



## 포스트모템: <레지던트 이블 4>

(Postmortem: *Resident Evil 4*)

작성자: 요시아키 히라바야시(Yoshiaki Hirabayashi)

작성일: 2013년 6월 26일

[이 글은 가마수트라 자매지인 [<게임 디벨로퍼 매거진>\[1\]](#)의 2005년 10월호에 실렸던 글로, <레지던트 이블 4>의 시네마틱스(cinematics) 팀장인 요시아키 히라바야시가 이 게임 프랜차이즈에서 가장 주목받는 게임 중 하나인 <레지던트 이블4>가 직면했던 점검사항들과 도전과제들에 대해 논하고 있다.]

<레지던트 이블(Resident Evil)> 시리즈는 팬층이 아주 넓어서, 그들의 기대에 부응하기 위해 우리는 [<레지던트 이블 4>\[2\]](#)을 완전히 새로운 게임으로 만들기로 했다. 이 시리즈는 워낙 오래되었기 때문에 우리는 게임 개조를 위해 팬들과 개발팀 모두의 피드백을 진심으로 원했다. 공포를 경험하는 새로운 방법을 선보이고 더 무서운 적들을 만들고, 아이템을 사용하는 새로운 방법들을 구현하는 등등의 관점에서 모든 것을 다시 본다는 의미다.

이 포스트모템에서 사용할 한 가지 요소는 게임 개발의 원형으로서 성공적으로 그래픽 스타일을 만들어내는 것으로, 이는 우리의 가장 큰 도전이기도 했다. 이 요소에 수반된 것들과 우리가 이루어낸 방법을 몇 가지 구체적인 사례를 들어 설명하겠다.

### 잘된 점

#### 1. 컷씬 통합

프로젝트를 시작하면서 우리가 초점을 맞추었던 한 가지는, 게임의 플레이하는 부분에서 사전 렌더링된 동영상으로 옮겨가는 방법에 대한 것이었다. 잘만 된다면 수용도를 결정적으로 높일 수 있는 요소 같았다. 보통은 게임플레이가 컷씬(cutscene)으로 옮겨갈 때 그 차이가 눈에 잘 띈다. 이 때 플레이어가 게임의 실제 부분이 아닌 그냥 영상일 뿐이라고 생각하게 될 수 있다. 높은 품질의 컷씬 그래픽으로 시네마틱스를 즐기는 사람들에게는 어필할 지 모르지만, 플레이어가 게임에 집중하게 한다는 측면에서는 게임에서 컷신으로 넘어가는 순간이 게임의 흐름을 방해할 수 있다. 우리는 게임플레이와 게임 동영상 사이의 이동을 자연스럽게 할 수 있다면, 플레이어가 전체 게임 경험 동안 방해받지 않고 게임에 몰입할 수 있을 것이라고 생각했다. 우리의 해결책은 컷씬들을 리얼타임으로 유지하는 것이었다.

<레지던트 이블 4>에 구현한 액션 버튼 시스템은 실시간 동영상 사용에 아주 좋은 보완이 되었다. 우리는 액션 버튼을 컷씬에 포함시켜서 플레이어들이 게임의 동영상과 상호작용할 수 있게 해주었다. 전통적인 게임 시나리오에서는 시네마틱스가 시작되고 플레이가 중단되면 플레이어가 적극적인 참여자에서 방관자로 변한다. 플레이어는 영상에 주의를 깊이 기울이지 않거나 심지어 컨트롤러를 내려놓을 수도 있는데, 어느 쪽이든 우리가 바라는 바는 아니다.

## 2. 기술 발전

현세대 콘솔의 하드웨어는 그동안 기술적으로 크게 향상되었고, 우리의 기술 자체도 하드웨어의 잠재력을 최대한 활용할 수 있을 정도까지 발전했다. 기술적으로 말하자면, 이런 발전은 예전에는 미리 구현된 컷씬에서만 가능했던 장면들, 예를 들어 복잡한 얼굴 애니메이션을 실시간으로 표현할 수 있게 해주었다. 지금까지 우리에게서 우리가 <레지던트 이블 4>에서 달성했던 종류의 복잡한 애니메이션을 실현할 수 있는 처리 능력이 없었다. 단순히 하드웨어의 능력 밖의 것이었다. 우리는 이 문제를 프로그래밍과 데이터를 지능적으로 패키징하여 해결했다. 빛의 투사나 폭발 같은 특수효과가 많이 필요한 영역들에도 동일한 해결책이 적용되었다.

실시간 동영상을 사용한 덕분에 게임 사양과 디자인에 따라 스토리의 요소들을

변화시키는 것도 수월하게 되었다. 예를 들어 영상을 미리 렌더링해두었을 때에는, 동영상 속의 캐릭터나 적이 어떻게든 바뀔 경우 영상을 만드는 데 사용된 모든 시간과 에너지는 수포로 돌아가게 되었다. 그러나 실시간 동영상을 사용함으로써 우리는 그저 기존의 모델 데이터에 새로운 모델을 덮어쓰면 되었다.

### 3. 워크플로우 개선

우리는 시리즈를 처음부터 다시 만들어야 했기 때문에, 목표에 집중하기 위해 데이터 변환과 운영, 버그 수리 같은 일들의 워크 플로우를 개선해야 했다. 우리는 이런 업무를 보다 효율적으로 완수하기 위한 방법을 찾아야 했는데, 그래서 결국 XML 데이터 운영 시스템을 사용하게 되었다. 워크플로우 운영방식은 귀중한 시간을 절약해 주었고, 우리는 이를 게임 디자인의 창조적인 요소들에 활용할 수 있었다.

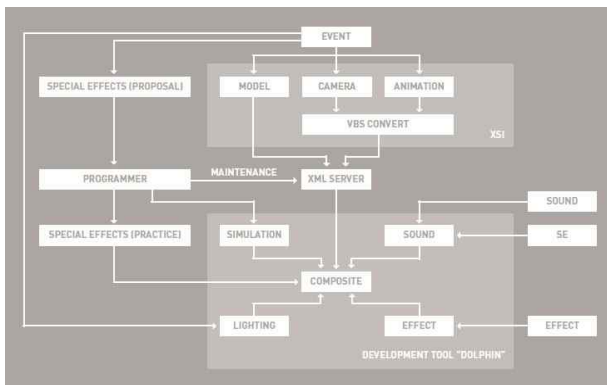


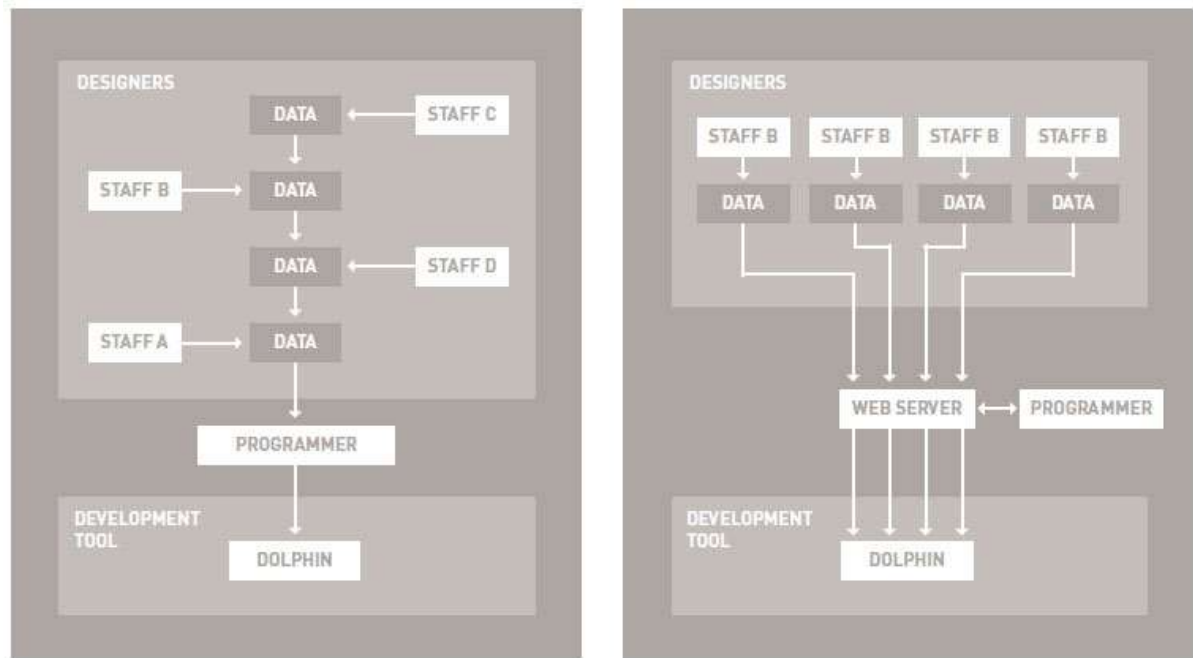
그림 1은 우리가 다른 것들 중에서도 특히 동영상 장면 전환, 장면 데이터 포스팅, 모델링, 텍스트처 데이터의 사소한 조정에 사용했던 시스템을 보여준다. XML 시스템은 우리가 데이터의 총량을 측정해서 지원에 부족함이나 넘침이 있었는지 평가할 수 있게 해주었다. 이

시스템은 또한 게임의 여러 장면 각각에 최적화된 모델을 여러 가지로 만들어 보는 프로세스를 수월하게 해주었다.

우리가 전에 사용했던 시스템(그림 2)을 보면, 새로운 시스템이 우리에게 도움이 되었던 이유를 정확하게 알 수 있을 것이다. 일본의 전통적인 개발 파이프라인은 아직도 꽤 위계적이다. 즉 예전의 시스템에서는 데이터를 개발 툴로 옮기는 프로그래머가 필요한데, 이렇게 되면 프로그래머의 작업 용량이 제한된다. 이런 상황은 시간을 엄청나게 낭비하게 했다.

기존의 시스템이 프로그래머들에게 작업량의 부담을 주었던 것과는 반대로, 새로운 시스템(그림 3)은 실제로 디자이너들이 더 자주, 더 직접적으로 기여할 수 있게

해주었고 그 결과 시간 낭비를 최소화할 수 있었다.



#### 4. 그럴듯한 이미지와 매력적인 캐릭터

그럴듯한 이미지를 만들어내는 것은 반드시 이미지들이 사실적이어야 한다는 것을 의미하지는 않는다. 오히려 이미지가 플레이어들의 마음을 끌어야 하며 게임의 세계 안에서 그럴듯해 보여야 한다. 예를 들어 <레지던트 이블 4>에서는 긴 치마나 긴 머리카락이 캐릭터의 움직임에 따라 자연스럽게면서도 약간 과장된 방식으로 움직인다. 이런 효과는 플레이어들을 게임에 몰두하게 하는 데 도움이 된다. 실제 세계에서 흔들리는 물체는 게임에서도 흔들린다. 그러나 현실에서는 그렇게 과장되어 아름답게 움직이지는 않을 것이다.

우리는 흔들림을 조금 반영했고, 모션이 아주 정밀하지는 않아도 플레이어가 기대하는 수준에 가깝게 만들었다.



더 매력적인 캐릭터를 만들어내는 것에 대해서는 의견이 아주 다양했다. 게임 디렉터였던 신지 미카미(Shinji Mikami)는 매력적인 캐릭터를 만들 때 가장 중요한 요소는 그럴듯한 표정과 동작을 만들어내는 것이라고 강조했다. 캐릭터는 주어진 상황에 적절한 감정과 표정을 표현해야 하며, 동시에 자신의 개성도 드러내야 한다.



우리는 캐릭터의 표정을 그럴듯하게 만든다는(그림 4와 5를 보라) 미카미의 목표를 달성하기 위해 엄청난 노력을 투자했고, 그 결과 캐릭터는 손가락 하나하나가 움직이고 이어져 있는 관절을 가지게 되었다. 그러나 우리는 여전히 더 정교한 디테일에 집중하고자 했다. 그러기 위해

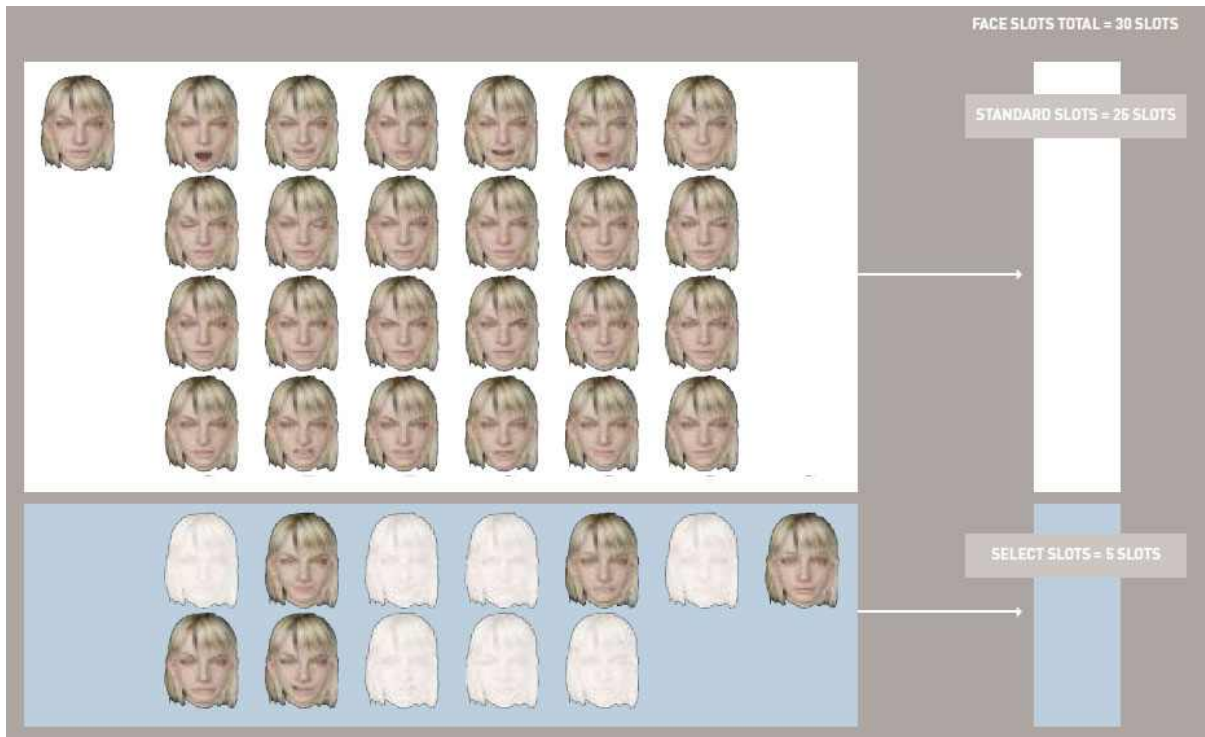
우리는 각 캐릭터마다 엄청난 수의 페이스 타겟(또는 표정)들을 사용했다.

우리는 캐릭터의 표정을 강조하는 조명 요소들을 추가함으로써 캐릭터가 느낄 수도 있는 긴장을 중요한 순간들에 옮겨 넣었다. 실제로 우리는 엄청난 수의 표정 패턴들을 만들어냈다. 또, 더 높은 품질의 특수한 텍스처를 사용해 표정들이 가능한 한 좋아 보이게 만들었다.

여주인공 에술리의 경우 총 36개의 페이스 타겟(face target)이 사용되었는데, 이것은 일반적인 숫자의 1.5배에 달한다. 늘어난 타겟의 효율성과 관계없이, 단 하나의 캐릭터에 그렇게 많은 데이터를 부여할 수 없었으므로 우리는 페이스 타겟 패키징이라는 방법을 사용했다. 36개의 페이스 타겟들이 준비되었지만, 그것들 모두가 모든 장면에 꼭 필요한 것은 아니었다. 우리는 각 장면에 필요한 페이스 타겟들을 그것들을 장면별로 구현하는 모델들에 포함시켰다.

처음에 나는 하나의 캐릭터에 사용될 수 있는 데이터의 양에 대해 프로그래머와 이야기를 나누고 각 캐릭터마다 페이스 타겟을 30개씩 사용하기로 했다. 우리는 이 페이스 타겟을 두 개의 범주로 나누었다(그림 6을 보라). 하나는 스탠다드 슬롯(Standard Slot)으로, 자주 사용되는 일반적인 슬롯이다. 다른 하나는 셀렉트 슬롯(Select Slot)으로, 각 장면에서 자주 사용되지 않는 타겟들이다. 우리는 하나의 패키지를 이 두 개의 슬롯으로 만들었다. 셀렉트 슬롯(Select Slot)의 표정의 수가

장면마다 달랐기 때문에 우리는 데이터 얼로케이션을 효율적으로 할 수 있었고, 그 결과 패키지에 항상 30개의 페이스 타겟만 있게 되었다. 장면에 따라 패키지를 변화시킴으로써 우리는 더 높은 품질의 얼굴 애니메이션을 만들어낼 수 있었다.



## 5. 게임플레이의 재창조

<레지던트 이블> 시리즈는 호러와 관련되어 있음에도 불구하고, 동시에 탄탄한 게임플레이와 높은 오락적 가치를 가진 것으로 알려져 있다. 우리는 게임의 오락적 가치를 높임으로써 <레지던트 이블 4>를 더 다양한 플레이어들에게 어필하는 게임으로 만들기를 원했다. 개발 초기에 우리는 시행 착오를 통해 게임이 으스스할 뿐 재미 요소는 적다는 것을 알게 되었다. 게임플레이 시스템과 공포 요소 모두를 다시 생각해봐야 했다. 우리는 이 게임이 소비자들에게 재미를 준다는 것을 확실하게 하기 위해 프로젝트를 네 번 다시 시작했다. 게임에는 공포 이상의 것이 있어야 했다.

게임플레이를 바꾸기 위해 우리는 환경의 상호작용, 그래픽 품질, 더 강렬한 액션 등의 측면에서 이전에는 실현 불가능하다고 생각했던 기능과 요소들을 구현하는데 도전했다. 우리는 <레지던트 이블>의 기존 원형에다가 게임 세계에 어울리는 다른

플레이 요소를 추가했다. 주인공 캐릭터의 근접 공격은 더 거칠게 구현되었는데, 왜냐하면 그는 경찰관이기에 이전 게임에서보다 접근전에 능숙해야 하기 때문이다. 적들은 더 지능적이고 훨씬 빠르게 움직이도록 구현되었다. 즉 적들은 무기를 집어 들고 문을 열 수 있다. 그리고 공격 계획에 대해 서로 대화한다. 이것은 <레지던트 이블> 시리즈의 공포 요소를 자극하는 새로운 방식이 되었는데, 플레이어에게 자신이 절뚝거리는 좀비나 단순한 몬스터가 아니라 감정과 무자비한 힘에 맞선다는 느낌을 주었기 때문이다.

우리는 시작 단계부터 프로젝트가 우리를 자극하고 도전하게 하도록 계획했다. 나는 우리가 우리 자신에게 박차를 가하고 그 목표를 달성하는 과정을 통해 팬들에게 재미있는 게임을 제공할 수 있었다고 생각한다.

## 잘못된 점

### 1. 단장하기

모든 게임에서 스토리라인의 목표는 플레이어에게 자신이 해야 하는 일에 대한 정보를 제공함과 동시에, 그 게임의 세계의 깊이와 그것에 대한 지각을 확장시키는 것이다. <레지던트 이블 4>의 스토리라인은 원래 실시간 동영상들에 의해 진행되었다. 우리는 게임 동영상에 신뢰감을 부여하는 데 꼭 필요한 이야기 요소들과 게임플레이를 결합시킨다는 목표를 달성하기 위해 고생했다.

게임은 시각적인 미디어인데, <레지던트 이블> 시리즈에서는 신선한 컨셉트를 만들어내기가 쉽지만은 않았다. 시리즈의 발전을 이어갈 비주얼을 개발하는 것은 더 어려웠다. 그래서 우리는 <레지던트 이블 4> 개발의 새로운 방향을 "컨셉트의 완전한 변화"라고 생각했다. 즉 우리는 이전의 모든 게임플레이 특성들과 공포 요소들을 조사하고 분석해서, 게임의 전체 컨셉트를 바꾸었다.



## 2. 데이터의 제약

<레지던트 이블 4>가 액션 중심의 게임이었기 때문에 우리는 한 번에 엄청나게 많은 수의 캐릭터를 등장시켜야 할 때가 많았다. (컷씬 이미지 A-D를 보라). 그 결과 우리는 이런 유형의 장면들을 위한 모델 텍스처를 주의깊게 적용해야 했다. 일반적으로는 적 캐릭터 하나가 약 3,000개의 폴리곤과 약 400K의 데이터 공간을 차지한다. 그러나 25명의 캐릭터가 동시에 화면에 등장하는 장면들이 몇 있다. 우리는 메모리와 이미지 저장 공간을 3,000개의 폴리곤 모델을 원래대로 유지하는데 필요한 수준까지 맞출 수가 없었다.



이 문제를 해결하기 위해 우리는 모델 당 데이터의 양을 장면마다, 심지어 장면 속의 컷들에까지 조절하기로 했다. 먼저 가장 많은 데이터가 필요한 모델(보통은 카메라에 가장 가까운 것들)을 구현했다. 그 다음에는 메모리 수준에 기초해 그것이 어떻게 그려지고 처리될 것인지를 결정했다. 전체 모델의 데이터 양을 제한하면 퀄리티가 떨어질 것이기에,

최상의 모델링이 필요한 일부의 모델들을 우선했다. 우리는 모델들을 텍스처링하는데 데이터를 분배하는 방법에 대해 세 가지 시나리오를 도출했는데, 그 후 이 시나리오들은 서로 결합함으로써 다양한 상황들에 사용될 수 있었다.

게임의 주된 적 가나도(Ganado)에 쓰인 고화질의 모델은 약 3,500개의 폴리곤으로 이루어져 있었다. 일반적으로는 동시에 많은 수의 가나도를 고화질로 구현하기가 어려웠다. 그래서 우리는 약 2,000개의 폴리곤을 가진 중간 화질의 가나도와 약 1,000 개의 폴리곤을 가진 저화질의 가나도를 추가로 만들었다. 이 세 개의 모델을 상황에 따라 교체했다. 텍스처도 같은 방식으로 취급했다. 고화질의 텍스처는 512x512 픽셀, 중간 화질의 텍스처는 256x256 픽셀, 저화질의 텍스처는 128x128 픽셀로 이루어졌다.

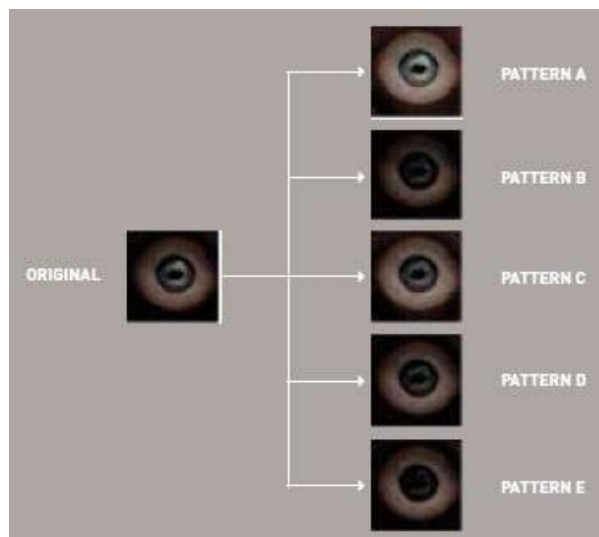
앞에서도 언급했듯이 이 모델들은 상황에 따라 교체될 수 있었는데, 장면에 필요한



모델들의 수와 주어진 장면에서 사용할 수 있는 데이터의 양이라는 두 가지 조건에 따라 이루어졌다. 우리는 데이터 용량을 항상 효율적으로 관리하고자 했으므로, 특정한 장면들에 쓰이는 카메라와 조명을 고려해 모델과 텍스처 데이터를 줄였다.

### 3. 텍스처 vs. 조명

우리는 모델 외에 각 장면의 텍스처도 조절했다. 다양한 조명들이 캐릭터를 비출 때 그 캐릭터가 보이는 방식을 어떻게 조절할 것인지와 관련해 문제들이 가끔 발생했다. 이런 문제는 특히 관통하는 조명이나 스포트라이트를 많이 사용할 때 발생했다. 우리는 이 문제를 각 장면이나 컷마다 텍스처를 만들어냄으로써 해결했다.



이 방법은 눈(그림 7) 같이 세세한 부분뿐만 아니라, 캐릭터의 다른 신체 부위와 배경 사물들에까지 사용되었다. 텍스처 조절은 오목한 대상이 너무 밝게 비춰지는 경우처럼 다이내믹 라이트가 사용되는 상황에서 발생하는 문제들을 해결하는데도 유용했다. 모델이 멀리에서 보이기만 하고 명암 정보가 거의 필요하지 않은 경우에는 데이터의 양을 축소했는데, 부피감이 시각적으로 잘 보이지 않으면 차이점이 눈에 띄지 않을 것이기 때문이다. 카메라의 위치로 인해 캐릭터의 그림자가 보이는 경우에는 저화질의 모델을 사용하지 않았다. 저화질 모델은 섬세한 조명의 디테일을 적절하게 구현할 수 없기 때문이다. 그 대신 우리는 저화질 텍스처의 고화질 모델을 사용해서 데이터를 적게 사용할 수 있었다.

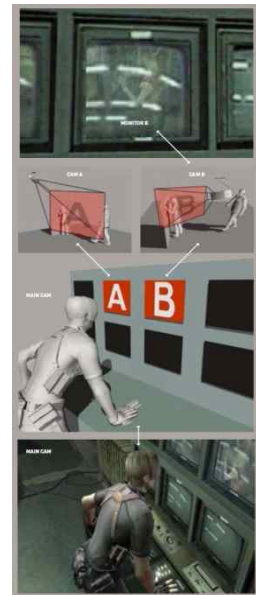
#### 4. 피사계 심도

해결해야 했던 또 다른 문제는 주어진 장면에서의 피사계 심도(depth of field)와 관련된 것이었다. 우리가 사용했던 한가지 방법은 효과를 흐릿하게 하는 것이었다. 즉 카메라의 움직임이 없을 경우 다양한 피사계 심도에 주어진 여러 이미지 조각들을 교체했다. 이 경우 처리 능력은 가벼웠고 하드웨어에는 지나치게 부담을 받지 않았다.

더 어려운 상황은 카메라의 움직임이 필요없는 장면에서 피사계 심도를 다룰 때 발생했다. 이런 경우 우리는 카메라로부터의 거리가 이미지를 흐릿하게(blur) 할 범위를 지나치게 정했다. 우리는 포토샵에서처럼 이미지 그 자체를 적극적으로 흐릿하게 했다. 이 방법은 인공적인 것이었지만, 실제 게임 내 카메라에 나타나는 피사계 심도를 이동시킬 수 있게 해주었다. 처리 능력이 첫 번째 경우에 비해 무거웠으므로, 우리는 크리에이티브의 관점에서 카메라를 움직이고 피사계 심도를 조절할 필요가 있을 때에만 이미지를 흐릿하게 했다.

#### 5. 멀티카메라의 부담

실시간으로 구현된 텍스처가 필요한 상황들이 몇 있었는데, 이는 여러 대의 카메라를 써야 했다. 여러 대의 카메라를 사용하면 한 대의 카메라만 사용할 때보다 처리 과정의 부담이 약 1.5배 늘어난다. 그러나 우리가 사용했던 워크플로우 시스템이 이미지들을 아주 효율적으로 조절할 수 있게 해주었으므로 우리는 이 방법을 사용하기로 했다. 이 기술은 레온이 스크린 상의 모니터를 들여다보는 장면에서 사용되었다. 모니터 상에 나타나는 "동영상"을 구현하기 위해 우리는 미리 구현된 동영상이 아니라, 실시간의 텍스처에 구현된 애니메이션 장면을 사용했다. 이 방법 덕분에 장면의 요소들을 변화시키기 위해 모니터 상에 캐릭터의 움직임을 구현할 필요가 없었고, 그 결과 아주 쉽게 변화를 줄 수 있었다.



보통은 하나의 컷씬에 단 한 대의 카메라 데이터가 사용되지만, 이 장면에서는 카메라 세 대의 데이터가 사용되었다(그림 8을 보라). 첫째, 메인 카메라는 전체 장면을 비춘다. 둘째, 카메라 A는 모니터 A에 나타나는 이미지를 비춘다. 셋째, 카메라 B는 모니터 B에 나타나는 이미지를 비춘다. 카메라 A와 카메라 B에 의해 비춰진 스크린 이미지들은 메모리에 저장하고 저화질로 낮춘 다음, 모니터에 쓰일 텍스처로 사용되었다. 이 작업을 마친 다음에 우리는 메인 카메라에 쓰일 이미지를 수집했다. 이 기술은 저장된 이미지 데이터에 대한 명쾌한 조절을 통해 선글라스와 차창 같은 것들에 반사된 이미지에도 사용되었다.

## 마치며

이상에서 보았듯이, <레지던트 이블 4>의 개발은 새로운 기술을 혁신하기보다 효율성에 기초해 진행되었다. 우리는 워크 플로우를 향상시키고 게임의 디테일에 집중적으로 초점을 맞춘 덕에 우리가 만들어내려고 했던 품질을 달성할 수 있었다. 프로젝트를 여러 차례 재시작했던 덕분에 우리는 이 게임과 서바이벌 호러 게임 전반에 대한 몇 가지 신선한 시각들을 취할 수 있었다. 나는 결과적으로 우리가 즐길 수 있으면서도 <레지던트 이블> 시리즈를 긍정적인 방향으로 발전시키는 데 도움을 주는 것을 만들어냈다고 생각한다. <레지던트 이블> 시리즈의 다음 게임은 차세대의 콘솔에 맞춰 계획되어 있으며, 도전과 기회의 새로운 상징이 될 것이다. 그 때에도, 우리 자신의 높은 기대를 다시 한번 충족시킬 수 있기를 바란다.