

EASY ACHIEVEMENT SYSTEM

DEVELOPED FOR UNITY3D BY BLAKE KENDRICK



Version 1.0

CONTENTS

Introduction..... 3

Features	3
Getting Started	3
Importing The Package	3
Folder Breakdown.....	4
Object Types	4
GUI Camera.....	4
Achievement Structure	4
Achievement Container/Component	4
Achievement Popup	5
Achievement Helper	6
Achievement Manager	6
Samples.....	6
Unlocking Achievements	6
Viewing Achievements	6
Tutorials.....	7
Tutorial – Set Up	7
Tutorial – Creating An Achievement.....	9
Incremental Achievements.....	9
Tutorial – Unlock an Achievement.....	10
Tutorial – Incrementing An Achievement.....	10
Tutorial – Locking An Achievement	11
Tutorial Displaying An Achievement.....	11
Tutorial – Create An Achievement Viewer	13
Additional Notes.....	14
Resources.....	14
Formats.....	14
Special Thanks	14
Contact	15

INTRODUCTION

This tool is designed to allow for quick and easy implementation of an achievements system within a game.

Using this tool will allow you to have achievements in your game in a matter of minutes and takes care of all the back end management without you having to do more than specify a name, description and a couple of images.

FEATURES

- Robust achievement system that allows an unlimited number of achievements.
- Static achievements.
- Incremental achievements.
- Simple API calls to unlock/lock/maintain achievements.
- Script included to use in conjunction with the Google Play Services plugin.

GETTING STARTED

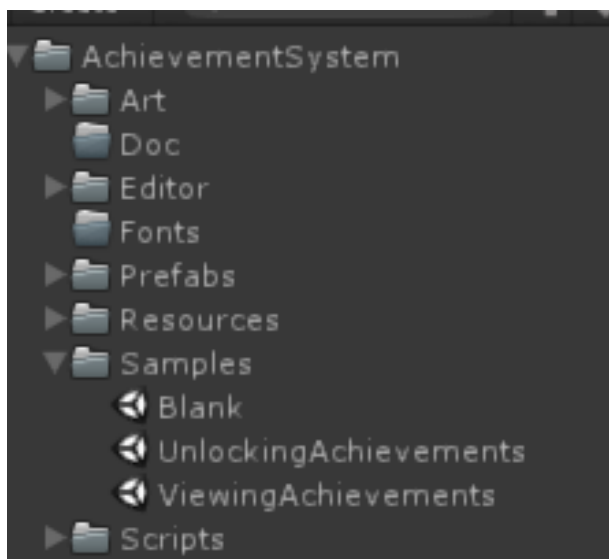
The first thing you will want to do is import the package into the current project you are working on.

IMPORTING THE PACKAGE

If not automatically imported when selecting from the Asset store follow the steps below.

1. From unity select **Assets>Import Package>Custom Package.**
2. Navigate to where you downloaded the .unitypackage file.
3. Click **Open.**

You should now have imported the package and the project tab should look something like below:



Now you have imported the package feel free to look around.

FOLDER BREAKDOWN

Documentation – Contains the documentation that you are reading now, plus a changelog (in future updates).

Editor – This folder contains the various helpers and editor extensions.

Fonts – A selection of fonts that are used in the samples, can be deleted or replaced as needed.

Prefabs – These are some ready to go assets; drag them into the hierarchy to try them.

Resources – Resources that are required for the system to work when running, most of these are fail safes in case you miss something. Ensures it always works.

Samples – In here are the default sample scenes, you can delete these, but it is worth looking at them to get a better understanding of the system.

Scripts – All the scripts used in the system. Feel free to look around these and edit them where you need to, I have commented the majority of the code, but you shouldn't need to edit much.

OBJECT TYPES

GUI CAMERA

This can be found in the Prefabs under AchievementCamera. This is a ready to go GUI camera solution that has all the required scripts already attached to start implementing Achievements.

THIS PREFAB NEEDS TO BE IN ALL SCENES WHERE ACHIEVEMENTS CAN APPEAR!!!

ACHIEVEMENT STRUCTURE

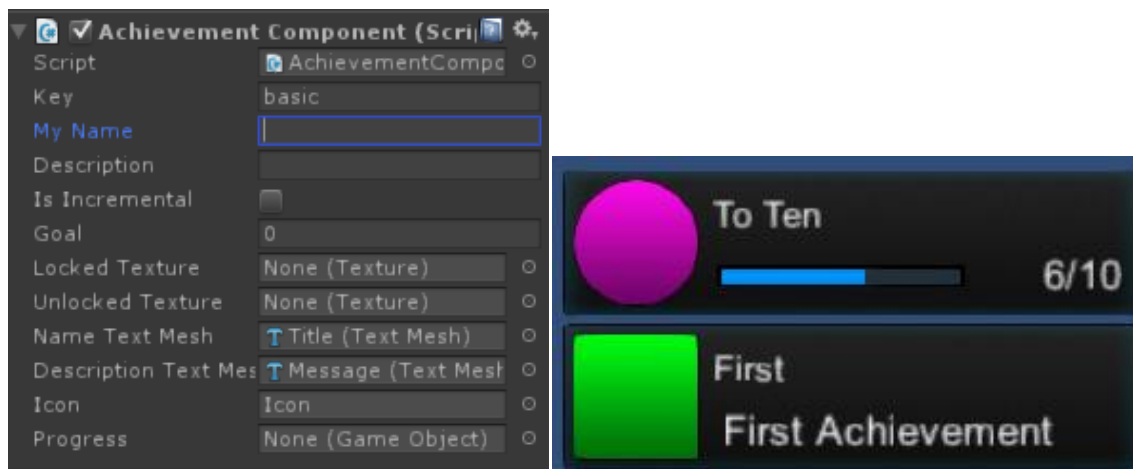
An achievement is comprised of the following information:

- **Key** – The key used to record and access an achievement.
- **Name** – The name of the achievement.
- **Description** – Description of the achievement, usually a few words to explain what is achieved.
- **Incremental** – A Boolean value to determine whether the achievement is incremental or not.
- **Goal** – If the achievement is incremental the number to reach.
- **Locked Texture** – The image to display when achievement is locked.
- **Unlocked Texture** – The image to display when achievement is unlocked.

ACHIEVEMENT CONTAINER/COMPONENT

An achievement container is used when you want to display the achievement permanently, for example when you want to display all achievements. These are also used to insert Achievements into the game, ideally these are set up on a screen that will ALWAYS be shown before any achievements are required to be unlocked.

Achievement Components are created when setting up achievements in the editor, the physical presence of the achievements ensures proper serialisation so achievements can always be accessed from the achievement manager.



The following components are to be used if using achievement components to display the achievements status:

- **Name (Text Mesh)** – This is the text mesh used to display the name of the achievement.
- **Description (Text Mesh)** – This is a text mesh which displays the achievements progress/description.
- **Icon (Game Object)** – The Game Object that will have the relevant texture applied to it.
- **Progress (Game Object)** – A Game Object containing a progress bar element

ACHIEVEMENT POPUP

An achievement popup is instantiated when an achievement is unlocked.



The components in this are set as follows:

- **Title (Text Mesh)** – This is the text mesh used to display the name of the achievement.
- **Description (Text Mesh)** – This is a text mesh which displays the achievements progress/description.
- **Icon (Game Object)** – The Game Object that will have the relevant texture applied to it.
- **Progress (Game Object)** – A Game Object containing a progress bar element

ACHIEVEMENT HELPER

The achievement helper is usually attached to the GUI camera and controls the instantiation of popups as required.

Components-

- **Popup (Game Object)** – This is to be linked to the Achievement popup to be displayed when an achievement increments/unlocks.
- **Popup Location (Transform)** – This is usually an empty Game Object with a Transform component, placed where the achievement is to be instantiated, serves as a marker.
- **Progress Bar Enabled** – This is a Boolean flag that dictates whether the progress bar is to be shown on incremental achievements.
- **Distance Between Popups** – This is a value which dictates the space between popups when more than one is to be displayed.
- **Max concurrent popups** – The maximum number of popups that can be displayed at one time.

Take a look at the **Achievements Camera** in the sample scenes to get a feel for how it works.

ACHIEVEMENT MANAGER

The Achievement Manager is the control mechanism for controlling all achievements, this class can be called to:

- **Unlock/Lock Achievements**
- **Increment Achievements**
- **Add/Remove Achievements from the game.**
- **Store Achievements**

SAMPLES

I have provided two samples in the **'Samples'** folder. Open these to get a feel for the different ways of using the system. To see them working be sure to hit **play** to allow the scripts to run.

UNLOCKING ACHIEVEMENTS

This shows a simple setup with three buttons and two unlockable achievements:

- **Increment** – Increments an incremental achievement.
- **Unlock** – Unlocks an achievement.
- **Reset** – Resets the achievement

VIEWING ACHIEVEMENTS

This sample just shows how the achievements can be displayed in containers, this could be used to build an achievement viewer.

TUTORIALS

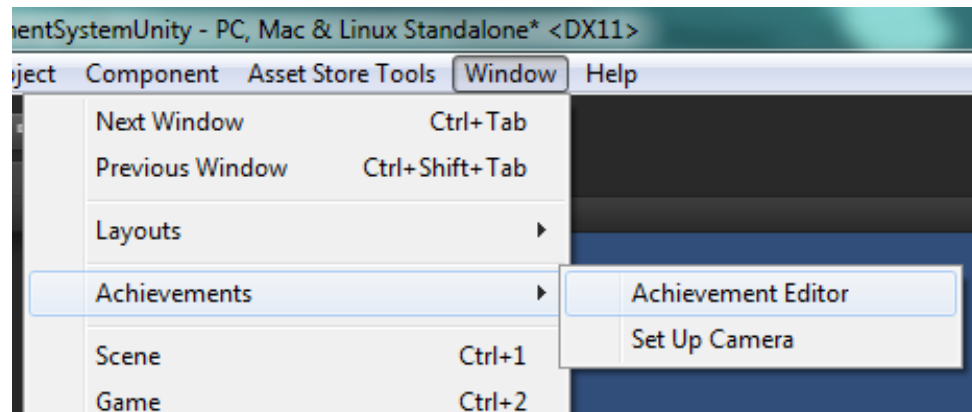
Below are a series of tutorials designed to familiarise you with using this system and provide you with a good foundation to produce achievements quickly and easily. The results of each tutorial will be stored in a **Tutorials** folder within the **Samples** folder.

TUTORIAL – SET UP

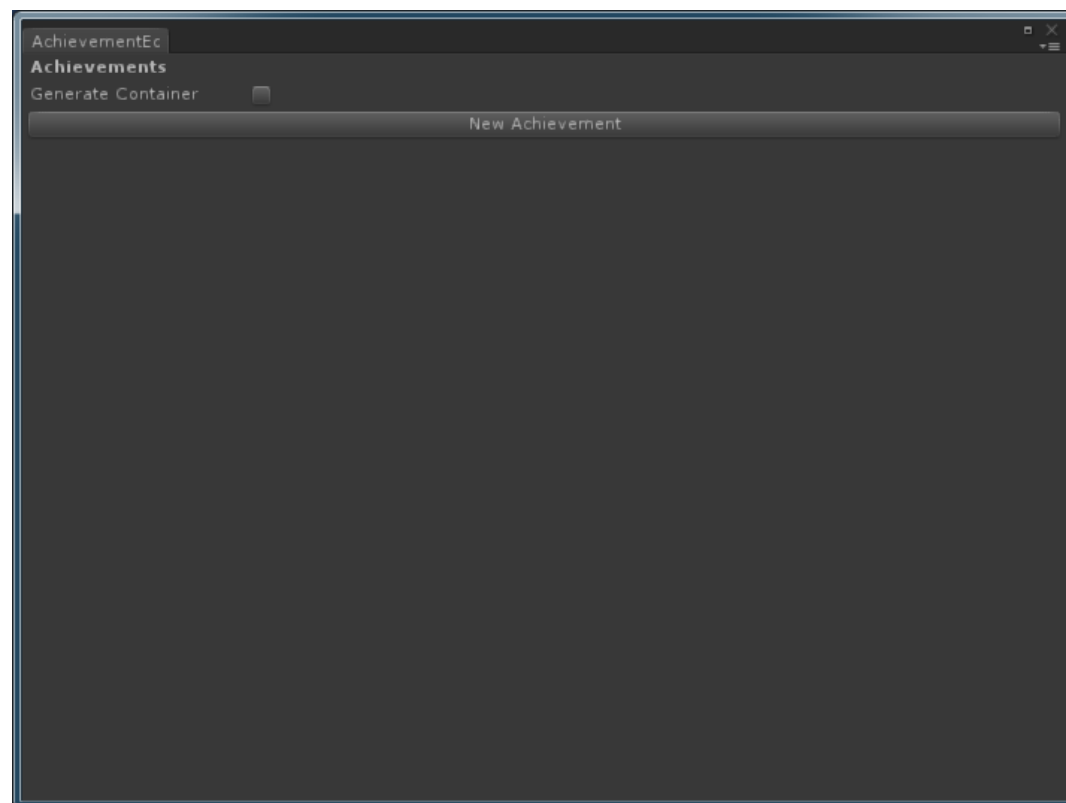
This tutorial will take you through the basic setup of the system. The system requires initialisation on the **FIRST SCREEN VIEWED WHEN THE GAME LAUNCHES**.

Let's Begin!

Step 1 – Create a new scene from **File>New Scene**.



Step 2 – Open the achievement editor from **Window>Achievement>Achievement Editor**.



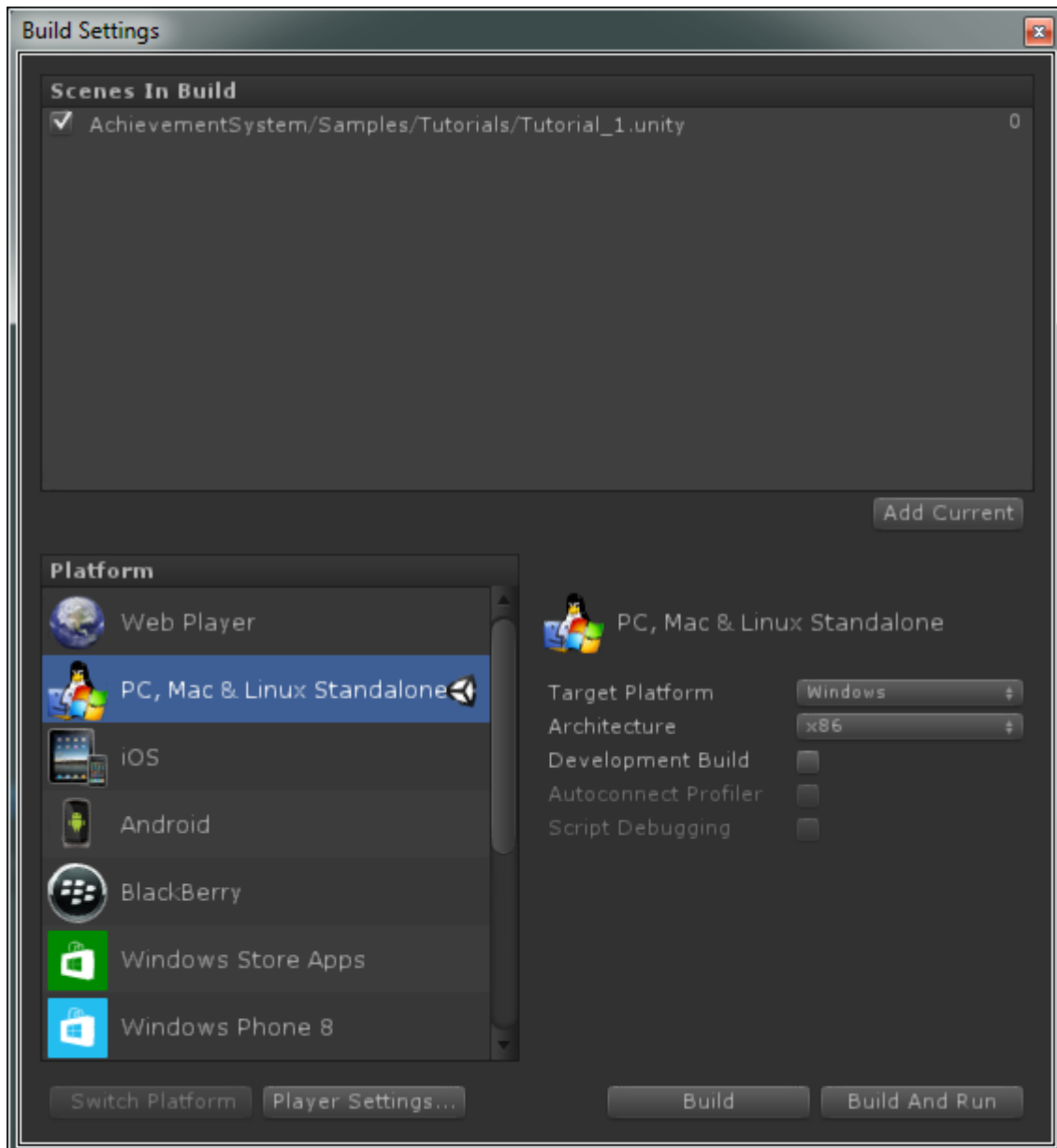
Step 3 – Make note of the newly created **Achievement Holder** in the **Hierarchy**, this is the main holding point for you achievements.

Step 4 – Decide whether you want to generate containers for achievements, this containers can be used to create an achievement viewer screen.

Note – If you create achievements then change this option you will have to edit any existing achievements to make use of containers.

Step 5 – Save the scene as whatever you want.

Step 6 – In **File>Build Settings** click on **Add Current** and ensure that the newly created scene is **AT THE TOP** of the build order.



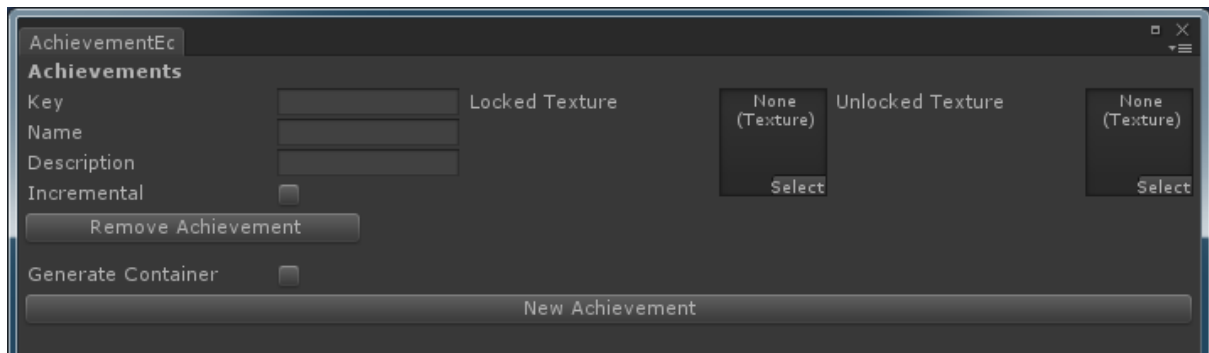
TUTORIAL – CREATING AN ACHIEVEMENT

This tutorial talks you through creating an achievement.

Step 1 – Open **Tutorial_1** from **Samples>Tutorials**.

Step 2- Open the Achievement Editor from **Window>Achievements>Achievement Editor**.

Step 3 – Click the **New Achievement** button. A blank achievement will be created.

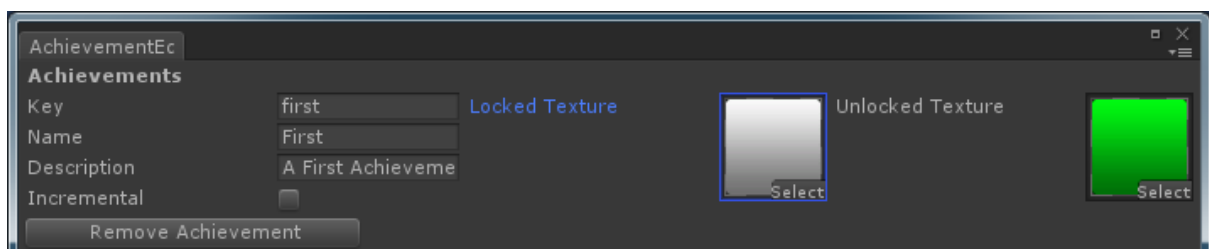


Step 4 – Define a key for the achievement.

IMPORTANT – THE KEY MUST BE UNIQUE!



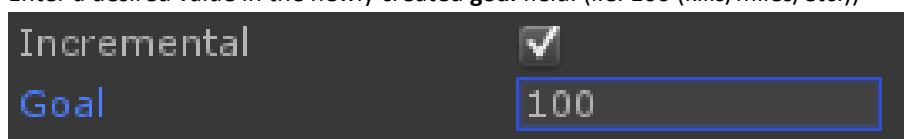
Step 5 – Fill out the rest of the details as required, there are some images for the locked and unlocked textures in **Art>Icons>Basic**.



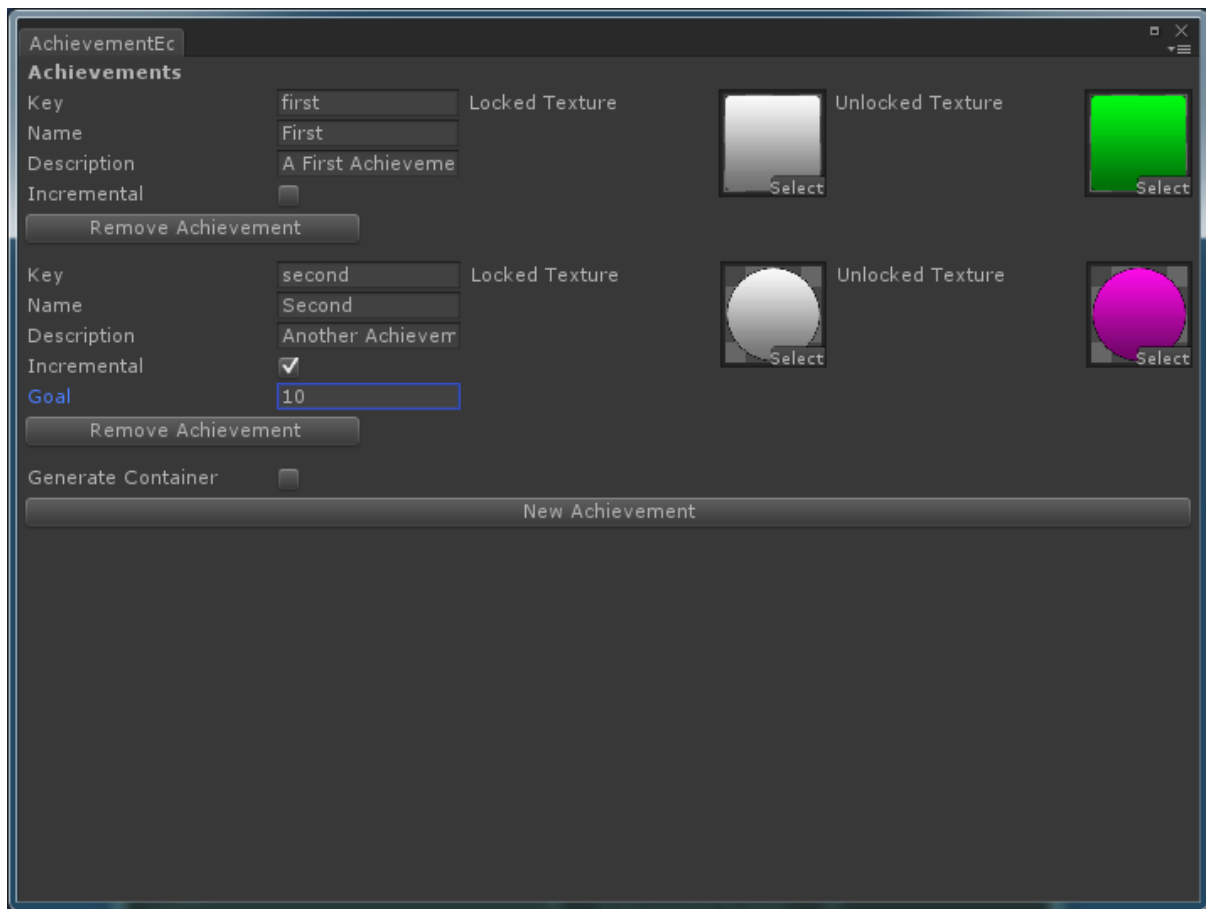
INCREMENTAL ACHIEVEMENTS

To make an incremental achievement:

- Click the checkbox marked **Incremental**.
- Enter a desired value in the newly created **goal** field. (i.e. 100 (kills/miles/etc.))



Step 6 – Play around with the system and add a couple of achievements as you desire.



Step 7 – Save the scene.

TUTORIAL – UNLOCK AN ACHIEVEMENT

To unlock an existing achievement all you have to do is call the following line of code from **ANY** script you desire:

```
AchievementManager.UnlockAchievement("key");
```

See **SampleUnlock.cs** for an example, this is used in the sample **UnlockingAchievements** on the **Unlock Button**.

TUTORIAL – INCREMENTING AN ACHIEVEMENT

To increment an existing achievement call the following code from **ANY** script:

```
AchievementManager.IncrementAchievement("key");
```

See **SampleIncrement.cs** for an example; this is used in the **UnlockingAchievements** sample, found on the **Increment Button**.

TUTORIAL – LOCKING AN ACHIEVEMENT

To lock an achievement:

```
AchievementManager.LockAchievement("key");
```

TUTORIAL DISPLAYING AN ACHIEVEMENT

To display an achievement:

Step 1 – Open **Tutorial_2** from the sample scenes.

Step 2 – Create the achievement camera by selecting **Window>Achievements>Set Up Camera**.

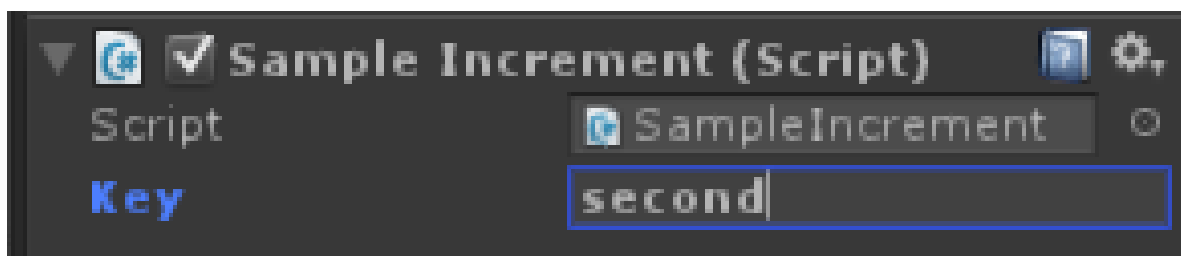
NOTE – Perform this step on EVERY scene you wish to allow achievement popups to appear on.

Step 3 – From the **Prefabs** folder drag the **Buttons** into the Hierarchy.

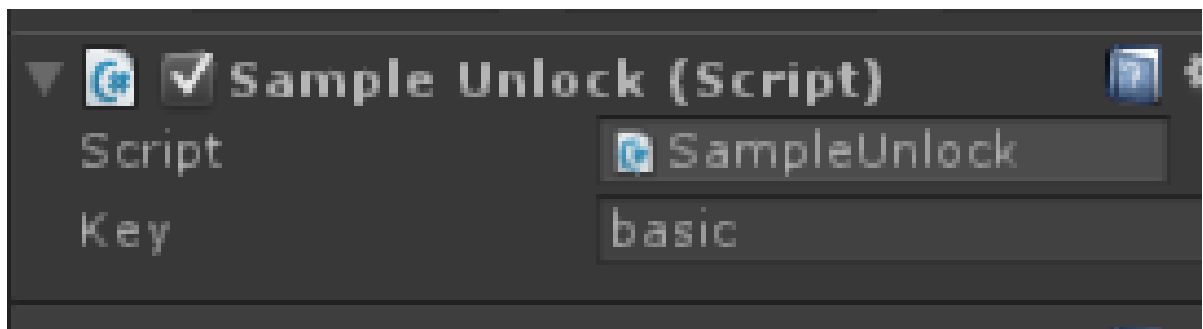


Step 4 – Take note of the scripts attached to each button.

Step 5 – In the **Increment Button** set the key in the “Key” field to an achievement stored in the Achievement Holder. (In this example “second”).



Step 8 – In the **Unlock Button** follow the steps described above and set the key.



Step 7– Hit play and press each button to see the effects.

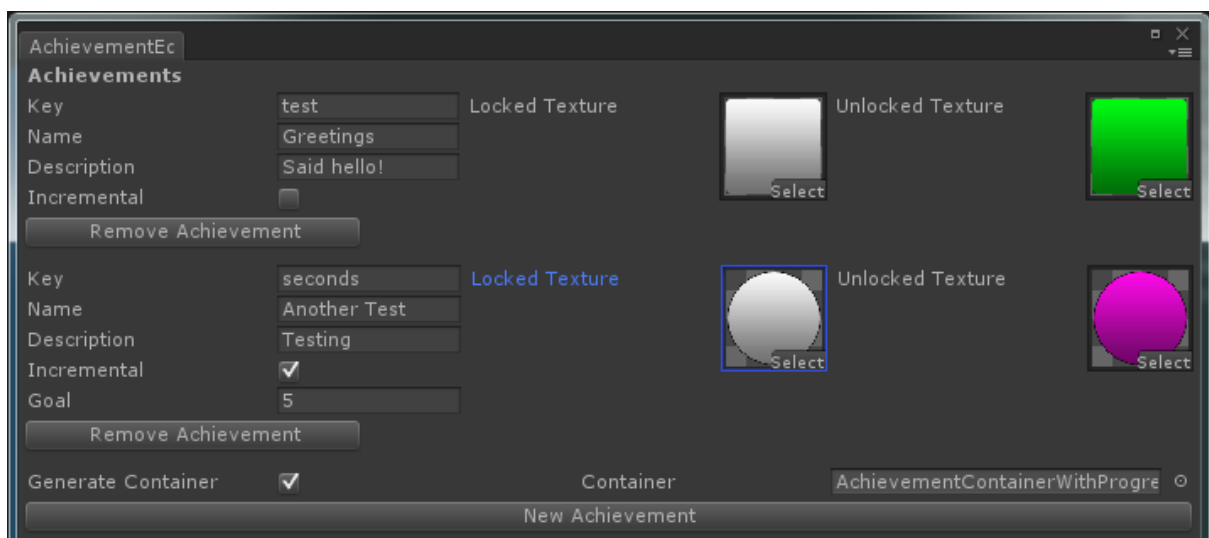


TUTORIAL – CREATE AN ACHIEVEMENT VIEWER

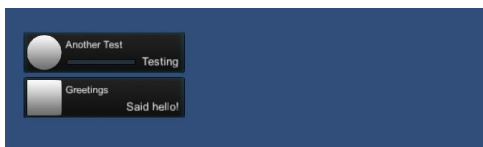
To create an achievement viewer:

Step 1 – Follow tutorials 1-2 **ENSURING THAT GENERATE CONTAINER IS SET TO TRUE IN THE ACHIEVMENT EDITOR.**

Note – If you want a progress bar displayed change the container from DefaultContainer to “AchievementContainerWithProgress” found in the Prefabs folder.

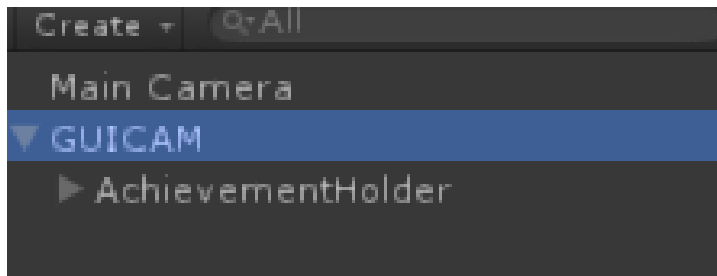


Step 2 – Drag **GUICAM** from the **Prefabs** folder into the Hierarchy.



(OPTIONAL) – Move the containers to desired locations on screen.

Step 3 – Attach the **AchievementHolder** to the **GUICAM** and store as a prefab.



Step 4 - Simply Instantiate the newly created prefab whenever you want to view achievements.

ADDITIONAL NOTES

Some points worth considering when working with this system.

When switching between scenes in the editor achievements will seem to lose their icons, this is due to the serialisation in Unity, it is essential that the first scene is allowed to play before playing any other scene.

Make sure that ALL achievements have both a name and a key or they will not display.

RESOURCES

Try to utilise the resources folder as much as possible. This will enable you to keep your hierarchy clean and organised.

It is worth noting that the resources in the resource folder when first using the system **SHOULD NOT BE REMOVED**. This is because of a failsafe in code that ensures that your system will work even if the system can't find your custom assets.

FORMATS

I made use of **PNG** and **PSD** for the majority of images; it is recommended that you use these formats, with images that have a size that is of a power of 2. Set high antialiasing and true colour in an images setting for best results.

SPECIAL THANKS

I would like to thank all the people on the Unity Forums for help, advice and support that I have had through the development of this project.

I would also like to thank:

Abbey Bottomley
Mark Kendrick & Maureen Lawson
Ilena Williams & Roger Williams

CONTACT

Email: blakekendrick1@hotmail.com

-OR

tsumiki15@gmail.com

Twitter: @TsumikiGameDev

Facebook: facebook.com/Tsumik1

Deviant-art: Koichi-Han

Unity Forums: Tsumik1