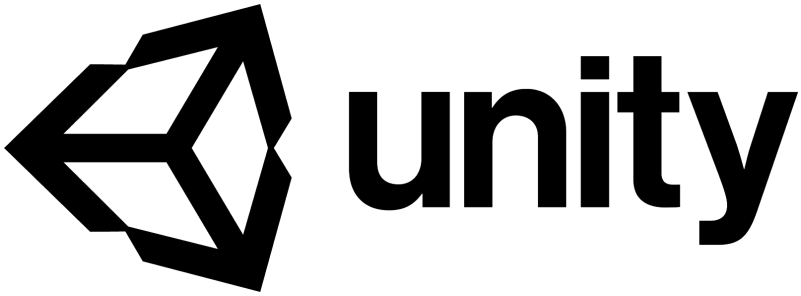
Accelerate Your C# Learning with Unity

Troubleshooting Guide



Welcome to the troubleshooting guide of **Accelerate Your C# Learning with Unity!** This guide will help you with troubleshooting any issues you may face while following the workshop materials.

Please choose your Operating System below to see the troubleshooting guide:

This is the troubleshooting guide for Windows. If that is not your operating system, please go back to the first page to choose the right troubleshooting guide.

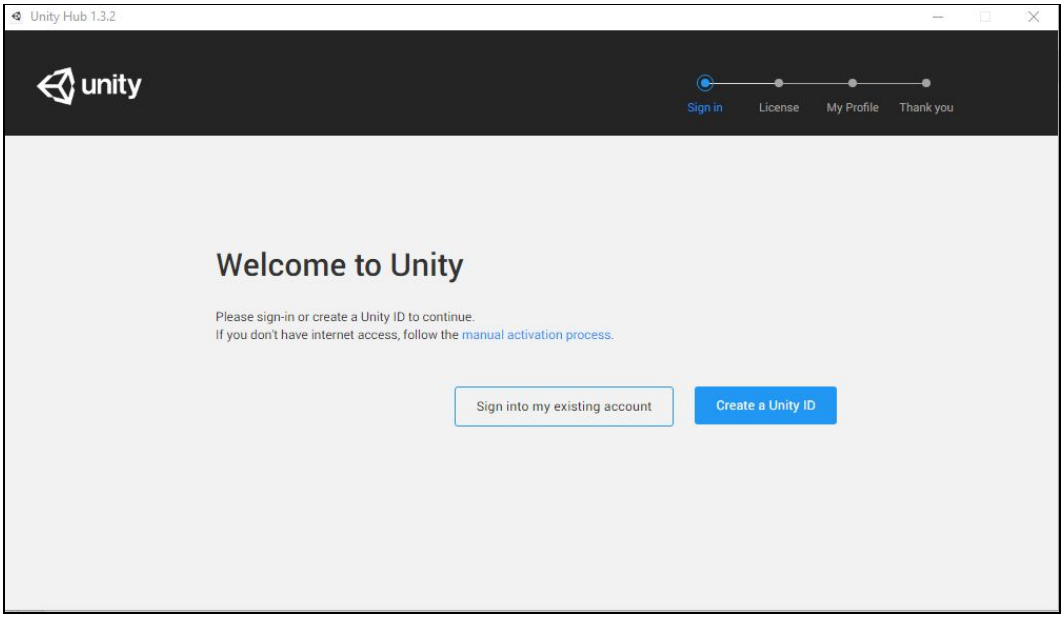
The following sections describe the possible problems that could occur or things to be aware of when following the workshop.

SECTION 1: EXTRACTING CODE

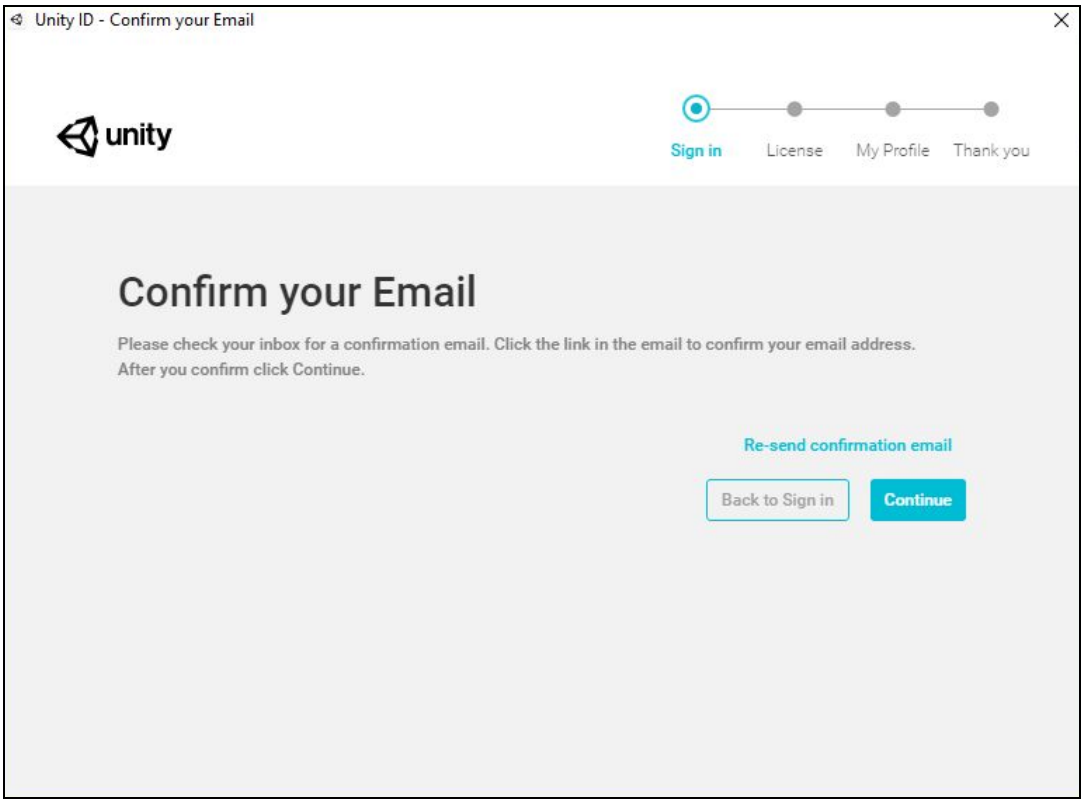
1. To download Unity and the project code, you will need about 2GB of space. This might be a lot for some people's computers. If you need to make space, a good resource is to download CleanMyMac which helps to sort old files and delete them. You can also empty your Trash can or uninstall Applications that you do not use regularly.
2. If you are using a **flash drive** to download the project, insert the flash drive into your computer. Select the name of the flash drive and copy and paste the PC folder and the README, right click and **Save** it onto your computer. **Do not** drag and drop the file onto your computer. When you have the file on your computer, eject the flash drive and pass it to the next person.
3. If you are **not** using flash drives, navigate to this web address to download the file: http://mlhlocal.host/unity-project. Do not be worried that it takes a long time, it is a very large file.
4. Ensure you **delete the /unity-starter-code part in the extract all dialog box**, or else there will be a nesting of folders.
5. Use 7-Zip / WinRAR for extracting the files, this will make the extraction much faster than the traditional Windows extractor.

SECTION 2: INSTALLING UNITY HUB

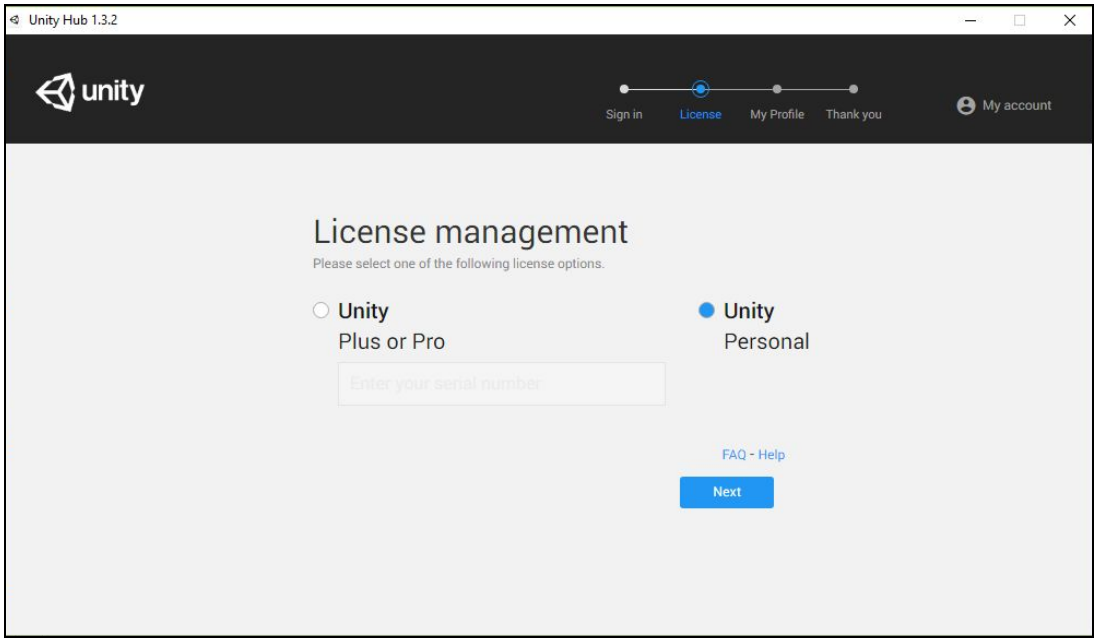
1. Once Unity Hub opens, click on create a Unity ID if you need to create one, or sign in using your existing username and password.



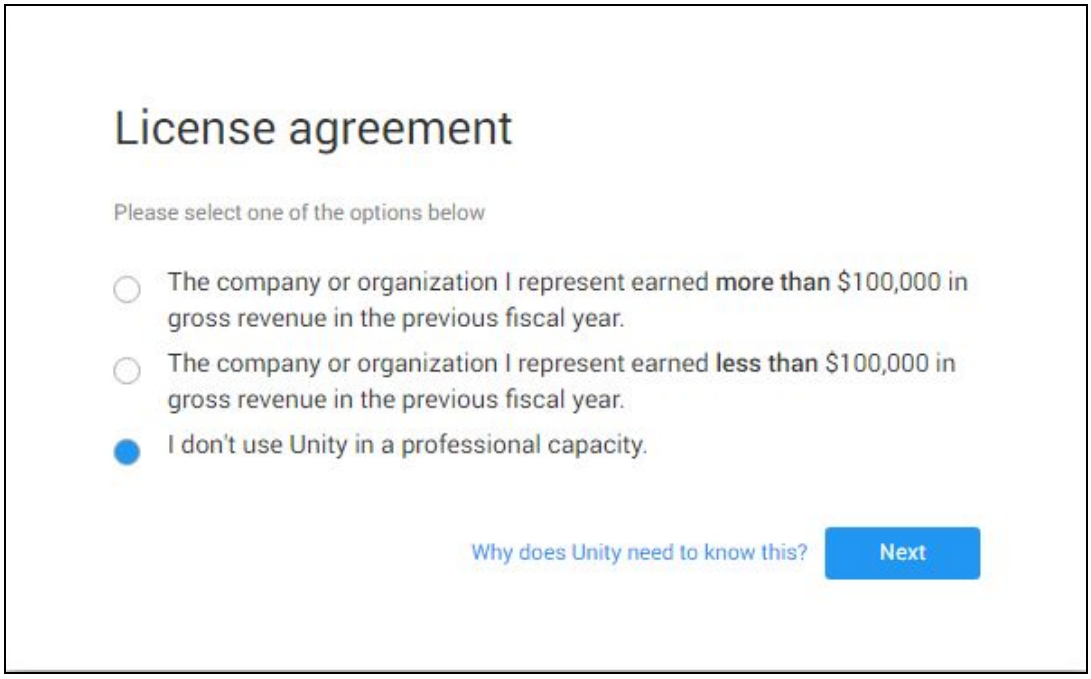
2. If you sign up using your email, after entering all your details, you will see this screen. Ensure you click the confirmation link in your email and then press **Continue**.



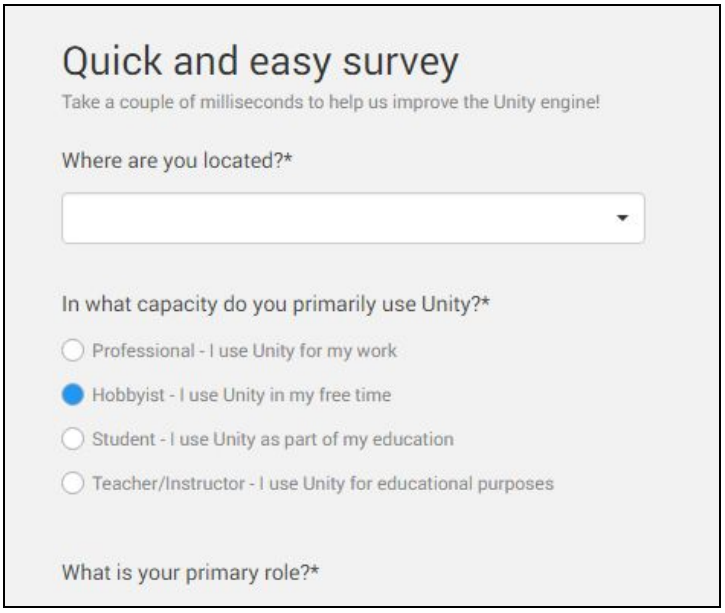
3. Unity Hub will then ask you to choose the type of license you have, pick **Personal** and click on **Next.**

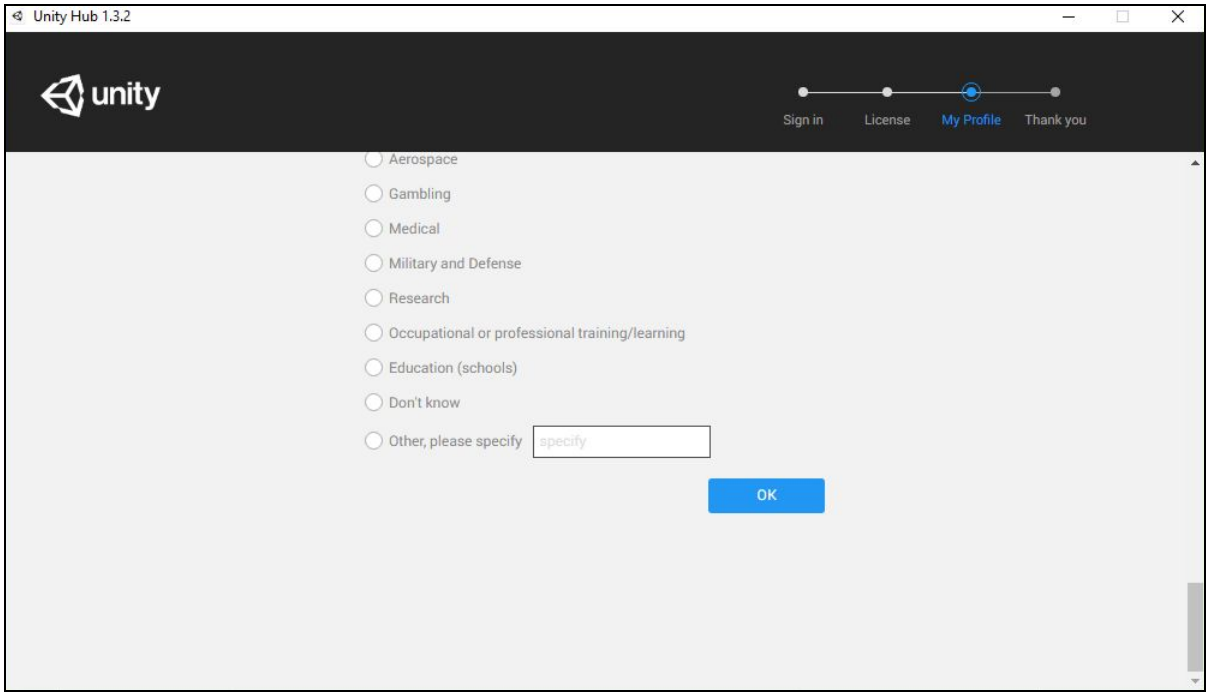


4. In the license agreement screen, choose the option most relevant to you (in this case, option 3) and click on **Next.**

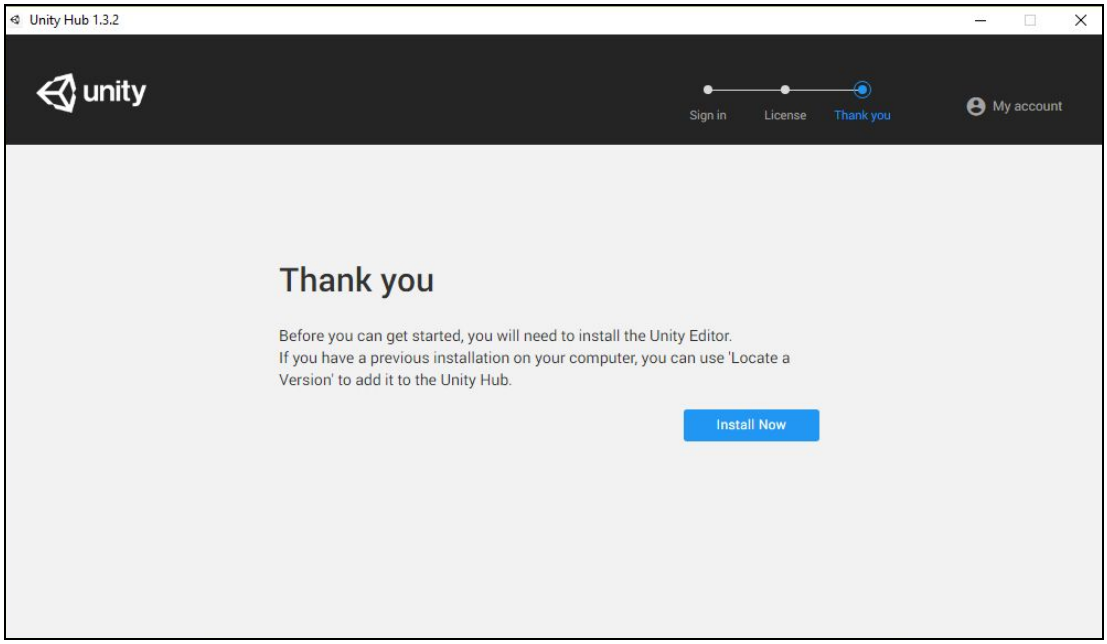


5. Unity Hub then asks you to fill a survey. Fill in all the details and click on **OK**.





6. Click **Install Now** and sit back while Unity Hub is installed!

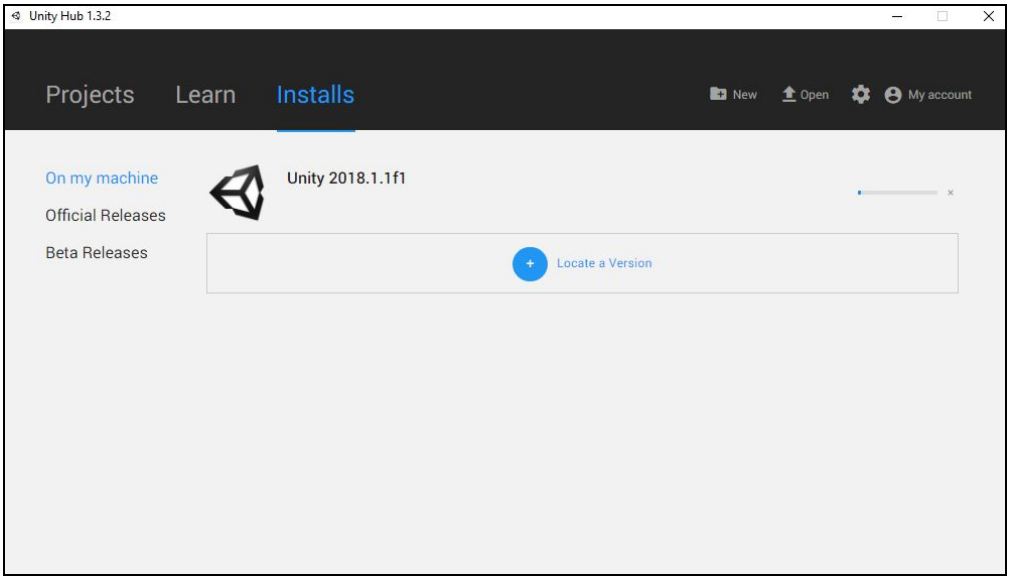


7. The terminal might suddenly open in the middle of installation - there is no need to worry, the terminal will auto close and install will complete.
8. Once Unity Hub is installed, there will be a checkbox that asks if you want to run unity hub (checked by default) - leave it checked as you will need to open Unity Hub for the next step.

SECTION 3: INSTALLING UNITY

1. Pick the Unity Hub option in the site (easiest way to download Unity) which is right next to version 1.1.

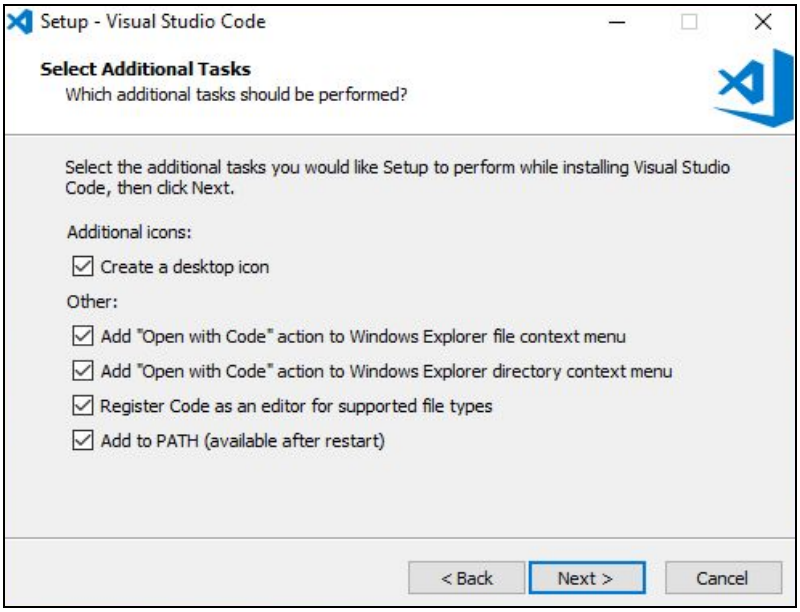
2. Check only the **Standard Assets** and **Unity Editor.** Uncheck everything else.
3. Once you click **Done**, you will see the progress bar with the version as shown in the image.



4. Computer might make a lot of noise, and the terminal might randomly appear for a few milliseconds. Don't worry, this is completely normal!

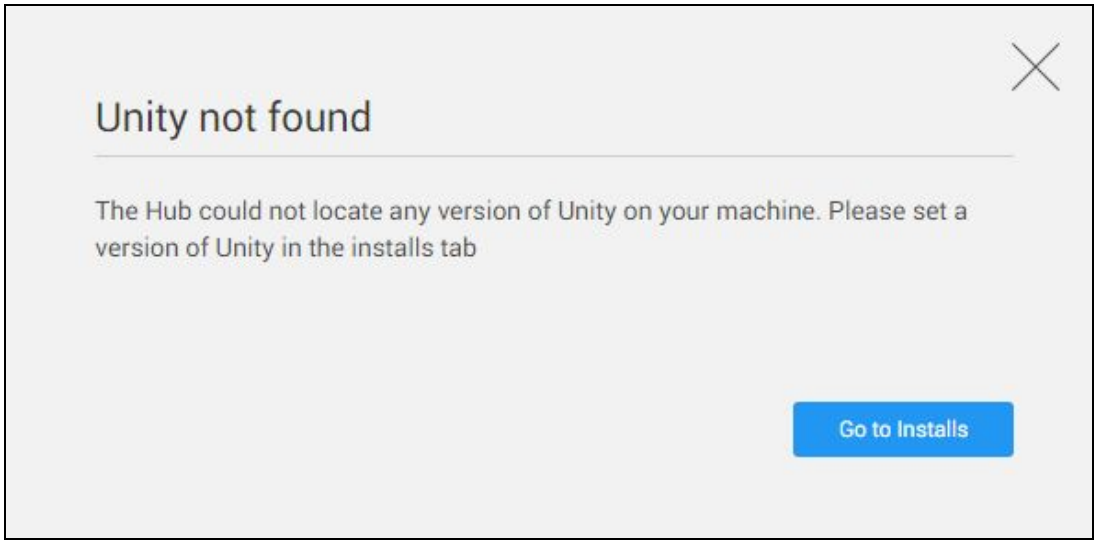
SECTION 4: DOWNLOAD AND INSTALL VS CODE

1. If the site does not show the correct version for your OS, scroll down to the bottom of the site and manually select the version that matches your OS.
2. Restart your PC after installation to ensure the PATH variable is set (not essential, but may be useful).
3. During install, choose all the options in this dialog box to make it easy for you to work with VS code (not essential, but may be useful).

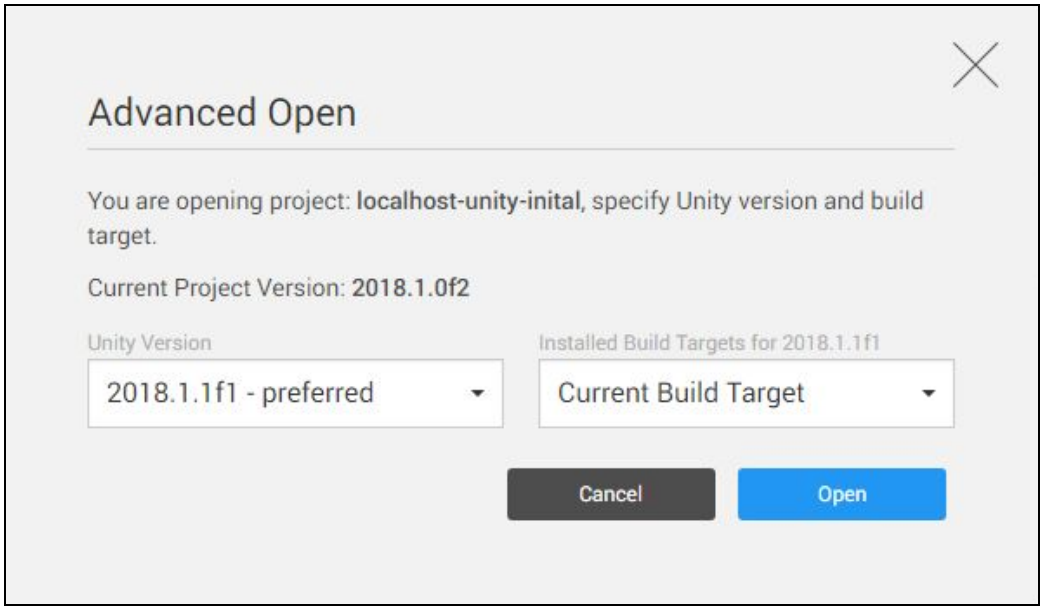


SECTION 5: SET UP CODE

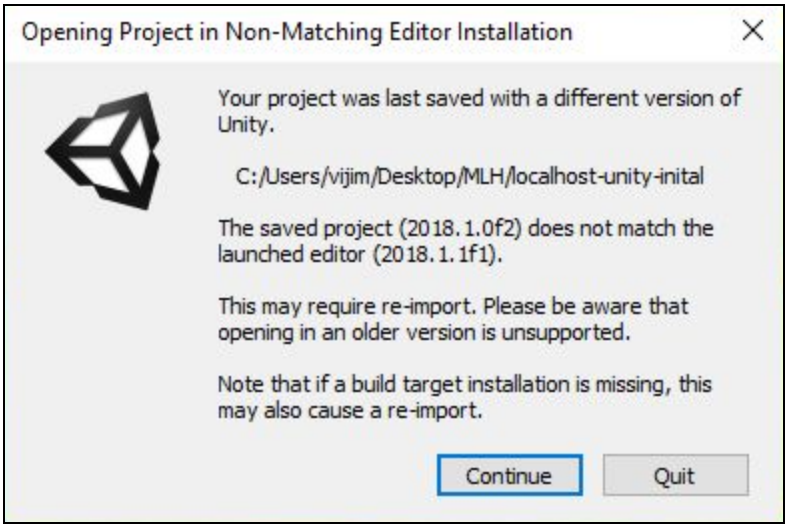
1. You need to wait until Unity is installed before clicking on **Open**, otherwise you will get this message -

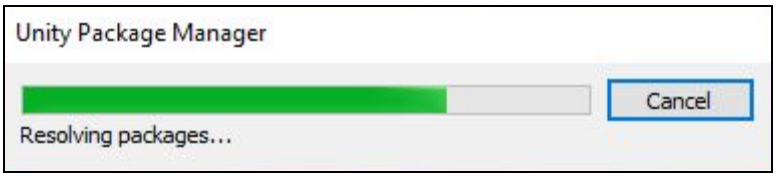


2. Once you browse and select the folder to open, you will be shown this dialog box, leave its default values in and click on **Open**.

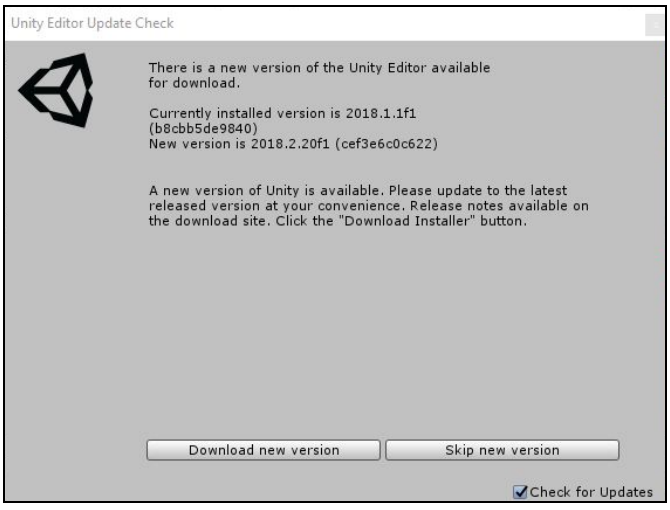


3. Since the project was built on a Mac, you might see these screens on Windows, asking you to re-import and switch target to Windows. Simply click on **Continue** and switch target respectively, and the code should load without any problems.





4. Sometimes, Unity Hub might ask you to update to the latest version of Unity. Do not do that since the project is built in Unity 1.1 and thus will run only on that version of Unity!

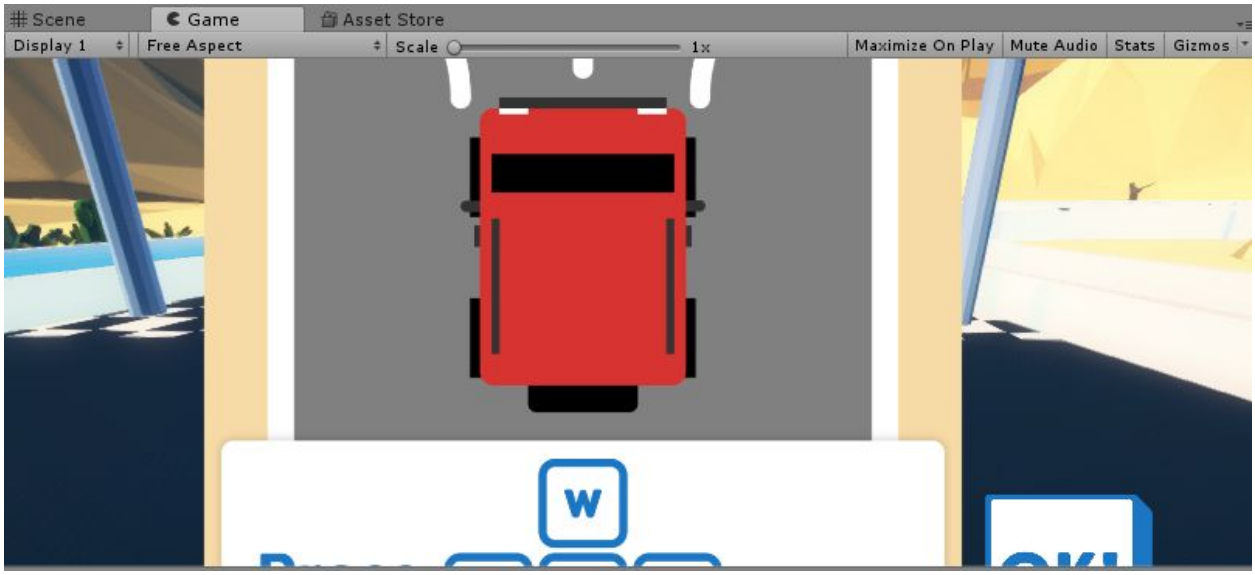


SECTION 6: SETTING EDITOR

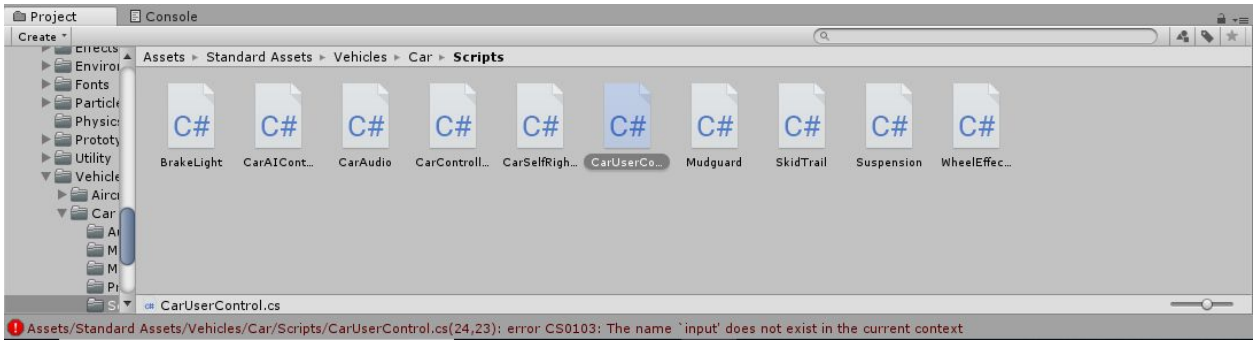
1. Visual Studio code usually creates an icon on your Desktop so you can navigate to that to choose Visual studio as the default editor.

SECTION 7: UPDATE THE CODE

1. When you run the game on Unity Hub, you may not be able to see all the elements in the game, even if the scale is at 1x. This does not mean that there is anything wrong, the final build will be full screen and will have all the components. For now, simply click on **OK** to start.



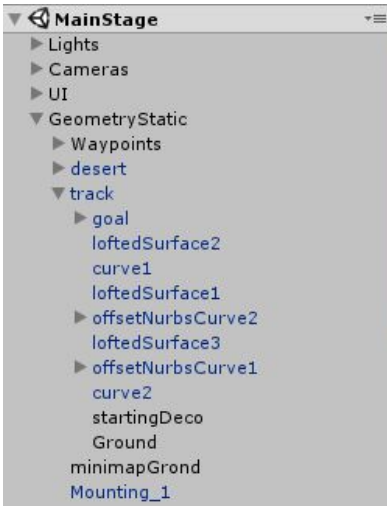
2. Remember to save your code in VS Code after each step (before running the code), so that you are always checking if the latest version of the code has no errors.
3. If there is an error in the code, the game will not run. You will see a message on the screen that says “All compiler errors have to be fixed ..” and if you see the bottom of Unity Hub (below the project window) you will see the error written in red.



4. Press the play button to start the game and press the play button again to stop the game. When the game is started, the play button looks like this  and normally it looks like 

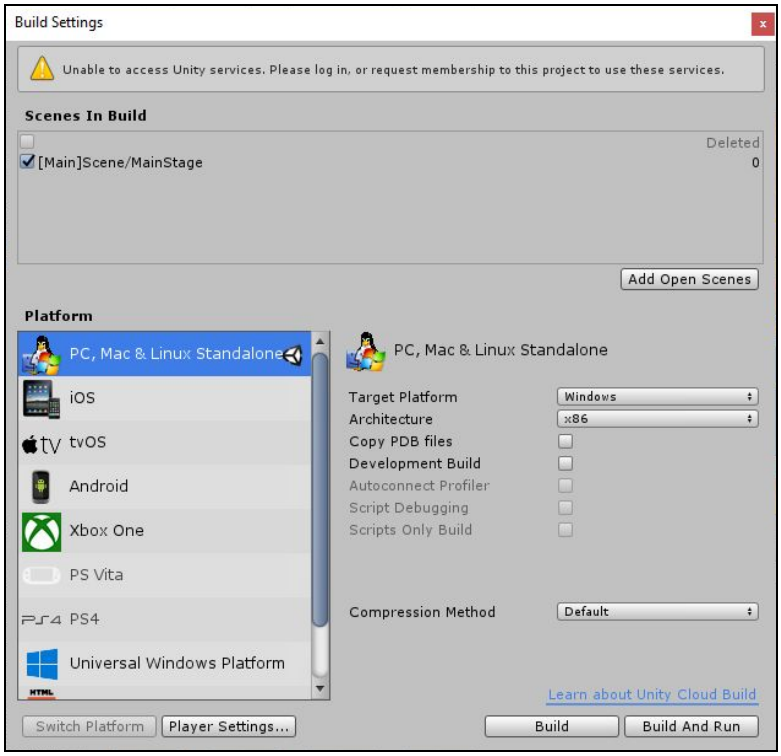
SECTION 8: COLLIDERS

1. Click on the correct component, if clicking from the UI seems tough, select the appropriate component from the Scene hierarchy (menu on the left hand side).
    a. **Road** : GeometryStatic > track > loftedSurface1
    b. **Wall 1**: GeometryStatic > track > loftedSurface2
    c. **Wall 2**: GeometryStatic > track > loftedSurface3



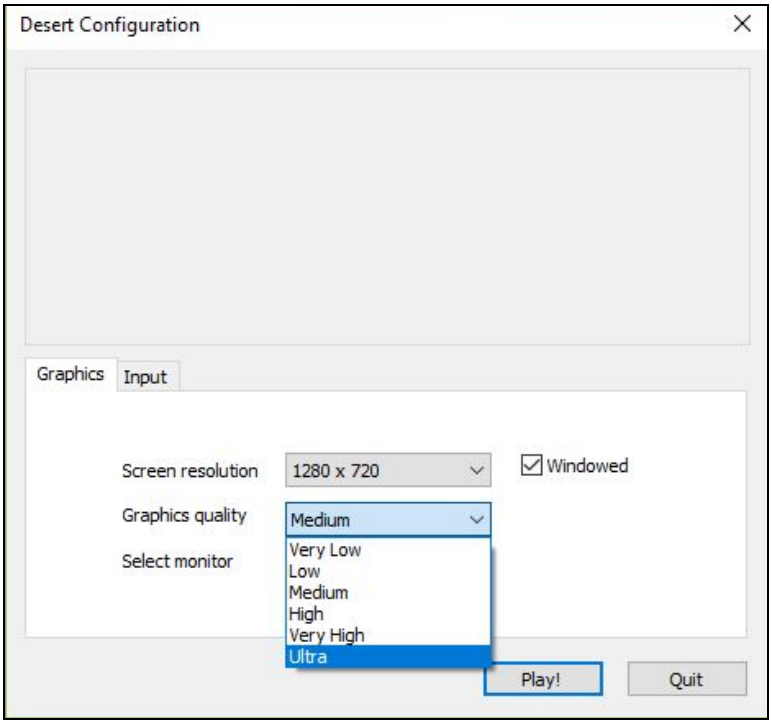
SECTION 9: BUILD PROJECT

1. Click on File > Build Settings, choose PC, Mac & Linux Standalone. Ensure the Target Platform and architecture match your machine and click on Build.



2. If you get the error “Missing project ID: … this build will not access Unity Services” just press Yes and the build will be created.

SECTION 10: TRY THE GAME


Desert Configuration
Graphics
Input
Screen resolution
1280 x 720
Windowed
Graphics quality
Medium
Select monitor
Very Low
Low
Medium
High
Very High
Ultra
Play!
Quit

This is the troubleshooting guide for OSX. If that is not your operating system, please go back to the first page to choose the right troubleshooting guide.

The following sections describe the possible problems that could occur or things to be aware of when following the workshop.

SECTION 1: DOWNLOAD PROJECT

1. To download the Unity project, you will need 1.97 GB of space. This might be a lot for some people's computers. If you need to make space, a good resource is to download CleanMyMac which helps to sort old files and delete them. You can also empty your Trash can or uninstall Applications that you do not use regularly.
2. If you are using a **flash drive** to download the project, insert the flash drive into your computer. Select the name of the flash drive and when you find the `Mac` file and the README, right click and **Save** it onto your computer. **Do not** drag and drop the file onto your computer. When you have the file on your computer, eject the flash drive and pass it to the next person.
3. If you are **not** using flash drives, navigate to this web address to download the file: http://mlhlocal.host/unity-project. Do not be worried that it takes a long time, it is a very large file.

SECTION 2: EXTRACTION

1. If you cannot find your download, double check where your downloads are going. They might be on the desktop or in another folder.
2. Double click the file to unzip it.

SECTION 3: INSTALLING UNITY HUB

1. Once you drag the Unity app icon into your **Applications** folder, to continue the process of installing the Hub, you will need to open your **Applications** folder.
2. If you create your User ID for Unity but are unable to sign in, double check whether you have confirmed your email address.

SECTION 4: INSTALLING UNITY

1. Double check that you download Unity version 2018 (2018 1.1f1) of Unity. Your game will not run if you do not have the correct version.

2. You have two options here: the **Downloads (Mac)** option or the **Unity Hub** option. The advised version is to complete the download from the **Unity Hub**.

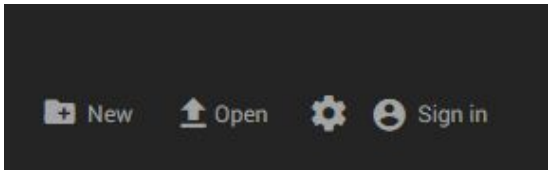
3. You will then be prompted to select Microsoft Visual Studio Community as an editor for writing C#. You can **skip** downloading this since it is very large and use a normal code text editor. If you do not currently have a text editor, VSCode is more lightweight than Visual Studio. Atom or Sublime are also good options.

4. To download Visual Studio Code navigate to: https://code.visualstudio.com. Select **Mac** as your operating system on Visual Studio Code's website and the Stable version as the package you want to download.

5. If the Visual Studio Code package does not start automatically downloading, you can select **Direct Download**.

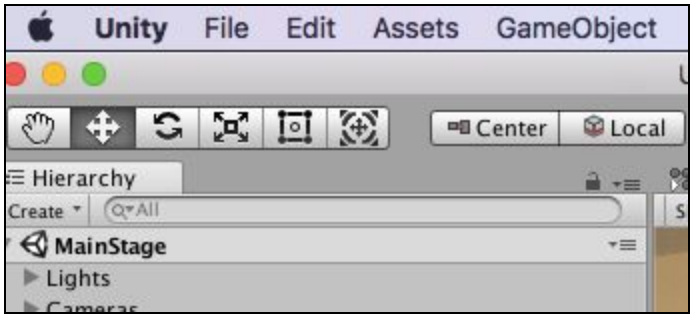
6. After you install Unity, restart your computer after installation to ensure the PATH variable is set (not essential, but may be useful).

7. Once Unity is installed, within Unity you now need to open the specific project. Select **Open** and navigate to the project folder. If your project does not open, you might not have downloaded the correct version of Unity.



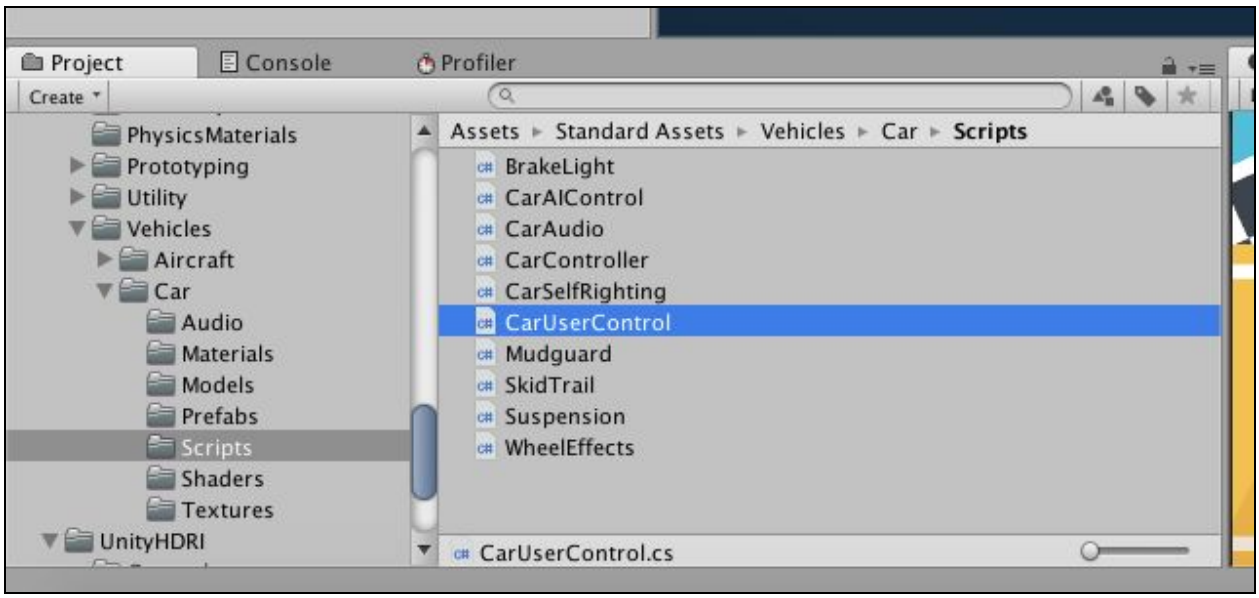
SECTION 5: SET YOUR TEXT EDITOR

1. You need to navigate from **Preferences > External Tools > External Script Editor**. Preferences is nested under the **Unity** panel.

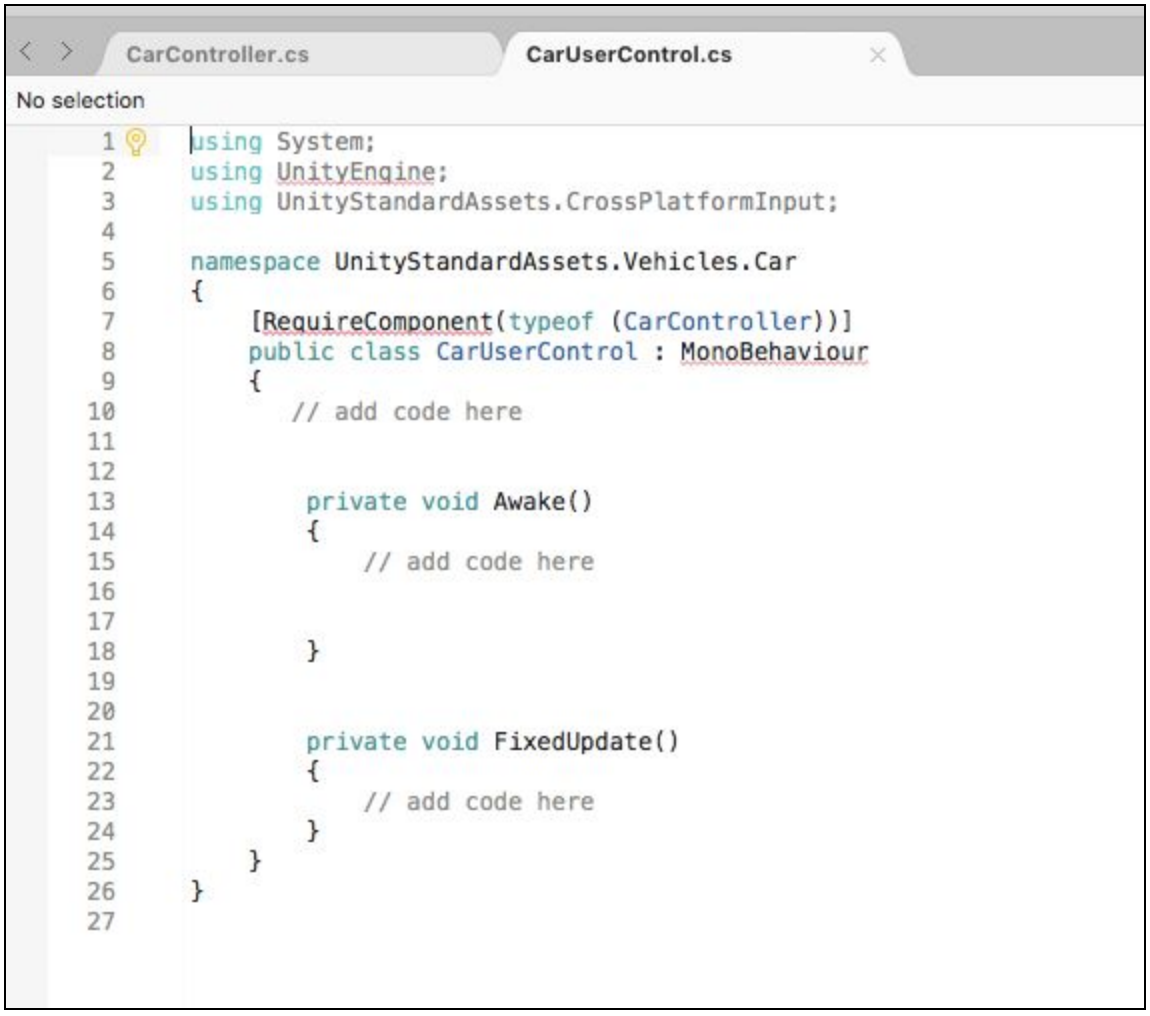


SECTION 6: FIX CARUSERCONTROL.CS

1. To get to the `CarUserControl.cs` file, navigate to **StandardAssets > Vehicles > Car > Scripts > CarUserControl.cs**. This is located on the bottom left panel of your main dashboard. If you have trouble locating this file, make sure you are in the **Project** panel of your game.



2. There are three different pieces of code that you will need to add to `CarUserControl.cs` to get the car to move. Double check whether you are writing your code in the right location, because there is also a `CarController.cs` file. You will know that you have the right file if there is a comment about how you should place your code within a certain location.



The final altered code sections should be:

CarController.cs | CarUserControl.cs

No selection

```
1   using System;
2   using UnityEngine;
3   using UnityStandardAssets.CrossPlatformInput;
4
5   namespace UnityStandardAssets.Vehicles.Car
6   {
7       [RequireComponent(typeof (CarController))]
8       public class CarUserControl : MonoBehaviour
9       {
10          {
11      Private CarController m_Car;
12  }
13
14
15
16      private void Awake()
17          {
18              m_Car = GetComponent<CarController>();
19
20      }
21
22
23      private void FixedUpdate()
24          {
25          float v = Input.GetAxis("Vertical");
26          float h = Input.GetAxis("Horizontal");
27
28      }
29  }
30  }
31
```

3. To more effectively solve errors in your code, you should save your code at each step so you can recognize where an error was created.
4. If you do have errors in your code, when you run your game you will get an error message at the top of your screen.

SECTION 7: PROJECT SETTINGS

1. In this section you need to set the input values for the API to listen to. To do so you need to access the Input Manager to select which keys control which direction the car in the game will drive. To get to the Input Manager you should navigate to: **Edit > Project Settings > Input > Axes > Vertical**. You now need to set both the Vertical and Horizontal controls.

   Vertical Controls:

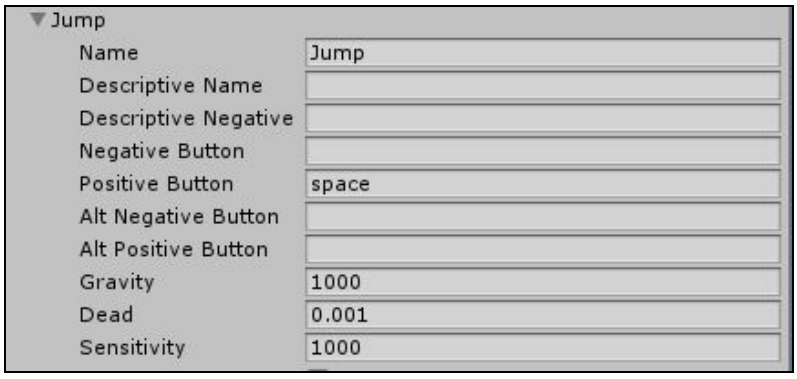
   Down as **Negative**
   Up as **Positive**
   S as **Alt Negative**
   W as **Alt Positive**

   Horizontal Controls:

   Left as **Negative**
   D as **Positive**
   A as **Alt Negative**
   Right as **Alt Positive**

If in your test run you find that your car is going in an odd direction when you hit the right key, chances are that a key is set incorrectly in this section.

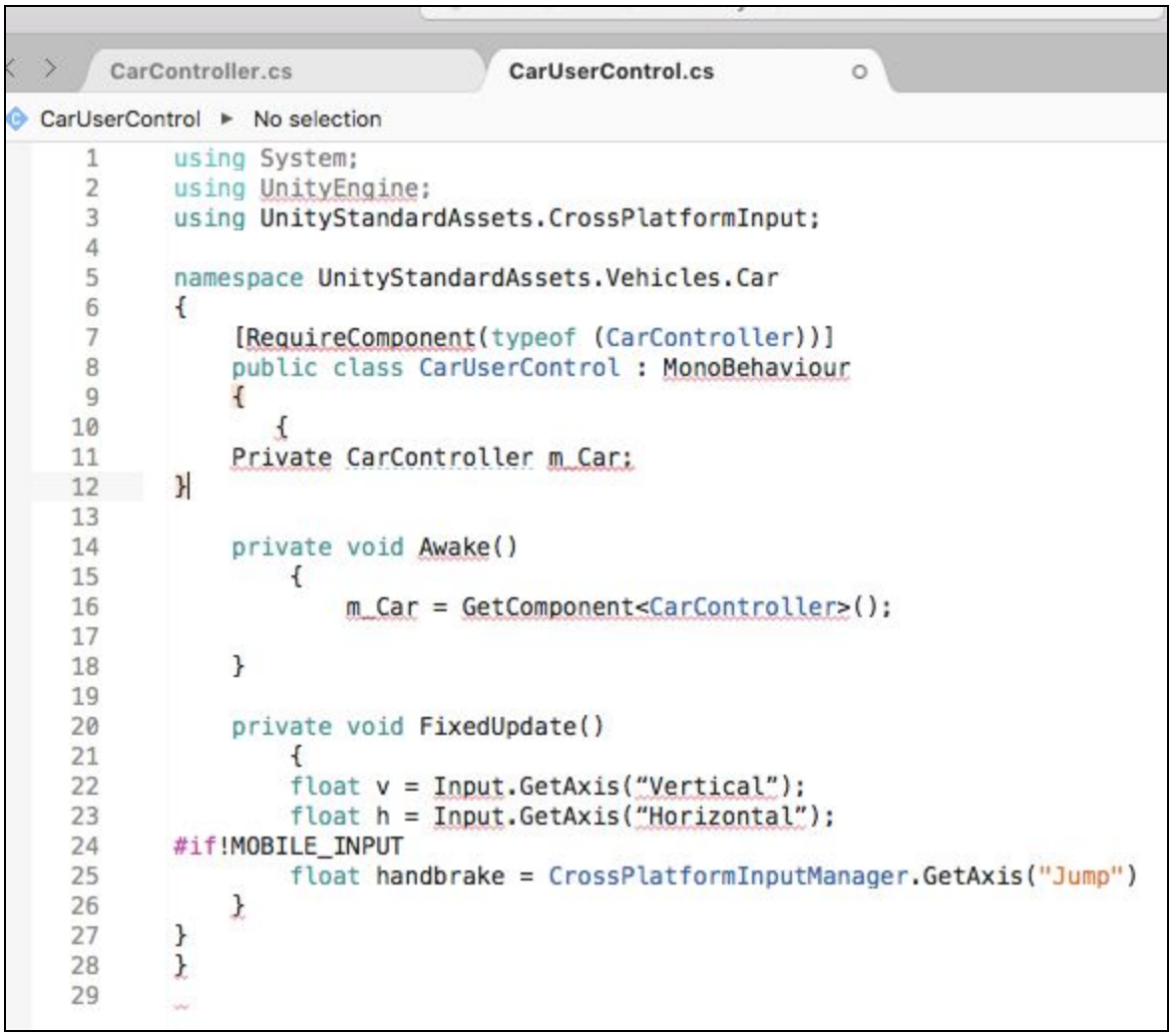
You will also need to alter the controls for the vehicle's handbrakes. Navigate to **Edit > Project Settings > Input > Axes > Jump**. Make sure there is "space" set for the **Positive Button** and nothing in the **Negative Button** area.



SECTION 8: HANDBRAKE

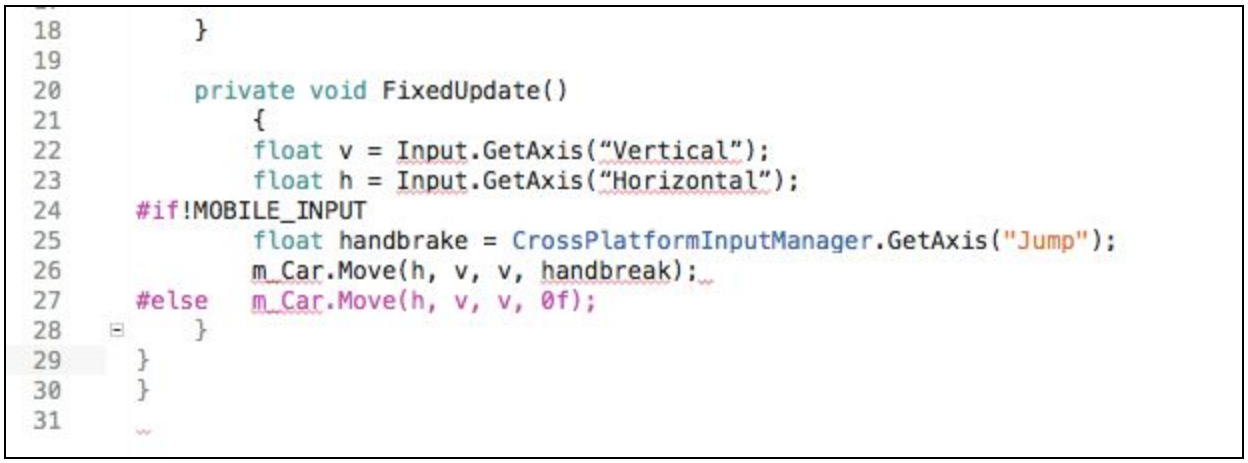
1. Now that we have put a handbrake into the controls, we need to update the code to reflect this. Navigate back to the `CarUserControl.cs` file that we previously edited.

2. There is a piece of code that we need to add that checks whether the game is played on mobile, and then creates a handbrake if the game is played on mobile. This code will be in the `private void FixedUpdate ()` section on line 20 in `CarUserControl.cs`. The final code will appear as follows:



SECTION 9: MOVE

1. In this next section we will add parameters to the `Move()` function. This function is responsible for moving the car models in the game. To start, you should still be in the `CarUserControl.cs` file, directly after the `float handbreak` function. Add the code `m_Car.Move(h,v,v,handbrake)`.
2. Add an `#else` statement in case the `handbreak` function is never initialized, then `m_Car.Move(h,v,v,0f)` The characters `0f` meets the criteria for passing a float value, but will not trigger the handbreak. The final code will appear as follows:



```
18          }
19
20          private void FixedUpdate()
21              {
22              float v = Input.GetAxis("Vertical");
23              float h = Input.GetAxis("Horizontal");
24      #if!MOBILE_INPUT
25              float handbrake = CrossPlatformInputManager.GetAxis("Jump");
26              m_Car.Move(h, v, v, handbreak);
27      #else   m_Car.Move(h, v, v, 0f);
28          }
29      }
30      }
31
```

4. The last part demands ending the section with `#endif.` If you run your code in later steps and this line is not there, you might encounter a continuously spinning cursor because the code is trying to run repeatedly. If this happens, close the program and add `#endif` to your code.

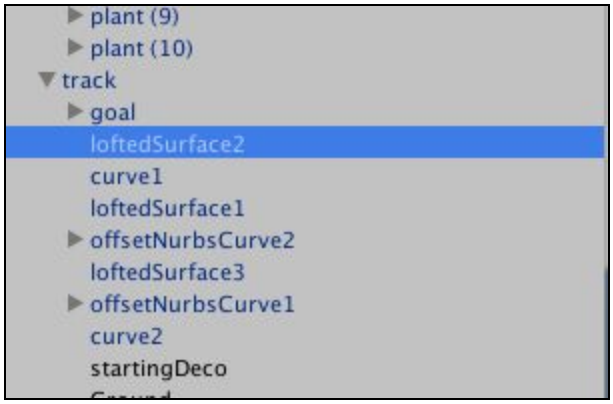
SECTION 10: CREATING COLLIDERS

1. If you are searching for the Inspector panel, you need to first select an Object in your scene. Once an object is selected, this will trigger the **Inspector Panel** to open and there will appear a button that says **Add Component.** Click on **Add Component > Physics > Mesh Collider**. Select **Mesh Collider,** and it will be added to the Inspector **Panel**.

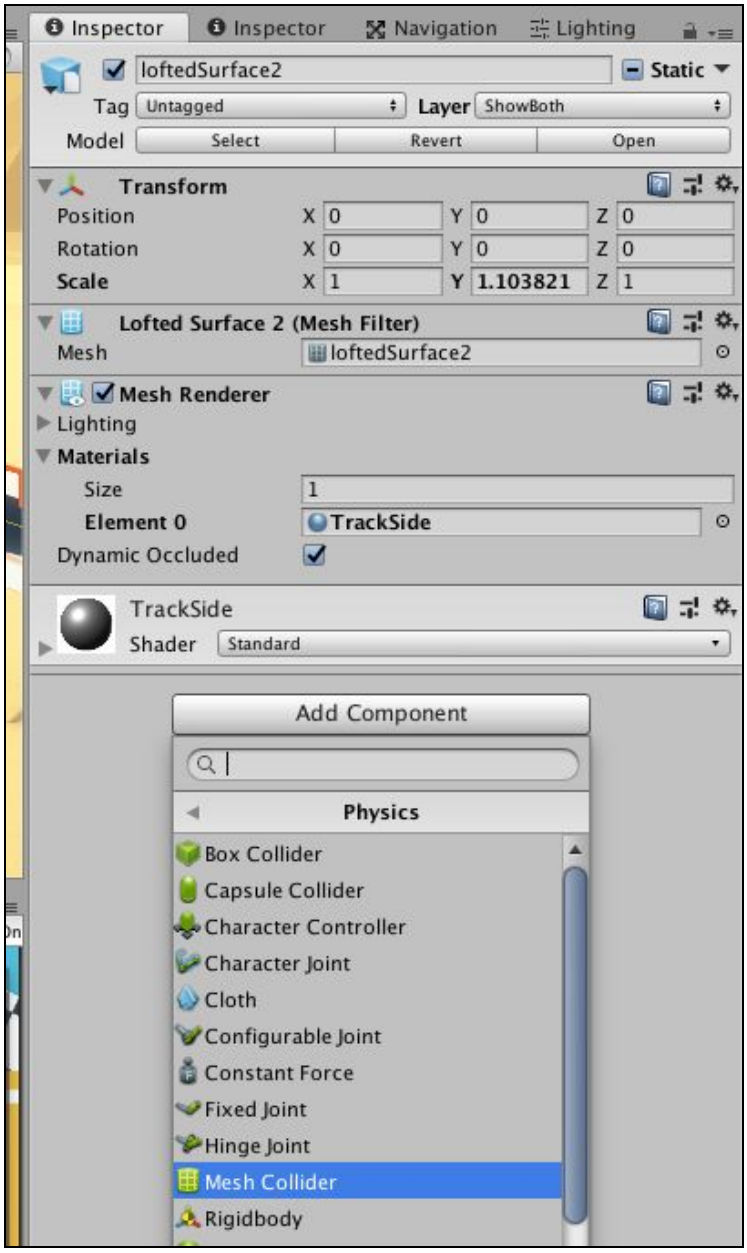
2. You need to create Mesh Colliders for three different kinds of objects:

1. **Road** : GeometryStatic > track > loftedSurface1
2. **Wall 1**: GeometryStatic > track > loftedSurface2
3. **Wall 2**: GeometryStatic > track > loftedSurface3

You can find these objects under **Track** in the left hand **Inspector** panel.

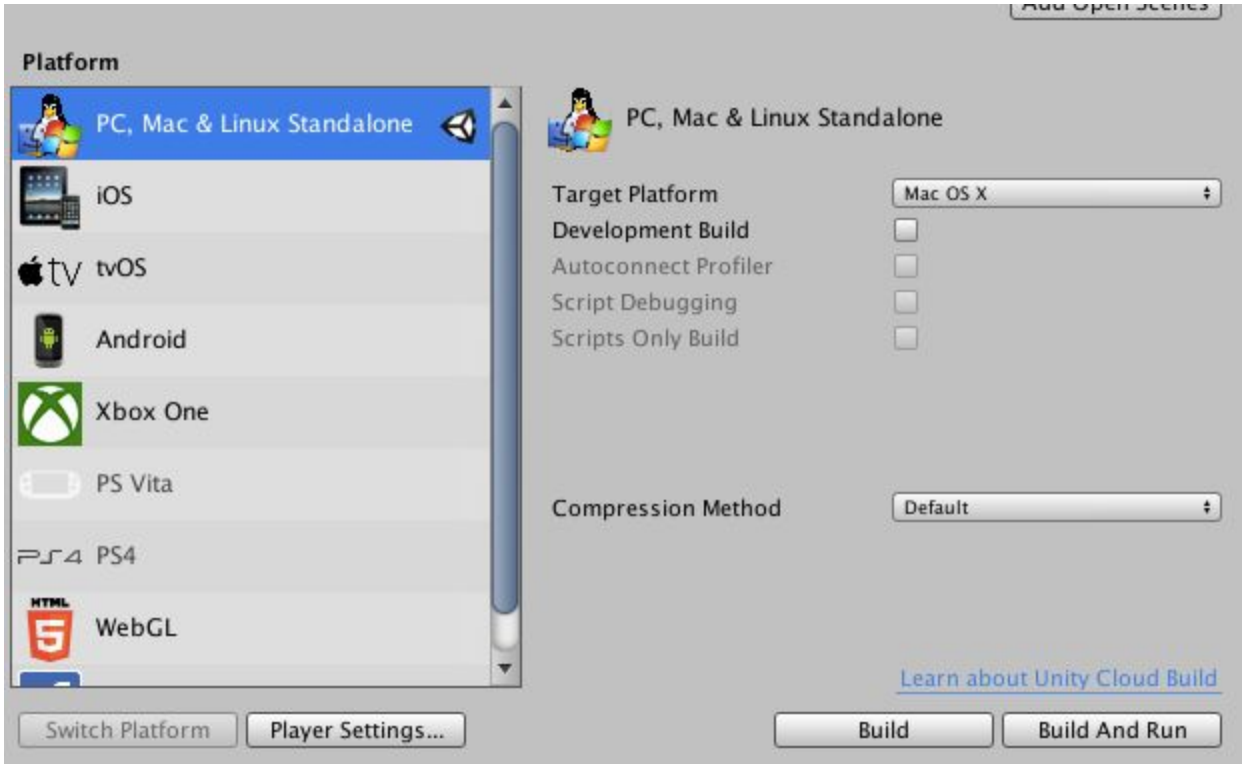


3. If done correctly, you **will** receive an error message. This is because the game now understands that your car should not be located in the wall.



SECTION 11: BUILD PROJECT

1. The final step is to Build the project. To do this, you need to navigate to **File > Settings**. Select Mac as the platform, then **Build.** You will get an error if you try to Build your project on the wrong platform.



2. You might get a screen that says “Missing Project ID.” Select **Yes** and your project will still be created.