Game Documentation

To insure a quick response to any issues with the asset please send all support requests to the following e-mail address:

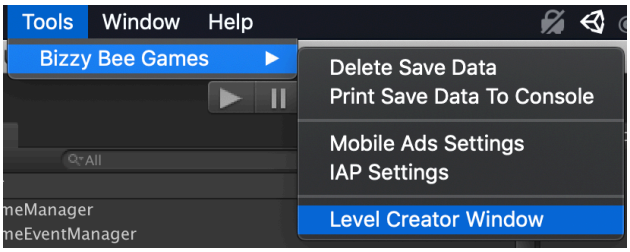
support@bizzybeegames.com

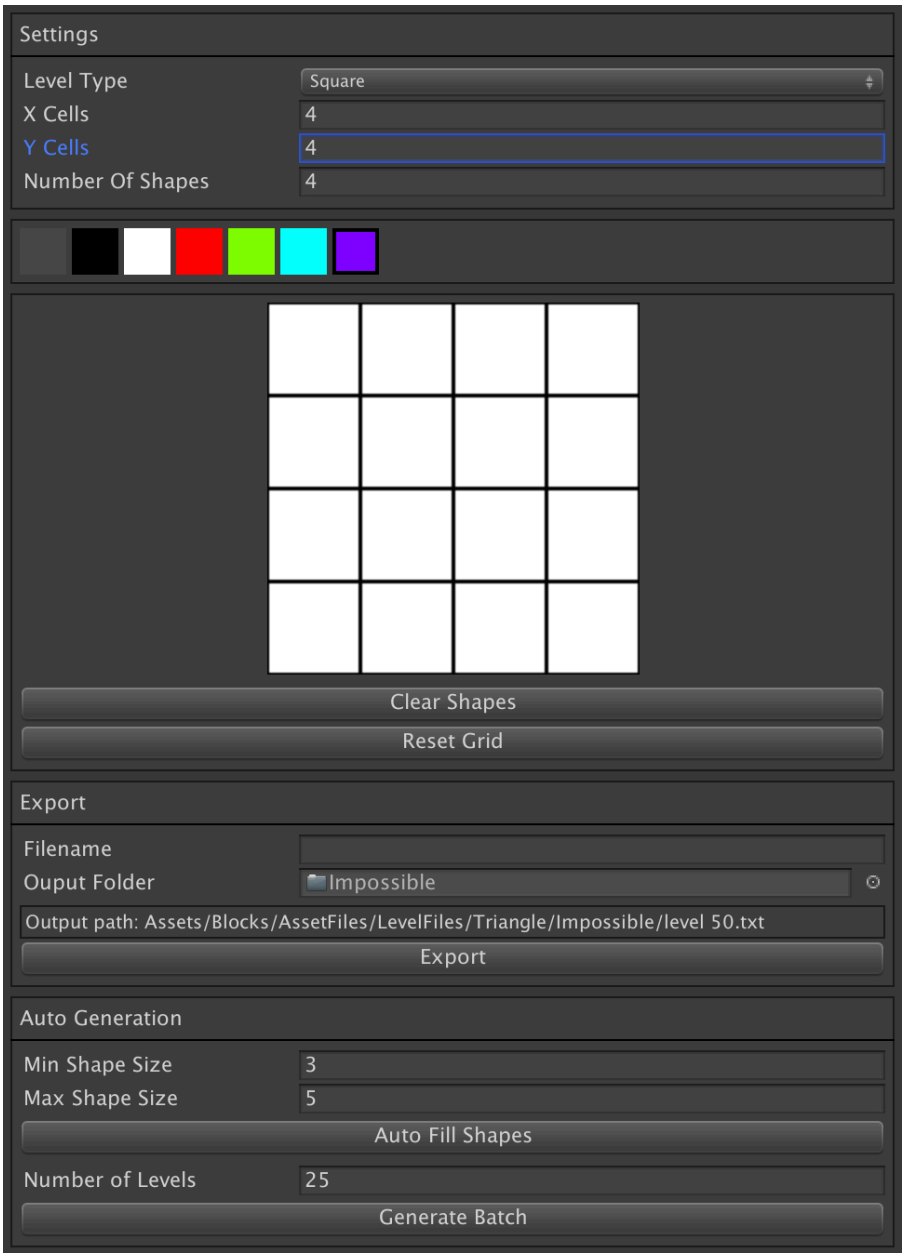
Please include the asset name and Unity version you are using. Thank you!

Table of Contents:

Creating Levels

The asset comes with a Level Creator editor window which is used to create new level files. To open the window select the menu item **Tools -> Bizzy Bee Games -> Level Creator Window**.





Settings

Level Type: The type of level you want to create, can be Square, Triangle, or Hexagon.
X/Y Cells: The size of the grid.
Number Of Shapes: The number of shapes you want to place on the grid.

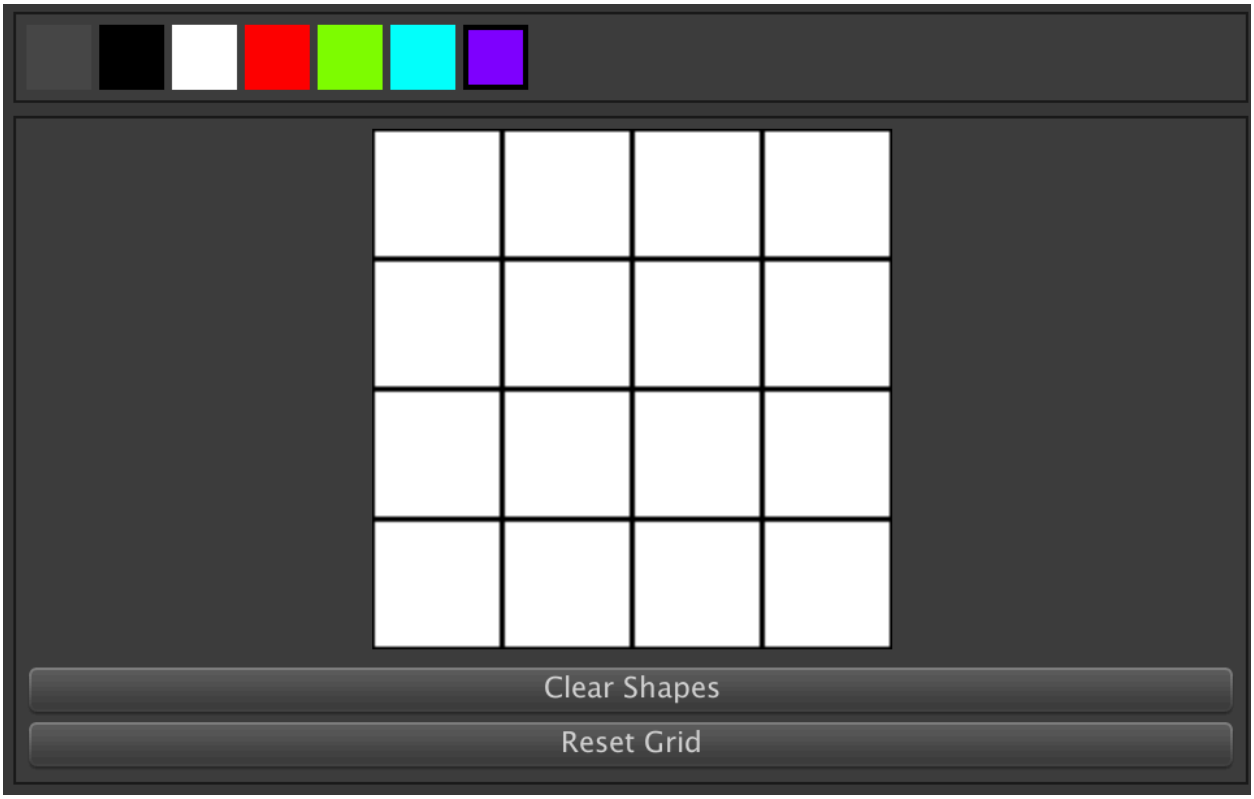
Export

Filename: The name of the level text file.
Output Folder: The folder to place the level file in. This is dragged from the Project window.

Auto Generation

Min Shape Size: The minimum number of cells in a single shape.
Max Shape Size: The maximum number of cells in a single shape.
Number of Levels: The number of levels to create when Generate Batch is clicked. This will use the grid layout (Blank/Block cells).

Manually Creating Levels



You can manually create levels by clicking on the color squares above the grid then click the cells on the grid to assign them to that color. The first 3 colors are special cell types:

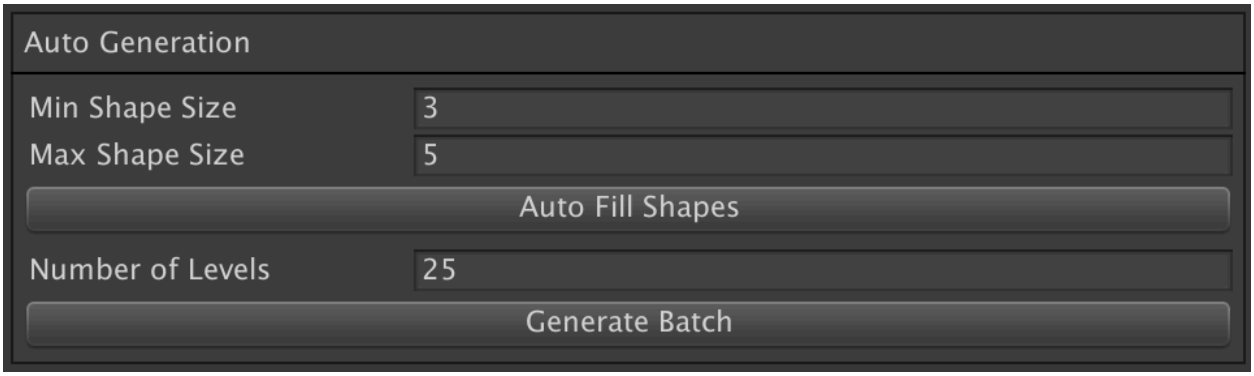
Grey: These are blank cells, when the game runs and the level is created those cells will not appear at all.
Black: These are “block” cells, no shape can occupy a block cell.

White: These are “empty” cells. A completed level should not have any white/empty cells in them. Using the white square essentially erases the cell and sets it back to default.

The color squares represent the different shapes in the level. The actual color will have no effect when the game runs, it is just so you can differentiate the different shapes on the grid in the editor window.

NOTE: You can also click and drag to quickly set a bunch of cells.

Auto Generate Levels



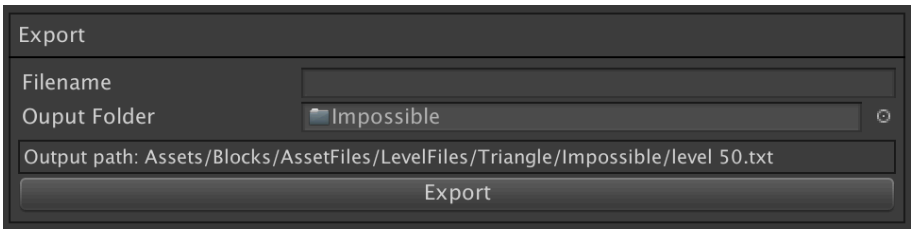
You can use the Auto Generate panel to automatically fill the grid with shapes or generate a bunch of levels at once.

The **Auto Fill Shapes** button will attempt to fill the grid with shapes. The number of shapes is set in the **Settings** panel (Number Of Shapes). Once it is done it will display the shapes in the grid on the editor window. You can then click then export the level.

The **Generate Batch** button will auto fill the grid with shapes then auto export it and it will do it for the number of levels specified.

These settings will not auto place blank or block cells, you must set those cells on the grid prior to clicking the buttons and it will fill the grid using the blanks/blocks you have set (Any shape cells placed on the grid will be removed before it auto fills it with shapes)

Export Levels



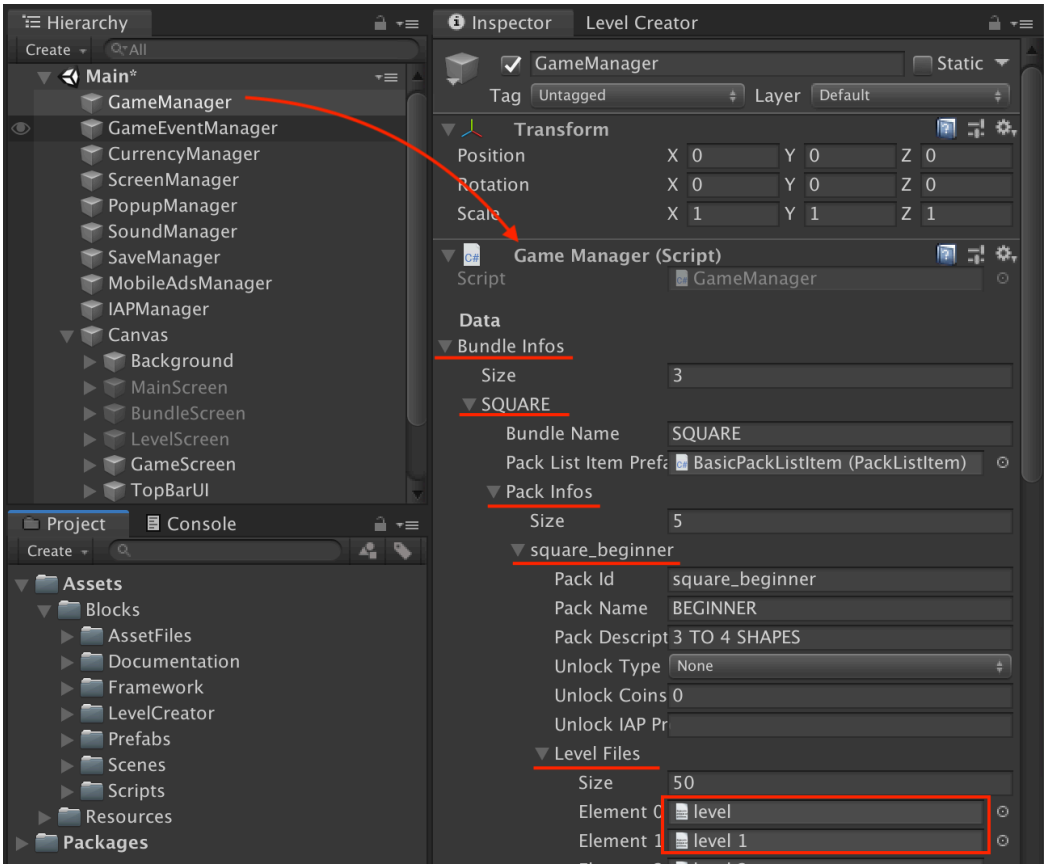
The Export panel is used to create the level text file. You can give the level file a name (Leaving it blank will name it “level” by default) and specify a folder to place the file in. This will not over write files with the same name, instead it will append a number to the end of the file name so it is unique.

These settings will be used when clicking the Generate Batch button as well.

Using Level Files

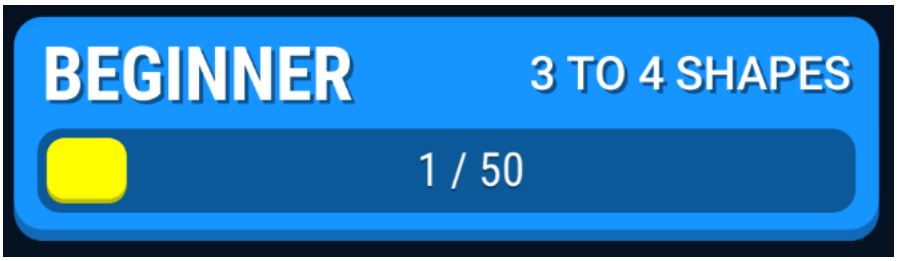
To use/play the generated level files in the game you need to assign them to a Bundle/Pack on the GameManagers inspector.

Select the GameManager in the Scenes Hierarchy then expand **Bundle Infos / <Bundle Name> / Pack Infos / <Pack Id> / Level Files** and drag the level files into the Level Files list.



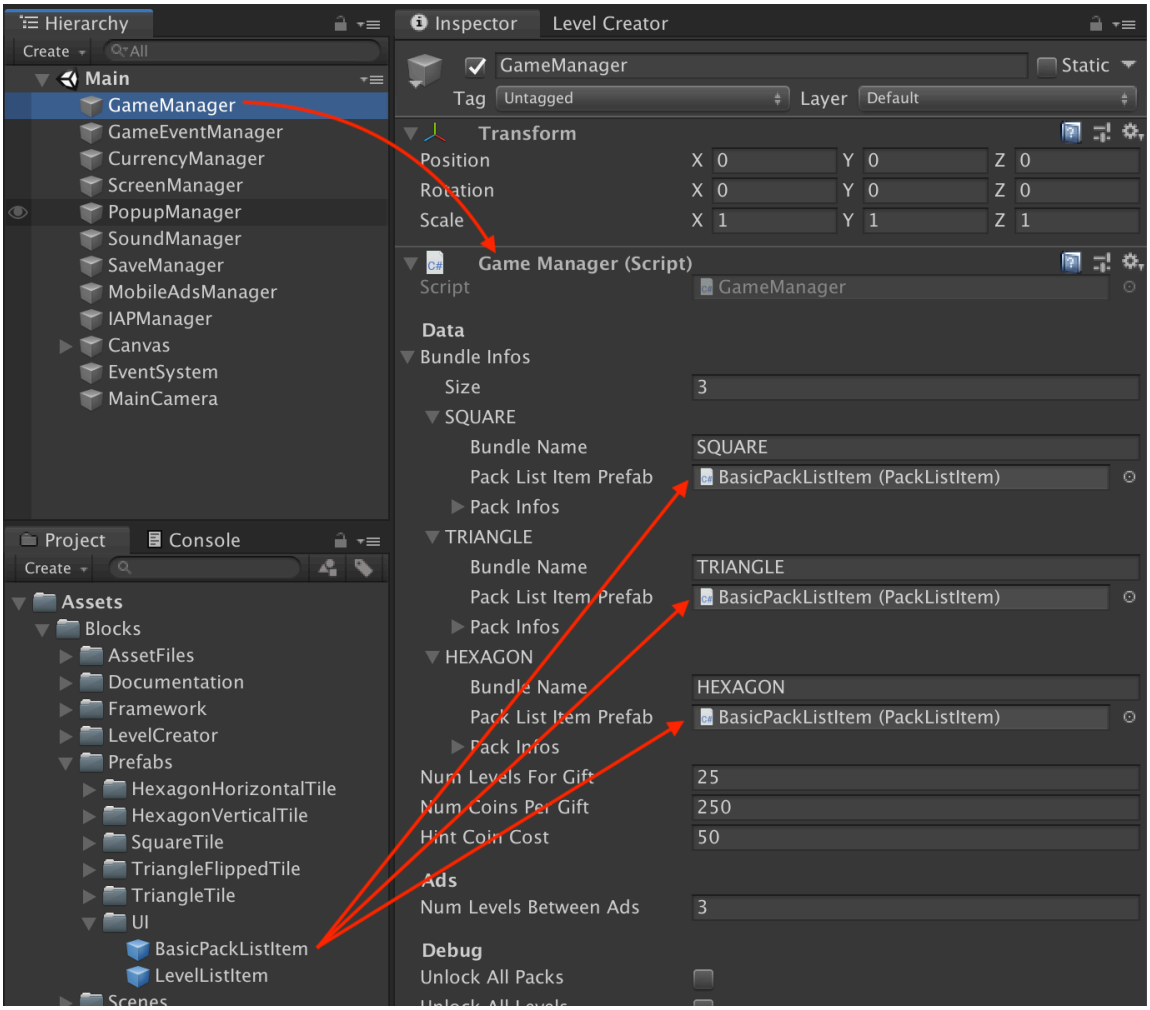
Project

Pack List Items



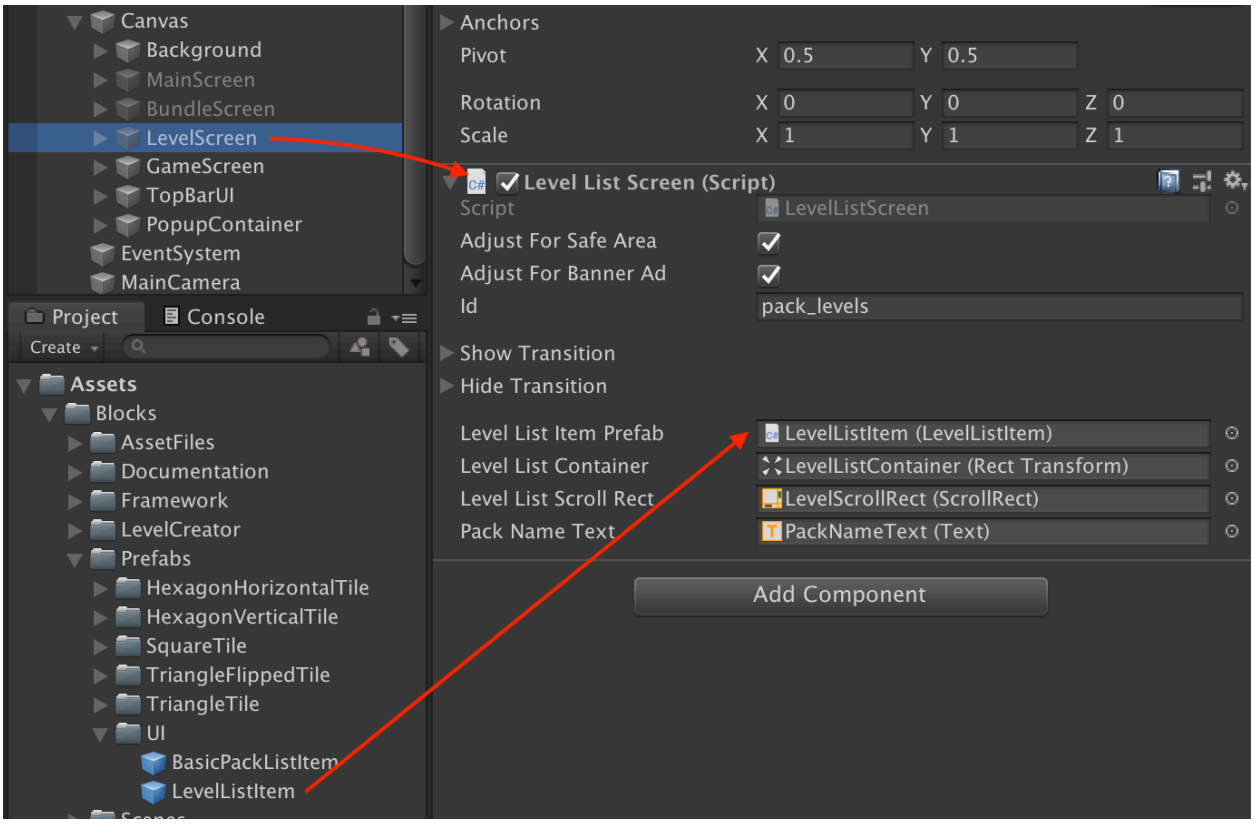
The pack list items are instantiated at run time for each pack in the game. The **Pack List Item Prefab** field located on the GameManager inspector under Bundle Infos is used to instantiate copies. The prefab that is used in the asset is located at **Prefabs/UI/BasicPackListItem**.

The asset uses the same PackListItem prefab for each Bundle however a different PackListItem prefab can be used for each Bundle.

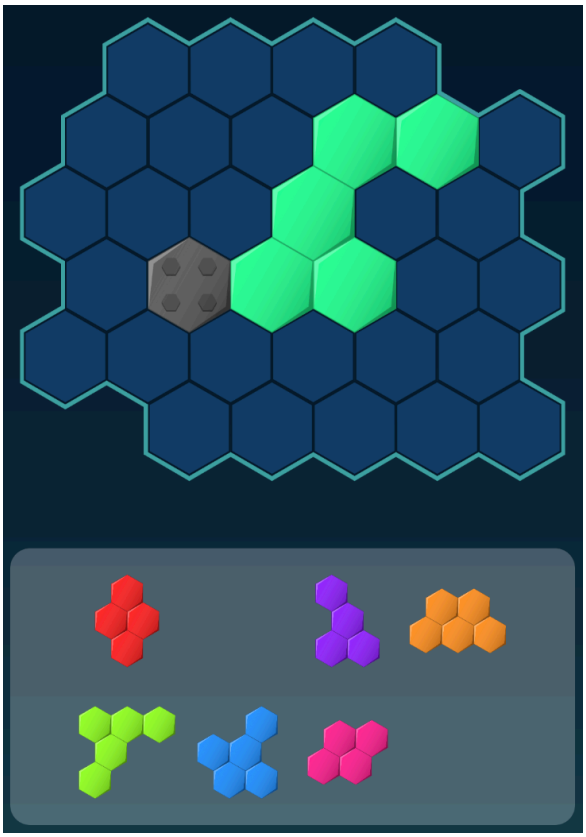


Level List Items

The level list items are generated at run time by the **LevelScreen** component attached to the LevelScreen GameObject. The **Level List Item Prefab** is used to instantiate copies, the prefab that is used is located at **Prefabs/UI/LevelListItem**:



Game Area

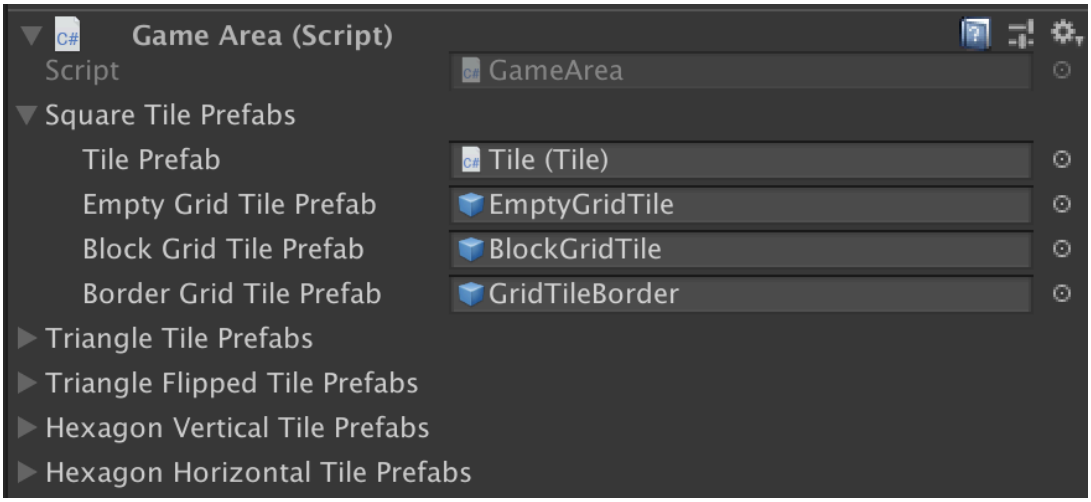


The GameArea component attached to the **Canvas / GameScreen / GameUIArea / GameUIContainer / GameArea** GameObject handles creating the grid / shapes you see on the GameScreen.

This section will show you how to change the various UI elements on this screen.

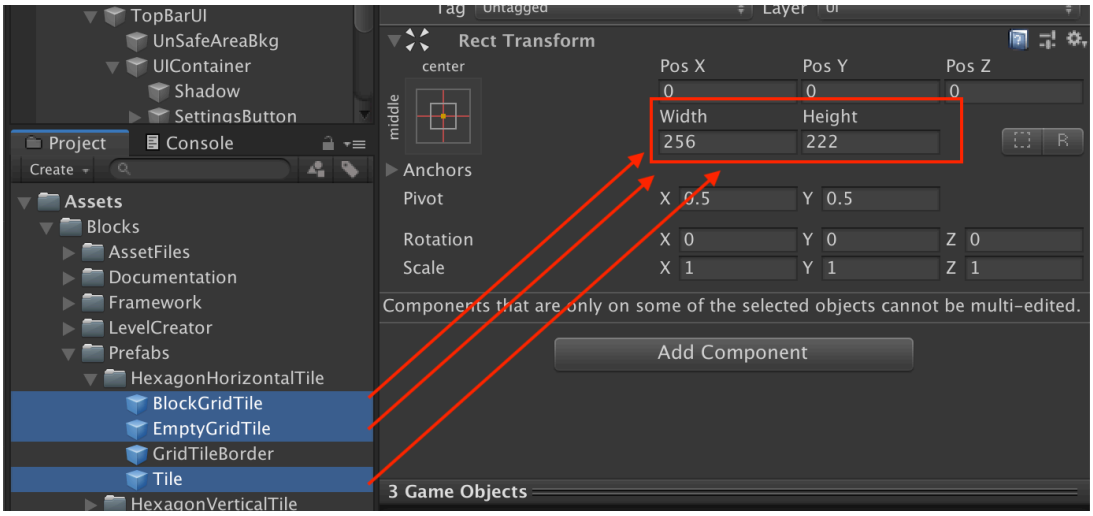
Tiles

The **Tile Prefabs** drop downs are where you set the prefabs for the individual tile cells. There are four prefabs for each level type:



Tile Prefab: This is the prefab that is used to create the colored shapes.
Empty Grid Tile Prefab: This is the prefab that is used for each cell on the grid.
Block Grid Tile Prefab: This is the prefab that is used for a block on the grid (The grid metal looking tile on the picture above).
Border Grid Tile Prefab: This is the prefab that should be used for the border around the grid. It is simply a tile that is placed behind each of the empty grid tiles.

In order for all the cells/shapes to line up properly, the Tile Prefab, Empty Grid Tile Prefab, and Block Grid Tile Prefab should all have the same width/height:



The **TriangleFlippedTile** prefabs are used for “upside down” triangles on Triangle type levels. They should also have the same width/height as the normal TriangleTile prefabs.

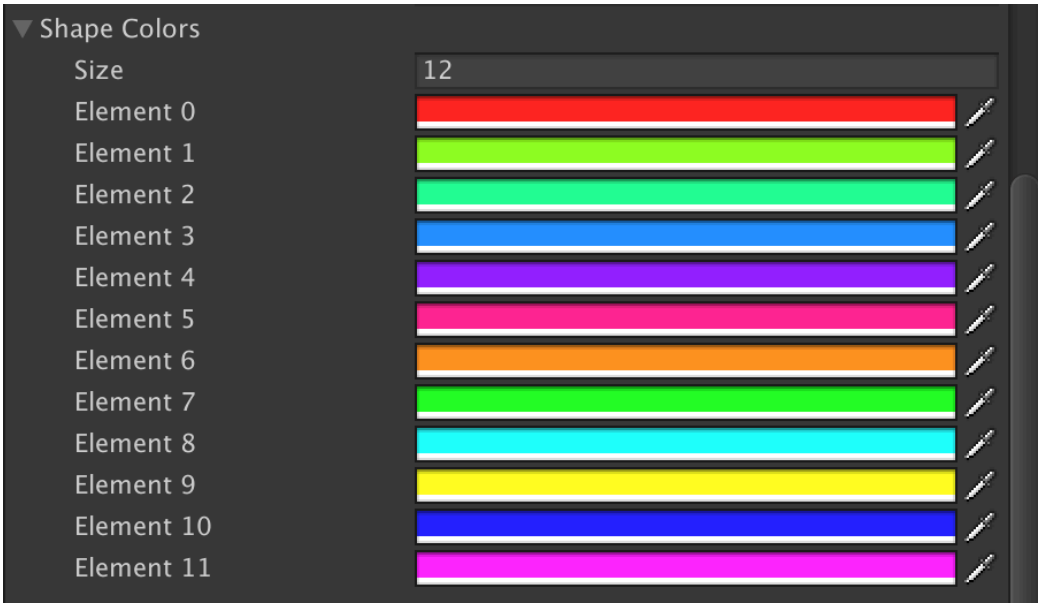
Equilateral Triangle / Regular Hexagons

You should design the triangles / hexagons to be equilateral triangles and regular hexagons:

https://en.wikipedia.org/wiki/Equilateral_triangle
https://en.wikipedia.org/wiki/Hexagon#Regular_hexagon

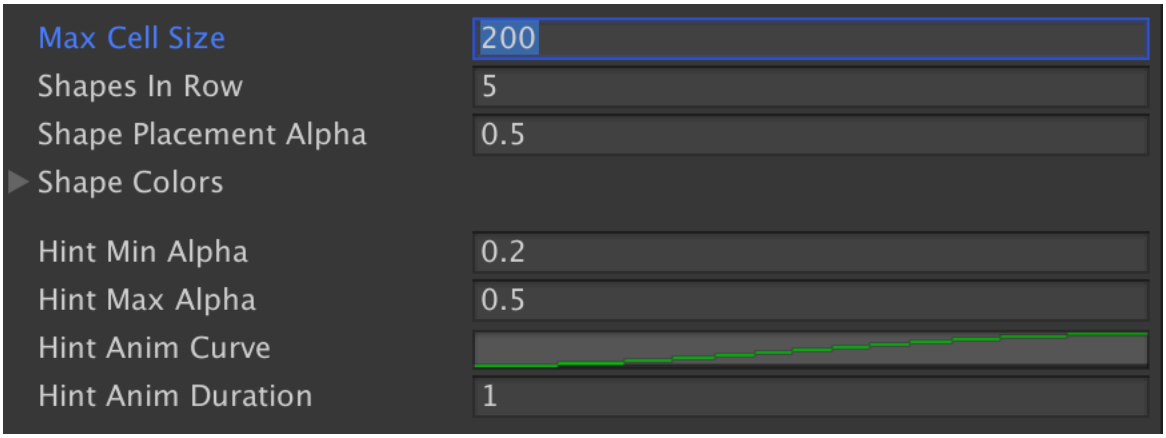
Shape Colors

The colors that are assigned to the shapes can be set in the **Shape Colors** list:



The colors are assigned sequentially so the first shape will get the first colors, the second shape will get the second color, etc. If there are more shapes in a level then there are colors then it will wrap around and start using the same colors over again.

Other Settings



Max Cell Size: The maximum width / heigh a cell can have on the screen.

Shapes In Row: Controls the maximum number of shapes that can go in a single row on in the “shapes container” under the grid.

Shape Placement Alpha: The amount of alpha to apply to the preview/placement location shape that appears on the grid when the player is dragging the shape around over the grid.

The Hint fields control the appearance of the hints when the hint button is used.

Coins

The coins amount is handle by the CurrencyManager in the Scenes hierarchy. The CurrencyManager in the Blocks asset has one currency setup “coins”. This is where you can set the starting coins that the player has when they first start the game. It also has settings for how the “out of coins” popup should appear.

Rewards

The player can get coin rewards for completing levels. The amount of levels the player must complete in a pack before being given a reward is set on the GameManager in the **Num Levels For Gift** field and the **Num Coins Per Gift** sets how many coins the player is awarded.

There is also a Reward Ad button on the “out of coins” popup. The popup will appear whenever the player tries to spend coins but does not have enough. The amount of coins the player is awarded for watching the reward ad is set on the CurrencyManager.