



포스트 모템: 벡터유닛의 Riptide GP

등록자 : 매트 스몰 Matt Small
가마수트라 등록일 : 2011년 9월 8일

안드로이드 상에서도 콘솔 같은 경험을 제공하기 위해 무엇이 필요할까? XBLA제목 “Hydro Thunder Hurricane” 에서 Riptide GP까지 벡터 유닛은 이동하였는데, 이 글에서는 구글 모바일 플랫폼을 위한 강력한 수중 레이싱 게임 창조한 이야기를 공유하고자 한다.

2010년 6월에, 벡터 유닛에 있는 한 팀은 조심스럽게 낙관적인 전망을 하고 있었다. 첫 게임으로 개발을 막 끝낸 Hydro Thunder Hurricane은 시간과 예산에 잘 맞춰 되어졌고, XBLA의 명성 있는 아케이드 슬롯(Arcade slots)중의 하나를 착수하기 위해 준비하고 있었다. 초기의 리뷰는 좋았으며, 판매 예상은 연말이전에 좋은 성과가 있을 것으로 보았다.

우리는 여전히 Hurricane을 위한 Tempest Pack DLC 을 통해서 들어오는 작은 수입금이 있었다. 그 수입은 충분히 나를 유지시켜주었고, 여름 내내 바쁘게 미술 작업하는 사람들과 연락하며 지낼 수 있게 해주었다. 엔지니어링 작업이 많이 필요하지 않았기 때문에, 나의 공동창업자인 Ralf Knoesel와 리드 프로그래머는 Dead Space 2에서 작업하기 위해 EA Visceral Studios에 연락을 취하였다. 그 동시에 우리는 새로운 프로젝트에 마이크로소프트와 다른 퍼블리셔들을 지속적으로 참여시키려고 하였다.

우리는 누군가로부터 얻은 아이디어나 새로운 컨셉이 있었고, 그 중에 어떤 것들은 굉장히 신선한 아이디어였으며, 우리는 막힘이 없었다. 한 가지 중요한 점은, 퍼블리셔들은 우리를 “레이싱 게임 스튜디오” 로 여겼고, 레이싱 장르에서는 우리는 최근 높은 위치를 갖게 되었고, 또한 새로운 레이싱 게임에 대한 큰 욕구도 없었다.

허리케인이 7월에 출시되었다. 플레이어들에게 받은 후기는 꽤 좋았으나, 판매는 우리가 기대한 만큼 잘나가지는 못했다. 그것으로 곧 두 번째 타이틀에 자기자금(셀프펀딩)을 선택하지 못한다는 것이 분명하게 되었다. 10월에는 Tempest Pack 작업을 마쳤고, Ralf는 여전히 EA와 연락 중이었고, 우리는 여전히 두 번째 프로젝트에 대해 분명한 것이 없었다. 다음번에는 무엇을 해야 할까라는 생각이 다급하게 다가오기 시작했다.

그 후 11월초에 우리는 NVIDIA에서 놀라운 소식을 전하는 전화를 받았다. 그 칩 메이커 회사는 dual-core Tegra 2 프로세서와 함께 안드로이드 폰과 태블릿 시장으로 움직이는 중이었다. 그리고 새로운 하드웨어의 역량을 선보일 수 있는 게임을 찾고 있는 중이었다. 그 회사의 누군가 허리케인 게임을 즐기고 있었고, 우리가 그들을 위해 허리케인과 유사한 어떤 것을 만들 수 있는지 알고자 했다.

우리가 찾고 있던 그러한 거래는 아니었으나, 도전할 만한 것이었다. 우리 중에 누구도 모바일 게임을 만드는데 작업해 본 사람이 없었다. 우리는 Tegra 2 하드웨어가 실제로 어떤 것을 할 수 있는지 모르고 있었고, 그 하드웨어가 360도로 허리케인을 위해 우리가 개발했던 수중 효과를 끌어낼 수 있을지도 의문스러웠다.

예산도 빠듯해서 개발비용의 대부분을 우리 스스로 해결해야 할 것 같았고, 2월까지 첫 시연 게임하고 5월까지 게임을 마무리 하려면 스케줄의 압박도 컸다

반면에 긍정적인 측면으로는, 우리는 어느 정도 안드로이드에 우리의 엔진을 이동하는 계획이 되어있었고 특정 최신 태블릿과 휴대폰을 위한 개발되어진 첫 번째 게임 중에 하나를 만드는 기회를 가지고 있다는 것이다. 사실 어려운 일이지만 재미있게 들렸다. 그리고 일은 일인 것 뿐 이다.

그래서 12월 내내 우리는 시간을 조율했고, NVIDIA는 우리의 자금문제를 도와주어 우리가 자금을 해결하도록 도와주었다. 무엇이 다가오는지 알기 때문에, 우리는 우리스스로가 휴가 기간에도 짧게 쉬고 2011년 1월을 시작하였고, 우리의 두 번째 게임인 Riptide GP라고 불리는 futuristic jet ski racer(미래적인 제트스키 경주자)를 착수하게 되었다.



An early shot from Alpha City, right after the MWC demo in March.

잘된 점들

1. Hurricane을 기반으로 일어서기

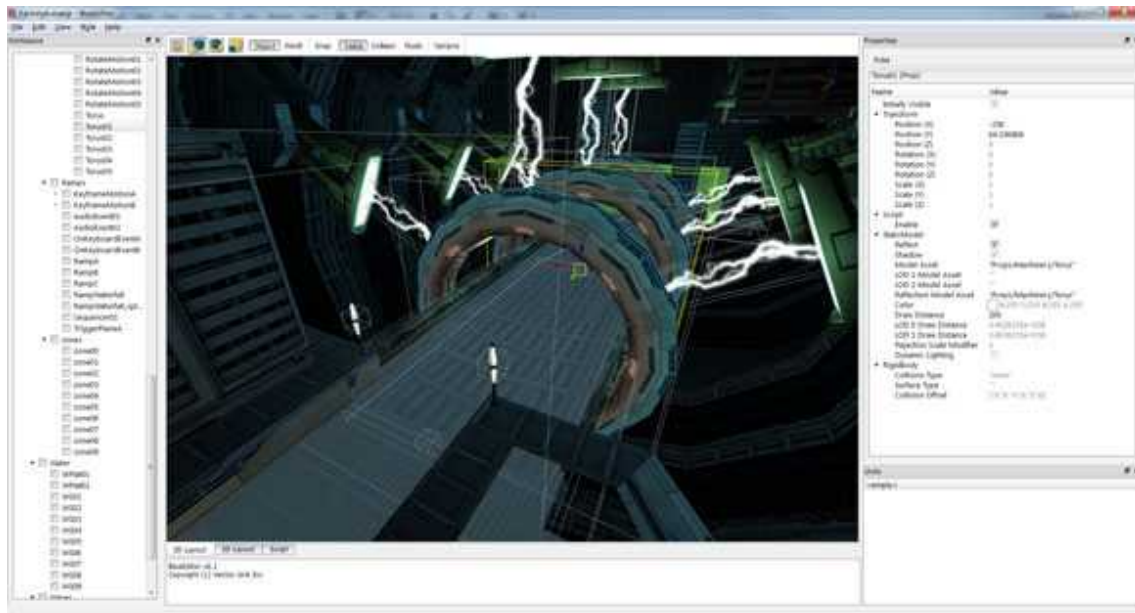
Hydro Thunder Hurricane가 없이는 간단하게 Riptide GP를 만드는 것이 우리가 가진 시간으로는 가능하게 보이지 않았다.

벡터 유닛은 허리케인을 개발했던 모든 기술을 보유하고 있었다. 그래서 1월초에 Riptide GP를 작업을 시작했을 때, Ralf는 안드로이드 NDK로 낮은 수준의 것들은 이동시키는 작업을 하는 동안에, 우리는 거의 전체 모습을 만드는 것이 가능했고, 크로스 플랫폼 엔진(cross-platform engine)은 PC상에서 개발내용을 시작할 수 있게 하였다.

내가 우리의 첫 번째 버티칼 슬라이드의 원형(프로토타입)을 시작하였을 때, Ralf는 재빨리

Tegra 2의 CPU가 가득찬 것을 발견해냈는데, 그가 허리케인에서 주로 해왔던 calc-intensive fluid 다이나믹 시뮬레이션을 처리하는 것 때문에 가득 찼던 것이었다.

그는 거의 360도를 약간 수정하거나 거의 수정하지 않으면서 모든 물, 파도, Bullet 물질 코드를 옮길 수 있었다. 이것은 코드에 도전하는 것이 아닐 수 없다. (자세한 것은 잘못된 점에서 읽을 것) 그러나, 일반적으로 말해서, 크로스 플랫폼 기반작업은 허리케인 상에 해 놓았고, 안드로이드로 이동은 비교적 순조롭게 되어졌다.



Major environment components were built and assembled in Maya, and then exported and brought into the Vector Engine game editor. We then add in PFX, animated objects, AI pathing, and water surfaces (click for full size).

또한 우리는 수중 경주게임 디자인의 2년 경험이 있다는 이점이 있었고, 우리는 그 2년 동안 실수를 통해 많이 배웠다. 우리는 몇 가지의 우리가 할 수 있는 것과 할 수 없는 일이 있다는 것을 알고 있는 그 지점에서 시작하였고, 또 몇 가지는 변화시켜보기를 원하였다.

예를 들면, 우리는 완전히 허리케인에 워터 심(water sim)에 결코 만족하지 않았다. 사실 그것은 매우 실제적이었다.100mph 정도의 빠른 속도의 실제 경주를 보았을 때, 보트는 물속에 잠기지 않는다. 물이 마치 아스팔트 같을 것이다. 그러나 게임에서는, 우리는 부드러운 느낌을 원하고 있다.

허리케인에서, 우리는 실제적인 워터 심으로 시작하였고, 후방 엔진의 종류를 사용하여 보트가 물속으로 잠기는 것을 부드럽게 하였다. Riptide GP를 위해, Ralf는 기초적인 시뮬레이션에 들어갔고 근본적인 부력과 저항력 계산을 좀 더 기분 좋고 부드러운 느낌을 주기위하여 수정하였다. 반면에, 여전히 실제적인 특성들을 지속적으로 유지할 수 있도록 하였다.

만약 우리가 처음부터 시작했었다면, 우리는 게임을 만들기 위해 최소 5개월 동안은 결코 발견이나 반복 같은 것을 할 수 있는 시간을 가질 수 없었을 것이다. 우리가 잘했던 못했던 이미 작업해 놓았던 허리케인이 있었기에, 수중 아케이드 레이스에서, Riptide GP에 그 경험들을 모두 적용시킬 수 있었다.

2. 디자인 규모

우리는 Riptide GP를 네 가지의 직접적인 기본적 디자인 목표를 가지고 시작하였다. 무엇보다도 첫째로, Tegra 2를 위한 쇼케이스에 보일 것으로써, 기술적으로나 시각적으로 인상적이어야만 한다. 둘째로, 모바일 플랫폼에서 사용될 것을 목표로 해서, 접근이 용이하고 쉽게 구할 수 있는 게임 메커니즘이고 짧게는 2-3분의 플레이 세션을 가져야만 한다.

셋째로, 우리는 “프리미엄” 가격을 정의내릴 수 있는 수준의 가치를 충분히 보여줄 게임을 원했다. 무료(Freemium) 또는 99센트짜리 앱 모델에 견주는 것이어서는 안 된다. 그러나 우리는 높은 수입 모델을 지지하는 시장에 이러한 새로운 장치는 충분하지 않다는 것은 예견하고 있었다. 그리고 마지막으로, 물론 우리는 단 1달 반 만에 두 사람으로 그 게임을 만들어 낼 수 있었다. 그것은 기본적으로 우리가 떠오는 플랫폼 상에 첫 번째 타이틀 중에 하나가 되기를 원했기 때문에 가능하였다.

우리의 두 번째 목표가 바로 간단함, 간결함이었기에 촉박한 시간에 잘 맞아 떨어졌다. 그러나 비주얼과 규모에 콘솔 품질을 만들어 위해서는 시간이 굉장히 도전적이었다. 충분한 내용을 창조해내기 위해, 환경에 있어서 모든 것을 모듈식으로 만들었고 기하학과 구성을 다른 용도에 맞게 고쳐서 어디에서든 가능하게끔 하였다.

이러한 재사용은 패키지를 작게 만드는 이점을 더할 수 있게 해주었다. 결국 우리는 구글의 50mb 제한 하에서 추가적 데이터 다운로드 없어도 되게 하였고, 로드(Load)시간은 로딩 바 (loading bar)가 필요가 없을 정도로 빠르게 되었다.

우리의 초기 디자인 스펙과 전체적 개발에 있어서, 우리의 핵심 목표에서 벗어나는 요소들은 가차 없이 버리려고 하였다. 그리고 다른 바람직한 특징들인 온라인 멀티 플레이어나 소비자용 특성은 기초적인 수준으로 남겨두었다.

핵심적인 게임플레이에 집중하려고 목표를 줄여서 집중하다 보니 일은 빠르게 진행되었다. 2달간의 개발이후에, 게임은 대강 나왔고 컨트롤하기에 아주 훌륭하게 느껴졌다. 그러나 전반적으로 다소 얇은 수준이었고, 초기에 삭제시켰던 메커니즘은 재도입하기로 결정하게 되었다. 우리는 스크린에서 thumb-swipes의 다른 종류로써 실행을 놀랍게 해낼 수 있는 추가적 부스트를 플레이어가 얻을 수 있게 추가시켰다. 바로 게임이 필요로 하는 엑스트라 킥 (extra kick: ahem.)을 놀랍게 제공하였다. 제스처를 재미있게 만들었고, 게임에서 부족한 위험에 보상하는 긴장감을 제공하는 것도 성공적으로 되었다.

3. 쉽게 하기

콘솔 타이틀을 수년간 작업한 이후로, Riptide GP는 모바일 게임으로서 우리의 첫 번째 디자인 경험이었다. 우리는 너무 쉬운 게임이 되길 원치 않으면서도, 한편으로는 게임은 난도와 컨트롤 메커니즘의 양쪽 조건에서 볼 때, 캐주얼 플레이 스타일에 더욱 적용될 수 있어야 한다고 느꼈다.

Hydro Thunder Hurricane 같은 콘솔에서의 레이싱 게임은 기술적으로 매우 어렵고 도전적인 경향이 있다. 그러나 자동차의 조작과 물체 사용능력을 마스터를 하여, 속도를 주의해서 유지하며, 코너 각도와 트랙의 넓이의 다양한 경주라인들을 완벽하게 이해하게 되면, 충분한 재미와 보상이 주어진다. 한편으로는, 모바일 레이싱 게임의 관례는 가속도계 조절과 전속력 자동속도계로써 플레이어 스스로 완벽하게 되어 질 수 없다.

우리는 그 점에 대항하고자 하기보다는 오히려 그 점을 포용하려 하였다. 우리가 개발한 차량의 스펙과 트랙의 레이아웃은 모두 매우 빠른 스피드로 몰수 있도록 되어있고, 플레이어가 코너를 잘못 돌았을 경우에 브레이크를 오로지 사용할 수 있도록 되어있다. 우리는 게임

이 빠르게 흘러가길 원했다. 대신 플레이어가 막길 수 있는 불특정하거나 어려운 코너들이 거의 없고 속도를 줄이지만 앞으로 계속 갈수 있도록 해놓았다.

결과적으로, Riptide GP는 매우 쉽게 이해해서 플레이 할 수 있게 만들어졌다. 대부분의 트랙들에서, 거의 사람들은 플레이 그 자체를 하도록 되어있고 마지막에 끝나게 되어있다. 게임에서 파도와 다른 라이더들과 상호작용을 하면서 코너를 돌 때 속도를 유지하는 방법을 배우는 것과 안전하게 스텐트 동작을 하는 것을 아는 것이 가장 높은 수준이며 도전적인 과제가 된다.

처음에는, 게임이 너무 쉽다고 생각 했었다. 그러나 친구들과 함께 테스트로 플레이를 해보았더니 어려웠고 다른 사람들도 우리 도움이 있어야 플레이를 할 수 있었다. 그리고 3가지 다른 수준의 레벨이 주어지면, 한 사람을 좌절시키거나 다른 사람을 괴롭히는 것 없이, 대충 쉽게 하거나 아주 어렵게 하는 게임인 두 가지 스타일 모두를 수용할 수 있다는 것을 알게 되었다.



A previs concept rendered in the *Hydro Thunder* engine on the 360, before we ported to Android. We had to lose normal maps and fogging, but otherwise managed to come pretty close to our visual target.

4. 첫째

Tegra 2 프로젝트에 뛰어들어 승리를 증명하는 것은 큰 위험을 감당해야 하는 것이었다.

처음 생산을 계획을 했을 때는 나는 해낼만한 목표를 세웠다. 하드웨어와 소프트웨어 업데이트를 하는데 좋은 환경을 제공하는 iOS상에서와 같지 않게, OS의 붕괴는 안드로이드 개발자들에게는 심각한 난관이었다.

Tegra에 집중함으로써, 우리는 비교적 기술적 수준과 꾸준한 성과를 끌어낼 수 있었다. 그리고 이러한 장치들이 새로웠기 때문에, 우리는 오로지 안드로이드 OS의 몇 가지 다른 버전에 대한 걱정을 해야만 했었다. 우리는 테그라를 하는데 스케줄상 시간이 정말 없었기 때문

에, 호환성 테스트를 위한 필요조건들 아주 간단히 해야 하는 것이 좋았다.

추가적으로, 우리는 배타적으로 그 플랫폼을 목표로 하고 있었기 때문에, NVIDIA 는 크게 도움이 되었다. 회사는 개발 하드웨어와 테스트하기 위해 곧 발매될 장치들까지 제공하였다. 그 회사의 기술팀은 우리가 만든 것을 평가하고 긍정적 피드백을 제공하며 도와주었다. 그리고 마케팅 팀은 우리를 새로운 장치를 발매할 제작자와 많은 회사에 소개시켜주었다. 그러한 도움으로 최대한 우리를 시장에 필요한 만큼 노출시킬 수 있었다.

5. 공정함

여기서 언급해야 할 이야기가 있다. 전통적인 게임의 재정적 모델은 퍼블리셔와 로열티로 되어 있다는 점이다.

Riptide GP을 위한 최선의 길은 전통적 시스템을 이용하여 관리하는 것이고 게임 자금을 우리가 가진 현금과 제 3자의 투자와 함께 섞어 자금을 마련하는 것이었다. 내가 서두에서 언급했던 것처럼, NVIDIA은 이러한 점으로 도움이 되었다. 그 회사는 우리에게 작은 스튜디오들에게 자금 개발로 도움을 주는 Flashman Capital이라는 샌프란시스코에 있는 회사를 소개시켜주었다. Flashman은 우리가 필요한 현금의 일부를 결제해 주었고, 총수입을 나누기로 하였다. 또한 우리는 또 다른 투자자를 찾아다녔고 우리에게 발생한 적자를 해결하였다.

여기에는 두 가지 유리한 점이 있었다. 첫 번째로, 우리는 IP를 지켜야 했고, 우리는 게임을 우리가 원하는 다른 플랫폼 어디에든 옮길 수 있었다. 두 번째로, 우리에게 더 좋았던 것은, 우리와 다른 투자자들이 게임 이득을 모두 공유하기로 했다는 것이다.

전통적 퍼블리셔 모델에서는, 개발자들은 퍼블리셔가 초기 투자를 개발자의 로열티 공유로부터 되돌려 받기까지 한 푼도 바라볼 수 없었다. 이전까지는, 퍼블리셔들은 여러 차례 그들의 초기 투자를 되돌려 받으려고 하였다. 총수입 공유 모델에서도, 다른 투자자들은 하루에 발생한 이익까지 공유하고, 심지어는 미래적 투자 레벨까지 공유하려고 한다.

결과적으로, 벡터 유닛은 개발 발전에 크게 투자하였다. 모든 추가적 세일즈에서 직접적 혜택을 볼 수 있으므로, 우리는 게임 산업의 성숙과 촉진에 동기 부여를 하게 되었다. 일반적으로 볼 때, 우리는 다른 경우에 비해 더욱 행복한 상황이고 전체적으로 이상할 정도로 공평한 대우를 받았다고 할 수 있다.

잘못된 점

1. 너무 큰 기대

1월 첫 주, 우리는 Vector Engine 을 Tegra2 Dev 하드웨어에서 가동시키기 위해 이동시켰고, 그것은 우리를 긴장시키는 일이었다.

2월초에 우리는 NVIDIA를 위해 엔진데모를 Mobile World Congress에서 선보이기 위해 전달

할 것에 서명을 하였다. 하드웨어의 실제 성과물로부터 기대할 수 있는 것을 우리는 알지 못하지만, 우리는 높은 기대를 가지고 있었다. 우리 모두 서류상 스펙을 작성해야 했고 NVIDIA로부터 값비싼 청구와 Tegra 2가 초기 모바일에서 콘솔 수준의 품질을 제공할 수 있는 얼리어답터 개발자가 되어야만 했다.

새로운 하드웨어를 위한 성과를 진짜가 되기 하기 위해서는 실제 현실은 조금 더 복잡하였다. 내가 이전에 언급했던 것처럼, 안드로이드 NDK로 360/PC 크로스 플랫폼 엔진을 이동하는 것은 순조롭게 되어졌다. 둘째 주까지, Ralf는 나의 프로토타입 레벨을 올렸고 Tegra 2 dev kit상에서 가동시켰고, Tegra 2하드웨어는 기대했던 것보다 높은 성과를 내었다. 그 CPU는 놀랍게 water sim과 유체 순환을 해치웠고 Bullet physics의 이행을 순조롭게 해내어 high-poly 모델을 소화시키고 우리가 요구하는 것보다 더 잘 해내었다.

GPU에 관한 또 다른 이야기도 있다. 우리는 매우 높은 입력속도를 가졌고 특히 게임에서 도전 받게 되었다. 스크린 상에서 절반은 거대한 반투명 물로 덮이는 일이 발생하였다. 우리가 360에서 크게 의존했던 복잡한 픽셀 shader가 빠른 속도의 레이싱 게임에는 전송 프레임률을 허락해주지 않을 것을 빨리 깨닫게 되었다.

우리는 shader(셰이더)를 유선화하였고, 우리가 per-vertex 레벨의 수준까지 모든 per-pixel 프로세스를 이동시켰고, 가능한 어느 곳에서든지 낮은 정밀도 순환을 바꾸려고 하였다. 우리는 반응을 깨뜨리는데 필요하나 복잡성을 유지하기 위해 mesh density를 증가시키고 물에서 따로 나오는 정상적인 물결을 떼어내었다.

내 오래된 skool 아티스트 툴킷을 오랜만에 꺼내어, 정점 색상으로 조명을 주었고, 분산된 텍스처로 세부그림을 넣었고, 가능한 GPU 파이프 오픈으로 유지하기 위해 텍스처와 메쉬 변환을 관리하려고 하였다.

NVIDIA는 이 과정에서 성과분석을 하는데 큰 도움을 주었고, 결국 우리는 처음으로 게임에 들어가는 MWC demo를 시각적 품질을 제공하며 만들어 낼 수 있었다. 최소한 보통 사람 눈에 는 그렇게 보였을 것이고, 콘솔이나 최신 PC상에서 볼때 좋게 보일 수 있는 정도였다.

모바일 플랫폼 상에서 기술적 진보의 비율은 사실 더디었고, 나는 지식이 있는 게이머들이 수년 안에 인게임 screenshot이 높은 사양의 콘솔이나 전화기에 만들어질 수 있을지 대답해 달라고 압박할 것이라 여겨진다. 모델링, 애니메이션, 그리고 셰이더(shader)조절은 2008년 Maya에서 되어졌다.

2. 창의적인 재사용 혹은 그것의 부족함

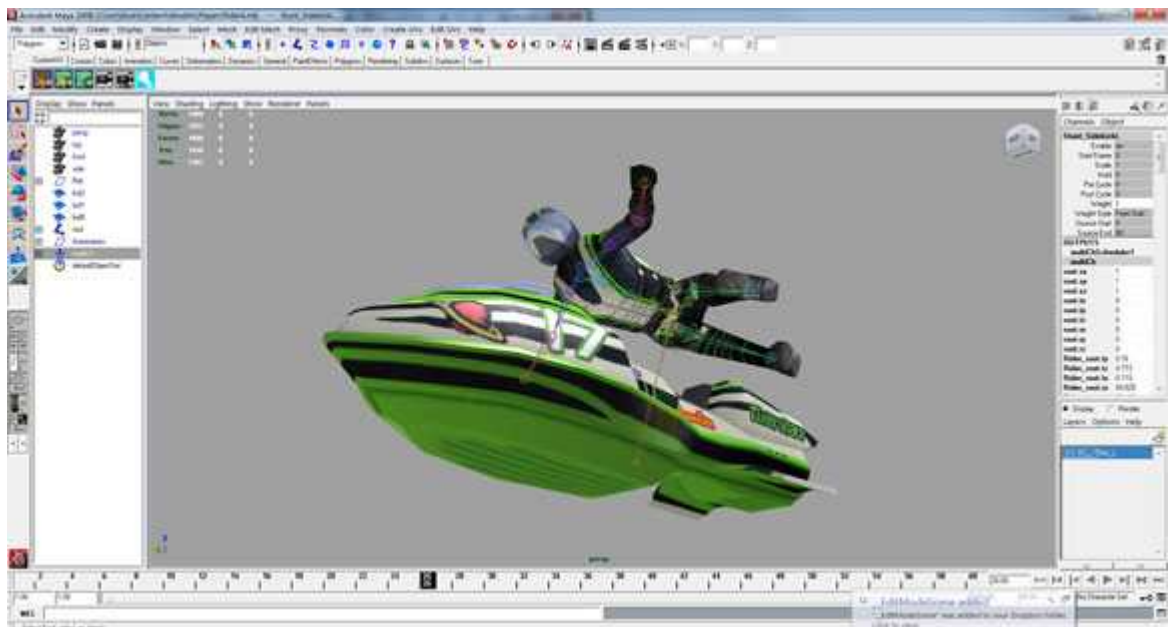
Modeling, animation, and shader adjustment was all done in Maya 2008 (click for full size).

Riptide GP의 개발에 대해 만들어진 내용물에 대해 비평을 받을 것이라는 것은 처음부터 알고 있었다. 그런 전면적인 코드와 달리, 우리는 허리케인에 있는 다른 내용물들, 즉 모델, 텍스처 라이브러리 등을 재사용할 수 없어서 처음부터 새롭게 시작해야만 했다.

몇 달 정도의 스케줄이 처음으로 지나고서, 나는 겁이 났다. 4주 동안 제트 스키, 캐릭터, 애니메이션을 만들어야 했고, 10주 동안 환경을 제작했고, 5주 동안 나머지 것들을 해야 했다. UI, 세부 효과, 극대화 등을 포함하여 해야 할 것이 기다리고 있었다.

환경을 해내는 스케줄이 가장 힘들었다. 평균 6-7일에 6개의 주요 레이스 트랙을 각각 끝마쳐야 했고 그 후 하루 이틀 동안 내가 원하는 최소한의 독특한 빛과 물의 특징을 주는 변화를 만드는데 시간이 들었다.

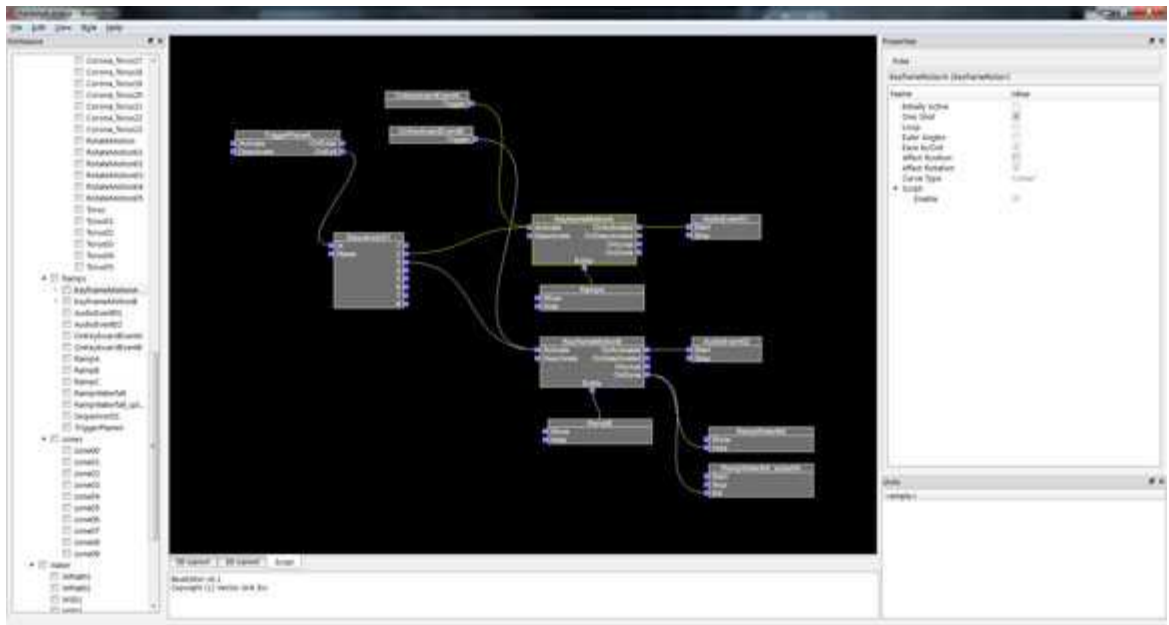
오로지 성실한 재사용과 사례들을 사용하는 것을 통해서만 가능할 것 같았다. 나는 어느 정도 해낼 수 있을 정도의 일이라 여겼고, 트랙을 만드는데 texture와 geometry를 재사용 혹은 재활용할 기회를 찾고 있었다. 그러나 내가 각각의 트랙을 다르게 만들고자 하는 시도와 바람은 포기되어야 했고, 내가 가진 많은 자료들을 최대한 재사용할 수 없었다.



결국, 나는 옛날 방식을 사용해야 했고, 열심히 오랜 시간을 들여서 마쳐야 했다. 전체적으로 보면 그것은 창의적인 작업이고 앞서 언급되었던 우리가 품격 있는 게임을 제공하는데 에너지와 투자를 해야 하는 것이었다. 나는 5월에 안드로이드 마켓에서 Riptide GP 를

발매하기위해 그전에 연마하고 세부조정을 하였다. 그러나 내가 게시할 때가 돼서야 안심하게 되었다.

모든 게임 스크립팅은 비주얼 스크립트 요소를 사용함으로 되어진다. 이러한 과정은 "Grindhouse"에서 애니메이션 램프(ramp)로 조절할 수 있다.



All game scripting is built up using visual scripting components. This sequence controls the animated ramps in "Grindhouse" (click for full size).

3. 작은 팀의 도전

우리는 우리 두 사람이서만 Riptide GP 를 만들어야겠다고 처음에 의도했던 것은 아니었다. 한두 달 동안의 artwork을 계약해야한다고 생각했고 사운드 디자인과 음악 같은 것도 예산에 할당해두었다.

그러나 Riptide GP의 자금을 우리 스스로 감당해야 했기 때문에, 비용을 낮추어야만 하는 동기부여가 있었다. 그래서 우리 일정을 빡빡하게 만들었고 아웃소싱을 감당할 시간적 여유가 없었다.

그래서 우리는 전체 흐름을 미리 보았고, 나의 일정은 예술부분의 일까지 담당해서 처음부터 끝까지 꼭 채워져 있었다. 그러나 Ralf가 일단 그래픽 엔진 최대화를 하고나자마자 Ralf의 프로그래밍 스케줄이 시작될 수 있었다. 그는 잡다한 수많은 콘텐츠 업무를 해야만 하였고, 그 일에는 Audiomicro와 같은 온라인 사이트에서 흐르는 사운드 디자인과 음악 소스, 스크린에 흐르는 자막, 세부 효과 등을 포함하고 있었다.

작은 팀에는 항상 문제가 존재한다. 우리 경우에는, 아주 작은 팀이었기 때문에 실제적인 보이지 않는 변수로 인한 들쭉 날쭉한 문제의 소지가 없었다. 프로젝트를 시작한 시점부터

전력질주를 하였고, 우선순위가 정해진 리스트대로 우리가 가진 시간 내에서 하기 위해 애썼다. 열심히 한 만큼, 그것은 창조적이고 보상받는 작업이었다. 그리고 어떻게든 우리는 시간 내에 모든 일을 마칠 수 있었다. 그러나 과정은 매우 어려웠다.

4. Yarr!

모바일에서 복제는 누구나 언급하지 싫어하는 문제점이다. 플랫폼을 가진 사람에게는 당황스러운 일이며, 개발자들이 가능한 홍보를 적게 해야 하는 것이 최선이라고 느낀다. 그러나 진짜 문제는, 앱 구매를 통하지 않고 엄격하게 기본적 게임 판매로 자금을 얻는 Riptide GP 같은 게임이 특히 문제가 된다고 생각한다.

우리는 구글의 충고를 받아들여 안드로이드 마켓 라이선스를 확인(license-checking)을 이행하였다. Ralf는 최소한 자동화된 부트(bot)를 전략적으로 작동시킬 방법을 모색하면서, 어디에서든 가능한 라이선스 코드는 남이 보지 못하게 감추었다. 만약 해커들이 정말 해킹해볼 거리를 찾는다면, 최소한 라이선스 코드를 우선 해결해야만 한다는 것을 우리는 알고 있었다.

사실 우리는 그렇게 열심히 일할 필요가 있었던 것은 아니었다. 게임이 처음 출시되었을 때, 자동적 라이선스 스트리핑 (automatic license stripping)은 작동하지 않고 있는 warez forums에 올라온 것을 보는 것만으로도 신이 나있었다. 그러나 24시간 안에, Riptide GP는 플레이를 하는데 있어 금이 난 버전(cracked version)으로 나타났다.

우리의 분석에 따르면, 전 세계적으로 볼 때 불법 복제와 합법적 복제의 비율을 9 대 1에 이른다. 우리의 추정으로는, 아마 우리의 희망이겠지만, 해적들의 대다수들은 게임을 사지 않을 것이다. 그래서 그 숫자들은 판매 손실로 해석할 수 없을 것이다.

그럼에도 불구하고 좀 짜증스러운 일은, 게임을 법적으로 사는 플레이어들에게 구글의 라이선스 확인(license check)은 다소 까다롭고 세심한 주의가 필요하다. 특히 유저가 모바일 인터넷 연결을 가지고 있을 때 그러하다.

우리는 이메일로 기술적 지원을 많이 받지 않는다. 나는 그 게임 자체를 말하는 것이 기쁘고 아주 안정적이라 생각하지만, 우리가 받은 이메일의 대부분은 라인선스 확인에서 실패해서 해결해달라는 것이었다. 사람들은 간단한 형식의 편지와 함께 문제해결의 해답을 얻을 수 있었으나, 그것은 시간이 소요되는 일이고 짜증나는 일이다. 그리고 우리에게 이렇게 라이선스 확인을 해야 할 정도의 가치 있는 것인지 의심을 가게 만든다.

5. 사랑은 어디에 있나?

XBLA상에서 Hydro Thunder Hurricane 가 나왔을 때, 우리는 그것을 알리는 언론홍보 활동을 대대적으로 하였다. 프리뷰를 위한 화면이나 비디오를 발매하여 보내고, 리뷰를 위한 리뷰 코드도 보내었다. 물론 우리는 Hydro Thunder가 프랜차이즈로 알려져 있다는 사실로 크게 도움을 받았다. 그러나 우리가 볼 수 있었던 것은 오리지널 인디 게임을 위해 콘솔과 마찬가지로 같은 전략을 세우고 그만큼 일을 해야 했다.

모바일 게임은 다르게, 특히 안드로이드 게임용을 위해 특히 광고 면을 얻어내기가 어려웠다. 심지어 Tegra-exclusive 게임을 위해서는 더욱 어려움을 겪어야 했다. 거의 몇 안 되는 주요 게임 사이트에서 모바일 보도(coverage)를 헌신적으로 제공하였고, 그렇게 한 것이 가장 큰 아이폰/아이패드 타이틀에 보도를 할 수 있었다. 메타 비평(Metacritic) 상에서는 당신은 아마 안드로이드 게임 리스트를 찾을 수 없을 것이다. 그리고 주류 게임 과정의 밖에서, CNet과 Wired 같은 주요 기술의 배출구는 주로 하드웨어와 게임 앱이 아닌 것들에

주로 집중하고 있다.

Riptide GP를 위해서는, 우리는 가능한 큰 파장을 일으키기 위해 Flashman Agency 마케팅 회사와 함께 파트너로 일하였다. 우리는 NVIDIA와 그 외 carrier/hardware 파트너사들과 함께 그 게임을 홍보하는 공개행사를 하였고 MWC, GDC, E3와 같은 무역전시장에서도 선보였다. 우리는 거의 모든 주요한 게임과 기술 출구에 우리의 보도 자료, 예고편, 스크린샷 및 기타 자료들 보내었다.

우리가 Riptide GP를 세상에 내놓기 위한 노력에 비한다면, 그 모든 것이 공평했다고 할 수 없을 것이다. Joystiq는 우리를 도와주었는데, 많은 작은 사이트에서부터 우리가 중간 크기의 안드로이드와 모바일 게임 사이트와 블로그까지 우리는 엄청난 거래량과 칭찬을 받게 되었다. 그러나 주요 게임사이트와 기술 사이트들은 쉽게 흥미를 갖지는 않았다.

그러한 사이트들에게는, 우리의 도전은 사회기반시설 중의 하나가 되었다. 안드로이드 마켓은 코드 메커니즘에 관해 어떠한 리뷰도 해주지 않기 때문에 언론으로 하여금 게임 복사본을 보내서 리뷰를 하게 만드는 것은 쉬운 일이 아니다. 또 하나의 추가적 도전이라면 사람들이 게임할 수 있도록 Tegra 하드웨어를 허용하는 사이트가 많지 않다는 것이다. 결국 우리는 게임을 여러 개의 핸드폰과 태블릿에 미리 설치하여 언론사에 스스로 보낼 수 밖에 없었다.

나는 주요 사이트들이 관심을 갖지 않는 것은 플랫폼 마켓을 부분적으로 공유하고 있어서 일 것이라 생각한다. 그러나 그것 또한 PC나 콘솔의 진짜 게임과 같지 않게, 모바일 게임은 여전히 알파하고 쓰다버릴 수준이라고 생각하는 주류 게임 언론의 일부에 대한 편견에서 나온 것일 수 있다고 생각한다.

그러나 최첨단 모바일 시장은 콘솔 시장보다 더욱 빠르게 성장하고 있고, 전통적인 게임 플랫폼에서 찾던 것에 견줄만한 게임경험을 제공하는 모바일 게임의 새로운 세대가 존재하고 있다. 소비자들도 이미 이것을 알고 있다. 여기에 던져지는 질문은 바로 언제 게임 프레스가 따라잡을 것인가이다. 마지막 게임으로부터의 한방. 그 새로운 빌보드 시스템은 우리가 새로운 자연적 트랙을 산업적인 측면과 함께 혼합할 수 있게 해주었다.

에필로그

Riptide GP는 안드로이드 마켓에서 5월 20일에 발매되었다. 다른 Tegra-exclusive 게임에 비교해 볼 때 판매실적은 꽤 좋은 편이다. 더불어 유저들은 좋은 리뷰와 코멘트로 우리를 격려해주고 있다. 그리고 게임은 유저 스코어를 평균 4.8/5 정도 받았다. iOS 버전은 작업 중이며, 우리는 다른 플랫폼에서도 사용할 수 있도록 하는 것을 고려하고 있다. 추가적으로, 항상 다운로드 받을 수 있는 내용과 다른 파생물의 가능성은 늘 존재한다.

가장 중요한 것은, 이러한 경험은 우리에게 새로운 거대한 시장에서 성공을 맛보게 해주었다는 것이다. 우리는 항상 작은 팀을 유지하여 일하고, 콘솔에서 모바일까지 관심을 이동시키는 것을 할 수 있기를 원한다. 아마 앞으로도 수년은 그럴 것이다.

그렇게 우리가 콘솔의 발전에 노력하는 것을 포기하겠다는 말은 아니다. 우리는 일부를 혼합시키겠다는 것이다. 콘솔은 여전히 기술적 우위를 가지고 큰 안정성을 제공하는 사이즈와 예산도 증가시키고 있다. 또한 콘솔 게임머들은 여전히 우리가 모바일에서 현재하는 것보다 게임 메커니즘이 좀 더 심도가 있고 섬세하기를 요구하고 있다.



A shot from the final game. The new billboard foliage system allowed us to mix a few naturalistic tracks in with the gritty industrial ones.

여전히, 모바일 게임 시장은 우리에게 작은 스케일의 프로젝트의 기회를 제공하고 있다. 그런 게임들은 작업하기에 재미있고 머리를 식히기 좋다. 무엇보다도 가장 좋은 것은, 우리가 사실 우리 스스로 자금을 댈 수 있고 우리의 노력만큼 수확을 얻을 잠재력이 있다는 것이다.

우리가 해온 작은 스튜디오 사정을 이해해달라고 말하기가 망설여진다. 그러나 최소한 가까운 미래에 우리가 곧 안정이 될 만큼 충분한 일이 있다. 그리고 현재 안정감도 정말 좋다.