Word Blocks Documentation

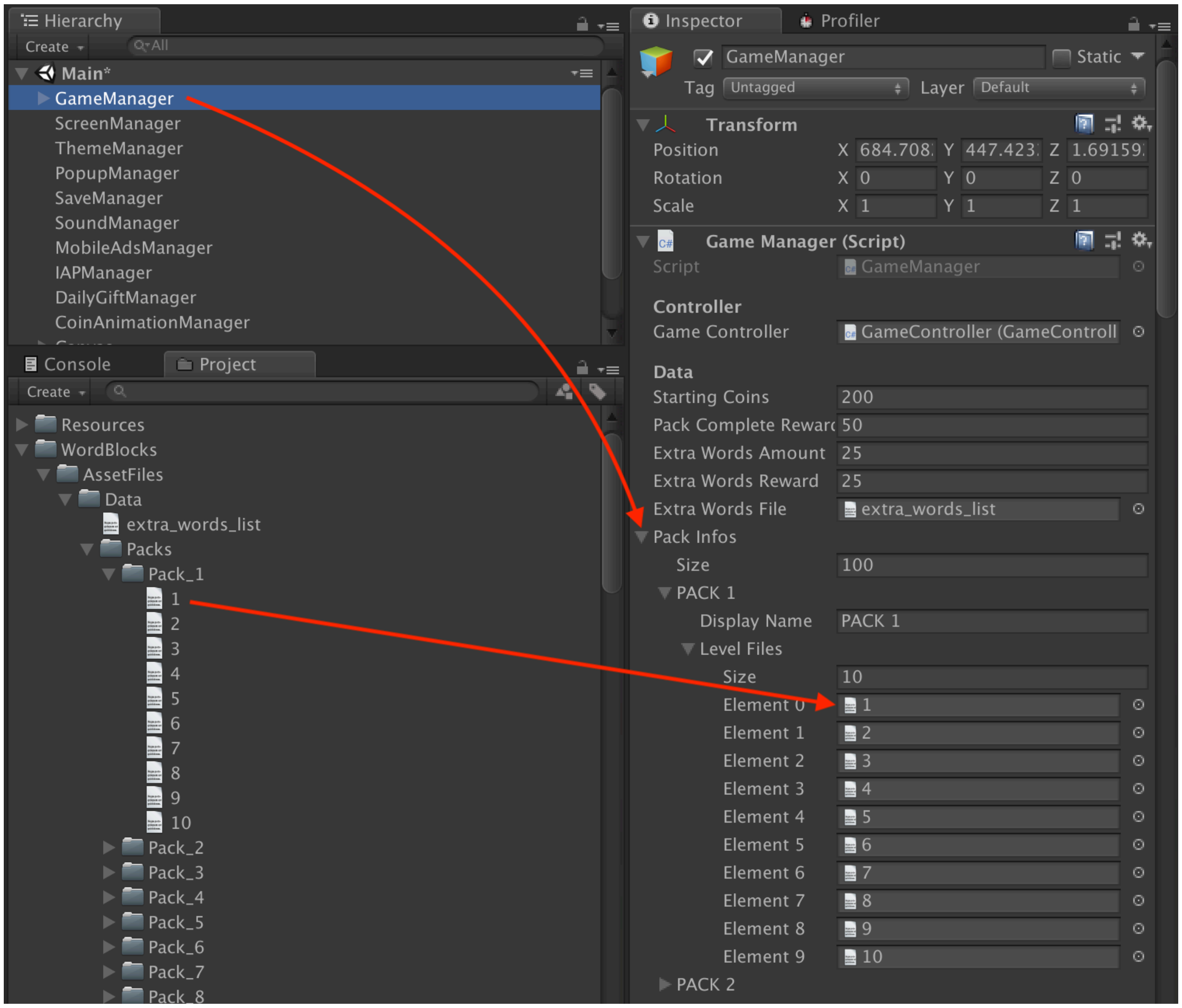
Creating Levels

Each level file is a text file which contains the **hint** on the first line followed by all **words** in the level on their own lines.

For example level one's hint is SCHOOL SUBJECTS and the words are GYM, ART, and DRAMA so it's level file contains:

SCHOOL SUBJECTS
GYM
ART
DRAMA

After you create the level text file it can be added to the game using the **GameManager**'s inspector. Click on the GameManager then expand **Pack Infos** and create or expand the pack you would like to add the level to and add the level file to the **Level Files** list.



Hierarchy:
- Main*
  - GameManager
  - ScreenManager
  - ThemeManager
  - PopupManager
  - SaveManager
  - SoundManager
  - MobileAdsManager
  - IAPManager
  - DailyGiftManager
  - CoinAnimationManager

Project:
- Resources
- WordBlocks
  - AssetFiles
    - Data
      - extra_words_list
      - Packs
        - Pack_1
          - 1
          - 2
          - 3
          - 4
          - 5
          - 6
          - 7
          - 8
          - 9
          - 10
        - Pack_2
        - Pack_3
        - Pack_4
        - Pack_5
        - Pack_6
        - Pack_7
        - Pack_8

Inspector: GameManager, Tag Untagged, Layer Default

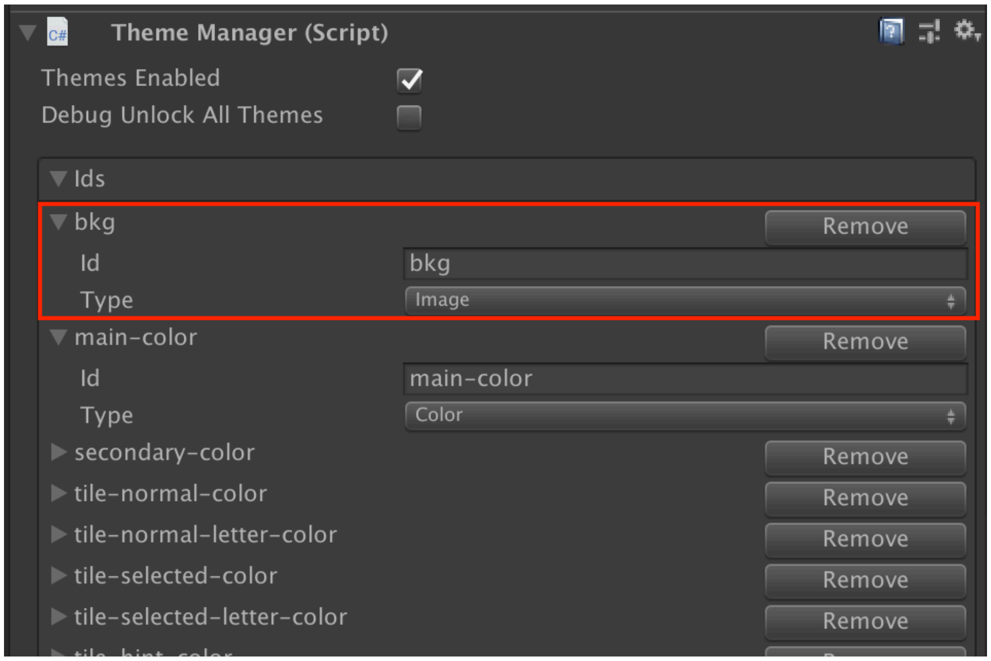
Transform	X	Y	Z
Position	684.708	447.423	1.69159
Rotation	0	0	0
Scale	1	1	1

Game Manager (Script)

Script	GameManager
Controller	
Game Controller	GameController (GameControll
Data	
Starting Coins	200
Pack Complete Rewar	50
Extra Words Amount	25
Extra Words Reward	25
Extra Words File	extra_words_list
Pack Infos	
Size	100
PACK 1	
Display Name	PACK 1
Level Files	
Size	10
Element 0	1
Element 1	2
Element 2	3
Element 3	4
Element 4	5
Element 5	6
Element 6	7
Element 7	8
Element 8	9
Element 9	10
PACK 2	

Themes

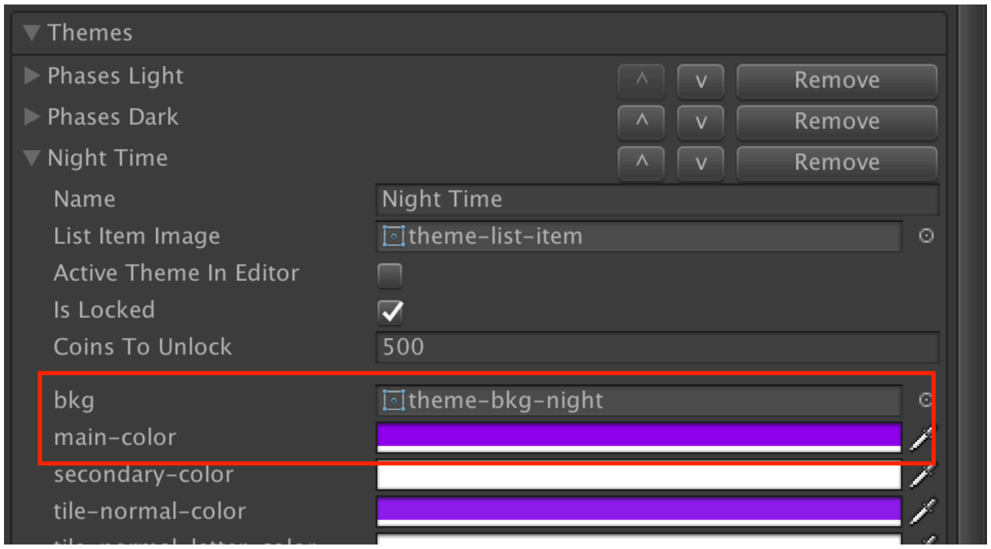
The themes are created and controlled using the **ThemeManager**. The ThemeManager works by first adding **Ids** and assigning them to a **type** (Image, Color, or Prefab).



```
Theme Manager (Script)
Themes Enabled              ✓
Debug Unlock All Themes     ☐

Ids
  bkg                                   Remove
    Id          bkg
    Type        Image
  main-color                            Remove
    Id          main-color
    Type        Color
  secondary-color                       Remove
  tile-normal-color                     Remove
  tile-normal-letter-color              Remove
  tile-selected-color                   Remove
  tile-selected-letter-color            Remove
```

A new field will appear in each **Theme** where you can set the Sprite, Color, or prefab reference for the Id you just created:



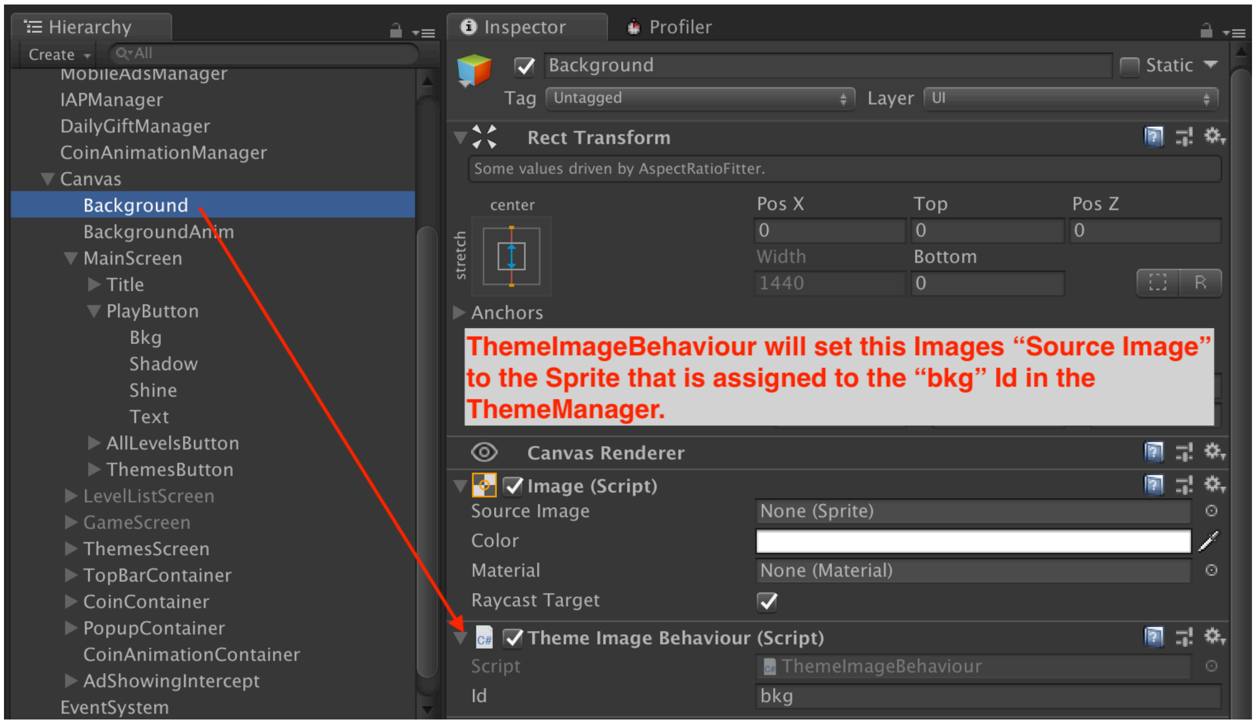
```
Themes
  Phases Light                  ^  v  Remove
  Phases Dark                   ^  v  Remove
  Night Time                    ^  v  Remove
    Name                    Night Time
    List Item Image         theme-list-item
    Active Theme In Editor  ☐
    Is Locked               ✓
    Coins To Unlock         500

    bkg                     theme-bkg-night
    main-color
    secondary-color
    tile-normal-color
```

To use the themes in the application you can use one of the three components: **ThemeImageBehaviour**, **ThemeColorBehaviour**, and **ThemePrefabBehaviour**. Or you can programmatically fetch the ThemeItem using the ThemeManager.

ThemeImageBehaviour

The ThemeImageBehaviour is used for theme Ids that are set to the Image type. When the game runs it sets the **Source Image** of the **Image component** that is attached to the same GameObject to the Sprite set in the ThemeManager for the specified Id. Example:



Background Static
Tag Untagged Layer UI

Some values driven by AspectRatioFitter.

center	Pos X	Top	Pos Z
stretch	0	0	0
	Width	Bottom	
	1440	0	

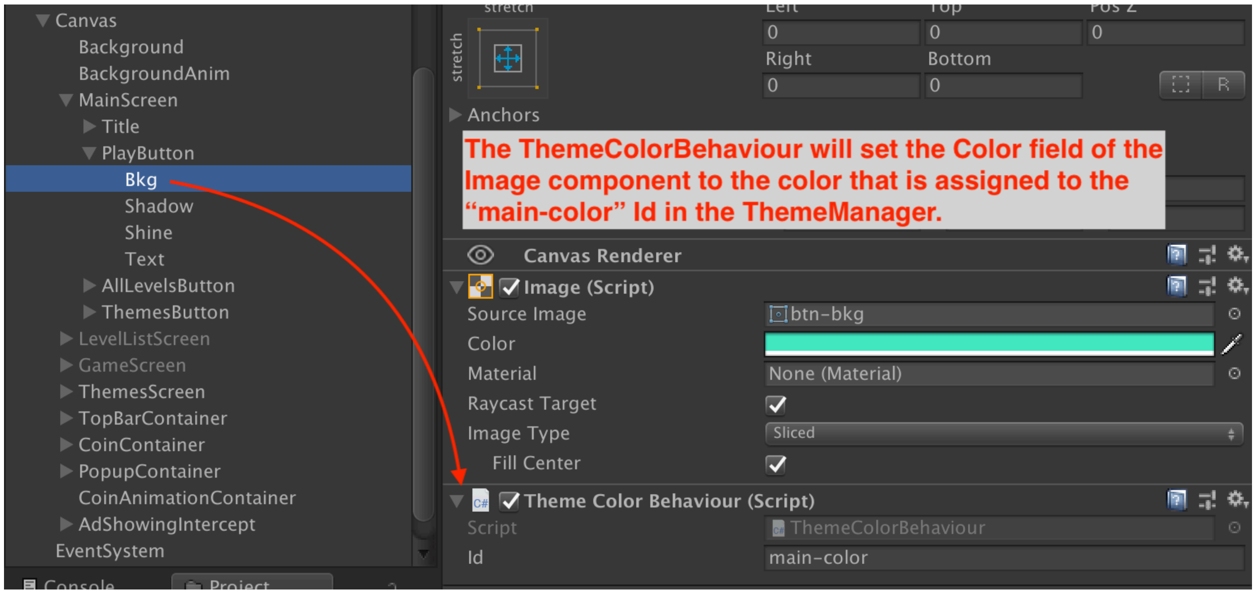
ThemeImageBehaviour will set this Images "Source Image" to the Sprite that is assigned to the "bkg" Id in the ThemeManager.

Image (Script)	
Source Image	None (Sprite)
Color	⬜
Material	None (Material)
Raycast Target	✓

Theme Image Behaviour (Script)	
Script	ThemeImageBehaviour
Id	bkg

ThemeColorBehaviour

The ThemeColorBehaviour is used for theme Ids that are set to the Color type. When the game runs it sets the Color of the Graphic component (Image, RawImage, Text) to the the color set in the ThemeManager for the specified Id. Example:



stretch	Left	Top	Pos Z
	0	0	0
	Right	Bottom	
	0	0	

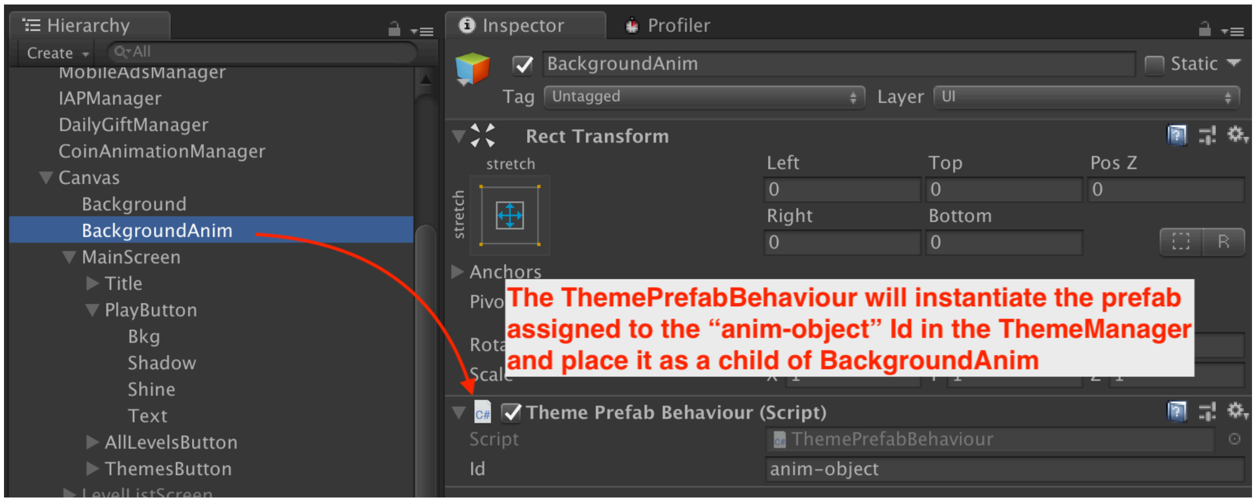
The ThemeColorBehaviour will set the Color field of the Image component to the color that is assigned to the "main-color" Id in the ThemeManager.

Image (Script)	
Source Image	btn-bkg
Color	🟩
Material	None (Material)
Raycast Target	✓
Image Type	Sliced
Fill Center	✓

Theme Color Behaviour (Script)	
Script	ThemeColorBehaviour
Id	main-color

ThemePrefabBehaviour

The ThemePrefabBehaviour is used for theme Ids that are set to the Prefab type. When the game runs it will instantiate the prefab set in the ThemeManager for the specified Id and set it as a child. Example:



BackgroundAnim

Tag Untagged · Layer UI

Left	Top	Pos Z
0	0	0
Right	Bottom	
0	0	

The ThemePrefabBehaviour will instantiate the prefab assigned to the "anim-object" Id in the ThemeManager and place it as a child of BackgroundAnim

Theme Prefab Behaviour (Script)

Script: ThemePrefabBehaviour

Id: anim-object

Programmatically

To fetch a ThemeItem for an Id you can call the following method:

ThemeManager.Instance.GetThemeItem(string id, out ThemeItem themeItem, out ItemId itemId);

This will get the ThemeItem object for the active theme with the given id. An example of this can be found in **GridLetterTile.cs** in the **SetColor** method where it gets the various colors for different states of the tile (normal, selected, etc).

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

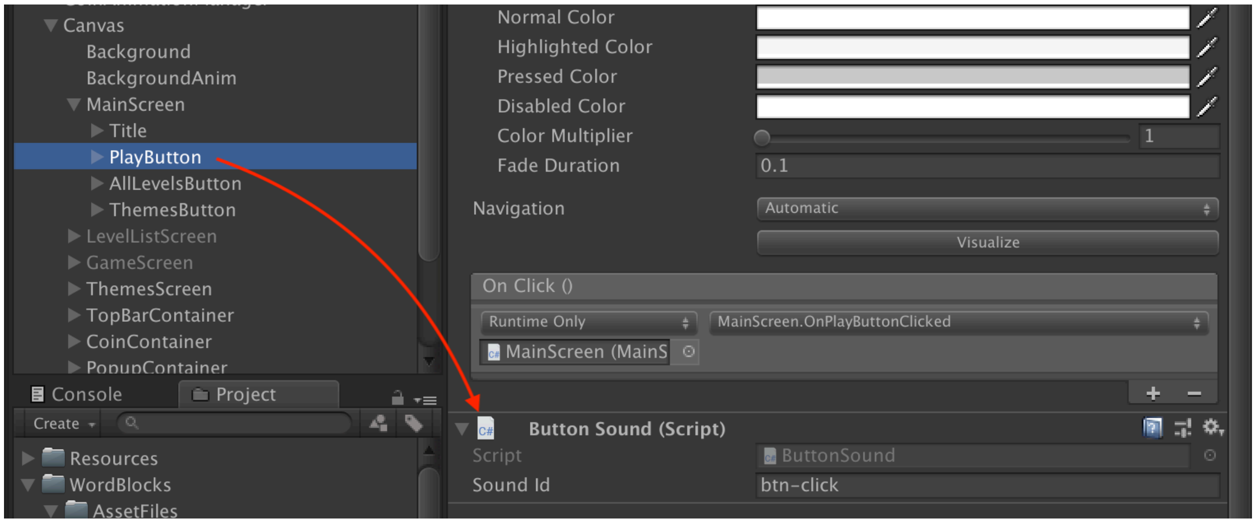
Id - The Id used to play the sound in the game.
Audio Clip - The sound file from your project.
Type - The type of sound (Sound Effect or Music), this is used to turn on/off all sounds of a particular type.
Play And Loop On Start - If selected the Audio Clip will play when the game starts and will loop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

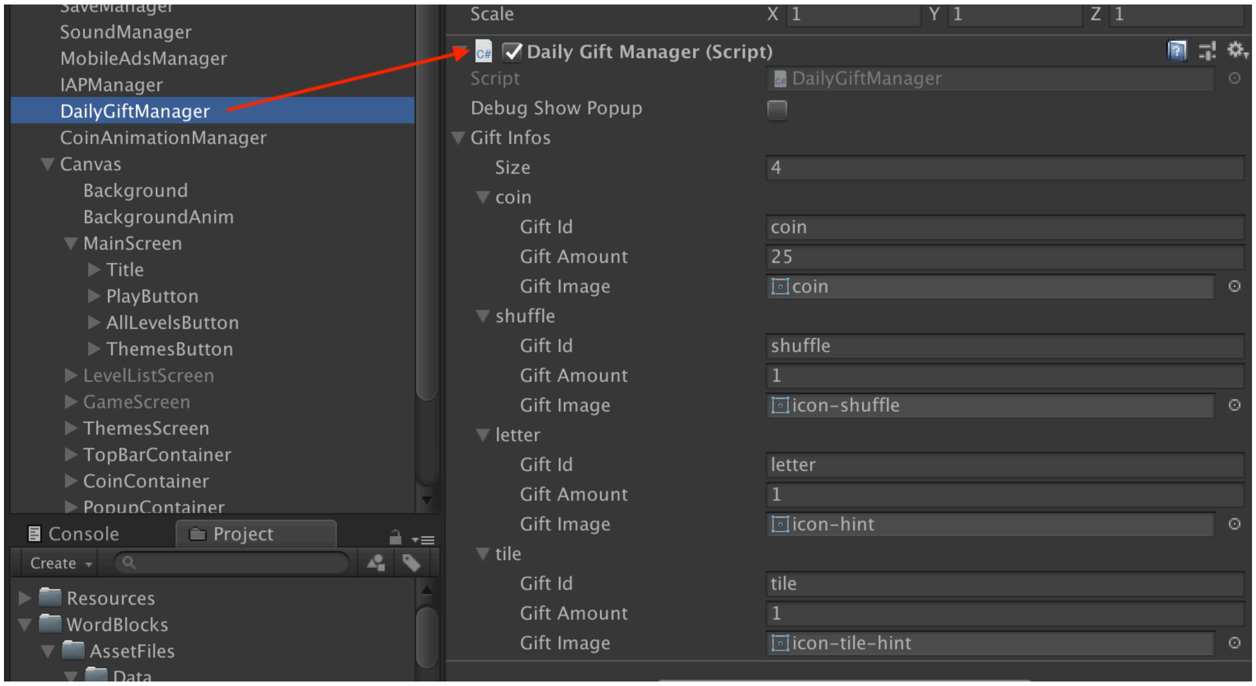
You can easily play a sound when a Button is clicked by adding the **ButtonSound** component to a GameObject with a **Button** component. The ButtonSound will play the sound with the specified Id every time the button is clicked.



Normal Color
Highlighted Color
Pressed Color
Disabled Color
Color Multiplier 1
Fade Duration 0.1
Navigation Automatic
On Click ()
Runtime Only MainScreen.OnPlayButtonClicked
MainScreen (MainS
Button Sound (Script)
Script ButtonSound
Sound Id btn-click

Daily Gifts

The DailyGiftManager is used to award a gift to the user when they open the application the first time each day. There are 4 gifts in the Word Blocks asset that can be given to the player: coins, shuffle, letter hints, and tile hints. The amount the player gets can be assigned on the DailyGiftManagers inspector.



Scale X 1 Y 1 Z 1

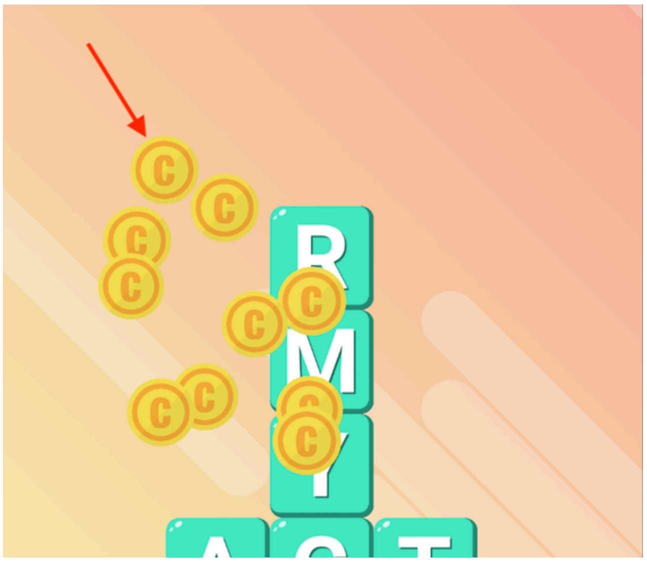
Daily Gift Manager (Script)

Script	DailyGiftManager
Debug Show Popup	☐
Gift Infos	
Size	4
coin	
Gift Id	coin
Gift Amount	25
Gift Image	coin
shuffle	
Gift Id	shuffle
Gift Amount	1
Gift Image	icon-shuffle
letter	
Gift Id	letter
Gift Amount	1
Gift Image	icon-hint
tile	
Gift Id	tile
Gift Amount	1
Gift Image	icon-tile-hint

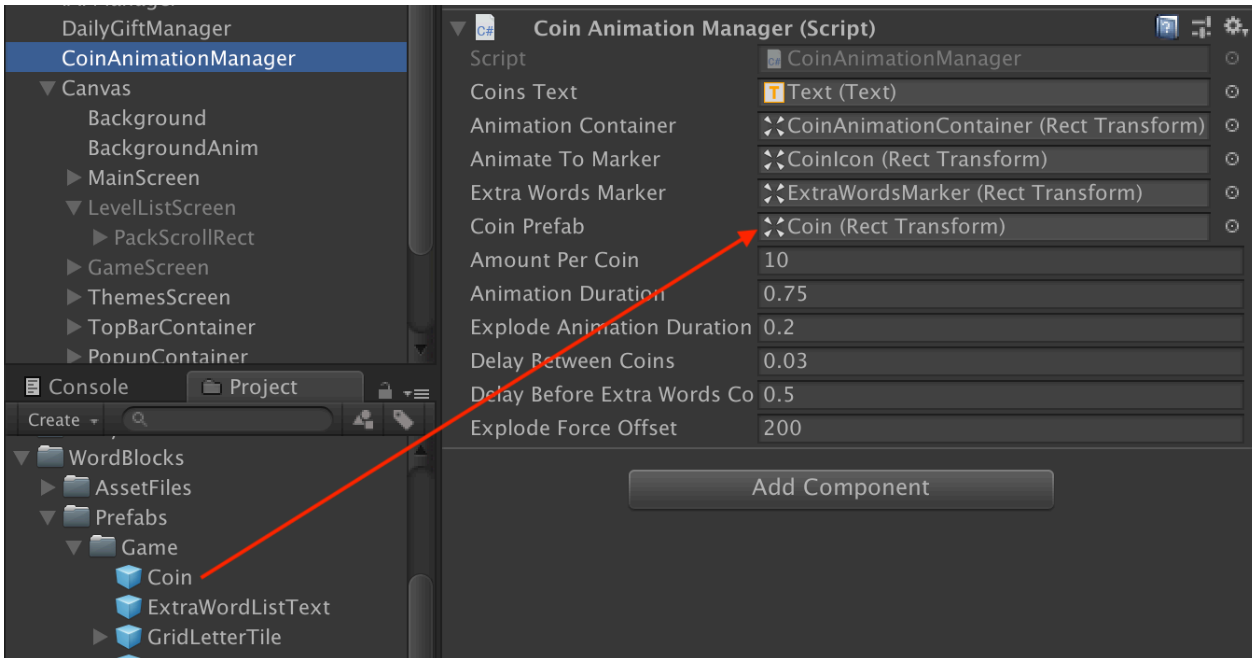
Project

In this section I will show you where you can find / edit some game elements that are instantiated when the game runs and are not located in the Scenes hierarchy.

Coins



The flying coins are created / controlled by the CoinAnimationManager. The **CoinPrefab** is the prefab that is instantiated when a coin needs to animated across the screen. The **Coin** prefab in the Prefabs folder is the coin that is being used in the asset.



DailyGiftManager
CoinAnimationManager
Canvas
Background
BackgroundAnim
MainScreen
LevelListScreen
PackScrollRect
GameScreen
ThemesScreen
TopBarContainer
PopupContainer

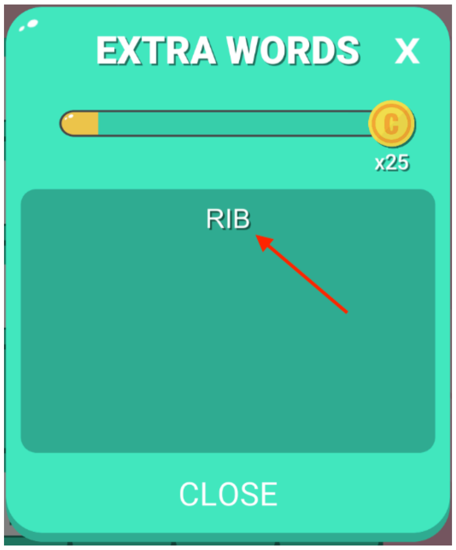
Console | Project

WordBlocks
AssetFiles
Prefabs
Game
Coin
ExtraWordListText
GridLetterTile

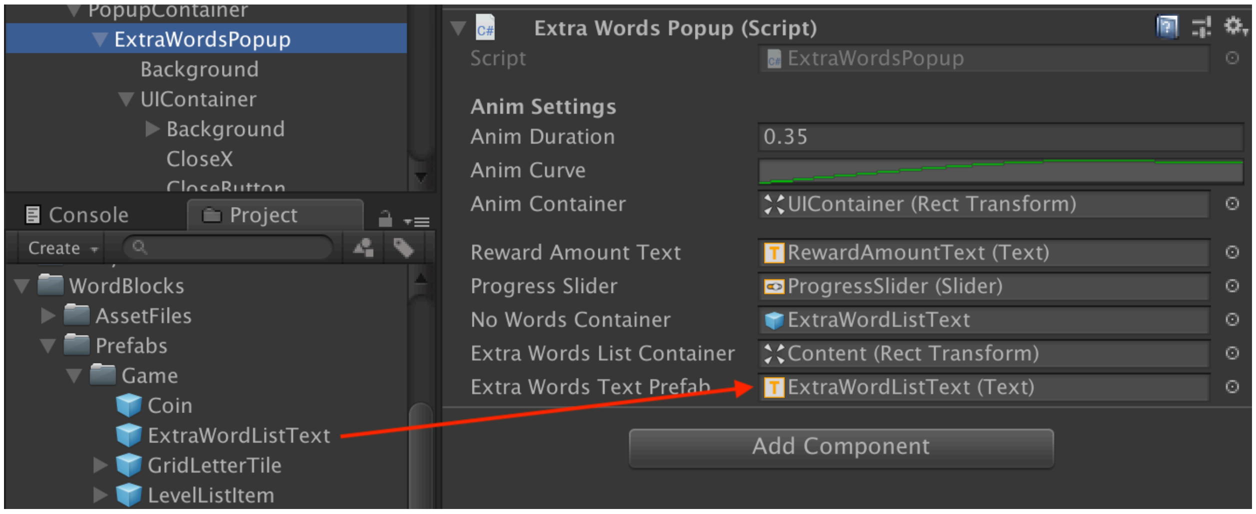
Coin Animation Manager (Script)

Script	CoinAnimationManager
Coins Text	Text (Text)
Animation Container	CoinAnimationContainer (Rect Transform)
Animate To Marker	CoinIcon (Rect Transform)
Extra Words Marker	ExtraWordsMarker (Rect Transform)
Coin Prefab	Coin (Rect Transform)
Amount Per Coin	10
Animation Duration	0.75
Explode Animation Duration	0.2
Delay Between Coins	0.03
Delay Before Extra Words Co	0.5
Explode Force Offset	200

Extra Words Text



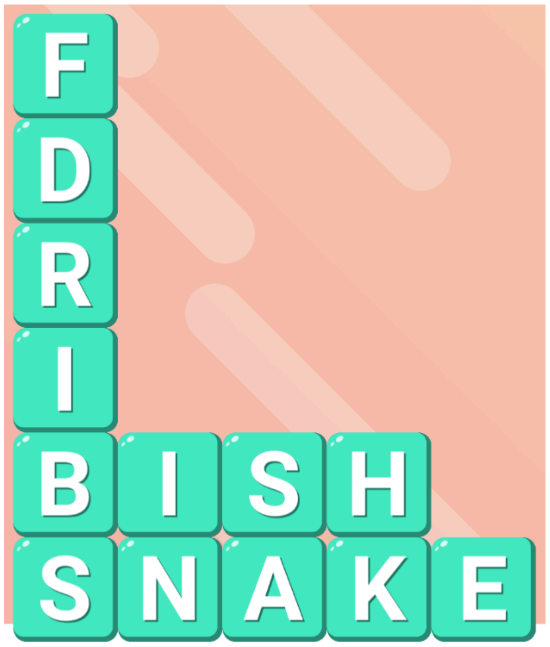
The extra words text is instantiated by the ExtraWordsPopup. The **Extra Words Text Prefab** is the prefab that is instantiated when the popup populates the list of found extra words. The ExtraWordListText prefab in the Prefabs folder is the one used in the asset:



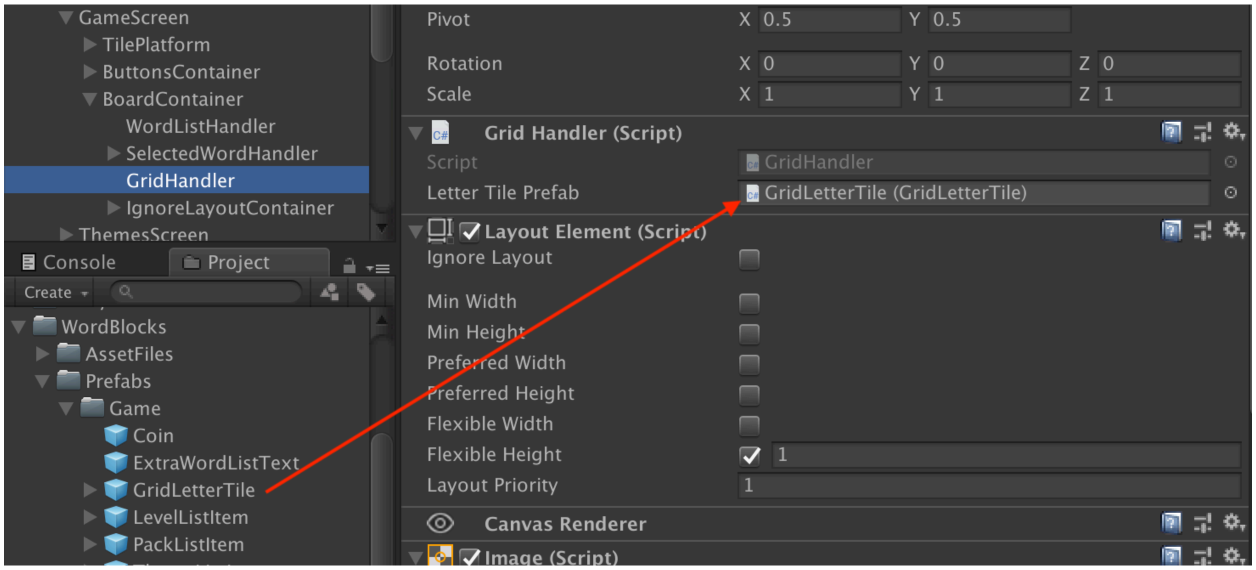
Extra Words Popup (Script)

Script	ExtraWordsPopup
Anim Settings	
Anim Duration	0.35
Anim Curve	
Anim Container	UIContainer (Rect Transform)
Reward Amount Text	RewardAmountText (Text)
Progress Slider	ProgressSlider (Slider)
No Words Container	ExtraWordListText
Extra Words List Container	Content (Rect Transform)
Extra Words Text Prefab	ExtraWordListText (Text)

Grid Letter Tiles



The **GridHandler** is responsible for instantiating and positioning the tiles that the player selects. The **Letter Tile Prefab** is the prefab that is instantiated for each cell in the grid. The GridLetterTile in the Prefabs folder is the prefab used in the asset:



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

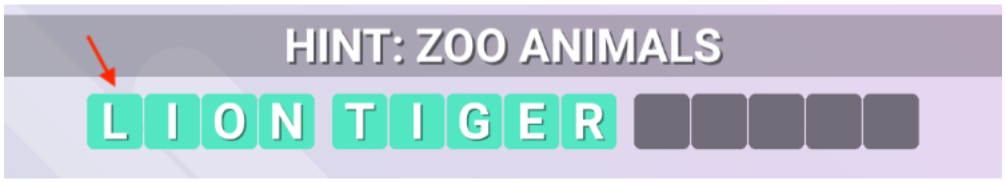
Grid Handler (Script)
Script: GridHandler
Letter Tile Prefab: GridLetterTile (GridLetterTile)

Layout Element (Script)
Ignore Layout
Min Width
Min Height
Preferred Width
Preferred Height
Flexible Width
Flexible Height ✓ 1
Layout Priority 1

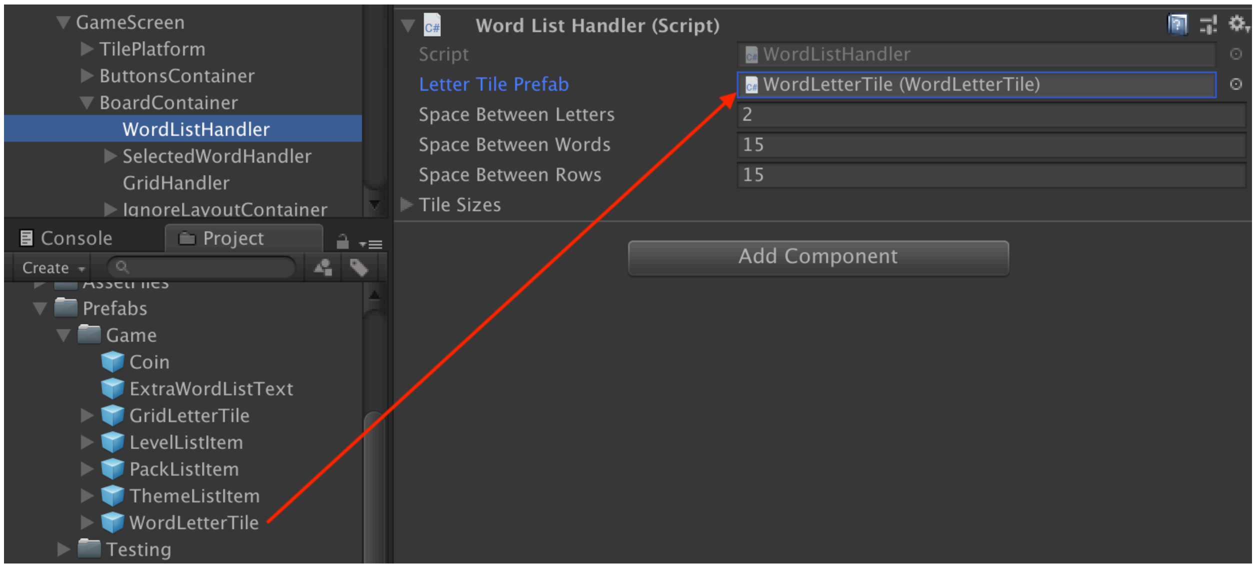
Canvas Renderer

Image (Script)

Word Letter Tile

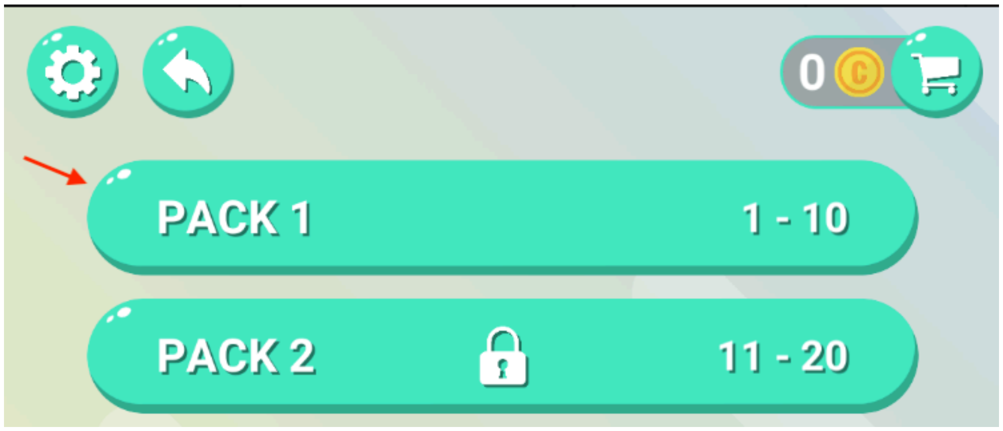


The **WordListHandler** is responsible for instantiating and positioning the tiles for the words at the top of the game screen. The **Letter Tile Prefab** is the prefab that is instantiated for each letter in each word. The WordLetterTile in the Prefabs folder is the prefab used in the asset:

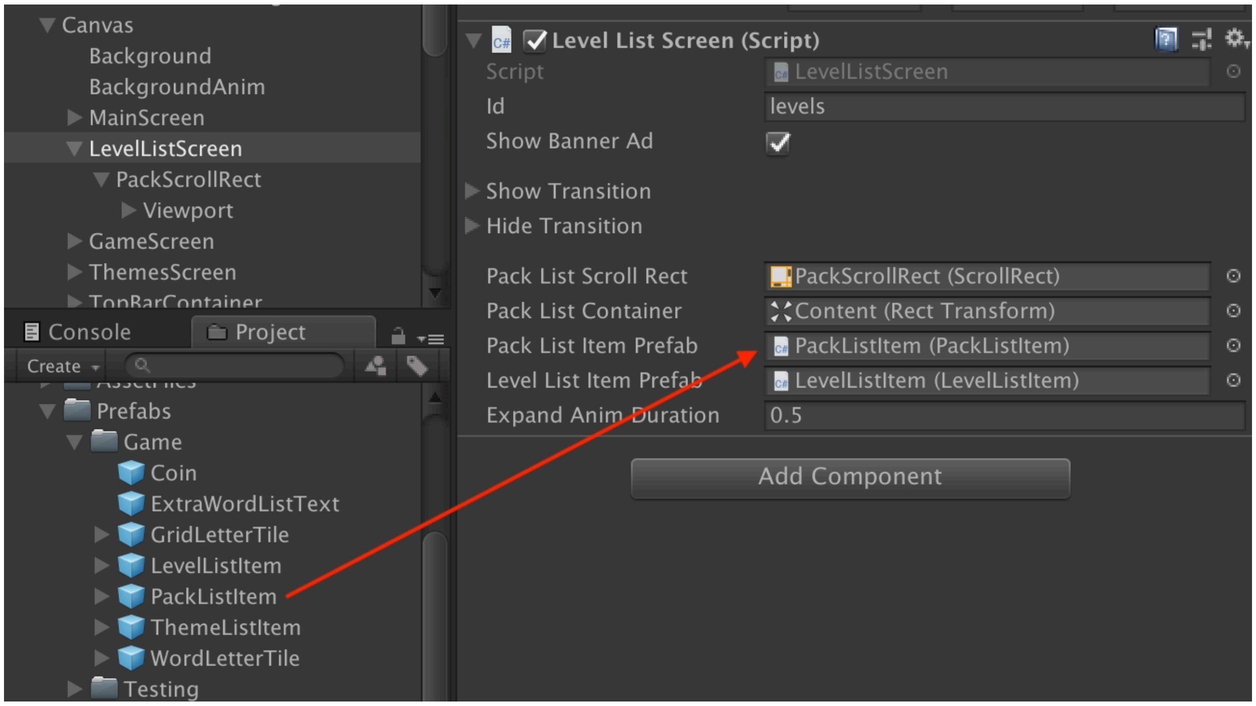


Script	WordListHandler
Letter Tile Prefab	WordLetterTile (WordLetterTile)
Space Between Letters	2
Space Between Words	15
Space Between Rows	15
Tile Sizes	

Pack List Item



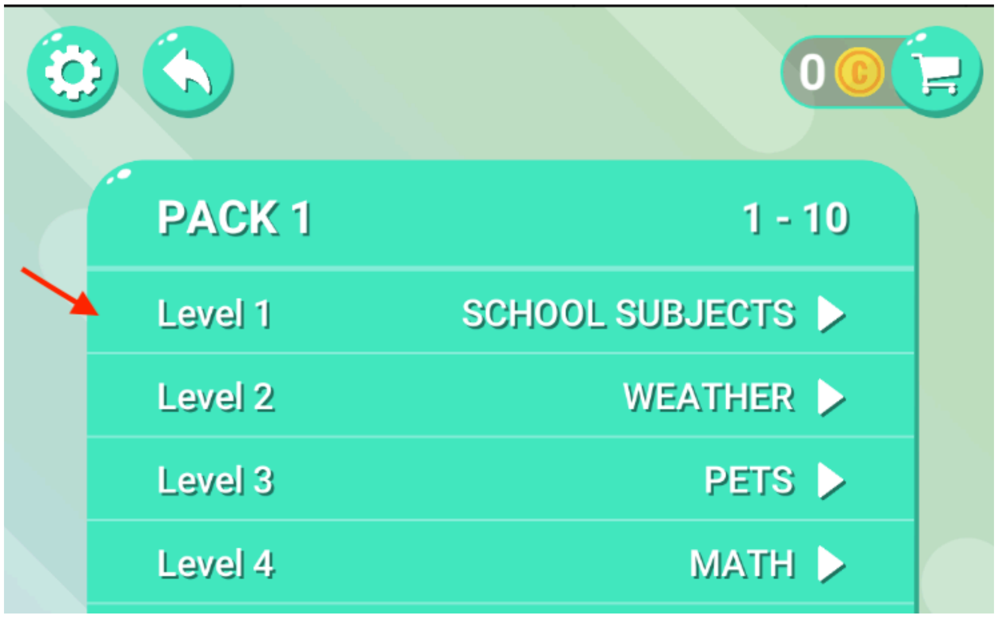
The **LevelListScreen** is responsible for instantiating the pack list items. The **Pack List Item Prefab** is the prefab that is instantiated for each Pack Info set on the GameManager. The PackListItem in the Prefabs folder is the prefab used in the asset:



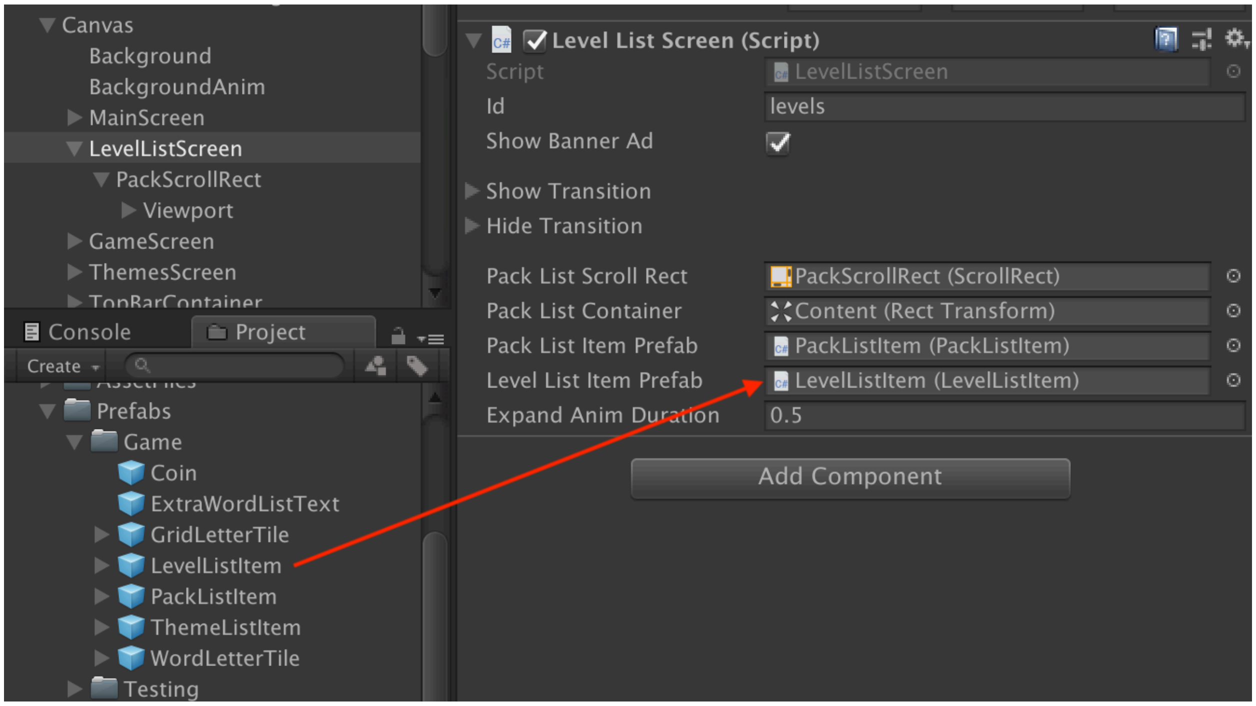
Level List Screen (Script)

Script	LevelListScreen
Id	levels
Show Banner Ad	✓
Show Transition	
Hide Transition	
Pack List Scroll Rect	PackScrollRect (ScrollRect)
Pack List Container	Content (Rect Transform)
Pack List Item Prefab	PackListItem (PackListItem)
Level List Item Prefab	LevelListItem (LevelListItem)
Expand Anim Duration	0.5

Level List Item



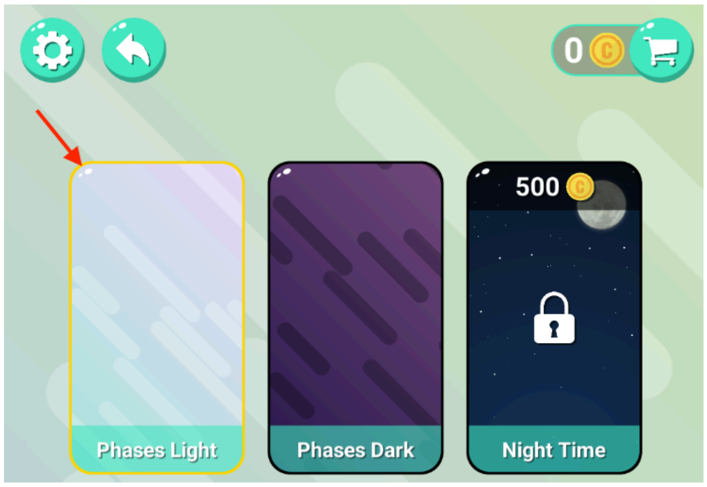
The **LevelListScreen** is responsible for setting up the ObjectPool which is used by individual PackListItems to instantiate copies of the **Level List Item Prefab**. The LevelListItem in the Prefabs folder is the prefab used in the asset:



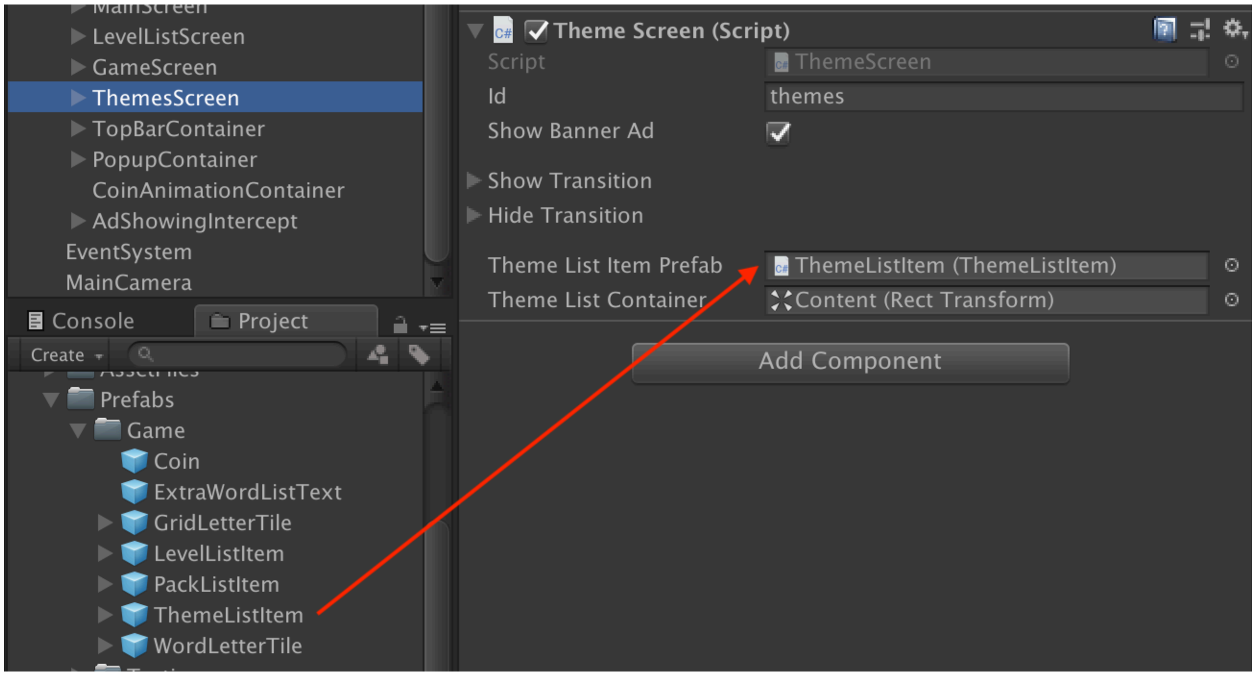
Level List Screen (Script)

Script	LevelListScreen
Id	levels
Show Banner Ad	✓
Pack List Scroll Rect	PackScrollRect (ScrollRect)
Pack List Container	Content (Rect Transform)
Pack List Item Prefab	PackListItem (PackListItem)
Level List Item Prefab	LevelListItem (LevelListItem)
Expand Anim Duration	0.5

Theme List Item

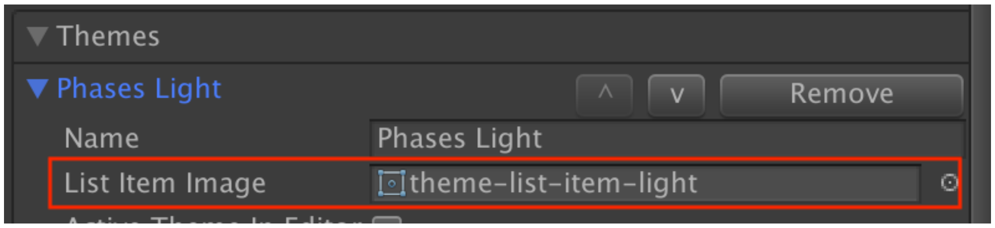


The **ThemesScreen** is responsible for instantiating the theme list items. The **Theme List Item Prefab** is the prefab that is instantiated for each Theme set in the ThemeManager. The ThemeListItem in the Prefabs folder is the prefab used in the asset:



Theme Screen (Script)
Script: ThemeScreen
Id: themes
Show Banner Ad: ✓
Show Transition
Hide Transition
Theme List Item Prefab: ThemeListItem (ThemeListItem)
Theme List Container: Content (Rect Transform)

The Sprite used as the image for the theme list item is set on the Theme in the ThemeManager:



Name: Phases Light
List Item Image: theme-list-item-light