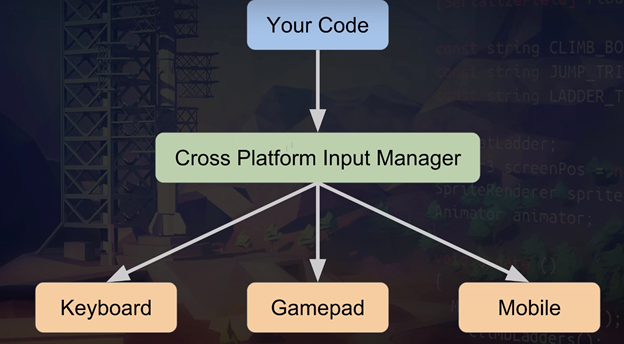
**Cross Platform Input**

**Objective**: Students will know how to explore the cross platform impact manager. Students will understand what happens with the input values when we use the keyboard. Students will be able to use a gamepad and set up for the ability to move the ship.

**Background Info:**

Lets think about cross platform input. In the last section our code directly read from the keyboard. A couple of problems with that: One, is that the player cannot remap those keys and normally in a game you would have some sort of flexibility over remapping your keys and the player would be able to change those things and two we don’t give the use the option to use a gamepad.

Now unity is a cross platform engine it's very good at providing you the ability to deploy your game to multiple platforms. Now obviously you don’t wan tot design a game for various platforms because the gameplay should be specific. ie no one wants to play a mobile game on their Xbox. Here’s where the Cross Platform Input Manager comes into play.



What it does is it lets the code talks to this middle layer which in turn talks to these different inputs and lets the players remap their keys. It also means that you can switch your platform to mobile or you can use a gamepad or you can use keyboard and they should all work with a few caveats.

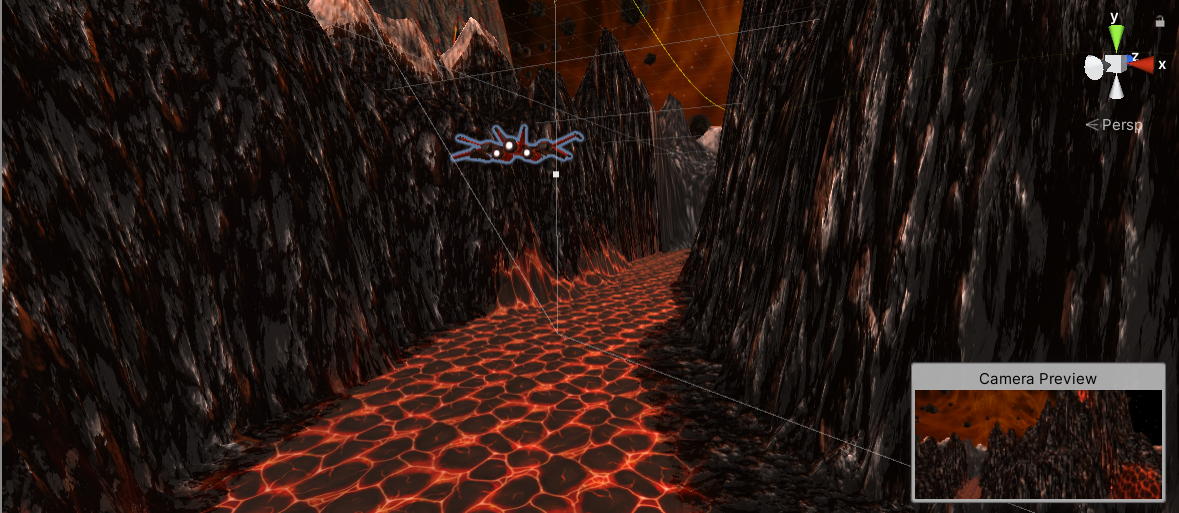
1. Okay lets jump into Unity. First we needs to start thinking about where our **Spaceship** should live in our game? and the preferred place would be as a child of the **Camera**. Since the ship always has to be in view of the camera a few units in front.



1. Double click on the **Camera** and click on the **Game** tab.

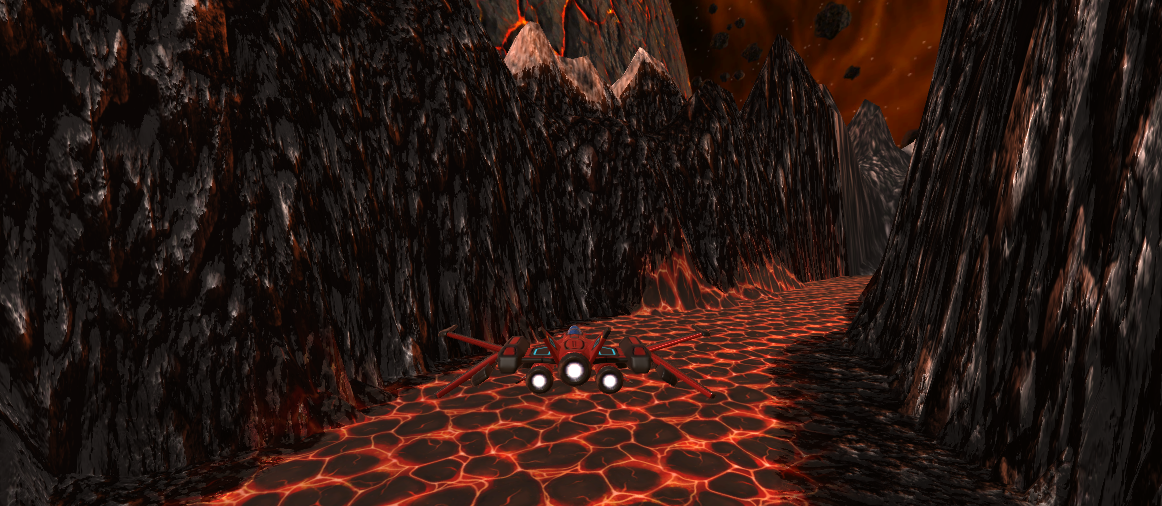


1. Chances are your ship is not currently in view lets go ahead and reset the Spaceships coordinates. so here’s a tip click on the **Scene** tab and place your view somewhere where you want the camera to be. For example I moved mine in between the valley.



1. Click on **GameObject** in the menu and select **Align With View**. Your ship should appear directly in front of the camera. Now you can adjust the ships position and place it an appropriate distance from the camera.

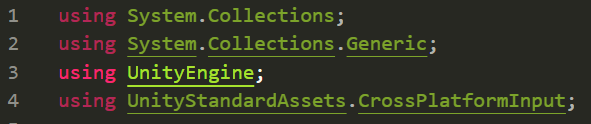


My final view was this 

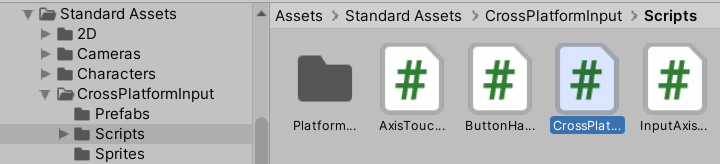
1. Now if you click play you should see the ship move steadily with the camera. This is because it’s a child of the Camera object.

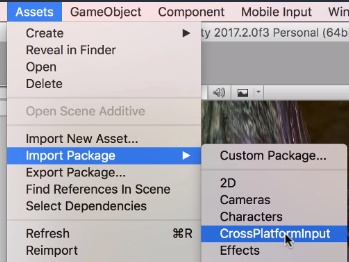
**Moving the ship**

1. Click on the player ship and **add component**. Add a new script and name it **Player**. Open this new script and under the imports add the cross platform input libray.

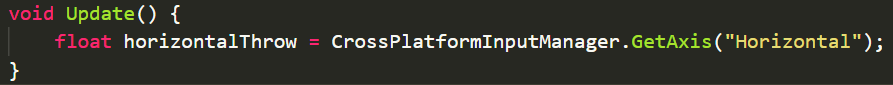


Now this only works provided that in the previous lecture you imported the **standard assets** and with your standard assets one of the folders you should have is **cross platform input** and inside of that is the **Scripts** folder is something called C**ross platform input manager**.

 If for whatever reason you don’t have this go to the menu and select **Assets** > **Import** **Package** > **CrossPlatformInput**



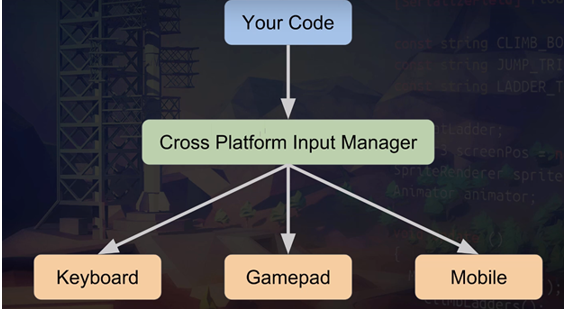
1. Back in your **Player**.**cs** file lets go into the Update method and type



GetAxis(“Horizontal”) - This cross platform input manager is going to say “I'm going to refer to everything by an axis”. Now if you use the A and the D key by default then they are the horizontal axis.

Go back to Unity. Go to the menu and click on **Edit**> **Project** **Settings** > **Input Manager**

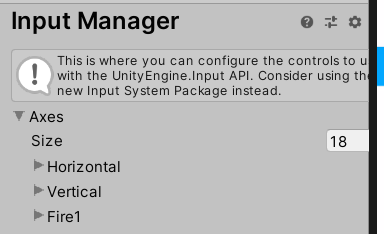
This is the key to the middle layer.



We can either interface with this through code



or on the editors side:



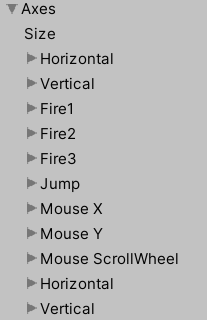
The point is that THIS string Horizontal



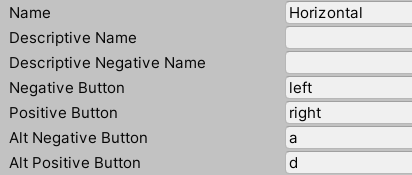
is targeting THIS Horizontal



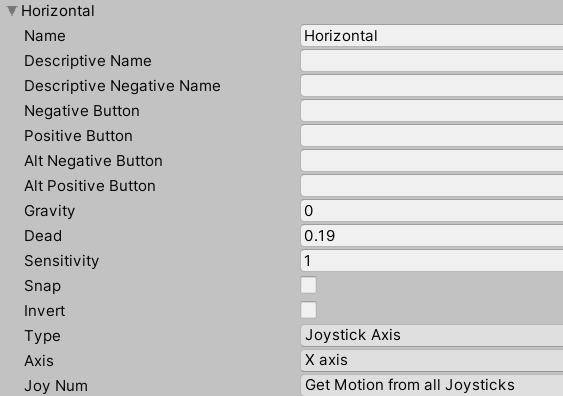
If you’ve also noticed there’s 2 Horizontal options.



The first one says the in order to move negative direction in the horizontal plane you need to either click **left** or **a.**



The second Horizontal option which deals with the joystick x axis.

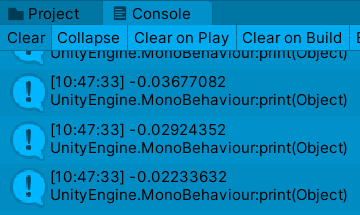


We’ll explore these settings in depth later right now I just wanted you to be exposed to them. Lets go back to the Unity Script.

1. Why did we name the variable **horizontalThrow**?? If you think of a gamepad what I call throw is if the gamepad joystick is central then there is no throw. If the gamepad stick is all the way to one side then there's a throw of one (100%) and if the stick goes to the other side then it would have a throw of -1 ( -100%). so lets go ahead and print out the horizontalThrow value and test it out.



Remember after you click play to make sure the console is showing at the bottom and click on your game window for your keyboard inputs to be read. If done correctly you should get logs after pressing A or D on your keyboard.



What unity does when you hit a key is the game can respond with keyboards similarly to a joystick. What it does is it builds up the value from 0 to 1 over a period of time.

If your gamepad is connected correctly you should be able to do this with the joystick too!