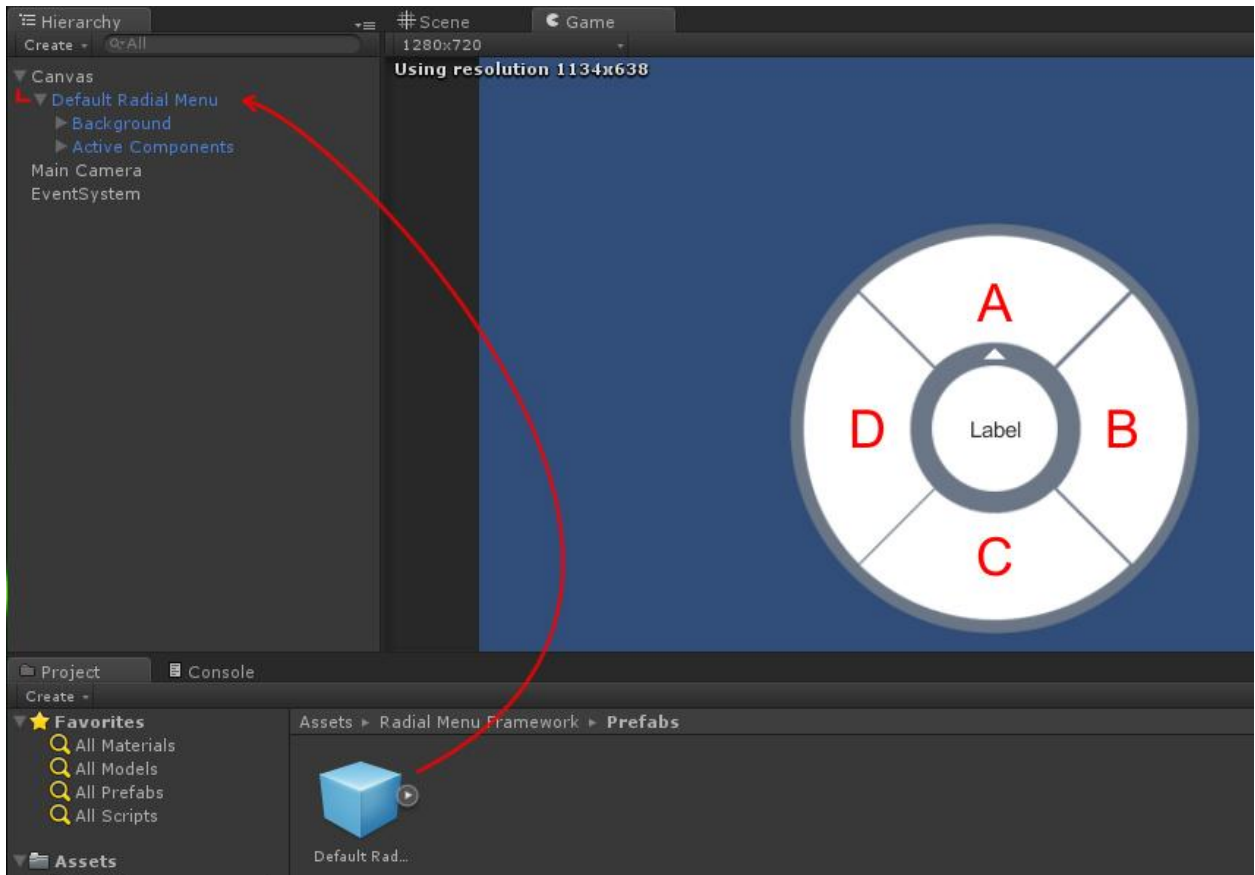


Radial Menu Framework 1.0.x Setup Guide

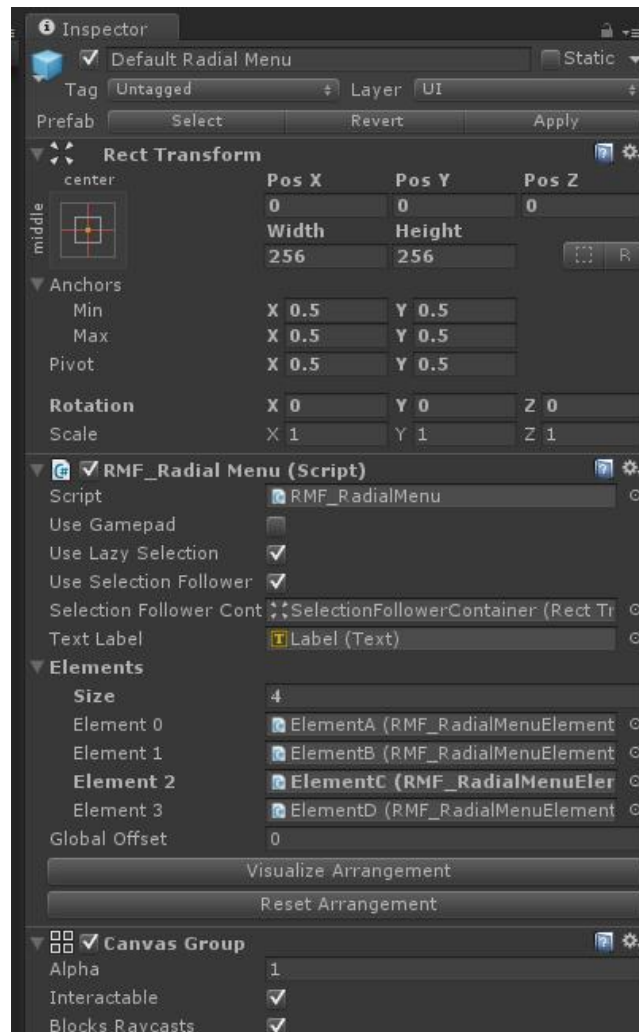
Thanks for your interest in the Radial Menu Framework! This guide will introduce you to how everything works.

The easiest method involves dragging and dropping prefabs into your canvas, and then modifying it to fit your needs. If needed, building one from scratch isn't too difficult once you understand how they are set up, which this guide will explain for you.

1. Look in the Radial Menu Framework \ Prefabs folder.



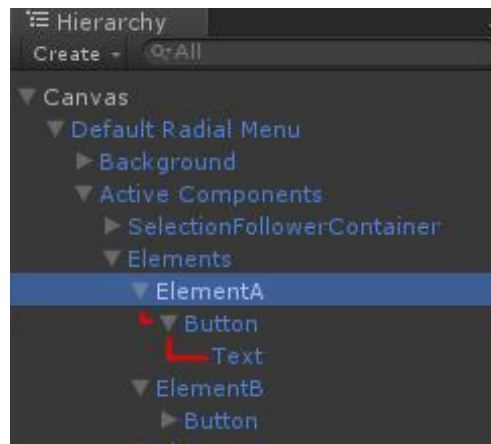
2. Drag and drop the “Default Radial Menu” prefab into your scene, underneath a canvas. (Please note that lazy selection and the selection follower features will NOT work if the canvas is a world-space canvas.)



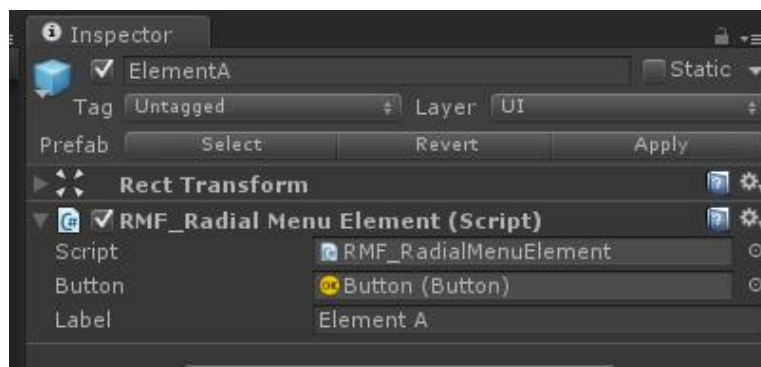
3. Set up the primary settings.

- a. **Use Gamepad:** Check this if you want to use a gamepad or some sort of joystick. The script uses the built-in “Horizontal” and “Vertical” axes, so if you need to change these, you can easily do so at the start of the update() function. It’s clearly labeled.
- b. **Use Lazy Selection:** Check this if you only want to point in the general direction of an element to be able to select it. With this off, you will only be able to select elements that are directly moused-over, or you will be completely bound by build-in navigation setup for gamepads.
 - i. *PLEASE NOTE: While lazy selection works perfectly fine for non-rectangular GUI images, if you do not have this on and you are using the mouse, please be aware that raycast detection will still be completely rectangular. Like anywhere else in UnityUI, if there are non-rectangular buttons overlapping, weird things will happen. Look for a free script called “Raycast Mask” that will help you with this.*
- c. **Use Selection Follower:** Check this if you want a little pointer to follow your mouse cursor / direction of joystick input. Relevant with joysticks, or if lazy selection is enabled when using a mouse.
- d. **Selection Follower Container:** This is the parent of the selection follower image. This should be a blank game object at the center of the radial menu. The selection follower image should be a child that is placed in a position directly upwards.
- e. **Text Label:** This is a label that will read the label from each radial menu element. Leave it blank if you don’t want there to be a label.

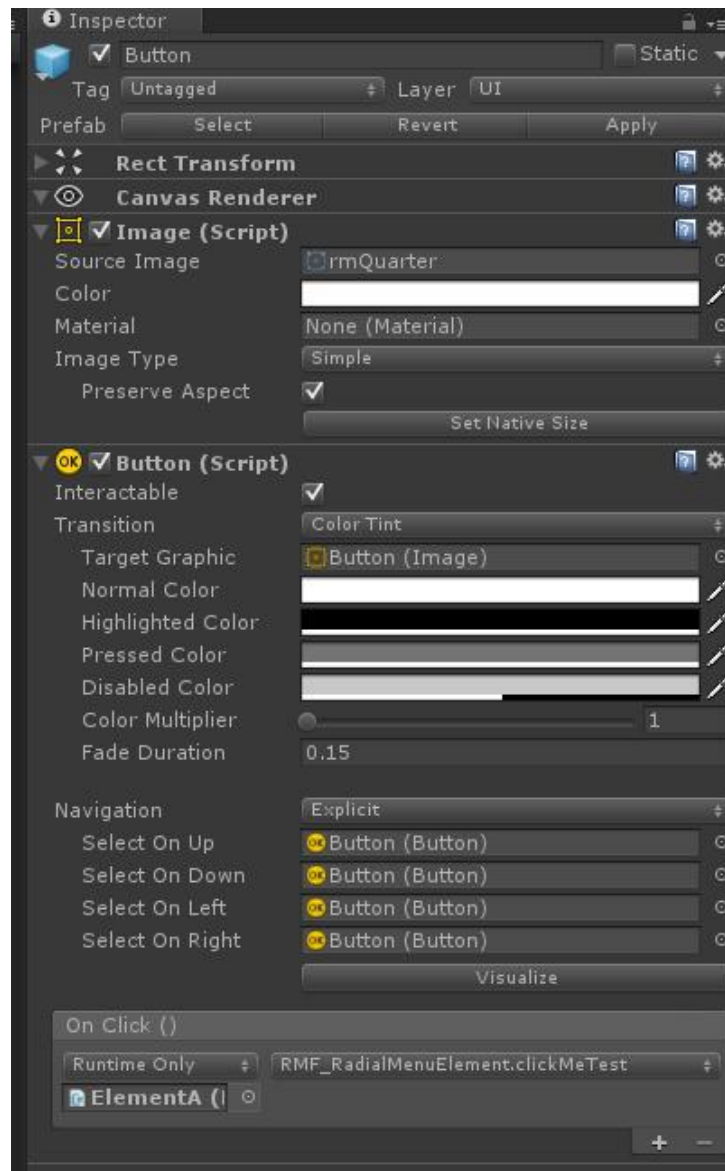
- f. **Elements:** These are the individual buttons that will appear on your radial menu. Set the size to the number of buttons you want. Once your elements are set up, you will be dragging and dropping them into these spots. This is order dependent: the first element in the list will be the first element displayed at runtime.
- g. **Global Offset:** This will rotate all element by the specified number of degrees.
- h. **Visualize Arrangement:** This will properly rotate all of the currently loaded elements into the places they will appear at runtime. This is for visualization only: positions will be reset to 0 and then recalculated at runtime.
- i. **Reset Arrangement:** This will move all the elements back to a rotation of 0, stacking them on top of each other. This makes it easy for you to move each button upwards at once so that the distance from the center of the radial menu is identical for each element.



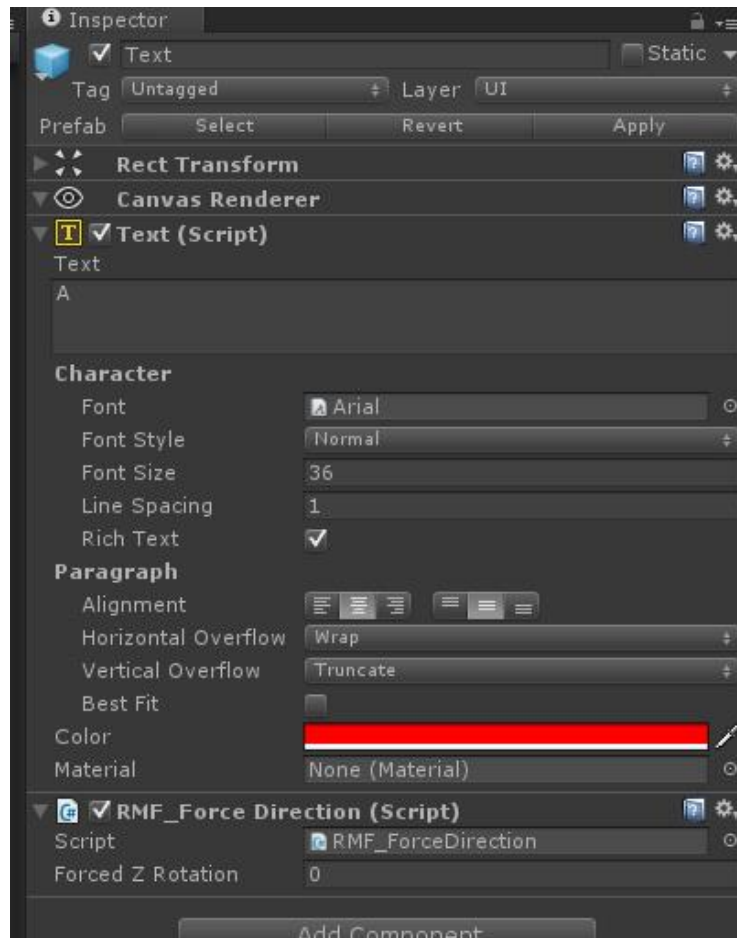
4. Our next step is to set up the elements. Elements are generally composed of 3 parts: The core element, the button, and the text/icon. The core element is a blank gameobject with the RMF_Element script attached. The button is a child of this core element, and then your text/icon (optional, of course) will usually be a child of the button. If you want to create more elements on the default radial menu, it's easiest to duplicate one of the existing ones. Select one of the elements, and press Control + D. Technically, in order for it to work, all you'd need to do is add it as a fifth element to the main radial menu script, however, you'll probably want to differentiate it.



- a. The “Button” field is where you drag and drop your button. This will be a child of the gameobject the RMF Radial Menu Element script is attached to.
- b. The “Label” field is what the radial menu will display when you have this element highlighted. You can leave this blank if you wish.



5. The button itself is set up identically to a normal UGUI button. The function to be called should be set up in the On Click() event handler. If you wish to specify navigation for a gamepad (If you're not using lazy selection), you'll want to set it to "Explicit". "Automatic" navigation probably isn't going to work properly.



6. The last thing to set up on the button is the text or icon. Because each element gets rotated, this means that text and icons are not going to appear with the correct orientation (Text could be vertical, slanted, etc). The **RMF_Force Direction** script solves this problem. The “Forced Z Rotation” parameter will set this UGUI element’s absolute rotation to the specified value; usually this will be 0. No matter how any of the object’s parents are rotated, it will always be facing in a certain direction. This also works in the editor.
7. When you’re all done setting up each of your buttons, drag and drop them in the order you want in the core radial menu script. You can also modify the Background objects if you desire, they are purely visual, so go as crazy as you want. Additionally, if you wish to change the appearance of the selection follower, you can do so under SelectionFollowerContainer > Pointer.

A few notes:

- If you need to change the size of the menu, adjusting the scale in the core rect transform works well.
- Be careful of anchors—if things start scaling in odd ways based on resolution, the lazy selection option might not work as intended. By default, all anchors are set to the dead center of the radial menu. This means it won't scale based on resolution by default.
- Make use of the “Visualize” and “Reset” arrangement buttons, they'll help you quickly edit things so they look as good as possible ingame.
- If using a gamepad/joystick, you don't need to explicitly define navigation if you use lazy selection.
- Lazy selection allows you to do a lot of crazy animation stuff with your buttons—you don't have to worry about them moving out from under your mouse pointer.
- Buttons support multi-component editing. Control + Click all the buttons in your elements to modify them all at once. Makes things easier for things like changing colors, or changing the image they use.

If you have any questions, feel free to email me at: brettg94@live.com

Thanks!