Word Search Documentation

Welcome to the Word Search documentation. Below you will find a brief overview of the important scripts used in the game, how they are setup, and how to change out some of the assets used.

BoardCreator.cs

The BoardCreator script is used to generate a Board object that describes how letters should be placed on the CharacterGrid and where the words are located on the board.

BoardCreator works by using a standard backtracking algorithm to place the words on the board one at a time. The algorithm is given a time limit for how long it can run until it stops. This is because we cannot determine it all the given words can even be placed on the board and it takes a really long time to find a placement for all words. So the algorithm will keep trying different placements for words and when the time runs out it picks the board with the most words placed on it and returns that one.

The algorithm is started by calling the public static method **BoardCreator.CreateBoard** and passing it a BoardConfig object that provides the settings for the board and also a callback that will be called when its done generating the Board.

BoardConfig Fields:

rowSize	The number of rows on the board.
columnSize	Then number of columns on the board.
words	The list of words to place on the board (Note: It is possible that not all of these words will be placed on the board)
filterWords	The list of words that should not appear on the board when placing random characters.
randomCharacters	A string that represents all the characters that can be randomly placed on the board when filling all the empty spaces that are not taken up by words.
algoTimeoutInMilliseconds	The amount of time in milliseconds that the algorithm will run for until it stops.

WordSearchController.cs

The WordSearchController script controls most of the game. It handles things like starting the creation of word search board based on what categories/difficulties are selected, saving/loading the game state, and checking if selected words are words that need to be selected.

Word Files:

Each category needs a word file that contains all the words that can be placed on a board. Each word must be placed on its own line and should not contain spaces. The word files that comes with the Word Search asset can be found in **AssetsFiles/Data/…**

Events:

There are two public System.Actions that are invoked by WordSearchController:

OnBoardStateChanged : Invoked when the ActiveBoardState property changes.
OnWordFound : Invoked when a new word on the board has been found by the player.

Inspector Properties:

Character Grid	Reference to the CharacterGrid component in the hierarchy.
Word List	Reference to the WordList component in the hierarchy.
Random Characters	The string of characters to use when placing random characters in empty spaces on the board after all the words have been placed.
Mode Infos	List of modes to use in the game.
Difficulty Infos	List of difficulties to use in the game.
Category Infos	List of categories to use in the game.
Filter Words	List of words that should never appear on the board. Use this to keep things like curse words from appearing on the board when random letters are being placed.

Inspector Properties - Mode Info

Mode Name	The display name for the mode.
Sort Type	How the list of words on the board is sorted and displayed in the WordList.

Is Timed	If checked then games started in this mode will be timed, when the time runs out the game is over.
Is Sequential	If checked then games started in this mode will have the words appear in the WordList one at a time, the player must find that word first before continuing to the next word.

Inspector Properties - Difficulty Info

Difficulty Name	The display name for the difficulty.
Board Row Size	The number of letter rows on a generated board.
Board Column Size	The number of letter columns on a generated board.
Max Words	The maximum number of words that can appear on the board.
Max Word Length	The maximum length of any word that can be placed on the board.
Timed Mode Start Time	The amount of time in seconds the player has at the start of the board if the selected mode is sequential.
Timed Mode Found Bonus	The amount of time in seconds that is added to the time remaining when a word is found on the board.

Inspector Properties - Category Info

Category Name	The display name for the category.
Category Icon	The display image for the category.
Word Files	The file that contains all the words that can be placed on a board for this category.

CharacterGrid.cs

The CharacterGrid script handles displaying a Board on the screen, highlighting words, and notifying the WordSearchController when a group of words is selected.

It uses the RectTransforms area its attached to as a the area to place the grid. It will attempt to fill this area with the letters for the given board and will scale the cells up/down to fill the space. If the cells reach the maximum size then the grid will be centered.

Inspector Properties:

Max Cell Size	The maximum size of a cell in the grid.
Spacing	The amount of horizontal/vertical space between the letters.
Padding	The amount of padding between the CharacterGrids RectTransform and the edge of the letter grid.
Letter Font	The font to use for the letters.
Letter Font Size	The font size to use for the letters, CharacterGrid will set the Text component to this font size then scale the GameObject up/down so the letter fits the cell. You should pick a large enough font so that the letters are always scaled down to fit the grid instead of scaled up so they don't look blurry.
Letter Color	The color of the letters.
Letter Offset In Cell	The amount of space of offset the letters in their cells. Use this if the font used has “extra space” at the bottom in order to center the letter in their cell.
Highlight Sprite	The Sprite to use for highlights.
Highlight Extra Size	Extra height / thickness to give to highlights.
Highlight Alpha	The alpha to give to highlights.
Highlight Type	Specifies how the colors for the highlight should be chosen.
Red Min	The minimum red value for Range type.
Red Max	The maximum red value for Range type.
Green Min	The minimum green value for Range type.
Green Max	The maximum green value for Range type.
Blue Min	The minimum blue value for Range type.
Blue Max	The maximum blue value for Range type.
Custom Colors	The list of colors to randomly choose from for highlights if the type is set to Custom.

WordList.cs

The WordList script handles displaying the words that are on the board that need to be found and also displaying what words have been found.

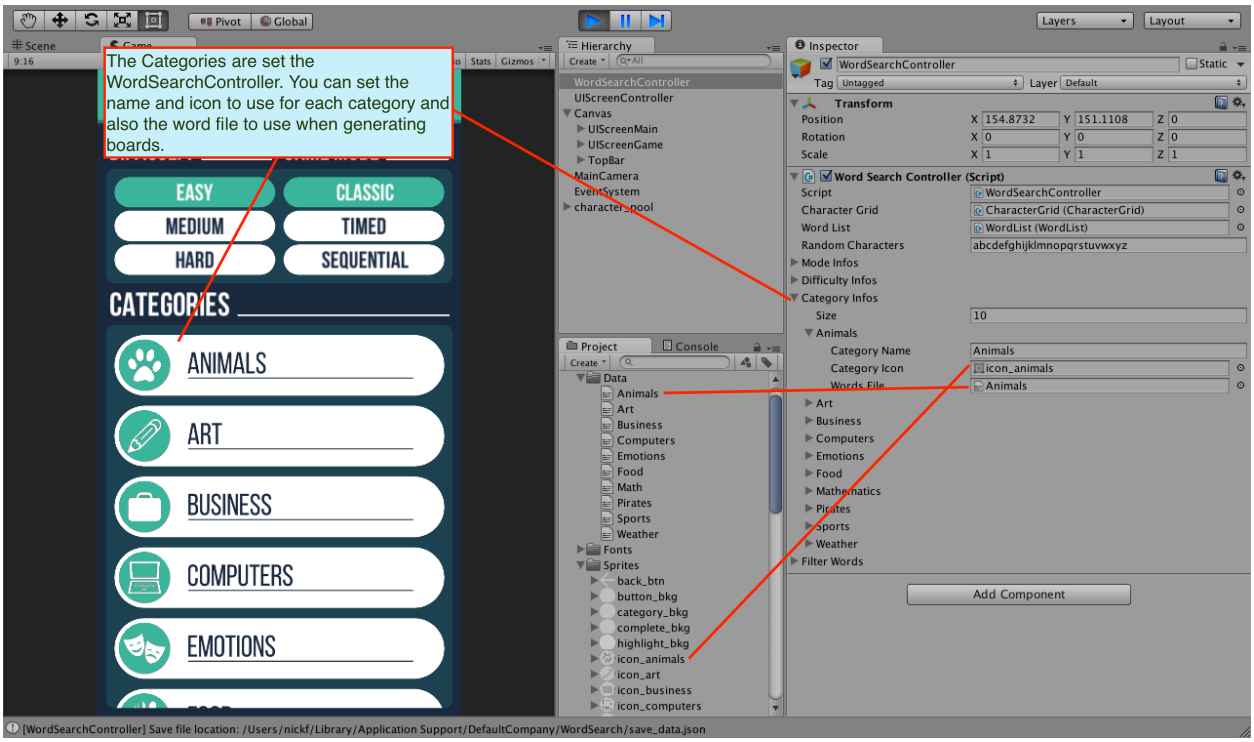
Inspector Properties:

Word List Container	Reference to the RectTransform where all the WordListItem objects will be placed in.
Word List Item Prefab	The WordListItem prefab that will be used for each word on boards that are not sequential.
Word List Canvas Group	Canvas group to nicely fade in the word list when the board is finished generating.
Sequential Container	Reference to the RectTransform where all the WordListItemSequential objects will be placed in.
Sequential Item Prefab	The WordListItemSequential prefab that will be used for each word on a sequential board.

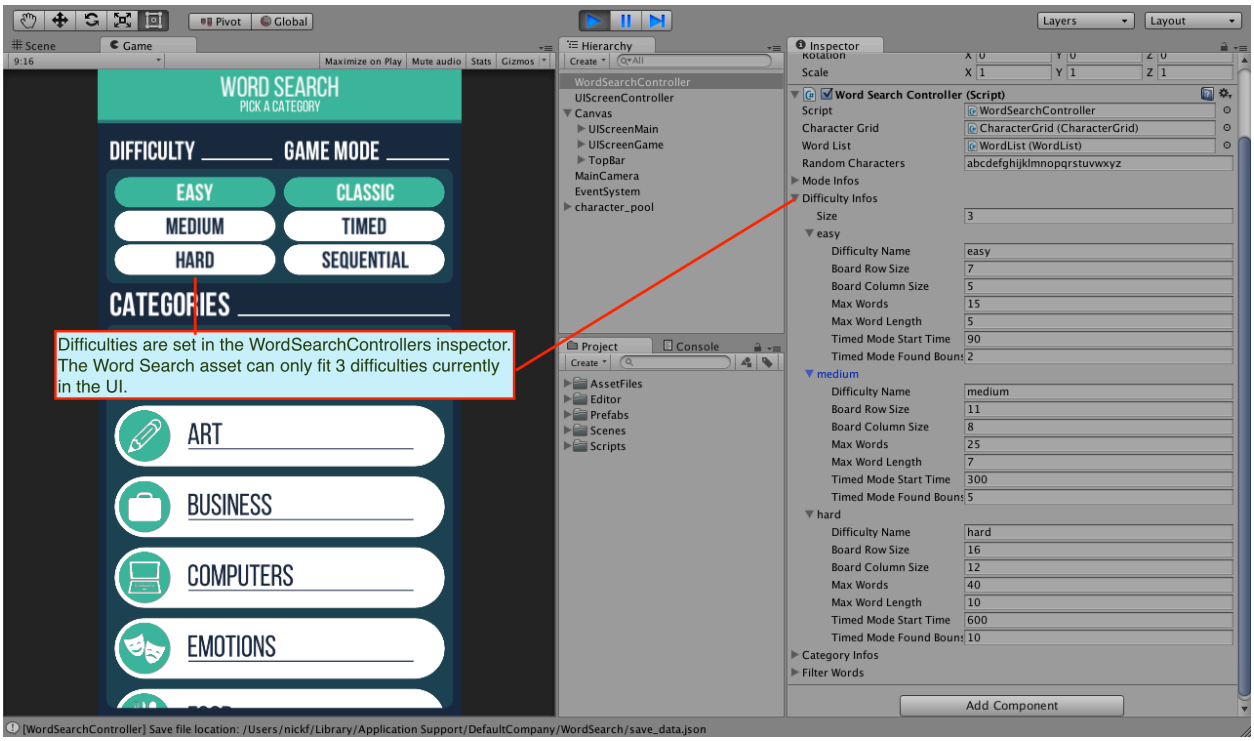
Project Setup

Below are some images that describe how some of the important parts of the project are setup.

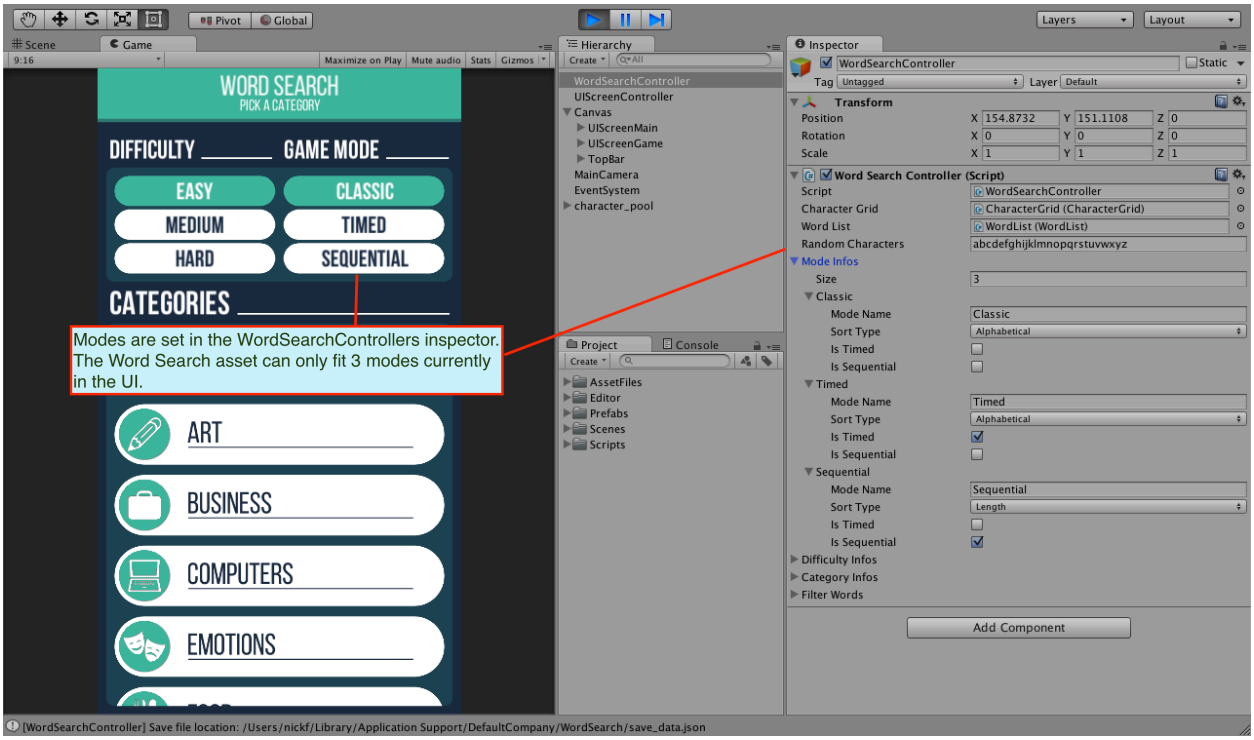
Categories



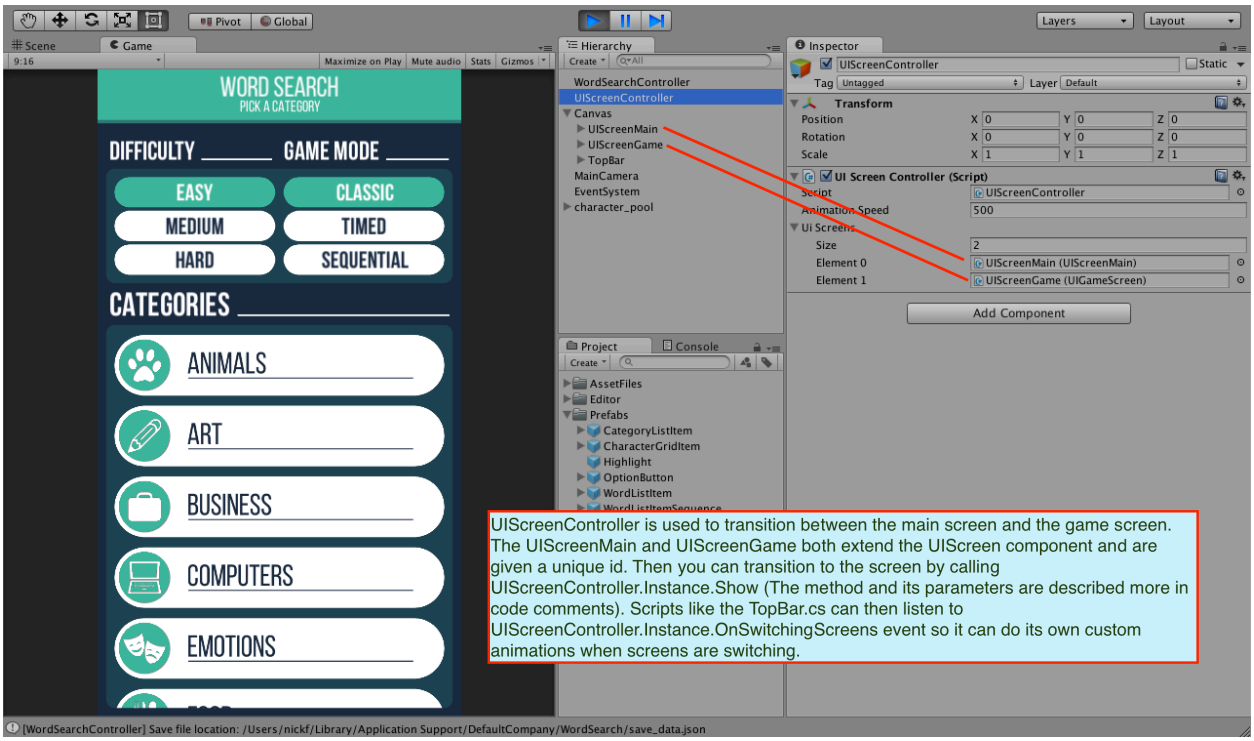
Difficulties



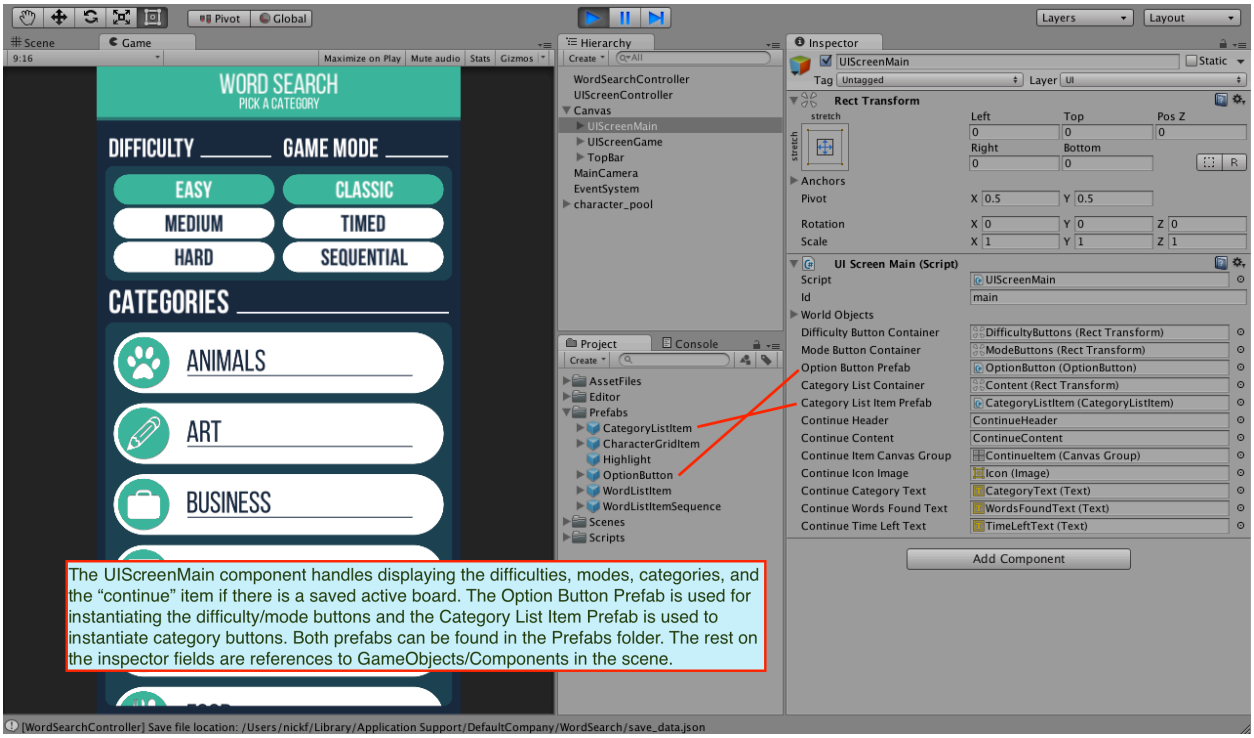
Modes



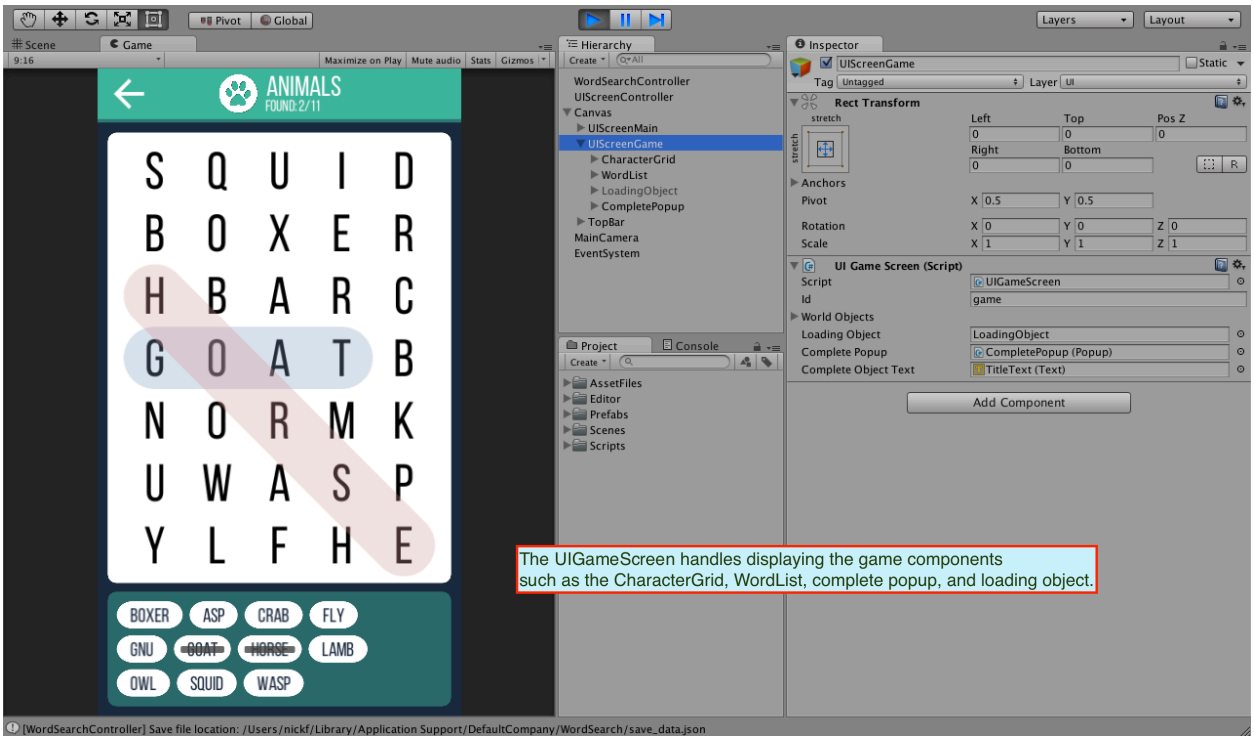
UIScreenController



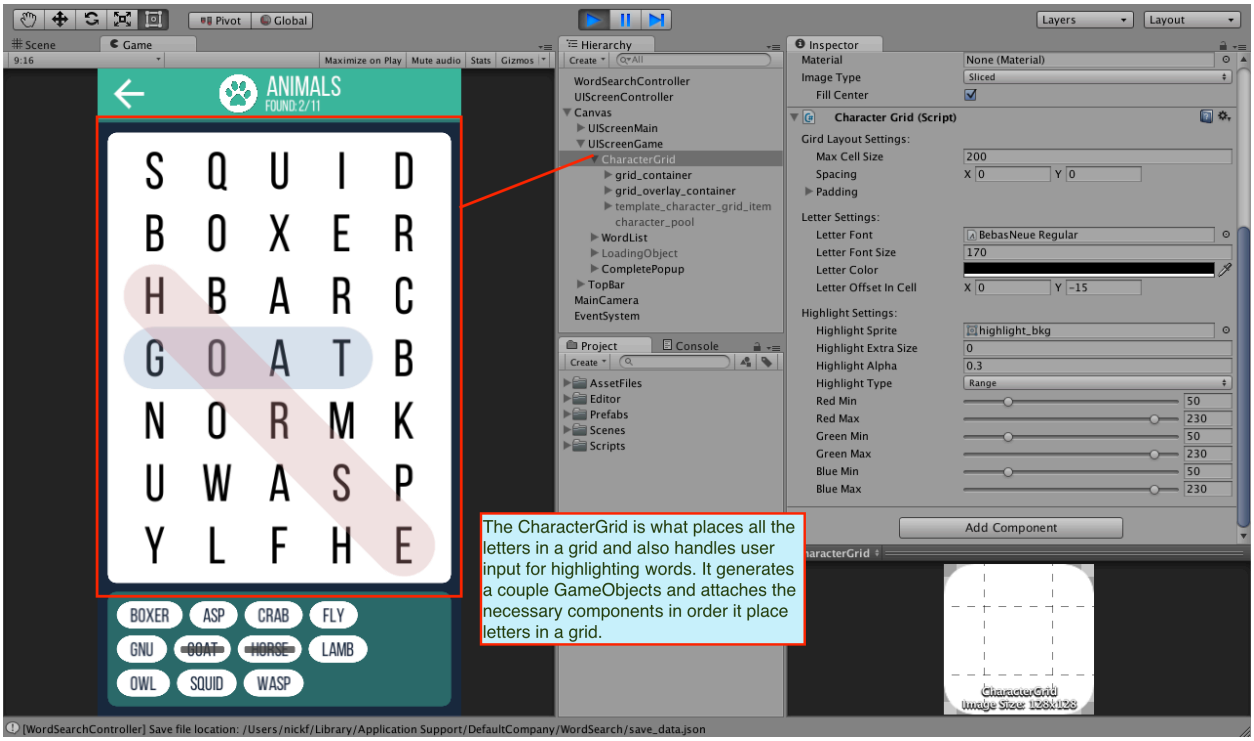
UIScreenMain



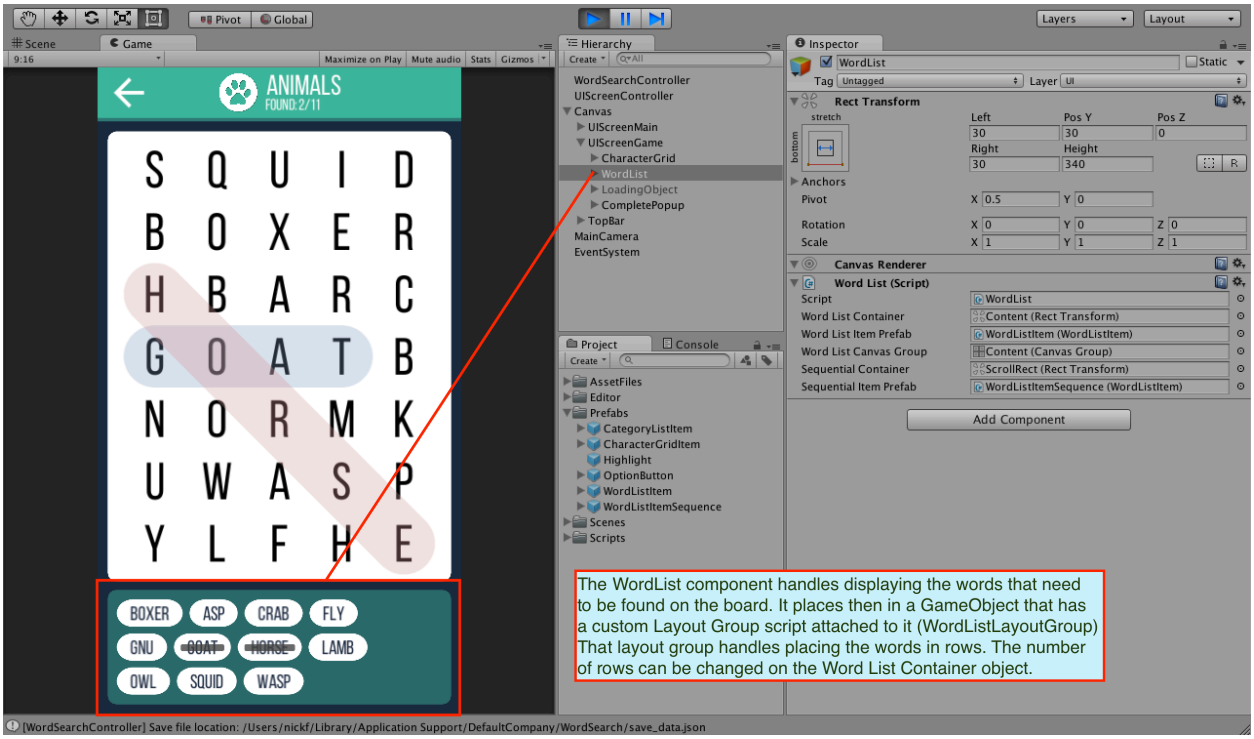
UIScreenGame



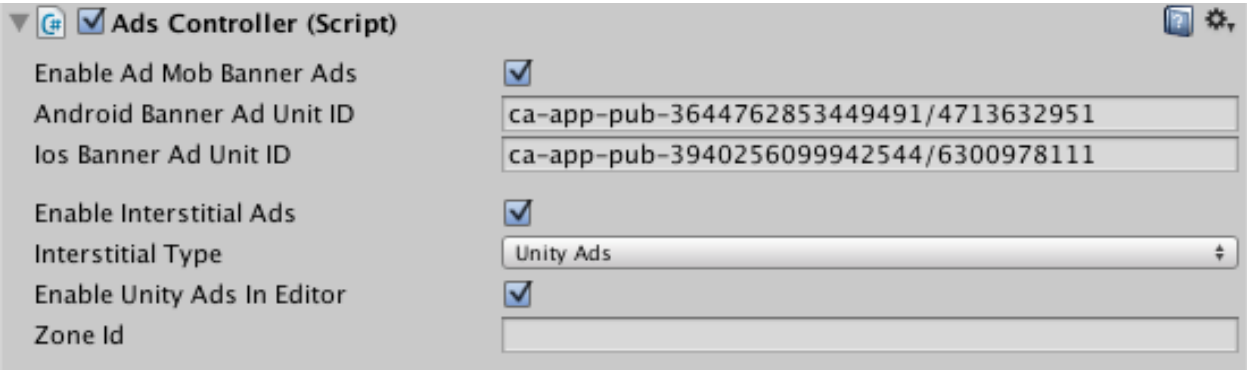
CharacterGrid

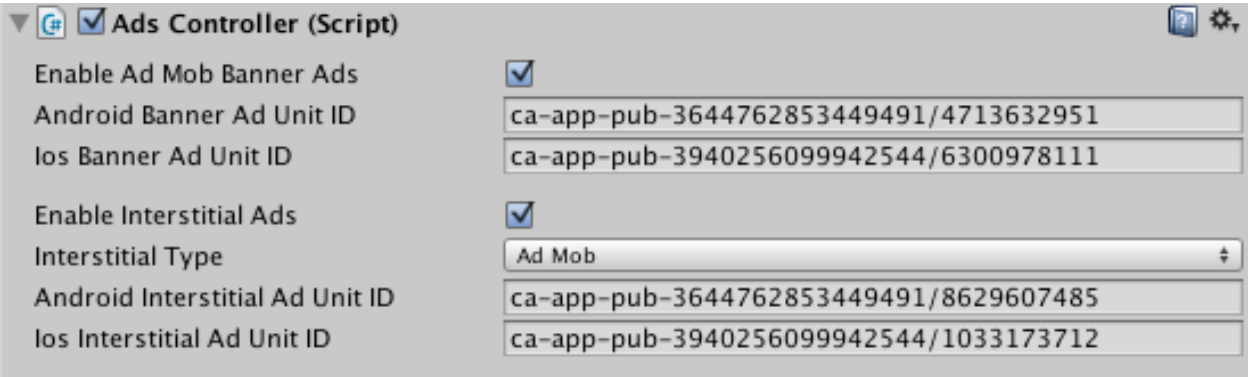


WordList


ANIMALS
FOUND: 2/11
S Q U I D
B O X E R
H B A R C
G O A T B
N O R M K
U W A S P
Y L F H E
BOXER
ASP
CRAB
FLY
GNU
GOAT
HORSE
LAMB
OWL
SQUID
WASP
WordSearchController
UIScreenController
Canvas
UIScreenMain
UIScreenGame
CharacterGrid
WordList
LoadingObject
CompletePopup
TopBar
MainCamera
EventSystem
Rect Transform
Canvas Renderer
Word List (Script)
Word List Container
Word List Item Prefab
Word List Canvas Group
Sequential Container
Sequential Item Prefab
Add Component
The WordList component handles displaying the words that need to be found on the board. It places then in a GameObject that has a custom Layout Group script attached to it (WordListLayoutGroup) That layout group handles placing the words in rows. The number of rows can be changed on the Word List Container object.

Ads



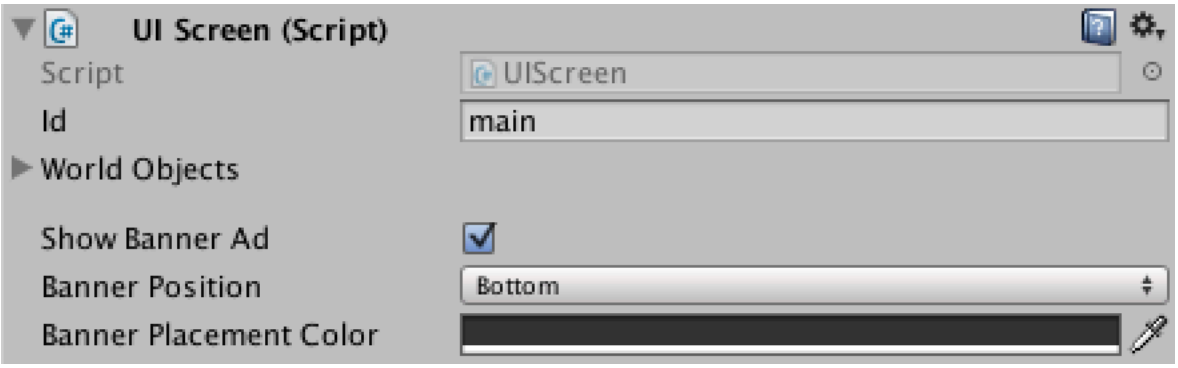


You can enable / disable banner and interstitial ads in the **AdsController** inspector. You can use either Unity Ads or AdMob for interstitial ads. AdMob ads will only appear on device, Unity Ads can be enabled to appear in the Unity Editor for testing purposes.

Interstitial ads will display right after a level is completed. You can set the number of levels that must be completed before a new interstitial ad appears by changing the **Num Level Start Before Ad Shown** field on the **WordSearchController**.

The AdMob Unit IDs that come with the asset are Googles test ids, you will need to replace them with your own if you would like to use AdMob.

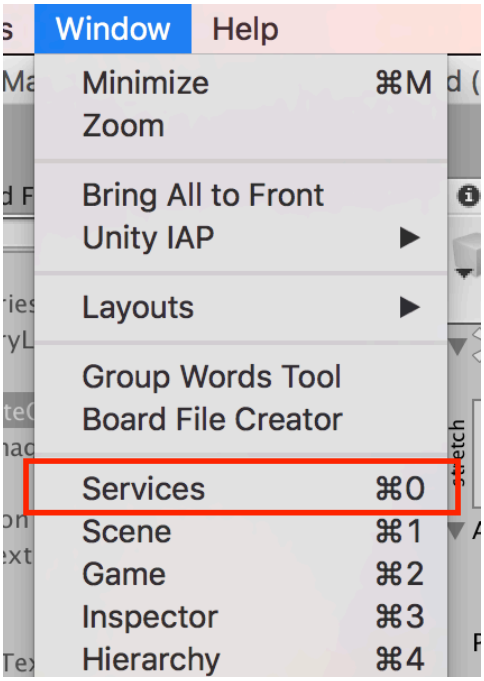
UIScreen / Banner Ads

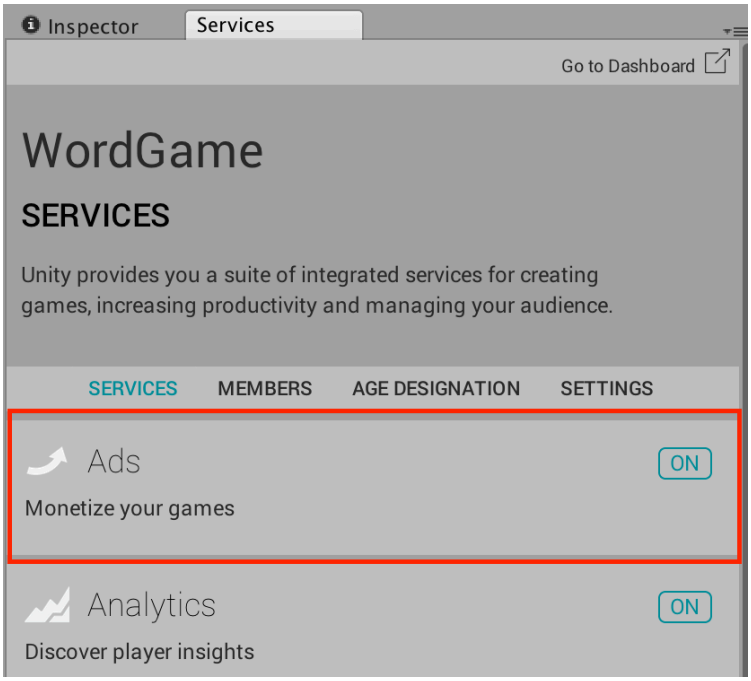


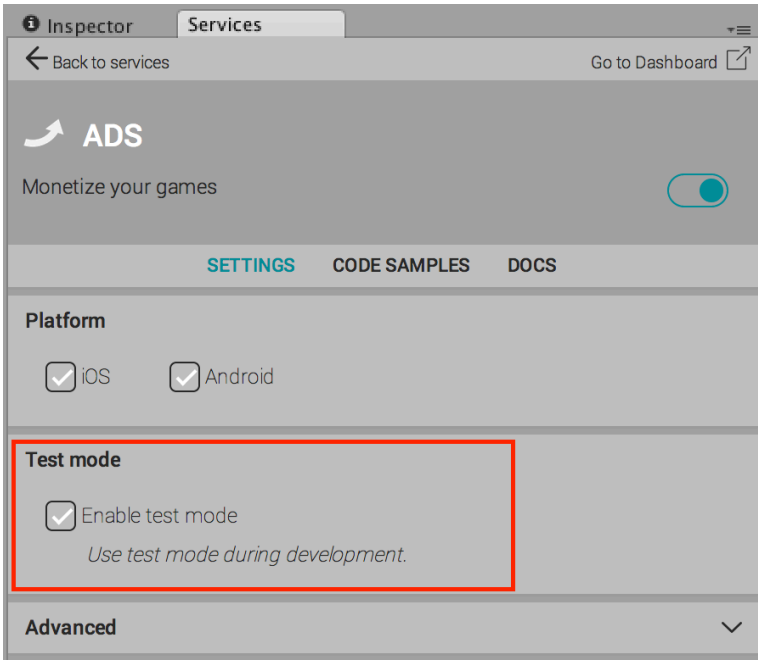
The UIScreen inspector can be used to enable / disable banner ads and also set the position of the banner ads for each screen. If banner ads are enable on a UIScreen then at run time the UIScreen will automatically adjust it's layout to make room for the banner ad so that it does not block any UI.

Unity Ads Setup

In order to enable Unity Ads in the project, navigate to **Window -> Services** and enable **Ads**. When testing Unity ads make sure you click the **Enable test mode** on the Ads Services panel.



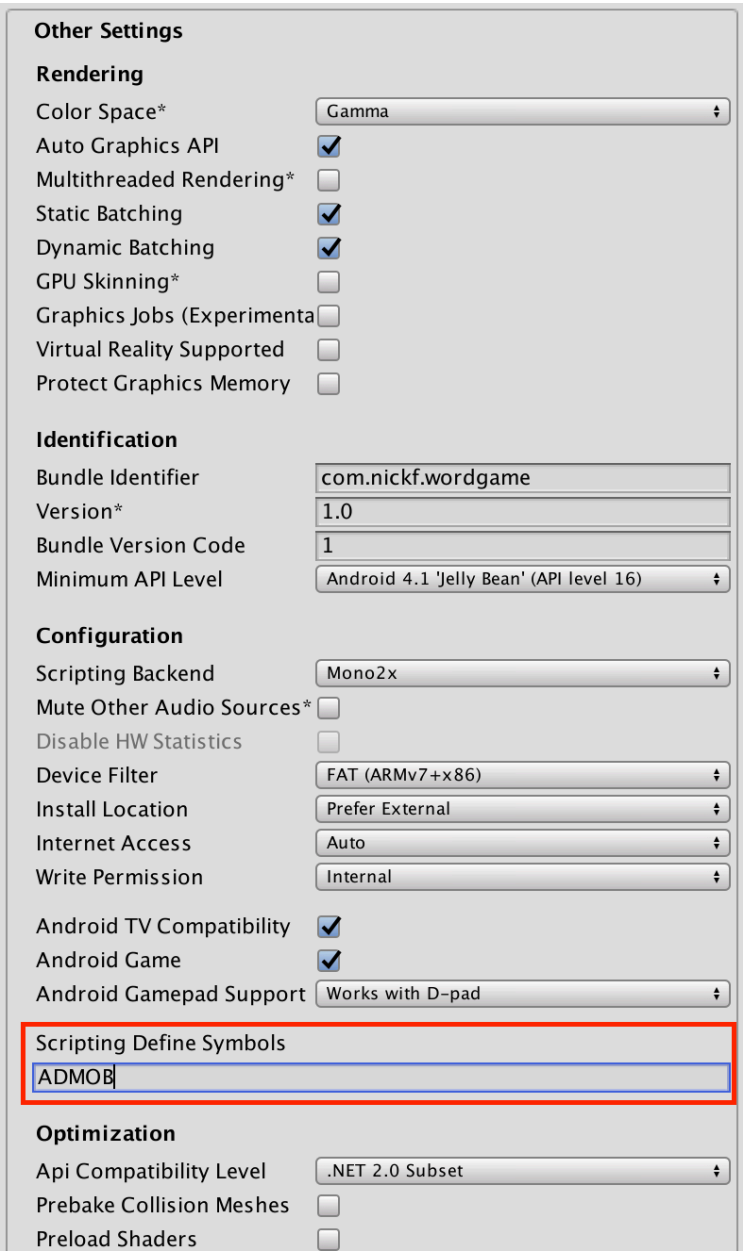




AdMod Setup

In order to enable AdMob first you need to import the AdMob unitypackage located at **Assets/WordGame/AdMob/GoogleMobileAds.unitypackage**. To import it simply double click in.

Next navigate to the **Player Settings** window and under the **Other Settings** add **ADMOB** to the **Scripting Define Symbols**.



NOTE: This setting is not shared between platforms. You need to add it to the Player Settings on both Android and iOS platforms.