

# 제 출 문

(재)한국게임산업개발원장 귀하

본 보고서를 『국제적 수준의 미래·수요 지향적 게임교육 프로그램 개발 연구』의 결과보고서로 제출합니다.

2006년 5월 31일

제출기관명 : (사) 한국 콘텐츠 학회

연구책임자 : 조용환 (충북대학교)

책임연구원 : 이창조 (우송대학교)  
이광재 (서남대학교)  
김영진 (청강문화산업대학)  
김미혜 (충북대학교)

공동연구원 : 장신환 (원광사이버대학)  
장창익 (세종사이버대학)  
오용선 (목원대학교)  
유석호 (공주대학교)  
최승권 (중부대학교)  
이준희 (세종사이버대학)  
이병록 (충북대학교)

연구조원 : 김송영 (충북대학교)  
조윤정 (충북대학교)  
최운수 (충북대학교)  
이준석 (충북대학교)  
홍성식 (충북대학교)  
남득기 (우송대학교)  
한득수 (전북대학교)  
최우영 (충북대학교)

국제적 수준의 미래·수요지향적 게임교육  
프로그램 개발 연구

2006. 5.

제 출 기 관 : 사단법인 한국콘텐츠학회

연구책임자 : 조 용 환

- 목 차 -

I. 서 론 .....	1
1.1 연구의 필요성 .....	1
1.1.1 수요지향적 게임 교육의 요구 .....	1
1.1.2 게임 산업체의 수요지향적 교육 강화 방안 요구 .....	3
1.1.3 대학의 게임 교육 강화를 위한 교육 로드맵 필요성 .....	5
1.1.4 본 연구의 독창성 .....	5
1.2 연구의 목표 .....	6
1.2.1 대학 게임교육 환경 개선안 제시 .....	6
1.2.2 게임교육 관련 대학의 교육 강화 .....	6
1.2.3 게임 교육계의 교육 강화 방안 제시 .....	7
1.2.4 수요지향적 게임 교육 강화 방안 구축 로드맵 제시 .....	8
1.3 연구개발 내용 및 범위 .....	9
1.3.1 대학 게임 교육 강화 방안의 목표 .....	10
1.3.2 현황진단 및 분석 .....	12
1.3.3 선진해외 교육프로그램 도입방안 .....	13
1.3.4 타당성검토 .....	15
1.3.5 마스터플랜 작성 .....	16
1.4 추진전략 및 방법 .....	18
1.4.1 연구 추진 전략 .....	18
1.4.2 연구방법 .....	18
1.4.3 연구 기대 성과 .....	18
1.4.4 활용 방안 .....	18

II. 게임 산업과 전문 인력 양성 .....	20
2.1 게임 산업의 특성과 현황 .....	20
2.1.1 세계 게임시장의 규모 및 전망 .....	20
2.1.2 세계 게임시장의 주요 트렌드 .....	20
2.2 게임 시장 전체 규모 및 동향 .....	24
2.3 플랫폼별 규모 .....	26
2.4 국내 게임 산업 주요 동향 .....	28
2.4.1 MMORPG와 캐주얼 온라인 게임 .....	28
2.4.2 게임 장르의 다양화 .....	31
2.4.3 해외 게임사의 국내 온라인 게임 시장 진출 .....	33
2.4.4 기업들 간의 인수 합병 .....	34
2.4.5 휴대용 게임기의 약진 .....	35
2.4.6 성인 아케이드 게임 시장 .....	36
2.5 게임기술 로드맵 .....	38
2.6 국내 게임 산업계 및 개발 업체들의 현황과 특성 .....	42
2.6.1 주력 개발 제품에 대한 설문 결과 .....	42
2.6.2 개발 계획 중인 장르에 대한 설문 결과 .....	44
2.6.3 개발 인력 채용에 관한 설문 결과 .....	47
2.6.4 대학과의 연계 계획에 대한 설문 결과 .....	50
2.7 국내 게임 산업 전문 인력 수요 분석 및 인력 수급 전망 .....	55
2.7.1 인력 수요 분석 .....	55
2.7.2 인력 수급 전망 .....	60
2.8 게임 산업 전문 인력 양성 관련 정책 .....	63
2.8.1 정부의 게임 인력 양성 계획 추진 현황 및 정책 방향 .....	64
2.8.2 핵심 전문인력 양성 강화 .....	66
2.8.3 연구기반 및 교육역량 강화 .....	75
2.8.4 실행형 산학협력 모델 구축 .....	79
2.8.5 취업환경 개선 및 내실화 .....	83
2.9 향후 정책의 방향 및 이에 대한 검토 .....	86

III. 게임산업 인력 양성 인식 및 사례 분석 .....	91
3.1 인력 양성에 대한 조사 .....	91
3.1.1 설문 내용 .....	91
3.1.2 설문조사 내용 분석 .....	97
3.2 게임 교육 교과 과정의 개선 .....	99
3.2.1 게임관련 학과 기존 교과 과정 .....	99
3.2.2 국내 게임 전문 인력 양성 기관의 사례 분석 .....	102
3.3 미국 게임 인력 기관의 사례 분석 .....	127
3.3.1 미국 게임 교육기관 .....	127
3.3.2 미국 교육기관의 교과과정 현황 .....	128
3.3.3 대학교별 교과과정 사례 .....	128
3.4 일본 게임 인력 기관의 사례 분석 .....	146
3.4.1 일본 게임산업의 특징과 게임교육 .....	146
3.4.2 일본 교육기관들의 특징 .....	147
3.4.3 전문 인력을 상세 구분한 교육 방식의 적용 .....	148
3.4.4 학원 및 사내 교육기관을 중심으로 본 교육 과정 분석 .....	150
3.4.5 정규 학위 대학과정에서의 게임 교육 .....	157

<b>IV. 게임 산업 전문 인력 양성을 위한 한국형 교육 시스템의 제안</b> .....	159
4.1 게임 산업 전문 인력 양성을 위한 한국형 교육시스템의 기본 방향 .....	159
4.1.1 국내 게임 교육 제도의 현실 .....	159
4.1.2 정규교육기관의 국내교육제도 .....	160
4.1.3 국내 게임 교육 시스템의 문제점 .....	164
4.1.4 선진해외 게임 교육프로그램 사례 .....	167
4.1.5 게임 산업 전문인력 양성을 위한 한국형 교육시스템의 필요성 .....	169
4.1.6 게임 산업 전문인력 양성을 위한 게임 교육개선방안 .....	173
4.2 게임 산업 전문인력 양성을 위한 교육시스템 .....	176
4.3 게임 산업 전문인력 양성을 위한 한국형 교육시스템 제안 .....	181
4.4 게임플레이어의 전문인력화 교육과정 제안 .....	183
4.4.1 고등학교 게임플레이어에 대한 교육과정 제안 .....	183
4.4.2 전문대학 이상의 게임플레이어에 대한 교육과정 제안 .....	184
4.4.3 한국형 게임교육시스템을 통한 인력양성 활용방안 .....	185
4.5 단계별 첨단 영상 게임 전문 인력 양성 사업 추진계획안 .....	187
<b>V. 결    론</b> .....	190
참고문헌 .....	196
부록. 카네기멜론대 ETC 교육 과정 분석 .....	202

# I. 서 론

## 1.1 연구의 필요성

### 1.1.1 수요 지향적 게임 교육의 요구

- 게임 산업계 필요수준의 게임 전문인력 양성 요구
  - 경쟁력 있는 차세대 게임 응용기술 개발 요구
  - 독창성(Originality) 및 문화 창의성(Creativity) 포함 요구
- ⇒ ● 글로벌 게임 대학교육시스템 필요  
● 열린 게임 대학교육시스템 필요  
● 온라인 게임 대학교육시스템 필요

#### 1) 산업계 수요지향적 게임 전문인력 양성 요구

현재 게임 관련 국내 정규 교육기관은 전문대학과 대학교를 중심으로 증가하다가 2005년에 들어서면서 원격대학과 전문대학을 중심으로 오히려 감소하고 있는 추세이다. 또한, 이들 정규 교육기관의 설립·운영의 역사는 매우 일천하고 선진 해외 게임교육에 대한 벤치마킹이 부족하여 이론 중심의 교육을 실시하는 등 “경쟁력을 갖춘 게임 전문 인력의 양성”을 필요로 하는 산업계의 요구에 부응하지 못하고 있는 실정이다. 이에, 본 □□대학의 수요 지향적 게임 교육 강화 방안에 대한 연구□□를 수립하여 다음과 같이 체계적인 대학 게임교육 방안을 모색할 필요가 있다.

- 게임의 속성(다학제적 컨버전스 특성)을 반영한 교육시스템 운영으로 게임 전문 인력 양성
- 국내 게임 산업계의 현실과 부합되는 대학교육시스템 운영(첨단 응용 기술)
- 게임 교육에서의 독창적이고 창의적인 교육 강화
- 미래지향적인 게임인력 양성(에듀테인먼트, 엔터테인먼트, 차세대 모바일, 온라인 등)

#### 2) 경쟁력 있는 차세대 게임 응용기술 개발 요구

게임 관련 기술은 기존의 IT 기술과는 달리 [표 I-1]에 제시된 것과

같은 다양한 기술들이 복합된 ‘컨버전스 기술’이며, 산업체와 대학이 협력하여 실제 상품화가 가능한 기술로 개발되어야 한다. 이에, 본 게임 교육 강화 방안에 대한 연구를 토대로 다음과 같이 응용기술 개발을 위한 체계적인 산학협력 방안을 위한 교육 강화 방안을 제시할 필요가 있다.

- 도전받는 온라인 게임의 위상 회복을 위하여 요구되는 차세대 게임 응용기술
- 선점한 온라인 게임 위주의 게임 응용기술에 고급기술 결합을 위한 인력 보강
- 게임 관련 기술에 더하여 창의력이 포함된 게임 응용기술
- 매체별, 장르별 게임 콘텐츠 응용기술

표 1-1. 게임 컨버전스 기술에 복합되는 다양한 세부 기술

대분류	세부기술명	특징	비고
CT	디지털 콘텐츠 핵심기반기술	가상현실, 인공지능, 디지털콘텐츠 저작도구기술, 게임엔진 제작 기반기술 등	디지털 미디어에 기반한 첨단 문화예술산업을 발전시키기 위한 기술을 총칭
	생활문화 디지털기술	가상의 정보공간을 매개체로 여러 사용자가 의사소통과 상호작용을 할 수 있는 사이버 커뮤니케이션 기술 등	
	문화재 복원 디지털기술	문화유적 탐사 발굴 원형추정과 복원 기술	
IT	차세대 네트워크 기반기술	차세대 이동통신, 대용량 광전송시스템, 초고속 인터넷 및 네트워크 기술	정보의 생성, 도출, 가공, 전송, 저장하는 모든 유통 과정에서 필요한 기술을 총칭
	정보통신 핵심부품기술	테라비트급 광통신부품, 고밀도 저장장치	
	정보처리시스템, SW핵심기술	멀티미디어 단말기, 전자상거래, 등	
ST	우주개발을 위한 위성핵심기술	통신, 기상, 지구탐사, 위성설계 및 개발 등	위성체, 발사체, 항공기 등에 관련된 복합기술. 기술 개발 결과가 타 분야에 대한 파급효과가 큰 종합기술을 총칭
	발사체 핵심기술	로켓 추진기관 기술, 소형 위성발사체 기술	
	차세대 항공기술	스마트무인기, 다목적 헬리콥터, 항공기 체계 종합 및 비행성능 기반 기술 등	
NT	나노소자 및 시스템 핵심 원천기술	나노전자소자, 나노정보저장기술, 가변파장 광소자 등	물질을 원자·분자 크기의 수준(10 <sup>-9</sup> nm)에서 조작·분석하고 제어할 수 있는 과학 기술을 총칭
	나노바이오 보건의술	나노분말소재, 나노바이오 물질합성 및 분석 기술, 의약 약물 전달시스템 등	
	핵심 나노기반 공정기술	나노측정, 나노화학공정, 나노소재 기술 등	
BT	생명공학 기초 기반기술	유전체 단백질체 연구, 생물정보학(Bioinformatics), 생물공정기술, 생명공학 안전성 평가기술 등	생체나 그 유래물질 또는 생물학적 시스템을 이용하여 제품을 제조하거나 그 공정을 개선하는 기술을 총칭
	첨단 농업·해양 생명공학기술	바이오 신약, 생체조직재생기술 등 보건의료 생명공학 핵심기술 개발과 유전자변형생물체기술 등	

### 3) 독창성(Originality) 및 문화 창의성(Creativity) 포함 요구



게임 산업은 새로운 아이디어와 문화적 수준을 요구하며 지속적으로 발전하는 '컨버전스 기술'이다. 따라서 게임 산업계는 독창성과 문화 창의성을 갖춘 게임 전문인력의 양성을 끊임없이 요구하고 있다. 이에 대응하여 대학에서는 다음과 같은 국제적 수준의 독창성과 문화 창의성 증진 방안을 모색할 필요가 있다.

- 게임 콘텐츠의 국제 경쟁력과 문화 독창성(차별화) 증진
- 지식·정보 문화가 창의적으로 결합된 고부가가치 산업 활성화
- 기반기술 및 기존기술에 게임 응용기술을 접목한 연계 활용
- 문화기반에 공학, 아트, 인문학이 복합된 컨버전스 문화상품 개발

#### 1.1.2 게임 산업체의 수요지향적 교육 강화 방안 요구

- 주입식 교육에서 비선형(Non-Linear) 창의성 교육으로 전환
  - 선진 게임교육 환경 및 과정 분석 (세계 게임 3강 벤치마킹: 미국, 일본, 유럽)
  - 게임 장르별, 교육기관별 산학협력 교육프로그램 제안
- ⇒ ● 글로벌 게임 대학교육시스템 필요  
● 열린 게임 대학교육시스템 필요  
● 온라인 게임 대학교육시스템 필요

#### 1) 주입식 교육에서 비선형(Non-Linear) 교육으로 전환

게임 기술은 새로운 아이디어와 독창성을 요구하는 컨버전스 기술로 게임 교육계는 전문인력의 창의성 증진을 끊임없이 요구하고 있다. 이에, 대학에서는 수요지향적 게임 교육 강화 방안을 제시하여 기존의 이론중심 교육을 탈피하고 비선형 파이프라인 형식의 교육을 실시함으로써 독창성 증진 방안을 모색할 필요가 있다.

- 세계화, 정보화시대, 다양한 디지털 지식기반의 게임 콘텐츠 경쟁시대 도래

- 사회 구성원 개별의 창의력에 의거하여 해결되는 문제가 날로 증가하고 있음
- 고급 인적자원은 지속적이고 비선형적인(Non-Linear) 실무 훈련과정을 통하여 개발됨
- 게임산업의 혁신적인 발전을 위한 기반은 게임 전문인력의 독창성에 의존함

### 2) 선진 게임교육 환경 및 과정 분석

국내 게임 정규 교육기관의 일천한 역사와 이론중심 교육을 탈피하기 위해서는 선진 게임 3강인 미국, 일본, 유럽 등의 정규 교육기관이 표방하는 실무중심 교육을 도입하여 전면적으로 개선할 필요가 있다. 이에, 선진 게임 교육기관에 대한 벤치마킹을 실시하여 실무중심 교육의 올바른 방향을 제시할 필요가 있다.

- 게임을 종합예술로 보고 다학제적이고 실무중심적인 교육
- 창의력에 의하여 무궁한 자원을 만들어낼 수 있다고 판단하는 선진 교육
- 산업계를 선도하고 있는 해외 선진국의 게임 관련 교육에 대한 세밀한 벤치마킹
- 선진 교육방식과 프로그램의 장단점을 분석하여 국내에 적용할 수 있는 모델 제시

### 3) 게임 장르별, 교육기관별 산학협력 교육프로그램 제안

	분야	산학협력내용	주요프로그램	비고
장르	온라인	온라인 게임 개발업체와의 정보 공유와 산학을 통한 실무교육 및 맞춤형인력 양성	온라인 게임개발 실무교육 및 인재양성	
	PC	PC 게임 개발업체와의 정보 공유와 산학을 통한 실무교육 및 맞춤형인력 양성	PC 게임개발 실무교육 및 인재양성	
	콘솔	콘솔 게임 개발업체와의 정보 공유와 산학을 통한 실무교육 및 맞춤형인력 양성	콘솔 게임개발 실무교육 및 인재양성	
	아케이드	아케이드 게임 개발업체와의 정보 공유와 산학을 통한 실무교육 및 맞춤형인력 양성	아케이드 게임개발 실무교육 및 인재양성	
기관	단기학원	게임교육학원과의 교육연계	개발인력 양성	
	대학	게임관련학과와의 게임교육 협력	실무교육 및 인재양성	
	대학원	게임관련 대학원을 통한 고급인력양성	게임연구 인력양성	
	전문가 그룹	게임개발 전문가들과의 교육방안 협력	전문가 그룹양성	

### 1.1.3 대학의 게임 교육 강화를 위한 교육 로드맵 필요성

- 창의성은 게임의 다양성과 품질을 향상시키는 가장 중요한 요인
  - 미진한 분야의 기술을 획기적으로 향상시키기 위한 실무형 고급인력 양성
  - 차세대 국가 성장 원동력으로서의 게임산업을 주도할 고급인력의 양성
- ⇒ ● **글로벌 게임 대학교육시스템 필요**  
● **열린 게임 대학교육시스템 필요**  
● **온라인 게임 대학교육시스템 필요**

보다 높은 수준의 수요 지향적 게임 교육 강화 방안은 게임 산업을 위한 고급 인력 양성에 결정적인 역할을 수행할 것이다. 단순 기술자만을 양성하는 교육은 산업계의 요구를 충족시키지 못하며, 궁극적으로 외국의 기술에 종속되는 결과를 가져올 수 있다. 종합예술인 게임에서의 창의력은 게임의 다양성과 게임의 질을 향상시키는 가장 중요한 요인이며, 이는 곧 부가가치 창출로 이어진다. 우리가 점유하고 있는 선두분야 기술을 지속적으로 발전시키고 아직 미진한 분야의 기술을 획기적으로 향상시키기 위해서는 차세대 게임 산업을 주도할 경쟁력 있는 고급인력이 반드시 필요하다.

이러한 목표를 달성하기 위해서는 본 연구의 단기목표에서 얻어진 연구결과를 바탕으로 전국 주요 게임 관련 대학의 학과에 제시하고, 이들을 연계하는 네트워크를 형성함으로써 차세대 국가 원동력으로 작용할 수 있는 수요 지향적 대학 게임교육시스템을 구축하여야 한다. 이를 위해서는 본 연구를 수행함에 있어 구체적이고 체계적인 교육 로드맵이 필요하다.

### 1.1.4 본 연구의 독창성

본 연구는 게임산업 분야의 경쟁력을 갖춘 개발인력의 양성과 그 활용의 수준을 높이기 위하여, 창의성의 개념을 포괄적으로 확대하여 이를 실현할 수 있는 대학 교육 시스템 로드맵을 제시하고, 단계적으로 이들을 전국에 확산하여 산학 네트워크를 통하여 상호 연계함으로써 공동활용 및 연계체제를 구축함을 주된 목적으로 설정한다.

상기 국내외 관련 선행연구에서 보는 바와 같이, 기존의 연구들은 단위대학, 학과 또는 교수개인의 교수법을 주된 대상으로 삼고 있으나, 본 연구는 대학간 연계 뿐 아니라 단기학원, 학부, 대학원 나아가 산업체 및 전문가(팀장급 이상) 그룹을 포함하는 포괄적이고 전 방위적인 연구라는 점

에서 일차적인 차별성을 갖는다. 본 연구가 지향하는 산학 간 연계체제의 구축은 국내 게임산업 발전 및 고급인력 양성을 위한 산학 연계 네트워크 구축으로 발전하며, 게임 장르별, 교육기관별, 산업체와의 연계성 등 세 가지 측면을 고려하고 있다.

## 1.2 연구의 목표

- 대학 게임교육 환경 개선안 제시
- 게임교육 관련 대학의 역량 강화
- 게임 교육계의 개선방안 제시
- 대학의 게임 교육 강화를 위한 교육 시스템 로드맵 제시

게임 교육기관에서 교육중인 교육과정을 파악하고, 특히 대학 게임 교육 환경의 개선안 및 관련 대학의 역량 강화를 위한 몇 가지 방법론을 제시한다. 게임 교육계의 교육 강화 방안으로 그에 따른 로드맵을 제시하고자 한다.

### 1.2.1 대학 게임교육 환경 개선안 제시

- 글로벌 게임 대학교육시스템
  - 열린 게임 대학교육시스템
  - 온라인 게임 대학교육시스템
- ⇒ 교육과정 개선 제안

### 1.2.2 게임교육 관련 대학의 교육 강화

- 산학 협력체계 구축
- 산학 공동연구(Project) 및 교육 수행

#### 1) 산학 협력체계 구축

- 연계체제 구축 : 대학 ⇔ 대학, 대학 ⇔ 산업체, 대학 ⇔ 연구소 간 연계체제 구축
- 대학 게임 교육과정 제안, 교육여건 제안, 게임콘텐츠의 분석(테스트)및 결과 제안

- 게임 제작과정(파이프라인)별 창의성 점검시스템 제안

## 2) 공동연구(Project) 수행

- 산학협력 공동 프로젝트의 중심 : 게임콘텐츠의 창의성 및 실용성 배가, 기술 및 설계 자문, 기술이전 및 기술지도
- 산학 공동연구의 중심 : 게임 개발성 증진, 분야별·장르별, 기타 종합적인 측면의 공동수행
- 산학협력 교육활동의 중심 : 인턴쉽(Internship), 캡스톤 설계 (Capstone-Design), 현장실습 및 견학 기획 및 실행

### 1.2.3 게임 교육계의 교육 강화 방안 제시

- 글로벌 게임 대학교육시스템
- 열린 게임 대학교육시스템
- 온라인 게임 대학교육시스템

본 연구에서 지향하는 대학 교육 강화 과정은 지역화와 국제화를 선도하는 학제적 지식과 전문성을 갖춘 디지털시대의 정보화를 선도하는 게임 전문인력 양성을 목적으로 하고 있다. 이를 구체적으로 실현하기 위해서 다음과 같은 특성화 교육체계를 수립하고자 한다.

#### 1) 글로벌 게임 대학 교육시스템

국제교류 프로그램을 기반으로 국제적인 감각을 소유하고 사회적 요구에 부응하는 게임전문인력 양성을 위한 대학 게임 교육시스템 : 교환학생제도, 교환교수제도, 외국인교수의 정규강좌 개설, 국제 게임교류전 등

#### 2) 열린 게임 대학 교육시스템

급변하는 사회변화와 요구에 따른 경쟁력과 전문성을 바탕으로 다양한 분야에 대한 이해력 증진을 위한 게임 대학 교육시스템 : 연계전공, 심화과정, 부전공, 다전공, 사제동행세미나 등

#### 3) 온라인 게임 대학 교육시스템

디지털시대의 게임전문 인재양성을 위하여 온라인으로 통합된 대학 게임교육을 실현하고, 전문적이고 실용적인 네트워크 교육체계를 수립함 : 사이버강좌, 교수의 홈페이지 활용, 인터넷 교육 등 인터랙션 방식의 원격·가상교육 체재구축 등

#### 1.2.4 수요지향적 게임 교육 강화 방안 구축 로드맵 제시

본 연구에서는 전국의 주요 포스트로 확산시켜 연계 네트워크를 형성함으로써 대학 게임 전문인력 양성을 위한 망 구성을 목표로 한다. 본 연구에서는 이를 위한 업무 로드맵을 제시하고자 하는 것이다.

- 마스터플래닝의 주요 과정
  - ① 한국콘텐츠학회를 중심으로 하는 대학 게임 교육 강화 방안 연구를 통하여, 국내 게임 교육기관의 교육 프로그램을 개선하는 방안 제시
    - 전국적 확산 및 네트워크 구축 방안 도출
    - 실무형, 산업형 게임 교육 프로그램 개발 및 보급
    - 인턴쉽(Internship), 캡스톤설계(Capstone-Design), 현장실습 및 견학 프로그램 개발 및 보급
  - ② 수요지향적 대학 게임 교육 강화 방안 시스템의 보급
    - 한국콘텐츠학회 학술위원회 게임콘텐츠분과, 엔터테인먼트분과, 애니메이션콘텐츠 분과, IT기반기술 분과의 위원장과 핵심 분과위원들로 구성되는 운영위원회 구성
    - 선진 사례 분석을 통한 교육 과정 개념 정립
    - 설문조사를 통한 학계, 연구계, 산업계, 관계 의견 수렴
  - ③ 지원 정책 및 전략 도출
    - 그룹별 맞춤형 교육과정과 교육환경의 개발 및 보급 (인턴쉽, 캡스톤설계, 현장 적응형 교육 프로그램, 창의적 게임 저

작을 위한 실험 실습실 환경 정의)

- 산학협력을 통한 공동 프로젝트의 기획 및 실행
- 산업계와 학계를 연계할 수 있는 구심점 제공
- 산학 연계 네트워크 구축 및 운영

### 1.3 연구개발 내용 및 범위

대학의 게임 교육 강화 방안						
연구방향	<ul style="list-style-type: none"> <li>■ 게임 교육 강화 방안 연구</li> <li>■ 교육과정 개선안 작성</li> <li>■ 산학 협력 구축 방안 마련</li> <li>■ 로드맵 작성</li> </ul>					
연구개발 주요방향	교육과정 개념정립	현황진단 및 분석	선진 해외 교육 프로그램 도입방안	선진 해외 교육 프로그램 도입 실행 가능성 검토	마스터플랜 제시	로드맵 구성
추진 전략	글로벌 게임 대학교육 시스템 구축		열린 게임 대학교육 시스템 구축		온라인 게임 대학교육 시스템 구축	
연구 목적	게임교육 환경 개선안 제시	게임 관련 대학의 역량강화	게임 교육계의 개선방안 제시	교육 강화시스템 로드맵 제시		
연구의 필요성	<b>1. 게임 산업계 요구</b> -경쟁력있는 고급 인력 양성 요구 -차세대 게임 응용기술 -독창성 및 창의성 요구	<b>2. 게임 교육계의 요구</b> -비선형 창의교육 -선진 게임교육 분석 -산학협력 프로그램	<b>3. 연구 필요성</b> -게임콘텐츠 품질향상 -고급 인력 양성 -차세대 국가원동력	<b>4. 본 과제의 독창성</b> -포괄적 전 방위 연구 -유관기관간 연계방안 -다면적 접근		

### 1.3.1 대학 게임 교육 강화 방안의 목표

- 해외 선진기관에서 진행되는 창의적 게임교육
- 게임교육에서 창의성이 필요한 영역
  - ⇒ 게임 교육과정 개선
    - 1) 선진프로그램 벤치마킹
    - 2) 학제간 융합 교육 프로그램

#### 1) 해외 선진기관에서 진행되는 게임교육

현재까지 우리나라 게임 교육은 게임 그래픽과 게임 프로그래밍 교육에 치중하여 온 것이 사실이다. 게임 개발의 후발 국가로서 선진국의 기술 수준에 도달하기 위해서는 이 두 분야의 교육을 먼저 실시한 것은 온라인 게임 등에서 어느 정도 효과를 보고 있다고 할 수 있다. 그러나 창의력을 바탕으로 게임을 기획하고 독자적으로 개발하지 않는 한 게임 하청공장으로로서의 역할 이상은 기대할 수 없다. 선진국들은 게임산업이 미래 국가발전의 주요 전략으로 자리잡게 될 것을 간파하고, [표 1-2]와 같이 체계적이고 종합적인 교육센터를 설립·운영하는 등 막대한 투자를 아끼지 않고 있다. 이러한 교육센터들은 실질적이면서 다양한 프로그램을 운영하며 창의력 배양과 향상을 목표로 양질의 교육을 실시하고 있다.

표 1-2. 해외 선진 게임 관련 정규 교육기관 현황(벤치마킹 대상)

구분	대학명	장점	특화내용	비고
미국	디지펜대학	<ul style="list-style-type: none"> <li>▶물리와 수학 등 기초과학을 포함한 다양한 학문을 습득하도록 하여 학사학위를 수여함.</li> <li>▶게임전문대학으로 전문가 양성이 목표</li> <li>▶HW 등의 서킷보드를 조립하여 컴퓨터 내부동작을 이해하는데 교육의 초점</li> <li>▶실전위주 게임프로그램 교과목 강조</li> </ul>	▶미국 유일의 4년제 컴퓨터대학교로 설립됨.	물리와 수학 등 기초과학을 바탕으로 한 창의성 교육
	카네기멜론대학 (ETC)	<ul style="list-style-type: none"> <li>▶다양한 전공의 학부생을 석사과정으로 유도</li> <li>▶팀 프로젝트 베이스의 결과물 도출과 대규모 평가회</li> <li>▶다수의 교육센터 순회연수 프로그램</li> <li>▶게임 산업체와 연계된 인턴십 운영</li> <li>▶게임관련 다양한 분야의 강사진 확보</li> </ul>	<ul style="list-style-type: none"> <li>▶대학 내 독립적인 ETC 운영 (Entertainment Technology Center)</li> <li>▶석사과정 교역의 등록금 수수</li> <li>▶게임관련 산업체와 고정적인 인력 공급관계 유지</li> <li>▶단기 연수 프로그램 운영</li> <li>▶세계 각지에 부설 교육센터 설치 운영</li> </ul>	다학제간의 교류를 통한 창의성 연계
	유켄대학	<ul style="list-style-type: none"> <li>▶게임산업에 종사하는 산업인력 훈련에 교육의 목표를 둠</li> <li>▶애니메이션, 그래픽 등의 최신의 컴퓨터 게임 디자인 기술을 교육.</li> <li>▶학생들의 적성에 따라 게임감독자, 게임설계자, 게임프로그래머 등의 훈련을 쌓을 수 있음.</li> </ul>	▶오랜 역사를 바탕으로 축적된 기술의 넓은 활용	적성을 살린 교육

일본	오사카 전기통신대학	▶전자/정보/통신을 바탕으로 한 교육인프라 ▶대학 및 대학원의 다양한 교육	▶디지털 게임 학부 및 대학원 과정 신설	신설학과를 통한 창의성 연계
	일본공학원전문학교	▶게임소프트/CG/디지털크리에이티브 코스 등 멀티미디어과를 통한 전문교육 가능	▶2D/3D 네트워크 게임의 작품제작 중심의 교육 ▶실무에서와 같은 기자재 및 학습환경	실무와 같은 학습
	일본전자 전문학교	▶IT, CG, 게임, 애니메이션의 분야로 현재 분야별 다양한 학년제로 교육 및 운영	▶3DCG 제작을 초보에서부터 전문가까지 배울 수 있는 과정 ▶업계가 필요한 게임개발자 육성 과정	전공분야별 다 학년제를 통한 교육
유럽	라벤스번 대학	분석중	분석중	인문학을 기초로 과학과 예술이 결합된 교육
	헐대학	▶런던대학의 분교로 시작 ▶기초학문 및 응용학문 특히 컴퓨터 관련 학문이 우세	▶컴퓨터과학, 전자공학 및 로봇학 텔레통신과학, 공학디자인 등 공학 및 네트워크가 강하다	기초학문을 바탕으로 한 응용학문

## 2) 대학 게임교육에서 필요한 영역

콘텐츠 분류	유형	특징	비고
시각 콘텐츠	<ul style="list-style-type: none"> <li>정지화상</li> <li>애니메이션</li> <li>디지털 비디오</li> <li>비주얼 아트워크</li> <li>게임그래픽 아트워크</li> </ul>	<ul style="list-style-type: none"> <li>직접적이다</li> <li>이해하기 쉽다</li> </ul>	<ul style="list-style-type: none"> <li>시각 디자인</li> <li>객체 표현</li> <li>게임 인터페이스</li> </ul>
청각 콘텐츠 (음악 콘텐츠 포함)	<ul style="list-style-type: none"> <li>게임 캐릭터의 음성</li> <li>음향효과</li> <li>전자음악</li> <li>녹음음악</li> <li>알고리즘 음악</li> </ul>	<ul style="list-style-type: none"> <li>쉬운 전달</li> <li>감성적이다</li> </ul>	<ul style="list-style-type: none"> <li>감성표현의 음악화</li> <li>음악표현의 창의성</li> </ul>
텍스트 콘텐츠	<ul style="list-style-type: none"> <li>대본화된 대화</li> <li>이야기 텍스트</li> <li>정보제공 텍스트</li> </ul>	<ul style="list-style-type: none"> <li>간결하다</li> <li>직접적이다</li> </ul>	<ul style="list-style-type: none"> <li>스토리</li> <li>레벨 디자인</li> <li>상황표현</li> </ul>
촉각 콘텐츠	<ul style="list-style-type: none"> <li>게임 컨트롤러</li> </ul>	<ul style="list-style-type: none"> <li>물리적 접촉</li> </ul>	<ul style="list-style-type: none"> <li>사실감증대</li> <li>현실과 연계 창의</li> </ul>
통합콘텐츠	<ul style="list-style-type: none"> <li>시청각 통합 콘텐츠</li> </ul>	<ul style="list-style-type: none"> <li>콘텐츠의 통합</li> </ul>	<ul style="list-style-type: none"> <li>사용자편리</li> <li>몰입감증대</li> </ul>

- 정부와 산업체간의 전문인력 양성 및 수급의 문제점
- 교육기관의 교육과 산업체에서 요구하는 수준과의 차이
- 게임은 기존 IT기술인력의 배출과정과 다름을 인정해야 한다.
- 국내 게임 교육기관의 교육과정에서의 가장 큰 문제점은 수요  
지향적 교육의 결여이다.
- 게임산업체에서 원하는 고급 인력을 양성한다.

- 게임 교육기관이 배출한 인력과 산업체가 요구하는 인력 간의  
수준차

- 게임은 과학과 예술이 만나는 종합예술의 관점에서 이해: 교육의 커리큘럼
- 교육 수혜자의 지적능력, 특기, 진로의 희망, 적응성 등 개인별 사안을 종합 판단
- 적정 수준의 교육이 실습, 실무와 함께 실시
- 특별강연과 세미나들은 관련 부분 전문가를 초빙하여 실시: 체계적이고 심도있는 교육의 질을 기대

### 1.3.2 현황진단 및 분석

- 문제점 분석
- 전망 제시
  - 국내외 교육기관의 향후 변화 추세 전망

#### 1) 문제점 분석

- 국내 게임 교육의 짧은 역사로 다학제적인 교육과정 미비
- 게임에 대한 인식의 부족으로 정규교육과정에 관련교과 부족
- 게임 교육 인력의 부족으로 창의성관련 저변부족
- 산학연의 유기적인 결합 부족으로 종합적이고 실무적인 교과 미개설

#### 2) 전망 제시

단순한 강의의 연속과 개별적이고 부분적이며 게임 각각의 부분이 유기적인 관계없이 진행되는 교육은 이제 더 이상 지속되어서는 시대의 흐름에 동참할 수 없다. 기존 교육기관들은 이러한 패러다임의 변화에 적응하지 못하면 교육생들이 외면할 것이고 자연적으로 도태될 수밖에 없다. 게임 교육시스템과 방법론의 변화는 이미 시작되었으며 바로 지금이 시행에 옮겨야 하는 적기이다.

- 게임은 철저한 TEAMWORK의 산물이다.
- 국내 게임 교육은 기술적인 측면에만 주력하고 있어 문제점이 많다.
- 특정 분야에만 치중한 게임 교육기관으로 인한 교육생들의 선

택 폭이 좁아짐으로 인한 문제

- 진취적인 교육시스템을 구성하기 위한 노력이 필요하다
- 프로젝트 진행을 통한 교육의 필요성
- 게임 교육시스템과 방법론의 변화는 지금 시작되고 있고 바로 실행해야할 과제이다.

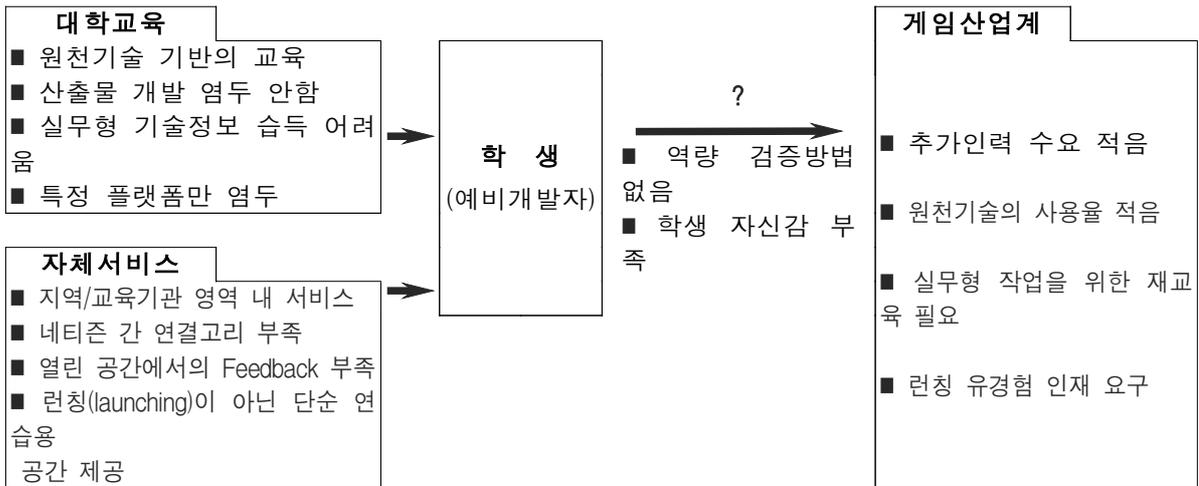


그림 1-1. 현재 게임교육기관의 애로

### 1.3.3 선진해외교육프로그램도입방안

- 문화콘텐츠 관련 전 분야의 해외 우수 교육 프로그램 비교 검토
  - 미국 디지펜 대학, 카네기멜론 대학, 유펜 대학, 영국 라벤스번 대학, 헐 대학 등을 포함한 세계 각국의 우수 게임 교육의 장점과 특화 내용 검토
- 벤치 마킹을 통해 국내 적용 가능한 모델 제시
- 도입 일정, 절차를 포함한 사전 검토 내용 제시
- 기존 국내 교육기관의 체계 및 내용 분석
- 문제점 분석
  - 국내 게임 교육기관의 문제점 및 원인 분석
- 전망 제시
  - 국내외 교육기관의 향후 변화 추세 전망

#### 1) 문화콘텐츠 관련 전 분야의 해외 우수 교육 프로그램 비교 검토

국내 게임 정규 교육기관의 일천한 역사와 이론중심 교육을 탈피하기 위해서는 선진 게임 3강인 미국, 일본, 유럽 등의 정규 교육기관이 표방하는 실무중심교육에 대한 벤치마킹 필요하다. 이에, 본 연구에서는 선진

교육기관에 대한 벤치마킹을 실시할 필요가 있다.

- 게임을 종합 예술로 보고 다학제적인 교육과정 개선에 전력
- 창의력으로 무궁한 자원을 만들어낼 수 있다고 판단
- 해외 선진국의 게임 관련 교육은 산업계를 선도
- 교육방식과 프로그램의 장단점을 분석하여 우리의 것으로 적용할 수 있는 모델을 제시

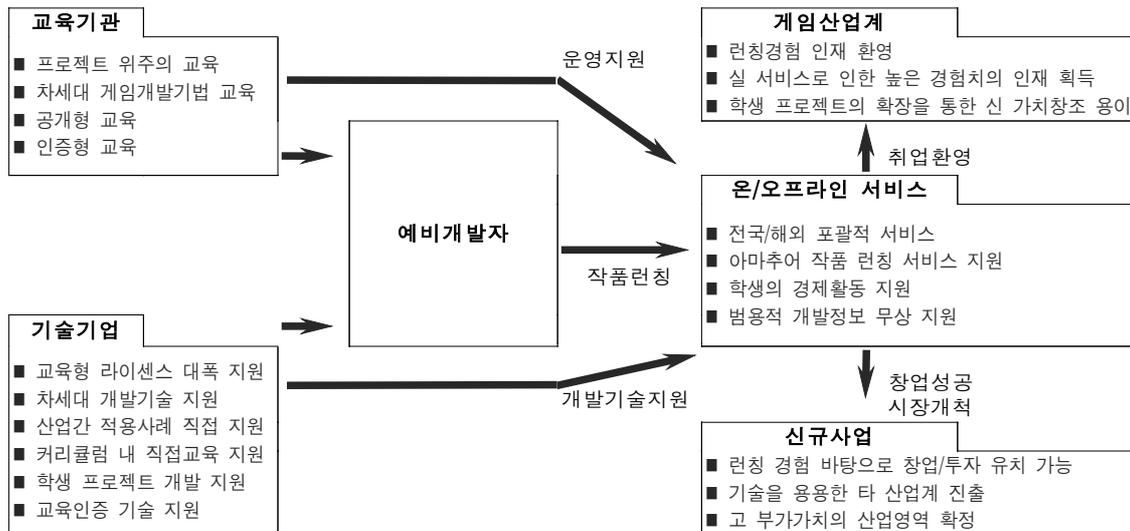


그림 1-2. 산업에 필요한 게임 교육혁신지점 - 프로젝트 중심 현장형 교육체제로 전환

## 2) 문제점 분석

- 국내 게임 교육의 짧은 역사로 다학제적인 교육과정 미비
- 게임에 대한 인식의 부족으로 정규교육과정에 창의성 관련교과 부족
- 산학연의 유기적인 결합 부족으로 종합적이고 실무적인 교과 미 개설

## 3) 전망 제시

- 대학, 전문대학, 정부산하 교육기관 및 민간 교육기관 모두 게임 개발의 기술적인 능력 함양
- 외국 교육모델을 벤치마킹하여 시스템을 구성
- 게임 교육 커리큘럼을 다양하게 하면서도 자율성을 부여하여 깊

- 이 있는 개발 능력을 함양
- 개발 전 게임콘텐츠의 관련 프로그램 분석 후 결과를 반영한 높은 수준의 결과물로 산출되어야 한다.

#### 1.3.4 타당성검토

- 한국 대학 게임교육계에서의 실현 가능성 여부
- 사업추진의 성과

#### 1) 대학 게임 교육강화 교육 시스템의 실현 가능성 제고

##### A. 성공 가능성 분석

- a. 일천한 국내 게임 교육의 역사와 상대적으로 급속한 발전을 이룩한 국내 게임 산업의 현장에는 산업계 요구 수준의 고급 인력 절대 부족
- b. 게임관련 교육기관의 증가로 배출 인력이 증가하고 있으나 기존 IT 기술 개발 수준의 교육으로 창의력이 요구되는 인력 부재
- c. 세계 수준의 게임 개발 프로그램 운영으로 타 교육기관과 차별화
- d. 각 교육기관 교육자들의 게임관련 고급 교육 요구 증가
- e. 동아시아 게임 산업의 발전으로 게임 산업 인력 수요 증가

##### B. 개설을 위한 예비 전략

- a. 최근 선진국의 국내 교육센터 진출 동향 예의 주시
- b. 외국 교육센터의 국내 개설의 장단점 철저 분석, 검토
- c. 외국 교육센터 벤치마킹의 결과를 홍보에 연계
- d. 산업계 미발표 게임콘텐츠에 창의성 증진 요소 제시, 결과물 산출

#### 2) 사업추진의 성과

- a. 국내 게임 산업의 지속적인 발전을 위한 인력 공급라인 구축
- b. 외국 교육센터 유학에 준하는 저비용 고급 교육 기회 부여
- c. 국내 교육 기관의 교육자 프로그램 운영으로 국내 게임 교육의

업그레이드 효과

- d. 산학협동으로 필요 인력의 적재적소에 공급
- e. 산학 협동이 실질적인 게임 결과물에 창의성 증진 효과
- f. 국내외 다양한 프로그램 운영으로 향후 재정 독립과 이익 창출

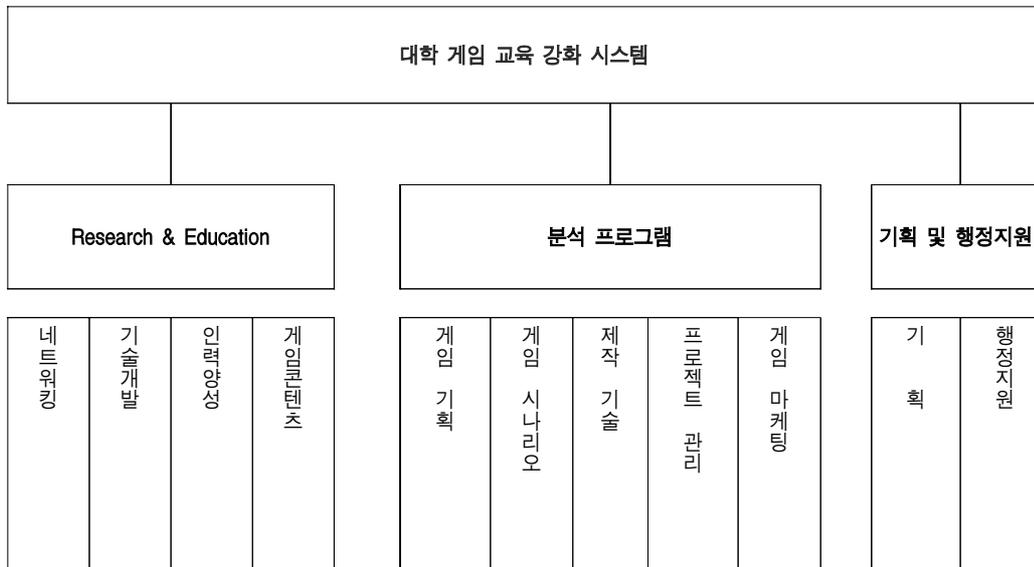
1.3.5 마스터플랜 작성

- 국내 게임 교육기관에 표준 교육 시스템으로 연계하는 방안
- 교육 실행 방안
- 관련 지원 정책 및 전략 도출

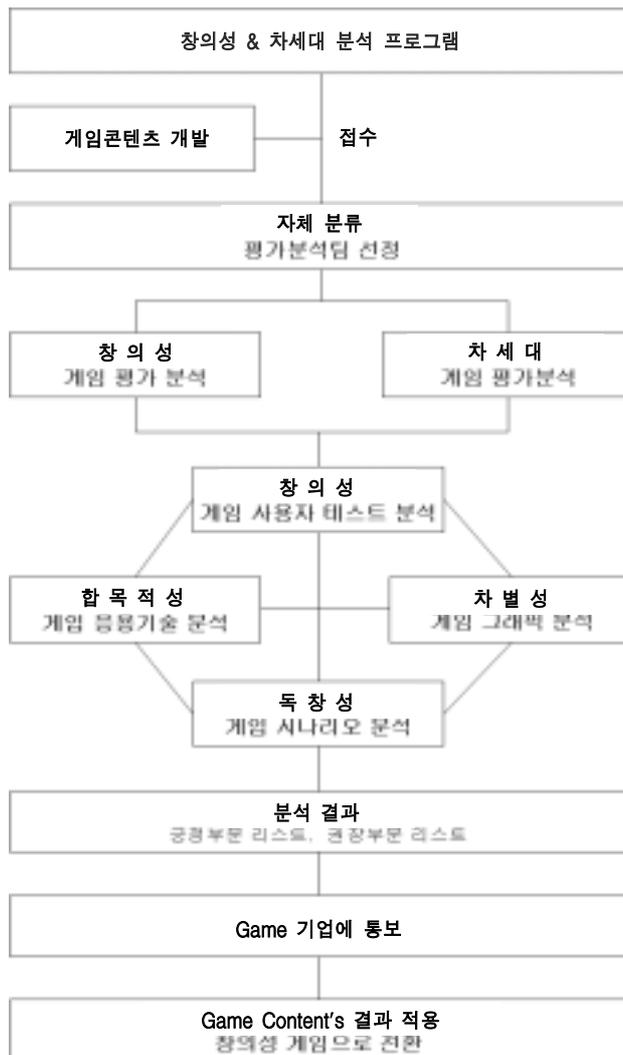
1) 연계방안

- 산업체 <----> 게임 교육계 : 실무형 교육과정 및 교육 여건 개선안, 기술개발, 인력양성, 네트워킹
- 게임교육계 <----> 게임 산업계 : 게임콘텐츠의 경쟁력 테스트, 기술개발, 인력양성, 네트워킹 등

2) 교육 강화 시스템 구축 실행 방안



<p>&lt; Research &amp; Education &gt;</p> <ul style="list-style-type: none"> <li>● 네트워킹 <ul style="list-style-type: none"> <li>- 기반 장비 구축</li> <li>- 관련연구 및 기술 DB화</li> <li>- 회원사 System</li> <li>- Homepage 운영 강화</li> </ul> </li> <li>● 기술개발 <ul style="list-style-type: none"> <li>- 3D 그래픽</li> <li>- 가상현실</li> <li>- 네트워크</li> </ul> </li> <li>● 인력양성 <ul style="list-style-type: none"> <li>- 학생 중심 교과과정 개선</li> <li>- 재교육 단기교육 과정</li> <li>- 전문가 (임원급)</li> <li>- 대학원생</li> <li>- 학원 신설</li> </ul> </li> <li>● 게임콘텐츠 <ul style="list-style-type: none"> <li>- 텍스트 콘텐츠</li> <li>- 시각 콘텐츠</li> <li>- 청각적 콘텐츠</li> <li>- 촉각적 콘텐츠</li> </ul> </li> </ul>	<p>&lt;분석 프로그램 &gt;</p> <ul style="list-style-type: none"> <li>● 게임 기획 <ul style="list-style-type: none"> <li>- 게임 제작 전반의 아이디어</li> <li>- 무에서 유를 창조하는 작업</li> <li>- 다학제적인 교육과정</li> </ul> </li> <li>● 게임 시나리오 <ul style="list-style-type: none"> <li>- 게임 진행의 스토리</li> <li>- 수준 높은 창의력이 요구되는 분야</li> <li>- 게임에 몰입하게 하는 창조적인 에너지 역할</li> </ul> </li> <li>● 제작 기술 <ul style="list-style-type: none"> <li>- 그래픽 기술 : 비주얼 객체를 만드는 기술</li> <li>- 음향 기술 : 게임 사운드의 제작</li> <li>- 프로그래밍 : 게임의 진행을 위한 소프트웨어 제작</li> <li>- 하드웨어 기술 : 게임이 이루어지는 플랫폼 제작</li> </ul> </li> <li>● 프로젝트 관리 <ul style="list-style-type: none"> <li>- 게임 제작의 완성을 위한 진행 관리</li> <li>- 철저한 창의적인 게임의 완성</li> <li>- 제작 단계별 최적의 진행 스케줄 관리</li> </ul> </li> <li>● 게임 마케팅 <ul style="list-style-type: none"> <li>- 게임의 홍보</li> <li>- 게임의 보급</li> <li>- 부가가치 창출</li> </ul> </li> </ul>
---	--



## 1.4 추진전략 및 방법

### 1.4.1 연구 추진 전략

게임교육계, 산업계에 실효적인 게임 창의성 증진을 위한 게임 창의성 증진 연구는 3가지 ①글로벌 게임 대학 교육시스템 ②열린 게임 대학 교육시스템, ③온라인 게임 대학 교육시스템을 추진 컨셉으로 설정 추진한다.

#### ● 글로벌 게임 대학 교육시스템

- 국내외 게임 연구자와의 협동연구체계 구성
- 교육내용의 국제적 통용 가능한 교육 과정 구성

#### ● 열린 게임 대학 교육시스템

- 창의적인 게임 교육시스템
- 연계전공, 심화과정, 부전공, 다전공, 사제동행세미나

#### ● 온라인 게임 대학 교육시스템

- 실용적인 네트워크 교육체계를 수립함 :
- 사이버강좌, 교수의 홈페이지 활용, 인터넷 교육 등 인터랙션 방식의 원격·가상교육 체제구축

### 1.4.2 연구방법

- 설문대상그룹 : 학생, 개발자, 기업 임원급, 교수의 4개군으로 분류
- 소 과제 연구관리 및 연계 : PD(프로그램 디렉터)제의 도입

### 1.4.3 연구 기대 성과

- 1) 게임에서의 교육 강화 및 산학 연계 중요성 제고
- 2) 게임 교육 강화 방법론 제시
- 3) 국내외 교육 모델 비교 분석

### 1.4.4 활용 방안

- 1) 국내 대학 게임 교육 시스템의 보급 및 운영을 위한 참고 자료로 활용
  - A. 국내 보급을 위한 고려사항 참조
  - B. 단기, 장기 프로그램 구성방안 참조
  - C. 2년제 석사 학위과정 프로그램 운영 방안 참조
- 2) 국내 대학 교육 시스템의 확장과 수익 창출을 위한 참고 자료로

활용

A. 산학연 연계 프로젝트 발굴 방안 참조

B. 프로젝트에 교육생 참여와 취업 연계 방안 참조

3) 국내 게임 관련 대학 커리큘럼의 발전적 모델 제시로 활용

A. 국내 게임 관련대학의 커리큘럼 대안 제시

B. 학점 인정제와 학점 교류제의 방안 제시

## II. 게임 산업과 전문 인력 양성

### 2.1 게임 산업의 특성과 현황

#### 2.1.1 세계 게임시장의 규모 및 전망

2005년 세계 게임시장은 지난 2004년 394억 달러에서 9.5% 성장한 약 432억 달러 규모로 추정되며, 2010년까지 13.2%의 연평균 성장률을 기록하면서 지속적인 성장세를 보일 것으로 전망된다. 시장 규모에 있어서는 콘솔게임(181억 7,900만 달러)과 아케이드게임(130억 3,800만 달러)순으로 시장을 형성하고 있으나, 온라인게임과 모바일게임이 2005~2010년 까지 가장 높은 연평균 성장률(33.1%, 33.9%)을 기록할 것으로 전망된다. 콘솔게임 시장은 5년 주기로 출시되는 새로운 콘솔에 따라 시장 규모가 좌우되기 때문에, 2005년 말부터 2006년 초 예정되어 있는 차세대 콘솔 출시로 시장 재도약에 대한 기대가 고조되고 있으며, 온라인게임은 중국을 비롯한 아시아 시장에서의 고성장과 미국/유럽 등으로의 시장 확대로 지속적인 성장세를 보이고 있다. 한편, 마이너스 성장률을 보이고 있는 PC게임 시장은 향후에도 시장규모의 지속적인 축소가 전망되고 있다.

#### 2.1.2 세계 게임시장의 주요 트렌드

##### 1) 콘솔게임 시장

콘솔게임 시장의 구조적 특징은 한마디로 플랫폼 3사 중심구도 속에 대형 퍼블리셔가 주도하는 시장이라고 할 수 있다. 콘솔 제조업체 중심으로 콘솔업체-퍼블리셔-개발자 연합이 각각 형성되어 있으며, 콘솔 제조업체들이 퍼블리셔들과 개발자들을 이끌고 있다. 그러나 EA 같은 대형 퍼블리셔는 콘솔업체의 주도 아래 있는 것이 아니라 오히려 시장을 주도하고 있어, EA가 Sony와 MS 중 어느 진영을 선호하느냐에 따라 시장판도가 바뀌기도 한다.

콘솔게임 시장은 차세대 콘솔 출시에 크게 영향을 받으며, 차세대 콘솔이 출시되는 5년을 기점으로 시장의 성장과 정체를 반복한다. 이미 MS

가 XBox 360을 2005년 11월 출시했으며, Sony의 PS3 출시도 2006년 초로 예정되어 있어, 차세대 콘솔 시장을 둘러싼 소니, MS, 닌텐도 3사간 주도권 경쟁은 점점 더 치열해질 전망이다.

표 II-1. 세계 게임시장의 규모 및 전망

(단위: 백만 달러)

권역별	세부시장	2004	2005	2006	2007	2008	2009	2010	'05~'10 CAGR
세계전체	콘솔게임	18,179	17,991	21,315	24,439	25,845	27,003	25,878	7.5%
	PC게임	3,450	3,398	3,322	3,170	3,073	2,981	2,860	-3.4%
	온라인게임	3,179	4,799	7,195	10,484	13,809	17,027	20,064	33.1%
	모바일게임	1,547	3,269	5,566	7,843	10,088	11,646	14,065	33.9%
	아케이드게임	13,038	13,696	14,278	15,094	15,827	16,587	17,346	4.8%
	계	39,392	43,153	51,677	61,029	68,642	75,243	80,214	13.2%
미주권	콘솔게임	6,629	6,322	7,251	8,741	9,038	9,137	8,714	6.6%
	PC게임	1,176	1,159	1,133	1,096	1,072	1,037	1,008	-2.8%
	온라인게임	698	1,002	1,520	2,431	3,171	3,804	4,436	34.7%
	모바일게임	306	640	1,130	1,596	2,020	2,396	2,877	35.1%
	아케이드게임	4,586	4,795	5,013	5,298	5,512	5,747	5,983	4.5%
	계	13,395	13,918	16,047	19,162	20,813	22,121	23,018	10.6%
유럽권	콘솔게임	4,290	4,108	4,729	5,730	5,973	6,090	5,835	7.3%
	PC게임	771	757	732	704	683	655	630	-3.6%
	온라인게임	461	787	1,285	2,055	2,967	3,846	4,746	43.2%
	모바일게임	458	1,107	1,910	2,672	3,403	3,721	4,579	32.8%
	아케이드게임	5,008	5,228	5,411	5,748	6,084	6,421	6,757	5.3%
	계	10,988	11,987	14,067	16,909	19,110	20,733	22,548	13.5%
일본	콘솔게임	3,944	4,521	5,850	6,106	6,708	7,315	7,158	9.6%
	PC게임	219	203	182	168	152	135	119	-10.1%
	온라인게임	202	289	362	429	514	592	670	18.4%
	모바일게임	402	478	545	612	671	729	780	10.3%
	아케이드게임	1,606	1,687	1,785	1,886	1,995	2,101	2,206	5.5%
	계	6,373	7,178	8,724	9,202	10,040	10,872	10,933	8.8%
중국	콘솔게임	84	92	112	131	145	161	177	14.0%
	PC게임	53	54	57	61	63	66	68	4.8%
	온라인게임	420	621	849	1,146	1,361	1,601	1,842	24.3%
	모바일게임	71	146	269	387	508	629	749	38.7%
	아케이드게임	102	130	137	145	153	161	168	5.4%
	계	730	1,043	1,424	1,870	2,230	2,617	3,005	23.6%
아·태권	콘솔게임	2,801	2,511	2,889	3,129	3,324	3,587	3,282	5.5%
	PC게임	1,130	1,123	1,096	1,019	982	968	906	-4.2%
	온라인게임	1,396	2,095	3,168	4,405	5,773	7,155	8,335	31.8%
	모바일게임	310	898	1,712	2,576	3,486	4,171	5,080	41.4%
	아케이드게임	1,738	1,857	1,933	2,018	2,083	2,158	2,232	3.8%
	계	7,375	8,483	10,798	13,147	15,649	18,039	19,836	18.5%
남미권	콘솔게임	430	437	484	602	657	712	712	10.3%
	PC게임	101	102	122	122	121	120	128	4.6%
	온라인게임	2	6	11	17	23	29	35	42.1%
	모바일게임	-	-	-	-	-	-	-	-
	아케이드게임	-	-	-	-	-	-	-	-
	계	533	545	617	741	801	861	875	9.9%

[자료원: PWC 2005, Global Entertainment and Media Outlook: 2005-2009, NPD Group 2005, DFC 2004, Ovum 2004, Iresearch 2005, CESA 2004, Jamma, 2005, IDC 2004, 2005, NRI, 2005]

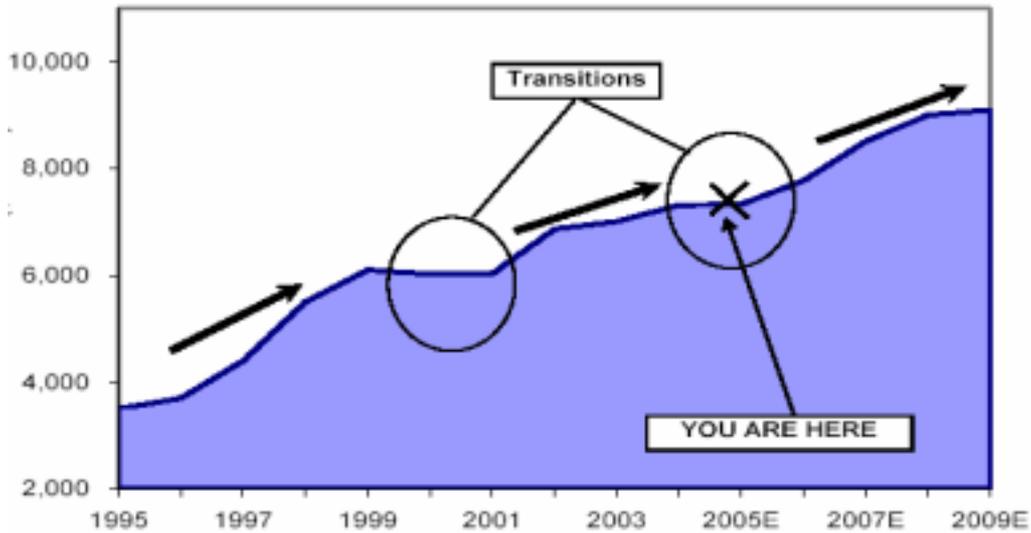


그림 II-1. 콘솔게임 시장의 성장주기

[자료원: NPD Group, NPD Funworld, TRSTS Video Game Service and Thomas Weisel Partners]

## 2) 온라인게임 시장

세계 온라인게임 시장에서는 다음과 같은 특징들이 나타나고 있다.

첫째, 세계 온라인게임 시장 역시 연평균 두 자리 수 성장률을 보이며 꾸준히 증가할 전망이다. 2004년 말 기준의 전세계 온라인게임 시장규모는 32억 달러 수준이며, 전체 게임시장에서 차지하는 비중은 약 8.1% 수준이다. 그러나 온라인게임은 컴퓨터 보급률의 증가, 브로드밴드 인프라의 확산/고속화/안정화에 힘입어 2005년~2010년까지 약 33.5%의 연평균 성장률을 보이며 고속성장 할 것으로 전망된다.

둘째, 거대 콘솔게임 퍼블리셔들의 시장 지배 현상이다. 국내 게임 퍼블리셔인 NcSoft가 매출 기준으로 전세계 온라인게임 시장점유율 1위를 기록하고 있으며, 나머지 시장은 전통적인 콘솔게임 퍼블리셔들이 지배하고 있는 실정이다.

셋째, 제한된 장르를 들 수 있다. 온라인게임은 주로 MUD게임, 그 중에서도 MMORPG 장르를 중심으로 시장이 성장하고 있다. MMORPG 이용자들은 대부분 하드코어 유저들로 일주일에 몇 시간씩을 온라인게임 이용에 할애하는 소비자들이다. 따라서 캐주얼 게임류의 보다 대중적인 게임들이 높은 인기를 끌게 된다면 온라인게임 시장은 더욱 폭넓게 확산될 수 있을 것이다.

### 3) PC게임 시장

PC 게임은 현재 시장 규모가 점점 축소되고 있는 사양 산업이라고 할 수 있다. 특히, 아시아 지역에서는 온라인게임의 인기와 맞물려 PC 게임 시장은 더욱 빠르게 축소되고 있다.<sup>1)</sup> 또, PC 게임은 불법복제에 취약하기 때문에 중국을 비롯한 아시아 시장에서는 시장 성장에 커다란 장애요인으로 작용하고 있다.

PC게임은 아직까지 인터넷 보급률이 낮은 지역을 중심으로 수요가 남아 있기 때문에 한동안은 시장이 유지되겠지만, 향후 온라인게임으로 통합되면서 시장은 빠른 속도로 축소될 전망이다.

### 4) 아케이드게임 시장

아케이드게임 시장은 게임 시장에서 콘솔게임 다음으로 높은 비중을 차지하고 있다. 이는 아케이드게임의 경우 소프트웨어와 하드웨어를 분리하지 않고 통합적인 상품으로 취급하기 때문에 상대적으로 시장규모가 크게 산정되고 있음에 기인한다.

아케이드게임은 소위 '오락실'로 불리는 전용게임장과 백화점을 비롯한 각종 레저센터 및 공원 등에 설치되어 있으며, 편의점이나 상점에서 작은 규모로 1~2대의 아케이드 게임을 설치하여 운영하는 경우도 있다.

아케이드 게임 시장은 2000년대 '펌프'류의 댄스게임의 인기 이후 이렇다 할 후속작이 없어 침체기를 맞이하고 있다가 2003년 이후 성인게임 시장의 성장으로 다시금 성장세를 보이고 있다. 아케이드 게임은 이후에도 경마, 메달 게임류의 성인게임 장르 위주로 완만한 성장을 보일 것으로 전망된다.

### 5) 모바일게임 시장

젊은 층을 중심으로 이용자수가 확대되면서 모바일게임 시장이 점차 활성화되고 있다. 최근에는 국내나 일본에서 단말기 액정의 고화소 및 대형화로 인해 3D 게임과 네트워크 게임 등 기존 유선인터넷에서 가능했던 게임들이 모바일로 이식되고 있다. 모바일 게임은 모바일 Entertainment 시장에서 음악 다음으로 주도적인 역할을 해 왔으며, 성장가능성에서는 가장 주목을 받고 있는 분야이다. 네트워크 및 단말기 고도화로 인해 3D 게임, 네트워크 게임 등이 최근 잇따라 출시되고 있는 가운데, 브랜드 파워와 Visual적

1) 국내의 경우에는 2005년에 국내에서 자체 생산되어 출시된 PC 게임 타이틀이 거의 전무할 정도임

인 측면이 소비자에게 어필하고 있는 추세다.

표 II-2. 게임 업종 매출 규모(전체)

(단위: 억원)

구분	2004년	2005년	성장률
아케이드 게임	4,031	4,197	4.1%
PC게임	802	701	-12.6%
비디오게임	2,118	2,495	17.8%
온라인게임	12,061	14,644	21.4%
모바일게임	1,785	2,101	17.7%
합계	20,797	24,138	16.1%

표 II-3. 게임 업종 매출 규모(콘텐츠/광고)

(단위: 억원)

구분	2004년		2005년		성장률	
	콘텐츠	광고	콘텐츠	광고	콘텐츠	광고
아케이드	4,031	0	4,197	0	4.1%	0.0%
PC	802	0	701	0	-12.6%	0.0%
비디오	2,118	0	2,495	0	17.8%	0.0%
온라인	11,863	198	14,424	220	21.6%	11.1%
모바일	1,785	0	2,101	0	17.7%	0.0%
총계	20,599	198	23,918	220	16.1%	11.1%

## 2.2 게임 시장 전체 규모 및 동향

게임 산업은 PC와 가정용 게임 콘솔, 업소용 게임기 등에서 컴퓨터 프로그램으로 만들어진 게임을 플레이하는 것으로 그 동안 국내 디지털콘텐츠 산업의 대표적인 산업으로 자리매김해 왔다. 현재 게임 산업의 분류는 대부분의 기관에서 플랫폼에 따른 분류를 채택하고 있으며 통상적으로 온라인 게임, PC 게임<sup>2)</sup>, 모바일 게임, 비디오 게임, 아케이드 게임 등 5개의 플랫폼별 분류를 채택하고 있다. 주지하다시피 한국은 2000년 이전에는 외국산 패키지 게임들의 소비시장 역할을 해왔으나 2000년을 전후해서 인기를 끌기

2) PC 게임은 'PC 패키지 게임'을 의미한다.

시작한 온라인 게임이 해마다 비약적으로 발전하며 세계 시장에서 게임 생산국으로서의 지위를 공고히 해왔다.

이렇게 온라인 게임이 발전하기 시작하면서 국내 게임 시장의 규모도 크게 증가하기 시작했다. 2001년 7,606억 원 규모였던 게임 시장은 연평균 성장률 33.5%를 기록하면서 2005년에는 2조 4,138억 원으로 성장했다. 2002년 이후 40% 내외를 유지했던 성장률은 2005년에 들어서면서 16%대로 떨어졌지만 시장 규모는 2조 5천억 원에 육박해 여전히 국내 디지털콘텐츠 최대의 시장으로서 산업 내에서 중요한 위치를 차지하고 있다.

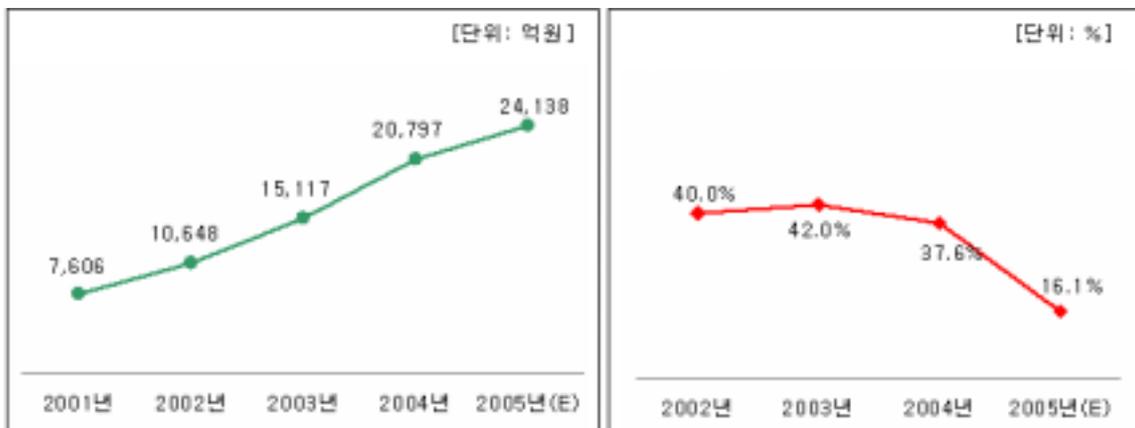


그림 11-2. 게임 시장 규모 및 성장률 추이

게임 시장이 해마다 큰 폭의 성장을 이어 가면서 디지털콘텐츠 시장에서 게임 시장이 차지하는 비중도 꾸준히 증가해 왔다. 2001년 26.3%를 기록했던 게임 시장의 비중은 급격히 증가해 2004년에는 30.2%를 기록했고 2005년에는 성장률이 떨어지면서 2004년도보다 다소 하락한 30.0%의 시장 비중을 기록했다. 이러한 하락에는 역시 한국 게임 산업의 주력이던 온라인 게임의 시장 성장이 다소 둔화된 것, 그리고 두 번째로 큰 시장을 형성하고 있는 아케이드 게임 시장의 성장률이 둔화된 것에서 그 이유를 찾을 수 있다. 온라인 게임의 경우 지난해 50%가 넘는 성장률을 기록했지만 올해는 21.4%의 성장률을 기록해 성장률이 크게 떨어졌으며 아케이드 게임의 경우에도 작년 20%에 가까운 성장률에서 올해는 4.1%의 성장률로 성장률이 크게 떨어졌다.

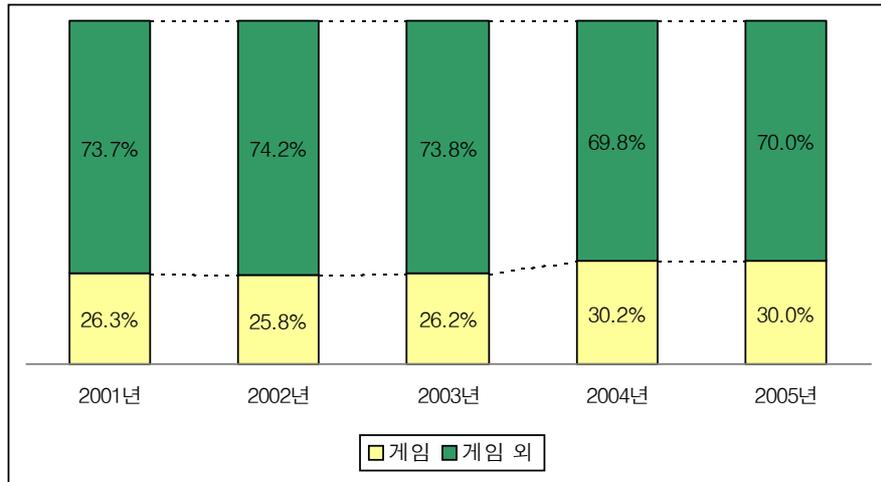


그림 II-3. 디지털콘텐츠 시장 중 게임 시장 비중

### 2.3 플랫폼별 규모

게임 업종에서 가장 큰 규모를 형성하고 있는 온라인 게임은 규모가 가장 크에도 불구하고 가장 높은 성장률을 기록했다. 지난 해 1조 2,061억 원이었던 온라인 게임 시장 규모는 올해 새로운 장르의 부분 유료화 게임이 큰 인기를 모으고 해외 온라인 게임이 수입되어 큰 인기를 모으는 등의 효과에 힘입어 전년 대비 21.4% 성장한 1조 4,644억 원에 달했다. 한편, 지난 몇 년간 경이적인 성장률을 기록하던 모바일 게임의 성장세는 한풀 꺾여 올해는 17.7%의 성장에 머물렀다.

표 II-4. 게임 업종 매출 규모(전체)

단위: 억원

구분	2004년	2005년	성장률
아케이드 게임	4,031	4,197	4.1%
PC게임	802	701	-12.6%
비디오게임	2,118	2,495	17.8%
온라인게임	12,061	14,644	21.4%
모바일게임	1,785	2,101	17.7%
합계	20,797	24,138	16.1%

표 II-5. 게임 업종 매출 규모(콘텐츠/광고)

단위: 억원

구분	2004년		2005년		성장률	
	콘텐츠	광고	콘텐츠	광고	콘텐츠	광고
아케이드	4,031	0	4,197	0	4.1%	0.0%
PC	802	0	701	0	-12.6%	0.0%
비디오	2,118	0	2,495	0	17.8%	0.0%
온라인	11,863	198	14,424	220	21.6%	11.1%
모바일	1,785	0	2,101	0	17.7%	0.0%
합계	20,599	198	23,918	220	16.1%	11.1%

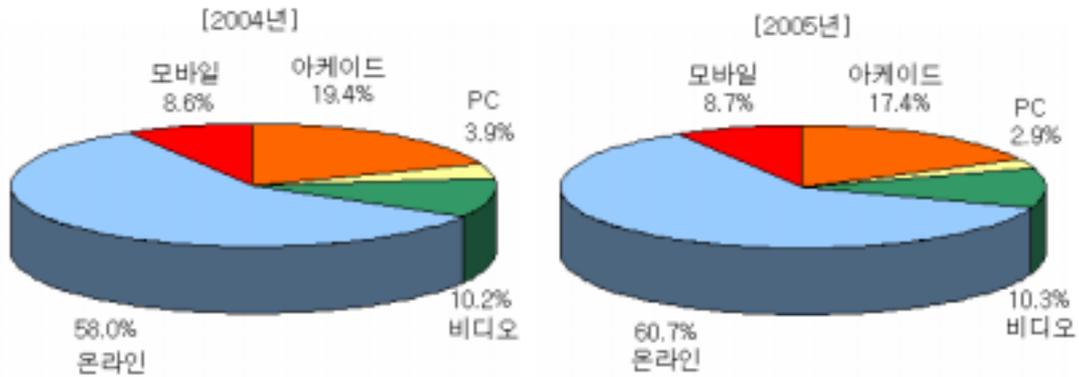


그림 II-4. 게임 시장 플랫폼별 비중

2005년의 경우 모든 플랫폼에서 성장률이 지난해보다 좋지 않아 온라인 게임과 모바일 게임은 성장률이 하락했음에도 불구하고 전체 게임 시장에서 차지하는 비중이 증가한 것으로 나타났다. 온라인 게임의 경우 지난해 58%의 시장 비중에서 올해는 60.7%의 시장 비중을 기록해 지난해에 비해 2.7% 증가한 시장 비중을 보였다. 모바일 게임도 지난해보다 0.1% 정도 상승한 8.7%의 시장 비중을 기록했으며 비디오 게임의 경우도 올해 성장률이 다소 증가하며 작년에 비해 0.1% 상승한 10.3%의 시장 증가를 보였다.

하지만 성장률이 크게 떨어진 아케이드 게임과 성장률이 감소세를 보인 PC 패키지 게임의 경우 지난해보다 시장 비중이 감소했다. 아케이드 게임의 경우 지난해 19.4%에서 2% 감소한 17.4%의 시장 비중을 보였으며 PC 게임의 경우 1% 떨어진 2.9%의 시장 비중을 기록했다.

## 2.4 국내 게임 산업 주요 동향

### 2.4.1 MMORPG와 캐주얼 온라인 게임

올해 한국 게임 시장의 성장률 감소는 가장 큰 규모를 차지하고 있는 온라인 게임과 아케이드 게임 시장의 성장률 감소와 그동안 초고속 성장을 해온 모바일 게임 시장의 정체가 가장 큰 요인으로 지목될 수 있다. 그중에서도 게임 시장 규모의 60%를 차지하고 있는 온라인 게임 시장의 성장률 감소는 게임 시장의 전체적인 성장률 감소에 가장 큰 영향을 미쳤다.

표면적으로 한국의 온라인 게임은 <카트라이더>와 <스페셜포스> 등 부분 유료화 온라인 게임의 약진과 <월드 오브 워크래프트(WOW)>의 성공적인 국내 시장 진입 등으로 성장률이 크게 증가했을 것이라고 생각되었으나 실제로 한국 온라인 게임 시장에서 중요한 역할을 차지하고 있는 MMORPG 시장이 침체를 겪고 있는 것으로 나타나며 기대했던 것만큼 큰 성장을 이루진 못했다.

#### 1) MMORPG의 시장 정체

한국의 대표적 디지털콘텐츠 산업이며 산업의 성장을 주도해 온 온라인 게임 산업은 올해 들어 다소 성장 정체의 조짐을 보이고 있다. 물론 온라인 게임이 올해 기록한 21.4%의 성장도 여전히 높은 성장률이라고 할 수 있고 게임 시장에서 차지하는 비중도 60% 수준으로 여전히 높은 편이다. 하지만 지난해에 비해 성장률이 크게 감소한 것은 사실이며 이는 그 동안 시장 성장의 견인차 역할을 해 왔던 MMORPG 분야의 성장이 둔화되거나 감소하고 있는 것이 가장 큰 이유로 꼽히고 있다.

이렇게 한국의 게임 시장 성장을 이끌어 왔던 MMORPG 분야의 성장이 둔화되고 있는 가장 큰 이유는 한국의 대표적 MMORPG로 자리 잡아 왔던 엔씨소프트의 <리니지>나 웹젠의 <뮤>, 그라비티의 <라그나로크> 등의 매출이 정체 혹은 감소하는 추세를 보이고 있는 반면, 이들의 뒤를 이을 게임으로 새롭게 등장한 대작 MMORPG들이 유저들로부터 그리 좋은 반응 얻어내지 못하고 있기 때문이다.

실제로 1세대라고 할 수 있는 MMORPG들의 분기별 매출 추이를

살펴 보면 2004년 1분기를 정점으로 점차로 매출이 감소하고 있는 양상을 볼 수 있다.<sup>3)</sup> 2004년 1분기부터 올해 4분기까지의 평균 매출 감소율은 약 6.3% 정도로 예상되며 이러한 매출 감소 추세는 올해를 지나 새로운 게임들이 대거 출시되는 내년에는 더욱 심화될 것으로 추정된다.

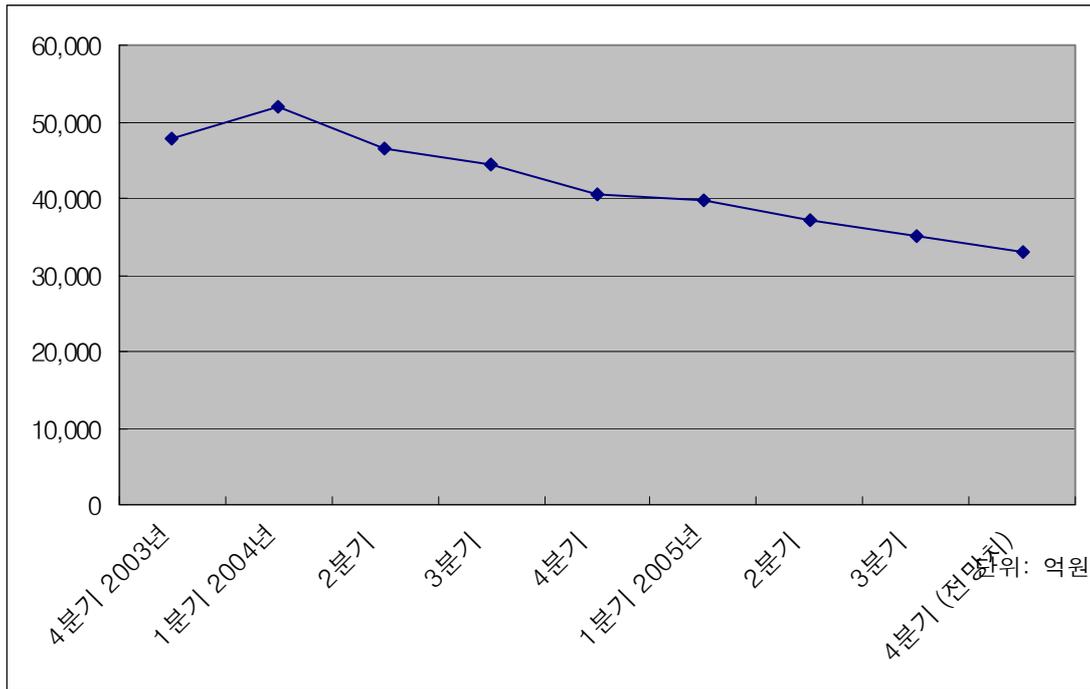


그림 II-5. '리니지'와 '뮤'의 MMORPG의 분기별 매출 추이

이렇게 기존에 인기를 끌던 MMORPG들의 수입 규모가 감소하고 있는 가운데 올해 출시된 대작 MMORPG들의 경우 전체적인 시장의 분위기 반전에 큰 도움을 주고 있지 못하다. 올해 기대를 모으던 대작 MMORPG의 하나인 NHN의 <아크로드>의 경우 오픈 베타 서비스를 실시한지 5개월 여 동안 정식 서비스를 출범시키지 못하고 있었으며, 결국 최근에는 부분 유료화로 시장에 내어 놓기로 결정한 상태이다. 엔씨소프트의 자회사 아레나넷에서 개발한 또 다른 기대작 <길드워>의 경우에도 해외에서는 인기를 얻고 있으나 국내 유저들에게는 좋은 반응을 얻고 있지 못하다.

그리고 올해 출시되어 시장 활성화에 큰 기여를 할 것이라 생각되었던 웹젠이나 한빛소프트, 넥슨의 새로운 대작 MMORPG 들은 제작 일정

3) 그래프의 MMORPG의 분기별 수입은 <리니지>와 <뮤>의 분기별 매출을 합해 정리한 것

이 지연되면서 아직 클로즈드 베타 테스트를 실시하고 있는 중이어서 MMORPG의 성장에 큰 도움을 주고 있지 못하고 있으며 기대 밖의 큰 선전을 거둔 써니YNK의 <로한>이나 이스트소프트의 <카발 온라인>도 아직 정식 서비스를 시작하지 않아서 MMORPG의 성장세 회복에는 크게 기여하고 있지는 못한 실정이다.

## 2) 캐주얼 온라인 게임의 대두

이런 시장 분위기 가운데서 20%가 넘는 온라인 게임의 성장세는 캐주얼 온라인 게임과 새로운 장르의 온라인 게임들이 주도하고 있다. 그 중 2004년 후반기부터 '국민 게임'으로 자리 잡아 온 넥슨의 <카트라이더>의 성공은 가장 놀라운 것인데, <카트라이더>는 2005년 한 해에도 놀라운 유저의 증가세를 보이면서 누적 회원 수 천만 명을 돌파했으며 현재 약 천 5백만 명의 누적 회원이 가입되어 있는 상태다.



<카트라이더>, 넥슨



<갯앰프드>, 원디소프트

※ 엔씨소프트의 <리니지>도 누적 가입자 1천만을 돌파했으나 1인당 복수 계정이 허용되어 제외

그림 II-6. 인기 캐주얼 게임 <카트라이더>와 <갯앰프드>

역대 캐주얼 게임 중에 누적 회원 가입 천만 명을 넘긴 게임은 CCR의 <포트리스>, 넥슨의 <크레이지 아케이드 비앤비>와 <메이플 스토리>, 그리고 <카트라이더>보다 조금 늦게 누적 회원 천만 명을 돌파한 원디소프트의 <갯앰프드> 등이 있다. 하지만 누적 회원 천만을 돌파한 다른 게임들은 모두 게임을 서비스하기 시작한지 1년이 넘는 시점에서 천만 명의 회원을 돌파한 것에 비해 <카트라이더>는 서비스 개시 약 7개월 만에 누적 회원 천만 명을 돌파해 큰 인기를 얻고 있음을 짐작할 수 있다.

표 II-6. 누적 가입자 1천만 명 돌파 게임

게임명	출시년도	가입자 천만명 돌파	개발사
포트리스	1999년	2001년	CCR
비앤비	2001년	2002년	넥슨
메이플 스토리	2003년	2005년	넥슨
카트라이더	2004년	2005년	넥슨
갯앰프드	2003년	2005년	원디소프트

물론 누적 가입자가 천만 명을 돌파한 게임들은 모두 게임 접속은 무료로 할 수 있고 게임 안에서 사용되는 아이템 판매를 통해 수익을 얻는 부분 유료화 게임들이어서 현재 대부분 월 정액제로 운영되고 있는 MMORPG에 비해 게임 가입자 확보가 쉽긴 하지만 누적 회원이 천만 명이 넘는 게임들이 속속 탄생하고 있다는 것은 게임이 많은 국민들에게 새로운 놀이 문화로 다가가고 있으며 이러한 게임 문화의 정착에 귀엽고 쉬운 게임 성을 가진 캐주얼 게임들이 크게 기여하고 있다는 점은 부인할 수 없는 점이다.

이렇게 보았을 때 결국 2005년의 게임 시장은 다소 부진했던 MMORPG 시장과 높은 성장을 보인 캐주얼 온라인 게임 시장이 시장의 분위기를 주도했던 한해라고 할 수 있다. 특히 2005년의 새로운 트렌드로 등장한 캐주얼 온라인 게임의 인기는 MMORPG 시장의 부진에도 불구하고 온라인 게임 시장을 성장으로 이끄는 중요한 동인이 되어 주었다.

#### 2.4.2 게임 장르의 다양화

가벼운 게임의 재미로 승부하는 캐주얼 게임들의 등장과 함께 2004년 말부터 2005년도까지 게임 시장에 등장한 새로운 트렌드 중 하나는 다양한 장르의 게임들의 출시가 늘어나고 있다는 점이다. 많은 시간을 투자해서 한 번에 오랫동안 게임을 진행해야 하는 RPG 형식의 게임과는 달리 스포츠, 격투, 레이싱, FPS 등의 게임은 모두 짧은 시간 안에 승부가 결정되는 명확한 구조의 게임으로 청소년 게임 사용자들에게 큰 인기를 끌고 있다.



<스페셜포스>, 드래곤플라이



<프리스타일>, 제이씨엔터테인먼트

그림 II-7. 인기 비 MMORPG 게임 <스페셜포스>와 <프리스타일>

실제로 2005년도 시장에서 크게 히트한 게임들은 제이씨엔터테인먼트의 농구 게임 <프리스타일>과 현재 네오위즈의 간판 게임으로 자리매김한 드래곤 플라이의 FPS 게임 <스페셜포스>, 새로운 국민 게임으로 떠오른 넥슨의 <카트라이더> 등이고 새롭게 등장한 RPG 중에서 히트하고 있는 게임은 블리자드의 <월드오브워크래프트> 외에는 없는 형편이어서 유저들의 게임 취향이 다양해졌다는 점을 반증해 주고 있다.

2년 전인 2003년에 게임 이용자들에게 큰 인기를 얻었던 게임들을 살펴보면, 현재 게임 시장에 장르의 다양성 측면에서 많은 진전이 있었음을 알 수 있다. 2003년 10월 PC방 인기 게임 순위<sup>4)</sup>에는 미국 게임 업체 블리자드의 패키지 게임 세 개를 제외한 모든 게임들이 RPG와 카드 게임으로 채워져 있는 데 비해 2005년의 PC방 인기 순위를 보면 RPG와 함께 FPS, 스포츠, 레이싱 등의 다양한 장르의 게임들이 높은 순위를 차지하고 있는 것을 알 수 있다. MMORPG의 시장 한계로 인한 다양한 장르의 게임 출시가 사용자들에게는 오히려 더 다양한 게임을 즐길 수 있는 환경을 가져다 준 것으로 풀이된다.

4) PC방 인기 게임 순위는 '게임트릭스(www.gametrics.com)' 자료 기준

### 2.4.3 해외 게임사의 국내 온라인 게임 시장 진출

2005년도에도 한국의 온라인 게임 시장 진출을 목표로 국내 시장에서 게임을 출시한 많은 해외 게임사들이 있었다. 먼저 2004년 말 오픈 베타 이후로 게임 이용자와 국내 게임 업계 관계자들의 커다란 반향을 불러 일으켰던 블리자드의 새로운 MMORPG <월드 오브 워크래프트>와 일본 패키지 게임의 강자 KOEI의 <대항해 시대 온라인>, 중국 스네일게임의 <항해세기>, 소니온라인엔터테인먼트의 <에버퀘스트2> 등 주로 MMORPG를 중심으로 한국 시장의 진출이 이루어졌다.

이중 블리자드의 <월드 오브 워크래프트>는 블리자드의 전략 시뮬레이션 게임인 <워크래프트>를 RPG로 바꾸어 시장에 내어 놓아 국내 게임 시장에 신선한 충격을 던져 주었다. <월드 오브 워크래프트>는 2004년 11월 말 국내에서 오픈 베타 서비스를 시작하면서 기존 게임인 <워크래프트>에서 가져온 탄탄한 세계관과 이를 바탕으로 한 일관성 있는 스토리, 치밀하게 연결된 퀘스트 구조, 균형 있게 잘 분화된 직업과 그를 이용한 전략적 파티 플레이를 선보여 사용자들의 절대적인 지지를 얻어내 2005년 최고의 기대작으로 떠올랐다.

특히나 치밀하게 잘 짜인 퀘스트 중심의 게임 플레이는 그동안 단순한 '핵&슬래시'<sup>5)</sup> 게임에 길들여져 있던 한국의 유저들과 게임 개발자들에게 회자되며 이후 국내 게임의 개발에 큰 영향을 끼치게 되었다. <월드 오브 워크래프트>에서 선보인 퀘스트 개념은 이후 등장한 모든 MMORPG의 필수 요소가 될 정도였다.

World of Warcraft	개발사 : Blizzard Entertainment 배급사 : 블리자드 코리아 서비스 : 상용화	항해세기	개발사 : Snail Games 배급사 : 나인브라더스 서비스 : 오픈 베타
Everquest2	개발사 : Sony Online Entertainment 배급사 : 감마니마 코리아 서비스 : 상용화	대항해 시대	개발사 : KOEI 배급사 : CJ Internet 서비스 : 오픈 베타

그림 II-8. 2005년 현재 국내 진출 해외 게임

5) Hack&Slash Game, 레벨을 올리거나 아이템을 얻으려는 목적으로 게임 상에 등장하는 NPC(Non-Player Character)들에 대한 단순한 살육을 반복하는 것이 게임 플레이의 주된 형식인 게임들을 일컫는다.

하지만 국내 MMORPG 게임 시장을 평정할 것으로 기대되던 <월드 오브 워크래프트>는 블리자드 코리아의 유료화 정책과 정식 서비스 초기의 운영 미숙 등으로 유저들의 큰 반발을 불러일으키며 이용자가 급감하는 사태를 맞게 되었다. 이로 인해 한 때 동시 접속자 수 40만 명이 된다고 이야기 됐던 이 게임은 유료화 직후 동시 접속자 수가 10만 명 이하로 떨어지며 국내 게임계에 당초 예상했던 것만큼 큰 파장을 미치지 못하는 못하며 절반의 성공으로 만족해야 했다.

<월드 오브 워크래프트>의 출시 이후 국내 게임 제작사들의 철용성으로 여겨졌던가 시장에 해외 게임들이 물 밀듯이 들어올 것으로 예상되었지만 <월드 오브 워크래프트> 이후 한국 MMORPG 시장에서는 뚜렷한 히트작이 생산되지 못하는 좋지 않은 시장 분위기가 이어지며 현재는 앞서 언급한 네 개 정도의 MMORPG가 국내 유저들에게 서비스 되고 있으나 역시 아주 좋은 반응을 얻고 있지는 못한 실정이다.

#### 2.4.4 기업들 간의 인수 합병

2004년도부터 본격적으로 시작된 게임 업계의 인수 합병 열풍은 2005년도에도 그대로 이어져 많은 개발사들이 대기업이나 대형 게임사들에 인수되거나 합병되었다. 게임 업계의 인수 및 합병은 게임 개발의 요구가 증대되며 대형 게임 개발사들이 개발력을 확보하기 위해, 대형 게임 회사들이나 게임 경험이 없는 대기업이나 국내 온라인 게임 시장에 경험이 없는 해외 게임 기업들이 게임 시장 진출의 발판을 마련하기 위해, 그리고 상장 기업과의 인수 합병을 통한 우회 상장을 위해 실시되었다.

2005년 한 해 가장 논란이 되었던 인수 합병 관련 사건은 중국 기업 산다의 액토즈소프트 인수와 일본 기업 소프트뱅크의 그라비티 인수를 들 수 있다. 산다는 2004년 12월 액토즈소프트를 3천 3백억 원의 매각 대금

- 6) <월드 오브 워크래프트>의 국내 서비스를 맡고 있는 블리자드 코리아 측은 유료화 이후 동시 접속자 수가 10만을 넘었다고 발표했다. 하지만 일부 네티즌들과 PC방 관계자들은 '3~4만 명 선일 것'이라고 반박하는 등 유료화 이후 동시접속자 수를 놓고도 많은 추측이 있었다.
- 7) <월드 오브 워크래프트> 이전에도 해외에서 명성을 얻었던 많은 MMORPG 게임들이 국내에 진출했으나 좋은 성과를 거두지 못하고 있거나 서비스를 그만두어야 했다. 대표적인 게임으로는 소니온라인엔터테인먼트의 <에버퀘스트>와 터바인엔터테인먼트의 <애서론스 콜>이 서비스를 중단해야 했으며 미씩엔터테인먼트의 <다크 에이지 오브 카멜롯>이나 지금은 엔씨소프트 소속인 리처드 개리엇이 제작한 오리진의 <울티마 온라인> 등도 적은 유저 수로 고전했다.

에 인수했고 이어 2005년에는 <라그나로크>를 일본 내에서 서비스 하던 경호 온라인의 모회사 소프트뱅크가 그라비티 최대 주주와 그 일가의 지분 약 58%를 4천억 원에 인수하면서 또 한 번 게임 업계에 큰 파장을 일으켰다.

한국의 액토즈소프트와 위메이드엔터테인먼트가 개발한 '미르의 전설2'를 중국 내에 서비스하며 큰 성장을 이룩한 산다의 경우 이들과 게임을 놓고 법정 소송을 진행 중이었기 때문에 이러한 소송에 대한 무마를 위해 아예 인수를 택한 것이 아니냐는 관측이 나돌기도 했었다. 또한 이제 한국 게임을 서비스하며 큰 부를 쌓은 중국 업체들이 한국 업체들을 인수하면서 온라인 게임의 기술력을 급속히 갖추어 한국 게임 산업에 큰 위협으로 등장할 것이라는 우려의 목소리가 컸었다.

그로부터 얼마 뒤 2005년 5월에는 그라비티가 소프트뱅크에 인수되었다. 산다의 액토즈소프트 인수에 이어 또다시 국내 게임을 현지에서 서비스하던 업체가 오히려 국내 게임 기업을 인수하자 논란의 목소리는 더욱 커졌다. 게다가 산다의 경우 <미르의 전설2>를 서비스하며 나스닥 상장까지 간 경우이고 소프트뱅크의 계열사 경호 온라인의 경우 자사 매출의 80~90%를 그라비티의 <라그나로크>에 의존하고 있었기 때문에 한국 게임들이 노력한 결과를 고스란히 외국 기업에게 빼앗기고 있다는 부정적인 인식들이 게임계에 확산되기도 했다.

이외에도 <광야>를 개발한 엔트리브소프트가 SKT 계열 연예 기획사인 IHQ에 인수된 것이나 CJ가 게임 포털 '넷마블'을 서비스하는 플래너스를 인수한 것 등 대기업이 게임 기업을 인수해 시장 진출을 꾀하거나 넥슨이 모바일 게임 회사 엔텔리전트를, 그리고 웹젠이 중소 개발사 3개를 인수한 것처럼 자체 개발력 강화를 위해 다른 개발사들을 인수한 사례도 2005년의 게임 기업 인수 합병에서 볼 수 있었던 사례였다.

#### 2.4.5 휴대용 게임기의 약진

2005년도 비디오 게임 시장의 경우 지난해에 비해 시장 성장률이 다소 증가한 17.8%를 기록했다. 가정용 콘솔의 양대 산맥인 PS2와 XBOX가 모두 플랫폼의 노후화 증세를 보이며 소프트웨어와 하드웨어 양쪽에서 판매량이 감소하고 있는 추세이고 XBOX의 경우 2005년 PS2의 경우 2006년에 차

세대기의 출시가 예정되어 있음에도 불구하고 국내 비디오 게임 시장이 증가세를 보이게 된 것은 휴대용 게임 시장의 성장이 큰 역할을 했다.

휴대용 게임기는 이전부터 'GBA' 등으로 휴대용 게임기 시장의 강자로 군림하며 전세계에서 점유율 1위를 기록하고 있는 닌텐도와 비디오 게임 시장 전체에서 43%의 점유율로 1위를 차지하고 있는 소니의 대결로 많은 관심을 모았었다. 1년 간 두 게임기의 전적은 전세계의 출하량에서는 닌텐도가 12월 현재 1천 3백만대, 소니가 1천만대 정도의 판매량을 기록하고 있어 닌텐도의 우세가 이어지고 있지만 국내에서는 퍼블리셔의 강력한 마케팅에 힘 입어 PSP 등의 휴대용 게임기의 성장세가 두드러졌다. PSP는 국내에서 20만대 이상 판매되면서 거치형 콘솔 플랫폼의 노후화로 인한 시장 감소를 상쇄하며 비디오 게임 시장의 성장을 이끌었다. PSP의 경우 소프트웨어의 판매량도 높은 편이어서 '모두의 골프'나 '릿지 레이서' 등 인기 타이틀의 경우 6만장 이상의 판매량을 기록하고 있는 것으로 알려지고 있다.

#### 2.4.6 성인 아케이드 게임 시장

올 한해 아케이드 게임 시장의 가장 큰 이슈는 작년 말부터 규제와 뇌물 스캔들에 시달렸던 스크린 경마 게임의 감소와 이를 대체하며 큰 폭으로 성장하고 있는 릴게임, 즉 성인용 도박 게임이었다.

지난 해 20% 가까이 성장하며 큰 폭의 성장세를 보였던 아케이드 게임이 2005년 4% 대의 성장률 감소를 겪게 된 가장 큰 원인은 스크린 경마 게임의 감소였다. 2005년 들어 스크린 경마 게임은 신규 게임장의 수요가 크게 줄어들면서 대표적인 몇몇 업체들의 경우 매출이 50% 내외로 감소하고 있는 것으로 알려졌다. 이에 따라 약 2천억 원대로 추산되고 있는 스크린 경마 등 성인 게임의 시장 규모는 크게 감소하게 되었다.

이와 같은 스크린 경마 게임의 감소에도 불구하고 아케이드 시장이 마이너스 성장으로 돌아서지 않은 것은 이를 대체하며 신규 업소가 큰 폭으로 증가하고 있는 릴게임의 등장 때문으로 추정된다. 2005년 들어서 본격적으로 등장하기 시작한 파친코 게임의 변형이라고 볼 수 있는 릴 게임은 이제 어느 곳에서나 게임장을 볼 수 있을 만큼 신규 게임장이 크게 늘어나고 있으며 인터넷 곳곳에 공략 사이트가 등장하고 있을 정도로 인기를 끌고 있

다. 이러한 인기를 바탕으로 스크린 경마 게임의 감소세를 상쇄하며 큰 폭으로 성장하고 있는 릴 게임 시장이지만 최근 이들 게임도 사행성 논란이 불거지며 영업 시간 규제, 단속 강화 등 규제를 위한 움직임이 일어나고 있어 2006년에도 올해와 같은 성장세를 이어갈 수 있을지는 의문시 된다.

아케이드 게임의 경우, 올해 들어 급격하게 확대되고 있는 릴게임장의 존재에도 불구하고, 스크린 경마 게임장의 신규 수요 감소와 청소년 게임장의 지속적인 시장 감소로 지난 해 대비 의미 있는 시장 확대를 이루지 못했다. 그러므로 만약 2006년 성인용 게임 시장에 대한 규제가 확대되며 이 분야 시장의 위축을 가져올 경우 아케이드 게임의 시장 규모는 2000년 초반과 같이 다시 마이너스 성장으로 돌아서는 상황을 맞을 수도 있을 것으로 예상된다.



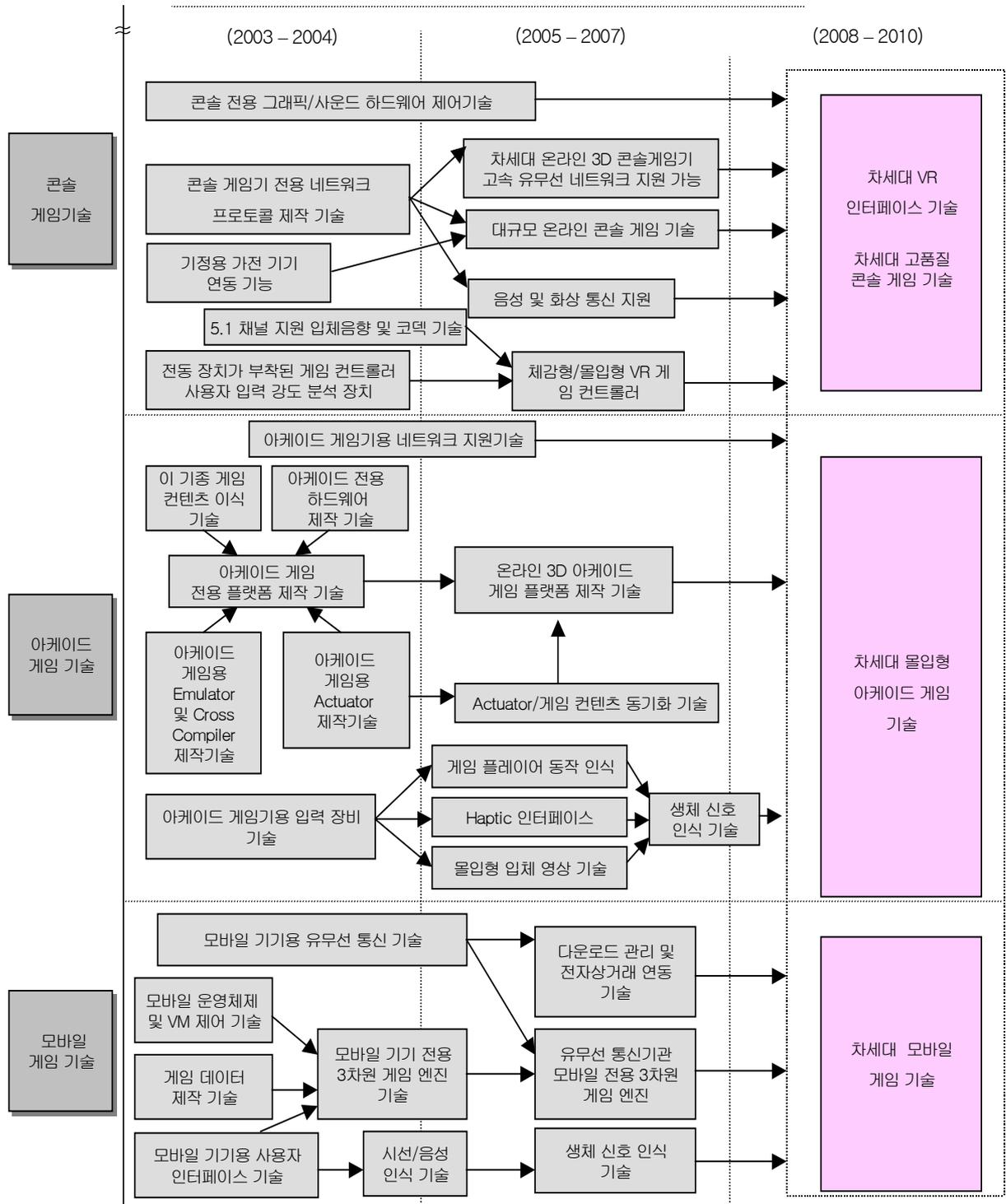


그림 II-10. 게임기술 로드맵 2/2

표 II-6. 국내 게임산업 장단점 분석표

구분	강 점	약 점
법제도	<ul style="list-style-type: none"> <li>· 음비게법, S/W산업진흥법, 컴퓨터프로그램 보호법 등 관계법령에 의한 게임산업 육성을 위한 각종 진흥시책 수립·추진</li> <li>· 정통부, 문광부, 산자부 등 관계부처간 협력 분위기 조성 중</li> </ul>	<ul style="list-style-type: none"> <li>· 관련 정부부처간 유기적인 업무협조 체계의 정책간 시너지효과 부족</li> <li>· 불법복제 게임물의 유통이 아직 근절되지 않고 있음</li> <li>※ '02 미국 IDSA에 의하면 '00~'01 한국의 게임 S/W불법복제규모는 4.9억 달러로 추산(세계 1위)</li> <li>· 정부주도의 경직된 심의제도 운영</li> <li>· 유통게임물의 품질에 대하여 소비자가 믿을만한 기준·정보 부족</li> </ul>
산업기반	<ul style="list-style-type: none"> <li>· 정부의 적극적인 산업육성 정책 추진</li> <li>· 세계 최고 수준의 초고속 유·무선 통신망 구축</li> <li>· 고급 사양의 PC, 휴대형 단말기 등 고성능 게임 H/W플랫폼 보급 확산</li> <li>· 폭넓은 게임 사용자층 확보, 전국적인 게임 인구 편재</li> <li>· 정부차원의 투자·용자 기금 운영</li> </ul>	<ul style="list-style-type: none"> <li>· 민간 산업구조의 미성숙               <ul style="list-style-type: none"> <li>- 퍼블리셔 기능을 수행할 민간업체가 부족</li> <li>- 소수 메이저업체와 다수의 중소기업간 콘텐츠개발 및 마케팅연계 부족</li> </ul> </li> <li>· 게임업체들이 온라인/PC게임 분야에 과도하게 편중된 산업구조               <ul style="list-style-type: none"> <li>- 온라인게임내에서도 특정장르(RPG)에 편중되어 문화차이가 있는 외국시장 진출이 어려움</li> </ul> </li> </ul>
기술	<ul style="list-style-type: none"> <li>· 세계 최고 수준의 온라인게임 서버구현 및 운영기술 보유</li> <li>· 우수한 초고속 유·무선 통신인프라로 인해 인터넷기반의 온라인게임콘텐츠 개발경험 및 기술 축적</li> <li>· 모바일온라인게임(2D) 콘텐츠 분야의 앞선 경험과 기술 보유               <ul style="list-style-type: none"> <li>- 우수한 무선인터넷플랫폼 표준규격(WIFI) 독자개발·보유</li> </ul> </li> <li>· 정부주도의 게임엔진(Dream3D) 개발·보급으로 게임콘텐츠제작에 활용되는 세계적 수준의 핵심요소기술 확보</li> </ul>	<ul style="list-style-type: none"> <li>· PC기반의 온라인게임에 편중된 기술               <ul style="list-style-type: none"> <li>- 콘솔 게임, 아케이드 게임, 3D모바일게임 분야의 핵심기술 취약</li> </ul> </li> <li>· 이기종간 연동기술 분야의 핵심기술 부족               <ul style="list-style-type: none"> <li>※ 영국 Creterion사는 2002년 모바일, PC, PS2, Xbox, Game Cube를 모두 지원하는 게임엔진 'RenderWare' 출시</li> <li>※ 일본 SQUARE사는 콘솔(PS2)과 PC를 연동하는 게임콘텐츠 개발 중</li> </ul> </li> <li>· 게임엔진 등 핵심기술 보유업체가 전문화되어 있지 못하고 콘텐츠제작업체와의 기술교류가 거의 전무함</li> </ul>

<p><b>인력 현황</b></p>	<ul style="list-style-type: none"> <li>· 게임관련 대학 및 학원 등 체계적인 교육기관의 증가와 내실 있는 커리큘럼의 개발·운영으로 인력수급 여건 향상</li> <li>· 게임에 대한 사회적 인식제고로 우수한 인력의 게임업계 유입 증가 중</li> <li>· 국내 게임산업의 활성화에 따라 국내외 게임 개발인력간 교류 활성화</li> </ul>	<ul style="list-style-type: none"> <li>· 개선된 인력여건에도 불구하고 교수요원 및 고급 전문인력은 여전히 부족 <ul style="list-style-type: none"> <li>- 시나리오/기획, 프로젝트관리, 고급 프로그래머 및 마케팅 전문인력 등</li> </ul> </li> <li>· 업체들간 과당 인력유치경쟁으로 핵심개발 인력의 업체간 이동이 잦음 <ul style="list-style-type: none"> <li>- 안정적인 콘텐츠개발이 어려움</li> </ul> </li> </ul>
<p><b>시장</b></p>	<ul style="list-style-type: none"> <li>· 국내 게임시장의 급성장으로 세계시장에서의 비중과 중요성 증가 <ul style="list-style-type: none"> <li>- 해외 메이저 게임업체들의 국내 시장에 대한 진출노력 증가</li> </ul> </li> <li>· PS2, Xbox 등 콘솔게임기의 국내 정식발매로 인해 이 분야의 신규시장 조성 중</li> <li>· 온라인/모바일게임 분야의 시장이 지속적인 성장 기조 유지</li> </ul>	<ul style="list-style-type: none"> <li>· 불법복제, 무분별한 번들제공 등 왜곡된 유통구조로 인해 PC용 패키지게임시장이 고사상태</li> <li>· 내수시장 규모의 한계로 인해서 충분한 경쟁환경을 조성하지 못하고, 완성도 있는 내수용 게임제작이 어려움</li> <li>· 외국 메이저 게임제작사의 국내시장 과다점유 <ul style="list-style-type: none"> <li>- 국내 주요 게임업체들의 자체 콘텐츠제작보다는 외국게임 도입 선호경향</li> </ul> </li> <li>· 해외시장에 대한 정보 및 마케팅 경험부족으로 인해 외국 시장진출의 어려움 <ul style="list-style-type: none"> <li>- 중국 등 해외시장에서의 현지업체와의 분쟁 증가</li> </ul> </li> </ul>
<p><b>사회 문화</b></p>	<ul style="list-style-type: none"> <li>· 게임포털사이트, 온라인 게임 동호회 활성화를 통한 건전한 게임 문화 정착</li> <li>· 정규 게임리그의 운영, 게임방송의 상업적인 성공 등 게임 저변 확대 <ul style="list-style-type: none"> <li>- World Cyber Game 개최로 국제적인 게임 문화 주도</li> </ul> </li> <li>· 게임을 새로운 여가활용 수단으로 인식하는 세대의 등장으로 게임 저변 확대</li> <li>· 단순한 오락이 아닌 기능성 게임 등장 <ul style="list-style-type: none"> <li>- 교육, 의료 분야에서의 활용</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>· 게임 중독, 사이버 범죄 등 사회적 역기능이 대두되고 있으나 이를 해소하기 위한 사회적 노력 부족 <ul style="list-style-type: none"> <li>- 아이템 불법 거래, 해킹</li> <li>- 청소년들의 과도한 게임 몰입</li> </ul> </li> <li>· 게임으로 인한 세대간 갈등, 가족 구성원간의 의사소통 단절 <ul style="list-style-type: none"> <li>- 가족 구성원들이 함께 즐길 수 있는 콘텐츠 부족</li> </ul> </li> <li>· 게임을 청소년들의 학업에 지장을 주는 요인으로 보는 편향적인 시각이 잔존</li> </ul>

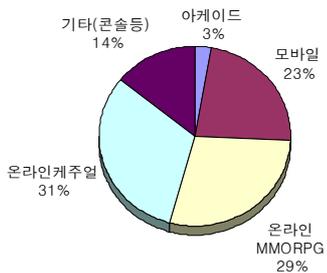
## 2.6 국내 게임 산업계 및 개발 업체들의 현황과 특성

- 국내 게임업체의 최근 추세와 향후 전망을 파악하고 나아가 국내 게임개발 전문 인력 양성 계획 수립을 위하여 수도권과 지방에 있는 33개 업체를 대상으로 설문을 실시하였다.
- 주요 설문 내용은 주력개발제품에 대한 설문, 개발 계획 중인 프로젝트, 개발인력채용, 대학과의 연계계획에 관한 것으로서 조사의 결과는 다음과 같다.

### 2.6.1 주력 개발 제품에 대한 설문 결과

#### 1) 주력개발 제품 현황

○ 주력개발제품에 대한 설문 중 설문업체에서 가장 선호하는 주력 개발 분야는 캐주얼 게임분야로서 약 31%로 가장 높았고, 그 뒤를 이어 온라인 MMORPG게임이 29%, 모바일 게임이 23%였다. 콘솔 및 PSP용 게임도 14%에 이르고 있으며 아케이드분야는 3%로 거의 시장에서 사라지고 있는 추세이다. 향후 주력개발 제품군은 MMORPG나 캐주얼 게임이 주류를 이룰 것으로 예상된다.



게임종류	업체 수	비율
아케이드	1	3%
모바일	8	23%
온라인MMORPG	10	29%
온라인캐주얼	11	31%
기타(콘솔, PSP등)	5	14%

그림 II-11. 주력개발 제품에 대한 설문 결과

#### 2) 주력 제품 개발에 투입된 인력 현황

- 주력제품 개발에 투입된 총 인력은 적게는 5명 미만, 많게는 100

명 이상이 투입된 업체도 있었다. 주력제품개발에 필요한 투입인력현황은 20여명을 투입한 경우가 46%로 가장 높았다. 그 뒤를 이어 5명 미만, 10명 미만 업체가 각각 18%, 20인 이상인 업체가 12%, 100여명이 이르는 업체도 6%나 되었다.

○ 이는 게임의 규모가 점점 대형화되고 그 시장도 국내뿐 아니라 해외진출도 겨냥하고 있다는 점을 의미하기도 한다. 현재 가장 우선순위로 꼽히는 나라가 중국으로서 국내에서 제작한 MMORPG 게임과 캐주얼 게임이 상당수 진출을 서두르고 있다.



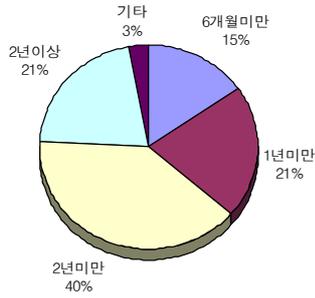
투입인원	업체 수	비율
5인 미만	6	18%
10인 미만	6	18%
20인 미만	15	46%
20인 이상	4	12%
기타	2	6%

그림 II-12. 주력개발 제품에 투입된 인력에 대한 설문 결과

### 3) 주력 제품 개발 기간 현황

○ 업체들의 주력게임 개발에 투입된 기간은 대부분 3년을 넘지 않는 것으로 조사되었다. 전체 조사업체 중 15%는 6개월 미만, 1년 미만과 2년 이상인 업체도 각각 21%, 그리고 40%가 1년 이상 2년 미만의 개발기간이 걸렸다.

○ 개발기간이 2년을 넘는 업체는 MMORPG나 온라인 캐주얼 게임을 개발하는 업체들이었다. 다만 일부업체는 MMORPG 개발에 3년 이상 소요된 것으로 조사되었다.



개발기간	업체 수	비율
6개월 미만	5	15%
1년 미만	7	21%
2년 미만	13	40%
2년 이상	7	21%
기타	1	3%

그림 II-13. 주력개발 제품에 투입된 기간에 대한 설문 결과

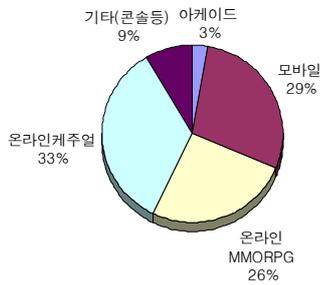
## 2.6.2 개발 계획 중인 장르에 대한 설문 결과

### 1) 개발 계획 중인 장르 현황

o 게임개발업체의 계획 중인 장르는 요즘 가장 인기를 끌고 있는 MMORPG와 캐주얼 게임, 모바일 게임 등이 전체의 88%에 달해 향후 게임 개발 장르의 주류를 이루었다. 업체들이 가장 선호하는 게임장르는 온라인캐주얼 게임으로 33%의 업체가 개발을 계획하고 있고, 그 뒤를 이어서 모바일 게임 개발업체가 29%, MMORPG게임 개발업체는 26%를 차지했다.

o 온라인캐주얼 게임 업체나 모바일 게임 개발업체가 많은 이유는 업체규모나 투자금액, 투입 가능한 인력을 고려한 것으로 조사되었다. 또 짧은 개발 기간과 사용자의 요구에 빠르게 대응할 수 있다는 점도 온라인캐주얼 게임과 모바일 게임의 개발 이유라고 할 수 있다.

o 게임시장 매출의 대부분을 차지하고 있는 MMORPG게임의 개발이 다소 낮은 이유는 방대한 프로젝트를 진행하기위한 예산의 확보에 어려움이 있는 것으로 나타났다.



게임종류	업체 수	비율
아케이드	1	3%
모바일	10	29%
온라인MMORPG	9	26%
온라인캐주얼	12	33%
기타(콘솔등)	3	9%

그림 II-14. 개발 계획 중인 장르에 대한 설문 결과

## 2) 계획중인 장르를 선택한 이유에 대한 현황

o 게임을 개발할 때는 많은 사항이 고려된다. 먼저 만들고자 하는 게임이 어떤 장르의 게임인지를 결정하기 위해서 최소한 게임개발사는 게임을 누가 주로 할 것 인지, 어느 정도 개발기간이 걸리는지, 비용은 얼마나 드는지, 투입될 인원은 얼마정도가 적당한지, 게임유통에 필요한 비용은 어느 정도인지, 게임을 통해 벌어들일 수 있는 수익은 얼마인지 등 많은 사항을 고려하게 된다.

o 이것은 개발한 게임이 시장에 진출 했을 때의 대중성과 수익성을 충분히 고려하기 위한 준비 사항으로 자본주의 사회에서 필수적으로 필요한 기업의 이익창출과도 맞물린 매우 중요한 이유이다. 그래서 대부분의 게임 개발사가 가장 중요하게 생각하는 부분은 대중성과 수익성이라고 볼 수 있다. 조사대상 업체의 대부분이 대중성과 수익성에 초점을 맞추고 게임을 개발한 것으로 조사되었다.



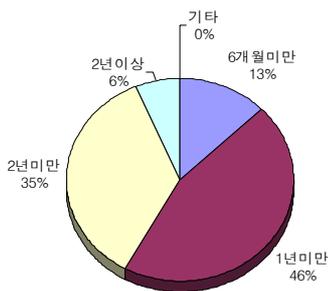
장르선택이유	업체 수	비율
수익성	1	6%
대중성	10	76%
기타	9	18%

그림 II-15. 개발 계획 중인 장르 선택한 이유에 대한 설문 결과

3) 계획한 제품 개발에 투입될 기간에 관한 현황

o 개발기간은 어떤 장르의 게임을 개발하느냐에 따라 차이가 있다. 주로 MMORPG나 온라인캐주얼 게임을 제작하는 업체의 상당수가 1년 미만의 제작기간을 갖는 것으로 조사되었다.

o 전체 업체 중 46%가 1년 미만의 개발기간을 소요할 것으로 예상하고, 1년 이상 2년 미만의 기간을 갖는 업체도 35%에 달했고, 6개월 미만인 업체는 13%, 2년 이상의 개발기간을 갖는 업체도 6%를 차지하였다. 전체적으로 몇 년 전에 비해 개발기간이 장기화되는 추세이다.



투입기간	업체 수	비율
6개월 미만	4	13%
1년 미만	14	46%
2년 미만	11	35%
2년 이상	2	6%

그림 II-16. 개발 계획 중인 제품개발에 투입될 기간에 대한 설문 결과

4) 개발 계획 중인 제품에 투입될 인원 현황

o 투입될 인력은 그래픽디자인분야와 프로그래머가 가장 많이 투입될 것으로 조사되어 향후 전문인력 양성에 있어서 프로그래머와 그래픽디

자이너 양성이 매우 많아야 할 것으로 예상된다.

o 투입될 인원 중 가장 높은 비중을 차지하는 분야는 그래픽 디자인 전문가로 46%의 비중을 차지해 게임의 비주얼이 얼마나 중요한 부분인지를 단적으로 보여주고 있다. 프로그래머의 비중도 34%를 차지해 게임개발의 핵심인력 구성은 여전히 프로그래머와 그래픽 디자이너인 것으로 조사되었다.

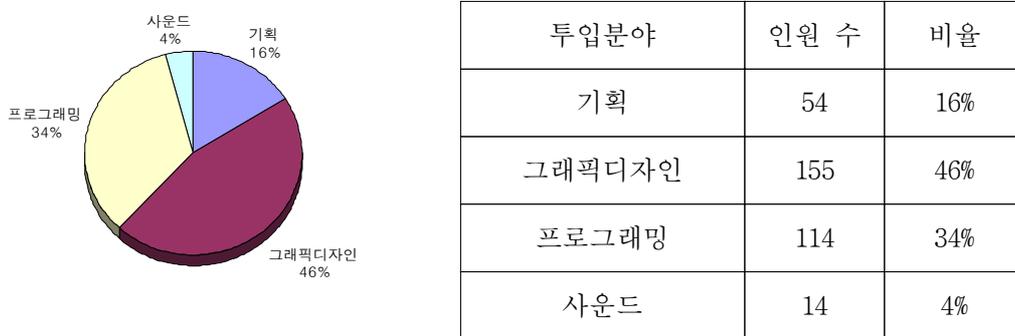


그림 II-17. 개발 계획 중인 제품개발에 투입될 인원예 대한 설문 결과

### 2.6.3 개발 인력 채용에 관한 설문 결과

1) 회사에서 가장 많이 필요로 하는 인력 및 분야별 인력에 대한 설문 결과

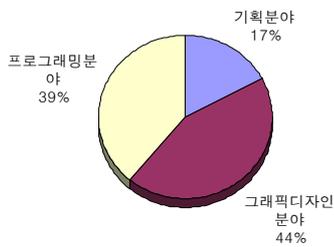
o 게임개발 업체가 가장 많이 필요로 하는 인력은 프로그래머로 64%의 업체가 선택하였다. 또 그래픽디자이너를 필요로 하는 업체도 33%에 달해 업체들의 주요 핵심인력이 어떤 쪽으로 분포되어 있는지 확인할 수 있었다.

o 채용분야별 인력은 그래픽 디자이너 44%(108명), 프로그래머 39%(95명), 기획 분야 17%(42명)정도의 인력을 뽑을 것으로 예상하고 있다. 이것은 위의 게임개발 업체가 필요로 하는 인력의 조사와는 약간 상반된 결과이기는 하나 전반적으로 프로그래머와 그래픽디자이너를 가장 많이 필요로 하는 것으로 조사되었다.



채용분야	업체 수	비율
기획	1	3%
그래픽디자인	11	33%
프로그래밍	21	64%

그림 II-18. 인력 채용 분야별 업체 수의 설문 결과



채용분야	인력 수	비율
기획	42	17%
그래픽디자인	108	44%
프로그래밍	95	39%

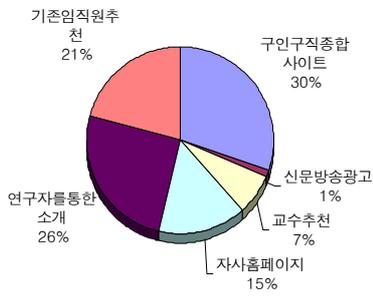
그림 II-19. 인력 채용분야별 인원수에 관한 설문 결과

## 2) 인력 채용방식에 대한 설문 현황

o 업체가 어떤 방식으로 인력을 채용할 지에 대한 질문에는 구인/구직종합사이트를 이용하겠다는 응답이 가장 높은 26%를 차지했다. 그 뒤를 이어 연고자를 통한 소개로 인력을 채용하겠다는 응답이 22%를 차지했고 기존 임직원의 추천을 받아 인력을 채용하겠다는 응답도 18%에 달했다.

o 여기서 연고자나 임직원추천으로 인력을 채용하겠다는 응답이 전체의 30%의 비중을 차지하는데 이것은 검증된 인력을 안전하게 채용하겠다는 것을 의미하기도 한다.

o 대학을 통한 추천이나 교수 추천은 6%로 매우 적었는데 이유는 배출인력의 질적인 신뢰도가 충분하지 않아서라는 인식이 지배적인 것으로 나타났다.



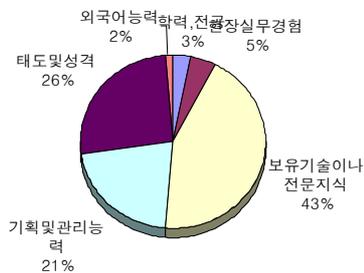
채용방식	업체 수	비율
구인/구직종합사이트	26	30%
신문, 방송광고	1	1%
대학을 통한 교수추천	6	7%
자사 홈페이지	13	15%
연고자를 통한 소개	22	26%
임직원 추천	18	21%

그림 II-20. 인력 채용 방식에 관한 설문 결과(중복선택)

### 3) 인력 선발 기준에 대한 설문 현황

o 선발 기준 중 중요한 것은 예상했던 대로 보유기술이나 전문지식을 갖고 있는 인력이 가장 우선순위가 높아 43%의 선호도를 나타냈고, 그 뒤를 이어 근무자의 근무태도 및 성격이 26%를 차지하여 업체 내에서 타 인력들과의 유대관계를 중요시하게 여겼다. 프로젝트 기획 및 관리능력도 21%에 달했다.

o 반면에 학력이나 출신학교, 전공, 성적 등은 선발기준에 큰 영향을 미치지 않는 것으로 나타났다. 이것은 어떤 학교를 졸업 했느냐 보다는 무엇을 할 줄 아느냐가 더 중요한 인력 선발기준이라는 점을 보여준다.



인력선발기준	업체 수	비율
학력, 전공	2	3%
현장실무경험	3	5%
보유기술, 전문지식	29	43%
기획 및 관리능력	14	21%
태도 및 성격	17	26%
외국어능력	1	2%

그림 II-21. 인력 선발 기준에 관한 설문 결과(중복선택)

## 2.6.4 대학과의 연계 계획에 대한 설문 결과

### 1) 대학과 가능한 산학협력 사업 현황

○ 대학과 업체간의 연계에 대한 설문에서는 많은 업체들이 인턴십 제도를 불신하는 것으로 나타났다.

○ 인턴십을 활용하지 않는 업체에서는 그 이유를 인력의 수준을 낮게 평가하여 인력이 업체 내에서 할 수 있는 역할도 매우 미미한 수준의 업무라고 판단하고 프로젝트 진행에 별 다른 도움이 되지 않을 것으로 평가하고 있었다.

○ 인턴십을 이용한 업체에서는 상당수가 인턴십 사원에 대한 업무 비중을 높게 평가하여 전문 인력의 보조 및 전공 관련 업무를 수행하는 것으로 나타났다.

○ 대학과 가능한 산학협력 사업 중 우선순위가 가장 높은 분야는 인력수급분야로서 46%의 업체가 선택하였고, 공동기술개발이나 기술개발 자문을 구하는 사업도 각각 20%에 달했다. 애로기술지원은 4%, 기존 인력 재교육은 10%의 업체가 선택하였다.

○ 인력수급이 가장 높은 이유는 대부분 대학교육을 이수한 전공자나 관련교육을 이수한 전문 인력을 선발하기 위한 수단으로 대학을 가장 많이 이용한다는 점을 의미한다. 그러나 애로기술지원에 관한 부분은 기업의 기술적 능력을 대학이 따라잡지 못하고 있음을 보여주는 예이다. 이 부분은 대부분 스스로 해결하거나 외국의 게임관련회사의 도움을 받고 있는 것으로 분석된다.



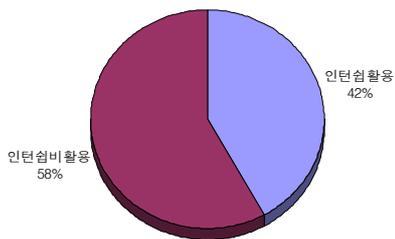
산학협력 사업	업체 수	비율
인력수급	23	46%
공동기술개발	10	20%
애로기술지원	2	4%
기존인력재교육	5	10%
기술개발자문	10	20%

그림 II-22. 대학과 가능한 산학협력 사업에 관한 설문 결과(중복선택)

2) 인턴쉽 제도의 활용 현황 및 인턴사원의 주 업무에 대한 현황

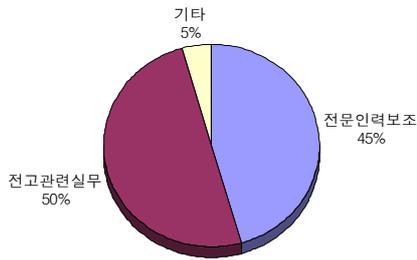
o 인턴쉽 제도를 활용한 경험에는 58%가 없다고 하였으며 조사 대상업체 중 42%가 인턴쉽 제도를 활용한다고 응답하였다.

o 인턴쉽 제도를 활용하는 업체에서 인턴사원들의 주된 업무는 전공 관련 실무에 50%가 투입되고, 전문 인력의 보조업무를 수행하는데 45%, 그리고 기타 사무실 잔업에 5%정도 투입되는 것으로 조사 되었다. 인턴사원들의 주된 업무가 전공 등의 실무에 투입되는 것은 인턴사원에 대한 신뢰도가 상당히 높은 것으로 생각되기 때문일 것으로 판단된다.



인턴쉽	업체 수	비율
활용한다	13	42%
활용하지 않는다	18	58%

그림 II-23. 인턴쉽 제도 활용에 관한 설문 결과



인턴사원의 업무	업체 수	비율
전문인력보조	10	45%
전공관련실무	11	50%
기타	1	5%

그림 II-24. 인턴사원의 주 업무에 관한 설문 결과(활용업체만)

### 3) 인턴사원의 업무수행 만족도에 관한 현황

○ 인턴ships을 활용하고 있는 기업 중 인턴사원의 업무수행에 대하여 47%는 만족하고 있었고 38%는 보통수준이라고 답을 했으며 15%는 불만족이라고 답을 해 전체적으로 인턴사원의 업무 수행도에 상당히 만족해하는 편이라 판단된다.

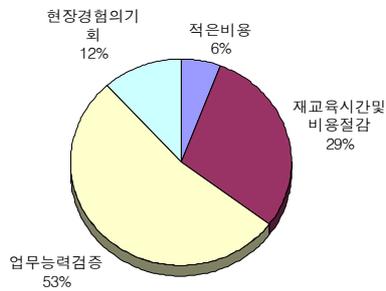
○ 인턴ships 제도를 이용하여 인턴사원을 채용하는 목적에서 채용하기 전에 업무능력을 검증하기 위한 목적이 53%로 가장 큰 것으로 나타났다.

○ 채용 전 업무능력 검증을 위해서 필요하다라는 의견도 29%를 차지했고, 학생들에게 현장경험을 할 수 있는 기회를 주기 위해서 라는 응답과 적은 비용으로 인력 활용을 하기 위해서 라는 답변은 적은 편이었다. 전반적으로 인력채용의 전 단계로 업무능력을 검증하거나 입사 전 재교육 차원에서 인턴ships을 활용하는 것으로 나타났다.



업무수행만족도	업체 수	비율
만족	6	47%
보통	5	38%
불만족	2	15%

그림 II-25. 인턴사원의 업무 수행 만족도에 관한 설문 결과

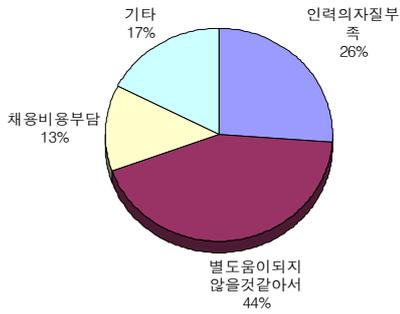


채용목적	업체 수	비율
적은비용	1	6%
재교육시간 및 비용절감	5	29%
업무능력검증	9	53%
현장경험기회	2	12%

그림 II-26. 인턴사원의 채용 목적에 관한 설문 결과

#### 4) 인턴쉽 제도를 활용하지 않는 이유에 관한 현황

○ 인턴쉽 제도를 활용하지 않는 이유에 대해서는 44%의 업체가 별 도움이 되지 않는 것 같다는 응답을 하였다. 대학의 인력에 대한 신뢰도가 많이 낮다는 점을 단적으로 보여주고 있다. 또 인력의 자질이 부족할 것이라는 의견, 채용비용이 부담이 될 것이라는 의견도 각각 26%와 17%를 차지하여 인력에 대한 신뢰도와 인건비부담에 대한 것이 가장 큰 이유인 것으로 나타났다.



비 활용 목적	업체 수	비율
인력의 자질부족	6	26%
별 도움이 되지 않을 것 같아서	10	44%
채용 비용부담	3	13%
기타	4	17%

그림 II-27. 인턴십 제도를 활용하지 않는 이유에 관한 설문 결과

### 5) 인턴사원의 근무기간에 관한 현황

○ 인턴사원을 채용 했을 경우 적당한 근무기간은 6개월 이상 1년 미만의 기간을 선택한 업체가 48%였다. 3개월 이상 6개월 미만 업체도 39%에 달해 전체의 87%가 1년 미만의 기간을 근무기간으로 선택하였다.

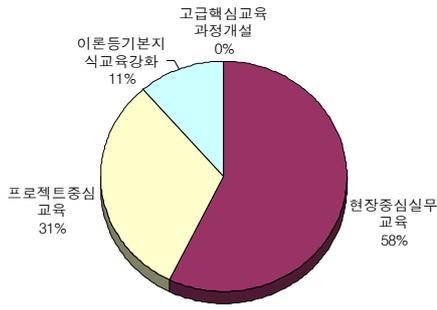


근무기간	업체 수	비율
3개월 이상	12	39%
6개월 이상	15	48%
1년 이상	3	10%
기타	1	3%

그림 II-28. 인턴사원의 근무기간에 관한 설문 결과

### 6) 대학의 전문인력 양성 시 강조되기를 바라는 사항

○ 업체에서 대학의 전문인력 양성을 위해 강조되기를 바라는 점은 현장 중심 실무교육 강화를 58%로 가장 많이 선택했고, 프로젝트 중심의 교육을 31%로 선택했다. 이론 등 기본지식 교육 강화는 11%에 그쳤다. 프로젝트 중심 교육은 실무위주의 진행형 교육을 의미하는 것으로 현장 중심 실무교육과 밀접한 관계가 있는 것으로 기업이 대학에서 교육하여 주기를 바라는 매우 중요한 부분이다.



전문인력 양성을 위해 강조되기를 바라는 점	업체 수	비율
고급핵심 교육과정	0	0%
현장중심 실무교육	20	58%
프로젝트 중심교육	11	31%
기본지식 교육강화	4	11%

그림 II-29. 대학의 전문인력 양성 시 강조 분야에 관한 설문 결과(중복선택)

## 2.7 국내 게임 산업 전문 인력 수요 분석 및 인력 수급 전망

### 2.7.1 인력 수요 분석

○ 본 통계는 게임업체 356개를 대상으로 조사한 2005년 게임백서를 인용한 결과로서 게임 인력은 2001년 23,594명(업체당 평균17명), 2002년 33,870명(18.2명), 2003년 39,104명(18.9명), 2004년 47,051명(19.1명)으로 증가세를 보이고 있다.

○ 직종별 선호 전공은 컴퓨터프로그래머, H/W 개발, 시스템엔지니어의 경우에는 60% 정도가 전자/공학계열을 선호하고, 그래픽디자이너의 경우 67.0%가 예술/디자인계열을 선호하는 것으로 나타나고 있다.

표 II-27. 게임 산업체의 직종별 선호 전공

구분	전자/공학계열	인문/사회계열	예술/디자인계열	경영/마케팅계열	게임관련학과	게임전문대학원졸업	구별하지 않음	기타	합계
게임PD	5.7%	11.8%	2.7%	4.6%	25.1%	4.2%	44.5%	1.5%	100.0%
그래픽디자이너	2.9%	0.4%	67.0%	0.4%	7.2%	2.2%	18.6%	1.4%	100.0%
컴퓨터프로그래머	59.4%	0.4%	1.1%	0.4%	13.1%	4.2%	20.5%	1.0%	100.0%
시나리오	2.2%	25.0%	2.2%	0.7%	14.7%	1.5%	50.7%	2.9%	100.0%
사운드크리에이터	7.0%	0.8%	27.3%	0.0%	7.0%	2.3%	52.3%	3.2%	100.0%
H/W개발	58.5%	0.0%	0.0%	0.0%	6.5%	2.4%	30.1%	2.4%	100.0%
시스템엔지니어	60.0%	0.6%	0.0%	0.0%	7.5%	1.9%	28.8%	1.3%	100.0%
게임운영자(GM)	7.1%	2.4%	0.0%	7.7%	14.9%	4.8%	61.9%	1.2%	100.0%
홍보/마케팅	1.3%	8.6%	0.4%	50.0%	0.9%	0.4%	36.6%	1.7%	100.0%
일반관리직	1.7%	10.8%	0.0%	42.9%	2.2%	0.0%	40.3%	2.2%	100.0%

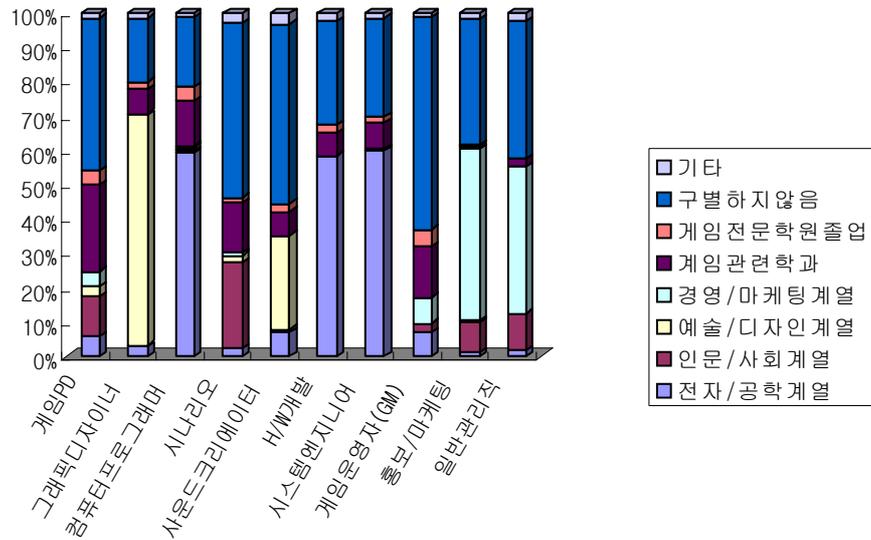


그림 II-30. 게임 산업체의 직종별 선호 전공

o 직종별 선호 학력은 대체적으로 '구별하지 않음'이 33.3%~57.6%로 매우 높은 비율을 보이며, 특히 시나리오, 사운드크리에이터, 게임운영자의 경우 50% 이상이 선호하는 학력을 구별하지 않고 있으나, 다른 항목들과 함께 고려해 보면, 대졸인력의 선호도가 40% 안팎이고 고졸이하학력 선호비율이 매우 낮은 것으로 보아 일반적으로 대졸이상의 학력을 선호함을 알 수 있다.

표 II-28. 게임 산업체의 직종별 선호 학력

구분	고졸	전문대졸	대졸	석사이상	구별하지 않음	기타	합계
게임PD	0.4%	7.6%	42.7%	1.1%	45.4%	2.7%	100.0%
그래픽디자이너	0.7%	16.8%	34.8%	0.7%	44.8%	2.2%	100.0%
컴퓨터프로그래머	0.7%	9.2%	44.2%	2.1%	41.7%	2.1%	100.0%
시나리오	0.7%	5.9%	37.0%	1.5%	52.6%	2.2%	100.0%
사운드크리에이터	0.8%	7.9%	30.7%	0.8%	57.5%	2.4%	100.0%
H/W개발	0.8%	9.5%	38.9%	2.4%	46.0%	2.4%	100.0%
시스템엔지니어	0.6%	8.8%	42.5%	2.5%	43.8%	1.9%	100.0%
게임운영자(GM)	1.8%	10.6%	27.1%	1.2%	57.6%	1.8%	100.0%
홍보/마케팅	0.4%	7.4%	55.0%	1.7%	33.3%	2.2%	100.0%
일반관리직	2.6%	12.9%	42.2%	1.7%	37.9%	2.6%	100.0%

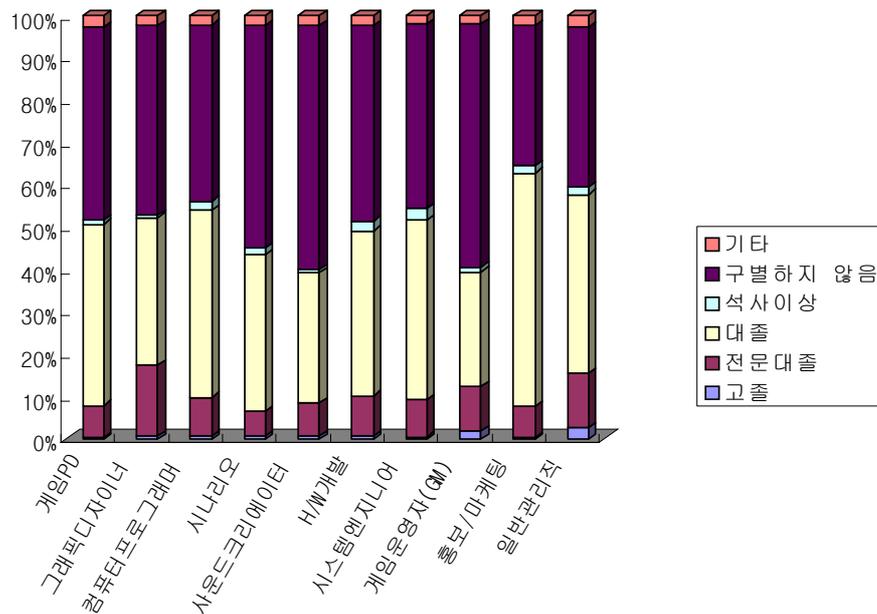


그림 II-31. 게임 산업체의 직종별 선호 학력

o 게임업체에서는 선호 인력에 대한 중요 기준으로 대체적으로 학력보다 개인적인 능력과 경력을 중요시 하고 있다. 게임제작에 대한 직무에 있어서는 포트폴리오 및 경력이 중요한 기준으로 40%안팎의 높은 비율을 차지하고, 특히 그래픽디자이너의 경우 포트폴리오가 63.2%로 매우 높게 나타나며, 컴퓨터프로그래머, H/W개발, 시스템엔지니어는 경력을 가장 중요시하고 있음을 알 수 있다.

표 II-29. 게임 산업체의 선호 인력에 대한 중요 기준

구분	포트폴리오	학력	경력	태도 성격	외모	구별하지 않음	기타	합계
게임PD	40.2%	1.1%	40.6%	11.1%	0.0%	5.0%	1.9%	100.0%
그래픽디자이너	63.2%	0.0%	24.6%	6.8%	0.0%	4.3%	1.1%	100.0%
컴퓨터프로그래머	31.3%	1.8%	51.2%	9.6%	0.0%	4.6%	1.4%	100.0%
시나리오	47.0%	0.0%	35.8%	6.0%	0.0%	9.0%	2.2%	100.0%
사운드크리에이터	44.6%	0.0%	37.7%	6.2%	0.0%	8.5%	3.1%	100.0%
H/W개발	29.3%	0.0%	50.4%	8.1%	0.0%	8.9%	3.3%	100.0%
시스템엔지니어	25.3%	1.9%	53.8%	10.1%	0.0%	6.3%	2.5%	100.0%
게임운영자(GM)	16.6%	0.6%	42.6%	30.8%	0.0%	7.7%	1.8%	100.0%
홍보/마케팅	14.8%	2.2%	52.2%	22.2%	1.7%	5.2%	1.7%	100.0%
일반관리직	11.3%	3.0%	41.7%	33.0%	0.0%	9.6%	1.3%	100.0%

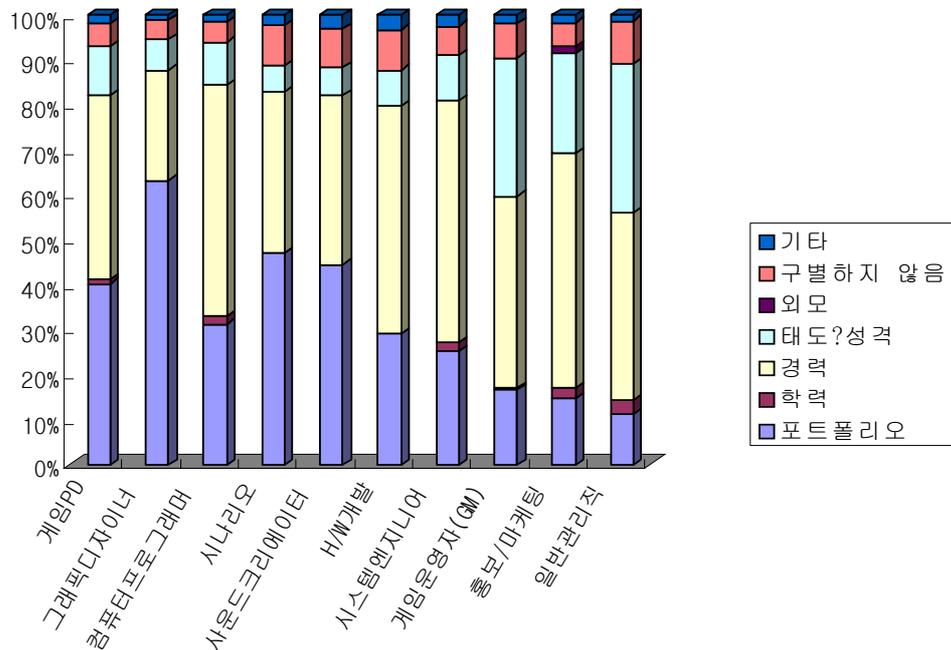


그림 II-32. 게임 산업체의 선호 인력에 대한 중요 기준

o 플랫폼별 필요 인력으로는 전반적으로 신규업체가 늘어남에 따라 그래픽디자이너와 컴퓨터프로그래머 인력을 상당히 필요로 하고 있다. 온라인 게임, 비디오게임, 모바일 게임의 경우 게임PD의 필요인력 요구가 많아지고 있으며, PC게임의 경우에는 홍보/마케팅 인력이 22.6%로 타 플랫폼에 비해 높은 요구를 보이고 있다.

표 II-30. 게임 산업체의 플랫폼별 필요 인력

구분	게임 PD	그래픽 디자이너	컴퓨터 프로그래머	시나리오	사운드 크리에이터	H/W 개발	시스템 엔지니어	게임 운영자 (GM)	홍보 마케팅	일반 관리직	합계
온라인 게임	12.0%	26.5%	23.7%	3.0%	2.9%	1.5%	3.3%	12.8%	8.4%	5.8%	100%
PC게임	7.5%	24.5%	34.0%	3.8%	0.0%	0.0%	0.0%	1.9%	22.6%	5.7%	100%
아케이드 게임	7.5%	14.1%	23.6%	4.0%	3.0%	11.6%	8.0%	3.0%	12.1%	13.1%	100%
비디오게임	19.5%	19.5%	22.0%	7.3%	2.4%	4.9%	2.4%	0.0%	9.8%	12.2%	100%
모바일 게임	14.2%	20.2%	36.3%	3.0%	1.3%	0.7%	2.6%	1.7%	12.9%	7.1%	100%

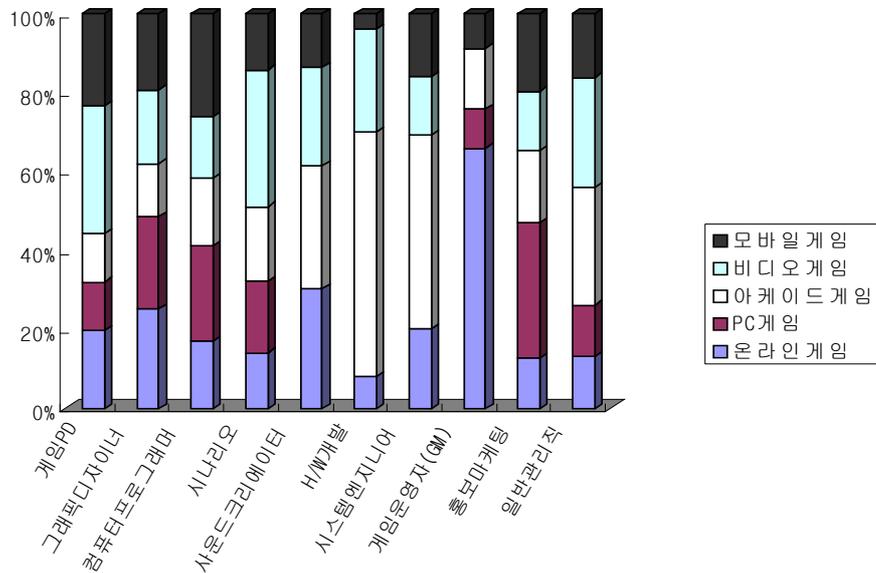


그림 II-33. 게임 산업체의 플랫폼별 필요 인력

o 업체의 인력 확보 어려움 요인은 '경력직 자질 부족'이 26.5%, '인력정보부족'이 19.9%, '저임금 회피'가 17.7%로 높게 나타났으며, '정규직 자질부족' 9.1%, '학원졸업생 자질 부족' 4.3%, '정규 졸업생 부족' 4.2%순으로 나타났다. 경력직 자질 부족을 해결하기 위한 특별교육프로그램 및 재교육프로그램 마련이 시급한 것으로 판단된다.

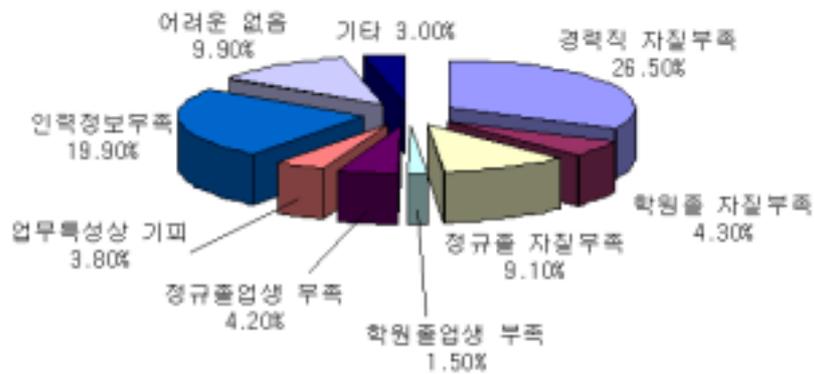


그림 II-34. 게임 산업체의 인력 확보 어려움 요인

## 2.7.2 인력 수급 전망

○ 직종별 수요증감 전망을 살펴보면 게임PD, 그래픽디자이너, 컴퓨터프로그래머, 홍보/마케팅의 경우 향후 3년간 '20~30% 증가할 것'이라는 비율이 40%이상 높게 나왔으며, 특히 컴퓨터프로그래머의 경우 필요인력의 수요가 높은 만큼 '50%이상 증가할 것'이라는 전망이 11.6%로 타 직종보다 높고 사운드크리에이터, 시스템엔지니어는 '20~30% 감소할 것'이라는 비율이 높게 나타나고 있다.

표 II-31. 게임 산업체의 직종별 수요 증감 전망

구분	50%이상 감소	20~30% 감소	변화 없음	20~30% 증가	50%이상 증가	합계
게임PD	2.0%	4.3%	43.4%	43.8%	6.6%	100.0%
그래픽디자이너	2.6%	6.3%	36.8%	48.0%	6.3%	100.0%
컴퓨터프로그래머	2.6%	3.7%	34.3%	47.8%	11.6%	100.0%
시나리오	3.3%	5.3%	53.9%	32.2%	5.3%	100.0%
사운드크리에이터	2.1%	9.1%	65.7%	18.9%	4.2%	100.0%
H/W개발	2.9%	6.6%	70.1%	17.5%	2.9%	100.0%
시스템엔지니어	3.0%	7.3%	61.8%	23.0%	4.8%	100.0%
게임운영자(GM)	2.3%	5.8%	52.6%	32.4%	6.9%	100.0%
홍보/마케팅	1.4%	4.5%	47.7%	41.9%	4.5%	100.0%
일반관리직	2.8%	7.3%	68.8%	19.7%	1.4%	100.0%

○ 온라인 게임 분야의 인력수요는 그래픽디자이너가 '20~30% 증가'의 비율이 50.4%로 높게 나타나고, 게임PD, 컴퓨터프로그래머, 게임운영자의 경우에도 40%이상으로 '20~30%증가'하리라 전망하고 있으며, 특히 컴퓨터프로그래머의 경우 '50%이상 증가'하리라는 기대가 11.5%로 높게 나타나 지속적인 성장 흐름과 온라인 게임 발전에 대한 기대심리가 계속되고 있음을 알 수 있다.

표 II-32. 온라인 게임 분야의 인력수요 전망

구분	50%이상 감소	20~30% 감소	변화 없음	20~30% 증가	50%이상 증가	합계
게임PD	1.7%	5.2%	40.5%	46.6%	6.0%	100.0%
그래픽디자이너	3.3%	7.3%	33.3%	50.4%	5.7%	100.0%
컴퓨터프로그래머	3.3%	2.5%	34.4%	48.4%	11.5%	100.0%
시나리오	4.2%	7.0%	50.7%	32.4%	5.6%	100.0%
사운드크리에이터	1.5%	8.8%	61.8%	22.1%	5.9%	100.0%
H/W개발	1.6%	6.6%	70.5%	19.7%	1.6%	100.0%
시스템엔지니어	2.5%	8.6%	59.3%	25.9%	3.7%	100.0%
게임운영자(GM)	2.1%	4.3%	41.5%	43.6%	8.5%	100.0%
홍보/마케팅	2.0%	2.9%	52.0%	39.2%	3.9%	100.0%
일반관리직	2.0%	6.1%	73.7%	18.2%	0.0%	100.0%

o 모바일 게임의 인력수요는 게임PD, 그래픽디자이너, 컴퓨터프로그래머, 홍보/마케팅의 경우 '20 ~ 30% 증가'의 응답이 40%이상으로 높게 나타났고, 그 외 직종에서는 변화 없는 것으로 분석되며, 고사양의 휴대폰 단말기의 확산과 더불어 모바일 게임 개발을 위한 게임제작과 관련된 직종의 인력에 대한 기대가 높은 것으로 보인다.

표 II-33. 모바일 게임 분야의 인력수요 전망

구분	50%이상 감소	20~30% 감소	변화 없음	20~30% 증가	50%이상 증가	합계
게임PD	1.1%	4.3%	38.0%	48.9%	7.6%	100.0%
그래픽디자이너	1.0%	7.3%	39.6%	43.8%	8.3%	100.0%
컴퓨터프로그래머	1.0%	7.3%	34.4%	49.0%	8.3%	100.0%
시나리오	0.0%	4.1%	51.0%	38.8%	6.1%	100.0%
사운드크리에이터	0.0%	13.3%	64.4%	20.0%	2.2%	100.0%
H/W개발	2.3%	7.0%	76.7%	11.6%	2.3%	100.0%
시스템엔지니어	2.0%	8.0%	68.0%	18.0%	4.0%	100.0%
게임운영자(GM)	0.0%	10.0%	64.0%	20.0%	6.0%	100.0%
홍보/마케팅	0.0%	6.8%	42.5%	46.6%	4.1%	100.0%
일반관리직	2.9%	10.1%	60.9%	21.7%	4.3%	100.0%

o PC 게임에서는 타 분야의 성장과는 반대로 PC게임 시장의 하락세에 상당히 위축된 심리가 반영되어 향후 3년간 인력수요에 거의 변화가 없을 것으로 예상되며, 다만 그래픽디자이너와 컴퓨터프로그래머의 인력이 다소 증가할 것으로 보인다.

표 II-34. PC 게임 분야의 인력수요 전망

구분	50%이상 감소	20~30% 감소	변화 없음	20~30% 증가	50%이상 증가	합계
게임PD	0.0%	7.1%	64.3%	21.4%	7.1%	100.0%
그래픽디자이너	0.0%	0.0%	58.3%	41.7%	0.0%	100.0%
컴퓨터프로그래머	0.0%	0.0%	41.7%	25.0%	33.3%	100.0%
시나리오	0.0%	0.0%	75.0%	25.0%	0.0%	100.0%
사운드크리에이터	0.0%	0.0%	100.0%	0.0%	0.0%	100.0%
H/W개발	0.0%	0.0%	100.0%	0.0%	0.0%	100.0%
시스템엔지니어	0.0%	0.0%	85.7%	14.3%	0.0%	100.0%
게임운영자(GM)	0.0%	0.0%	71.4%	28.6%	0.0%	100.0%
홍보/마케팅	0.0%	0.0%	58.3%	33.3%	8.3%	100.0%
일반관리직	0.0%	0.0%	84.6%	15.4%	0.0%	100.0%

o 아케이드 게임에서의 인력 수요는 향후 조심스러운 기대와 부정적 전망이 서로 대립되어 그래픽디자이너와 컴퓨터프로그래머의 경우 '20~30% 증가'하리라는 예상이 20%이상 나타났으나 다른 직무의 경우 '50% 이상 감소'하리라는 전망이 다른 게임분야에 비해 상당히 높게 나타나고 있다.

표 II-35. 아케이드 게임 분야의 인력수요 전망

구분	50%이상 감소	20~30% 감소	변화 없음	20~30% 증가	50%이상 증가	합계
게임PD	9.5%	0.0%	66.7%	19.0%	4.8%	100.0%
그래픽디자이너	8.3%	0.0%	33.3%	50.0%	8.3%	100.0%
컴퓨터프로그래머	8.0%	0.0%	32.0%	52.0%	8.0%	100.0%
시나리오	12.5%	6.3%	56.2%	18.8%	6.3%	100.0%
사운드크리에이터	12.5%	0.0%	68.7%	18.8%	0.0%	100.0%
H/W개발	10.0%	5.0%	45.0%	30.0%	10.0%	100.0%
시스템엔지니어	10.0%	0.0%	50.0%	30.0%	10.0%	100.0%
게임운영자(GM)	14.3%	0.0%	57.1%	21.4%	7.1%	100.0%
홍보/마케팅	4.3%	4.3%	39.1%	43.5%	8.7%	100.0%
일반관리직	8.0%	4.0%	56.0%	32.0%	0.0%	100.0%

o 비디오 게임은 현재 시장이 점차 개척·형성되고 있는 가운데 다른 플랫폼과 비교하여 높은 성장이 예상되며, 특히 컴퓨터프로그래머의 경

우 '20~30% 증가'라는 응답이 37.5%, '50%이상 증가'라는 응답이 25%로 인력에 대한 수요와 성장의 기대가 높게 나타나고 있다.

표 II-36. 비디오 게임 분야의 인력수요 전망

구분	50%이상 감소	20~30% 감소	변화 없음	20~30% 증가	50%이상 증가	합계
게임PD	0.0%	0.0%	37.5%	50.0%	12.5%	100.0%
그래픽디자이너	0.0%	0.0%	37.5%	62.5%	0.0%	100.0%
컴퓨터프로그래머	0.0%	0.0%	37.5%	37.5%	25.0%	100.0%
시나리오	0.0%	0.0%	66.7%	33.3%	0.0%	100.0%
사운드크리에이터	0.0%	16.7%	66.7%	0.0%	16.7%	100.0%
H/W개발	0.0%	20.0%	60.0%	20.0%	0.0%	100.0%
시스템엔지니어	0.0%	20.0%	60.0%	0.0%	20.0%	100.0%
게임운영자(GM)	0.0%	16.7%	83.3%	0.0%	0.0%	100.0%
홍보/마케팅	0.0%	12.5%	50.0%	37.5%	0.0%	100.0%
일반관리직	0.0%	52.0%	75.0%	0.0%	0.0%	100.0%

## 2.8 게임 산업 전문 인력 양성 관련 정책

전문 인력 양성이라는 부분은 그 특성 상 기업이나 학교 수준에서의 단기적인 교육 시스템 개선이나 진행을 통해 결과가 바로 얻어지는 분야라기보다는 정부 단위의 중장기적인 목표 설정이나 정책 결정이 이루어진 후 상당 기간의 꾸준한 실천을 통하여 성과를 얻어낼 수 있는 분야이다. 즉, 단기적이고 지엽적인 수준의 실행 방안보다는 광범위하고 거시적인 정책적인 안목과 전략이 필요한 부분인 셈이다.

국내의 경우, 90년대 말 본격적인 인터넷 시대 개막과 함께 게임산업에 대한 관심이 높아지며 게임 인력 양성을 위한 정부의 정책이 다양한 부처와 기관을 통해 수립되어 진행되고 있는 상황이다. 이번 장에서는 게임 산업 전문 인력 양성을 위해 정부가 현재까지 추진해온 정책의 내용과 앞으로의 정책 변화 내용 등을 살펴보도록 하겠다.

### 2.8.1 정부의 게임 인력 양성 계획 추진 현황 및 정책 방향

최근의 정부의 게임 인력 양성은 정보통신 산업의 발전 및 디지털 콘텐츠 산업의 육성 관점에서 이루어지고 있는 것이 큰 특징이라고 할 수 있다. 게임산업의 주무부서라 할 수 있는 문화관광부는 게임 인력 양성의 중요성을 깊이 인식하고 산하 기관인 한국게임산업개발원을 중심으로 작성한 “게임산업진흥 중장기계획 (2003년~2007년)”의 주요 사업 목표 중 한 부분으로써 게임 인력 양성에 노력해오고 있다. (표 참조)

표 II-37. “게임산업진흥 중장기계획 (03년~07년)”에서의 인력 양성 관련 내용

6대 중점 추진 사항	<ul style="list-style-type: none"> <li>- 게임 산업 인력 수급분석 및 인력DB 구축</li> <li>- 산학 협력 인력 양성 시스템 구축</li> <li>- 게임 전문인력 연수·해외교류 기회 확대</li> <li>- 대학 학계 연구기반 및 교육역량 강화지원</li> <li>- 게임 아카데미 기능 재정립</li> <li>- 게임 인력의 취업 환경 개선</li> </ul>
----------------	---

이는 연차적인 계획에 따라 다양한 형태의 사업으로 실제 진행되었는데 다음의 표는 2005년 국내의 게임 관련 정부 산하 기관인 한국게임산업개발원이 2005년 한 해 동안 인력 양성을 위해 진행한 사업의 개략적인 내용이다.

표 II-38. 2005년 한국게임산업개발원 게임 인력 양성 사업 개요

사 업 명	주요 실적
산학연계 인력양성 현장화	<ul style="list-style-type: none"> <li>○ 게임연구센터(GRC) 지정 및 지원               <ul style="list-style-type: none"> <li>- 덕성여대 등 10개 대학</li> </ul> </li> <li>○ 게임 인력 교류 활성화(학술지원)               <ul style="list-style-type: none"> <li>- 한국해세학회, 한국디지털콘텐츠경영포럼, 한국게임마케팅포럼</li> </ul> </li> <li>○ 청소년교육과정 지원('04.11~'05.7)</li> <li>○ 지역 인력양성기관 육성               <ul style="list-style-type: none"> <li>- 강릉, 대구, 대전, 부산 4개 지역</li> </ul> </li> <li>○ 게임인력 해외교류 활성화(해외연수)               <ul style="list-style-type: none"> <li>- 연수기간 : '05.7.18~'8.12</li> <li>- 연수장소 : 카네기멜론대학 ETC</li> <li>- 연수인원 : 총 11명</li> </ul> </li> <li>○ 게임분야 우수인력 조기발굴(고교지원)</li> <li>○ 게임교재 6종 개발</li> </ul>
게임전문인력양성	<ul style="list-style-type: none"> <li>○ 게임정규과정(2년제) 4기 50명 배출 (누계 취업률 90%)</li> <li>○ 게임전문가과정 총 4개과정 94명 교육               <ul style="list-style-type: none"> <li>- 비디오게임 개발자과정 실시 (수료 : 8명, '05.4)</li> <li>- 게임마케팅전문가과정 실시</li> </ul> </li> </ul>

	(수료 : 19명, 05'5) - 게임분야 공무원 교육과정 실시 (수료 : 51명, 05'5) - 게임분야 실업계교과 연수과정 실시 (수료 : 16명, 05'7)
게임원격교육 활성화	○ 게임원격강의 콘텐츠 8차 9과목 개발중 - 9차 9과목은 10월부터 개발 예정 ○ 게임원격교육 8, 9기 134과목 운영 - 수강생 3,148명 ○ 전국 44개 대학과 게임원격교육 MOU체결 - 주성대, 부산정보대 학점연계 ○ KOCCA와 통합 LMS 구축 - 개발 전과정 협의 및 공동 진행

이 외에도 정보통신부는 “게임 제작 동아리 지원 사업” 및 정보통신 분야 학과에 관련한 다양한 지원 사업을 통하여, 노동부는 게임 관련 국가검정 자격 제도의 도입을 통하여, 산업자원부는 게임사관학교의 설립 지원 등을 통하여 정부의 각 부서가 게임 전문 인력 양성의 중요성을 인식하고 다양한 정책 사업을 진행하고 있는 상황이다.

특히, 문화관광부는 최근 빠르게 변화하고 있는 세계 게임 산업의 추세와 국내 시장의 변화, 이에 따른 국내 게임 전문 인력의 공급 및 수요 형태 변화에 적응하기 위하여 “2010 게임산업 전략위원회” 를 구성하고 “창의적인 게임인력 양성” 을 목표로 2010년까지의 중장기 게임 전문 인력 양성 계획을 수립했다. 그 내용을 세부적으로 살펴보면 다음과 같다.

표 II-39. “2010 게임산업 실행전략” 에서의 게임 인력 양성 관련 내용

목표	소 과 제	세부 과제 및 주요 내용
창의적인 게임인력 양성	핵심 전문인력 양성 강화	- 게임 전문 인력의 국제 역량 강화 - 핵심 부분 및 미래 게임기술 연구인력 양성 - 교육 기관별 핵심 교육과정 표준화 - 업계의 기존 인력 재교육 지원 - 취약 분야 교육 기반 강화 - 게임아카데미 역할 및 기능 재정립
	연구기반 및 교육역량 강화	- 게임교육기관 평가제도 구축 및 우수기관 선정 지원 - 국내외 인력 양성 관계기관간 협력 네트워크 구축 - 중장기 게임 인력 수급 균형 정책 수립
	실행형 산학협력 모델 구축	- 산 학 공동 프로젝트 활성화 - 산학연계 게임 공모전 위상 강화 및 통합 - 인턴십 제도 및 주문식 교육 모델 정립
	취업환경 개선 및 내실화	- 게임 산업에 특화된 공동 채용 박람회 개최 - 실효성 있는 국가 검정 제도 개선 - 병역특례제도의 활성화 및 효율화 제고

## 2.8.2 핵심 전문인력 양성 강화

### 1) 게임 전문 인력의 국제 역량 강화

#### □ 목적/필요성

- 국내 게임산업의 국제경쟁력을 위해 국제적 감각과 역량을 가진 전문 인력의 필요성 증가
- 선진화된 해외 교육 프로그램의 효과적인 수용을 위한 체계적인 해외 연수 프로그램 필요

#### □ 현황 및 문제점

- 해외 마케팅 및 퍼블리싱 전문 인력의 수요 절대 부족은 국내 게임산업 성장에 장애요인으로 발생
- 해외 연수에 대한 특성화 및 장기적인 목표 설정 미흡

#### □ 사업 내용

- 해외 연수 및 교류 프로그램 확대 추진
  - 교수 전문요원 해외연수프로그램 추진
    - 학계, 업계 핵심전문가 대상 “교수요원 해외연수” 기획 부여
    - \* 기획, 마케팅 등 취약분야에 대한 전문 교수인력 확보에 중점
  - 업계 전문인력의 단기 해외교류프로그램 추진
    - 해외 전문 연구기관 단체 대상 단기 연수 교류 및 N/W 강화
    - 해외 게임산업 현황, 마케팅 기법, 기획 개발과정 등 체험 기획 부여
- 대학의 게임 수출 마케팅 전문인력 교육 강화를 위한 전공 설립 및 표준 교육과정 지원
  - 게임 수출 마케팅 교육을 전담하는 게임 마케팅 전공 설립을 추진

- 게임 마케팅 교수요원 양성과 관련 표준 교육과정 및 표준 교재개발을 지원
- 게임산업에 대한 지식과 기술, 수출에 필요한 언어와 경험을 겸비한 게임 수출 마케팅 전문인력 양성

□ 연도별 추진체계

- o 2006~2007 : 해외 연수 및 교류 프로그램 재정비, 해외 마케팅 교육을 위한 교원 확보 및 교재 개발
- o 2008~2010 : 해외 연수 프로그램 안정화, 해외 마케팅 교육 필수 과목화 추진

□ 추진주체

- o 한국게임산업개발원, 국내외 인력관련 유관기관, 게임업계 및 게임학회, 교육인적자원부

□ 소요 예산 (단위 : 억원)

연 도	2007년	2008년	2009년	2010년	합 계
예 산	10	10	12	12	44

2) 핵심 부분 및 미래 게임기술 연구인력 양성

□ 목적/필요성

- o 창의적인 게임 기획과 게임 디자인을 위한 인력 양성 프로그램을 연구 및 보급
- o 기술 발전에 근거한 차세대 네트워크 및 차세대 3D 그래픽스 기술 개발 기술 인력 양성

□ 현황 및 문제점

- o 게임 업계에서 요구하고 있는 게임 기획에 필요한 창의성에 중점을 둔 교육이 부족

- 그래픽 기술의 발전에 대응하는 최신의 고급 기술을 가진 인력의 부재

□ 사업 내용

- 창의적 기획 인력 양성을 위한 프로그램 및 커리큘럼 개발
  - 업계의 요구 및 시장 트렌드에 따른 창의적 기획 인력의 핵심 역량에 대한 분석을 통한 육성 프로그램 개발
  - 해외 선진기관에서 진행되는 창의적 기획 프로그램 분석
  - 국내의 CT대학원에서 창의성 있는 게임기획과 스토리텔링에 대한 교육 적용 및 커리큘럼 개발
- 산학협력을 통한 차세대 3D 그래픽스 기술 및 AI/Physics 개발 체계 구축
  - 고급 그래픽스 기술 개발체계 구축
    - NVidia, ATI등 외국 그래픽 카드 제조사와의 협력체계 구축
    - 차세대 그래픽스 하드웨어 및 셰이더에 기반한 렌더링 및 애니메이션 기술 개발을 위한 산학 컨소시엄 지원
  - AI / Physics 산학연 공동연구 개발체계 구축
    - AI / Physics와 관련한 국내 대학 및 연구소의 고급 연구 인력의 활용
    - 산학협력을 통해 대학 및 연구소가 보유한 고급 기술을 게임 AI로 적용 연구 추진
- 차세대 네트워크 기술의 특허 확보 및 관련 인력 양성
  - 유비쿼터스 시대와 적합한 대규모 동시접속을 이룩할 수 있는 네트워크 발전 방향을 수용하는 특허 확보 및 통신기술자 양성

□ 연도별 추진체계

- 2006~2007 : 창의적인 기획 인력 양성, 차세대 기술 인력 양성에 대한 로드맵 작성

- 2008~2009 : 기획 및 기술 인력 양성체제 구축, 서버 기술의 특허 확보
- 2010 이후 : 고급 기획 및 기술 인력 배출

□ 추진주체

- 한국게임산업개발원, 인력관련 유관기관, 게임 업계 및 협회·단체, 관련 학계

□ 소요 예산 (단위 : 억원)

연 도	2007년	2008년	2009년	2010년	합 계
예 산	6	7	8	10	31

3) 교육 기관별 핵심 교육과정 표준화

□ 목적/필요성

- 산업계의 인력 수요 및 향후 게임 산업 발전 방향에 부응하는 표준적인 교육 과정의 개발 및 보급을 통하여 우수한 수준의 인력 양성이 가능한 기반 구축

□ 현황 및 문제점

- 국내 게임 교육 기관의 양적 증가에도 불구하고 교육과정의 체계화 미흡
- 교육 내용, 방식, 목표, 효과 등에 대한 불명확성 및 검증 체계 미흡으로 인해 게임 인력 양성 시스템 자체에 대한 업계의 불신 지속

※ 2004년 이후 단계적으로 고시되고 있는 교육인적자원부의 게임 관련 “표준교육과정”의 경우는 일부 비정규 교육 기관의 학점은행제 운영을 위한 기준으로 활용되는 반면, 최근 교육 부분의 자율화 풍조에서 대학 등 정규 교육 기관의 교육 시스템에 대한 영향력에는 한계가 있는 것으로 평가

□ 사업 내용

- 게임 산업계의 인력 요구 사항이 반영된 실무형 프로젝트 교육 과정 개발 및 보급
  - 게임 산업계의 인력 수요에 대한 교육 요구 사항 조사 및 분석을 기반으로 '실무형 프로젝트 교육 모델' 개발
    - 프로그래밍, 그래픽, 사운드, 기획 등 각 업무별 최소 요구 사항 등이 반영된 구체적인 실습 모형 제시
  - 대학, 고등학교, 학원 등 교육 기관별 특성이 고려된 구체적인 형태의 '실무형 프로젝트 교육 과정' 개발 및 보급
  
- 교육인적자원부 표준교육과정 운영 방안 개선
  - 한국교육개발원의 게임그래픽 분야 등 미 개설 분야에 대한 표준교육과정 개설 및 운영의 지속 진행
  - 기 고시된 게임프로그래밍 및 게임디자인(기획) 부분의 표준교육과정에 대한 게임 산업계의 지속적인 의견 수렴, 수정안을 지정 및 고시
  - 표준교육과정의 보급 및 활용을 위해 협회나 정부 차원의 게임 교육 기관 검증 및 우수 교육 기관 지원 추진

□ 연도별 추진체계

- 2006~2008 : 실무형 프로젝트 교육 모델 개발, 게임그래픽 분야 표준교육과정 개설, 기 고시된 표준교육과정에 대한 확장 및 수정안 고시
- 2009~2010 : 교육 기관별 실무형 프로젝트 교육 과정 개발 및 보급

□ 추진주체

- 문화관광부/한국게임산업개발원, 교육인적자원부/한국교육개발원, 한국게임학회 등 정규/비정규 교육기관

□ 소요 예산 (단위 : 억원)

연 도	2007년	2008년	2009년	2010년	합 계
예 산	1	1.5	2	2	6.5

4) 업계의 기존 인력 재교육 지원

□ 목적/필요성

- 개발 및 서비스 경험을 가지고 있는 기존 인력의 재교육을 통하여 보다 지속적이고 연속적인 국내 게임 산업 발전의 토대 마련
- 단기적으로 국내 게임 산업 인력의 효율적인 재배치는 물론 중장기적으로는 핵심 인력 및 기술 인력의 고도화 효과 기대

□ 현황 및 문제점

- 국내 게임 산업체의 규모 확대 및 게임 개발 프로세스 변화에 따라 기존 인력의 재교육 필요성 지속 증가
- 국내 게임 산업계, 특히 개발 분야에서 경력직이 차지하는 비중 및 역할 지속적으로 증가하나 재교육 시스템에 대한 실질적인 운영 사례는 매우 드문 상황임

□ 사업 내용

- 업체별 심화 교육 수요 분석
  - 업체별 인사 담당자 및 재교육 실무자 협의회 구성
  - 업체 및 게임 산업계 근무자 대상 심화 교육에 대한 수요 조사
- 재교육 프로그램 개발 및 운영 지원
  - 일차적으로는 수요 분석을 기반으로 우선순위에 따라 기존인력 재교육 프로그램의 일부만을 우선 개발하여 시범 실시 추진
  - ※ 일괄적인 대규모 접근 보다는 시범 실시를 통한 산업계 및 기존 인력의 반응 및 평가 분석을 통해 단계적이고 점진적으로

확대 실시

- 시범사업에 대한 평가 분석을 통해 재교육 프로그램 실시 분야의 확대 여부 검토 후 기존 인력 재교육 프로그램의 단계적 개발 진행

□ 연도별 추진체계

- o 2006~2007 : 업체별 인사 담당자 및 재교육 실무자 협의회 구성, 교육 수요 분석, 시범 실시
- o 2008~2010 : 재교육 프로그램의 단계적 확대 진행

□ 추진주체

- o 한국게임산업개발원, 한국게임산업협회 등 게임 관련 협회, 국내 게임 산업 관련 업체, 게임 인력 양성 포럼

□ 소요 예산 (단위 : 억원)

연 도	2007년	2008년	2009년	2010년	합 계
예 산	5	1	1.5	1.5	9

5) 취약 분야 교육 기반 강화

□ 목적/필요성

- o 개발 분야 등 특정 분야 위주의 현재의 인력 양성 제도의 개선을 통해 중장기적인 산업 발전에 따른 인력 수급의 불균형 해소
- o 취약한 분야에 대한 교육 기반 강화를 통해 산업 경쟁력 확보

□ 현황 및 문제점

- o 국내 게임 인력 양성 기관의 대다수는 부문별로는 프로그래밍, 그래픽에, 플랫폼별로는 온라인에 편중
- o 게임 인력의 편중화는 게임 관련 일부 핵심 기술의 경우에 외국 기반 기술에의 종속이 예상되는 게임 인력 공급의 구조적 불균형

## 현상 초래

### 사업 내용

- 부문별 계층별 취약 분야에 대한 수요 조사 및 선정
  - 부문별 계층별 인력 현황 및 인력의 질적 양적 수요 조사, 취약 분야 선정
  - 게임트렌드에 따른 취약 분야 변화에 대한 조사 정례화
  
- 취약 분야 관련 교육 프로그램 개발 및 운영 지원
  - 취약 분야 교육 과정 및 프로그램 개발 및 보급
  - 취약 분야 관련 우수 교육 기관에 대한 교육 환경 개선 지원 사업 실시

### 연도별 추진체계

- 2006~2007 : 취약 분야 수요 조사 및 선정
- 2008~2010 : 취약 분야별 세부 로드맵 및 프로그램 개발 보급

### 추진주체

- 한국게임산업개발원, 게임 관련 협회, 국내외 학계 및 업계

### 소요 예산 (단위 : 억원)

연 도	2007년	2008년	2009년	2010년	합 계
예 산	1.5	2.5	5	5	14

## 6) 게임아카데미 역할 및 기능 재정립

### 목적/필요성

- 게임아카데미의 교육 방향 및 체질 개선을 통하여 게임 산업 전반에 기여할 수 있는 특수 교육 기관으로서의 역할 강화

□ 현황 및 문제점

- 최근 게임 인력 양성 교육 기관의 양적 질적 성장에 따라 정부가 지원하는 교육기관으로써의 게임아카데미의 위상 재정립 필요성 증가
- 현재 게임아카데미가 진행하고 있는 게임 교육콘텐츠 개발 및 보급 사업 및 게임 원격 사업의 경우, 대상 분야에 있어 국내 게임 산업계의 취약 분야 교육으로의 전환 필요성 증가

□ 사업 내용

- 사설 교육 기관과의 차별화 전략 수립
  - 재교육 전문 기관 또는 취약 분야 전문 교육 기관 등의 운영 방향 및 세부 추진 내용 조사 및 분석
  - 게임아카데미의 역할 및 차별화 방향 및 전략 도출을 통한 기능 재정립
- ‘게임 교육 자문 협의회’ 구성, 게임아카데미의 운영에 대한 평가 및 자문 체계 마련
  - 시장 환경 변화에 적절히 대응할 수 있는 게임아카데미 기능 및 운영 상태에 대한 지속적인 외부 평가 및 자문
  - 관련사업에 대한 정기적인 평가 및 신규사업에 대한 의견 수렴

□ 연도별 추진체계

- 2006~2007 : 게임아카데미 역할 재정립 검토 및 결정
- 2008~2010 : 게임아카데미 역할에 대한 정기적인 외부 평가 및 자문 실시

□ 추진주체

- 문화관광부, 한국게임산업개발원, 학계 및 업계

□ 소요 예산 (단위 : 억원)

연 도	2007년	2008년	2009년	2010년	합 계
예 산	0.5	0.5	0.5	0.5	2

### 2.8.3 연구기반 및 교육역량 강화

#### 1) 게임교육기관 평가제도 구축 및 우수기관 선정 지원

□ 목적/필요성

- 게임교육기관에 대해 교육내용 및 교육체계 등에 대한 평가를 통해 내실 있는 운영과 적절한 게임 인력 양성 도모

□ 현황 및 문제점

- 실무에 바로 투입할 수 있는 기술능력을 보유한 인력이라는 업계의 수요에 대해 교육기관의 성과 미흡
- 교육기관에 대한 정기적인 평가 부재로 배출 인력에 수준의 향상성 유지가 어려움

□ 사업 내용

- 게임교육기관 평가 기준 수립 및 평가 실시
  - '게임학과 평가 위원회'를 구성하여 평가기준과 방법에 대한 검토 및 평가안을 마련하고 전국 게임학과를 대상으로 평가 실시
- 우수 교육기관 선정 및 지원 정책 추진
  - 매년 평가 후 우수 교육기관 선정 및 사례 발표회 개최
  - 선정된 우수 교육기관에 실질적인 인센티브 부여
    - 연구 및 운영 지원금의 지원, 인턴쉽 우선 배정 또는 해외 연수 등 특전 부여

연도별 추진체계

- 2006~2007 : 교육기관 평가 기준 수립 및 시범 평가 실시
- 2008~2010 : 4/2년제 대학/대학원 평가 및 우수교 포상

추진주체

- 한국게임산업개발원, 인력양성 유관기관, 게임업계 및 협회·단체

소요 예산 (단위 : 억원)

연 도	2007년	2008년	2009년	2010년	합 계
예 산	5	5	6	7	23

2) 국내외 인력 양성 관계기관간 협력 네트워크 구축

목적/필요성

- 게임 인력 기관간 협력 네트워크 구축을 통해 국가 전체 게임 인력 양성 시스템의 내실화 및 효율화 도모
- 우수한 해외 게임교육기관과의 교류 활성화를 통해 국제적 역량 강화

현황 및 문제점

- 게임 인력 양성 관련 정부 관계 기관(문화관광부, 정보통신부, 교육인적자원부, 노동부, 산업자원부, 외교통상부, 여성부 등) 간의 협력 미비에 따른 효과적인 인력 양성 체계 부재
- 해외 유명 게임 개발국 교육기관과의 교류협력 사업이 저조하여 국내 게임기술 발전 진척도 미진 및 관련 인력 양성의 국제 트렌드 반영 미흡

사업 내용

- 국내 게임 인력 양성 관련 협의체 구성 및 운영
  - 관련 정부 기관 담당자를 모두 포함하는 '게임 인력 양성 정책

실무 회의' 구성

- 게임 산업계 인사 담당자, 학계, 비정규 교육기관 관계자, 정부 기관 담당자('게임 인력 정책 실무 회의' 참여자)가 참여하는 '게임 인력 양성 포럼' 구성, 정기 개최

- o 해외 게임교육기관의 교류 협력 관련 협의체 구성 및 운영
  - '세계 게임 교육기관 교류협력 포럼' 구성
  - 공동 컨퍼런스 및 워크샵 개최, 국제게임컨퍼런스(KGC) 확대 개최 추진

□ 연도별 추진체계

- o 2006~2007 : 게임 인력 양성 포럼, 게임 인력 양성 정책 실무 회의 구성
- o 2008~2009 : 해외 게임교육기관 교류 협력 포럼 구성 및 워크샵 개최
- o 2010 이후 : 해외 교육 네트워크 확대

□ 추진주체

- o 한국게임산업개발원, 한국교육개발원, 한국산업인력공단, 기타 정부 내 게임 인력 양성 관련 기관

□ 소요 예산 (단위 : 억원)

연 도	2007년	2008년	2009년	2010년	합 계
예 산	1	1	1	1	4

3) 중장기 게임 인력 수급 균형 정책 수립

□ 목적/필요성

- o 게임 산업의 중장기적인 발전에 기여할 수 있는 창작 인력, 개발 인력, 연구 인력, 서비스 인력 전반에 대한 중장기 인력 수급 균

## 형 방안 마련

### □ 현황 및 문제점

- IT 산업 및 문화 산업의 특성을 고루 갖추고 있는 게임 산업의 특성 상 특정 산업이나 특정 정부 부처 관점에서의 인력 양성 계획 수립에의 한계점 존재
- 정확한 수요 예측 및 교육 기관 별 차별성이 부재된 상황에서 특정 분야에 집중된 게임 관련 정규, 비정규 교육 기관의 양적 확대로 전체 게임인력의 수급 균형이 미진

### □ 사업 내용

- 게임 산업 인력 수급 실태에 대한 정기적인 분석
  - 범정부 차원의 국내 게임 산업 실태 및 인력 양성 현황에 대한 진단 실시, 정기적으로 연차 보고서 발간
- 범정부 차원의 중장기 게임인력 수급 균형 정책 수립
  - '게임 인력 양성 포럼' 및 '게임인력정책 실무회의' 를 통한 초기 연구 진행 및 타당성 조사
  - 관련 정부 기관이 공동으로 참여하는 중장기 게임 인력 수급 균형 정책 수립, 관련 기관간 협력 추진

### □ 연도별 추진체계

- 2006~2007 : 국내 게임 산업 실태 및 인력 수급 현황 파악
- 2008~2009 : 범정부 차원의 게임 산업 인력 수요 공급 실태 연차 보고서 발간 및 수급 균형 정책 수립
- 2010 이후 : 정책 사업 추진 및 사후 관리

### □ 추진주체

- 정부 내 게임 인력 양성 관련 부처 (문화관광부, 정보통신부, 교육인적자원부, 노동부, 산업자원부, 여성부 등), 관련 협단체

□ 소요 예산 (단위 : 억원)

연 도	2007년	2008년	2009년	2010년	합 계
예 산	1	1.5	2	2	6.5

## 2.8.4 실행형 산학협력 모델 구축

### 1) 산 학 공동 프로젝트 활성화

□ 목적/필요성

- 산·학·관 협력 네트워크 강화를 통해 현장중심의 게임 전문 인력 양성, 실무중심의 교수 학습모델 개발 및 산학프로그램의 활성화 및 게임교육의 체질개선
- 산·학·관 협력을 통한 게임 전문인력 DB 구축으로, 교육과 게임인력 활용 부문간의 유기적인 연계체제 및 효율적인 인력 활용 기반 마련

□ 현황 및 문제점

- 산학협력의 핵심인 현장 전문인력의 교육 프로그램 참여에 대한 제도적 인프라 미흡
- 게임인력 교육 이후 현장과의 연계 측면이 미흡한 산학 협력 프로그램의 실시로 인력활용의 실질적인 효과 부재

□ 사업 내용

- 산·학 협력 프로젝트 발굴 및 지원
  - 프로젝트 중심의 교과 과정 개편 추진 및 산학 협력 프로젝트 우수 사례 발표회 정례화
  - 현장 전문인력의 교과 운영 참여 확대 지원
    - 업계 수요중심의 교육체제 구축, 평가(커리큘럼, 교육내용 등)
  - 게임관련 국가기술자격검정 확대, 내실화

- 우수인력 확보를 위한 「병역특례제도」 개선 및 대안 마련  
(병무청 협조)

- 산·학·관 협력을 통한 게임 전문 인력 DB구축
  - 각 교육기관 졸업생 현황 및 업계 수요 인력에 대한 조사에 근거한 게임전문 인력 정보를 통합, 관리/검색 시스템 구축
    - 국가 게임전문인력 종합 DB 구축, 게임인력정보 통합 관리
  - 종합 DB 구축과 함께 게임인력 이력 인증시스템 체계화

□ 연도별 추진체계

- 2006~2007 : 산·학·연 협력 인력양성 프로젝트 개발 추진
- 2008~2009 : 국가 게임전문인력 종합 DB 및 인력정보 통합 관리 시스템을 구축
- 2010 이후 : 게임인력 이력 인증시스템 체계화

□ 추진주체

- 한국게임산업개발원, 인력관련 유관기관, 게임업계 및 협회·단체

□ 소요 예산 (단위 : 억원)

연 도	2007년	2008년	2009년	2010년	합 계
예 산	10	10	2	2	24

2) 산학연계 게임 공모전 위상 강화 및 통합

□ 목적/필요성

- 실무형 프로젝트 진행을 검정하기 위한 수단 및 기업 현장과의 연계를 강화하는 방향으로 게임공모전의 역할 제고

□ 현황 및 문제점

- 다수의 공모전을 통하여 응모되는 작품의 중복성 및 산업체로 연

결되어 제품화하기 어렵다는 점에서 공모전의 실효성 부족

- 공모전 수상작의 시제품화에 대한 검증방법 및 산업체와의 연결 창구 필요

#### □ 사업 내용

- 게임공모전을 통한 우수 동아리 발굴/지원 및 콘텐츠 활용 활성화
  - 게임공모전을 통해 우수 동아리 및 프로젝트 팀 발굴 및 지원
    - 공모전 방식의 범위를 '개발' 분야에서 '창작', '연구', '서비스' 분야 전체로 단계적으로 확대 실시
  - 공모전을 통해 모집된 다양한 콘텐츠를 게임업체와 연결하는 채널 마련, 수상작의 콘텐츠 활용 방안 추진
    - 시장 수용에 적용될 수 있는 심사 기준 마련
- 프로젝트 실습 학점제 등 실무형 프로젝트 교육 환경 구축 사업과 공모전과의 연계 시행
  - 산학 협력 프로젝트 공모전 개최를 통해 산학협력 활성화 유도
  - 프로젝트 실습 학점제(재학생이 산학협력을 맺은 기업현장에서 수업을 받도록 하는 제도) 응용 적극 도입, 결과물을 공모전과 연계

#### □ 연도별 추진체계

- 2006~2007 : 공모전 수상작 DB구축 및 심사 기준 정비
- 2008~2010 : 공모전 분야 확대 및 프로젝트 공모전 개최

#### □ 추진주체

- 한국게임산업개발원, 인력관련 유관기관, 게임업계 및 협회·단체

□ 소요 예산 (단위 : 억원)

연 도	2007년	2008년	2009년	2010년	합 계
예 산	5	4	2	2	13

3) 인턴십 제도 및 주문식 교육 모델 정립

□ 목적/필요성

- 교육의 질적 수준을 제고하여 현장 적응력과 국제 경쟁력을 갖춘 게임개발 전문인력 양성
- 기업의 수요에 근거한 교과과정 및 커리큘럼을 통해 실무에 바로 투입될 수 있는 맞춤형 게임인력을 양성

□ 현황 및 문제점

- 노동부, 중소기업청, 기타 부처 산하 인턴십 프로그램들이 다양화라는 잇점에 비해 비용-편익면에서 효율성 미흡
- 업계와 교육기관 간 주문식 교육을 시도하고 있는 IT분야에 비해, 게임산업의 경우 교육기관의 역사가 짧아 업계와 밀착된 인력 양성 여건 부재

□ 사업 내용

- 기존 인턴십 제도에 대한 각계의 의견 수렴 및 개선 방향 수립
  - 인턴십 제도의 성공을 위한 학생, 대학, 기업이라는 3자 주체의 이해관계와 요구의 합리적인 접점 연구
  - 인턴십에서 기업과의 연계활동 강화 요구에 따른 활동 분석 (학생 입장)
  - 인턴십의 제도적 활용에 필요한 산학연계를 도모할 수 있는 교과목 개발 선행(교육기관의 입장)
  - 인턴십 활성화를 위한 정부의 제도적인 지원방안 도출
- 산·학 연계의 주문식 게임교육 제도 연구

- 수요 지향적 게임 인력양성 좌담회 개최
- 선진 교육시스템 연구를 위한 교수 게임 해외 연수 지원사업 추진
- 게임분야 해외교수초빙지원사업(신규 및 계속) 지원 사업 추진
- 기업 맞춤형 게임인력 양성 사업 추진

연도별 추진체계

- o 2006~2007 : 인턴쉽 제도 및 주문식 교육 모델 수립
- o 2008~2010 : 모델 적용 및 사후 관리

추진주체

- o 한국게임산업개발원, 인력양성 유관기관, 게임업계 및 협회·단체

소요 예산 (단위 : 억원)

연 도	2007년	2008년	2009년	2010년	합 계
예 산	6	8	10	12	36

## 2.8.5 취업환경 개선 및 내실화

### 1) 게임 산업에 특화된 공동 채용 박람회 개최

목적/필요성

- o 특화된 게임 채용박람회를 계획하여 유망기업에 유능한 인재를 연결하고 구직자에게 쉽게 일자리 확보 기회 제공

현황 및 문제점

- o 국내 문화콘텐츠 산업이 빠른 성장에 따른 각 분야별 우수 인재에 대한 수요 상승률에 비해 인력 공급 미진
- o 취업 희망 인력과 업계의 희망 인력 간의 거리를 좁히고 직접적이고 업계의 수요인력 조건에 적절한 채용의 장 마련이 필요

사업 내용

○ 게임 인력 수요시장의 트렌드 분석

- 게임인력의 분야별 수요 수준 및 인력 시장 현황 분석을 통한  
구직자와 구인자의 현황 파악

※ 게임인력 수급실태 연차 보고서 활용 및 업계 의견 수렴

○ 온/오프라인 게임산업 채용박람회 개최

- 온라인 채용박람회를 적극 활용하여 수시 채용 활성화
- 오프라인 채용박람회를 통한 인력의 채용 범위 및 기회 확대

연도별 추진체계

○ 2006~2007 : 게임인력 수요시장 트렌드분석 및 채용 박람회 개최

○ 2008~2010 : 게임분야 온/오프라인 채용박람회 확대

추진주체

- 한국게임산업개발원, 인력양성 유관기관, 게임업계 및 협회·단체

소요 예산 (단위 : 억원)

연 도	2007년	2008년	2009년	2010년	합 계
예 산	5	5	5	5	20

2) 실효성 있는 국가 검정 제도 개선

목적/필요성

- 실질적인 업무와의 연계를 위한 자격증 제도 개선을 통해 게임  
인력의 자질을 평가하는 공인 자격증의 실효성 증진

현황 및 문제점

- 게임인력에 대한 체계적이고 개선된 검증방법이 부재하여 게임

관련 자격증 소지자에 대한 기대가 저조

- 현업에 투입할 수 있는 작업능률과 현재의 자격증 수단 간의 간극으로 인해 인력 자질 판단의 근거로서의 자격증 의미가 퇴색

□ 사업 내용

- 국가검정 자격시험제도의 개선 방안 수립 및 추진
  - 자격시험의 수준, 시장 환경 수요 반영, 실무중심의 능력 기준, 시험방식의 합리성 효율성 제고 등에 대한 분석
  - 타 분야의 국가공인자격시험과 동등한 수준으로 자격시험의 질적 수준 제고를 위한 개선 방향 및 전략 도출
- 자격시험제도의 활성화 지원사업 추진
  - 자격증의 현장 활용성에 대한 제도화 연구
  - 게임교육기관과 연계한 자격증 가산점제도 도입(학점 취득) 검토

□ 연도별 추진체계

- 2006~2008 : 국가검정 자격시험제도의 개선 방안 수립
- 2008~2010 : 자격시험제도의 활성화 지원

□ 추진주체

- 한국게임산업개발원, 인력양성 유관기관, 게임업계 및 협회·단체

□ 소요 예산 (단위 : 억원)

연 도	2007년	2008년	2009년	2010년	합 계
예 산	2	2	2	2	8

3) 병역특례제도의 활성화 및 효율화 제고

□ 목적/필요성

- 병역특례제도 적용 대상 및 범위에 대한 제고를 통해 중소기업체의 인력 활용 및 우수 인력의 현장 체험 기회 확대

□ 현황 및 문제점

- 현재의 병역 특례 제도의 범위가 제한, 업계의 활용도 및 만족도가 미흡
- 게임업계 개발자 중 병역특례자의 비중이 높음에도 불구하고, 인력 부족난을 겪는 중소기업보다는 대형 우량업체에 집중되는 경향

□ 사업 내용

- 병역특례제도의 운영 현황 및 업계 분석을 통한 개선 방안 마련
  - 정부 정책 사업에 대한 인센티브로서 병역특례제도 활용 검토
  - 우수 인재 확보의 관점에서 제도 개선안 도출
  - 단계별 병역특례제도 개선안 추진

□ 연도별 추진체계

- 2006~2007 : 병역 특례 제도 개선을 위한 인력 현황 조사
- 2008~2010 : 병역 특례 제도 개선 방안 마련

□ 추진주체

- 한국게임산업개발원, 게임업계 및 협회·단체, 병무청, 인력양성 유관기관

□ 소요 예산 (단위 : 억원)

연 도	2007년	2008년	2009년	2010년	합 계
예 산	1	1	1	1	4

## 2.9 향후 정책의 방향 및 이에 대한 검토

앞서 살펴본 바와 같이 향후, 정부의 게임 전문 인력 양성 방향은

기존의 방향에서 상당 부분 변화될 것으로 예상되고 있다. 이를 표로 요약하면 다음과 같이 정리할 수 있다.

표 II-55. 게임 전문 인력 양성과 관련된 정부 정책의 변화 방향

구 분	기존 내용	향후 변화 내용
게임산업 발전의 기본 방향	게임의 산업화 주력	글로벌화 산업인프라 강화 선도기술 확보 법 제도 기반 마련 게임의 건전 문화 조성
인력 양성 방향	개발 인력 중심	창작, 연구, 서비스 인력으로 확대
	신규 인력 중심	기존 인력의 재교육 추가
	온라인게임 및 모바일게임 중심	전체 플랫폼으로 영역 확대
정부 지원 부서	문화관광부, 정보통신부 등 개별	정책 실무 협의체 구성
인력 양성 중점	양적 확장	질적 발전

표의 내용을 자세히 살펴보면 다음과 같다. 우선, 게임 산업의 발전 방향이 산업화 단계에서 벗어나 점차 글로벌화 되어 가고 있으므로 이에 따른 전문 인력 양성 방향의 변화가 불가피할 것으로 보인다. 기존의 기술 중심의 인력 양성 정책은 향후 마케팅과 운영 등 분야로 변화되며 글로벌화된 인력 양성의 필요성이 더욱 빠르게 확장될 것이며, 국내 소비 시장의 건전화 를 위한 소비자 환경, 즉 “게임의 건전 문화 조성” 도 향후 게임 인력 양성에 있어 새로운 화두가 될 것으로 보인다.

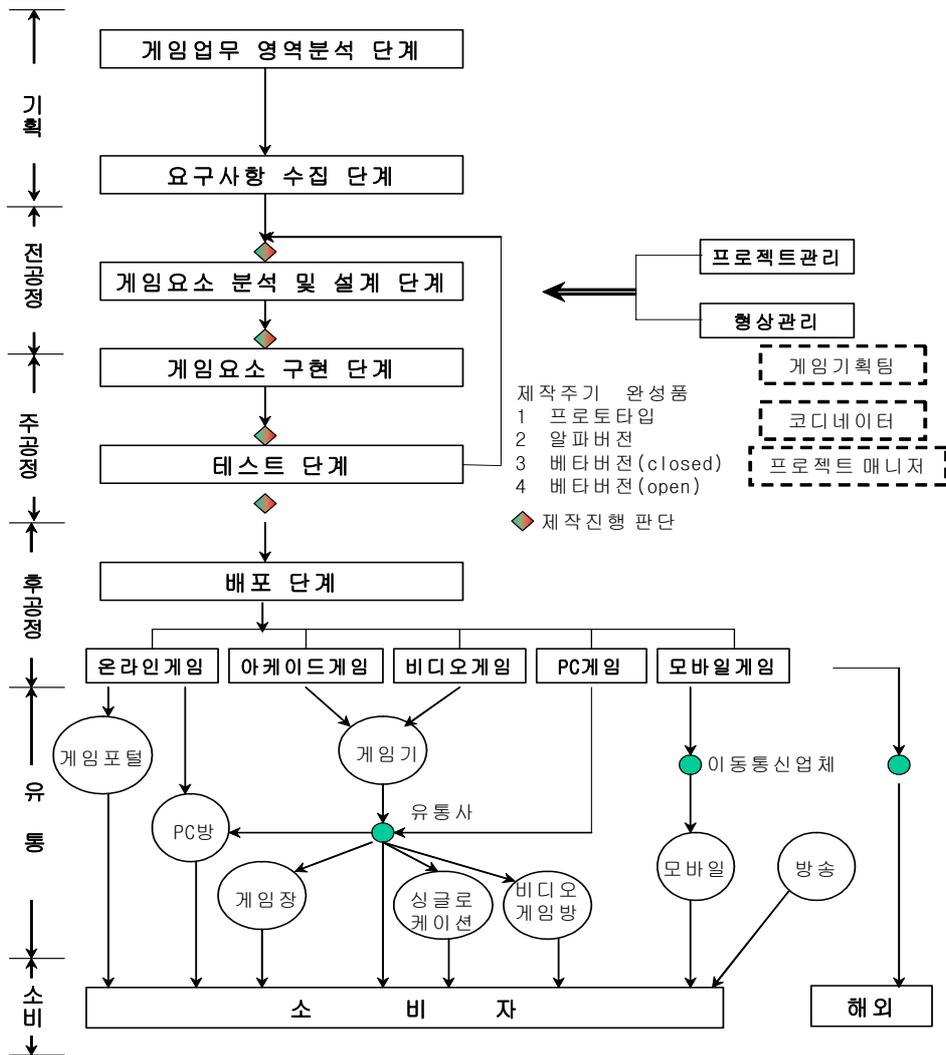
둘째, 게임 전문 인력의 방향을 기존의 “개발” 인력 중심에서 “창작”, “연구”, “서비스” 인력 부분으로 다변화하는 것도 정부 정책의 큰 변화의 흐름이라고 할 수 있다. 이는 애니메이션 등의 산업에서 국내 산업 기반이 어려움을 벗어날 수 없었던 요인을 사전에 방지하는 한편, 저가의 인건비를 기반으로 한 중국이나 인도의 게임 산업 확장을 견제하기 위한 사전 포석이 될 것이다. 특히, 그래픽스 분야나 네트워크 분야에서의 원천 개발 기술의 선진국 종속을 막기 위하여 연구 인력 분야의 양성 정책이 빠르게 강화될 것으로 판단된다.

다음의 표와 그림은 개발 중심의 관점에서의 게임 산업의 직무별 세부 직업의 다양성을 나타내는 표이다. 실제로 정부의 기존 인력 양성 방향이 계획대로 “개발”의 측면에서 향후 “창작”, “연구”, “서비스” 분야로 확장될 경우, 게임과 관련된 다양한 직업의 등장과 이에 따른 교육 수요는 지속적으로 확대될 것으로 판단된다.

표 II-56. 게임산업 관련 직업군별 세부 직업명

직업군	세부직업명
게임 프로덕션(Production)	프로듀서(Executive Producer)
	보조프로듀서 (Assistant Director, Associate Director)
게임 오퍼레이션 (Game Operation)	품질관리 매니저(QA: Quality Assurance Manager)
	게임 마스터(GM: Game Master)
게임 엔지니어그룹 (Engineer Group)	기술 책임(Lead Programmer)
	클라이언트 엔지니어(Client: 3D, 2D, AI, Tool)
	서버 엔지니어(Server: Server Programmer / D/B)
	시스템 엔지니어(System Engineer)
게임 비주얼아트 (Visual Art)	아트 디렉터(AD: Art Director)
	컨셉 아티스트(Concept Artist/Sketch Artist)
	3D모델러 (3D Modeler)
	애니메이터(Animator)
	유저 인터페이스 디자이너(UI Designer)
	특수효과(F/X Director/Technician)
게임 오디오 (Audio)	기술감독(Technical Director)
	오디오 기술(Sound Engineer)
	오디오 작곡 (Sound Composer)
게임 마케팅그룹 (Marketing Group)	악기연주 (Sound Performer)
	마케팅 디렉터(Marketing Director)
	마케터(Marketer)
게임디자인 (Game Design)	마케팅플래너(Marketing Planner)
	게임 시스템 디자이너(Game System Designer)
	레벨 디자이너(Level Designer)
	시나리오작가(Scenario Writer)
경영지원 및 법률 (Administration/Legal)	크리에이티브 디렉터(Creative Director)
	프로젝트 매니저(PM: Project Manager)
	재무/회계(Financial / Accounting)
	비즈니스 디렉터(Business Director)
	법률 전문가(Legal Expert)

이는 다음의 그림을 통해서도 확인 할 수 있다.



출처 : 문화콘텐츠진흥원(2005), 문화콘텐츠산업 인력구조 및 직무 분석

그림 II-35. 게임제작 및 유통과정 흐름도

실제로 게임 산업은 창작과 개발 외에도, 테스트 및 배포 등과 관련된 유통과 소비가 산업에 있어 매우 중요한 비중을 차지하고 있으므로, 정부의 정책 변화는 교육 시장 및 교육 수요자에게 다양한 형태의 환경 변화를 전개시키는 한편, 게임 산업에 대한 직업군으로써의 관심의 확대와 이에 따른 교육 수요의 증가, 교육 내용에 대한 기대치 증가를 연속적으로 야기할 것으로 보인다.

셋째, 신규 인력과 특정 플랫폼 위주의 인력 양성 정책도 대폭 개선되는 추세에 있다. 이미 세계 우위를 점유하고 있는 온라인게임 및 모바일게임 분야에서의 다양한 개발 경험을 타 플랫폼으로 공유, 발전시키는 한편, 노하우 및 기술 전수가 발전적으로 확대될 수 있도록 기존 개발 인력에 대

한 재교육과 타 플랫폼에 대한 교육 영역 확대에 새로운 정책 방향이 맞추어지고 있기 때문이다. 따라서, 향후의 게임 전문 인력 양성에 있어서는 기존 인력의 재교육과 타 플랫폼에 대한 고려 등 다양성과 개성화 부분이 강조되어야 할 것이다.

마지막으로, 개별화되어 경쟁적으로 추진되던 정부 내의 인력 양성 방향도 향후에는 어느 수준 정리가 될 것으로 보인다. 즉, 문화관광부, 정보통신부, 산업자원부, 노동부 등에서 중복적으로 이루어진 각개의 게임 인력 양성 계획이 보다 큰 틀에서 정리되는 흐름이 마련될 것으로 판단된다. 이러한 변화는 향후 게임 인력 양성 분야에 있어서도 양적 성장보다는 질적 성장이 중시되는 시기가 도래하고 있음을 의미한다. 즉, 양적 성장 위주의 정부의 교육 지원 방향이 이제 질적 성숙 단계로 변화하고 있으며, 이에 따라 산업계의 요구를 적극적으로 수용할 수 있는 한 단계 발전된 실무형의 교육이 이루어져야 함을 암시하고 있는 셈이다.

### III. 게임산업 인력 양성 인식 및 사례 분석

#### 3.1 인력 양성에 대한 조사

게임 산업 인력 양성에 대한 인식 조사에서는 게임 인력을 가장 필요로 하고 이에 따라 게임인력에 대한 요구사항이 분명한 게임 개발사들을 통하여 실시하였다. 보다 많은 대상의 선정과 많은 인원에 대하여 조사함이 바람직하겠으나 조사비용과 지역의 한계 등으로 범위를 축소하여 대전지역으로 국한 하였다. 그러나 최근 게임 개발 회사들의 상황과 요구사항을 수렴할 다양한 게임 장르별 분포를 고르게 하도록 노력하여 12개사 약 40명의 의견을 토대로 게임 인력 양성에 대한 의견 및 교육과정에 대한 사항을 분석 하였다.

설문조사의 내용은 크게 2가지로 구분된다. 첫째는 현재 우리나라의 상황에서 게임 산업계가 요구하는 게임인력을 어떻게 양성하는 것이 좋은지에 대하여 질문을 하였고, 두 번째는 국내에서 게임교육을 실시하는 4년제 대학의 커리큘럼을 제시하고 업계의 요구 수준에 따른 교육과정의 수정 및 변경을 요구하였다. 설문에서 제시한 4년제 대학의 커리큘럼은 대전 소재 우송대학교 게임멀티미디어 학과의 커리큘럼을 활용하였다.

##### 3.1.1 설문 내용

다음은 게임 산업 인력 양성에 대한 인식 조사에 활용한 설문지이다.

국제적 수준의 미래 · 수요 지향적  
게임 교육 프로그램 개발을 위한  
게임 개발사 설문조사

안녕하십니까?

먼저 우리나라의 게임 산업 발전을 위한 노력에 감사를 드리며 귀사와 귀 연구원들의 무궁한 발전을 기원합니다. 저는 귀 게임 개발사 및 산업체 의견을 반영하여 국제적 수준의 미래 수요 지향적 게임 교육 프로그램 개발을 위해 설문조사를 수행하고 있는 한국 콘텐츠학회 회장, 충북대학교 조용환 교수입니다.

본 설문조사는 통계에 관한 법률에 의해 선생님의 응답은 절대 비밀이 보장되니 대학 인력 양성차원에서 보더라도 잠시만 시간을 내어서 설문에 응답해주시면 감사하겠습니다.

본 조사와 관련해 문의사항이나 의견 개진이 있으시면 아래로 연락해주시기 바랍니다. 성심성의껏 응답해 드리겠습니다.

2006년 4월

조사 기관 : 한국 콘텐츠 학회 ( 043 - 272 - 8830 )

협력 기관 : 우송대학교 게임멀티미디어학과

책임연구원 : 충북대학교 전기전자컴퓨터 공학부 교수 조용환

1. 우리 지역의 게임시장규모(업체수, 근무자 수)는 어느 정도라고 생각하십니까?

일반적 게임개발사 ( \_\_\_\_\_ 개, \_\_\_\_\_ 명)

성인용 게임개발사 ( \_\_\_\_\_ 개, \_\_\_\_\_ 명)

2. 우리 지역의 경우 특화를 시켜야 할 게임의 장르는 어떤 것이 우선되어야 한다고 생각되십니까(복수 응답 가능)?( 1, 2, 3)

① 아케이드 게임    ② PC게임    ③ 비디오게임    ④ 온라인 게임

⑤ 모바일 게임    ⑥ 기타(\_\_\_\_\_)

3. 우리 지역의 게임시장의 장점은 무엇이라 생각하십니까(복수 응답 가능)?  
(1, 2, 7)

① 기획                    ② 제작                    ③ Test bed                ④ 마케팅

⑤ 캐릭터                ⑥ 시나리오                ⑦ 음향                    ⑧ 그래픽

⑨ 기타(\_\_\_\_\_)

4. 게임 산업육성을 위해 지역별로 다음의 기술요소와 제작단계별 상대적 최적지에 대한 귀하의 의견은?

구 분	기술요소	제작단계	의 견
대전/충청권			
광주/제주/호남권			
춘천/강원권			
대구/경북권			
부산/경남권			

※ 기술요소 및 제작단계 예시 : 캐릭터, 시나리오, 음향, 그래픽, 제작, 마케팅 등

5. 우리 지역의 게임 인력의 부족분야는?

교육기관 유형	현재 인력	적정인력	부족인력
기획분야			
제작분야			
그래픽분야			
Test bed 분야			
마케팅 분야			
기타( )분야			

6. 우리 지역 게임시장의 규모는 적정하다고 생각하십니까? ( )

- ① 매우 적정하다      ② 적정한 편이다.      ③ 그저 그렇다  
 ④ 적정치 못한 편이다      ⑤ 전혀 적정치 않다

7. 우리 지역 게임시장의 성장가능성은 어떻다고 보십니까? ( )

- ① 매우 높다.      ② 높은 편이다      ③ 그저 그렇다  
 ④ 낮은 편이다      ⑤ 매우 낮다

8. 우리 지역의 국내 마케팅(판로)의 잠재력은 어떻다고 보십니까? ( )

- ① 매우 높다.      ② 높은 편이다      ③ 그저 그렇다  
 ④ 낮은 편이다      ⑤ 매우 낮다

9. 우리 지역의 해외 마케팅(판로)의 잠재력은 어떻다고 보십니까? ( )

- ① 매우 높다.      ② 높은 편이다      ③ 그저 그렇다  
 ④ 낮은 편이다      ⑤ 매우 낮다

10. 귀 회사의 주력 분야를 선택해 주십시오.(복수응답가능)

- 1) 컴퓨터 H/W
- 2) 통신·네트워크 H/W
- 3) 컴퓨터 S/W(응용 S/W)
- 4) 게임
- 5) 콘텐츠(멀티미디어)

11. 귀 회사에서 사용 중인 프로그램 언어와 도구를 선택하여 주십시오.

(복수응답 가능)

- 1) C                    2) C++                    3) C#                    4) Java  
 5) Pascal            6) COBOL            7) Visual C++    8) Visual Studio .Net  
 9) Jbuilder        10) Visual Basic 11) Power Build 12) Delphi  
 13) VHDL        14) 기타(                    )

12. 귀 회사에서 사용 중인 운영체제를 선택하여 주십시오. (복수응답 가능)

- 1) MS Windows 계열            2) MS 닷넷            3) 유닉스  
 4) 리눅스                    5) RTOS(                    )

13. 다음의 교과목에 대하여 살펴보고 2,3,4학년 교과목 중에서 실무 중심  
 현 교육과정으로의 개편을 통한 인재 육성을 위해서는 어떤 과목이 폐지되  
 었으면 하는지를 직접 마크 하여 주시기 바랍니다.

학년	1학기				2학기			
	이수구분	교과목명	학점	시간	이수구분	교과목명	학점	시간
2	교양필수	대학영어3	2	3	교양필수	대학영어4	2	3
	교양선택	외국어 및 자유선택	3	3	교양선택	외국어 및 자유선택	3	3
	전공필수	C프로그래밍2	3	3	전공필수	게임프로그래밍1	3	3
		자료구조	3	3		프로젝트수행방법론	3	3
	전공선택	컴퓨터그래픽디자인1	3	3	전공선택	컴퓨터그래픽스디자 인2	3	3
		웹 프로그래밍1	3	3		웹 프로그래밍2	3	3
		2D게임제작	3	3		JAVA프로그래밍2	3	3
		JAVA프로그래밍1	3	3		3D컴퓨터그래픽스1	3	3
	개설학점		23	24	개설학점		23	24
3	교양필수	직업윤리	2	2	교양필수	현대사회와 인간관계	2	2
		편입-생활영어1	-3	-3		편입-생활영어2	-3	-3
	교양선택	외국어 및 자유선택	3	3	교양선택	외국어 및 자유선택	3	3
	전공필수	웹 프로젝트연구	3	3	전공필수	게임프로젝트연구	3	3
		게임프로그래밍2	3	3	전공선택	온라인 게임개발	3	3
	전공선택	3D컴퓨터그래픽스2	3	3		모바일 프로그래밍1	3	3

		데이터베이스	3	3		시스템관리	3	3
		디지털컨텐츠분석 및 기획	3	3		디지털 애니메이션1	3	3
		네트워크프로그래밍	3	3		산업현장연구2	-3	
		현장실습	-2					
		산업현장연구1	-3					
	개설학점		23(8)	23(3)	개설학점		20(6)	20(3)
4	교양선택	외국어 및 자유선택	3	3	교양선택	외국어 및 자유선택	3	3
	전공선택	졸업프로젝트	3	3	전공선택	정보보호	3	3
		디지털사운드 처리	3	3		디지털컨텐츠 마케팅	3	3
		모바일 프로그래밍2	3	3		특수영상처리론	3	3
		디지털 애니메이션2	3	3		최신기술동향연구	3	3
		인공지능						
		산업현장연구3	-3					
		개설학점		18(3)	18	개설학점		15(3)
개설학점 총계	* 교양필수 : 26(6)학점 기초필수 : 18학점 교양선택 : 18학점 * 전공필수 : 21학점 전공선택 : 75(14)학점						총계 : 157 (20)학점	

14. 위 13번 문항을 기본으로 실무 중심형 인재 육성을 위하여 추가 하고자 하는 교과목이 있으시면 기입하여 주시기 바랍니다.


\* 끝까지 응답해 주셔서 대단히 감사합니다.

### 3.1.2 설문조사 내용 분석

대전지역 게임 및 콘텐츠 관련 업체의 실무 담당자들을 대상으로 약 30명에 대하여 설문 조사를 하였으나, 성의 없는 답변을 제외한 25명에 대한 설문 조사 검토 내용은 다음과 같은 결과를 받았다.

표 III-1. 설문 조사 결과

답 문 항	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1														
2	10	8	4	12	4									
3	3	10	2	8		1		5	5					
4														
5														
6	1	9	10	4	1									
7		10	8	6	1									
8	3	6	11	5										
9	2	6	7	6	4									
10	1		6	22	9									
11	3	21		3			19	7		7		11		
12	25		1	6										
13														
14														

---

### 1) 대전지역의 게임 업체수 및 근무자 수

위 답에 대한 각 설문응답자는 많은 편차를 보이며 답을 하였다. 이는 대전광역시 지역내에 얼마나 많은 게임 업체와 사람이 존재 하는지를 제대로 파악하지 못하고 있으며, 이를 전담하는 기관 조차 없는 것으로 여겨지는 사항이다.

### 2) 중점 육성 게임 분야

대전지역에서 육성하여 성장 시킬 수 있는 게임 분야로는 아케이드 게임과 온라인 게임이라 보고 있으며, 특히 PC 게임 시장의 답을 유추하면 아케이드 게임 개발을 PC 상에서 하기때문인 것으로 사려된다.

### 3) 대전의 게임 산업 장점

대전 지역의 게임 산업의 장점은 제작기술을 보유하고 있다는 것으로 여겨진다. 이는 대덕 연구단지를 중심으로 한 대학 및 연구원의 창업자들이 포진되어 있기 때문으로 여겨진다.

### 4) 산업의 성장가능성

각 업체의 관계자들은 대전지역이 향후 게임 산업 시장에서 중요 역할을 수행할 것으로 전망하고 또한 향후 게임시장의 성장 가능성을 높게 보고 있다. 그러나 마케팅 및 판로에 있어서는 부정적인 견해를 갖고 있다. 이는 향후 해당 인력을 배출해야 되는 숙제가 된다.

### 5) 개발 환경

대부분의 개발사들은 C와 C++를 사용하고 MS의 Windows 계열의

환경에서 개발을 수행하고 있다. 이는 대전지역내에서의 연구인력간의 교류를 쉽게 할 수 있는 환경이 될 수 있으며, 또한 대규모 프로젝트를 공통으로 수행 할 수 있을 것이다.

## 6) 교육과정

설문에서 현 우송대학교 게임멀티미디어학과 교육과정에서 수정되어야 할 부분과 추가해야 될 부분에 대한 질문의 사항은 다음과 같다.

표 III-2. 업체가 요구하는 과목

<ul style="list-style-type: none"> <li>- 프로젝트 수행</li> <li>- 알고리즘</li> <li>- 소프트웨어개발방법론</li> <li>- 업체와의 공동 프로젝트 수행</li> <li>- 실무자 초청강연</li> <li>- Linux 관련 교과목</li> <li>- Win32 프로그래밍</li> <li>- DirectX &amp; OpenGL</li> <li>- 인터넷 영어</li> </ul>	<ul style="list-style-type: none"> <li>- 고급 C++</li> <li>- 심화 자료구조</li> <li>- 프로젝트 작업</li> <li>- 엔진 분석</li> <li>- 게임 개발 방법론</li> <li>- 소프트웨어 설계</li> <li>- 게임 디자인</li> <li>- 산업현장과의 공동 프로젝트 수행</li> <li>- 프로그램 설계</li> <li>- 운영체제</li> </ul>
---	--

## 3.2 게임 교육 교과 과정의 개선

### 3.2.1 게임관련 학과 기존 교과 과정

기존 4년제 게임학과 교과 과정과 업체에서 요구한 사항을 기본으로 기존 교육 과정과 수요지향적으로 변경한 교육과정은 다음과 같다.

표 III-3 기존 교육과정

학년	1학기				2학기			
	이수구분	교과목명	학점	시간	이수구분	교과목명	학점	시간
1	교양필수	대학영어1	2	3	교양필수	대학영어2	2	3
		대학외국어1(일본어)	2	3		대학외국어2(일본어)	2	3
		컴퓨터1	3	3		컴퓨터2	3	3
		대학과 인생	2	2		문화와 예술	2	2
	기초필수	멀티미디어개론	3	3	기초필수	기초수학	3	3
		컴퓨터과학개론	3	3		디자인개론	3	3
		게임학개론	3	3		C프로그래밍1	3	3
	개설학점		18	20	개설학점		18	20
2	교양필수	대학영어3	2	3	교양필수	대학영어4	2	3
	교양선택	외국어 및 자유선택	3	3	교양선택	외국어 및 자유선택	3	3
	전공필수	C프로그래밍2	3	3	전공필수	게임프로그래밍1	3	3
		자료구조	3	3		프로젝트수행방법론	3	3
	전공선택	컴퓨터그래픽디자인1	3	3	전공선택	컴퓨터그래픽스디자인2	3	3
		웹 프로그래밍1	3	3		웹 프로그래밍2	3	3
		2D게임제작	3	3		JAVA프로그래밍2	3	3
		JAVA프로그래밍1	3	3		3D컴퓨터그래픽스1	3	3
	개설학점		23	24	개설학점		23	24
3	교양필수	직업윤리	2	2	교양필수	현대사회와 인간관계	2	2
		편입-생활영어1	-3	-3		편입-생활영어2	-3	-3
	교양선택	외국어 및 자유선택	3	3	교양선택	외국어 및 자유선택	3	3
	전공필수	웹 프로젝트연구	3	3	전공필수	게임프로젝트연구	3	3
		게임프로그래밍2	3	3	전공선택	온라인 게임개발	3	3
	전공선택	3D컴퓨터그래픽스2	3	3		모바일 프로그래밍1	3	3
		데이터베이스	3	3		시스템관리	3	3
		디지털콘텐츠분석 및 기획	3	3		디지털 애니메이션1	3	3
		네트워크프로그래밍	3	3		산업현장연구2	-3	
		현장실습	-2					
		산업현장연구1	-3					
	개설학점		23(8)	23(3)	개설학점		20(6)	20(3)
4	교양선택	외국어 및 자유선택	3	3	교양선택	외국어 및 자유선택	3	3
	전공선택	졸업프로젝트	3	3	전공선택	정보보호	3	3
		디지털사운드 처리	3	3		디지털콘텐츠 마케팅	3	3
		모바일 프로그래밍2	3	3		특수영상처리론	3	3
		디지털 애니메이션2	3	3		최신기술동향연구	3	3
		인공지능						
		산업현장연구3	-3					
	개설학점		18(3)	18	개설학점		15(3)	15
개설학점 총계	* 교양필수 : 26(6)학점 기초필수 : 18학점 교양선택 : 18학점 * 전공필수 : 21학점 전공선택 : 75(14)학점						총계 : 157 (20)학점	

표 III-4. 변경 예상 교육과정

학년	1학기				2학기			
	이수구분	교과목명	학점	시간	이수구분	교과목명	학점	시간
1	교양필수	대학영어1	2	3	교양필수	대학영어2	2	3
		대학외국어1(일본어)	2	3		대학외국어2(일본어)	2	3
		컴퓨터1	3	3		컴퓨터2	3	3
		대학과 인생	2	2		문화와 예술	2	2
	기초필수	멀티미디어개론	3	3	기초필수	기초수학	3	3
		컴퓨터과학개론	3	3		디자인개론	3	3
		게임학개론	3	3		C프로그래밍1	3	3
	개설학점		18	20	개설학점		18	20
2	교양필수	대학영어3	2	3	교양필수	대학영어4	2	3
	교양선택	외국어 및 자유선택	3	3	교양선택	외국어 및 자유선택	3	3
	전공필수	C프로그래밍2	3	3	전공필수	게임프로그래밍1	3	3
		자료구조	3	3		프로젝트수행방법론	3	3
	전공선택	컴퓨터그래픽디자인1	3	3	전공선택	컴퓨터그래픽디자인2	3	3
		웹 프로그래밍1	3	3		웹 프로그래밍2	3	3
		2D게임제작	3	3		JAVA프로그래밍2	3	3
		JAVA프로그래밍1	3	3		3D컴퓨터그래픽스1	3	3
		개설학점		23	24	개설학점		23
3	교양필수	직업윤리	2	2	교양필수	현대사회와 인간관계	2	2
		편입-생활영어1	-3	-3		편입-생활영어2	-3	-3
	교양선택	외국어 및 자유선택	3	3	교양선택	외국어 및 자유선택	3	3
	전공필수	웹 프로젝트연구	3	3	전공필수	게임프로젝트연구	3	3
		게임프로그래밍2	3	3	전공선택	온라인 게임개발	3	3
	전공선택	3D컴퓨터그래픽스2	3	3		모바일 프로그래밍1	3	3
		데이터베이스	3	3		시스템관리	3	3
		디지털콘텐츠분석 및 기획	3	3		디지털 애니메이션1	3	3
		네트워크프로그래밍	3	3		산업현장연구2	-3	
		산업체와 연계한 프로젝트 실습 1	3			산업체와 연계한 프로젝트 실습 2	3	
		산업현장연구1	-3					
	개설학점		23(8)	23(3)	개설학점		20(6)	20(3)
4	교양선택	외국어 및 자유선택	3	3	교양선택	외국어 및 자유선택	3	3
	전공선택	졸업프로젝트	3	3	전공선택	정보보호	3	3
		디지털사운드 처리	3	3		디지털콘텐츠 마케팅	3	3
		모바일 프로그래밍2	3	3		특수영상처리론	3	3
		디지털 애니메이션2	3	3		최신기술동향연구	3	3
		인공지능						
		산업현장연구3	-3			산업현장연구4		
		산업체와 연계한 프로젝트 실습 3				산업체와 연계한 프로젝트 실습 4		
	개설학점		18(3)	18	개설학점		15(3)	15
개설학점 총계	* 교양필수 : 26(6)학점 기초필수 : 18학점 교양선택 : 18학점 * 전공필수 : 21학점 전공선택 : 75(14)학점						총계 : 157 (20)학점	

산업체에서 요구하는 실무중심형은 대학에서 교육된 내용을 개발자 입장의 교육은 실제 제품으로 개발을 하였느냐 하는 것과, 유통 및 마케팅 측면의 교육은 실제로 게임을 사용자에게 유통하고 서비스를 해보았느냐는 것이 주된 요구사항이다. 이를 해결하기 위해서는 학생들이 실제 게임 개발사의 환경을 체험하고 학습할 수 있는 방법이 필요한데 산업체 측에서는 단기간에 학생들을 인턴쉽으로 보내는것에 대하여 부정적이며, 산업체의 요구를 수용하여 처리하는 교육과정은 1주일에 최소 2일정도 및 최소 1년, 적당한 관계는 2년정도 회사와 연계를 갖고 프로젝트를 수행할 수 있도록 하는 방안을 본 수요지향적 교육과정 개편에 담았다.

이는 추후 교육과정 개편시에 해당 교육과정이 산업체에서 요구하는 실무중심형 게임 인재의 양성에 반영되도록 할 예정이다.

### 3.2.2 국내 게임 전문 인력 양성 기관의 사례 분석

이번 장에서는 앞서 살펴본 국내 게임 산업 환경에서의 전문 인력 양성과 관련된 교육 시스템의 문제점과 개선 방향을 보다 구체적으로 살펴본다는 측면에서 국내의 게임 전문 인력 양성 기관의 실제 사례를 살펴보도록 하겠다. 국내의 게임관련 교육기관은 크게 정규 교육기관과 비정규 교육기관으로 나눌 수가 있다. 정규 교육기관은 다시 고등학교, 전문대학(전문학교포함), 대학교(산업대학 포함), 원격대학(교), 대학원으로 나눌 수가 있다. 그리고 비정규 교육기관은 고등학교 및 정규 대학, 대학원을 제외한 교육기관으로 세부적으로는 정부기관 산하단체에 의해 설립된 교육기관과 일반 개인 사설 학원으로 나누어진다.

표 III-5. 게임관련 정규 교육기관 추이(비정규기관 제외)

구분	2001.6	2002.4	2003.5	2004.3	2005.4	
고등학교	2	2	3	4	4	
전문대학	16	22	28	39	36	
대학교	일반	3	7	10	18	18
	원격	-	8	13	13	3
대학원(석사)	2	5	8	8	8	
전체	23	44	62	82	69	

이러한 게임과 관련된 교육기관은 정규교육기관과 비정규교육기관 모두 해마다 꾸준히 증가하는 추세이다. 2001년 6월 기준으로 게임관련 정규 교육기관은 23개였던 것이 해마다 꾸준히 증가하여 2004년에는 82개까지로 급속하게 증가하였다.

그러나 2005년 들어와서는 각 대학교의 구조조정으로 인하여 게임 관련학과가 다른 과로 통합되거나 폐지되어 게임기관 자체의 수는 총 69개 기관으로 교육기관의 수가 가장 많았던 2004년도에 비해서는 13곳이 줄어든 상태이다. 이러한 교육기관 수의 감소현상은 특히 원격대학과 전문대학에서 많이 나타나는데 전문대학은 3개의 교육기관에서, 원격대학 10개의 교육기관에서 게임관련 학과가 다른 과에 통합되거나 폐지되었다.

2005년 4월 기준 현재 게임관련 정규 교육기관은 전체 69 기관의 분포를 살펴보면 고등학교가 4개(6%), 전문대학이 36개(52%), 대학교가 18개(26%), 4년제 원격대학은 3개(4%), 대학원 8개(12%)이다. 특히 전체 정규 교육기관에서 전문대학이 차지하는 비중이 52%로서 게임관련 정규교육기관의 과 반수를 넘는다. 이를 보면 우리나라 게임 인력 양성이 주로 전문대학을 중심으로 이루어지고 있다는 것을 알 수 있다.

게임은 그 성격상 프로그래밍이라는 공학적인 요소와 디자인이라는 예체능요소가 모두 포함되는 관계로 게임 관련 정규교육기관에 개설된 게임 관련 학과가 속한 계열은 크게 공학계열과 예체능계열로 나눌 수 있다. 공학 계열에 속해 있는 학과는 총 62개, 예체능계열에 속해 있는 학과는 총 11개, 그리고 기타 계열에 속해 있는 학과는 총 2개로 공학계열에 속해 있는 경우가 압도적으로 많다.

교육기관별로 개설된 게임관련 학과는 고등학교가 4개 학교에 6개 학과, 전문대학은 36개 학교에 38개 학과, 대학교는 18개 학교에 18개 학과, 원격대학교는 3개 학교에 4개 학과, 대학원은 8개 학교에 9개의 학과가 있다. 이러한 각 교육기관에 개설된 게임관련 학과 명칭은 매우 다양하다. 공학계열에 속해 있는 학과는 주로 게임공학, 게임학과, 컴퓨터게임제작, 게임 프로그래밍 등의 이름으로 개설되어 있고 예체능계열에 속해있는 학과는 게임디자인, 게임애니메이션 등의 이름으로 학과가 개설되어 있다.

다음의 내용은 특정 학교와 학과의 내용이긴 하지만, 대다수의 게임

관련 학과가 경험하는 문제점을 살펴보고 개선 방향의 한 가지 사례를 제시한다는 점에서 나름대로 의미가 있다고 판단하여 본 자료의 내용에 추가하게 되었다. 참고로 이절에서 인용된 표 및 그림의 일부는 2005 게임백서를 참조한 것이다.

## 가. 고등학교

### 1) 서울디지털고등학교 전자게임과의 개요

현재 실업계와 일반계, 특수 목적 고등학교로 구분되어 있는 국내 고등학교 교육 체계 상에서도 게임 교육을 전문으로 진행하고 있는 고등학교가 이미 여러 곳 개설되어 있는 상황이다. (표 참조)

표 III-6. 게임 교육을 진행하고 있는 고등학교 현황

고교명	학과개설 년도	학과명	교원	입학 정원	졸업현황(인원)		
					계	취업	진학
한국애니메이션	2001년	컴퓨터게임제작	3	25	46	5	41
수원공업	2004년	디지털게임과	3	36	-	-	-
한국게임과학	2004년	게임과	19	100	-	-	-
울산애니원	2003년	컴퓨터게임개발과	4	25	-	-	-
미림여자정보과학	2004년	게임애니메이션	15	50	64	36	28
서울디지털	2001년	전자게임과	11	99	177	42	120
합 계			55	335	287	83	189

이 중 가장 초기에 설립된 서울디지털고등학교의 개요를 살펴보면 다음과 같다.

- 유성실업학교(1975) -> 청지공업고등학교(1995) -> 서울디지털고등학교 (2002) 로 변경, 2001년 전자게임과 및 인터넷통신과 신설
- 사립 고등학교로써 서울시 용산구 이태원 2동 214-4 소재
- 전자게임과 3학급 구성

### 2) 서울디지털고등학교 전자게임과의 교육 목표 및 방침

전자 게임과 교육목표는 다음과 같다. 컴퓨터 게임의 기획 및 게임 시나리오를 작성하고 컴퓨터 프로그램과 게임제작 도구를 사용하여 게임 제작 능력을 기른다. 또한 인터넷을 통한 정보 검색, 웹 문서 디자인, 창업 등에 대한 실무 능력을 배양하여 현장에 바로 적응할 수 있는 인력을 양성한다.

전자 게임과 교육방침은 게임 제작 요소 기술인 새로운 아이디어 발상, 창의적인 시나리오 창작 및 게임의 놀이 전략을 중심으로 창의성의 게임 개발 전략의 개발, 그래픽 표현기법, 화면 연출, 게임알고리즘 및 게임프로그래밍 기술, 게임인공지능 구현, 게임기획 및 게임 디자인 등의 게임개발 교육 과정을 통하여 현장 게임 개발 능력을 배양하고 게임의 특성인 상호작용(Interactive)의 게임 동작을 구현 할 수 있는 게임 이론과 현장 기술을 구현 하며 미래에 대비할 수 있는 현실 및 미래지향적인 교육에 역점을 두고 있다.

### 3) 서울디지털고등학교 전자게임과의 학과 특색 및 졸업생 진로

전자 게임과 학과의 특징은 창의적인 게임 시나리오 작성, 게임 프로듀싱 및 게임기획, 게임디자인, 게임그래픽, 게임프로그래밍, 게임사운드의 이해 등의 게임개발 및 게임제작의 제반 과정을 교육하며, 특히 게임분석 및 게임연구, 게임 동아리 및 그룹 스터디 등을 통한 팀별 학습활동 활성화와 개인별 적성과 소질에 따른 개인 특성화교육, 산학협력을 통한 현장 기술 교육 등을 통하여 게임별 교육계획을 수립하게 하여 교육계획이 정상적으로 수행하고 있는가를 점검 및 평가하는 교육 방식을 적용하고 있다.

전자 게임과 진로 및 취업분야를 살펴보면 게임개발사, 게임프로듀서, 게임시나리오 작가, 게임기획창작 전문가, 게임그래픽 디자이너 및 게임그래픽 분야, 게임프로그래머 및 게임프로그래밍 분야, 게임 캐릭터 및 캐릭터디자인 분야, 원화작도 분야, 네트워크 게임 분야, 프로그래머 활동, 게임스쿨 등의 게임제작 교육기관 강사, 인터넷 PC 및 게임방, 게임리그업체 진출, 게임작가 및 게임평론가, 게임평론가, 게임관련 벤처 창업, 영상산업 분야, 디지털애니메이션 분야, 게임마케팅 분야 등에 진출하는 것을 예상하고 있다.

전자 게임과 취득 자격증으로는 게임기획 전문가, 게임그래픽 전문가, 게임 프로그래밍 전문가, 컴퓨터그래픽스 기능사, 워드프로세서 1, 2, 3급, 정보기기 기능사, 컴퓨터 활용능력, 인터넷 정보 검색사 등 자격 취득을 권장하고 있다.

#### 4) 서울디지털고등학교 전자게임과의 교육 과정

표 III-7. 서울디지털고등학교 전자게임과의 교육 과정

구분	교과	과목	기준 단위	선택주제		학기별 단위수						계
				학교	학생	1학년		2학년		3학년		
						1	2	1	2	1	2	
보통 교과	국민공통 기본교과	도덕	2			1	1					2
		국어	8			2	2	2	2			8
		사회	6			3	3					6
		국사	4			2	2					4
		수학	8			4	4					8
		과학	6			3	3					6
		정보사회와 컴퓨터	6			3	3					6
		체육	4			1	1	1	1			4
		음악	2			1	1					2
		미술	2					1	1			2
	영어	8			4	4					8	
	선택교과	작문	8							2	2	4
		시민윤리	4							2	2	4
		수학 I	8					2	2	2	2	8
		생활과 과학	4					2	2			4
		영어 I	8					2	2	2	2	8
		체육과 건강	4							1	1	2
		일본어 I	6					2	2	1	1	6
	한문	6					1	1	1	1	4	
전문 교과	계열필수	공업입문	2	2			1	1			2	
		정보기술기초	4	4		2	2				4	
		전기전자통신제도	8	8				4	4			8
	이론및실 습교과	공업기술	2	2						1	1	2
		진로와 직업	4	2						1	1	2
		컴퓨터게임제작 I	6	6				3	3			6
		컴퓨터게임제작 II	24	24				7	7	5	5	24
		컴퓨터게임제작 III	10	10						5	5	10
		컴퓨터게임제작 IV	18	18						9	9	18
		컴퓨터그래픽스	4	4		2	2					4
		컴퓨터음악	4	4						2	2	4
프로그래밍	12	12					6	6		12		
재 량 동	교과재량	애니메이션기초	10	10		5	5				10	
	창의적재 량	인성교육	2			1	1				2	

특별활동	12			2	2	2	2	2	2	12
보통교과 이수단위	96			24	24	13	13	11	11	96
전문교과 이수단위	96			4	4	21	21	23	23	96
재량활동 (12단위)	12			6	6					12
특별활동 (12단위)	12			2	2	2	2	2	2	12
총 이수 단위	216			36	36	36	36	36	36	216

## 나. 2년제 대학

2005년 현재 게임 관련 학과는 전문대학 30개 학교에 개설되어 있다. 전문대학의 게임학과 개설은 전체의 52%를 차지하고 있으며 이러한 배경을 바탕으로 전문대학은 게임 산업에서 필요한 인재를 양성하여 게임 산업 현장에 빠르게 배출하고 있는 상황이다.

전문대학에서 게임관련 교육은 공학계열과 예체능 계열에 나누어 개설되어 있는데 전문대학에 개설된 게임관련 학과는 현재 공학계열에 31개, 예체능계열에 8개가 개설되어 있다.

공학계열에 개설된 주요 학과로는 게임멀티미디어, 게임프로그래밍, 컴퓨터게임과가 있다. 군장대학, 두원공과대학, 숭의여자대학, 창신대학, 청강문화산업대학, 한국재활복지대학, 혜천대학에는 컴퓨터게임학과가 개설되어 있으며, 동아방송대학, 영진전문대학에는 게임프로그래밍과가 개설되어 있다. 예체능 계열에는 경동정보대학, 동강대학, 동아방송대학, 서라벌대학, 울산과학대학, 주성대학 등에 게임 디자인 및 게임 애니메이션과가 개설되어 있다.

표 III-8. 전문대학 계열별 학과 현황

계열	세부 전공	학과 수
공학계열	게임&소프트웨어전공	1
	게임그래픽학과	1
	게임기획학과	1
	게임멀티미디어	1
	게임멀티미디어과	1
	게임웨어	1
	게임전공	1
	게임정보미디어과	1
	게임콘텐츠전공	1

	게임제작과	1
	게임창작과	1
	게임컨설팅과	1
	게임프로그래밍과	1
	게임프로그래밍전공	2
	게임프로그래밍학과	1
	게임학과	1
	멀티미디어게임	1
	사이버게임	1
	사이버게임과	1
	온라인게임전공	1
	인터넷게임	1
	컴퓨터게임&그래픽전공	1
	컴퓨터게임과	6
	컴퓨터게임전공	1
	컴퓨터게임제작과	1
소계		31
예체능계열	게임디자인	1
	게임디자인과	1
	게임디자인전공	1
	게임애니메이션과	1
	게임애니메이션전공	2
	게임캐릭터디자인과	1
	게임컨텐츠디자인	1
소계		8

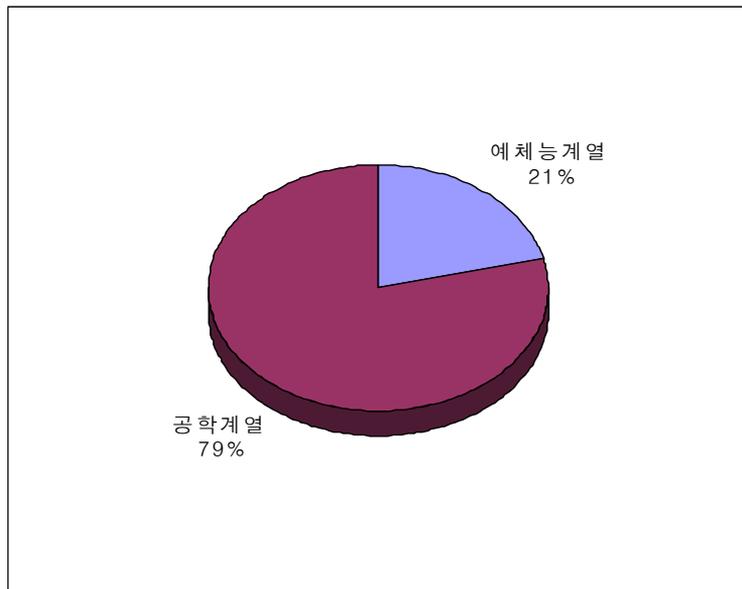


그림 III-1. 전문대학 계열별 게임학과 분포

표 III-9. 전문대학 게임관련 학과 현황

대학명	년제	모집단위	세부전공
경남정보대학	2	인터넷응용계열	온라인게임전공
경동정보대학	3	게임애니메이션과	게임애니메이션과
계원조형예술대학	2	게임웨어	게임웨어
계명문화대학	3	멀티미디어학부	게임콘텐츠전공
군장대학	2	컴퓨터응용계열	컴퓨터게임전공
김천과학대학	3	컴퓨터정보계열	아동컴퓨터게임전공
김천대학	3	컴퓨터멀티미디어계열	게임멀티미디어과
대경대학	2	컴퓨터정보과학부	인터넷게임
대구과학대학	3	컴퓨터계열	게임&소프트웨어전공
대구미래대학	2	공학계열	게임창작과
대덕대학	2	컴퓨터게임제작과	컴퓨터게임제작과
동강대학	2	컴퓨터정보계열	게임애니메이션전공
동부산대학	2	게임컨설팅과	게임컨설팅과
동아방송대학	2	게임제작계열	게임디자인전공 게임프로그래밍전공
동주대학	2	멀티미디어계열	멀티미디어게임
두원공과대학	2	전자/정보/건축계열	컴퓨터게임과
서라벌대학	2	게임콘텐츠디자인	게임콘텐츠디자인
서울정보기능대학	2	컴퓨터게임과	컴퓨터게임과
승의여자대학	3	공업계열	컴퓨터게임과
영진전문대학	3	컴퓨터정보계열	게임프로그래밍전공
용인송담대학	3	컴퓨터게임정보과	컴퓨터게임정보과
울산과학대학	3	컴퓨터정보학부	게임애니메이션전공
장안대학	2	게임전공	게임전공
전남과학대학	2	IT계열	게임제작과
주성대학	2	디자인/컴퓨터	게임디자인과
창신대학	2	공학계열	컴퓨터게임과
청강문화산업대학	2	게임/애니메이션계열	컴퓨터게임과
한국재활복지대학	2	컴퓨터게임개발과	컴퓨터게임과
혜전대학	2	컴퓨터과	게임애니메이션전공
혜천대학	2	컴퓨터멀티미디어계열	컴퓨터게임&그래픽전공

전문대학의 게임학과는 소속된 계열에 따라 다소 상이한 교육과정과 교과목의 구성을 보이고 있다. 공학계열에 개설된 게임학과는 컴퓨터 프로그래밍 관련 교과목의 비중이 높은 반면에 예체능계열에 속한 게임학과는 디자인, 애니메이션, 그래픽에 관련된 교과목 비중이 높다.

이와 같이 전문대학은 계열별로 게임 관련 전공이 세분화되어 있으며 각 게임학과가 속한 계열의 특성에 맞는 교육과정을 구성하고 있다. 그리고 세부적인 교육의 진행 방법은 게임 산업 현장에서 바로 투입되어 일할

수 있는 인재를 양성하기 위해 현장 실습 교과목과 게임 프로젝트 진행과 관련된 수업을 병행해서 진행하고 있다.

### 1) 청강문화산업대학 게임 교육 시스템의 특징

청강문화산업대학의 컴퓨터게임과는 게임애니메이션계열에 개설되어 있다. 이 학교의 컴퓨터게임과에서는 산업 현실에서 필요로 하는 높은 수준의 전문 게임제작 기술 인력의 육성을 위하여 다양하고 효과적인 교육 과정을 제공하는데 교육목표를 두고 있다.

세부적인 교육과정을 살펴보면 첫 학기에는 게임의 개요와 게임산업 전반에 대한 이해, 게임제작을 위한 다양한 요소인 게임그래픽, 게임프로그래밍, 게임기획, 게임사운드 등에 대한 기초적인 교육을 실시한다.

그러고 난 뒤에는 학생 개개인의 적성과 관심도에 따라 게임기획 분야, 게임그래픽 분야, 게임프로그래밍 분야, 게임사운드분야의 의 4개 분야 중 하나를 선택하여 공부하게 되는 일종의 세부 전공제도를 적용하고 있다.

이러한 세부 분야에 대한 집중적인 교육은 게임산업만이 가지는 고유한 특성, 즉 공학-예술-인문학이 연계된 컴퓨터게임 제작 과정의 종합 문화산업적인 특성을 최대한 고려한 형태의 교육과정이며, 최종 학기에는 4개 세부 분야별 학생들이 단일한 소규모 팀을 구성하여 실제 게임 제작 프로젝트를 수행하여야만 졸업을 할 수 있는 시스템을 운영하고 있다.

청강문화산업대학 컴퓨터게임과의 특징은 첫째, 게임프로그래밍, 게임그래픽, 게임기획, 게임사운드로 세분화된 전공 심화 교육을 실시하고 있다. 게임제작을 위해 필요한 요소들에 대한 전반적인 이해를 통해 자신의 적성과 능력에 맞는 게임제작의 세부 분야를 선택하고, 각 분야별로 전문화된 심화 교육을 본인의 목표와 취향에 따라 선택 수강하는 방식을 통해 보다 창의적이고 전문화된 게임 개발 교육 시스템을 제공하고 있다.

둘째, 같은 계열에 속한 애니메이션과, 만화창작과 등 유관 학과의 실질적인 연계 교육을 실시하고 있다. 세부 선택 분야인 게임프로그래밍, 게임그래픽, 게임기획, 게임사운드 간의 공동 프로젝트 진행은 물론, 동일 계열의 학과와 수업 공유, 공동 졸업작품전 실시를 통해 실질적인 '원소스 멀

티유즈'가 가능하도록 하고 있다.

셋째, 국내외 유명 게임전시회 및 공모전 참여를 통한 체험 교육을 실시하고 있다. 2년여 간의 수업 기간 중 게임산업 현장에 계신 교수님과 관련 전문가들의 도움을 통해 스스로의 게임을 완성하고, 이를 G-star (지스타)와 국내외 각종 공모전을 통해 공개함으로써 자신의 수준과 능력을 산업체의 눈을 통하여 직접 점검해 보는 교육을 실시하고 있다.

넷째, 실질적인 산학 연계를 목표로 하는 현장 교육을 실시하고 있다. 정규적인 수업은 물론, 수시로 개최되는 게임산업 관련 전문가의 특강이나 간담회 등을 통하여 게임산업계의 현실을 보다 가깝게 체험하는 한편, 스튜디오 활동과 각종 산학 연계 프로젝트에 직접 참여하여 효과적이고 실질적인 현장 교육이 되도록 지원하고 있다.

다섯째, 해외 유학의 기회가 주어지는 국제 교육을 실시하고 있다. 우수한 자질과 열정을 가지는 학생의 경우, 국내 대학교나 전문 교육 기관으로의 지속적인 학습 활동을 지원하는 것은 물론, 2학년 과정 중 일본의 JAM과 프랑스, 미국, 중국 등으로의 연수 지원 프로그램 실시를 통하여 보다 창의적이고 체계적인 능력을 가질 수 있도록 지원하고 있다.

이러한 학과 특성을 바탕으로 실제 수업진행은 창의적인 게임 시나리오 작성, 게임 기획 및 마케팅, 게임 그래픽, 게임 프로그래밍, 게임 사운드의 이해 등 게임 개발 및 제작의 제반과정과 관련된 세부적인 내용과 소양을 교육하고 있다. 그리고 각종 게임 동아리 활동 및 그룹 스터디 등을 통한 팀 별 학습 활동 활성화와 개인별 적성과 소질에 따른 개인별 교육 계획 수립 및 적용, 실질적인 산학 협력을 통한 현장 기술 교육 연계 등을 실시하고 있다.

청강문화산업대학 컴퓨터게임과는 1999년 학과 개설 이래 학과의 편제와 교육 시스템 운영에 있어 다양한 시도와 시행착오를 거쳐 현재의 교육 시스템에 이르고 있다. 사실 상당수의 게임 관련 교육 기관이 청강문화산업대학 컴퓨터게임과와 상당히 유사한 형태의 과정을 거치고 있는 것으로 보이는데, 본 사례 소개가 이러한 손실을 최소화하는데 도움이 되었으면 한다. 청강문화산업대학 컴퓨터게임과의 게임 교육 시스템의 변화 내용과 주요 문제점은 다음과 같은 것이 있었다.

## 2) 학과 개설 당시 (1999년)

- 개요 : 입학 정원 전원은 동일 조건으로 입학, 1학기 기본 수업 후, 1학년 2학기 게임프로그래밍/게임그래픽 분야별 고정 정원으로 전공 분리, 교과과정은 전공 2개로 독립적으로 구성
- 문제점 : 각 교과과정은 해당 전공 분야만 기술 위주로 집중 교육하여 타 전공 및 전체 프로세스에 대한 이해도가 떨어짐

## 3) 게임디자인 분야 추가 및 전공별 독립 운영 (2000년 ~ 2001년)

- 개요 : 입학 전형 시부터 게임프로그래밍/게임그래픽/게임디자인(기획) 세 전공 분야로 고정 정원으로 분리 모집하여 마치 3개의 독립과처럼 운영, 교과과정은 전공 3개 과정으로 독립적으로 구성하되 타 전공 비중을 약간 높임
- 문제점 : 입학 시 전공별 지원률 및 커트라인 상이로 전공 간 위화감 발생, 각 교과과정은 기존 연도에 비하여 타 분야의 비중을 늘렸으나, 여전히 전공별 수업이 독립적으로 운영되는 탓에 타 전공 및 전체 프로세스에 대한 이해도가 떨어짐

## 4) 전공 제도 폐지 및 세부 분야 선택 방식으로 전환 (2002년 이후 ~)

- 개요 : 입학 전형 시 분야 구분 없이 모집, 프로그래머 등 비제작 인력 배제
- 개선방향
  - ① 1학년 1학기 모든 학생이 의무적으로 이수해야 하는 공통 기초 과정 (게임그래픽/프로그래밍/기획/사운드/어학 등) 수강 후, 본인의 적성 및 관심도에 따라 1학년 2학기부터 분야별 인원 제한 없이 개인별 관심 분야 자율 선택
  - ② 교과과정은 다양한 과목 pool 이 혼재되어 있는 단일한 1개로 개편, 학생 개개인이 자율적으로 선택하여 수강하는 개별적인 교과 시스템 진행
  - ③ 입학생 전원은 게임그래픽과 게임프로그래밍, 게임디자인을 희

망하는 학생 각 1인 이상이 포함된 복합 프로젝트팀 구성, 2학년 2학기까지 게임 작품 제작

- ④ 게임 제작을 학생 개인의 졸업 기본 조건으로 강화하는 한편, G-STAR 등의 전시를 기본 목표로 해서 게임 제작에 대한 구체적인 목표 부여

표 III-10. 청강문화산업대학 컴퓨터게임과 교과과정표

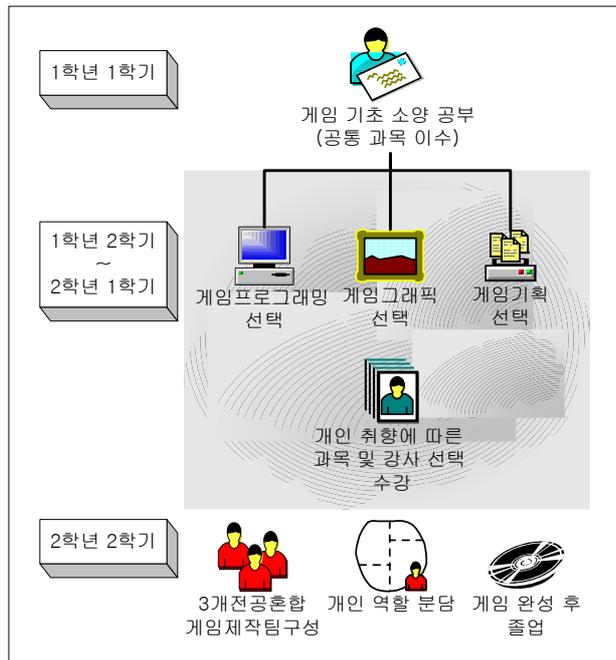
구 분		교 과 목 명 (국문)	교 과 목 명(영문)	
교양과목	공통필수	기독교문화와의만남 I or 채플 I 중 택1 기독교문화와의만남 II or 채플 II 중 택1 일과보람있는삶 봉사활동 청강문화체험		
	일반선택	생활레포트I 몸의 이해 생각과 표현 우리문화의 이해 생활레포트II 대학생 리더쉽 세계문화의 이해 문화콘텐츠의 이해		
	어학선택	교양일어 1 또는 교양영어 1 교양일어 2 또는 교양영어 2		
전ier과목	필수	게임학개론 및 아이디어 발상훈련 (공통)	Introduction of Computer Game	
		게임그래픽기초 (G)	Introduction of Game Graphics	
		게임프로그래밍기초 (P)	Introduction of Game Programming	
		게임기획기초 (D)	Introduction of Game Design	
		게임사운드 기초 (S)	Introduction of Game Sound	
		게임스토리보드제작 (제작)	Project Training of Game Storyboard	
		게임제작프로젝트실습 I (제작)	Project Training of Game Development I	
		게임제작프로젝트실습 II (제작)	Project Training of Game Development II	
		졸업작품연구 (제작)	Advanced Game Development	
	산업현장실습 (공통)	Internship Training		
	선택	1학년 1학기	게임 컨셉디자인 (공통)	Concept Design for Computer Game
			기초드로잉 (G)	Drawing
			2D CG 의 이해와 실습 (G)	2D Computer Graphics
			게임모델링 (G)	Training of Game Modeling
			게임자료구조 (P)	Data Structure for Computer Game
			게임시스템운영체제 (P)	Operating System for Computer Game
			객체지향프로그래밍 (P)	Object-Oriented Programming
			콘텐츠비즈니스정보기술 (D)	Introduction of Contents Business
			게임분석기초 (D)	Introduction of Game Analysis
			게임제작기법의이해 (D)	Introduction of Game Development Method
대중음악과 게임(S)			Popular Music and Game	
게임컴퓨터음악(S)	Game Computer Music			
1학년 2학기	웹게임제작 (공통)	Training of Web Game Development		
	2D게임캐릭터제작 (G)	Training of 2D Game Character		
	게임원화제작 (G)	Training of Game Artwork		

		디지털페인팅 (G)	Digital Painting
		매핑과 라이팅 (G)	Mapping and Lighting
		게임그래픽프로그래밍 (P)	Game Graphic Programming
		게임데이터베이스프로그래밍 (P)	Database Programming for Game Development
		게임툴프로그래밍 (P)	Game Tool Programming
		게임알고리즘 (P)	Introduction of Game Algorithms
		게임기획사례연구 (D)	Case Study of Game Design
		게임분석 및 설계 (D)	Game Analysis and Design
		게임인터페이스 기획 (D)	Game Interface Design
		게임기획서 작성 (D)	Training of Game Proposal Preparation
		게임 배경음악과 효과음 (공통)	Game Back Ground Music and Sound Effect
		게임음향학(S)	Game Acoustics
		게임사운드디자인(S)	Game Sound Design
	2학년 1학기	게임네트워크의 이해 (공통)	Introduction of Game Network
		인간과컴퓨터상호작용 (공통)	Human Computer Interface
		컴퓨터애니메이션 (G)	Training of Computer Animation
		3D게임캐릭터제작 (G)	Training of 3D Game Character
		게임배경 및 오브젝트제작 (G)	Training of Game Background and Object
		게임그래픽인터페이스 (G)	Game Graphic Interface
		3D게임프로그래밍 (P)	3D Game Programming
		네트워크프로그래밍 (P)	Network Programming
		게임엔진분석 (P)	Analysis of Game Engine
		온라인게임프로그래밍 (P)	Online Game Programming
		고급게임기획 (D)	Advanced Game Design
		게임제작프로젝트관리 (D)	Management of Game Development Project
		게임디자인실무기법 (D)	Business Practice of Game Design
		게임레벨디자인 (D)	Game Level Design
		게임마케팅기초(D)	Introduction of Game Marketing
		게임음악제작실습(S)	Project Training of Game Music
		디지털음향편집(S)	Digital Sound Editing
		2학년 2학기	디지털컨텐츠관리 (공통)
	게임특수효과 및 가상현실기술 (공통)		Introduction of Game Virtual Reality
	게임그래픽포트폴리오 (G)		Portfolio for Game Graphics
	게임동영상제작 (G)		Training of Game Trailer Movie
	게임시스템개발 및 최적화기법 (P)		Game Development Optimization
	게임인공지능 (P)		Introduction of Artificial Intelligence
	게임마케팅론 (D)		Introduction of Game Marketing
	게임 비즈니스의 이해(D)		Introduction of Game Business
	멀티플랫폼게임기획(D)		Multi-platform Game Design
	게임이벤트연출 (D)		Producing of Game Event
	게임음악작곡과편집(S)		Game Music Composition and Editing
	게임사운드비즈니스(S)		Game Sound Business

5) 청강문화산업대학 컴퓨터게임과의 교육 시스템 개선의 방향

앞서 살펴본 바와 같이 청강문화산업대학 컴퓨터게임과는 다양한 시행착오를 거친 후 현재와 같이 과목 및 세부 분야에서는 자율 선택권을 부여하되 최소한의 기본적인 소양과 게임 제작 경험은 엄격하게 관리하는 방식을 채택하고 있다.

이러한 현재의 교육시스템을 구성하는 기본적인 방향을 요약하면 다음과 같다.



- \* 방학 중 교내 특강 실시 및 전문 과정 수강 상담
- \*\* 학기 중 분야 및 수준별 학생 세미나 참석 활성화
- \*\*\* 계열 우수 학생 해외 연수 및 특강 참여 지원

그림 III-2. 컴퓨터 게임과의 교육 시스템

- ① 국내 게임 산업계 현실을 고려한 학과 목표의 재설정
  - 교육 목표의 설정 : 게임 개발의 경험을 보유한 말이 통하는 개발 인력의 양성
  - 목표 배출 인력 : 그래픽/프로그래밍/기획 각 해당 분야 중급 수준 실무 개발자
- ② 기존의 문제점이 개선된 교육시스템 운영 시도

- 틀 위주가 아닌 소양과 기술을 접목하는 교과과정 개발, 운영
- 프로그래밍/그래픽/기획 분야의 상호 교류 가능성 확대
- 최근의 학생 개성 다양화 추세에 맞춰 학생 선택권을 최대한 부여하도록 노력

③ 학과 및 개인별 정체성의 확립 노력

- 중장기 관점의 청강문화산업대학 컴퓨터게임과의 전략 수립 지속 시도
- 게임 산업의 광범위한 인력 구조를 최대한 단순화하여 교과 과정에 수용
- 기존 학교 및 학과 홍보 위주의 전시회 성격을 학업의 궁극적 목표로 부여
- 현장실습, 과제진행, 개인상담, 각종 특강 등을 통해 개인별 목표수립 유도

④ 산업계 동향의 반영 및 목표 배출 인력 층의 설정

- 게임 산업 특성에 적합한 산학 협력 모델 개발 및 운영
- 게임 산업 인력 양성 관련 정부 정책 및 각종 연구에 적극적으로 참여
- 게임 관련 직무 분석을 통한 정확한 목표 배출 인력 대상 선정 노력 지속 진행

5) 청강문화산업대학 컴퓨터게임과의 전문 인력 양성 방향

청강문화산업대학 컴퓨터게임과가 지향하고자 하는 전문 인력 양성의 방향은 “게임 개발의 경험을 보유한 말이 통하는 개발 인력의 양성” 으로 요약할 수 있다.

이를 위해 학과는 크게 다섯 개의 교육 흐름을 진행하고 있으며 각각의 흐름은 다음과 같다.

우선, “정규 교과 과정” 은 입학생 전체가 반드시 수강하여야 하는 ‘학과 기초 공통 과정’ 과 학생 개개인이 적성과 목표에 따라 선택하는 ‘개인별 선택 과정’ 으로 구분하여 운영하고 있다. 특히, 다양한 과목이 제공되는 ‘개인별 선택 과정’ 은 게임그래픽, 게임프로그래밍, 게임디자인(기획)의 개략

적인 흐름과 각 분야별 표준 과정은 학과 차원에서 학생에게 제공되지만, 기본적으로 어떠한 분야의 과목을 선택하여 듣는지는 학생 개인의 선택에 달려있는 방식으로써 최근의 융합 환경에 걸맞는 학생을 양성할 수 있는 방식이라 할 수 있다.

두 번째 흐름으로는 정규 교과과정 내에 “게임 제작 과정”을 별도로 운영하되 이는 전공 필수 개념을 도입하여 해당 과정을 반드시 이수하게 하고 있다. 1학년 1학기에는 아이디어 발상과 게임 컨셉 제작을 실습하며, 1학년 2학기에는 ‘게임스토리보드 제작’ 과목을 통하여 본인의 의사와 관계없는 임의의 팀을 계속 구성하여 다섯 개의 소규모 프로젝트를 경험하게 된다. 이를 기반으로 1학년 말에 임시 게임 제작팀을 구성하게 되며, 2학년 1학기의 ‘게임제작프로젝트실습 I’ 과목, 2학년 2학기의 ‘게임제작프로젝트실습 II’ 과목, ‘졸업작품연구’ 과목을 통하여 최종적인 게임을 제작하여 각종 공모전과 G-STAR 전시회에 출품하는 과정을 직접 경험한다. 이는 실제 실무 제작 경험을 가진 졸업생을 배출하고자 하는 목표에 따라 진행되는 일종의 특별 과정으로써 현재 청강 컴퓨터게임과의 가장 중요한 교육 흐름이라고도 할 수 있다.

세 번째 흐름은 졸업 이후의 “청강 게임 스튜디오(CCRC)”를 통한 실무 확장형 교육 개념이다. 2년이라는 전문대학의 한계를 극복하기 위하여 구상된 스튜디오 개념은 일종의 ‘2+1’의 확장형 교육 시스템의 사례로써 희망 학생은 일정한 선발 과정을 거쳐 2학년 졸업 후 실제 게임 산업 형태와 유사한 학교 내 스튜디오에서 1년여의 기간 동안 직접 연구원 및 개발자로서 게임 제작에 참여하는 실무형 교육을 받게 된다. 스튜디오는 기본적으로 산학 연계의 모델을 가지고 있으며, 제작 작품은 상용화와 연계되어 실제 참여 학생의 경력과 실무 능력 향상에 큰 도움을 주게 된다. 이는 최근 국내 게임 산업계에서의 인력 채용이 신입 보다는 경력 위주로 이루어지고 있는 현실에 대한 일종의 해결책인 동시에 전문대학의 2년 과정이 지니는 한계를 극복하기 위한 보완책이기도 하다. 특히, 계열 내 애니메이션과와 만화창작과가 이미 성공적으로 운영하고 있는 각각의 스튜디오와 직간접적으로 프로젝트를 연계 수행함으로써 이론적으로 배운 이른바 ‘원소스멀티유즈’ 개념을 직접 체험하고 자신의 창의력을 반영시키는 경험을 쌓게 된다.

네 번째 흐름은 위의 세 가지 흐름을 뒷받침하는 추가적인 교육 보완 시스템이다. 2003년 이후 방학 중에는 외부 전문가가 진행하는 1-2주간의 '전문 제작 워크샵' 이 합숙 과정을 통해 실시되며 이는 향후 고등학교 및 상위 과정과의 연계를 통해 확장될 예정이다. 또한, 학과 개설 과목의 숫자나 범위의 부족함을 보완하기 위해 2004년부터 학생에게 게임아카데미의 온라인 교육을 권고하여 이를 추가 점수 형식으로 학점에 반영하고 있으며, 각종 전문가 특강과 세미나 참석은 물론 창의력 부여를 위한 뮤지컬이나 영화 단체 관람 등의 문화 체험 행사도 2005년부터 학과 차원에서 전면적으로 실시하고 있는 상황이다.

마지막인 다섯 번째 흐름은 학생의 수업 외적 활동과 관련된 부분인데, 학생회 및 동아리 활동의 활성화와 학기 중 취업이나 산학 연계 활동, 해외 우수 교육 기관으로의 연수 지원 등을 들 수 있다. 이는 게임 제작 교육이 갖는 가장 독특한 특징이라 할 수 있는 학생 개개인의 열정과 노력을 최대한 지원하기 위한 방안이다. 학생과 관련한 지원은 정보통신부나 문화관광부의 동아리 지원 사업이나 각종 공모전을 통하여 다양한 성과를 보이고 있는데, 청강문화산업대학 컴퓨터게임과가 주력하는 또 하나의 독특한 교과 과정이기도 하다.

#### 다. 4년제 대학

2005년 현재 게임 관련 학과는 18개 대학교에 개설되어 있다. 대학교의 게임학과 개설은 전체의 26%를 차지하고 있으며, 전문대학과 함께 고려하면 전체 게임관련 교육기관의 78%를 차지하기 때문에 대학에서의 인재 양성은 국내게임 산업의 발전에 중요한 역할을 담당하고 있다.

대학교의 게임관련 학과는 공학계열과 예체능 계열에 나누어 개설되어 있는데, 대학교에 개설된 게임관련 학과는 공학계열에 89%, 예체능계열에 11%가 개설되어 공학계열로 편중되어 있음을 알 수 있다.

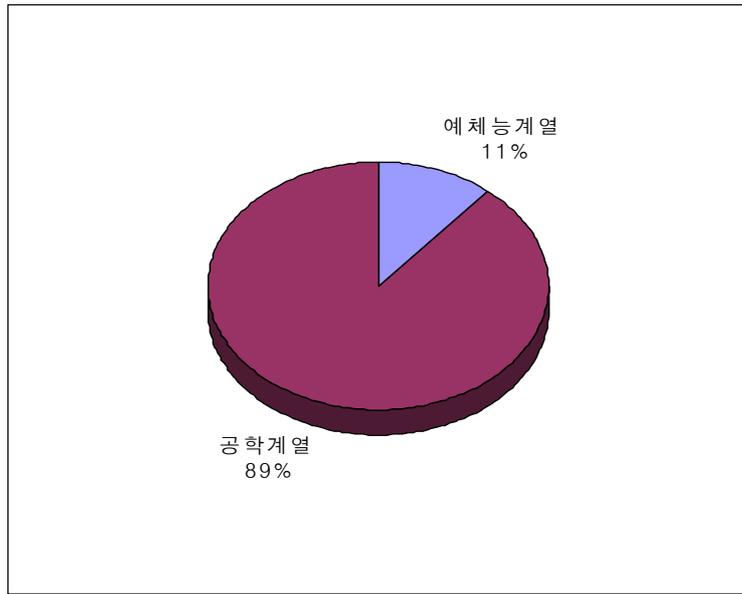


그림 III-3. 대학교 계열별 게임학과 분포

2004년까지는 대학교 게임교육기관의 수가 급속히 증가했으나, 2005년에는 18개로 변화가 없었다. 이는 각 대학이 새로운 게임학과 개설보다는 게임전문 인력 양성에 중점을 두고 있다고 해석할 수 있다.

표 III-11. 대학교 계열별 게임학과 현황

계열	세부 전공	학과수
공학계열	게임개발학과	1
	게임공학	1
	게임공학과	3
	게임공학전공	1
	게임디지털콘텐츠공학	1
	게임멀티미디어공학	1
	게임멀티미디어전공	1
	게임멀티미디어학과	1
	게임전공	1
	전산게임학과	1
	컴퓨터게임	1
	컴퓨터게임산업전공	1
	컴퓨터게임전공	1
	컴퓨터게임학전공	1
소계		16
예체능계열	게임디자인학과	1
	게임애니메이션전공	1
소계		2

표 III-12. 대학교 게임관련 학과 현황

대학명	모집단위	세부전공
건양대	전산게임학과	전산게임학과
공주대	게임디자인학과	게임디자인학과
극동대	정보통신학부	게임디지털콘텐츠공학
대구한의대	멀티미디어학부	컴퓨터게임산업전공
동명정보대	게임공학과	게임공학과
동서대	디지털콘텐츠학부	게임멀티미디어공학
배재대	공과대학	게임공학과
영산대	멀티미디어공학부	컴퓨터게임
예원예술대	만화게임영상학부	게임애니메이션전공
우송대	게임/IT계열	게임멀티미디어학과
위덕대	게임학부	게임멀티미디어전공
중부대	컴퓨터정보학부	게임공학
탐라대	게임개발학과	게임개발학과
한국산업기술대	게임공학과	게임공학과
호남대	게임애니메이션학부	컴퓨터게임학전공
호서대	컴퓨터공학부	게임공학전공
호원대	컴퓨터학부	컴퓨터게임전공
홍익대	소프트웨어게임학부	게임전공

대학교의 교육 내용은 해당 학과나 전공이 소속된 계열이나 학부에 영향을 받고 있다. 게임 전공에 필수적인 교과목은 전공이 소속된 계열이나 학부에 무관하게 개설되어 있지만, 그 외 교과목은 전공의 소속에 따라서 영향을 받고 있다.

공학계열에 개설된 게임학과의 경우는 컴퓨터 프로그래밍 관련 교과목의 비중이 높은 반면에 예체능계열에 속한 게임학과의 경우는 디자인, 애니메이션, 그래픽에 관련된 교과목 비중이 높다.

대학교의 교과과정은 영역별 전문인력 양성을 위한 커리큘럼을 가지고 있으며, 게임 산업의 요구를 반영하여 교육과정을 편성하고 있다. 또한 졸업 프로젝트를 통하여 게임 산업 현장에서 바로 투입되어 일할 수 있는 인재를 양성하고 있다.

### 1) 우송대학교 운영 사례

우송대학교 게임멀티미디어학과는 2004년도에 신설되었다. 게임멀티미디어학과 교육목표는 송대학교 게임멀티미디어학과는 차세대 핵심성장동력(GEOK)으로 부상하는 게임미디어콘텐츠를 비롯한 디지털문화산업분야에

창조적 역할을 수행할 실무형 인력양성의 메카로서, 정보기술(IT)과 문화기술(CT) 중심의 맞춤형 산학교육시스템을 구축하고, 컴퓨터와 인터넷 그리고 무선통신기술을 기반으로 하는 게임소프트웨어를 개발하기 위한 총체적인 기반기술을 체계적으로 교육함으로써 컴퓨터게임, 게임애니메이션, 그리고 휴대용 모바일 단말기게임 등 게임소프트웨어의 인프라 개발 전문가를 양성하고 있다.

고부가가치 게임멀티미디어 정보산업 분야에서는 창의력을 갖춘 고급인력을 필요로 하며, 특히 게임 및 멀티미디어 기술의 기본 이론과 활용능력을 갖춘 고급인력과 전문적인 능력을 지닌 고급 소프트웨어 개발 인력이 매우 부족한 실정이다. 이러한 요구에 부응하기 위하여 게임멀티미디어학과에서는 게임 및 멀티미디어시스템 기술과 콘텐츠 제작에 필요한 기획 및 기술의 이론과 실무가 병행하는 교육을 실시하여 고 부가가치 정보산업에 필요한 실무형 인재 교육양성에 그 목표가 있다.

게임멀티미디어학과는 첨단 게임 기술 및 멀티미디어 콘텐츠의 기획, 개발, 응용 분야에 전문 인력의 수요가 급증함에 대비하여 우수한 교수진 및 최신시설을 갖추고 산업체와 프로젝트형 수업을 통해 차세대 고부가가치 문화 산업 창출에 선도적인 역할을 담당할 고급 실무 인력을 양성하고 있습니다. 특히, 2004년도에 신설된 학과로서 교육인적자원부 NURI(지역대학 혁신 역량 강화 사업)사업의 “첨단 영상·게임 산업 전문 인력 양성사업”에 선정되어 향후 5년간(총사업규모 150억) 게임분야에 집중 투자하여 국내외 최고의 게임멀티미디어 관련 교육기관으로 특성화를 하고 있으며, 국제공인 Web3D 인증센터 및 밀착형 산학협력을 통하여 게임산업 실무 인재 양성 과정이 산학프로그램으로 함께 이루어지고 있습니다. 특히, (재)한국게임산업개발원이 지정한 지역인력양성기관(대전게임아카데미)으로 선정되어 대전광역시 및 대전엑스포과학공원, 첨단산업진흥재단과 공동 사업 추진 및 한국전자통신연구원 디지털콘텐츠 사업단, (주)이머시스, (주)레인버스스튜디오, (주)모션테크놀러지, (주)에버소프트, (주)드림스케이프 등 게임영상 관련 산업체와의 밀착형 산학교육프로그램을 운영 하여, 전문인력 양성에 힘쓰는, 대전을 대표하는 게임학과로서 성장하고 있다. 특히, 학생들의 특기 적성별 교육과정을 도입 프로그래밍/기획/그래픽/게임음향 파트별 전문화된 교육을 밀착

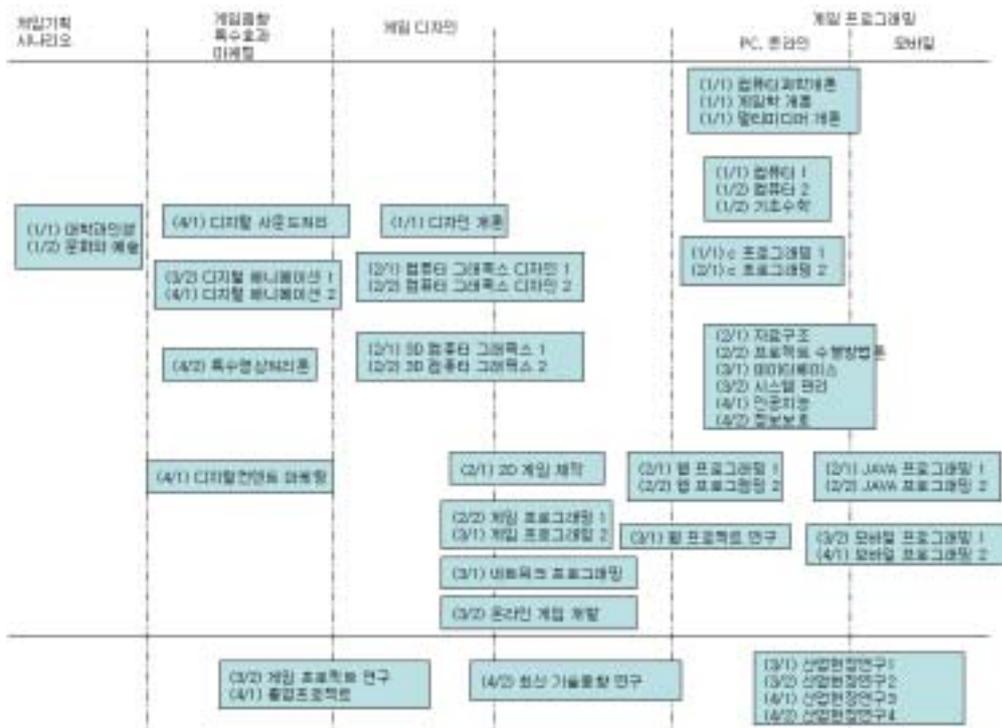


그림 III-4. 우송대학교 게임 멀티미디어학과 교과과정

형 산학협력을 통하여 상품화교육을 강화하여 게임개발 프로젝트 경험, 산업체 현장실습 교육, 콘텐츠 창작력을 위한 애니메이션 전공 및 디자인전공과의 연계교육이다. 이와 같은 교육목표를 달성하기 위하여 다음과 같은 전략을 가지고 있다. 첫째, 게임업체와 산학공동의 개발연구를 통해 생산적인 교육체계를 구축하여 기반지식과 실무능력을 극대화 한다. 둘째, 연구중심 교육환경을 구축하고 차세대 신기술개발 연구에 집중하여 명실 공히 세계적인 연구 경쟁력을 갖춘다. 셋째, 게임업체의 전문 인력을 적극 유치하여 산업현장에 대한 기술교육을 강화한다. 넷째, 학생들의 창업동아리를 활성화 하여 풍부한 개발경험을 갖출 수 있도록 한다. 게임공학전공은 아래 표와 같이 게임프로그램에 관련된 교과목이 압도적으로 많지만, 게임기획과 게임그래픽 관련 교과목, 게임음향 교육도 균형감 있도록 제공되고 있다.

우송대학교 게임멀티미디어학과 주요 분야는 게임기획 및 시나리오, 게임 디자인, 게임프로그래밍(PC, 온라인, 모바일), 게임음향, 특수효과 및 마케팅 분야로 나눌 수 있다. 우송대학교 게임멀티미디어학과의 교과과정의 분포는 프로그래머 개발을 중점으로 짜여져 있으며, 특히 산업체의 연계를 위한 교과과정이 많이 포함되어 있는 현실이다.

라. 비정규 교육기관 : 게임아카데미

1) 게임아카데미의 개설 목적 및 목표

21세기 지식정보시대의 고부가가치 전략산업인 문화콘텐츠(CT) 산업의 핵심인 게임산업의 '소수정예 전문인력 양성'을 목표로 게임아카데미가 설립되었다.

게임아카데미는 차세대 국가 성장 동력인 문화콘텐츠(CT)분야의 핵심인 게임산업의 게임개발 인력양성을 통한 문화산업의 국가경쟁력을 확보하는 것을 목적으로 한다.



이 외에도 다양한 게임교육 인프라 구축으로 지식문화강국 실현의 기반을 구축하는 것도 목표의 하나이다.

2) 게임아카데미의 사업 연혁

- 게임산업의 체계적인 육성 및 지원을 위해 (재)게임종합지원센터 설립 (1999. 02) → (재)한국게임산업개발원 명칭 변경 (2002. 01)
- 게임인력양성을 위한 전문위원회(전문인 22명) 구성 (1999. 09)
- 게임아카데미 개원식 및 개강(한국방송회관) (2000. 11)
- 정규교육과정 제 2기 선발 및 개강 (2001. 07)
- 게임아카데미 이전(목동 한국방송회관 → 논현동) (2001. 10)
- 게임원격교육 호서대 등 5개 대학과 협정서 체결 (2001. 12)
- 게임원격교육 사이트 Open (2002. 03)
- 정규교육과정 제 3기 선발 및 개강 (2002. 04)
- 특수교육과정 개설(개발자/ 청소년 과정) (2002. 05)
- 게임원격교육 1차 16과목 콘텐츠 개발 완료 (2002. 05)
- 게임원격교육 1기 수강생 356명 교육 운영 (2002. 07)
- 게임원격교육 영남대 등 18개 기관과 협정서 체결 (2002. 08)
- '02년도 게임교재 개발위원회(산·학·연·관 25명) 구성 (2002. 10)
- 게임원격교육 2차 16과목 콘텐츠 개발 완료 (2002. 11)
- 게임원격교육용 서버 및 보안 시스템 구축 완료 (2002. 11)
- 정규교육과정 제 1기 졸업식 및 작품시연회 개최 (2002. 12)
- 교육용 영상 제작실 구축 (2002. 12)
- 게임원격교육 2기 수강생 331명 교육 운영 (2002. 12)
- 정규교육과정 제4기 선발 및 개강 (2003. 02)
- 게임원격교육 한국문화콘텐츠진흥원과 통합 LMS 구축 (2003. 02)
- 게임원격교육 한국외대 등 37개 대학과 협정서 체결 (2003. 02)
- 게임원격교육 2차 22과목 콘텐츠 개발 완료 (2003. 03)

- 게임교육 콘텐츠(교재) 1차년도 개발 완료 (2003. 06)
- 정규교육과정 제 2기 졸업식 및 작품시연회 개최 (2003. 07)
- 게임교육 콘텐츠(교재) 1차년도 보급 (2003. 07)
- 게임원격교육 3기 수강생 6,385명 교육 운영 (2003. 07)
- 사이버아카데미 사업 공동 추진 협정 체결(KOCCA) (2003. 08)
- 게임원격교육 홍익대 등 8개 대학과 추가 협정서 체결 (2003. 08)
- 게임원격교육 4차 16과목 콘텐츠 개발 완료 (2003. 10)
- 게임교육 콘텐츠(교재) 2차년도 개발 착수 (2003. 12)
- 게임원격교육 4기 수강생 3,512명 교육 운영 (2003. 12)
- 게임원격교육 KOCCA와 인터넷 전용선 사용 계약 체결 (2004. 01)
- 정규교육과정 제 5기 선발 (2004. 02)
- 게임원격교육 공주대 등 39개 대학과 협정서 체결 (2004. 02)
- 게임원격교육 5기 수강생 4,393명 모집 (2004. 03)
- 게임 시나리오과정 신규 개설 (2004. 05)
- 게임 마케팅과정 및 공무원 게임교육과정 신규 개설 (2004. 07)
- 비디오게임 교육관련 SCEK와 업무협약 체결 (2004. 08)
- 게임원격교육 경기대 등 10개 대학과 추가 협정 체결 (2004. 08)
- 게임원격교육 6기 수강생 2,396명 모집 (2004. 09)
- 모바일게임관련 (주)다음커뮤니케이션과 업무협약 체결 (2004. 09)
- 비디오게임 교육관련 제페토와 업무협약 체결 (2004. 09)
- 게임분야 청소년교육과정 지원사업 업무협약 체결 (2004. 10)
- 게임분야 인력시장 활성화 지원사업 추진 (2004. 10)
- 미국 카네기멜론대학과 게임인력양성 업무협약 체결 (2004. 10)
- 게임 교육콘텐츠(교재) 2차년도 보급사업 추진 (2004. 11)
- 게임연구센터(GRC) 지정·기반지원사업 1차년도 추진 (2004. 11)
- 게임분야 해외연수 지원사업 1차년도 추진 (2004. 11)
- 게임원격교육 7기 수강생 1,927명 모집 (2004. 11)
- 단계별통합추진 관련 KOCCA와 연구용역 ISP 사업추진 (2004. 12)
- 미국 카네기멜론대학 ETC 게임분야 해외연수(24명) 실시 (2005. 01)
- 정규교육과정 제 4기 졸업식 및 작품시연회 개최 (2005. 02)
- 정규교육과정 제 6기 선발 (2005. 02)
- 2005 Game Developer Conference Expo 총 9명 참가 (2005. 03)
- 비디오게임 교육과정 개설 (2005. 04)
- 게임 마케팅과정 개설 (2005. 05)
- 2005 E3(Electronic Entertainment Expo) 총 4명 참가 (2005. 05)
- 공무원 교육과정 개설 (2005. 05)
- 게임원격교육 8기 수강생 1,622명 모집 (2005. 05)
- 게임관련 대학연계 교류사업 추진 (2005. 06)
- 산업체 실업계 고교교사 연수과정 개설 (2005. 07)
- 게임원격교육 9기 수강생 1,767명 모집 (2005. 07)
- 비디오게임 교육과정 개설 (2005. 09)
- 2005 동경게임쇼 참가 (2005. 09)
- 모바일 3D게임 교육과정 개설 (2005. 10)
- 중국통상전문가 교육과정 개설 (2005. 10)
- 게임원격교육 10기 수강생 1,870명 모집 (2005. 10)

3) 게임아카데미의 교육 체계

○ Off-Line 게임교육

- 정규교육과정 : 2년 8학기제 운영

(1학년 기본과정, 2학년 프로젝트과정)

- 특수교육과정 : 게임전문가과정 및 공무원교육과정 운영

(교육기간 ▪ 내용 과정별 상이)

○ On-Line 게임교육

- 원격교육과정 : 연 3기수(1기수 10주) 운영

4) 게임아카데미의 교육 과정 및 커리큘럼

표 III-13. 게임디자인(GD)

구분	1Q	2Q	3Q	4Q	5Q	6Q	7Q	8Q
1	게임학총론	시나리오창작 I	시나리오창작 I	게임인터페이스	게임프로젝트	게임프로젝트	게임프로젝트	게임프로젝트
2	게임저작물	게임디자인 I	게임디자인 II	게임디자인 III				
3	아이디어발상	게이머행동론	영상연출 II	게임산업의 이해				
4	게임소재론 I	게임소재론 II	게임소재론 III	게임소재론 IV (군사학)				
5	게임그래픽의 이해	커뮤니케이션	PD론	게임아키텍처				
6	게임마케팅	영상연출 I	보드게임 디자인론	게임그래픽의 이해				
7	게임사운드	2D프로그래밍의 이해	소프트웨어공학	3D프로그래밍의 이해				
8	외국어	외국어	외국어	외국어				
9	외국어	외국어	외국어	외국어				
10	외국어	외국어	외국어	외국어				

표 III-14. 게임그래픽(CG)

구분	1Q	2Q	3Q-1	3Q-2	4Q	5Q	6Q	7Q	8Q
1	소묘	인체소묘 (누드크로키)	중간프로젝트	인체소묘	게임기획론	게임프로젝트	게임프로젝트	게임프로젝트	게임프로젝트
2	소묘	조소 II		조소 III	모션연구				
3	조소 I	3D 게임그래픽기초		캐릭터디자인응용	포트폴리오제작론				
4	조소 I	3D 게임그래픽기초		의상 및 소도구 디자인	메카닉디자인				
5	게임학개론	3D 게임그래픽기초		3D 게임모델링 I	3D 게임모델링 II				
6	디지털이미지 프로세싱	게임영상편집		3D 게임모델링 I	3D 게임모델링 II				
7	벡터그래픽	조형 디자인		3D 게임모델링 I	3D 게임맵핑 II				
8	2D 게임그래픽	색채 디자인		3D 게임맵핑 I	3D 게임맵핑 II				
9	게임영상기획	캐릭터디자인		3D 게임맵핑 I	3D 애니메이션 I				
10	인터페이스디자인	사운드 디자인		게임영상연출	3D 애니메이션 I				

표 III-15. 게임프로그래밍(GP)

구분	1Q	2Q	3Q-1	3Q-2	4Q	5Q	6Q	7Q	8Q
1	Programming I (C&C++)	Programming II (C&C++)	중 간 프 로 젝 트	3D Programming	3D Engine	게 임 프 로 젝 트	게 임 프 로 젝 트	게 임 프 로 젝 트	게 임 프 로 젝 트
2	Programming I (C&C++)	Programming II (C&C++)		3D Programming	3D Engine				
3	Programming I (C&C++)	Programming II (C&C++)		3D Programming	3D Advanced Feature				
4	Windows Programming & MFC	Network Programming		3D Programming	Network Engine				
5	Windows Programming & MFC	Network Programming		3D Programming	Network Engine				
6	프로그래밍 실습	프로그래밍 실습		3D Programming 실습	3D Engine 실습				
7	프로그래밍 실습	네트워크 실습		3D Programming 실습	Network 실습				
8	System Programming (OS Program)	게임수학		3D Programming 실습	게임프로젝트				
9	게임학개론	3D 그래픽의 이해		게임개발 프로세스	게임특강				
10	Data Structure (자료구조)	Data Structure (알고리즘)		Computer System Architecture	게임사운드				

### 3.3 미국 게임 인력 기관의 사례 분석

#### 3.3.1 미국 게임 교육기관

표 III-16. 미국 게임 교육기관

학교명(URL)	위 치	개 요
Carnegie Mellon University ETC	Pittsburg www.cmu.edu	석사과정에 Entertainment Technology Center를 운영하면서 컴퓨터 공학분야와 카네기 멜론 대학의 강점인 연극이론 분야의 커리큘럼을 복합하여 구성하고 있다. EA 및 디즈니사와 깊은 산학 협동 프로그램을 추진하고 있고, 기본 코스 이외에는 프로젝트 중심으로 교육을 구성하고 있고 근래 가장 성공적인 운영 사례로도 꼽힌다.
DigiPen Institute of Technology	Redmond, WA www.digipen.edu	게임개발에 대한 최초의 학사학위 인정 게임스쿨로 3D Computer Animation과 Real Time Interactive Simulation등의 학과가 있으며, Nintendo of America의 지원을 받고 있다. 온라인 교육과정이 개설되어 있어서 사이버 교육도 활발한 편이다.
University of North Texas	Denton, TX apache.csci.unt.edu	학사과정과 석사과정에서 Computer Game Design and Programming, Directed Study 등의 코스가 개설되어 있다.
DevelopMentor	Los Angeles, CA/ Boston, Portland, UK www.develop.com	3D 그래픽 프로그래밍, OpenGL, Direct3D, C++ 등의 프로그래밍 교육과정이 개설되어 있다.
Full Sail Real World Education	Winter Park, FL www.fullsail.com	게임 디자인, 컴퓨터 애니메이션 등에 학사학위를 부여하며, 다양한 게임관련 커리큘럼을 갖고 있다.
Georgia Institute of Technology	Atlanta, GA www.cc.gatech.edu	GVU 센터에서 교육을 제공하며 애니메이션, VR, 게임 프로그래밍 및 개발에 관한 과정이 있다.
University of Southern California	Los Angeles, CA www.usc.edu/dept/cs interactive.usc.edu	게임개발과 직접 관련된 멀티미디어의 과정이 있으며, Computer Science 과정이 개설되어 있다.
University of Michigan	Ann Arbor, MI www.eecs.umich.edu/cse	컴퓨터 게임 디자인과 컴퓨터 게임개발 교육과정이 설치되어 있다.
School of Communication Arts	Raleigh NC www.ncsca.com	SoftImage, 3DMax, and Alias Power Animator 등의 교육과 그래픽 디자인 교육과정이 설치되어 있다.
MIT Artificial Intelligence Laboratory	Cambridge, Mass www.ai.mit.edu	MIT의 AI Lab
Pratt Institute	Brooklyn, NY www.pratt.edu/ad/cgim/index.html	Computer Graphics와 Interactive Media 교육과정이 있다.
Rensselaer Polytechnic Institute	Troy, NY www.rpi.edu/dept/hss/hssf/mesetundergrad.htm	RPI는 M&M(Minds & Machines), EMAC(Electronic Media, Arts, & Communication)를 중심으로 게임/로봇, AI, 디지털 뮤직 등의 교육과정이 설치되어 있다.
University of Advancing Computer Technology	Tempe, AZ www.rpi.edu/dept/hss/hssf/mesetundergrad.htm	UACT는 우수한 컴퓨터 기술 교육연구소로 학사학위를 수여하며, 멀티미디어 관련 교육을 제공한다.

### 3.3.2 미국 교육기관의 교과과정 현황

o 2년제 대학의 교육시스템은 대학교에서 Associate Program 등과 함께 취급되어지므로 “대학교” 항목에서 다루도록 한다.

o 일반 게임학원들과 유사한 Training Center 등의 시설이 활발하게 진행되는 것에 반하여, 게임을 전문적으로 연구하는 정규 교육과정인 대학교는 아직도 초기 단계라고 할 수 있다.

o 몇몇 대학교의 Multimedia과 등에서 Game Graphic과 Programming 교과 과정을 일부 강의 하고 있다.

o 본격적으로 게임만을 교육하는 대학교인 DigiPen Institute of Technology는 수학, 물리학, 컴퓨터, 게임프로젝트 등을 가르치는 Real Time Interactive Simulation과 미술, 애니메이션, 필름, 컴퓨터그래픽, 작문, 프로젝트 등을 가르치는 3D Computer Animation학과로 구성되어 있다.

표 III-17. 교과과정 현황

설치대학/학과명/설치년도	교육영역	과목 수	특 징	URL
DigiPen Institute of Technology/ Real Time, 3D Computer Animation/1994	GD GG GP	2 21 31	본격 Game 대학교로서 Program 위주의 Real Time Interactive Simulation과, Graphic위주의 3D Computer Animation 과가 있음	digipen.edu
University of Advancing Computer Technology/ Bachelor of Arts in Multimedia Degree	GG GP	10 5	일반적인 Multimedia과의 교과 과정 중 다수의 게임 관련 과정 들이 있음	uact.edu

### 3.3.3 대학교별 교과과정 사례

게임 전문 대학원이 아닌 기존의 컴퓨터 학과를 중심으로, 대학원 내에서 소수의 게임 관련 교과과정 등이 설치되어 있으며, 교과내용으로는 Graphics and Internet Design, Video Production, Multimedia Design and

Production, User Interface Design & Evaluation, Modeling and Design, Project Introduction 등이 있다.

표 III-18. 교과과정 현황

설치대학/학과명/설치년도	특 징	URL
Carnegie Mellon University ETC/1999	- 디즈니가 개발한 오픈 엔진인 Pand3D 엔진을 활용한 프로젝트 중심 교육 - 단기간 다작 프로젝트 추진 방식을 통한 d 창작성 향상에 중심	www.cmu.edu
Georgia Institute of Technology/Master of Science In human Computer Interaction/1997	- Human Computer Interaction 전공의 대학원에 Game 관련 교과 과정이 소수 포함	cc.gatech.edu
University of Southern California/Master of Science is Computer Science in Multimedia and Creative Technologies School of Cinema-Television	- 대학 및 대학원에 다수의 관련 교과과정이 있음	usc.edu

### 1) DigiPen대학의 개요

- 미국내 최초의 게임관련 학위 프로그램을 실시
- 본부 소재지 : Redmond, WA, USA
- Motto: Produce People as well as Software for making Games.
- Areas of Expertise: 전공 세분화 영역
  - Game Design
  - Game Art and SFX
  - Game Sound and SFX
  - Game Software and Programming

### 2) DigiPen 대학 게임 교육의 특성

- 이론과목과 함께 강도 높은 실습과정과 함께 연동되는 프로젝트 교육과정
- 매년 4-6명 단위로 팀을 구성하고 수업과정의 내용을 이해한 후 창작품을 제출

표 III-19. 게임디자인 전공 학점 구성

학수번호	과목명	이수 학점
GAM 100	Introduction to Game Design & Production	3 학점
GAM 150	Project I	3 학점
GAM 200 & 250	Project II	8 학점
GAM 300 & 350	Project III	10 학점
GAM 400 & 450	Project IV	10 학점
GAM 390	Internship I	1-5 학점
GAM 490	Internship II	1-5 학점

- 게임 디자인 및 제작 과목 이외에는 철저하게 프로젝트 및 인턴 과정으로 수업을 진행하고 있는 것이 큰 특징이다.
- Real Time Interactive Simulation Program  
- 4년제 학부 Courses

표 III-20. 교과과정(Digipen Institute of Technology ①)

학기	교과내용
1	Algebra and Trigonometry
	Linear Algebra and Geometry
	Computer Environment
	Operating System I - Man-Machine Interface
	High Level Programming I
	Project Introduction
2	Calculus and Planar Analytic Geometry
	Operating System II - OS Interface to Graphics and Sound
	Hardware
	High Level Programming II
	Game Implementation Techniques II
3	Project I
	Calculus and Solid Analytic Geometry
	Computer Graphics I
	High Level Programming II
	Game Implementation Techniques II
4	Project II - Part 1
	Computer Graphics II
	Computer Networks I - Interprocess Communication
	Computer Networks II - TCP/IP Communication Protocol
	High Level Programming IV
5	Data Structures
	Project II - Part 2
	Advanced Computer Graphics I
	Low Level Programming I
	Graphics File Formats and Data Compression Techniques
6	Algorithm Analysis
	3D Computer Animation Production
	Project III - Part 1
6	Surface Modeling
	Advanced Computer Graphics II

	Low Level Programming II
	Image Processing
	3D Computer Animation Production II
	Project III - Part 2
7	Advanced Surface Modeling
	Motion Dynamics
	Ray Tracing I
	Multimedia Aspects of Game Making I
	Project IV - Part 1
8	Solid Modeling
	Waves, Optics, and Aero-Dynamics
	Ray Tracing II
	Multimedia Aspects of Game Making II
	Project IV - Part 2

- 3D Computer Animation Program
- Associate Degree Courses (2년제 학위과정)

표 III-21. 교과과정 (Design Institute of Technology ②)

학 기	교 과 내 용
1	Algebra and Trigonometry
	Linear Algebra and Geometry
	Computer Environment
	Operating System I - Man-Machine Interface
	High Level Programming I
	Project Introduction
2	Computer Graphics II
	Computer Networks I - Interprocess Communication
	Computer Networks II - TCP/IP Communication Protocol
	High Level Programming V
	Data Structures
	Project II - Part 2
3	Advanced Computer Graphics I
	Low Level Programming I
	Image Processing
	Algorithm Analysis
	3D Computer Animation Production II
	Project III - Part 1
4	Surface Modeling
	Advanced Computer Graphics II
	Low Level Programming II
	Image Processing
	3D Computer Animation Production II
	Project III - Part 2

### 3) Carnegie Mellon University ETC 과정

- ETC 설립 배경
  - CMU에서 Entertainment 분야의 중요성을 깨닫고 새로운 교육 필드를 조성하고자 설립하였다.
  - Computer Science 전공 교수 Randy Paush와 Drama 전공 교수 Donald Marinelli 의 만남을 통해서 ETC가 만들어졌다.
  - Fun 중심
  - 예술과 공학을 결합시킨 연구
  - ETC의 철학은 예술과 기술의 두 분야 사람들을 동등한 위치에서 협동하여 함께 작품을 구상하고 개발하도록 하면 컴퓨터 게임과 같이 여러 분야가 모인 종합적인 분야에 있어서 좋은 성과를 낸다는 관찰에 기반한다.

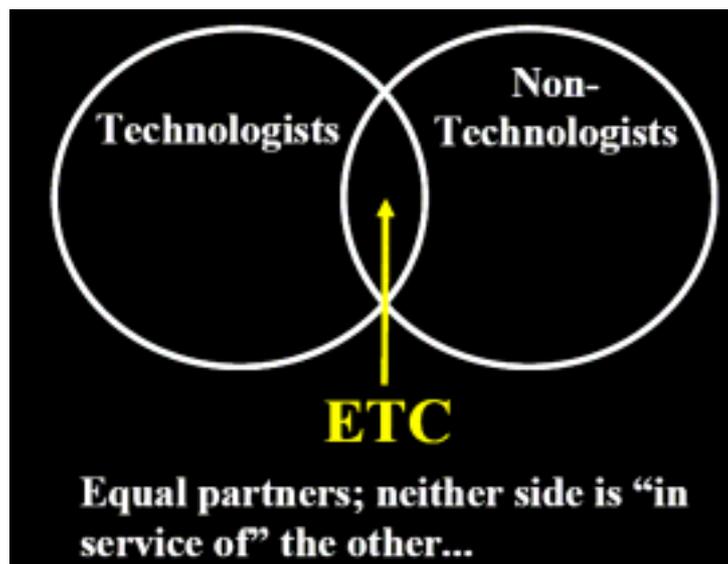


그림 III-5. ETC교육철학의개요

- Carnegie Mellon 대학에 있는 2년제 석사 교육 과정
- 예술(Arts)과 기술(Technology)이 혼합된 교육 과정(Entertainment Technology) 으로 미국내 게임 관련 교육기관 중 성공한 교육과정으로 알려졌다.
- 초창기 1999년에 8명의 학생과 시범적으로 출발하였으며 2001년

도 봄학기에 첫 졸업을 하였다.

○ ETC 구성 인원

- 14명의 faculty : 교수 요원들로 드라마, 사운드, 영화, 게임 시나리오 등 각 분야의 교수진이 분포되어 있다.
- 4명의 staff : 기술적인 부분을 담당하는 사람들로 Maya, Program, Sound 등의 Mentor 지원하고 있다.
- TA(Train assistance) 운영 : 프로젝트를 진행하면서 프로그램과 그래픽 쪽의 의문 사항을 해결해 준다.

ETC는 전기한 바대로 2명의 공동 책임자가 있으며 그 밖에는 주로 lecturer 트랙의 인원들로 구성된다고 한다.

구성은 3명의 컴퓨터 과학 강사, 3명의 드라마분야 강사(즉흥연기, 스토리구조, 게임디자인), 2명의 필름메이킹 및 포스트프로덕션 강사, 1명의 디지털 뮤직 강사, 1명의 컴퓨터그래픽 강사로 이루어져 있고 이밖에 인간-컴퓨터 인터랙션, 엔터테인먼트공학, 법률사항 등을 담당하는 겸임교수들이 있다.

이상은 지난 7월 Kris의 사전 오리엔테이션 자료의 내용을 기준으로 한 것이지만, 2차에 걸친 본인의 경험으로는 이밖에 실습을 도와주는 테크니션과 수명의 조교들이 수업을 돕고 있었으며, 졸업생들을 잘 활용하여 겸임교수나 전임교수로 활용하였다. 1차 때의 조교 대표가 이번 2차 때에는 졸업생으로서 테크니션으로 근무하고 있었다.

○ ETC 교육 과정의 특징 (매학기 프로젝트 수업을 진행)

- 강의식 수업 대폭 축소
- 전체 학생들이 청강할 수 있는 그래픽, 기술, 드라마 등 교양 과목 수준의 강의 채택
- 원하는 수업을 들을 수 있는 Optional 수업 형태 도입
- 각 분야별 심화 과정을 소그룹 형태의 토론식 수업으로 진행

ETC는 2개 학년의 석사과정을 운영한다. 표준 커리큘럼은 다음과 같다.



그림 III-6. ETC 표준 커리큘럼

ETC의 교육과정은 강의식 수업이 오히려 많지 않다는 특징이 있다. 첫 학기는 앞에서 말한 대로 5회의 BVW 프로젝트가 주를 이룬다. 이 프로그램이 2년간 교육과정의 분위기를 좌우한다. BVW를 통하여 성취감 및 자신감을 획득하게 되며, 이 과정을 통한 하드 트레이닝으로 이후 스스로 심화 공부를 할 수 있는 자신감을 얻게 된다. 그리고 2학기부터는 1학기 이상 단위의 장기 프로젝트를 수행한다. 프로젝트의 결과물은 프로토타입이나 예술적 창작품이어야 한다.

Introduction to Entertainment technology 코스에서는 기술을 기반으로 하는 콘텐츠에 대하여 고찰한다. Improvisational acting은 Brenda Harger라는 여배우출신 패컬티가 진행하는데, 즉흥연기를 통하여 상대방의 의견을 판정하지 않고 수용하는 자세를 키우며 소극적으로 물러서지 않고 적극적으로 나서게 하는 훈련을 쌓는다.

The Visual Story는 영화분야 출신인 Ralph Vituccio가 진행하는데, 영화의 스토리 구조와, 스토리를 비주얼로 진행하는 방법에 대하여 터득하고 토론한다. 특히 비주얼 수단에 의하여 시청자의 느낌을 변화시키는 것에 대하여 신선한 많은 사실들을 배울 수 있었다.

Electives는 정해진 수업이 아니라 학생들이마다 다양한 욕구 충족을 위하여 자기 자신이 골라 수강하는 형태의 수업이다. Game Design, Sound Design 등이 여기에 해당한다.

1학년 2개 학기가 끝나고 나면 많은 학생들이 EA나 디즈니와 같은 현장에서 인턴쉽을 밟는다. 본인은 지난 겨울 1차 연수때 당시 인턴쉽을 끝내고 온 학생들의 역동적이고 자신감 있는 경험담 발표에 깊이 매료된 기억이 있다. 당시에는 EA의 공동 창립자인 고든까지 와서 사회에 첫발을 디딜 준비에 대한 코멘트를 하는 것을 경청한 바 있다.

이 인턴쉽이 지속되면 2학년 1학기나 2학기 과정을 대체할 수도 있다고 한다.

- 독자적 커리큘럼
  - Improvisational Acting (즉흥연기)
  - 독창적 수업으로서 원래 카네기멜론 대학이 가지고 있는 장점인 드라마와 연극 분야의 과목을 접합시켜서 만들어 낸 과목이다.
- The Visual Story
  - 영화의 스토리 구조
  - 영화와 비주얼적인 관계 구조
  - 새로운 미디어와의 접목 관계에 대해서 토론
- 다양한 선택 과목 (Electives)
  - 다양한 학생들의 심화된 전공 교육의 욕구와 고착화된 수업 방식을 탈피하여 다양한 수업을 추구하기 위해서 커리큘럼을 자기 자신이 디자인하여 만족감을 극대화(즉 타 대학 과목 등을 신청하여 이수 할 수 있다)
  - ETC가 제공하는 선택과목
    - Game Design
    - Introduction to Sound Design : Design & Media
    - Maya
    - Pre-approved Carnegie Mellon Electives
    - Computer Science
    - Robotics
    - Electrical Engineering
    - Film-Mechanical Engineering
    - Design

- Drama
- Music
- Art- English
- Philosophy
- Psychology
- Public Policy & Management
- 업체 Internship 의 활성화
  - 주로 EA 및 디즈니 인터랙티브사 등 관련회사에서 실습
  - 2학년 1학기, 2학기 과정
  - 회사와 공동 작업에 참여 할 경우 수업을 프로젝트로 대체함
- ETC는 EA와 밀접한 관계를 맺고 있음
  - EA는 매년 40% 정도의 ETC 졸업생을 받음
  - 매년 여름에 적어도 10명 정도의 인턴을 받고 있음
  - 2003년 가을에는 EA의 연구진을 ETC에 상주하게 하여 ETC 교육과정을 연계함
- 인턴쉽을 통한 산학 연계
  - 산학 연계를 통하여 현장 중심의 교육과정을 연구
  - 학생들의 진로 및 방향 제시 가능
  - ETC 프로젝트 코스는 팀의 완성도를 높이는 교과로서 실제적으로 기업에서 필요로하는 작업을 같이 할 수 있는 사람을 키워냄 (Communication)
- BVW (Building Virtual World) 프로젝트 수업 방식
  - Panda3D 엔진을 활용하여 다양한 가상환경의 콘텐츠를 제작하고 있음.
  - 한학기에 5번의 프로젝트로 2주마다 팀을 구성
  - 팀 구성은 일반적으로 4명
    - Intangible : 사운드 편집, 스토리보드, 팀 관리 등
    - Programmer : python을 이용한 프로그램 구현
    - Modeler : Maya를 이용한 3D 모델링
    - Texturing : 원화 및 2D 관련 그래픽
  - 매 프로젝트마다 팀 구성원을 바꿈
  - 평가의 2/3은 각 프로젝트의 결과물로 점수 반영
  - 평가의 1/3은 학생들이 다른 팀원을 평가하여 점수에 반영

- 첫 학기에 행해지는 만큼 2년간의 교육과정 분위기를 좌우
  - 성취감 및 자신감 획득
  - 실제적으로 기업에서 필요로 하는 작업을 같이 할 수 있는 사람 육성
  - Hard Train 이후 자신의 분야를 스스로 공부할 수 있는 자질을 키움

#### 4) BVW (Building Virtual World)

이상과 같은 철학을 바탕으로 백그라운드가 다른 학생들을 팀으로 묶고 시간을 적게 줌으로써 팀원들 간에 고집을 부릴 수 있는 시간을 줄이고 서로 협동하는 법을 가르치는 단기 프로젝트 프로그램을 개발하였는데 이를 BVW라고 한다.

팀 하나의 규모는 4인을 기준으로 하고 있으며, 패컬티의 지도 아래 2주 내에 자유로운 주제를 가지고 작품을 만든다. 그 중의 일부는 자금 지원을 받는 산학 협동의 형태로 진행되기도 한다.

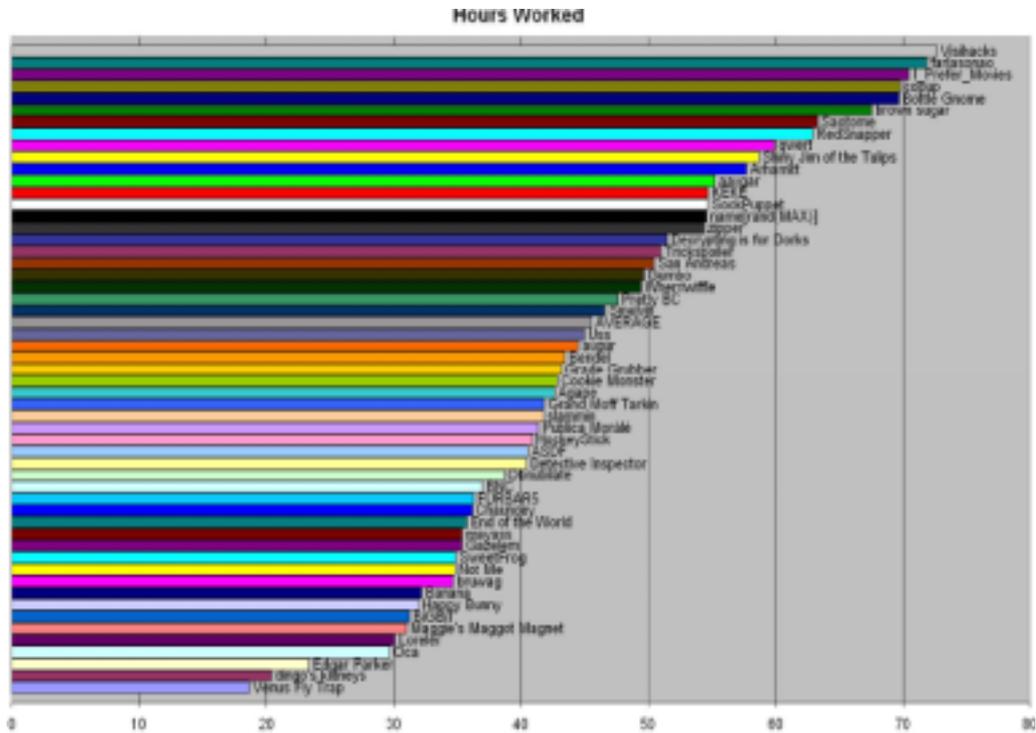
보통은 프로그래머 1명, 모델러(3D 그래픽 디자이너) 1명, 2D 1명, 그리고 사운드 등 기타를 담당하는 intangible 1명으로 구성된다. 동일한 업무를 맡는 사람이 2명 이상이면 서로간의 다툼이 있을 수 있기 때문에 각기 다른 역할을 맡기를 권고하며, 더 오랜 기간을 주면 낭비하는 시간이 있고 팀원간 다툼 시간이 있기 때문에 2주만 주는 것이라고 한다.

팀을 구성할 때에는 서로 친분이 있는 학생들끼리 모이는 것을 방지하기 위하여 패컬티들이 임의로 섞어 구성한다. ETC에서는 첫 학기중에 이러한 BVW를 5회(10주간) 되풀이하므로 다른 사람들과 팀 프로젝트하는데 익숙해지고 창의적인 작품 구성에 자신감을 갖게 된다.

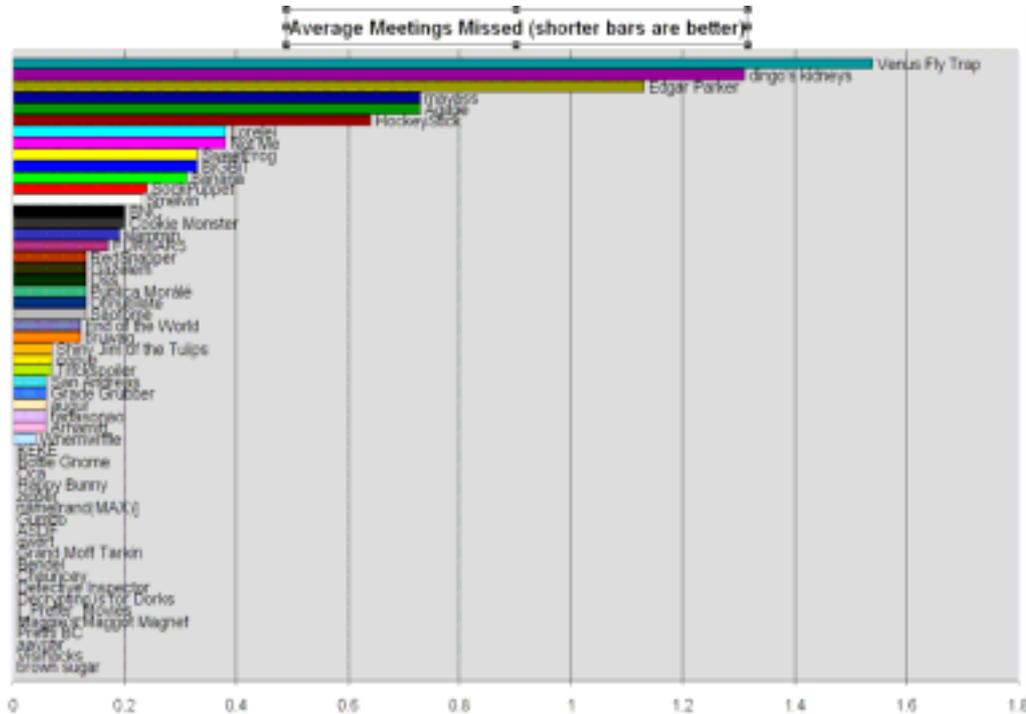
5회 프로젝트를 하는 동안 가급적 같은 팀에 거듭 소속되는 것을 방지하도록 섞어놓기 때문에, 각자 20명의 다른 사람들과 만나 일해 볼 기회를 얻게 된다.

평가는, 결과물에 대한 평가점수를 2/3로 하고 여기에 다른 팀원이 각자에게 내린 평가점수를 1/3 반영시킨다. 이렇게 한 학기 5개 프로젝트를 끝내고 나면 어떤 사람이 일하기 편한 사람이고 또 그렇지 않은지가 자연스

렵게 드러난다. 예컨대 각자가 일한 시간과, 모임에 빠진 횟수를 조사하면  
 다음처럼 그래프를 얻을 수 있다.



(a) BVW 팀원 관리 차트(Hours Worked)



(b) BVW 팀원 관리 차트(Average Meeting Missed)



또, 아이디어 기여도와 같이 일하기 좋은 정도를 조사하면 다음 그래프가 산출된다. 이렇게 함으로써 팀 작업이라는 공동작업 중에도 각자의 노력을 요구하게 되고 팀원과의 작업을 잘 하려는 훈련이 쌓인다.

2주 내에 수행하여야 하기 때문에 구현에 걸리는 시간을 최소화하여야 한다. 그러므로 개발 도구가 중요하게 되는데, ETC에서는 배우기 쉬운 스크립팅 언어인 Python으로 공개 게임 엔진인 Panda3D를 호출하도록 하며, 그래픽 도구는 Maya를 기본으로 하고, 우리 한국 교수들이 여름에 갔을 때에는 3D MAX도 준비해 놓고 있었다. 이 도구 특히 프로그래밍 관련 도구들은 비교적 매우 단순하기 때문에 구현보다는 아이디어 자체에 집중할 수 있다는 장점이 있다. 다만 짧은 기간과 도구의 한계 때문에 국내에서와 같은 복잡한 게임을 구성하기는 쉽지 않아 보였다. 그러므로 결국 ETC는 기술을 가르친다기 보다는 예술과 기술이 합한 엔터테인먼트 콘텐츠 분야의 작품을 창의하고 협동하여 완성하는 경험과 즐거움을 훈련시키는 것이 주 목표인 것으로 판단하였다.

지난 겨울 1차 연수시에는 우리 한국 연수단도 이러한 BVW를 2회에 걸쳐 실습하였는데, 그때의 의견으로 반영하여 이번 여름의 2차 연수에서는 3주간의 1회 BVW로 개편하였다.

- 프로젝트 학습 평가 기준
  - 학생들의 주당 BVW 활동 시간
  - BVW팀 미팅 불참 척도
  - 창의력 기여도
  - 팀작업의 친밀도
  
- ETC의 학제적 연계전공을 통한 프로젝트 코스
  - 3~7명으로 팀 구성
  - 프로젝트의 주목적은 순수 프로젝트를 진행하는 것에 있고, 부목적으로는 다른 이들 혹은 외부인과의 원활한 프로젝트 진행에 있음
  - 프로젝트 진행을 위해 기술자와 비기술자의 적절한 안배가 요구

- 결과물은 교수들이 놀라워할 정도의 창작물이어야 함
- 회사와 Co-work을 하는 경우 3,4학기 프로젝트 수업을 대치

### 5) 진출

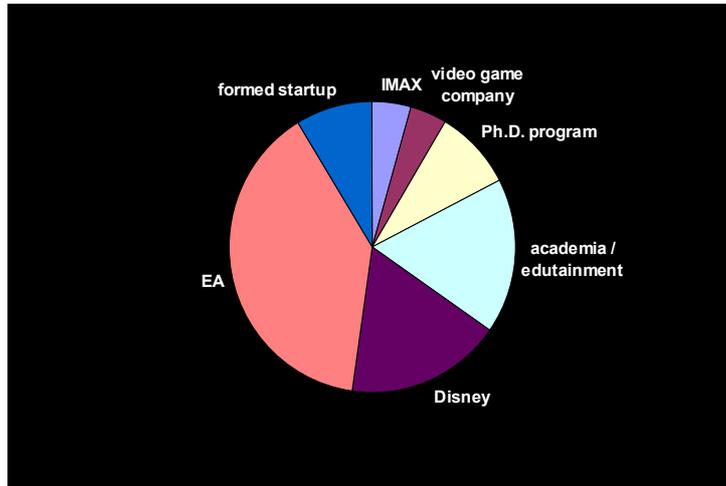


그림 III-8. CMU ETC 졸업생 진출 현황

이와 같은 프로젝트 중심의 커리큘럼에 따라 실제로 업계에서 필요로 하는, 함께 일할 수 있는 대화 가능한 인재를 수년간 키워냈기 때문에 EA, Disney 등 이 분야 우수 대기업들이 ETC졸업생을 리크루트하려고 적극적이며, 또 이를 수월하게 하기 위하여 재학생의 인턴쉽도 적극 유치하고 있다고 한다. Randy의 말로는, 앞으로 5년간 매년 10명의 졸업생을 보내 달라고 하는 EA의 편지를 자기 책상서랍에 넣어놓고 있다고 하며 이것은 다른 기업도 마찬가지라고 자랑하고 있었다. 다음은 작년 졸업생의 진출 현황이다.

이렇게 졸업생의 진출이 수월하다 보니 장학금이 일체 없는 고비용의 교육임에도 우수한 인재들이 몰려들고 있고 ETC는 이러한 지망자들 중에서 베스트들을, 분야별로 골고루 뽑아올 수 있어 더 수준을 높일 수 있는 선순환 구조를 확립한 것으로 보였다. 이것은 입학생들에게 기초 교육을 시킬 필요가 없고 처음부터 전문가 수준으로서 바로 프로젝트에 투입 가능하다는 데에 대해서 매우 부러웠다. 다음 도표는 처음 몇 명에 불과했던 학생수가 이 선순환 구조의 확립에 따라 갈수록 늘고 있음을 보여준다.

6) University of Southern California

- Master of Science in Computer Science in Multimedia and Creative Technologies

표 III-22. University of Southern California 교과과정

교과내용
Computer Graphics
Introduction to Digital Image Processing
Computer Vision
3D Graphics and Rendering
Advanced Topics in Computer Vision
Artificial Intelligence
Geometric Modeling
Computational Geometry
Specification and Design of User Interface Software
Microsystems Technology for Multimedia

- MFA(Master of Fine Arts in Interactive Media (Graduate))
  - MFA는 컴퓨터 기반의 엔터테인먼트 과정들을 제공한다.
  - 이 과정의 기본 개념은 실제 실행하는데 있어서의 창조적인 표현과 경험 그리고 우수성을 강조하고 있다.
  - 특히 인터랙티브 엔터테인먼트 트랙은 게임 디자인과 혁신을 강조한다.
  - 이과정은 전통적인 영화나 TV에서의 기술과 경험에 바탕을 둔 상호적 효과의 이해를 요구하고 있다.
  - 학생들은 최소 평균 3.0(A=4.0)을 취득해야하고 모든 과정에서 C(2.0)를 취득해야 해당 과정을 통과한 것으로 인정한다.
  - 졸업 기준은 56학점 취득(42필수, 16 선택)해야 한다.

표 III-23. 교과과정 (Master of Fine Arts)

학기	과정명	학점	비고
1학년, 1학기	Survey of Interactive Media	2	
	Interactive Media Seminar	1	
	Design for Interactive Media	4	
	Production I	4	
1학년, 2학기	Interactive Media Seminar	1	
	Visual Expression	2	
	Introduction to Interactive Writing	2	
	Electives	4	
2학년, 1학기	Interactive Experience Design	4	
	Experiments in Interactivity I	2	
	Electives	4	
2학년, 2학기	Interactive Design and Production	4	
	Experiments in Interactivity II	2	
	Preparing the Interactive Project	2	
3학년, 1학기	Advanced Interactive Project	4	
	Business of Interactive Media	2	
	Electives	4	
3학년, 2학기	Advanced Interactive Project	4	
	Internship in Cinema/Television	2	
	Electives	2	
선택 과목(최소 6학점)	Producing Practicum	2	
	History of Animation	2	
	Introduction to Computer Animation	2	
	Visual Effects	2	
	Interactive Animation	2	
	Virtual Reality and Stereoscopic Animation	2	
	Seminar in Film/Television and New Technologies	4	
	Fundamentals of Animation	2	
	Censorship in Cinema	4	
	Culture, Technology and Communications	4	
	History of the International Cinema After World War II	4	
	Television, New Media and Culture	2	
	Seminar in Film and Television Genres	4	
	Designing Online Multiplayer Game Environments	2	
	Programming for Interactivity	4	
	Game Design Workshop	4	
	Special Topics	2-4	
	Directed Research	1-12	
	Introduction to Production Design	2	
	Introduction to Production Design	2	
	Non-Theatrical Aspects of Film and TV Producing	2	
	Producing for Independent Films	2	
	Developing and Selling Your Film or TV Projects	2	
	Writing the Short Script	2	
Advanced Scene Writing Workshop	2		
Screenwriting Fundamentals	2		
Writing from the Unconscious	2		
선택 과목(4학점 이상)	Computer Graphics	3	
	File and Database Management	3	
	3D Graphics and Rendering	3	
선택 과목	3D Animation and Character Design	2	
	Introduction to Computer Animation	2	
	Culture, Technology and Communications	4	
	Seminar in Film and Television Genres	4	
	Interface Design for Games	2	
Advanced Visual Design for Games	2		

	Usability Testing for Games	2	
	Usability Testing for Games	2	
	Design and Technology for Mobile Experiences	2	
	Sound Design for Games	2	
	Business and Management of Games	2	
	Industry Workshop	4	
	Critical Theory and Analysis of Games (cross-listed with Critical Studies, CTCS)	4	
	Anatomy of a Game	4	
	Game Studies Seminar	4	
	Designing Online Multiplayer Game Environments	4	
	Programming for Interactivity	4	
	Intermediate Game Development	4	
	Advanced Game Development	4	
	Game Design Workshop	4	
	Intermediate Game Design Workshop	4	
	Advanced Game Project	4	
	Experimental Game Topics	4	
	Special Topics	2-4	
	Character Development and Storytelling for Games	4	

○ 학사학위 과정(Bachelor of Arts in Interactive Entertainment (Undergraduate))

- 이 과정은 School of Cinema-Television와 관련하여 문학, 미술 그리고 과학(Letters, Arts and Sciences) 대학에서 수여한다.
- 학생들은 문학, 미술 그리고 과학 대학에서 일반 대학 과정을 포함한 사전 과정을 수강하고 전공 과정에서 School of Cinema-Television의 커리큘럼에서 선택된 10학점의 하위 과정과 34학점의 고급과정을 포함하여 128학점을 이수하면 된다.

표 III-24. 교과과정 (Bachelor of Arts)

	과정명	학점	
필수 과목	Introduction to Cinema	4	
	Introduction to Interactive Entertainment	4	
	Fundamentals of Cinema Technique	2	
	Cinematic Communication	4	
	Programming for Interactive Media	4	
	Intermediate Game Development	4	
	Game Design Workshop	4	
	Intermediate Game Design Workshop	4	
3개과목 선택	Interface Design for Games	2	
	Advanced Visual Design for Games	2	
	Usability Testing for Games	2	

	Design and Technology for Mobile Experiences	2	
	Sound Design for Games	2	
	Business and Management of Games	2	
	Game Studies Seminar	2	
	Designing Online Multiplayer Game Environments	2	
	Advanced Game Development	2	
	Fundamentals of Animation	2	
	3D Animation and Character Design	2	
	Introduction to Computer Animation	2	
최소 1개 과목 선택	Character Development and Storytelling For Games	4	
	Film and/or Television Genres	4	
	Culture, Technology and Communications	4	
	Critical Theory and Analysis of Games	4	
	Anatomy of a Game	4	
최소 1개 과목 선택	Industry Workshop	4	
	Advanced Game Project	4	
	Experimental Game Topics	4	

- 비디오 게임 디자인과 관리 부전공 과정(Minor in video game design and management(Undergraduate))
  - 비디오 게임 디자인 부전공 과정은 비디오 게임 산업에 특화하여 인터랙티브 엔터테인먼트 분야를 준비하는 학생들에게 이론적인 개념과 실질적인 기술을 학습하게 한다.
  - 이과정은 학생들에게 22학점을 요구한다.

표 III-25. 교과과정 (Minor in video game design and management)

	과정명	학점	비고
핵심과정(20학점)	Video Game Production	4	
	Programming for Interactivity	4	
	Game Design Workshop	4	
	Designing and Producing Video Games	4	
	Intermediate Game Design Workshop	4	
선택과정 (2-4학점, 1개 과정 선택)	Introduction to Interactive Entertainment	4	
	Advanced Project Development	3	
	Designing Online Multiplayer Game Environments	2	
	Introduction to Animation	2	
	Introduction to Computer Animation	2	
	Level Design and Development for Video Games	4	
Advanced Game Design Workshop	4		

- 심대를 위한 단기 특별 코스 : 비디오게임 개발 방법을 교육하는

### 여름 캠프

- 어려서부터 비디오 게임에 익숙해진 세대는 게임을 플레이 할 뿐만 아니라 자신이 고안한 게임을 만들어 보고 싶어 하며, 이들을 위하여 십대를 위한 게임 캠프가 열리고 있다.
- 카네기멜론 대학에서도 방학 중에 고교생들을 모아서 특별 교육을 실시하기도 하고 대학원생들이 개발한 프로젝트 중에서 교육용 게임인 중동평화를 위한 평화만들기 (Peace Maker) 라는 교육용 게임을 고교생들에게 시험 적용해 보고 결과를 분석하고 있기도 하였다.
- 이런 과정들은 뉴욕 대학과 같은 대학들뿐만 아니라, 캠프 회사에 의해서 제공 되어왔다. 이 캠프는 2004년부터 250억 시장규모의 게임산업에 종사하는데 관심을 가지고 있는 대학생들과 10대들을 위한 여름방학 과정이 시작되었다.
- 미국내 게임 개발 여름 캠프 과정들이 실제로 얼마나 많이 존재하는지에 대해서 정확한 숫자를 아는 것은 불가능하지만, 캠프 전문가들은 이 수가 점차 증가하고 있으며, 또한 이 강좌들이 점차 교양으로 성장해가고 있다고 전망되고 있다.

## 3.4 일본 게임 인력 기관의 사례 분석

### 3.4.1 일본 게임산업의 특징과 게임교육

일본 게임산업의 경우 기존 만화산업과 애니메이션 산업이 가지고 있는 노하우와 강점을 배경으로 게임 소프트웨어 부분에서 급속한 성장을 보이고, 있으며 일본 반도체의 기술력을 바탕으로 한 세계 게임기 시장에서도 세계 제일의 위치를 차지하고 있다.

일본 게임산업은 초기 일본 게임산업이 오락실용 아케이드 게임에서 시작되었지만 패미컴, 겐보이, 세가 새턴, 닌텐도 울트라64, 플레이스테이션, 드림캐스트 등의 비디오 게임기의 개발에 힘입어 비디오 게임기기 시장에서 폭발적인 성장을 해 왔으며, 이와 더불어 비디오 게임기에서 지원되는

게임 소프트웨어 산업도 급성장하게 되었다.

이러한 과정 속에서 일본 게임산업계는 자신들의 위치를 유지하고, 새로운 시장에 도전하기 위해 아케이드 게임과 비디오 게임을 제작할 수 있는 전문 인력이 필요하게 되었고, 자연스럽게 아케이드 게임과 비디오게임에 관련된 전문인력들이 학원에서 양성되기 시작했다.

### 3.4.2 일본 교육기관들의 특징

첫째로 게임 교육이 철저하게 산업계와의 연관 속에서 이루어지고 있음.

- 각 학원들은 전문 게임 제작업체로부터 게임 제작에 관련된 각종 노하우를 제공받아 교육 과정에 활용하고 있으며, 게임 제작에 필요한 전문 기술을 교육생에게 강의한 다음 일정 수준에 다다른 교육생들을 다시 게임 제작업체에 파견하여 연수를 시키고 있다.
  - 게임 제작업체 측면에서는 자신들이 가지고 있는 게임제작 환경을 교육생들에게 교육할 수 있기 때문에 교육생들이 자신의 회사에 입사할 경우 쉽게 게임제작환경에 익숙해 질 수 있는 효과를 얻을 수 있다. 따라서 기존 게임 제작자와의 연결이 용이해져서 새로운 게임 개발에 소요되는 일정을 단축할 수 있게 된다.
- 두 번째 특징은 단순히 게임에 관련된 기술만을 교육하지 않고, 콘텐츠 중심의 미디어 믹스(Media Mix)의 문화를 가르치고 있다.
  - 예를 들어 비디오 게임기용 프로그램은 워드프로세서와 같은 하나의 단순한 소프트웨어로서의 가치뿐만 아니라 게임 콘텐츠 내용의 평가에 의한 가치를 인정받고 있다.
  - 따라서 일본 게임산업은 만화나 애니메이션과 같은 콘텐츠 중심의 문화상품과 밀접히 연계되어 발전하고 있다. 이러한 일본 게임산업의 현실은 교육현장에서도 절실히 드러난다. 일본 산업계는 만화, 애니메이션, 아케이드 게임, 비디오 게임을 제작 판매하면서 쌓은 노하우들을 교육기관에 전수하고 있으며, 교육기관에서는 일본 오락산업이 가지고 있는 노하우를 바탕으로 새로운 전문인력들을 양

성하고 있는 것이다.

- 일본 게임 교육은 오락산업이 가지고 있는 강점과 노하우를 바탕으로 차세대 오락산업의 주역, 특히 아케이드와 비디오 게임산업의 새 인재를 양성하는 방향으로 게임교육이 이루어지고 있다.
- 결국 일본 게임이 계속해서 세계 게임 시장에서 인정받고 있는 것은 바로 이러한 시스템에서 기인한다고 볼 수 있다. 게임산업이 쌓아온 노하우가 게임교육 기관을 통해 후배들에게 쉽게 제공되고 있으며, 이 노하우를 바탕으로 게임 제작에 관련된 전문 기술을 습득한 후 자신들이 배운 게임 제작 기술을 현장에서 다시 실습함으로써 학원생들이 쉽게 게임산업에 응할 수 있도록 하고 있기 때문이다.
- 또한 만화나 애니메이션과 같은 콘텐츠 산업과 연계된 게임산업의 성장은 콘텐츠 산업의 노하우를 게임 교육에 도입함으로써 좀더 창의적인 제품을 개발할 수 있는 인재를 양성하는 좋은 교육 교재로 사용되고 있는 것이다.

### 3.4.3 전문 인력을 상세 구분한 교육 방식의 적용

- 일본 게임 교육기관은 게임산업에 필요한 전문 인력들을 세부적으로 나누어 교육하고 있는데, 주로 게임 프로그래머, 게임 프로듀서, 캐릭터 디자인, 그래픽 디자인, 애니메이션 디자인, 시나리오 작가, 사운드 작가 등으로 나누어 교육을 하고 있다.
- 이러한 특화된 교육방식은 각 분야에서 갖고 있는 노하우를 각 분야의 지망생들에게 전문적이고 집중적으로 교육할 수 있도록 하기 위한 것이며, 역으로 기존 산업 분야에서 이러한 각 분야의 인재들을 필요로 함을 반영한 것으로 볼 수 있다.
- 즉 일본 게임산업에서 필요한 인재가 자사의 노하우를 쉽게 받아들이고 이를 응용할 수 있는 기술자들이기 때문에 일본 게임 교육기관들은 게임에 대한 학문적인 능력을 교육하기보다는 게임산업이 가지고 있는 노하우를 쉽게 받아들이고, 이 노하우를 응용할 수 있는 능력을 갖추는데 교육 목적을 설정하고 있다.

표 III-26. 교과과정의 구성 단계

단계 구분	교육 내용
초급단계	게임 제작에 필요한 기초적이고 전반적인 과목들을 교육하여 각 분야의 사람들과의 커뮤니케이션 능력을 배양
중급단계 (심화과정)	게임산업의 노하우들을 각 전문분야에 따라 체계적으로 교육
고급단계 (응용과정)	게임산업의 노하우들을 각 전문분야에 따라 심층적으로 교육 현장 실습을 통한 현장 실무 적응능력 배양

- 마지막 단계에서는 학원에서 배운 교과 내용을 바탕으로 현장 실습을 할 수 있도록 구성하여 게임산업에의 조기 적응 능력을 키우도록 하고 있다.
- 정리하면 일본 게임 교육기관은
  - 게임산업에 필요한 현장적 전문 기술자를 양성하는 것을 목표로 하고 있으며,
  - 이를 위해 게임산업이 갖고 있는 노하우를 바탕으로 세부분야의 기술적인 영역을 교육하고 있다.
  - 또한 게임산업에 교육생들이 쉽게 적응할 수 있도록 현장학습 기간을 두고 있으며, 이를 통해 게임산업이 요구하는 인재들을 배출함과 동시에 산업계에 쉽게 적응할 수 있도록 교육하고 있다.

표 III-27. 일본 게임 교육기관 (아카데미)

교육기관명	위치	개요
디지털 엔터테인먼트 아카데미 (DEA)	東京都新宿區北新宿1-5-2 佐藤ビル	유명한 게임메이커 20개사가 출자한 신동협동학원 2년제, 3년제 과정 개설
동경 게임 디자인 학원	東京都 시부야구 요요기 3-57-6	게임디자이너, 게임아티스트, 게임캐릭터디자이너, 게임CG크리에이터, 게임기획, 시나리오 라이터, 게임평론, 잡지 라이터, 게임 사운드크리에이터, 게임디자인크리에이터 과정 등의 게임 전문학원
휴먼 아카데미	東京都 新宿區 다가다노바바 4-4-2	일본에서 가장 규모가 큰 학원으로 게임기획, 프로데뷰, 3DCG, SOFTIMAGE 마스터, 캐릭터디자이너, 게임 크리에이터, CG크리에이터 강좌 등을 개설(전국 체인)
요요기 애니메이션 학원 (YAG)	東京都 시부야구 요요기 20-3 DL-11	일본 최대 애니메이션 학원으로 게임CG애니메이션게임학과, 게임디자이너, 게임캐릭터무비디자이너, 게임사운드디렉터과 등을 개설
어뮤즈먼트 미디어 종합학원	東京都 시부야구 2-29-8	CG아티스트, CG영상코스, CG디자인코스, 게임프로그래밍, 게임 기획, 시나리오, 게임그래픽, 게임크리에이터 연구코스 등을 개설
Vantan	東京都 시부야구 3-22-14-11MDC	2년제 게임 시나리오라이터, 게임그래픽스, 게임프로그래밍, 게임사운드, 게임개발연구코스 등 개설

KCE	東京都 미나토구 록뽀기 1-4-30	일본 게임개발사인 코나미가 직접 운영하는 학원으로 게임프로그래밍, 영상디자인, 음악사운드 코스 등의 강좌 개설
Digital Hollywood	東京都 치요다구 간다 14F	일본 및 미국에 분교가 있는 멀티미디어 학원 Windows NT 3DCG 애니메이션, 크리에이티브, 프로듀서, 비주얼사이언스 등의 강좌 개설
실리콘 할리우드 학원	東京都 미나토구 도라노몽 1-16-4	CG 마스터 코스, 게임 마스터 코스, 네트워크 마스터 코스, CG크리에이터 코스, 게임 크리에이터코스, 네트워크 크리에이터 코스, Alias, 그래픽디자인입문, 그래픽 디자인 스페셜 코스

### 3.4.4 학원 및 사내 교육기관을 중심으로 본 교육 과정 분석

#### ○ 교과과정 현황

- 모든 교육과정을 수료하는데 2년 이상의 기간이 책정되어 있다.
- 교과과정이 세부적이기는 하나, 네트워크 게임에 대한 교육 과정은 많지 않으며, 올해 신설되었기 때문에 그 기능 면에서 얼마나 큰 효과를 발휘할 수 있는지는 좀 더 시간을 두고 지켜봐야 할 것으로 판단된다.

#### ○ 교과과정 분석

- 게임 제작 및 게임 제작 교육의 긴 역사를 바탕으로 ,하나의 교과과정을 여러 개의 교과로 세분화 함으로써 실무현장에 직접적인 도움이 되도록 하고 있다.
- 강사진도 유명 게임 제작자를 위촉하여 현장경험 중심의 강의를 진행하게 한다. 이것은 게임이라는 문화가 사회의 일부로서 편입되어 있고, 문화적으로도 상당한 영향력을 발휘할 수 있는 상황하에서만 가능한 것이라 할 수 있다.
- 선진국 게임 학원의 가장 큰 특징 중 하나는 기초과학이나, 자기 관리 및 협동 정신 등 인간 관계형성에 연관된 교과과정을 운영하고 있다는 점이다. 게임 제작자는 다양하고 건전한 상식을 구비하고 있어야 한다는 전제 하에서, 일반 교양강좌를 비롯한 수학, 기하학, 논리학 등의 교과과정과 공수도, 꽃꽂이, 다도 교실 및 게임 디자이너의 자질 교육 등을 수행함으로써 학생 개개인에게 게임의 '문화'를 이끌어 가는 자부심을 심어주는 효과를 발휘하는 것이다.

- 게임 제작에 직접 참여하게 함으로써 게임 제작에 대한 실질적인 경험을 축적할 수 있도록 하고 있다. 교과과정 역시 이를 뒷받침 할 수 있도록 실습 중심의 교과과정으로 구성되어 있다. 이러한 교과과정은 일반적인 게임 제작의 단계를 답습하기보다는, 게임 제작 '교육'이라는 측면에서 학생들에게서 효과적인 제작의 방향성과 실무를 습득할 수 있도록 한 운영이라 할 수 있다.
- 운영에 있어서도 강의실 중심의 전통적인 면대면 강의와 함께, 위성 방송, TV 및 인터넷을 병행 이용하고 있으며, 방송강좌의 경우 학생을 방송 스태프로 직접 참여시킴으로써 실무 능력 배양에도 직접적인 도움을 주고 있다.
- 교과과정 현황 및 분석 : HUMAN CREATIVE SCHOOL (HCS)
  - 기본 2년의 교육 기간을 가지고 있으며, 그 과정은 멀티미디어, '3D 무비', '인터넷', '크리에이티브 플래닝', '상품 제작' 등으로 나뉜다. 졸업 이후 3년 차에는 게임 제작 연구과정 도 개설되어 있으며 해외 유학 코스도 있다.
  - TV 방송을 통해 강의를 하는 것이 특징이다. 이 TV 방송의 스케줄은 일일 강의 계획이 세부적으로 나와 있다.
  - 과정 중 '크리에이티브 플래닝' 과정이나 '상품 제작' 과정은 게임 제작의 비즈니스적인 측면을 고려한 과정으로 보이며, 이러한 것들은 전문적인 일본 게임업계의 현실을 잘 보여주는 것이라 할 수 있다.
  - 졸업 후 3년 차에 개설된 '게임 제작 연구 과정'은 게임 제작에 아카데미즘적인 부분을 도입한 것이라고 볼 수 있으며, '게임 제작'도 아카데미즘적인 연구의 한 부분이 될 수 있다는 것을 보여준다.
  - 일본의 학원 중에서는 유일하게 네트워크 게임 과정이 개설되어 있다. 기간은 역시 2년이다.

표 III-28. 학원 현황 (HUMAN CREATIVE SCHOOL)c

설립	90년
전화번호	0120-37-9898
주소	본관 東京都 武蔵市吉祥寺北町 1-2-12 (〒) 180-0001 2호관 東京都 武蔵市吉祥寺本町 2-17-12 (〒) 180-0001
홈페이지	http://www.human.co.jp
교육이념	- 自學, 自育, 自創 - 게임업계에 필요한 기술자를 양성한다.
강사진	20명(게임개발 경험자)
주요장비	DOS/V, Macintosh 9500, SGI 02
수강료	135만円(입학금 25만円, 수업료 60만円, 실습비 20만円, 교제비 30만円)
특징	- TV방송 강의 (SKY Perfect TV! Ch. 771) 과목 : Game Creator 양성과정 (입문편) 수강료 : 방송수강 180,000円(別) / Video 수강 250,000円(別) 교제대금 : 50,000円(別) 지불방법 : 신용카드(1회 일시불) / 지로, 은행 입금 / 12회분할 입금 해 약 : 수강료는 반환하지 않음
	- Net-work 게임 과정 개설('99.4 / 일본최초) <서버설치> 1년차 필수과목 : Net-work 概論, Server概論, 프로그램 基礎, Net-work 게임 Programming, Net-work 實習 2년차 필수과목 : NET-work 게임 Programming, 응용실기, C言語上級, C++, 작품제작

표 III-29. 교과과정(HUMAN CREATIVE SCHOOL)

과정	기간	과목	비고
Multi-media	1년 前期	기초학습	
	1년後期~2年次	프로그램 또는 그래픽 코스 선택	
3D Movie	1年次	영상제작 기초학습	
	2年次	CG기술, 3D영상제작	
Internet	1年次	Net-work 기초학습	
	2年次	Server Application	
Creative Planning	1年(新設)	기획, 계획, 제작과정, Business(Notebook 사용)	수강료 150만円
상품제작	1年(新設)	완전 OJT형 게임 개발	
졸업			
선택	3年次	게임 제작연구 과정	
		해외유학	

○ 교과과정 현황 및 분석 : 요요기 ANIMATION 학원

- '영상학부', '게임 크리에이터 종합', '예술학부', '출판학부', '종합학부', '야간학부' 등 학부의 개념을 도입하였다, 실제로 우리가 관심을 가져야 할 부분은 게임 크리에이터에 관계된 것이겠지만, 사실상 여러 문화 장르의 유기적인 통합이 게임의 발전을 가져올 수 있다고 봤을 때, 이 학원의 체계에 대해서는 상당히

- 배울 부분이 많다고 보인다.
- 각 학부에는 여러 종류의 학과가 존재하며, 교육 기간도 각 학과에 따라 다르다, 각 학부의 학과는 아주 세밀하게 나뉘어져 있으며, 각 분야의 전문가를 양성하기 위한 교육편성이라고 볼 수 있다.
  - 이 학원의 가장 큰 특징은 게임 제작 외에 문화교실이라든지 어학 교육, 교양 강좌 등이 존재한다는 것이다. 문화 교실에서는 공수도, 꽃꽂이, 다도 등을 교육하고, 자유선택강좌에서는 영어 회화나 수학, 국어(일본어), 사회 등도 가르친다.
  - 그 외에도 성공한 인물을 조사, 분석하여 희망자에 한해 무료로 수강할 수 있도록 하는 '석세스 세미나'나 위성방송수업 등 다양한 프로그램도 구비하고 있다.

표 III-30. 학원 현황 (요요기 ANIMATION)

설립	20년 이상 (영상교육 목적)
전화번호	0120-310-042
주소	東京, 名古屋 Higasi, Yokohama, Sendai, Sapporo, Hirosima, 大宮, Okinawa
홈페이지	http://www.yag.ac.jp
수업방침	상황에 대응 할 수 있는 프로를 育成
수강료	74만円 (입학수수료 2만円, 입학금 18만円, 연간 수업료 38만円, 연간 설비비 16만円) ※실습교재비, 행사비, 등(18만円~21만円), 학과별 필수 교재비 별도
주요장비	DOS/V 600대, Macintosh 300대, SGI 02 30대, 모션캡처 1대
특징	- 위성방송수업(CS Net-work, Star Com) 전국 요요기 Animation 학원을 연결하는 위성방송수업 (강사와 원생들간의 질의 응답가능), 학원생이 위성방송 스태프로 활동, 대량의 영상데이터를 실시간으로 빠르게 전달
	- BEP (SUCCESS SEMINAR) 현재 성공한 인물(60명)을 조사, 분석하여 강의 희망자에 한해 무료로 수강
	- 문화교실 공수도 교실, 꽃꽂이 강습 (1회 500円), 茶道教室(1회 300円)
	- 자유선택강좌 (전부 무료이며 중복수강 가능) (우수생은 무료로 5박 6일 美 LA연수), Computer Graphic 기초강좌, 일반교양강좌 (영어, 수학, 국어, 사회 등)
	- 학과 중복 강의시 수강료 할인
	- 야간 학습시 특전
	석식 제공, 전용기숙사(년42만円/졸업 후 취업이 결정된 사람에 한해서 2년까지) 대여(13,000円 / 朝, 夕食 제공)
	- 2000년 입학기념 Event 학비면제+12만円 또는 36만円 장학금 증정 (선발), 입학 축하금 10만円 증정 (전원)
	- 특기생 시스템 : 출석률, 성적, 향학열을 고려 년 12만円 학습 지원금 증정(년간 300명 이상 수여)
	- 특기생에서 50명 전후로 최우수생을 선발 년 36만円 수여
- Pro 제도 : 졸업생 대상으로 우수한 기획제안 또는 제작활동 지원 하나의 Project당 100만円까지 제공, 99년 졸업생부터 본격적으로 실시	

표 III-31 교과과정 (요요기 ANIMATION 學院)

학 부	학 과	기간(년)	비 고
Animation 영상학부	Animator	2	
	Digital Painter, Artist	2	
	Animation 감독	2	
	배경미술 감독	2	
	Animation AV Operator	2	
	Animation 음향감독	2	
	Special make-up Artist	2	
	SFX 감독	2	
Game Creator	CG Design	2	
	Multi-media Creator	1	
	Game Design	2	
	Motion Capture Design	2	
	Sound Director	1	
	Character Design	2	
	Game 시나리오	2	
	Digital 映像	2	
예술학부	성우, Talent	1	
	뮤지컬 舞臺, 연극	2	
	뮤지컬 Artist	2	
출판학부	만화, 코믹 Pro 양성	2	
	Entertainment Novels	2	
	Visual 일러스트	2	
종합학부 (2000년 新說)	Media 예술종합	3	동 경
	Animation 종합	3	동 경
야간학부	Game Creator	1	동 경 大 阪
	CG Creator		
	Inter-net Creator		
	Animator		
	Animation 연출		
	Comic		
	Trendy 일러스트		
	Novels		
	시나리오 Writer		
	보컬 Talent		
	성우 Talent		
Hit-make Produce			

○ 교과과정 현황 및 분석

- DIGITAL ENTERTAINMENT ACADEMY (DEA)

- 교과 과정은 초반에 독자 세미나를 통해 파트별로 강의하며, 6개월 이후에 세부적인 교육이 이루어진다.
- 이 학원의 특징은 일본의 우수 게임 기업들이 출자한 학원이라는 점으로서, 산학협동을 통한 실무진 배출이라는 측면에서 상당히 긍정적이며 여러 기업들이 공동으로 출자하여 게임실무진들을 키운다는 점에서 국내에서 모델로 삼을 만한 게임학원이라고 볼 수 있다.
- 오전반, 오후반 수업을 통해 시간이 없는 학생들을 위한 3년 코스도 개설해놓고 있다

표 III-32. 학원 현황(DIGITAL ENTERTAINMENT ACADEMY)

설립	1991. 10. 1
전화번호	03-5330-2870
주소	東京都 新宿區 北新宿 1-5-2 사토오 B/D 4F (우)169-0074
홈페이지	http://www.d-e-a.co.jp
교육이념	- 매력 있는 人間 育成 - 새로운 시스템을 개발할 수 있는 기술자의 育成 유저를 즐겁게 하는 게임 창작자의 육성
수업방침	Creator에 필요한 능력을 종합적인 분야에서 교육
출자회사	KONAMI, SEGA, SONY, SQUARE, KOEI, 닌텐도, Microsoft, BANDAI, ASCII (출판) 등 20개사
강사진	정규 강사 이외에 14명의 각 분야별 특별 강사
특징	산학협동체험
수강료	147만円 (입학금 25만円, 실습비 30만円, 수업료 60만円, 교재비 25만円 소비세 7만円)
모집요강	- 입학자격 : 18세 이상 고등학교 졸업이상 - 정 원 : 2년 코스 230명 3년 코스 오전, 오후 각 80명 - 시험구분 : 일반입학-면접 추천입학-서류접수, 면접 특기생-서류심사, 면접, 필기시험, 작품평가 (약간 명은 입학금, 수험료 면제)  ※ 조기 지원 시에는 학비 분납이 가능
학원생활	- OPEN (09:00), CLOSE (18:00, 필요시 20:00까지) - 강의시간 • 2년 코스 필수과목 : 09:30~15:00, 선택과목 : 15:00~16:40 • 3년 코스 (2년제와 수강과목은 동일) (오전반) 필수 : 09:30~12:40 선택 : 15:00~16:40 (오후반) 필수 : 09:30~16:40 선택 : 수강 없음 • 1년 코스 4월 하순 개강 年 2월 하순 종강

표 III-33. 교과과정 (DIGITAL ENTERTAINMENT ACADEMY)

학 기	과 목	
1년 前期 (共通)	프로그램, 사운드, 그래픽, 기획기초, 각종 선택과목 (Seminar)	
(選 擇)	프로그램 EXTRACT 과정	그래픽 디자인 과정
1년 後期	C言語 중심, Assembly 言語, Algorithm, 교 學, Direct-X, C ++, 각종 선택과목 (Seminar)	2D CG, 3D CG (LW 3D), Design, Real 그래픽, 원화 각종 선택과목 (Seminar)
2년	Algorithm, Assembly 言語, 프로그램 제작 (Windows), 프로그램 제작 (Net), 각종 선택과목 (Seminar)	2D CG 上級 3D CG 上級 (Softimage 3D), Design, Real 그래픽, 원화 作品制作, 각종 선택과목 (Seminar)
Master	2년차 종료 후 희망자에 한해 (99년도입), 실제 개발과 같은수준의 교육	
졸 업		
3년 코스	2년 코스와 동일 (수강시간은 오전, 오후 선택이 가능), 2000년 신설됨	

표 III-34. 교과과정 자료 (DIGITAL ENTERTAINMENT ACADEMY)

종류(Seminar)	과 목	내 용
프로그램	R3000	Playstation의 Main Switch의 명령 및 개요
그 래 픽	Design	그리고, 화면구성, 묘화력,
	原 畫	Art와 기술을 습득하기 위한 방법
	美 術	본격적인 원화 제작
	3D CG	3D와 영상제작능력 향상
	2D CG	디자인 중심의 간단한 게임 제작
사 운 드	음악제작	실기중심
	음 악	음악이론, 樂曲分 , 作曲法 等
	DTM	초보자대상, MIDI제작 기초
企 劃	기 획	시나리오를 게임적 발상으로 양성
	Project Manager	자신을 이론적으로 표현하는 방법, 리더가 되기 위한 Know-how등
	시나리오 실습	효율적인 문장 기술방법부터 세계관설정, 게임시나리오 작성까지
	Analog게임	Table-talk, 보드게임, 카드게임 등을 분석하여 이해력 향상
	게임 디자인	디자이너가 되기 위한 자질, Point, 업무이해, 기획력, 게임업계 근무방법 등
게임제작	Team 결성	학원에 신청하여 허락 후 결성 (1년차 2월)
	기획서 제출	창작성이 있는 기획서만 허락 (2년차 5~6월)
	사양서 제출	설계도 와 사양서 제출 (2년차 5~6월)
	제출	2년차 7월
	제출	2년차 9월
	완성품 제출	2년차 12월
	제작 발표회	발표회에서 전시 (2년차 1월)
	Monitoring	제품의 객관전 평가 (DEA 학생)

### 3.4.5 정규 학위 대학과정에서의 게임 교육

- 야마카타에 있는 토호쿠 예술공과대학은 '미래디자인학부 게임디자인코스'를 개설하고 있다.
- '미래디자인학부'는 e-비즈니스, 정보설계, 게임디자인, 영상설계의 4개 코스(전문강습계열)를 갖추고 있는 종합디자인학과이며, 게임디자인코스는 "게임콘텐츠 기획의 발상법을 게임 기획의 기초지식의 강의, 게임작품의 시연평가, 팀에 의한 기획제작체험 등을 통하여, 종합적으로 가르치고 있다.
- 또한 게임이 갖고 있는 상호소통성을 활용한 새로운 가능성을 모색한다."라고 대학안내에도 나와있듯이, 단순히 게임제작사로의 취업양성소적인 측면뿐만이 아니라 '콘텐츠 일반에 적용할 수 있는 기획력 양성'을 메인으로 하는 학과, 코스(일반대학적용어)라는 것이 독특한 면이다.
- 시라카미 교수를 중심으로 주식회사 남코에서 프로듀서직에 종사하고 있는 현역개발자인 나카무라 씨가 교수로 재직하고 있는 것도 주목해야만할 점이다.
- '미래디자인학부'의 학년별 커리큘럼
  - 1학년 1학기부터 컴퓨터, 콘텐츠, 플래닝 세 부문의 기초강연을 하고,
  - 2학년 2학기부터 4개의 전문분야로 나가는 커리큘럼으로 짜여져 있다.
- 영상, 웹콘텐츠, 게임기획, 상품기획과 같은 '기획'직에 필요한 표현수단과 생각하는 법을 가르치고, 학생은 '컴퍼니 시스템'에 의해 유사회사를 조직하여 재학 중부터 실천적으로 비즈니스, 기획을 한다.
  - 그 후 실제로 게임이나 프로그램, 상품 등을 제작하여, 졸업연구로 하고 있다.
- 상품으로서의 게임, 또는 거기에 달려있는 콘텐츠를 여러 방법으로 모색하여 실제로 한명부터 여러사람이 유사하게 만들어 나가면서, 현재 게임제작사에서는 당연시되고 있는 '팀으로 작품을 만드는' 흐름을 체험할 수 있다.
- 앞의 과정만을 보자면 게임스쿨에서도 하고 있는 실습과 별로 차이가 없어 보인다. 단, 그것과는 조금 다르다고 생각한 것은 실제로 보게 된 졸업연구

에서였다.

- 게임의 모션, 웃음을 표현한 게임, 캐릭터 비즈니스가 게임에 직결한다는 것, 게임언론의 '게임뇌'에 대한 반응을 조사한 것, 다양하게 그려진 '드라큐라'의 경로를 더듬어 자신만의 '드라큐라'상을 디자인 한 것, 게임적인 인터페이스를 사용한 포토앨범 등, 게임에서 길러진 발상을 다른 어플리케이션에 적용하여 실질적인 게임제작만이 아닌 '게임'을 이용한 여러가지 연구대상으로 한 것이 있었다.
- 한편 주식회사 남코의 '드래곤 클로니클 온라인'의 개발현장이나 휴대전화 콘텐츠의 제작에 인턴으로 참가한 학생이 실제 개발현장에서 접한 경험을 발표한 것도 독특하다. 이 학부에서는 기회가 있으면 그러한 제작사와의 연계도 즉시 행할 준비가 되어있다고 한다. 또한 졸업작품 중에는 실제로 상품화가 검토되는 것도 있다고 한다.
- 강의에도 남코의 현역 개발자가 몇 번이고 방문하여 개발현장의 살아있는 강연을 들을 수 있는 기회도 준비되어 있다. 이른바 예술과는 다르게, 지금의 게임제작은 집단작업이라서, 잘못하면 회사끼리의 연계에 의한 총력전의 양상을 보이고 있다. 한편, 휴대용 게임기, 휴대전화용 콘텐츠의 제작도 활발하다.
- 다양한 판도에 대응할 수 있는 기획력을 몸에 익히기 위해 여러 시도가 행해지고 있는데, 이 학교는 첫 졸업생 배출시기부터 졸업작품 중에서 두 점을 대학측에서 사들이기 시작하였다. 이렇게 저작권을 사들인 학생 작품의 제작 전과정을 관리하고 소스 등을 보관하고 관리하면서 후배들의 교육자료로 활용한다. 지금까지는 유화, 조각, 일본화, CG, 포스터 등, 이전부터 확립되어있던 분야의 작품이 대상이었으나, 당당히 게임관련 콘텐츠가 그것들과 동등하게 평가되고 있다.

## IV. 게임 산업 전문 인력 양성을 위한 한국형 교육 시스템의 제안

### 4.1 게임 산업 전문 인력 양성을 위한 한국형 교육시스템의 기본 방향

본 장에서는 앞서 살펴본 국내 외 게임 교육 기관의 분석 내용을 기반으로 현 시점에서의 국내 게임 교육 프로그램 및 시스템의 문제점을 살펴보고, 이를 극복하기 위한 “한국형 교육시스템”의 필요성과 기본 방향을 제시하고자 한다.

#### 4.1.1 국내 게임 교육 제도의 현실

최근의 국내 게임 교육 기관은 그 성격에 따라 고등학교, 2년제 대학 및 대학교, 대학원 등의 정규 교육 기관과 특수 목적의 교육 기관 및 일반 사설 학원 등으로 구성된 비정규 교육 기관으로 크게 구분 지을 수 있다. 이는 일반적인 교육 현실과 크게 다를 바 없는 구조이지만, 보다 구체적인 내용을 살펴보면 타 분야와는 다른 다소 독특한 특성이 있음을 알 수 있다.

우선, 게임 정규 교육 기관은 대부분 2003년에서 2004년 사이에 양적으로 크게 성장한 바 있는데, 2005년 4월을 기준으로 고등학교 4곳, 2년제 교과과정을 운영하는 전문대학 36곳, 4년제 대학교 18곳, 대학원 8곳 및 온라인 대학 3곳 등 총 69개 교육 기관에서 게임 교육과 관련된 학과를 개설 운영하고 있다. 현재 게임 관련 국내 정규 교육기관은 [표 VI-1]과 같이 전문대학과 대학교를 중심으로 증가하다가 2005년에 들어서면서 원격대학과 전문대학을 중심으로 오히려 감소하고 있는 추세이다. 또한, 이들 정규 교육기관의 설립·운영의 역사는 매우 일천하고 선진 해외 게임교육에 대한 벤치마킹이 부족하여 이론 중심의 교육을 실시하는 등 “창의성을 갖춘 게임 전문 인력의 양성”을 필요로 하는 산업계의 요구에 부응하지 못하고 있는 실정이다.

표 IV-1. 게임 관련 국내 정규 교육 기관 추이

구 분	2001년 (5월)	2002년 (4월)	2003년 (5월)	2004년 (3월)	2005년 (4월)
고등학교	2	2	3	4	4
전문대학	16	22	28	39	36
대학교(일반)	3	7	10	18	18
대학교(원격)	-	8	13	13	3
대학원	2	5	8	8	8
전 체	23	44	62	82	69

(출처 : 한국게임산업개발원 (2005), "2005 대한민국 게임백서", p.280)

#### 4.1.2 정규교육기관의 국내교육제도

국내 정규 교육기관 중 전문대학과 대학교에 공통으로 해당하는 교육시스템의 문제점을 나열하면 아래와 같다.

##### 1) 전문대학

2, 3년제 전문대학 교과 과정을 가지고 게임 개발에 대한 교육은 기간이 부족하다. 현업에서 MMORPG를 개발하는데 있어 적어도 2-3년 정도의 기간이 소요되고, 다른 여타 캐주얼 온라인 게임의 경우도 적어도 1-2년 정도의 시간이 소요되는 현실을 고려할 때 이에 대한 개발 관련 교육을 2년 안에 한다는 것은 매우 어렵고 난이도 있는 문제라 할 것이다.

2년제 대학 교과 과정의 경우에 있어 1학년 1학기에 보통 개념을 잡기 위한 교과 과정이 편성되어 있고, 1학년 2학기에 본격적으로 게임개발에 필요한 과목을 수강하기 시작한다. 그러나 대부분의 대학에서와 같이 졸업 작품으로서 게임 제작을 요구하고 있는 학교에서는 졸업 작품의 게임 완성도를 높이기 위하여 2학년 1학기부터 시작을 하는데 학생들 입장에서 이를 다시 보면, 아직 게임 개발에 대한 전체적 개념을 잡기 전에 게임 개발을 시작하는 것이라 볼 수 있다. 이러한 위험성은 간혹 게임개발에 대한 이론적 완성도 없이 눈에 보이는 산출물을 생산하기 위한 우회적인 방법론만을 습득할 수 있는 위험성이 있다 할 것이다.

대부분의 전문대학교수인 경우 아직도 실제 게임 개발 경험을 보유한 인력이 상대적으로 부족한 현상이다. 이는 게임업계와 학계의 산

학교류가 활발하게 이루어지고 있지 않기 때문으로 볼 수 있으며, 또한 2년제 대학의 입학 지원 감소로 인한 기존 학과가 전략적인 차원에서 변화한데 기인한 원인도 있다. 반면에 전문대학교수 인력 중 그나마 업계 경험을 바탕으로 게임 개발 인력을 양성하고 있는 교수 인력의 경우 게임 개발 경험을 바탕으로 경쟁력 있는 교과 과정을 소화하고 있다. 그러나 이러한 개발적인 측면은 아직도 많이 수정 보완되어야 하는 점들이 있다.

전문대학간의 경쟁 심화와 입학 지원 감소로 인한 구조적인 문제로 기존의 게임관련 학과에서 보유해야 하는 적정 교수 인력을 확보하는데 어려움이 있다. 이러한 문제로 기인하여 교육부에서도 교수 인력의 확보기준을 제시하여 독려하고 있으나 실제로 게임 교육을 위한 다양한 분야의 교수 인력을 전임 교원으로 확보하는 데는 어려움이 있는 것이 사실이다. 이러한 구조적인 문제는 정부 지원책에 의하여 전향적으로 검토될 필요가 있다.

표 IV-2. 국내 교육기관 현황(대학)

번호	학교명	학과명	홈페이지	지역
1	경남정보대학	인터넷응용계열, 인터넷컨텐츠전공	www.kit.ac.kr	부산
2	경동정보대학	게임애니메이션과	http://ga.kdct.ac.kr/	경북
3	계명문화대학	미디어커뮤니케이션전공	http://www.km-c.ac.kr	대구
4	계원조형예술대학	게임웨어학과	www.kaywon.ac.kr	경기
5	공주영상정보대학	모바일게임과	http://game.kcac.ac.kr	충남
6	국립한국재활복지대학	컴퓨터게임개발과	http://www.hanrw.ac.kr/game	경기
7	군장대학	컴퓨터응용계열, 컴퓨터게임전공	http://www.kunjang.ac.kr	전북
8	극동정보대학	게임디지털컨텐츠공학	http://www.kdu.ac.kr	충북
9	김천과학대학	컴퓨터정보계열, 아동컴퓨터게임전공	http://computer.kcs.ac.kr/	경북
10	김천대학	컴퓨터멀티미디어계열, 게임 멀티미디어과	http://computer.nazero.net/	경북

남해전문대학, 대경대학, 대구과학대학, 대구미래대학, 대덕대학, 동강대학, 동부산대학, 동아방송대학, 두원공과대학, 부산경상대학, 서강정보대학, 서라벌대학, 서울정보기능대학, 성덕대학, 순천제일대학, 순천침암대학, 송의여자대학, 안동정보대학, 양산대학, 여주대학, 영진전문대학, 용인송담대학, 울산과학대학, 장안대학, 재능대학, 전남과학대학, 주성대학, 창신대학, 청강문화산업대학, 한국재활복지대학, 한림성심대학, 혜전대학, 혜천대학, 부산정보대학, 서울호서전문대학교 등 총 45개 대학

## 2) 대학교

게임인력양성을 위한 전문교수인력이 부족하다는 것은 전문대의 경우와 마찬가지로 제일 큰 문제로 대두되고 있으며, 다음으로는 전임 교수들의 상당수가 IT관련 분야 출신으로 게임 개발 실무경험이 부족하다는 것이 문제이다. 따라서 게임 산업 전문 인력 양성을 위한 미래 지향적인 교과과정 편성에도 문제가 발생하게 된다. 공학계열로

소속된 대부분의 학과에서는 주로 프로그래밍 관련 교과목이 많이 편성되어 있으며, 산업현장과는 거리가 있는 단편적인 프로그래밍 기술 중심으로 교육이 진행되고 있는 실정이며 프로젝트 교육을 진행하는데도 어려움이 있는 실정이다. 그동안에 산업체의 수요를 무시하고 게임 관련학과가 양적으로 확장되어 산업체에서 요구하는 개발능력을 갖춘 인력을 배출하지 못하는 것도 사실이다.

대학교 전임교수들은 게임업체와의 교류가 미흡하여 산학협력을 추진하는데 어려움이 있다. 대부분 학과의 교과 내용이 유사한 면이 많은데, 이는 일부 대학 혹은 대학교에서 게임 관련학과를 개설할 때 새로 전임교수를 채용하기보다는 기존 교수들이 소속을 변경한 경우가 많기 때문으로 판단된다. 특성화도 부족하며 실령 특성화를 추진한다고 하더라도 교육의 효율성이 떨어지는 것이 사실이다. 대학교의 전임교원 채용 기준을 만족하는 게임업체의 경험을 가지고 있는 교수 인력 풀이 많지 않다는 점이 문제이다. 학과의 특성을 고려한 채용기준을 완화하거나 새로운 신인교수 발굴 양성하기 위하여 정부의 지원 정책이 절실히 필요하다.

표 IV-3. 국내 교육기관 현황(대학교, 대학원)

번호	학교명	학과명	홈페이지	지역
1	건양대학교	전산게임학과	<a href="http://www.cs.konyang.ac.kr">http://www.cs.konyang.ac.kr</a>	충남
2	경기대학교	일반대학원 게임웨어학과	<a href="http://gameware.kyonggi.ac.kr/">http://gameware.kyonggi.ac.kr/</a>	경기
3	경운대학교	컴퓨터공학부 게임전공	<a href="http://www.ikw.ac.kr/">http://www.ikw.ac.kr/</a>	경북
4	고려대학교	게임기술연구센터	<a href="http://www.korea.ac.kr">http://www.korea.ac.kr</a>	서울
5	공주대학교	게임 디자인학과	<a href="http://game.kongju.ac.kr">http://game.kongju.ac.kr</a>	충남
6	대구한의대학교	멀티미디어학부 게임전공	<a href="http://www.dhu.ac.kr/">http://www.dhu.ac.kr/</a>	대구
7	대불대학교	디지털게임애니메이션학과	<a href="http://home.daebul.ac.kr/~game">http://home.daebul.ac.kr/~game</a>	전남
8	동국대학교	영상정보통신대학원 게임제작전공	<a href="http://home.dongguk.ac.kr">http://home.dongguk.ac.kr</a>	서울
9	동명정보대학교	게임공학과	<a href="http://game.tit.ac.kr/">http://game.tit.ac.kr/</a>	부산
10	동서대학교	인터넷공학부 게임&멀티미디어공학전공	<a href="http://multi.dongseo.ac.kr/">http://multi.dongseo.ac.kr/</a>	부산

명지대학교, 배재대학교, 상명대학교, 세종대학교, 숭실대학교, 아주대학교, 연세대학교, 영산대학교, 예원예술대학교, 우송대학교, 위덕대학교, 전주대학교, 중부대학교, 중앙대학교, 추계예술대학교, 탐라대학교, 한국산업기술대학교, 한성전문학교, 호남대학교, 호서대학교, 호원대학교, 홍익대학교 등 총 32개 대학

### 3) 비정규 교육기관

비정규 교육 기관의 경우에도 문화관광부 산하의 게임아카데미를

비롯하여 노동부와 산업자원부, 한국 상공회의소 등 정부 기관이 개설한 다양한 형태의 직업 교육 기관을 비롯하여 서강대학교 게임교육원 등 대학 부설 교육원, 기타 방송사나 게임사 부설 교육 기관, 개인이나 단체 기반의 일반 사설 학원 등 다양한 형태의 게임 전문 인력 양성 기관이 대부분 1년 전후의 단기 과정 위주로 산업계에서 요구하는 인력 양성을 진행하고 있다.

이러한 게임 전문 인력 양성 시스템의 양적 확장과 발전은 최근 급속도로 성장한 국내 게임 산업 환경에서 단기간에 많은 수의 게임 관련 인력을 배출함으로써 관련 인력 수급에 적지 않은 도움을 준 것은 사실이다.

하지만, 현재의 국내 게임 전문 인력 양성 시스템에는 여러 가지 문제가 존재하는 것 또한 사실이다.

표 IV-4. 국내 교육기관 현황(비정규 교육기관)

번호	비정규 교육기관명	과정명	홈페이지	지역
1	온게임스쿨	게임기획전문가, 게임그래픽, 게임프로그래밍	<a href="http://www.ongameschool.com">http://www.ongameschool.com</a>	서울 강남
2	월드아트스쿨	3DMax게임그래픽과정, 3DMax게임기획과정, 게임프로그래밍	<a href="http://worldschool.co.kr">http://worldschool.co.kr</a>	서울 강남
3	아트센터스쿨오브디자인	게임디자인, 게임프로그래밍, 게임그래픽, 게임원화디자인	<a href="http://www.artcenterschool.com/">http://www.artcenterschool.com/</a>	서울 용산
4	그린컴퓨터아트학원	게임그래픽, 게임프로그래밍 게임기획 마야3D 애니메이션 캐릭터 애니메이션 게임그래픽 게임기획 게임 캐릭터 애니메이션 게임프로그래밍 마야3D 애니메이션	<a href="http://www.greenart.co.kr">http://www.greenart.co.kr</a>	서울 종로
5	게임월드아카데미	게임기획PD 게임그래픽 게임프로그래밍 게임컨셉디자인	<a href="http://www.gameworldschool.co.kr">http://www.gameworldschool.co.kr</a>	서울 강남
6	MBC아카데미게임학원	비디오게임그래픽과정 PC게임그래픽	<a href="http://www.mbcdigitalacademy.com">http://www.mbcdigitalacademy.com</a>	서울 서초
7	게임스쿨	게임프로그래밍전문가 게임그래픽전문가 3D MAX 게임기획전문가	<a href="http://www.gameschool.co.kr/">http://www.gameschool.co.kr/</a>	서울 서초
8	소니아트산업디자인학원	3DMax게임그래픽과정 3DMax 게임기획 과정 게임프로그래밍 과정	<a href="http://www.isonyart.com/">http://www.isonyart.com/</a>	서울 강남
외 22개 학원/ 게임 그래픽& 프로그래밍 중심				

#### 4.1.3 국내 게임 교육 시스템의 문제점

국제적 수준의 미래 수요 지향적인 인력 양성을 위해서는 현재 국내 게임교육프로그램에 몇 가지 문제점들을 다음과 같이 들 수 있다.

##### 1) 외부요인에 의존한 교육시스템

우선, 가장 큰 문제점은 최근의 국내 교육 현실과 관련하여 상당수의 정규, 비정규 교육 기관이 게임 분야에 대한 구체적인 인력 양성의 목표와 비전을 통해 설립되어 운영되기 보다는 기존 교육 시스템에서의 교육 인원의 감소나 구조 조정 등의 외부적 요인에 의해 기존의 학과나 학원이 명칭만을 변경하여 운영하는 사례가 적지 않게 존재한다는 점이다. 이러한 사례는 결국 충분하지 않은 준비나 부실한 교과과정 진행으로 이어져 기존의 교원이나 시설 재활용 위주의 뺄질식 교육 시스템을 초래하게 되어 교육대상자에게 열악한 수준의 교육 서비스를 제공하는 한편, 산업계 및 교육 수요층의 게임 교육에 대한 불신을 확산시키는 부작용을 일으키거나, 최악의 경우에는 해당 게임 관련 학과나 학원의 조기 폐쇄나 통합 현상 까지도 실제로 나타나고 있는 상황이다.

##### 2) 국내 게임 교육의 짧은 역사로 다학제적인 교육과정 미비

세분화되고 전문화된 특정 영역 위주의 현재의 교육 시스템이 이른바 융합 (convergence) 시대의 대표적인 산물이라고 할 수 있는 게임의 속성을 제대로 반영하지 못해 오히려 교육의 효율성을 떨어뜨리고 있다는 점을 들 수 있다. 2005년 한국게임산업개발원의 조사에 따르면 국내 정규 게임 교육 기관에 포함된 69개 교육 기관의 75개 학과는 각각 공학 계열 기반이 62곳, 예체능 계열 기반이 11곳, 기타 비즈니스 또는 e-sports 기반이 2곳으로 구분되는 것으로 나타났다.

실제로 국내에 존재하는 대다수의 정규 게임 교육 기관이 게임 교육의 기반을 게임소프트웨어 개발 기반만의 공학 중심으로 진행하고 있거나 그래픽 디자인만을 중심으로 한 예체능 중심으로 구분지어 진

행하고 있다는 점을 실제 각 교육 기관의 교과과정 분석을 통해서 쉽게 파악할 수 있다. 이는 대다수의 게임 관련 교육 기관에서 게임만이 가지는 독특한 특성, 즉, 공학을 기반으로 예술과 인문학이 고루 배합되어 있는 문화 상품으로써의 특징을 갖는 점을 제대로 반영하지 못하는 교육이 이루어지고 있음을 의미한다. 물론, 이러한 문제점은 최근 학제와 영역을 뛰어넘는 다양한 모델을 도입함으로써 실험적으로 해결되고 있긴 하지만, 기본적으로 국내 대다수의 교육 기관에서는 교육 구성원이나 해당 기관의 영역적 특성으로 인하여 특정 분야 중심의 진행이라는 이른바 “절름발이식 게임 교육”의 한계를 교육생에게 강요하고 있는 실정이다.

3) 산학연 유기적 결합 부족으로 종합적이고 실무적인 교과과정 미비  
단순한 강의의 연속과 개별적이고 부분적이며 게임 가가의 부분이 유기적인 관계없이 진행되는 국내 게임 산업계의 현실과 동떨어진 교육 시스템 운영을 들 수 있다.

국내 게임 산업 분야의 현재는 PC, 임베디드 단말이나 콘솔 등의 플랫폼, 심지어는 초고속인터넷망이나 개인이동통신 네트워크의 특성까지를 반영하는 특성을 가지고 있는 다소 구조적으로 복잡한 분야로 성장하고 있다. 즉, 실제 산업 현장에서의 게임 제작은 직접 개발해본 경험과 노하우를 필요로 하는 실무 제작 기반의 분야인데, 국내의 교육 현실은 일단 게임을 가르칠 수 있는 전문 교육 인력 확보 자체도 어려울 뿐더러 각 교육 기관의 특징적 한계로 인해 실무형 교육 운영에 있어 다양한 어려움을 겪고 있는 상황이다. 실제로, 4년제 대학교와 대학원은 학술적 학문적 연구에 치중하거나 특정 분야에 집중하는 한계로 인하여 배출 인력에게 충분한 게임 개발 경험을 보장하지 못하고 있으며, 국내 게임 교육의 52%를 차지할 정도로 그 비중이 높은 전문대학은 2년이라는 짧은 교육 기간으로 인해 충분한 전문성을 가진 인력을 배출하지 못하고 있다. 아울러, 입시와 연계된 고등학교는 산업 교육의 한계를 절감하고 있으며, 비정규 교육 기관은 1년 남짓한 기간을 활용하기 위하여 개발 툴 운영 위주의 한계적인 인력만을

양산하고 있는 수준이다.

이러한 상황에서 게임과 관련된 세 개의 국가 공인 자격 제도를 운영하고 있는 정부조차도 개발 경험이나 능력을 파악하기 힘든 기존의 이론 평가 위주의 국가 검정 제도의 현실적인 한계로 인하여 게임 자격증 제도의 확실한 위상을 확립하지 못하고 있고, 교육인적자원부 또한 최근에서야 게임 분야에 있어서의 “표준교과과정” 개설의 필요성을 절감하는 수준 정도에 그치고 있는 상황이다.

#### 4) 수요 지향적 창의성 관련 교과부족

게임 산업은 새로운 아이디어와 문화적 수준을 요구하며 지속적으로 발전하는 ‘융합 기술’이다. 따라서 게임 산업계는 독창성과 문화 창의성을 갖춘 게임 전문인력의 양성을 끊임없이 요구하고 있다. 게임 교육에서의 창의성 교육의 결여는 단순 기술자만을 양성하는 교육이며 산업계의 요구를 충족시키지 못하게 되어, 궁극적으로 외국의 기술에 종속되는 결과를 가져올 수 있다. 종합예술인 게임에서의 창의력은 게임의 다양성과 게임의 질을 향상시키는 가장 중요한 요인이며, 이는 곧 미래의 부가가치 창출로 이어진다. 우리가 점유하고 있는 선두분야 기술을 지속적으로 발전시키고 아직 미진한 분야의 기술을 획기적으로 향상시키기 위해서는 창의력이 풍부한 고급인력이 반드시 필요하다는 점을 강조하고자 한다.

이는 국내 교육 시스템과 관련한 모든 주제에서 항상 다뤄지는 단골 메뉴와 같은 문제이지만, 게임 분야가 향후 차지하게 될 산업적 문화적 역할을 고려할 때 가장 절실하고 심각한 문제일 수 밖에 없는 사안이기도 하다.

최근 게임은 최신의 정보 통신 기술을 집대성하여 해당 국가나 지역의 문화를 담는 대표적인 문화 상품으로써 거듭 나고 있으며, 영화나 드라마, 애니메이션, 만화 등의 소재 시발점으로써 그 중요성을 더하고 있는 상황이다. 하지만, 최근의 국내 게임 산업의 흐름은 성공한 게임과 유사한 아류작을 양산하여 시류와 경제성을 따르려 할 뿐, 독창적인 시도나 다양성이 부족하고 기능성 게임 등의 신규 분야에 대

한 관심은 미미한 것이 현실이다.

이러한 상황에서, 미래를 책임지게 될 교육 분야에서도 창의적이고 개성을 강조하는 교육이 결여되어 기술 및 기능의 모방, 혹은 경제성만을 강조하는 게임 위주로 개발이 이루어질 경우에는 향후 유비쿼터스 및 통신 방송 융합 시대에 있어 가장 대중에게 친숙하면서도 발전 가능성이 높은 게임 산업 분야의 국가 경쟁력 확보가 어려워짐은 물론, 디지털콘텐츠 분야 자체의 해외 종속을 심각하게 걱정하여야 할 상황을 야기할 수 있을 것이다.

한편, 앞서 살펴본 국내 게임 교육 현실의 문제점은 최근의 산업 현실과 맞물려 더욱 심각한 상황을 맞이하고 있다. 이는, 단 시간에 급작스럽게 성장한 국내의 게임 산업 현실이 온라인게임 및 모바일게임 분야에서 상대적으로 우월한 지위를 차지하고 있지만, 국내 게임 시장 수요 자체의 한계, 세계적인 게임 시장의 주류라 할 수 있는 아케이드게임과 콘솔게임 분야에서의 역할 부족, 잠재 경쟁국인 중국 및 인도 시장의 유동성 등으로 인하여 산업의 발전 방향에 큰 변화가 있을 수 있기 때문이다. 특히, 현재의 국내 게임 개발과 관련된 전문 인력 양성과 관련하여서는 메이저 업체를 비롯한 대다수의 산업체가 기존 개발자나 신규 인력을 발굴하여 재교육시키기 보다는 높은 순이익과 현금 유동성을 기반으로 이른바 “빼가기” 식의 스카우트 체계 위주로 개발진을 유지하고 있는 것 또한 중장기적으로 전문 교육 기관 중심의 체계적인 게임 전문 인력 양성의 개선이 더욱 중요해지는 이유가 되고 있다.

#### 4.1.4 선진해외 게임 교육프로그램 사례

현재까지 우리나라 게임 교육은 게임 그래픽과 게임 프로그래밍 교육에 치중하여 온 것이 사실이다. 게임 개발의 후발 국가로서 선진국의 기술 수준에 도달하기 위해서는 이 두 분야의 교육을 먼저 실시한 것은 온라인 게임 등에서 어느 정도 효과를 보고 있다고 할 수 있다. 그러나 창의력을 바탕으로 게임을 기획하고 독자적으로 개발하지 않는 한 게임 하청공장으로

의 역할 이상은 기대할 수 없다. 선진국들은 게임산업이 미래 국가발전의 주요 전략으로 자리잡게 될 것을 예측하고, [표 3]과 같이 체계적이고 종합적인 교육센터를 설립·운영하는 등 막대한 투자를 아끼지 않고 있다. 이러한 교육센터들은 실질적이면서 다양한 프로그램을 운영하며 창의력 배양과 향상을 목표로 양질의 교육을 실시하고 있다.

표 IV-5. 해외 선진 게임 관련 정규 교육기관 현황

구분	대학명	장점	특화내용	비고
미국	디지펜대학	<ul style="list-style-type: none"> <li>▶물리와 수학 등 기초과학을 포함한 다양한 학문을 습득하도록 하여 학사학위를 수여함.</li> <li>▶게임전문대학으로 전문가 양성이 목표</li> <li>▶H/W 등의 서킷보드를 조립하여 컴퓨터 내부동작을 이해하는데 교육의 초점</li> <li>▶실전위주 게임프로그램 교과목 강조</li> </ul>	▶미국 유일의 4년제 컴퓨터대학교로 설립됨.	물리와 수학 등 기초과학을 바탕으로 한 교육
	카네기멜론대학 (ETC)	<ul style="list-style-type: none"> <li>▶다양한 전공의 학부생을 석사과정으로 유도</li> <li>▶팀 프로젝트 베이스의 결과물 도출과 대규모 평가회</li> <li>▶다수의 교육센터 순회연수 프로그램</li> <li>▶게임 산업체와 연계된 인턴십 운영</li> <li>▶게임관련 다양한 분야의 강사진 확보</li> </ul>	<ul style="list-style-type: none"> <li>▶대학 내 독립적인 ETC 운영 (Entertainment Technology Center)</li> <li>▶석사과정 교역의 등록금 수수</li> <li>▶게임관련 산업체와 고정적인 인력 공급관계 유지</li> <li>▶단기 연수 프로그램 운영</li> <li>▶세계 각지에 부설 교육센터 설치 운영</li> </ul>	다학제간의 교류를 통한 연계교육
	유엔대학	<ul style="list-style-type: none"> <li>▶게임산업에 종사하는 산업인력 훈련에 교육의 목표를 둠</li> <li>▶애니메이션, 그래픽 등의 최신의 컴퓨터 게임 디자인 기술을 교육.</li> <li>▶학생들의 적성에 따라 게임감독자, 게임 설계자, 게임프로그래머 등의 훈련을 쌓을 수 있음.</li> </ul>	▶오랜역사를 바탕으로 축적된 기술의 넓은 활용	적성을 살린 교육
일본	오사카 전기통신대학	<ul style="list-style-type: none"> <li>▶전자/정보/통신을 바탕으로 한 교육인프라</li> <li>▶대학 및 대학원의 다양한 교육</li> </ul>	▶디지털 게임 학부 및 대학원 과정 신설	신설학과를 통한연계교육
	일본공학원전문학교	▶게임소프트/CG/디지털크리에이티브 코스 등 멀티미디어과를 통한 전문교육 가능	<ul style="list-style-type: none"> <li>▶2D/3D 네트워크 게임의 작품제작 중심의 교육</li> <li>▶실무에서와 같은 기자재 및 학습환경</li> </ul>	실무와 같은 학습으로 실무형 교육개발
	일본전자전문학교	▶IT, CG, 게임, 애니메이션의 분야로 현재 분야별 다양한 학년제로 교육 및 운영	<ul style="list-style-type: none"> <li>▶3DCG 제작을 초보에서부터 전문가까지 배울 수 있는 과정</li> <li>▶업계가 필요한 게임개발자 육성 과정</li> </ul>	전공분야별 다 학년제를 통한 교육
유럽	헐대학	<ul style="list-style-type: none"> <li>▶런던대학의 분교로 시작</li> <li>▶기초학문 및 응용학문 특히 컴퓨터 관련 학문이 우세</li> </ul>	▶컴퓨터과학, 전자공학 및 로보틱텔레통신과학, 공학디자인등 공학 및 네트워크이 강하다	기초학문을 바탕으로 한 응용학문

#### 4.1.5 게임 산업 전문 인력 양성을 위한 한국형 교육시스템의 필요성

앞서 살펴본 바와 같이 현재 국내의 대다수의 게임 교육 기관 및 시스템은 상당히 큰 문제에 직면해 있음을 알 수 있다. 이는 빠르게 변화하고 있는 게임 산업의 특성과 중국 및 인도를 중심으로 한 경쟁국의 발전을 고려해 볼 때 하루빨리 개선해야만 하는 문제점들이라고 할 수 있다.

물론, 교육과 관련된 문제는 단 시간에 해결할 수 있는 부분이 아니다. 특히, 빠르게 변화하는 정보 통신 기술의 첨단에서 있는 게임 산업 분야에 있어 교육 방향을 특정한 방향으로 지정하는 것 자체가 어리석은 판단일 수도 있다. 하지만, 앞서 살펴본 국내 게임 교육 시스템의 문제점은 향후 국내 게임 산업이 얼마나 발전하고 도약하느냐를 떠나 생존 자체를 판가름하는 중요한 변수가 될 것은 자명해 보인다. 이는 콘텐츠 산업 분야 특히 디지털콘텐츠 분야에 있어 전문 인력이 콘텐츠 개발에 있어 어떠한 역할을 하느냐를 고려해 볼 때 너무나 명확한 문제이기도 하다. 앞의 문제점들을 해결하기 위한 구체적인 방법으로는 다양한 해결책이 있을 수 있겠지만, 각각의 문제를 사안별로 해결하기 보다는 국내의 게임 산업만이 가지는 독특한 특성을 해결하기 위한 보다 구체적인 방향성이 있어야 할 것으로 판단된다.

즉, 온라인게임과 모바일게임을 중심으로 단기간 내에 성장한 국내 게임 산업만의 독특한 환경에 적합한 교육 시스템을 구축하고 이를 각 교육 기관의 특성과 방향성에 맞게 실천해 나가는 것이 바람직해 보인다는 것이다.

본 연구 보고서에서는 이를 “한국형 교육시스템”으로 정의하고 “미래 지향적 창의성 개발 교육”의 두 가지 내용을 기본 방향으로 부여하고자 하였다.

##### 1) 미래 지향적 창의성 개발 교육

게임 산업이 여타 제조 산업 분야나 정보 통신 산업 분야와 구분되는 가장 큰 특성은 게임 산업 자체가 바로 “콘텐츠 산업”이라는 점일 것이다. 즉, 원가를 낮추고 생산성을 극대화하는 노력보다는 개개인의 창의성을 기반으로 한 창작적인 요소가 더 중요한 산업이란 것

이다.

따라서, 현재의 개발 도구나 툴 (tool) 위주, 결과물 완성도 위주의 교육시스템에서 벗어나 창의적 개발 인력 양성을 위한 기초적인 교육 시스템 개편이 요구된다.

물론, 창의력 교육은 단기간에 완성되기 보다는 오랜 기간에 걸쳐 필요한 것으로, 창의적인 게임 개발자는 단시간에 걸친 집중적인 교육을 통해 만들어지기 보다는, 어린 시절부터 지속적으로 접하는 기초 교육과 환경으로부터 자연스럽게 창의력을 익히고 이를 생활 속에 적용해가는 습관이 생겼을 때 자연스럽게 만들어지는 것이다. 이러한 관점에서, 전체 청소년의 50% 이상이 상당한 시간을 온라인게임 등 게임 문화에 할애하고 있는 현실에서는 게임의 유해성을 거론하며 이를 하지 못하게 하는 해결책을 찾기보다는 오히려 건강하고 건전하게 게임을 즐길 수 있는 방법적인 부분을 교육하는 것이 중장기적으로 디지털콘텐츠 산업을 양성화하고 창의적인 개발 인력을 양성하는데 크게 도움이 될 것으로 보인다. 즉, 현재 시행되고 있는 영어나 컴퓨터 교육과 마찬가지로 관점에서 게임 등의 디지털콘텐츠를 바르게 선택하고 활용할 수 있는 교육을 “8차 교육 과정 개편” 등에 반영 하는 것은 건전한 게임 문화 형성은 물론 궁극적으로는 창의적인 게임 개발 능력으로 이어질 것이라는 점이다.

이러한 기초 창의성 교육의 저변을 기반으로 “한국형 교육시스템”에서는 창의성 향상을 위하여 아이디어 발상 훈련이나 게임 심리 등의 교과목 개설은 물론, 독서 프로그램, 문화체험 프로그램 등 다양한 부가 활동이 포함되어야 할 것으로 판단된다. 특히, 동아리나 팀 활동을 통한 다양한 체험 학습은 창의성 향상에 있어 밑거름 역할을 하므로, 고등학교 및 대학의 정규 교과 과정에 이를 여러 가지 방법으로 포함시키는 것도 고려해 볼 만한 사안이라고 생각된다.

최근 급속도로 진행되고 있는 정보통신 기술과 컴퓨터나 이동 단말기 등의 하드웨어 발전은 가상현실 등의 새로운 미디어 형태를 양산할 것으로 예상되는 바, 이러한 환경에서는 기존의 게임과는 전혀 새로운 형태의 게임성이나 인터페이스가 요구되는 등 더욱 창의적인 계

임 개발 인력의 경쟁력이 더욱 돋보일 것으로 보이며, 이 시점에서는 해당 국가나 지역의 건전한 게임 이용 문화가 자연스럽게 다양한 게임 개발의 토양이 되어 실제 게임 개발 시장에 상당한 영향을 줄 것이다. “한국형 교육시스템”의 가장 중요한 특징이 될 창의성 개발 교육은 향후 국내 게임 산업이 다양한 플랫폼에서 성공적으로 발전할 수 있는 가장 탄탄한 토대가 될 것이다.

## 2) 실제 개발 경험을 중시하는 실무형 교육

고등 교육시스템이 가지는 가장 중요한 포인트는 바로 전문성이라고 할 수 있다. 하지만, 게임과 관련하여서는 이러한 전문성은 때때로 해가 되기도 한다. 즉, 게임 프로그래밍 혹은 게임그래픽의 한 분야만의 집중적인 교육을 받는 피교육자가 전체적인 게임 제작의 흐름을 이해하지 못하는 경우가 종종 있는데, 이는 게임이 프로그램 기술, 혹은 그래픽 기술, 혹은 게임성의 하나만으로 완성될 수 없는 독특한 성향을 가지고 있기 때문이다. 이는 실제 산업 현장에서 개발자 간의 의견 충돌을 야기하거나 심지어는 시장에서 외면받는 게임 제작물로 표출되기도 한다.

“한국형 게임 교육시스템”의 또 다른 기본 방향으로는 게임 교육의 전체적인 방향성 측면에서 특정 분야에 한정된 교육보다는 제작 경험 자체가 동등하게 중시되는 실무 제작형의 종합적인 교육이 진행되어야 한다는 점을 들 수 있다. 즉, 특정 분야 위주의 획일적이고 단순화된 교과과정 운영 보다는 게임을 구성하는 다양한 분야 전체에 대한 종합적인 관점과 이해를 실제 제작 경험을 수반하는 교육 시스템을 통해 교육생에게 제공하는 것이 필요하다는 것이다.

실제로, 게임에 적용되는 각 기술 요소나 예술 요소는 각각의 개체로 독립적으로 존재하는 것이 아니라 다른 분야와의 상호 연계 - 예를 들면 프로그래밍과 그래픽, 그래픽과 사운드, 프로그래밍과 인터페이스 등- 를 통해 운영, 진행되게 된다. 이러한 사실은 실제 팀 구성을 통한 게임 개발 경험을 통해서야 교육 대상자에게 제대로 전달되어질 수 있으며, 이는 게임 관련 전문 인력 양성이 게임 제작 과정

전반에 대한 종합적인 이해와 실무를 기반으로 이루어져야 함을 반증한다.

또한 이러한 실무형 교육시스템은 구체적으로는 게임 전문 인력 양성과 관련한 각 교육 기관의 역할 구분을 통하여 산업계가 필요로 하는 다양한 계층의 인력 수요를 적절하게 공급하는 방안이 전제되어야만 한다. 이는 향후 더욱 다양화되고 전문화될 게임 제작 기술 발전에 있어서 개발과 연구가 어중간하게 진행되는 현재의 교육 시스템에서 배출되는 인력이 경쟁력 향상에 도움이 되지 않는다는 이유 때문이다.

2년제나 4년제 대학은 개발 경험 중심의 실무형 교육을 통해 전문 개발자를 양성하고, 대학원이나 재교육 과정 등에서는 이에 대한 학문적 학술적 깊이를 늘려 관련된 연구 인력을 양성하는 식의 역할 분담이 이루어지지 않는다면 이미 게임 개발 인력의 구성과 다양성이 상대적으로 탄탄한 경쟁국의 장벽을 넘는 것이 불가능한 것은 물론, 가까운 장래에 핵심 게임 제작 기술마저도 해외에 종속될 우려마저 높기 때문이다. 따라서, 대학원은 게임 저변 기술 연구에 적합한 실무형 교육을 실시하고, 전문대학은 실제 게임 제작을 책임질 개발 인력을 양성하는 등의 역할 구분을 통해 기술 선도 인력과 핵심 개발 인력을 동시에 양성하는 전략이 필요할 것이다.

실무형 교육을 중시하는 “한국형 게임 교육시스템” 이 자리를 잡기 위해서는 정부 차원의 각종 지원 정책과 제도 개선도 절실히 보인다. 특히, 학습자가 실제로 구현해 본 게임 결과물이나 연구물을 산업계와 연계시킬 수 있는 공모전의 활성화나, 게임 교육과 관련한 최소한의 가이드라인이 되는 “표준교과과정” 을 실무형 교육시스템으로 변경하는 작업은 매우 큰 역할을 할 것으로 판단된다. 이 외에도 게임 자격증의 개선과 관련하여 실제 개발 경험 및 능력에 대한 평가 방식이 적용되는 것은 중장기적으로 정규 및 비정규 교육기관의 게임 교육의 방향을 바꾸는 중요한 요소가 될 것이다. 이러한 관점에서 2004년부터 한국교육개발원이 단계적으로 지정 공시하고 있는 게임 분야 “표준교과과정” 에 게임 관련 기술 전체를 이해할 수 있는 공통 교과

목이나 실무 개발 과정이 포함되는 것은 다행스러운 일이라 할 수 있다.

이 외에도 현재 상당히 부족한 실무 위주의 게임 관련 전문 교육 인력을 정부 차원에서 양성하거나 기존 교수 인력에 대한 지속적인 재교육 환경을 제공하는 것도 중장기적으로 게임 전문 인력 양성의 흐름을 건전하게 변화시키는 중요한 요소가 될 것이다.

#### 4.1.6 게임 산업 전문 인력 양성을 위한 게임 교육개선방안

국제적 수준의 미래, 수요 지향적 게임교육 프로그램 개발을 위해 무엇보다 중요한 것은 경쟁력 있는 교과 과정을 운영하기 위하여 우수한 교수 요원의 확보가 가장 중요하다. 게임을 개발하기 위하여 게임을 종합 엔터테인먼트적인 콘텐츠 개발로서 인식하여야하고 이를 제작하기 위한 교육이 필요하다. 그러나 아직 대학 교과 과정은 이를 뒷받침하기 위한 교과 과정이 부족하다 할 것이다. 현재 교육부 게임 교육 표준 교과 과정에 따르면 그래도 이를 수행할 수 있는 제도적 장치는 어느 정도 제시하고 있다고 볼 수 있다. 그러나 현실적으로 교육적인 측면에서 하나의 학과에서 학생들에게 기술 기반의 프로그래밍, 예술 기반의 그래픽, 문학적 기반의 기획적인 요소를 충분히 교과 과정으로 소화하고 있다고는 판단할 수 없다. 이는 아직 게임 개발을 위한 교과 과정이 개발 중이고 발전하고 있기 때문이기도 하지만 게임 교육을 이런 콘텐츠적인 측면에서 충분히 소화하기 위한 교수 인력의 절대적 부족에서도 기인한다 할 것이다. 또한, 교과 과정이 기술적으로 업계에 비하여 뒤처지고 있다. 전문대학과 대학교를 중심으로 게임 산업 전문 인력 양성을 위한 개선 방안을 다음과 같이 제시하고자 한다.

##### 1) 전문대학

2년제 대학의 경우 게임 교육에 필요한 한계적 임계 시간이 절대적으로 부족한 상태이다. 그러므로 이를 시스템적으로 극복할 수 있는 게임 산업적 역할 분담이 이루어져야 한다. 이러한 분야별 역할 분담은 우선 2년제 대학은 짧은 교육 기관에도 불구하고 단점을 극복할

수 있는 교육 과정을 개설하여야 한다.

이러한 대안으로 2+2 연계 교육과정을 통한 게임 개발 교육에 있어 연속성 확보를 제안할 수 있다. 2+2 연계 교육 과정은 이미 다른 분야에서도 많이 시행되고 있는 사항으로 이를 게임 분야에서도 확대 적용할 필요가 있다. 실질적으로 이를 운영하기 위하여서는 학문적 연결성을 위하여 전문대학과 대학교 게임 학과 간에 긴밀한 협조가 필요하며, 이를 위한 교과 과정도 상호 협력적인 모델로서 운영되어야 할 것이다. 실무 연수 인력 양성을 위한 2+2의 연계과정으로서 전문대학 졸업생을 대상으로 (재)한국게임산업개발원 산하인 게임아카데미와의 수직적 연계를 통하여 프로젝트 중심의 교육을 실시함에 따라 상호 보완적으로 운영할 수 있다.

아울러 전문대학에서 게임 관련 전공을 이수하지 않은 학생들의 경우 게임 분야에 진출하기 위하여 전문적인 실무 프로젝트 중심의 게임 교육기관인 게임아카데미를 통하여 업계의 필요한 인력 충원 및 전문대학에서 자신의 적성에 맞지 않은 전공을 이수한 인력을 게임산업으로 유도한다. 게임 개발에 있어 상대적인 시간적 제약 요건에 맞추어 요즘 중요시 되고 있는 게임 운영에 특화하여 교육 과정을 개설할 수 있다. 이는 게임의 개발 환경에 대한 전반적 이해를 바탕으로 마케팅적인 요소를 이해하여 게임의 원만한 운영자로서의 역할을 2년제 대학에서 전담하여 인력을 배출할 수 있을 것이다.

짧은 교육 기관에 맞추어 현장 중심의 실질적인 게임 교육이 될 수 있도록 프로젝트 교육을 강화하는 것이다. 이를 위하여 새로운 교과 과정을 개발할 필요가 있으며, 한 학기를 16-20학점 단위의 단일 프로젝트 교과 과정의 운영을 고려할 수 있다. 이러한 단일 교과과정의 장점은 프로젝트를 진행하면서 학생들의 요구 사항을 능동적으로 운영할 수 있으며, 학생들은 게임 개발에 대한 전체적인 경험을 업계의 신입사원이 될 예비 지원자로서 사전 경험하게 되는 장점이 있을 수 있다.

또한 이러한 프로젝트 교육은 산학 연계 프로그램으로 확장하여 보다 현장 중심 적인 교육이 될 수 있을 것이다. 전문대학의 특성상 전

임 교원들의 시수 보장에 따른 산업체 인력을 활용한 다양한 과목의 개설이 어려운 상황이다. 이러한 측면을 해결하기 위하여 전임 교원들의 시수를 어느 정도 선에서 유통성 있게 운영하는 것이 필요하다.

이렇게 확보된 시간을 산업체 인력에게 배정하여 학생들로 하여금 다양한 현업과 관련된 과목을 이수하게 할 수 있다. 또한 학생들에게 교과목을 선택적으로 선택할 수 있는 시스템을 더욱 활성화하고 학생들이 선택하지 않은 과목은 과감히 폐강하여 과목 간 경쟁 유발 체계도 갖추는 것이 타당하다.

표 IV-7. SWOT분석(전문대학)

<p><b>강점</b></p> <ul style="list-style-type: none"> <li>○ 현장 중심의 교육이 가능</li> <li>○ 산업체 경험 교수의 상대적 이동 편리</li> <li>○ 놀이, 기술, 그리고 예술적 종합엔터테인먼트로서의 게임 콘텐츠 개발을 위한 교육 과정 운영의 효율성 확보</li> </ul>	<p><b>약점</b></p> <ul style="list-style-type: none"> <li>○ 교육 기간이 짧다</li> <li>○ 교과과정의 개발 및 적용이 상대적으로 다른 산업 분야에 비하여 미약</li> <li>○ 산업체에 비하여 기술적 우위 미확보</li> <li>○ 구조적인 문제로 인하여 다양한 게임 분야의 교수인력 확보의 어려움 상존</li> </ul>
<p><b>기회</b></p> <ul style="list-style-type: none"> <li>○ 온라인 게임 업계의 특성상 다양한 계층의 게임 운영 인력을 요구</li> <li>○ 다양한 장르의 게임 개발로 인하여 다양한 개발 인력 풀 요구</li> <li>○ 게임 산업에 대한 사회 인식의 긍정적인 변화</li> <li>○ 정부 기관이 주요 성장 동력원으로서 게임 산업을 인식</li> </ul>	<p><b>위협</b></p> <ul style="list-style-type: none"> <li>○ 게임산업의 구조 조정 가능성</li> <li>○ 전문대학의 입학 자원 축소로 인한 경쟁 가열</li> <li>○ 국내 게임 산업의 장르 편향에 따른 인력 구성의 편향</li> <li>○ 전문대학의 구조적 한계로 인한 우수 인력 확보의 어려움</li> </ul>

## 2) 대학교

게임인력양성을 위한 전문교수인력이 절대적으로 부족하며, 전문교수인력을 양성하기 위하여 신진교수 발굴과 기존 전임교수들의 재교육을 위한 정부의 지원이 절실히 필요하다. 현장감 있는 프로젝트 교육을 진행하기 위하여 산업체 인력을 교수 요원으로 채용할 수 있도록 정책적으로 정부에서 지원한다. 또한 이러한 프로젝트 교육은 산학연계 프로그램으로 확장하여 게임제작 실무 중심의 교육과정을 운영하도록 한다. 게임업체와의 교류를 활성화하고 산학협력을 통하여

학생들로 하여금 다양한 업체의 경험을 하게하고 이와 관련된 과목을 이수하게 한다. 교과과정 개편사업을 통하여 업체에서 필요한 교육이나 특성화를 유도하여 학생들의 경쟁력을 향상시킨다. 지원 시에 대학교의 교과과정도 자문해 준다.

전문대학 졸업 인력을 수용하기 위하여 전문대학과 대학교의 게임학과 간에 긴밀한 협조가 필요하며, 이를 위한 교과 과정도 상호 협력적인 모델로서 운영되어야 할 것이다. 대학 게임동아리 지원사업을 통하여 학생들에게 게임제작 경험을 유도한다. 학내에서 게임제작에 관한 지도를 받기 어려운 경우에도 게임 동아리 지원사업을 통하여 게임업계의 전문가들의 지도를 통하여 게임제작 경험할 수 있다.

표 IV-8. SWOT분석(대학교)

<p><b>강점</b></p> <ul style="list-style-type: none"> <li>○ 게임업체 경력 우수 교수 채용에 유리</li> <li>○ 게임에 대한 사회적 인식 변화로 우수 학생의 확보 가능</li> <li>○ 이로 인한 연구 인력 양성 가능</li> <li>○ 정부지원을 받고 있는 대학교의 경우 산학협력이 활성화되고 있음</li> </ul>	<p><b>약점</b></p> <ul style="list-style-type: none"> <li>○ 게임업체 경력이 있는 전임교수의 부족</li> <li>○ 게임업체에 비하여 기술적 열세</li> <li>○ 공학계열로 게임학과의 편중</li> <li>○ 채용조건으로 인하여 다양한 게임 분야의 교수인력 확보의 어려움</li> </ul>
<p><b>기회</b></p> <ul style="list-style-type: none"> <li>○ 다양한 장르의 게임 개발로 인하여 다양한 개발 인력 풀 요구</li> <li>○ 게임 산업에 대한 사회 인식의 긍정적인 변화</li> <li>○ 정부 기관이 주요 성장 동력원으로서 게임 산업을 인식</li> </ul>	<p><b>위험</b></p> <ul style="list-style-type: none"> <li>○ 국내 게임산업의 구조 조정 가능성</li> <li>○ 국내 게임 산업의 장르 편향에 따른 인력 구성의 편향</li> <li>○ 많은 유사한 게임학과가 양적으로 급히 확장되어 경쟁 과열</li> </ul>

## 4.2 게임 산업 전문 인력 양성을 위한 교육시스템

게임 관련 전공이 설치되어 있는 대학을 중심으로 4년제 대학 10개와 전문대학 11개 대학을 표본 조사하여 커리큘럼을 분석한다.

조사대상은 호서대학교, 공주대학교 등 4년제 대학과 혜천대학, 계명문화대학 등 전문대학, 울산애니원고등학교, 게임과학고등학교를 포함하여

6개 고등학교에 대해 교육과정을 조사하였다.

표 IV-8. 커리큘럼 조사 대상 고교 및 대학

구 분	내 용
대학교	호서대학교, 홍익대학교, 공주대학교, 동명대학교, 영산대학교, 예원대학교, 우송대학교, 위덕대학교, 중부대학교, 한국산업기술대학교
전문대학	대경대학, 동강대학, 부산경상대학, 순천청암대학, 영남이공대학, 장안대학, 재활복지대학, 혜천대학, 경동정보대학, 계명문화대학, 영진전문대학
고등학교	게임과학고등학교, 한국애니메이션고등학교, 울산애니원고등학교, 서울디지털고등학교, 수원공업고등학교, 미림여자고등학교

1) 고등학교 교육과정 분석

게임관련 고등학교의 경우 고등학생에게 맞는 전문 교재 및 지도 교사가 부족한 것으로 나타나고 있다. 이는 고등학교의 특성상 대학처럼 자유롭게 교재를 선택하거나 교사를 채용하기 어려운 제도적 한계에 의한 것이라고 볼 수 있다.

그래픽 관련 교육과정은 프로그램 분야에 비해서도 게임으로 특화 시키기에는 절대적으로 과목수가 부족하다고 볼 수 있다.

표 IV-9. 게임관련 고등학교 교과과정

구 분	프로그래밍 과정	그래픽디자이너 과정	기획 및 마케팅 과정
고등학교	컴퓨터프로그래밍 프로그래밍 일반 유닉스프로그래밍 C++ 프로그래밍 API 프로그래밍 DirectX 프로그래밍 Visual Basic프로그래밍 비주얼 C++ 프로그래밍 윈도우프로그래밍 3D 게임프로그래밍	게임그래픽 포토샵 일러스트레이트 3D MAX 플래시 애니메이션일반 캐릭터제작 디자인일반	게임기획 게임마케팅 게임기획 및 시나리오작성 게임산업 게임개론

2) 대학교 교육과정 분석

게임관련 전공 대학은 크게 게임프로그래머와 게임그래픽디자이너 분야로 전공을 분리할 수 있다. (표본조사 한 10개의 대학의 경우 게임그래픽디자이너 양성을 목적으로 하는 학교는 30%에 불과하고 프로그래밍 전공이 많은 실정이다) 대학의 교육과목은 크게 프로그래밍,

그래픽디자인 및 기획 분야에 대한 과목 중심으로 이루어지고 있다. 게임프로그램 과정은 10개 대학의 80% 정도가 중심 교육과정으로 교육을 진행할 정도로 중요한 부분을 차지하고 있다. 게임그래픽 과정을 중심으로 운영하는 대학은 프로그래밍 관련 과목을 거의 개설하지 않고 있다. 기획 및 마케팅과정은 교과과정에서 많은 비중을 차지하지는 않지만, 대학교의 경우 기획 및 마케팅과정이 교과과정의 하나로 개설되어 있다.

대학교 게임 관련 학과의 교과 분야별 비중 및 게임 관련 학과의 교과과정을 분석한 결과를 살펴보면 아래 표와 같다.

표 IV-10. 대학교 게임 관련 학과 과목 비중

구 분		과목 비중		
		프로그래밍 과정	그래픽디자이너 과정	기획 및 마케팅 과정
호서대학교		28%(13 과목)	13%(6 과목)	20%(9 과목)
공주대학교		9%(4 과목)	29%(13 과목)	20%(9 과목)
동명대학교		25%(12 과목)	11%(5 과목)	8%(4 과목)
영산대학교		24%(9 과목)	16%(6 과목)	18%(7 과목)
예원대학교		2%(1 과목)	40%(17 과목)	19%(8 과목)
우송대학교		20%(9 과목)	16%(7 과목)	9%(4 과목)
위덕대학교		50%(14 과목)	14%(4 과목)	7%(2 과목)
중부대학교		28%(8 과목)	10%(3 과목)	21%(6 과목)
한국산업대학교		30%(13 과목)	7%(3 과목)	15%(7 과목)
홍익 대학교	그래픽디자인 전공	2% (1 과목)	45%(19 과목)	12%(5 과목)
	게임 S/W 전공	26%(10 과목)	5%(2 과목)	10%(4 과목)

표 IV-11. 대학교 게임 관련 교과과정

구 분	프로그래밍 과정	그래픽디자이너 과정	기획 및 마케팅 과정
대학교	게임 프로그래밍 컴퓨터 프로그래밍 3D 플래시게임 C 프로그래밍 API 게임 프로그래밍 비주얼 게임프로그래밍 DirectX 프로그래밍 JAVA 프로그래밍 임베디드게임프로그래밍 게임알고리즘 연구 네트워크 프로그래밍 고급윈도우 프로그래밍 고급네트워크프로그래밍 시스템프로그래밍 보안프로그래밍 모바일 게임프로그래밍 온라인게임프로그래밍 게임엔진설계	컴퓨터그래픽스 게임그래픽디자인 평면디자인(2D) 입체디자인(3D) 게임캐릭터디자인 게임애니메이션연구 웹애니메이션연구 영상디자인 디지털애니메이션 2D게임오브젝트디자인 3D게임오브젝트디자인 게임인터페이스 디자인 게임동영상제작 컴퓨터그래픽실무 일러스트레이션 디지털페인팅연구 3D 모델링	게임학개론 게임시스템분석 온라인게임개론 게임제작개론 게임분석론 게임디자인 게임소재론 게임기획 게임디자인 게임시나리오 창작 웹컨텐츠기획 게임마케팅 게임영상기획 게임총론 게임기획 및 연출 게임스토리텔링 게임산업연구 게임프로젝트관리론

3) 전문대학 교육과정 분석

2·3년제 전문대학의 교과과정은 대학교와는 약간의 다른 점을 찾아볼 수 있다. 대학교는 프로그래밍과정과 그래픽과정이 뚜렷하게 개설과목에 차별을 보였으나, 전문대는 프로그램과정과 그래픽디자이너 과정에 비슷한 수준의 교과목이 개설된 사례가 많으며, 기획 및 마케팅에 대한 과목이 전혀 개설하지 않는 대학도 30%에 달하고 있다. 게임전공 대학은 크게 게임프로그래머와 게임그래픽디자이너 분야로 전공을 분리할 수 있으며, 4년제와 비슷한 양상을 보이고 있다. 전문대학 게임 관련 학과의 영역별 교과 비중과 게임 관련 학과의 교과과정이 아래 표에 제시되어 있다.

표 IV-12. 전문대학 게임 관련 학과 과목 비중

구 분	과목 비중		
	프로그래밍 과정	그래픽디자이너 과정	기획 및 마케팅 과정
순천청암대학	38%(11 과목)	24%(7 과목)	7%(2 과목)
대경대학	38%(10 과목)	15%(4 과목)	12%(3 과목)
영남이공대학	25%(7 과목)	14%(4 과목)	0
부산동강대학	30%(8 과목)	22%(6 과목)	0
동강대학	19%(5 과목)	18%(5 과목)	0
장안대학	31%(11 과목)	31%(11 과목)	8%(3 과목)
재활복지대학	30%(9 과목)	33%(10 과목)	10%(3 과목)
혜천대학	18%(5 과목)	18%(5 과목)	7%(2 과목)
경동정보대학	13%(5 과목)	45%(17 과목)	8%(3 과목)
계명문화대학	29%(10 과목)	6%(2 과목)	11%(7 과목)
영진전문대학	44%(17 과목)	5%(2 과목)	5%(2 과목)

표 IV-13. 전문대학 게임 관련 교과과정

구 분	프로그래밍 과정	그래픽디자이너 과정	기획 및 마케팅 과정
전문대학	비주얼베이직 C언어프로그래밍실습 자바프로그래밍 비주얼C++ C실무 C++프로그래밍 모바일 프로그래밍 C++실무 위피 유닉스프로그래밍 게임프로그래밍 온라인게임 프로그래밍 다이렉트X실습 모바일게임 프로그래밍 무선인터넷 프로그래밍 웹프로그래밍 객체지향언어실습 윈도우즈프로그래밍 클라이언트/서버 프로그래밍 인터넷프로그래밍 임베디드프로그래밍 게임엔진제작 게임제작물	게임그래픽실습 게임캐릭터 디자인 게임애니메이션 3D게임 애니메이션 컴퓨터그래픽 캐릭터및배경디자인 디지털영상편집 3D그래픽 컴퓨터응용 디자인 기초디자인 컴퓨터일러스트레이션실습 멀티미디어 디자인 영상촬영및편집 디지털이미지처리 실무 디지털애니메이션 실무 게임동영상제작 게임캐릭터제작 게임배경제작 게임드로잉기법 게임제품디자인 방송콘텐츠제작 웹애니메이션 제작	게임학개론 게임시나리오 제작 게임기획 및 제작 게임소재론 게임디자인 게임문화 게임매체기론

4) 게임업계에서 요구하는 교과과정 분석(심층면접 실시)

게임업체에서 요구하는 프로그래밍 언어는 C 및 C++ 로서 C계열 언어이다. 게임그래픽은 3D MAX와 포토샵이 가장 많이 사용하는 것으로 나타났으며, 주로 실무위주의 교육을 원하고 있다. 게임기획 분야에 설문결과는 다양한 응답이 나타났다.

표 IV-14. 게임업체에서 요구하는 교과과정

구 분	프로그래밍 과정	그래픽디자이너 과정	기획 과정
게임업계에서 요구하는 교과과정	C 프로그래밍 C++ 프로그래밍 JAVA 프로그래밍 서버프로그램 서버프로그램 고급 SKVM WIPI GNEX GVM BREW WIPI VC++ 프로그래밍 API 프로그래밍 Direct3D 프로그래밍 STL MFC 국내외 유상엔진	3D MAX 3D MAX 고급 오브젝트제작 도트작업 애니메이터 페인터 포토샵 일러스트레이터 고급 드로잉 및 원화 작업	게임스토리 기획 매뉴얼제안서 작성 밸런싱 조정 기획서 작성 방법 시나리오 게임디자인 게임기획 기초지식 창의성 개발 엑셀 중급 MS프로젝트 중급 게임디자인 레벨디자인 게임기획 기초지식

#### 5) 교육과정 비교 분석

게임관련 고교 및 대학의 교과과정과 현업에서 요구하는 교과과정이 다소 차이를 보이고 있다. 게임업계는 유상엔진 및 개발 툴 관련 교육을 원하는 반면, 현재 정규교육기관은 유상엔진 교육 및 개발 툴 관련 교육이 부족하거나 전무한 실정이다. 게임 프로그래밍은 C, C++, JAVA 정도의 사용능력을 원하고 있으며, 대학은 다양한 과목으로 관련 언어를 교육하고 있는 것으로 나타났다. 게임기획자 과정 역시 정규교육과정과 약간의 차이를 보이고 있다. 즉, 현업에서는 창의적 요소, 실질적으로 기획을 하는데 필요한 프로젝트 관리 등에 관한 교육을 필요로 하나, 정규교육과정은 게임기획과 기획서 작성방법에 초점을 맞추고 있어 현업에서 요구하는 교육과정과 차이를 보이고 있다.

게임업체에서 요구하는 교육과정보다 정규교육과정에서는 많은 교과과정을 제공하고 있으며, 게임업체에서 요구하는 상용엔진, 개발 툴 등에 대해서는 교육이 미흡한 상태이다. 정규교과과정을 끝내고 게임업체에 입사하면 바로 개발에 투입이 가능하도록 실무에 대한 교육을 하는 데는 미흡한 상태이다.

### 4.3 게임 산업 전문 인력 양성을 위한 한국형 교육시스템 제안

고등학교, 대학교 및 전문대학의 교과과정을 토대로 현업에서 요구하는 교과과정을 연구하여 교과과정을 도출하였다. 고등학교, 대학 및 업체에서 요구하는 교육과정은 어느 정도 일치하나, 각 정규과정의 요구 수준이 다르게 나타났다. 고등학교, 대학교 및 전문대학의 교과과정과 업계에서 요구하는 교과과정에서 가장 큰 차이점은 상용 툴 및 게임엔진에 대해 교육을 하기 원하나, 정규교과과정에서는 이러한 교육이 부족한 상태이다.

업계에서 요구하는 실무위주의 교육을 달성하기 위해서 고등학교, 전문대학 및 대학교의 교육목표를 다르게 설정하여, 교육의 내실화를 추구하였다. 고등학교의 교육과정으로는 게임개발에 대한 기초 지식 습득을 교육목표를 설정하여 추진할 수 있도록 모바일게임 제작을 위한 교육과정을 제

시 하였다.

컴퓨터프로그래밍 과정은 C프로그램을 기초로 모바일분야의 WIPI 등 게임엔진 관련 분야를 교과과정으로 제시한다. 모바일 분야로 특화함으로써, 그래픽 분야 교과과정은 2D와 원화 및 드로잉 작업을 그래픽분야의 교과과정으로 제시한다. 게임스토리, 게임매뉴얼 제작 및 일정관리 분야를 교육과정으로 제시하여, 모바일 게임제작을 원활히 하고자 한다. 전문대학의 교육과정으로는 게임개발 과정에 실무능력 습득을 교육목표로 설정하고 추진할 수 있도록, 모바일게임 및 PC게임 제작을 위한 교육과정을 제시 하였다.

모바일 및 PC게임 분야를 특화함으로써, 컴퓨터프로그래밍 과정은 WIPI(C, JAVA)을 기반으로 C++(MFC, API), DirectX을 교육과정으로 채택함으로써, PC게임 제작도 가능하도록 한다. 게임업계에서 요구하는 실무위주의 교육이 가능하도록 모바일 및 PC게임 관련 게임엔진도 교육과정으로 추가한다.

그래픽 분야 교과과정은 2D와 원화 및 드로잉 작업을 그래픽분야의 교과과정으로 채택하고, 3D 오브젝트 제작 및 애니메이션 작업까지도 교육과정으로 제안하므로 PC게임 제작도 가능하게 한다. 게임기획 분야로 게임스토리, 게임매뉴얼 제작, 게임기획 및 기획서 작성, 게임디자인, 프로젝트 관리뿐만 아니라 마케팅 실무를 교육과정에 적용하여 프로젝트 동아리로서 능력을 배양한다.

대학교의 교육과정으로는 독자적인 게임개발 능력의 보유를 교육목표로 설정하여 창업동아리로서의 역할을 완전히 수행할 수 있도록 하기 위한 교육과정을 제시하였다. 모바일, PC 및 온라인게임 분야 까지 특화하기 위해, 컴퓨터프로그래밍 과정은 C++(MFC, API), DirectX, 서버프로그래밍 교육과정을 채택하여 온라인게임 제작을 가능하게 한다.

게임업계의 요구사항을 반영하여 온라인게임엔진, 즉 상용화엔진의 교육을 추가하여 게임업계에서 요구하는 실무위주의 교육과정을 채택한다. 그래픽 분야 교과과정은 2D와 원화 및 드로잉 작업, 3D 오브젝트 제작뿐만 아니라 애니메이션 작업까지도 교육과정으로 제안한다.

게임기획 분야로 게임스토리, 게임매뉴얼 제작, 게임기획 및 기획서

작성, 게임디자인, 프로젝트 관리뿐만 아니라 마케팅 실무, 창의성 개발, 레벨디자인, 밸런싱 조정을 교육과정에 적용하여 창업동아리로서의 능력을 배양하고자 한다.

표 IV-15. 게임개발자를 위한 교과과정(안)

구 분	목 표	과 목		
		프로그래밍	그래픽	기획
고등학교	게임개발에 대한 기초 지식 습득 학습동아리 수준 주학습분야 : 모바일	C 언어 WIFI 등 모바일 관련 엔진	2D(도트작업) 드로잉 및 원화	게임스토리 게임매뉴얼 제작 일정관리
전문대학	게임개발 전과정 실무능력 습득 프로젝트동아리 수준 주 학습분야 : 모바일 및 PC게임	C(WIFI) C++(MFC, API) JAVA(WIFI) 모바일 및 PC게임(DirectX) 관련 엔진	2D(도트작업) 드로잉 및 원화 3D(오브젝트작업) 애니메이션	게임스토리 게임매뉴얼 제작 게임기획 및 기획서 작성 게임디자인 프로젝트 관리 마케팅 실무
대학교	독자적인 게임개발 능력 보유 창업동아리 수준 주 학습 분야: 온라인게임(모바일 및 PC게임 포함)	C++(MFC, API) DirectX 등 서버프로그램 온라인게임엔진 (상용엔진 포함)	2D(도트작업) 드로잉 및 원화 3D 중급 애니메이션	게임스토리 게임매뉴얼 제작 게임기획 및 기획서 작성 게임디자인 창의성 개발 레벨디자인 밸런싱 조정 프로젝트 관리 마케팅 실무 및 제안서작성

#### 4.4. 게임플레이어의 전문인력화 교육과정 제안

게임플레이어 양성을 위한 교육과정을 크게 고등학교와 대학으로 구분하여 교육과정을 제안한다.

##### 4.4.1 고등학교 게임플레이어에 대한 교육과정 제안

고등학교 게임플레이어에 대한 교육과정을 다음과 같이 제안한다.

- 1) 고등학교의 게임플레이어에 대해서는 건전한 아마추어 플레이어와

길드 및 클랜의 리더로 양성하는 것을 목표로 하여 기초소양, 플레이어 능력 배양을 위한 교과과정 및 대외활동분야로 나누어서 교과과정을 제안한다.

- 2) 고등학교 게임플레이어는 기초소양으로 언어예절을 교과과정으로 제안하며, 또래 친구사이에서의 상담사 역할을 할 수 있는 상담전문가 과목을 추가하여 게임이 또래 집단의 건전한 문화로 자리 잡을 수 있도록 유도한다.
- 3) 게임플레이어로서의 능력 배양을 위해 게임공략 분석 방법을 교과과정으로 배정하여 게임플레이어로서의 능력을 배양하고자 한다.
- 4) 게임플레이어의 대외활동 분야를 제안하여 게임홍보, 게임안내 및 게임테스터로서의 역할을 수행하도록 교과과정에 추가한다.

#### 4.4.2 전문대학 이상의 게임플레이어에 대한 교육과정 제안

전문대학 및 대학교를 통합하여 대학으로 명칭하며 교과과정은 아래와 같이 제안한다.

- 1) 대학 게임플레이어는 사이버 대중문화 형성 및 전파 요원을 목표로, 사이버 한류문화 전파를 활성화하는 역할을 담당한다.
- 2) 아마추어 e-Sports 요원, 길드 및 클랜의 리더, 핵심요원 양성을 위해 관련 교과과정을 추가한다.
- 3) 기초소양, 플레이어 능력 배양을 위한 교과과정 및 대외활동분야로 나누어서 교과과정을 제안한다.
- 4) 대학 게임플레이어는 기초소양으로 게임전문상담, 인터넷 예절, 사이버문화 형성을 추가하여 게임이 건전한 문화로 자리 잡을 수 있도록 한다.
- 5) 게임플레이어로서의 능력을 배양 하기위해 게임공략 분석 방법 및 정리를 교과과정으로 추가하여 게임플레이어로서의 능력을 배양 한다.
- 6) 게임플레이어의 지식을 활용하기위해 게임기획-아이디어 정리 방

법을 교과과정으로 제안한다.

- 7) 게임플레이어의 대외활동 분야를 게임홍보, 게임안내, 게임테스터, 서포터, 게임분석요원 및 GM으로서의 역할을 수행하도록 교과과정에 추가한다.

표 IV-16. 게임플레이어 관련 교과과정(안)

구 분	목 표	분 야		
		기 초 소 양	플 레 이	대 외 활 동
고등학교	건전 아마추어 플레이어 길드 및 클랜의 리더 양성	언어 예절 또래 상담	특정게임 공략 및 분석	게임홍보 및 안내 게임테스터
전문대학 및 대학교	사이버 대중문화 형성 및 전파 요원 (사이버 한류 전파 등 ) - 아마추어 e-Sports 요원 양성 - 길드 및 클랜의 리더 및 핵심요원 양성	언어예절 게임전문상담 인터넷 예절 사이버문화 형성	특정게임 공략 분석 특정게임 분석 정리 게임기획-아이디어	게임홍보 및 서포터 게임분석요원 GM 게임테스터

#### 4.4.3 한국형 게임교육시스템을 통한 인력양성 활용방안

전문대학 및 대학교를 통합하여 대학으로 명칭하며 교과과정을 통한 인력양성 방안을 다음과 같이 제시한다.

- 1) 이러한 교과과정을 통해 게임플레이어 동아리의 인력을 e-Sports 인력으로 활용한다.
  - 게임플레이어의 소양교육 및 게임업체와의 산학협력을 통해 게임플레이어 인력을 e-Sports 인력으로 활용한다.
- 2) 게임플레이어 동아리의 인력을 분석/테스트 요원으로 활용한다.
  - 게임플레이어 동아리를 게임테스터로서 활용하기 위하여 게임업체와 협력을 추진하여 분석/테스트 인력으로 활용한다.
- 3) 서포터 인력으로 활용한다.
  - 게임뿐만 아니라 상업화 홍보를 위한 인력으로 활용한다.
- 4) 게임중독 치료 상담사로 활용한다.
  - 게임에 대한 경험이 풍부한 게임플레이 동아리 출신 인원을 또

- 래상담사로 활동할 수 있도록하여 점차 전문상담사로 육성한다.
- YWCA, 청소년단체 등 청소년 전문 상담프로그램을 운영하는 기관과 협력하여 운영한다.
  - 초기에는 자원봉사활동으로 유도하여 활동케 하고, 점차 게임 분야의 전문상담사로 육성한다.

#### 4.5 단계별 첨단 영상 게임 전문 인력 양성 사업 추진계획안



그림 IV-1. 프로젝트 중심 혁신형 인재 교육 시스템 구축 사업



그림 IV-2. 자립형 게임 동아리 및 게임 벤처 설립 운영



그림 IV-3. 영상 게임 산업 특수효과 교류 협력 지원사업

## V. 결 론

온라인게임과 모바일게임을 중심으로 빠르게 성장하고 있는 국내의 게임 산업은 산업 자체의 발전과 더불어 게임 인력의 양성에 대한 요구를 빠르게 증가시켰으며, 최근 다양한 정규/비정규 교육 기관을 통하여 양적으로 매우 활발한 인력 양성이 이루어지고 있는 상황이다. 이러한 게임 전문 인력 양성 시스템의 질적 양적 확장과 발전은 최근 급속도로 성장한 국내 게임 산업 환경에서 단기간에 많은 수의 게임 관련 인력을 배출함으로써 관련 인력 수급에 적지 않은 도움을 준 것이 사실이다.

하지만, 현재의 국내 게임 전문 인력 양성 시스템에는 여러 가지 문제가 존재하는 것 또한 사실인데, 이는 향후 국내 게임 산업의 발전을 어렵게 함은 물론, 날로 치열해 저가는 경쟁 환경에서 산업계의 생존 자체를 위협하는 요소로써 작용하고 있는 상황이다.

이러한 문제점 중 교육의 구조적인 문제점을 우선 살펴보면, 일부 신규 게임 교육 기관의 불명확한 게임 교육의 목표와 부실한 교과과정 진행으로 인한 산업계 및 교육 수요층의 게임 교육에 대한 불신 확산, 융합 (convergence) 시대의 대표적인 산물이라고 할 수 있는 게임의 속성을 제대로 반영하지 못한 특정 분야 중심의 진행이라는 이른바 “절름발이식 게임 교육”의 한계 등을 지적할 수 있다.

이 외에도 교육의 질적인 부분에서는 게임 개발 경험을 보장하지 못하고 있는 이론 위주의 교과 과정, 창의성 교육이 결여된 채로 기술 및 기능, 경제성만을 강조하는 교육 내용으로 인하여 향후 게임 산업 분야의 국가 경쟁력 확보가 어려워짐은 물론, 디지털콘텐츠 분야 자체의 해외 종속을 심각하게 걱정하여야 할 상황을 야기하고 있다.

이러한 문제는 교육 환경, 학생, 교수 등 여러 방면에서 그 원인을 찾을 수 있으나 게임 선진국과 비교하여 본다면, 해외 선진국의 경우 게임 교육이 특성화된 전문 교육기관을 중심으로 하여 산업체와의 밀접한 산학협력을 통하여 산업체에서 요구하는 전문 인력을 양성하기 위한 교육을 실시하고 있지만 국내는 아직 산학협력이 활발하게 이루어지는 단계까지 이르지 못하는 점, 게임과 관련된 전문적인 통합 교육을 담당할 실무 경험이 풍부한

전문 교수 인력이 절대적으로 부족하다는 점도 포함시킬 수 있을 것이다.

이와 같은 현실적 어려움을 해결하고 산업계에서 필요로 하는 우수한 게임 전문 인력을 양성하고자 정부에서도 여러 가지 지원 정책이나 프로그램을 마련하여 실행해 왔으나, 지원 정책과 프로그램의 대부분은 장비 지원과 같은 교육 환경 개선이나 학생들을 위한 단기 특화 프로그램 마련에 그쳤었다. 이로 인해 가장 중요한 부분인 산업계에서 필요로 하는 현장 실무형 인력을 양성하기 위한 프로그램 자체나 교수 인력 확충 방안은 부족한 실정이다.

이러한 현실적 문제점은 게임 산업의 특성과 중국 및 인도를 중심으로 한 경쟁국의 비약적 성장을 고려해 볼 때, 향후 국내 게임 산업이 얼마나 발전하고 도약하느냐를 떠나 생존 자체를 판가름 하는 중요한 변수이므로 대학, 정부, 기업 모두가 적극 나서서 게임 전문 인력 양성 환경의 개선에 앞장서야만 할 것이다.

본 연구 보고서에서는 이러한 한국 게임 교육의 현실적인 문제점들을 해결하기 위해 국내의 게임 산업만이 가지는 독특한 특성을 고려하는 “한국형 게임 교육시스템”의 개념을 마련하고 이를 제안하였다.

“한국형 게임 교육 시스템”은 기본적으로 “미래 지향적 창의성 개발 교육”, “실제 개발 경험을 중시하는 실무형 교육”의 기본 방향을 토대로 한국 게임 교육의 문제점을 개선하는 방향을 제안하는데 초점이 맞춰져 있다.

우선, 현재의 개발 도구나 툴(tool) 위주, 결과물 완성도 위주의 교육시스템에서 벗어나 창의적 개발 인력 양성을 위한 기초적인 교육시스템을 개편하여야 하며, 실제 교과 과정에 있어서도 창의성 향상을 위하여 아이디어 발상 훈련이나 게임 심리 등의 교과목 개설은 물론, 독서 프로그램, 문화 체험 프로그램 등 다양한 부가 활동이 포함되어야 할 것을 제안하였다.

“한국형 게임 교육시스템”의 또 다른 기본 방향으로서는 게임 교육의 전체적인 방향성 측면에서 특정 분야에 한정된 교육보다는 제작 경험 자체가 동등하게 중시되는 실무 제작형의 종합적인 교육이 진행되어야 한다는 점을 제안하였다. 즉, 특정 분야 위주의 획일적이고 단순화된 교과과정 운영보다는 게임을 구성하는 다양한 분야 전체에 대한 종합적인 관점과 이해를

바탕으로 실제 제작 경험을 수반하는 교육 시스템을 교육생에게 제공하는 것이 필요하다는 것이다.

실무형 교육을 중시하는 “한국형 게임 교육시스템” 이 자리를 잡기 위해서는 정부 차원의 각종 지원 정책과 제도 개선도 절실히 보인다. 특히, 학습자가 실제로 구현한 게임 결과물이나 연구물을 산업계와 연계시킬 수 있는 공모전의 활성화나, 게임 교육과 관련한 최소한의 가이드라인이 되는 “표준교과과정” 을 실무형 교육시스템으로 변경하는 작업은 매우 큰 역할을 할 것으로 판단된다. 이 외에도 실제 개발 경험 및 능력에 대한 평가 방식이 취업이나 게임 자격증 등의 국가 자격 검정 제도에 반영되는 것 등은 중장기적으로 정규 및 비정규 교육 기관의 게임 교육의 방향을 바꾸는 중요한 요소가 될 것이다.

또한, 현재 상당히 부족한 실무 위주의 게임 관련 전문 교수 인력을 정부 차원에서 양성하거나 기존 교수 인력에 대한 지속적인 재교육 환경을 제공하는 것도 중장기적으로 게임 전문 인력 양성의 흐름을 건전하게 변화시키는 중요한 요소가 될 것이다. 특히, 게임 개발과 관련된 실무 경험이 부족한 기존 교수 인력에 대한 실무 능력 향상 지원을 위한 재교육 방안을 마련하고 이를 적극 시행하는 것이 현실적으로 중요하다고 보며, 나아가 이를 더욱 발전시키기 위해서는 산업체와 공동으로 노력하여 나가기 위한 산학협력 기반을 갖추는 것이 양질의 게임전문 인력 양성을 위해 가장 중요한 역할을 할 것이다. 나아가 중장기적인 관점에서 본다면, 교수 요원을 국내 게임 산업 발전을 위한 차세대 기술 개발을 위한 잠재적인 연구 요원으로 활용할 필요가 있다고 보며, 이에 대한 프로그램도 마련하여 적극 시행할 필요가 있다 할 것이다.

본 연구의 결과인 “한국형 게임 교육시스템” 을 실제로 구체화하기 위해서는 다음과 같은 추가적인 정책 방향 연구와 실천이 필요할 것으로 판단된다.

가장 우선적으로 추진되어야 하는 사업은 실제 “한국형 게임 교육시스템” 의 보급을 추진하기 위한 기반을 조성하는 사업일 것이다. 구체적으로는 교수 인력양성 사업 추진을 위한 관련 분야 기관 및 인력들의 네트워크 구축이 선행되어야만 할 것이다.

네트워크는 전국적으로 구성되어야 하며, 지역별로 지역네트워크를 구축하여 지역적 사업이 원활하게 추진될 수 있도록 하여야 할 것이다. 전국 네트워크는 한국게임산업개발원이 중심이 되어 네트워크를 형성하여 나가며, 지역네트워크는 지역적 사업을 담당할 지역게임아카데미 또는 지역 문화산업지원센터를 중심으로 네트워크를 구축할 필요가 있다. 또한 지역에는 지역 단위의 포럼을 결성하여 지역 대학 게임분야 교수, 관련 산업체, 특화교육기관 등의 상호 교류가 이루어지도록 하며, 이를 통해 산학협력 및 인력 양성 프로그램 개발 등이 자율적으로 수행될 수 있는 기반을 마련하여 나가야 할 것이다. 또한, 이러한 네트워크를 지원할 수 있는 웹사이트의 구축, 지원과 함께 이의 효율적인 운영도 필요하다. 아울러 중장기적으로는 외국 선진 게임 교육기관과의 교류가 필요하므로 해외 네트워크도 같이 구축할 필요가 있다.

둘째, “한국형 게임 교육시스템”을 효율적으로 보급하기 위한 조직이나 기관이 필요한데, 이를 담당하기 위한 기반 시설로서 전문 연구 기관이 구축되어야 할 것이다.

전국 단위 또는 지역 단위의 교육 네트워크가 구축되면, 우선 수요가 많을 것으로 예상되는 분야부터 교육 프로그램을 실행하여 나가야 할 것이다. 예를 들어, 실무 분야 교수인력 양성을 위한 단기 연수 프로그램과 같은 프로그램을 지역 아카데미를 중심으로 개설하여 운영하여 나가는 것이다. 이와 같은 시범 교육 프로그램을 단계적으로 수행하여 나가면서 점차 프로젝트 교육, 특화 교육, 실무 체험 교육 등과 같은 산업계 요구 중심의 프로그램을 마련하고 시행하여 나가야 하는데 전문 연구 기관은 이에 대한 체계적인 연구를 지원하여야 할 것이다.

셋째, 실무 능력 향상을 위한 프로그램의 수행을 위해서는 산학 협력이 필요하므로 이 분야에 중심을 두어 교육 프로그램을 운영할 필요가 있다. 물론, 산학협력 프로그램은 학교가 주가 되는 것이 아니라 기업이 주가 되어 시행되도록 함으로써 기업이 갖는 앞선 기술이나 경험이 교수요원에게 전수될 수 있도록 하여 나가야 할 것이다. 산학 협력 프로그램의 구체적인 종류로는 프로젝트 교육, 특화 교육, 실무 체험 교육, 산학 교류, 산학 공동 개발 등이 있으나 이의 시행은 각 지역의 기업이나 학교의 사정을 감안하여

계획을 수립하고 시행할 필요가 있다.

아울러 산학 협력을 위한 실무 능력 향상 교육 프로그램 운영에 있어 대상이 되는 기업을 국내 기업으로만 한정하기 보다는 외국 기업과의 산학 협력이 이루어지도록 다양한 지원 프로그램을 마련하여 시행할 필요가 있다. 또한 외국 선진 교육 기관의 교수자나 게임 개발자의 교환이나 교류 프로그램의 시행을 통해 해외의 교수 요원을 국내에서 활용할 수 있도록 제도적으로 지원하여 나갈 필요가 있다.

넷째, “한국형 게임 교육시스템”의 단기적 프로그램 시행에만 치중하기 보다는 중장기적으로는 교육 프로그램의 중심을 연구 개발 분야로 돌려 연구 개발 분야의 인력을 위한 중장기 연수 프로그램을 운영하여 나갈 필요가 있다. 연구 인력 양성을 위한 프로그램은 내용 및 국가적 우선 순위에 따라 지역 단위 및 전국 단위 네트워크를 활용하여 확장시켜야 할 것이다.

우선 국내 연구 기반 활성화를 위해 학술 세미나, 컨퍼런스, 산학 공동 연구 등의 프로그램을 마련하여 시행할 필요가 있으며, 나아가 해외 교류 및 해외 공동 연구로 이어질 수 있도록 지속적으로 발전시켜 나갈 필요가 있다. 아울러 연구기반의 인력양성을 촉진시키기 위해서는 대학 또는 지역별로 연구소/산학협력 지원 센터를 설립하여 운영하여 나감으로써 제도적으로 연구기반을 마련하여 줄 필요가 있다.

마지막으로 국내 게임 교육 기관의 현황은 모바일이나 온라인게임 등 특정 분야에 치우치는 경향이 강해 기업체에서 필요로 하는 다양한 분야의 인력이 양성되지 못하는 실정이므로 이를 타개하기 위한 교육시스템에 대한 준비가 필요하다는 점을 들 수 있다. 이러한 교육 분야는 교육 수요가 많지 않을 것이라는 특수성을 감안하여 지역 단위 보다는 고려하여 전국 단위의 네트워크를 활용하는 것이 바람직할 것으로 보인다. 우선적으로 유비쿼터스 분야, 무선(제어) 분야, VR 분야, 아케이드 분야, 콘솔 분야, 사운드 분야 등을 교육 분야로서 설정할 필요가 있다고 본다.

이러한 특수 분야의 교육 활성화를 위해서는 그 무엇보다도 학교-기업 간 교류가 필요한 분야이므로 이 분야에 대한 프로그램의 개발과 지원이 필요하다 할 것이다.

앞에서 제안한 “한국형 게임 교육시스템” 이 효율적으로 정착되기 위해서는 우선 현재 국내 게임 분야 교육에 관한 정책 수립 및 운영을 정부 및 각 기업의 중요한 하나의 사업 분야로서 명확히 설정하여 이에 대한 정책을 수립하고 추진할 필요가 있다. 또한 그렇게 하기 위해서는 이 분야의 사업을 담당할 전담 조직의 신설이 필요하며 나아가 사업 추진을 필요한 재원의 확보가 우선적으로 이루어져야 할 것이다.

또한, “한국형 게임 교육시스템” 의 빠른 성과를 위해서는 지방 거점도시를 중심으로 기 설치된 지역 게임아카데미를 전국 대학 게임 교수인력 양성 사업을 위한 교육 운영기반으로 적극 활용하면 기존 수도권 중심의 국내 게임교육 운영 기반이 지방에까지 확대하여 시행하게 되는 것이므로 그만큼 빠른 시간 내에 최소한의 비용을 활용하여 국내의 게임 교육 여건을 개선시킬 수 있을 것이다. 이는 기존의 게임아카데미의 역할이 수도권 중심에서 지역까지 확대되어 시행되게 되는 부수적인 효과를 거둘 수 있으므로, 기존의 시설 및 환경을 활용해 국내 게임 교육 활성화 및 산업 발전에 기여할 수 있는 계기로 활용될 수 있을 것으로 본다.

또한, 관련 학회나 전문 교육 기관을 상호 연결하고 협력하여 교육을 시행하여 나감으로써 장점을 활용하여 나갈 수 있는 기반을 마련하여 나가는 것도 매우 바람직한 방향으로 판단된다. 보다 효율적이고 효과가 높은 교육이 이루어지도록 하기 위해서 국내의 다양한 전문 기관이 상호 협력하고 교육에 필요한 요소들을 활용하여 나갈 필요가 있는 것이다.

## [ 참고 문헌 ]

1. 문화관광부 (2004), "2004 문화산업백서".
2. 사단법인 기독교윤리실천운동 (2003), 제 3 회 온라인 게임문화 포럼 "청소년의 온라인 결재 문제점과 대책 방안" 토론회 자료집.
3. 이재현 (2001), "인터넷과 온라인 게임", 커뮤니케이션북스.
4. 정보통신부 (2003), "통신방송융합에 대비한 정책 및 법·제도 정비방향".
5. 정보통신윤리위원회 (2005), "정보통신윤리위원회10년사".
6. 정보통신정책연구원 (2000), "정보통신기술이 콘텐츠산업에 미치는 파급효과 분석", 연구보고 00-17.
7. 정보통신정책연구원 (2002), "통신.방송 융합에 따른 법.제도 개선 및 산업 정책 연구", 연구보고 02-38.
8. 청소년보호위원회 (2001), "컴퓨터게임이 청소년에게 미치는 영향에 관한 연구".
9. 최유한 (2002), "컴퓨터 게임의 이해", 문화과학사.한국게임산업개발원 (2005), "2005 대한민국 게임백서".
10. 한국게임산업개발원 (2003), "게임 산업 진흥 중장기 계획 (2003년~2007년)", 2003년 12월 대통령 보고 자료.
11. 한국게임산업개발원 (2003), "교육용 게임 시장 및 개발 전략".
12. 한국게임개발자협회 (2002), "한국 게임개발자 컨퍼런스 2002".
13. 한국전산원 (2001), "차세대 인터넷 응용 서비스/기술 연계도 및 서비스 모델 연구", 국가정보화센터 공동 연구 보고서.
14. Andrew Rollings & Dave Moris / 최경원 역 (2001), "Game Architecture and Design", 제우미디어.
15. ETRI 컴퓨터소프트웨어연구소 (2003), "온라인게임 기술의 현황 및 전망", 게임동아리지원사업 관련 특강 자료.
16. Rusel DeMaria & Johnny L.Wilson / 송기범 역 (2003), "The Illustrated History of Electronic Games" .
17. 정보통신정책연구원, 유지연, 정보통신정책 제13호권 16호 지식기반사회에서의 e-learning 현황 및 전망, 2005
18. 한국소프트웨어진흥원(KIPA), 미국 온라인게임 시장에 대한 전략적 접근:

- 과학적 사용자 분석과 진출 전략, 2005. 4
19. Mobile Content Forum, 일본 모바일시장 동향 2005
  20. Korean Click, 국내 온라인포털 사이트 이용행태 분석, 2004. 6
  21. Laramee 공저, 김지인 외 역, 해외 게임 비즈니스 가이드: 미주, 유럽지역, 정보문화사, 2005. 1
  22. Asian Daily, 2005, 9
  23. ACTeN, E-Content Report 6, 2004
  24. AEN, Survey Research on e-Learning in Asian Countries-Fiscal Year 2002, 2003
  25. AEN, Survey Research on e-Learning in Asian Countries-Fiscal Year 2002, 2003
  26. Alan Bergstein, "Introducing Health It World Technology for Healthcare-2004 media Kit", health-itworld.com, 2004
  27. Angelina Benedetti, "eBooks at kcls.org", King County Library System eBooks in the public Library Conference, 2004.3
  28. ANKOS, "ProQuest 5000", ProQuest Information and Learning Annual Meeting, 2004.6
  29. Baird Research, Communications Services, Wireless Data, 2005.8
  30. Bank of America Securities, Rollin' Over? 2005-2007 Video Game Industry Outlook, 2005. 3
  31. Bank of America Securities, Video Game Industry Overview: Important Topics for the Video Game Industry in 2005, 2005. 2
  32. Baskerville, Global Mobile Forecasts to 2010, 2005
  33. Bear Sterns, Asian Online Gaming, 2005. 1
  34. Bonnie Hawkwood, "New Sources of eBook Content for Library-Gale Virtual Reference Library", Thomson & Gale, 2004.3
  35. Book Conference, "Book Conference Draft Program", Book Conference.com, 2004.7
  36. BuddeComm, Mobile Communications-Mobile Data, 2005.8
  37. Budde Com, Europe (Western) Telecoms, Mobile & Broadband Overview

- and Analysis 2004, 2005. 3
38. CIBC World Markets, Shanda Entertainment, 2005. 11
  39. CSFB, China Internet Sector, 2005. 8
  40. Centrosim, Tiscali, 2005. 11. 9
  41. City Group Smith Barney, EUROPEAN TELECOMS - CHOPPY WATERS, BUT WELL WORTH THE RIDE, 2005. 4
  42. CityGroup, Latin America: Telecom Services and Media Review, 2005. 10
  43. CSFB, Asian Daily, 2005. 9 ~ 10
  44. CSFB, Baidu, 2005. 9
  45. DATAMONITOR, Digital Rights Management , 2005
  46. DCITA, A Guide to Digital Rights Management, 2005
  47. Deutsche Bank, CHINA INTERNET SECTOR: THE FOUR NET EMPEROR, 2005. 8
  48. Deutsche Bank, CHINA INTERNET WIRELESS VAS, 2005. 10
  49. Deutsche Bank, WVAS, ads and travel strong, games lackluster, 2005.10
  50. DFC Intelligence, The Online Game Outlook 2004, 2004. 8
  51. Digital Vector, Global Animation Industry Strategy, Trends and Opportunities, 2005
  52. Double Click, Eruope Online Ad, 2005. 9
  53. EIAA, Mediascope Europe Study 2005: Executive Summary, 2005. 10
  54. EITO, European Information Technology Observatory 2005
  55. Equity Research, Internet Portal & Commerce, 2005. 7
  56. Frost & Sullivan, U.S. E-Learning System Market, 2004
  57. Gartner, Akamai's Deal to buy speeder will Narrow the CDN Market, 2005
  58. Gartner, Apple Governance Principles to Improve Content Management, 2005
  59. Gartner, Application Delivery Network Go Beyond Traditional content Distribution Network, 2005

60. Gartner, Beyond content Delivery : The Evolution of ECDNs , 2003
61. Gartner, Content and Document Management Market Shows Growth, 2004
62. Gartner Dataquest, "Market Focus- Portable Digital Audio Players, Worldwide, 2004-2010", 2005
63. Gartner, 2005 E-Learning Client Issues, 2005
64. Gartner, Forecast: Content and Document Management Software, Worldwide, 2000-2008, 2004
65. Gartner, Forecast : Enterprise Content Management, Worldwide, 2005-2009, 2005
66. Global Mobile, 2005.5, 2005.6, 2005.7
67. Goldman Sachs, Sixth Annual Goldman Sachs Global Internet Conference: Internet Roadmap, 2005. 5
68. Hoover's, [www.hoovers.com](http://www.hoovers.com)
69. IDC, Asia's Content Management (CMS) Market Growth, 2004
70. IDC, worldwide and U.S corporate elearning forecast 2004-2008, 2004
71. IDC, Quantifying the Business Benefits of enterprise Content Delivery Networks, 2005
72. Informa Telecom and Media, Global Mobile forecast to 2010
73. iResearch, China Online Education Research Report 2004, 2005
74. iResearch, [www.iresearch.com](http://www.iresearch.com)
75. IRN Research, [www.irn-research.com](http://www.irn-research.com)
76. IT media, [www.itmedia.co.jp](http://www.itmedia.co.jp)
77. JAPA(JAPAN AS POPCULTURE ASSOCIATION), [www.ppp.am](http://www.ppp.am)
78. Joe Gonnella, "How Standard Drive Our Business", Barnes&Noble Inc, 2004.8
79. Marche du film, World film market trend 2005
80. Meta group, Mobile Digital Rights Management: Critical for Vendors, But Users?, 2004. 10
81. MGAIN, Mobile Entertainment Industry and Culture, 2005

82. Moors & Cabot, 2Q05 US Wireless Data and Content Review, 2005.8
83. Morgan Stanley, China Internet, 2005.9
84. MRG ,IPTV Global Forecast 2005 to 2008, 2005
85. OECD, Digital Broadband Content: Mobile Content-New Content for New Platforms, 2005 5.3
86. OECD, "Digital Broadband Content: Music", Working Party on the Information Economy, 2005. 6. 8
87. Ovum, Wireless Regional Forecasts, 2005.9
88. Paul Budde, Global-Mobile-Subscriber Statistics and Forecasts, 2005.8
89. Park Associates, Digital Entertainment in the Home: The Home Computer & Connectivity, 2004
90. PEW INTERNET, PEW INTERNET & AMERICAN LIFE PROTECT, August, 2005
91. Pro-music, [www.pro-music.org](http://www.pro-music.org)
92. PWC, Global Entertainment and Media Outlook 2005-2009, 2005
93. Research and Consultancy Outsourcing Service, Digital Rights Management-State of the Industry, 2005
94. Simba Information, "Business Information Markets : 2005-2006, 2005
95. Tamara DeSouza, Candy Hon, Jenny Kim, Eric Lee, Charlene Leong, Ryerson University, " eBook- An Uphill Battle for Acceptance", The Electronic Document Systems Foundation, 2004. 5
96. Thomson, IPTV The need for standard, International Broadcasting convention 2005
97. University of California, Berkeley, The Role of US Higher Education in the Global E-Learning Market, 2002
98. WestLB, The third screen, 2005.10
99. World Economic Forum, Global Competitiveness Report 2004-2005, 2004
100. Yankee Group, Mobile Entertainment Opportunities and Challenges, 2005
101. ALIC, 日本eラーニング白書 2005/2006, 2005

102. DCAJ, デジタルコンテンツ白書2005, 2005.8
103. Dentsu, 情報メディア白書2005, 2004.12
104. JETRO, 日本の映画産業の動向、2005.5
105. JETRO, 日本のアニメーション産業の動向、2005.7
106. JETRO, 日本の音楽産業の動向, 2005.7
107. MCF(Mobile Content Forum), 日本モバイルコンテンツ白書 2005, 2005.7
108. NRI(野村総合経済研究所), これから情報・通信市場で何が起こるのか—IT市場ナビゲーター2005版, 2005.3
109. 日本経済産業省, コンテンツプロデュース機能の基盤強化に関する調査研究, 2004. 2
110. 日本経済産業省, アニメーション産業の現在と課題, 2004.9
111. 日本経済産業省, 音楽産業の課題と現状について, 2002.3
112. 日本経済産業省, 日本デジタル産業白書 2005, 2005
113. 日本レコード協会, 日本のレコード産業 2005、2005.3
114. 日本データベース振興センター, データベースサービス実態調査, 2005.6
115. 第一生命経済研究所 携帯音楽プレーヤ登場で注目を集めるHDD市場, 2005.4
116. 野村総合研究所(NRI),これから情報通信市場で何が起るのか"IT市場ナビゲーター2005", 2005.3
117. インプレス、ケータイ白書2005、2004. 12
118. インプレス、インターネット白書2005、2005.6
119. 新桂网, 正版网络音乐下载举步维艰 合法网站仅20家2005.6.30
120. 中国博客, 在线音乐发展还需过三关, 2005.6.30

## [부록] 카네기멜론대 ETC 교육 과정 분석

우 송 대 학 교    교수 이 창 조  
청강문화산업대학    교수 김 영 진  
서 남 대 학 교    교수 이 광 재

### 1. 보고서의 개요

가. 본 보고서의 작성 목적: 본 보고서는 2005년 7월부터 8월까지 한국게임산업개발원의 위탁에 따라 미국 피츠버그 현지에서 4주에 걸쳐 진행된 CMU ETC 특별 교육 과정을 통하여 확인된 Panda3D 의 게임 개발 및 교육 도구로써의 특징 및 장단점에 대한 내용을 실제 프로젝트 진행을 경험한 관점에서 정리하는 것을 목적으로 작성되었음

나. 보고서 작성자 및 관련 문의 : 이 창 조 (cjlee@wsu.ac.kr)  
김 영 진 ([indytel@chol.com](mailto:indytel@chol.com))  
이 광 재 (kjlee@seonam.ac.kr)

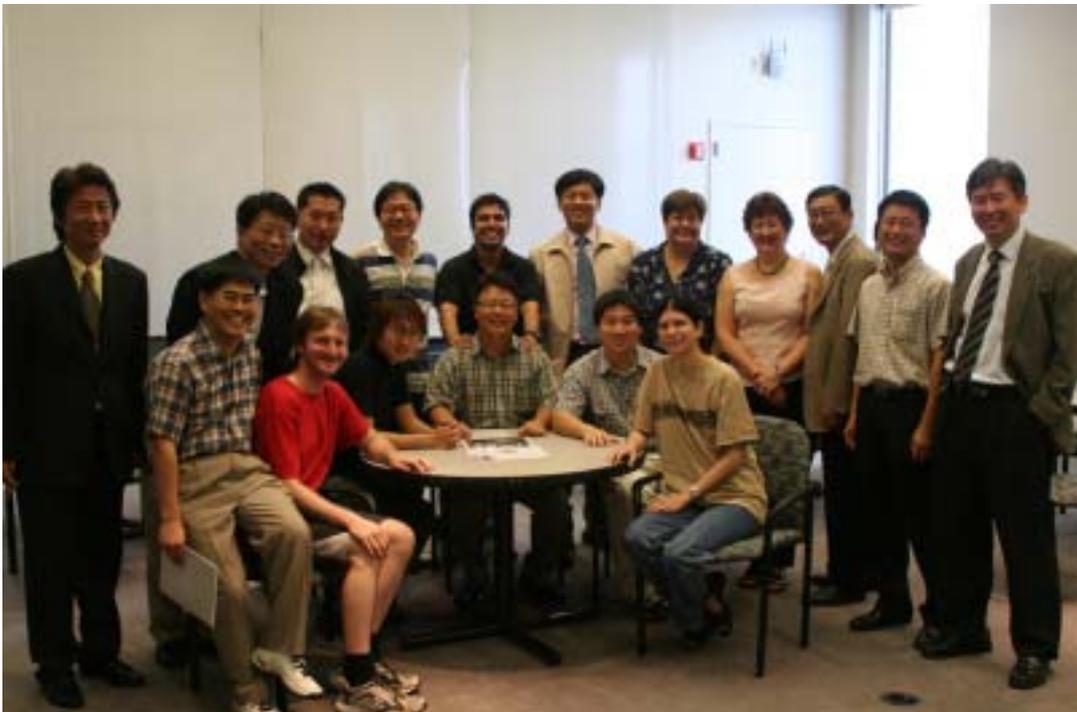


## 2. CMU ETC 단기 연수 과정 체험 개요

가. 참여 과정 명칭 : “카네기멜론대 게임분야 교수인력 해외연수”

나. 연수기관 및 기간

- 연수기관 : 카네기멜론대학 부설 Entertainment Technology Center
- 연수기간 : 총 4주간 (2005.7.18 ~8.12)
- 국내 담당 기관 : 문화관광부 산하 한국게임산업개발원



다. 해당 과정 개설 목적

- 글로벌 게임 전문교수 인력양성 위한 해외 우수대학과 연수프로그램을 통해 선진기술 습득 및 글로벌 마인드 고취
- 국내 게임 전문교수 인력양성 체계 구축 및 신진 교수인력 발굴 및 육성의 확대

라. 신청자격 및 선발방법

- 지원 대상

- . 게임분야 해외 연수 후, 전파교육이 가능한 게임관련 교육 종사자
- . 국내 게임관련 대학 교수 및 강사
- . 신진 교수인력 (게임관련 석, 박사/교육 분야에 진출희망 개발자)
- 선발방법
  - . 1차 서류전형 : 지원서 온라인 접수
  - . 2차 영어 인터뷰 및 심층 면접 (서류전형 통과자 대상)



마. 해당 연수 지원 및 진행 과정

- 지원 과정
  - . 6/23 (목) 기획실 협의 후 6/24 (금) 지원서 온라인 접수
  - . 6/30 (목) 영어 인터뷰 및 심층 면접 실시
  - . 7/7 (목) 합격자 발표 및 오리엔테이션
- 실제 연수 진행 과정
  - . 7/16 (토) 출국
  - . 7/18 (월) ~ 8/12 (금) 연수 과정 진행
  - . 8/15 (월) 입국

### 3. CMU ETC 의 게임 관련 단기 연수의 교육 구성

가. 개 요 : 금번 단기 연수 과정은 CMU ETC 가 진행하는 과목 전체를 리뷰할만한 시간적인 여유가 없는 4주간이라는 짧은 시간동안 이루어 졌다. 이러한 문제를 극복하기 위해 CMU ETC 는 한국게임산업개발원 (KGDI) 를 위한 특별 과정을 준비하고 있었는데, 이는 오전 강의 와 오후 프로젝트 및 과제 부여를 동시에 진행하는 병행 방식으로 진행되었다.

나. KGDI 를 위한 특별 과정 커리큘럼의 구성 : CMU ETC 측에서는 자신 들의 교과과정 중 중요한 부분을 하루 한 두 시간 정도의 특강 형식으로 요약해 소개하는 방식으로 연수를 진행하였는데 연수 실시 이전에 제안된 커리큘럼의 내용은 다음의 표와 같았다.

[표 6] CMU ETC 의 KGDI 특별 과정 커리큘럼

Week	Lectures	Required Readings / Assignments	Project Description
Week 1	1. Western Storytelling Overview (Games & Film) 2. Characters in Games & Film 3. What is an 'Interactive' Story? Guest lectures	1. Assignments & readings 2. There will be assignments and readings, prior to the lectures .	Project #1: This is a one-week test project assignment: Build a five-minute Interactive Experience that tells a prototypical story, including one or more characters, and that embodies Indirect Control.
	The theory behind this week is to understand ETC's approach, through lectures, readings, and the project assignment, and to become acquainted with a common vocabulary to serve the remainder of the program. Key topics include principles of storytelling, including a review of traditional climactic plot structure and diverse approaches to episodic plot development, principles for creating characters within stories, and principles of the design of interactive systems (for		

	<p>indirect control of the game player). The assignment (producing a five-minute interactive experience within the first week) allows the ETC faculty and participant faculty to interact on the key topics of story, characters, and indirect control of the game player, to lay a foundation of understanding for the fundamentals of ETC's approach. At the same time, this one-week test assignment also allows the ETC faculty to guide the participant faculty in fast cycle project management and production.</p>		
Week 2	<p>1. Telling a Story Through Visuals 2. Story Structure 3. Emotional Engineering</p> <p>Guest lectures</p>	<p>1. Assignments &amp; readings</p>	<p>Project #2: This project is 3-week project: This project will be both story &amp; game design driven.</p>
	<p>This week enhances understanding of game design and the psychology of engaging, interactive games. Key topics include visuals and key aspects of computer graphics, the unfolding of the structure of a story at multiple levels, the dynamics of level design, and principles of emotional engineering. Also, in the first week of project assignment #2, participants are guided to plan, and reflect upon, the aspects of teamwork, project management, and production of a three-week project. Participants will also be exposed to specific initiatives within the ETC aimed at enhancing the verisimilitude of NPCs (Non-Playing Characters) in games through creation of algorithms based upon Improvisational Acting exercises and techniques. This research within the ETC was highlighted at the Game Developer's Conference in San Francisco to significant industry interest and positive reaction.</p>		
Week 3	<p>1. Game Design as an Art 2. Game Genres 3. Triangularity &amp; Indirect Control</p> <p>Guest lectures</p>	<p>1. Assignments &amp; readings</p>	<p>Project #2: Continued.</p>
	<p>Week 3 deepens understanding of game design, emphasizing it as a form of art, rather than science, but one with its own internal and proven structure and dynamics. We will also review various genres of games analyzing their unique properties and shared similarities. It</p>		

	<p>will also be a time for a deeper review of emotional engineering, triangulation and indirect control of the game player. In the second week of project assignment #2, the ETC faculty will evaluate each participant team on aspects of team management and production, compared to original plans, allowing adaptation to meet the objectives of time, resources, and quality, within the limits of the resources and the time schedule.</p>		
	<p>1. Types of Game Players 2. Prototyping 3. Game Balance</p> <p>Guest lectures</p>	<p>1. Assignments &amp; readings</p>	<p>Project #2: Continued, completed, and Final Presentations and Critiques.</p>
Week 4	<p>Week 4 includes a review of players, software production methods, and the role of choice reduction within interactive games. First is a review of prototypical game players and the key elements of success in addressing their needs. This will contain a retrospective view of the key software development technique used in game production, namely, rapid prototyping, and the reasons that rapid prototyping is important, as compared to other methods of producing games. Finally, we review the issues of "game balance" or "play balance" in the context of interactive games in which the user is given latitude in game play, while the game has been designed to limit or reduce the number of choices that the user may choose. In the final week of project assignment #2, the team must realize the results of their efforts, by producing a game that has been designed around a story and is indirectly controlled by the principles of game design. Final critiques of the results will provide an opportunity for reflection on the ETC approach, their own experience, their degree of "success" and "lessons learned." This guarantees understanding of at least some of the complexities of entertainment technology and the principles of "entertainment technology education" as designed and practiced in the graduate program at Carnegie Mellon's Entertainment Technology Center.</p>		

#### 4. CMU ETC 의 실제 수업 진행 내용 분석

가. 개요 : 앞서 살펴본 표에서 제안된 내용을 기반으로 구체적으로 진행된 강의는 커리큘럼 개요 소개, 프로그래밍, 공학 일반, 그래픽 기술, 사운드 개요 및 기술, 마케팅, 프로젝트 매니지먼트, 팀원 관리 등의 각 분야별로 진행되었다. 특히, 각 수업은 학기 중에 직접 수업을 진행하는 해당 교수가 교수진의 입장에서 해당 과목의 필요성을 강조하고 수업의 핵심 내용을 실제로 강의하는 한편, 이와 관련된 토의가 병행되는 방식으로 진행되었다. 금번 연수에 참가한 모든 대상자가 연구자를 포함해 모두 교수 요원이어서 해당 토론은 게임 개발자 양성을 위한 해당 교과목의 필요성과 효과적인 교수 방법에 대한 내용에 집중되었다. 본 장에서는 실제로 진행된 각각의 강의 중 현재 본 교에서 진행되고 있는 강의에 크게 도움이 될 만한 강의의 개요와 내용을 기술하고자 하였다. 다만, CMU ETC 교육의 핵심이라 할 수 있는 프로젝트 수업에 대한 내용은 이번 장에서 다루지 않고 다음 장에서 소개하였다.

[표 2] KGDI 단기 연수를 위한 실제 수업의 진행 시간표

주 차	월	화	수	목	금
1주차	Opening Discussion	US Game Industry/Economics	Philosophy of BVW	Game Design	SCS Orientation-Collaborating with CMU
2주차	ETC overview	Storytelling in Games	Storytelling in Games II	Storytelling in Games III	Sound Technology
3주차	Alice	Game Processes & Methodologies	Project Mgmt in Practice	Intro to the Role of the Producer	Janet Murray
4주차	Project Work	Improv	Visual Language	Visual Metaphor	Ending Discussion

#### 나. Project Management in Practice (강사 : Dan Irish)

##### (1) 수업의 개요

- 프로젝트 관리의 핵심인 작업 분석 및 일정 수립에 대한 원칙

소개

(2) 실제 수업 내용

- 게임 제작에서의 프로젝트관리 토픽
  - . 모든 프로젝트 관리자들에게 프로젝트 관리는 매우 힘든 업무
  - . 게임개발의 일정수립은 점차적으로 복잡함
- 작업들의 분할
  - . 각 작업들은 상세하게 분할하고 일정상에 자세히 나타나야 함
  - . 단위 작업공정(process)들을 정의해야 함
  - . WBS (Work Breakdown System) 포함 공정들의 목록화 필요
  - . 각 분할된 작업들은 Best, Worst, Most Likely로 구분하여 추정함
- MS 프로젝트의 사용
  - . 매일 작업관리의 결과를 MS 프로젝트 파일에 갱신함
  - . 데이터 양식은 유연하고, 방어적이고 과학적이어야 함
  - . 프로젝트 관리 작업의 우선순위를 정의함
  - . 관리의 초점은 주공정경로의 작업들을 성공적으로 완료하는 것임
- 의존성과 임시적대용체(Placeholders)
  - . 의존성은 프로젝트의 전체일정을 늦추게 하는 주요 원인
  - . 가능한 작업들간에 의존성을 낮추어야 함
  - . 임시적대용체는 각 작업의 최종결과물 대신에 사용할 수 있는 것
    - . 특히 아트 작업 결과물에 대해서 효과적으로 사용할 수 있음
- 마일스톤 승인 테스트(MAT: Milestone Acceptance Tests)
  - . 마일스톤의 승인 테스트는 실험기준을 마련해서 실행해야 함
  - . 기획단계에서 그 내역을 결정해야 함
    - (예: 테스트방법, 합격기준, 평가지수 등)

다. Game Production Processes & Methodologies (강사 : Dan Irish)

(1) 수업의 개요

- 대표적인 소프트웨어 제작과정들 소개
- 게임개발에 적절한 소프트웨어 제작과정 소개 및 게임제작사례 소개
- 소프트웨어 팩토리(Software Factory) 효율성(Efficiencies) 및 사례

(2) 실제 수업 내용

- 게임개발에 있어서의 공정(process)과 방법들(methodologies)
  - . 개발프로젝트에 맞는 것을 선택, 사용하는 것이 중요함
- 대표적인 소프트웨어 제작과정
  - . Code-like-hell, fix-like-hell : 제작계획 없이 곧바로 개발하거나 제작계획이 개발단계에서 무시되는 개발 방식. 가장 공통적이고 오랫동안 적용되어온 개발방식
  - . Increments to Completion : 유한개의 콤팩트한 증가분(increment)들로 나누어서 따로 개발한 후 통합하여 소프트웨어를 제작하는 방식. 게임은 개발 초기에 플레이 가능하게 만들어짐
  - . Cascade : 전체 팀이 모두 개발과정의 한 단계에 집중적으로 참여하여 완료하고 다시 다음단계를 동일한 방법으로 수행하는 방법, 게임개발에는 부적합
  - . Iterate-until-you-drop : 가장 유연한 제작 방법. 객체지향 소프트웨어 개발에 적합. 성공적인 제작을 위해 적절한 툴과 기술적 기획이 필수적 요구
- 게임개발에 적절한 소프트웨어 제작과정 소개 및 게임제작사례 소개
  - . Standard development model : 비디오게임 개발에 가장 공통적인 적용, 개발 후반 단계에서 리스크들이 발생. 오랫동안 유일한 방법으로 받아들여졌음
  - . Forward loading development model : 상용적으로 성공하는

게임을 만드는 것을 목표로 유지. 프로토타입을 확실하게 구현.  
비용이 지출되는 만큼 리스크는 감소. 리스크가 많은 프로젝트는 초기에 중단

- Agile Project Management 소개 및 게임제작사례 소개
  - . Jim Highsmith에 의해 소개
  - . “Iterate-until-you-drop”에 민첩성을 추가한 방법
  - . 조직화되고 훈련된 팀에는 훌륭한 방법
  - . 사례연구 : RocketBowl 등
- 주공정경로(critical path) 기획(planning) 및 사례연구
  - . 주공정경로는 전체 일정에 직접적인 영향을 주는 경로
  - . 집중적인 관리가 필요한 경로
  - . 작업활동들 사이의 의존성을 최소화 해야 함
  - . 모든 팀원들이 항상 일정을 알 수 있게 함
- 긴급사태 대비 계획(Contingency Planning)
  - . 초과작업에 대한 고려를 예산상에서 반영되어야 함
  - . 휴가 및 공유일 등에 대한 상황을 예측하고 계획에 반영해야 함
  - . 테스트를 고려한 일정들도 계획에 반영해야 함
- 소프트웨어 팩토리(Software Factory) 효율성(Efficiencies)
  - . 틀, 프로그램 코드, 문서들을 재사용
  - . 개발은 개발목표를 정확히 달성할 수 있도록 엄격히 통제 필요
- 조직화와 성공적인 공정들
  - . 사용법이 간결하고 쉽고 효과적임
  - . 관리상 나타난 문제들을 명료화하도록 관련자들에게 문의할 것
  - . 다른 사람들의 필요성을 예상할 것
- 리스크 관리
  - . 모든 가능한 리스크들을 찾고 목록화하여 관리해야 함
  - . 각 리스크에 관리 담당자를 정하여 책임을 부여함

- . 만약 리스크가 너무 높다면 프로젝트를 중단해야 함

## 라. Basic Storytelling (강사 : Chris Klug)

### (1) 수업의 개요

- 스토리텔링의 기본 요소 및 개념 소개
- 사례 분석을 통한 스토리텔링의 중요성 강조

### (2) 실제 수업 내용

- 스토리의 요소 : 발단(Beginning), 전개(Middle), 결말(End)
- 스토리란 무엇인가?
  - . 스토리는 갈등을 통해 감정적으로 변화를 일으켜야함
  - . 스토리는 기대와 결과사이의 중간매개
  - . 스토리는 시작순간의 감정의 정도(value)와는 다른 결말로 종료
  - . 스토리는 관중(audience)에게 하는 약속
  - . 스토리는 긍정적으로 생각한다면(hopefully) 당신의 관점(view)
- 캐릭터(Character)
  - . 캐릭터는 이야기를 풀어나가는 체계(syntax)
  - . 인간이 받는 압박하에서 만들어지는 선택(행동)으로 차별화
- 비트(Beats), 장면(Scenes), 시퀀스(Sequences)
  - . 비트는 액션/리액션안에서 행동의 주고받음(exchange)으로 정의
  - . 시퀀스는 어떤 이전의 장면보다도 훨씬 큰 충격(impact)이 최고조에 달하는 장면들(2-5)의 연속.
- A-story; B-story
  - . A-스토리는 줄거리(Plot)를 의미
  - . B-스토리는 영웅의 내적인 여정(Hero's Inner Journey)을 의미. 주관적 스토리(subjective story)라고 불림
- 호기심을 일으키는 우발적 사건(Inciting Incident)
- 스토리 체크리스트(Story checklist):

- 질문들의 연속(Series of Questions)

마. Improvisational Acting (강사 : Brenda Harger)

(1) 수업의 개요

- 즉흥 연기 기술 교육을 통한 게임 디자인의 향상 도모
- 실시간 이야기와 장면 연출 실습을 통해 창의성을 개발

(2) 실제 수업 내용

- 즉흥 연기 연습 하기의 기본 원칙
  - . 팀 구성( Team building)
  - . 자발성 연습 (Exercising Spontaneity)
  - . 청취 능력 개발(Developing Listening skills)
  - . 무에서 유를 창조해내는 아이디어 (Building ideas from nothing)
  - . 이야기 전개와 기본적 기술 습득(Learn basic elements of storytelling)
  - . 창의적 사고 (Creative thinking)
  - . 자발적 문제 해결(Spontaneous problem-solving)
  - . 발표 기술(Presentation Skills)
  - . 캐릭터 창조에 대한 의미 연구(Exploring what it means to create a character.)
- 즉흥 연기의 3가지 규칙
  - . 즐기며 함께 놀이하기
  - . 이야기 구조 제공
  - . 파트너 칭찬하기
- 친숙해지기 위한 게임 실습
  - . 둘러서서 박수치기(Circle applause)
  - . Enemy defender
  - . Tag
  - . Everybody's it
  - . Explosion tag

- . 손벽치기(Blob Tag) 등을 직접 실습
- 청취 그리고 주고 받기 게임 실습
  - . Counting to twenty (20까지 세기)
  - . Unrelated conversation (서로 연관 없는 대화 하기)
  - . Without a letter (문자 없이)
- 캐릭터 개발 게임 실습
  - . Undulations (변화들)
  - . Status/Master Servant (지위/ 주인 하인)
  - . Faces (얼굴들)
  - . Smith Jones Scene
  - . To be thought lists (고려되어야할 사항들에 대한 목록)

바. The New Poetics (강사 : Don Marinelli)

(1) 수업의 개요

- 새로운 시론 (The New Poetics) 에 대한 설명
- Janet Murray의 글을 중심으로 한 새로운 해석 소개

(2) 실제 수업 내용

- 새로운 시론의 기본 요소 : 줄거리(Plot), 인물(Character), 주제 (Theme), 대사(Diction), 음악(Music), 배경(Spectacle), 상호 작용 (Interactivity), 몰입(Immersion), 변형(Transformation)

- 몰입

- . “사실보다 더 사실적인 세계를 창조하라 (Janet Murray)”
- . Heim은 “가상현실”을 “실제로는 존재하지 않지만 효과 면에서 사실과 같은 사건이나 존재”라고 묘사
- . “허구에 대한 자발적 믿음 (Coleridge)”

- agency

- . 행위만으로는 agency가 성립되지 않음.
- . 게임 제작자들이 때때로 시간당 반복(인터랙션)의 숫자에 집중

- . 체스는 매우 적은 혹은 빈번하지 않은 동작을 가지는 게임이지만 . 디지털 환경의 특성에 따른 agency의 한 형태가 공간 항해(spatial navigation)
- . 선형 이야기는 하나의 복잡한 인간의 사건을 포함하는 하나의 형태로 나아가게 됨
- 변형
  - . 디지털에서의 이야기는 단순히 목격하는 것이 아닌 이야기 자체를 연기하는 기회를 제공함으로써 또 다른 강력한 요소를 더함.
  - . Janet Murray에 의하면 연기(enactment)란 참여자가 어떤 연락을 연기하거나 공식화된 요소로부터 자신들만의 이야기를 구성해나가도록 초대받는 경험
  - . 디지털 내러티브의 변화무쌍하고 모자이크 된 구조 즉 다중적 방법으로 동시에 행해지는 동작을 표현할 수 있는 능력으로 인해 가능
  - . 복잡한 하이퍼텍스트는 일종의 정신적 단편과 같은 내부적 독백을 포착하는 이상적인 매체

#### 사. Interactive Entertainment (강사 : Don Marinelli)

##### (1) 수업의 개요

- 사회 또는 청소년 사이에서 일어나고 있는 변화와 그에 대한 인식
- interactive entertainment의 특징 및 사례 분석

##### (2) 실제 수업 내용

- 가속되고 있는 세대 간의 변화에 대한 적응
  - . 요즘 젊은이들의 심리적, 사회적인 현실 밑바닥에는 Theme의 동적인 매력을 추구하고 있기 때문임.
  - . 따라서 요즘 젊은이들이 과거 세대들과 다른 사회적, 심리적, 지적인 요소가 무엇이나에 대한 연구 필요
- 시간과 공간에 대한 변화된 인식

- . 새로운 인식을 가진 청소년들을 교육시키는 교육자들도 그들의 인식에 주목할 필요가 있음.

- . 게임의 "상호작용의 역동성(dynamic of interactivity)" 등에 주목해야 함

- Interactive Entertainment 의 특징

- . Immersive Environment : 현재 비디오, 사운드, Haptic 등의 기술을 통해 몰입환경의 구현이 가능

- . Digital Mobility : 언제, 어디서나 상상의 세계에 몰입할 수 있음

- . Transformation of Player into Actor : Player가 단순한 Audience 에서 Actor로의 역할 전환이 가능

- . Beyond Winning or Losing : Loser도 Replay를 함으로써 다시는 같은 실패를 되풀이 하지 않도록 하는 환경이 제공됨

- . Navigable Virtual Worlds : 마음대로 가상의 세계를 여행할 수 있을 뿐 아니라 그 여행을 통해 감정을 공유할 수 있음

- . Massively Multiplayer Online Worlds : 수평적인 사회에서 각자의 역할에 따라 Play할 수 있는 환경 제공

- . Non-Linearity : 사용자의 선택을 통해 변화 무쌍한 경험이 가능

- . Multi-Sensorial Stimulation : 모든 감각기관을 동시에 필요로 하는 환경을 구축함으로써 청소년들의 호기심을 자극

아. Finding Pattern and Structure in Music (강사 : Roger B. Dannenberg )

(1) 수업의 개요

- 게임 등 엔터테인먼트 분야에서의 음악의 직간접적 중요성 강조
- 실제 감성과 접목되는 음악 및 효과음 관련 기술 현황 설명

(2) 실제 수업 내용

- 음악 연구의 구분 : 음악이란 감정, 시간, 추상적 구조의 조합물  
. 정형 이론

- . 표기법
- . 추상적 구조
- . 피치, 시점, 하모니 등에 관한 이론
- 컴퓨터 반주의 개요
  - . 모노포닉 : 화음 없는 단선율을 의미
  - . 폴리포닉 : 여러 채널로 구성된 다선율을 의미
- 음악의 이해 : 음악 내에 존재하는 패턴과 구조를 인식하기
  - . 표피적인 구조 : 피치, 음량, 하모니, 음표
  - . 심층적인 구조 : 악구 사이의 관계성, 악보 따라가기
  - . 음악이 주는 감정
- 게임을 위한 음악의 특징
  - . 멜로디는 정할 수 없지만 코드(화음)은 대체로 일정한 특징이 있음
    - . 악구 단위 합성
    - . 음표 단위 합성
- 장래의 도전 과제
  - . 악보를 이해하고 청취만으로 악보를 따라가는 기술
  - . 현대 오케스트라나 재즈 4중주의 연주를 합성해 내기
  - . 작곡물 분석, 주제부 구별, 리듬의 모티브, 내재율, 전개 기술

## 5. 3D 게임 개발 도구 Panda3D의 개요

가. Panda3D란? (자료 출처: <http://www.panda3d.org/>)

Panda3D is a 3D engine: a library of subroutines for 3D rendering and game development. The library is C++ with a set of Python bindings. Game development with Panda3D usually consists of writing a Python program that controls the Panda3D library. Panda3D is unusual in that its design emphasis is on supporting a short learning curve and rapid development. It is ideal whenever deadlines are tight and turnaround time is of the essence. For example, in a class called [Building Virtual Worlds](#) at the [Entertainment Technology Center](#), interdisciplinary groups of four students are asked to create virtual worlds in two weeks each. Screenshots of their projects are visible throughout this site. Panda3D is what makes this rapid turnaround possible. Panda3D was developed by [Disney](#) for their massively multiplayer online game, [Toontown](#). It was released as free software in 2002. Panda3D is now developed jointly by Disney and [Carnegie Mellon University's Entertainment Technology Center](#).

나. Panda3D의 주요 특성

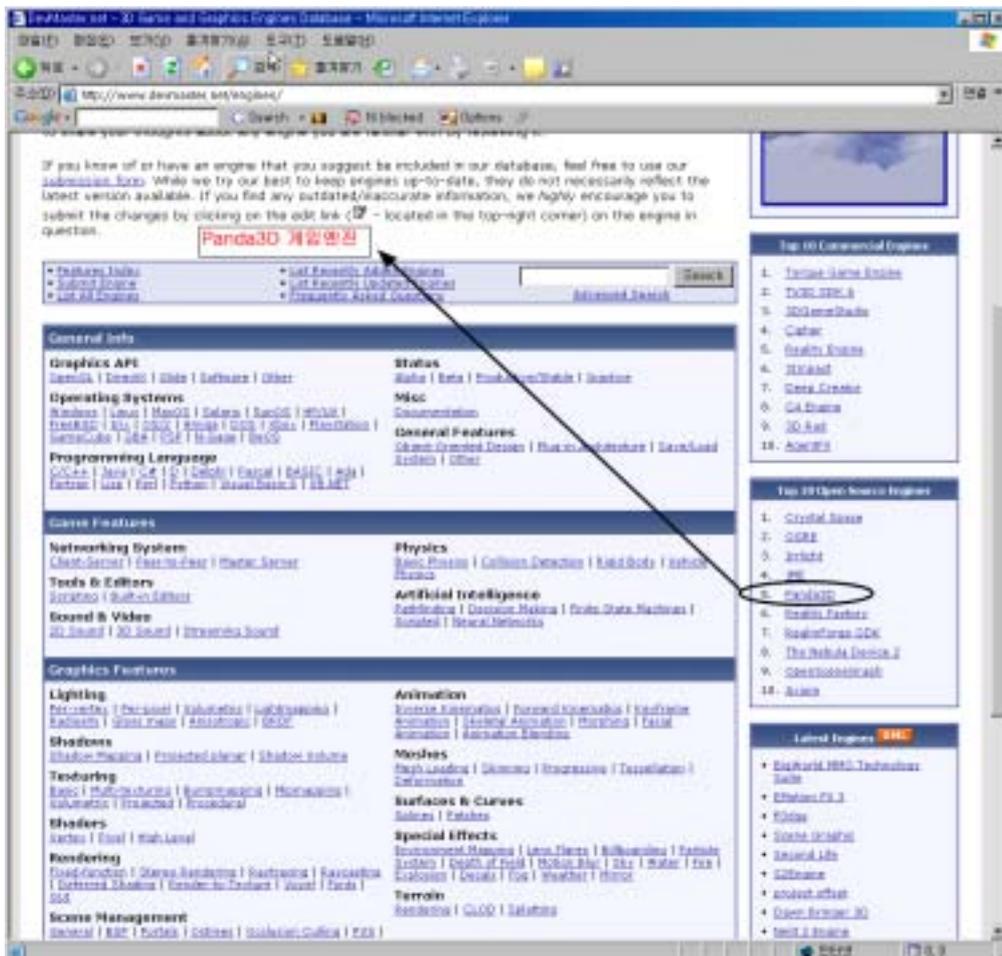
- Short learning curve
- Rapid development
- Library callable from C++ or Python
- Rock solid stability
- Free software license
- Support for Windows and Linux
- Simple installation
- Collection of free models and artwork

- Multiple active development teams
- . Disney Imagineering
- . Entertainment Technology Center
- Lots of documentation
- . Extensive manual
- . Collection of code samples
- . API Reference materials

다. Panda3D 에 관한 평가

Panda3D 는 2006년 4월 기준으로 오픈 소스 엔진 5위에 오를 정도로 우수한 엔진으로 평가 받고 있다.

(자료 출처: <http://www.devmaster.net/engines>)



## 6. Python과 Panda3D의 주요 특징 소개

### 가. Python의 정의

파이썬은 강력한 기능과 함께 명확한 문법을 가지고 있다. 모듈, 클래스, 예외 처리 및 고차원의 동적 자료형, 동적인 자료형 결정 기능을 가지고 있다. 많은 시스템 호출과 라이브러리 그리고 윈도우 시스템(X11, Motif, Tk, Mac, MFC)에 대한 인터페이스를 가지고 있다. C, C++를 사용하여 새로운 모듈을 쉽게 만들어 낼 수 있다.

파이썬은 프로그래밍 가능한 인터페이스를 필요로 하는 애플리케이션에 확장 언어로 사용될 수 있다.

### 나. Python의 유래

그리스 신화에 나오는 동물로 Parnassus 산에 숨어 살고 있는 커다란 뱀으로서 아폴로(Apollo)가 죽였다. 아시아, 아프리카, 오스트레일리아에 살고 있는 커다랗고 독성 없는 뱀으로서 먹이를 질식시켜 죽인다.

### 다. Python 언어

1989년 네델란드인 귀도 반 로섬 (Guido Van Rossum)에 의하여 개발되었으며, 인터프리터 방식의 객체지향 스크립트 언어이기 때문에 어떤 시스템에서든지 OS와 관계없이 파이썬 해석기만 있으면 실행할 수 있다(platform independent, portable). 객체지향 스크립트 언어이기 때문에 C나 C++ 같은 언어에 비해 생산성(개발 시간)이 매우 뛰어나다. 또한 언어의 구조가 오류를 줄일 수 있도록 되어 있다.

### 라. Python과 Panda3D 의 참고 사항 및 사이트

- Official Website <http://www.python.org>
- Current Python Documentation <http://docs.python.org>
- Python Documentation <http://www.python.org/docs>

- Python Tutorial. Written by Guido Van Rossum, the author of Python.

<http://docs.python.org/tut/tut.html>

- <http://www.panda3d.org/>

#### 마. Panda3D 시작

텍스트 파일을 생성하고 확장자를 .py로 저장한다. PYPE와 IDLE는 Python의 특별한 텍스트 에디터이지만 모든 텍스트 에디터로 작성이 가능하다. 다음과 같이 Python 파일에 텍스트를 입력하면

```
import direct.directbase.DirectStart  
run()
```

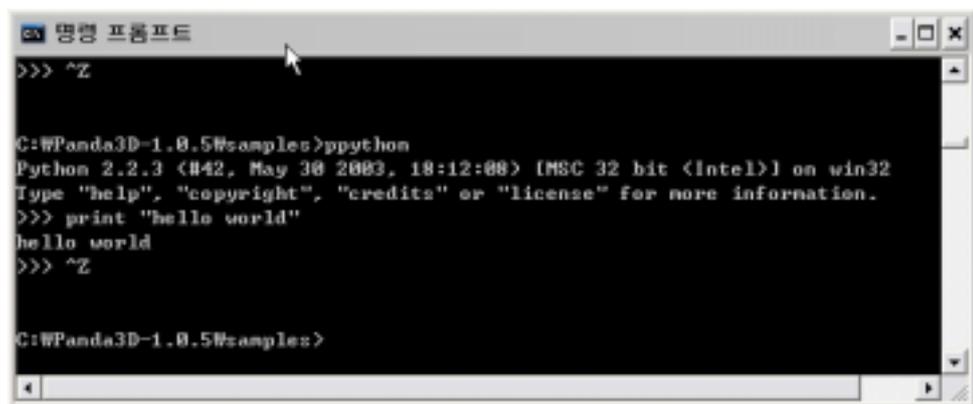
위의 내용을 입력한 후 확장자를 filename.py로 저장 후 도스명령어 창에서

```
ppython filename.py
```

를 입력하면 실행되며, 그 결과는 빈 창이 생성된다.

Panda3D에서 python 파일을 실행하는 방법은 두 가지가 있다.

- 1) ppython filename.py
- 2) ppython 입력 후 인터랙티브 모드에서 실행하는 방법이 있다



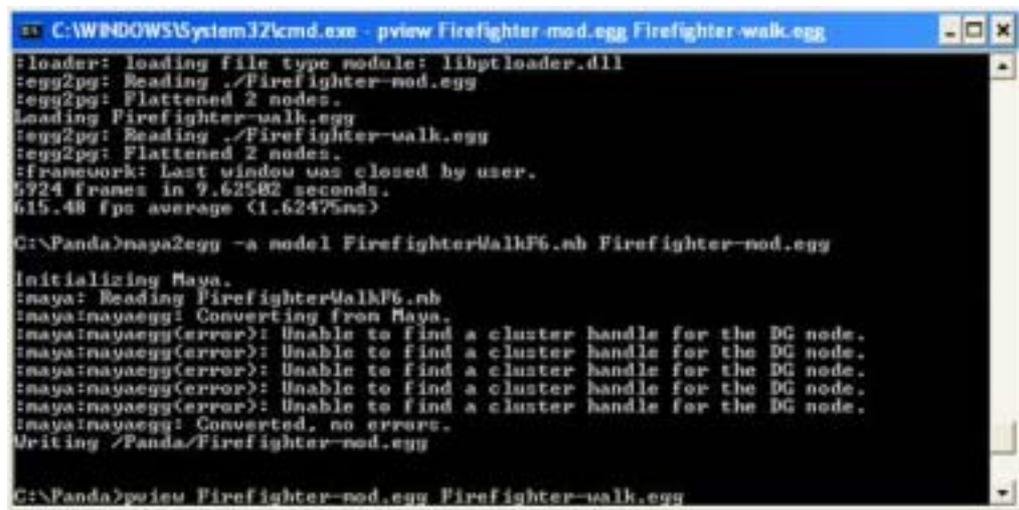
```
명령 프롬프트  
>>> ^Z  
  
C:\WPanda3D-1.0.5\samples>ppython  
Python 2.2.3 (#42, May 30 2003, 18:12:00) [MSC 32 bit (Intel)] on win32  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print "hello world"  
hello world  
>>> ^Z  
  
C:\WPanda3D-1.0.5\samples>
```

### [그림 1] 실행 결과

위 그림은 인터랙티브 모드에서 “hello world”를 수행한 결과이며, 중지하는 방법은 Ctrl - z 키를 입력한다.

또한 Maya 및 3D 스튜디오 맥스에서의 모델링 데이터는 egg 파일로 변환한다.

변환하는 방법은 maya2egg filename.egg filename.mb를 수행한다.



```
C:\WINDOWS\system32\cmd.exe - pview Firefighter-mod.egg Firefighter-walk.egg
:loader: loading file type module: libytloader.dll
:egg2pg: Reading ./Firefighter-mod.egg
:egg2pg: Flattened 2 nodes.
Loading Firefighter-walk.egg
:egg2pg: Reading ./Firefighter-walk.egg
:egg2pg: Flattened 2 nodes.
:framework: Last window was closed by user.
6924 frames in 9.62582 seconds.
615.48 fps average (1.62475ms)

C:\Panda>maya2egg -a node1 FirefighterWalkF6.mb Firefighter-mod.egg

Initializing Maya.
:maya: Reading FirefighterWalkF6.mb
:maya:mayaegg: Converting from Maya.
:maya:mayaegg(error): Unable to find a cluster handle for the DG node.
:maya:mayaegg(error): Unable to find a cluster handle for the DG node.
:maya:mayaegg(error): Unable to find a cluster handle for the DG node.
:maya:mayaegg(error): Unable to find a cluster handle for the DG node.
:maya:mayaegg: Converted, no errors.
Writing ./Panda/Firefighter-mod.egg

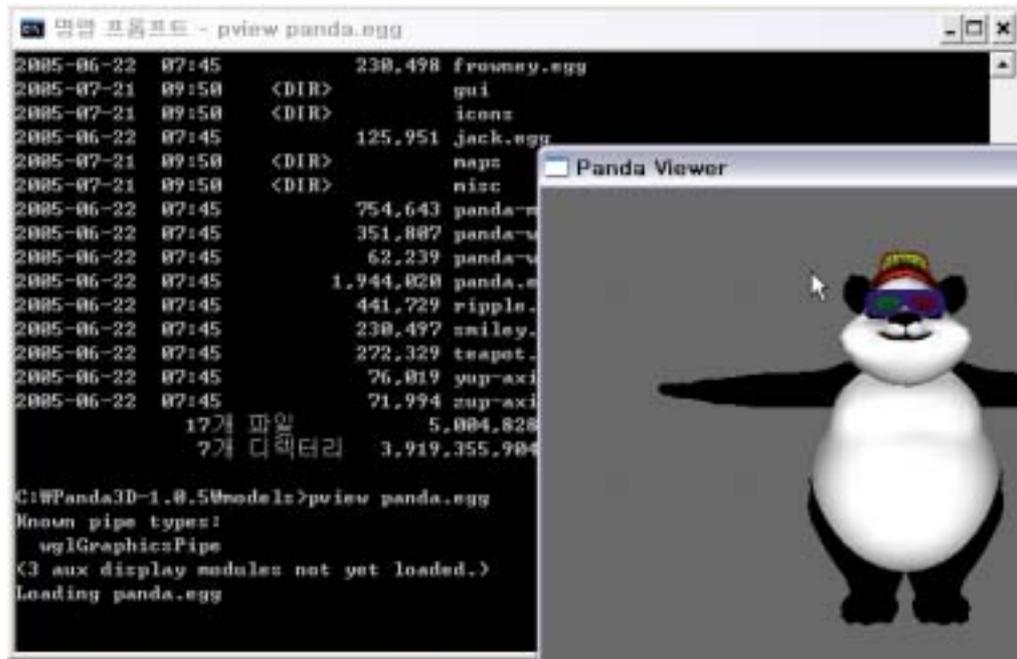
C:\Panda>pview Firefighter-mod.egg Firefighter-walk.egg
```

### [그림 2] 실행 결과

변환된 filename.egg 데이터를 도스 창에서 다음과 같이 입력한 후 확인할 수 있다.

#### **pview filename.egg**

위 명령어의 결과 egg 데이터를 확인 할 수 있으며, 창에 마우스 커서를 위치한 후 c, l, w 키를 누르면서 중앙위치, 라이트, 와이어프레임 결과 등을 확인 할 수 있다. 또한 마우스 왼쪽버튼은 상하좌우 중간버튼은 회전, 오른쪽버튼은 앞뒤로 이동할 수 있다. 그 외에 자세한 옵션은 -h 사용으로 자세히 익힐 수 있다.



[그림 3] 실행 결과

### 1) Loading models

only model:	obj1=loader.loadModel('models/model.egg')
model with animations:	obj2=Actor.Actor('model/model.egg', {'anim1':'models/modelAnim1.egg', 'anim2':'models/modelAnim2.egg'})
control functions:	obj2.play('anim1') obj2.stop('anim1') obj2.loop('anim2') obj2.pose('anim2', 0)

### 2) Intervals

Actors:	int1=actorInterval('anim1', loop=1, duration=100, startFrame=2, endFrame=102, playRate=.5)
Sounds:	int2=soundInterval('sound1' loop=1, duration=100, volume=.5, startTime=.1)
Functions:	int3=Func(obj.setPos, Point3(0,0,0)) int4=Func(obj.hide)
Lerp: (linear interpolation)	int5=LerpFunc(obj.setX, 0, 1000,duration=10) int5=obj.posInterval(duration=1, Point3(0,0,0), Point3(1000,0,0))
Sequences & Parallels:	inOrder=Sequence(int1,int2,int3,int4,int5) alltogether=Parallel(int1,int2,int3,int4,int5)
control functions:	int.start() int.finish() int.pause() int.resume() int.setT(1) int.getT()

	int.isPlaying() int.isStopped() int.loop()
--	--

### 3) Messages

accepting & ignoring messages:	class Foo(DirectObject.DirectObject): self.accept('event1',self.fun1) self.acceptOnce('even2',obj.show) self.ignore('event2') self.ignoreAll()
sending messages:	messenger.send('event1') interval.setDoneEvent('event2')
debugging messages:	messenger.toggleVerbose() print messenger messenger.clear()

### 4) Collision detection

handler:	collHandEvent=CollisionHandlerEvent() collHandEvent.addInPattern ("enter%in")
traverser:	collTraverser=CollisionTraverser() base.cTrav=collTraverser collTraverser.getNumColliders() collTraverser.clearColliders()
collision spheres:	collSphere=CollisionSphere( x, y, z, radius) cSphereStr ='CollisionHull' collSphereNode=CollisionNode( collSphereStr ) collSphereNode.addSolid(collSphere)
selective collisions:	collSphereNode.setIntoCollideMask(BitMask32.bit(1)) collSphereNode.setFromCollideMask(0) collTraverser.addCollider(collSphereNode, collHandEvent) accept( 'enter' + collSphereStr, collideFnc)
debugging:	debugCollObj=(node.attachNewNode(collSphereNode)) debugCollObj.show()
on Collisions:	def collideFnc(collEntry): print collEntry.getFromNode().getName() print collEntry.getIntoNode().getName()

### 5) Transformations, Rotations, Scaling, etc.

to scale:	obj.setScale(VBase3(0,0,0)) or obj.setScale(0,0,0) obj.setSx(0) obj.setSy(0) obj.setSz(0)
to move:	obj.setPos(VBase3(0,0,0)) or obj.setPoint(0,0,0) obj.setX(0) obj.setY(0) obj.setZ(0)
to rotate:	obj.setHpr(VBase3(0,0,0)) or obj.setHpr(0,0,0) obj.setH(0) obj.setP(0) obj.setR(0)
to color:	obj.setColor(VBase4(1,1,1,1)) or obj.setColor(1,1,1,1) obj.clearColor()

### 6) Sounds

to Load:	loader.loadSfx('sounds/sound.wav') loader.loadMusic('sounds/sound.wav')
control functions:	sound.play() sound.stop() sound.setTime(.5) sound.setVolume(0)

	<pre>base.disableAllAudio() base.enableAllAudio() base.enableMusic(bEnableMusic) base.enableSoundEffects(bEnableSoundEffects)</pre>
--	---

## 7) Tasks

Creating:	<pre>def exampleTask(task): if task.time &lt; 2.0: return Task.cont print 'Done' return Task.done</pre>
Task Manager:	<pre>taskMgr.add(exampleTask,'aTask') taskMgr.Remove('aTask') taskMgr.doMethodLater(10, aFunc, 'otherTask')</pre>

## 8) Finite State Machines

Creating:	<pre>class NewStyle(FSM.FSM): def enterRed(self): print "enterRed(self,'%s','%s'" % (self.oldState, self.newState) def filterRed(self, request, args): if request == 'advance': return 'Green' return self.defaultFilter(request, args) def exitRed(self): print "exitRed(self,'%s','%s'" % (self.oldState, self.newState)</pre>
Using:	<pre>aFSM= NewStyle('a new FSM') print aFSM.state aFSM.request('Red') aFSM.request('advance')</pre>

## 9) Lighting

Ambient Light:	<pre>aLight = AmbientLight( 'ambientLight' ) aLight.setColor( Vec4( 0.4, 0.4, 0.4, 1 ) )</pre>
Directional Light:	<pre>dLight = DirectionalLight( 'directionalLight' ) dLight.setPoint(Vec3(0,0,0)) dLight.setDirection(Vec3( 0.3, .2, .10 )) dLight.setColor(Vec4( 0.4, 0.4, 0.4, 1 ))</pre>
Point Light:	<pre>pLight = PointLight( 'pointLight' ) pLight.setPoint(Vec3(0,0,0)) pLight.setColor(Vec4( 0.4, 0.4, 0.4, 1 ))</pre>
Spot Light:	<pre>spot = Spotlight( 'spotlight' ) spot.setPoint(Vec3(0,0,0)) spot.setColor( Vec4( 0.4, 0.4, 0.4, 1 )) spot.setDirection(Vec3( 0.3, .2, .10 )) spot.setLens(PerspectiveLens()) spot.getLens()</pre>
general stuff:	<pre>lightAttributes=LightAttrib.makeAllOff() lightAttributes.addLight( aLight) render.attachNewNode(aLight.upcastToPandaNode()) render.node().setAttrib(lightAttributes )</pre>

## 7. Panda3D 를 활용한 프로젝트 진행 사례 분석

### 가. 프로젝트 진행 개요

프로젝트의 역할: 해당 프로젝트의 진행은 카네기멜론대의 ETC의 한국게임산업개발원 위탁 특별 과정인 4주간의 교육 과정과 관련하여 교육생을 위한 팀 단위 진행 프로젝트로써 실시되었음

프로젝트의 진행: 4주간의 교육 기간 중 첫 주인 1주차는 개별 개발 도구에 대한 개인 소양 능력 평가 및 기본 사용법 교육을 위해 활용되었으며 실제 프로젝트 진행은 2주차 ~ 4주차까지의 3주간의 기간을 통하여 진행됨

프로젝트 팀의 구성: 교육생 11명을 2개 팀으로 구분하여 진행

### 나. BVW 과정으로써의 프로젝트 진행의 역할

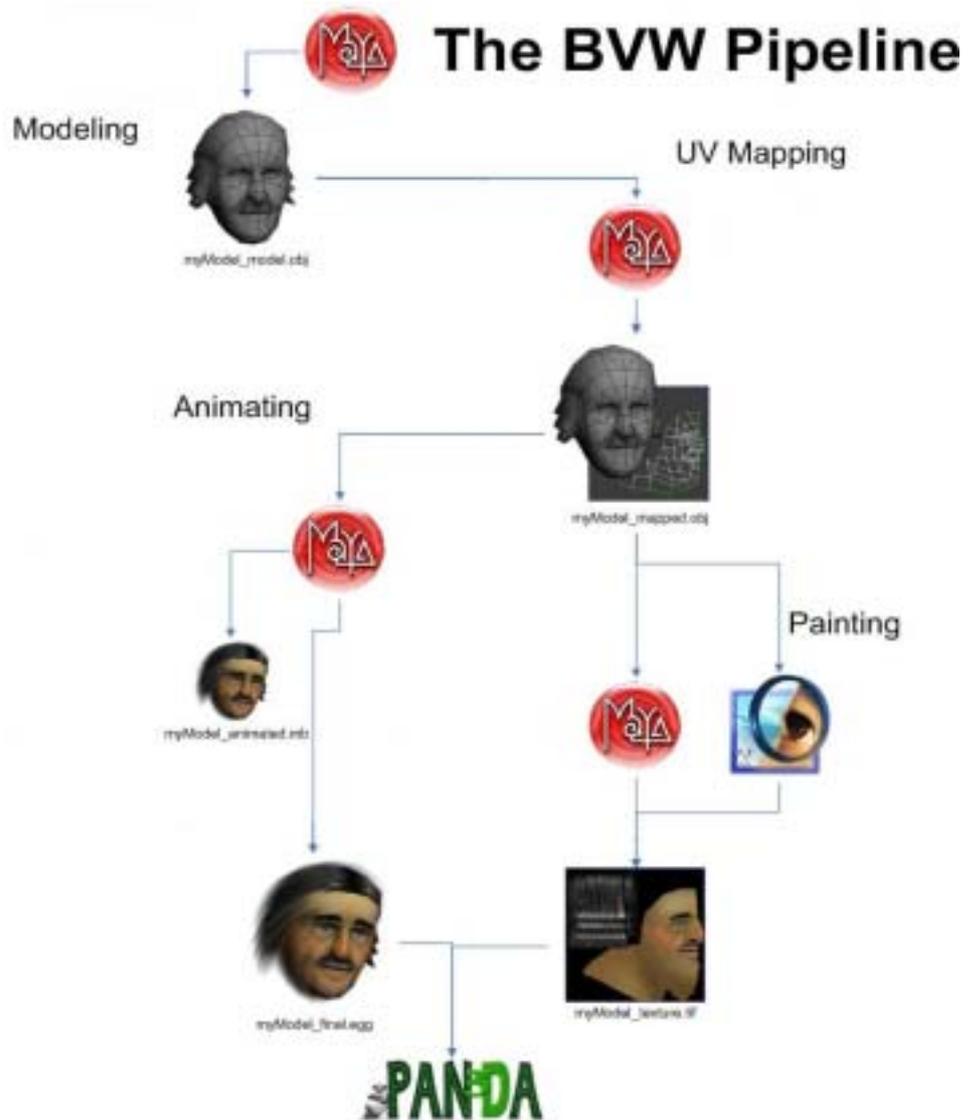
프로젝트 진행의 목표: 해당 프로젝트의 진행은 프로그래밍, 기획, 그래픽의 분야별 수준 향상 및 신기술 습득 등의 개별적인 기술 향상을 위한 목적으로 진행되기보다는 팀을 이루는 각 개인별 전공 및 특징점을 하나의 단일한 게임 아이디어를 구체화하고 이를 직접 구현하는 전체 과정의 실제적 체험을 위해 고안되었음

프로젝트 진행과 관련된 참고 사항: 해당 프로젝트는 CMU ETC의 정규 교과과정으로 편성되어 있으며, BVW (building a Virtual World) 라는 명칭으로써 한 학기 동안의 기간 동안 최소 5개 이상의 팀 프로젝트를 통하여 실제 산출물을 개발해내는 방식으로 운영되고 있음. 실제 재학생의 경우, 그래픽 프로그래밍 사운드 기획 등으로 다양한 배경과 전공을 가지는 학생들이 개인적인 친분이나 교류와 관계없이 무작위로 구성된 팀에서 “아이디어 발상 게임 기획 게임의 제작 테스트 프리젠테이션 및 활용” 등의 전 과정을 단기간에 체험하는 경험을 가지게 됨

프로젝트 팀의 구성: 프로젝트 팀은 서로 다른 전공 및 역할을 맡은 4~7인으로 구성되며 (최소 3명, 최대 12명으로 구성) 해당 팀은

해당 프로젝트가 진행되는 2 3주의 기간 동안만 한시적으로 운영  
됨

- . 일반적인 팀 구성: 모델러 1, 페인터 1, 프로그래머 1, 사운드 1  
프로젝트 팀원의 역할 구분
- . 게임 그래픽 디자이너: 게임에 사용되는 캐릭터나 게임 배경의  
그래픽 작업을 담당. 일반적인 경우에는 1개 팀 당 2명 이상이  
적당하며 모델러와 페인터로 역할 구분. 사용하는 개발 도구는  
3DMax 또는 Maya
- . 게임 프로그래머: 실제 게임의 알고리즘 및 자료 구조를 설계하  
고 개발하는 역할 담당. BVW 를 위한 개발 도구로는 Alice 또는  
Panda3D 사용, 2005년 기준으로는 Panda3D 가 주로 활용됨
- . 게임 사운드 디자이너: 게임의 배경이 되는 BGM (Back ground  
Music) 이나 각종 효과음을 준비하는 역할 담당. CMU ETC 의  
경우에는 별도로 준비되어 있는 편집실을 통하여 작업을 진행



[그림 4]

다. 팀 프로젝트 진행 개요

1) 팀 구성 내용

프로젝트 기간: 2005년 7월 25일 ~ 8월 12일 (3주간)

팀A: 장희동, 송병호, 이광재, 김미진 김영진 교수

팀B: 장신환, 장창익, 조용환, 윤임모, 이대운, 이창조 교수

프로젝트 매니지먼트/기획

사운드 디자인 및 에디팅

인터페이스 분석 / 동영상 제작  
그래픽 디자인 및 모델링:  
프로그래밍

프로젝트 진행 목표

- . PC 기반
  - . 별도의 플랫폼을 통한 게임 개발 경험
  - . 결과 중심 보다는 과정 중심의 프로젝트 진행
  - . 진행 전 과정에서의 전체 팀원의 참여 중시
- 프로젝트 개발 환경
- . 구동 환경: PC / Windows
  - . 인터페이스: Jam O Drum
  - . 프로그래밍 개발 환경:
  - . 진행 전 과정에서의 전체 팀원의 참여 중시

## 2) 실제 프로젝트 진행 시의 주요 일정

팀 구성 및 브레인스토밍: 7월 25일 (월)  
프로젝트 관련 조정 회의 및 발표 자료 준비: 7월 27일 (수)  
1차 프리젠테이션(기획 안 발표): 7월 28일 (목)  
프로그래밍/그래픽/기획/사운드 세부작업 시작: 8월 2일 (월)  
2차 프리젠테이션(프로토타입 발표): 8월 5일 (금)  
진행 문제점 확인 및 집중 작업 실시 개시: 8월 8일 (월)  
최종 조정 회의 및 작업 스케줄 추가 조정: 8월 10일 (수)  
3차 프리젠테이션(최종 발표): 8월 12일 (금)

## 라. 팀 프로젝트 진행 내용

### 1) 개발 목표의 설정

CMU ETC 에서 부여한 과제 조건 (요약)

- . 위험 또는 곤란한 상황에 처한 캐릭터와 이를 위험 상황에서 벗어나게 해주는 캐릭터를 등장시키는 상황 구성

. 개발 환경은 Maya, Panda3D 를 활용

## 2) 브레인스토밍 과정

팀원 상호간 긍정을 원칙으로 한 브레인스토밍을 통하여 다양한 상황을 구성하고 이를 다른 팀원에게 설명

팀 A의 아이디어는 낙하산이 없어 위험에 빠진 엄마 거북 및 거북이 알을 낙하산을 가지고 뛰어내린 네 마리의 아기 거북이가 구한다는 내용으로 선정

팀 B의 아이디어는 몬스터 성에 잡혀 판다로 변한 남자 친구를 구하는 여주인공 앨리스의 모험과 대장 몬스터를 물리친 후 판다가 마법에서 풀려 남자친구를 구하고 몬스터성을 탈출하는 시나리오로 선정

게임의 분위기는 가족간 사랑 (휴머니티) 을 최대한 강조하는 한편 카네기 교수진을 몬스터로 표현하여 학생 입장에서는 흥미 유발

## 3) 개발 플랫폼 및 개발 환경의 결정

팀A는 Jam O Drum 장비를 대상으로 개발하며, 팀B는 3D 게임으로 구현하기로 결정

그래픽 작업은 Maya 개발 후 egg 파일 변환을 통하여 Panda3D 프로그램으로 통합하기로 하였으며 Jam O Drum 과 관련된 프로그램 상의 어려움은 Framework 및 데모 프로그램의 분석을 통하여 해결하기로 방향을 잡음



[그림 5] 개발 플랫폼으로 결정한 Jam O Drum의 모습

4) 프로젝트 진행 초기 컨셉 아트 디자인 (김미진 교수 작성)

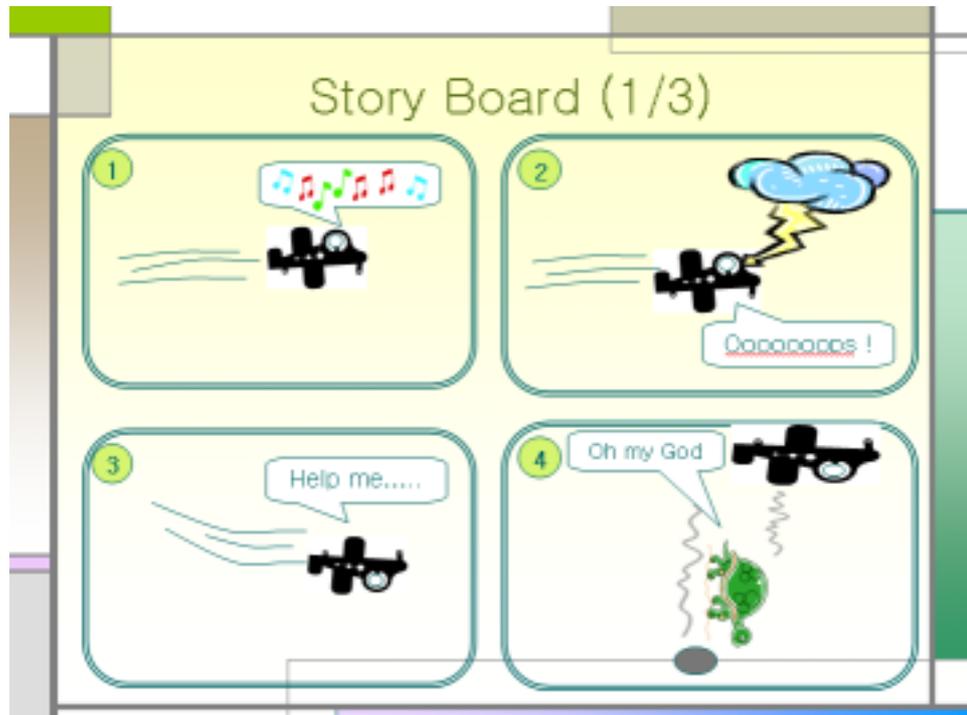


[그림 6] 게임 등장 캐릭터의 초기 스케치

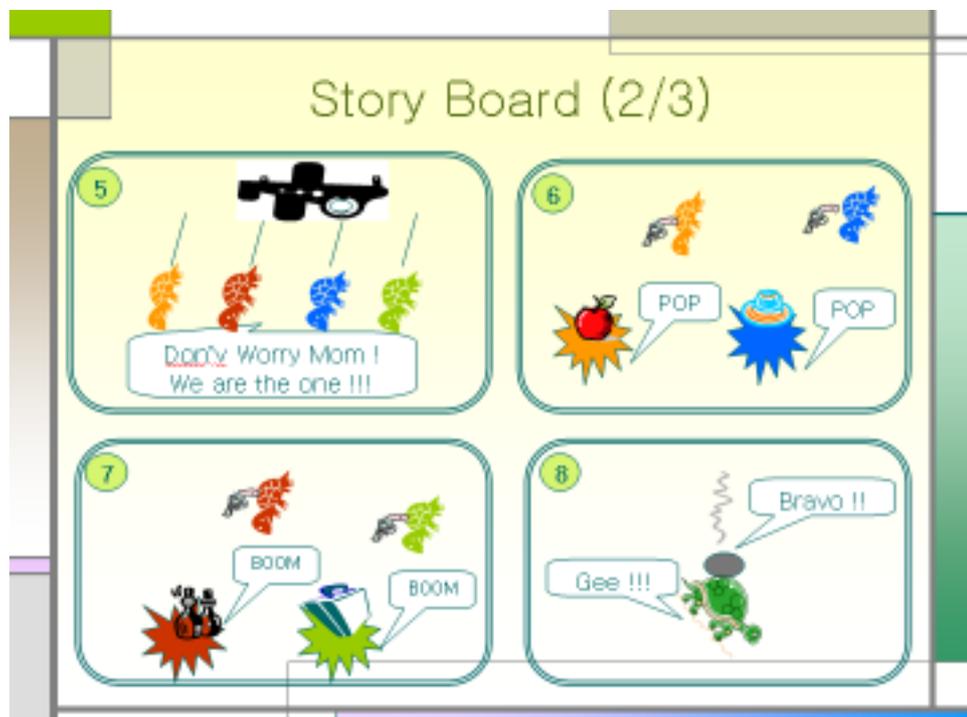


[그림 7] 위험 상황에 대한 초기 디자인

5) 초기 스토리보드 문서



[그림 8] 스토리보드 문서 (첫 번째 장)



[그림 9] 스토리보드 문서 (두 번째 장)



[그림 10] 스토리보드 문서 (세 번째 장)

6) 초기 계획에서의 팀 내부 개발 일정 (안)

Concept Design (7/28 Thu)

. Idea and High Level Concept Design (7/26)

. Concept Design Draft (7/27)

. Presentation of Concept Design (7/28)

Demo Prototype World (8/5 Fri)

. First Prototype (8/2)

. Second Prototype (8/4)

Present Complete Final World (8/12 Fri)

. Alpha Version (8/9)

. Beta Version (8/10)

. Play Test and Calibration (8/11)

7) 초기 계획에서의 게임 월드의 구성 안

동작	물리규칙	이벤트
낙하운동	수직방향: 일정한 가속도로 떨어짐	



	동작	전하면서 흔들리는 동작
우 방향전환동작	우측 방향 10도 회전 동작	몸의 자세는 머리와 꼬리를 회전축으로 하여 45도 회전하면서 흔들리는 동작
엄마를 꼭 잡는 동작	엄마를 두 팔로 꼭 잡는 동작	몸의 자세는 기부의 나팔이나 다리를 두 손으로 잡는 동작

10) 초기 계획에서의 새끼 거북이의 동작 구성 안



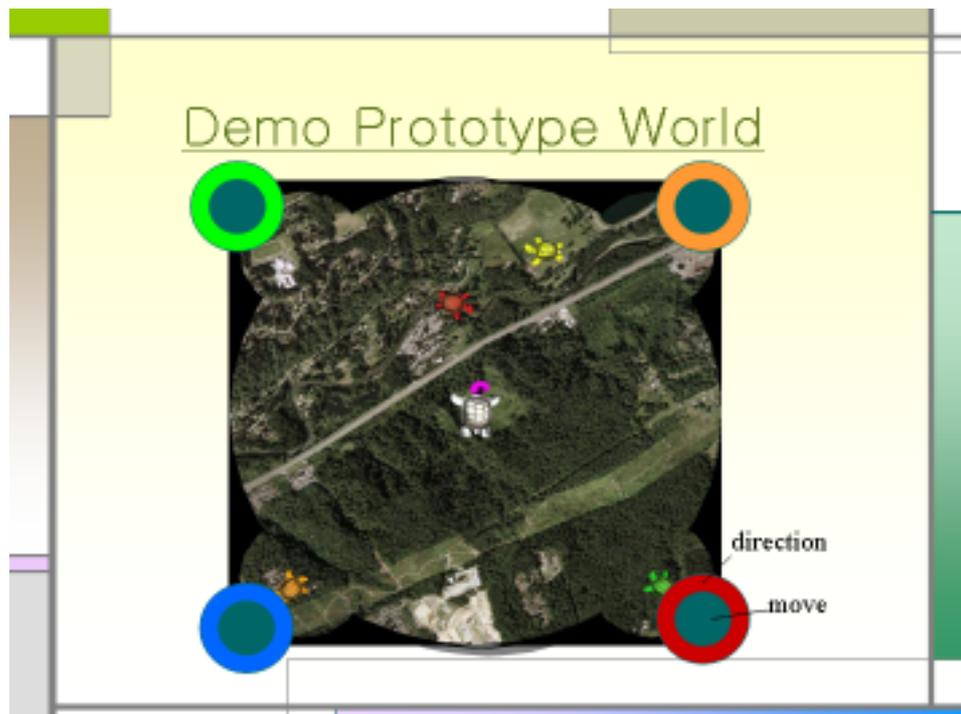
11) 프로토타입 데모 단계에서의 작업 진행 상황 및 일정 변경 내용

### The Present State/Time Sched

Activity	20%	40%	60%	80%	100%
<u>Intro Visual</u>			8 Aug.	8 Aug.	8 Aug.
Building World					
Character Arts				8 Aug.	9 Aug.
Play Mechanics				8 Aug.	9 Aug.
Visual Effects			8 Aug.	8 Aug.	9 Aug.
Sound		8 Aug.	8 Aug.	9 Aug.	9 Aug.
Ending Visual		9 Aug.	9 Aug.	10 Aug.	10 Aug.
Integration/Test				8 Aug.	11 Aug.

[그림 12] 프로토타입 데모 단계에서의 중간 일정표

12) 프로토타입 데모 단계에서의 화면 구성 내용



## [그림 13] 프로토타입 데모 단계에서의 화면 구성 내용

### 13) 프로토타입 데모 단계에서의 플레이 수정 안

엄마 거북이 붙잡기

[1안] 특정 신호를 통하여 엄마를 붙잡을 수 있는 시간을 지정해 주는 방법

[장점] 해당 시간 외에는 엄마를 조정하여 박진감이나 동적인 움직임 지정 가능

[단점] 유희 시간에 플레이어가 할 일이 없음

[구현방안] 엄마를 랜덤하게 이동 또는 회전 시키다가 멈춘 시점부터 5초에서 7초간 잡을 수 있는 시간 부여

[2안] 플레이어가 조정이 가능한 경우 자유롭게 붙잡는 것을 허용해 주기 (난이도 조정 포함)

[장점] 플레이어는 계속 플레이에 집중함

[단점] 유희 시간에 플레이어가 할 일이 없음

[구현방안] 한 마리의 거북이가 먼저 붙잡을 경우, 특정 시간의 사운드를 주거나 신호를 주어 해당 시간 사이에 나머지 세 마리가 모두 붙으면 성공, 시간 초과 시엔 거북이를 다시 떨어뜨림

착륙 과정

[1안] 새로운 미션 주기 (예: 계속 두들길 경우에만 낙하산이 퍼주게 조정)

[장점] 게임이 계속 진행되어 긴장감을 유지함

[단점] 사용자가 혼란을 일으킬 수 있음

[구현방안] 일단 네 마리의 거북이가 엄마를 붙잡으면 화면에 “Hit” 라는 표시를 줌

[2안] 거북이 네 마리가 엄마를 잡으면 동영상으로 바로 이동 (난이도 조정을 통하여 거북이가 엄마를 잡기 어렵게 할 경우 사용)

[장점] 플레이어는 미션을 완수한 후 충분한 보상을 받음

[단점] 엔딩 동영상이 길면 지루해 짐

[구현방안] 네 마리가 모두 성공하면 제어권을 플레이어로부터 회수하고 동영상으로 연결하며 이 경우 음악 선택이 매우 중요함

기타 사항

장애물 등장 시 장애물에 부딪치면 새끼 거북이들을 뱅글뱅글 돌린 후 시작 지점으로 돌아가게 할 예정임

패드 부분에 동그라미를 그려주지 않을 경우, 게임을 더욱 재미있게 해주는 효과가 있지 않을까 생각됨

14) 최종 프로젝트 결과물 목록

구분	파일명	세부내용
프로젝트 진행문서	storyboardPt.ppt, 게임디자인.xls 등	각종 기획 문서 및 프리젠테이션 파일, 담당자 별 작업 지시서 등
사운드 데 이터	Play.mp3, end.mp3, Ef cheering.mp3 등	게임 진행 단계별 배경 음악 파 일, 게임 효과 음향 파일 등
동영상 데 이터	Slide00.png, Slide27.png	인트로 동영상, 성공 및 실패 각 각의 엔딩 동영상 관련 파일 등
모델링 데 이터	Mother_turtle.egg, Little_turtle.egg, boo_B.jpg 등	.egg 포맷의 기본 모델 파일, 텍스 처 파일, 애니메이션 관련 화일 등
프로그램	Indy.py, Run Test.bat	프로그램, 구동용 Panda3D 프로 그램 파일, 기타 관련 문서 등

15) 단계별 동영상 구성 내용

작업 개요: Panda3D 의 동영상 지원 기능이 부족함에 따라  
정지 화면을 중심으로 스토리 중심의 애니메이션을 구성한 후,  
별도 제작한 사운드와의 동기를 고려하여 프레임 단위로 구동  
하는 방식을 적용



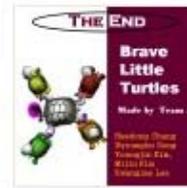
[그림 14] 인트로 동영상용 파일 작성 내용(팀A)



Slide00.PNG

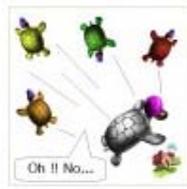


Slide01.PNG



Slide02.PNG

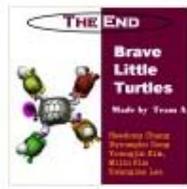
[그림 15] 미션 성공 시의 엔딩 동영상 작성 내용



Slide00.PNG



Slide01.PNG



Slide02.PNG

[그림 16] 미션 실패 시의 엔딩 동영상 작성 내용



graduate.jpg



slide00.png



slide01.png



slide02.png



slide03.png



slide04.png

[그림 17] 인트로 동영상용 파일 작성 내용(팀B)

<표 1> 인트로 화면을 표현하는 클래스 구현

```
class IntroVisual(DirectObject):
    def __init__(self):
        self.visualPlane = loader.loadModelCopy('plane/plane')

        self.visualPlane.setPosHprScale(0,8,100,0,0,0,5.8,1,4.4)
```

```

self.visualPlane.setTransparency(1)

self.introTexs = self.loadTextureMovie(5, 'Intro_png/slide',
                                         'png', padding = 2)

self.intro = loader.loadMusic('sounds/intro')

def start(self):
    base.disableMouse()
    self.visualPlane.reparentTo(render)
    camera.setPos(0,0,100)
    self.introTask = taskMgr.add(self.textureMovie, "introTask")
    self.introTask.fps = 0.06
    self.introTask.obj = self.visualPlane
    self.introTask.textures = self.introTexs
    self.intro.play()

def textureMovie(self, task):
    currentFrame = int(task.time * task.fps)

    if currentFrame < len(task.textures):
        task.obj.setTexture(task.textures[currentFrame],1)
    else :
        self.visualPlane.remove()
        messenger.send('startGame')
        taskMgr.remove('introTask')

    return Task.cont

def loadTextureMovie(self, frames, name, suffix, padding = 1):
    % i) return [loader.loadTexture((name+"%0"+str(padding)+"d."+suffix)
                                   for i in range(frames)]

```



[그림 18] 시작화면



[그림 19] 카라오케 몬스터 란도



[그림 20] 리포트 몬스터 크리스



[그림 21] 대장몬스터 단



[그림 22] 대장 몬스터로부터 마법에 풀려 판다에서 남자친구로  
변환

#### 16) 사운드 데이터 작성 내용

작업 개요: 초기 기획 회의 및 프로토타입 작성 이후 확정된 게임 플레이 내용에 맞는 다양한 배경 음악을 광범위하게 조사하여 선곡 후, 전체 팀 구성원의 합의에 따라 정해진 테마 음악을 중심으로 게임에 적용하기 위한 편집 작업을 실시하는 방식으로 진행. 게임의 효과 극대화 와 흥미 요소 유발을 위해 전체 팀원이 참여한 레코딩 작업을 통하여 대다수의 효과 음향은 직접 작업하여 활용

게임 배경 음악 작업 내용

- . 게임 메인 플레이용 파일: play.mp3 / 8,129 KB / 4분 20초 (반복)
- . 인트로 동영상용 파일: begin.mp3 / 1,496 KB / 47초
- . 미션 성공 시를 위한 파일: end.mp3 / 491 KB / 50초
- . 엔딩 크레딧을 위한 파일: credit.mp3 / 1,749 KB / 55초

## 17) 모델링 데이터 작성 내용

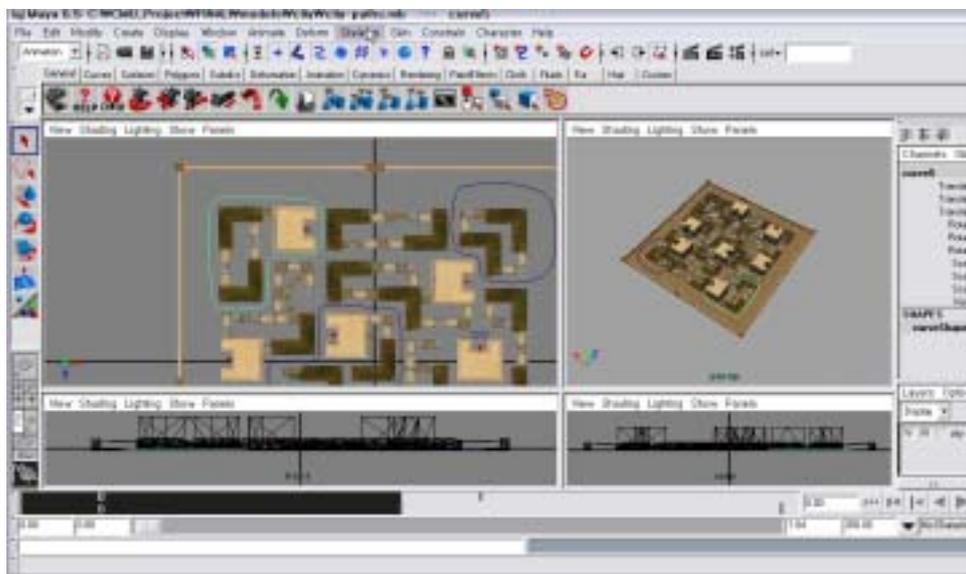
작업 개요: 본 프로젝트의 핵심 내용이었던 3D 방식의 캐릭터 구현을 위한 각종 모델 작성 및 이와 관련된 텍스처 작업을 Maya 및 확장 도구를 활용하여 진행. 프로그래밍 파트와의 원활한 의사 소통을 통하여 모델의 중복을 최소화 하고 텍스처와 애니메이션 데이터 등을 적극적으로 응용하는 방식을 채택함으로써 프로그램의 부하를 상당 부분 감소시킴. 기본적인 캐릭터 이외에도 Jam O Drum 을 위한 기본 마스크 데이터와 패드용 인터페이스, 리얼한 효과를 주기 위한 지형 화면 구성을 직접 작성하거나 지원함



[그림 23] 주요 캐릭터인 거북이의 텍스처 파일



[그림 24] 3D게임 맵 모델



[그림 25] 몬스터의 경로를 위한 Mopath 설정 작업화면

<표 2> Mopath 문을 이용 몬스터의 경로 설정

```

from direct.directutil import Mopath
from direct.interval.MopathInterval import *
(중간생략)

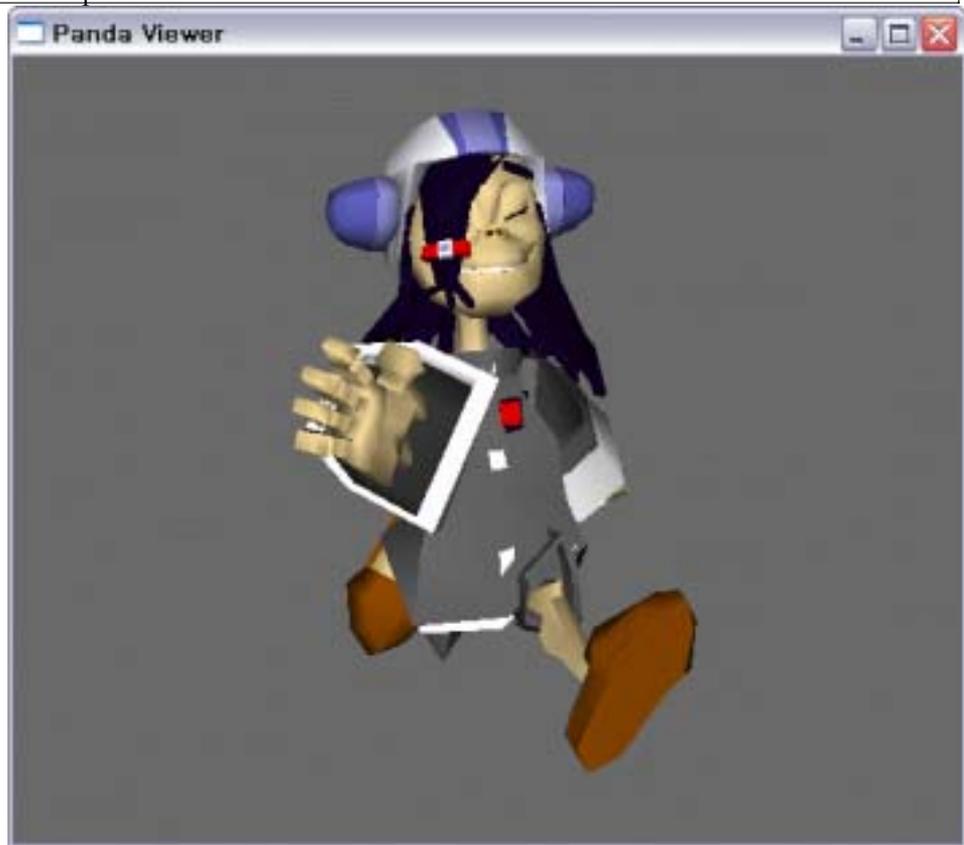
```

```

#paths
self.path1 = Mopath.Mopath()
self.path1.loadFile('models/city/path1')
self.path1.fFaceForward = 1
self.path1Interval = MopathInterval(self.path1, self.monster1_holder)
self.path1Interval.loop(playRate = 0.1)

self.path2 = Mopath.Mopath()
self.path2.loadFile('models/city/path2')
self.path2.fFaceForward = 1
self.path2Interval = MopathInterval(self.path2, self.monster2_holder)
self.path2Interval.loop(playRate = 0.1)
self.path4 = Mopath.Mopath()
self.path4.loadFile('models/city/path4_boss')
self.path4.fFaceForward = 1
self.path4Interval = MopathInterval(self.path4, self.monster4_holder)
self.path4Interval.loop(playRate = 0.1)
#end_paths

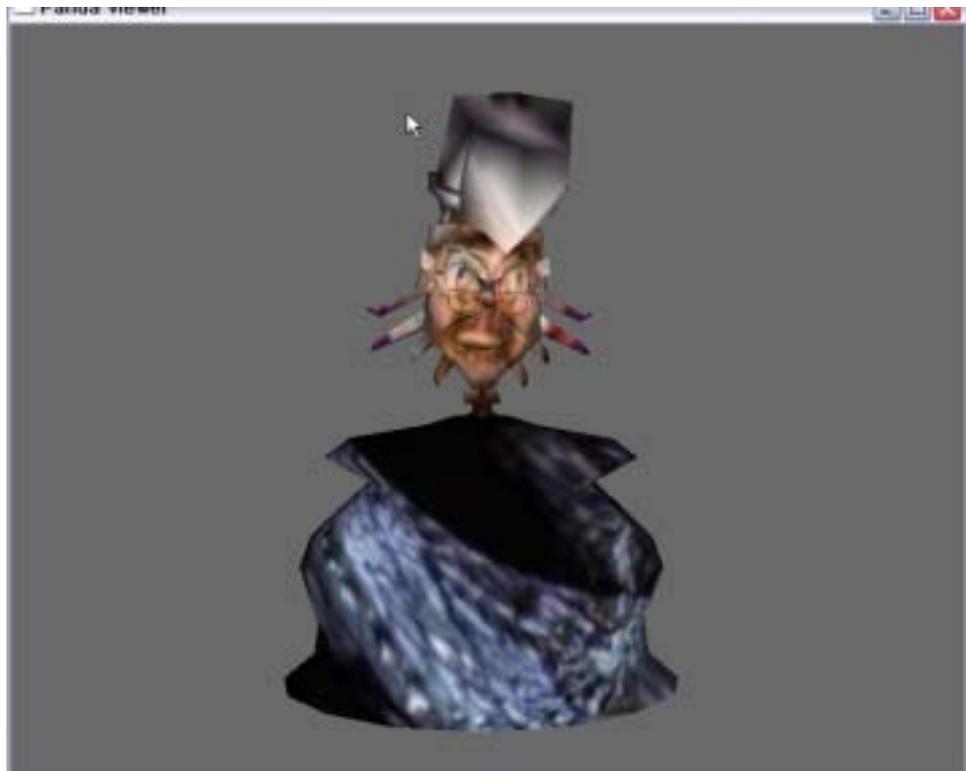
```



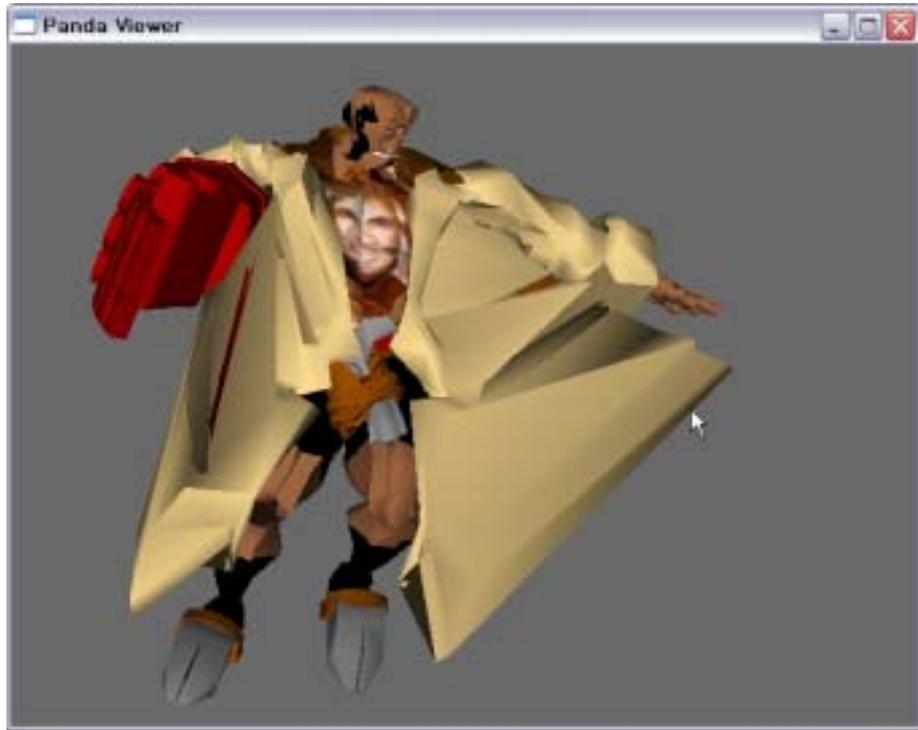
[그림 26] 여주인공 캐릭터 모델



[그림 27] 마법에서 풀린 남자 친구 캐릭터 모델



[그림 28] 리포트 몬스터 크리스



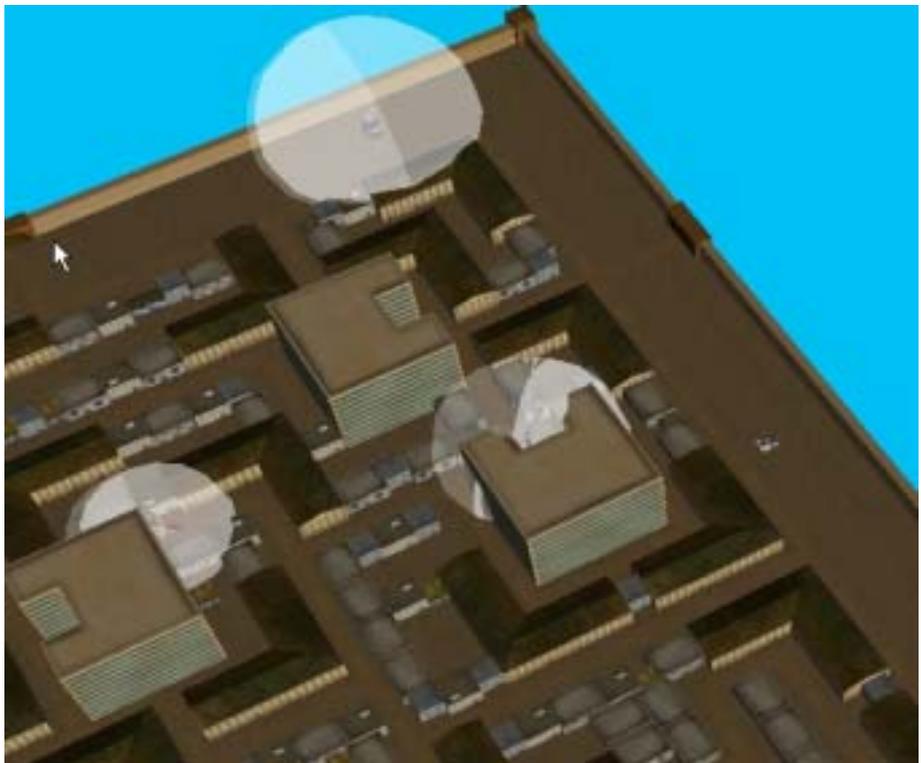
[그림 29] 카라오케 몬스터 란도



[그림 30] 대장 몬스터 단



[그림 31] 충돌체크를 하기 위한 구 의 모습



[그림 32] Mopath 설정에 따른 몬스터 모델의 움직임과 충돌체

## 크용 구의 모습

<표 3> 충돌체크를 위한 함수정의 부분

```
def setupCollisions(self):
    ##Collision Code##
    # use an event collision handler (sends events on collisions)
    self.cHandler = CollisionHandlerEvent()

    self.cHandler.setInPattern("%in into %fn")

    # make a traverser and make it the default traverser
    base.cTrav = CollisionTraverser()
    self.pusher = CollisionHandlerPusher()

    #setup a collision solid for the player
    cNode = CollisionNode('player')
    cNode.addSolid(CollisionSphere(0,0,2,1.5))
    cNode.setIntoCollideMask(BitMask32.bit(1))
    cNode.setFromCollideMask(BitMask32.bit(0))
    self.playerColl = self.player.attachNewNode(cNode)
    self.playerColl.show() #충돌체크를 위한 구의 모습을 보는 명령어
    base.cTrav.addCollider(self.playerColl, self.pusher)
    self.pusher.addCollider(self.playerColl,self.player, base.drive.node())

    #make a collision solid for monster1
    cNode = CollisionNode('monster1')
    cNode.addSolid(CollisionSphere(0,0,2,12))
    cNode.setIntoCollideMask(BitMask32.allOff())
    cNode.setFromCollideMask(BitMask32.bit(1))
    self.monster1Coll = self.monster1.attachNewNode(cNode)
    self.monster1Coll.show() #충돌체크를 위한 구의 모습을 보는 명령어
    base.cTrav.addCollider(self.monster1Coll , self.cHandler)
    self.accept('player into monster1', self.monster1Event)
```

위의 표에서 충돌체크를 위한 구의 모습을 확인 후 이 부분을 코멘트처리를 하면 구의 모습은 사라진다.

게임 인터페이스 작업 내용

- . Jam O Drum 조작 패드 디자인 작업: pan\_g.png, pan\_o.png, pan\_r.png, pan\_y.png



[그림 33] Jam O Drum 용 인터페이스 디자인 내용

### 18) Panda3D 용 프로그램 작성

작업 개요: 인터프리터 형식의 언어인 Python 을 기반으로 한 Panda3D 프로그램의 작성을 통하여 팀이 구현하고자 하는 내용을 실제로 구현하는 역할을 진행. 개발팀 전체가 Panda3D 는 물론이고 Python 에 처음 접하는 상황이었으므로 해당 언어 및 라이브러리의 원리 및 구조, 주요 특성 등을 이해하며 단계적으로 기능을 확장하는 방식으로 작업을 진행. 모델 및 애니메이션의 구동, 사운드 및 동영상 지원, Jam O Drum 인터페이스 프로그래밍, Task 운용, 기본 충돌 체크 등의 대부분의 기능은 비교적 성공적으로 프로그래밍을 완료하였으나 복합 속성의 다중적 충돌 체크 및 충돌 처리 기능에서 발생한 문제점으로 인하여 이와 관련된 Panda3D 의 문제점을 파악하고 이를 포럼에 재확인 시키는 한편, 일부 기 개발 기능을 대치하는 수준으로 프로젝트를 종료함

실제 프로그래밍 작업과 관련된 문제점 해결 사례 일부 예시  
 . Jam O Drum 인터페이스 프로그래밍: Jam O Drum 을 위한 skeleton 코드를 기반으로 프로그램을 작성 가능. 단, 프로그램 개발 단계에서는 마우스와 키보드를 활용하는 디버깅 작업이 가능하게 지원하고 있으나, 패드의 속성을 파악하기 위해서는 직접 Jam O Drum 장비에서의 테스트가 반드시 필요한 것으로 판단됨. 실제 본 프로젝트에서는 패드의 민감성 등을 고려하여 해당 부분과 관련된 내용을 define 문

형식으로 선언하여 프로젝트 매니저가 테스트를 할 수 있도록 하였음

<표 4> 테스트를 고려한 각종 변수 define 문의 일부

```

TURN_RATE = 90 #Degrees Turtle can turn in 1
second
ACCELERATION = 2 #Turtle acceleration in
units/sec/sec
MAX_VEL = 0.4 #Maximum Turtle velocity in
units/sec
MAX_VEL_SQ = MAX_VEL ** 2 #Square of the Turtle velocity
DEG_TO_RAD = pi/180 #translates degrees to radians for
sin and cos
GROUND_V = 0.01 # Ground Moving Velocity default
= 0.01
RADIOUS = 1.65
COL_DIST = 0.05

MIN_OBS_POS = 0.5
MAX_OBS_POS = 1.0

DOOLY_MOVE_SCALE = 0.45 # hit_Red only
DOOLY_DEC_SPEED = 0.85 # UpdatePosition Only
DOOLY_WIND_EFFECT = Vec3(0.20,0,0) # UpdatePosition Only

```

실제 낙하 효과를 위한 프로그래밍: 착각을 불러올 정도의 실감나는 낙하 효과의 반영을 위하여 위성 사진을 부착한 지형 모델이 고정된 카메라와 거북 캐릭터를 향해 다가오도록 프로그래밍 하는 방식으로 해결. 이 때 지형이 다가올 때 발생하는 해상도의 문제는 다중의 크기를 가지는 위성 사진 여러 장을 활용하여 문제점을 극복하였음. 특히, 이러한 파일 교체 시 발생하는 차이점을 극복하고 좀 더 리얼한 낙하 효과를 위해서는 구름 모델을 별도로 작성한 뒤 가속도와 알파 값 변화를 활용하여 구름이 가끔 다가오는 시각적 효과를 제공 하였음



[그림 34] 실제 낙하 효과를 주기 위하여 Google Earth 위성 사진을 다중 사이즈로 적용

- . Panda3D 에서의 모델 및 애니메이션 활용: View 가 고정되는 Jam O Drum 이 가지는 시각적 제한점을 극복하기 위하여 각 캐릭터 모델에 x, y, z 방향의 다양한 변화와 회전, 움직임을 부여한 애니메이션 효과를 추가하여 중력이 0인 낙하 현상을 재현하려 하였으며, 각종 효과를 위해 엄마 거북, 아가 거북, 장애물의 기본 모델과 지형 모델 외에 Jam O Drum 의 활용을 위한 각종 모델을 함께 사용하였으나 별 문제없이 구동됨을 확인하였음

<표 5> 예제를 통해 살펴본 Panda3D 에서의 모델 및 애니메이션의 로딩

```

self.g_babyturtle = Actor.Actor("model_a/little_turtle.egg",
                                {
"move": "model_a/little_turtle_move.egg",
"catch": "model_a/little_turtle_catch.egg",
"r_turn": "model_a/little_turtle_direct_right.egg",
"l_turn": "model_a/little_turtle_direct_left.egg",
"idle": "model_a/little_turtle_ani.egg"})
self.g_babyturtle_tex = loader.loadTexture("model_a/new_turtle_g.jpg")
self.g_babyturtle.setTexture(self.g_babyturtle_tex, 1)
self.g_babyturtle.reparentTo(render) #Attach it to render
self.g_babyturtle.setPosHpr(0.77, 0.77, 0, 225, 0, 0)
self.g_babyturtle.setScale(0.001)
self.g_babyturtle.velocity = Vec3(0,0,0) #Initial velocity
self.g_babyturtle.actorInterval("idle",playRate=0.5).loop()

```

- . Jam O Drum 마스크 및 팀 고유의 인터페이스 디자인 적용과 관련된 문제점 해결 방안: Jam O Drum 을 위한 skeleton 코드에서 기본적으로 제공하는 마스크 사용 방식은 이번 프로젝트와 같이 고유의 인터페이스 디자인을 적용하면서 동시에 다양한 모델과 텍스처를 중복된 좌표에서 고정된 카메라를 향해 이동시킬 경우 depth 문제 등 여러 형태의 비주얼한 문제가 발생할 수 있음을 확인하였음. 이러한 문제들은 카메라 위치 및 transparency 등의 속성값 지정을 통하여

## 해결함

<표 6> Jam O Drum 마스크 및 모델 간 좌표 값 중복에 따른 비주얼 문제 해결 방법의 예

```
self.visualPlane.remove()
#JOD mask, used for putting a black shell around the sides
# of the table, os projection aligns to table better. Can
# be removed for visual effect if desired

camera.setPos(0,0,10)
camera.setHpr(0, 90,0)

self.blackMask = loader.loadModel("models/jamodrummask")
self.blackMask.reparentTo(render)
self.blackMask.setBin("unsorted", 0)
self.blackMask.setDepthTest(False)
self.blackMask.setScale(.85)
self.blackMask.setColor(0,0,0)
self.blackMask.setPos(0,0,2)
self.blackMask.wrtReparentTo(camera)

self.pan_r = loader.loadModelCopy('model_a/plane')
self.pan_r_tex = loader.loadTexture('pic/pan_r.png')
self.pan_r.setTexture(self.pan_r_tex,1)
self.pan_r.setTransparency(1)
self.pan_r.setAlphaScale(0.5)
self.pan_r.reparentTo(render)
self.pan_r.setBin("unsorted", 0)
self.pan_r.setDepthTest(False)
self.pan_r.setPos( 1,0.98, 0.3)
self.pan_r.setHpr(0, 90,135)

# 중간 내용 생략

camera.setPos(0,0,5)
camera.setHpr(0, 90,0)

base.setBackgroundColor(0.383, 0.387, 0.25)
```

- 불필요한 리소스의 제거와 task manager 의 운용: 게임의 진행에 따라 더 이상 불필요해진 각종 리소스의 제거는 아주 간단한 방법으로 처리할 수 있으며, 게임의 진행 종료에 따른 동영상의 구동과 관련된 task manager 의 전환 역시 별다른 어려움 없이 구동 가능. 단, 프로그램 수준의 여러 개의 클래스를 정의하여 구동하는 방식의 예제나 개발 사례는 찾

아볼 수 없었음

<표 7> 프로그램 진행 중에서의 리소스 제거 및 task manager  
의 중단 예제

```
def unloadRest(self):  
    print "game is over from success, deleting everything"  
    self.endingTurtle.remove()  
    self.ground.remove()  
    self.clouds.remove()  
    taskMgr.remove('drop')  
    self.blackMask.remove()  
    self.pan_g.remove()  
    self.pan_y.remove()  
    self.pan_r.remove()  
    self.pan_o.remove()  
    self.successMovie()
```

## [참고]

### 1. 팀 A 프로젝트 소스

```
1 import direct.directbase.DirectStart
2 from direct.showbase.DirectObject import DirectObject
3 from direct.actor import Actor
4 from direct.interval.IntervalGlobal import *
5 from pandac.PandaModules import *
6 from direct.task import Task
7 from JOD.globals import *
8 from math import sin, cos, pi, sqrt
9
10 # Turtle Five Present
11 # 2005 08 10 14:30 Version
12
13 TURN_RATE = 90 #Degrees Turtle can turn in 1 second
14 ACCELERATION = 3 #Turtle acceleration in units/sec/sec
15 MAX_VEL = 0.3 #Maximum Turtle velocity in units/sec
16 MAX_VEL_SQ = MAX_VEL ** 2 #Square of the Turtle velocity
17 DEG_TO_RAD = pi/180 #translates degrees to radians for sin and cos
18 GROUND_V = 0.01 # Ground Moving Velocity default = 0.01
19 RADIOUS = 1.65
20 COL_DIST = 0.05
21 OBS_POS = 0.9
22
23 DOOLY_MOVE_SCALE = 0.5 # hit_Red only
24 DOOLY_DEC_SPEED = 0.9 # UpdatePosition Only
25 DOOLY_WIND_EFFECT = Vec3(0,0,0) # UpdatePosition Only
26
27 #Set flag for whether on the Jam O Drum or on a computer
28 #flag: 0=Computer, 1=Jam O Drum
29 flag = 1
30
31 #Set flag for hard driving with baby_turtles
32 #v_flag: 0=normal, 1=hard drive
33 v_flag = 0
34
35 if flag==0:
36     from JOD import drumpads
37     from JOD import spinners
38 else:
39     from JOD import drumpads_JOD
40     from JOD import spinners_JOD
41
42 class JamODrum(DirectObject):
43     def __init__(self):
44         #render.explore()
45         self.accept('escape', sys.exit, [0])
46         #read in calibration from config.prc
47         drLeft = base.config.GetFloat( 'display region left', 0.0 )
48         drRight = base.config.GetFloat( 'display region right', 1.0 )
49         drBottom = base.config.GetFloat( 'display region bottom', 0.0 )
50         drTop = base.config.GetFloat( 'display region top', 1.0 )
51         base.win.getDisplayRegion( 0 ).setDimensions( drLeft, drRight, drBottom, drTop )
52
53     if flag==0:
54         self.drumpadControls=drumpads.DrumPads()
55         self.spinnerControls=spinners.Spinners()
56     else:
57         self.drumpadControls=drumpads_JOD.DrumPads()
58         self.spinnerControls=spinners_JOD.Spinners()
59     self.drumpadControls.Start()
60     self.spinnerControls.Start()
61
62     #Hide mouse Cursor
63     props = base.win.getProperties()
64     props.setCursorHidden(1)
65     base.win.requestProperties(props)
66     #might need#
67     #base.disableMouse()
68
69     #place camera, moving can be bad for your projection, be careful
70     #camera.setPosHpr(0,0,10, 0, 90, 0)
71
72     #Jamo model, for table, keep
73     self.jamODrumModel = loader.loadModel("models/jamodrum")
74     #self.jamODrumModel.reparentTo(render)
```

```

75     self.jamODrumModel.setPos(0,0,0)
76     #self.jamODrumModel.wrtReparentTo(render)
77
78     #detect all hardware
79     self.spinner0 = self.jamODrumModel.find("**/spinner02")
80     self.spinner1 = self.jamODrumModel.find("**/spinner03")
81     self.spinner2 = self.jamODrumModel.find("**/spinner04")
82     self.spinner3 = self.jamODrumModel.find("**/spinner01")
83     self.pad0 = self.jamODrumModel.find("**/pad02")
84     self.pad1 = self.jamODrumModel.find("**/pad03")
85     self.pad2 = self.jamODrumModel.find("**/pad04")
86     self.pad3 = self.jamODrumModel.find("**/pad01")
87
88     # set up listening for hits and spinners
89     taskMgr.add(self.polling, "padAndSpinnerPoller")
90     for x in range (NUM_STATIONS):
91         temp="SPIN_"+str(x)+"_1"
92     #         self.accept(temp, self.spinCCW, [x])
93
94         temp="SPIN_"+str(x)+"_2"
95     #         self.accept(temp, self.spinCW, [x])
96
97         temp="HIT_"+str(x)
98     #         self.accept(temp, self.hit, [x])
99
100    #####
101    #####
102    #start your stuff from here                                     #
103    #####
104    #####
105        self.intro = Func(self.introMovie)
106        self.game = Func(self.game)
107        self.loadPlaySounds()
108        self.music.play()
109        Sequence(self.intro,
110                Wait(30),
111                self.game).start()
112
113    def game(self):
114        print "starting the game"
115        self.visualPlane.remove()
116        #JOD mask, used for putting a black shell around the sides
117        # of the table, os projection aligns to table better.  Can
118        # be removed for visual effect if desired
119
120        camera.setPos(0,0,10)
121        camera.setHpr(0, 90,0)
122
123        self.blackMask = loader.loadModel("models/jamodrummask")
124        self.blackMask.reparentTo(render)
125        self.blackMask.setBin("unsorted", 0)
126        self.blackMask.setDepthTest(False)
127        self.blackMask.setScale(.85)
128        self.blackMask.setColor(0,0,0)
129        self.blackMask.setPos(0,0,2)
130        self.blackMask.wrtReparentTo(camera)
131
132        camera.setPos(0,0,5)
133        camera.setHpr(0, 90,0)
134
135
136        base.setBackgroundColor(0.383, 0.387, 0.25)
137
138
139        # from here Indy code
140        self.loadGeometry()
141        self.loadTurtles()
142        self.initCollisions()
143        self.motherRandomMove()
144
145        self.dropTask = taskMgr.add(self.drop, 'drop')
146        self.dropTask.last = 0
147        self.dropTask.MISSION_COMPLETE = 0
148        self.g_hold = 0
149        self.r_hold = 0
150        self.b_hold = 0
151        self.y_hold = 0
152        self.hold_no = 0
153
154        ### "introMovie"
155        def introMovie(self):

```

```

156     print "intro movie"
157     base.setBackgroundColor(0, 0, 0)
158     camera.setPos(0,0,0)
159     camera.setHpr(0,0,0)
160     self.visualPlane = loader.loadModelCopy('model_a/plane')
161     self.visualPlane.setPosHprScale(0,8,0,0,0,0,5.2,1,4.0)
162     self.visualPlane.reparentTo(render)      #reparent to render
163
164     self.visualPlane.setTransparency(1)
165
166     self.introTexs = self.loadTextureMovie(27, 'video/intro/Slide',
167                                           'png', padding = 2)
168
169     self.introTask = taskMgr.add(self.textureMovie, "introTask")
170
171     self.introTask.fps = 1
172
173     self.introTask.obj = self.visualPlane
174
175     self.introTask.textures = self.introTexs
176
177     def textureMovie(self, task):
178         currentFrame = int(task.time * task.fps)
179         if currentFrame < len(task.textures):
180             task.obj.setTexture(task.textures[currentFrame],1)
181             if currentFrame == 1 :
182                 self.begin_music.play()
183             if currentFrame == 16 :
184                 self.ef_boys.play()
185             if currentFrame == 22 :
186                 self.LetsCatch.play()
187
188         else :
189             taskMgr.remove(task)
190
191     return Task.cont
192
193     def loadTextureMovie(self, frames, name, suffix, padding = 1):
194         return [loader.loadTexture((name+"%0"+str(padding)+"d."+suffix) % i)
195               for i in range(frames)]
196
197     ### "successMovie"
198     def successMovie(self):
199         print "success movie"
200         base.setBackgroundColor(0, 0, 0)
201         camera.setPos(0,0,0)
202         camera.setHpr(0,0,0)
203         self.visualPlane = loader.loadModelCopy('model_a/plane')
204         self.visualPlane.setPosHprScale(0,8,0,0,0,0,5.2,1,4.0)
205         self.visualPlane.reparentTo(render)      #reparent to render
206
207         self.visualPlane.setTransparency(1)
208
209         self.successTexs = self.loadTextureMovie(3, 'video/good/Slide',
210                                                 'png', padding = 2)
211
212         self.successTask = taskMgr.add(self.successTextureMovie, "successTask")
213
214         self.successTask.fps = 0.25
215
216         self.successTask.obj = self.visualPlane
217
218         self.successTask.textures = self.successTexs
219
220     def successTextureMovie(self, task):
221         currentFrame = int(task.time * task.fps)
222         if currentFrame < len(task.textures):
223             task.obj.setTexture(task.textures[currentFrame],1)
224             if currentFrame == 1 :
225                 self.ef_cheering.play()
226             if currentFrame == 2 :
227                 self.ending_music.stop()
228                 self.credit_music.play()
229         else :
230             taskMgr.remove(task)
231
232     return Task.cont
233
234     ### "failMovie"
235     def failMovie(self):
236         print "fail movie"

```

```

237     base.setBackgroundColor(0, 0, 0)
238     camera.setPos(0,0,0)
239     camera.setHpr(0,0,0)
240     self.visualPlane = loader.loadModelCopy('model_a/plane')
241     self.visualPlane.setPosHprScale(0,8,0,0,0,0,5,2,1,4,0)
242     self.visualPlane.reparentTo(render)      #reparent to render
243
244     self.visualPlane.setTransparency(1)
245
246     self.failTexs = self.loadTextureMovie(3, 'video/fail/Slide',
247                                         'png', padding = 2)
248
249     self.failTask = taskMgr.add(self.failTextureMovie, "failTask")
250
251     self.failTask.fps = 0.25
252
253     self.failTask.obj = self.visualPlane
254
255     self.failTask.textures = self.failTexs
256
257 def failTextureMovie(self, task):
258     currentFrame = int(task.time * task.fps)
259     if currentFrame < len(task.textures):
260         task.obj.setTexture(task.textures[currentFrame],1)
261         if currentFrame == 1 :
262             self.ef_crash.play()
263         if currentFrame == 2 :
264             self.credit_music.play()
265     else :
266         taskMgr.remove(task)
267
268     return Task.cont
269
270
271 def loadPlaySounds(self):
272     #Loading the main music box song
273     self.music = loader.loadMusic('sound/play.mp3')
274     self.music.setVolume(.7) #Volume is a percentage from 0 to 1
275     self.music.setLoopCount(1) #0 means loop forever, 1 (default) means
276                               #play once. 2 or higher means play that
277                               #many times
278     self.begin_music = loader.loadMusic('sound/begin.mp3')
279     self.begin_music.setVolume(.7) #Volume is a percentage from 0 to 1
280     self.begin_music.setLoopCount(1)
281
282     self.ending_music = loader.loadMusic('sound/end.mp3')
283     self.ending_music.setVolume(.7) #Volume is a percentage from 0 to 1
284     self.ending_music.setLoopCount(1)
285
286     self.credit_music = loader.loadMusic('sound/credit.mp3')
287     self.credit_music.setVolume(.7) #Volume is a percentage from 0 to 1
288     self.credit_music.setLoopCount(1)
289
290     #Loading the open/close effect
291     #loadSFX and loadMusic are identical. They are often used for organization
292     #(loadMusic is used for background music, loadSfx is used for other effects)
293     self.ef_akmom = loader.loadSfx('sound/ef_akmom.mp3')
294     self.ef_akmom.setVolume(1)
295     self.ef_akboys = loader.loadSfx('sound/ef_akboys.mp3')
296     self.ef_akboys.setVolume(1)
297     self.ef_boys = loader.loadSfx('sound/ef_boys.mp3')
298     self.ef_boys.setVolume(1)
299     self.ef_cheering = loader.loadSfx('sound/ef_cheering.mp3')
300     self.ef_cheering.setVolume(1)
301     self.ef_crash = loader.loadSfx('sound/ef_crash.mp3')
302     self.ef_crash.setVolume(1)
303     self.ef_help = loader.loadSfx('sound/ef_help.mp3')
304     self.ef_help.setVolume(1)
305     self.ef_push = loader.loadSfx('sound/ef_push.mp3')
306     self.ef_push.setVolume(1)
307     self.ef_spin = loader.loadSfx('sound/ef_spin.mp3')
308     self.ef_spin.setVolume(1)
309     self.ef_akjang = loader.loadSfx('sound/ef_akjang.mp3')
310     self.ef_akjang.setVolume(1)
311     self.ef_akkim = loader.loadSfx('sound/ef_akkim.mp3')
312     self.ef_akkim.setVolume(1)
313     self.ef_aklee = loader.loadSfx('sound/ef_aklee.mp3')
314     self.ef_aklee.setVolume(1)
315     self.ef_aksong = loader.loadSfx('sound/ef_aksong.mp3')
316     self.ef_aksong.setVolume(1)
317     self.ef_hold = loader.loadSfx('sound/ef_hold.mp3')

```

```

318         self.ef_hold.setVolume(1)
319         self.LetsCatch = loader.loadSfx('sound/LetsCatch.mp3')
320         self.LetsCatch.setVolume(1)
321
322         #The open/close file has both effects in it. Fortunately we can use intervals
323         #to easily define parts of a sound file to play
324         #self.lidOpenSfx = SoundInterval(self.lidSfx, startTime = 0)
325         ##### end of the "loadPlaySounds(self)"
326
327     def playSound(self, sound):
328         sound.play()
329
330     def unloadPlaySounds(self):
331         #Loading the main music box song
332         self.ef_boys = loader.loadSfx('sound/ef_boys.mp3')
333         self.ef_boys.setVolume(1)
334         self.ef_cheering = loader.loadSfx('sound/ef_cheering.mp3')
335         self.ef_cheering.setVolume(1)
336
337     # How can make Music stop
338     self.music.stop()
339
340     self.ending_music.play()
341
342     # self.music.remove()
343     # self.ef_akmom.remove()
344     # self.ef_akboys.remove()
345     # self.ef_push.remove()
346     # self.ef_spin.remove()
347
348     ##### end of the "unloadPlaySounds(self)"
349
350     def loadGeometry(self):
351
352         self.ground = loader.loadModel("models/ground")
353         self.ground.reparentTo(render)
354         self.ground.setPos(0,0, 80)
355         self.ground.velocity = 0
356
357         self.clouds = loader.loadModel("models/clouds")
358         self.clouds.reparentTo(render)
359         self.clouds.setPos(0,0, 40)
360         self.clouds.setScale(1.2, 1.2, 1)
361         #self.clouds.wrtReparentTo(camera)
362
363         self.obstacles = loader.loadModelCopy("model_a/boo")
364         self.obstacles_tex = loader.loadTexture("model_a/boo_R.jpg")
365         self.obstacles.setTexture(self.obstacles_tex, 1)
366         self.obstacles.reparentTo(self.clouds)
367         self.obstacles.setScale(0.005)
368         self.obstacles.setPos(random.uniform( 0.8, 0.8), random.uniform( 0.8, 0.8), 7)
369
370         self.obstacles_1 = loader.loadModelCopy("model_a/boo")
371         self.obstacles_1_tex = loader.loadTexture("model_a/boo_B.jpg")
372         self.obstacles_1.setTexture(self.obstacles_1_tex, 1)
373         self.obstacles_1.reparentTo(self.clouds)
374         self.obstacles_1.setScale(0.005)
375         self.obstacles_1.setPos(random.uniform( 0.8, 0.8), random.uniform( 0.8, 0.8), 10)
376
377         self.obstacles_2 = loader.loadModelCopy("model_a/boo")
378         self.obstacles_2_tex = loader.loadTexture("model_a/boo_G.jpg")
379         self.obstacles_2.setTexture(self.obstacles_2_tex, 1)
380         self.obstacles_2.reparentTo(self.clouds)
381         self.obstacles_2.setScale(0.005)
382         self.obstacles_2.setPos(random.uniform( 0.8, 0.8), random.uniform( 0.8, 0.8), 12)
383
384         self.obstacles_3 = loader.loadModelCopy("model_a/boo")
385         self.obstacles_3_tex = loader.loadTexture("model_a/boo_BB.jpg")
386         self.obstacles_3.setTexture(self.obstacles_3_tex, 1)
387         self.obstacles_3.reparentTo(self.clouds)
388         self.obstacles_3.setScale(0.005)
389         self.obstacles_3.setPos(random.uniform( 0.8, 0.8), random.uniform( 0.8, 0.8), 6)
390
391         self.moveClouds = self.clouds.posHprInterval(5, Vec3(0,0,6), Vec3(15,0,0))
392         self.alphaClouds = self.clouds.colorInterval(1, Vec4(1,1,1,1), Vec4(1,1,1,0))
393
394         self.change_Obstacles_Position = Func(self.chgObstaclesPos)
395
396         self.loopClouds = Parallel(self.moveClouds,
397                                   self.alphaClouds,

```

```

398                                     self.change_Obstacles_Position )
399     self.loopClouds.loop()
400
401     ##### end of the "loadTurtles(self)"
402
403     def chgObstaclesPos(self):
404         self.obstacles.setPos(random.uniform( OBS_POS, OBS_POS), random.uniform( OBS_POS, OBS_POS),
7)
405         self.obstacles_1.setPos(random.uniform( OBS_POS, OBS_POS), random.uniform( OBS_POS, OBS_POS),
10)
406         self.obstacles_2.setPos(random.uniform( OBS_POS, OBS_POS), random.uniform( OBS_POS, OBS_POS),
12)
407         self.obstacles_3.setPos(random.uniform( OBS_POS, OBS_POS), random.uniform( OBS_POS, OBS_POS),
6)
408     ##### end of the "chgObstaclesPos(self)"
409
410     def unloadTurtles(self):
411         self.momturtle.remove()
412         # This is for Baby Turtles
413         self.r_babyturtle.remove()
414         self.g_babyturtle.remove()
415         self.b_babyturtle.remove()
416         self.y_babyturtle.remove()
417
418         self.ignore("SPIN_0_1")
419         self.ignore("SPIN_1_1")
420         self.ignore("SPIN_2_1")
421         self.ignore("SPIN_3_1")
422         self.ignore("SPIN_0_2")
423         self.ignore("SPIN_1_2")
424         self.ignore("SPIN_2_2")
425         self.ignore("SPIN_3_2")
426         self.ignore("HIT_0")
427         self.ignore("HIT_1")
428         self.ignore("HIT_2")
429         self.ignore("HIT_3")
430
431         self.mommmove_random.finish()
432
433     ##### end of the "unloadTurtles(self)"
434
435     def loadRescueTeam(self):
436         # This is for Family Model
437         self.unloadPlaySounds()
438         self.endingTurtle.reparentTo(render) #Attach it to render
439
440         self.gameOver = Func(self.unloadRest)
441
442         Sequence(self.endAnim,
443                 self.gameOver).start()
444     # self.dropTask.MISSION_COMPLETE = 1
445     # self.endingTurtle.remove()
446     # taskMgr.remove('drop')
447     ##### end of the "loadRescueTeam(self)"
448
449     def unloadRest(self):
450         print "game is over from success, deleting everything"
451         self.endingTurtle.remove()
452         self.ground.remove()
453         self.clouds.remove()
454         taskMgr.remove('drop')
455         self.blackMask.remove()
456     # self.music.stop()
457     # self.successMovie()
458     # self.ending_music.play()
459
460     def unloadRest2(self):
461         print "game is over from fail, deleting everything"
462         self.endingTurtle.remove()
463         self.ground.remove()
464         self.clouds.remove()
465         taskMgr.remove('drop')
466         self.blackMask.remove()
467         self.music.stop()
468         self.failMovie()
469         self.credit_music.play()
470
471     def loadTurtles(self):
472         # This is for Mother Turtle
473         self.momturtle = Actor.Actor("model_a/mother_turtle",
474                                     {"idle":"model_a/mother_turtle_ani.egg"})

```

```

475 self.momturtle.reparentTo(render) #Attach it to render
476 self.momturtle.setPosHpr(0, 0, 0, 180, 0, 0)
477 self.momturtle.setScale(0.001)
478
479 self.momturtle.actorInterval("idle",playRate=0.5).loop()
480
481 # This is for Baby Turtles
482 self.r_babyturtle = Actor.Actor("model_a/little_turtle.egg",
483 {
484     "move":"model_a/little_turtle_move.egg",
485     "catch":"model_a/little_turtle_catch.egg",
486     "r_turn":"model_a/little_turtle_direct_right.egg",
487     "l_turn":"model_a/little_turtle_direct_left.egg",
488     "idle":"model_a/little_turtle_ani.egg"})
489 self.r_babyturtle_tex = loader.loadTexture("model_a/new_turtle_r.jpg")
490 self.r_babyturtle.setTexture(self.r_babyturtle_tex, 1)
491 self.r_babyturtle.reparentTo(render) #Attach it to render
492 self.r_babyturtle.setPosHpr( 0.88, 0.88, 0, 45, 0, 0)
493 self.r_babyturtle.setScale(0.001)
494 self.r_babyturtle.velocity = Vec3(0,0,0) #Initial velocity
495 self.r_babyturtle.actorInterval("idle",playRate=0.5).loop()
496
497 self.g_babyturtle = Actor.Actor("model_a/little_turtle.egg",
498 {
499     "move":"model_a/little_turtle_move.egg",
500     "catch":"model_a/little_turtle_catch.egg",
501     "r_turn":"model_a/little_turtle_direct_right.egg",
502     "l_turn":"model_a/little_turtle_direct_left.egg",
503     "idle":"model_a/little_turtle_ani.egg"})
504 self.g_babyturtle_tex = loader.loadTexture("model_a/new_turtle_g.jpg")
505 self.g_babyturtle.setTexture(self.g_babyturtle_tex, 1)
506 self.g_babyturtle.reparentTo(render) #Attach it to render
507 self.g_babyturtle.setPosHpr(0.88, 0.88, 0, 225, 0, 0)
508 self.g_babyturtle.setScale(0.001)
509 self.g_babyturtle.velocity = Vec3(0,0,0) #Initial velocity
510 self.g_babyturtle.actorInterval("idle",playRate=0.5).loop()
511
512 self.b_babyturtle = Actor.Actor("model_a/little_turtle.egg",
513 {
514     "move":"model_a/little_turtle_move.egg",
515     "catch":"model_a/little_turtle_catch.egg",
516     "r_turn":"model_a/little_turtle_direct_right.egg",
517     "l_turn":"model_a/little_turtle_direct_left.egg",
518     "idle":"model_a/little_turtle_ani.egg"})
519 self.b_babyturtle_tex = loader.loadTexture("model_a/new_turtle_o.jpg")
520 self.b_babyturtle.setTexture(self.b_babyturtle_tex, 1)
521 self.b_babyturtle.reparentTo(render) #Attach it to render
522 self.b_babyturtle.setPosHpr( 0.88, 0.88, 0, 135, 0, 0)
523 self.b_babyturtle.setScale(0.001)
524 self.b_babyturtle.velocity = Vec3(0,0,0) #Initial velocity
525 self.b_babyturtle.actorInterval("idle",playRate=0.5).loop()
526
527 self.y_babyturtle = Actor.Actor("model_a/little_turtle.egg",
528 {
529     "move":"model_a/little_turtle_move.egg",
530     "catch":"model_a/little_turtle_catch.egg",
531     "r_turn":"model_a/little_turtle_direct_right.egg",
532     "l_turn":"model_a/little_turtle_direct_left.egg",
533     "idle":"model_a/little_turtle_ani.egg"})
534
535 self.y_babyturtle_tex = loader.loadTexture("model_a/new_turtle_y.jpg")
536 self.y_babyturtle.setTexture(self.y_babyturtle_tex, 1)
537 self.y_babyturtle.reparentTo(render) #Attach it to render
538 self.y_babyturtle.setPosHpr(0.88, 0.88, 0, 315, 0, 0)
539 self.y_babyturtle.setScale(0.001)
540 self.y_babyturtle.velocity = Vec3(0,0,0) #Initial velocity
541 self.y_babyturtle.actorInterval("idle",playRate=0.5).loop()
542
543 self.endingTurtle = Actor.Actor("model_a/test.egg",
544 {"ending":"model_a/test_ani.egg"})
545 self.endingTurtle.setPosHpr(0, 0, 0, 180, 0, 0)
546 self.endingTurtle.setScale(0.001)
547 self.endAnim = self.endingTurtle.actorInterval("ending", playRate=0.1)
548
549 #SPIN_X_Y: X=station number (0:green,1:blue,2:red,3:yellow)
550 # Y=direction(1:left,2:right)

```

```

551 self.accept("SPIN_0_1",self.leftGreen)
552 self.accept("SPIN_1_1",self.leftBlue)
553 self.accept("SPIN_2_1",self.leftRed)
554 self.accept("SPIN_3_1",self.leftYellow)
555 self.accept("SPIN_0_2",self.rightGreen)
556 self.accept("SPIN_1_2",self.rightBlue)
557 self.accept("SPIN_2_2",self.rightRed)
558 self.accept("SPIN_3_2",self.rightYellow)
559 self.accept("HIT_0",self.hitGreen)
560 self.accept("HIT_1",self.hitBlue)
561 self.accept("HIT_2",self.hitRed)
562 self.accept("HIT_3",self.hitYellow)
563
564 ##### end of the "loadTurtles(self)"
565
566 def motherRandomMove(self):
567     self.momturtle.rmove = self.momturtle.posInterval(8, Point3(0.2, 0.4, 0))
568     self.momturtle.lmove = self.momturtle.posInterval(5, Point3( 0.4, 0.4, 0))
569     self.momturtle.umove = self.momturtle.posInterval(3, Point3(0, 0.5, 0))
570     self.momturtle.dmove = self.momturtle.posInterval(6, Point3(0, 0, 0))
571
572     self.momturtle.xspin = self.momturtle.hprInterval(3, Vec3(180, 0, 0))
573     self.momturtle.yspin = self.momturtle.hprInterval(5, Vec3(0, 360, 0))
574     self.momturtle.zspin = self.momturtle.hprInterval(2, Vec3(0, 0, 900))
575
576     self.playSound_akmom = Func(self.playSound, self.ef_akmom)
577     self.playSound_help = Func(self.playSound, self.ef_help)
578
579     self.mommmove_random = Sequence(self.momturtle.rmove,
580                                     self.playSound_akmom,
581                                     self.momturtle.yspin,
582                                     Wait(3),
583                                     self.momturtle.lmove,
584                                     Wait(5),
585                                     self.playSound_akmom,
586                                     self.momturtle.zspin,
587                                     self.momturtle.umove,
588                                     self.momturtle.zspin,
589                                     self.momturtle.dmove,
590                                     Wait(1),
591                                     self.playSound_help,
592                                     self.momturtle.xspin)
593     self.mommmove_random.loop()
594 ##### end of the "motherRandomMove(self)"
595
596 def initCollisions(self):
597
598     self.cHandler = CollisionHandlerEvent()
599     self.cHandler.setInPattern("crash %in")
600
601     # This traverser will automatically be run every frame.
602     base.cTrav = CollisionTraverser()
603     self.pusher = CollisionHandlerPusher()
604
605     # from here, indy's code
606     #setup a collision solid for the camera/player
607
608     # Mother Turtle
609     self.cNode = CollisionNode('momturtle')
610     self.cNode.addSolid(CollisionSphere(0,0,0,100))
611     self.cNode.setIntoCollideMask(BitMask32.bit(1))
612     self.cNode.setFromCollideMask(BitMask32.allOff())
613     self.momturtle_coll = self.momturtle.attachNewNode(self.cNode)
614 #     self.momturtle_coll.hide()
615
616     # Baby Turtles
617     self.cNode = CollisionNode('y_babyturtle')
618     self.cNode.addSolid(CollisionSphere(0,0,0,100))
619     self.cNode.setIntoCollideMask(BitMask32.bit(1))
620     self.cNode.setFromCollideMask(BitMask32.allOff())
621     self.y_babyturtle_coll = self.y_babyturtle.attachNewNode(self.cNode)
622
623     self.cNode = CollisionNode('r_babyturtle')
624     self.cNode.addSolid(CollisionSphere(0,0,0,100))
625     self.cNode.setIntoCollideMask(BitMask32.bit(1))
626     self.cNode.setFromCollideMask(BitMask32.allOff())
627     self.r_babyturtle_coll = self.r_babyturtle.attachNewNode(self.cNode)
628
629     self.cNode = CollisionNode('g_babyturtle')
630     self.cNode.addSolid(CollisionSphere(0,0,0,100))
631     self.cNode.setIntoCollideMask(BitMask32.bit(1))

```

```

632 self.cNode.setFromCollideMask(BitMask32.allOff())
633 self.g_babyturtle_coll = self.g_babyturtle.attachNewNode(self.cNode)
634
635 self.cNode = CollisionNode('b_babyturtle')
636 self.cNode.addSolid(CollisionSphere(0,0,0,100))
637 self.cNode.setIntoCollideMask(BitMask32.bit(1))
638 self.cNode.setFromCollideMask(BitMask32.allOff())
639 self.b_babyturtle_coll = self.b_babyturtle.attachNewNode(self.cNode)
640
641 # Baby Turtles2 pusher
642 self.cNode = CollisionNode('y_babyturtle')
643 self.cNode.addSolid(CollisionSphere(0,0,0,100))
644 self.cNode.setIntoCollideMask(BitMask32.bit(3))
645 self.cNode.setFromCollideMask(BitMask32.bit(3))
646 self.y_babyturtle_coll_push = self.y_babyturtle.attachNewNode(self.cNode)
647
648 self.cNode = CollisionNode('r_babyturtle')
649 self.cNode.addSolid(CollisionSphere(0,0,0,100))
650 self.cNode.setIntoCollideMask(BitMask32.bit(3))
651 self.cNode.setFromCollideMask(BitMask32.bit(3))
652 self.r_babyturtle_coll_push = self.r_babyturtle.attachNewNode(self.cNode)
653
654 self.cNode = CollisionNode('g_babyturtle')
655 self.cNode.addSolid(CollisionSphere(0,0,0,100))
656 self.cNode.setIntoCollideMask(BitMask32.bit(3))
657 self.cNode.setFromCollideMask(BitMask32.bit(3))
658 self.g_babyturtle_coll_push = self.g_babyturtle.attachNewNode(self.cNode)
659
660 self.cNode = CollisionNode('b_babyturtle')
661 self.cNode.addSolid(CollisionSphere(0,0,0,100))
662 self.cNode.setIntoCollideMask(BitMask32.bit(3))
663 self.cNode.setFromCollideMask(BitMask32.bit(3))
664 self.b_babyturtle_coll_push = self.b_babyturtle.attachNewNode(self.cNode)
665
666 # Obstacles (Four)
667 self.cNode = CollisionNode('obstacles')
668 self.cNode.addSolid(CollisionSphere(0,0,0,30))
669 self.cNode.setIntoCollideMask(BitMask32.allOff())
670 self.cNode.setFromCollideMask(BitMask32.bit(1))
671 self.obstacles_coll = self.obstacles.attachNewNode(self.cNode)
672 # self.obstacles_coll.show()
673
674 self.cNode = CollisionNode('obstacles_1')
675 self.cNode.addSolid(CollisionSphere(0,0,0,30))
676 self.cNode.setIntoCollideMask(BitMask32.allOff())
677 self.cNode.setFromCollideMask(BitMask32.bit(1))
678 self.obstacles_1_coll = self.obstacles_1.attachNewNode(self.cNode)
679 # self.obstacles_1_coll.show()
680
681 self.cNode = CollisionNode('obstacles_2')
682 self.cNode.addSolid(CollisionSphere(0,0,0,30))
683 self.cNode.setIntoCollideMask(BitMask32.allOff())
684 self.cNode.setFromCollideMask(BitMask32.bit(1))
685 self.obstacles_2_coll = self.obstacles_2.attachNewNode(self.cNode)
686 # self.obstacles_2_coll.show()
687
688 self.cNode = CollisionNode('obstacles_3')
689 self.cNode.addSolid(CollisionSphere(0,0,0,30))
690 self.cNode.setIntoCollideMask(BitMask32.allOff())
691 self.cNode.setFromCollideMask(BitMask32.bit(1))
692 self.obstacles_3_coll = self.obstacles_3.attachNewNode(self.cNode)
693 # self.obstacles_3_coll.show()
694
695 #set up bitmasks
696 #this object can only collide into things
697
698 #add this object to the traverser
699 # base.cTrav.addCollider(self.wall.node(), self.actEvent)
700
701 base.cTrav.addCollider(self.momturtle_coll, self.cHandler)
702 # self.pusher.addCollider(self.momturtle_coll, self.momturtle, base.drive.node())
703
704 base.cTrav.addCollider(self.obstacles_coll, self.cHandler)
705 base.cTrav.addCollider(self.obstacles_1_coll, self.cHandler)
706 base.cTrav.addCollider(self.obstacles_2_coll, self.cHandler)
707 base.cTrav.addCollider(self.obstacles_3_coll, self.cHandler)
708
709 #base.cTrav.addCollider(self.y_babyturtle_coll, self.cHandler)
710 #base.cTrav.addCollider(self.r_babyturtle_coll, self.cHandler)
711 #base.cTrav.addCollider(self.g_babyturtle_coll, self.cHandler)
712 #base.cTrav.addCollider(self.b_babyturtle_coll, self.cHandler)

```

```

713
714     self.accept("crash_r_babyturtle", self.crash_r)
715     self.accept("crash_g_babyturtle", self.crash_g)
716     self.accept("crash_b_babyturtle", self.crash_b)
717     self.accept("crash_y_babyturtle", self.crash_y)
718     self.accept("crash_momturtle", self.crash_m)
719
720
721     ## #add this object to the traverser
722     ## #
723     base.cTrav.addCollider(self.wall.node(), self.actEvent)
724
725     ## base.cTrav.addCollider(self.momturtle_coll.node(), self.pusher)
726     ## self.pusher.addCollider(self.momturtle_coll, self.momturtle, base.drive.node())
727
728     ## base.cTrav.addCollider(self.obstacles_coll.node(), self.pusher)
729     ## base.cTrav.addCollider(self.obstacles_1_coll.node(), self.pusher)
730     ## base.cTrav.addCollider(self.obstacles_2_coll.node(), self.pusher)
731     ## base.cTrav.addCollider(self.obstacles_3_coll.node(), self.pusher)
732
733     base.cTrav.addCollider(self.y_babyturtle_coll_push, self.pusher)
734     self.pusher.addCollider(self.y_babyturtle_coll_push, self.y_babyturtle, base.drive.node())
735     base.cTrav.addCollider(self.r_babyturtle_coll_push, self.pusher)
736     self.pusher.addCollider(self.r_babyturtle_coll_push, self.r_babyturtle, base.drive.node())
737     base.cTrav.addCollider(self.g_babyturtle_coll_push, self.pusher)
738     self.pusher.addCollider(self.g_babyturtle_coll_push, self.g_babyturtle, base.drive.node())
739     base.cTrav.addCollider(self.b_babyturtle_coll_push, self.pusher)
740     self.pusher.addCollider(self.b_babyturtle_coll_push, self.b_babyturtle, base.drive.node())
741
742     ##End Collision Code##
743
744     #Updates the positions of objects
745     def crash_r(self, cEntry):
746         print "entering crash_r"
747         self.r_babyturtle.reparentTo(render)
748         self.ef_akjang.play()
749         self.r_hold = 0
750         self.r_babyturtle.velocity = Vec3(0,0,0)
751         self.r_babyturtle.posHprInterval(3, Vec3( 0.88, 0.88, 0), Vec3(765, 720, 720)).start()
752     #
753     self.r_babyturtle.setPosHpr( 0.88, 0.88, 0, 45, 0, 0)
754     ##### end of the "crash_r(self)"
755
756     #The Work to do when Mother and Baby make collision
757     #Updates the positions of objects
758     def crash_g(self, cEntry):
759         print "entering crash_g"
760         self.g_babyturtle.reparentTo(render)
761         self.ef_akkim.play()
762         self.g_hold = 0
763         self.g_babyturtle.velocity = Vec3(0,0,0)
764         self.g_babyturtle.posHprInterval(3, Vec3(0.88, 0.88, 0), Vec3(225, 720, 720)).start()
765     ##### end of the "crash_g(self)"
766
767     #The Work to do when obstacl and Baby make collision
768     #Updates the positions of objects
769     def crash_b(self, cEntry):
770         print "entering crash_b"
771         self.b_babyturtle.reparentTo(render)
772         self.b_hold = 0
773         self.ef_aklee.play()
774         self.b_babyturtle.velocity = Vec3(0,0,0)
775         self.b_babyturtle.posHprInterval(3, Vec3( 0.88, 0.88, 0), Vec3(135, 720, 720)).start()
776     #
777         self.b_babyturtle.setPosHpr(8, Point3( 0.88, 0.88, 0)).start()
778     ##### end of the "crash_r(self)"
779
780     #The Work to do when obstacl and Baby make collision
781     #Updates the positions of objects
782     def crash_y(self, cEntry):
783         print "entering crash_y"
784         self.y_babyturtle.reparentTo(render)
785         self.y_hold = 0
786         self.ef_aksong.play()
787         self.y_babyturtle.velocity = Vec3(0,0,0)
788         self.y_babyturtle.posHprInterval(3, Vec3(0.88, 0.88, 0), Vec3(315, 720, 720)).start()
789     #
790         self.y_babyturtle.setPosHpr(0.88, 0.88, 0, 315, 0, 0)
791     ##### end of the "crash_y(self)"
792
793     def crash_m(self, cEntry):

```

```

793     print "entering crash_m"
794     self.ef_akmom.play()
795
796     """
797     self.r_babyturtle.reparentTo(render)
798     self.g_babyturtle.reparentTo(render)
799     self.b_babyturtle.reparentTo(render)
800     self.y_babyturtle.reparentTo(render)
801
802     if ( self.g_hold == 1 ) :
803         self.g_babyturtle.velocity = Vec3(0,0,0)
804         self.g_babyturtle.posHprInterval(3, Vec3(0.88, 0.88, 0), Vec3(225, 720, 720)).start()
805     if ( self.r_hold == 1 ) :
806         self.r_babyturtle.velocity = Vec3(0,0,0)
807         self.r_babyturtle.posHprInterval(3, Vec3( 0.88, 0.88, 0), Vec3(45, 720, 720)).start()
808     if ( self.b_hold == 1 ) :
809         self.b_babyturtle.velocity = Vec3(0,0,0)
810         self.b_babyturtle.posHprInterval(3, Vec3( 0.88, 0.88, 0), Vec3(135, 720, 720)).start()
811     if ( self.y_hold == 1 ) :
812         self.y_babyturtle.velocity = Vec3(0,0,0)
813         self.y_babyturtle.posHprInterval(3, Vec3(0.88, 0.88, 0), Vec3(315, 720, 720)).start()
814
815     """
816     self.g_hold = 0
817     self.r_hold = 0
818     self.b_hold = 0
819     self.y_hold = 0
820
821     self.hold_no = 0
822     self.momturtle.posHprInterval(3, Vec3(0, 0, 0), Vec3(180, 0, 0)).start()
823
824     ##### end of the "crash_r(self)"
825
826     #Updates the positions of objects
827     def updatePos(self, obj, dt):
828         newPos = Vec3(obj.getPos()) + Point3(obj.velocity*dt)
829
830         if newPos.lengthSquared() < RADIOUS:
831             # newPos = newPos + (DOOLY_WIND_EFFECT * dt)
832             # obj.velocity = obj.velocity * DOOLY_DEC_SPEED
833             obj.setPos(newPos)
834
835         ##### end of the "updatePos(self, obj, dt)"
836
837     #This updates the Turtle's position.
838     def updateTurtle(self, dt):
839         #Finally, update the position as with any other object
840         self.updatePos(self.r_babyturtle, dt)
841         self.updatePos(self.g_babyturtle, dt)
842         self.updatePos(self.b_babyturtle, dt)
843         self.updatePos(self.y_babyturtle, dt)
844         ##### end of the "updateTurtle(self, dt)"
845
846     #Process hold momturtle
847     def holdMom(self, obj):
848         obj.velocity = 0
849         self.ef_hold.play()
850         obj.wrtReparentTo(self.momturtle)
851         # self.hold_move = Sequence(obj.posInterval(0.5, Point3(0, 0, 2), obj.getPos()),
852         # obj.hprInterval(0.5, Vec3(180, 0, 0)),
853         # obj.posInterval(0.5, Point3(j, k, 0), obj.getPos()))
854         # self.hold_move.start()
855         # self.mommmove_random.pause()
856
857         ##### end of the "updatePos(self, obj, dt)"
858
859     # This is main loop for this program
860     def drop(self, task):
861         dt = task.time - task.last
862         task.last = task.time
863
864         if self.ground.getZ() > 0:
865             return Task.cont
866
867         # Drop Speed Change
868         self.ground.velocity += GROUND_V * dt
869
870         temp_z = self.ground.getZ() + self.ground.velocity*dt
871         self.ground.setZ(temp_z)
872         self.ground.setH((2*task.time)%360)
873

```

```

874     if temp_z > 1 :
875         if ( task.MISSION_COMPLETE == 0 ) :
876             self.ef_crash.play()
877             self.unloadTurtles()
878             self.unloadRest2()
879     else :
880         #update Turtle position
881         if ( task.MISSION_COMPLETE == 0 ) :
882             self.updateTurtle(dt)
883             if ( ( self.momturtle.getPos() self.r_babyturtle.getPos()).lengthSquared() < COL_DIST) :
884                 if ( self.r_hold == 0 ) :
885                     self.holdMom(self.r_babyturtle)
886                     self.r_hold = 1
887                     self.hold_no += 1
888             if ( ( self.momturtle.getPos() self.g_babyturtle.getPos()).lengthSquared() < COL_DIST) :
889                 if ( self.g_hold == 0 ) :
890                     self.holdMom(self.g_babyturtle)
891                     self.g_hold = 1
892                     self.hold_no += 1
893             if ( ( self.momturtle.getPos() self.b_babyturtle.getPos()).lengthSquared() < COL_DIST) :
894                 if ( self.b_hold == 0 ) :
895                     self.holdMom(self.b_babyturtle)
896                     self.b_hold = 1
897                     self.hold_no += 1
898             if ( ( self.momturtle.getPos() self.y_babyturtle.getPos()).lengthSquared() < COL_DIST) :
899                 if ( self.y_hold == 0 ) :
900                     self.holdMom(self.y_babyturtle)
901                     self.y_hold = 1
902                     self.hold_no += 1
903
904             if ( self.hold_no == 4 ) :
905                 self.unloadTurtles()
906                 self.loadRescueTeam()
907                 self.loopClouds.finish()
908                 task.MISSION_COMPLETE = 1
909
910     return Task.cont
911     ##### end of the "drop(self, task)"
912
913
914     #called when a drum pad is hit
915     def hitRed(self):
916         heading = self.r_babyturtle.getH()
917         if heading > 360 : heading = heading - 360
918         dt = globalClock.getDt()
919
920         heading_rad = DEG_TO_RAD * heading
921
922         newVel = (
923             Vec3(sin(heading_rad), cos(heading_rad), 0) * ACCELERATION * dt)
924         # Vec3(sin(heading_rad), cos(heading_rad), 0) * DOOLY_MOVE_SCALE)
925         newVel += Vec3(self.r_babyturtle.velocity)
926
927         if newVel.lengthSquared() > MAX_VEL_SQ:
928             newVel.normalize()
929             newVel *= MAX_VEL
930             if v_flag == 1 :
931                 newVel = self.r_babyturtle.velocity
932             self.r_babyturtle.velocity = newVel
933             self.ef_push.play()
934
935     # return Task.cont
936     ##### end of the "hitRed(self)"
937
938     #called when any pad is spun left
939     def leftRed(self, speed):
940         self.r_babyturtle.actorInterval("l_turn",playRate=0.5).loop()
941
942         heading = self.r_babyturtle.getH()
943         dt = globalClock.getDt()
944
945         #Heading is the roll value for this model
946         #Change heading if left or right is being pressed
947         heading = dt * TURN_RATE
948         self.r_babyturtle.setH(heading %360)
949
950         self.ef_spin.play()
951     ##### end of the "leftRed(self, speed)"
952
953     #called when any pad is spun right
954     def rightRed(self, speed):

```

```

955         self.r_babyturtle.actorInterval("r_turn",playRate=0.5).loop()
956
957         heading = self.r_babyturtle.getH()
958         dt = globalClock.getDt()
959
960         #Heading is the roll value for this model
961         #Change heading if left or right is being pressed
962         heading += dt * TURN_RATE
963         self.r_babyturtle.setH(heading %360)
964
965         self.ef_spin.play()
966         ##### end of the "rightRed(self, speed)"
967
968         #called when a drum pad is hit
969         def hitGreen(self):
970             heading = self.g_babyturtle.getH()
971             if heading > 360 : heading = heading - 360
972             dt = globalClock.getDt()
973
974             heading_rad = DEG_TO_RAD * heading
975             newVel = (
976                 Vec3(sin(heading_rad), cos(heading_rad), 0) * ACCELERATION * dt)
977             newVel += Vec3(self.g_babyturtle.velocity)
978             #Clamps the new velocity to the maximum speed. lengthSquared() is used
979             #again since it is faster than length()
980             if newVel.lengthSquared() > MAX_VEL_SQ:
981                 newVel.normalize()
982                 newVel *= MAX_VEL
983                 if v_flag == 1 :
984                     newVel = self.g_babyturtle.velocity
985             self.g_babyturtle.velocity = newVel
986
987             self.ef_push.play()
988             # return Task.cont
989             ##### end of the "hitGreen(self)"
990
991         #called when any pad is spun left
992         def leftGreen(self, speed):
993             self.g_babyturtle.actorInterval("l_turn",playRate=0.5).loop()
994
995             heading = self.g_babyturtle.getH()
996             dt = globalClock.getDt()
997
998             #Heading is the roll value for this model
999             #Change heading if left or right is being pressed
1000             heading -= dt * TURN_RATE
1001             self.g_babyturtle.setH(heading %360)
1002
1003             self.ef_spin.play()
1004             ##### end of the "leftGreen(self, speed)"
1005
1006         #called when any pad is spun right
1007         def rightGreen(self, speed):
1008             self.g_babyturtle.actorInterval("r_turn",playRate=0.5).loop()
1009
1010             heading = self.g_babyturtle.getH()
1011             dt = globalClock.getDt()
1012
1013             #Heading is the roll value for this model
1014             #Change heading if left or right is being pressed
1015             heading += dt * TURN_RATE
1016             self.g_babyturtle.setH(heading %360)
1017
1018             self.ef_spin.play()
1019             ##### end of the "rightGreen(self, speed)"
1020
1021         #called when a drum pad is hit
1022         def hitBlue(self):
1023             #Thrust causes acceleration in the direction the ship is currently facing
1024             heading = self.b_babyturtle.getH()
1025             if heading > 360 : heading = heading - 360
1026             dt = globalClock.getDt()
1027             heading_rad = DEG_TO_RAD * heading
1028             newVel = (
1029                 Vec3(sin(heading_rad), cos(heading_rad), 0) * ACCELERATION * dt)
1030             newVel += Vec3(self.b_babyturtle.velocity)
1031             if newVel.lengthSquared() > MAX_VEL_SQ:
1032                 newVel.normalize()
1033                 newVel *= MAX_VEL
1034                 if v_flag == 1 :
1035                     newVel = self.b_babyturtle.velocity

```

```

1036         self.b_babyturtle.velocity = newVel
1037
1038         self.ef_push.play()
1039         # return Task.cont
1040         ##### end of the "hitBlue(self)"
1041
1042     #called when any pad is spun left
1043     def leftBlue(self, speed):
1044         self.b_babyturtle.actorInterval("l_turn",playRate=0.5).loop()
1045
1046         heading = self.b_babyturtle.getH()
1047         dt = globalClock.getDt()
1048
1049         #Heading is the roll value for this model
1050         #Change heading if left or right is being pressed
1051         heading = dt * TURN_RATE
1052         self.b_babyturtle.setH(heading %360)
1053
1054         self.ef_spin.play()
1055         ##### end of the "leftBlue(self, speed)"
1056
1057     #called when any pad is spun right
1058     def rightBlue(self, speed):
1059         self.b_babyturtle.actorInterval("r_turn",playRate=0.5).loop()
1060
1061         heading = self.b_babyturtle.getH()
1062         dt = globalClock.getDt()
1063
1064         #Heading is the roll value for this model
1065         #Change heading if left or right is being pressed
1066         heading += dt * TURN_RATE
1067         self.b_babyturtle.setH(heading %360)
1068
1069         self.ef_spin.play()
1070         ##### end of the "rightBlue(self, speed)"
1071
1072     #called when a drum pad is hit
1073     def hitYellow(self):
1074         heading = self.y_babyturtle.getH()
1075         if heading > 360 : heading = heading - 360
1076         dt = globalClock.getDt()
1077
1078         heading_rad = DEG_TO_RAD * heading
1079         #This builds a new velocity vector and adds it to the current one
1080         #Relative to the camera, the screen in Panda is the XZ plane.
1081         #Therefore all of our Y values in our velocities are 0 to signify no
1082         #change in that direction
1083         newVel = (
1084             Vec3(sin(heading_rad), cos(heading_rad), 0) * ACCELERATION * dt)
1085         newVel += Vec3(self.y_babyturtle.velocity)
1086         #Clamps the new velocity to the maximum speed. lengthSquared() is used
1087         #again since it is faster than length()
1088         if newVel.lengthSquared() > MAX_VEL_SQ:
1089             newVel.normalize()
1090             newVel *= MAX_VEL
1091             if v_flag == 1 :
1092                 newVel = self.y_babyturtle.velocity
1093         self.y_babyturtle.velocity = newVel
1094
1095         self.ef_push.play()
1096         # return Task.cont
1097         ##### end of the "hitYellow(self)"
1098
1099     #called when any pad is spun left
1100     def leftYellow(self, speed):
1101         self.y_babyturtle.actorInterval("l_turn",playRate=0.5).loop()
1102
1103         heading = self.y_babyturtle.getH()
1104         dt = globalClock.getDt()
1105
1106         #Heading is the roll value for this model
1107         #Change heading if left or right is being pressed
1108         heading = dt * TURN_RATE
1109         self.y_babyturtle.setH(heading %360)
1110
1111         self.ef_spin.play()
1112         ##### end of the "leftYellow(self, speed)"
1113
1114     #called when any pad is spun right
1115     def rightYellow(self, speed):
1116         self.y_babyturtle.actorInterval("r_turn",playRate=0.5).loop()

```

```

1117
1118     heading = self.y_babyturtle.getH()
1119     dt = globalClock.getDt()
1120
1121     #Heading is the roll value for this model
1122     #Change heading if left or right is being pressed
1123     heading += dt * TURN_RATE
1124     self.y_babyturtle.setH(heading %360)
1125
1126     self.ef_spin.play()
1127     ##### end of the "rightYellow(self, speed)"
1128
1129     #####
1130     #####
1131     #All your code above this, do not touch below.
1132     #####
1133     #####
1134
1135     #used for filtering out cross activation
1136     def crossActivation(self, task):
1137         if task.time < .3:
1138             self.ignore("HIT_0")
1139             self.ignore("HIT_1")
1140             self.ignore("HIT_2")
1141             self.ignore("HIT_3")
1142         else:
1143             self.revert()
1144         return Task.cont
1145
1146     def revert(self):
1147         taskMgr.remove('padTask')
1148         self.accept("HIT_0",self.hitGreen)
1149         self.accept("HIT_1",self.hitBlue)
1150         self.accept("HIT_2",self.hitRed)
1151         self.accept("HIT_3",self.hitYellow)
1152
1153     def setMaskDistanceFromCamera(self, newDist):
1154         if (newDist > 0):
1155             self.blackMask.setY(newDist)
1156             self.blackMask.setScale(newDist/camera.getZ())
1157
1158     # def spinCCW(self, station, speed):
1159     #     print "station",station,"spun counter clockwise"
1160
1161     # def spinCW(self, station, speed):
1162     #     print "station",station,"spun clockwise"
1163
1164     def hit(self, station):
1165         #causes pad delay to work, so no more cross activation.
1166         taskMgr.add(self.crossActivation,'padTask')
1167         print "station to see",station,"was hit"
1168
1169     def polling(self, task):
1170         self.drumpadControls.Poll()
1171         self.spinnerControls.Poll()
1172         return Task.cont
1173
1174 class IntroVisual(DirectObject):
1175     def __init__(self):
1176         base.setBackgroundColor(0, 0, 0)
1177         self.visualPlane = loader.loadModelCopy('plane/plane')
1178         self.visualPlane.setPosHprScale(0,8,0,0,0,0,5,2,1,4,0)
1179         self.visualPlane.reparentTo(render) #reparent to render
1180
1181         self.visualPlane.setTransparency(1)
1182
1183         self.introTexp = self.loadTextureMovie(25, 'IntroV/Slide',
1184                                             'png', padding = 2)
1185
1186         self.i0Vmusic = loader.loadMusic('sound/begin.mp3')
1187         self.i1Vmusic = loader.loadMusic('sound/ef_boys.mp3')
1188         self.i2Vmusic = loader.loadMusic('sound/ef_help.mp3')
1189         self.i3Vmusic = loader.loadMusic('sound/LetsCatch.mp3')
1190
1191         self.introTask = taskMgr.add(self.textureMovie, "introTask")
1192
1193         self.introTask.fps = 1
1194
1195         self.introTask.obj = self.visualPlane
1196
1197         self.introTask.textures = self.introTexp

```

```

1198
1199     def textureMovie(self, task):
1200         currentFrame = int(task.time * task.fps)
1201         if currentFrame < len(task.textures):
1202             task.obj.setTexture(task.textures[currentFrame],1)
1203             if currentFrame == 0 :
1204                 self.i0Vmusic.play()
1205             if currentFrame == 16 :
1206                 self.i1Vmusic.play()
1207                 self.i0Vmusic.setVolume(.5)
1208             if currentFrame == 18 :
1209                 self.i2Vmusic.play()
1210                 self.i0Vmusic.setVolume(.2)
1211             if currentFrame == 20 :
1212                 self.i3Vmusic.play()
1213
1214         else :
1215             taskMgr.remove(task)
1216
1217         return Task.cont
1218
1219     def loadTextureMovie(self, frames, name, suffix, padding = 1):
1220         return [loader.loadTexture((name+"%0"+str(padding)+"d."+suffix) % i)
1221                 for i in range(frames)]
1222
1223     base.disableMouse()
1224
1225     #i = IntroVisual()
1226     j=JamODrum()
1227     #e=lastMovie()
1228     run()

```

## 2. 팀 B 프로젝트 소스

```
1 # Woosong Univ. Dept. of GameMultimedia Chang Jo Lee
2 # 8. 12. 2005
3 from pandac.PandaModules import * # basic Panda modules
4 from direct.showbase.DirectObject import DirectObject # for event handling
5 from direct.actor.Actor import Actor
6 from direct.task import Task
7 from direct.actor import Actor
8 from direct.interval.IntervalGlobal import *
9 import math
10 from direct.gui.DirectGui import *
11 #import hlview
12 from direct.directutil import Mopath
13 from direct.interval.MopathInterval import *
14
15 BULLET_LIFE = 4 #How long bullets stay on screen before removed
16 BULLET_REPEAT = 0.1 #How often bullets can be fired
17 BULLET_SPEED = 40 #Speed bullets move
18 BULLET_SCALE = 1
19
20 class World(DirectObject):
21     def __init__(self):
22         self.loadModels()
23         self.loadSounds()
24         self.bullets = []
25
26         self.accept("escape", sys.exit) #Escape quits
27
28         #self.keys = {"fire" : 0, "jump": 0, "crouch": 0}
29
30         self.keys = {"fire1" : 0, "fire2" : 0,"fire3" : 0,"jump": 0, "crouch": 0}
31         self.accept("1", self.setKey, ["fire1", 1])
32         self.accept("1 up", self.setKey, ["fire1", 0])
33         self.accept("2", self.setKey, ["fire2", 1])
34         self.accept("2 up", self.setKey, ["fire2", 0])
35         self.accept("3", self.setKey, ["fire3", 1])
36         self.accept("3 up", self.setKey, ["fire3", 0])
37
38         self.accept('startGame', self.start)
39
40     def start(self):
41         #base.disableMouse()
42         base.enableMouse()
43         base.setBackgroundColor(0, 0.75, 0.95)
44         camera.setPosHpr(Vec3(0,0,10),Vec3(0, 35,0))
45         self.setupLighting()
46         self.setupIntervals()
47         self.setupCollisions()
48
49         self.gameTask = taskMgr.add(self.gameLoop, "gameLoop")
50         self.gameTask.last = 0 #Task time of the last frame
51         self.gameTask.nextBullet = 0 #Task time when the next bullet may be fired
52
53         # setup key controls
54         self.acceptOnce("arrow_up",self.walk)
55         self.acceptOnce("arrow_left",self.turn,[ 1])
56         self.acceptOnce("arrow_right",self.turn,[1])
57
58         self.accept("arrow_up up",self.stopWalk)
59         self.accept("arrow_left up",self.stopTurn)
60         self.accept("arrow_right up",self.stopTurn)
61
62         # start camera task
63         taskMgr.add(self.cameraFollowTask,'cameraFollowTask')
64
```

```

65         # start music
66         #SoundInterval(self.music1).loop()
67         self.music1.play()
68     # end start
69
70     def loadModels(self):
71         # load a modeloot
72         #self.player = Actor.Actor("sonic/sonic",{"walk":"sonic/sonic_run","eat":"sonic/sonic_win"})
73         self.player = Actor.Actor("sonic/sonic_win",{"walk":"sonic/sonic_run","eat":"sonic/sonic_win"})
74         self.player.reparentTo(render)
75         self.player.setScale(4)
76         #self.player.setPosHpr(46, 53,0,270,0,0)
77         self.player.setPosHpr(46, 55,0,270,0,0)
78         self.player_forwardNode = self.player.attachNewNode('forward')
79         self.player_forwardNode.setPos(0, 5,1)
80         self.player_forwardNode.reparentTo(self.player)
81
82         self.report = loader.loadModelCopy("models/bullet3")
83         self.note = loader.loadModelCopy("models/bullet2")
84         self.heart = loader.loadModelCopy("models/heart")
85
86         #monsters
87         self.monster1_holder = render.attachNewNode("monster1")
88         #self.monster1 = Actor.Actor("models/child_rando",{"rando_walk":"models/child_rando"})
89         self.monster1 = Actor.Actor("models/monster1/monster1",{"monster1_walk":"models/monster1/monster1"})
90         self.monster1.found = 0
91         self.monster1.health = 0
92         self.monster1.setScale(1)
93         self.monster1.setH(180)
94         self.monster1.reparentTo(self.monster1_holder)
95         self.monster1_walk = self.monster1.actorInterval("monster1_walk", playRate = 1)
96         self.monster1_walk.loop()
97
98         self.monster2_holder = render.attachNewNode("monster2")
99         self.monster2 = Actor.Actor("models/monster2/monster2",{"monster2_walk":"models/monster2/monster2"})
100        self.monster2.found = 0
101        self.monster2.health = 0
102        self.monster2.setScale(0.6)
103        self.monster2.setH(180)
104        self.monster2.reparentTo(self.monster2_holder)
105        self.monster2_walk = self.monster2.actorInterval("monster2_walk", playRate = 1)
106        self.monster2_walk.loop()
107
108        ## self.monster3_holder = render.attachNewNode("monster3")
109        ## self.monster3 = Actor.Actor("models/child_rando",{"rando_walk":"models/child_rando"})
110        ## self.monster3.found = 0
111        ## self.monster3.health = 0
112        ## self.monster3.setScale(0.35,0.35,0.35)
113        ## self.monster3.setH(180)
114        ## self.monster3.reparentTo(self.monster3_holder)
115        ## self.monster3_walk = self.monster3.actorInterval("rando_walk", playRate = 1)
116        ## self.monster3_walk.loop()
117
118        self.monster4_holder = render.attachNewNode("monster4")
119        self.monster4 = Actor.Actor("dragon/dragon_don_walk",{"rando_walk":"dragon/dragon_don_attack"})
120        self.monster4.found = 0
121        self.monster4.health = 0
122        self.monster4.setScale(0.6)
123        self.monster4.setH(180)
124        self.monster4.reparentTo(self.monster4_holder)
125        self.monster4_walk = self.monster4.actorInterval("rando_walk", playRate = 1)
126        self.monster4_walk.loop()
127
128        ## self.monster5_holder = render.attachNewNode("monster5")
129        ## self.monster5 = Actor.Actor("models/child_rando",{"rando_walk":"models/child_rando"})
130        ## self.monster5.found = 0

```

```

131     ## self.monster5.health = 0
132     ## self.monster5.setScale(0.35,0.35,0.35)
133     ## self.monster5.setH(180)
134     ## self.monster5.reparentTo(self.monster5_holder)
135     ## self.monster5_walk = self.monster5.actorInterval("rando_walk", playRate = 1)
136     ## self.monster5_walk.loop()
137
138     #end monsters
139
140     #load Panda
141     self.panda = Actor.Actor ("panda", {"panda_walk":"panda walk"})
142     self.panda.setPos(0,1.5,0)
143     self.panda.setH(270)
144     self.panda.setScale(0.2)
145     self.panda.reparentTo(self.monster4_holder)
146     self.panda_walk = self.panda.actorInterval("panda_walk", playRate = 1)
147     self.panda_walk.loop()
148
149     self.man = Actor.Actor ("models/pilot/pilot model", {"man_idle":"models/pilot/pilot idle"})
150     self.man.setPos(0,1.5,0)
151     self.man.setH(270)
152     self.man.setScale(0.05)
153     self.man.reparentTo(self.monster4_holder)
154     self.man_idle = self.man.actorInterval("man_idle", playRate = 1)
155     self.man_idle.loop()
156     self.man.hide()
157
158     #paths
159     self.path1 = Mopath.Mopath()
160     self.path1.loadFile('models/city/path1')
161     self.path1.fFaceForward = 1
162     self.path1Interval = MopathInterval(self.path1, self.monster1_holder)
163     self.path1Interval.loop(playRate = 0.1)
164
165     self.path2 = Mopath.Mopath()
166     self.path2.loadFile('models/city/path2')
167     self.path2.fFaceForward = 1
168     self.path2Interval = MopathInterval(self.path2, self.monster2_holder)
169     self.path2Interval.loop(playRate = 0.1)
170
171     ## self.path3 = Mopath.Mopath()
172     ## self.path3.loadFile('models/city/path3')
173     ## self.path3.fFaceForward = 1
174     ## self.path3Interval = MopathInterval(self.path3, self.monster3_holder)
175     ## self.path3Interval.loop(playRate = 0.1)
176
177     self.path4 = Mopath.Mopath()
178     self.path4.loadFile('models/city/path4_boss')
179     self.path4.fFaceForward = 1
180     self.path4Interval = MopathInterval(self.path4, self.monster4_holder)
181     self.path4Interval.loop(playRate = 0.1)
182
183     ## self.path5 = Mopath.Mopath()
184     ## self.path5.loadFile('models/city/path5')
185     ## self.path5.fFaceForward = 1
186     ## self.path5Interval = MopathInterval(self.path5, self.monster5_holder)
187     ## self.path5Interval.loop(playRate = 0.1)
188     #end paths
189
190     # load an environment
191     self.viron = loader.loadModel("models/city/city0")
192     self.viron.reparentTo(render)
193     self.viron.setScale(0.27)
194     self.viron.setCollideMask(BitMask32.bit(0))
195
196     # end loadModels
197
198     def loadSounds(self):

```

```

199
200     # load 2 sound effects
201     self.eatSound = loader.loadSfx("sounds/scream.wav")
202     self.reportSound = loader.loadSfx("sounds/report.MP3")
203     self.endSound = loader.loadSfx("sounds/ending.MP3")
204     self.endSound.index = 0
205     self.endSound.stopped = 1
206     self.monster1Sound = loader.loadSfx("sounds/monster1")
207     self.monster2Sound = loader.loadSfx("sounds/monster2")
208     #self.monster3Sound = loader.loadSfx("sounds/monster3")
209     self.monster4Sound = loader.loadSfx("sounds/monster3")
210     #self.monster5Sound = loader.loadSfx("sounds/ending.MP3")
211     self.turnSound = loader.loadSfx("sounds1/PENGIE_SCREAM_1.wav")
212     #self.onemiterSound = loader.loadSfx("sounds1/jawstHEME.wav")
213     #self.mySound1=SoundInterval(self.monster1Sound)
214     # load background music
215     self.music1 = loader.loadMusic("sounds/mi.MP3")
216     self.music1.index = 0
217     #self.music2 = loader.loadMusic("sounds/fiveo.MP3")
218
219     # end loadSounds
220
221     def setupLighting(self):
222         # start with a blank light attribute
223         lightAttribute = LightAttrib.makeAllOff()
224
225         # some directional light
226         dirLight = DirectionalLight('dirLight')
227         dirLight.setColor(Vec4(0.6,0.6,0.6,1.0))
228         dirLightNP = render.attachNewNode(dirLight.upcastToPandaNode()) # crashes without upcast
229         dirLightNP.setPos(Vec3(0.0, 10.0,10.0))
230         dirLightNP.setHpr(Vec3(0.0, 26.0,0.0))
231         lightAttribute = lightAttribute.addLight(dirLight) # add to attribute
232
233         # add a little extra ambient lighting just because
234         ambientLight = AmbientLight('ambientLight')
235         ambientLight.setColor(Vec4(0.25,0.25,0.25,1.0))
236         ambientLightNP = render.attachNewNode(ambientLight.upcastToPandaNode())
237         lightAttribute = lightAttribute.addLight(ambientLight)
238
239         # apply to world
240         render.node().setAttrib(lightAttribute)
241     # end setupLighting
242
243     def setupIntervals(self):
244         self.playerWalk = self.player.actorInterval("walk")
245         self.playerTurn = self.player.actorInterval("turn")
246     # end setupIntervals
247
248     def collide(self, collEntry):
249         print "Hello!\n"
250
251     def setupCollisions(self):
252         ##Collision Code##
253         # use an event collision handler (sends events on collisions)
254         self.cHandler = CollisionHandlerEvent()
255
256         self.cHandler.setInPattern("%in into %fn")
257
258         # make a traverser and make it the default traverser
259         base.cTrav = CollisionTraverser()
260         self.pusher = CollisionHandlerPusher()
261
262         #setup a collision solid for the player
263         cNode = CollisionNode('player')
264         cNode.addSolid(CollisionSphere(0,0,2,1.5))
265         cNode.setIntoCollideMask(BitMask32.bit(1))
266         cNode.setFromCollideMask(BitMask32.bit(0))

```

```

267     self.playerColl = self.player.attachNewNode(cNode)
268     self.playerColl.show()
269     base.cTrav.addCollider(self.playerColl, self.pusher)
270     self.pusher.addCollider(self.playerColl, self.player, base.drive.node())
271
272     #make a collision solid for monster1
273     cNode = CollisionNode('monster1')
274     cNode.addSolid(CollisionSphere(0,0,2,12))
275     cNode.setIntoCollideMask(BitMask32.allOff())
276     cNode.setFromCollideMask(BitMask32.bit(1))
277     self.monster1Coll = self.monster1.attachNewNode(cNode)
278     self.monster1Coll.show()
279     base.cTrav.addCollider(self.monster1Coll, self.cHandler)
280     self.accept('player into monster1', self.monster1Event)
281
282     #make a collision solid for monster2
283     cNode = CollisionNode('monster2')
284     cNode.addSolid(CollisionSphere(0,0,2,12))
285     cNode.setIntoCollideMask(BitMask32.allOff())
286     cNode.setFromCollideMask(BitMask32.bit(1))
287     self.monster2Coll = self.monster2.attachNewNode(cNode)
288     self.monster2Coll.show()
289     base.cTrav.addCollider(self.monster2Coll, self.cHandler)
290     self.accept('player into monster2', self.monster2Event)
291
292
293     #make a collision solid for monster3
294     ## cNode = CollisionNode('monster3')
295     ## cNode.addSolid(CollisionSphere(0,0,2,8))
296     ## cNode.setIntoCollideMask(BitMask32.allOff())
297     ## cNode.setFromCollideMask(BitMask32.bit(1))
298     ## self.monster3Coll = self.monster3.attachNewNode(cNode)
299     ## #self.monster3Coll.show()
300     ## base.cTrav.addCollider(self.monster3Coll, self.cHandler)
301     ## self.accept('player into monster3', self.monster3Event)
302
303     #make a collision solid for monster4
304     cNode = CollisionNode('monster4')
305     cNode.addSolid(CollisionSphere(0,0,2,16))
306     cNode.setIntoCollideMask(BitMask32.allOff())
307     cNode.setFromCollideMask(BitMask32.bit(1))
308     self.monster4Coll = self.monster4.attachNewNode(cNode)
309     self.monster4Coll.show()
310     base.cTrav.addCollider(self.monster4Coll, self.cHandler)
311     self.accept('player into monster4', self.monster4Event)
312
313     #make a collision solid for monster5
314     ## cNode = CollisionNode('monster5')
315     ## cNode.addSolid(CollisionSphere(0,0,2,8))
316     ## cNode.setIntoCollideMask(BitMask32.allOff())
317     ## cNode.setFromCollideMask(BitMask32.bit(1))
318     ## self.monster5Coll = self.monster5.attachNewNode(cNode)
319     ## #self.monster5Coll.show()
320     ## base.cTrav.addCollider(self.monster5Coll, self.cHandler)
321     ## self.accept('player into monster5', self.monster5Event)
322
323     def monster1Event(self, cEntry):
324         print "Monster 1 Encounter!"
325         self.path1Interval.pause()
326         self.monster1_walk.pause()
327         self.music1.index = self.music1.getTime()
328         self.music1.stop()
329         self.monster1Sound.play()
330         self.monster1.found = 1
331         print self.monster1
332
333     def monster2Event(self, cEntry):
334         print "Monster 2 Encounter!"

```

```

335     self.path2Interval.pause()
336     self.monster2_walk.pause()
337     self.music1.index = self.music1.getTime()
338     self.music1.stop()
339     self.monster2Sound.play()
340     self.monster2.found = 1
341
342     ## def monster3Event(self, cEntry):
343     ##     print "Monster 3 Encounter!"
344     ##     self.path3Interval.pause()
345     ##     self.monster3_walk.pause()
346     ##     self.monster3.found = 1
347
348     def monster4Event(self, cEntry):
349     print "Monster 4 Encounter!"
350     self.path4Interval.pause()
351     self.monster4_walk.pause()
352     self.music1.stop()
353     self.monster4Sound.play()
354     self.monster4.found = 1
355
356     ## def monster5Event(self, cEntry):
357     ##     print "Monster 5 Encounter!"
358     ##     self.path5Interval.pause()
359     ##     self.monster5_walk.pause()
360     ##     self.music2.play()
361     ##     self.monster5.found = 1
362
363     def walk(self):
364     taskMgr.add(self.walkTask,'walkTask')
365     self.playerWalk.loop()
366     # end walk
367
368     def stopWalk(self):
369     taskMgr.remove('walkTask')
370     self.playerWalk.pause()
371     self.acceptOnce("arrow_up",self.walk)
372
373     # end stopWalk
374
375     def walkTask(self,task):
376     speed = 7.0 # ft/s
377     dt = globalClock.getDt()
378     dist = speed*dt
379     angle = self.player.getH()*math.pi/180.0
380     dx = dist*math.sin(angle)
381     dy = dist* math.cos(angle)
382     self.player.setPos(Vec3(self.player.getX()+dx,self.player.getY()+dy,0))
383     return Task.cont
384     # end walkTask
385
386     def turn(self,dir):
387     taskMgr.add(self.turnTask,'turnTask',extraArgs=[dir])
388     self.playerTurn.loop()
389     self.ignore("arrow_left")
390     self.ignore("arrow_right")
391     # end turn
392
393     def stopTurn(self):
394     taskMgr.remove('turnTask')
395     self.playerTurn.pause()
396     self.acceptOnce("arrow_left",self.turn,[ 1])
397     self.acceptOnce("arrow_right",self.turn,[1])
398     #self.acceptOnce("arrow_down",self.walk,[ 1])
399     # end stopTurn
400
401     def turnTask(self,dir):
402     speed = 80.0 # degrees/s

```

```

403         dt = globalClock.getDt()
404         angle = dir*speed*dt
405         self.player.setH(self.player.getH() angle)
406         return Task.cont
407     # end turnTask
408
409     def cameraFollowTask(self,task):
410         speed = 17.0 # ft/s
411         offset = Vec3(0,18,5)
412         dt = globalClock.getDt()
413
414         currPos = camera.getPos()
415         desiredPos = self.player.getPos()+self.player.getQuat().xform(offset)
416         direction = Vec3(desiredPos currPos)
417         if (direction.length())>speed*dt):
418             direction.normalize()
419             camera.setPos(camera.getPos()+direction*speed*dt)
420         else:
421             camera.setPos(desiredPos)
422
423         camera.lookAt(self.player,Point3(0,0,1))
424         return Task.cont
425     # end cameraFollowTask
426
427     def eat(self,cEntry):
428         # stop other animations
429         self.stopWalk()
430         self.stopTurn()
431         # don't move while animating
432         self.ignore("arrow_up")
433         self.ignore("arrow_left")
434         self.ignore("arrow_right")
435
436         # eat bamboo
437         self.targets.remove(cEntry.getIntoNodePath().getParent())
438         cEntry.getIntoNodePath().getParent().remove()
439         pandaEat = Sequence(Parallel(SoundInterval(self.eatSound),self.player.actorInterval("eat")),
440                               Func(self.doneEat))
441         pandaEat.start()
442     # end eat
443
444     def doneEat(self):
445         self.acceptOnce("arrow_up",self.walk)
446         self.acceptOnce("arrow_left",self.turn,[ 1])
447         self.acceptOnce("arrow_right",self.turn,[1])
448     # end doneEat
449
450     def hitPanda(self):
451         # stop other animations
452         self.stopWalk()
453         self.stopTurn()
454         # don't move while animating
455         self.ignore("arrow_up")
456         self.ignore("arrow_left")
457         self.ignore("arrow_right")
458
459         # hit the panda
460         print "OUCH! You hit the Sonic."
461         currH = self.player.getH()
462         pandaStunned = Sequence(Parallel(SoundInterval(self.hitSound),
463                                         Sequence(
464                                             self.player.hprInterval(.25,Vec3(currH 20,0,0)),
465                                             self.player.hprInterval(.5,Vec3(currH+20,0,0)),
466                                             self.player.hprInterval(.25,Vec3(currH,0,0))
467                                         )),
468                               Func(self.doneEat))
469         pandaStunned.start()
470     # end hitPanda

```

```

471
472     def collide(self, collEntry):
473         print "Hello!\n"
474
475     def setKey(self, key, val):
476         self.keys[key] = val
477
478         if ((key == "fire3") & (val == 0)):
479             print "Stopping the song"
480             #stop sound
481             #Save the current time of the music box song
482             self.endSound.index = self.endSound.getTime()
483             self.endSound.stopped = 1
484             self.endSound.stop()           #Stop the music box song
485         if ((key == "fire3") & (val == 1)) & (self.endSound.stopped):
486             print "Starting the song"
487             #start sound
488             #Reset the time of the music box song so it starts where it left off
489             self.endSound.setTime(self.endSound.index)
490             self.endSound.stopped = 0
491             self.endSound.play()           #Play the music box song
492
493     def gameLoop(self, task):
494         dt = task.time - task.last
495         task.last = task.time
496
497         #check to see if the ship can fire
498         #if self.keys["fire"] and task.time > task.nextBullet:
499             #self.fire(task.time) #If so, call the fire function
500             #And disable firing for a bit
501             #task.nextBullet = task.time + BULLET_REPEAT
502             #self.keys["fire"] = 0 #Remove the fire flag until the next spacebar press
503
504         #check to see if the ship can fire
505         if self.keys["fire1"] and task.time > task.nextBullet:
506             self.fire1(task.time)
507             task.nextBullet = task.time + BULLET_REPEAT
508             self.reportSound.play()
509
510             #for monster1
511             if self.monster1.found == 1:
512                 self.monster1.health += 0.1
513                 print self.monster1.health
514
515         if self.keys["fire2"] and task.time > task.nextBullet:
516             self.fire2(task.time)
517             task.nextBullet = task.time + BULLET_REPEAT
518             self.eatSound.play()
519
520             #for monster2
521             if self.monster2.found == 1:
522                 self.monster2.health += 0.1
523                 print self.monster2.health
524
525         if self.keys["fire3"] and task.time > task.nextBullet:
526             self.fire3(task.time)
527             task.nextBullet = task.time + BULLET_REPEAT
528
529             #for monster4
530             if self.monster4.found == 1:
531                 self.monster4.health += 0.1
532                 print self.monster4.health
533
534         #update bullets
535         newBulletArray = []
536         for obj in self.bullets:
537             self.updatePos(obj, dt)           #Update the bullet
538

```

```

539         #Bullets have an expiration time (see definition of fire)
540         #If a bullet has not expired, add it to the new bullet list so that it
541         #will continue to exist
542         if obj.expires > task.time: newBulletArray.append(obj)
543         else: obj.remove()           #Otherwise remove it from the scene
544     #Set the bullet array to be the newly updated array
545     self.bullets = newBulletArray
546
547     if self.monster1.health > 3:
548         print "Defeated monster 1!!!!!"
549         self.monster1.remove()
550         self.playerEat = Sequence(Parallel(SoundInterval(self.eatSound),self.player.actorInterval("eat")),
551                                   Func(self.doneEat))
552         self.monster1.health = 0
553         self.monster1.found = 0
554         self.playerEat.start()
555         self.music1.setTime(self.music1.index)
556         self.music1.play()
557
558     if self.monster2.health > 3:
559         print "Defeated monster 2!!!!!"
560         self.monster2.remove()
561         self.playerEat = Sequence(Parallel(SoundInterval(self.eatSound),self.player.actorInterval("eat")),
562                                   Func(self.doneEat))
563         self.monster2.health = 0
564         self.monster2.found = 0
565         self.playerEat.start()
566         self.music1.setTime(self.music1.index)
567         self.music1.play()
568
569     ## if self.monster3.health > 3:
570     ##     print "Defeated monster 3!!!!!"
571     ##     self.monster3.remove()
572     ##     self.playerEat = Sequence(Parallel(SoundInterval(self.eatSound),self.player.actorInterval("eat")),
573     ##                               ## Func(self.doneEat))
574     ##     self.monster3.health = 0
575     ##     self.monster3.found = 0
576     ##     self.playerEat.start()
577
578     if self.monster4.health > 3:
579         print "Defeated Boss monster 4!!!!!"
580         self.monster4.remove()
581         self.playerEat = Sequence(Parallel(SoundInterval(self.eatSound),self.player.actorInterval("eat")),
582                                   Func(self.doneEat))
583         self.monster4.health = 0
584         self.monster4.found = 0
585         self.playerEat.start()
586         self.transform()
587
588     ## if self.monster5.health > 3:
589     ##     print "Defeated monster 5!!!!!"
590     ##     self.monster5.remove()
591     ##     self.playerEat = Sequence(Parallel(SoundInterval(self.eatSound),self.player.actorInterval("eat")),
592     ##                               ## Func(self.doneEat))
593     ##     self.monster5.health = 0
594     ##     self.monster5.found = 0
595     ##     self.playerEat.start()
596
597     return Task.cont
598
599     def transform(self):
600         remove_panda = Func(self.destroy, self.panda)
601         reveal_man = Func(self.reveal, self.man)
602         start_end = Func(self.end)
603         Sequence(self.panda.scaleInterval(1,3),
604                 self.panda.scaleInterval(1,0),
605                 remove_panda,
606                 reveal_man,

```

```

607             Wait(3),
608             start_end).start()
609
610     def end(self):
611         taskMgr.remove('cameraFollowTask')
612         self.enviro.remove()
613         messenger.send('begin_end')
614
615     def reveal(self, obj):
616         obj.show()
617
618     def destroy(self, obj):
619         obj.remove()
620
621     #Updates the positions of objects
622     def updatePos(self, obj, dt):
623         obj.velocity = obj.velocity + Vec3(0,0,15)*dt
624         newP = obj.getP() + (40*dt)
625         obj.setP(newP)
626         newPos = obj.getPos() + (obj.velocity*dt)
627         obj.setPos(newPos)
628
629     #Report
630     def fire1(self, time):
631         direction = self.player_forwardNode.getPos(render) - self.player.getPos(render)
632         direction.normalize()
633         pos = self.player.getPos() + Vec3(0,0,0.5) #offset
634         bullet = loader.loadModelCopy("models/bullet3")
635         bullet.reparentTo(render)
636         #bullet.setScale(BULLET_SCALE)
637         bullet.setScale(0.01)
638         bullet.setPos(pos)
639         bullet.setColor(1,1,1)
640         bullet.setHpr(self.player.getHpr())
641         #Velocity is in relation to the ship
642         bullet.velocity = direction * BULLET_SPEED
643         #Set the bullet expiration time to be a certain amount past the current time
644         bullet.expires = time + BULLET_LIFE
645
646         #Finally, add the new bullet to the list
647         self.bullets.append(bullet)
648
649     #Note
650     def fire2(self, time):
651         direction = self.player_forwardNode.getPos(render) - self.player.getPos(render)
652         direction.normalize()
653         pos = self.player.getPos() + Vec3(0,0,0.8) #offset
654         bullet = loader.loadModelCopy("models/bullet2")
655         bullet.reparentTo(render)
656         #bullet.setScale(BULLET_SCALE)
657         bullet.setScale(0.01)
658         bullet.setPos(pos)
659         bullet.setColor(1,1,0)
660         bullet.setHpr(self.player.getHpr())
661
662         #just for note
663         bullet.setH(bullet.getH() + 180)
664
665         #Velocity is in relation to the ship
666         bullet.velocity = direction * BULLET_SPEED
667         #Set the bullet expiration time to be a certain amount past the current time
668         bullet.expires = time + BULLET_LIFE
669
670         #Finally, add the new bullet to the list
671         self.bullets.append(bullet)
672
673     #Heart
674     def fire3(self, time):

```

```

675         direction = self.player_forwardNode.getPos(render) self.player.getPos(render)
676         direction.normalize()
677         pos = self.player.getPos() + Vec3(0,0,0.6) #offset
678         bullet = loader.loadModelCopy("models/heart")
679         bullet.reparentTo(render)
680         #bullet.setScale(BULLET_SCALE)
681         bullet.setScale(0.01)
682         bullet.setPos(pos)
683         bullet.setColor(1,0,0)
684         bullet.setHpr(self.player.getHpr())
685         #Velocity is in relation to the ship
686         bullet.velocity = direction * BULLET_SPEED
687         #Set the bullet expiration time to be a certain amount past the current time
688         bullet.expires = time + BULLET_LIFE
689
690         #Finally, add the new bullet to the list
691         self.bullets.append(bullet)
692
693     # end class World
694     class IntroVisual(DirectObject):
695         def __init__(self):
696             self.visualPlane = loader.loadModelCopy('plane/plane')
697
698             self.visualPlane.setPosHprScale(0,8,100,0,0,0,5.8,1,4.4)
699
700             self.visualPlane.setTransparency(1)
701
702             self.introTexs = self.loadTextureMovie(5, 'Intro_png/slide',
703                                                     'png', padding = 2)
704
705             self.intro = loader.loadMusic('sounds/intro')
706
707         def start(self):
708             base.disableMouse()
709             self.visualPlane.reparentTo(render)
710             camera.setPos(0,0,100)
711             self.introTask = taskMgr.add(self.textureMovie, "introTask")
712             self.introTask.fps = 0.06
713             self.introTask.obj = self.visualPlane
714             self.introTask.textures = self.introTexs
715             self.intro.play()
716
717         def textureMovie(self, task):
718             currentFrame = int(task.time * task.fps)
719
720             if currentFrame < len(task.textures):
721                 task.obj.setTexture(task.textures[currentFrame],1)
722             else :
723                 self.visualPlane.remove()
724                 messenger.send('startGame')
725                 taskMgr.remove('introTask')
726
727             return Task.cont
728
729         def loadTextureMovie(self, frames, name, suffix, padding = 1):
730
731             return [loader.loadTexture((name+"%0"+str(padding)+"d."+suffix) % i)
732                   for i in range(frames)]
733
734     class EndVisual(DirectObject):
735         def __init__(self):
736             self.visualPlane2 = loader.loadModelCopy('plane/plane')
737
738             self.visualPlane2.setPosHprScale(0,8,100,0,0,0,5.8,1,4.4)
739
740             self.visualPlane2.setTransparency(1)
741
742             self.introTexs = self.loadTextureMovie(8, 'fire_work/end',

```

```

743                                     'png', padding = 2)
744
745
746     self.fireworks = loader.loadMusic('sounds/fireworks')
747     self.onlyyou = loader.loadMusic('sounds/final')
748
749     self.accept('begin_end', self.start)
750
751     def start(self):
752         print "starting end movie"
753         base.disableMouse()
754         self.visualPlane2.reparentTo(render)
755         lightAttribute = LightAttrib.makeAllOff()
756         render.node().setAttrib(lightAttribute)
757         print "resetting the camera"
758         camera.setPosHpr(0,0,100,0,0,0)
759         print "camera is now at"
760         print camera.getPos()
761         self.introTask = taskMgr.add(self.textureMovie, "endTask")
762         self.introTask.fps = 3
763         self.introTask.obj = self.visualPlane2
764         self.introTask.textures = self.introTexs
765         Sequence(Func(self.playSound, self.fireworks),
766                 Wait(10),
767                 Func(self.playSound, self.onlyyou)).start()
768
769     def playSound(self, sound):
770         sound.play()
771
772     def textureMovie(self, task):
773         currentFrame = int(task.time * task.fps)
774
775         if currentFrame < len(task.textures):
776             task.obj.setTexture(task.textures[currentFrame],1)
777
778         return Task.cont
779
780     def loadTextureMovie(self, frames, name, suffix, padding = 1):
781
782         return [loader.loadTexture((name+"%0"+str(padding)+"d."+suffix) % i)
783                 for i in range(frames)]
784
785
786 # If world is run by itself, the laucher code is embedded here.
787 # It is *critical* that this if statement remain so the world
788 # can run correctly from the actual launcher.
789 if (__name__=="__main__"):
790     import direct.directbase.DirectStart
791     from direct.gui.DirectGUI import DirectButton
792     import sys
793
794     def startWorld():
795         clickToStart.ignore("space")
796         clickToStart.remove()
797         intro.start()
798         #world.start()
799         #messenger.send('begin_end')
800
801     world = World()
802     intro = IntroVisual()
803     end = EndVisual()
804
805     clickToStart =
DirectButton(image="clicktostart.tif",command=startWorld,clickSound=None,rolloverSound=None,pressEffect=0)
805     clickToStart.setScale(base.getAspectRatio())
806     clickToStart.accept("space",startWorld)
807     clickToStart.accept("escape",sys.exit)
808     run()
809 # end launch code

```

## 8. 게임 인터페이스 Jam-O-Drum 의 분석

### 가. Jam - O-Drum 개요

Jam - O-Drum은 원형 또는 정사각형의 테이블 위에서 4인 (초기엔 6인용)이 동시에 게임에 참여할 수 있는 협동 게임 도구이다. PC에서 생성된 그래픽이 테이블 위의 프로젝터로부터 테이블에 투사되고, 각 사용자는 휠 형태의 방향지시 입력 장치(X축 가변)인 턴테이블(Turntable)과 이벤트 입력 장치(Y축 가변 가능)인 드럼 패드(Drum pad)를 이용하여 2차원 제어가 가능하도록 설계되어 있다. 특히 사운드에 주안점을 두어 각 사용자의 전용 스피커를 두고 있으며 MIDI(Music Instrument Digital Interface)를 통하여 연결된다. 본 보고서에서는 Jam - O-Drum의 개요와 하드웨어 구성을 살펴보고 국내용 Jam - O-Drum의 구성방안과 활용에 대하여 기술한다.

#### (1) Jam - O-Drum의 역사

Jam - O-Drum은 1998년 7월, 음악 협동성을 증진하는 프로그램을 개발하기 위하여 시작된 프로젝트였다. 비 서구문화에서 나타나는 집단적 음악 행위에서 영감을 얻어 Tina Blaine(CMU ETC 교수)이 Jam - O-Drum 이란 상호 연동 드럼테이블을 개발한 것이다.

Jam - O-Drum은 다수의 사람들이, 음악적 사전 지식의 유무에 상관없이 협동적으로 MIDI 장비와 실시간 그래픽 프로젝터를 통해 시각, 청각적 실제의 체험을 공유하도록 고안 되었다. 즉 협동적으로 음악을 체험하며 그래픽화 하는 도구로서 시작되었으며 게임 플레이어 또는 사용자의 자의식 증진과 비사회성 행동 특성의 개선에 주안점을 두고 있는 것이 특징이다.

## (2) Jam - O-Drum 프로젝트의 초기 목표

Jam - O-Drum의 초기개발 당시 목표는 음악과 감성의 가시적 체험도구로서 초심자들에게 집단 공동체적 음악 체험을 가능하도록 하였다. 다양한 음악을 다수의 사람들이 동시에 얼굴을 마주하며 연주하면서 색다른 음악과 음악의 시각화를 시도하였다. 또한 연주과정에서 단독으로 시도하는 것이 아니고 다수의 사람들이 동시에 참여하면서 협동심을 증진하도록 하였다. 나아가 사용자들에게 신체적 운동성을 부과하는 효과를 얻을 수 있도록 하였다

### 나. Jam - O-Drum의 구성

#### (1) Jam - O-Drum의 하드웨어

Jam - O-Drum은 크게 테이블부, 프로젝터부, 데이터 처리부 3부분으로 구성되어 있다. 테이블 부는 프로젝터로부터 그래픽이 전사되는 스크린이면서 입력 장치가 되는 양방향 휠과 드럼이 각각 4개(4인용)씩 부착되어 있다. 또 테이블 아래에는 3D 입체 음향의 스피커 4개가 개별적인 사운드 출력을 위해 부착되어 있다.

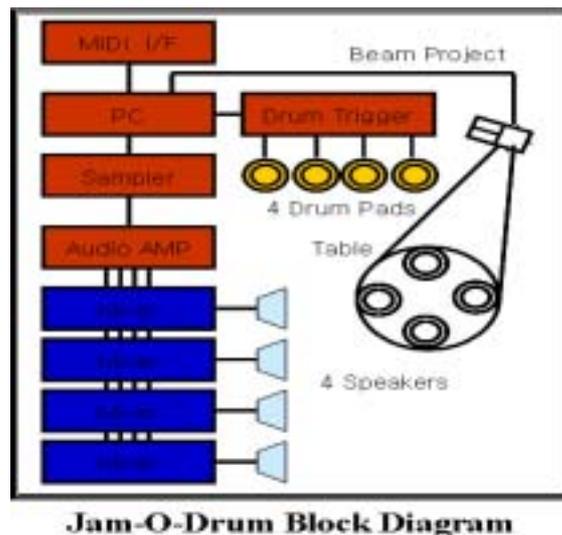


그림.1 Jam - O-Drum의 블록 다이어그램

프로젝터 부는 PC로부터 생성된 그래픽 데이터를 테이블에 뿌려주는 부분으로 천정 고정형과 테이블 일체형의 두 가지 모델로 설계되어 있다. 데이터 처리부는 PC를 중심으로 하며 휠과 드럼 패드로부터 제어 데이터를 입력 받고, 프로젝터와 스피커를 통해 그래픽과 사운드를 출력한다. 보조적으로 PC의 부팅과 프로그램 로드과정에서는 키보드와 마우스 그리고 PC 모니터를 사용하고 있었다. 그림.1은 Jam - O-Drum의 전체 블록 다이어그램을 보여준다.

### 가) 테이블의 설계

테이블은 철제 프레임 위에 합판을 붙여 제작하였다. 휠과 드럼 패드의 일체와 같은 높이로 하여 편평하게 하였고 흰색으로 칠하여 그래픽 이미지의 스크린 역할을 할 수 있도록 하였다. 프로젝터 분리형의 경우 테이블 아래에 스피커를 부착하였는데 사운드의 전달을 용이하도록 테이블에 구멍을 뚫고 스피커 보호 철재로 마감하였다. 그림.2와 그림.3은 각각 테이블 구성도와 스피커 부착 상태를 보여준다.

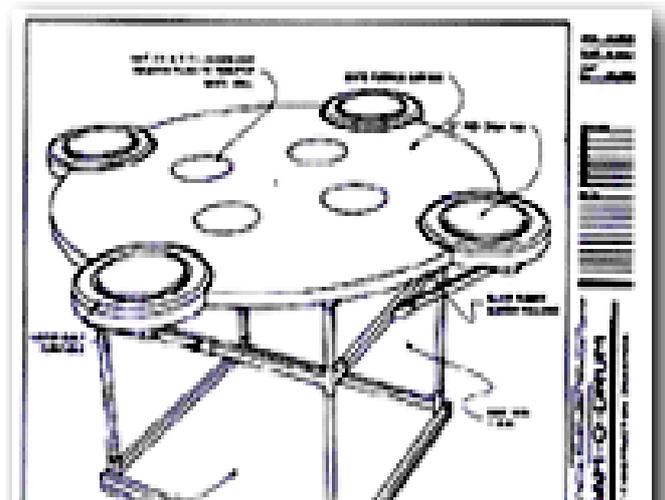


그림.2 테이블의 설계도



그림.3 테이블 아래의 스피커

Jam - O-Drum의 최신 버전의 경우, 사운드 시스템은 6.1 채널에 서브 우퍼를 사용하고 있어 박진감 넘치는 게임 사운드와 3D 입체 음향을 제공하여 게이머들에게 더욱 실감나는 게임을 할 수 있도록 설계되어 있다. 4개의 독립적인 스피커와 룸 스피커 그리고 서브 우퍼의 지원을 위해 사운드 믹서를 사용한다.

#### 나) 드럼 패드와 턴테이블

드럼 패드와 턴테이블은 Jam-O-Drum의 입력장치이다. 드럼 패드는 Jam-O-Drum의 초기 설계 목적에서 eh 알 수 있듯이 주로 사운드 입력 장치로 사용된 것이다. 그러나 단순한 사운드 입



그림.4 드럼패드와 턴테이블



그림.5 드럼 패드

력만이 아니라 게임의 이벤트 발생 입력으로 사용되기도 한다. Jam-O-Drum의 드럼 패드는 두껍고 부드러운 인조 가죽 형태의 고무랩으로 쌓여 있으며 10인치크기의 Single Zone Pad로 MIDI 드럼 모듈과 연결 된다.

그림.4는 드럼패드와 턴테이블이 테이블에 부착되어 있는 모습이다. 4명의 사용자를 구분하기 위하여 개별적으로 구분되는 컬러를 드럼패드와 턴테이블에 비춘 모습이다. 그림.5는 드럼 패드의 아래 부분으로 테이블의 프레임에 안정적으로 부착된 모습을 보여 준다. 드럼 패드는 잭과 라인을 통하여 그림.6의 드럼 모듈에 연결된다.



그림.6 드럼 패드와 연결되는 드럼 모듈

턴테이블은 드럼 패드를 보호하면서 동시에 좌우 방향 전환의

정보를 입력하는 입력장치의 하나이다. 구조적으로 턴테이블은 원형 베어링 위의 단순한 휠에 불과하며 정보의 입력은 턴테이블과



그림.7 턴테이블의 하부

동시에 돌아가는 Optical Encoder 에 의해서 이루어진다. 즉 턴테이블의 좌우 회전이 엔코더의 바퀴를 돌려 상대적인 변화 값을 얻게 되는 것이다. 턴테이블은 나무재질 위에 테이블과 같이 흰색으로 코팅되어 있으며 저항력을 크게 하기 위하여 고무벨트를 둘러렸다.

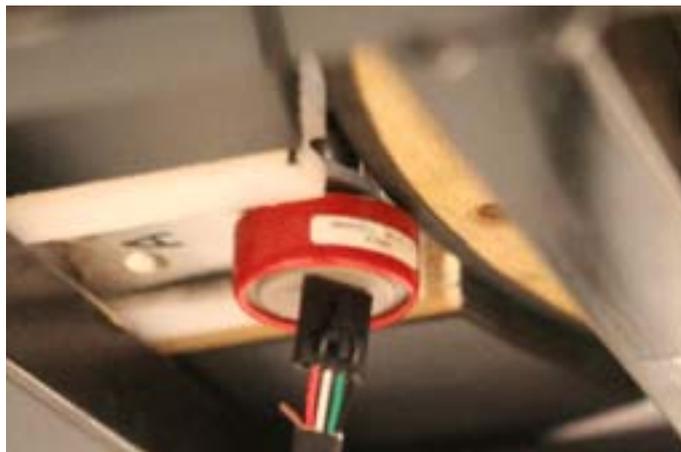


그림.8 고해상도 4핀 로터리 엔코더

그림 .7은 턴테이블을 뒤집은 모습으로 원형의 베어링이 깔려 있으며 외부에는 고무벨트가 감겨져 있다 로터리 엔코더의 출력은

라인을 통하여 그림.9의 Opti-PAC 이라는 Optical Interface for PC to Arcade Control 장치에 연결된다. 4개의 로터리 엔코더 출력이 하나의 Opti-PAC에 연결되고 Opti-PAC과 PC는 USB로 연결된다.

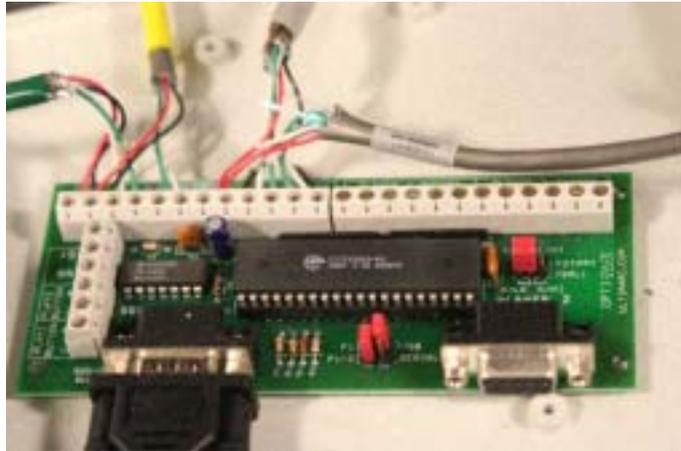


그림.9 Opti-PAC 인터페이스

## (2) Jam - O-Drum의 소프트웨어

Jam - O-Drum 용 게임 프로그램이나 응용 프로그램을 작성하는데 있어서 특별한 게임엔진을 사용하거나 특수한 프로그램 도구를 사용할 필요 없이 일반적인 PC 게임과 동일하게 작성할 수 있다. 일반 PC게임 과의 차이점은 그래픽이 전사되는 스크린 역할의 테이블 규격 때문에 정방형 그래픽 윈도우로 제한된다는 특성이 있다. 또한 사용자의 제어입력 도구가 드럼패드와 턴테이블이고 스크린 위에 위치하므로 드럼과 턴테이블의 위치를 정확하게 스크린에 전사해야 한다. 더불어, 사용자의 제어 입력에 따라 각 사용자의 전용 캐릭터를 사용하는 경우 드럼과 턴테이블에 전사되는 색깔을 구분하여 캐릭터와 컬러를 연계할 수 도 있다. 만일 불가피하게 정방형 윈도우를 사용할 수 없을 경우에는 4개의 드럼과 턴테이블 내부로 공간이 제약되기 때문에 활성 윈도우가 작아지게 된다.

앞에서도 언급한 바와 같이 현재 Jam - O-Drum 에서 제어입

력은 턴테이블과 드럼패드에 국한되어 있다. 이러한 단순함은 Jam - O-Drum 활용에 있어 나름대로 충분한 의미가 있으나 보다 다양한 제어 입력이 필요한 게임 프로그램의 경우에는 제약이 있을 수 밖에 없다. 따라서 게임 프로그램의 작성시 프로그래머는 게임의 제어입력이 많은 요구되는 게임은 지양하고 단순한 입력 만으로도 게임의 제어뿐 아니라 흥미를 유발 할 수 있도록 설계 해야 한다. 이후, Jam - O-Drum의 발전된 모형을 설계할 경우 매우 다양한 변형 포인트가 있지만 게임의 Jam - O-Drum의 본래 목적인 협동성이나 게임 참여자의 편리성을 고려하여 게임 프로그램을 작성해야 할 것이다.

Jam - O-Drum의 초기 개발 목적의 또 다른 중요한 포인트는 새로운 음악적 장치로서의 역할도 충분히 수행하는 것이었다. 그만큼 Jam - O-Drum의 음악적 출력 특성이 탁월하다는 의미이다. 그림.1의 Jam - O-Drum 블록 다이어그램에서도 알 수 있듯이 입력장치인 드럼 패드는 MIDI와 연결되는 디지털 드럼이고 또한 개별적인 스피커를 장착하여 사운드의 표현을 특히 강조하고 있다. 이런 하드웨어적 특성을 고려한다면 Jam - O-Drum은 음악적 요소를 게임에 반영하는 것뿐 아니라 게임의 새로운 장르인 음악게임용 장치로서도 충분한 응용 가능성이 있으므로 음악적 표현을 목적으로 하는 게임프로그램의 작성도 고려할 수 있다.

### (3) Jam - O-Drum의 게임 프로그램

그림.10은 Jam-O-Drum의 다양한 응용 프로그램과 게임 장면을 보여준다. 먼저 (a)는 Experience Music Project로 Jam-O-Drum의 사용자들이 그림 패드를 이용하여 음악을 연주하고 연주 음악의 입력 특성(주파수, 진폭 등)에 따라 다양한 그래픽이 테이블 위로 전사되는 협동 음악연주 또는 음악을 비주얼화한 프로그램이다. Jam-O-Drum 초기 목적의 달성을 위한 주요 프로그램으로 많은 의미가 부여된 듯 하다. 4. Jam-O-Drum의 응용에서 다양한 활용

방안을 논하고자 한다. (b)는 Circle Maze 로서 4개의 동심원 트랙이 있고 각 트랙에는 랜덤하게 발생한 통로(패스)가 선으로 그려져 있다. 각각의 트랙은 사용자의 턴테이블과 매칭되어 있으며 외부로부터 시작하여 중심으로 불이나 사람을 진행시키는 게임인데 각각의 트랙 통로가 연결되어야만 중심으로 전진할 수 있도록 고안되어 있다. 전형적인 Jam-O-Drum 협동게임 중 하나이다. 계



a) Experience Music Project



(b) Maze



(c) Quad



d) Brave Little Turtles (2005 한국 연수팀)

그림.10 Jam-O-Drum의 게임 및 응용 프로그램

임에 참여하는 4명이 각각 정확하게 트랙을 돌려 패스를 일치시키고 전진시켜 중심에 이르는 시간을 단축하는 게임으로 팀 대항의 기록경기라고 할 수 있다. (c)Quad 는 게임 플레이어가 병아리가 되어 사악한 여우로부터 동료 병아리를 구하는 협동 게임이다. (d)는 2005년 하계 한국 연수단의 한 팀이 프로젝트로 설계한 협동 게임이다. 위기에 처한 객체와 다른 객체가 도움을 주거나 구하는 미션이 해결 과제였다. 상황은 먼저, 거북이 가족이 소풍을 가는 도중 비바람과 벼락으로 헬리콥터가 요동하고 이때 거북알(baby)이 떨어진다. 순간 엄마 거북(mom)이 낙하산도 없이 알을 구하기 위해 낙하하는데 여기까지가 게임 인트로(Introduction) 부분으로 2D 그래픽과 사운드를 이용하여 작업하였다. 이후 부분

은, 헬리콥터에 남아있던 4마리의 용감한 아기 거북이(4 brave little turtles)들이 낙하산을 메고 엄마 거북을 구하기 위해 낙하한다. 제한된 시간 안에 4마리의 아기거북이 들이 모두 엄마 거북을 잡지 못하면 미션을 실패로 끝이 나고, 성공하면 엄마 거북 자체가 낙하산으로 변하여 거북 알과 어린 거북 4마리를 모두 구하고 끝이 나는 시나리오로 구성되었다. 인트로 이후 부분은, 각각의 거북과 엄마로의 접근을 방해하는 방해물(풍선과 구름)을 3D, 2D 그래픽으로 처리하였고 필요한 사운드의 일부는 팀원들이 직접 녹음하여 사용하기도 하였다. 게임제어를 위한 입력은 4명이 각각의 터테이블을 이용하여 회전방향을 결정하고 드럼패드를 이용하여 전진 방향을 제어하는 형식이다. 모두 제한된 시간 안에 엄마 거북을 잡아야만 미션이 종료되는 협동게임으로 짧은 기간에 작성되었기에 몇몇 미완의 문제는 남아 있었다 그러나 비교적 주어진 과제를 충실히 수행하였고 CMU-ETC 교수진과 관계자들이 직접 게임을 수행하고 난 후의 평가는 충분히 만족할 만한 수준이었다.

기타 위에 열거하진 않았으나 Jam-O-Drum을 이용한 많은 게임들이 학생들의 프로젝트로 생산되고 있었다. 게임을 구분하면 사운드 중심 게임(Jam-O-Elvis, Hip Hop, DJ sez, Musica), 사운드를 그래픽화한 비주얼 게임(Rhythm Breaker), 게임 플레이어들이 공동으로 미션을 수행하는 협동게임(Circle Maze, Musica, Quad, MATCH-N-HATCH), 그리고 자동차 경주 게임과 같이 개인적인 경쟁 게임(Rhythm Breaker, Car Racing) 등이 있었다.

## 다. Jam - O-Drum의 실제

### (1) Jam - O-Drum의 외관

다음은 Jam - O-Drum의 외관을 보여준다



(a) 상부 프로젝터



(b) 하부 테이블

#### 그림.11 프로젝터 천정고정형 Jam - O-Drum

그림.11은 초기 설계모델로 테이블 위의 프로젝터가 건물 천정에 고정된 형태이다. 상부 프로젝터의 위치에 따라 하부 테이블이 고정되어 이동성의 제약이 있다. 그림.12는 초기모델의 Jam - O-Drum을 개선하여 프로젝터와 테이블을 일체화 하였다. 테이블 위에 철제 프레임을 사용하여 원형 케이스를 만들고 그 안에 프로젝터를 설치하였다. 이모델의 경우 프로젝터로부터 테이블로 그래픽이 직접 전사되지 않고 케이스 내부의 거울을 통해 반사되는 이미지가 테이블에 전사 되도록 설계한 것이 특이하다. 프로젝터가 테이블과 일체화 되어 이동성은 확보되었으나 무게는 어림잡아 수백 킬로그램으로 여전히 이동에는 용이하지가 않아 보였다.



(a) 전체의 모습



(a) 상부 케이스 내부의 프로젝터(거울반사)

그림.12 프로젝터 일체형 Jam - O-Drum

## (2) Jam - O-Drum의 하부

Jam - O-Drum의 하부구성은 철재 프레임과 프레임 위의 테이블이 상자 역할을 하고 내부에 PC, MIDI 장비, 오디오 믹서와 스피커, 그리고 USB 인터페이스 장비들과 복잡한 케이블들이 위치한다. 그리고 상부의 프로젝터로 연결되는 비디오 케이블도 하부의 PC로부터 인출된다. 많은 수의 기기들이 좁은 공간에 위치하므로 매우 복잡하게 배치되어 있었는데 실험실 수준의 장비이기 때문이라 생각했다. 하나의 상품으로 만들어지기 위해서는 보다 치밀한 설계와 기기의 배치 및 배선의 정리가 필요할 것으로 여

겨진다. 그림 13은 Jam - O-Drum의 하부 구성을 보여준다.



a) 하부 프레임 구조



b) 하부 프레임 내부의 PC



(c) PC 후면의 케이블



(d) 오디오 믹서



(e) 케이블 허브(USB)



(f) USB 인터페이스

그림.13 Jam - O-Drum의 하부 구성

## 라. Jam - O-Drum의 응용

Jam-O-Drum 제작자인 Tina Blaine은 비서구인(non-western)들의 합동 연주에서 영감을 얻어, 공동으로 참여하며 음악을 만들어내는 새로운 음악 기기로서 Jam-O-Drum을 창안하였다. 그래서 초기의 주요 목적은 협동 음악기기로서의 역할이었으며 점차 음

악과 비주얼의 융합으로 발전하여 눈으로 보며 즐기는 디지털 기기로 발전하였다. 나아가 협동게임이라는 새로운 아케이드 게임의 장르를 만들었으며 미국 내 여러 미래박물관과 엔터테인먼트 전시관에서 전시를 이어가고 있다. 본 연수기간 동안에 Jam-O-Drum을 이용한 게임 개발이라는 팀 프로젝트를 수행하면서 Jam-O-Drum의 활용 분야를 숙고하였고 나름대로 다양한 게임 프로그램이나 응용프로그램이 개발된다면 다음에 논의하는 다양한 응용분야를 개척할 수 있을 것이라 생각한다.

### **(1) 에듀테인먼트 활용**

Jam-O-Drum의 가장 큰 장점은 협동적으로 게임이나 디지털 응용 활동을 수행할 수 있다는 점이다. 따라서 학교 공간에서 교육과 게임을 접목한 프로그램들이 Jam-O-Drum 용으로 개발된다면 높은 참여도와 흥미 있는 교육 도구로 사용될 수 있을 것이다.

또한 정신박약아, 특수학급, 언어장애, 행동장애, 학습장애, 부적응아, 자폐증, 정서불안 등으로 고통 받는 아동들을 위해서도 훌륭한 교육도구 치료도구로도 사용이 가능할 것이다.

### **(2) 실버 아케이드용으로 활용**

Jam-O-Drum의 또 다른 장점은 사용이 매우 쉽고 커다란 그래픽과 사운드를 충분히 활용할 수 있다는 점이다. 이 특성은 위 에듀테인먼트의 활용에서도 중요한 요소로 작용할 수 있으며 특히 노령화 사회로 들어선 우리나라의 현실에서 실버 세대를 위한 운동성과 오락성을 부여하는 아케이드 게임 도구로서의 활용도 기대할 수 있다.

### **(3) 가족 참여 오락기기로 활용**

복수의 사용자가 동시에 게임을 수행할 수 있으므로 온 가족이 참여하여 할 수 있는 프로그램이 개발된다면 공공의 장소에서나 아케이드 게임 장 기타 공개된 장소에서 건전한 가족오락 문화를 형성할 수 있을 것이다. 아직까지 오락과 게임에 대한 부정적인 이미지가 남아있는 현실에서 가족 단위의 게임활동은 이러한 고정관념을 바꿀 수 있는 기회를 제공할 수 있다.

#### **(4) 공동 협동심 배양에 활용**

조직사회에서 공동의 목적을 달성하기 위해서 가장 중요한 것이 협동심이다. 이를 위해 기업들은 여러 형태의 연수과정을 통해 협동심을 배양한다. 이와 같은 이벤트성 행사에 Jam-O-Drum과 적절한 프로그램의 지원에 따른 활용은 다양한 효과를 얻을 수 있다.