**GAME PROGRAMMING**

**텀 프로젝트 최종보고서**

**컴퓨터학과 2013210107**

**김정수**

1. **구현 환경**

* Windows 8.1 / Unity 5.4.1f1 Personal
* Android sdk 24.0.3 / android-ndk-r10e / jdk1.8.0\_60
* Oculus Platform SDK V1.8.0
* 제공 받은 GearVR / Galaxy Note 5

1. **실행 방법 / 조작 방법**

VR 착용 후, 화면 상의 Target을 기어vr 버튼 클릭을 통한 shot으로 죽이면 게임이 시작한다.



Player는 정해진 길을 따라 자동으로 이동하게 되며 주변을 살펴 조준하여 target을 폭파시킨다. 화면 상단에 자신의 score도 확인해 볼 수 있다.

Editor에서 실행 시, FPSController\_2 🡪 OVRCameraRig 🡪 TrackingSpace 의  
Camera\_rotate script를 활성화 시키면 마우스 이동으로 화면이동이 가능하다.  
shot : 마우스 좌클릭 또는 Ctrl .

1. **다운 / 구매하여 사용한 Asset**
2. **Let’s Try Assets**  
   : 이 게임의 가장 기본이 되는 asset.  
   Raycasting을 통해 사물을 쏘고, 맞은 사물에 대한 collision효과를 구현하는데 참고하였다. 처음 시작공간은 이 asset 그대로 사용하였다.
3. **Subway Level Kit**  
   : 처음 시작 배경 이후, 도시로 넘어가는 과정에서 터널로 사용한 asset.  
   터널의 일부분과 연기 particle을 가져와 사용하였다.  
   이 중, railroad를 구성하는 나무판자와 철로 object를 가져와 하나짜리 railroad를 조립하여 나중에 구현한 Bezier spline에 적용시켜 게임에서 Player가 이동하는 경로를 railroad로 보여주는데 사용하였다.
4. **Stylized Simple Cartoon City**  
   : 터널 통과 후, 게임의 배경이 되는 도시 및 자동차/탱크/헬리콥터 asset.  
   헬리콥터는 정지해 있었지만 그냥 script를 넣어 프로펠러 회전 및 경로 설정을 통해 이동하게 만드니 꽤 그럴듯하게 비행기 같았다.
5. **Nature Starter Kit 2**  
   : 기존 cartoon city object를 사용하였을 때, 직접 mobile로 실행하니 상당히 느려지는 현상이 생겼다. 혹시 city object의 퀄리티가 너무 높아 다른 asset으로 대체하려 다운받은 asset인데 결론적으로는 사용하지 않았다.  
   다만, skybox와 city의 배경이 되는 terrain을 꾸밀 때, 나무와 texture 정도만 가져왔다.
6. **Tanks! Tutorial**  
   : 이 게임에는 이 asset은 적용시키지 않았다. 처음에 tank게임을 만들려고 했을 때,  
   참고하고 공부하려 다운 받은 asset이다. 다만, 너무 완성도가 높은 asset이라 참고하기가 까다로워 주제를 바꾸게 되었다.  
   이 게임에서는 물체가 폭발하는 효과음과 효과 particle만 가져와서 사용하게 되었다.
7. **처음 제안 내용 대비 최종적으로 완성한 정도**

처음 제안 내용은 Tank 게임이었다. 코딩 실력이 부족하여 우선적으로 완성도 있는   
게임을 하나 만들어 보려는 것이 목적이었다. 하지만 막상 Tutorial을 통해 구현해 본 결과 이미 완성되어 있는 탱크 게임이 있어 새로이 만들기가 힘들었다.  
대충 해당 Tutorial 게임을 카메라의 위치나 클릭 기능을 수정하는 등 Mobile 모드로 바꾸어 보았지만, VR을 적용시키려 하니 까다로워 주제를 바꾸게 되었다.  
처음 제안 내용이 구체적이지 않아, 최종 완성도는 높다고 생각한다.  
다만, Target이 되는 object들이 뒤로 밀린다던가, AI를 넣는다던가 하는 좀 더 세부적인 부분을 넣지 못한 아쉬움은 있다.

1. **변경사항 혹은 미구현사항이 있을 경우 그 이유와 내용**

\* 주제변경 🡪 VR게임에 좀 더 어울리는 주제.  
\* 인공지능 AI 🡪 AI를 적용하고는 싶었지만 적절히 어디에 어떻게 적용할지 떠오르지 않음.  
\* Target이 맞았을 때의 효과 (ex. 사람이 맞으면 넘어진다.)  
🡪 래그돌, 메카님 등을 이용하면 된다고 들었지만, 게임의 다른 부분에 시간을 할애하다보니 여기까지 신경쓰지 못함.  
\* 무엇보다 게임을 이해하기 위해서는 직접 실행해보면 될 것 같다.

1. **기술적인 문제를 어떻게 해결했는가**

모든 문제는 google 검색과 Unity 홈페이지 스크립트 레퍼런스를 참고하여 해결하였다.  
(<https://docs.unity3d.com/ScriptReference/index.html>)

**a) Bezier spline**   
: 이 게임의 가장 핵심적인 부분이다. 수업에서 배운 부분이라 바로 활용하기도 좋고, 이해하기가 수월했다. 실제 따라하면서 구현한 곳은 다음과 같다.  
(<http://catlikecoding.com/unity/tutorials/curves-and-splines/>)

이를 통해 배우면서 Unity editor도 다룰 수 있다는 점이 매우 좋았다.  
다만, 완성도가 높아 배울 것은 많았지만 좀 더 내가 구현할 수 있는 부분이 떨어졌던 것 같다. 이렇게 습득한 Bezier spline 기능을 가지고 기찻길을 만드는 등 script를 조금씩 더 수정하여 게임에 적용시켰다.

**b) 폭발 효과**  
: 각각의 object에 대해 보이지 않는 Cube를 하나 두고 그 cube와 collision이 발생하면   
폭발 및 소리를 구현하였다.  
터널에 있는 형광등의 경우는 유리깨지는 소리를 이용하여 그럴 듯 하게 바꾸어 보았다.  
소리나 배경음악의 경우는 직접 원하는 소리파일을 찾아 적용하였다.

**c) 기어VR과의 호환성**  
: 자바 jdk, 안드로이드 sdk, ndk를 버전에 맞게 설정하고,   
폰 device id를 알아내어 osig(Oculus Signature File Generator)를 이용하여만든 파일을  
<project>/Assets/Plugins/Android/assets/ 폴더에 넣고 빌드하여야 실제 모바일에서 실행할 수 있다.  
(<https://dashboard.oculus.com/tools/osig-generator/>)

**d) 카메라에 UI가 보이지 않는 문제**  
: 에디터에서는 보이는 UICanvas가 직접 VR상에서는 보이지 않는 문제가 발생하였다.  
이는 실제 VR착용시, UI는 Camera space가 아닌 world space에 그려지기 때문이었는데, 따라서 해당 화면에 Score 및 조준점을 표시하는 UI를 world space로 옮겨주면 된다.  
나는 다른 방식을 사용하였는데, UI만 인식하는 Camera를 새로 만들어주어,   
VR의 CenterEyeAnchor의 자식으로 놓아주었다. 해당 방법은 다음을 참고하였다.  
(<http://answers.unity3d.com/questions/1027642/vr-how-to-display-a-canvas-with-oculus.html>)

실행 시, VR에서도 UI가 잘 나타남을 알 수 있었다. 다만 약간 UI와 배경이 좀 이질적이라  
어색한 감이 있는데, 이를 해결하는 것도 또 다른 하나의 문제인 것 같다.

**e) 모바일에서의 최적화 문제**  
: 실제 모바일에서 실행하니 많은 문제가 있었다. 특히나 속도문제가 가장 심했다.  
구글링을 해보니 Drawcall이 많을 경우(즉, object들이 너무 많을 경우) 이런 현상이 많이 발생한다고 하여, simple city asset의 object들을 static한 것은 하나로 합치기 위해 많이 찾아보았다. mesh bake라는 asset을 다운받아 이용하면 여러 개의 object를 하나로 합치고 또한 atlas도 최적화 시킬 수 있다 하였지만, 유료라서 다른 방법을 사용하기로 하였다.  
우선적으로 최대한 움직이지 않는 object들은 모두 static으로 바꾸어 메모리 위에 올려두었고, 카메라의 시야를 짧게하여 ( Far : 1000 -> 60 ) 최대한 그려지는 object를 줄였다.  
좀 나중에 알게된 사실이지만, 처음 가로등을 표현하기 위해 point light를 많이 사용하였는데,  
이 light가 상당히 부담이 가는 작업이라고 한다. 따라서 이 light를 모두 삭제시키고 나니 꽤 끊기지 않고 실행할 수 있었다.  
가장 속도를 높여주었던 것은 rendering 방식이었다.  
Edit 🡪 Project Settings 🡪 Player 에서 android setting을 보면, rendering path를 정할 수 있다. 기본으로 Forward가 설정되어 있어 그렇게 하였을 때는 상당히 끊겼었는데, 구글링을 한 결과 모바일에서는 Legacy Vertex Lit으로 하는 것이 가장 최적이라고 한다.  
이렇게 바꾸어주니 상당히 속도가 빨라짐을 느낄 수 있었다.

**7. 팀원 간 역할 배분**

개인 프로젝트

**8. 기타 작성하고 싶은 내용**

처음에는 학구열이 넘쳐 물리적인 기능을 직접 구현해보고 싶었지만, 결론적으로는 게임성만 남게 되었다. 그래도 bezier curve, spline을 직접 공부해보고 구현할 수 있어서 재미있었다. 처음에는 점수를 얻을만한 구현을 생각하니까 떠오르는 아이디어도 없고 막막했지만, 그냥 그런 것을 포기하고 만들다 보니 추가하고 싶은 기능들이 하나하나 생겨났다. 그때마다 해결해 나가다 보니 뭘 해야할지에 대한 부담감도 없고 더 재미가 생겼던 것 같다.