Accelerometer Based Control Asset Package

This unity asset is prepared to use accelerometer as an input controller with a game. You can calibrate on the fly while Pitch, Yaw and Roll (x, y, z) angles are given within the configurations. The prefab of "Accelerometer Based Control" gives you direct access to angles with calibration. The usage of the prefab is given in 8 sample game scenes. First 6 scenes are related with aircraft control, whereas in scenes 7 and 8, a classic labyrinth game is given in two different styles.

What is new in Version 1.2:

- Sample scenes added;
  - Simple Free Fly Scene with realistic control and dynamic value change
  - Simple Free Fly Scene with action control and dynamic value change
  - Detailed Free Fly Scene with realistic control and dynamic value change
  - Detailed Free Fly Scene with action control and dynamic value change
  - Free Fly Sample Game
  - Corridor Fly Sample Game
  - Rotate Labyrinth Sample Game
  - Roll a Ball Sample Game
- Code improvement by performance and readability.
- Made simple to use (drag / drop style) accelerometer based control (as prefab)

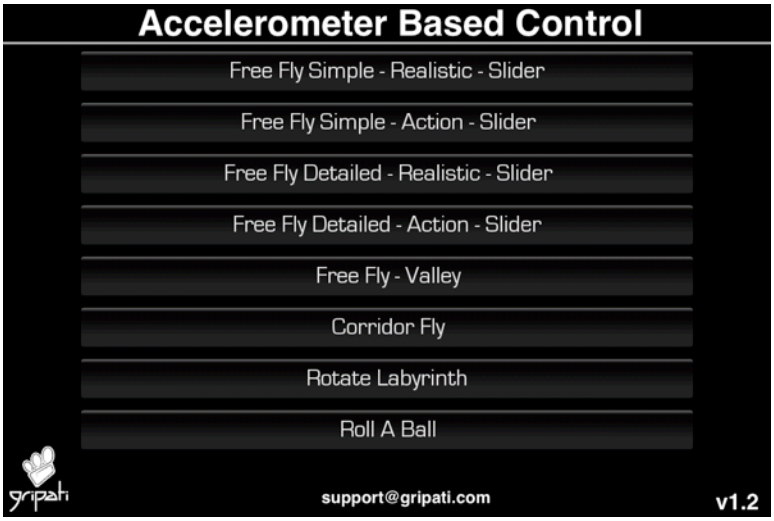
~~What is new in Version 1.1: Added Free Fly Scene with just a basic aircraft, terrain and core accelerometer based control script. This Scene is prepared according to help developers to use the accelerometer based control script after your own configuration.~~

How To Use Package (v1.2)

1.Import the Package (If there are errors about MotionBlur or ImageEffectBase: See Motion Blur Error Solution below)

2.Open Scenes > 0 MainScene. The initial game scene should bee seen as given below.

Note that: Here, the Screen resolution is iPhone 4G Wide (960x640)



3. Hit Play button using Unity Remote. (Check it from here)

If there are errors about MotionBlur or ImageEffectBase

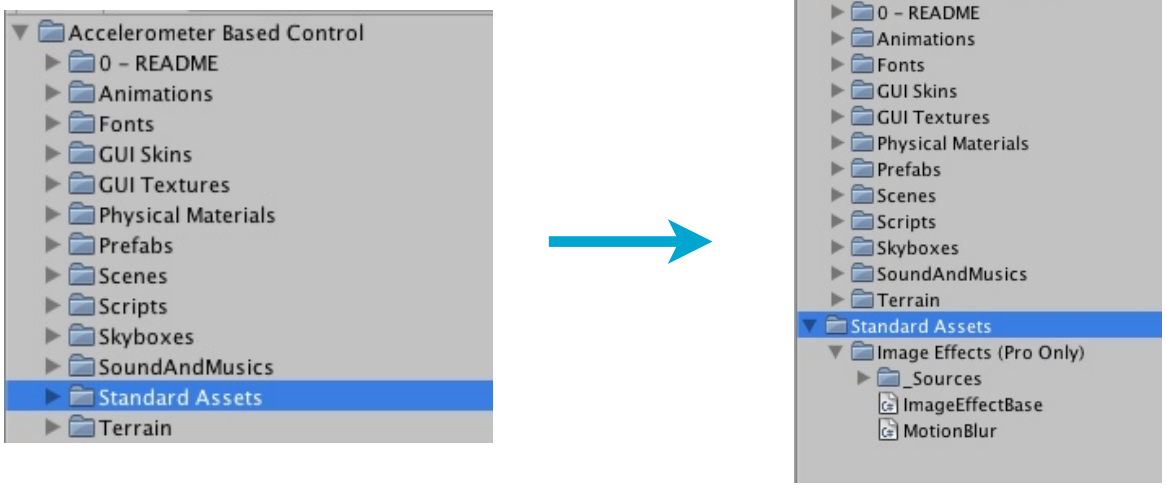
Import some Standard Assets (Assets -> Import Package -> Image Effects)

Image Effects - Motion Blur

Particles - Fire - Fire1

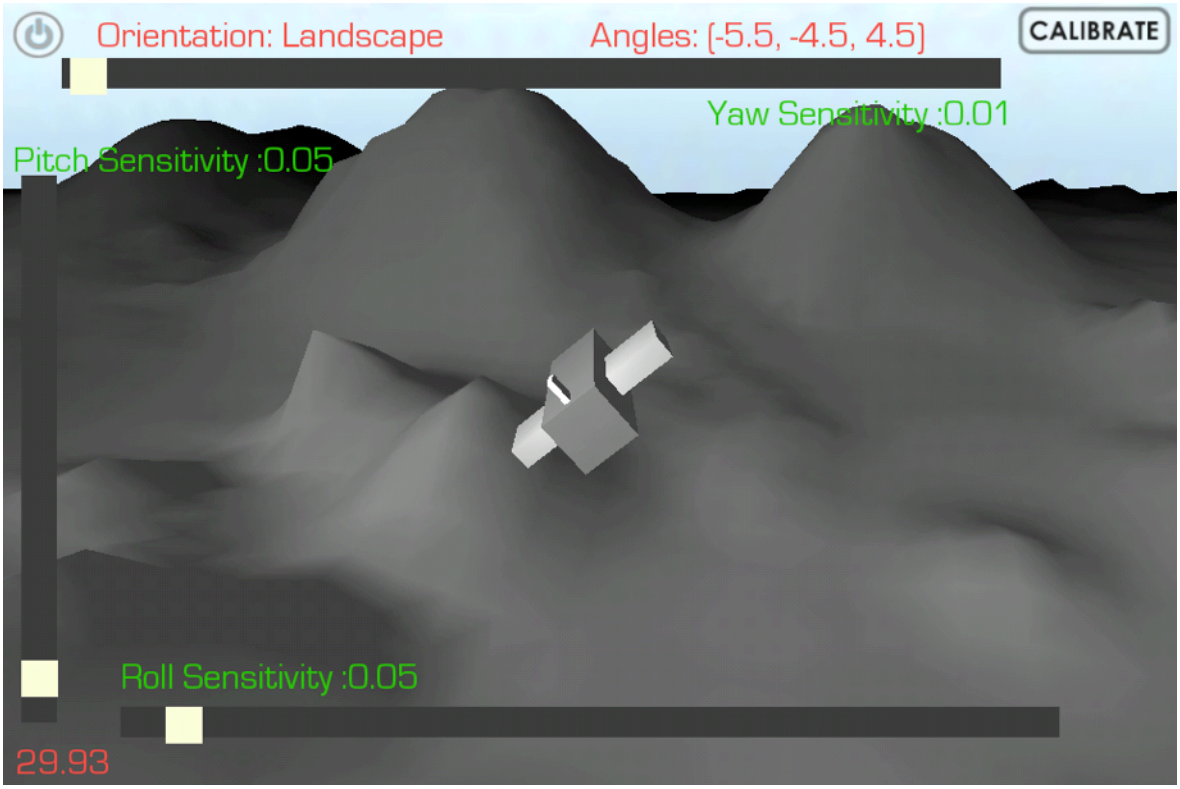
OR

Move the "Standard Assets " Folder outside the Game Folder



Scene1 - Free Fly Simple - Realistic - Slider

You can see the configuration parameters with initial values. Here, you can control your simple aircraft with Unity Remote. In this scene, you can change Yaw, Pitch and Roll angle sensitivity. In addition, you can calibrate according to your phone holding and you can change the aircraft control system.



Scene 2 3 4- Free Fly Simple & Detailed - Slider

There are two mode on Free Fly: Realistic and Action. In realistic mode, the turning angles are added to aircraft as cumulative. This mode (realistic) is almost the same with real aircrafts; if you rotate your phone the aircraft is started turning and if you stay the same position, the aircraft is turning continuously. However, in Action control the aircraft is rotating with your phone but not cumulative; if you rotate your phone the aircraft is started turning and if you stay the same position, the aircraft stays the same position.

In Scene2, you can use the aircraft with the action mode with same configurations with the scene1.

In Scene3 and Scene4, the terrain and the aircraft is detailed, but the configurations are the same with Scene 1 and Scene2. An example screenshot is given below from Scene3.

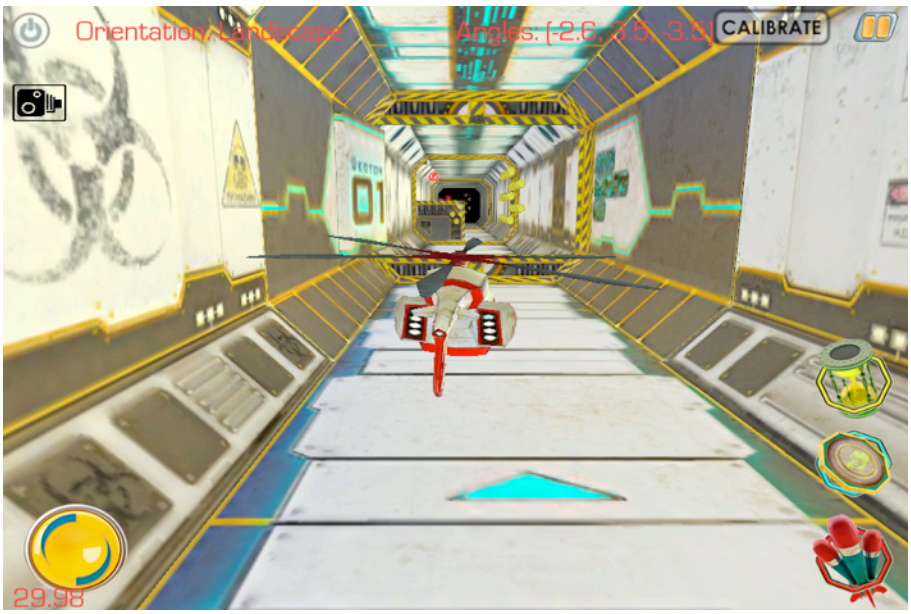
Scene5- Free Fly Game Sample

In Scene5, you can fly in a valley with an aircraft using accelerometer with action based control. This scene is an sample usage of the free fly using accelerometer controller. The screen-shoot is given below;

Scene 6 - Corridor Fly Sample Scene

You are controlling an aircraft and trying to avoid crash to any obstacle. In this example game, you can use bonus materials as many as you can. In addition, in this version; the collected coins are not seen or working for something else.

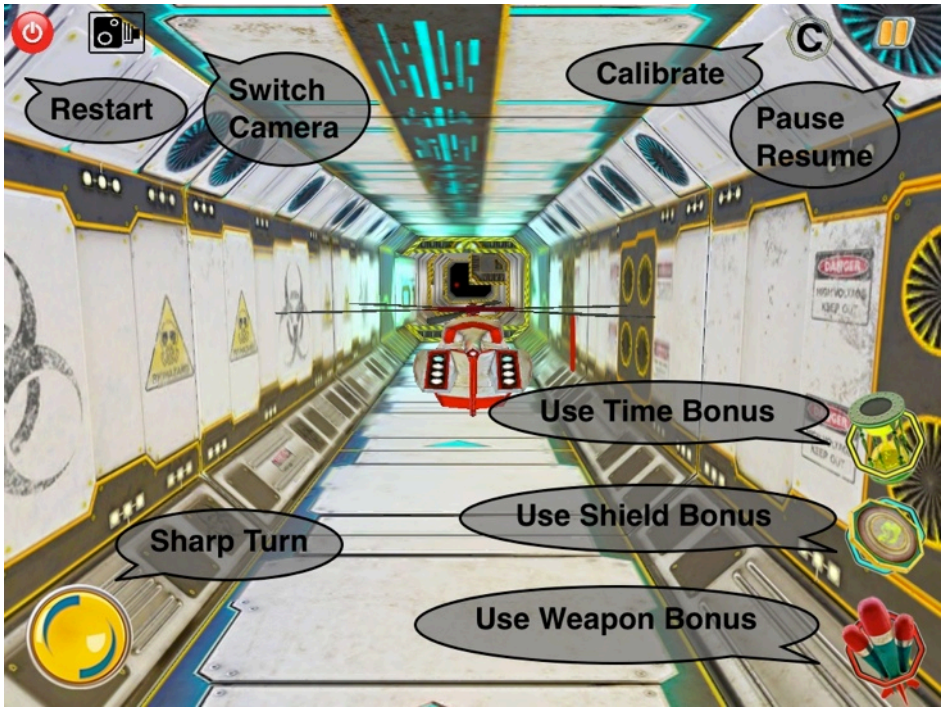
The main logic is given about the game, not the whole. However, you can easily change the game dynamics and use it.



After building and running the game on your iOS device, you can "Calibrate" the accelerometer according to your device position. Then you can easily play the game.

The main difference from other systems is; the accelerometer data is converted to "Pitch", "Roll", "Yaw" angle as smooth as possible using academic researches. The conversion makes the movements as like "Gyroscope" but better! Why? Because if your movement is fast, then the motion is also fast (because it is depend on acceleration of the device).

There are three bonus items in the corridor;
- Weapon : Burst next obstacle! Each aircraft has its own weapon.
- Shield : You don't need to worry about obstacles. Limited time only.
- Time: Slow down everything except for your aircraft. Escape easily and pickup more... Limited time only.



How To Play (see: imageGameWithDescription):
Move your device to control the aircraft. Note: If it is not controlling via your device, then make GameController (in Hierarchy view) -> ControllerXCode -> isDebugMode : false (Otherwise, it waits a keyboard input)

"C" button: To calibrate accelerometer according to your device's current position.

"IO" button: To restart the game.

Bonus buttons: To use bonus items. There are three bonus items : Weapon, Time, Shield.

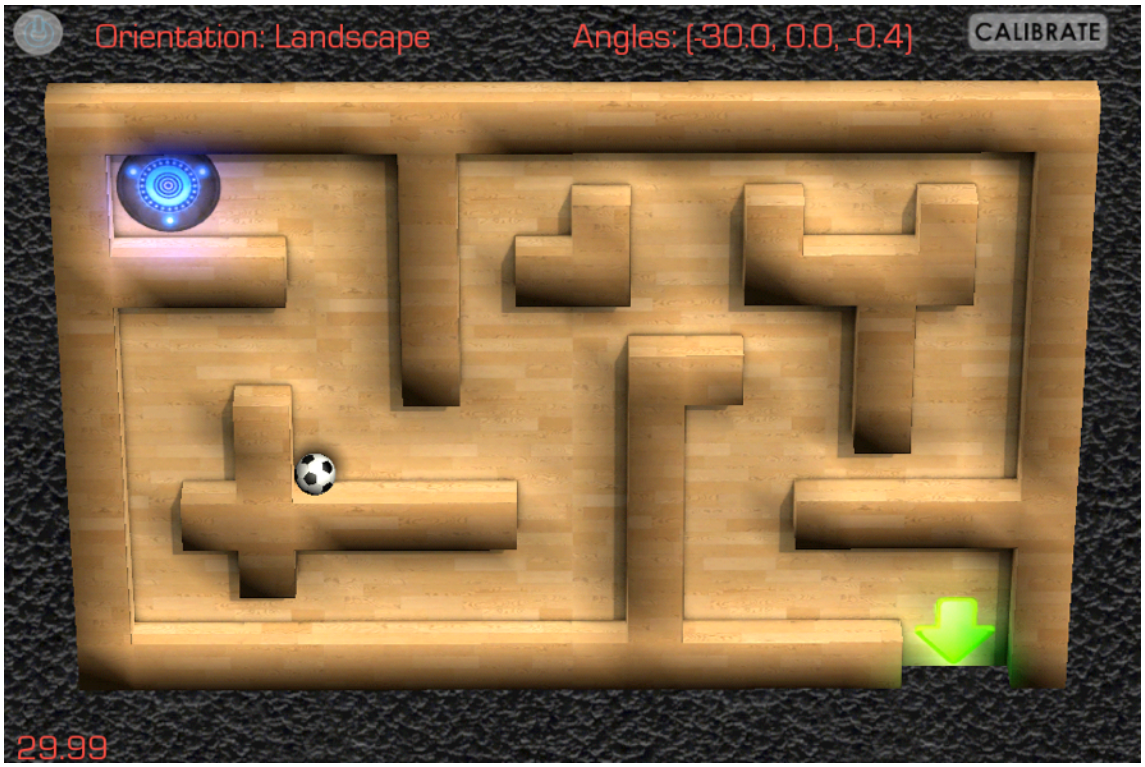
Turn Button: You can make a sharp turn using this button.

Camera Button: There are two camera options (Standard, Action). You switch your camera option on the fly!

To Change Aircraft2 to Aircraft4
GameController (in Hierarchy view) -> ControllerXCode -> Initial Aircraft -> Aircarft4 (There are 2 aircraft to use Aircraft2 and Aircraft4)

Scene 7 - Rotate Labyrinth

In this example scene, you can play a classic labyrinth game. However, here you are controlling the rotation of the whole labyrinth and the ball is moving according to rotation of the labyrinth. Your aim is to reach the green area. Whenever reach there, you will restart from the blue light (starting point).

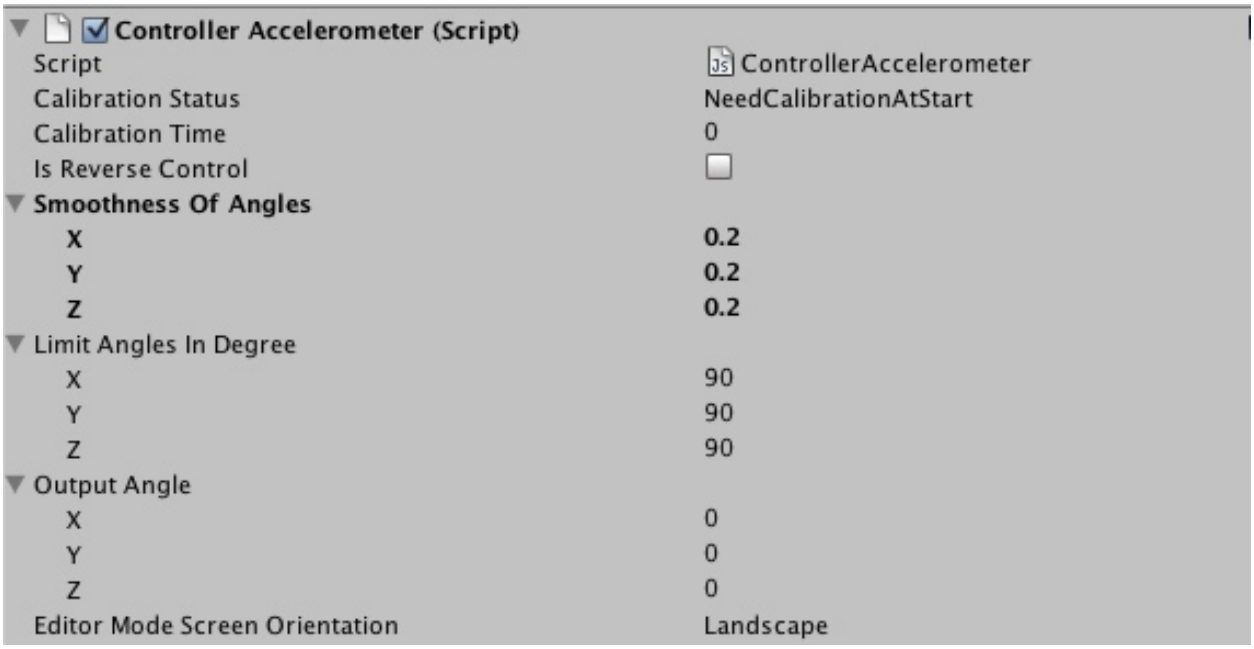


Scene 8 - Roll A Ball

In this example scene, you can play a classic labyrinth game. However, here you are controlling the torque (also force) of the ball. Your aim is to reach the green area. Whenever reach there, you will restart from the blue light (starting point).

Detail of Script:

Controller Accelerometer;



This is the main script where all other player controller takes the values from. It takes acceleration data as an input according to device orientation and gives the output angles with the configuration.

Calibration Status :
It defines the calibration status as 'NeedCalibrationAtStart', 'NeedCalibration', 'InCalibrating' or 'Calibrated'.

Calibration Time :
When you set CalibrationStatus as NeedCalibration flag true, the calibration offset values are started calculating. The current accelerometer value is averaged over this time. This is used for handling some error condition caused by user hand shaking.

Example:

If calibration time is 0, then current accelerometer value is calculated as calibration offset (not averaged).

If calibration time is 2, then accelerometer values are averaged with 3 Frames as calibration offset.

Is Reverse Control

This flag determines the reverse control. Sometimes, users can't control with default setting: if you hold down the phone, the aircraft position becomes down. However, in reverse control, if you hold down the phone, the aircraft position becomes up.

Smoothness of Angles

The acceleration raw data is smoothing according to smoothness value. Within this value, user hand shaking can be visible on aircraft or not. The value should be [0 , 1].

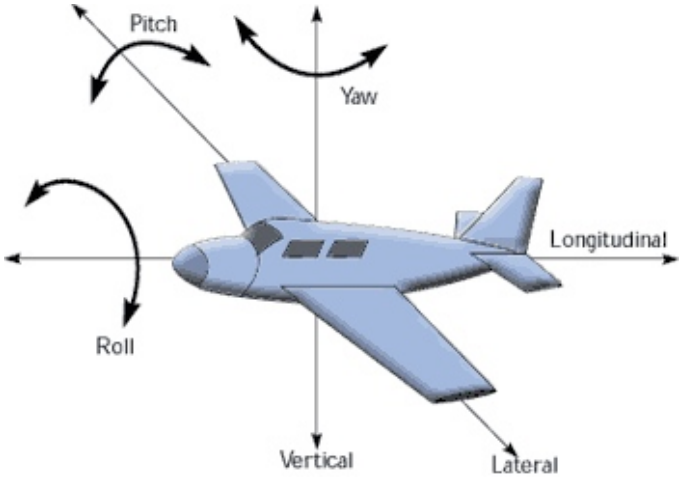
Limit Angles

The acceleration output angles can be limited in [-180,+180] degree.

Output Angle

The acceleration output angles. The angles in axis and the names are;

X = Pitch, Y = Yaw, Z = Roll



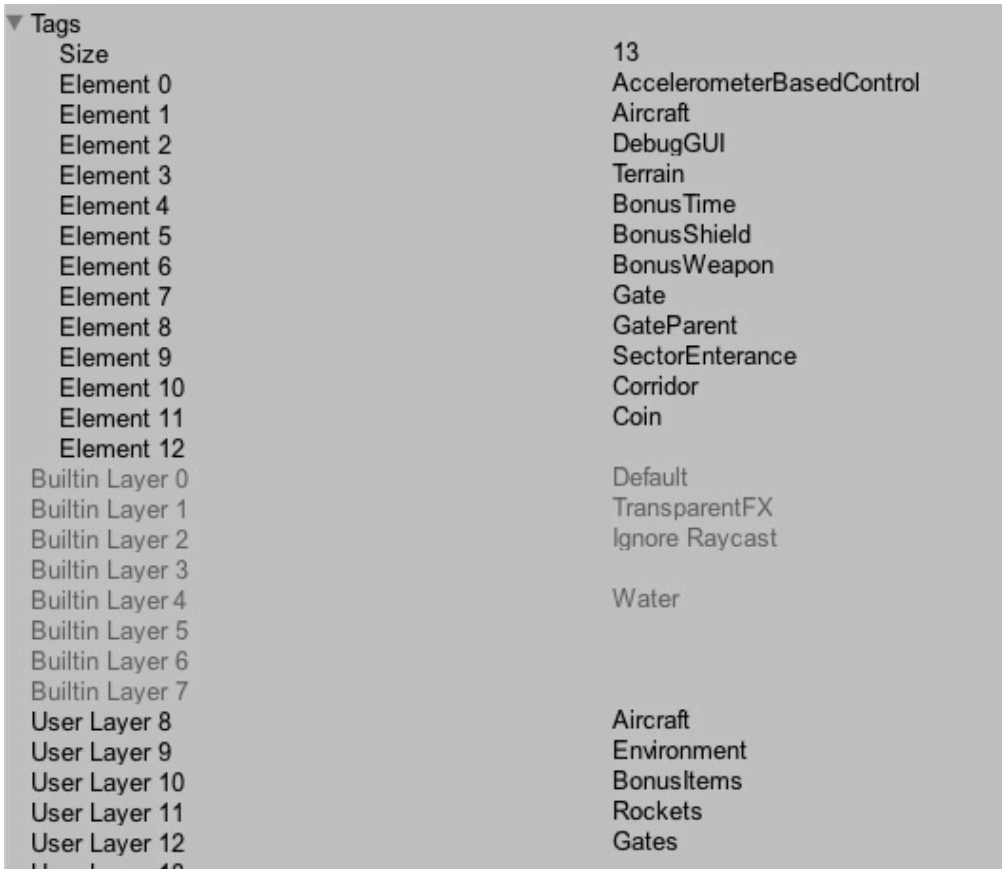
Editor Mode Screen Orientation

If you are debugging the code using Unity Remote

You can change the orientation of the user home. This value is changing the calculation of the accelerometer data. If your game is playing via portrait you can set it portrait, otherwise choose the right landscape value.

About Layers And Tags

In Unity there may be some errors or problems with layers (exp: layer names or tags can not be seen). In order to avoid this problem, the given tags and layers are used. If you can't see the tag and layer names just renamed it as given the image below.



About "Corridor Fly"

This package has one scene only and it contains an important part of a iOS game "Corridor Fly".

You can check Details of the iPhone / iPad / Android game "Corridor Fly":

Check the website and watch the trailer: http://www.corridorfly.com

Contact Information

You can send email to "support@gripati.com" regarding to question about this Asset Store Package with the topic name "Asset Store: Accelerometer Based Control".

We hope you like and enjoy your aircraft control system.

Kind Regards,

Developer Group of Gripati Digital Entertainment