

# LevelMenu

---

[Asset Store Link](#)

© 2017 Justin Garza

PLEASE LEAVE A REVIEW OR RATE THE PACKAGE IF YOU FIND IT USEFUL! Enjoy! :)

## Contact

---

Questions, suggestions, help needed?

Contact me at:

Email: [jgarza9788@gmail.com](mailto:jgarza9788@gmail.com)

Cell: 1-818-251-0647

Contact Info: [justingarza.net/contact](http://justingarza.net/contact)

## Description/Features

---

This Asset allows you to easily build a LevelMenu for any level based game, and manage LevelData, such as score, locked/unlocked, and whether or not the level has been won.

- Update the layout of each LevelPage on the fly.
- Vertical and Horizontal scrolling.
- Easily to use API to manage LevelData.
- Animation for showing stars.
- Animation for Unlocking of Levels.

## Terms of Use

---

You are free to add this asset to any game you'd like However:

please put my name in the credits, or in the special thanks section. :)

please do not re-distribute.

## Table of Contents

---

1. LevelButton.cs
2. LevelPage

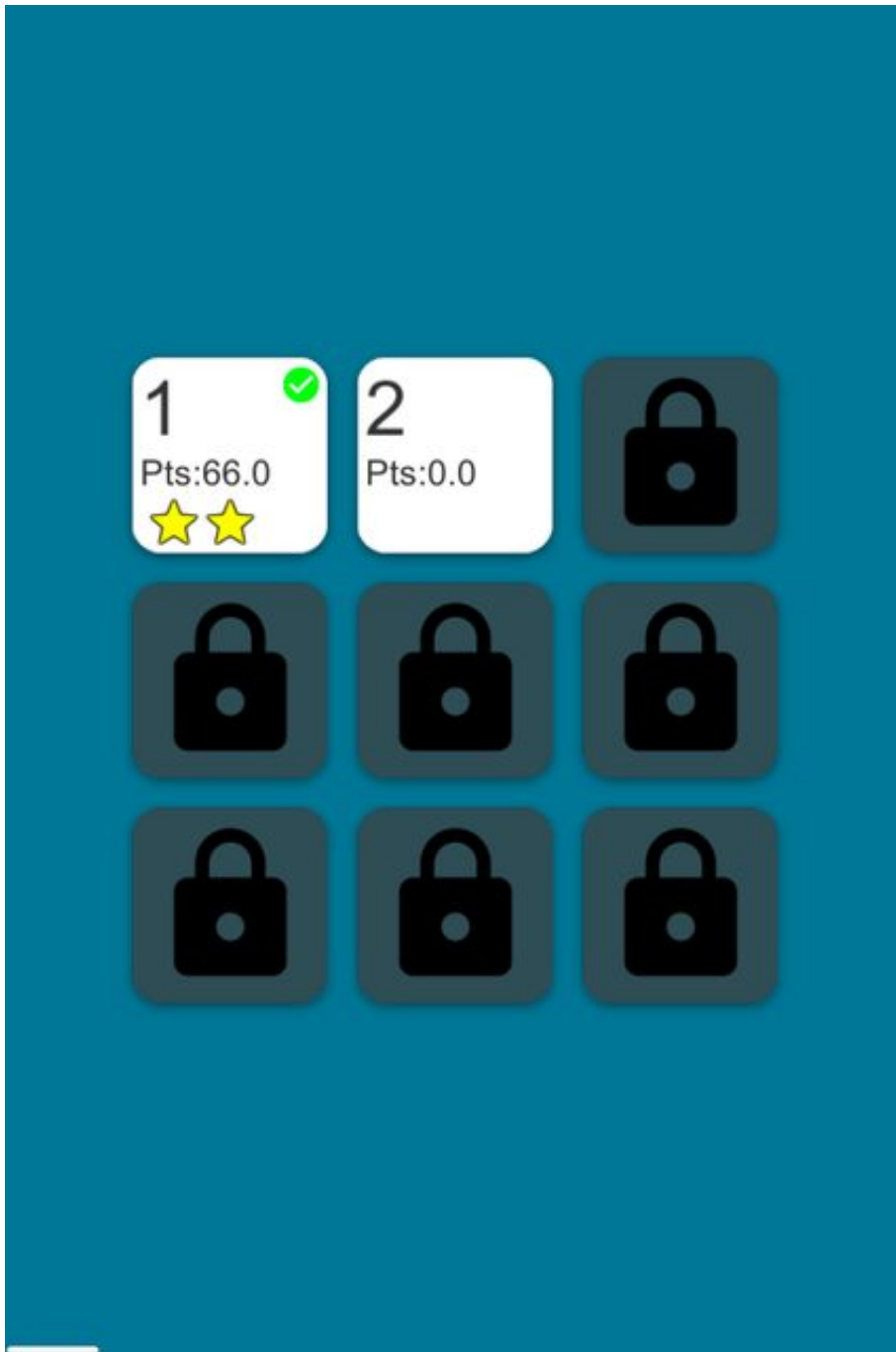
3. LevelMenu.cs
4. LevelData.cs
5. BackgroundParallax.cs
6. LevelPageNum.cs
7. Fader.cs

## LevelButton.cs

---

A LevelButton is a UI Button with a LevelButton.cs and specific children. This Object conveys specific information to the player and allows them to go into a specific level/scene.

The Image below shows 9 LevelButtons in 3 different ways.

**Level1:**

A green checkbox in the corner conveys that the level has been won.

The text “Pts:66.0” shows the score the player got in this level.

The 2 stars show how many stars they got in the level.

**Level2:**

Hasn't been won yet.

And no points, nor stars have been recorded for this level.

### Level3 - 9:

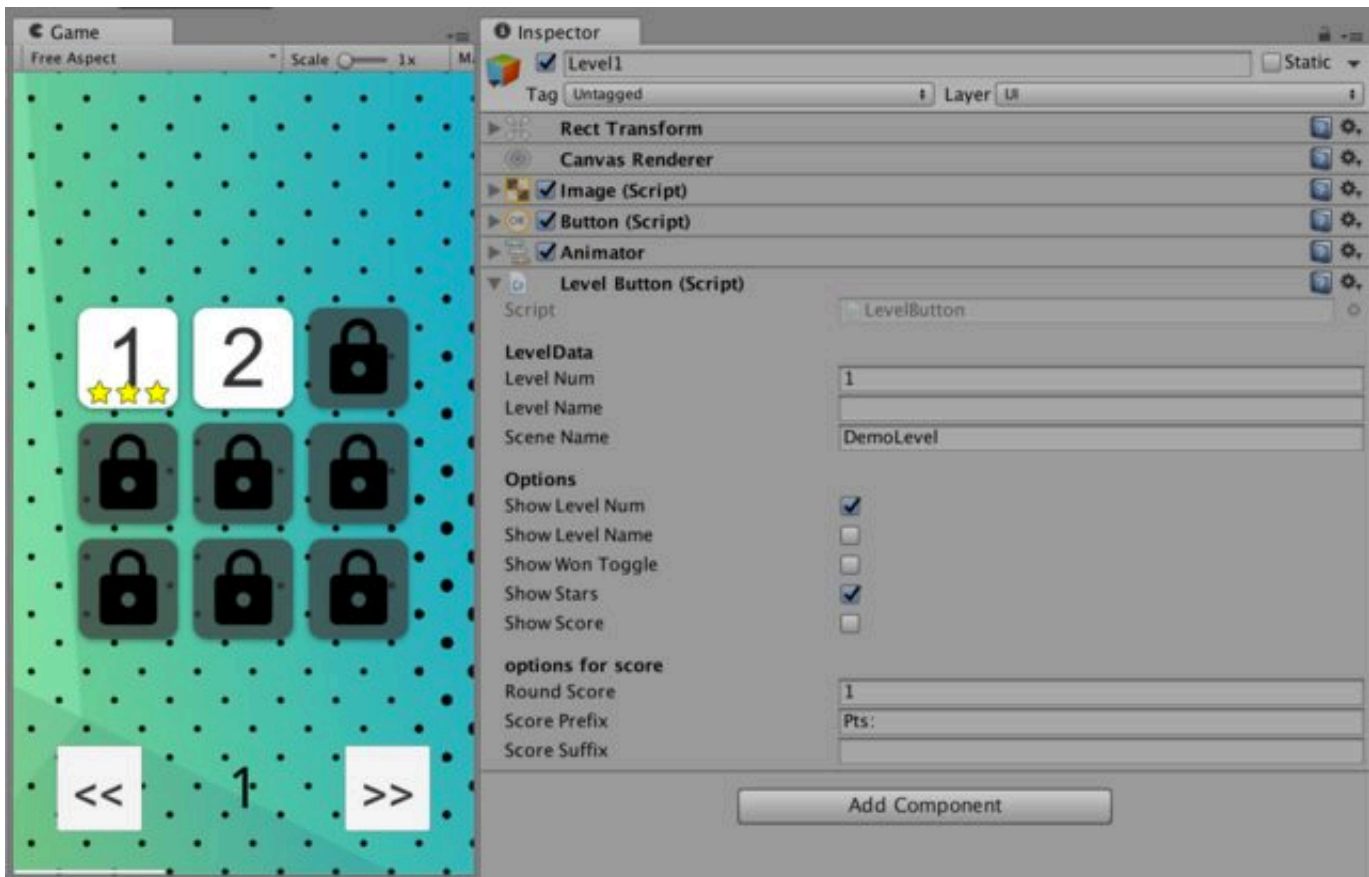
Are currently locked.

### Customizing the LevelButton

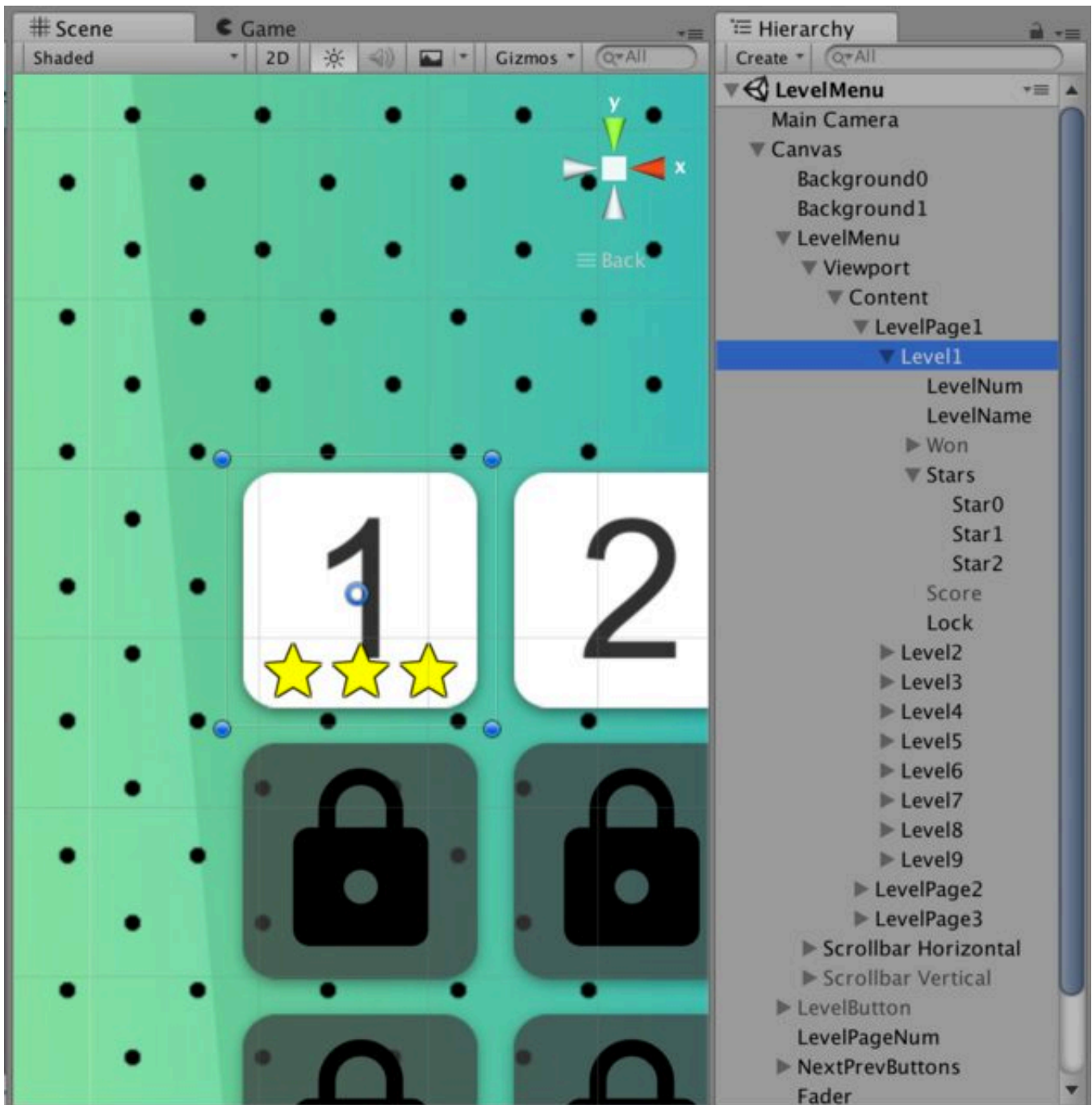
In order to customize the look of the LevelButton you can look at the LevelButton.cs script attached to the object, and/or the LevelButton's Children

### LevelButton.cs

The LevelData fields are filled in by the LevelMenu.cs. But you can change the Options & Options for Score in order to change how the LevelButton will look. Read the LevelButton.cs script to know more.



LevelButton's Children The LevelButton's Children can be changed. as long as they are not renamed, or the components are not removed. so fill free to change the fonts of the Text, resize the Score, or change the Image Source for the stars.



Please note that here are animator components on the Stars, and Lock Objects. Fill free to edit the animations as you see fit.

## LevelPage

a LevelPage is a group of LevelButtons that are displayed together. there is no script attached to these objects, however It's important to know the terminology to understand the next portion of this document.

# LevelMenu.cs

---

The LevelMenu.cs script is attached to the Scroll View Object in the Demo.

It's purpose is to destroy and rebuild the LevelPages with the correct number of LevelButtons, delete stored data about the Levels, and manage the 3 states of the ScrollBars.

## **RebuildLevelPages:**

if true this will delete and re-create the LevelPages based on the current values.

## **AutoRebuildLevelPages:**

if true this will delete and re-create the LevelPages after each update.

## **DeleteLevelData:**

if true this will delete all the LevelData stored for every level.

## **LevelPageConstraint:**

The level page constraint. (Flexible, Fixed Column Count, Fixed Row Count)

## **ConstraintCount:**

The constraint count, for columns or rows (if needed).

## **LevelsPerPage:**

Number of LevelButtons per each LevelPage

## **SelectedLevelPage:**

the current selected LevelPage (Updates when the scrollBar states gets reset to Idle).

## **SelectedLevelPageFloat:**

the current selected LevelPage as a float.

## **LevelButton:**

the Prefab that would be the template for all LevelButtons.

## **ButtonSize:**

The size of the LevelButtons.

## **Spacing:**

The spacing of the LevelButtons.

## **Directions:**

The direction you wish to scroll.

## **CurrentState:**

the current state of the scrollBar.

### **RecoilSpeed:**

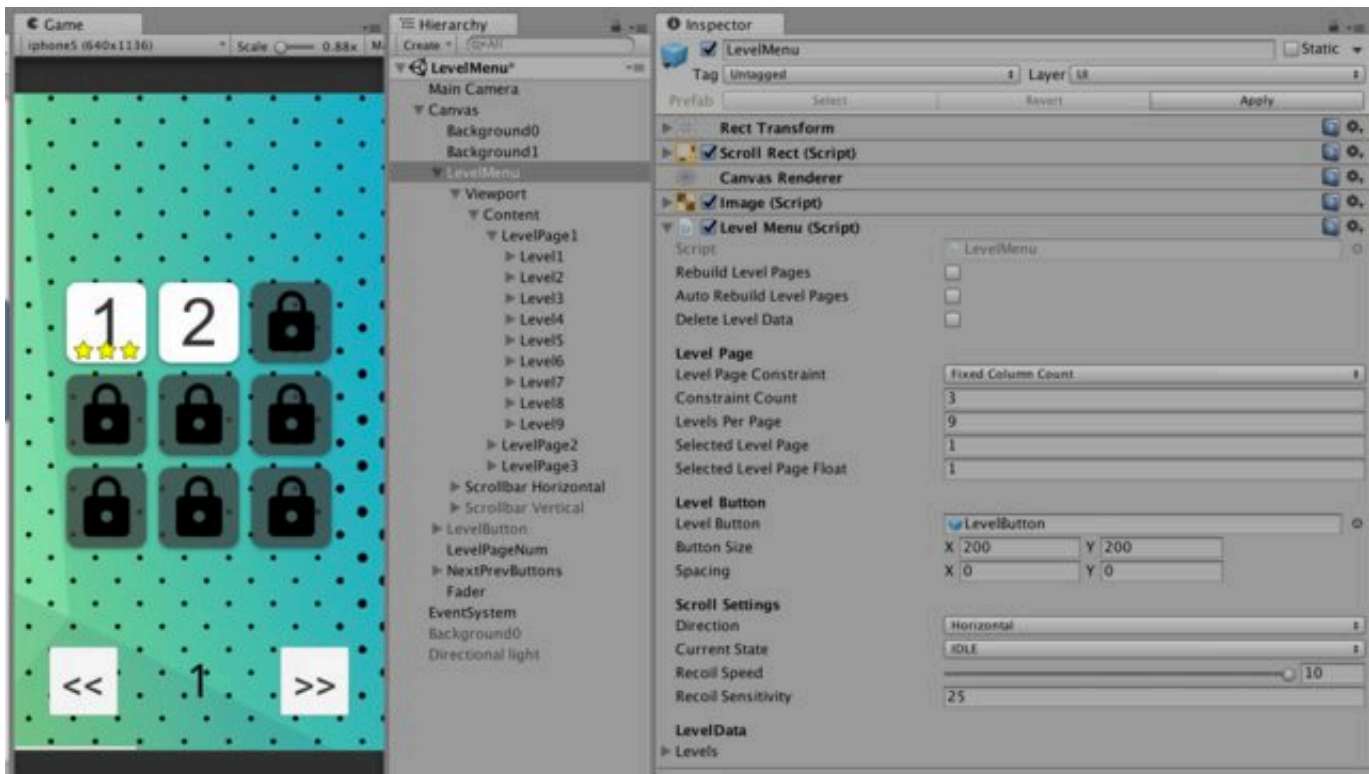
Speed of the scrollBar while Recoiling.

### **RecoilSensitivity:**

How much you will need to scroll for recoil to move to the next LevelPage.

### **Levels:**

An Array that contains all the LevelNums, LevelNames, and SceneNames.



For more information please read [LevelMenu.cs](#).

## **LevelData.cs**

---

This Script contains Methods to easily get/set LevelData for each level.

Methods are listed below:

`getStarCount(int LevelNum)`

`setStarCount(int LevelNum, int StarCount)`

`AnimateStars(int LevelNum, bool animate = true)`

`getAnimateStars(int LevelNum) getScore(int LevelNum)`

setScore(int LevelNum, float score)  
isLocked(int LevelNum)  
setLock(int LevelNum, bool Locked)  
Unlock(int LevelNum)  
Lock(int LevelNum)  
UnlockNextLevel()  
AnimateUnlock(int LevelNum, bool animate = true)  
getAnimateLock(int LevelNum)  
isWon(int LevelNum)  
setWon(int LevelNum, bool Won)  
goToLevel(int LevelNum)  
goToLevelMenu()  
ResetLevelData()  
ResetLevelData(int Upto)  
StarSum()  
ScoreSum()  
WinSum()  
UnlockLevels*StarSum()*  
*UnlockLevelsScoreSum()*  
UnlockLevels\_WinSum()  
getMaxUnlockedLevel()

for more information please read LevelData.cs

## BackgroundParallax.cs

---

This script should be attached to an object that you would like to move based on the value of the scrollbar.

### **pos0:**

The position of this object when the first page is in focus

### **pos1:**

The position of this object when the last page is in focus

### **overPullFactor:**

Determines how much this object will move when scrolling past the first or past the last page. 1 means it doesn't move at all and 1000 means it will move lot.

## LevelPageNum.cs

---



Displays the number according to the current page that is being viewed.

## **Fader.cs**

---

Fades out an image when the scene starts.