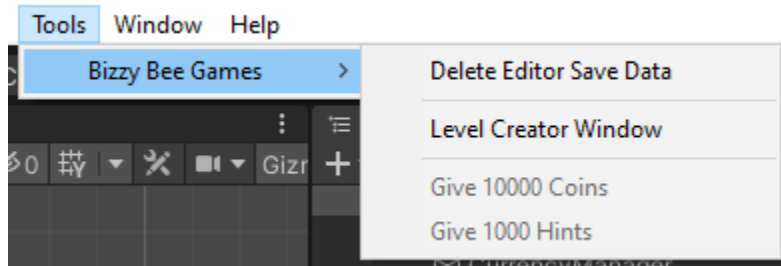


Picture Coloring Documentation

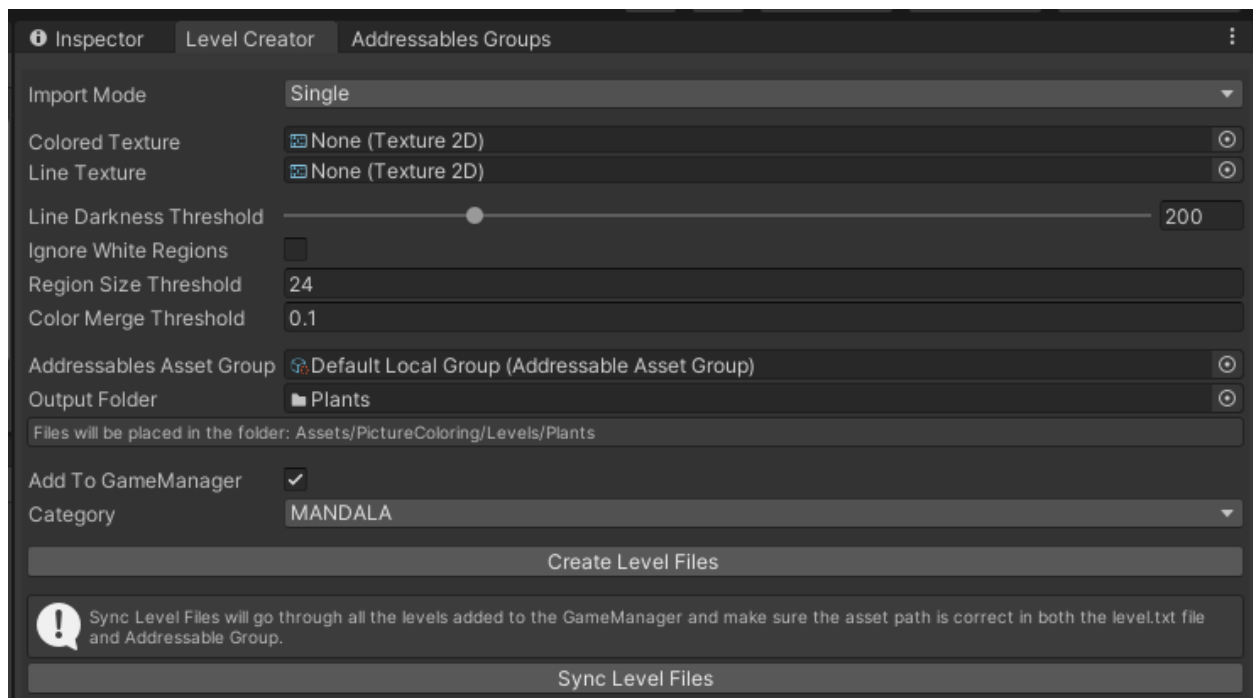
Creating Level Data Files	2
Level Creator Window	3
Create Level Files	4
Image Requirements	5
Batch Mode	6
Project	7
Category List Item	8
Library Level List Items	8
My Works List Items	9
Color List Items	10
Picture Settings	12

Creating Level Data Files

Use the Level Creator Window to generate the level files needed to play the game. To open the window select the menu item Tools -> Bizzy Bee Games -> Level Creator Window.



Level Creator Window



Import Mode: Single lets you import one image from the project window. Batch lets you import multiple images at once by selecting the folder where the images are located (The folder does not need to be in your project, it can be anywhere on your computer).

Colored Texture: The texture from the project folder which should be used as the colored in image. (See requirements below)

Line Texture: The texture from the project folder which should be used as the lines. (See requirements below)

Line Darkness Threshold: The “darkness” threshold for a pixel to be considered a line or not. For instance, setting to 255 means only solid black (#000000) pixels will be considered line

pixels.

Ignore White Regions: If selected, any white areas on the colored texture will be ignored, ei. the player will not need to color them in to complete the level.

Region Size Threshold: The minimum number of pixels in a region to be considered a region the player must color in.

Color Merge Threshold: The threshold for similar colors to be merged together (EX. Two shades of red which look the same to the naked eye). Set to 0 for no color merging.

Set Power Of 2: Sets the “Non Power Of 2” setting on the images for power of 2 compression.

Filename: The name to get the generated files. If blank then the name of the Colored Texture file will be used.

Addressable Asset Group: The addressable asset group that the generated files will be added to. Addressables are used to load level files when the game runs.

Output Folder: The folder to place the generated files into. The generated files must be placed in the Resources folder, if the selected folder is not in a Resources folder then a Resources folder will be created and the images will be placed into that folder instead.

Add To GameManager: If selected, the generated level file will be added as a level to the selected Category in the GameManager.

Category: The Category from the GameManagers Categories list to add the generated level to.

Create Level Files

The **Create Level Files** button will generate a few files:

1. **level.txt** : Small text file with the id of the level and the path to the folder it is located in. The game will use the path stored in level.txt to load the other files using the Addressables system.
2. **bytes.bytes** : Contains information about the level such as the different regions that can be color and the colors that are assigned to each region.
3. **atlas_#.png**: One or more texture atlases used to draw the levels in the game.

If you did not selected Add To GameManager then you will have to manually drag the level.txt from the Resources folder into a level on the GameManager:

If Add To GameManager was enabled then you can run the game and see the new level. To manually add the level to the game, select the **GameManager** in the scenes hierarchy and expand the Categories, category name you want to add the level to, the Levels and drag the level.txt file into the **Level File** field:

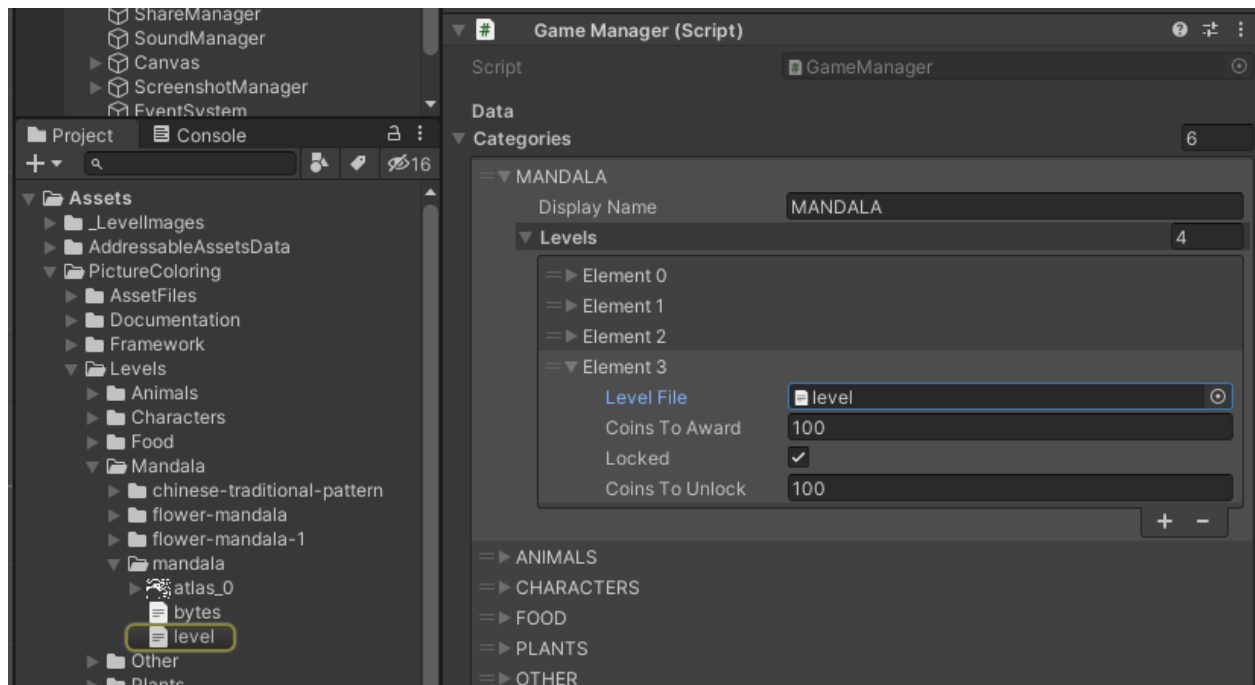


Image Requirements

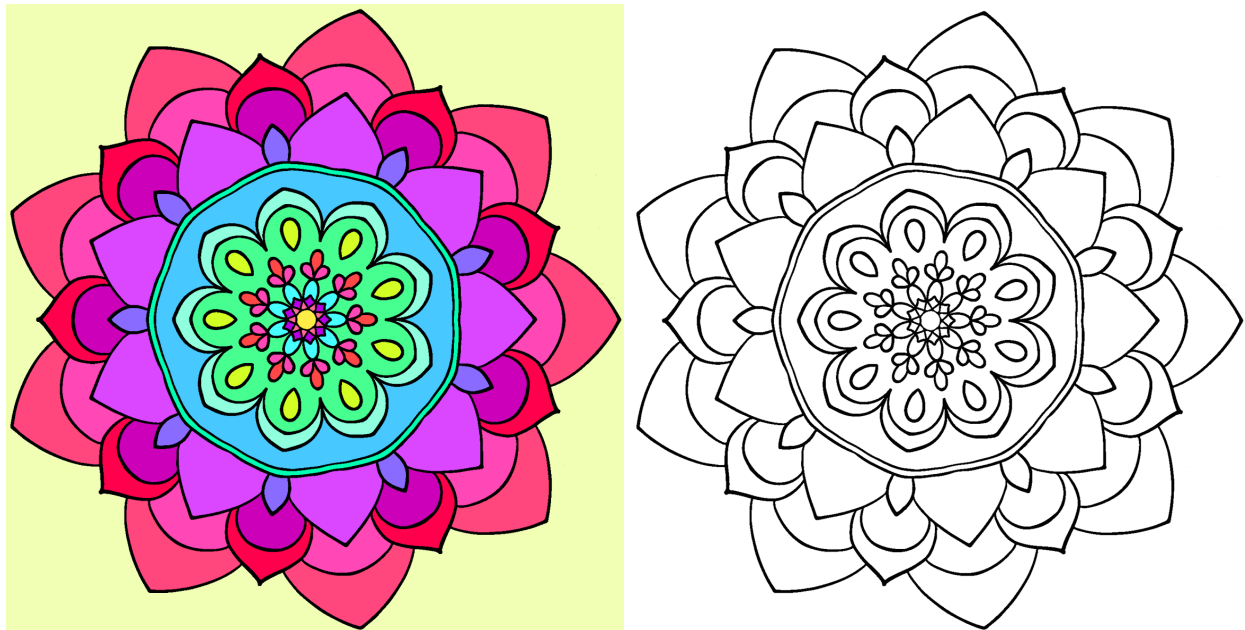
Requirements for the **Colored Texture**:

1. Needs to be the exact same size as the Line Texture.
2. Needs to have Read/Write enabled. (Not necessary for Batch mode)
3. Fully colored in, white regions can be set to be ignored (Not needed to be colored in)
4. No alpha, any regions with alpha pixels will default to ignored and appear as white in the game.

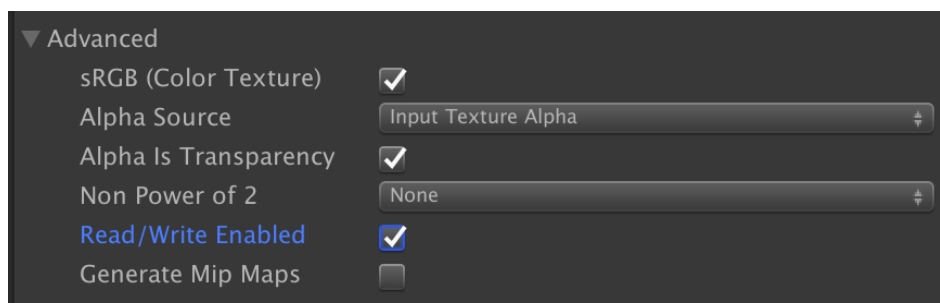
Requirements for the **Line Texture**:

1. Needs to be the exact same size as the Colored Texture.
2. Needs to have Read/Write enabled. (Not necessary for Batch mode)
3. Needs solid black lines with either white background. NO COLOR.

Example **Colored Texture** and **Line Texture**:



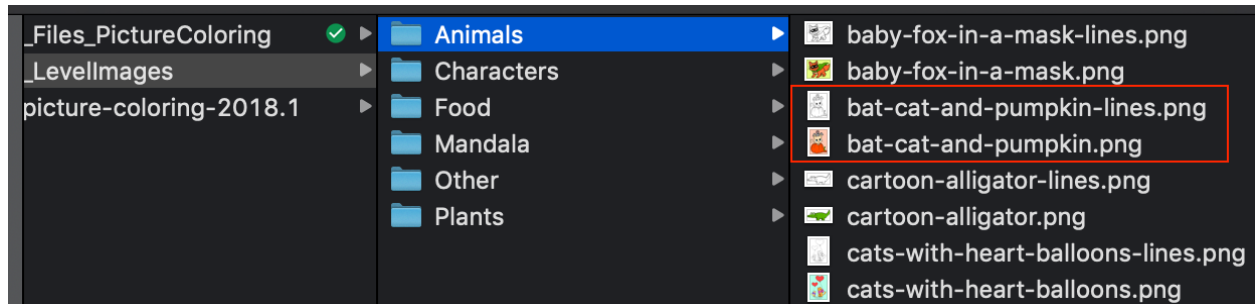
To set the textures as Read/Write enabled, select the image in your Project window and then in the Inspector, expand **Advanced** and select **Read/Write Enabled**. Then click apply at the bottom of the Inspector:



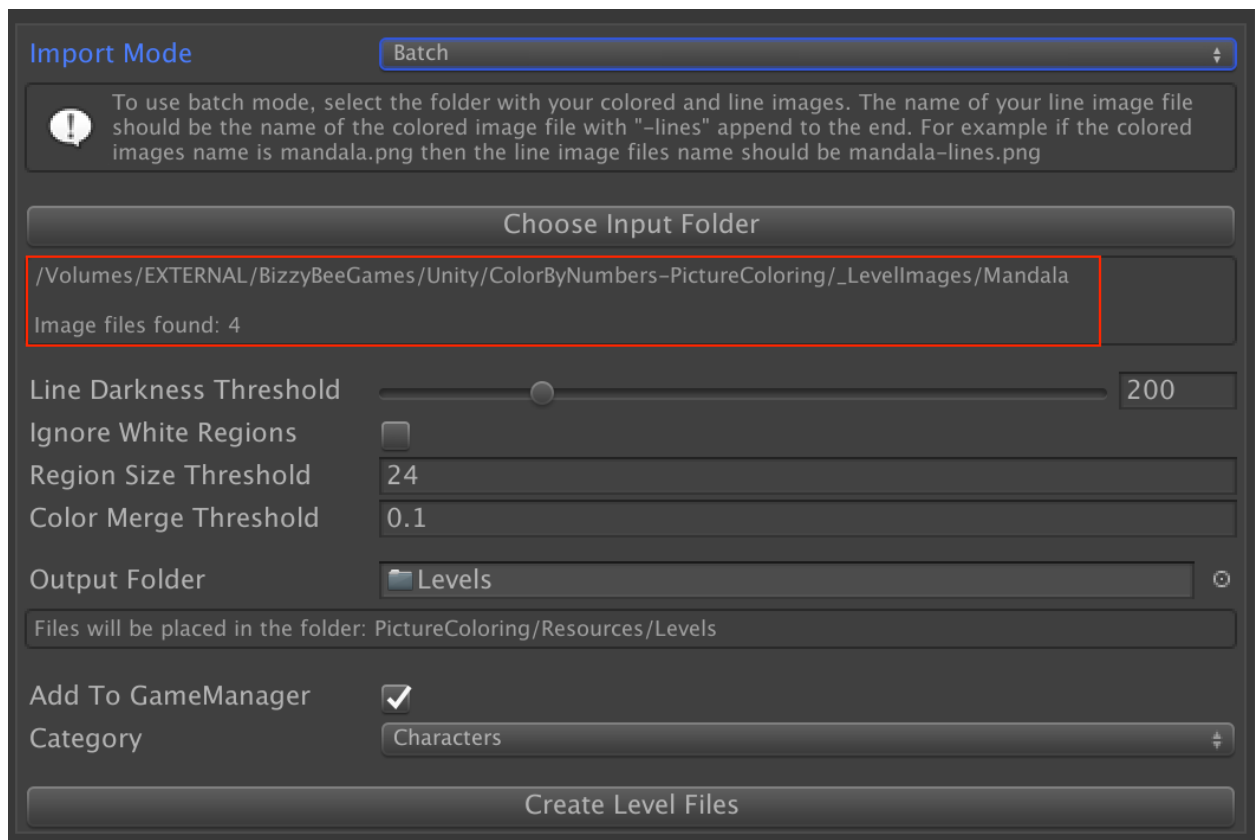
Batch Mode

Setting the window to Batch import mode will allow you to select a folder from your computer and have all PNG images in that folder imported one after another.

The line image will need to be the same name as the colored image with “-lines” append to the end. For example, if you have a colored image called “mountains.png” then the line image will need to be named “mountains-lines.png”.



When you select a folder the window will tell you how many level images it found, it will also print an error message to the Unity Console if it couldn't find the lines image for any of the images.

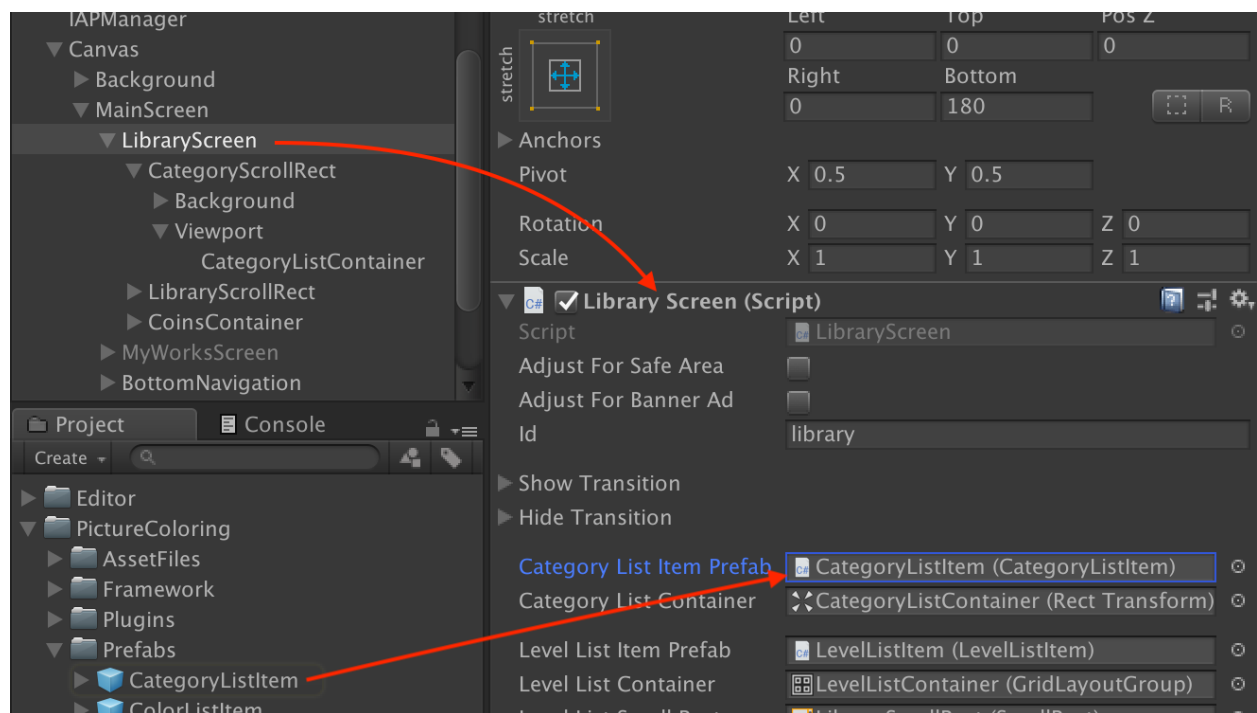


Project

This section will show you where UI elements that are changed / instantiated by scripts are located and how to edit them.

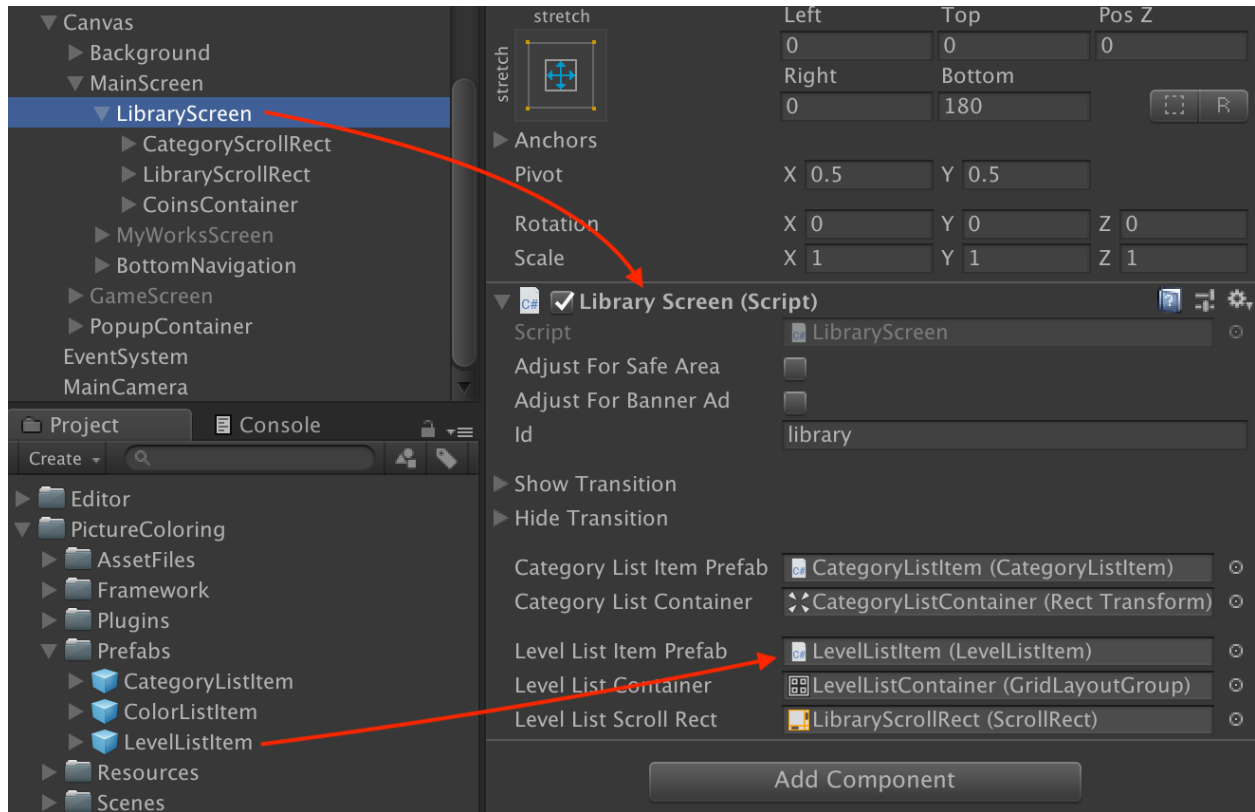
Category List Item

The category list items are generated at run time by the **LibraryScreen** component which is attached to the LibraryScreen GameObject. The LibraryScreen will instantiate a new **Category List Item Prefab** for each Category in the GameManager. The Category List Item Prefab that is being used in the asset is located at **PictureColoring/Prefabs/CategoryListItem**



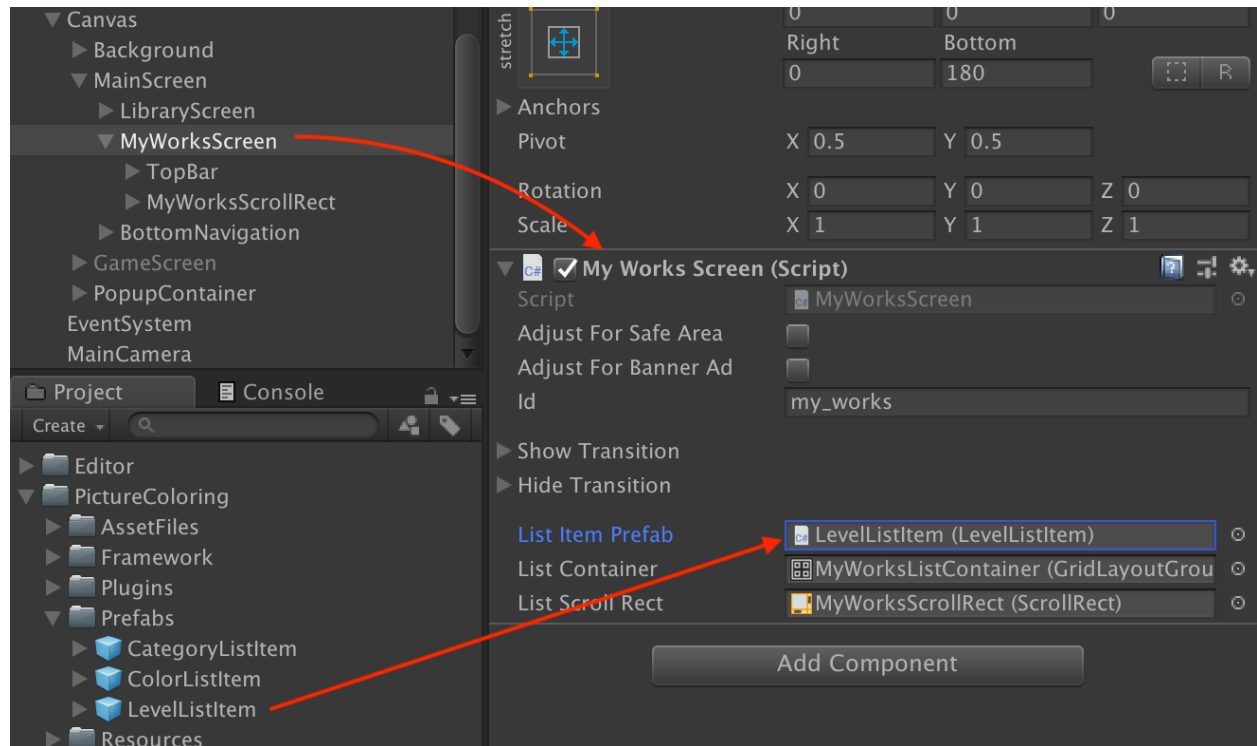
Library Level List Items

The level list items are generated at run time by the **LibraryScreen** component which is attached to the LibraryScreen GameObject. The Level List Item Prefab that is being used in the asset is located at **PictureColoring/Prefabs/LevelListItem**:



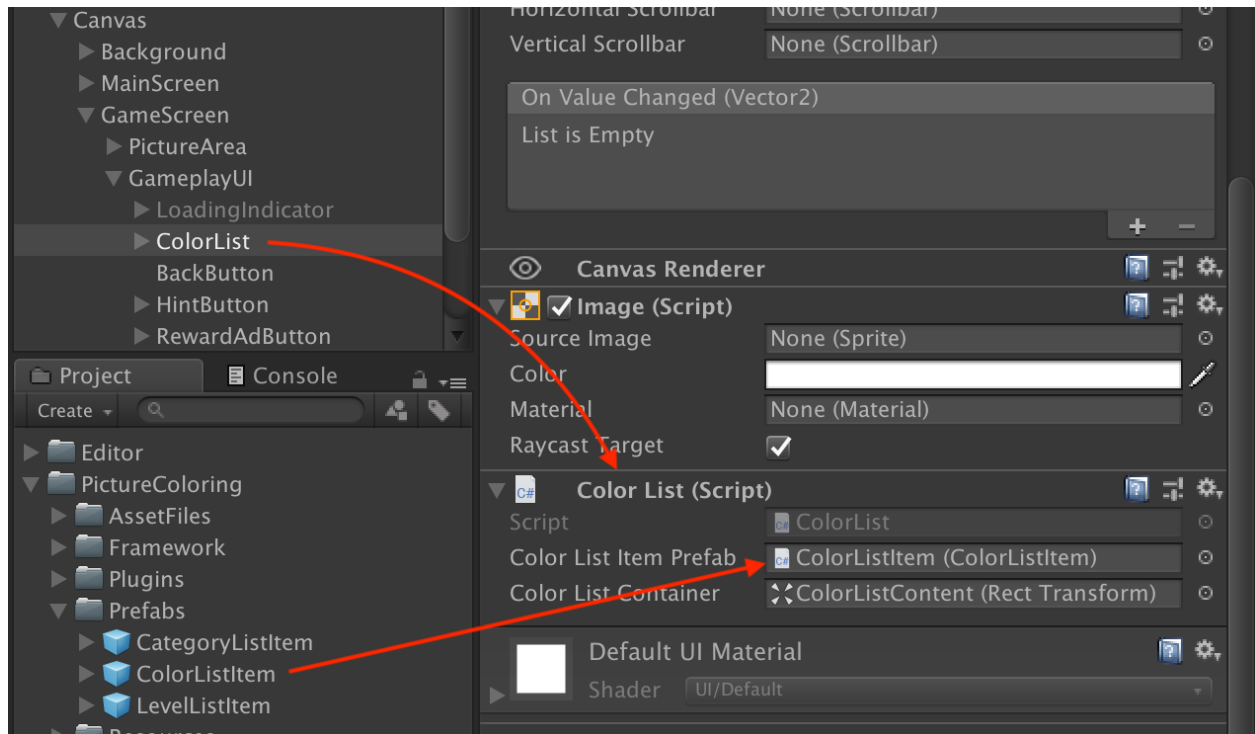
My Works List Items

The level list items on the My Works screen are generated at run time by the **MyWorksScreen** component which is attached to the MyWorksScreen GameObject. The **List Item Prefab** is used to instantiate copies of the level list items and uses the same prefab as the LibraryScreen which is located at **PictureColoring/Prefabs/LevelListItem**:

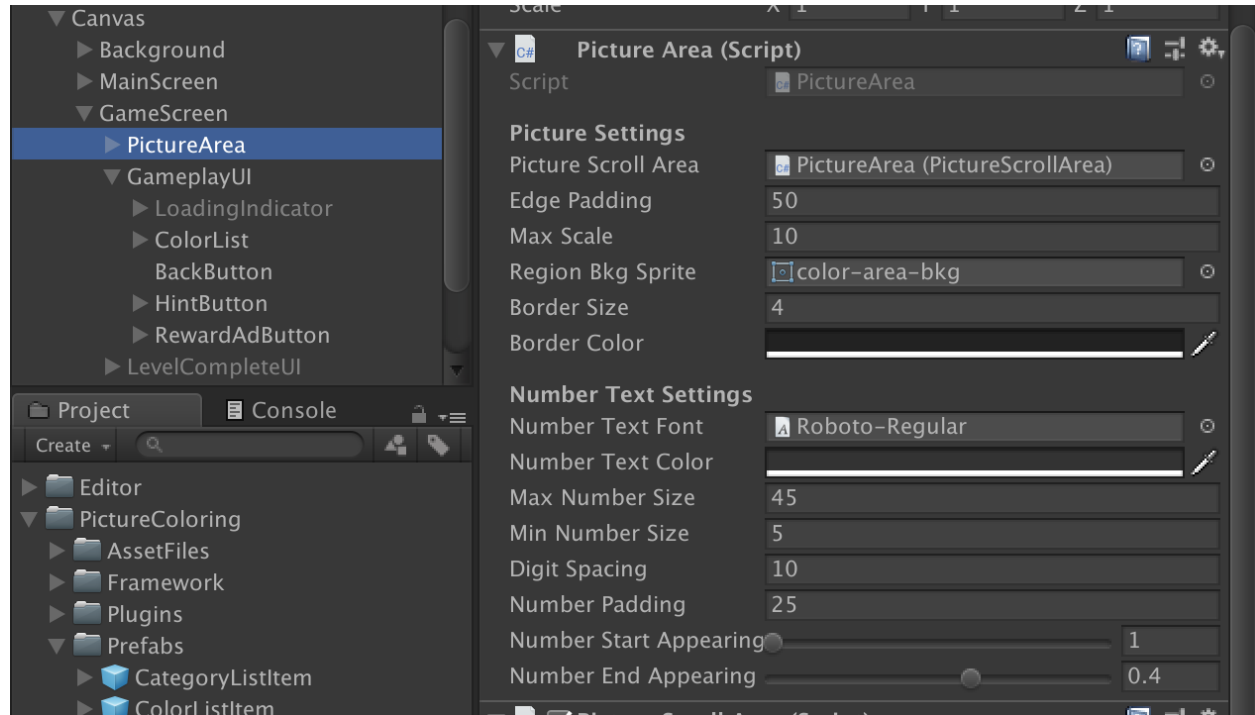


Color List Items

The color items on the GameScreen are generated when a level starts by the **ColorList** component attached to the ColorList GameObject. It uses the **Color List Item Prefab** to generate a new color list item for each color in the level. The Color List Item prefab that is being used in the asset is located at **PictureColoring/Prefabs/ColorListItem**:



Picture Settings



The appearance of the picture on the GameScreen can be changed using the settings on the **PictureArea** component which is attached to the PictureArea GameObject.

Edge Padding: The amount of space between the image and the edge of the PictureArea when the image is zoomed all the way out (At it's min scale / the start of the level).

Max Scale: The maximum amount that the image can be zoomed in by. In the above screenshot the image can be zoomed in to 10x its size.

Region Bkg Sprite: This is the sprite to use for the Image that appears when a color is selected. The sprite that comes with the asset is a repeating checkered pattern.

Border Size: The thickness of the border that appears around the image.

Border Color: The color of the border that appears around the image.

Number Text Font: The font to use for the numbers that appear on the image.

Number Text Color: The color to use for the numbers that appear on the image.

Max Number Size: The maximum size in pixels that a number can be on the image, this is to control large regions from having super big numbers.

Min Number Size: The minimum size in pixels a number can be on the image.

Digit Spacing: The amount of spacing in pixels between digits in a number if the number is larger than 9. (Ex. The number 12, the amount of space between the 1 and 2).

Number Padding: The amount of padding to apply to a number in its region so it's not super close to the lines.

Number Start Appearing: Controls when the numbers start appearing relative to the zoom/ scale of the image.

Number End Appearing: Controls when the smallest numbers appear on the image, settings to 0 means the image will have to be zoomed in all the way to see the smallest numbers.