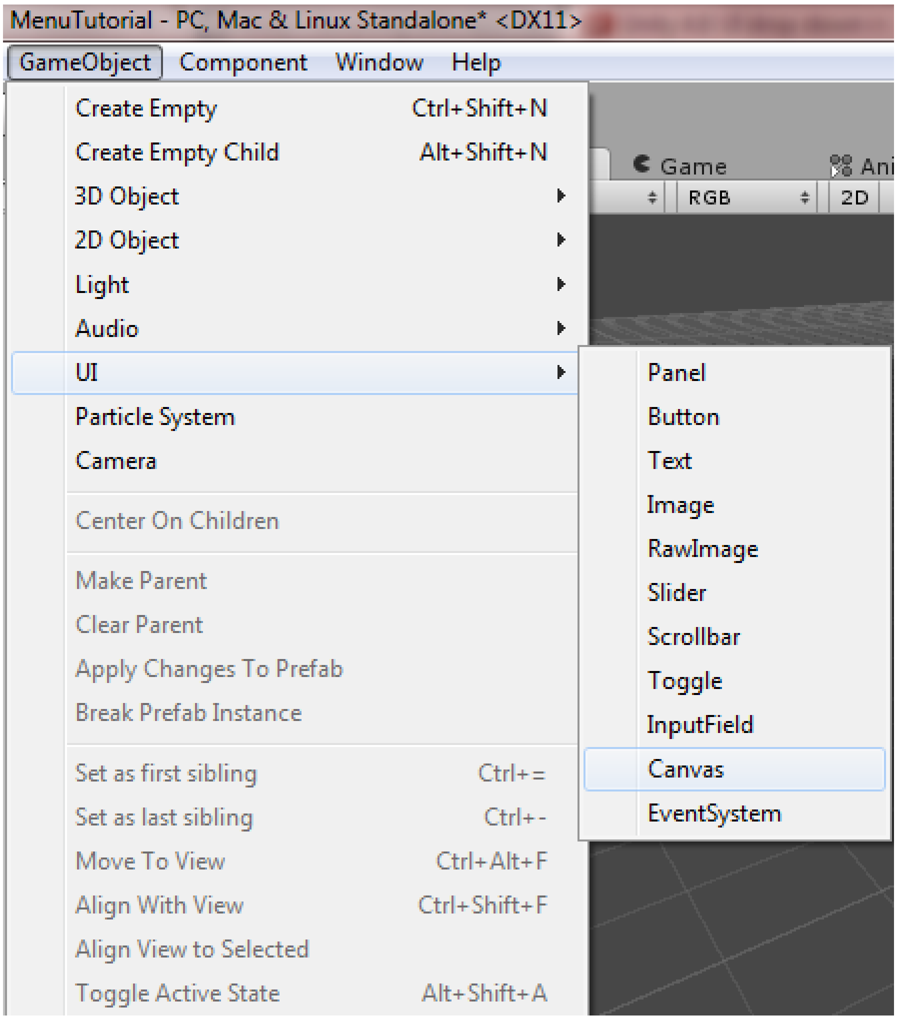
How to make a scrollable panel using the 4.6 Unity User Interface.

Let's start with a simple panel that just take a basic Text box prefab that can be added to dynamically.

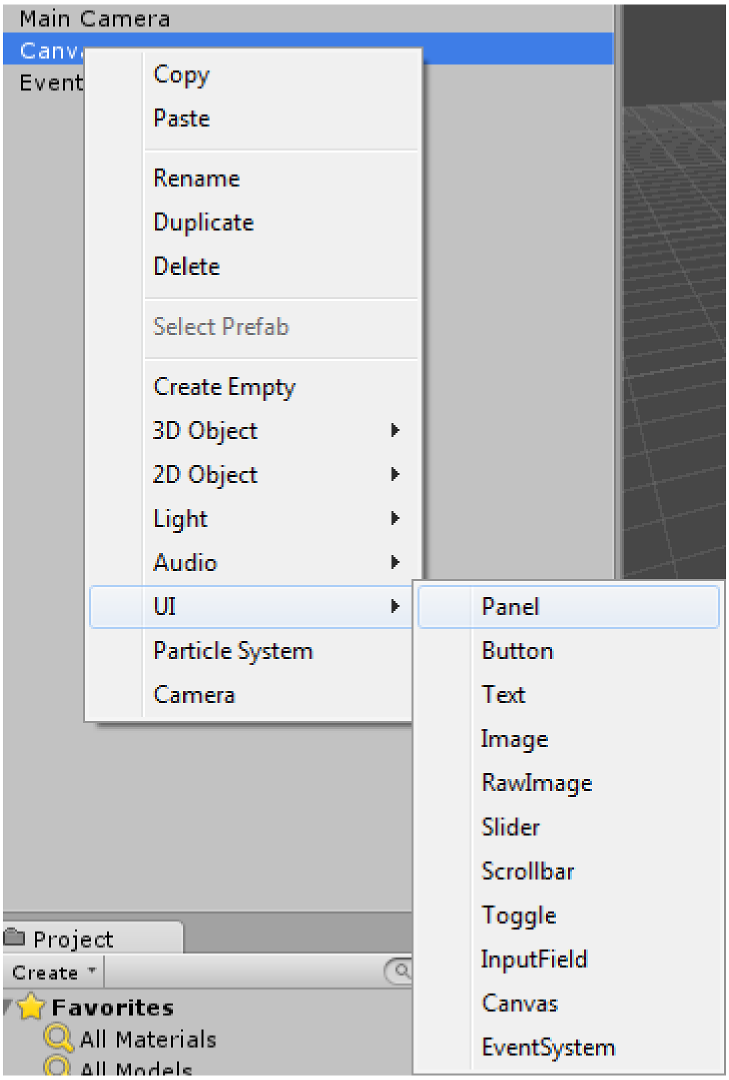
Before we start I should say this started off being based on a YouTube tutorial from Stuart Spence found here https://www.youtube.com/watch?v=TRLsmuYMs8Q but I had some issues with scaling on different resolutions so I eventually moved the way of creating the menu to a Richard Gubb tutorial here https://www.youtube.com/watch?v=-FcrLi49RTg both are excellent tutorials. Note that the UI changed from the Richard Gubb tutorial so if you create a menu bar like his tutorial then in the Horizontal Layout Group uncheck the Force child expand.

OK create a new Project and add a Canvas:

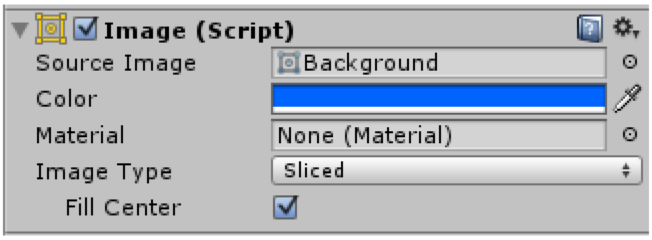


To this we'll add a basic background image so we have a backdrop to hide the terrain. This isn't about making a great looking menu just a functional one!

OK so right click the Canvas and add a panel.



Change the color to whatever you want and set the alpha to 255. Click on the Color panel here:



☑ Image (Script)

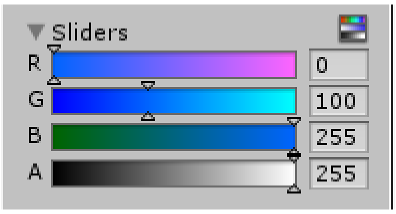
Source Image: Background

Color

Material: None (Material)

Image Type: Sliced

Fill Center: ☑



Sliders

R: 0

G: 100

B: 255

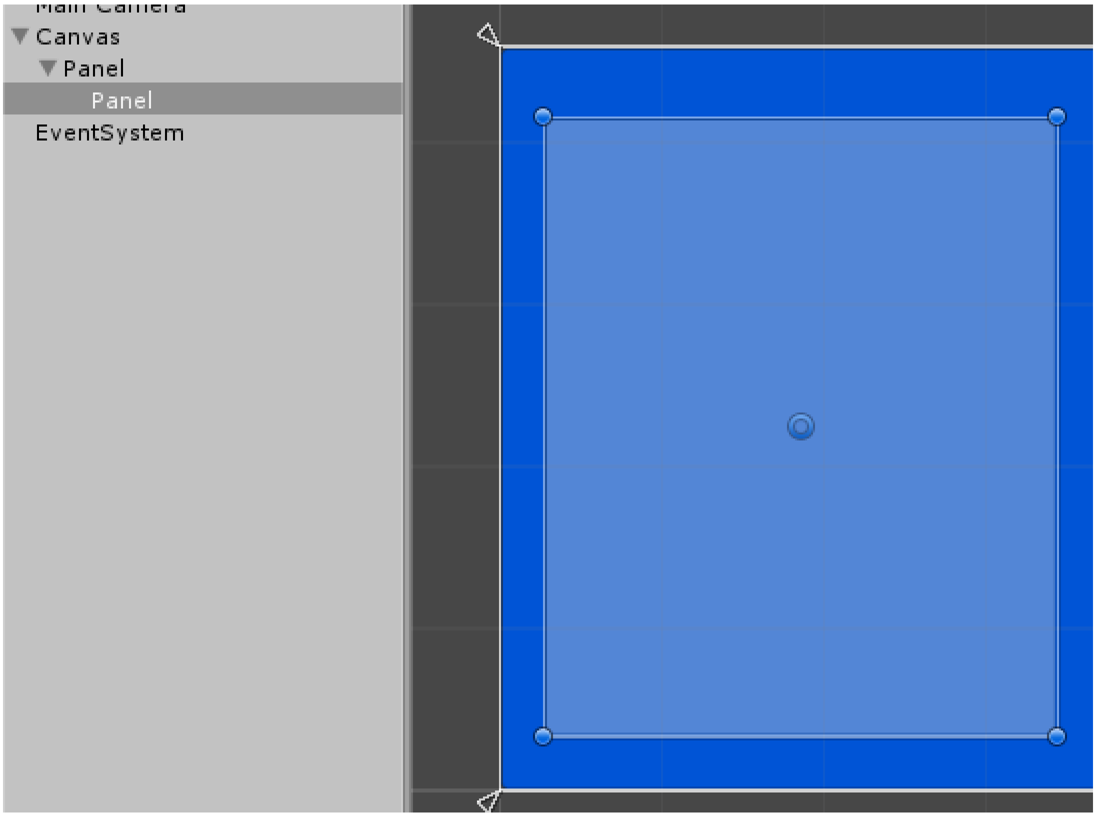
A: 255

Choose whatever settings you want, setting the A value to 255 removes any opacity but who you setup the aesthetics is up to you.

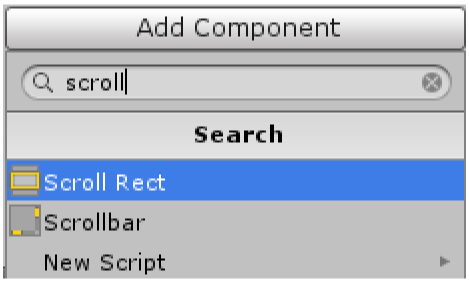
Now click on the 2D button or swing the Camera round so you can see the canvas face on. Then zoom out, a lot, the new canvas system is vast so zoom out until you can see full canvas.

Now we want to make a child panel so right click the current panel and select UI -> Panel.

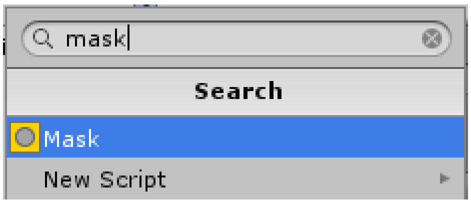
Resize the panel until you end up with something that suits your needs, this works for me:



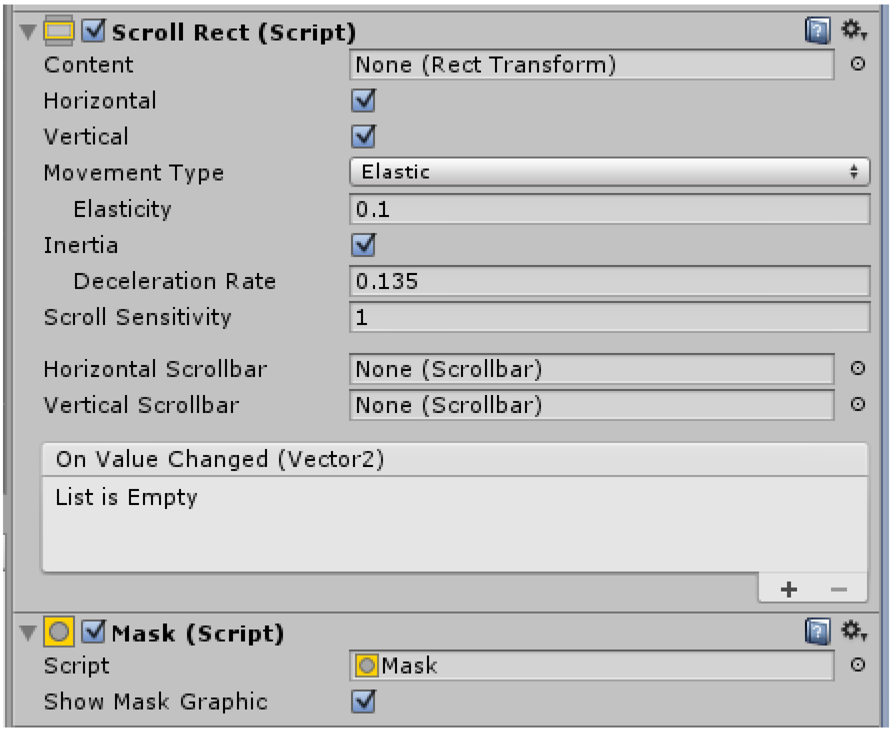
This is going to be the maximum size of the menu it just restricts the size that can be seen on screen. To do this we add a mask component and the ScrollRect component to the panel so with the new panel selected click on Add Component and type scroll.



We'll leave the setting for the time being but add a mask so click on Add Component again and type in mask.



You should now have both components on your panel:



☑ Scroll Rect (Script)

Content: None (Rect Transform)

Horizontal: ☑

Vertical: ☑

Movement Type: Elastic

Elasticity: 0.1

Inertia: ☑

Deceleration Rate: 0.135

Scroll Sensitivity: 1

Horizontal Scrollbar: None (Scrollbar)

Vertical Scrollbar: None (Scrollbar)

On Value Changed (Vector2)

List is Empty

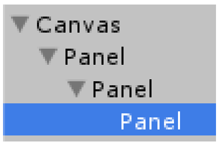
☑ Mask (Script)

Script: Mask

Show Mask Graphic: ☑

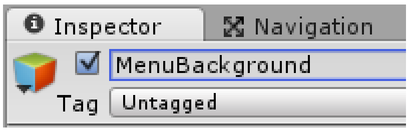
The first thing in the ScrollRect is the Content. For this we need another Panel so right click the panel we've just added the ScrollRect to and UI - > Panel.

So now we have this:



```
Canvas
  Panel
    Panel
      Panel
```

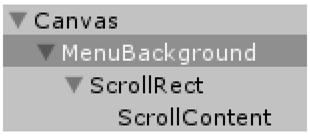
Probably a good idea to rename some of these so we don't get confused, I'm going to use MenuBackground, ScrollRect & ScrollContent and we rename by altering the name in the inspector.



☑ MenuBackground

Tag: Untagged

And we end up with this:

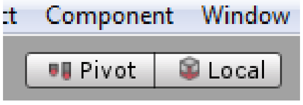


```
Canvas
  MenuBackground
    ScrollRect
      ScrollContent
```

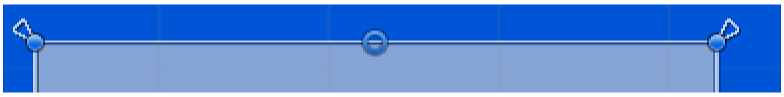
OK now we can tell them apart add the ScrollContent to the ScrollRect panels ScrollRect component's Content slot and un-tick Horizontal on the ScrollRect panels ScrollRect component! Got that?

Now click on the ScrollContent and move the pivot point to the top. This is really important, the new UI bases the starting point of the panel and the default start point of a scroll bar on the pivot. It makes a massive difference.

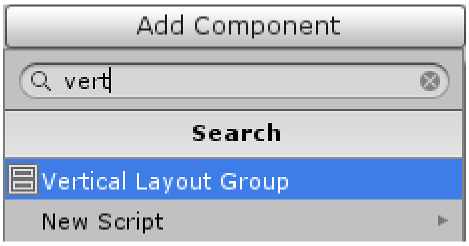
First make sure the pivot toggle is selected rather than the center toggle:



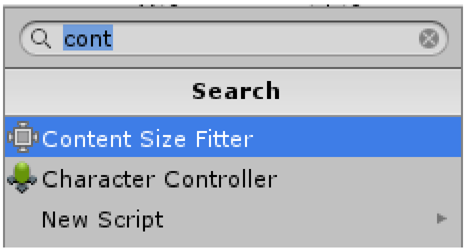
Then drag the pivot to the top center so you end up with this:



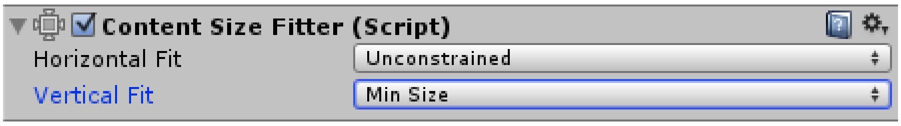
The pivot should snap into position at the top and center. So back to Add Component, this time we're on the ScrollContent panel and we add a Vertical Layout Group.



Leave the default settings and Add Component again, this time Content Size Filter:



On here we set the Vertical fit to Min Size:



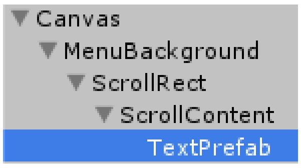
☑ Content Size Fitter (Script)

Horizontal Fit: Unconstrained

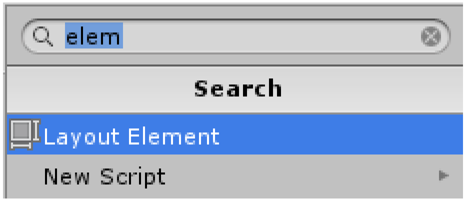
Vertical Fit: Min Size

Notice the panel effectively disappears as there's nothing in there so the panel's height is 0.

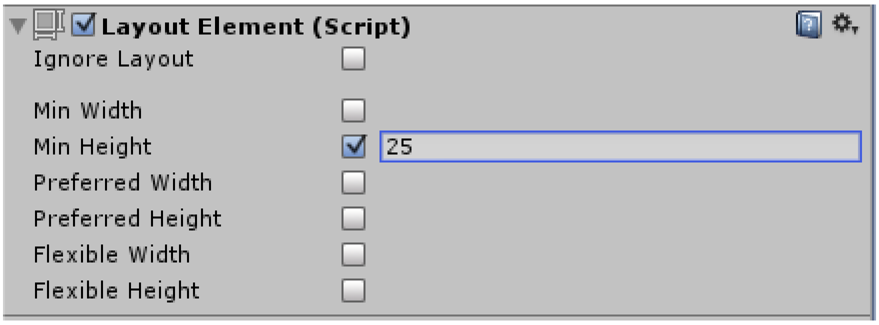
OK we're ready to add a Text box, this is going to be our prefab and it could be anything from a text box to a panel with multiple child objects but let's keep it simple and just do a text box. Right click our ScrollContent panel and select UI -> Text. Rename it TextPrefab so we now have:



But if you look in the panel there's no text there. The reason is the Text panel has no minimum size so the Vertical Fit from our Content Size Filter is overriding the size. So back to Add Component and add a Layout Element:



The tick the Min Height and set it to whatever's reasonable for your needs.



☑ Layout Element (Script)

Ignore Layout □

Min Width □

Min Height ☑ 25

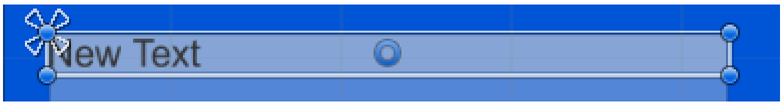
Preferred Width □

Preferred Height □

Flexible Width □

Flexible Height □

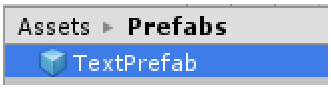
And the text now appears



In the Text I've selected the Best Fit so the text scales but you might want different settings.

Now we drag the Text element off into the Assets. Create a folder for Prefabs and let's keep it simple and call it Prefabs.

Right click on Assets and select Create -> Folder then rename it Prefabs. Drag the TextPrefab into that folder and you should end up with this:



We can delete the TextPrefab that's a child of the ScrollContent panel. So we're now ready for the code bit where we take the prefab and instantiate it multiple times. Don't worry I'll show all the code, really all we need is an array of some sort to store a list of text that we can add to the panel.

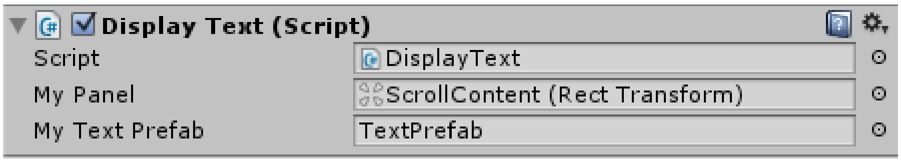
Create another folder for our scripts, right click the Assets folder and Create -> Folder, then rename it Scripts.

Go into that Scripts folder and right click on a blank area, select Create -> C# Script and call it DisplayText. This is the contents.

```
using UnityEngine;
using System.Collections;
using System.Collections.Generic;
using UnityEngine.UI;

public class DisplayText : MonoBehaviour {

        public RectTransform myPanel;
        public GameObject myTextPrefab;
        List<string> chatEvents;

        private float nextMessage;
        private int myNumber = 0;
        private GameObject newText;

        // Use this for initialization
        void Start () {
                chatEvents = new List<string>();

                chatEvents.Add("this");
                chatEvents.Add("is");
                chatEvents.Add("a");
                chatEvents.Add("test");
                chatEvents.Add("for");
                chatEvents.Add("showing");
                chatEvents.Add("strings");
                chatEvents.Add("and");
                chatEvents.Add("displaying");
                chatEvents.Add("in");
                chatEvents.Add("a");
                chatEvents.Add("scrollable");
                chatEvents.Add("panel");
                chatEvents.Add("in");
                chatEvents.Add("the");
                chatEvents.Add("new");
                chatEvents.Add("GUI");
                chatEvents.Add("really we can put anything in this");
                chatEvents.Add("anything at all");
                chatEvents.Add("Knock, knock");
                chatEvents.Add("who's there");
                chatEvents.Add("Doctor");
                chatEvents.Add("Doctor who?");
                chatEvents.Add("I refuse to participate in a joke older than I am");
                chatEvents.Add("yeah right older than you are!");
                chatEvents.Add("It is older than me honest");
                chatEvents.Add("as if, remember I know how old you are");
                chatEvents.Add("Look there's  clear evidence that they started in 1930's");
                chatEvents.Add("Maybe but Dr Who didn't did it");
                chatEvents.Add("Well OK but Dr Who's older than me");
                chatEvents.Add("Yes but that joke didn't start with the first episode");
                chatEvents.Add("Right I've had enough of this");
                chatEvents.Add("Stop being so melodramatic");

                nextMessage = Time.time + 1f;

        }

        // Update is called once per frame
        void Update () {
                if(Time.time > nextMessage && myNumber < chatEvents.Count)
                {
                        GameObject newText = (GameObject)Instantiate(myTextPrefab);
                        newText.transform.SetParent(myPanel);
                        newText.GetComponent<Text>().text = chatEvents[myNumber];
                        myNumber ++;
                        nextMessage = Time.time + 1f;
                }
        }
}
```

OK I'm keeping it simple and just creating a string List that I fill with rubbish in the Start() function. Drag the script onto the Canvas (it can go onto any GameObject but it keeps it neat if we keep it on the Canvas).

Then we need to drag the two Public variables the RectTransform for the ScrollContent panel and the TextPrefab. Highlight the Canvas so we can see the script and empty slots and drag on the following:



☑ Display Text (Script)

Script	DisplayText
My Panel	ScrollContent (Rect Transform)
My Text Prefab	TextPrefab

The Update() function the script waits until the nextMessage time is reached and then instantiates a clone of the TextPrefab and sets it's text component to equal the corresponding value in our List.

Whilst we're talking about lists I added the

```
using Systems.Collections.Generic;
```

So we can use lists and

```
using UnityEngine.UI;
```

So we can reference UI elements within a script.