



※ 본 아티클은 CMP MEDIA LLC와의 라이선스 계약에 의해 국문으로 제공됩니다

에잇몽키의 Darkest of Days 제작 후기 (Postmortem: 8Monkey's Darkest of Days)

Jeff Russell
2009. 11. 26

http://www.gamasutra.com/view/feature/4201/postmortem_8monkeys_darkest_of_.php

[이 심층적인 제작 후기에서 8 몽키 랩스는 PC 및 Xbox 360 시간 여행 슈터인 Darkest of Days의 제작에 관해 이야기하며 이 야심작의 제작 시 잘 풀렸던 일과 그렇지 못했던 일을 가감 없이 소개한다.]

시작하며

“시간 여행은 미래다”

- Mark Doeden

Darkest of Days(암흑의 시대)는 에잇 몽키 랩스(8 monkey labs)가 개발하고 팬텀 EFX 가 퍼블리시한 1 인칭 시간 여행 슈팅 게임이다. 이 타이틀은 2009 년 9 월 8 일 PC 와 Xbox 360 상에서 출시됐고 이번 휴가 기간에는 유럽에도 판매할 예정이다. 양 플랫폼 용의 무료 데모가 제공되므로 게임 맛보기가 가능하다.

Darkest of Days 는 아이오와 시더폴에 위치한 8 몽키 랩스가 출시한 첫 타이틀이다. 이 작품은 한 신생 스튜디오의 성장 과정의 산물이라고 할 수 있다. 8 몽키는 소규모였고 지금도 그러하다. 제작의 절정기에도 정식 직원은 10 명에 불과했다. 사람들은 우리를 ‘독립 개발자’라고 부르지 모르지만, 우리는 스스로를 그렇게 생각해본 적이 없다.

개발이 진행되는 동안 좋은 쪽으로 그리고 나쁜 쪽으로 수 많은 우여곡절을 겪었다. 예컨대 2008 년 6 월 8 몽키는 퍼블리싱 파트너인 팬텀 EFX 와 함께 흥수 때문에 사무실을 잃어버렸다.

며칠 사이에 8 몽키와 팬텀은 모든 장비와 직원을 두 번 옮겼다(1 차 백업 장소 역시 물이 계속 불어나 연결이 끊어져 버렸다). 한 주일 내에 사무실 건물을 2 번 잃어버렸다면 제대로 힘을 발휘하지 못할 거라고 생각할지 모른다. 그건 그렇지 않다.

다음 소개하는 이야기는 게임 제작, 사기의 고조와 저하, 관련자들의 기여에 대한 이야기의 극히 일부에 지나지 않는다. 하지만 이 개략적인 글에서 이 게임의 배경과 제작 과정이 어떠했는지는 알 수 있을 것이다. 어쩌면 이 글은 읽는 사람이나 쓰는 사람 모두에게 유익할 것이다.

잘 풀렸던 일

“처음부터 그럴 듯 하게 해놔야 한다. 그래야 그 후 굉장한 작품이 되도록 다듬을 수 있다.”

- Andres Reinot, Engineer

1. 대단한 팀

8 몽키가 설립된 첫날 마모셋 엔진(the Marmoset engine)의 개발을 시작했다. 사업 첫 해에는 이를 개발하고 시간 여행 모형 슈터(당시에는 이름이 없었음)의 시안을 제작하면서 보냈다. 이 시기에 개발진은 4 명이 전부였다. 이전에 게임 제작 경험이 있는 사람이 한 사람도 없었다. 모두 초보였다.

이는 직원의 수준이나 Darkest of Days 를 어떻게든 만들고야 말겠다는 결연한 리더십에 대한 적나라한 증거이다. 경험도 없었고 예산도 작았지만 가용 자원을 최대한 활용하는데 탁월했다. 팀원들은 여러 가지 일을 했고(특히 레벨 설계 시) 단순한 업무로 감당할 수 있는 수준을 넘어서는 능력을 보여주었다.

팀이 작았기 때문에 구성원들 모두가 게임의 많은 부분에 대해 주인 의식이 있었다. 그렇다 보니 자신의 능력을 최대한 발휘할 수 있는 환경이 조성됐다. 팀이 커져갈 때조차도 강한 주인 의식은 여전했다.

Darkest of Days 는 게임을 제작한 팀에서 유래한 독특한 색채를 띤다. 초보로서 어쩔 수 없었던 실수들도 분명 있었지만 좋은 점도 많았다. 제작 과정 중에 저격 임무, 훔친 제플린 레벨, 최종 대결에서 이용되는 굉장히 만족스러운 마이크로웨이브 건(microwave gun) 같은 것들을 기초부터 하나씩 만들었다. 대부분의 사람들이 우리 게임을 보고 독특하다고 이야기했고, 우리는 기분이 좋아졌다.



2. 아티스트 확보

게임의 아트 애셋(art assets), 특히 캐릭터는 많은 부분 외주 계약을 이용했다. 8 몽키 같은 작은 회사라면 그렇게 하는 게 맞다. 정식 현장 직원을 확보하는 데는 상당한 시간과 비용 문제가 따른다. 우리 같은 경우는 사람들을 아이오와로 옮겨 오도록 하는 게 쉽지 않았다. 그래서 아트의 많은 부분을 외주 업체에 의존했다.

회사의 수석 아티스트(현장직)는 다중적 종재자로서 아트 업계 등에서 확고한 거래선을 많이 만들었다. 프리랜서는 사용하기에 위험이 따르는 게 보통이고 직업 윤리나 역량이 천차만별이다.

아트 업계와의 탄탄한 연계를 바탕으로 8 몽키는 최고의 사람들, 나아가 프로젝트에 가장 잘 맞는 사람을 물색할 수 있었다. 인터넷 상에서 임의로 고른 사람들에게서 얻어진 결과물의 품질과 일관성에 놀라지 않을 수 없었다. 강력한 현장 아트 재능과 연계해 Darkest of Days 의 아트 보급로는 순조롭게 작용했다.

또한 엔진에 내장된 툴백(Toolbag)이라는 아트 툴 덕도 보았다. 팀 내에서, 그리고 여러 외부 업체에서 이를 이용했다.

8 몽키의 툴에 대한 철학은 최종 결과물과 최대한 비슷한 환경에서 일을 하는 게 가장 좋다는 것이다. 이는 게임 엔진을 단지 게임을 위해서만이 아니라 아트 툴로서도 이용한다는 의미다. 툴백과 레벨 에디터인 해비탯(Habitat)을 이용해 이를 현실로 만들었다. 이들 탁월한 툴이

없었더라면 우리가 이 정도 규모의 프로젝트를 절대 감당하지 못했을 것이다. 개발이 끝나갈 무렵 툭백을 아트 업계에 무료로 배포했고 반응이 굉장했다.

3. 전용 엔진(좋았던 점)

'병아리와 라디오를 가지고 로봇을 만들려는 사람처럼.....'

- Jake Splichal, Level Designer

마모셋 엔진은 Darkest of Days 에 이용하기 위해 만들어졌다. Darkest of Days 의 제작에 본격적으로 돌입하기 전에 이의 개발을 시작했다. 게임의 초기 시안은 기술적 측면에서 결점이 꽤 있었지만 프로젝트가 진행되는 과정에서 이 엔진은 스튜디오가 가진 최고의 자산이었다.

이 엔진은 유용한 기술 기반이자 틀셋 역할을 했는데, 스튜디오의 모든 사람들이 이를 속속들이 알고 있을 정도다. 대부분의 경우 마모셋은 Darkest of Day 가 필요로 하는 사항에 정확히 부합하는 유일한 해결책으로 인식되었다. 강력하면서도 이용하고 유지하기가 간편했다. 코드가 20 만 줄도 채 되지 않았다. 자라나는 스튜디오에 잘 어울리는 도구였다.

이 전용 코드는 게임에 매우 독특한 느낌을 가져왔다. 마모셋의 애니메이션 시스템, 조명 알고리즘 및 셰이더, 그리고 특수효과는 독창적이었다. 때문에 게임의 여러 소소한 측면에서 다른 게임들과 미묘한 차별화가 나타났다.

아울러 엔진 덕분에 자유로운 실험과 신속한 시안 제작이 가능했다. 코드베이스의 처리는 신속했다(전면 리빌드 시 걸리는 시간이 5 분이 채 안됨). 게다가 새로운 기능을 통합하는 일도 어렵지 않았다. 즉시 미리 보기 모드가 있는 레벨 디자인 용의 인-엔진 에디터와 결합해 이터레이션(iteration) 시간을 전반적으로 짜임새 있게 유지했다.

프로젝트가 마무리될 무렵, 엔비디아와 협력하여 Physx API 를 통한 GPU 가속 피직스(GPU accelerated physics) 지원을 추가했다. 게임의 PC 버전은 이 기술을 파티클 시스템(particle system)으로 이용하며, 파편과 연기 및 충격 효과를 새롭게 추가했다.

결론적으로 마모셋은 적은 비용으로 여러 유명 엔진에서 볼 수 있는 강력함과 기능들을 활용할 수 있게 해주었고 아울러 우리 기술을 제어할 수 있게 해주었다. 무엇보다 우리는 우리만의 탄탄한 기술 기반을 갖추게 되었고 이를 바탕으로 향후의 프로젝트들을 진행해나갈 수 있을 것이다.

4. 오랑우탄 같은 물골?

Darkest of Days 는 성공할만한 이유가 있다. 19 세기 미국의 한 군인이 리틀 빅혼 전투에서 때 이른 죽음을 맞기 바로 전에 시간으로부터 끌려 나와 미래로 소환된 후 여러 시대로 돌아가 종종 미래의 첨단 무기로 수 많은 사람들을 죽이며 역사를 바로잡도록 훈련을 받는다. 흥미롭게 들리지 않은가? 사실 흥미롭다.

Darkest of Days 는 이토록 흥미로운 가정을 바탕으로 하고 있고, 그것만으로도 시선을 끌 수 있다는 것을 알았다. 여러 시대를 서로 뒤섞어 정말 멋진 게임이 탄생할 가능성이 있었다. 팀의 모든 구성원들이 그러한 가능성을 보았다. 예산도 제한적이고 무명이었지만 마케팅 팀의 도움으로 프리뷰나 리뷰 보도 등 여러 출판물에서 이를 다루었고 PAX 에서도 좋은 자리를 차지할 수 있었다.

아무도 눈치 채지 못한 와일드 카드 하나는 에이전트 텍스터(Agent Dexter)였다. 원래 텍스터는 게임 안의 친구 내지 안내자로서 임무 목표를 하달하고 내러티브를 읽는 역할이다. 그의 존재는 초기 설계 문서에 이미 자리하고 있지만 그 당시 이를 대단하게 생각하는 사람은 없었다.

그 후 그의 다이얼로그가 만들어지기 시작했다. 텍스터는 게임자와 다투거나 그를 조롱할 뿐 아니라 온갖 외설적이고 촌스러운 재담을 늘어놓기도 했다. 완벽한 음성 연기로 팀과 게임자 모두에게 즉각적 호응을 얻었다. 다음 예시 구절을 보면 그가 어떤 성격인지 대번에 알 수 있다.

“이는 여기서 케케묵은 집단 성교 갈아질 거야. 미국 역사 상 가장 암울했던 시절로 곧장 가게 돼. 오랑우탄보다 더 추한 모습이지.”

- Agent Dexter, *Darkest of Days*



5. 숨가쁜 전투

초기 게임 디자인의 주된 부분은 일인칭 슈터의 넓고 생동감 있는 전쟁터를 묘사하는 일이었다. 사람들이 사실적인 전쟁 장면에 몰입되고 전면적 파괴와 생명 손상을 보고 압도적 감정을 느끼도록 만들고 싶었다. 슈팅 게임을 보면 5 명이 만나서 싸우는 장면은 아주 많다. 그런데 우리는 그 이상을 원했다.

첫 작품을 만드는 팀에게는 버거운 면이 있었다. 한번에 수백 명의 NPC 를 화면에 재현하고 묘사할 수 있어야 했고, 그에 따른 게임플레이 문제도 다루어줘야 했다. 모든 AI 가 잘 기능하는 것을 넘어 이들의 행동이 큰 집단에서나 작은 집단에서 또는 개인적으로 그럴듯해 보이도록 만들어야 했다.

렌더링 및 애니메이션 시스템은 수 백 가지 캐릭터에 따른 부하를 견딜 수 있어야 했다. 레벨 설계자는 대본에 나온 디즈니랜드 식의 레이아웃과 거대한 전투에서 빛어지는 총체적 혼란 사이의 가는 선 위에서 곡예를 펼쳐나갔다. 아울러 풀어야 할 크나큰 설계적 문제들도 있었는데, 가령 플레이어가 전장에 있는 백 명의 캐릭터 중 하나일 때 그에게 어떻게 힘을 부여할 것인가 등의 문제 같은 것들이었다.

성능 문제를 제어할 수 있게 되기 전까지, 그리고 몇 가지 게임플레이 문제에 답을 얻기 전까지 전전긍긍하며 보내던 시기도 있었다. 과연 성공할 수 있을지 확실치 않았기 때문이다. 게임에서 대량 살상과 거대한 규모에 대한 감각을 빼면 남는 게 없었으므로 이에 계속 들러붙을 수밖에 없었다.

마침내 우리 모두는 우리가 거둔 성과에 크게 놀라워하게 됐다. 한번에 20-100 명의 적과 교전하는 일은 지금까지 대부분의 액션 게임에서 경험할 수 없던 일이다. 아울러 결국 구현해낸 전쟁터 장면의 그 혼란스러운 느낌은 정말 대단하다고 생각한다. 이 웅장한 전쟁터는 분명 게임이 가진 강점이다.

잘 풀리지 않았던 일

“판도라 상자를 열어버린 게야. 흥포한 코뿔소의 XX 아래 처박힌 말벌 떼의 집을 건드려 버린 셈이지. 이제 어쩔 건가. 친구.”

- Agent Dexter, *Darkest of Days*

1. Xbox 360 포트

결국 Darkest of Days 를 PC 뿐 아니라 Xbox360 에도 올리기로 결정을 내렸다. 안 그래도 자원이 공색한 상태였는데, 어쩔 수 없이 외부 업체를 섭외해 8 몽키의 엔지니어와 함께 마모셋 엔진과 게임 코드를 Xbox360 에 올리는 작업을 했다.

계약 업체 직원들은 유능하고 근면했고, 회사간 의사 소통은 무난히 이루어졌다. 작업은 마침내 성공적으로 끝났다. 360 에 올렸을 때 처음 시도에서 MS 의 인증을 통과해버렸다. 그런데 뭐가 잘못되었느냐고?

우선 게임을 Xbox 에 올리겠다는 결정이 게임 제작이 본격화된 이후 1 년 만에 내려졌다. 이 때까지 엔진은 거의 2 년 동안 PC 위주로 개발이 이루어지고 있었다. 따라서 처음부터 기술 설계에는 콘솔에 대한 지원이 없었다. 콘솔 버전을 만들자는 결정은 사업적으로는 타당하지만 기술 기반에 대한 대대적인 수리를 요하는 일이었다.

다행스럽게도 대부분의 미들웨어를 양 플랫폼에서 유지할 수 있었다. SpeedTree, Physx, OpenAL 에는 모두 Xbox 360 틀이 있었다. 그래서 그래픽 코드, 리소스 로딩 및 관리, 메모리 사용에 초점을 맞춰 작업을 진행했다.

렌더링 코드는 거의 완전히 다시 써야 했으므로, OpenGL 기반의 PC 버전에 추가하여 보조 Direct3D 렌더러를 만들었다. 새롭게 스레드된 아카이브 기반 로딩 시스템에 맞춰 로드 시간을 다시 조정해야 했다(PC 로드 타임이 이제 엄청나게 빠르다는 나쁘지만은 않은 부작용 때문이다). 그리고 메모리 절감을 위해, 그리고 단편화 처리를 위해 전체 코드베이스를 세세히 검사했다.

그 결과 일정에 곤란을 겪었다. 포트 작업의 분량을 처음에 과소평가해서 일정을 그냥 지나치는 경우가 발생했다. 포트 작업은 PC 쪽의 일정 문제로 더욱 악화되었는데, 그곳에서 문제가 생기면 당연히 콘솔 버전에서는 훨씬 더 많은 지연이 일어날 수밖에 없었다. 8 몽키 엔지니어링 팀은 프로그래밍 리소스를 만드느라 예상보다 많은 시간을 콘솔 문제 해결에 할애했다.

가까스로 성공하기는 했다. PC 버전의 시각 효과와 컨텐츠는 콘솔 버전에서 대부분 그대로 유지할 수 있었지만 성능과 메모리를 위해 일정 부분은 크기를 줄여야 했다. 최종 콘솔 버전의 출시일 바로 앞에 놓인 마감일까지 기간이 너무 짧아서 그 때까지 밤을 새우거나 주말 작업을 하기가 일쑤였다.

근본적인 실수는 당연히 계획에 있었다. 애당초 Darkest of Days 의 콘솔 버전을 출시하겠다는 생각을 했다면(또는 비상 시에 대한 계획이 있었다면) 기술 및 아트를 초기부터 더 효과적으로 선택할 수 있었을 것이다. 그렇지만 뒤늦은 시도로 인해 여러 면에서 궁지에 몰렸고, 앞을 좀 더 내다봤더라면 피할 수 있었을 힘든 선택들을 강요 받았다.

2. 전용 엔진(좋지 않았던 점)

마모셋 엔진의 개발은 8 몽키가 잘 선택한 일이었지만 Darkest of Days 에 대한 가장 유리한 선택은 아니었을지도 모른다.

아무 것도 없는 상태에서 엔진을 개발하자면 시간이 걸린다. 많은 시간이 걸린다. 우리 경우는 매우 적은 비용으로 그렇게 했지만(불과 2 명의 엔지니어가 제작) 시간은 상당히 많이 걸렸다. 엔진의 기능이 모두 갖추어지고 쓸 만해지기까지 1 년 가까이 걸렸다. 이렇게 첫해를 보내는 동안 팀 내 나머지 사람들은 Darkest of Days 게임플레이의 시안 제작을 시도했다. 그러나 이 단계에서 우리의 노력은 대부분 엔진 개발로 인해 제약을 받았다. 어찌됐든 엔진 개발에 주력할 수 밖에 없었다.

제작이 본궤도에 올라 팀이 커지고 게임 콘텐츠를 본격적으로 열심히 만들기 시작했을 때조차 엔진이 항상 작용한 것은 아니었다. 도대체 완성된 것인지가 의심스러웠다(어떤 엔진을 과연 완성이라고 말할 수 있겠는가). 프로젝트 과정에서 엔지니어링 작업이 게임 코드가 아닌 엔진 유지에 전용되는 상황이 끊이지 않았다. 마지막 해 즈음이 되어서야 마모셋은 비로소 어느 정도 안정을 찾기 시작했다(Xbox360 포팅 작업은 제외). 그러나 이미 낭비해버린 시간은 어찌 할 것인가.

그렇다면 외부 게임 엔진을 쓰는 것이 옳았을까? 물론 그렇게 하면 개발 일정을 크게 단축했을 테고 더 빨리 시장에 진출할 수 있었을 것이다.

하지만 게임의 품질은 과연 더 좋았을까? 그건 말하기 어렵다. 게임의 외관이나 느낌은 분명 현저히 달랐을 것이고, 현재의 독특한 매력을 갖진 못했을 것이다. 또한 마모셋과 마찬가지로 기술적 문제가 전혀 없지도 않았을 것이고 특정 기술 설계 요건은 만족시키기 훨씬 어려웠을 것이다(드넓은 외부 환경, 수 많은 NPC 가 아주 좋은 사례이다).



3. 지옥에서 온 여행 대리인

“Darkest of Days는 누구든지 5,000 만 달러를 들여서라도 만들고 싶은 그런 게임이었다. 하지만 우리가 가진 거라곤 100 만 달러 밖에 없었다.

- Aaron Schurman, Writer / Producer

Darkest of Days 는 미국 남북전쟁, 리틀빅혼 전투, 1 차 세계대전, 2 차 세계대전, AD 79 년 베수비우스 산 폭발이라는 5 가지 명확한 시대를 배경으로 한다. 이러한 다양성은 게임이 지닌 장점 중 하나이다. 게임에서는 임무를 선택할 때 시대도 함께 선택할 수 있다. 하지만 이를 쉽게 이루어낸 것은 아니다.

자산 재사용이 정말 힘들었다. 각 시대는 그 시대 고유의 사운드, 레벨, 캐릭터, 건물, 환경 아트를 필요로 했다. 게임에 나오는 시대의 수를 따져보면 한 시대 고유의 아트 애셋이 최종 제품에서 '게임 시간'의 15-20%만 차지할 수 있다는 의미가 된다. 아트 팀은 기동, 캐릭터, 애니메이션을 쉴 새 없이 만들어야 했다. 게임 제작이 끝나갈 무렵이 되자 둘 또는 그 이상의 시대에 공통으로 사용할 수 있도록 애셋을 교묘히 만드는 법을 터득했다. 사방에 놓인 장애물 때문에 미쳐버리고 싶은 유혹을 간신히 참아냈다.

엔지니어링과 디자인 팀 역시 시대적 다양성에 시달리기는 마찬가지였다. AI 기능은 특정 시대 전용으로 만들어졌고 게임의 나머지 부분에서 사용되지 않았다(남북전쟁의 진형 전투가 아주 적절한 사례이고, 로마 병사들 또한 그 같은 사례에 속한다).

전투의 서로 다른 성질이 게임 디자인에 어려움을 던져주었다. 게임에는 24 가지 무기가 나오는데 대부분 시대 특유의 것이고 모두 균형을 잡아주고 조정을 해주어야 했다. 일부 미래형 총도 여러 시대에 사용되었고, 이들 또한 여러 시나리오에 맞춰 조정해야 했다.

Darkest of Days 는 게임 하나를 만들 자원을 가지고 5 개의 게임을 만드는 것 같았다. 그럴지 않아도 인원이 적은데 정말 버거운 일이었다. 무리였다고 말하면 온건하게 표현한 것이다. 이로 인해 게임의 특정 부분을 우리가 원했던 수준으로 퍼블리시할 수 없는 경우가 많았다. 다채로움은 이 게임에 잘 구현되어 있고 사람들의 관심을 끄는 주요한 부분이지만 품질을 위해 어느 정도 희생된 면이 없지 않다.

4. 인공 지능이라고?

"컴퓨터는 왜 로봇 같이 영리하지 못할까?"

- Mark Doeden, Director

전쟁터의 AI 는 서서히 게임 설계의 최대 난제로 다가왔다. 이에 따른 복잡성을 처음에 과소평가했다. 프로젝트가 진행됨에 따라 AI 에 대한 접근방식에 대대적인 변화가 불가피함을 절감했다.

Darkest of Days 의 AI 를 완전히 다시 만든 것만 3 차례나 된다. 각기 다른 엔지니어가 이를 맡아서 했다. 처음의 2 차례에서 얻은 교훈을 바탕으로 만들어진 3 차이자 최종적인 수정본은 가장 강력하긴 했지만 미흡한 점이 여전히 있지 있었다.

주요 설계 요건 중 하나는 성능이었다. 이전 버전들은 캐릭터가 10 개나 20 개 정도가 넘으면 제 성능을 발휘하지 못했다. 그런데 게임 설계에는 수백 개가 필요하다. 그래서 첫 번째 해야 할 일은 평균적인 성능을 달성하는 것이었다. 작동이 시원치 않은 일부 AI 미들웨어를 제거했고 그 결과 비전 및 경로 찾기 등의 하위 알고리즘을 다시 써야 했다.

새롭게 구현된 것들은 보다 효율적이고 조율하기가 더 용이했다. 하지만 지능적 캐릭터와 수많은 캐릭터 사이의 균형을 맞추기가 쉽지만은 않았다. 결과를 놓고 보니 넓은 전쟁터에서 효과가 매우 좋게 나타났다. 캐릭터는 저마다 스스로 생각했고 따라서 전투는 정말 짜임새가 있었다.

그런데 게임자가 멈춰서 소수의 NPC 를 자세히 보면 환상이 깨지는 경향이 있었다. 대규모 집단의 행동을 이상적으로 조정하다 보면 개개의 NPC 가 엉뚱하거나 그냥 방치된 듯 보였다. AI 의 수와 질 간의 끝없는 전투는 프로젝트의 나머지까지 계속됐다.

게임 캐릭터들에게 필요한 다양한 행동이 문제를 복잡하게 했다. 인간인과 군인, 인간과 동물, 그리고 무엇보다 3 가지 유형의 미래 전령사 등 서로 다른 5 가지 시대에 따른 캐릭터들이 있었다. 다채로움을 위한 노력이 다시 한번 진행됐고 일정 품질을 위해 많은 시간을 허비했다.

대규모 전투에서의 성능과 행동이라는 AI 시스템의 주요 목표가 마침내 달성됐다. 그러나 소규모 교전에서는 우리가 원했던 지능 수준에 결코 도달하지 못했다. 이 문제에 대한 해법을 알았더라면 게임 품질이 더욱 일관성을 띠었을 것이다.

5. 설계 시의 갈등

“서로의 파이에 손가락을 집어 넣는 일은 없어야 한다.”

- Jack Monahan, Level Designer

게임의 개념과 줄거리는 구성이 잘 되었지만 게임 설계의 구체적 부분들은 항상 명확하지는 않았다. 그래서 팀원들은 게임을 어떻게 균형을 잡고 조율을 해야 할지 망설이곤 했다. 각자가 팀과 상의도 하지 않고 혼자서 해법을 즉흥적으로 만들곤 했는데 이를 게임에 구현하면 설계 팀장이나 팀원들로부터 되짜를 맞았다. 이는 좌절, 중복 작업, 노력 낭비라는 결과를 초래했다.

이는 소유 의식이라는 동전의 이면이기도 하다. 팀원들이 개인적 기여에 대한 감정이 너무 강해서 변화가 필요할 때마다 마찰이 생기기 시작했다. 우리는 하나의 목표를 향해 나아가야 했다. 그런데 모두의 노력을 한곳에 집약하는데 때때로 어려움이 있었다.

완성으로의 길은 멀었고 게임이 좋아 보이지 않거나 실행이 잘 안되었던 시기도 있었다. 아직까지 별로 좋아 보이지도 않는 무언가를 위해 계속 작업을 한다는 것, 그리고 모든 것을 완전히 변경해야 할 필요성에 저항하는 것은 난감한 일이 아닐 수 없다. 결국에는 게임의 기본에 매달리도록 작업 초점을 유지할 수 있었다. 하지만 핵심 미케닉을 다듬을 필요가 있었을 때 이를 고치기 위한 아이디어와 제안에 수많은 시간을 허비하며 의견 교환을 했다.

외부 게임플레이 테스트가 없다가피 했다는 것도 설계 불확실성의 또 다른 원인이었다. 설계 테스트를 위해서라기보다는 버그를 찾기 위해 외부 테스트가 이루어졌고, 게임 개발자라면 알아야 하는 ‘재미 있는지 여부’라는 중심적 질문에는 집중할 수 없었다. 레벨 설계, AI, 무기 균형에 대해 실제 행해진 것보다 훨씬 더 많은 테스트를 행할 수 있었을 것이다. 팀원들이 상당한 내부 테스트를 하긴 했으나 설계 논쟁을 해결하는데 거의 도움이 되지 못했고 팀에 알려진 것도 얼마 되지 않았다.

프로젝트가 끝나갈 무렵이 되자 설계 문제가 어느 정도 사라졌다. 보다 많은 논의를 거쳐 보다 정기적으로 설계 이터레이션이 이루어졌다. 외부 테스트도 늘리려고 했다. 모든 사람들이 일에

흠뻑 취해 갔고, 게임 프로젝트에서 으레 그렇듯이 제작 막바지로 갈수록 최상의 성과들이 나타났다.



끝마치며

“맥주 마실 시간이군. 다음 차편에 올라타 시원한 맥주나 마시러 가는 게 어때?”

- Agent Dexter, *Darkest of Days*

이 프로젝트에 문제가 있었느냐고? 물론이다. 우리는 작은 팀이고 예산도 작다(2,000 만 달러가 투입되는 게임이 아니다). 게다가 초보자들이 겪는 실수들도 있었다. 신생회사로서 새로운 기술로 새로운 IP 를 창출하고 있었다. 불가능한 일이 아니냐는 소리도 여러 번 들었다. 처음부터 험난한 여정을 예상하고 있었다.

어찌됐든 게임은 독창적이었고 여러 측면에서 기존의 틀을 깨뜨리는 것이었다. 판에 박힌 슈팅 게임에서 벗어나 뭔가 새롭고 차별화된 무엇을 만들고 싶었다. 그렇게 하는데 성공을 거뒀다.

사람들의 반응은 다채롭다. *Darkest of Days* 의 독특한 느낌을 좋아하는 사람들도 있고 단점들을 지나치지 않는 사람도 있다. 이는 좋은 점에서나 좋지 않은 점에 있어 기존의 많은 슈터들과는 다르다. 이 게임은 현재 주류와의 차별성에 매력을 느끼는 충성스러운 매니아층을 작게나마 가지고 있다.

Darkest of Day 의 제작기간을 1 년 더 늘렸더라면 재미있었을 것이다. 예산 압박 때문에 그렇게 할 수 없었다. 그러나 결과는 흥미로웠을 것이다. 프로젝트가 진행됨에 따라 사람들의 숨씨가 갈수록 좋아지는 것 같았고 막바지에는 정말 감이 확 오는 느낌이었다. 우리의 새로운 기술과 경험을 이용해 우리가 어디로 나아갈지 지켜보는 것도 재미있을 것이다.

Darkest of Days 는 결국 돌풍을 일으켰다. 그곳에 없었다니 정말 유감이다. 무료 T-셔츠 등 모든 게 다 있었는데.

게임 정보

정식 개발 직원: 4-10 인

임시 계약직: 14 인

예산: 170 만 달러

개발 기간: 시안 및 엔진 개발 1 년, 제작 2.5 년

출시 일자: 2009 년 9 월 8 일

플랫폼: 윈도우 PC, Xbox 360

권장 기기 사양: 2.6 Ghz 인텔 코어 2 듀오, 2GB RAM, GeForce 8800

적용 소프트웨어: 비주얼 스튜디오, 모도, 머드박스, 마야, XNormal

적용 기술: 마모셋 엔진, OpenGL, PhysX, SpeedTree, OpenAL

크기: 코드 20 만 행, 소스 콘텐츠 80GB