2014

Heathen Engineering

ON SCREEN KEYBOARD

BY: JAMES MCGHEE

Table of Contents

Whats New

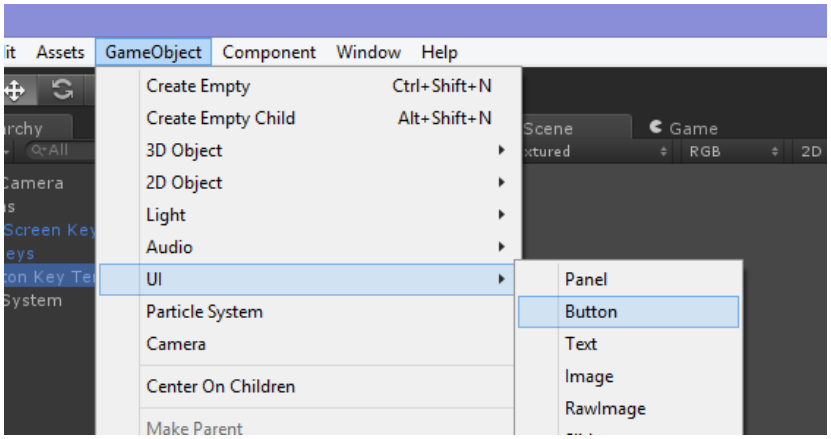
V2 introduces a number of fundamental changes though we maintain the easy to use keyboard generator and directional davigation; below you will find a short list of some of the key changes between v1 and v2.

- Use of UnityEnginer.UI aka uGUI
  That is we have removed calls to renderer and TextMesh in favor of the new Button, Canvas, etc. features.
  This change makes the keyboard much faster and most importantly much more flexible and easy to edit, stretch, skew and so forth.
- Removed Color data from the OnScreenKeyboard component
  As we now use the standard Unity GUI tools we have removed the coloring functions of the OSK as Unity's built in features are very easy to use and better equipped for a wider array of possibilities.
- Added KeyCode
  The OnScreenKeyboardKey now carries a KeyCode as well as the usual upper and lower values, this is handy if you want to pass the output of the OSK keyboard onto an existing controller that is expecting traditional KeyCode input.
- New Generated Structure
  Rows are now grouped under their own GameObject; we are also now using the Vertical and Horizontal Layout Group components to auto layout the keys of generated boards.
- Unity Navigation
  Since v2 uses standard Unity GUI elements we are able to leverage its built in navigation and have removed the manual directional navigation that we had in v1. Note that this is done by setting the navigation mode on generated keys to Explicate and then mapping the keys in the same way we did in v1. The only notable difference here is that Unity handles the navigation internally e.g. you don't need to call any navigate function.
  Just like with v1 the generated hook up will likely need adjusting in particular around large keys like the space bar; in this case you can use standard unity GUI navigation features to set it how you like even setting it to automatic which works well for large keys.

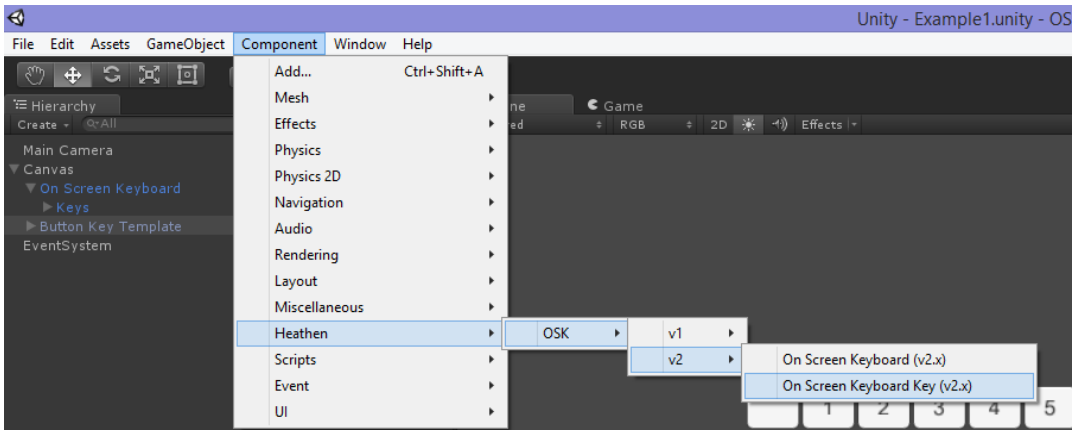
Creating a Keyboard

To create a keyboard you first want to create a key template; this template is simply a standard Unity Button with a bit of extra code to facilitate our key events and navigation.

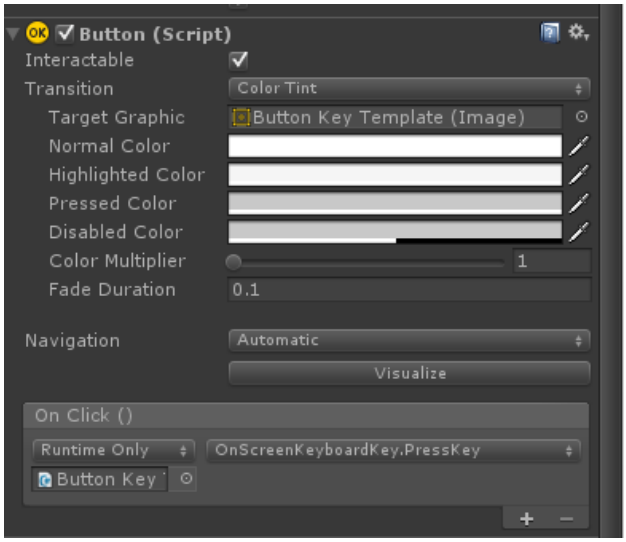
1. Create a new Button
   Click GameObject > UI > Button



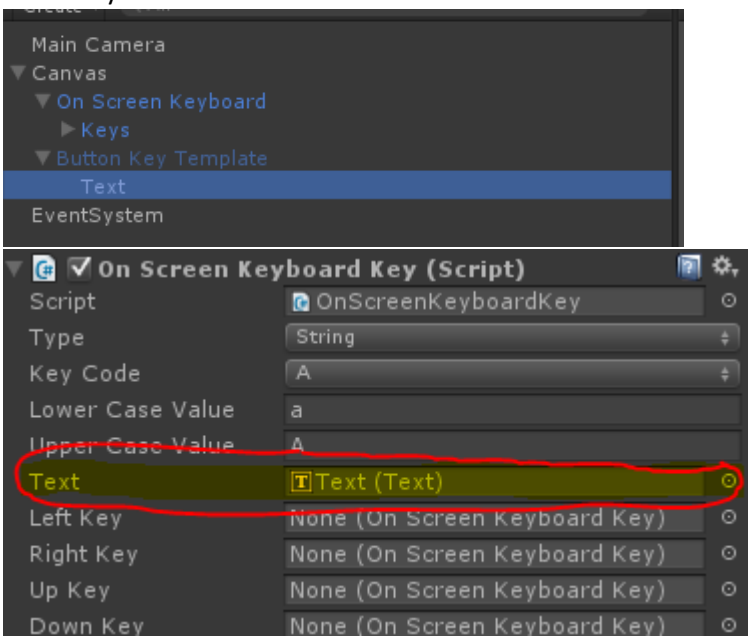
2. Add 'On Screen Keyboard Key (v2.x)' component
   Click Component > Heathen > OSK > v2 > On Screen Keyboard Key (v2.x)



3. Connect the On Click
   In the button script's On Click() select the template game object i.e. the same GameObject the button script is connected to and select the OnScreenKeyboardKey.PressKey function. Your script should look similar to the below:



4. Reference the text
   Simply set the Text field of the On Screen Keyboard Key to refrence the Text GameObject that was created when you added the button; it should be located under the button in the Hierarchy view.

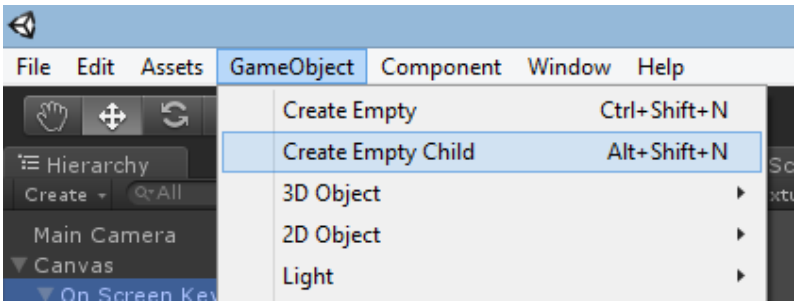


You can of course style this however you like; its simply a button and you can do with it all the things you would do with any other button. Just keep in mind we will be duplicating it so anything you do should be done at its scope or lower e.g. at or under it in the Hierarchy so it duplicates as well.

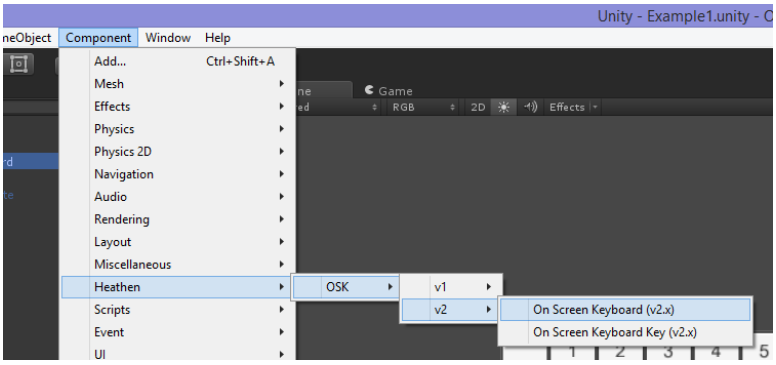
Next we are going to setup the keyboard itself; in this case we just need a new empty GameObject under our Canvas.

Note if you not aware when you added the Button Unity would have placed it under a Canvas either an existing one if you had one or it would have created one. Unity UI elements go here as it's this that sorts how the UI will be rendered in terms of transform (screen space, world space, etc.)

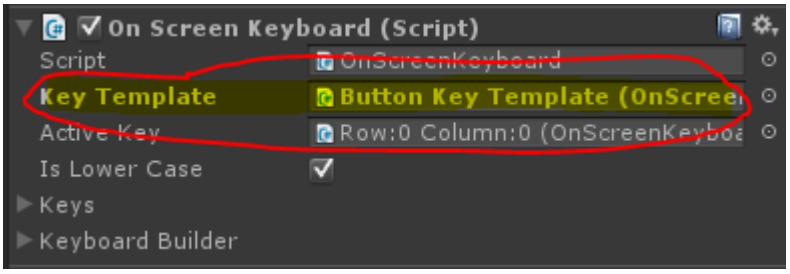
1. With your Canvas of choice selected click GameObject > Create Empty Child



2. Next add the OnScreenKeyboard script component via:
Component > Heathen > OSK > v2 > On Screen Keyboard (v2.x)



3. Reference your key template
Drag your key template to the Key Template field of the On Screen Keyboard component
Your On Screen Keyboard script should look something like this



4. Generate or Hand create keys
At this point you have 2 options; you can generate keys either from a built in setup by clicking the QWERTY or AZERTY buttons or you can manually define a layout by defining the row count, key count and settings for each key.
In either case once you are do you need to click the 'Generate Objects' button which will generate key layout and connect the navigation keys appropreatly.

Advanced Notes

Non generated keyboards;

You do not have to generate your keyboards you can layout your keys manually in any pattern or structure you like all you need do is make sure the keys are children of the keyboard as the keyboard will at run time poll for all keys under it and link for key events.

Adding and removing keys at runtime

You can add and remove keys at run time by simply adding and removing them from the keys list on the keyboard.