Instructions:

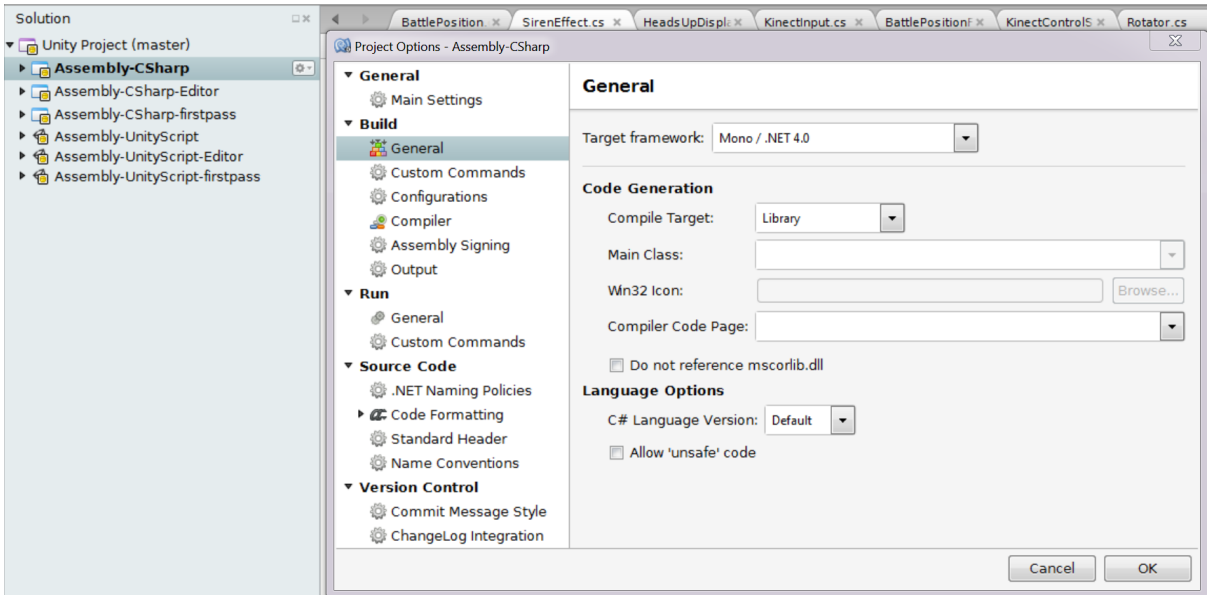
VR shooter 3.0 developed by team awesome.

1. Important Notices

a) To run the scripts, please use .NET 4.0 to compile/build our scripts.

How to: Right click “Assembly-CSharp” – Options – General – Target framework – Mono /.NET 4.0

Also “Assembly-CSharp-Editor” need to use 4.0 as well.



b) Hardware problems with unity + Kinect. You need to save everything and restart Unity, when you wish to run the game in Unity’s debug mode.

An alternative way to run the game is to build the game, and start with *.exe file. You can build the game from “Files” – “Build” or “Ctrl + B”

c) Kinect might lose track of the player during the game. Make sure nobody is around.

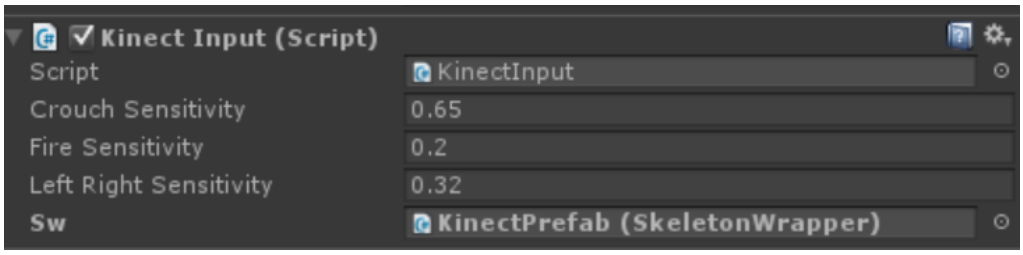
d) You need to do the Kinect Calibration gesture to start the game. You can have a look at our Youtube video about gestures: http://youtu.be/iuf7ziCP7KE

e) Our game do have audio, but be careful. When using DVI cable for the connection of Oculus Rift, the audio output device might be changed to DVI. You can manually set your audio output device in control panel.

2. Kinect &Oculus Rift, and Unity project tips.

a) Require Unity Pro to work with Oculus, so alternatively you can disable the oculus, but enable the main camera attached to the player.

b) Some parameters of KinectInput.cs script attached to OVRPlayerController gameObject need to be adjusted for different game environment. i.e. crouchSensitivity.



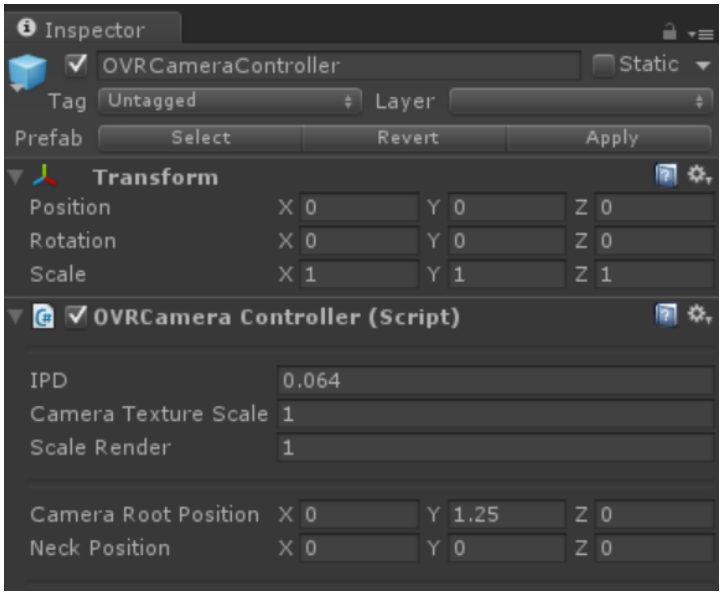
You can see the comments of these parameters in KinectInput.cs for help. If the detection of Kinect is not perfect, you can adjust these parameters for better performance.

c) Make sure that Kinect is 2 meters in front of the player, and there is no other players/big objects blocking the current player.

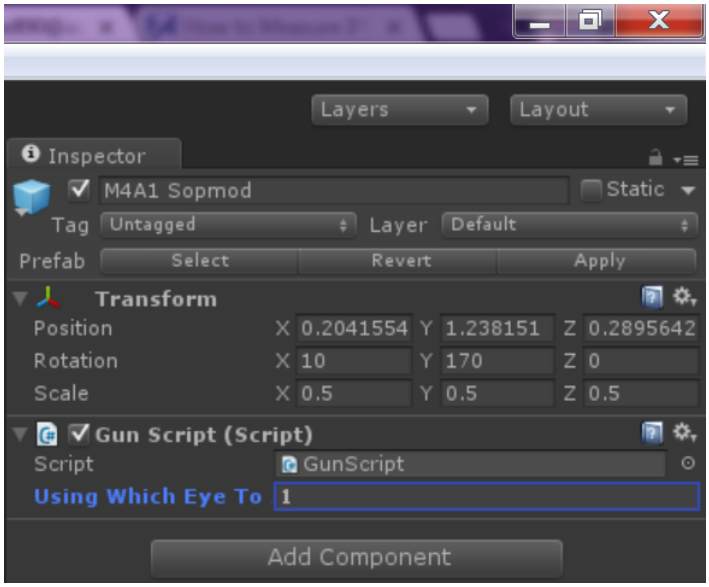
d) Oculus Rift calibration. You can do a calibration before using the device.

How to calibrate: http://www.roadtovr.com/measure-ipd-oculus-rift-configuration-utility/

Then we can arrange the IPD (interpupillary distance) in OVRCameraController:



d) The default aiming setting is using right eye to aim. You can set the parameter of this in GunScript.cs You can also change the game difficulty by setting the HP of enemy, enemies's shooting frequency, player's health point and so on.



In the figure above, we can set which eye to aim, 1:right eye; 2:left eye; 0:both eye

Using 2 eyes to shoot is a crazy idea, but we reserve the option for crazy players.

g) The in-game heads-up displays might break to 2 lines, with small screen resolutions. Try to use full screen during the game ☺

If any problem with marking, i.e. does not have account for Unity Pro, please contact Christof.

Work division

Sorry we did not use GitHub to store. Unfortunately something wrong with the wireless of the campus, we tried to upload the project but all faied, it's too big (>1.2GB). So we don't have a log file.

In our group, we merged all our works together at week 8, and Christof helped us to upload this project onto Github.

The following part is the work division edited by each member:

Henry:

Enemy Shooting and Targetting

Bullet Animations and Effects

Scene Transitions using Spline Controller

Oculus Heads-up Display

Player and Enemy Interactions (i.e. Getting shot)

Kevin:

Game anlaysis

Player's exercising actions/gestures anlaysis

Enemies' locations and player's game character's location moving path and stopping locations

Assets design.

enemies design.

Police car design and police car lights' effects.

warehouse design and lights' effects.

Yang:

Scripts implementation.

KinectInput.cs:

- Detect the player's height, position
- Detect player's action/gesture during game.

KinectContrlscript.cs:

Player character react when detect the input from Kinect, such as crouch, left/right dodge, and start shooting.

Barrel.cs:

Control the event of barrel explosion

BattlePosition.cs, BattlePositionFinal.cs:

Each battle position is a place where player fights with enemies.

Initialize enemies, enable barrels, set left/right dodge restricts, when player arrives a new battle position.

Activate splines so player will move between battle positions

InitialPosition.cs

Perform pre-game Kinect calibration before game

Countdown before game

BulletFly.cs EnemyBulletFly.cs:

Describe the behaivour of a bullet

Enemy bullet can slow down the time scale

GunScript.cs EnemyGunScript.cs:

Gun shooting trigger, shooting animation, audio clip, bullet spawn

And calculating the position for bullet (bullet target)

SpawnBullet.cs EnemySpawnBullet.cs:

The gameObject at gun muzzle, spawn bullets at a certain frequency

Warehouse scene:

Place enemies and barrels based on Kevin's game design.

Draw splines between each battle position, and design triggers so enemies will appear when player arrive a new position.

Sound effects

Others:

Experiment with Kinect and Oculus Rift to improve performance

i.e. robustness of gesture detection, moving speed of head's orientation

Tuning/debugging the game

i.e. adjust size of colliders, re-initialize the player’s left, middle, right position at each battle position.

Q&A

Q: How to play our game?

A: Youtube video: http://youtu.be/iuf7ziCP7KE

Q: Why the enemies are not moving?

A: Our game aims to motivate players to do more exercise (dodging), if the enemy is moving, player will be distracted by the enemies' movements, and will pay less attention to physical exercises.

Q: Why do you put a mirror here?

A: To debug/demonstrate the crouching action of the character. However the reflection in Unity is not so good.

Thank you for marking!

Q: Why there is a cursor in the middle of the game scene?

A: You can use Alt+Enter to enter full screen. And if the mouse is still there, use alt+Tab to switch to Desktop, and switch back.

If problem still occurs. Try to run it in Unity…

Thanks for marking.