Infinite Color Jump
Game documentation and HowTo guide.

INFINITE COLOR JUMP

This document contains:

Package Description and features

Infinite Color Jump is an action packed game full of challenge and fun. The game is ready to release straight out of the box, and it can also be easily customized to make it even more engaging to your players. The game supports PC/Mac, iOS, Android. It can be played with the mouse, keyboard, gamepad, or touch controls!

How to Play?

Click to jump up. Pass through blocks of the same color as you.

Features:
- Game ready for release straight out of the box, just build and play!
- Works on all platforms, PC, Mac, iOS, Android, etc
- Supports multiple resolutions and aspect ratios, automatically.
- Supports Mouse, Keyboard, Gamepad, and Touch controls.
- Easily customizable with lots of options to control game difficulty.
- Great learning resource with commented scripts and documentation.
- All assets included: graphics, sounds, and code.

Current version 1.23

Update history

1.23 (03.11.2017)
- Support for Unity 5.5, 5.6, and 2017.

1.22 (10.07.2016)
- Support for versions of Unity higher than 5.3.
- Better UnityAds support for Unity 5.2 and above.

1.2 (13.03.2016)
- Completely replaced the graphics of the game with something less generic and more thematic. The game is now called Infinite Color Jump.

1.1 (19.02.2016)
- Added a dynamic character shop. You can add any number of new characters to the shop and it will automatically create the shop buttons and accommodate the grid to fit all the characters you have.
- Minor fixes and changes.

1.0 (30.01.2016)
- Initial version

Credits

The sounds are courtesy of <u>the free sound project</u>.

Music is Horns by Kevin MacLeod (Public Domain)

Credits go to these authors for their great sound samples: **titaniumturner, panikko, Oddworld, fins, boulderbuff64, Isaac200000, Harris85**

Please rate my file, I'd appreciate it 🙂

Overview of the game's library contents

Let's take a look inside the game files. Open the main CSGAssets folder using Unity3D 4.6.9 or newer. Take a look at the project library, usually placed on the right or bottom side of the screen. Here are the various folders inside:

- **Animations:** Holds the animation clips made with Unity's built-in animation system.
- **FLA:** Holds the object graphics made with Flash CS3. These are vector graphics than can be easily scaled without loss of quality and then exported as PNG to be used in Unity.
- **Fonts:** Holds the font used in the game.
- **Prefabs:** Holds all the prefabs used in the game. These are distributed to various folders for easier access, Buttons, Enemies, Objects, etc. It also holds all the canvases in the game which are used to hold buttons and other UI elements.
- **Scenes:** The first scene that runs in the game is MainMenu. From this scene you can get to the Game scene.
- **Scripts:** Holds all the scripts used in the game. Each prefab contains one or more of these scripts.
- **Sounds:** Holds all the sounds used in the game. Jump,Item, etc
- **Textures:** Holds all the textures used in the game which are used as sprites in Unity.

Customization Guide

Getting started

Infinite Color Jump Template (CSG) is considered a complete project, and as such is supposed to work as the starting point of your planned game, rather than an addition to an existing project. That said, you may of course pick and choose some of the scripts/models to import into your existing project, but CSG works best as a starter kit which you can customize any part of to your liking.

The Game Controller

The Game Controller is the main prefab that controls all the progress of the game from start to finish. It controls the UI of the game, creates enemies and items and checks the level up condition.
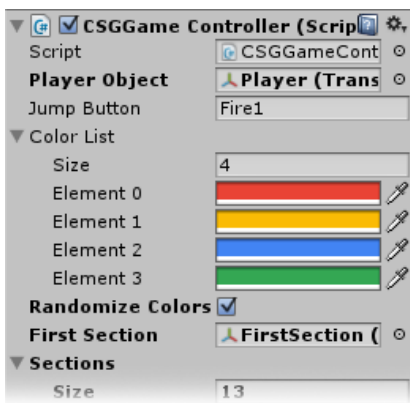
Player Object – This is the player that you control. It must be assigned from the level.

Jump Button – The button that makes the player jump. This is defined from the Input Manager, and corresponds to the gamepad, keyboard, mouse, and touch.

Color List – A list of all the colors in the game. This affects the blocks, color balls, and the player. Read more about colors here.
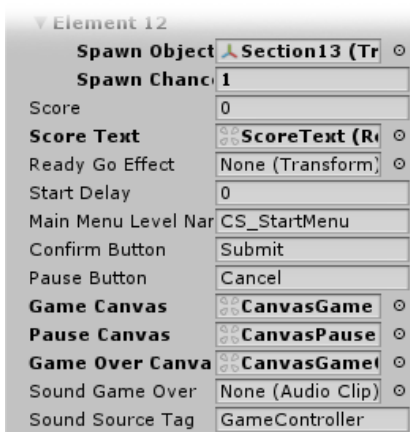
Randomize Colors - Randomize the list of the colors at the start of the game. This will make your game a bit more varied

First Section – The first section in the game. This is assigned from the scene, and all sections are created after it.

Sections – A list of sections in the game, which are spawned based on their spawn chance.

Score – The score of the game. Score is earned by shooting enemies and getting streaks.

Score Text – The text object that displays the score, assigned from the scene.

ReadyGoEffect – The effect displayed before starting the game.

Start Delay – How many seconds to wait before the player control starts.

Main Menu Level Name – The level of the main menu that can be loaded after the game ends.

Confirm Button – The keyboard/gamepad button that will restart the game after game over.

Pause Button – The keyboard/gamepad button that pauses the game.

**User Interface –** Various canvases for the UI, assign them from the scene.

Sounds – Various sounds that play during the game.

Sound Source Tag – The audio source from which the Game Over sound plays.

How Colors Work

The game is designed in a way that allows you to setup the list of colors in the Game Controller object, and then refer to those colors through the various elements in the game: Blocks, Color Balls, and the Player.

By default the Game Controller has 4 colors defined in it, like this:
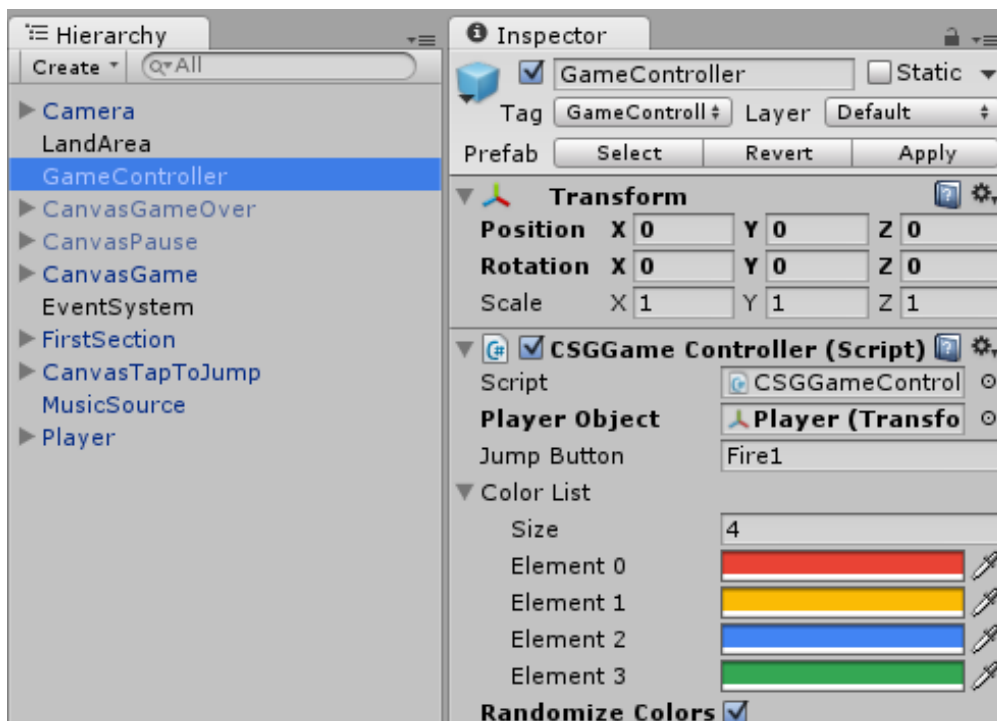
Each of these colors has an element number in the list, so the first color (red) is indexed at 0, while the second color (yellow) is indexed at 1, and so on. You can change these colors and the entire game will change accordingly.

The Player

The player object can jump and move through blocks of the same color, but will die if it touches a block with a different color. The current color of the player is set through the value of **Color Index**, and it refers directly to the list of colors in the gamecontroller.
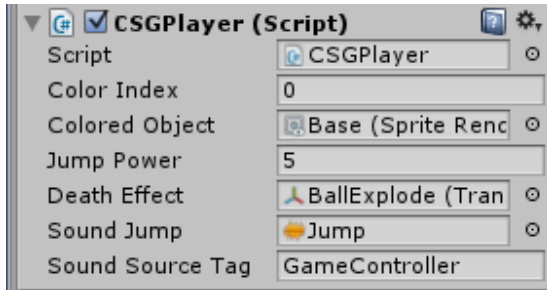
Color Index – The index of the color of this object. It corresponds to the color list defined in the gamecontroller (0 is the first color, 1 is the seconds color, etc). [Read more about it here.](#)

Colored Object – The part of the player that will get colored.

Jump Power – How high this player can jump.

Death Effect – The effect created at the position of this player when it dies.

Sound Jump – The sound that plays when the player jumps.

**Sound Jump –** The source from which this sound plays.

Now you have a new item right at the bottom of the shop, which you can try to unlock with money, if you have enough. When you pick the item and start the game, the player graphic will be replaced with the icon of the shop item.

Editing Sections

The fastest way to add a new section to the game is to duplicate an existing one and work from there.

For our example let's say we want to make a new section similar to Section12. To duplicate a prefab in the project pane select it and then press Ctrl-D. Now we have a new Section13 object, drag it to the game area so we can easily edit it.

Notice a few things about the section: It has a thin red horizontal line which indicates where the next section will be created. You can control this by changing the value of Section Gap in the section component.
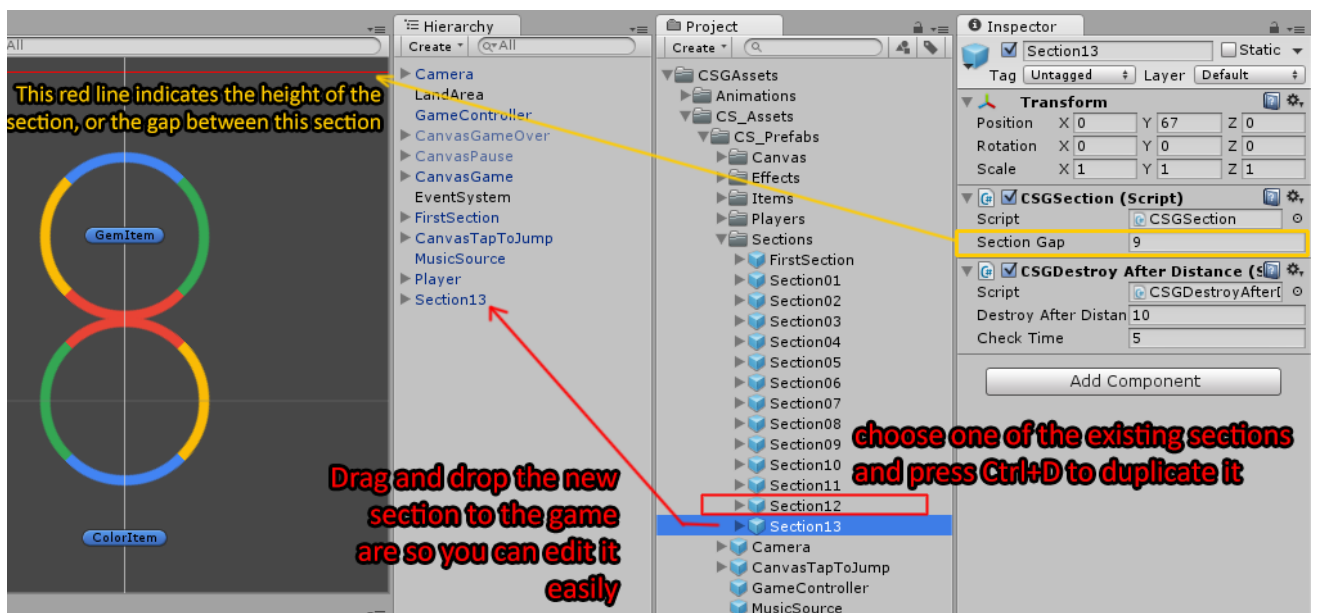

Another thing you'll notice is the GetItem and ColorItem. These are objects that are spawned in the section when it is created. The gem item is the one we pick up and gain score, and the color item is the one we pick up which changes our color. There are several types of color balls that change the color of the player differently (2 colors, 4 colors, etc).
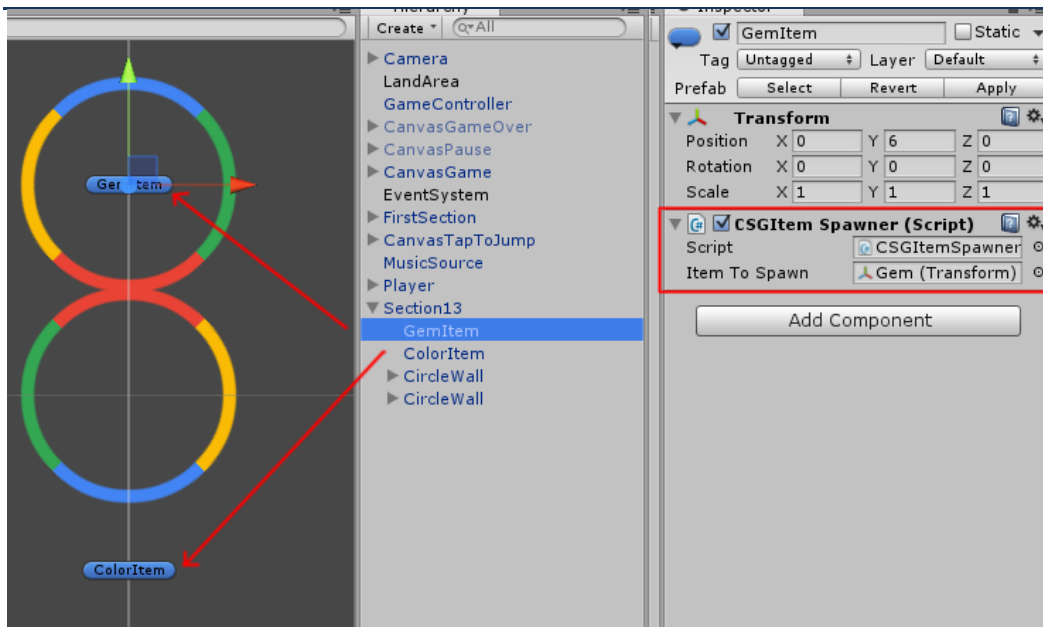
Now take a look at the CircleWall object, which has a MovingObject component attached to it. This makes the whole circle move, in our case rotate in a pendulum motion. You can always add waypoints to the moving object to make it move from one point to another.
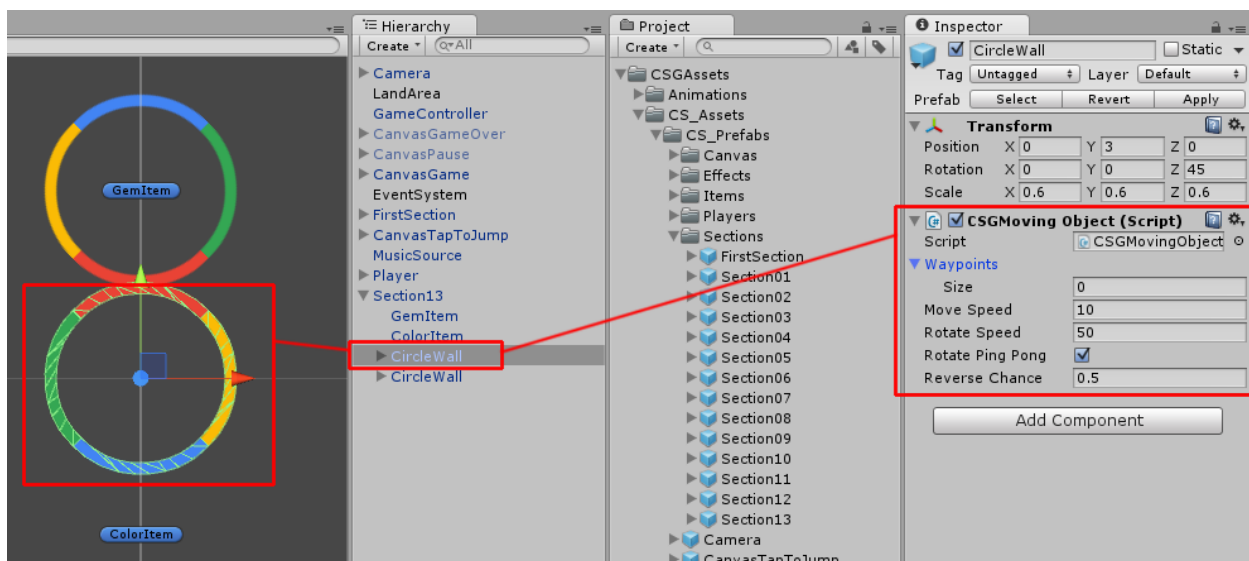

Inside the CircleWall there are several block objects that are colored and can kill the player if he touches them when he has a different color. The color of a block is set using an index which corresponds to the color list in the gamecontroller. So for example if the block is set to index 0 it will have the first color in the gamecontroller list, and if set to 1 it will have the second color. This allows you to change the entire color scheme of the game from the gamecontroller alone.
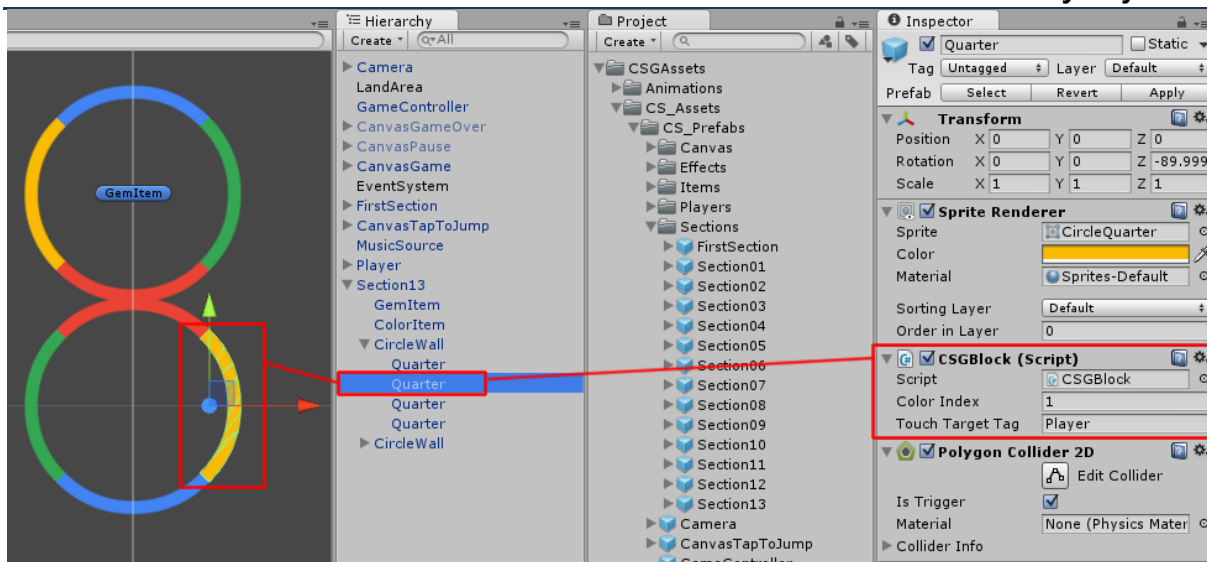
These are the parts making up a section. You can change them up and mix and match different moving parts with different blocks to make new sections. When we are done editing a section we should save the changes by pressing **Apply** in the top right part of the object inspector. Now delete the edited object from the scene as we don't need it anymore.

To add Section13 to the list of sections in the gamecontroller, increase the sections list size, and then drag&drop Section13 to the new slot. Here you can also set the appearance chance of this section relative to the other sections.
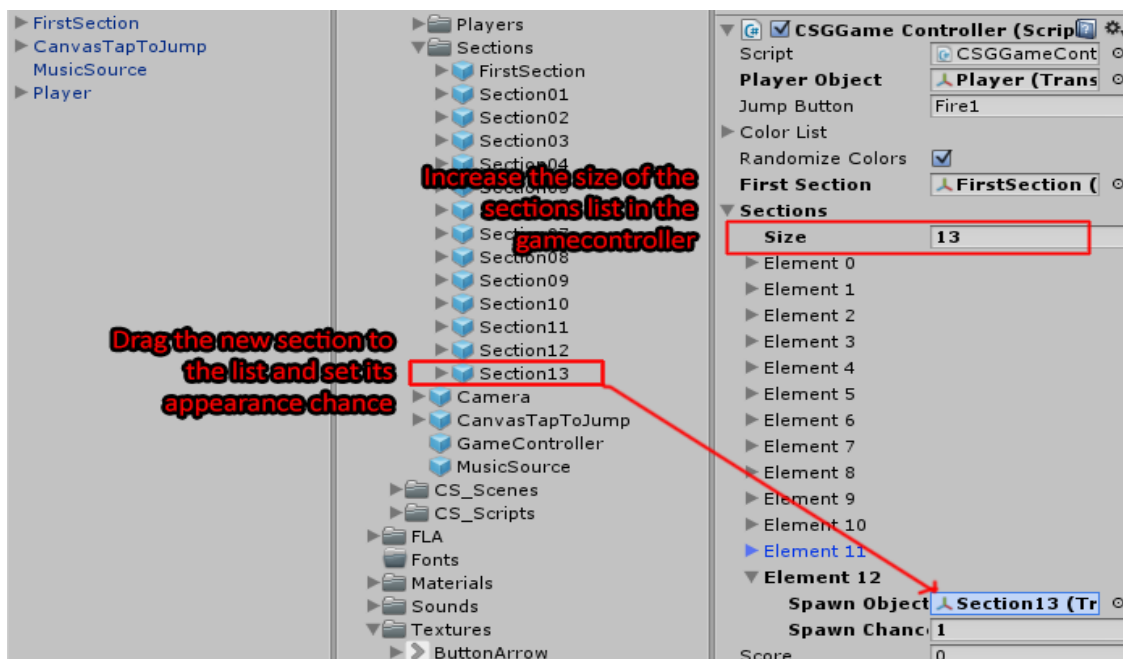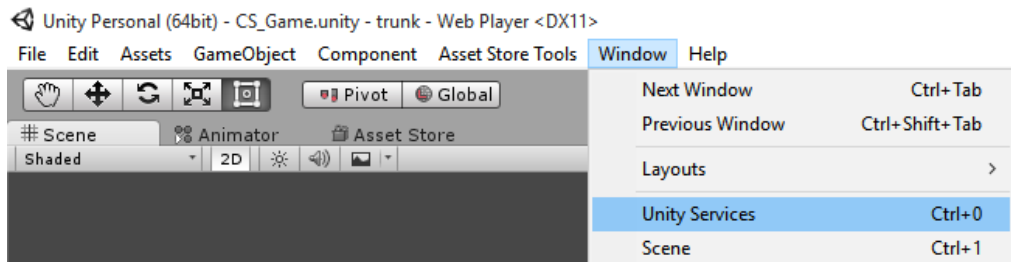
UnityAds Integration

Since Unity 5.2 UnityAds integration has been simplified, here's how you can have fullscreen video ads in your game:
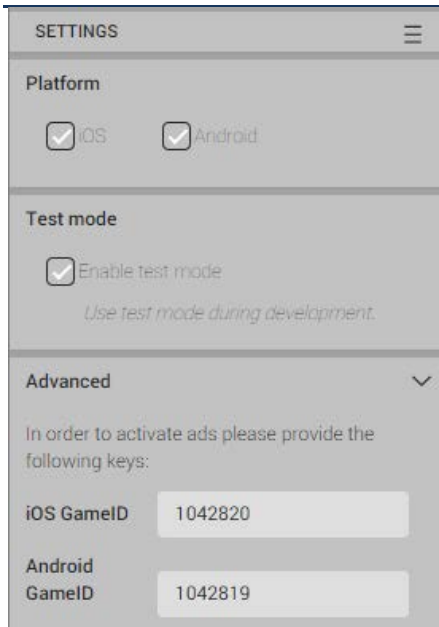
Activating UnityAds service

From the top menu choose Window > Unity Services.

In the new window that appears, choose your organization so that an ID will be generated. A list of services appears, choose the Ads service and turn it on. Then choose the age group this game will be targeted towards, and save the changes.
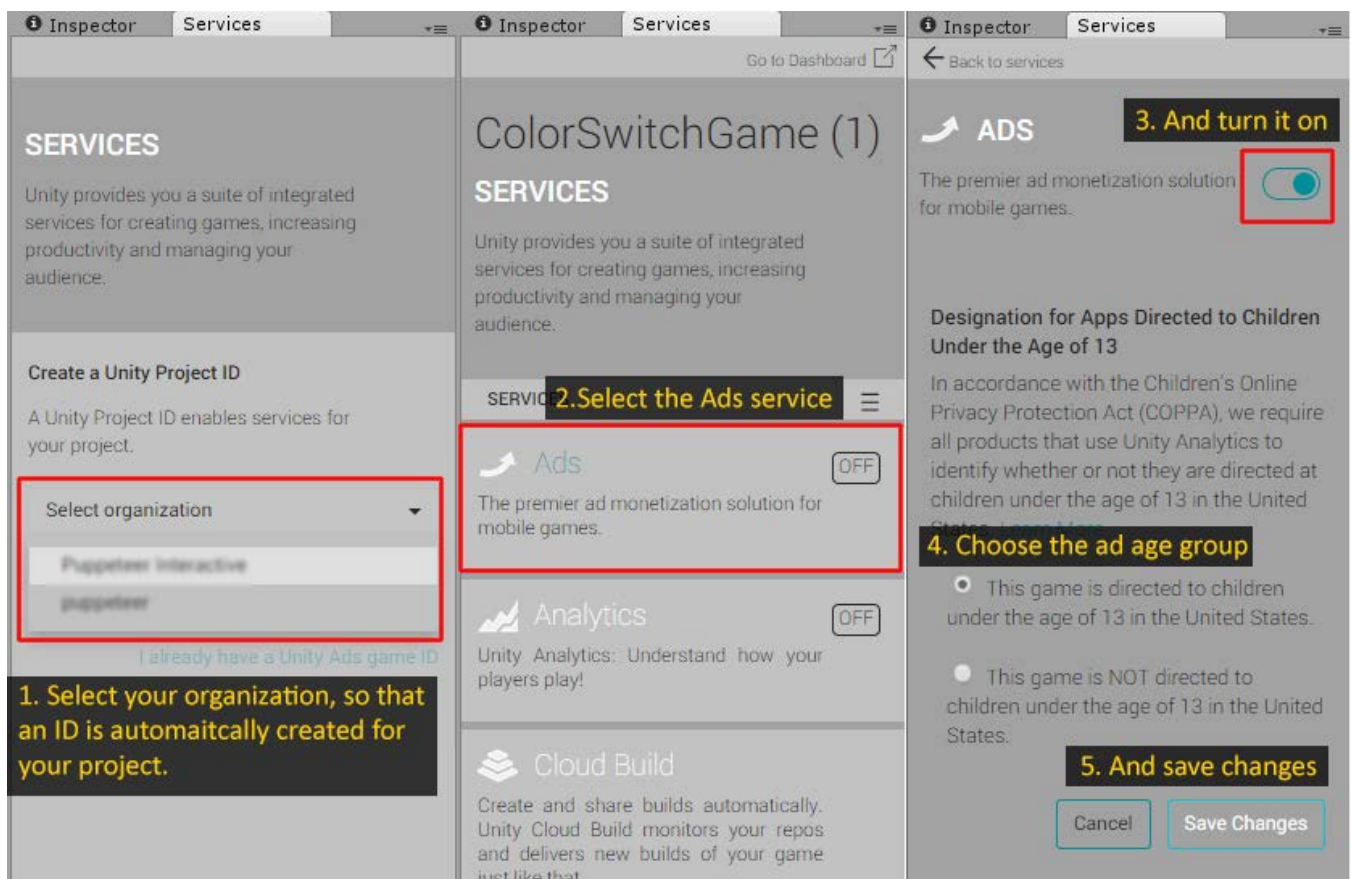
Once you save the changes, a new screen appears. Here you can set which platform the ads are targeted to, Android or iOS.

It's advised that you Enable Test Mode while working on your game to avoid any potential problems.
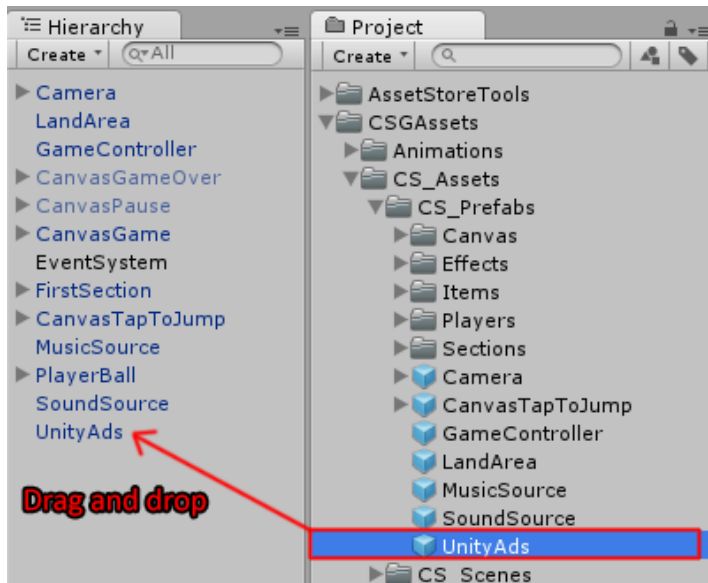
At the bottom of the screen you can see the IDs for the iOS platform and the Android platform.

Showing ads in your game

Before testing out the ads, make sure you are in a compatible platform in the editor, such as Android or iOS.
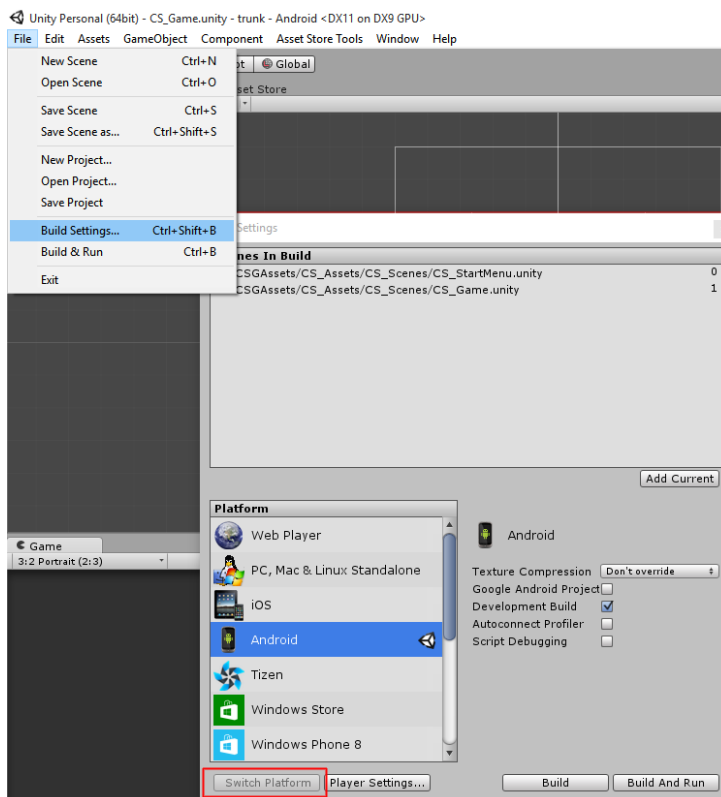
Drag the UnityAds prefab from your project to the game area in the level you want to show ads in.

Click on the UnityAds object in the level, here you can set the number of times the level needs to be loaded before showing an ad. By default it loads 3 times before showing an ad.
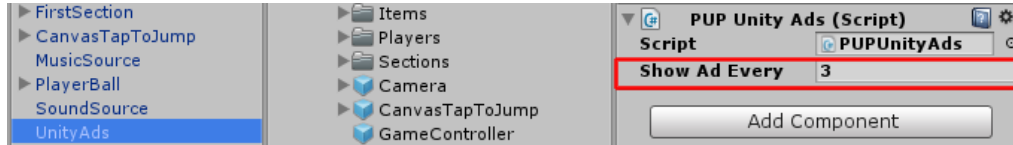
That's all, if you test the game now in the editor and restart the level several times you will be presented with a blue test ad screen.

In the actual game released on Android/iOS the game would pause and present you with a full screen video advertisement.

Before releasing a game, make sure you uncheck **Enable Test Mode.**

For more info about integrating UnityAds read this:

http://unityads.unity3d.com/help/monetization/integration-guide-unity

Does this package work on mobile?

Yes, this package has been successfully tested on both Android and iOS devices. The scripts for each lock type include controls for mobile that are detected automatically based on the platform it's built on.

My sprites are not showing on iOS

Sprite-based textures made with the new Unity 4.3 can sometimes disappear when working on the iOS platform.

You can notice this by opening a scene playing it. When you switch from your current platform to the iOS platform the sprite textures become invisible.

To solve this we must change the texture compression format for iOS. Follow these steps:

1. Click on a texture in the project view.

2. Click on the override for CSGone button on the right side.
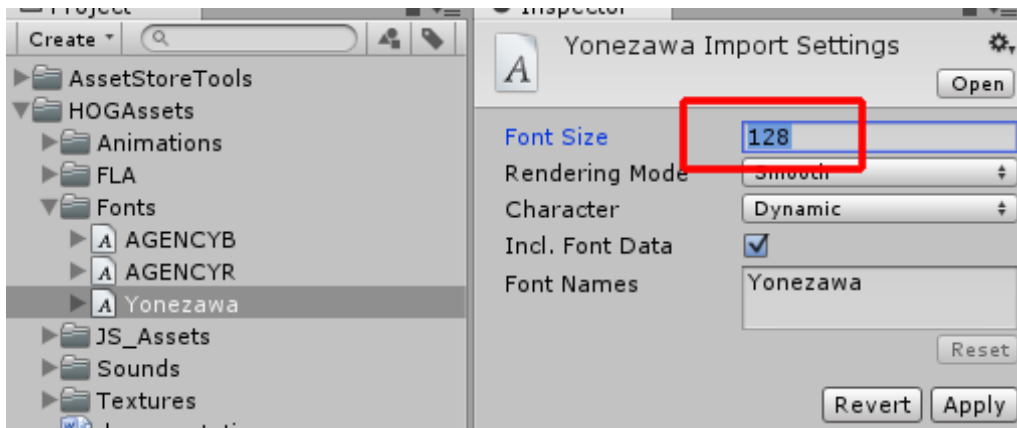
3. Change the format to 16bit.

4. Click Apply.

How to change font in the game?

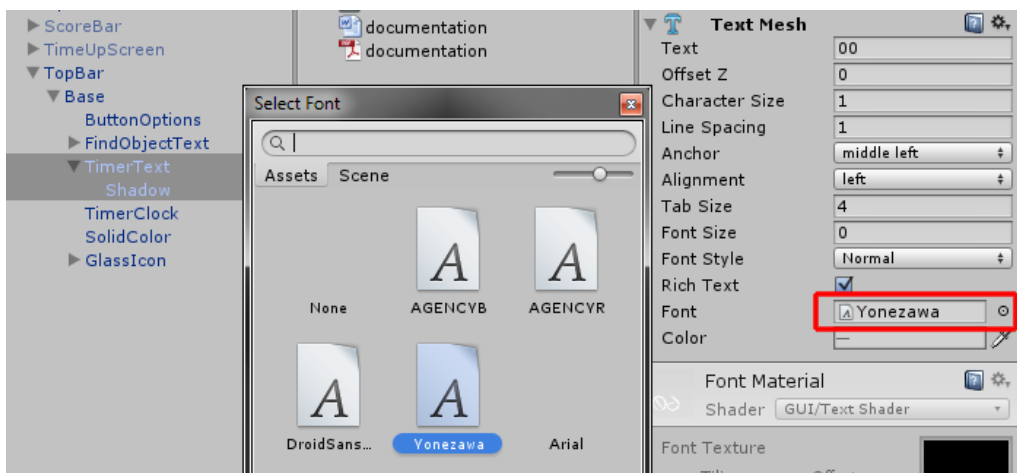To change a font in the game do the following:

Find a font you like and drag the .ttf file over to the Fonts folder in your game.
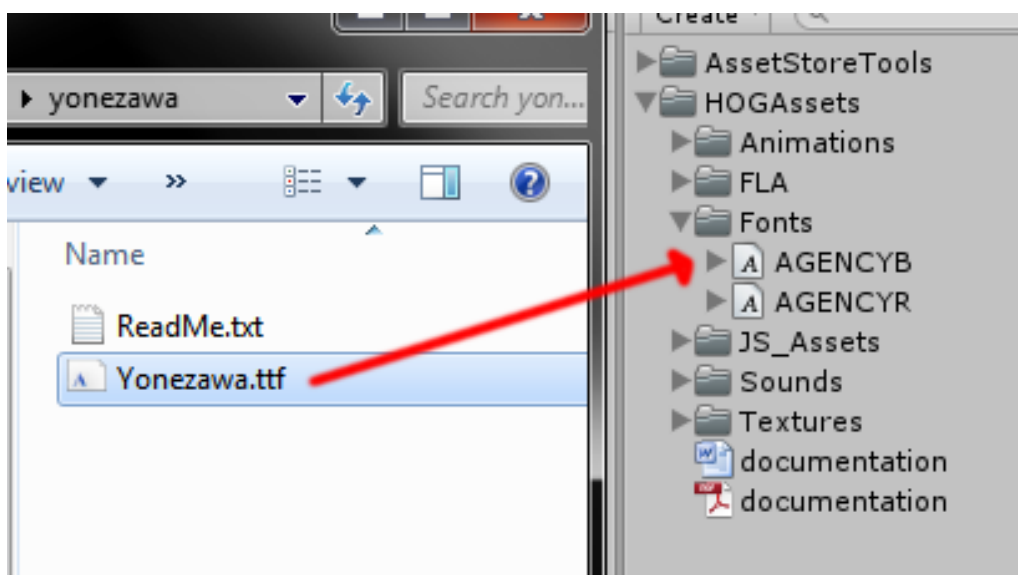
Click on the font you added and edit its attributes. I personally set all my fonts to a high number (and then scale the text object down) so that they look crisper in-game.

Select any text object in the game and change its font to the new font you have. Sometimes the text might disappear, but it's normal. Just write something in the text box above and it will refresh. Also, make sure you change the text for the shadow; you can select both the main text and its shadow and edit them together.

Click here to see the full catalogue of Asset Store files!

It is highly advised, whether you are a designer or a developer to look further into the code and customize it to your pleasing. See what can be improved upon or changed to make this file work better and faster. Don't hesitate to send me suggestions and feedback to puppeteerint@gmail.com

Follow me on twitter for updates and freebies!

Good luck with your modifications!