Private & Confidential

Unity Game Engine Camera System

LS Created by Lawrence Selly (Unlicensed)
Mar 14, 2018 • 5 min read

A guide to installing, running, and adding your own settings to the igloo 360 unity camera system.

Prerequisites

Before staring you must have completed the following steps.

1. Downloaded the current Igloo Camera package from Igloo Vision.
2. You should have unity installed, and updated (to whichever camera package you require)
3. Extract the igloo camera package to somewhere accessible, such as your desktop.

Installing the package

It is recommended that you first import the camera package into a new Unity project, this helps eliminate potential issues during installation. To import the camera package from within the Unity Editor go to *Assets > Import Package > Custom Package* and open the file with a '.unitypackage' extension included in the zip folder.

Configure Project Settings

Go to *Edit > Project Settings > Player* and apply the following changes

- Default is Fullscreen = False
- Run In Background = True
- Display Resolution Dialog = Enabled

The Igloo camera package requires some custom Input settings to be compatible with an Xbox controller. This can be done one of two ways;

> ⓘ **Note - The next step is important, you will encounter errors in play mode if you do not apply the following settings.**

1. If you are importing the camera package to a new Unity project then you can replace the InputManager.asset file found in *YourProjectDirectory > ProjectSettings > InputManager.asset* with the one included in the camera package

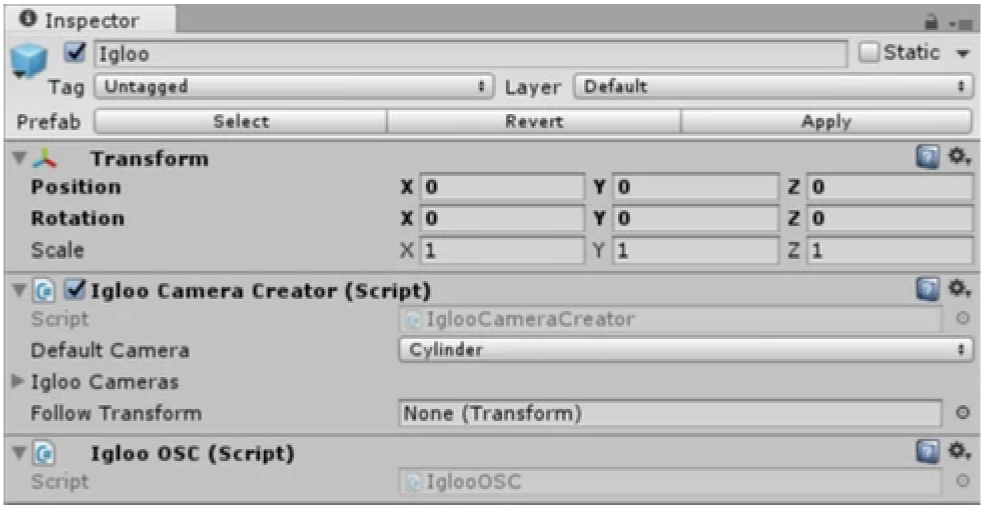
the one included in the camera package.

2. If you are importing the camera package to an existing project you may need to manually add the required input axes. If you have not made any changes to the Input manager you can simply replace the InputManager.asset file as described above. Alternatively you should create a new Unity project and replace the InputManager.asset file as described above, then copy the Input Manager settings found in *Edit* > *Project Settings* > *Input* to your project.
3. You must change the

Scene Setup

For an example of how to set up a scene to be compatible with an Igloo, open the demo scene found in *IglooTools* > *Scenes* > *DynamicCameraCreation*
There are two main prefabs **Igloo** and **IglooPlayer**. Which can be found in *IglooTools* > *Prefabs*

> **Igloo** - Must be included in the scene, this prefab is used to generate a 360 software camera which sends the output to the Igloo Warping application.



Inspector

☑ Igloo □ Static

Tag Untagged Layer Default

Prefab Select Revert Apply

Transform

Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

☑ **Igloo Camera Creator (Script)**

Script IglooCameraCreator
Default Camera Cylinder
Igloo Cameras
Follow Transform None (Transform)

Igloo OSC (Script)

Script IglooOSC

Default Camera	
Cylinder	Should be used for an Igloo Cylinder or Dome structure
Cube	Should be used for an Igloo Cube structure
Unwrap	This is a new type of camera system that renders to a single large texture, for 5 camera systems
Cylinder6	This is a special camera rig for small cylinder applications that run 6 projectors
Cube3	This is a take on the unwrap camera, but for a 3 screen system for inside a cube.

> *Note - The default camera can be overridden once the application has been built by specifying the following command line argument settings in a shortcut of the .exe '-iglooCylinderCam' '-iglooCubeCam'.*

The Igloo Camera can also be created and destroyed via code using the public functions defined in the *IglooCameraCreator.cs* script.

CreateCamera(int index) - Takes one parameter of type int which specifies which camera type will be created, if an Igloo camera already exists in the scene it will be automatically be destroyed before a new one is created.

DestroyCamera() - Takes no parameters and destroys the Igloo camera if one exists in the scene

Follow Transform

You can assign a Game Object for the Igloo Camera to follow. If you are using the IglooPlayer prefab it will be followed automatically and no GameObject needs to be assigned.

IglooPlayer - You don't need the custom Igloo controller but it makes navigation within the igloo much much easier as can be seen in this video.

This works using a custom Xbox controller with a gyroscope attached to it, which provides orientation data via OSC, so the direction the user is facing in the Igloo is known. Using the custom Xbox controller creates a natural pointing device on the screen so that when you press forward on the xbox controller you move in the direction you are physically facing.

The *IglooFirstPersonController* component attached to the **IglooPlayer** prefab has a number of control methods depending on which hardware controller is being used (Keyboard Mouse, GyrOSC, Igloo Custom Xbox) these can be specified using the drop down menus under the 'Igloo Settings' section of the parameters.

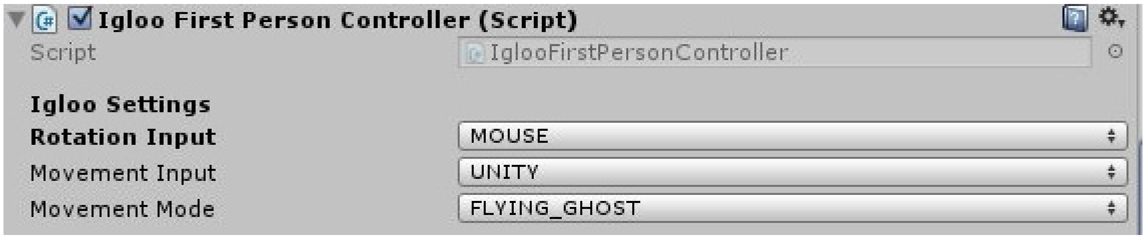
> ⓘ For testing moving the player within the Editor you can use either the **Keyboard and Mouse** or **GyrOSC**.

Keyboard and Mouse

Use the following settings on Igloo *IglooFirstPersonController*

Rotation Input - MOUSE

Movement Input - UNITY



Igloo First Person Controller (Script)

Script	IglooFirstPersonController
Igloo Settings	
Rotation Input	MOUSE
Movement Input	UNITY
Movement Mode	FLYING_GHOST

GyrOSC

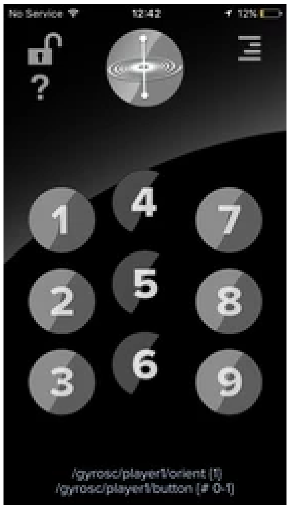
Use the following settings on *IglooFirstPersonController*

Rotation Input - GYROSC

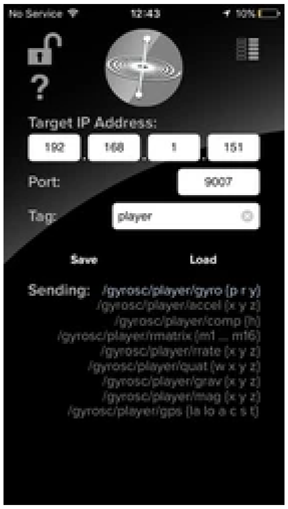
Movement Input - GYROSC

For testing purposes we use an iOS app called GyrOSC which allows us to steam gyroscope data from a mobile device to Unity over a network, here's how

- Download the GyrOSC (iOS) app.
- Join the same wifi network on your computer running unity and your phone running GyrOSC
- Open GyrOSC and copy the settings below, replacing Target IP Address with the IP address of the computer running Unity.
- When Unity is running the numerical buttons control movement {1- Forward, 2- Backward, 3- Left, 9- Right}



/gyrosc/player1/orient (i)
/gyrosc/player1/button (# 0-1)



Target IP Address: 192 . 168 . 1 . 151

Port: 9007

Tag: player

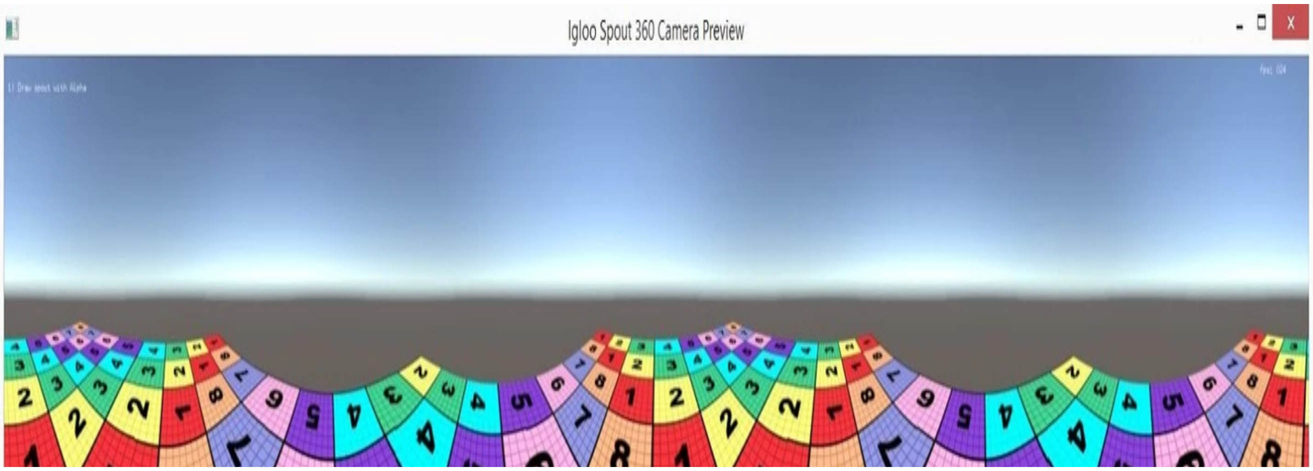
Sending: /gyrosc/player/gyro (p r y)
/gyrosc/player/accel (x y z)
/gyrosc/player/comp (h)
/gyrosc/player/rmatrix (m1 ... m16)
/gyrosc/player/rrate (x y z)
/gyrosc/player/quat (w x y z)
/gyrosc/player/grav (x y z)
/gyrosc/player/mag (x y z)
/gyrosc/player/gps (la lo a c s t)

Building the project

When building the project for the first time, follow the following procedure.

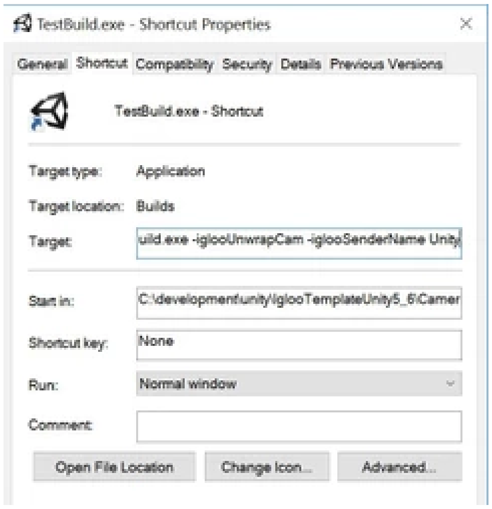
Run Scene

Included in the package is an app for previewing the output of the 360 Camera, *Igloo Preview.* When running your Unity scene you should see the 360 degree view being displayed in the Igloo Preview app, similar to the image below.



Command Line Arguments

To use command line arguments, create a shortcut of the built .exe, then right click > properties then add any arguments to the 'Target' text field.



Target type: Application

Target location: Builds

Target: uild.exe -iglooUnwrapCam -iglooSenderName Unity

Start in: C:\development\unity\iglooTemplateUnity5_6\Camer

Shortcut key: None

Run: Normal window

Comment:

The following command line arguments can be used

argument	use
-igloo3DEnabled	Enables 3D Mode on start
-iglooSpoutReset	Enables sending a reset message '/spout/reset' via OSC on creation to port 9001
-iglooSenderName' [NAME]	Where the following [NAME] argument specifies the spout sharing name
-iglooCameraType [NAME]	Where the following [NAME] argument specifies the type of camera and must be one of the following Cylinder Cube Unwrap Cylinder6

OSC Protocol

Player

address	use

address	use
/gameEngine/rotationInput/mouse	set rotation mouse
/gameEngine/rotationInput/gyrOSC	set rotation GyrOSC
/gameEngine/rotationInput/XIMU	set rotation XIMU
/gameEngine/movementInput/unity	set movement Unity (Standard)
/gameEngine/movementInput/gyrOSC	set movement GyrOSC
/gameEngine/movementMode/walking	set movement Mode walking
/gameEngine/movementMode/flying	set movement Mode flying
/gameEngine/movementMode/ghost	set movement Mode ghost
/ximu/euler	Three float parameters with gyro data [0] - pitch , [1] - roll , [2] - heading

Camera

address	use
/gameEngine/setCameraType	0: Cylinder, 1:Cube, 2:Unwrap, 3:Cylinder6
/spout/reset	
/gameEngine/camera/standardFOV	set FOV standard
/gameEngine/camera/wideFOV	set FOV wide
/gameEngine/camera/superWideFOV	set FOV superwide
/gameEngine/3D/separation	set 3D separation
/gameEngine/3D/lensShift	set 3D lensShift
/gameEngine/3D/enabled	set 3D enabled
/gameEngine/3D/invert	set 3D invert
/gameEngine/cameraTilting	set camera tilting On/Off
/gameEngine/camera/farClipDistance	set Far Clipping distance

System

address	use
/unity/quit	quit Unity
/unity/sleep	set sleep On/Off
/gameEngine/getSourceInfo	

address use

Be the first to add a reaction