**Setting up the project for development**

**Requirements:**

Must be running windows.

Must have a GearVR-compatible phone if you wish to test the game on android.

Follow this link: <https://cgcookie.com/2015/12/16/quick-start-guide-gear-vr-unity/>, or follow the summary below.

Download latest version of Unity (<https://unity3d.com/unity> -> Personal -> Learn More -> Try Personal -> download installer) and install.

Install Android Studio [https://developer.Android.com/Studio/index.html](https://developer.android.com/studio/index.html)

Find the file path of the Android SDK and save it somewhere.

Install most recent JDK (google java jdk and download the Java Se Development kit), find the file path of the JDK, should be in C:\ProgramFiles\Java and appear as jdk-somenumbers or jdksomenumbers) and save the file path somewhere.

Download project from github: https://github.com/takoda1/ReliefFinally.

Open Unity and open the root folder of ReliefFinally to open the project.

In the editor window, go to edit->preferences.

Click External Tools.

Under Android, it should say SDK and JDK, put the saved file paths of the SDK and JDK into these boxes.

Next, in the editor, go to File -> Build Settings, click Android, then click Switch Platform.

In the same window, click on player settings, then the Android logo in the top bar.

Click Other Settings and under Identification, change the Minimum API level to 19 or above.

If you are using Visual Studio to edit code, make sure to also set the… (Experimental version 4.6 leads to crashes when run on the device)

In the same player settings window, click XR Settings and enable Virtual Reality Supported.

Now follow the directions in the **Building for the Gear VR section** of <https://cgcookie.com/2015/12/16/quick-start-guide-gear-vr-unity/>.

**Using Visual Studio to edit code**

asdf

**Debugging the game on the Android device.**

Must have Android Studio or at least developer tools to debug.

It is easiest to have the Android device connected by USB for debugging. If you wish, you can also wirelessly connect your device for debugging by following the instructions under **Connect to a device over Wi-Fi** here: <https://developer.android.com/studio/command-line/adb.html>.

First you have to enable USB debugging on the Android device. In settings, find the Build Number setting, usually under About Phone, and click it at least seven times. This should enable Developer options. Next click on Developer options in the Settings screen. Scroll down in the Developer options screen and enable the USB debugging setting. Now, whenever the phone is connected to a machine it should prompt you to allow or prevent USB debugging from the said machine. Allow debugging from your machine when you connect the Android device.

There are two methods to see Logcat logs from the Android device. Non-gui Adb Logcat and Gui monitor. Both can be found deep within your file system and run from there, but it is easier to set them as system PATH variables and run them with short commands from the Command prompt (Instead of running C:\Users\username\AppData\Local\Android\sdk\platform-tools adb logcat, you can just type adb logcat and execute).

To set system PATH variables in windows 10, type View Advanced System Settings in the search bar, click Environment variables in the window that pops up, then under System variables, click Path to highlight it then click Edit. To add new environment variables, click New, then either type in the path or browse for it. The path for logcat and monitor.bat should be something like C:\Users\username\AppData\Local\Android\sdk\platform-tools and C:\Users\username\AppData\Local\Android\sdk\tools.

Now, open a command prompt (type cmd in search bar -> press enter) and type either adb logcat or monitor.bat and press enter to execute.

**Adb Logcat summary**

To view a specific log, click on the desired log to stop new ones from scrolling the logs up.

To make the logs continue scrolling after stopping the logs, press shift+enter.

To exit adb logcat press ctrl+c.

If you just type adb logcat, you get every single log from the android device, which is undesireable if you just want logs from the game. Instead, use the logcat options to filter for what you want: <https://developer.android.com/studio/command-line/logcat.html>.

**Monitor summary**

In the bottom half of the monitor, it should be displaying all logs from Logcat. In the search bar at the top, you can type in any string that you are looking for in the logs and it will automatically filter those logs for you to view.

**Project structure**

**Overview**

**Controllers**

Project compatible with either the GearVR controller or StratusXL controller.

**Scene Structure**

Scenes

MainMenuScene

TutorialScene

GrassyPlainsScene

BarnacleWatersScene

SnowyMountainScene

GrassyPlainsSceneControlled

BarnacleWatersSceneControlled

SnowyMountainSceneControlled

Scenes are split into three categories: the menu scene, the game scenes, and the controlled game scenes. The menu scene is MainMenuScene, the game scenes are TutorialScene, GrassyPlainsScene, etc, and the controlled game scenes are GrassyPlainsSceneControlled etc.

**Menu Scene Structure**

<http://primeyoullc.net/steel-series-stratus-xl-game-controller-mapping-unity-3d/>





Playercontroller: stratus xl controller axis already correspond. Just need sprint functionality+footsteps. GearVR Controller: move +look around +footsteps