포스트모템: 아발란치 스튜디오의 <레니게이드 옵스>

(Postmortem: Avalanche Studios' Renegade Ops)

작성자: 악셀 린드버그(Axel Lindberg)

작성일: 2012년 6월 8일

모든 일은 아발란치 스튜디오(Avalanche Studios)가 <저스트 코스 2(Just Cause 2)>의 베타 버전을 테스트하는 중에 시작되었다. QA 부서로부터 버그 신고가 계속 쏟아져 들어오고 있었다. 번쩍이는 아이디어나 기막힌 기능을 생각해 낼 시간이 결코 아니었다. 사실, 게임 개발 단계에서 "베타"의 목적은 열정적이고 창조적인 마법사들을 잘 훈련 받은 버그 수리 군인들로 바꿔놓는 것이다.

베타 단계에서 하는 일은 게임을 더 낫게 만들겠다는 꿈을 키우는 대신 게임에 만들지 못한 부분을 그만 받아들이는 것이다. 베타는 모든 게임 개발자들이 겪어야 하는 필수적인 전투인 셈이다. 베타 단계를 거치지 않은 게임은 이상하게 돌아가고, 작동을 멈추고, 대개는 플레이 할 수 없게 될 것이기 때문이다.

이 전투 중에, 나는 유독 성가신 버그를 고치고 있다가 갑자기 회의에 불려가게 되었다... "여러분, 새 프로젝트를 시작하면 어떨까요? 세가(Sega)가 우리 엔진을 사용하고 오리지널 IP 가 있는 XBLA/PSN/STEAM 게임의 컨셉 피치를 들어보고 싶어해요. 어떻게 생각해요?"

우리의 대답은 "세상에나! 신나게 일해봅시다!"쯤 되었다. 짜릿한 순간이었다.

왜 레니게이드 옵스 Renegade Ops 인가?

게임 컨셉을 정하기 전에 우리가 제일 먼저 한 일은 현재 엑스박스 라이브 아케이드(Xbox Live Arcade)와 플레이스테이션 네트워크(PlayStation Network)에 어떤 게임들이 출시되어 있는지를 살펴 보는 것이었다.

이 때는 체어엔터테인먼트(Chair Entertainment)에서 <섀도우 콤플렉스(Shadow Complex)>라는 게임을 출시한 무렵이었다. 우리는 그 게임을 무척 좋아했다. NES 와 SNES 콘솔용으로 나왔던 닌텐도(Nintendo)의 고전 게임 <메트로이드(Metroid)>를 에픽(Epic)의 언리얼 엔진(Unreal Engine)을 사용해서 개조했는데, 현대적인 느낌을 굉장히 잘 입혔다.

새로운 IP 를 사용했고, 많은 메카닉(mechanic)이 오늘날의 게이머들의 기대에 부응하도록 업데이트되고 심화되었으나, 80 년대 <메트로이드>의 인기에 기여했던 핵심 철학과 컨셉은 완전히 보존되어 있었다. 그 게임이 불러일으킨 향수가 우리를 열광케 했다.

우리가 만들 게임도 같은 아이디어 선 상에 있었으면 했다. 게임 황금 시대의 컨셉에 현대적 감각을 덧입혀 2011 년으로 끌어 오는 것이다.

어떤 게임에서 영감을 받을 수 있을지에 대해 많이 토론했다. 대규모 예산을 들인 게임들에 비하면 규모가 매우 작을 것이었으므로, 아발란치 스튜디오의 강점과 경험에 부합하는 게임에서 영감을 끌어와야 했다. 액션이 매우 뛰어나고, 파괴감이 크며, 탈 것들이 혼재하고, 장난스러운 마초적 태도가 깃들어 있어야 했다.

머지 않아 주로 영감을 얻을 소스들을 골라낼 수 있었다. <Jackal(NEX 1988)>, <Desert Strike(Genesis/Mega Drive 1992)>, 오래 전 토요일 아침에 보곤 했던 만화 <지아이 조(GI Joe)>가 조건을 완벽하게 만족시켰다. 이로서 프레임워크가 명확하게 정해졌고, 팀 내에서 아주 초기단계부터 비전을 통일하는데 도움이 되었다.

포스트모템

프로젝트가 시작되기 전에 세웠던 목표들은 상당히 명확했다. 달성하기에는 어려운 점이 있었을지언정 말이다.

- 대규모 예산이 들어간 게임처럼 보이고, 들리고, 느껴져야 하지만 용량은 작아야 한다.
- 아발란치 엔진 Avalanche Engine)에 온라인 기술을 위한 토대(멀티 플레이어)를 마련해야 한다.
- 메타크리틱(Metacritic) 리뷰 점수가 85%를 넘어야 한다.

Renegade Ops 리뷰들을 읽어보면, 우리가 첫 번째 목표는 달성한 것 같다는. Tom Chick 은 1UP 의 게임 리뷰에서 다음과 같이 썼다 "...<레니게이드 옵스>는 <저스트 코스 2>에서의 화려한 액션 영화같은 느낌을 잘 살렸다. 그러나 탑뷰 방식의 트윈 스틱 슈팅 게임(twin stick shooter)의 한계에 갇혀 있다."

두 번째 목표 또한 달성했다. 현재 아발란치 엔진(에 멀티플레이어가 존재한다. 그로 인해서 우리가 이후 프로젝트에서 무엇을 할 수 있게 되었는지에 대해서 정말 말하고 싶지만, 내가 지금 그 정보를 공유하면 앞으로 어떤 정보도 공유할 수 없게 될 것이기 때문에 슬프지만 말할 수 없다.

세 번째 목표와 관련해서 <레니게이드 옵스>는 메타크리틱에서 81 점을 받았다. 나는 그 점수에 만족하고, 객관적으로 볼 때 합당한 점수라고 생각한다. 그러나 우리의 목표가 85 점이었기 때문에 목표를 달성하지 못한 것은 맞다.

이렇게 얼핏 보기에는 괜찮아 보인다. 이제 <레니게이드 옵스> 개발 과정이 어땠는지 좀 더 깊이 파고 들어가보자. 무엇이 잘 되었는가? 무엇이 잘 안 됐는가? 중간 결과물과 베타 버전을 만들어내면서 우리의 생활은 어땠는가? 계속 읽어주기 바란다.

잘된 점

프로젝트는 관련 프로세스와 개개인 모두에게 발전의 기회이다. 이 개발 후기는 그 기회를 다른 사람들과 공유하고자 작성되었다. 이 섹션이 그 목적에 맞게 흥미롭거나 유용한 정보를 제공하기를 바란다. 우리의 접근법에서 성공적이었다고 생각하는 측면에 대해서 먼저 이야기하는 것으로 시작하겠다.

1. 빌드 중심의 개발

<레니게이드 옵스>의 가장 중요한 목표는 재미있는 게임이어야 한다는 것이었다. 이 점이 우리에게 최우선 순위가 되었고, 우리는 팀 내에서 "재미"에 대해 많은 의견을 나눴다. 우리는 "재미(fun)"가 자연스럽게 모든 결정의 주요 요소가 되는 접근법을 찾고 싶었다. 이를 위한 가장 믿을만한 방법은 개발하는 동안 개발자들이 직접 게임을 아주 많이 플레이 해보고 "재미"를 감소시키는 요인들에 대해서 솔직하게 토론하는 것이다.

우리는 내부적으로 이러한 접근법을 "핵심 메카닉에 집중한 빌드 중심의 개발"이라고 불렀다. 이는 무엇보다 게임이 항상 플레이 가능한 상태로 유지돼야

함을 의미한다. 게임 플레이가 불가능하도록 만드는 충돌이나 비슷한 문제점들이 생기면 우리 팀은 즉시 하던 일을 모두 멈추고 그것을 고쳤다.

또 우리는 개발중인 게임의 최신 버전이 우리가 다음에 할 작업을 결정하는 요소가 되도록 했다. 보통 개발 목표를 기록한 서류에 주로 근거해 작업하는 것과는 다른 방식이었다. 게임을 플레이 해보고 나서 이렇게 말하는 식이었다: "좋아. 지금 운전감이나 슈팅감은 아주 좋아! 근데 슈팅용 지프에 질렸어. 더 크고, 재미있는 적이 필요해. 전투를 더 신나고 다채롭게 만들 수 있는지 보자고."

그리고 나서 우리는 가능한 적의 유형 리스트가 들어있는 고수준(high-level) 설계 문서를 확인했다. "탱크! 크고 무서운 걸로! 좋았어! 다음 주엔 이 탱크를 만들기 시작하자고!"

내 말을 오해하지는 마시라. 우리도 프로젝트에 대한 전체적인 기획이 있었다. 그 기획서에는 단계적으로 달성해야 할 목표들과 퍼블리셔에 대한 우리의 의무도 구체적으로 담겨 있었다. 다만 Sega 가 뭐라하지 않을 범위 내에서 우리의 본능을 따를 수 있는 여유가 최대한 많이 주어지는 방식으로 기획을 짜기는 했다 (Sega 덕분이다!). 예를 들어, 중간 목표를 이런 식으로 세웠다. "적 유형을 두 개 개발하여 적용한다." 그러나, 우리가 고수준 설계 문서에 열거된 유형 가운데 어느 것을 선택할 것인지는 구체적으로 밝히지 않았다.

2. 최소한의 문서화

그러면 왜 이렇게 야단법석인가? 음, 게임 개발은 "재미있는", "직관적인", "집중하게 하는" 같은 애매한 단어들을 특징으로 하는 상품을 개발하는 일이 모두 그러하듯, 매우 복잡하고 약간 혼란스러운 과정이다. 문서화에 너무 의존할 때의 문제점은 게임 개발이 본래 엄청나게 반복적인 일이라는 점이다. 지면상에, 또는 이론상으로 설계가 아무리 잘되어 있다 할지라도 실제 게임에서 그대로 구현하기 위해서는 트라이얼 앤 에러(trial and error) 방식으로 다듬어 가는 과정이 필요하다. 게다가 게임 플레이 시스템 상의 사소한 변화 하나가 잠재적으로 (그리고 종종) 게임 전반에 큰 영향을 미친다.

이런 것들이 멋드러진 문서를 작성하는데 시간을 쏟아 부을 생각을 사라지게 하는 주요 요인들이다. 이러한 문서에 담긴 내용들은 대개 아이디어를 게임에 적용해 테스트하는 순간 낡은 정보가 되어버리기 때문이다. 그래서 실제로 게임을 만드는 시간보다 서류를 다시 작성하고 업데이트 하는데 더 많은 시간을 보낼 수도 있다.

우리는 문서화에 시간과 에너지를 가능한 적게 썼다. 대신에 대내외적으로 말로 커뮤니케이션하고 지식을 퍼트려 나갔다. 게임에 대한 고수준의 아이디어와 그 아이디어를 어떤 식으로 구현할 것인지만 문서화 되어 있으면, 우리는 기록을 멈추고 게임 개발에 집중했다.

내부적으로, 우리는 팀 내에 개방적이고, 자유롭고, 자발적인 커뮤니케이션을 통해 협동이 이루어지는 문화를 만들어 내려고 했다. 누군가가 어떤 내용을 이해하지 못했을 경우, 우리는 서버에서 업데이트가 되어 있을 수도 있고 되어 있지 않을 수도 있는 서류를 확인하기 보다는, 시간을 내서 그와 얼굴을 마주하고 이야기하는 것을 선호했다 (서류가 업데이트 되어 있다면, 그건 누군가가 매일 서류를 업데이트 하는데 많은 시간을 쓰고 있다는 것을 의미하기 때문에, 내가 보기에는 업데이트가 되어 있든, 되어있지 않든 손실이었다.)

외부적으로, 우리는 Sega 와 매우 건강하고, 개방적인 관계를 맺을 수 있었다. 일주일에 여러 번 퍼블리셔와 정기 회의를 했는데, 어떤 종류의 토론이든 가능한 자리였다. 더 긴급한 사안이 생길 때에는 이메일을 주고 받는 것보다 전화로 이야기하는 것을 선호했다.

이에 더해, 중간 결과물들을 기록할 때에는 시청각 녹음장치를 사용해서 해당 아이디어나 결정에 이르기까지 팀에서 겪었던 과정을 녹음했다. 실제 문서에는 감당할 수 있을 수준만큼만 세부사항을 포함시키고, 아주 세부적인 사항들은 말로 녹음했다. 녹음된 것은 항상 참고할 수 있었고 다른 이해 당사자들과 공유할 수도 있었다.

3. 핵심 메카닉에 집중

우리가 프로젝트 시작 단계에서 결정한 또 다른 중요한 사항은 게임의 핵심 메카닉에 많은 노력을 기울이자는 것이었다. 우리에겐 그런대로 괜찮은 수준의 기능들을 무한정 생산해 낼 인력이 없었기 때문에, 가능한 한 오래 가져갈 핵심 메카닉 몇 가지에 집중할 필요가 있었다. 나는 프로젝트의 범위와 상관없이 핵심 메카닉이 훌륭한 게임을 만들어 내는 데 있어서 가장 중요한 요소라고 생각한다.

우리는 핵심 메카닉이란 기술의 관점에서 봤을 때 게임의 가장 단순한 형태라고 정의 내렸다. <레니게이드 옵스>를 개발할 때는, 위에서 아래를 내려다보면서 차량을 이용해서 발사하는 게임을 만들자고 결정했기 때문에, 핵심 메카닉이 매우 간단했다. 위에서 아래를 내려다보는 카메라 뷰, 차량 운전, 전투를 위한 적. 이 세 가지 메카닉 각각의 필수적인 요소들이 핵심 메카닉이 되었다. 한편, 업그레이드, 내러티브, 미션 등의 다른 메카닉과 기능들은 모두 핵심 메카닉을 여러 방식으로 뒷받침하게 될 것이었다.

<레니게이드 옵스>의 핵심 메카닉은 다음과 같이 정의했다.

핵심 기술

1. 플레이어 관점 (탑뷰, 카메라 위치와 기능)

2. 기본 인터랙션 (차량 운전 제어, 조준/발사)

3. 기본 과제 (적, 복구 시스템, 파괴, 복구 아이템)

그러나 내러티브는 핵심 기술이 아니다 (<레니게이드 옵스>에서 스토리는 게임 화면 밖에서 나오는 2D 만화에 캐릭터들 간의 대화를 들을 수 있게 하여 제시된다). 이것은 핵심 메카닉이 아닌 것에 초점을 두고 에너지를 들인 예이다. 이 때문에 우리는 꽤 많은 프로덕션 비용을 들여야 했다.

나는 내러티브 결과물이 마음에 든다. 특히, Giorgio Cantú 가 작업한 2D 원화가 좋다. 게임에 전반적으로 적당한 정도의 특색을 입혀주는 느낌이 들기 때문이다. 하지만, 우리가 시간을 되돌릴 수 있다면 내러티브 작업을 더 단순하게 했을 것이다.

그래도 전반적으로 우리는 핵심 메카닉 몇 가지에 집중해서 개발할 수 있었다. 핵심 메카닉으로 성공하는 것이 우리의 최우선 과제였다. 우리는 기본적인 드라이빙, 슈팅, 적의 행동이 경쟁 게임과 비슷한 수준이 되기 전에는 업그레이드 시스템이나, 부차적인 무기들, 버튼-매쉬(button-mash) 시퀀스, 기타 비슷한 것들을 작업하지 않기로 했다.

4. 측정 가능한 성공에의 집중

게임의 핵심 메카닉에 초점을 두는 것과 더불어, 팀원 모두가 구체적이고, 측정 가능한 결과를 낼 수 있는 업무에 집중하도록 많은 에너지를 기울였다.

각각의 마일스톤을 계획할 때, 우리는 그 하나하나가 어떻게 확실하고 측정가능한 방식으로 게임의 퀄리티를 향상시킬 수 있을지 스스로에게 물었다. 이것은 퍼블리셔 뿐만 아니라 팀을 위한 일이기도 했다.

우리는 제작 과정의 우선순위를 팀이 프로젝트 진행에 더 신이 나고 자신감이 생기는 방향으로 정하려고 노력했다. 그 덕분에 팀, 회사, 퍼블리셔 모두가 일찍부터 프로젝트에 강한 자신감을 갖게 되었다.

이로 인해 팀에는 예상치 못했던 긍정적인 에너지가 지속적으로 유입되었고, 개발 기간 내내 팀의 사기가 매우 높았다. 우리는 버티컬 슬라이스(Vertical Slice)를 제출하기 오래 전에 이미 이런 말들을 하고 있었다: "와! 이 게임 너무 재미있어. 크게 성공할거야! 지금 작업하고 있는 걸 어서 게임에 넣었으면 좋겠어!"

우리가 제출한 버티컬 슬라이스는 성공적이었다. 최종 게임의 축소판에 해당하는, 제대로 된 데모를 실제로 제출한 프로젝트는 그 때가 처음이었다. 우리가 버티컬 슬라이스용으로 선택한 미션은 이후에도 사실 크게 바뀌지 않았다.

앞으로 프로젝트를 진행할 때에도, 이렇게 긍정적 에너지가 좋은 결과를 낳을 수 있도록 노력할 것이다. 그러한 에너지 속에서 일하는 것 자체가 너무 좋기 때문이다. 다만 주의할 점이 있는데, 그건 '잘못된 점'에서 다루도록 하겠다.

5. 경험이 풍부한 팀

<레니게이드 옵스> 개발팀은 경력 개발자들을 주축으로 구성되었다. 그들의 경험이 프로젝트를 궤도에 올리고, 올바른 방향으로 빠르고 지속적으로 이끌어 가는데 주요한 요소였다. 처음에는 8 명으로 시작했는데, 주요 프로덕션 단계에 이르자 팀원이 최대 16 명까지 늘었다. 프로젝트 시작 단계부터 일을 시작한 8 명은 게임 개발의 여러 영역에서 역할을 해야 했다. 모든 이들이 그들을 가장 필요로 하는 영역에서 최선을 다해 열심히 일했다.

그러나, 경력 개발자들을 팀원으로 두었을 때의 가장 큰 장점은, 어떤 기능이나 시스템에 대해서 의견을 나누는 것만으로도 위험요소를 줄이고 낭비를 최소화할 수 있다는 것이다. 어떤 경우에는 시간을 소모하기만 한 적도 있었다. 그러나 실제

시스템에 먼저 적용하고 나서 제대로 돌아가지 않는 것을 깨달은 후 원점으로 돌아가는 것보다는 시간이 적게 들었다.

이런 식의 의견 교환 과정을 통해서, 각각의 해결책을 적용하기 전에 그것의 한계를 알 수도 있었다. 그 덕분에 각 시스템이 자리잡은 후까지 기다리지 않고 그 한계 안에서 게임이 돌아갈 수 있도록 게임 설계를 변경할 시간을 벌 수 있었다.

잠깐! 내가 좀 전에 게임 개발의 '초-반복적인' 특성에 대해서 쓰지 않았나? 게임을 만드는 데에 시간을 할애해야 한다고? 그렇다. 하지만 문서화에 관한 내용이었을 뿐, 전반적인 커뮤니케이션에 대한 얘기는 아니다. 커뮤니케이션은 다른 무엇보다 중요하다. 특히 팀원 각자의 경험을 활용함에 있어서 말이다.

잘못된 점

여행자여, 조심하라. 이제 길이 점점 어두워질지니... 이제부터 우리가 저지른 잘못과, 그것이 우리 팀, 프로젝트, 최종 게임에 미친 영향에 대해 말해 보겠다.

1. 탐욕: 모든 걸 가져야겠어

<레니게이드 옵스>를 향한 우리의 야망은 너무 컸다. 우리처럼 작은 팀과 소규모의 예산으로, 3 배는 더 되는 예산이 들어간 것처럼 느껴지는 게임을 만들려고 했다. 따라서 처음부터 우리가 성공할 가능성은 적었다. 그러나 다른 XBLA/PSN 게임들과 비교해보면 우리 예산은 꽤 많았다.

우리는 모든 것을 쏟아 붓자는 우리끼리의 합의에 따라 초과 근무를 용납할 수 없을 정도로 많이 했다. 아발란치 스튜디오의 주된 목표 가운데 하나가 무슨 수를 써서라도 초과 근무를 피하는 것이었기 때문에 윗분들이 마땅치 않아 했지만 말이다. 우리가 트리플 에이(triple-A), 최상의 품질에 대한 의지를 어느 정도 굽혔다면, 프로젝트를 진행하는 동안 실수에 대비할 여지를 더 많이 둘 수 있었을 것이다.

"트리플 에이(triple-A)"라는 말의 문제점은 뜻이 매우 주관적이라는 것이다. 어떤 사람들에게는 "트리플 에이"가 예산 규모의 문제일 뿐이다. 특정 액수 이상의 예산을 투자하면 그 게임은 "트리플 에이"급 게임이 된다. 또 어떤 사람들에게는 "트리플 에이"가 평단의 반응을 의미한다. 그런가 하면, 어떤 이들은 매출을 기준으로 삼기도 하고, 또 어떤 이들은 이상의 모든 것을 기준으로 삼는다.

기본적으로, "트레플 에이급 게임을 만듭시다."라고 말하면 그 말을 들은 사람들 각자는 저마다의 방식으로 그 말을 이해할 가능성이 높다. 이런 점이 <레니게이드 옵스>를 만들 때 문제가 되었다. 왜냐하면, 세가, 아발란치 경영진, 개발팀 각자가 "트리플 에이급 게임"에 대한 관점이 달랐기 때문이다. 그래서 저마다 기대치와 우선순위가 달랐고, 타협에 애를 먹었다. 모두가 가장 중요한 점에 대한 의견이 달랐기 때문이다.

2. 욕망: 열정이 지나칠 때

초과 근무와 관련해 말하자면, 팀원들은 게임의 성공에 매우 열정적이고 헌신적이어서 솔직히 잔뜩 흥분된 분위기였다. 비정상적이었다. 여기에서 내가 배운 점은 긍정적 에너지가 지속적으로 돌고 이것이 점점 더 큰 영향을 끼치는 것은

분명 좋은 일이지만, 모든 면에서 기대치가 계속 상승됨에 따라 팀이 제멋대로인 상태가 될 수 있다는 것이다.

우리가 중간 결과물을 내놓을 때마다 이전보다 더 잘하려고 애쓰면 애쓸수록, 우리는 스스로에게 훨씬 더 잘해야 한다는 압박을 가하게 되었다. 프로젝트 기간이 18 개월이었는데, 만약 이보다 훨씬 오래 진행되었다면, 우리는 언젠가 무너졌을 것이다.

다행히도 그렇게 되진 않았다. 대신 이런 경험을 통해 정상적인 근무 시간 내에서 긍정적인 에너지를 유지하며 최상급의 게임을 만들어 내는 개발자들이 얼마나 대단한지를 제대로 알게 되었다.

3. 과잉: 충분함을 넘어가면 너무 많다

기능 측면에서 <레니게이드 옵스>의 범위가 너무 넓었다. 언뜻 보기에 이것은 야망이 커서 최종 게임에 들어갈 기술의 양에 너무 열중한 첫 번째 잘못과 비슷해 보인다. 그러나 증상을 보면 완전히 다른 문제라는 것을 알 수 있을 것이다.

우리의 "욕망"(열정과 큰 야망) 덕분에 <레니게이드 옵스>의 핵심 게임 플레이는 프로덕션 가치가 높았고 트피플 에이급 예산이 들어간 느낌이 있었다. 이는 우리의 건강과 여가시간을 희생해서 얻은 성공이라고 주장하고 싶다. 우리의 "과잉"은 이것 말고, 인 게임 컷신(in-game cutscene), 분할 스크린, 온라인 멀티플레이어, 보스전 등과 같이 게임의 핵심을 벗어난 것들과 관련이 있다.

우리는 이것들을 게임의 핵심 기능들과 같은 수준으로 게임에 적용할 시간이 없었다. 이들 가운데 한 두 가지를 더 빨리 포기했더라면, 시간을 굉장히 많이 절약해서 다른 기능의 품질을 훨씬 더 높은 수준으로 만들 수 있었을 것이다.

혹는 그 시간에 핵심 경험의 일부로 계획했던 시스템들을 적용할 수도 있었을 것이다. 예를 들어, 각 레벨 내 체크포인트 시스템을 적용했다면, 플레이어들이 죽고 난 후 전체 미션을 다시 시작해야 할 필요가 없을 것이다. 또한, 버그가 너무

많이 생겨서 개발 완료 단계에서 제거해야 했던 “인 게임 컷신 건너뛰기"도 적용했을 것이다.

4. 무지: 항상 축복인 것만은 아니다

이제 와서 <레니게이드 옵스>의 플레이 성향을 보면, 3, 4 명 플레이어 온라인 모드의 사용이 너무 적다. 그 부분에 투자한 것에 비해 결과가 너무 미약하다는 생각을 하면 머리가 아프려고 한다.

게임 출시 전에 이걸 알아차릴 수는 없었던 걸까? 이런 점은 눈 앞에 뻔히 있는데도 잘 발견되지 않는다는 것이 신기하다. 육감과 게이머의 본능에 충실했음에도 불구하고 우리 팀 아무도, 심지어 회사도 3, 4 명 플레이어 모드를 좋아한 적이 없었다는 사실을 완전히 잊었다.

짧게 여러 차례 사격하는 식의 플레이는 재미있었지만, 서너 명의 플레이어가 함께 움직이면 게임이 너무 정신 없게 진행되었다. 모든 게임이 그렇지는 않지만, <레니게이드 옵스>는 그랬다. 그리고 우리는 앉아서 서로에게 “아, 이런! 우리 4 인 모드도 테스트해야 하잖아? 드르렁!”하며 불평하는 와중에도 그 걸 생각하지 못했다.

5. 허영심: 한계를 받아들이지 못하다

<레니게이드 옵스>는 처음부터 실수의 여지가 없는 프로젝트였다. 우리가 헌신적으로 따랐던 첫 계획에는 알파 버전, 베타 버전, 출시를 위한 한 달을 포함해서 프로젝트에 주어진 총 시간이 16 개월이었다. 시간이 충분하지 않을 것이라는 걸 직감했지만, 본격적인 프로덕션 단계에 이르면 상황을 바꿀 수 있을 것이라는 생각으로 그대로 진행했다. 7 개월 뒤, 본격적인 프로덕션 단계에 이르렀을 때, 버티컬 슬라이스의 품질과 수준은 매우 만족스럽고 자랑스러웠지만, 계획은 여전히 여유가 없었다.

결국 프로젝트는 두 달 연기되었고, 이는 몇 가지 부정적인 결과를 낳았다. 가장 중요한 것은 아주 중요한 홍보 행사를 2, 3 주 차이로 놓쳤다는 것이다. 마케팅 측면에서나 일반에 공개하는데 큰 도움이 될 행사였는데 아쉽다. 다른 하나는

우리가 프로젝트 기간 내내 계획을 따라잡느라 정신이 없어서, 프로젝트가 제대로 진행되고 있는지에 대해 깊게 생각해 볼 시간이 없다는 느낌을 받았다는 점이다. 우리는 게임이 출시되고 난 후에야 프로젝트를 돌아보고 평가할 수 있었다.

그렇다면 이런 일이 생기지 않도록 하기 위해서 무엇을 할 수 있었을까? 무엇보다 프로젝트 계획을 세울 때 개발의 마지막 세 단계에 각각, 최소한 한 달 이상의 기간을 배정해야 할 것이다. 각각 한달 안에 해치우는 건 현실적이지 않고, 우리가 직접 경험했다.

두 번째로, 우리가 일정을 따라잡느라 프로젝트의 진행 상황에 대해서 생각해 볼 시간이 없다는 느낌이 또 들 때에는, 작업을 모두 중단시키고 상황 파악에 신경을 써야 할 것이다. 어쩌면 개발 프로세스 상에 문제가 있을 수도 있고, 프로젝트 범위를 조정해야 할 수도 있다. 그 부정적인 순환 구조를 깨트리는 해결책이라면 어떤 것이든지 적합하다고 생각된다.

요약

<레니게이드 옵스> 개발에 참여할 수 있어서 정말 좋았다. 개발하는 동안 우여곡절을 겪으면서 웃고, 춤추고, 울고, 잠 못 자고, 포스트잇으로 만든 작은 비행기 서로에게 수백 개씩 날렸다.

Giant Bomb 의 <레니게이드 옵스> 리뷰에 나온 다음 문장이 한 마디로 정리해준다. "크고, 시끄럽고, 좀 바보 같고, 억수로 재미있다."

읽어주셔서 감사합니다!