

# Welcome to Paintcraft!

Thanks so much for the interest in this package – we think you're really going to like using Paintcraft to build and launch your high-quality drawing applications.

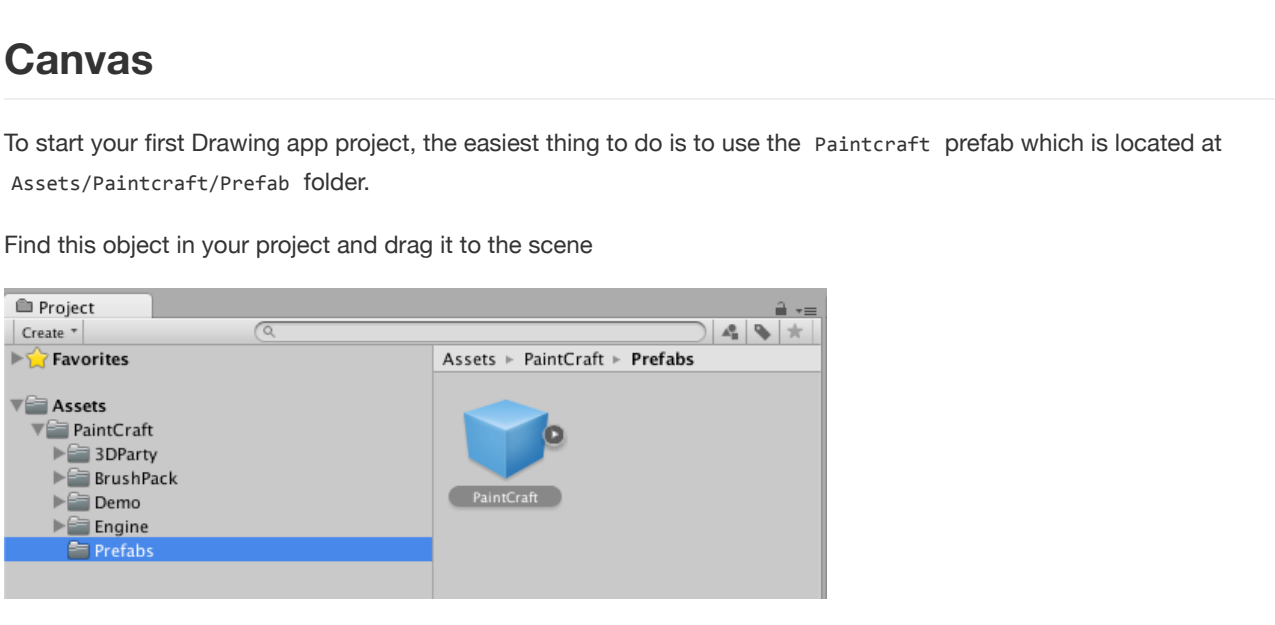
We are doing our best to make this package artist friendly and really flexible so it will satisfy any of your needs. This package is intended to walk anyone familiar with Unity3D through the basics of Paintcraft package in just about 25 minutes. Hopefully you'll learn how to make drawing app as fast as possible.

And if you wish to try how it works right now just open one of the demo scenes located at the `/demo/` folder or `/Demo/ColoringBook/` contains complete application

## Package Structure

```
PaintCraft
├── /3DPrefab/
│   ├── /nodeInspector/ # (1) OpenSource graph editor used in brush configuration
│   └── /BrushPack/ # (2) Contains built-in tools (Ink, Cryon, etc)
├── /Demo/
│   ├── /ColoringBook/ # (3) Complete project (you can use it for reskinning)
│   ├── /TwoPlayerBasic/ # (4) Simple config for one player drawing
│   ├── /TwoPlayerOneCanvas/ # (5) Two players draw on the same canvas
│   ├── /TwoPlayerTwoCanvases/ # (6) Two players draw on own canvases simultaneously
│   ├── /Egging/ # (7) Paintcraft source code (normally you don't need to go there)
│   └── /Prefabs/
│       └── /Paintcraft.prefab # (8) Minimal working drawing config.
```

1. Brush configuration is really flexible. You can control your line almost at any point. And this plugin used to make it more user friendly, here is an example of one of the brush config



2. We included several brushes to this package so you can freely use them in your project, and for sure you can play with any parameters.
3. This is a live project with picture selection and 2 players split screen drawing. You can test it here in the [browser](#) or on your mobile [Windows Store Android](#)
4. Simple example. Minimal canvas configuration (you can do the same if you follow this instruction)
5. Here 2 players draw simultaneously on the same canvas. Each player can use own brush and colors
6. The same as above, but each player has own independent canvas.
7. This is a source code of this package, if you are a programmer maybe it will be interesting to check the source code.
8. This is a configured canvas prefab, you can drag it to your scene and you will get working drawing app with minimum configuration - ...

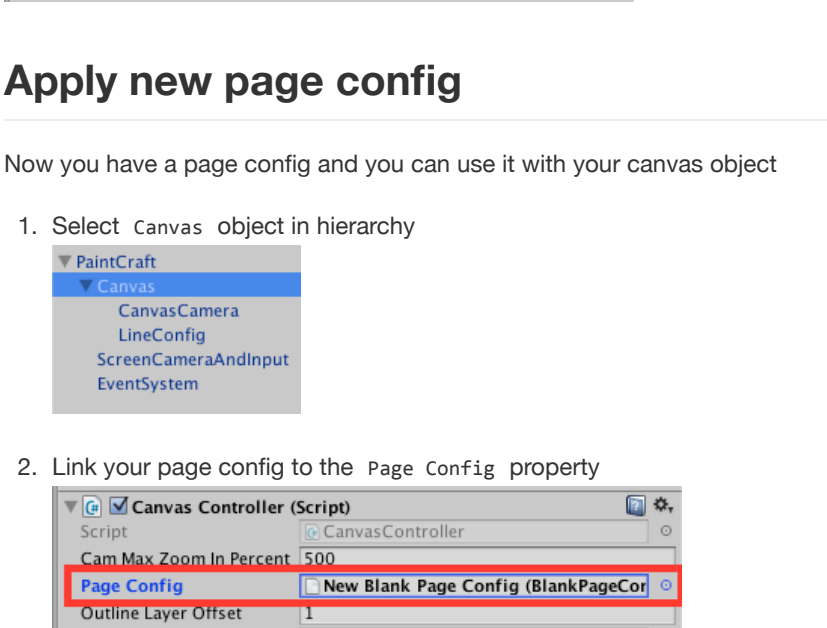
In next parts we will show how to create coloring book from scratch

## Canvas Prefab

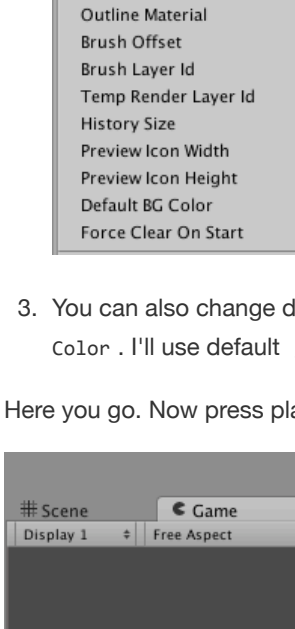
### Canvas

To start your first Drawing app quickly, the easiest thing to do is to use the `Paintcraft.prefab` which is located at `Assets/Interact/Prefab` folder.

Find this object in your project and drag it to the scene



If you have a brand new scene you will get this structure



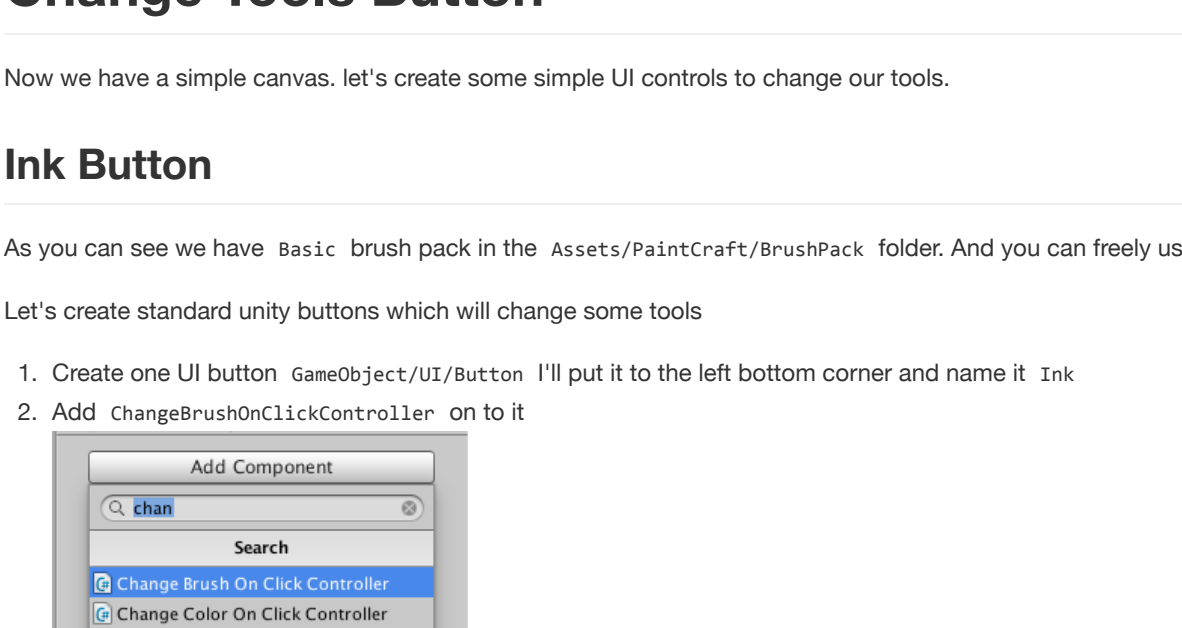
This object already contains necessary configs, but it won't work immediately because we need to tell our canvas where we want to draw.

### Page Config

Let's create a new Page Config. This object will keep configuration just the size our canvas will be.

We'll create a new folder for this `Assets/Configs`

And then click right mouse button and select `Create/Paintcraft/BlankPageConfig` menu

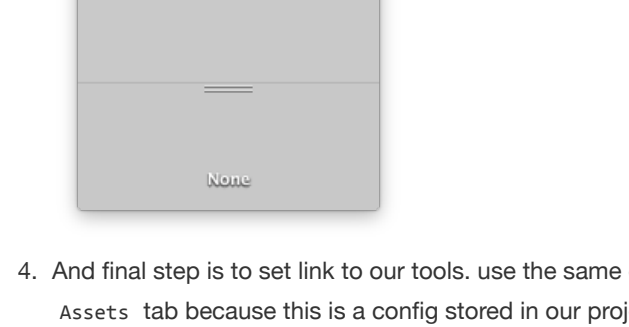


And now let's provide several settings for this config. Select this config in the project window and adjust settings in the inspector

- Unique Id # (0) this settings must be unique across all pages inside your project
- Size # (1) Actual size of your result image (like 1920x1080)

1. We use this id to store some settings like preview icon and other meta data
2. It's not mandatory to have canvas the same size as the screen, so we can make it smaller or bigger as you wish.

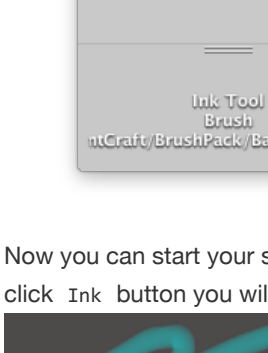
Here is what we can get



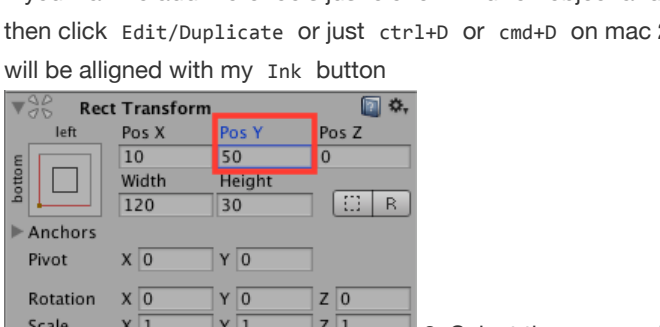
### Apply new page config

Now you have a page config and you can use it with your canvas object

1. Select `Canvas` object in hierarchy

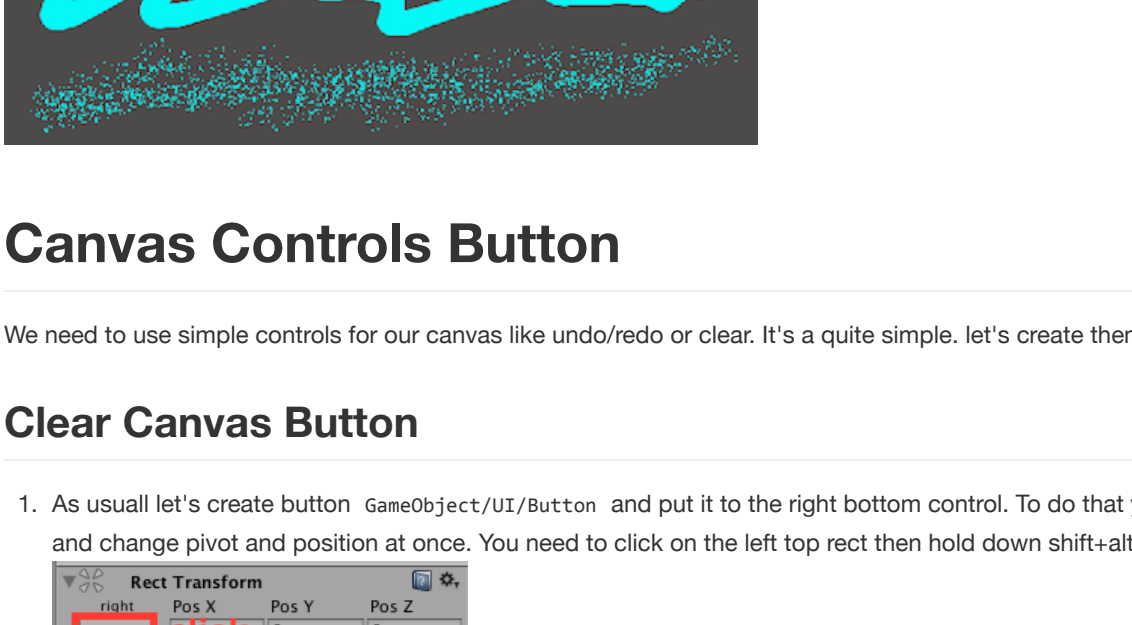


2. Link your page config to the `Page Config` property



3. You can also change default background color of your canvas in the same `Canvas Controller` component change default bg color. I'll use default gray color

Here you go. Now press play button and you will be able to draw on your screen



## Change Tools Button

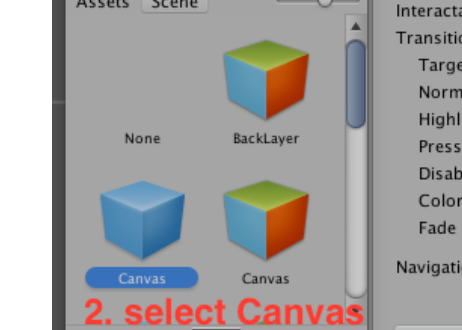
Now we have a simple canvas, let's create some simple UI controls to change our tools.

### Ink Button

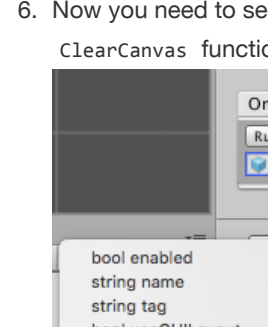
As you can see we have `Basic` brush pack in the `Assets/Paintcraft/BrushPack` folder. And you can freely use them in your projects.

Let's create standard unity buttons which will change some tools

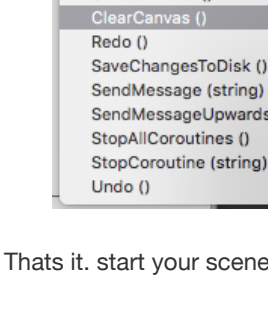
1. Create one UI button `GameObject/UI/Button` I'll put it to the left bottom corner and name it `Ink`
2. Add `ChangeEventOnClickController` on to it



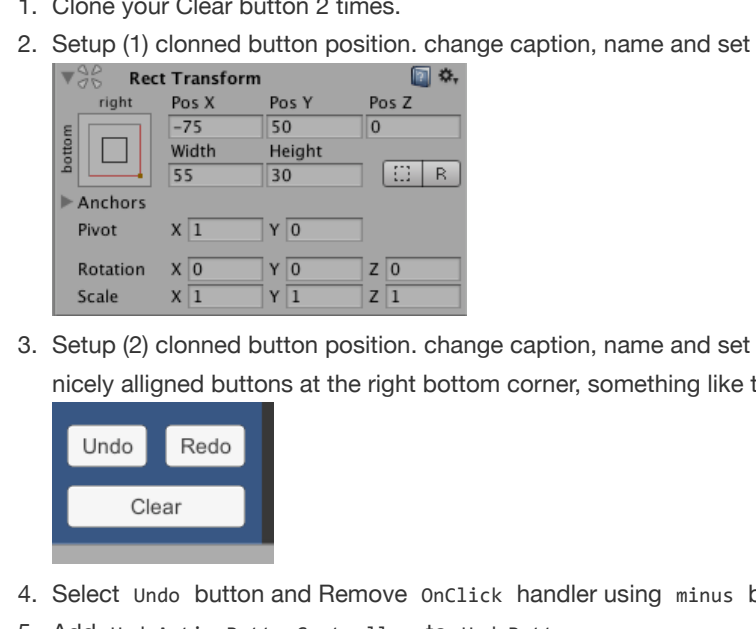
3. Now you need to set `LineConfig` reference to our prefab subcomponent on our scene `Paintcraft/Canvas/LineConfig`. Actually you can click right click on the right side of this property and it will show you just one possible object, because we have only one right now



4. And final step is to set link to our tools, use the same circle and select `Ink` config from our basic pack. Don't forget to switch to `Assets` tab because this is a config stored in our project and not a scene object

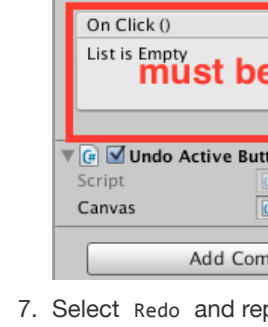


Now you can start your scene and try to draw with the brush (because it's already selected in our line config as default tool) and after you click `Ink` button you will draw with ink



### Spray Button

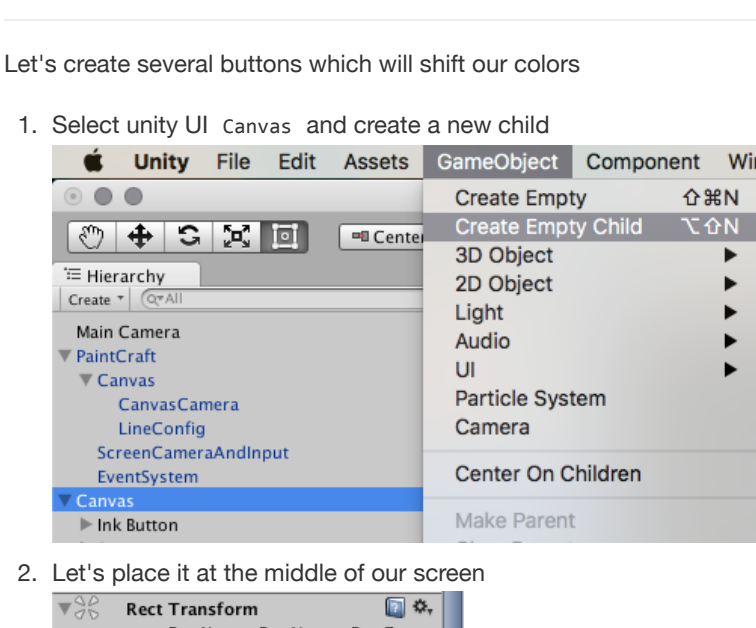
If you want to add more tools just clone `InkButton` object and change reference to the tool 1, to clone object select `Ink` button and then click `Edit/Duplicate` or just `ctrl+D` or `cmd+D` on mac 2. Change position of this object, I'll change `PosY` in the rect transform so it will be placed with my `Ink` button



3. Select the `Spray` object as you did it in previous fourth step 4. And now rename your button and text value to your tool name `Spray`

And now let's change our default tool in our line config. Select `Paintcraft/Canvas/LineConfig` object. Click on `Brush` and select `Ink`

Now when you start your scene you will draw with `Ink` and you will be able to switch this tool to `Spray` and back.

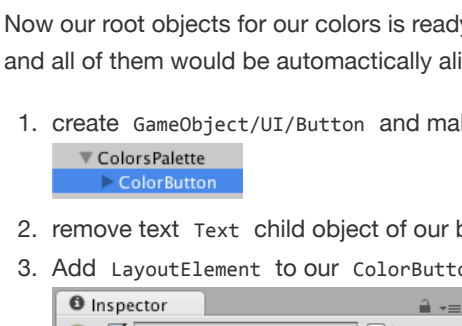


## Canvas Controls Button

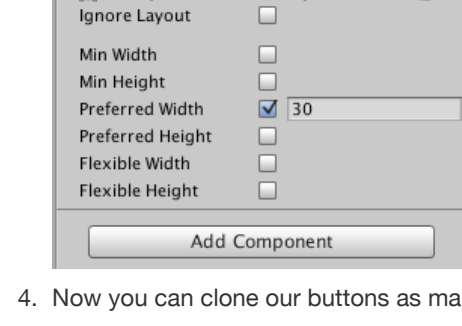
We need to use simple controls for our canvas like undo/redo or clear. It's a quite simple, let's create them

### Clear Canvas Button

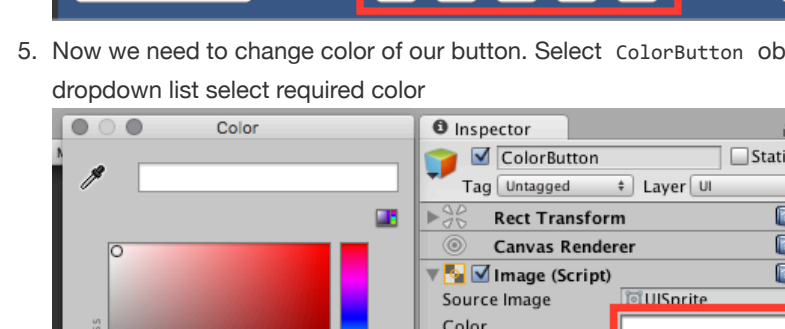
1. As usual let's create button `GameObject/UI/Button` and put it to the right bottom control. To do that you can select `RectTransform` and change pivot and position at once. You need to click on the left top right then hold down shift-alt click the bottom right corner



2. Set small offset, set `PosX=-10` `PosY=10` so it won't be right on the corner and change `width=120`.
3. Name this button as `Clear`
4. now we need to add event to this button. Click + in `onClick()` event

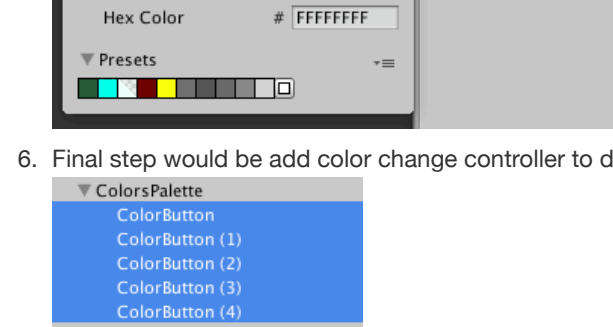


5. Click on the object field and select `Canvas`



You can also drag and drop `Paintcraft/Canvas` scene object to this property

6. Now you need to select the `Canvas.ClearCanvas()` function. To do that click on `No Function` button select `CanvasController` and `ClearCanvas` function

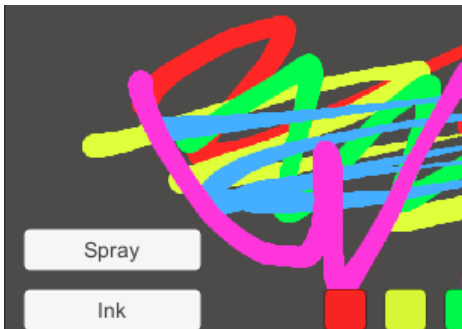


That's it, start your scene, try to draw something and then clear canvas

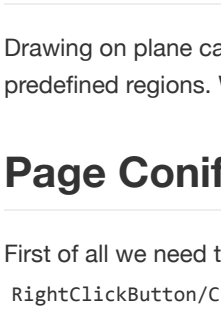
### Undo/Redo

Let's repeat button creation process for undo/redo

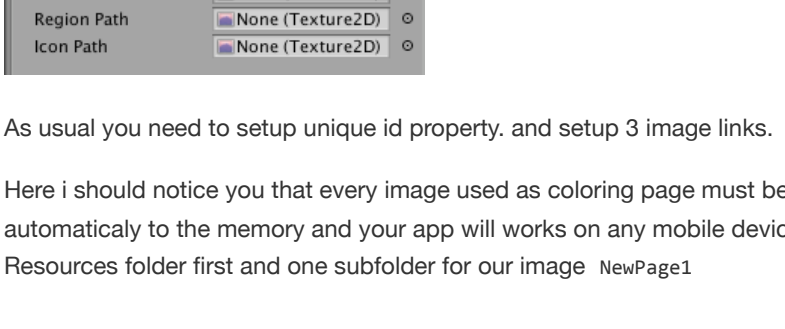
1. Clone your `Clear` button 2 times.
2. Setup (1) cloned button position, change caption, name and set following position: `PosX=-75` `PosY=50` `width=55`



3. Setup (2) cloned button position, change caption, name and set following position: `PosX=-10` `PosY=50` `width=55`. You should get neatly aligned buttons at the right bottom corner, something like this:



4. Select `undo` button and Remove `onClick` handler using minus button
5. Add `UndoActiveButtonController` to `UndoButton`
6. Select `Canvas` for this controller you will get this. Make sure that `onClick()` is cleared properly



7. Select `redo` and repeat steps 4 to 6 but this time add `RedoActiveButtonController`

Your systems buttons are ready and you can start your scene and check that everything is works properly. You can even see that `undo/redo` is clickable only if you have something to undo or redo. By default history size is 10.

## Line Color and Thickness

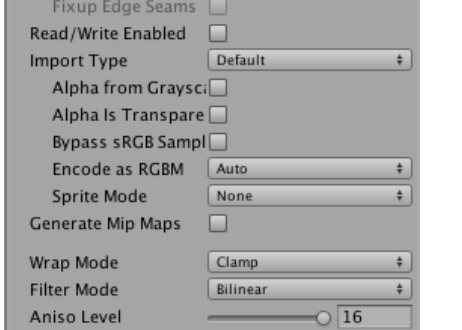
### Color

Let's create several buttons which will shift our colors

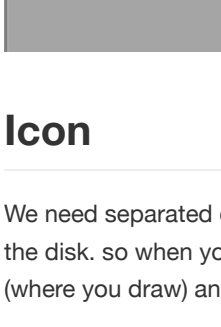
1. Select unity UI `Canvas` and create a new child



2. Let's place it at the middle of our screen



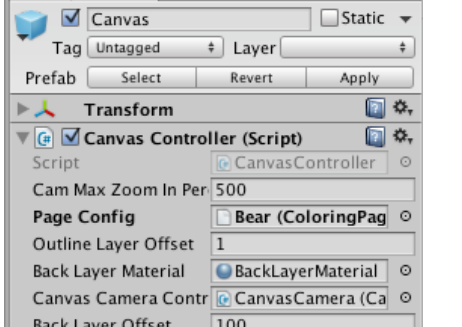
3. Name it `ColorsPalette` and set `PosY=10`
4. Add standard unity `HorizontalLayoutGroup` script on it set `spacing=10` and `childForceExpand=false`
5. On the same `ColorsPalette` object add `ContentSizeFitter` and set `horizontalFit=PreferredSize` so your object will looks as



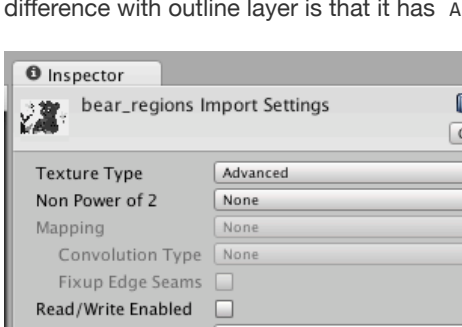
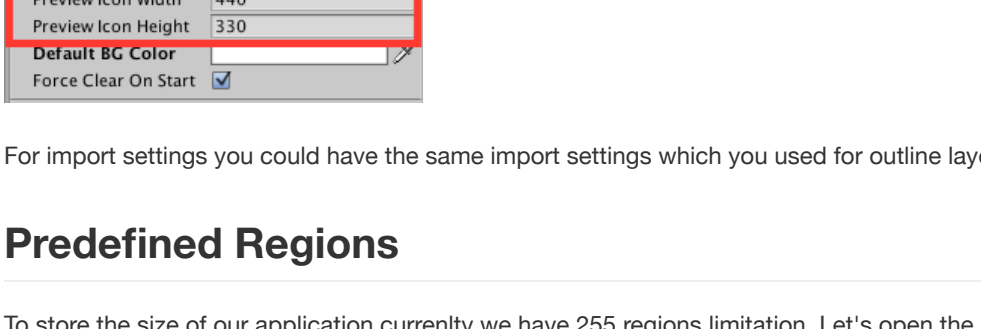
Now our root objects for our colors is ready and it would have nice feature. It would grow automatically and you can add several colors and all of them would be automatically aligned, so let's do that

1. create `GameObject/UI/Button` and make it as child for our `ColorsPalette` element and name it 'ColorButton'

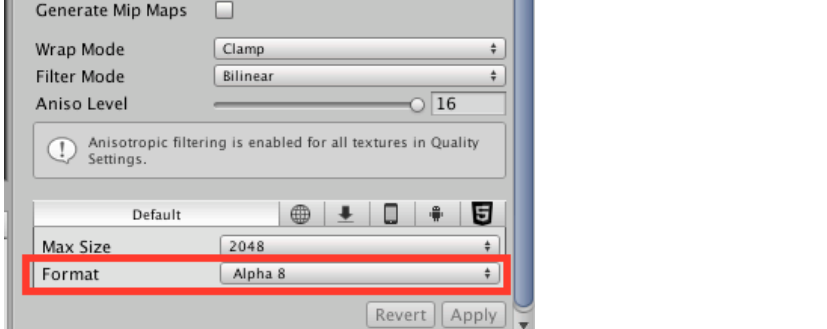
2. remove text `Text` child object of our button, so you won't have any caption on it
3. Add `LayoutElement` to our `ColorButton` and set 'PreferredWidth=30'



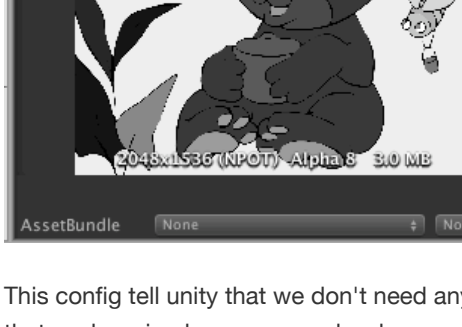
4. Now you can clone our buttons as many times as you want to have in your final app. I'll make 5 colors, here is hierarchy and how it looks on scene



5. Now we need to select color of our button. Select `ColorButton` object `ImageComponent` and click on `Color` property and in dropdown list select `changed color`



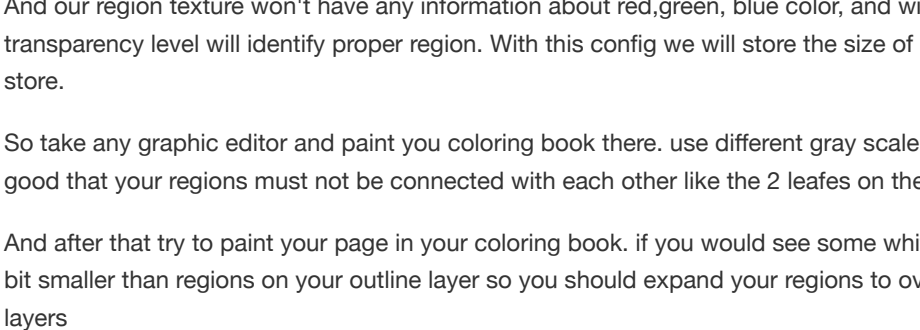
then add `ChangeColorOnClickController` And finally copy our colors from our `Image` component to this controller. To do that you need select component menu little gear at the top right corner of our component and select `Copy Color From Image`



and then select `LineConfig` as you did that in text change step.

Now check all your color buttons and make sure that all of them has link to `LineConfig` and every one has the same color as button.

Click play button and try to change line of your color

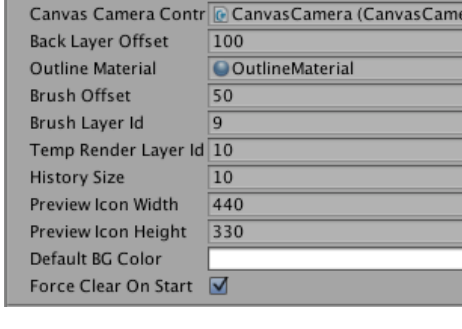


## Coloring Page

Drawing on plane canvas it's cool but let me show you how to create a coloring page, where all you lines would be restricted to predefined regions. We would need to setup 3 image and I'll explain you where each of them is used for.

### Page Config

First of all we need to create a new page config go to your `Configs` folder and create new page config file, but this time select `RightClickButton/Create/Paintcraft/ColoringPageConfig`

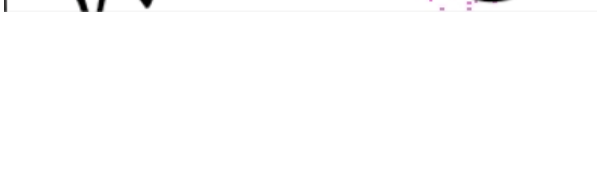


As usual you need to setup unique id property, and setup 3 image links.

Here I should notice you that every image used as coloring page must be stored somewhere in `Resources` folder so it won't be loaded automatically to the memory and your app will works on any mobile device even if you will have on handred pages. So let's create `Resources` folder first and one subfolder for our image `NewPage1`

### OutlineLayer

In coloring page we use 2 layers. One layer is the same as on blank page where we draw our graphics, and 2 layer contains our actual image. On the screenshot below we hasn't drawn anything and our back layer is blank and white. Outline layer contain Bear image.



And as you see Outline layer must have transparency, you could have any colors on it, something like shades color injections and so on, it's not mandatory to have black and white image. You can open demo project pictures and see that it has some transparent shades.



Then you need to fix import settings. By default unity could use power of 2 settings so your image would be stretched, to fix that select your outline image and then in inspector change `TextureType=Advanced` and then `Non Power of 2 = None` and also we don't need `MipMaps` so you could disable them as well. Here you could check what i have.



### Icon

We need separated downscaled icon because of the following reason. When you make any changes on the canvas we store this state on the disk, so when your player would comeback it will be able to work with this image. And we do not store mix of the background layer (where you draw) and outline layer and it's just a lines which you made. The stored layer is looks like this



And we need to blend this with outline layer. It won't work on older device if we will blend this with original outline layer, because each image would consume too much memory, and this is why we need to add downscaled image.

It depends on you which size of image do you want to have. If you will open canvas controller you will see the settings for the icon size by default it 448x330



For import settings you could have the same import settings which you used for outline layer

### Predefined Regions

To store the size of our application currently we have 255 regions limitation. Let's open the import settings of our region image, main difference with outline layer is that it has `AlphaFromGrayScale` option and `Alpha` format



This config tell unity that we don't need colors except of red, green, blue, color, and will have just alpha transparency and each transparency level will identify proper region. With this config we will store the size of our result package which you will submit to the store.

Do you remember what is RGB and how colors usually looks in inspector. Every color is a mix of red and blue color, and each of this color could have a value from 0 to 255 like in the unity color picker



And our region textures won't have any information about red, green, blue color, and will have just alpha transparency and each transparency level will identify proper region. With this config we will store the size of our result package which you will submit to the store.

So take any graphic editor and paint you coloring book there, use different gray scale for every region which you want to have, and it's good that your regions must not be connected with each other like the 2 leaves on the bear image above.

And after that try to paint your page in your coloring book, if you would see some white dots or glitches - that's mean your regions is a bit taller than regions on your outline layer so you should expand your regions to overlap all semitransparent pixels on your outline layers

### Final config

At this point you should have all 3 images, let's setup links to them in our config file



If your images would be stored not in `Resources` folder, this inspector will notify you about this. Also if you will change location of this images you would need to setup this link again, unity won't change links to them automatically as it does for scene or prefab objects.

Now you could select this config in your `CanvasController` component. Also I'll change default background color to `white`



And now you could start your scene and start drawing your image

