

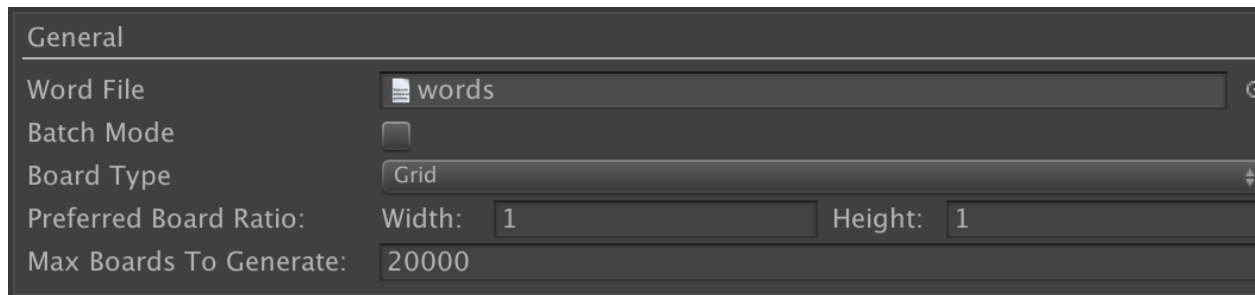
# Word Connect Documentation

<b>Level Builder Window Tool</b>	<b>2</b>
General Settings	2
Import Settings	3
Level Settings	3
Letters/Words In Level	4
Possible Words To Add	4
Word Grid	5
Export	5
Batch Mode	6
<b>Sounds</b>	<b>8</b>
<b>Project Setup</b>	<b>9</b>
GameController	9
UIController	11
Screen Controller	16
Coin Controller	18
PopupController	19
PackItemUI	20
CategoryItemUI	21
LevelItemUI	22
LetterWheel	23
WordBoardGrid	24

# Level Builder Window Tool

The Level Builder window tool is used to create the level files that are loaded when the game runs. To open the window, select the menu item **Tools > Bizzy Bee Games > Word Connect Level Builder**.

## General Settings



The screenshot shows a window titled "General" with the following settings:

Word File	words	
Batch Mode	<input type="checkbox"/>	
Board Type	Grid	
Preferred Board Ratio:	Width: 1	Height: 1
Max Boards To Generate:	20000	

**Word File** - The file that contains all the words in the game. This is the games dictionary and should contain all possible words, each on a newline.

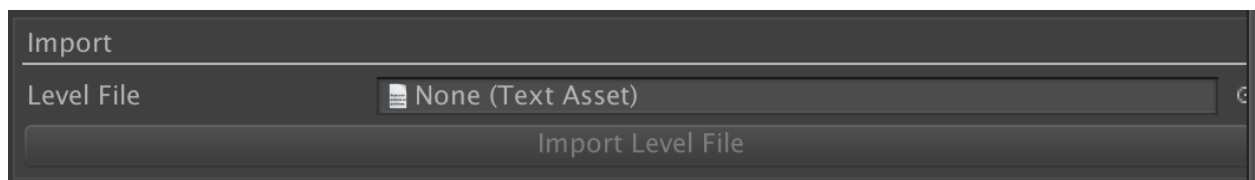
**Batch Mode** - Enables batch generation mode. Batch mode is used to created multiple randomly generated levels.

**Board Type** - The type of board this level is for. Grid will generate/display a grid of words, List will simply display the list of words (No grid).

**Preferred Board Ratio** - The ratio (rows by columns) that the grid should attempt to match. This is used to generate grids that best match the ratio to be displayed in. (Example 1x1 would create mostly squared grids where as 1x2 would created more portrait looking grids)

**Max Boards To Generate** - The way the grid is generate is by generating a bunch of random boards then picking the best one based on the Preferred Board Ratio. This is the maximum number of boards that the algorithm will generate before terminating.

## Import Settings

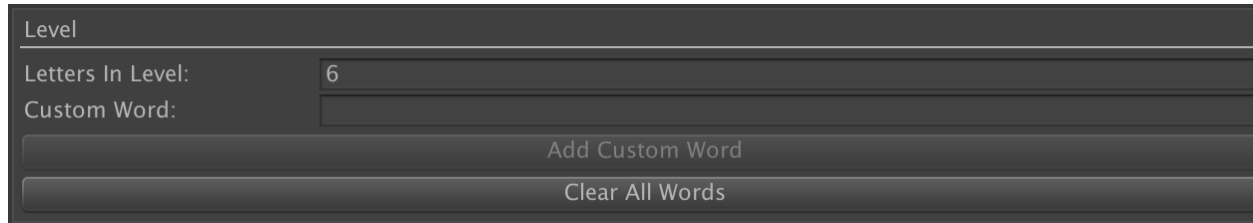


The screenshot shows a window titled "Import" with the following settings:

Level File	None (Text Asset)
Import Level File	

This section of the window is used to import already created level files in order to edit them. To use, drag a level file into the Level File field and click Import Level File.

## Level Settings

A screenshot of the 'Level Settings' window. It has a dark grey background. At the top, there's a label 'Level' followed by a horizontal line. Below this, there are two input fields: 'Letters In Level:' with the value '6' and 'Custom Word:' which is empty. Below the 'Custom Word' field is a button labeled 'Add Custom Word'. At the bottom of the window is a button labeled 'Clear All Words'.

**Letters In Level** - Sets the preferred number of letters for the level. This is the number of letters that will appear in the letter circle in the game. Words will be suggested in the section below based on this number.

**Custom Word** - Used to enter a word manually for the level instead of selecting one in the “Possible Words To Add” section.

## Letters/Words In Level

A screenshot of the 'Letters/Words In Level' window. It has a dark grey background. At the top, there's a label 'Letters In Level' followed by a horizontal line. Below this, the letters 'A C E H R S' are displayed. Below the letters is a label 'Words In Level' followed by a horizontal line. Below this, there's a list of words: 'SEARCH', 'REACH', 'SHARE', 'CHASE', and 'EACH'. To the right of each word are three buttons: 'Bonus', a star symbol (\*), and a minus symbol (-).

The **Letters In Level** section shows a preview of the letters that will appear in the letter circle when the game runs. The **Words In Level** section shows all the words that have been selected and will appear in the level. There are three buttons beside each word:

- **Bonus** : Clicking this button will assign the words as the bonus word. The bonus word will appear outside the grid for Grid type levels (This button does nothing for List type levels)
- \* **[star]** : Clicking the star button will add coins to the word which when found in the game will award those coins to the player.
- - **[minus]** : Clicking the minus buttons will remove the word from the level.

## Possible Words To Add

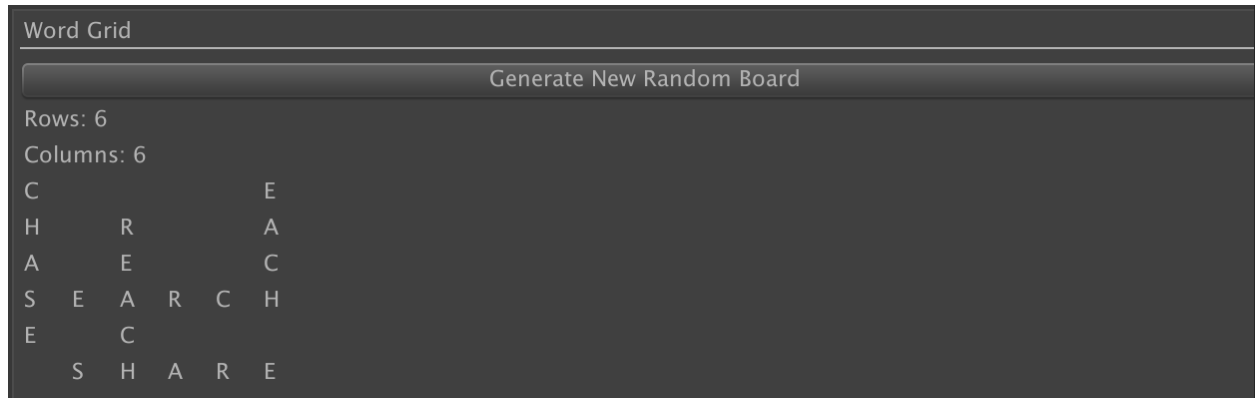


This section shows a list of words that can be added to the level that won't go over the Letters In Level amount. This list can be sorted by a number of ways to help find words of interest.

**Rank** is where the word appeared in the word file. The word file that comes with the asset is sorted by most used words in the English dictionary. The sort types are as follows:

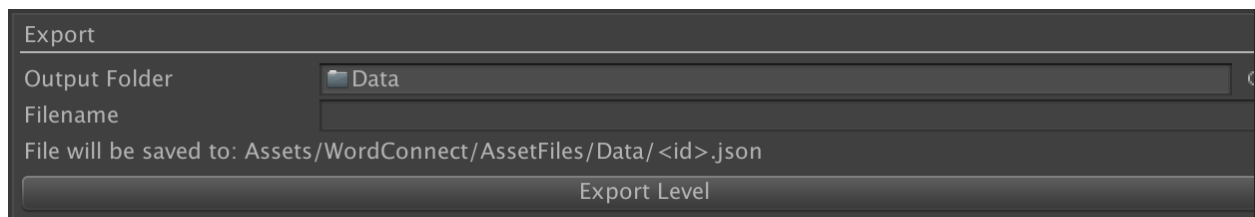
- **Rank** : Sorts the words by rank, smallest to largest (Words that appear earlier in the word file)
- **Rank Shortest First** : Sorts the words by length then by rank so the shortest words appear first.
- **Rank Longest First** : Sorts the words by length then by rank so the longest words appear first.
- **Alphabetically** : Sorts the words alphabetically.
- **Alphabetically Shortest First** : Sorts the words by length then alphabetically so the shortest words appear first.
- **Alphabetically Longest First** : Sorts the words by length then alphabetically so the longest words appear first.

## Word Grid



This section shows a preview of the grid if Grid type is selected. Clicking **Generate New Random Board** will generate a new board but it might not be different depending on the number of possible boards that could be generated since it is trying to find the most optimal board based on the settings in the General section.

## Export



This section is used to export the level when all the desired words have been added. To set the **Output Folder** simply drag a folder from the Project window in Unity into the field. If a **Filename** has not been set the id of the level will be used.

## Batch Mode

Selecting **Batch Mode** in the General settings will switch the window over to batch generation mode. Batch mode is used to create a multiple of randomly generated levels at once. The table below will describe all the settings for batch generation mode.

Batch Level Creation	
Letters In Level:	6
Number Of Levels	1
Min Words Per Level	0
Max Words Per Level	0
Min Word Length	0
Max Word Length	0
Length Pick Type	Random
Rank Pick Type	Random
Choose Bonus Word	<input type="checkbox"/>
Every # of Levels	1
Include First Level	<input type="checkbox"/>
Min Bonus Word Length	0
Max Bonus Word Length	0
Re-Use Words	<input type="checkbox"/>
Load Words From Folder	<input type="checkbox"/> None (Object)
Min Re-Use Word Length	0
Max Re-Use Word Length	0

**Letters In Level** Sets the number of letters for each level. This is the number of letters that will appear in the letter circle in the game.

**Number Of Levels** The number of levels to generate.

**Min Words Per Level** The minimum number of words each level must have.

**Max Words Per Level** The maximum number of words each level can have. Words will continue to be chosen until this number has been reached. Setting this to 0 means there is no maximum, the algorithm will continue choosing words until there are no more valid words for the level.

**Min Word Length** The minimum number of letters a word must have in order to be considered for a level.

**Max Word Length** The maximum number of letters a word can have in order to be considered for a level.

**Length Pick Type** Determines the behavior when choosing words for a level.

- **Random** means length is not taken into consideration when choosing a letter.
- **Longest First** means the longest possible word will be chosen first.
- **Shortest First** means the shortest possible word will be chosen first.
- **Distributed Longest First** means the longest word will be chosen first, followed by the second longest, then the third, etc

- **Distributed Shortest First** is the opposite of Distributed Longest First meaning the shortest word is chosen first then the second shortest, all the way to the longest word at which point it wraps back to the shortest.

**Rank Pick Type** Determines the behavior when choosing words for a level. First Length Pick Type is used to get a list of words by length to choose from then **Rank Pick Type** is used to choose which word from that list is chosen.

- **Random** means a random word is chosen from the list.
- **Most Common First** means the word with the least Rank value (Appears first in the word file) will be chosen.
- **Least Common First** means the word with the highest Rank value (Appears last in the word file) will be chosen.

**Choose Bonus Word** If selected then a bonus word will be chosen for the level. The bonus word in batch generation mode will also be starred meaning it will have coins on it when the level is played in the game.

**Every # of Levels** Determines which levels get the bonus word. For example, setting to 1 means every level will choose a bonus word, setting to 8 means every 8th level will choose a bonus word (Level 8, level 16, etc)

**Include First Level** If selected then the first level generated will get a bonus word.

**Min Bonus Word Length** The minimum word length for a bonus word.

**Max Bonus Word Length** The maximum word length for a bonus word.

**Re-Use Words** If selected then words will be re-used in levels, if false then words that have been used in previous levels will not be re-used again.

**Load Words From Folder** Drag a folder with already generated level files into this field and when the batch generation starts all level files in this folder will be loaded and their words will be considered used when generating new levels.

**Min Re-Use Word Length** The minimum length a word must be in order to be re-used. For example, setting to 4 means all words of length 2 and 3 will not be re-used only words greater than or equal to 4.

**Max Re-Use Word Length** The maximum length a word can be in order to be re-used. For example, setting to 6 means all words of length greater than 6 will not be re-used.

# 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);
```

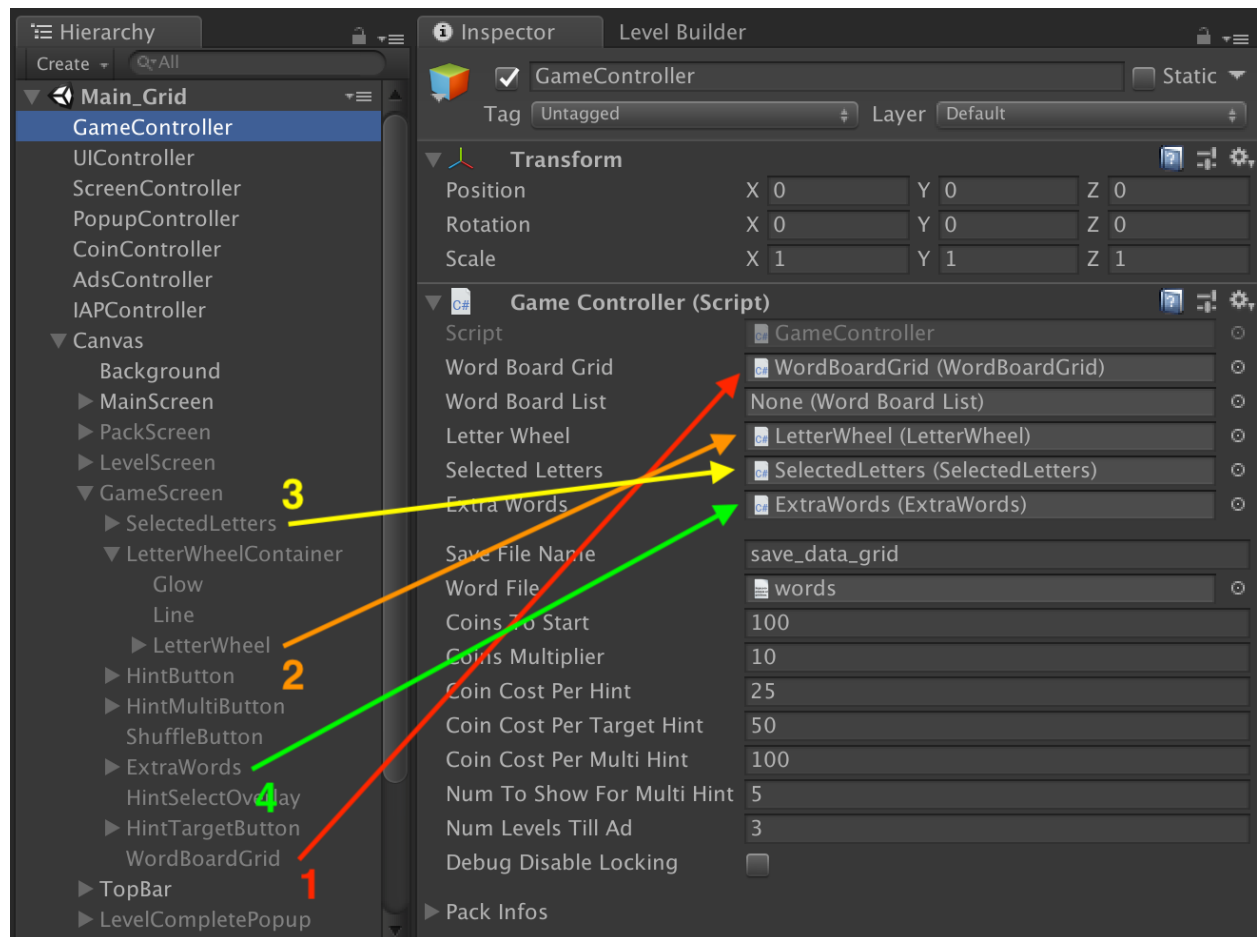
You can easily play a sound when a Button is clicked by adding the **ButtonSound** component to a GameObject with a **Button** component. The ButtonSound will play the sound with the specified Id every time the button is clicked.



# Project Setup

## GameController

The GameController handles most of the core game logic. It's used to start levels and controls what happens when the user selects words or hints.



**1 Word Board Grid** - The Word Board Grid handles displaying the main grid on the screen. It shows blanks cells for words that have not been found and when a word is found it will animate in the word onto the board.

**2 Letter Wheel** - The Letter Wheel component handles displaying the letters that are in the level and notifying a listener when letters are selected.

**3 Selected Letters** - The Selected Letters component handles displaying the letters that have been selected in the Letter Wheel.

**4 Extra Words** - The Extra Words component handles displaying the number of extra words that are found in the level.

## Other GameController Fields:

**Save File Name** - The name the save file will have when the game saves.

**Word File** - The file that contains all the words in the game. This file is used when checking for extra words.

**Coins To Start** - The amount of coins that the player starts with when they first play the game.

**Coin Multiplier** - This multiple is applied whenever the player receives coins.

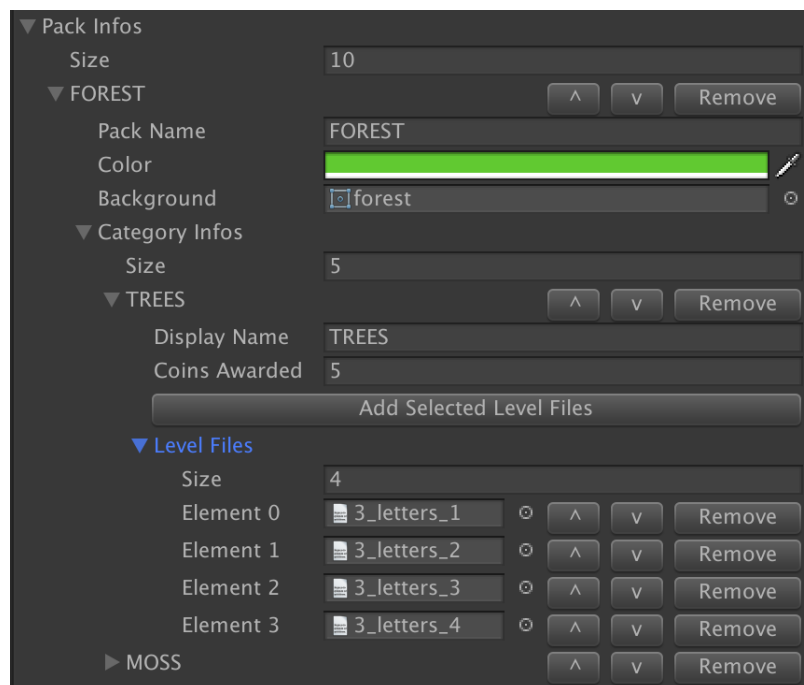
**Coin Cost Per Hint** - The amount of coins each hint costs.

**Coin Cost Per Target Hint** - The amount of coins each target costs.

**Coin Cost Per Multi Hint** - The amount of coins multi-hint costs.

**Num To Show For Multi Hint** - The amount of letters to show on the board when a multi-hint is used.

**Debug Disable Locking** - If this is selected then all levels will be playable on the level screen regardless of the progress. This only applies in the Unity editor, if the game is built to device this will be set to false.



The game is split up into Packs/Categories/Levels. Packs contain any number of Categories and Categories contain any number of Levels. Levels Files are the files that are generated using the **Level Creator Window Tool**.

### Pack Info:

**Pack Name** - The display name of the pack.

**Color** - The color of the pack. This color will be used for various elements on the pack screen game screen.

**Background** - This Sprite will be used as the background when a level in this pack is played.

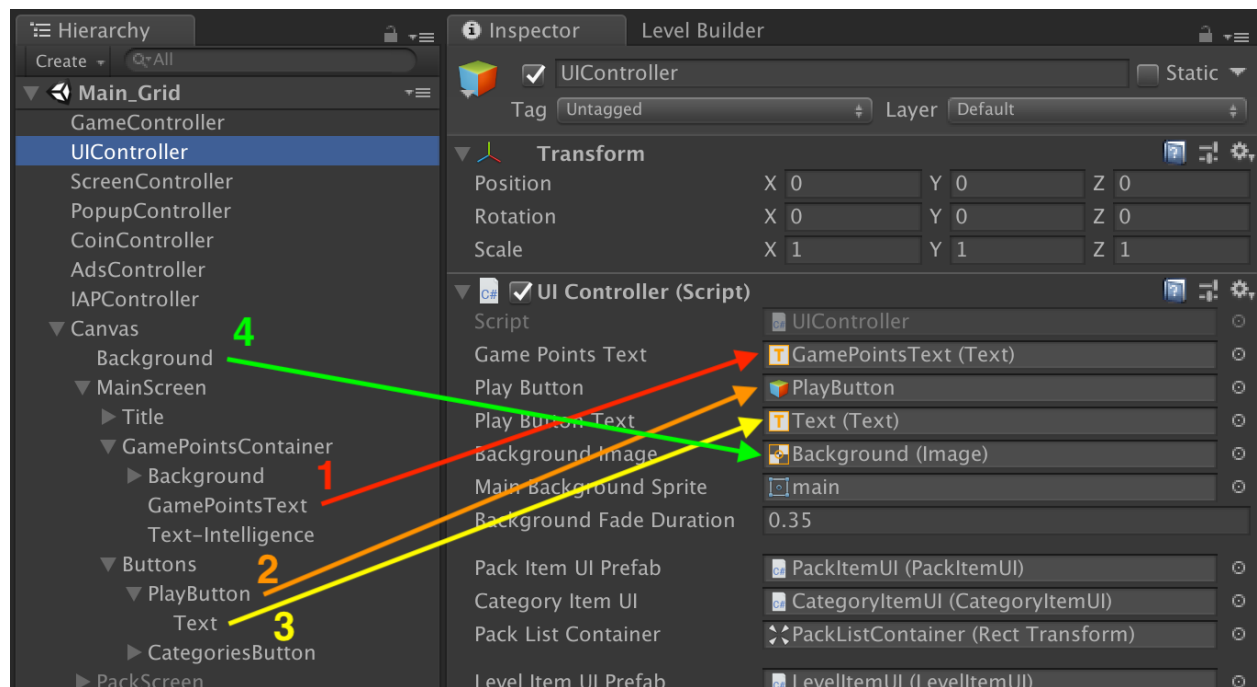
### Category Info:

**Display Name** - The display name of the category.

**Coins Awarded** - The amount of coins that are awarded when all levels in this category have been completed.

## UIController

The UIController handles updating most of the UI elements on various screens. It is also responsible for creating the Pack, Category, and Level UI list items.



**1 Game Points Text** - The Text component on the Main screen to update when the player receives game points from completing levels.

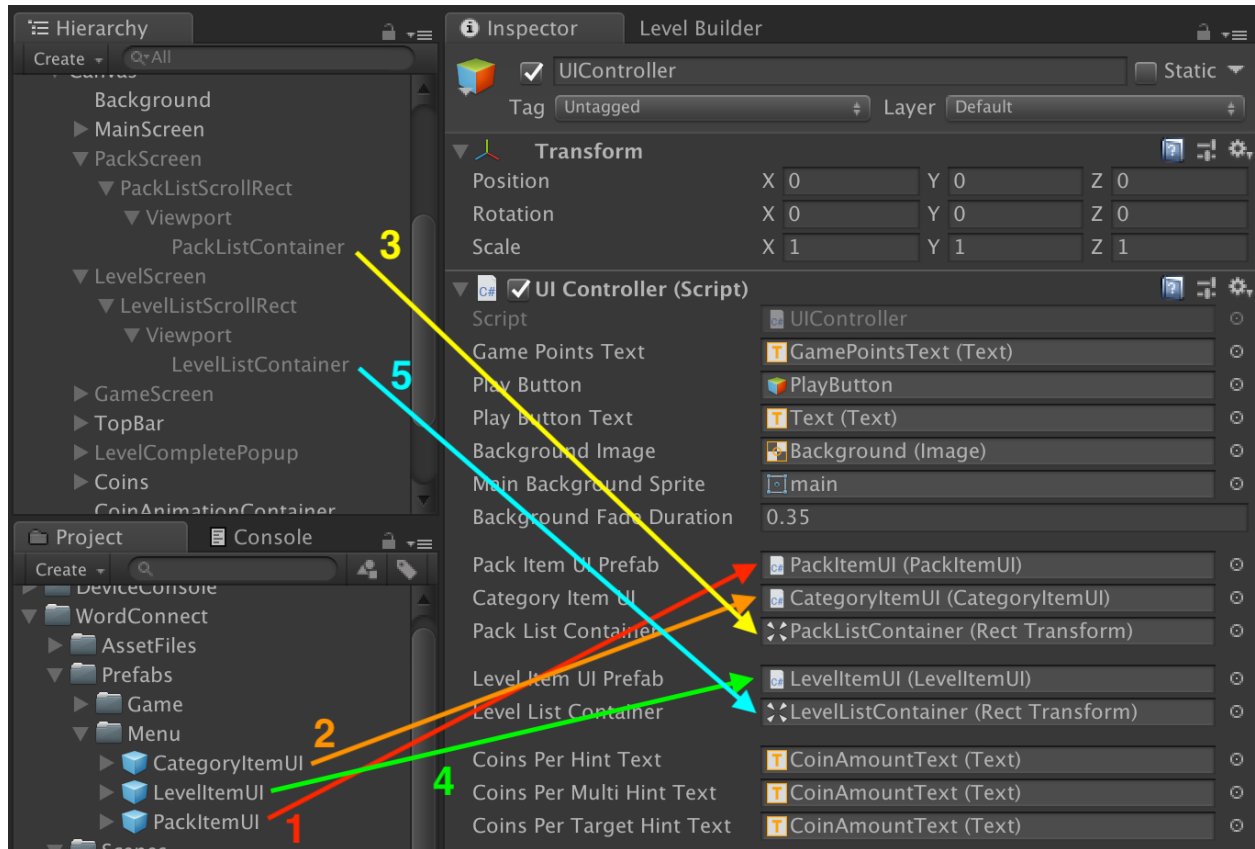
**2 Play Button** - A reference to the Main screens play button, this buttons GameObject will be set to de-active when all levels in the game have been completed.

**3 Play Button Text** - The Text component that will display the next level number to play.

**4 Background** - The Image component that will display the background sprite for the current active level.

**Main Background Sprite** - The Sprite that will be used as the background on the Main screen and Pack screen.

**Background Fade Duration** - The animation duration in seconds for fading in a new background when the background Sprite changes.



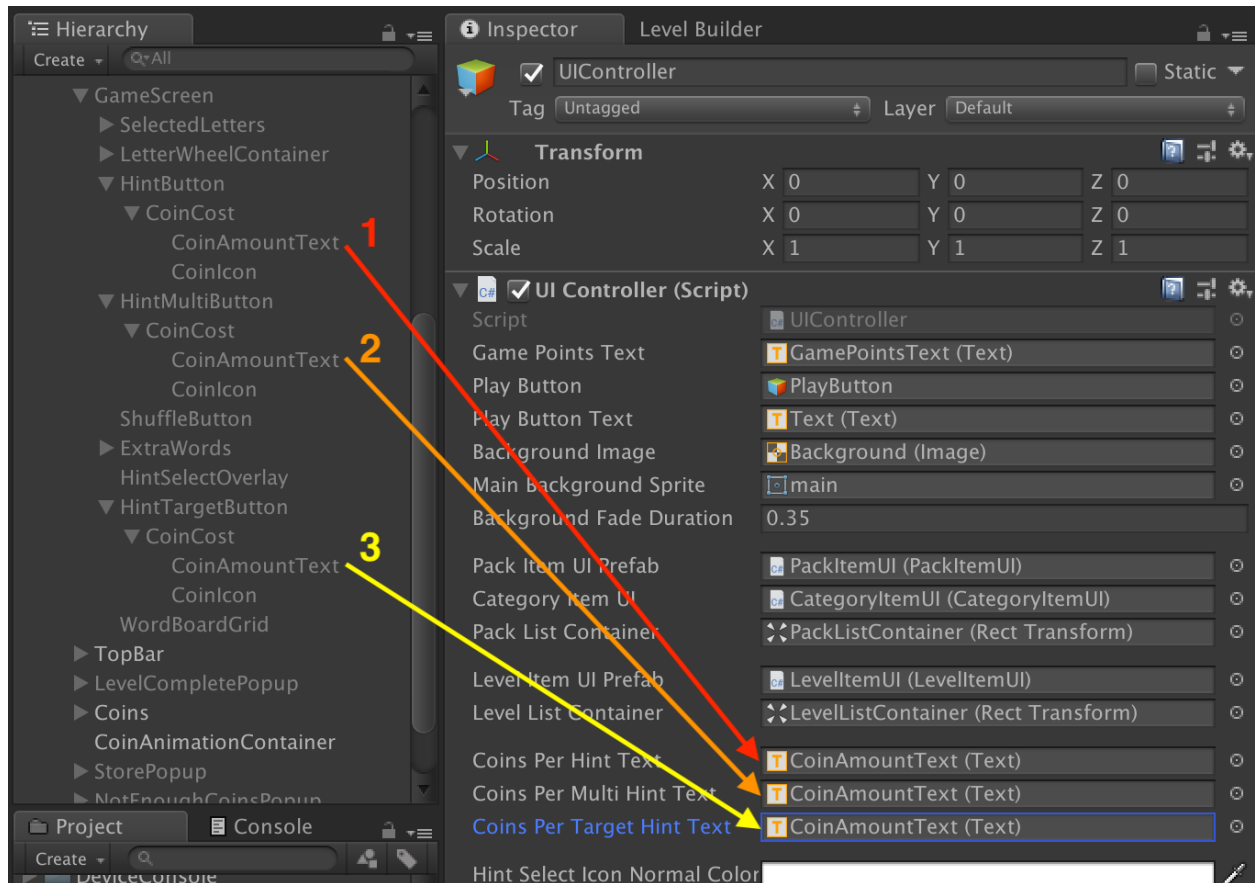
**1 Pack Item UI Prefab** - The prefab that will be used to create pack UI items for each pack in the game.

**2 Category Item UI Prefab** - The prefab that will be used to create category UI items for each category in the game.

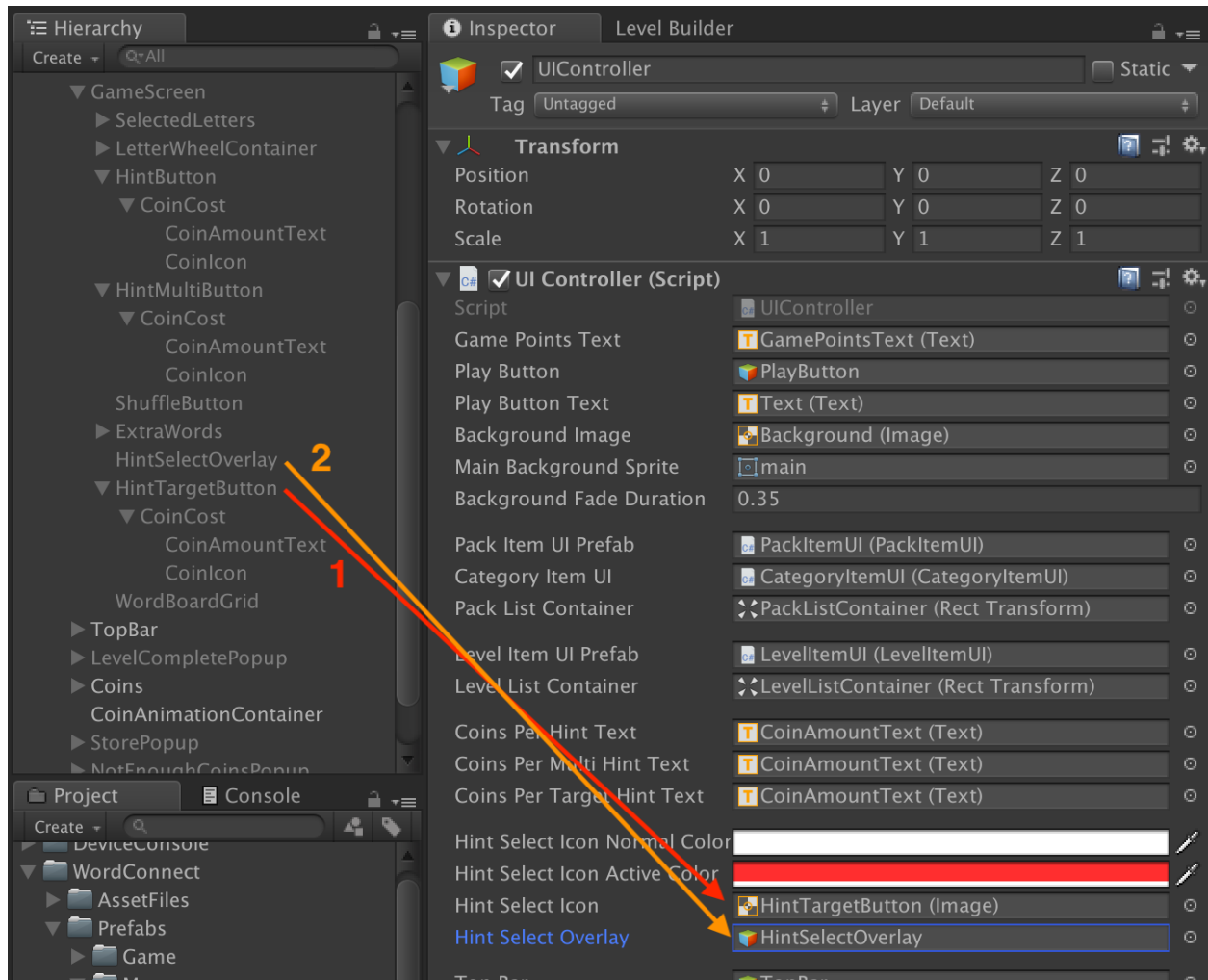
**3 Pack List Container** - The Transform that pack UI items will be played in.

**4 Level Item UI Prefab** - The prefab that will be used to create level UI items for each level in the game.

**5 Level List Container** - The Transform that category UI items will be played in.

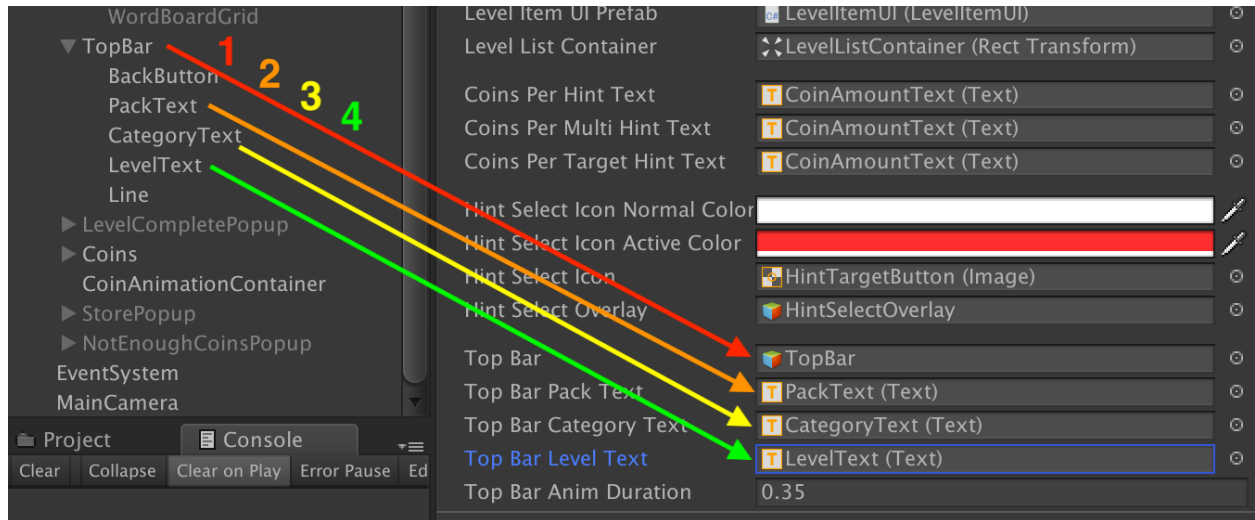


- 1 Coins Per Hint Text** - The Text component that will display the amount of coins a hint costs.
- 2 Coins Per Multi Hint Text** - The Text component that will display the amount of coins a multi-hint costs.
- 3 Coins Per Target Hint Text** - The Text component that will display the amount of coins a target hint costs.



**1 Hint Select Icon** - The Image component that will have its color set when the target hint is selected.

**2 Hint Select Overlay** - The GameObject that will be set to active when the target hint is selected. This is meant to block all other UI elements other than the cells on the WordBoardGrid so the user can only select a cell on the board.



**1 Top Bar** - The GameObject which has all top bar UI elements as its children. This GameObject is faded out when the Main screen is presented and faded in when the user navigates to any other screen.

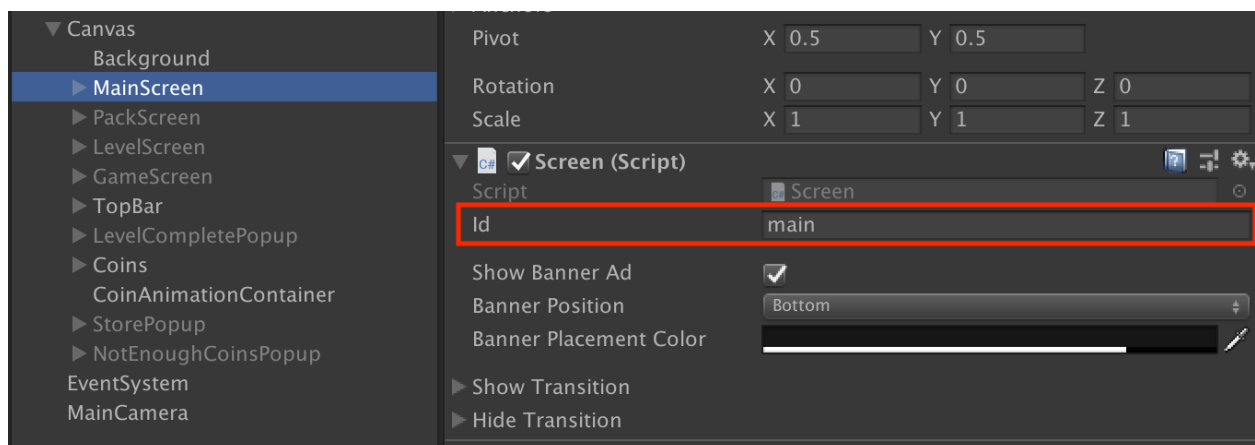
**2 Top Bar Pack Text** - The Text component which will display the name of the selected pack on the Level screen.

**3 Top Bar Category Text** - The Text component which will display the name of the category for the active level on the Game screen.

**4 Top Bar Level Text** - The Text component which will display the level number for the active level on the Game screen.

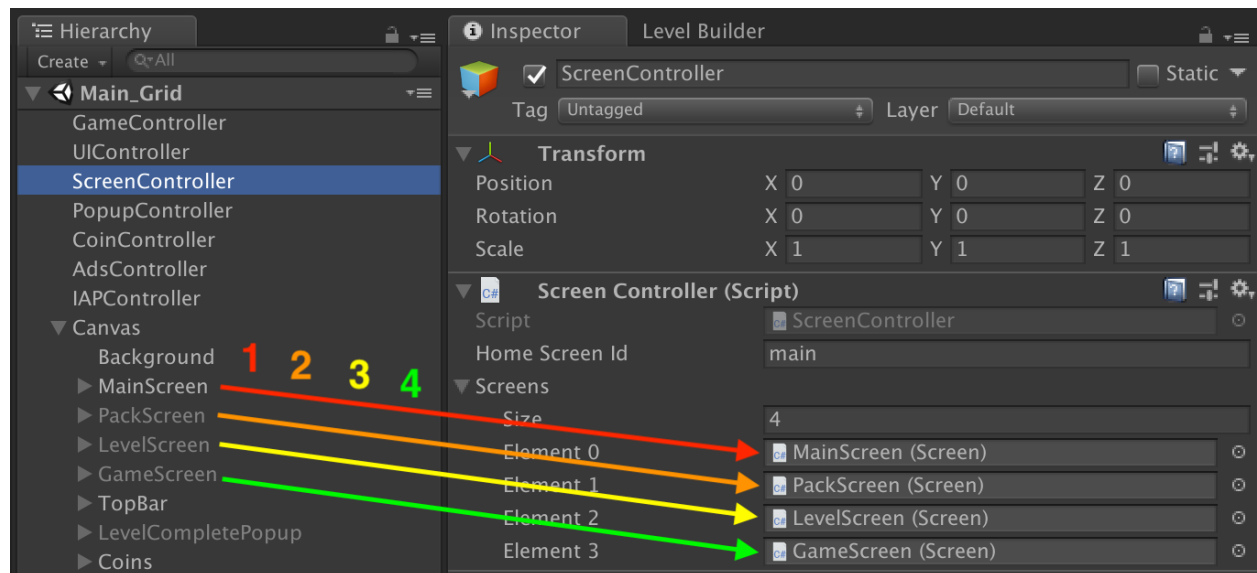
## Screen Controller

The screen controller handles transitioning between the screens in the game. A screen can be transition to using **ScreenController.Instance.Show(screenId)**. The screenId is the Id that is set on the screen:



The ScreenController keeps track of the screens that have been shown, calling the method **ScreenController.Instance.Back()** will transition from the current screen to the last shown Screen.

There are four screens in the Word Connect Game asset:



**1 MainScreen** - The first screen that the player sees when the app launches.

**2 PackScreen** - The screen that displays the list of packs and categories. Selecting a category transitions to the Level Screen,

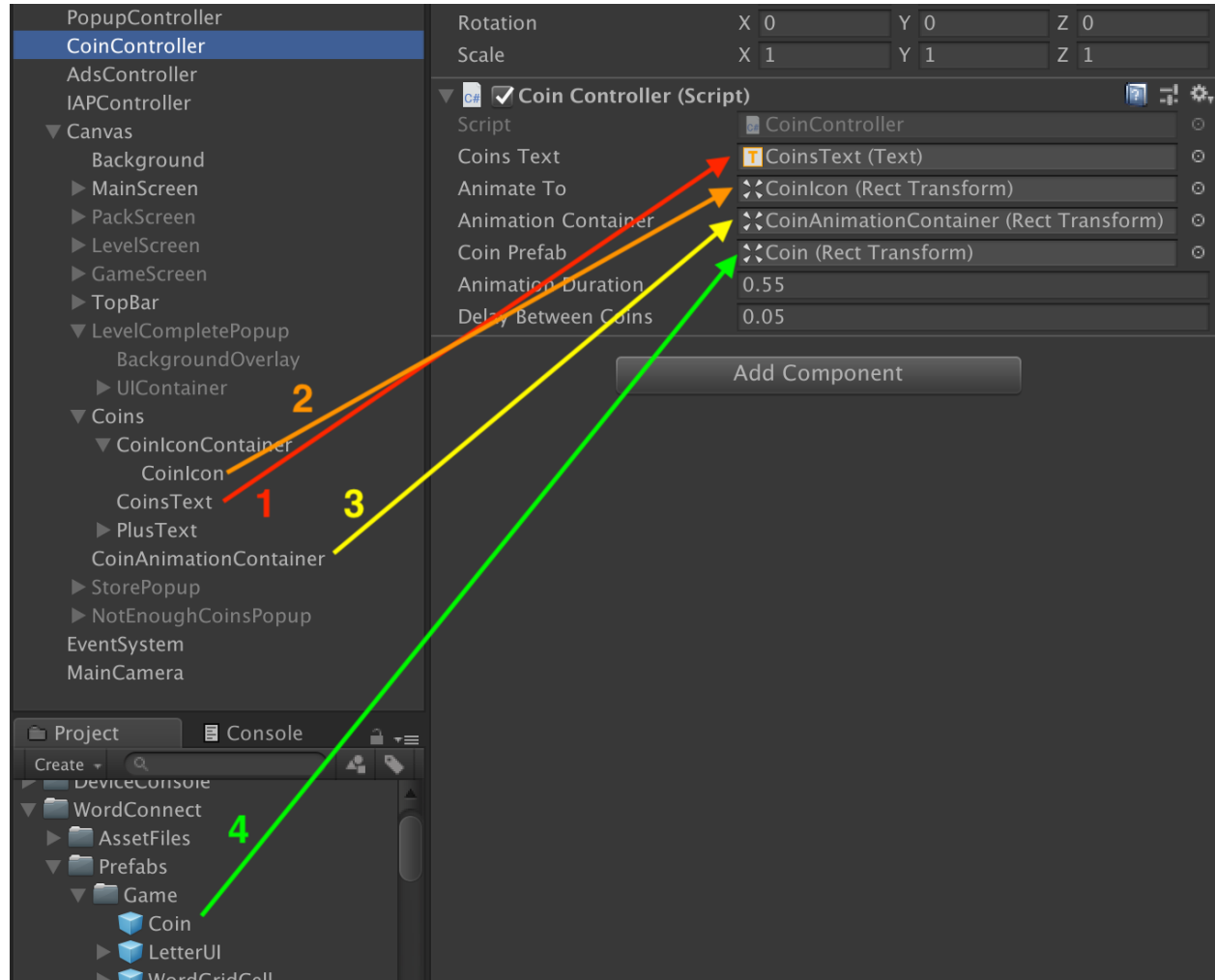
**3 LevelScreen** - The screen that displays the levels for the selected category.

**4 GameScreen** - The screen that displays when a level is being played. Selecting a level or clicking play on the Main screen transitions to this screen.



# Coin Controller

The CoinController handles updating the text that shows the amount of coins the player currently has and animating coin objects when the player receives coins.



**1 Coins Text** - The Text component to update when the player receives coins.

**2 Animate To** - The RectTransform that coin objects will animate to when the player receives coins.

**3 Animation Container** - The Transform that instantiated coin objects will be placed in.

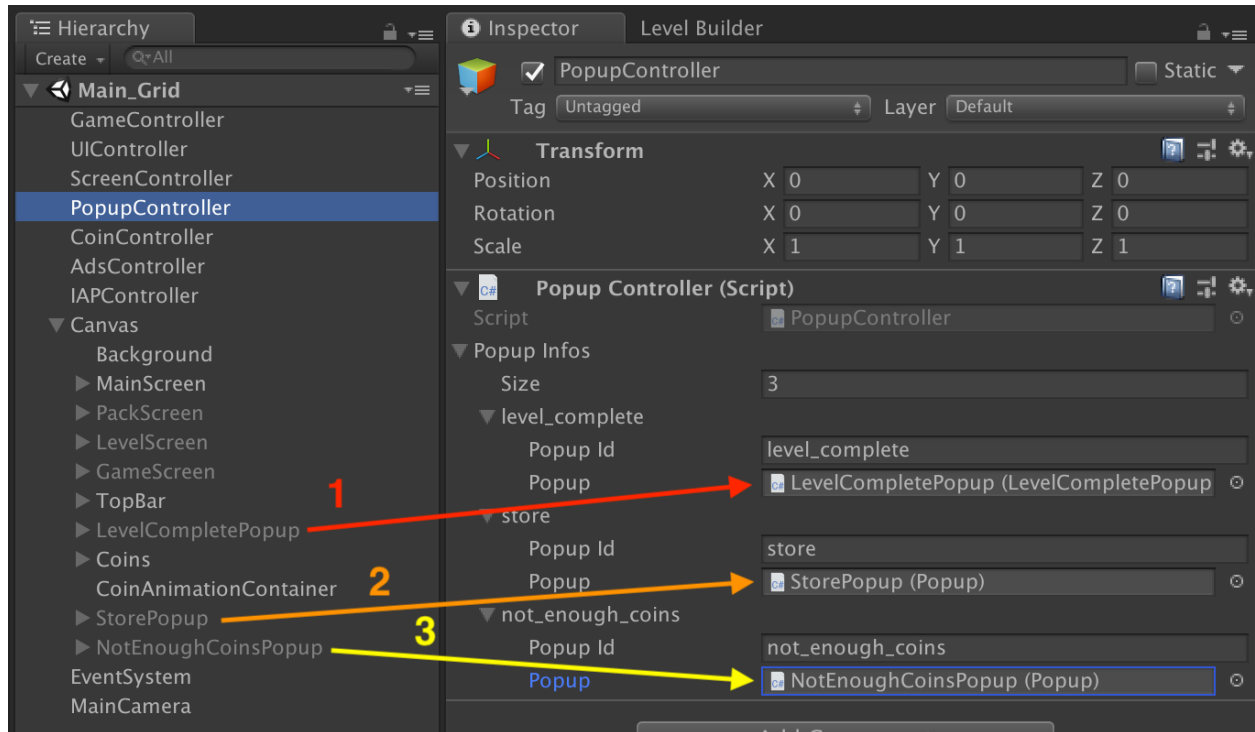
**4 Coin Prefab** - The prefab that will be instantiated when coins need to animate.

**Animation Duration** - The amount of time in seconds the coin animation takes.

**Delay Between Coins** - The amount of time in seconds each coin will wait until starting its Animation.

# PopupController

The PopupController handles displaying popups. A popup can be displayed by calling `PopupController.Instance.Show(id, data, onPopupClosed)`. The id is the Popup Id set on the PopupController, data is an object array containing the data that the popup is expecting, and onPopupClosed is a callback which is invoked when the popup closes.



**1 LevelCompletePopup** - Displayed when a level is completed.

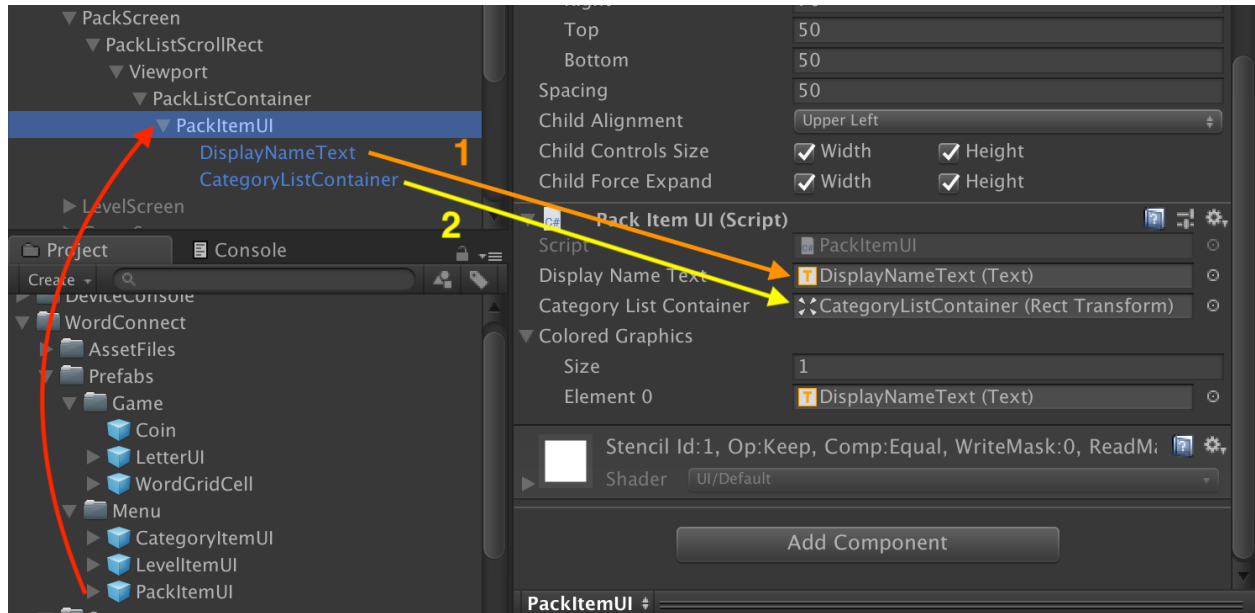
**2 StorePopup** - Displayed when the player coins are clicked.

**3 NotEnoughCoinsPopup** - Displayed when the player attempts to use a hint but does not have enough coins.

# PackItemUI

The PackItemUI component handles how a pack should be displayed in the pack list. A PackItemUI prefab exists in the Prefabs/Menu folder and is used by the UIController when instantiating new pack UI list items to be placed in the pack list.

To edit/change the look of a pack UI item drag the PackItemUI prefab into the PackListContainer found in the PackScreen:



**1 Display Name Text** - The Text component that will display the name of the pack.

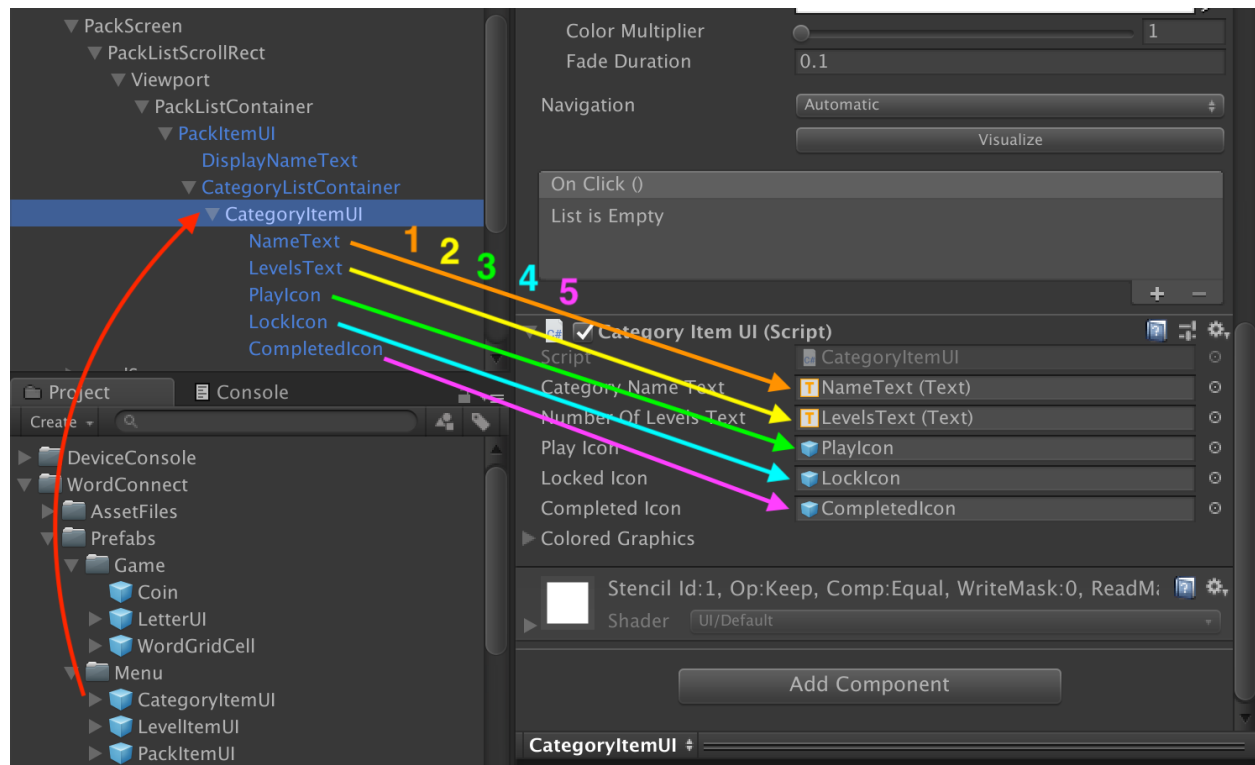
**2 Category List Container** - The Transform that CategoryItemUIs will be instantiated and placed in.

**Colored Graphics** - The list of Graphic components (Text, Image, etc) who's color field will be set the color of the pack.

# CategoryItemUI

The CategoryItemUI component handles how a category list item will be displayed. A CategoryItemUI prefab exists in the Prefabs/Menu folder and is used by the UIController when instantiating new category list items to be placed in a packs category list.

To edit/change the look of a category item first drag a PackItemUI prefab into the PackListContainer found in the PackScreen. Next drag a CategoryItemUI prefab into the PackItemUI's CategoryListContainer:



**1 Category Name Text** - The Text component that will display the name of the category.

**2 Number Of Levels Text** - The Text component that will display the to/from level numbers.

**3 Play Icon** - The GameObject that will be set to active if this category is the next playable category.

**4 Locked Icon** - The GameObject that will be set to active if this category is locked.

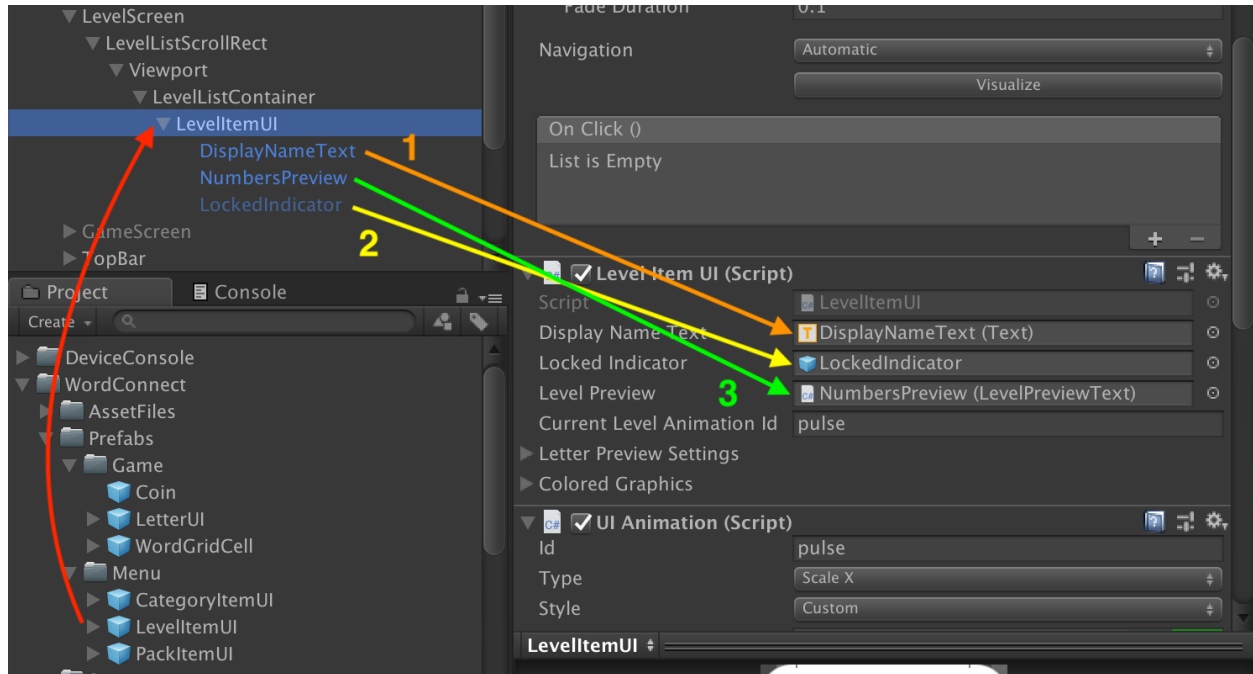
**5 Completed Icon** - The GameObject that will be set to active if all levels in the category have been completed.

**Colored Graphics** - The list of Graphic components (Text, Image, etc) who's color field will be set the color of the pack that this category belongs to.

# LevelItemUI

The LevelItemUI component handles how a level list item will be displayed. A LevelItemUI prefab exists in the Prefabs/Menu folder and is used by the UIController when instantiating new level list items to be placed on the Level Screen.

To edit/change the look of a level item drag the LevelItemUI prefab into the LevelListContainer found in the LevelScreen:



**1 Display Name Text** - The Text component that will display levels number.

**2 Locked Indicator** - The GameObject that will be set to active if the level is locked.

**3 Level Preview** - The LevelPreview component that displays the levels letters in a circle.

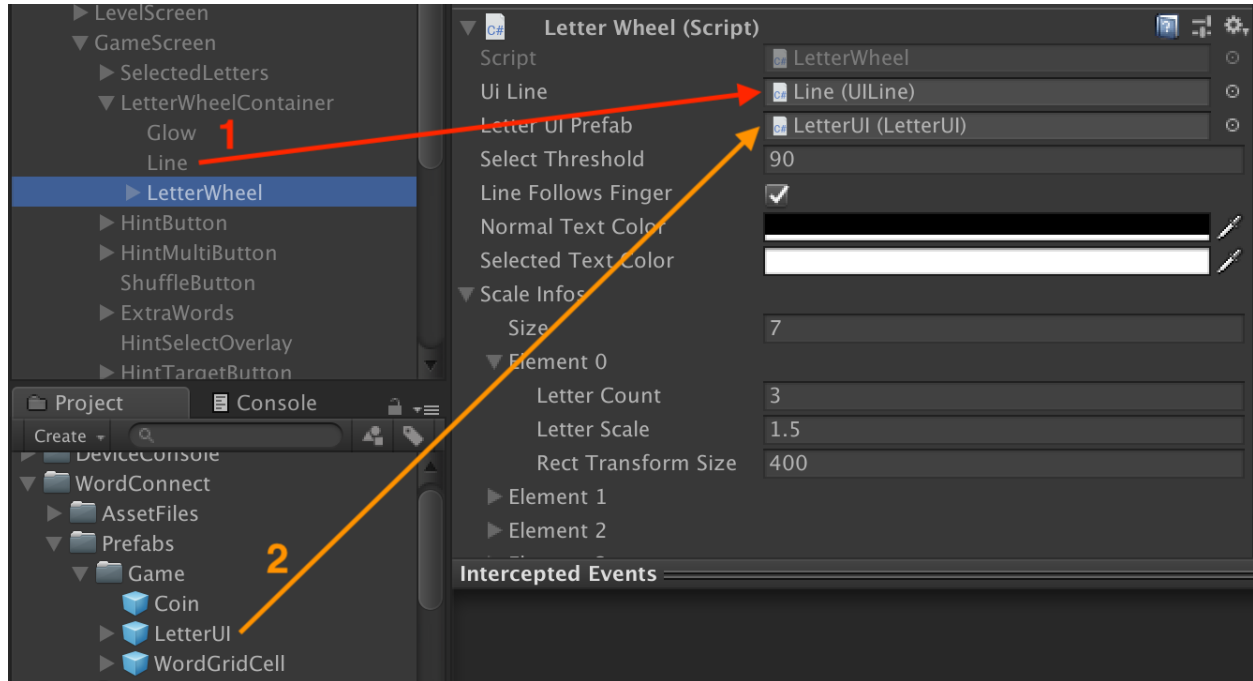
**Current Level Animation Id** - The UI Animation Id that should play when the LevelItemUI is the active/next level to be played.

**Letter Preview Settings** - List of settings to set on the LevelPreview object. Letter Count is the number of letters in the level for this setting to be used.

**Colored Graphics** - The list of Graphic components (Text, Image, etc) who's color field will be set the color of the pack that this level belongs to.

# LetterWheel

The LetterWheel component handles displaying the letters in the active level in a circle and invoking an event when the user selected letters.



**1 Ui Line** - The UILine component that draws the line between selected letters.

**2 Letter UI Prefab** - The LetterUI prefab that will be instantiated for each letter in a level.

**Select Threshold** - The distance the mouse/finger needs to be from a letter in order to selected it.

**Line Follows Finger** - If selected then the line will be drawn from the last selected letter to the location of the mouse/finger.

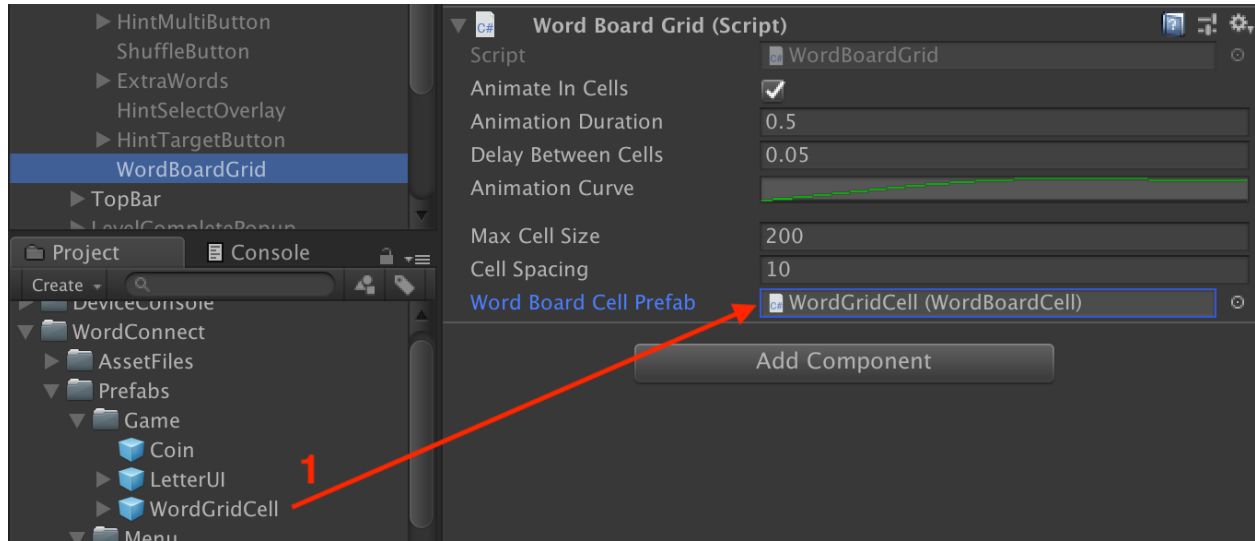
**Normal Text Color** - The text color for un-selected letters.

**Selected Text Color** - The text color for selected letters.

**Scale Infos** - List of settings to apply to the LetterWheel and it's letters. Letter Count is the number of letters in the level for this setting to be used. Letter Scale is a scale that is applied to each letter object, Rect Transform Size is the size to set the LetterWheels RectTransform.

# WordBoardGrid

The WordBoardGrid component handles displaying a Grid level. It creates the objects required to display the levels grid and animates in the letters as they are found.



**1 Word Board Cell Prefab** - The WordBoardCell prefab to use when instantiating cells in the grid.

**Animate In Cells** - If selected then the letters will animate in when a word is found.

**Animation Duration** - The duration of a single cells animation.

**Delay Between Cells** - The amount of time to wait before starting each cells animation when a word has been found.

**Animation Curve** - The curve to use when animating in a cell.

**Max Cell Size** - The maximum size a single cell can be on the screen when scaling the grid to fit the bounds of the WordBoardGrids RectTransform.

**Cell Spacing** - The amount of spacing between cells in the grid.