실리적인 게임 플레이테스팅: Wii 기반 사례 연구
(Practical Game Playtesting: A Wii-Based Case Study)

Gareth Griffiths

가마수트라 등록일(2009. 1. 15)

(http://www.gamasutra.com/view/feature/3894/practical_game_playtesting_a_.php)

[Sidhe의 Griffiths는 GripShift 개발자들의 플레이테스팅 방법 및 그 피드백을 기반으로, Wiimote 개조에서 난이도 조정에 이르기까지 최신 Speed Racer 게임의 Wii 버전을 향상시킬 수 있는지에 대해 깊이 있게 논의한다.]

게임을 최초로 플레이테스팅하는 일은 믿기 어려울 만큼 힘든 작업이다. 그 준비 과정에 대해서는 이야기하지 않더라도, 테스팅 과정에서 직면하게 되는 많은 부정적인 반응에 대해 이야기해 보도록 하겠다.

플레이어가 처음으로 게임을 접하게 되었을 때에는 많은 문제점들이 야기된다. 보고서를 작성하는 일에 있어서, 필자는 단조롭고 몹시 지루하여 쓰러질 것 같은 순간을 버티면서 이 정보를 개발자들에게 전해주어야 한다는 사명감으로 작업한다.

그러나 게임이 아직도 우리 손에 있는 동안 이 문제들을 결국 찾아내어 맞붙을 수 있을 때면 긴 터널의 끝이 보이기 시작한다. 일단 출시되고 나면 너무 늦은 것이며, 그 때에는 돌이킬 수 없는 결과만이 남는다.

항상 자신의 게임이 완벽할 것이라고 믿는 경향이 있는데(여기에는 필자 자신에게도 책임이 있다), 분명 그 고된 작업을 하고 나서도 무엇인가 잘못된 것이 남아있어서야 되겠는가! 그러나 불과 2 분도 채 안되어 진실은 드러나고 만다. 어떤 것은 난이도가 너무 높거나, 플레이어가 어떤 문제를 해결하는 방법을 이해하지 못한다.

여기서 유의해야 할 것은 플레이테스팅 세션 중에 여러 가지 문제들을 조명해 볼 수는 있지만, 시간이나 예산 등의 사항들은 안타깝게도 고려할 볼 수가 없다는 것이다.

본 기사는 ***Speed Racer*** *게임의 플레이테스트*에 착수하는 방법을 살펴보고, 발견한 문제점들 몇 가지를 조명해 본 뒤 이를 해결하기 위해 한 일들에 대해 논의한다. 희망적인 것은 플레이테스팅과 사용자들의 경험을 적용해 보는 과정이 왜 유익한지 쉽게 이해할 수 있다는 것이다

***Speed Racer* 에 우리가 한 일은 무엇인가? 그 다음에는 무엇을 할 것인가?**

Sidhe는 *Speed Racer* 의 Wii 및 PS2 버전에 대해 작업하였다(본 기사는 전적으로 Wii버전에 대해서만 다룰 것임).

이것은 우리의 첫 Wii 타이틀이었으며, 10 개월 개발 기간이라는 매우 촉박한 시간표를 갖고 있었다. 이는 곧 우리가 무엇을 하든, 애초부터 제대로 된 것이어야 함을 의미했다. 그러나 쉽게 상상할 수 있듯 일이 처음부터 제대로 되라는 법은 없었다. 예상했던 것들과 미처 예상 못한 것들까지 여러 가지 다양한 문제들이 제기되었다.

그러나 가까스로 제시간에 맞출 수 있었으며 매우 좋은 평을 얻었다. 사실 그것이 기반으로 한 영화 자체보다도 끊임없이 더 좋은 점수를 얻은 몇 안 되는 영화 라이선스 게임 중 하나가 되었다.

그처럼 튼실한 게임을 제작할 수 있었던 방법 중 한 가지는 실제로 게임을 하게 될 플레이어들에게 플레이테스팅을 해 봄으로써 가능하였다 (테스트 대상이 9-14 세인 경우). 일단 투입되면 세션의 모든 부면, 이를테면 게임플레이부터 플레이어의 표현들에 이르기까지, 그리고 그들이 투입된 장치를 사용되는 방식까지 모두 기록되었다. 불필요하게 강력한 방법일 수도 있으나, 모든 것이 게임을 세밀하게 조정하는 데 사용될 자원인 것이다.

플레이테스팅 세션을 진행하면서, 필자는 이를 항상 두 부분으로 나눈다. 처음 세션은 네 명의 플레이어로만 진행하여 게임의 주요 특징들을 전반적으로 탐구해 보고 플레이어가 게임에 접근하는 방식에 대한 기본 감을 잡고 이해하는 데 주력한다.

그 다음 발견된 문제점들을 한 데 묶어, 남은 세션 동안 참조해 볼 좋은 기초를 만든다. 물론 더 많은 문제점들이 발견되겠지만, 기반이 될 무엇인가가 있다는 것은 언제나 좋은 일이며, 이 과정이 아주 잘 작용한다는 것을 알게 되었다. 다음으로 이 플레이테스팅 세션을 필자의 자체 방법들과 결합시킴으로써 보고서에 첨부할 후보 솔루션들을 찾아낼 수 있다.

문제점들의 예

Speed Racer 는 자동차 전투 레이싱 게임이기는 하나, 플레이어가 사용할 수 있는 전형적인 무기는 없다는 것이 흥미로운 특징이다. 대신 차량 자체가 무기이며, Wiimote를 활용함으로써 플레이어들은 여러 "Car-Fu" 작전들을 수행해야 하였다.

이를 위해서, 우리는 제어 기능은 치밀한지, 플레이어가 복잡한 여러 수단들을 동시에 수행할 필요가 있는지 확인해야 했다. *Speed Racer* 라는 명칭 자체가 곧 극히 빠른 속도로 진행되는 게임을 의미했기 때문이다. 무엇이든 이로 인해 방해를 받을 경우 제대로 작동하지 않을 것이 뻔하였다.

Car-Fu 게임이지만 사용하지는 않는다!

우리는 플레이어들이 Car-Fu 를 집어 들고 흥분을 자아내며 사용해 주길 바랬다. 그런데 우리가 직면한 첫 번째 문제점이 바로 플레이어들이 Car-Fu 를 사용하지 않는 것이었을 때 얼마나 끔찍했을지 상상해 보라. Car-Fu 는 게임의 중요한 요소였기 때문에 플레이어들이 “이를 집어 들지 않으면” 우리에게는 큰 문제가 아닐 수 없었다.

플레이어들이 이 게임을 단순히 레이서의 관점에서 보는 것이 문제였다. 한 시간 반이 지나도록 Car-Fu를 사용할 수 있다는 것을 잊고 있는 것 같았다. 따라서 게임 중에 이용 가능하다고 상기시켜 주는 것이 한 가지 방법이었으나(여기에 또다시 가시성과 어포던스의 문제가 대두된다), 이는 플레이어의 몰입을 깰 것이었기 때문에 Car-Fu에 대해 직접적으로 지시하고 싶지는 않았다.

대신 좀 더 미묘한 방법이 필요했다. 가장 좋은 방법은 그저 Car-Fu 가 게임 내에서 어떻게 발생하는지를 보여주는 것이었다.

이는 상대방이 서로 Car-Fu 를 하게끔 만듦으로써 가능했다. 이렇게 플레이어에게 기능을 상기시켜 줌으로써 자신들도 똑같이 할 수 있음을 깨닫게 되었다. 이는 또한 플레이어가 공격을 받게 되면서 도전 정신을 유도하고, 즉각 복수하고 싶게 만드는, Car-Fu 를 수행하여 갚아 주고 싶게 만드는 좋은 방법이기도 하였다.

그 기능을 사용할 수 있음을 깨닫게 되자, 게임의 짜릿함은 급상승하였다.

Wiimote 를 정확하게 사용하지 않는다!

드라이브 게임에서 Wiimote의 사용은 우리가 직관적으로 생각해 낸 것이었다. 이 게임이 *Mario Kart 휠* 추가 장치를 지원하기는 하였으나, 이 기능이 없이도 플레이 할 수는 있었다. 우리가 미처 예측하지 못한 것은 이 장치를 플레이어가 사용하도록 탑재하는 방식이다.

플레이어에게는 Wilmote 를 보유할 방법을 보여주는 작은 애니메이션이 주어졌고, 우리는 그것이면 충분하리라 생각했다. 그러나 대다수의 플레이어들이 이 애니메이션을 탑-다운 방식으로 인식하고 픽업트럭 핸들을 사용하듯이 차를 조종하려 한 것이다.

물론 그런 식으로는 움직이지 않았고, 차량이 필요에 따라 반응하지 않고 벽에 충돌하고 말았을 때에는 틀림없이 모두 낙심하였을 것이다.

한동안 이런 상황이 지속되었고, 우리가 이미지나 애니메이션을 조정해보려고 노력했으나, 플레이어들은 Wiimote 를 다루는 방법을 결국 알아내지 못했다. 이것이 큰 문제였던 것은 게임이 미궁 속에 빠져 아무도 플레이 하는 법을 알 수 없었던 것이다.

이 문제에 대한 해결책은 사실 아주 간단했다. 사람들이 지시문을 읽는 것을 관찰한 뒤 깨달은 것은, 그들이 애니메이션과 이미지를 갖고 있지만 기준 프레임을 갖고 있지 않아 각자가 그 내용을 달리 해석한 것이다.

따라서 화면 로딩을 위해, 흑백의 단순한 텔레비전 스크린을 삽입하고, 그 앞에 Wiimote 을 두었다. 우리는 이 이미지가 Wiimote 를 다루는 기준 프레임처럼 기능하길 바랬다. 그리고 실제로 효과가 있었다! 다음 번 플레이테스트를 했을 때, 플레이어들이 Wiimote 를 집어 들고 즉시 운전을 할 수 있게 되었다.

차량 조종: 극히 불안정함

이 문제는 차량 조종에 너무 익숙해 있던 나머지 무시했던 것으로서 문제가 될 수 있으리라고는 생각하지 못했다.

플레이어가 차량제어를 하기 시작하고 나자, 조정 장치가 너무 민감하여 직선으로 가는 것처럼 간단한 일조차 매우 힘들다는 것을 알게 되었다 이 때문에 모퉁이를 제대로 돌려고 해도 항상 장애물에 부딪히고 말았다. 이는 플레이어들에게 있어서는 분명 매우 부정적으로 비춰졌을 것이다.

차량 조종장치를 개조함으로써 극적으로 게임을 향상시킬 수 있었다. 그림 1 은 전후를 비교한 그래프를 보여주고 있다.

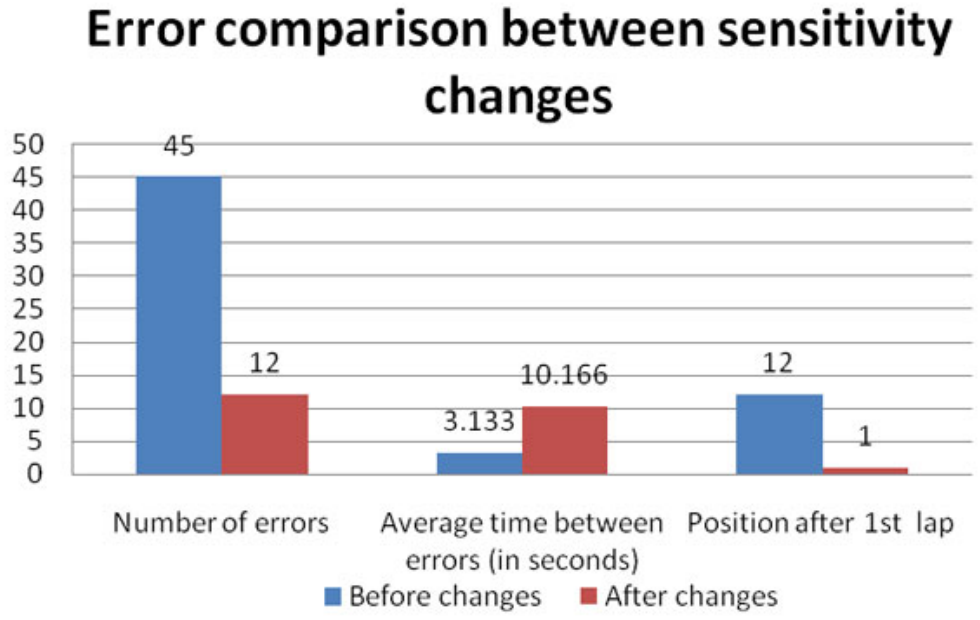

그림 1. 민감도 변화 전후의 오류 비교

그림에서 볼 수 있듯이, 오류의 수가 극적으로 감소하였으며, 오류 간 평균 시간은 증가하였다. 그에 더하여, 플레이어들은 처음 경주에서 승리하고 나서 강하고 긍정적인 피드백을 주었고, 계속 하고자 하는 의욕을 북돋워 주었다.

충돌 문제

측면 충돌 수행 시, 그림 2 에서 볼 수 있듯이 Wiimote 좌측 혹은 우측을 빠르게 충돌하는 일이 수반되었다.

그림 2. 충돌 방법의 예

이로 인해 차는 그 방향으로 약간 미끄러져 원하는 대로 차에 부딪히게 되어 있었다. 이 액션을 수행하는 데에는 문제가 없었지만, 예상했던 결과는 아니었다. 이 움직임의 정점에서, Wiimote 이 멈추는 대신, 물리 법칙이 반동을 주어 급격한 움직임이 나타났다. 이 때문에 반대 방향으로도 동일한 세기의 움직임이 발생한 것이다.

안타깝게도, 게임에서 이 기능을 수행했을 때 그림 3 에서 볼 수 있듯이 차는 플레이어가 원하는 방향의 반대로 움직였다.

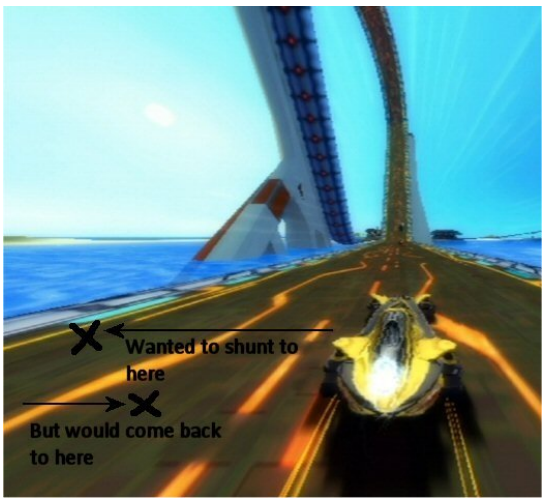

그림 3. 실제로 발생한 상황

이로 인해 플레이어들은 낙심하게 되었는데, 차가 자신이 원하는 대로 움직이지 않았기 때문이다.

이 문제에 대한 해결책은 플레이어의 액션과 소프트웨어를 조화시키는 것이었다. 플레이어는 항상 강하게 충돌하려고만 했으며, 그 절반쯤으로 충돌하는 일은 없음을 알게 되었다.

따라서 Wiimote 의 민감도를 아주 약간 낮추기로 결정했다. 이는 곧 플레이어가 충돌을 시도했을 때에, 동작이 끝난 후 후속 작용의 정도를 떨어뜨리는 결과를 가져왔다.

UI 문제

아, 공포의 UI! 확신하건데, 개발자들은 자기만의 UI 관련 문제를 갖고 있으며 여기에는 예외가 없었다. UI 는 필요할 때에 정보를 중계해 준다. 예를 들어, 사용자에게 건강 수치가 낮은지, 또는 활력 증강이 가능한 지 등을 알려준다.

그에 더하여, 차로부터 나오는 불길의 색상은 건강 상태에 따라 변화한다. 따라서 색이 짙은 적색이면 차가 기진맥진한 상태이며 폭발하기 직전임을 보여준다.

그러나 우리는 초기에만 이 기능을 사용하고, 도전 스테이지가 올라갈수록 후반에는 제거하기로 결정했다. 플레이어가 UI 사용에 더 익숙해지고 차량의 상태에 대해 더 잘 인식하게 되기를 바랐기 때문이다. 하지만 결과는 그렇지 않았다.

처음 이 결정을 반영했을 때, 플레이어의 차량이 계속 폭발하였고 그 이유도 알지 못했다. 여기서 문제는 플레이어들이 게임이 그 상태를 알려주는 데 익숙해진 나머지 더 이상 그 상태를 보지 못하자, 그것이 없이도 게임을 하는 방법을 터득한 것이 아니라, 게임을 망쳐 버리고 만 것이다.

이 문제를 해결하기 위해 맨 처음 한 일은 단순히 이 기능을 완전히 제거하여 플레이어로 하여금 UI 가 어떻게 작용하는지를 학습하게 하는 것이었다. 안타깝게도 이러한 접근법은 완전히 실패하여 플레이어들은 매번 같은 장소에서 실패하고 말았다. 결과가 좋지 않았다.

그래서 남은 한 가지 방법은 모두 원상태로 되돌려 플레이어의 건강수치가 낮거나 증진이 가능할 경우 신호를 주도록 하는 것이었다. 그런데 흥미로운 것은 플레이어들이 중요한 상태 변화에 대해 게임이 알려주는 정보에 얼마나 의존하고 있는지를 잘 알려준 사례였다.

나보다 얼마나 더 앞서 있거나 뒤쳐져 있는 거지?

Speed Racer 에서 트랙의 눈에 띄는 특징 한 가지는 매우 길다는 것이다. 여기서 문제는 플레이어가 때로는 다른 차들과 얼마나 떨어져 있는지 전혀 알 수 없다는 것과, 경주 관련 정보를 원한다는 것이다.

다른 레이싱 게임에서는 이런 종류의 기능이 필요하지 않기 때문에 흥미를 자아냈다. 플레이어가 자신의 포지션 정보를 볼 수 있으면 그것으로 보통 만족해 했다. 하지만 경주가 너무 길어지면 그 이상의 정보가 필요했다.

이 문제에 대한 해결책은 포지션 정보를 스크린 좌측에 포함시키는 것이다. 1 위와, 2 위, 플레이어 본인, 플레이어 직후의 플레이어를 볼 수 있다. 이 정보는 플레이어의 포지션과 관련되어 있고 그 시간 간격도 알 수 있기 때문에 이로써 플레이어에게 필요한 정보를 줄 수 있었다.

플레이어는 다음 플레이어와는 얼마나 떨어져 있으며, 1 위와는 얼마나 떨어져 있는지를 즉시 확인할 수가 있다. 이는 플레이어에게 도움이 될 뿐 아니라, 게임의 도전 욕구 또한 증가시켜 준다는 것을 알게 되었다. 한 마디로 윈-윈 시나리오인 것이다. 그 예가 그림 4 에 제시되어 있다.

그림 4. 레이싱 UI

토론

상기 문제들에 대해 우리가 찾아낸 해결책들은 게임 내에서 찾아낸 다른 문제들에 대해서 여러 가지 방법으로 채택 및 사용하였으며, 플레이테스팅이 시간을 들였지만 세션이 개발과 동시에 진행되어 개발 과정에 부담이 되지는 않았다. 또한 조처를 취하지 않았을 경우 게임 내에 계속 남아있을 만한 문제들 상당수를 해결하였다.

반드시 사전에 주어진 상황에서 예측되는 결과에 대해 물어보아야 한다. 당시 고심해야 할 견해 제시가 줄어들수록, 차후에 부정적인 반응은 더더욱 극심해 질 것이기 때문이다.

플레이테스팅의 최종 부분은 변경한 사항들이 실제로 효과가 있었는지를 확인하는 것이다. 이는 상대적으로 쉬운 문제이며, 다른 세션을 진행하면서도 수행할 수 있다. 문제가 또다시 제기되는지 여부를 관찰함으로써 해결 여부를 신속하게 판단할 수 있을 것이다.

지속적으로 나타나는 문제도 있겠지만, 분명 극적인 향상은 있을 것이다. 세션 내내 문제점들에 눈을 고정시켜보면 이제 변경해야 할 일은 손쉬운 몇 가지에 지나지 않음을 알게 될 것이다.

전체 게임은 수 개월 동안 계속 진화하였으며, 게임 내 세세한 조정이 가능하였다(시간이 허락하는 한). 어린이든 성인이든 누구나 일단 시작하면 즉시 차를 운전할 수 있고 모든 액션 수행이 가능할 것이다.

결론

나쁜 소식을 들고 개발자들에게 돌아가면 그들을 의기소침하게 만들 수 있다는 것을 필자도 안다. 그들도 기진맥진할 정도로 일하고 있기 때문이다. 또한 부정적인 요인들의 목록을 들고 찾아가면 누구든 그리 달갑지 않을 것임은 분명하다.

설령 그렇다 하더라도 이 방면에 있어서는 기꺼운 마음으로 자유로운 의사소통을 나누는 것이 필수적이다.

개발자들은 자신들이 비평을 받고 있는 것이 아니라 자신들이 하고 있는 위대한 일을 향상시킬 수 있는 정보가 주어지는 것임을 이해해야 한다. 운이 좋게도 Sidhe 에서 일하는 필자는 항상 사람들이 기꺼이 듣고, 아이디어를 받아들이고, 보고서를 건설적으로 사용하는 모습을 보고 있다.

외부 플레이어들과 플레이테스팅을 정기적으로 수행하는 것은 게임 개발 시 중요한 정도가 아니라, 필수 관건이다. 플레이어들로부터 직접적인 피드백을 받음으로써만이 게임 내에 문제점이 무엇인지를 알 수 있게 된다.

매일 게임을 하는 사람들은 같은 방식으로 단기간에 많은 문제들을 대하는 요령을 익히게 되기 때문에 신선한 눈으로 게임을 바라볼 수 없게 된다. 실제 세상에서 온 게이머들이라면 그렇게 되는 일이 없을 테니 말이다.