



※ 본 기사는 CMP MEDIA LLC와의 라이선스 계약에 의해 국문으로 제공됩니다

포스트모템: Double Fine의 *Brütal Legend*

캐롤라인 에스몰도크(Caroline Esmurdoc)
가마수트라 등록일(2010. 03. 25)

“크게 가지 않으면 집으로 가라”라는 격언을 소프트웨어 개발 노력에 적용할 수 있다면, 아마 *Brütal Legend* 제작에 적용할 수 있을 것이다. 예전에 우리가 *Psychonauts* 를 작업했을 때, Double Fine 은 다시 한 번 혁신에 사활을 걸었다. 이때 Tim Schafer 의 마음 속의 Full Throttle 측에서 게임 하나가 태어났다. *Brütal Legend* 는 헤비메탈 정신으로 녹아드는 3 인칭 오픈월드 전략형 액션 어드벤처 인터랙티브 라이드 게임이다.

개발 스토리는 아주 간단하게 시작되었다. *Psychonauts* 를 출하한 후, Double Fine 은 *Brütal Legend* 의 정신을 포착하기 위해 고안된 컨셉워크 콜렉션, 피치 문서 및 게임 트레일러를 만들었다. 우리가 말했던 대부분의 퍼블리셔들은 게임 컨셉에 관심을 보였으나, 그들의 질문은 공통적으로 우리가 그 게임을 통해 나아가는 방향이 어디인지를 잘 모르겠다는 것이었다.

예를 들면 이런 질문이었다 “왜 헤비메탈이죠? 락이나 컨트리 아니면 힙합은 어떤가요? 왜 밴드 공연 매니저로 연주하고 싶으신거죠? 락의 신으로 연주하는 건 어떤가요?”

Vivendi Universal Games 의 한 퍼블리셔는 피치 미팅이나 다른 미팅에서 이러한 질문을 하지 않았다. 그들은 이 게임이 무엇을 위한 것인지 이해했고, 이 게임이 될 수 있을 것이라고 하면서 계약했다.

우리는 게임의 멀티플레이어 모드에 우선 초점을 맞추면서 개발을 시작했다. 우리의 생각은 전에 한 번도 멀티플레이어 게임을 만들어본 적이 없기 때문에 이 문제를 해결하는 데 최고 주안점을 둔다는 것이었다. 그리고 16 개월의 시간이 걸렸다.

Rocktober 2006 에서, 우리는 퍼블리셔에게 전체 플레이가 가능한 Ironheade vs. Tainted Coil 스커미쉬를 납품했다. Vivendi 의 요청에 따라, 우리는 싱글 플레이어 캠페인에 초점을 맞추었는데, 그 범위가 초기 디자인을 훌쩍 뛰어넘었다.

이 기간은 우리가 Jack Black 의 목소리와 다른 많은 유명 탤런트들을 게임에 추가했던 시기이며, 또한 우리가 마지막에 출하했던 캠페인 경험에 대한 비전을 끈고히 해 주는 다른 개선점들을 추가한 시기이다.

2007 년 6 월 우리는 싱글 플레이어 게임의 의미있는 첫 번째 부분을 납품했고 게임 컨셉에 대한 모든 변경사항이 예정보다 늦어진다는 것 또한 재빠르게 수용했다.

우리의 첫 번째 일정 수정은 프로젝트를 10 개월까지 연장시켰고, 두 번째 일정 수정은 7 개월 더 연장시켰다. Vivendi Universal Games/Sierra 배너 하에서 2008 년 5 월 첫 출시하기로 예정되었던 *Brütal Legend* 는 Electronic Arts 에 의해 퍼블리시되어 2009 년 10 월 13 일 최종 출하되었다.

Double Fine 은 혁신을 장려하지만 이는 곧 우리가 항상 기능이나 접근법이 어떻게 될지를 예측하기 위해 이전 경험에 의존할 수는 없다는 뜻이기도 하다. *Brütal Legend* 에서, 연속적인 반복 및 컨셉 정제의 관행은 수많은 아이디어의 프로토타이핑을 유도했으며, 이 중 많은 것들이 살아남아 최종 게임에 탑재되었다. 그러나 이에 못지 않게 Perforce 의 연대기에서 디지털 화석이 된 채로 남겨진 것들도 많이 있었다.

Brütal Legend 개발 중 제대로 진행된 것과 잘못 진행된 것들의 목록을 10 개로 간추린 결과 상당한 문제점이 드러났다. 가장 놀라웠거나 충격적이었던 몇 가지 교훈을 아래에 정리했다.



잘된 점

1. 창조적 한계 뛰어짜기

Brütal Legend 는 헤비메탈 세계의 과장된 우스꽝스러움(아직 굉장히 진지함)이 인터랙티브하게 융합된 것이다. 우리는 메탈 광 팬들이었고 이 세계를 생활 속으로 가져오는 것을 영광으로 생각했다. *Brütal Legend*는 다음과 같이 간단한 목록에서 시작했다. 헤비메탈 앨범 표지에서 찾아볼 수 있는 모든 것이 구현된 게임: 크롬강, 피 웅덩이, 화산, 동굴, 불을 내뿜는 메탈 비스트, 레이저 표범, 블레이드헨지 및 비어헨지, 도미나트릭스, 라텍스와 체인, 몸에서 떨어져 나온 죽지 않은 머리...

여기에서 우리는 핵심적인 게임플레이를 추가했다. 브라우저 어드벤처 게임을 만들고 싶었다. 이 게임에서 플레이어는 게임 과정에 걸쳐 략의 신으로 진화하는 헤비메탈 공연 매니저가 된다. Xbox 360 과 PlayStation 3 상에서 *Brütal Legend* 는 액션 게임의 근력과 RTS 의 우아함을 갖출 수 있다.

Vivendi 와의 관계에서 우리는 일찍이 RTS 가 콘솔 공간에서 부적합한 단어라는 점을 알게 되었기 때문에 그 명칭을 더 이상 부르지 않았고, Electronic Arts 또한 그랬다. 게임을 시장에서 액션 타이틀로 광범위하게 포지셔닝하기 위해서였다.

우리는 우리의 RTS 가 콘솔의 이점을 활용하여 플레이어가 액션의 중심에 있게 하고 싶었다. 플레이어가 여러 가지 난폭한 동작과 능력을 수행할 수 있는 캐릭터를 직관적으로 통제하게 하고 또한 플레이어가 해당 캐릭터와 개인적으로 유대관계를 가질 수 있게 하고 싶었다. 그리고 캐릭터가 오만하게 대화하는 수십 분대들을 동시에 지휘할 수 있게 하고 싶었다.

RTS 접근성 문제를 해결하는 데 있어 우리의 가장 큰 문제점 중 하나는 팀 명령이었다. 수많은 시도와 끝도 없는 포커스 테스트가 있었지만, 결국 우리는 명령 인터페이스를 단순하게 통일하기로 결정했는데, 여기서 AI 는 전쟁터에 있게 하고 싶은 플레이어로 행동한다.

플레이어는 다음과 같은 4 가지 명령을 가질 수 있다.

1) "따라와(Follow)", 이 명령에서 Eddie 는 자신의 분대에게 특정 위치로 이동하여 가능하면 공격하라는 명령을 한다. 그러면 그 명령은 전체 맵을 둘러싼 Eddie 주변에서 상대적으로 큰 거리인 "샤우팅 레인지" 안에서만 주어질 수 있다. 이때 Eddie 가 그의 분대 근처에 있을 때는 "따라와" 명령이 강제되지 않게, Eddie 가 멀리 떨어져 있을 때는 명령이 강제되도록 함으로써 강제(경로의 적/장애물 무시) 및 비강제(길을 따라 적들과 교전) 문제를 해결한다.

2) “막아라”, 이 명령에서 Eddie 는 자신의 군대에게 이동을 멈추고 그 자리에 있으라는 명령을 하여, 카메라를 마주한 상태에서 가장 유리한 방식(정면에서 난투, 뒤쪽에 배치, 후미에서 지원)으로 군대를 정렬할 수 있다.

3) “이동하라”, 이 명령에서 군대가 목적지에 가까워질 때까지 이동을 하게 되는데 여기에서 근처의 적들과 교전을 할 수도 있다.

4) “돌격하라”, 이 명령에서는 플레이어의 군대가 공격하지 않은 경우 근처의 모든 적군들의 평균 위치에 가장 가까이 있는 적으로 비강제 이동하고, 플레이어의 군대가 공격을 한 경우 현재 활동을 철수했다 하더라도 공격 위치로 강제 이동한다.

Double Fine 이 콘솔 RTS 를 구체화하는 바람에 우리는 초기 사전제작 회의에 참여할 수 없었지만 고된 반복과 재창조 및 재작업은 몇 년 동안 계속되었다. 우리는 주기적인 “Mandatory Hour of Fun” 세션에서 진행 상황을 테스트했는데, 여기서 전체 팀이 최신 빌드를 플레이한 다음 한 그룹으로서 회의를 하여 잘 진행되었던 점 그리고 실망스러웠던 점이나 개선될 수 있는 점에 대해 토론했다. 아이디어 교환 및 지속적인 반복에 대한 추진력을 위한 이러한 공개 포럼은 개발 과정에 걸쳐 핵심적인 게임 메커니즘에 심오한 변화를 가져왔다.

2. 스크럼

Brütal Legend 에 대한 작업을 시작하기 전에, Double Fine 팀은 *Psychonauts* 를 개발하는데 5 년의 시간을 보냈다. 이 중 마지막 2 년은 엄청나게 고된 크런치의 시간이었었는데 여기서 회사는 초기 퍼블리셔를 잃었고 게임 출시 직전에 회사 문을 닫을 뻔했었다.

우리의 첫 번째 타이틀을 출하했을 때의 희열이 사라졌을 때, 우리의 눈에는 Double Fine 이 다른 스튜디오가 했던 방법으로 게임을 개발하지 않은 것이 확실했고 다양한 제품 개발 시스템을 적소에 배치해야 할 것으로 보였다.

Psychonauts 에서 공포스러운 크런치가 있었던 주요 원인은 레벨이 빌드된 후에도 우리가 게임 기능을 계속해서 개발했었기 때문이다. 각 게임 메커니즘이 개선되면 모든 레벨에 대한 재작업이 있었다. 거품을 칠하고, 행구고, 반복하는 꼴이었다.

Double Fine, 특히 Tim 은 게임을 플레이하고 게임으로 살고 게임으로 숨쉬고 시간이 지날수록 게임에 더 몰입하고, 게임을 재미있고 또 재미있게 만드는 것을 끊임없이 반복해야 했다.

방법론에 대한 연구 후, 우리는 애자일 소프트웨어 개발의 장점에 이끌려서 스크럼을 채택하기로 결정했다. *Brütal Legend* 를 개발하기 시작한 처음 몇 달, 팀은 스크럼을 연습했고 최초의 결말은 정말 인상적이었다.

시스템에 대한 기능, 신속한 프로토타이핑 및 반복, 교차 학문적인 팀, 프로세스에 참여하는 사람들 그리고 잠재적으로 출하가능한 여러 소프트웨어의 개발에 스크럼을 집중함으로써 게임이 개발 초기 단계, 그것도 아주 초기 단계에서 플레이가 가능하게 되었다. 첫 번째 달에는 렌더러, 지형 및 플레이가능한 캐릭터(Eddie Riggs)를 갖추었고, 두 번째 달에는 2 명의 Eddie 가 지형 근처에서 핫 로드(Druid Plow)를 운전할 수 있었고, 세 번째 달에는 Eddie 가 하이필드 지형 근처에서 Druid Plow 로 끌도 없이 수많은 정신이상자들을 치고 갈 수 있었다. 환희가 느껴졌다.

우리는 메타 게임 창작뿐만 아니라 마이크로 프로젝트에도 스크럼을 적용했다. 개발 과정 거의 초기에, 우리는 어떻게 RTS 를 만들지에 대한 개념도 없었고 RTS 를 만드는 데 적합한 엔진도 보유하고 있지 않았었다. 우리는 재고 PC 엔진으로 프로토타입을 만들어서 이 두 문제를 해결했으며 우리 팀의 여러 멤버들은 이에 곧 익숙해졌다. Unreal 2.5 엔진에, *Brütal Legend* 의 디자인 수요는 기존의 FPS 엔진을 사용해 게임을 개발하려는 시도가 어려운 것으로 입증되었지만, UnrealScript 의 유연성에 처음으로 접근함에 따라 우리는 개발 PC 에 대한 초기 RTS 개념 중 일부를 테스트할 수 있었다.

이러한 접근법을 통해 우리 디자이너들과 게임플레이 프로그래머들은 엄청난 생산성을 거둘 수 있었으며 동시에 프로그래밍 팀은 우리의 새 엔진 구축을 위한 작업에 들어갔다. 이를 통해 개발 도중 우리가 직면할 수 있는 디자인 문제점 및 UnrealScript 를 통해 무엇인가를 신속하게 반복할 수 있는 기회를 엿볼 수 있었으며, 이것은 우리의 새 엔진을 구축하는 방법에 대한 매우 값진 방향과 *Brütal Legend* 를 정의하기 위한 메커니즘에 대한 통찰력을 제공해 주었다.

우리의 성공에도 불구하고 스크럼에 대한 눈에 띄는 부정적인 면이 있었다. 스크럼을 구현하면서 사전제작이 너무 오래 있다가 제작으로 이어지게 되었다. 스크럼은 방어적인 프로그래밍을 장려하지도 않았고 잘 확장된 디자인 시스템의 프랙티스를 조장하지도 않았다. 우리는 많은 시스템들을 아이디어를 입증하고 확장하여 향후 납품본에서 정제하는 데 충분하도록 80%까지 가져왔다. 그러나 개발 단계의 마지막 몇 달쯤에 나머지 20%가 노력의 다른 80%였다는 것을 알았다. 이것은 가장 심한 타격을 입은 팀 멤버들에게 뜻밖의 크런치를 유도했다. 프로젝트가 제작 단계에 들어가게 되면, 신뢰성과 아트 크리에이션이 낮아질 위험이 있기 때문에, 폭포수형 접근방식이 스크럼보다 더 적합했다.

이러한 경고에도 불구하고, 스크럼은 팀으로 하여금 다듬어지지 않은 이론을 신속하게 테스트할 수 있게 하여 최고의 이론을 유지하게 할뿐만 아니라 개발 전 과정에 걸쳐 지속적인 반복을 통해 그 이론들이 빛이 날 수 있도록 했다. 이것은 스튜디오에게 있어 의미있는 과정 개선이었다.



3. Bert 의 붓

Double Fine 은 2 명의 정규직 테스터를 고용한다. 우리는 이 테스터들이 게임의 일일 빌드를 스모크 테스트하고 우리가 다음 납품본에 넣기로 한 새 기능들을 구현하는 데 의존했다. 초기부터 분명했던 사실은 테스터들만으로는 게임의 안정성 요구를 계속 유지할 수 없으며, 프로젝트 기간 동안 정규직 테스트 직원을 고용하는 것 또한 재정적인 측면에서 가능하지 않았다는 것이다. 우리의 솔루션은 자동화된 테스트 시스템을 개발하는 것이었는데, 우리는 이 시스템의 이름을 사랑스럽게 RoBert 라고 지었고, 뒷 글자만 따서 Bert 라고 했다. Bert 가 우리의 소프트웨어 테스트 엔지니어가 된 것이다.

자동화된 테스트는 일종의 실험으로 시작되었고 초기 시스템을 투입하는 데는 약 한 달 정도가 걸렸다. 첫 번째 간단한 스크립트는 게임의 모든 캐릭터와 오브젝트를 불러냈는데 모두 경고와 충돌을 찾아내는 데 유용한 것으로 입증되었다.

정기적으로 테스트를 실행한 결과 이러한 충돌을 시기적절하게 발견할 수 있었으며, 코드베이스에 대해 수행된 다소 최근의 변경에 대한 충돌 원인의 범위를 좁혀갈 수 있었다. 프로그래머들은 자동화된 테스트를 로컬로 실행하기 시작했으며 이에 따라 프로그래머들은 체크인하기 전 먼저 위험한 코드 변경을 테스트해 볼 수 있게 되었다. 테스트는 또한 버그를 재현해 내는 데도 도움이 되었다.

특히 꾀가 많은 프로그래머 한 명은 잠 들어 있는 Xbox 360 과 PlayStation 3 를 빌려와서 자동화된 테스트를 실행하자는 아이디어를 제안하기도 했다. 팀 멤버들은 항상 테스트를

종료하고 나서 자신의 머신을 다시 가져갈 수 있었지만, 일주일 내내 테스트를 실행하려면 잠들어 있는 머신을 사용하는 것이 유용하고 효율적이었다.

자동화된 테스트 시스템이 성공적이었기 때문에 우리는 이것을 확장하여 최후까지 싸우는 두 군대를 이용한 테스트를 포함시켰고 각 유닛이 얼마나 파워풀한가를 판단하기 위해 테스트의 균형을 맞추었다. 그 다음 입력이 시뮬레이션될 수 있는 경우 봇(Bot) 기능이 추가되었다.

이를 통해 플레이어가 실제 게임을 할 때 이동을 정확하게 수행할 수 있게 하기 위한 테스트를 수행할 수 있었다. 봇은 공격 콤보, 그 외 모든 분대의 다양한 이동을 수행하는데 사용되었고 또한 월드에서 stuck-spot 을 찾는 데도 사용되었다. 또한 멀티플레이어 테스트에도 사용되었는데, 이것은 우리의 P2P 록스텝 네트워크 시스템에서 싱크가 맞지 않는 것(de-sync)을 찾는 데 굉장히 중요했다.

가용한 머신이 있는 한, 1v1 및 팀 멀티플레이어 테스트는 정기적으로 실행될 수 있었다. 결국, 봇 시스템은 캠페인을 통해 플레이될 수 있도록 확장되었다. 두 번째 미션을 통한 플레이, Motorforge 실행, 장비 업그레이드 또는 완성 전 각 미션 실패를 위해 테스트 설정이 추가되었고 메모리 리포팅도 곧 캠페인 테스트에 추가되어 프로그래머들이 메모리 사용량과 누출량을 추적할 수 있게 되었다.

자동화된 테스트를 원격으로 실행하고 크래시 리포트를 관련자 및 책임자들에게 이메일로 전송하는 것이 가능해졌다. 이렇게 함으로써 여러 개의 테스트를 동시에 실행하는 것이 크게 단순화되었고 상이하게 구성가능한 빌드를 사용하여 테스트가 실행되는 경우도 생기게 되었다. 초기에는, 대부분의 테스트들이 디버그 빌드를 사용해 실행되었기 때문에 프로그래머들은 크래시된 머신에 연결하여 보다 쉽게 문제를 해결할 수 있었다.

프로젝트 말미에는 테스트가 특수 릴리즈 빌드를 실행하여 릴리즈전용 크래시를 찾아냈다. 시스템에는 약간의 제한이 있었다. 예를 들면 프로그래머가 남아있는 크래시를 해결하기 전에 봇 팜에서 크래시된 머신이 기계 소유자에 의해 재부팅되는 것. 그리고 봇은 결코 경로나 운전을 학습하지 않으며 일시 중지 또는 프론트 엔드 메뉴에서 테스트를 실행할 수 없었다. 우리는 이러한 점들을 개선할 것이며 향후 프로젝트를 위해 RoBert 로 추가 확장을 수행할 것이다.

4. 툴 슈퍼파워

Double Fine 은 함께 추가될 때 능률화된 개발 파이프라인을 만드는 다양한 툴을 작성했다. 하나의 툴만 선택해서 그 툴이 어떻게 효율적인 개발을 가능하게 했는지 나타내는 것은 어렵다. 여기에 2-3 개 정도의 무기가 설명되어 있다.

MUE(Multi-User-Editor)는 우리의 월드 제작 과정의 백본을 형성하는 협업형 오픈월드 빌딩 툴이다. 이 툴의 기본 기능은 디자이너들과 아티스트들이 우리의 대규모 월드에서 동시에 작업할 수 있게 하는 것이다.

Brütal Legend 월드는 상이한 “타일”로 구분되는데, MUE 를 이용하면 개별 타일을 체크아웃하고 다양한 개인이 타일을 편집할 수 있다. 또한, 월드에 거주하는 수천 개의 엔티티 중 일부 또는 전부를 체크아웃하고 다양한 개인이 편집할 수 있다.

MUE 는 또한 하이 맵 지형을 편집하는 데 적합한 툴 세트를 제공한다. 이것은 ZBrush 라이트와 약간 비슷하지만, 하이 맵 지형을 편집하는 데 특히 최적화되어 있다.

조각 및 스무딩 툴 외에도, MUE 에는 아티스트들이 메시 조각을 가져와서 지형에 이 조각을 정렬시킨 다음 버튼 클릭 한 번으로 “스탬프”할 수 있게 하여 하이 맵 지형의 형태를 만드는 “지형 스탬핑”으로 알려진 혁신적인 조각 툴이 있다.

이것은 지형에 대해 특정한 높이, 각도 또한 너비 요구사항을 갖는 중요한 게임플레이 영역을 만들 때 특히 유용한 기법이었다. 지형 조각 외에도, MUE 는 아티스트들이 혼합된 지형 재료와 주변 메시(풀, 덩굴 등)를 실시간으로 채택할 수 있게 한다.

MUE 는 Maya 창 내부에서 실행하지만 후드 아래에는 파워풀한 SQL 데이터베이스가 있다. 데이터베이스 백엔드를 보유한다는 것이 흥미롭게 들리지 않을 수는 있지만, 이것은 데이터 손실에 대한 강력한 보호수단의 역할을 하여 신속한 월드와이드 변경을 가능하게 했다. 아티스트가 모든 포스터 트리를 타이어 트리라고 바꾸고 싶어하는 경우, 한 줄 검색 및 교체 파이썬 스크립트를 실행할 수 있는데, 이는 또한 데이터 체크인 및 체크아웃을 처리했다.

두 번째 툴(Rigator 라고 함)은 리깅 및 애니메이션 툴 파이프라인을 지원하기 위해 만들어졌다. 약 150 개의 독특한 캐릭터(이 중 일부는 40 개 남짓의 얼굴 모양) 및 약 300 개의 영화 신을 만들고 애니메이션하는 위압적인 작업에 마주하게 됐을 때, 캐릭터 리깅 및 애니메이션 파이프라인을 가능한 한 많이 자동화하는 것이 중요했다. 많은 분량의 아트 수정이 내부적으로 이루어졌기 때문에, 우리는 또한 쉽게 변경을 할 수 있어야 했다.

이를 위해서 우리는 툴에 의해 자동으로 할당된 Maya 레퍼런싱 및 명명 규칙을 사용하여 사용자 입력을 최소화하기 시작했다. 그 결과 사용자와 파일 간의 불일치가 최소화되었다.

캐릭터 리그를 만들고 편집하기 위해, 우리는 자동화된 과정을 만들었고 그 과정에서 사용자가 간단한 골격 윤곽을 캐릭터의 비율에 맞게 맞추고 버튼 하나만 누르면 캐릭터가 가졌던 고유한 크기 또는 형태를 기준으로 적절한 위치와 명명을 통해 복잡한 애니메이션 컨트롤을 즉시 만들 수 있게 하였다. 이 툴을 개발하기 전, 과정은 수동으로 완료하는데 1 내지 2 일이 걸렸지만 애니메이터들은 이제 캐릭터를 즉시 리깅할 수 있게 되었다.

또한, 고유한 캐릭터에는 많은 고유한 애니메이션이 필요하기 때문에, 우리는 캐릭터의 고유한 자세와 애니메이션을 쉽게 저장하고 게임의 다른 캐릭터로 로드할 수 있게 하는

라이브러리를 관리하는 툴을 개발했다. 이 Anim Toolbox 는 애니메이션 과정의 속도를 높이는 동시에 만든 사람에게 상관없이 모든 캐릭터 애니메이션 세트가 독창적으로 결합되도록 하며, 인게임 애니메이션뿐만 아니라 게임의 영화 신에도 적용되었다. 이러한 워크플로우를 활성화하지 않았다면 우리는 엄청난 양의 애니메이션을 필요한 시간 안에 만들지 못했을 것이다.



5. 빅 네임

라이센스를 받은 음악 트랙의 수나 서명한 유명 배우/뮤지션의 수를 떠벌리는 것은 *Brütal Legend*에서는 가능하지만, 다른 게임의 경우 거의 없었다. 게임에 포함시킬 각 아티스트와 트랙을 선택하는 것은 엄청나게 끔찍하고 힘든 과정이었지만, 그 만큼의 가치가 있었다.

라이센스를 받은 음악의 경우, Tim 씨와 음악 감독 Emily Ridgway 씨가 본질과 진실성이 담길 노래를 선정했다. 각 노래는 게임에 알맞게 사용되도록 조정되어야 했고 메탈 팬들이 수용할 수 있는 것이어야 했다. *Brütal Legend*가 이 장르를 진정으로 잘 반영할 수 있게 하기 위해, 헤비메탈의 현장감이 느껴질 수 있도록 해야 했다. 음악, 가사 및 영상이 게임의 크리에이티브 디렉션 역할을 했기 때문이다.

일단 노래가 선정되고 나면, 100 개 이상의 트랙에 대한 라이선싱 과정(각각 마스터 및 싱크 계약서)을 거쳐 다중 권리 보유자들에게서 대부분, 그리고 해체한지 오래된 밴드로부터 일부를 찾아서 협상하고 예산을 짜고 서명하고 감사하는 데 초인적인 노력을 기울였다.

우리는 각 노래에 대해 개별적으로 허가를 얻었고, 경우에 따라 Double Fine 에서 *Brütal Legend*를 대신해 개인적으로 밴드에 연락을 취해 아티스트가 게임용 원곡을 창작하게

했다. 예를 들면 리타 포드의 Betrayal 같은 곡이다. *Brütal Legend* 의 최종 사운드트랙은 웅장했다. 잘 알려지지 않는 라이선싱 과정에서 트랙을 확인하고 선택할 때는 흥분되던 기분이 곡 사용 허가를 얻고 나면 뭔가 뿌듯함으로 바뀌었다.

음악 콘텐츠에 대한 우리의 꿈은 시작부터 고상했기 때문에, 처음에는 Eddie Riggs 의 목소리 역으로 Jack Black 을 캐스팅할 계획이 아니었다. 그가 *Psychonauts* 의 팬이자 메탈 팬이라는 말을 듣고 우리는 기대를 갖고 그에게 접근했고, 그는 게임에서 목소리 연기를 해주기로 계약을 하게 된 것이다.

Jack Black 외에도, 게임에서 캐릭터를 구체화하고 캐릭터의 헤비메탈을 대한 기여를 기릴 수 있는 목소리를 제공하기 위해 대단한 경력을 가진 다른 수많은 뮤지션들을 확인하고 찾았는데 모두에게 연락하고 협상하고 일정을 잡고 방향을 제시해야 했다. Double Fine 에서는 캐릭터에 대해 성격과 유머를 항상 정의했는데, *Brütal Legend* 는 캐스팅과 녹음이 잘 수행된 예라고 볼 수 있다. 아 이번은 예외이다. 캐스팅이 좀 더 잘되고 유명했기 때문에.

놀랍게도, 뮤직 사운드트랙 및 캐릭터 목소리를 유명 배우로 캐스팅하는 추세의 증가, 특히 Jack Black 목소리를 Eddie 역할로 한 것은 Tim 의 작품에 영감을 주었고 팀에게 동기 부여가 되었으며 퍼블리셔들의 흥미를 일깨웠다.

잘못된 점

1. 미들웨어

Brütal Legend 개발 주기가 시작되었을 때 다중 콘솔에 대한 새로운 차세대 기술 구축의 전망은 우리가 생각해 보기에 위압적인 시도였으며, 스크럼의 프랙티스나 자체적인 야망도 게임에 필요한 모든 시스템을 구성할 수 없게 했다.

우리는 게임 개발 초기에 미들웨어를 사용하는 데 따른 장점을 인지하게 되었다. 통합에 적절한 미들웨어를 선택함으로써 게임을 조기에 시작할 수 있게 됐으며 새 아이디어를 신속하게 테스트할 수 있게 되었다.

이러한 장점으로 인해 우리는 개발 주기 시작 단계에서 강력한 데모를 만들 수 있었는데 이것은 곧 게임을 피칭할 때 잠재적인 퍼블리셔들 그리고 납품본에 대한 퍼블리셔의 기대가 다소 누그러지는 사전 제작 중 현재의 강력한 마일스톤에게 우리의 게임 컨셉을 전달하는데 도움이 되었다.

Double Fine 은 오디오(FMOD), 피직스(Havok), UI(Scaleform), 립싱크(Annosoft - 툴 및 오프라인) 그리고 비디오(Bink)에 미들웨어를 사용했다.

다른 미들웨어 패키지로 잘못 시작한 적도 있었지만, 상용 소프트웨어 패키지 전체에 걸쳐 개발 위험을 분배한다는 결정이 합리적이었고 개발 초기에 이러한 패키지를 통합하라는 요구가 개발 수명 주기 전체에 걸쳐 성과를 거뒀다.

다양한 시스템에 대한 상용 소프트웨어에 의존한 것이 유익했던 만큼, Double Fine 이 적절한 미들웨어를 선택하는 확률은 .500 였다. 우리는 결국 수많은 미들웨어 소프트웨어 패키지를 선택하여 통합했지만 나중에는 다 폐기해버렸다. 프로젝트가 진화함에 따라 이러한 패키지가 우리의 요구를 충족시켜주지 못했기 때문이다. 우리가 마지막에 출하한 미들웨어는 통합 시에 여전히 엄청난 투자를 필요로 했고, 결국 부각되는 안정성 및 성능 문제를 해결해야 했을 때보다 훨씬 더 많은 직원이 필요했다.

여러 미들웨어 업체들은 그들 시스템에 새로운 요구가 생김으로 인해 우리가 이전에는 발견되지 않았던 소프트웨어의 버그와 결함을 노출시켰다고 알려왔다. 이 소식은 우리를 기쁘게도 했고 낙담시키기도 했고, 결국 스트레스가 많은 개발 주기 동안 관리 비용이 많이 들게 되었다. 개별 미들웨어 업체에서 우리 했던 기술 지원을 받는 것이 엄청난 위안이 되었다. 그들의 시기적절한 관심과 우리 타이틀에 대한 협업 의지가 없었다면, 우리는 아마도 게임을 하나도 출하하지 못했을 것이다.

우리가 이미 향후 프로젝트에 대해서도 계속해서 특정 미들웨어 패키지를 사용한다는 계획을 인지하면서, 우리는 개발 주기 후반부에 가서 다른 패키지를 사용하는 데 노력을 기울일 것이며 우리의 요구를 특히 잘 충족시키는 자체 소프트웨어를 구성할 것이다. 부적절하거나 쓸모없는 미들웨어를 코드베이스에 통합한 다음 다른 것을 재통합하거나 자체 솔루션을 생산하는 데 아까운 시간이 낭비되었고, 굉장한 엔지니어링 시간과 노력이 들어갔으며, 미들웨어 지원 및 디버깅을 했던 것처럼 우리는 결국 *Brütal Legend* 의 출시를 통해 사용을 결정했다.



2. 11 번째 시간 툴

개발 시작 시부터, 우리는 팀에 보다 빠른 반복 시간을 보장하기 위해 우수한 툴에 우선순위를 두어야 함을 인지했다. 레거시 팀 구조, 고유한 복잡성, 콘텐츠 창작 리더십에서의 불충분한 투입, 빈약한 기획 및 불운이 모두 혼합되어 팀에 중요한 몇 가지 툴의 납품에 상당한 지연이 발생하게 되었다. 이 지연은 마일스톤 납품을 위태롭게 하고 아티스트 생산성을 저하시켜, 결국 다른 엔지니어링 노력을 희생하면서까지 예기치 못한 긴급 중재를 할 수 밖에 없게 했다.

공교롭게도, 우리는 밝은 컬러의 Xbox *Psychonauts* 그래픽에서부터 Xbox 360 과 PlayStation 3 *Brütal Legend* 그래픽의 선명하고 고도로 정교한 풍부함을 얻기 위해 어떤 것을 취해야 하는지에 대해 크게 과소평가했다.

먼저, *Psychonauts* 에서 경험했던 전용 툴 프로그래머의 부족을 해결하기 위한 시도로, 우리는 *Brütal Legend* 의 엔지니어링 팀 내에 툴 그룹을 만들고 4 명의 엔지니어를 두었다. 이것은 *Psychonauts* 를 개선시킨 동시에, 이 접근법의 문제점은 엔지니어링 노력의 나머지 부분에서 툴 작업의 분리를 쉽게 하여 게임의 기능을 전달하는 과정에서 사용자에게 대한 툴의 접근성을 격리시키는 것이었다.

종종, 이러한 분리는 구현 시 비효율성, 툴 디자인과 이것이 지원하는 런타임 기능 간의 불일치를 야기했다. 어떤 경우, 이러한 분리는 일체의 툴 지원 없이 런타임 팀이 아티스트 페이싱 기능을 완성할 수 있는 경우 스프린트로 이어졌고, 그 기능은 쓸모없게 되어버렸다.

두 번째 중대한 실수는 효과적인 툴 우선순위 설정과 연관된 것이었다. 콘텐츠 창작으로부터의 강력한 지침 없이, 툴 팀은 초기에 툴 개발 및 빌드 인프라와 같은 백엔드 서비스에 초점을 두었다. 그 취지는 이러한 서비스가 향후 툴 개발에 대한 생산성 승수의 역할을 할 수 있다는 것이었다. 불행하게도, 분배 툴이나 빌드 코드가 없는 상태에서는 초기 투자가 콘텐츠/게임 패이싱 툴의 개발을 크게 지연시켰다.

이후에, 우리는 전체적인 제작 환경으로 확대될 수 있는 단기 요구 충족과 툴 빌딩 간의 올바른 균형을 찾기 위해 고군분투했다. 이러한 비효과적인 우선순위 설정이 미치는 영향의 한 가지 예가 바로 MUE의 1년 개발 지연이었다. MUE는 어떤 상황에서건 거대한 일일 수 있지만, 우리의 경우 단기 임시변동에서부터 툴의 제작 준비 실제 버전에 이르기까지 늦은 시작과 험한 길로 인해 훨씬 더 어려웠다.

그리고 약간의 불운도 있었다. 우리는 COLLADA 기반 Maya 익스포터 파이프라인의 얼리 아답터가 되기로 결정했다. 처음에는 COLLADA 1.3-준수 파서를 구현하는 데 얼마의 시간이 걸릴지에 대해 과소평가했었다. 그 다음, COLLADA 1.4에서 우리 계획에 큰 변화가 필요하게 됐을 때, 파서를 다시 구성해야 했다.

최종적으로 기능을 다 갖추고 나면, 우리는 유지보수 진행이나 예기치 못한 한계에 부딪혔고 결국 용이한 상호운용성 또는 툴이나 파이프라인 유틸리티의 평균 증대와 같은 예상된 상승세를 인지하는 데 실패했다. 어쨌든, 결과적으로 “COLLADA”이라는 단어는 불경스러운 말이 되었고 Double Fine 스크럼 미팅에서 불만 섞인 소리들이 나왔다. 되돌아보면, 이것은 적어도 나에게 있어 유머였다.

지나고 나서 보니, 우리는 런타임 기능의 개발과 함께 우수한 툴 창조 가능하게 한 프로그래밍 팀 구조를 만들어야 했는데 어쩌면 개발 초기에 엔지니어링 팀의 더 많은 부분을 툴에 할당했어야 했을지도 모르겠다.

앞으로 나가기 위해, 우리는 먼저 대부분의 클라이언트-패이싱 작업으로 시작하는 툴을 구성한 다음, 콘텐츠 창작자들의 요구를 충족하기 위해 반복적으로 툴을 개선시키고자 한다. 아울러 현재의 요구를 대변하는 투자를 위해 이론적인 미래의 이익을 도출할 것을 약속하는 기술 투자에 대한 집중을 분산시킬 것이다.

3. 콘텐츠 사태

*Brütal Legend*는 소형 게임이 아니다. 다행히, 우리는 우리가 직면했었던 것이 무엇인지 그리고 데이터/빌드 인프라에 집중적으로 투자된 것이 무엇인지를 안다고 생각했다. 끔찍하게 실패했던 것은 우리가 총 콘텐츠 푸시를 과소평가했다는 점, 그리고 더 중요한 것은 제작 마무리 단계에서 거대한 콘텐츠 스파이크를 예상하지 못했다는 점이다. 게임 시작에서부터 2008년 말까지, 우리의 데이터 격변 비율과 데이터 성장 비율은 상당히

안정적이었으며 직원 및 팀 생산성의 증가치와 대략적으로 일치했다. 이것은 기술에 의해 예측 및 계획되고 지원되었다.

그러나 그 때, 2009 년 1 월, 모든 것이 폭발했다. 갑자기 말이다. 개발 3 년 후 우리는 최적화 및 포장된 게임 데이터의 약 2.5GB 를 누적했으며 그 후 4 개월이 지나지 않아 9GB 로 뛰어올랐다.

이것의 주된 이유는 동시에 다양한 팀으로부터의 자산 전달이 크게 증가했기 때문이었다. 예를 들어, 몇 주만에 0 개의 로컬라이즈 파일이 약 100,000 개가 된 것이다.

우리는 Jack Black 인트로 및 우리의 모든 주 메뉴를 위한 고해상 비디오 자산을 무더기로 수신했다. 늦게서야 추가 오디오 작업을 계약하기로 결정했으며, 새 앰비언스와 사운드 이펙트가 신속하게 게임에 추가되었다.

다양한 유형의 콘텐츠에 걸쳐 이렇게 동시적인 큰 증가는 우리의 전체 인프라, 특히 빌드 머신, Perforce 서버 및 네트워크 백본에 엄청난 부담을 안겨 주었다. 문제를 악화시키기 위해, 우리는 폭포효과를 살펴보기 시작했다. 한 대의 시스템(예: 10,000 개의 .wav 파일의 체크인)에 대한 큰 히트가 Perforce 를 끔찍 못하게 할 경우 모든 의존형 시스템(우리의 빌드 서버 및 개별 체크인)에서 병목현상이 발생하게 되고 그러면 이러한 병목현상은 또 다른 병목현상을 야기했다.

이것은 결코 실패하지 않는다는 당사의 파이프라인과 빌드 인프라에 대한 신뢰이지만, 우리는 심각한 성능 저하를 무수히 경험했는데 이중 많은 것들이 엔지니어링 팀의 긴급 중재를 필요로 했다. 이렇게 예기치 못한 급한 진화는 가능한 최악의 시간(데모 또는 마일스톤 준비 도중)에서 항상 생산성 손실을 야기했다.

또한 이러한 대량의 콘텐츠 덤프는 우리의 런타임 시스템에 상당한 긴장감을 심어주었다. 음성 시스템의 라인당 메모리 오버헤드가 수 만개의 라인을 취급하기 위한 준비를 갖추지 못했고, 결국 듀얼 레이어 DVD 에 꼭 맞추는 우리의 능력에 혼란이 발생했다.

전면적으로, 이러한 콘텐츠의 예기치 못한 증가는 우리 IO, 메모리 및 프로세싱 프로파일 전체에 걸쳐 파급효과를 야기했다. 그리고 증가율이 높고 또 예측되지 못했기 때문에 이 시스템에 대해 논쟁하는 엔지니어들은 그들에게 할당된 작업에서 손을 떼고 긴급 진화 작업에 투입되었다.

이제 점점 더 우리는 콘텐츠 창작자들과의 작업에 대해 훨씬 더 많이 알게 되어 그들이 창작하기로 계획하는 전체 데이터 량을 미리미리 추정하고 이러한 수치를 우리의 기술 디자인에 포함시켜 우리가 제품의 최종 요구를 충족할 수 있도록 할 것이다. 또한, 우리는 필요한 것으로 입증될 경우 새 용량을 온라인으로 신속하게 가져오기 위한 더 나은 위치를 점할 수 있다는 희망으로 확장가능한 데이터 인프라에 더 많이 투자할 계획에 있다.

그러한 개선과 약간의 운으로 인해, 콘텐츠 사태 처리가 우리의 향후 프로젝트에 대해 자랑하는 무언가가 되었으면 한다.



4. 조력가와 실행가

Brütal Legend 가 제작 단계로 들어섰을 때, 팀의 핵심적인 실행 직책에서 인원부족 현상이 있을 것이라는 사실이 분명해 졌다. 이 요구를 충족하기 위해, 우리는 조력 부문(예: 디자인 및 제작) 고용을 위해 제쳐둔 직책을 재활당하여 실행가(예: 관리자 및 프로그래머) 인원을 더 많이 보충했다.

실행가쪽으로 인원을 더 많이 보충해야 했던 동시에, 우리는 제작 단계로 깊이 들어갈 때까지 이러한 직원 교환에 의해 전체적인 효율성이 떨어졌다는 사실을 인지하지 못했다. 효율성 저하 이유는 재작업을 해야만 하는 디자인이 통계적으로 증가한 점, 관리 실수가 늘어난 점, 그리고 전체적인 의사소통의 잘못에 의한 것이었다.

디자인 부서의 인원부족으로 인한 대가는 기능이 완전히 계획되기 전에, 종종 대충 논의만 되었던 아이디어들이 실행되고 있음을 의미하는 것이었다. 기능이 부분 또는 전체적으로 실행되고 난 후에만 심각한 결함이 발견되는 경우도 가끔 있었다.

이는 또한 시작을 잘못된 후 요구되는 재작업에 추가되기도 전에 많은 의사결정과 규격이 항상 밀려 있음을 의미했다. 이것은 종종 종속성 체인의 맨 위에 있기 때문에, 디자인 병목현상은 팀의 나머지 부분에 유해한 영향을 미쳤다. 특히 우리의 광범위한 게임 범위를 생각해 볼 때 말이다.

같은 인원부족 증상은 제작 부서에서도 발견되었다. 프로듀서들은 커뮤니케이션, 종속성을 해결하는 것, 팀이 합리적인 시간 안에 작업량을 보장하는 것, 그리고 프로젝트를 일정에 맞게 유지하는 것에 책임이 있다. 프로듀서들은 팀이 도달할 앞길이 잘 닦이도록 해야 한다. 인원 부족으로 인해, 제작 책임 중 일부를 리더가 지게 되거나, 안전망이 없어 초과근무와 스트레스로 비효율성이 전달되었다.

생각해 보면, 조력 및 실행 직책의 직원을 고용하는 것에 똑같이 우선순위를 두었어야 했다. 게임을 만드는 우리의 능력을 충족하기 위해 게임의 범위를 정하는 것은 거의 확실히 더 작은 게임을 의도한 것일 수 있지만, 더 많이 마무리될수록 더 원활하게 관리되었다.

5. Double Fine, 서비스를 받다

2009년 6월, Activision Entertainment Holdings, Inc.는 Double Fine 을 상대로 소송을 제기했는데, 소송 이유는 계약 파기였고 Electronic Arts 의 Rocktober 13 게임 출시를 막기 위한 잠정적 금지명령을 얻기 위해서였다. 2 달도 안 되어 사건은 재판없이 합의되었다. 필자는 본 사설 이 사건에 대해 일체 말할 수는 없지만 퍼블리셔들간의 전환과 이후의 *Brütal Legend* 의 개발에 대한 소송의 영향으로 인해 잘못된 무엇인가로 고소당할 수 있다는 것을 알게 되었다.

옛날로 약간 거슬러 올라가보자. 우리는 Vivendi 가 Activision 에 합병되고 소송이 퍼지는 동안 양 퍼블리셔들과의 연락이 끊어질 때까지 2 년 동안 Vivendi 에서 다양한 그룹과 협업하여 성공적으로 작업했다.

합병 발표와 그 후 이어진 Vivendi 직원과 퍼블리셔의 연락 감소는 팀의 내부 불안과 사기 저하를 야기했다. 특히 이들의 관계가 이전에 굉장히 조화로웠던 후에는 더욱 그러했다.

회사 미팅에는 종종 우리 퍼블리셔의 현재 상황에 대해 적으나마 우리가 알았던 것, 그리고 Double Fine 을 위해 계획되었을 수 있는 다양한 결과 에 대한 실망스러운 논의가 포함되었다.

이 혼란스러운 불확실성은 몇 달 동안 지속되었는데, 이 기간 동안 리더들은 퍼블리셔 지불이 없을 경우 돈이 씨가 말라가는 것을 지켜보면서 예정된 마일스톤을 달성하도록 계속해서 팀에 동기를 부여했다.

우리는 출하할 게임을 목록화한 Activision 의 공식 언론 발표를 통해 그들이 *Brütal Legend* 를 퍼블리싱하지 않을 것임을 알게 되었다. 그 목록에 우리 게임은 없었다. 또 다시 팀은 불안에 떨어졌다. 공식적인 새 퍼블리셔 탐색이 시작되었는데 이는 이미 가혹한 개발 주기 중에 있던 팀과 나, 여러 팀 리더들을 당혹케 했다.

Electronic Arts 와 게임 계약을 다시 체결한 후에도 Double Fine 과 Activision, Electronic Arts 간의 법적 교류가 시작되기 전에 짧은 유희기간만을 즐길 수 있었다. 팀의 대부분은

Electronic Arts 가 퍼블리싱할 게임을 선정했음을 발표한 2008 년 12 월과 소송이 합의된 2009 년 7 월 사이에 전개된 드라마로부터 보호되었다. 그러나 Double Fine 의 리더십은 아니었고, 방심과 스트레스가 개개인 및 납품본에 영향을 미쳤다.

게임이 Alpha 로 가자마자 소송이 제기되었다. Gold Master 가 제출되기 전에 그 소식이 들렸다는 이유였다. 팀과 나, 팀 리더들의 간부들은 가장 중요하고 가장 크런치가 심했던 제작 기간 중에 정보 수집, 선언 작성, 소송 해석, 증인 인터뷰 및 비게임 제작과 같은 모든 골치 아픈 작업을 맡게 되었다. 소송은 팀과 회사, 우리의 제품 및 우리의 낙관론에도 영향을 미쳤다. 어느모로 보나 잘못되었던 것이다.



Double Fine, 메탈을 위하여

전설에서부터 장소, 생명, 언어학에 이르기까지 게임에 포함된 모든 것은 Double Fine 에 의해 개별적으로 수작업되었다. 75 개 밴드, 107 개의 라이선스된 헤비메탈 트랙과 무수히 많은 헤비메탈 앨범 커버는 게임의 크리에이티브 디렉션에 영감을 주었다.

Kath Soucie, Zach Hanks 및 Jennifer Hale 과 같은 베테랑 비디오 게임 배우들의 역할을 수행하는 스타 목소리와 함께 Jack Black, Rob Halford, Lemmy Kilmister, Lita Ford, Ozzy Osbourne, Tim Curry, Kyle Gass, David Cross, Brian Posehn 및 Wil Wheaton 등 유명인들의 목소리는 게임에 쓰여지고 기록되고 편집되고 통합된 40,000 개의 대화를 통해 150 개 이상의 오리지널 캐릭터에 생명을 불어넣었다.

Glenn Tipton 과 K.K. Downing 은 직접 Eddie, Ophelia 및 Doviculus 의 오리지널 기타 리프를 만들고 녹음했다. Eddie 가 군대에 명령을 내리고 악을 물리치기 위해 메탈의

파워를 이용하는 수단인 도끼 하나, 기타 한 대 그리고 핫 로드를 어떻게 사용하는지 설명하기 위해 다음 몇 가지 단어를 인용해 보겠다 -- 3 인칭 전투, 싱글 플레이어, 멀티 플레이어, 오픈 월드, 드라이빙, RTS, 어드벤처.

결국, 가족과 친구들의 인내와 포용력 그리고 Electronic Arts 의 깊은 지원으로 버텨낸 83 명의 개발자들은 불확실성과 실망 앞에서 그들의 기술과 지독한 끈기에 대한 극도의 헌신을 증명해 보였다. 퍼블리셔 플레이트 이동, 휘몰아치는 계약 협상, 소송, 잘못된 방향과 방향 재설정, 포커스 테스트 충격 및 지나친 의욕을 극복한 것이다.

이제 독점적인 게임 엔진, 견고한 툴 파이프라인, 재능과 경험을 갖춘 직원 그리고 창작의 자유와 혁신을 위한 기업 방침으로 무장한 Double Fine 은 다음 번의 웅장한 여행길에 오르기 위한 유리한 위치를 차지했다.

게임 데이터

개발자: Double Fine Productions, Inc.

퍼블리셔: Electronic Arts

플랫폼: Xbox 360, PlayStation 3

출시일: Rocktober 13, 2009

내부 개발자 수: 74 명

외부 계약자 수: 9 명

예산: 2,400 만 달러

개발 기간: 4.5 년

대표적인 워크스테이션: Dell Quad Core Xeon, 4GB RAM, 150GB, NVidia GeForce7800, Windows XP

사용된 소프트웨어: MS Visual Studio 2005, Autodesk Maya 2008, ZBrush, Adobe Photoshop, Adobe After Effects, ProTools, Sound Forge, Perforce, Microsoft Xbox360 XDK, Sony PlayStation 3 SDK, 내부 개발 툴 모음

총 코드 라인:

게임(C++, 미들웨어 제외): 554,736

게임(Lua): 42,745

툴(C++ & C#): 52,263

툴(MEL): 85,294

툴(Python): 34,079

총 퍼포스 체크인 수: 164,863 개