

포스트모템: 벡터 유닛의 '하이드로 썬더 허리케인'

매트 스몰

[EA와 스톰프론트 스튜디오(Stormfront Studios)의 베테랑 매트 스몰(Matt Small)은 Xbox Live Arcade용 하이드로 썬더 허리케인(Hydro Thunder Hurricane)을 개발하면서 큰 스튜디오에서 작은 소규모 회사에 이르기까지 그가 이끈 소규모 팀의 변천사를 통해 겪은 성공과 실패를 공유한다.]

"그래서, 나는 게임과 관련된 사소한 일을 중단하고 게임을 하는 것에 대한...이러한 생각을 하고 있었어."

2007년 여름 밤 내 채팅 창에는 위와 같은 말이 오고 갔다. 나의 친구 랄프 크노에셀(Ralf Knoesel)과 나는 아동용 작은 게임 프로젝트를 진행하고 있었던 것 같다. 아니면 최소한 이 프로젝트에 대해 논의하고 있었다. 문제는 그 당시 우리가 우리의 삶으로 생각했던 월급을 받는 다른 직업을 가지고 있었다는 것이다. 그렇기에, 프로젝트 진도는 매우 더디게 나갔다.

내가 아는 모든 개발자들은 언젠가는 현 상황에서 탈출해 그들만의 게임을 만들겠다는 백일몽에 빠져 있다. 랄프와 나는 회사를 설립하기 위해 각각 12년 넘게 노력했다. 최근에는 EA 레드우드 쇼어스(EA Redwood Shores)의 아트 리더로서, 랄프는 스톰프론트 스튜디오의 프로그래밍 리더로서 활동했다. 우리는 약간의 돈을 모았고, 더 맨(The Man) 작업에 실증이 난 상태였다. 랄프의 뜻밖의 제안은 내가 필요로 했던 모든 시도였다.

2008년 1월, 우리는 작업을 그만두었다. 개인적인 자산을 모두 청산하고 우리 PC를 업그레이드하여 벡터 유닛(Vector Unit)이란 새로운 회사를 설립했다. 우리는 직원이었을 때는 생각지 못한 세세한 여러 사항을 처리했다. 작은 비즈니스 관리, 계약 협상, 단체 건강 보험 등에 대한 내용이었다.

우리가 만들기로 한 게임은 처음에는 SF 러시(SF Rush)와 하이드로 썬더(Hydro Thunder) 같은 고전 아카이드 레이서로의 회귀인 스케일은 작지만 거대한 XBLA나 PSN용 워터 레이싱 게임이었다.

우리는 역동적인 수면으로부터의 탈출을 좋아하기 때문에 물을 소재로 한 게임을 생각했다. 위 스포츠(Wii Sports) 같은 미니 게임은 제외하고 지난 10년 동안 관창은 워터 레이싱 게임은 출시되지 않았다. 또한, 보트를 이용해 전투를 즐기는 첫 Xbox용 블러드 웨이크(Blood Wake)가 출시되기 전 수 년 동안 우리는 함께 일을 했다. 우리는 우리가 만들 수 있을 거라고 판단한 무엇인가에 대해 착수하기로 했다.

6개월 후 우리는 친구들 앞에서 수 많이 연습했고 잘 때도 읊었던 파워포인트 피치, 게임 엔진, 경기하기 쉬운 PC 프로토타입을 갖게 되었다. 우리는 이를 이곳 저곳에 알렸고 결국 Microsoft Game Studios의 관심을 이끌어 냈다.

하이드로 썬더 라이선스를 충족하기 위해 우리의 디자인을 변경하면 어떨까라는 생각이 마이크로소프트와의 대화 가운데 나왔다. 우리는 우리의 프로토타입을 믿었지만 알려진 라이선스로 인해 새로운 스튜디오로서 첫 작품에 매우 필요한 언론의 관심을 받을 것이라는 것을 알고 있었다.

MGS는 최근엔 사용하지 않은 미드웨이 게임즈(Midway Games)로부터 라이선스를 구입한 워너 브라더스로부터 판매권을 얻었다. 우리는 우리의 디자인 문서를 다시 작성하고 범위를 3배 늘려 계약 협상을 통해 우리의 갈 바를 법적으로 정했다.

2008년 4월 우리는 최종적으로 거래에 서명을 했고 첫 기금과 360개의 첫 개발 키트를 받았다. 하이드로 썬더 허리케인의 진행 상황은 가다 서다를 반복했다.

"가다"는 약간 과장일 수 있었다. 우리는 여전히 팀을 구성하고 사무실을 찾으며 업데이트된 프로토타입에 우리의 새 디자인을 만족스럽게 적용해야 했다. 그리고 360에서 구동 중인 우리의 PC 엔진을 구해야 했다.

우리가 벡터 유닛을 설립할 때 기존의 게임 개발 스튜디오에서 대규모 팀을 관리했던 경험을 통해 좀 더 작은 게임 프로젝트 개발을 원활하게 관리할 수 있을 것으로 생각했다. 오히려, 더 쉬울 것이라 생각했다. 돌아서 생각해 보니 어느 정도 맞는 판단이었다. 작은 게임 개발이 우리에게 보여주는 수 많은 작고 놀라운 소식을 우리는 믿지 않았다.

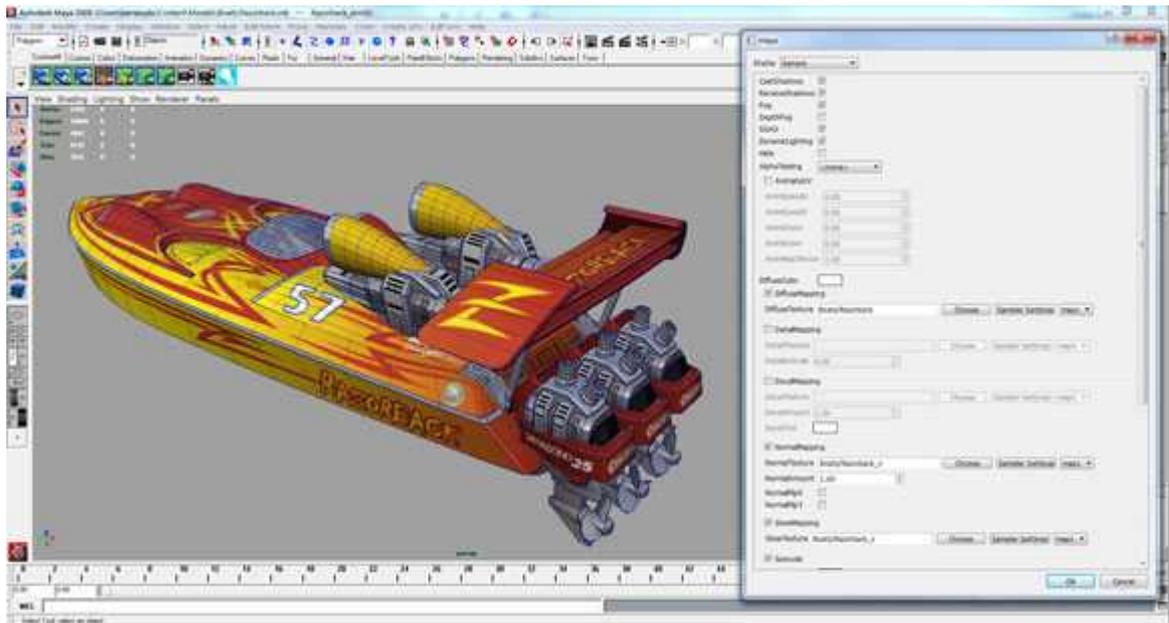
잘한 일

1. 틀이 있었던 초기

우리의 원 프로토타입을 작업했던 초기, 게임 엔진에 대한 작업 이전에 수행했던 첫 기술적인 작업은 제작 툴 세트에 관한 것이었다. 우리는 콘텐츠 제작이 우리에게 중요한 일이고, 우리의 핵심 팀을 작게 유지하고 대부분의 아트를 외주를 주는 것이 계획이었다.

계약자가 최소 빌드 타임을 재빨리 반복하도록 하고 기존의 자산을 다시 구축할 필요없이 필요한 대로 기능을 추가할 수 있도록 조정하기 쉬운 파이프라인이 필요했다.

우리는 수 많은 타사 게임 개발 세트를 연구했다. 하지만 이들에 대해 부정적이었다. Unreal같은 엔진은 우리의 예산 범위를 초과했고 우리가 실제로 필요로 수준보다 더 많은 파워를 전달했다. 그리고, XNA와 토크(Torque)같은 독립 세트는 우리가 필요하다고 판단하는 낮은 수준의 제어를 제공하지 않았다. 그래서, 우리는 애초에 우리만의 것을 만들기로 결정했다. 이는 우리의 최고 결정 중 하나가 되었다.



기본 아트 자산 파이프라인이 최우선 고려 사항이었다. 랄프는 데이터 중심의 셰이더 시스템을 개발했다. 이 시스템은 게임 셰이더 코드를 사용한 셰이드 뷰포인트에서 완전한 하드웨어 렌더링을 지원하는 셰이더 플러그인을 통해 Maya로 통합된다. 아티스트들은

일반적이거나 상세한 맵핑 같은 셰이더 레이어를 추가하고, 변경 결과를 즉시 볼 수 있으며, 게임으로 내보내지 않고 자유롭게 반복할 수 있었다. 개발 과정 동안 이러한 단순한 기능은 개발 시간을 수 주 단축하는데 도움이 되었다. 아티스트들은 자연스럽게 느끼는 방법으로 작업을 할 수 있었다.

게임 에디터, PFX 에디터, 기타 툴은 모두 공통 포맷(xml 또는 json)으로 내보낸다. 이러한 포맷은 게임 자체가 실행 시간에 구축되고 업데이트 자산을 요구한 대로 제작한다. 아티스트들에게 있어서 이는 가장 폴리 헤비(poly-heavy)하고 텍스트가 많은 자산 측면에 대해서도 Maya에서 보내기 버튼을 누르는 것부터 게임에서 자산을 바빠 돌아다니게 하는데 걸리는 시간은 30초를 넘지 않고 보통 이보다 훨씬 적다. 이 툴은 또한 모든 파라미터를 변경할 때 다시 구축해야 할 필요를 줄이는 생생한 원격 측정을 지원한다.

최종적으로, 개발하는 동안 우리는 게임의 작업 PC 빌드를 유지했다. 최종 자산이 항상 360에서 테스트받고 TV에서 평가된다고 해도 PC 빌드는 우리가 외주업체를 위한 개발 또는 디버그 키트를 제공해야 하는 의무감에서 벗어날 수 있도록 하고, 팀원들이 집에서 일하고 융통성있게 시간을 사용하도록 한다.

또한, 모든 것을 구축하지 않았더라는 것을 말하고 싶다. 우리는 오디오를 위해 FMOD, 충돌과 강제 물리(rigid body physics)를 위해 Bullet을, 버전 제어를 위해 진정으로 놀라운 Subversion(특히, TortoiseSVN)을 사용했다.

2. 배위에 있는 나!

우리가 워터 기반 게임을 엄청나게 좋아하는 팬이라는 것을 처음에 언급했다. 요동치는 파도를 헤치고 나가는 파워풀한 보트의 느낌을 주는 것이 게임의 성패를 좌우하는 핵심이 될 것이라는 것을 처음에 감지했다.

유체 역학 시뮬레이션은 게임 엔진의 첫 부분으로서 온라인으로 실현될 부분이었다. 로우-폴리 "유체" 선체 주변을 모델로 한 부력과 흐름 계산을 기반으로 하지 않았고 물에서 실질적인 보트의 움직임을 꽤 정확하게 모델로 삼아 작업했다.

이러한 정확성으로 인해 우리는 선체의 모양을 기반으로 사실적으로 생동감있는 다양한 보트 제어 모델을 제작할 수 있었다. V 모양의 고속 모터보트 선체는 속력을 내면 사실감있게 물 밖으로 떠오른다. 쌍동선 폰툰(catamaran pontoon)은 번갈아 측면 안정성을 유지하고, 바닥이 평평한 선체 수상비행기가 고속으로 수면을 가로질러 난다.

그러나, 이러한 리얼리티는 필요 이상으로...사실적이라는 것을 초기에 깨달았다. 실질적인 연안의 모터 보트는 일반적으로 잔잔한 수면에서 최대 100mph의 속도를 낼 수 있지만, 우리는 보트가 파도가 치는 상황에서도 200mph가 넘는 속도를 낼 수 있도록 했다.

우리는 인공적인 다운 포스를 추가했고 고속으로 부력을 줄였으며 과장된 아카이드 레이싱 경험에 필요한 정신 없이 빠른 속도를 실현할 수 있을 때까지 수많은 수를 변경했다.

우리는 프로토타입 단계에서 초기에 사용자 테스트를 시작했고 시간을 낼 수 있는 친구들을 불러 모아서 이 게임을 어떻게 하는지 방법을 알려주지 않고 컨트롤러를 주고 게임을 해보라고 했다. 첫 경험의 결과는 초라했다. 수 주 동안 우리는 우리의 테스트 트랙에서 안정적으로 레이싱을 했다. 우리의 첫 테스트 대상은 이 벽에서 저 벽으로 왔다 갔다 세계 부딪히는 데에 게임 플레이의 대부분이었다.

이는 고전적인 디자인의 어려웠고, 디자이너는 제어 방식에 너무 가까이 있어서 결국 이러한 특이한 성격에 익숙해진다는 사실을 다시금 깨달았다. 여러 차례 반복한 후 우리는

단순화된 회전 모델을 제작할 수 있었다. 이 모델은 충분히 리얼리티가 살아 있지만 게임을 처음하는 사람이 제어하려면 직관에 의지해야 했다.

첫 프로토타입의 구동 방법은 결국 HTH를 사용한 것에서 여전히 요원했다. 하지만, 컨셉을 성공적으로 실현하기 위해 그리고 게임의 반응이 좋을 것이라고 퍼블리셔를 확신시키기 위해 거의 충분했다 사용자 테스트를 통해 얻은 교훈을 개발 기간 동안 계속 상기했고 방법을 조정하고 개선하는데 이를 사용했다. 우리는 우리가 제공한 제품에 대해 자부심을 느낀다. 한 리뷰어가 말한 대로 "이는 버터와 같은 것을 다루는 것이다."

3. The Man과 친구하기

이는 마치 졸졸 따라다니며 아부하는 꼴이다. 하지만 이를 "올바른 일"로서 열거하지 않은 것은 태만이다. **Microsoft Game Studios**는 훌륭한 퍼블리싱 파트너이다.

변덕스러운 퍼블리싱 파트너와의 좋지 않은 관계는 잘 짜인 계획을 망칠 수 있다. 퍼블리셔가 디자인 권한에 달려있고 그들이 계약상 지불해야 할 의무가 있는 마지막 순간까지 마일스톤 지불을 보류하여 유능한 소규모 독립 회사가 당황하는 경우를 봤다.

MGS는 훌륭했다. 우리의 프로듀서는 우리가 원하는 창조적인 자유를 주었고 실질적으로 게임을 좀 더 잘 만들 수 있도록 편집 피드백을 제공했다. 그는 항상 우리가 협의하지 못한 점에 대해 논하고 타협하려고 했다. **MGS** 팀은 복잡하고 상세한 차트와 도표를 제공할 필요없이 우리의 **SCRUM-lite** 스케줄링 방법을 사용해서 함께 작업하는 데에 동의했다. 그리고, 이들은 항상 마일스톤 승인에 즉시 지불했다.

우리는 이 모든 것을 무료로 얻었다. 퍼블리셔와의 관계는 의식적인 노력을 기울여 관리해야 한다. 무엇보다 우리는 적기(**red flag**)를 피하기 위해 무진 애를 썼다. 우리는 정시에 마일스톤을 제공했고 이들을 사전에 완전하게 테스트해서 우리가 합의한 대로 인도물을 작업하도록 한다.

일정의 후반부로 갈수록 우리는 우선적인 내부 **MAT** 테스트를 통해 원활하게 성공적으로 준수했다. **MSG**가 제기한 문제 또는 우려 사항에 재빠르게 대처했다. 그리고, 우리는 새로운 콘텐츠와 데모가 준비된 빌드 플로우를 계속 제공했다. 그래서, 항상 새롭고 매력적이거나 웃긴 무언가가 있어서 리뷰 미팅에서 자랑스럽게 선보일 수 있었다.

이러한 노력은 성공적이었다. 우리는 마일스톤을 거절당한 적이 없었다. 마일스톤을 통해 돈을 벌고 회사는 탄탄대로를 걸었다. 그리고, 우리의 프로듀서를 행복하게 하여 우리가 원하는 창조적인 휴식의 장을 주었다.



4. PWNage

우리에게는 우리의 원 PC 프로토타입에서 매우 초기에 작동하고 있는 분할 스크린 멀티플레이어가 있었다. 분할 스크린을 통해 시간을 AI에 투자하기 전에 우리의 레이싱 경험을 플레이테스트를 했다. 그리고 우리는 대부분의 게임 메커닉스(특히, 물리적인 보트의 지나간 자리와 앞 보트를 뒤따라가는 모습)를 현지의 멀티플레이어 경험에 맞추었다. 편안한 긴 의자에 앉아 팔꿈치를 부딪히며 레이싱하고 서로 공격적으로 대화하며 사진 판정 승리에 깽 소리를 내는 경험은 게임의 중요한 매력이 될 것이라고 우리는 판단했다.

온라인 게임을 할 때 마이크로소프트의 QNet 프레임워크를 선택했다. 이 프레임워크를 통해 우리는 360에서 처음으로 빠르게 할 수 있었다. 그 후 지연 효과와 기타 네트워킹 아티팩트(networking artifact)에 대해 물리학 및 충돌 테스트의 균형을 맞추려고 몇 달 동안 조율 및 실험을 했다.

물 물리학 엔진으로 우리는 보트가 씰룩씰룩 움직일 수 있도록 하고 보트가 상이한 파고에서 움직일 때 자연스럽게 물위에서 흔들리고 물 위에서 미끄러지도록 한다. 반듯하게 운행하고 네트워크 지연으로 인한 러핑 효과(lerping effect)를 빠르게 드러내는 차와는 반대이다. 우리는 프로젝트 내내 클라이언트의 물리 보간법(physics interpolation)을 지속적으로 변경했다. 그리고, 현재 시장에서 모든 레이싱 게임의 라이벌이 된 매우 자연스러운 온라인 게임이 되었다.

개발 단계에서 후에 온라인으로 나온 추가적인 디자인 개선 사항 중 하나는 멀티플레이어에서 “루저 헬퍼” 시스템으로의 스위치였다. 이 스위치는 따라잡기 위해 트레일링 플레이어에 추가적인 부스트와 기회를 제공한다. 우리의 프로듀서가 이러한 제안을 했고, 나를 포함한 우리 팀의 일부가 초기에는 이에 회의적이었지만 첫 테스트 결과는 괜찮았다. 레이스는 점점 치열해지면서 흥미진진하게 되었다. 대부분의 노련한 플레이어는 지속적으로 이길 수 있었지만, 대부분 1/10초 또는 1/100초 차이로 승부가 갈렸다.

HTH에서 4명의 플레이어가 함께 할 수 있는 분할 스크린 레이싱을 포함하는 것과 로컬 플레이어가 온라인에서 함께 레이싱을 할 수 있도록 지원하는 것은 플레이어와 리뷰어

모두에게 호평을 받았다. 이들의 다수는 요즘의 레이싱 게임에서 분할 스크린이 없는 것에 대해 애용해했다. 우리에게 있어서 이는 엄청난 성공이며 향후의 모든 게임에서 우리의 작업에 큰 영향을 미칠 것에 대해 의심치 않았다.

5. 직원 고용의 융통성

저축과 프로젝트 기간 동안의 마일스톤 지불을 통해 전적으로 꾸려지는 벡터 유닛을 통해 우리는 처음부터 직원을 최소로 하고 독립 계약자와의 아웃소싱을 통해 게임을 독점적으로 제작해야 한다고 생각했다.

우리의 고용 계획은 레이싱 환경을 만들고 텍스처링하는 작업을 나와 함께 하기 위해 랄프, 1명의 사운드 디자이너, 2~3명의 아티스트 외에도 프로젝트 기간 함께 할 프로그래머가 필요했다. 프로젝트에 도움을 준 대부분의 사람들이 현장과 현장 외에서 일했음에도 불구하고 이 정도가 프로젝트 기간 동안 필요한 직원의 대부분이었고, 우리의 “내부” 팀에 대해 생각한 것이었다. 보트와 아트 작업의 대부분, UI 디자인, 기타 잡무는 “외주”를 주었다. 즉, 좀 더 작고 신중한 작업은 현장 외의 지역이나 해외에서만 일하는 개인이나 그룹에 넘겼다.

우리는 직원 고용을 꼼꼼하게 계획했지만 사실 굉장히 운이 좋아서 수 개월 만에 훌륭한 팀을 구성할 수 있었다. 이러한 관점에서 보면 경제 불황이 우리에게 도움이 되었다. 2009년 SF Bay Area에서만 수 천 명의 게임 개발 전문가들이 일자리를 잃었다.

운 좋게도 이러한 비극은 희망을 가져다 주었다. 프로젝트를 시작할 때 우리는 실력있는 친구들과 일하기 원했던 예전 동료들을 고용할 수 있었다.

이로 인해, 우리는 아주 무난하게 우리의 프로젝트를 진행할 수 있었고 2009년 가을 완전한 제작이 가능할 정도로 성장했다. 개발에 제동이 필요할 때 우리는 소방수 역할의 단기 계약자를 추가했다.

계약자가 거의 매일 우리와 긴밀히 작업을 했기 때문에 우리는 비효율적인 지출과 아웃소싱한 프로젝트에 종종 발생하는 간접 운영비를 줄일 수 있었다. 현장 외에서 콘텐츠를 제작하는 것에 대해 우리는 해외의 아티스트와 현지 샌프란시스코 사무소가 있는 아트 아웃소싱 그룹인 컨셉 아트 하우스(Concept Art House)와 일차적으로 작업했다. 이러한 현지 팀과의 작업과 반응은 우리의 내부 간접비용을 줄이는데 추가적으로 도움이 되었다.

정직원이 2명뿐이라서 사람들은 종종 벡터 유닛은 2명이 운영하는 상점같다는 인상을 받기도 한다. 사실과 크게 다르지 않다. 이곳 저곳에서 도움을 주는 사람들의 숙련된 경험, 독창성 및 책임감없이 하이드로 썬더 허리케인은 태어날 수 없었다.

잘못된 일

1. 작은 팀

EA에서 20명의 아티스트로 이루어진 팀을 관리하고 있을 때 한 명이 아프게 되면 이는 작업 진행에 차질을 빚게 되었다. 당신에게 3명의 아티스트가 있으면 제작 측면에서 33%를 잃는 것이다.

이것이 소규모 스튜디오를 운영하면서 첫 프로젝트에서 호되게 겪은 하나의 교훈이었다. 제작 역학은 극도로 변화한다. 생산성 변동은 프로젝트 관리에 있어서 자연스런 부분이다. 직원은 병가, 육아 휴직, 퇴사를 한다. 일부 팀원은 다른 팀원보다 좀 더 생산적이고 일부 작업은 당신이 예측한 것보다 더 오랜 시간이 걸릴 수 있다.

그로 인해, 게임은 파악해서 쉽게 해결해야 했던 수 많은 문제가 있는 채로 출시되었다. 그래서 다른 것들 중에 일부는 리더 보드 활용에 매우 불쾌해 했다. 이러한 사항이 대다수의 플레이어에 통보되지는 않았지만 이러한 버그는 최종 제품의 전체적인 완성도를 떨어뜨렸다.

우리는 개발 후반부에 내부 QA 연락 담당자를 고용할 예정이어서 내부/외부 커버리지를 지휘하고 이러한 문제를 다루도록 할 예정이다.

3. 네트워킹 에지 케이스

네트워크 멀티플레이어가 HTH의 핵심이었고 “잘한” 일 중 하나였지만 이행 측면에서 원래 제공하려던 것보다 훨씬 어려움이 많았다.

어려운 부분은 플레이어 경험과는 관련되지 않았다. 지연 처리, 물리학적 동기화, 데이터 대역폭 관리 등이 어려웠다. 이러한 모든 사항은 노력과 인내를 필요로 했지만, 우리의 소규모 네트워킹 팀(한 명)이 이를 테스트하고 해결해서 최소한 예측가능하고 해결될 수 있었던 문제였다.

난해한 문제는 용역 측면의 모든 커뮤니케이션이었다. 라이브 서버와 대화하고 떨어진 플레이어, 현지 및 네트워크 접속이 끊긴 플레이어에게 친절하게 대하는 것들이었다. 우리가 예상한 네트워킹 일정을 자세히 살필 때 마이크로소프트의 Microsoft 라이브러리는 상대적으로 포괄적인 라이브 관련 문제의 대부분을 해결할 패키지 솔루션을 들고 나올 수 있다. 실질적으로 우리는 대부분의 솔루션을 우리 스스로 제작해야 함을 발견했다.

이를 혼합하는 것은 작은 스튜디오가 해결해야 할 또 다른 도전 과제였다. 당신에게 오직 6명의 사람이 있을 때 8명의 플레이어가 하는 온라인 게임을 스트레스 테스트하는 것은 어려운 일이다.

VMC는 그들 측면에서 스트레스 테스트를 꽤 많이 수행했고 그들과 마이크로소프트의 팀은 종종 우리의 내부 MP 테스트를 위해 함께 작업했다. 그렇기는 하지만 게임이 출시된 후 그들만 알 수 있는 몇 개의 연결성 버그가 있었고, 우리는 동시에 빠르게 게임에 참가하고 떠나는 수 많은 사람들을 통해 최종적으로 실제 세상에서의 테스트 키친(test kitchen)을 보유했다.

운 좋게도 우리는 추후 출시한 타이틀 업데이트(Title Update)에서 이러한 문제를 해결할 수 있었다. 하지만, 우리가 진심으로 게임을 즐기길 바랐던 바로 그 사람들(게임을 출시 초기에 구입해서 온라인 커뮤니티를 자리잡게 한 핵심 플레이어들)의 일부는 가끔 발생하는 결함에 당황했다.

4. 사용자 경험 숲

하이드로 썬더 허리케인은 플레이가 지겹지 않은 몇 안되는 게임 중 하나이다. 거의 매일 우리는 하루에 30분 이상 멀티플레이어 전투에서 치열하게 싸우거나 Leaderboard Opponent 분할 시간 기능을 통해 각자의 시기가 최고가 되도록 했다.

수많은 수정과 조율을 통해 각을 넓히고 파도의 높이를 조정하는 등의 플레이 세션이 나왔다. 그리고 게임을 즐겨하는 사람들을 통해 HTH에서 시시각각의 경험으로 게임이 쉽고 즐겁게 되었다.

그러나, ‘경기성(playability)’이라는 나무에 중점을 둔 상태에서 우리는 사용자 경험이라는 숲을 간과했다.

HTH의 싱글 플레이어 게임은 플레이어에게 수 많은 콘텐츠를 제공한다. 8개의 트랙이 있고 각 트랙에는 Race, Gauntlet, 세 개의 Ring Master가 있다. 풀어야 하고 플레이를 할 40개의 이벤트가 있다(챔피언십 시즌은 포함하지 않는다).

이러한 모든 이벤트와 레이싱을 할 보트는 크레딧 리더를 사용해서 열린다. 이벤트에서 3단계 이상을 오르면 크레딧을 받고 각 아이템은 특별한 크레딧 스킴에서 열 수 있다.

우리가 너무 늦게 알아차린 한 가지 유용한 문제는 바로 모든 플레이어가 모든 이벤트를 좋아하지 않는다는 것이다. 일부는 레이싱만 좋아하고, 일부는 Gauntlet을 좋아하지만 Ring Master는 싫어한다. 또 다른 사람은 이와 반대이다. 불행하게도 게임의 선형 언락(linear unlock) 시스템은 당신이 무엇을 좋아하는지 상관하지 않고 모든 것을 플레이하도록 한다.

또 다른 문제는 난이도 범위가 너무 광범위하다는 것이다. 기초적인 이벤트는 하드코어 레이싱 팬들에게는 너무 쉽다. 반면에, 일반적인 플레이어(와 약간 하드코어적인 플레이어)는 종종 전문가(Expert) 수준에서의 살인적인 난이도에 당황했다.

그래서 게임에는 모든 플레이어를 위한 재미있는 경험이 많이 있지만, 모든 이벤트와 난이도가 모든 플레이어에게 재미있기만 한 것은 아니다. 그리고, 게임은 모든 플레이어가 모든 이벤트와 난이도를 경험하도록 몰고 간다.

그때 사정으로는 모든 게임 구조에 대해 무엇인가를 하기에는 실질적으로 너무 늦었다. 우리가 처음부터 끝까지 게임을 하기 위해 내부적으로 충분한 시간을 마련해 놓고 있었다면 또는 좀 더 많은 긴 포맷의 외부 유용성 테스트에 좀 더 많은 투자를 했었다면 우리가 이러한 문제를 통보하고 해결할 시간을 확보해서 좀 더 강력한 경험을 얻었을 것이라고 생각한다.

5. DLC 예견

Microsoft Certification 테스트를 통과하기 위해 우리는 대표적인 다운로드가능 콘텐츠(DLC)에 대한 "플레이스홀더 자산"을 포함해야 했고, 우리는 글자 그대로 이를 제공했다. 불행하게도 우리는 우리의 첫 확장판(2010년 10월 출시된"Tempest Pack")에 포함하기로 결정한 완전한 기능 세트를 예견하지 않았다.

수 많은 자세한 사항(예를 들면, 자세하게 어떠한 Achievements를 얻는지 또는 어떻게 새로운 보트와 이벤트를 열 것인지)이 DLC 개발 과정 도중 기본적으로 발생하거나 변경되었다. 우리가 원하는 DLC를 얻기 위해 우리는 타이틀 업데이트(TU)로 기본 게임을 패치해야 한다는 것을 깨달았다. 불행히도 각 TU는 cert를 반드시 거쳐야 한다. 즉, 이들은 공짜가 아니며 작은 개발업체는 TU의 수가 최소로 되게 해야 한다.

우리는 버그 수정 번들을 통해 하나의 TU를 발행했고 기본 게임 출시 후 약 1달/DLC 콘텐츠가 완전하게 선보이기 전 1달인 8월에는 DLC 기능을 기대했다. 그렇다 하더라도 TU가 출시된 후 추가하는 것에 대해 우리가 생각한 새로운 기능들이 있었다. 불행히도 이러한 점에서 우리는 우리가 가진 것을 사용해서 작업해야 했다.

기본 게임 개발과 동시에 DLC를 창조하기 위해 이것이 훨씬 더 효과적이었을 것이다. 이는 필요한 대로 기능을 자유로이 추가하거나 도입할 수 있고 완전한 패키지를 함께 시험할 수 있고 단일 인증 위해 이들을 제안할 수 있도록 했을 것이다. 게임이 출시되고 출시일을 전략적으로 맞출 때 DLC를 보유하는 추가적인 혜택과 말이다.

물론, 이러한 문제는 우리에게 전적으로 달려있지 않다. DLC에 대한 최종적인 결정은 기본

게임용 콘텐츠 컴플리트(Content Complete)가 히트한 바로 그 즈음인 2010년 2월에서야 내려졌다. 다음 번엔 이것이 향후 프로젝트에 대해 추진해야 하는 무엇인가라고 알고 있다.



결론

당신이 1~2년의 경험이 있는 게임 개발자라면 당신은 아마도 마음속에 “나만의 게임을 만든다면 어떻게 다른 게임과 차별화를 할 것인가”란 주제를 가지고 프로젝트에서 해야 할 것과 하지 말아야 할 것을 나열한 긴 목록으로 가지고 있을 것이다.

나의 경우 하이드로 썬더 허리케인은 이러한 목록을 현실로 이를 기회가 있었다. 이 게임을 만들면서 가장 흐뭇했던 시간은 대부분의 경우 우리의 목록이 괜찮다는 것을 느꼈을 때였다. 우리는 보유하길 항상 원했던 툴 세트를 작성하고, 유능한 계약자로 구성된 작고 유대가 긴밀한 팀과 함께 일하며, 좋은 평판을 받은 속편을 아카이드 레이싱 역사에서 가장 사랑을 받은 프랜차이즈 중 하나로 선택했다. 우리는 이를 제때 예산에 맞게 제공했다. 또한, **Summer of Arcade**로 뽑히기 까지 했다.

이 프로세스에서는 문제가 없었다. 이전 프로젝트 경험에서는 메워야 했던 허점, 완전하게 잘못될 것으로 판명된 가정, 회피하고 넘어지는 수많은 작고 놀라운 일들과 위험한 일들이 있었다.

우리의 목록에 있는 항목 중 많은 부분은 다시 작성해야 했고 다시 말이 나오지 않도록 두꺼운 아주 두꺼운 유성 마커로 줄을 그어 놓아야 했다.

그러나, 결국 우리는 당신이 시도를 하지 않는다면 당신은 배울 수 없다. 그렇지 않은가?

다음 목록은 진행중인 작업이다.

데이터 박스

개발업체: 벡터 유닛(Vector Unit)

퍼블리셔: Microsoft Game Studios

출시일: 2010년 7월 28일

플랫폼: Xbox LIVE Arcade

개발 기간: 14달(초기 프로토타입 기간은 제외)

내부 팀 규모(최대): 7

인월(Man month), 코드: 26

인월(Man month), 컨텐츠: 45

예산: 마린의 집 한 채 값 정도

코드 라인: 170K

개발 툴: Autodesk Maya 2008, Adobe Photoshop, Crazy Bump, TortoiseSVN, Dev Studio 2008

타사 **SDK**: FMOD, Bullet