사후분석: Crystal Dynamics의 *Tomb Raider: Underworld*

Eric Lindstrom
가마수트라 등록일(2009. 9. 2)

(http://www.gamasutra.com/view/feature/4122/postmortem_crystal_dynamics_tomb_.php)

[최신 Tomb Raider 타이틀의 제작자들이 제공한 이 심층 사후 분석은 원래 금년 초에 Game Developer 잡지에 실렸던 것이며, 이 큰 호평을 받은 최신판을 제작하는 과정에서 무엇이 잘 되었고 무엇이 잘못 되었는지를 정확하고 설명하고 있다.]

Tomb Raider: Underworld 의 개념 단계는 그 전신인 *Tomb Raider: Legend* 가 최종 QA 단계에 있고 제출이 임박했을 때 시작되었다. *Tomb Raider: Legend* 에 대한 예비 조사는 매우 고무적이었으며, 기존의 기능에 활용할 수 있는 사용하지 않은 잠재력이 아직도 많이 있다고 생각했다.

이 잠재력은 더 짧은 시간에 더 크고 더 뛰어난 Lara Croft 모험을 만들기 위해 기존의 기능을 활용하는 것과 콘텐츠에 집중하기에 충분하다는 생각이 들었다. 여러 가지 면에서 이것이 팀이 이룩한 것이지만 게임 개발에서 항상 일어나듯이 현실은 예상보다 복잡했다.

개발 중에 잘못된 것의 상당 부분이 예상했던 함정과 관련이 있었고 이를 피하려고 노력했는데도 불구하고 함정에 빠졌다는 것은 특히 흥미로운 일이다. 때때로 실수를 저지르기는 하지만 다른 경우에는 어쩔 수 없는 일을 겪고 이를 해결하기 위해 최선을 다하기 때문에 대부분의 사후분석에서는 단지 "우리가 잘못한 것"이 아니라 "잘못된 것"을 말한다는 점에 주의를 기울이는 것이 중요하다.

작년에 Game Developer 에 실린 기사(2008 년 12 월 호의 "What Went Wrong?")에서는 특히 왜 게임 개발에서는 동일한 실수를 반복해서 저지르는 것으로 보이는가에 관한 질문을 던졌다. 이를 고려하여

알려진 문제를 피하기 위한 우리의 방법이 어떤 식으로 부족했는지의 측면에서 "잘못된 점"의 일부를 논의할 것이다.

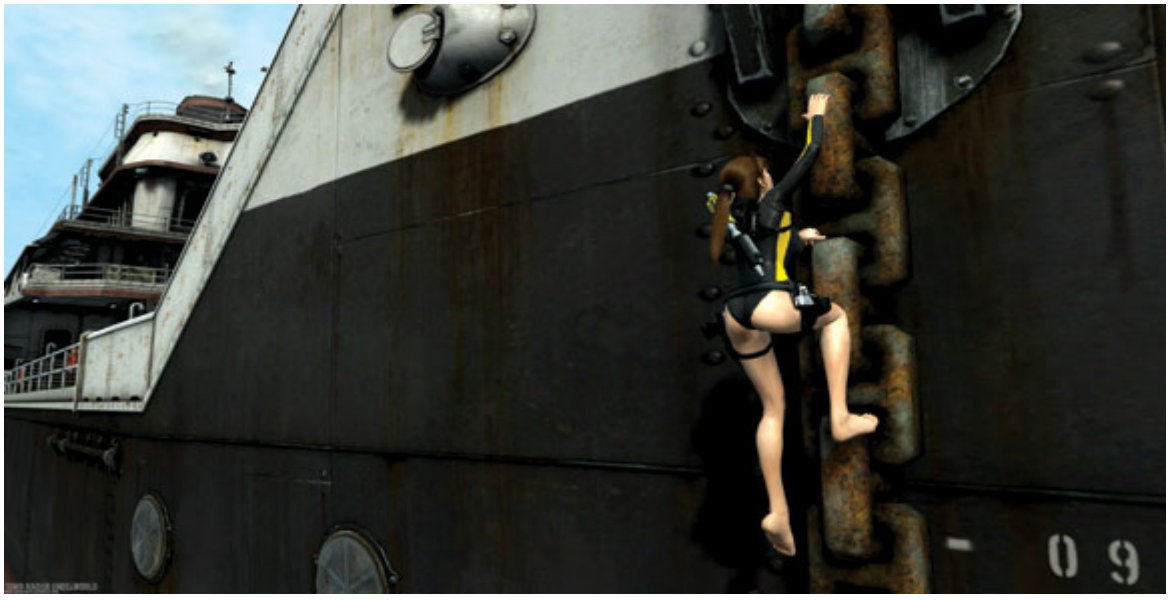
잘된 점

1. 긴 알파(Alpha) 기간

우리는 마지막에 마무리 시간을 더 많이 갖기 위한 여지를 만들기 위한 비교적 짧은 베타(Bea) 기간에 대한 준비를 하고 해결되지 않은 프리프로덕션(preproduction) 문제를 처리하기 위하여 예외적으로 긴 알파(Alpha) 기간을 수립했다. 이 때문에 아트 제작과 관련하여 좋은 결과가 나왔으며 게임이 그토록 보기 좋게 나오게 된 이유 중의 하나이다. 또한 우리는 프리프로덕션 단계에서 나온 상당한 양의 디자인 결함을 극복했으며 코드 면에서는 핵심 기능의 대부분을 기대에 맞출 수 있게 되었다.

이 긴 알파 기간 중에 우리가 올바르게 한 또 한 가지는 범위를 여러 차례 축소했던 것이다. 이것은 우리의 첫 번째 "차세대" 타이틀이었으며, 우리는 복잡성의 문제를 여러 번 과소평가했다. 우리는 게임이 너무 크다는 평가 및 결정을 내린 다음 준비해 놓은 한계에 맞추기에 충분한 양만큼 콘텐츠를 축소하곤 했다. 그 다음에 또 두 달 후에 게임이 아직도 너무 커서 범위를 추가로 축소해야 하는 상황에 직면하곤 했다.

원래 게임을 위해 계획된 거의 모든 기능 및 영역이 규모만 축소되어 최종 버전에 포함되었으며 그 수가 줄어든 방식으로 서로 연결되었다는 사실에서 알 수 있듯이 게임은 이 정도의 축소를 처리할 수 있도록 디자인되었다. 우리는 중요한 요소 전체를 빠뜨리지 않고 이곳 저곳에서 사소한 부분을 제거하여 범위를 축소할 수 있었다.



2. **초점**

후속 작품을 만들 때 내용을 조금 다르게 하고 기능을 조금 추가시킨 상태에서 이전 작품과 똑같은 게임을 만드는 것은 쉽게 저지르는 실수이다. 똑바로만 가게 하는 것은 줄이고 자유 탐사는 더 늘린다는 것 같은 새로운 중요한 목적을 세웠다는 사실에도 불구하고 우리는 대체로 이런 함정에 빠졌다. 하지만 팀의 많은 사람들은 팀을 규합하는 한편 게임을 위한 고유 판매 제안으로 사용하기 위하여 초점이 추가로 필요하다는 것을 알았다. 이 결과로 서사적인 탐험 퍼즐의 개념이 나오게 되었다.

여러 층의 연결된 퍼즐을 이용하여 대규모의 게임 내 장치 및 영역을 만들어서 이 게임은 이전의 *Tomb Raider* 게임들과 비교해서도 뛰어난 표현력을 가지게 되었으며 게임 전체에 게임 노력을 기울이는 것에 대한 시험대도 얻게 되었다. 이로 인해 이러한 퍼즐에 전념하는 하위 팀을 만들게 되었으며, 이러한 퍼즐은 복잡한 구조를 갖는 것으로 밝혀졌다.

이 필요성을 예상하고 처음부터 적절하게 계획을 세웠다면 더 좋았겠지만 커져가는 우려를 파악하고, 이 특별한 하위 팀을 만들고, 더 많은 직원과 자원을 여기에 투입하고, 전담 프로듀서를 배정했다는 사실은 우리가 개발 중에 예상하지 못했던 문제를 얼마나 잘 해결했는지를 보여주는 하나의 예이다.

3. 제작 유연성

Tomb Raider: Underworld 라는 엄청난 것의 방향을 크게 바꾸는 것은 어려운 일이었지만 제작 과정의 분해에 직면하여 우리는 많은 성공적인 변화를 이루었다. 처음에 우리는 이전 게임에서 얻은 교훈을 바탕으로 *Tomb Raider* 활동 영역의 대부분을 구성하는 정글의 폐허를 설계하는 최상의 방법을 알고 있다고 생각했지만 현실에 부딪혀서는 다시 혼란에 빠졌으며 그에 따라 변경을 했다.

하나의 분명한 예가 레벨 디자인 및 아트와 관련이 있다. 처음부터 우리는 모두 재미있고 아름다우며 믿을 수 있는 환경을 만들기 위해서는 첫날부터 두 분야 모두 협력하여 동작하게 만드는 것이 중요하다는 데 동의했다. 이전의 모험은 디자인에서 생성된 블록 메쉬(block mesh)나 아름답게 조성된 무덤으로 시작했는데, 디자인에서 생성된 블록 메쉬에서 나온 형상은 그럴 듯한 폐허로 전환하는 것이 극히 어려웠고 아름답게 조성된 무덤은 재미있게 기어오를 수 있는 기회나 적절하게 제작된 게임플레이를 제공하지 못하였다.

우리의 해결책은 멕시코나 태국 같은 지역별로 레벨 디자이너와 환경 아티스트의 짝을 짓고 재미있는 한편 미적으로도 적당하도록 만들기 위해 환경 아키텍처에서 반복하면서 서로 협력하게 하는 것이었다.

이것이 올바른 방향이라고 밝혀지기는 했지만 우리의 초기 과정 중의 상당수는 해결되지 않았다. 하지만 이러한 초기 시도에 결함이 있었다는 사실을 받아들이려는 의지 덕분에 진행 중에 있는 제작 워크플로(workflow)를 변경하는 큰 일을 몇 차례나 수행할 수 있었다.

과정을 이렇게 자주 변경하는 것은 때때로 혼란스럽고 좌절감을 느끼게도 했지만 우리가 잘못된 패러다임을 고수했다면 상황은 훨씬 더 나빴을 것이다. 결국 우리의 제작 방법은 완전하지는 않았지만 처음부터 끝까지 우리를 이끌어 줄 것이라고 생각했던 것보다는 나았다.

4. 메트릭

플레이어가 그토록 많고 다양한 환경 요소와 그토록 많은 방식으로 상호 작용할 수 있는 *Tomb Raider: Underworld* 와 같은 게임에서는 메트릭(metric)이 매우 중요하다. 메트릭은 Lara Croft 를 그리드에서 내리고 Lara Croft 가 현재 우리가 갖고 있는 유동적으로 움직이는 캐릭터로 발전하는 것을 막았던 트랙터 컨트롤(tractor control)을 제거하는 기반을 구성한다.

Tomb Raider: Legend 는 개발 과정 중 후반부까지 점프 거리, 돌출부에 대한 매개변수 및 그 밖의 메트릭에서 변화를 겪었고, 그 결과로 디자이너와 아티스트는 양쪽 모두 매우 많은 시간 동안 재작업을 해야 했으며(아티스트가 가장 힘들었다) 우리는 후속 작품에서는 이를 피하기로 결심했다.

우리의 수석 레벨 디자이너와 시스템 디자이너는 개발 초기에 전체 세트가 정의될 때까지 플레이어 상호 작용의 모든 요소에 대한 메트릭을 수립하기 위해 공동으로 작업했다. 일부 변경은 나중에 이루어졌고 허점이 발견되어 수정할 필요가 있었지만 전체적으로 볼 때 우리가 큰 변경 없이 메트릭을 유지하고 시행한 정도는 지난 노력에 비해 크게 향상된 것이었다. 이 성공이 없었다면 우리가 게임에 포함시킨 게임상의 부동산의 양은 크게 줄어들었을 것이다.

5. 공정성

이 게임에 대한 작업을 하는 사람들 중 대부분은 팀의 규모, 협력의 정도, 그리고 우리가 시각표에 맞추어야 했던 게임 자산의 양의 측면에서 볼 때 이렇게 큰 프로젝트에서 일한 적이 없었다. 여기에 이렇게 재능 있는 사람들이 얼마나 열정적이었고 최종 게임의 품질에 노력을 했는지도 더하면 그 결과는 폭발적이었을 것이다.

실제로 견해의 차이가 있었는데 이 중 상당수는 큰 차이를 보였고 분명히 상당한 스트레스가 있었으며 가끔 파국에 이르기도 했지만 전문가적 기질은 극도로 높았으며 이는 사람들이 어울리는 것을 훨씬 넘어서는 방식으로 제작에 큰 긍정적인 영향을 주었다.

범위를 축소하거나, 재작업 또는 작업 취소를 하거나, 프로세스 변경이 필요할 때 사람들은 성숙함을 보여주었고 자신만을 내세우는 일은 적었다. 우리는 게임에서 누군가가 좋아하는 요소를 삭제하려고

하는 것에 대해 화를 내지 않았다. 모든 사람들이 제출 전의 미칠듯한 버그 수정 기간(bug-fixing days)을 포함하여 프로젝트 내내 이성과 공정함을 유지했다.

이보다 작은 프로젝트와 약한 스트레스도 사람들과 팀을 망쳐놓았기 때문에 이것은 잘된 점의 목록에 오를 만하다. 이 같은 프로젝트에서 많고 긴 갖가지 고난을 손상 입지 않고 헤쳐나가는 것은 다른 요소들과 마찬가지로 프로젝트의 궁극적인 성공에 기여했기 때문이다.

잘못된 점

1. 기술의 공유

Tomb Raider: Underworld 를 시작할 때 Crystal Dynamics 는 스튜디오로서 내부 개발의 성배, 즉 미래의 모든 게임을 위한 출발점으로 사용하기 위한 견고하고 강력한 공유 코드 기반을 추구하기로 결정했다. 이 독점 엔진은 우리의 게임에서 요구할 공통 기능을 제공할 수 있는 전담 엔지니어가 확대 및 유지할 것이며, 팀 프로그래머는 자신의 게임에 고유한 기능에 중점을 둘 수 있다.

Tomb Raider: Underworld 와 공유 코드 기반은 둘 다 *Tomb Raider: Legend* 에 바탕을 둘 것이기 때문에 우리의 빡빡한 제작 시각표에도 노력을 합칠 수 있을 것이라고 스튜디오는 믿었다.

Tomb Raider: Legend 를 만들었던 사람들 중 상당수를 포함하여 많은 재능 있고 부지런한 프로그래머들이 공유 코드 팀에 투입되었기 때문에 기술은 문제가 아니었다. 여기서 가장 큰 문제는 과도한 야심이나 복잡한 의존성이 아닌 것으로 나타났다(물론 이들이 문제였던 것은 분명했다).

문제는 소유권과 우선순위에 더 큰 관련성이 있었다. 팀 내에서는 일정이 어긋나기 시작해서 되돌려야 할 필요가 있을 때는 제작 일정을 다시 맞추기 위하여 함께 모여서 임무를 다시 정의하고 필요한 모든 집합적인 변경을 할 수 있다.

하지만 공유 기술 그룹은 팀에 있지 않았다. 이 그룹의 강령은 팀에게 서비스를 제공하는 것이었지만 서로 충돌하는 요구를 가진 복수의 팀에게 서비스를 제공해야 했다. 각 팀의 관점에서 볼 때 공유 기술 그룹은 우리의 수석 엔지니어에게 보고를 하지 않는 프로그래머들의 핵심 그룹이었다. 이것은 우리가 청원과 설득을 통해서만 영향을 줄 수 있을 정도로 의존도가 컸으며 이들의 진행 상황을 살펴보는 것이 제한되었기 때문에 일정을 수립할 수 없는 경우도 종종 있었다.

처음부터 우리는 *Tomb Raider: Underworld* 의 기반을 발생기의 전개되는 코드 기반에 두는 것이 막대한 위험, 즉 커다란 잠재적인 함정이었다는 것을 알고 있었다. 그렇다면 우리는 왜 눈을 크게 뜨고 이 함정에 걸어 들어갔는가? 당시에는 잠재적인 보상이 위험을 감수할 가치가 있으며 적당한 사람들이 문제를 해결하려고 노력하고 있는 것으로 보였기 때문에 우리가 직면하고 있었던 과제 중의 일부가 궁극적으로는 장래에 더 많은 효율을 낼 수 있을 것이라는 것을 알면서 공유 기술 그룹과 함께 전진하여 우리가 할 수 있는 최선을 다했다.

하지만 궁극적으로는 모든 알려진 위험을 극복할 수 있는 우리의 능력을 과대평가했기 때문에 우리가 두려워하던 일 중 일부가 실현되었다.

2. 리팩터링(Refactoring)

초기의 엔지니어링 작업 중의 하나는 플레이어 코드를 더욱 견고하게 만들고 새로운 기능을 받아들이는 능력을 늘리기 위하여 플레이어 코드를 리팩터링하는 작업이 포함되었다. 이 노력은 우리의 기능 집합을 효율적으로 확장할 수 있게 하기 위해서는 중요했지만 이 리팩터링이 상황이 계속되는 동안 플레이를 할 수 없게 되는 방식으로 엔진을 분해하고 재조립한다는 것을 의미한다는 것을 당시에는 이해하지 못했다.

디자인 팀은 레이아웃 작업을 수행하는 편안한 기간을 예상하고, 상호 작용을 실험하고, 더욱 발전된 설정을 만들고, 그렇지 않을 경우 *Tomb Raider: Legend* 에서는 가능하지 않았던 탐험적이고 반복적인 디자인에 참여하고 있었다. 하지만 우리의 프리프로덕션 희망은 플레이어 기능의 상당 부분이 오프라인이 되면서 사라졌다.

일상적인 지연과 일정 차질이 이 어두운 기간 동안 지속되었으며, 그 동안 엔진 부품은 여기 저기 흩어져버렸다. 문제는 우리가 많은 핵심 플레이어 기능이 동작하지 않는 상태에서 제작에 들어가고 알파 단계에 깊이 들어가야 하게 될 때 복잡해졌다. 이것은 초기 디자인 및 레이아웃 노력에 매우 심각한 영향을 주었으며 이 효과는 최종 제품까지 내내 계속되었다.

근본적으로 이것은 잘못된 의사 소통의 문제였는데, 잘못된 의사 소통은 모든 프로젝트에 잠복해 있는 전통적인 문제점이다. 우리는 처음부터 이 잘못된 의사 소통을 해결하려고 노력했으며 우리의 제작 규모를 생각하면 상당히 잘했지만 이 특별한 불운은 되돌릴 수 없는 정도에서 고유한 것이었다. 의사소통을 향상시킨다는 것은 미리 정보 전달을 향상시킨다는 것에 관한 것이지만 동일하게 중요한 것으로서 잘못된 의사소통이 나타날 때 이를 빠르게 해결하는 것이었으므로 우리는 이것을 우리가 반복했던 전통적인 실수 중의 하나로 생각하지 않는다.

모든 것을 다 잡을 수는 없으며 리팩터링의 효과는 슬며시 빠져나갔으며 일단 시작된 다음에는 되돌릴 수 없었다. 이것은 특히 나쁜 잘못된 의사 소통이었으며 우리는 그저 참고 견딜 수 밖에 없었다. 결국 많은 귀중한 실제 디자인 시간이 시작할 때 사라졌으며 문제를 일찍 파악했다면 선적된 게임에 훨씬 더 나은 레이아웃을 갖게 되었을 것이다.

3. 프리프로덕션

우리의 프리프로덕션은 이전의 두 문제, 즉 공유 기술 노력과 리팩터링이 이전에는 플레이할 수 있었던 구조를 플레이할 수 없게 만든 방식 때문에 흔들렸지만 우리의 선적 일자를 이동하고 있던 시점에서는 선택이 없었다. 이것은 두 주요 분야에서 우리가 기대했던 것보다 훨씬 적은 것을 프리프로덕션에서 가져온다는 것을 의미했다.

첫째 이것 때문에 우리는 디자이너가 늦게까지 구조를 플레이할 수 없을 페이퍼 메트릭에 바탕을 두고 레이아웃을 만드는 상황에서 Tomb Raider: Legend 와의 시기와 비슷한 입장에 처하게 되었다. 이것이 좋은 디자인 관행을 어긴 것은 분명하지만 디자인 팀이 이제 막 이러한 동일한 구조를 많이 가진 Tomb Raider 게임을 이제 막 완료했기 때문에 우리는 이러한 구조 없이도 오랫동안 그럭저럭

헤나갈 수 있었으므로 손해를 억누를 수 있다고 생각했다. 하지만 상황은 최적인 것과는 거리가 멀었다.

둘째 *Tomb Raider: Underworld* 를 Xbox 360 으로 이식하는 것 외에 이것은 당시에 차세대 콘솔(console)이었던 것을 위해 만들어진 우리의 첫 번째 프로젝트였다. 우리가 예상한 것보다 훨씬 더 복잡하고 콘텐츠 집약적인 것으로 나타났으며 레이아웃을 만들 수 있는 능력이 프리프로덕션 단계에서 너무 손상되었기 때문에 제작의 아트 측면을 적절하게 시험하거나, 측정하거나, 조사할 수 없었다.

시간에 맞게 적절한 선적 기회를 갖기 위해서는 특정 날짜까지는 제작에 들어갈 필요가 있었기 때문에 우리는 우리의 아트 제작의 요구와 과정을 제대로 이해하기 전에, 그리고 이와 동등하게 중요한 것으로서 아트를 외주로 보내기 위해 나중에 필요할 파이프라인과 지지 구조를 개발하기 전에 프리프로덕션을 마쳤다.

또한 우리 팀에는 아직도 *Tomb Raider: Anniversary* 의 완료 작업을 하고 있는 팀원을 위해 맡아 놓은 자리가 있었기 때문에 특정 분야에서 인원이 부족했다. 결국에는 우리가 개념 단계, 프로토타이핑 단계 및 프리프로덕션 단계를 계획하기는 했지만 실제로는 개념 및 프로토타이핑 작업을 일부만 하고 나서 프리프로덕션을 완료하기 전에 바로 제작에 뛰어든 것이었다.

이 압력의 상당 부분은 야심과 능력 사이에서 일어난 충돌에서 발생했으며, 가지고 싶어하는 설계상의 게임을 보는 것과 정해진 날짜까지 이것을 가질 수 없다는 것을 제작 데이터가 나타낸다는 것을 알지만 어느 쪽도 변경하지 않으려고 하는 태도 사이의 인지적 불협화 때문에 악화되었다.

당시에, 그리고 프로젝트 전반에 걸쳐서 우리는 이전 단계가 완전히 완료되기 전에 제작의 다음 단계로 이동하는 것이 위험하다는 것을 알았지만 우리는 어떻게든 그렇게 했다. 이것은 바로 반복된 실수 중의 하나이다. 그렇다면 우리는 왜 그렇게 했을까? 인지적 불협화가 대답의 일부이다. 희망적 사고는 다른 부분이다. 하지만 불완전한 세계에서 대체로 당신이 무릅쓴 모든 위험은 실패할 수 있는 도박이며 노력이 끝날 때는 이것처럼 당신이 무릅쓴 위험 중에서 결과가 나오지 않은 것만을 설명할 필요를 가지는 것으로 끝난다.

4. 불가항력의 사고

이것은 일부 사후분석에서 "Too Many Demos"라는 제목을 붙이는 범주이거나 외부에서 팀으로 찾아오는 다른 어떤 문제이다. 우리에게는 분명히 데모(demo)가 너무 많다는 문제가 있었지만 이것은 준비되지 않은 우리에게 떨어진 문제 분류의 한 항목이었을 뿐이었다.

무엇보다도 우리가 처음부터 이 문제를 피하려고 했기 때문에 데모는 특히 사태를 악화시켰다. 우리는 장기 데모 일정을 요구했으며 이를 받아들였다. 우리는 일정에 있지 않은 데모에 대한 일부

요청을 받아들이기를 거부했으며 퍼블리셔(publisher)는 이를 존중했다. 하지만 다양한 환경 때문에 일부 일정에 없는 데모를 수용했던 미끄러운 경사면(slipper slope)도 있었다.

가끔 우리는 마케팅의 측면에서 특정 데모가 2 주의 제작 시간을 투입하더라도 제품의 성공을 위해서는 더 나은 장기적인 선택처럼 보였던 딜레마에 직면했다. 데모와 훌륭한 홍보의 중요성을 무시했기 때문에 위대한 게임이 소란스러운 시장 속으로 사라져버렸다는 이야기는 너무나 많다.

이러한 데모가 종종 제품의 질을 저하시키는 것은 사실이고 팀의 초점을 벗어나게 하는 것도 사실이지만 결국에는 느끼는 만큼 고통스러운 일이더라도 때때로 데모를 하는 것이 최상이다. 이것은 눈을 크게 뜨고 함정으로 걸어 들어가는, 즉 미리 알면서도 실수를 하는 것의 분명한 예이지만 데모를 거부하거나 이를 위한 더 나은 계획을 세우는 것이 해답이 아니기 때문에 반복적으로 발생한다. 끝에 이르러서만 함께 나오는 것이 아닌 게임을 만드는 것이 해답이다.

또한 팀 전체에 걸쳐서 결혼, 신혼여행, 제작 기간 중의 출산, 그리고 부적절한 시기에 프로젝트에서 떠나는 일 등이 너무 많았으며 그 중에서 무엇보다도 큰 영향을 준 것은 리더 중에서 이런 일이 생겼다는 것이다. 제작 중간에 아트 디렉터가 떠났는데 다른 회사로 간 것이 아니라 다른 업계로 떠난 것이었기 때문에 이에 관해 우리가 할 수 있는 일은 별로 없었다.

제작이 끝날 무렵에 수석 디자이너가 출산을 해서 일을 할 수 없었는데, 이것 또한 우리가 어쩔 수 있는 일이 아니었다(우리는 어떻게 하고 싶지도 않았는데 아이가 예뻤기 때문이다!). 그리고 가장 비극적이었던 것은 우리의 수석 레벨 디자이너가 제작 기간의 상반기에 갑작스럽게 사망한 것이다.

우리의 노력에서 이러한 핵심 인물들은 영리하고 능력이 있었을 뿐만 아니라 *Tomb Raider: Legend* 의 성공의 일부여서 이런 종류의 게임을 만드는 방법을 상세하게 알고 있었기 때문에 극히 중요했다.

여러 가지 이유로 *Tomb Raider* 게임은 만들기 어려웠기 때문에 이런 사람은 드물며 우리의 일정 아래에서는 이들을 대체할 수 없었다.

우리는 이 틈을 메우기 위해 팀에 남아 있는 사람들에게 새로운 책임을 맡게 해서 이러한 손실에 대한 상쇄를 했으며 이 팀원 중 일부는 일을 훌륭하게 처리하여 실제로 우리를 구했지만 이러한 손실이 없었다면 훨씬 더 부드럽게 제작을 했고 더 나은 최종 제품이 나왔으리라는 것은 부인할 수 없다.

5. 범위

우리가 너무 많을 것을 하려고 했다는 것은 사실이다.

가능한 멀리 뻗으려고 하고 자신의 팔이 실제 길이보다 더 길다고 생각하는 것은 인간의 본성이지만 어떤 게임은 다른 게임보다 규모에 더 민감하다. 수도쿠(sudoku) 게임을 만든다면 시간이 다 지나고 게임을 마칠 때까지 바둑판 무늬를 만들 수 있지만 서로 겹치는 기능 집합에 시작, 중간, 그리고 끝이 있는 게임의 경우에는 적절하게 크기를 조정한다는 것이 훨씬 더 어려우며 제작 중에 규모를 변경하는 것은 훨씬 더 어려운 일이다.

우리의 범위와 관련되고 우리가 베타에 이르는 과정에서 이루었던 축소와 관련된 성공담이 있지만, 최종 분석에서 우리는 너무나 많은 것을 하려고 했고, 그렇게 할 필요성에 빠졌으며, 우리가 지키려고 했던 품질 표준에 맞게 모든 것을 해내려고 노력했다.

규모가 우리의 마감시한을 계속해서 앞섰던 이유는 대체로 차세대 개발의 복잡성을 과소평가하였고 모든 문제를 파악하면서 프리프로덕션에 충분한 시간을 쏟지 않았기 때문이었다.

너무 큰 게임을 만들어서 생기는 가장 큰 피해는 크런치 타임(crunch time)이 아니었으며(과거의 게임과 비교하여 크런치 타임은 더 적었다), 경험에 허점이 없었기 때문에 게임의 일부를 완성하지 않고 놔둔 것도 아니었다. 가장 큰 손해는 우리에게 매우 중요한 요소, 즉 마무리에 발생했다. 우리는 마무리를 할 충분한 시간을 갖기 위해 전례가 없는 양으로 일정을 늘렸으며 그리고 나서 또 어느 정도 늘렸다.

하지만 차세대 개발의 예상치 못한 복잡성은 스튜디오보다 큰 팀에서 일어나는 의사소통과 함께 우리의 마무리 시간과 그 이상을 보고 이를 소비해버렸다. 우리는 전혀 새로운 종류의 알려지지 않은 것을 처리하고 있었기 때문에 이것 또한 반복된 "잘못된 점" 시나리오 중의 하나였는데도 불구하고 우리의 마무리 시간을 과소평가하는 실수를 더욱 이해 가능하게 만들었다. 하지만 이것은 앞으로도 진행되는 경우는 아닐 것이며 따라서 팀은 다음 번에는 충분한 마무리 시간을 수립하지 않는 흔히 저지르는 실수를 피하기 위한 준비를 해야 한다.

위험과 보상

우리는 Tomb Raider: Legend 의 성공에 힘을 입어 강력한 후속 작품을 만들려고 한 것이었지만 그 자체만으로 주요한 제품이 되었다. 이 게임을 만든 많은 팀원의 노고, 재능, 유연성, 헌신, 그리고 레벨 수석 때문에 처음에 모든 사람들이 합리적으로 예상했던 것보다 더 크고 풍부한 작품이 나온 것이다. 일부 리뷰에서 *Tomb Raider: Underworld* 를 첫 번째 게임 이후의 최상의 *Tomb Raider* 게임으로 환영한 것은 매우 만족스러운 것이었으며 팬의 반응은 경이적이었다.

그렇다면 우리가 막으려고 했던 그렇게 많은 것들이 여하튼 잘못된 것은 왜일까? 여기서 설명한 범주는 훨씬 더 많은 범주 중에서 겨우 다섯 개일 뿐이다. (잘못된 빌드와 긴 빌드 시간은 충분히 언급할 만하여 추가된 차별화는 아마도 다른 모든 문제를 합한 것보다 혈압을 더 상승시켰을 것이다.) 함정에 대한 사전 지식을 갖추었다면 이러한 함정을 완전히 피할 수 있게 되지 않을까?

그렇지 않다. 창의적인 노력은 그 자체의 성격 때문에 혼란스러운 것이며 창의적인 혼란, 야심, 그리고 헌신이 제작 과정에서 발생하는 우려 사항과 퍼블리셔의 마감 시한 및 목적이라는 단단한 벽에 충돌하면 많은 동일한 함정에 빠지게 되며 심지어는 이러한 함정에 향하게 된다. 모든 알려진 함정을 피하기 위해 가능한 모든 것을 하는 것은 혁신과 창의력을 상실하게 만들 것이기 때문에 이렇게 하는 것은 해답이 아니다. 잠재적인 함정을 인식하고 그 중 하나에 빠지는 것의 부정적인 영향을 민첩하게 완화시킬 준비를 하는 것이 방법이다.