Sudoku Documentation

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

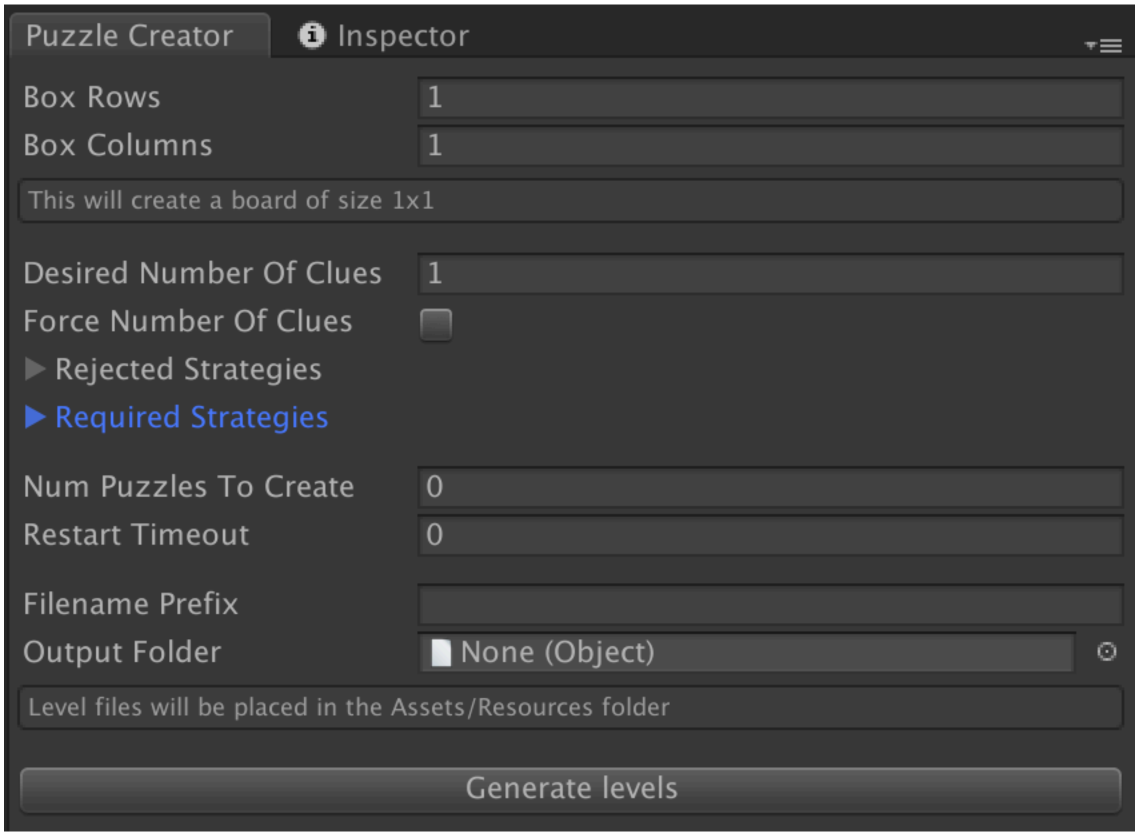
support@bizzybeegames.com

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

Table of Contents:

Creating Sudoku Puzzles

The game works by loading pre-generated puzzle files that are created with the Puzzle Creator editor window. To open the window select the menu item **Tools -> Bizzy Bee Games -> Sudoku Puzzle Creator**:



Puzzle Creator | Inspector

Box Rows: 1
Box Columns: 1
This will create a board of size 1x1

Desired Number Of Clues: 1
Force Number Of Clues: ☐
▶ Rejected Strategies
▶ Required Strategies

Num Puzzles To Create: 0
Restart Timeout: 0

Filename Prefix:
Output Folder: None (Object)
Level files will be placed in the Assets/Resources folder

Generate levels

Box Rows and **Box Cols** sets the size of puzzle that you want to create. These specify the number of rows/columns in one of the nine boxes on a Sudoku puzzle. So for instance setting them to 3 and 3 will create a classic 9x9 sudoku puzzle.

Desired Number Of Clues sets the number of "numbers" that appear on the puzzle when the player first starts the puzzle. The algorithm will try to get as close to this number as possible while retaining a valid sudoku puzzle (Only one solution to the puzzle). If **Force Number Of Clues** is selected then it will continue to try new random puzzles until it can create one with the exact number of desired clues.

NOTE: The generation process may take a long time if Force Number Of Clues is selected. It may also never complete if the desired number of clues is set so low that that there is no possible valid board. For instance, on a 9x9 sudoku puzzle, the lowest possible number of clues is 17 so setting Desired Number Of Clues to 16 will cause the the algorithm to run forever.

Rejected Strategies is all the sudoku solving strategies you do not want to be used in order to solve the sudoku puzzle. For instance, if you wanted to create an easy puzzle, select all except for Hidden singles. This will create puzzles where only naked/hidden single strategies are required to solve the board and will reject any puzzles where more advanced strategies are needed.

Required Strategies is all the sudoku solving strategies you want to be required to solve the puzzle. Only puzzles where one of the selected strategies is used to solve the puzzle will be generated. For instance, if you wanted to create a hard puzzle, select Hidden Pairs/Hidden Triples. This will create only puzzles that require one of those strategies in order for it to be solved.

Num Puzzles To Create sets the number of puzzle files you wish to generate and click the Generate Levels button. This will place your level files in the Resources folder.

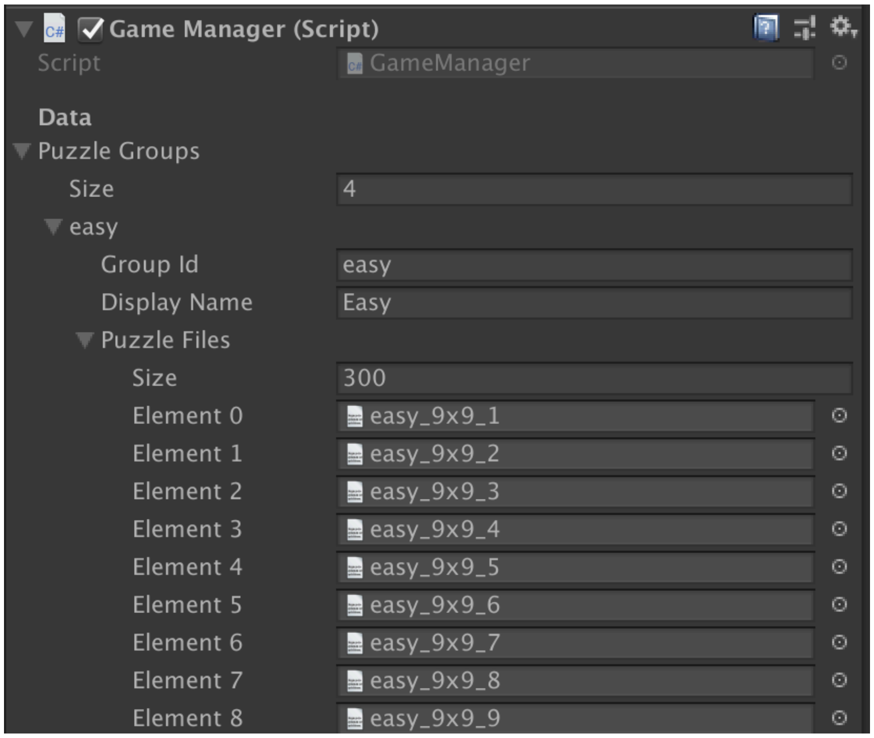
Restart Timeout is the amount of seconds before the algorithm gives up on the current puzzle, discards it, and tries again with a new random placement of numbers. This is required because sometimes on larger puzzles it will create a starting sequence of numbers that is very hard for the algorithm to solve so the best course of action is to discard it and try again with different starting numbers. Setting this to 0 will disable this and the algorithm will continue trying until the puzzle is solved.

Filename Prefix is the prefix to give all level files that are generated, the file name will be of the following format: prefix_#.txt

Output Folder is the folder to place generated puzzle files in, this is set by dragging a folder from the Project window.

Using Puzzle Files In Game

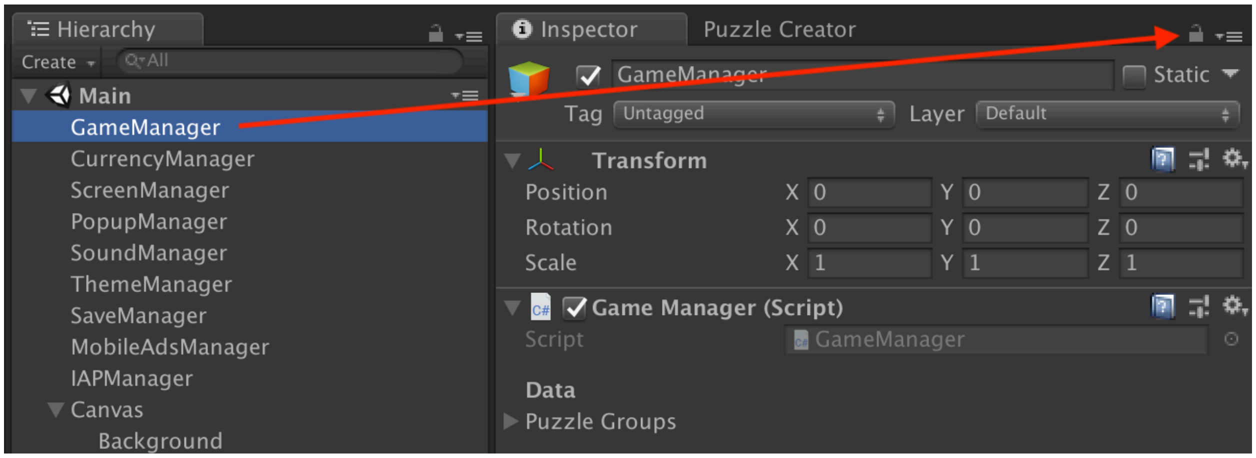
To add puzzle files generated using the Puzzle Creator window, select the **GameManager** in the Main scenes hierarchy and expand **Puzzle Groups**. Expand a puzzle group or create a new group and add the generated puzzle files to the **Puzzle Files** list:



Game Manager (Script)

Script	GameManager
Data	
Puzzle Groups	
Size	4
easy	
Group Id	easy
Display Name	Easy
Puzzle Files	
Size	300
Element 0	easy_9x9_1
Element 1	easy_9x9_2
Element 2	easy_9x9_3
Element 3	easy_9x9_4
Element 4	easy_9x9_5
Element 5	easy_9x9_6
Element 6	easy_9x9_7
Element 7	easy_9x9_8
Element 8	easy_9x9_9

TIP: To add a bunch of puzzle files at once, select the GameManager then click the lock button at the top of the Inspector window:



Main
GameManager
CurrencyManager
ScreenManager
PopupManager
SoundManager
ThemeManager
SaveManager
MobileAdsManager
IAPManager
Canvas
Background

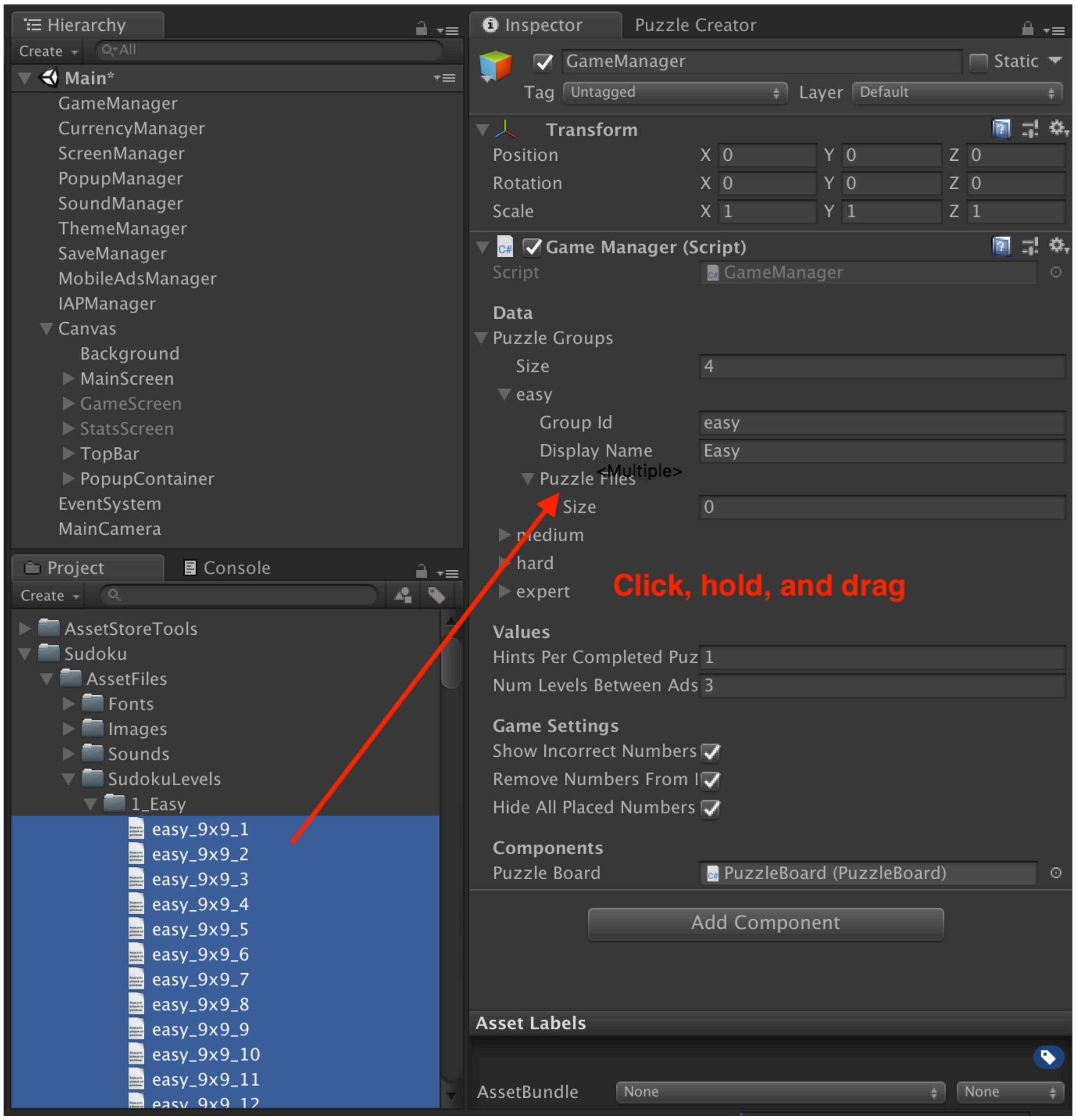
GameManager
Tag Untagged
Layer Default

Transform

Position	X	0	Y	0	Z	0
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

Game Manager (Script)
Script GameManager
Data
Puzzle Groups

With the Inspector locked on the GameManager, you can then select other objects without the Inspector changing away from the GameManager object. Now you can select a bunch of puzzle files from the Project window and drag them to the Puzzle Files list in one of the Puzzle Groups to add them all at once:



```
Main*
  GameManager
  CurrencyManager
  ScreenManager
  PopupManager
  SoundManager
  ThemeManager
  SaveManager
  MobileAdsManager
  IAPManager
  Canvas
    Background
    MainScreen
    GameScreen
    StatsScreen
    TopBar
    PopupContainer
  EventSystem
  MainCamera
```

```
AssetStoreTools
Sudoku
  AssetFiles
    Fonts
    Images
    Sounds
    SudokuLevels
      1_Easy
        easy_9x9_1
        easy_9x9_2
        easy_9x9_3
        easy_9x9_4
        easy_9x9_5
        easy_9x9_6
        easy_9x9_7
        easy_9x9_8
        easy_9x9_9
        easy_9x9_10
        easy_9x9_11
        easy_9x9_12
```

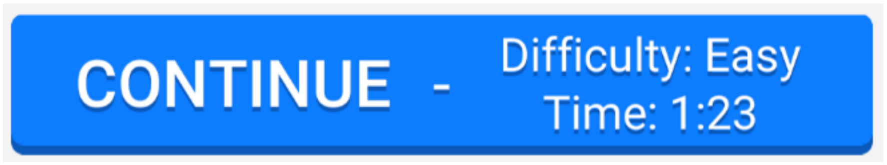
```
GameManager                         Static
Tag Untagged         Layer Default
Transform
  Position   X 0   Y 0   Z 0
  Rotation   X 0   Y 0   Z 0
  Scale      X 1   Y 1   Z 1
Game Manager (Script)
  Script                      GameManager
  Data
  Puzzle Groups
    Size                      4
    easy
      Group Id                easy
      Display Name            Easy
      Puzzle Files  <Multiple>
        Size                  0
    medium
    hard
    expert
  Values
  Hints Per Completed Puz     1
  Num Levels Between Ads      3
  Game Settings
  Show Incorrect Numbers      ✓
  Remove Numbers From I       ✓
  Hide All Placed Numbers     ✓
  Components
  Puzzle Board                PuzzleBoard (PuzzleBoard)
Asset Labels
AssetBundle   None   None
```

Project

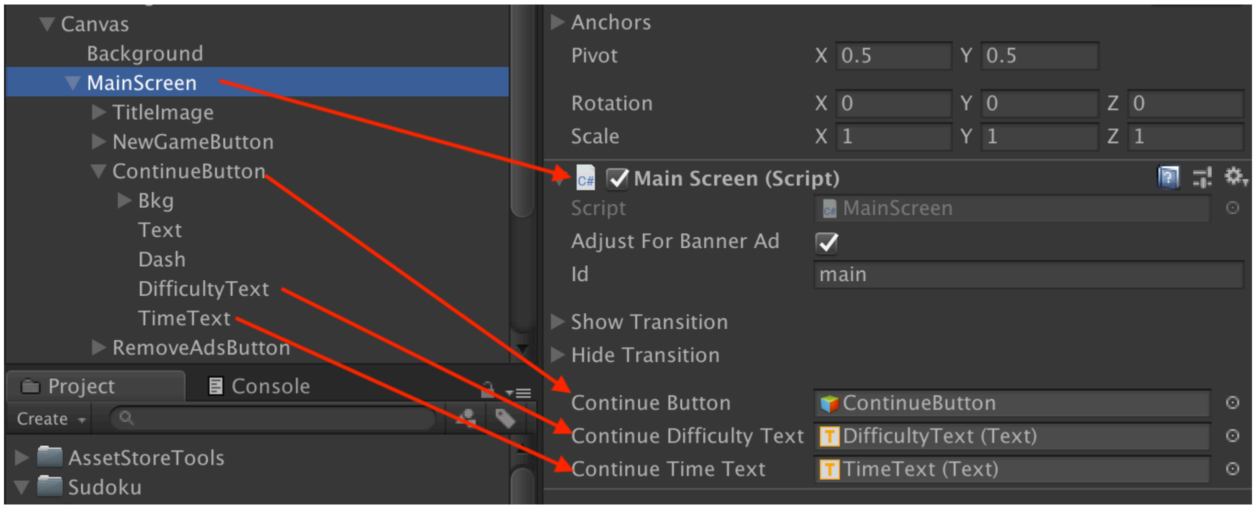
This section will show you where UI elements that are changed / instantiated by scripts are located and how to edit them.

NOTE: All Image and Text components colors are set by the ThemeGraphicBehaviour component when the game runs. If you want to change colors you will need to change them in the ThemeManager. Check the Themes section of this Documentation for more information.

Main Screen - Continue Button



The continue button on the main screen is controlled by the **MainScreen** script attached to the MainScreen GameObject.

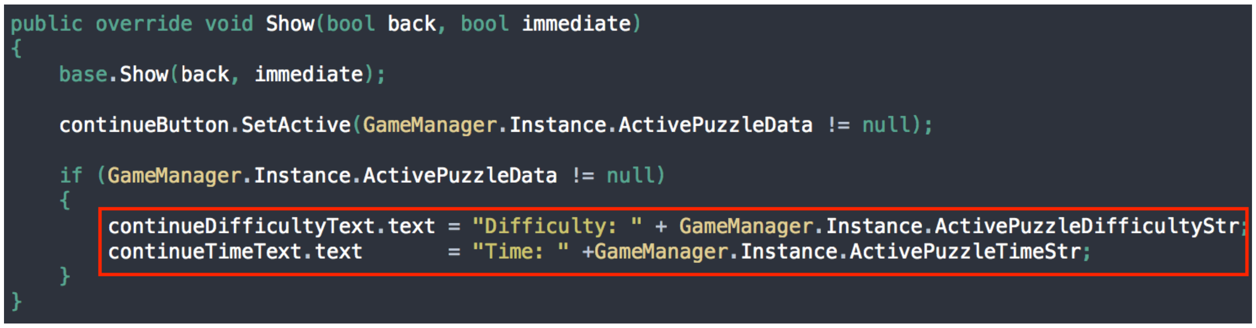


```
▼ Canvas
    Background
  ▼ MainScreen
    ▶ TitleImage
    ▶ NewGameButton
    ▼ ContinueButton
      ▶ Bkg
        Text
        Dash
        DifficultyText
        TimeText
    ▶ RemoveAdsButton
Project    Console
Create
▶ AssetStoreTools
▼ Sudoku
```

```
▶ Anchors
Pivot                     X 0.5    Y 0.5
Rotation                  X 0      Y 0      Z 0
Scale                     X 1      Y 1      Z 1
✓ Main Screen (Script)
Script                    MainScreen
Adjust For Banner Ad      ✓
Id                        main
▶ Show Transition
▶ Hide Transition
Continue Button           ContinueButton
Continue Difficulty Text  DifficultyText (Text)
Continue Time Text        TimeText (Text)
```

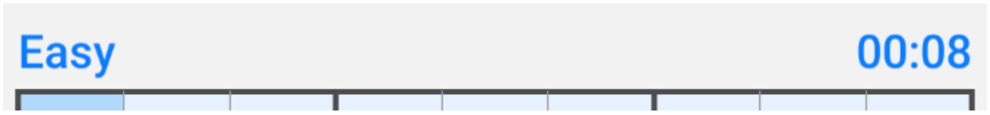
When there is a puzzle that can be continued the MainScreen will set it to active and if there is no puzzle to continue it is set to de-active.

The **DifficultyText** and **TimeText** are also set by the MainScreen script in the **Show** method:

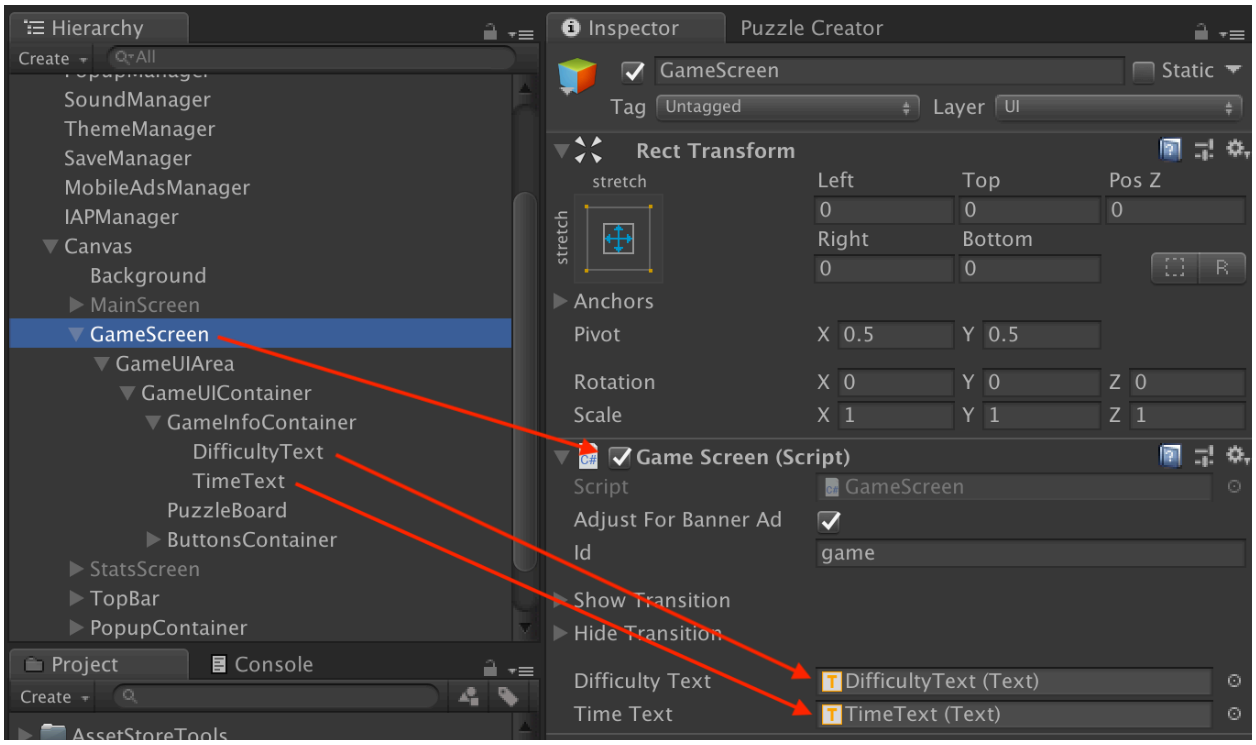


```
public override void Show(bool back, bool immediate)
{
    base.Show(back, immediate);

    continueButton.SetActive(GameManager.Instance.ActivePuzzleData != null);

    if (GameManager.Instance.ActivePuzzleData != null)
    {
        continueDifficultyText.text = "Difficulty: " + GameManager.Instance.ActivePuzzleDifficultyStr;
        continueTimeText.text       = "Time: " +GameManager.Instance.ActivePuzzleTimeStr;
    }
}
```

Game Screen - Difficulty / Time Text



The Difficulty and Time text that appear above the puzzle board on the GameScreen are set by the **GameScreen** script:

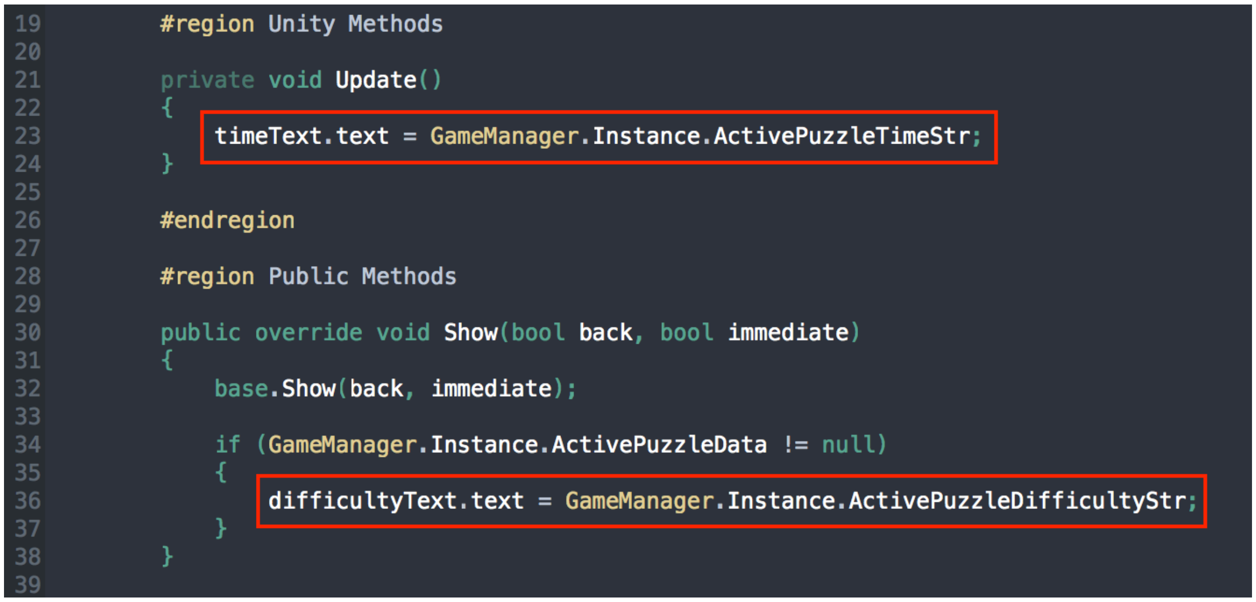


GameScreen Static
Tag Untagged Layer UI

Rect Transform
stretch
Left 0 Top 0 Pos Z 0
Right 0 Bottom 0
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

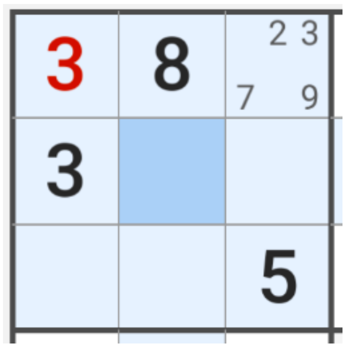
Game Screen (Script)
Script GameScreen
Adjust For Banner Ad ✓
Id game
Difficulty Text DifficultyText (Text)
Time Text TimeText (Text)

The **Time Text** is updated every frame in the Update loop and the **Difficulty Text** is set when the screen shows:

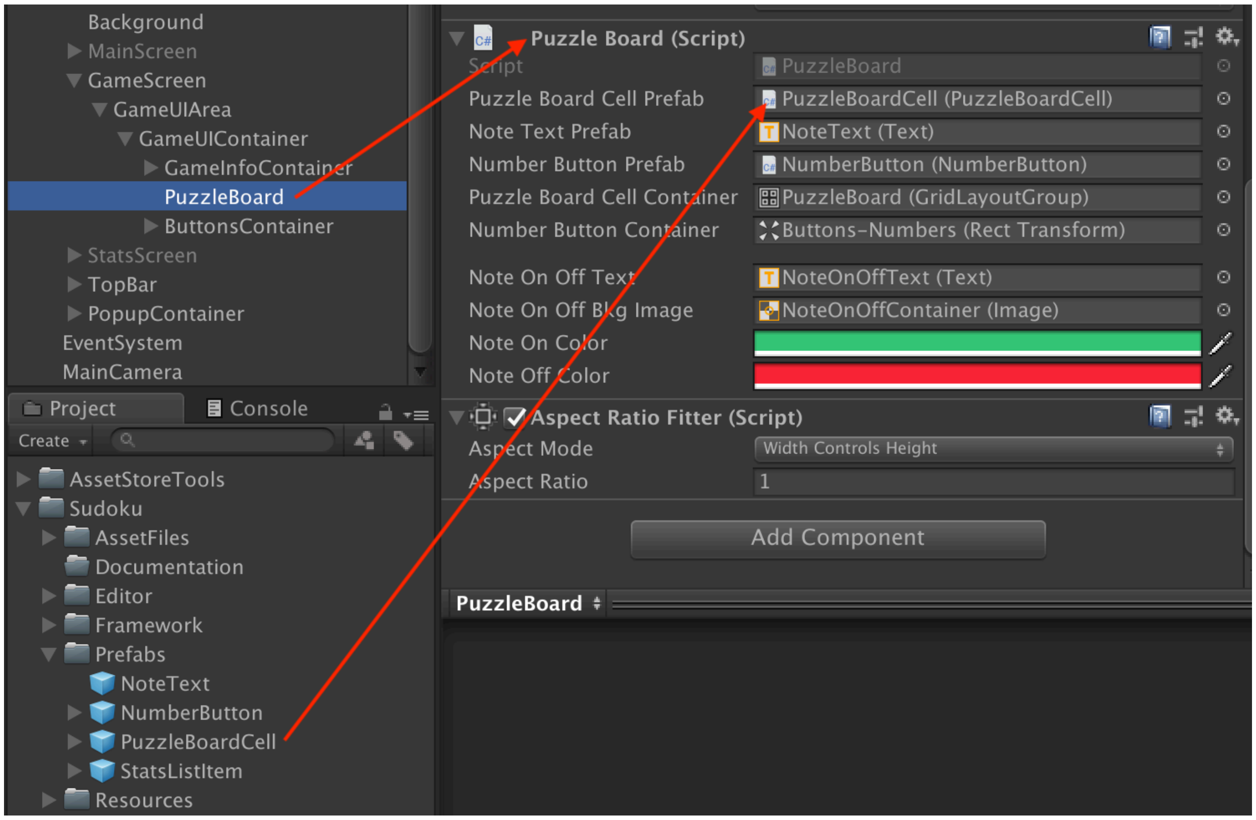


```
        #region Unity Methods

        private void Update()
        {
            timeText.text = GameManager.Instance.ActivePuzzleTimeStr;
        }

        #endregion

        #region Public Methods

        public override void Show(bool back, bool immediate)
        {
            base.Show(back, immediate);

            if (GameManager.Instance.ActivePuzzleData != null)
            {
                difficultyText.text = GameManager.Instance.ActivePuzzleDifficultyStr;
            }
        }

```

Puzzle Board - Cells



The **PuzzleBoard** script is responsible for creating and putting together the cells that make up the puzzle board. When a game starts a copy of the **Puzzle Board Cell Prefab** is instantiated for each cell in the puzzle (So for a 9x9 puzzle there will be 81 cells). The **PuzzleBoardCell** prefab located in the **Prefabs** folder is used in the asset.



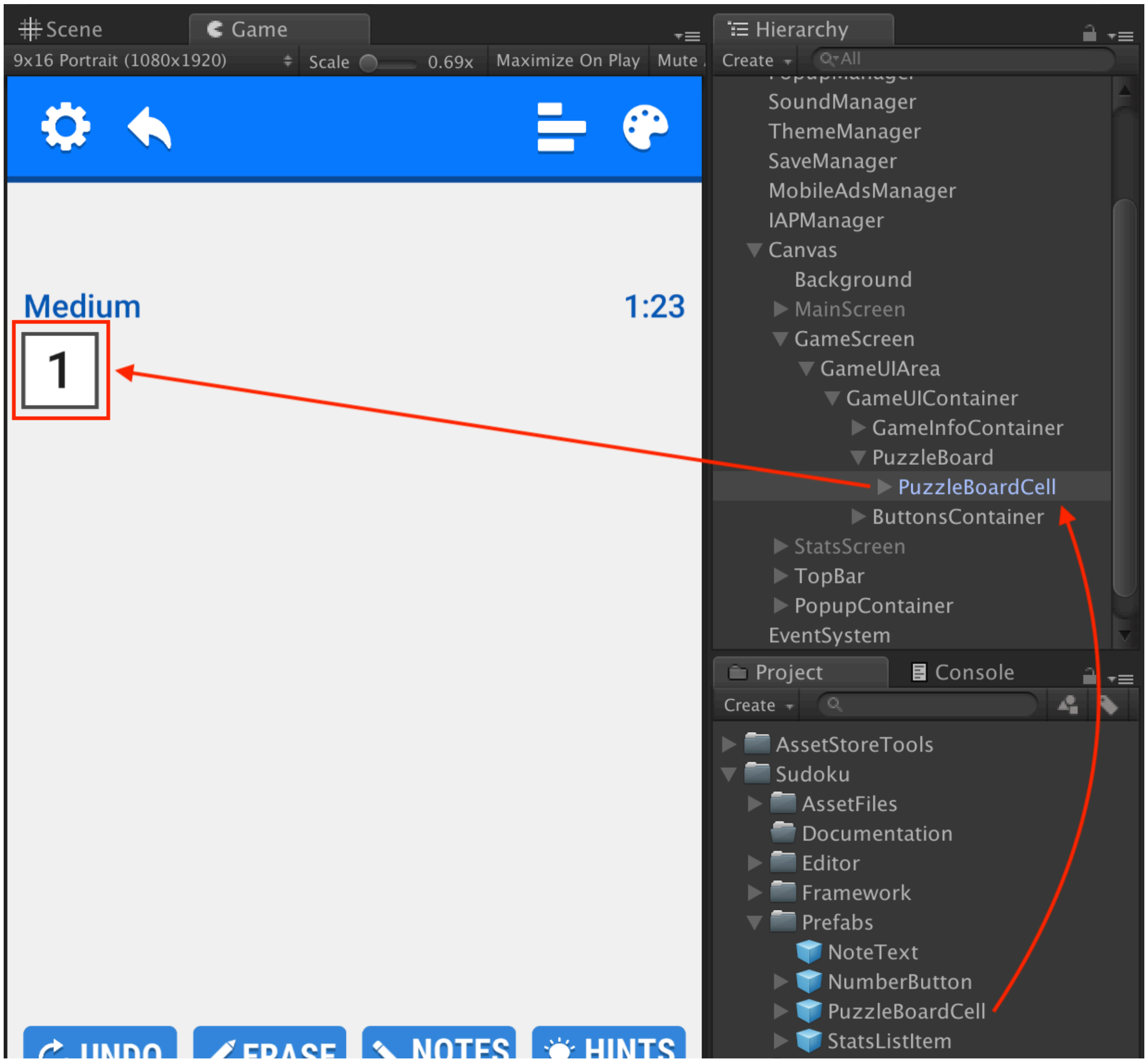
Puzzle Board (Script)

Script	PuzzleBoard
Puzzle Board Cell Prefab	PuzzleBoardCell (PuzzleBoardCell)
Note Text Prefab	NoteText (Text)
Number Button Prefab	NumberButton (NumberButton)
Puzzle Board Cell Container	PuzzleBoard (GridLayoutGroup)
Number Button Container	Buttons-Numbers (Rect Transform)
Note On Off Text	NoteOnOffText (Text)
Note On Off Bkg Image	NoteOnOffContainer (Image)
Note On Color	🟩
Note Off Color	🟥

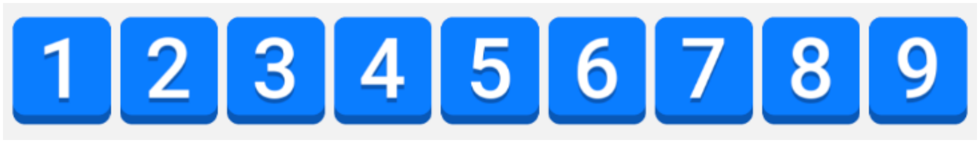
Aspect Ratio Fitter (Script)

Aspect Mode	Width Controls Height
Aspect Ratio	1

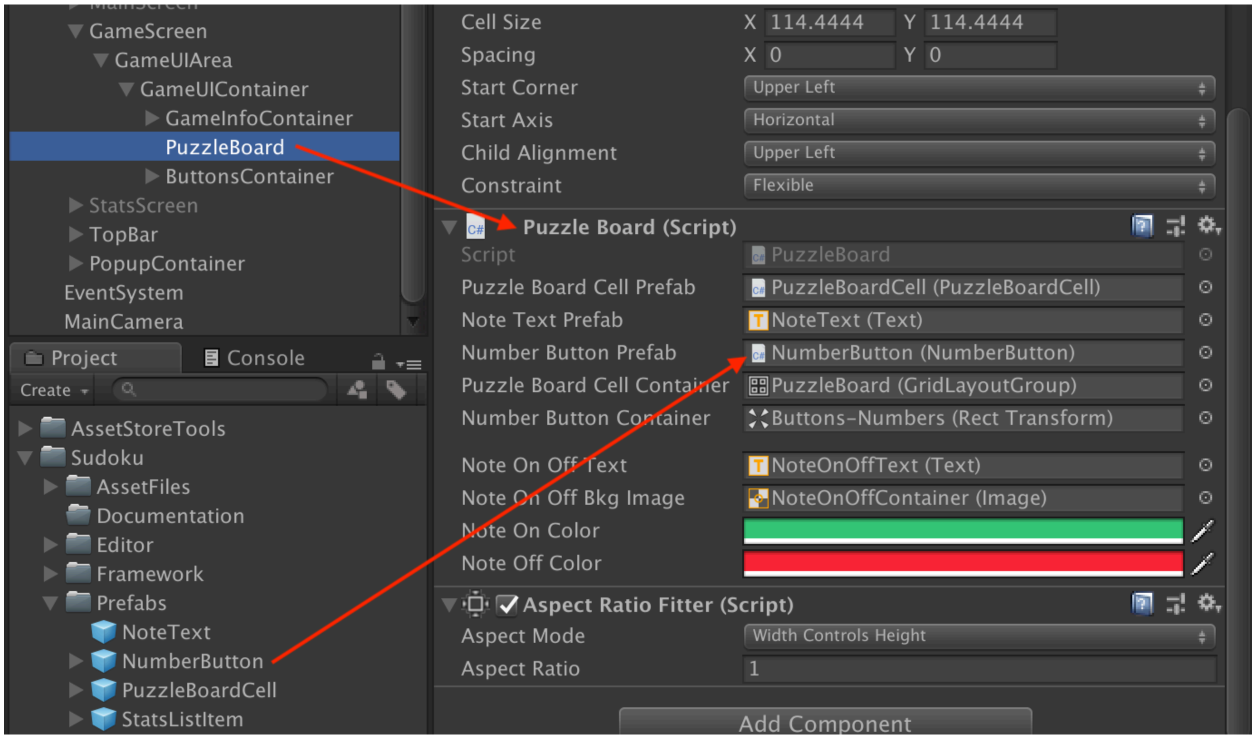
To edit the PuzzleBoardCell, drag it under the PuzzleBoard GameObject:



Game Screen - Numbers

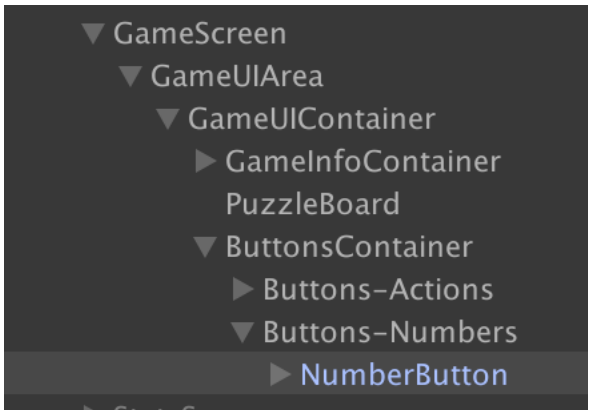


The **PuzzleBoard** script is responsible for creating the number buttons that appear under the puzzle board. When a game starts a copy of the **Number Button Prefab** is instantiated for each number needed in the puzzle (So for a 9x9 puzzle there will be 9 buttons). The **NumberButton** prefab located in the **Prefabs** folder is used in the asset.

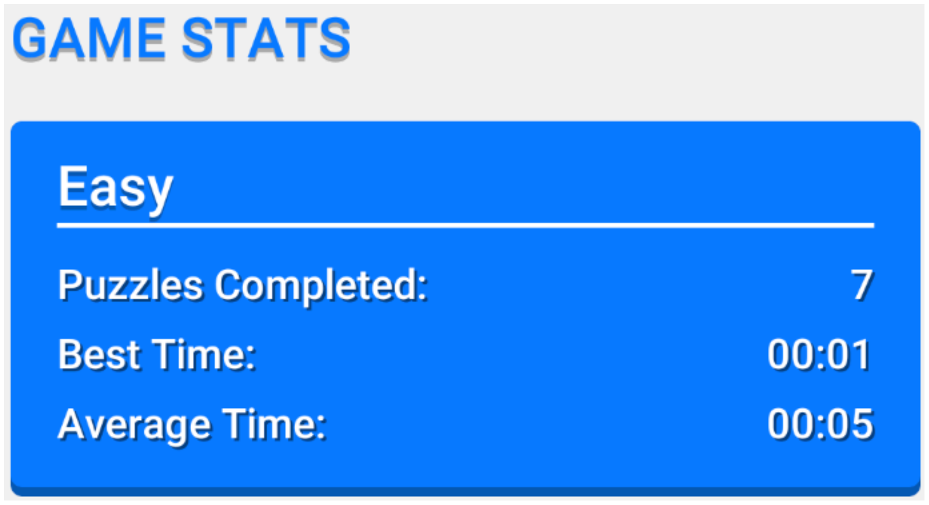


Cell Size	X 114.4444 Y 114.4444
Spacing	X 0 Y 0
Start Corner	Upper Left
Start Axis	Horizontal
Child Alignment	Upper Left
Constraint	Flexible
Puzzle Board (Script)	
Script	PuzzleBoard
Puzzle Board Cell Prefab	PuzzleBoardCell (PuzzleBoardCell)
Note Text Prefab	NoteText (Text)
Number Button Prefab	NumberButton (NumberButton)
Puzzle Board Cell Container	PuzzleBoard (GridLayoutGroup)
Number Button Container	Buttons-Numbers (Rect Transform)
Note On Off Text	NoteOnOffText (Text)
Note On Off Bkg Image	NoteOnOffContainer (Image)
Note On Color	🟩
Note Off Color	🟥
Aspect Ratio Fitter (Script)	
Aspect Mode	Width Controls Height
Aspect Ratio	1

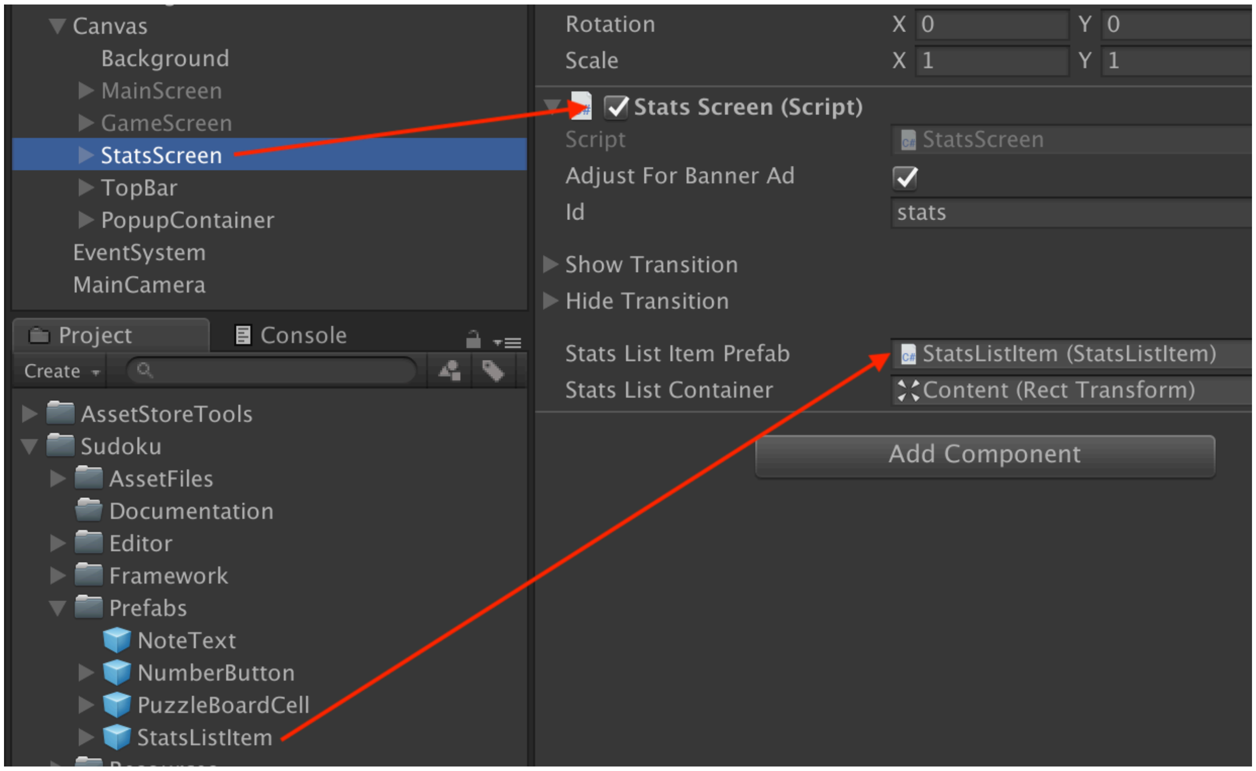
To edit the NumberButton, drag it into the **Buttons-Numbers** GameObject



Stats Screen - Game Stats List Item



The StatsScreen script attached to the StatsScreen GameObject is responsible for instantiating copies of the **Stats List Item Prefab** for each Puzzle Group that exists in the GameManager Puzzle Groups list. The **StatsListItem** prefab located in the **Prefabs** folder is used in the asset:



Rotation X 0 Y 0
Scale X 1 Y 1
Stats Screen (Script)
Script: StatsScreen
Adjust For Banner Ad: ✓
Id: stats
Show Transition
Hide Transition
Stats List Item Prefab: StatsListItem (StatsListItem)
Stats List Container: Content (Rect Transform)

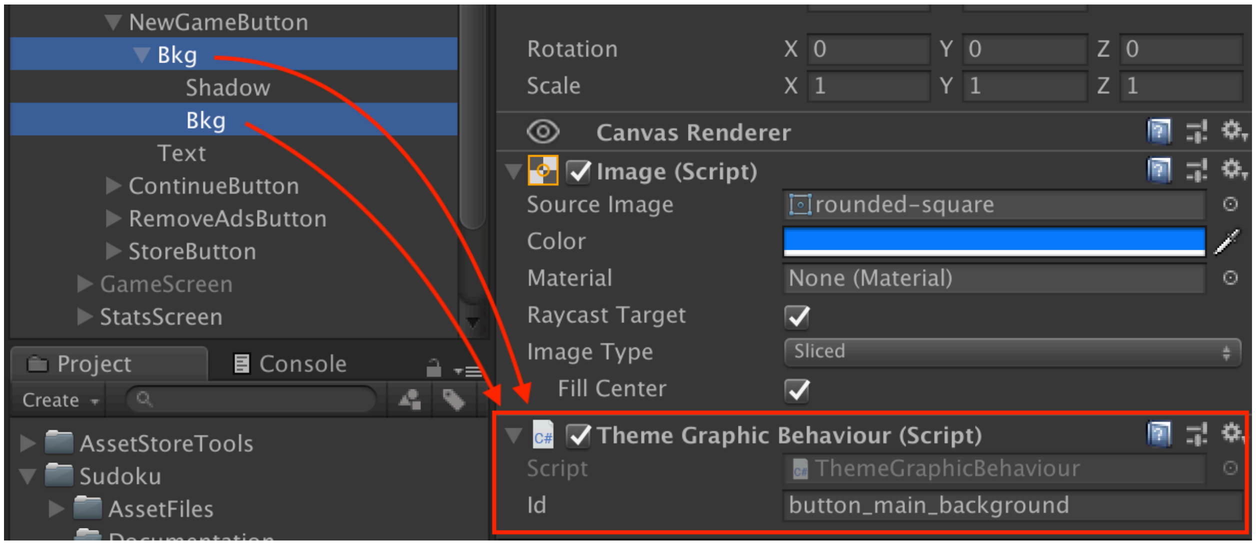
Themes

The **ThemeManager** is responsible for editing / adding new color themes.

Changing Colors In The Asset

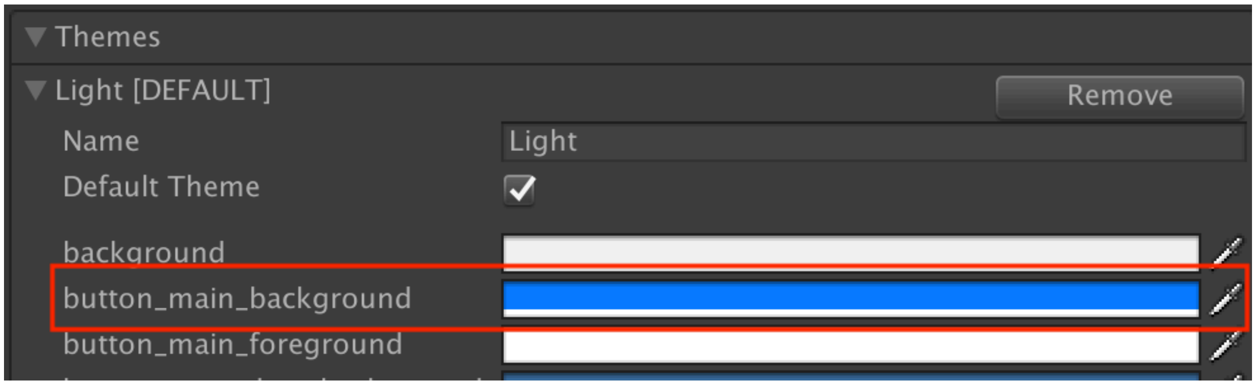
All Image and Text components colors are controlled using the ThemeManager and ThemeGraphicBehaviour component. If you would like to change the color of certain Image/ Text components you will have to change the color on the ThemeManager. To know what color to change check the **Id** that is set on the **ThemeGraphicBehaviour** component.

For example, if we wanted to changed the color of the “NEW GAME” button on the main screen, select the **Bkg** GameObjects that have the Image components on them and we see that the Id on the Theme Graphic Behaviour is set to **button_main_background**:



Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Canvas Renderer
Image (Script)
Source Image rounded-square
Color
Material None (Material)
Raycast Target ✓
Image Type Sliced
Fill Center ✓
Theme Graphic Behaviour (Script)
Script ThemeGraphicBehaviour
Id button_main_background

Then over on the **ThemeManager**, expand the Themes box and locate the button_main_background ids to change the background color of the button:

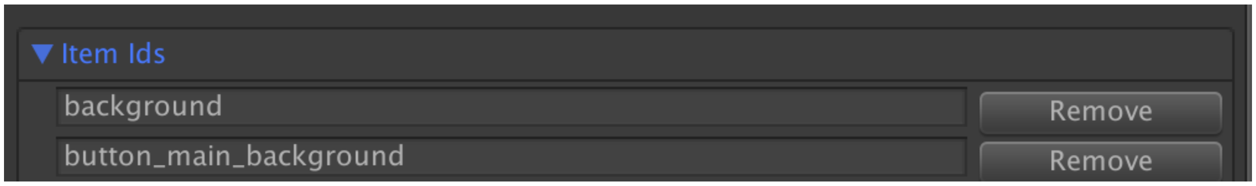


Name Light
Default Theme ✓
background
button_main_background
button_main_foreground

Adding New Theme Color Items

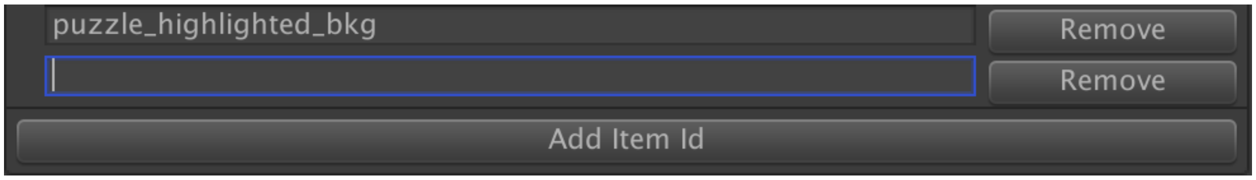
All the button colors are controlled by the button_main_background id. So if you wanted to change the color of just the “NEW GAME” button on the main screen and no other buttons you would have to create a new color id.

To do so, first create a new Id in the **Item Ids** list on the **ThemeManager**. Expand the Item Ids list:



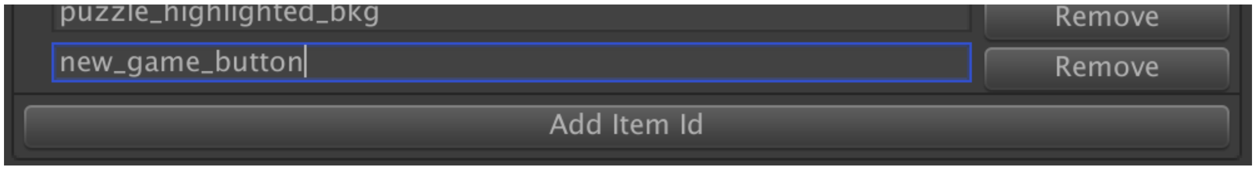
background
button_main_background

Then click the **Add Item Id** button at the bottom of the list:



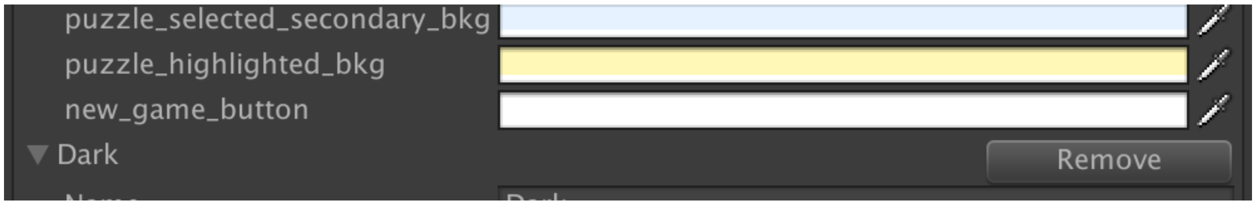
puzzle_highlighted_bkg

We will set the new id to **new_game_button**:



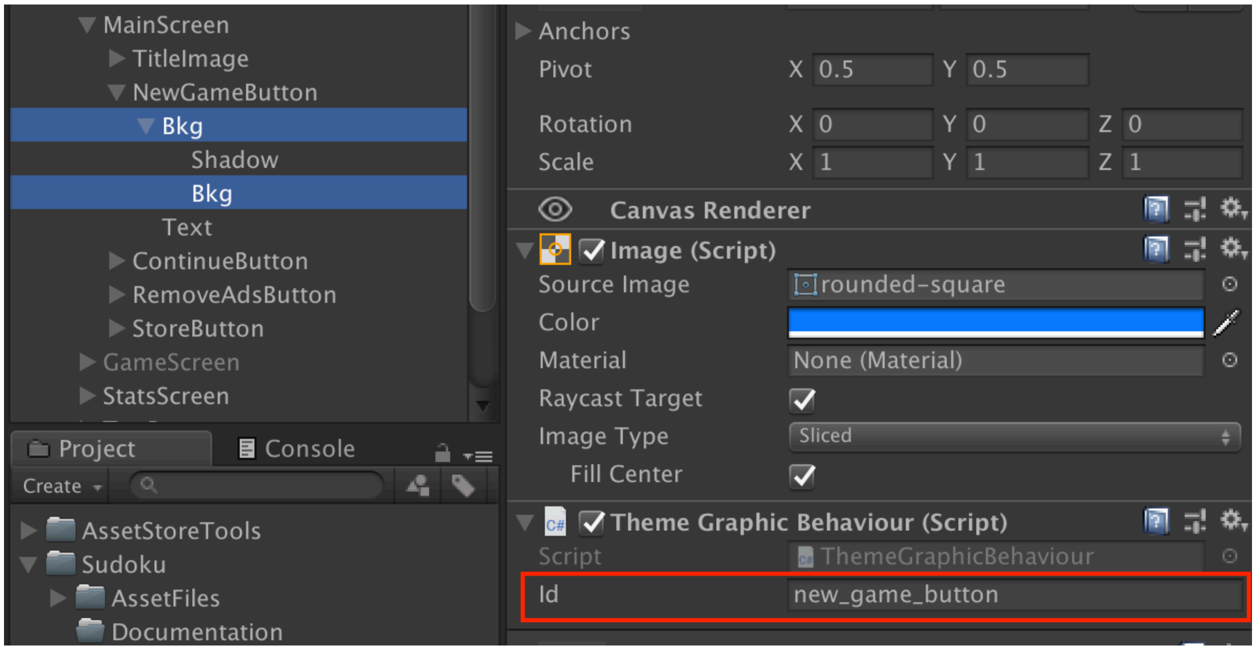
puzzle_highlighted_bkg
new_game_button

A new color item will appear in each theme with the new_game_button id:



puzzle_selected_secondary_bkg
puzzle_highlighted_bkg
new_game_button

Set the color of the new_game_button item for each theme. Then select the **Bkg** GameObjects in the MainScreen under the **NewGameButton** object and change the **Id** on the ThemeBehaviourComponent to the new_game_button id we just created:



```
▼ MainScreen
  ▶ TitleImage
  ▼ NewGameButton
    ▼ Bkg
        Shadow
        Bkg
      Text
  ▶ ContinueButton
  ▶ RemoveAdsButton
  ▶ StoreButton
▶ GameScreen
▶ StatsScreen
```

Anchors			
Pivot	X 0.5	Y 0.5	
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Canvas Renderer

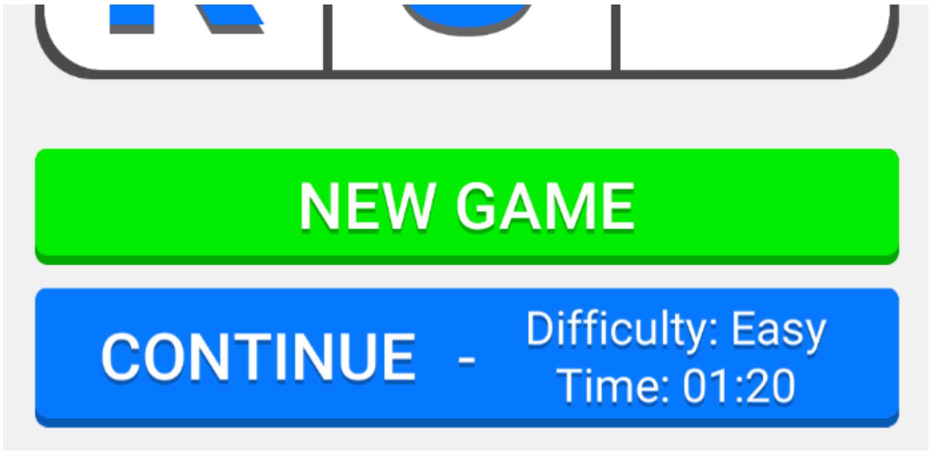
Image (Script)

Source Image	rounded-square
Color	
Material	None (Material)
Raycast Target	✓
Image Type	Sliced
Fill Center	✓

Theme Graphic Behaviour (Script)

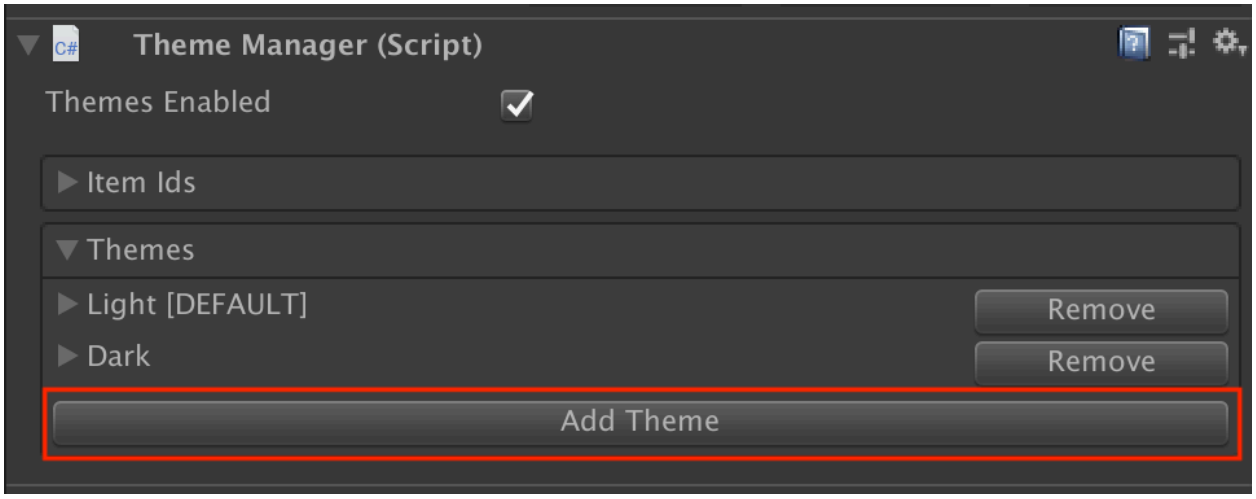
Script	ThemeGraphicBehaviour
Id	new_game_button

Now when you run the game the button will be set to the new color:

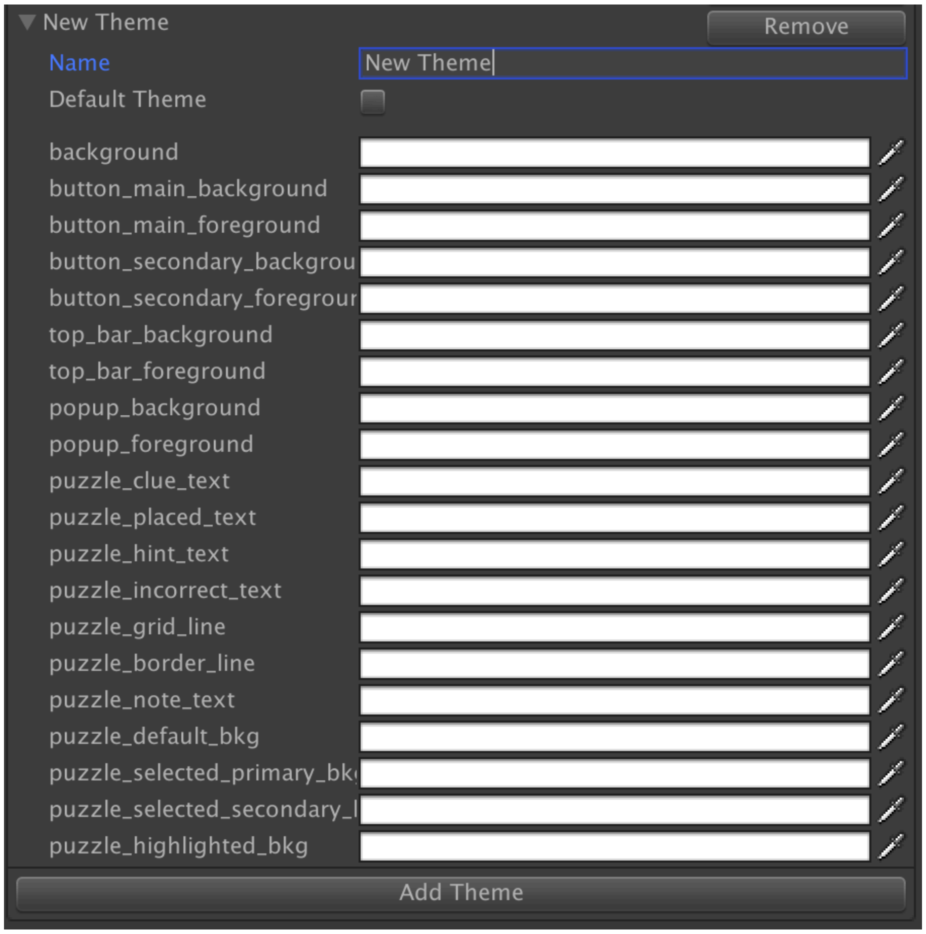


Adding New Themes

To create a new theme simply select the **Add Theme** button on the **ThemeManager:**

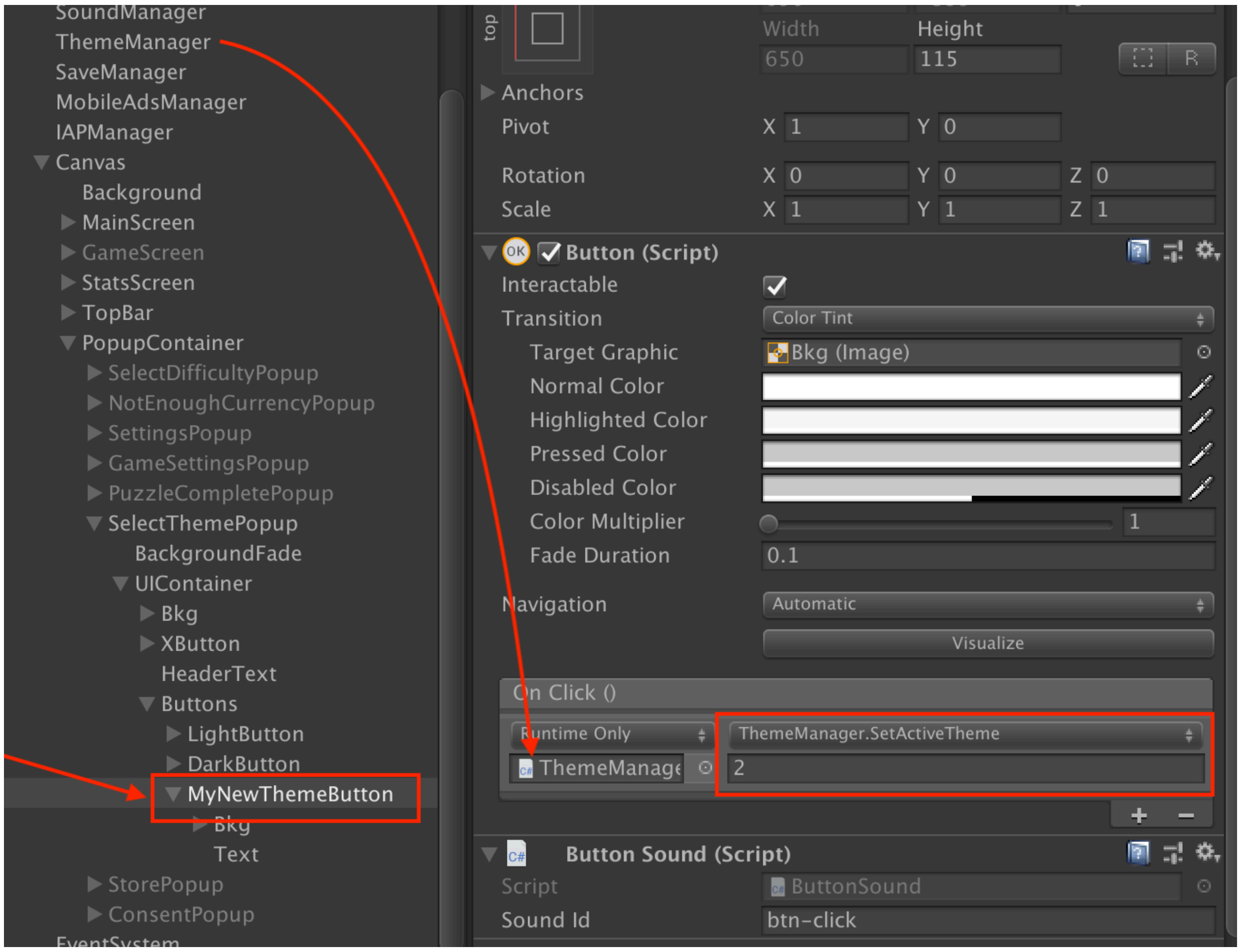


You will then need to set the color of all the existing theme color items or everything will be set to white when using that theme.



Name	New Theme
Default Theme	
background	
button_main_background	
button_main_foreground	
button_secondary_backgrou	
button_secondary_foregrour	
top_bar_background	
top_bar_foreground	
popup_background	
popup_foreground	
puzzle_clue_text	
puzzle_placed_text	
puzzle_hint_text	
puzzle_incorrect_text	
puzzle_grid_line	
puzzle_border_line	
puzzle_note_text	
puzzle_default_bkg	
puzzle_selected_primary_bk	
puzzle_selected_secondary_l	
puzzle_highlighted_bkg	

Once you have set all the colors you will need a way for the player to select the theme. Add a new button to the **SelectThemePopup.** You can then have button call the **SetActiveTheme** method on the **ThemeManager** and pass it the index of the theme you want to set active when the button is clicked:



SoundManager
ThemeManager
SaveManager
MobileAdsManager
IAPManager
Canvas
Background
MainScreen
GameScreen
StatsScreen
TopBar
PopupContainer
SelectDifficultyPopup
NotEnoughCurrencyPopup
SettingsPopup
GameSettingsPopup
PuzzleCompletePopup
SelectThemePopup
BackgroundFade
UIContainer
Bkg
XButton
HeaderText
Buttons
LightButton
DarkButton
MyNewThemeButton
Bkg
Text
StorePopup
ConsentPopup
EventSystem

Width 650
Height 115
Anchors
Pivot X 1 Y 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Button (Script)
Interactable ✓
Transition Color Tint
Target Graphic Bkg (Image)
Normal Color
Highlighted Color
Pressed Color
Disabled Color
Color Multiplier 1
Fade Duration 0.1
Navigation Automatic
Visualize

On Click ()
Runtime Only ThemeManager.SetActiveTheme
ThemeManage 2

Button Sound (Script)
Script ButtonSound
Sound Id btn-click

Sounds

Sounds in the game are controlled using the SoundManager. On the SoundManager's inspector you will find a number of Sounds Infos already created and used in the game.

Sound Info fields

Id - The Id used to play the sound in the game.
Audio Clip - The sound file from your project.
Type - The type of sound (Sound Effect or Music), this is used to turn on/off all sounds of a particular type.
Play And Loop On Start - If selected the Audio Clip will play when the game starts and will loop forever unless it is stopped.
Clip Volume - Sets the volume of the sound when it is played.

Playing Sounds

Sounds can be played by calling the **Play** method on the SoundManager like so:

```
SoundManager.Instance.Play(string id);
```

You can easily play a sound when a Button is clicked by adding the **ButtonSound** component to a GameObject with a **Button** component. The ButtonSound will play the sound with the specified Id every time the button is clicked.