Object Smashing Game Template

Game documentation and HowTo guide.

This document contains:

Package Description and features

Object Smashing Game 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, etc. It can be played with the mouse or touch controls.

How to Play?

Click the moving objects to smash them, and don't let them go off screen or you'll lose a life. Also beware the bomb!

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 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 1.0

Update 1.01 (01.11.2017)

- Update for Unity 5.5, 5.6, and 2017.

Credits

The sounds are courtesy of the free sound project.

Music is Skipping in the No Standing Zone (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 OSGAssets folder using Unity3D 5.3.0 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:

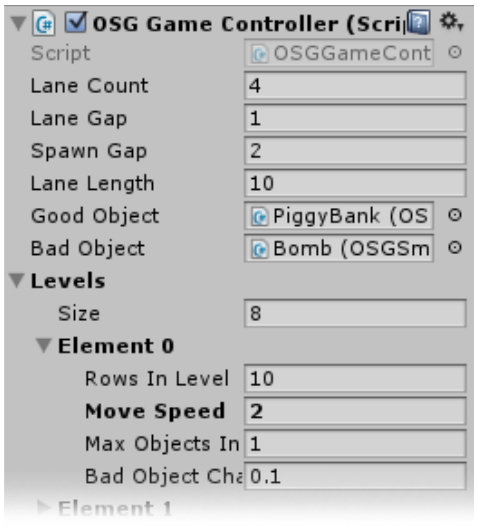
- **Animations:** Holds the animation clips made with Unity's built-in animation system.
- **FLA:** Holds the object graphics made with Flash CS3. These are vector graphics than can be easily scaled without loss of quality and then exported as PNG to be used in Unity.
- **Fonts:** Holds the font used 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. Hit,Smack, etc
- **Textures:** Holds all the textures used in the game which are used as sprites in Unity.

Getting started

Object Smashing Game Template (OSG) 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 OSG 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.



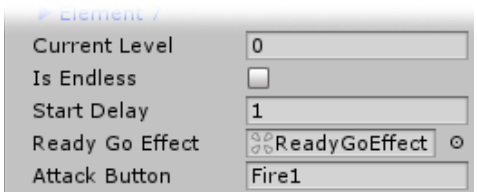
Lane Count – The number of lanes in the stage. Each lane can hold an object.

Lane Gap – The horizontal distance between each two lanes.

Spawn Gap – The vertical distance between each two rows of objects

Lane Length – The path of the object along a lane. The object starts moving from the top of the path and disappears when reaching the bottom of the path.

Good Object – The good object which you smash and get points for.



Bad Object – The bad object which you must not smash or you'll lose a life.

Levels – A list of levels that you advance through in the stage. In each level you can set the number of rows to create, the movement speed of objects in the lanes, the maximum allowed number of objects in a single row, and the chance for a bad object to appear.

Current Level – The current level we are on. 0 is the first level, 1 is the second.
Is Endless – If set to true the game will not end when you reach the last level.
Show Delay – How long to wait before showing the targets.
Ready Go Effect – An effect that appears at the start of the stage.

Attack Button - The attack button, click it or tap it to smash objects.

Hit Effect – The effect that appears when we hit an object.

Bonus Effect - The bonus effect that shows how much bonus we got when we hit a target.

Score - The score of the player.

Lives Object – The heart-shaped lives object that shows you how many lives we have left.

Lives – The number of lives the player has.

Lose Life Delay – A delay to prevent the player from losing many lives together. If you lose a life, you will not lose another life for some time.

Score Text - The score text object which displays the current score of the player.

Level Up Messages - A list of messages that appear randomly when you level up.

Canvases – These UI screens are assigned from the scene for each level.

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.

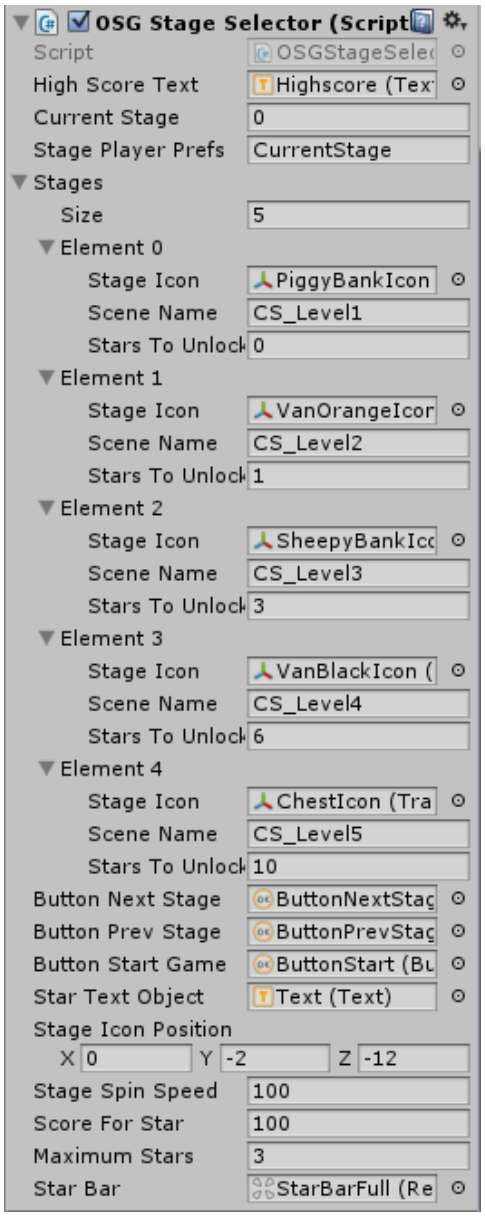
User Interface – Various canvases for the UI, assign them from the scene.

Sounds – Various sounds that play during the game.

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

The Stage Selector

The stage selector allows you to have several stages in the game, each with its own smashable objects and background. Each level must be unlocked by earning a certain number of stars. Each star can be earned when getting a certain score in a stage. You can customize all of these settings.



High Score Text – The text object that shows the high score we got in the current stage.

Current Stage – The current stage we are in. 0 is the first stage, 1 the second.

Stage Player Prefs – This is the name of the record that holds the current stage we are in. This is saved locally.

Stages – A list of stages you can unlock and play.

Stage Icon – The 3D model that appears and spins when we select a stage.

Scene Name – The scene that is loaded when we start this stage.

Stars To Unlock – The number of stars needed in order to unlock this level.

Button Next/Prev/Start – These buttons assigned from the scene let us switch stages and start the stage.

Start Text Object – The text inside the start button which also shows how many stars we need to unlock a level.

Stage Icon Position – The position where the 3D spinning icon is placed.

Stage Spin Speed – How fast the 3D icon spins.

Score For Star – The number of points needed in a level in order to get a star. This is calculated based on the high score we got in the level, so for example if we have 100 highscore we get 1 star, and for 200 highscore we get 2 stars.

Maximum Stars – The maximum stars we can get in a stage.

UnityAds Integration

Since Unity 5.2 UnityAds integration has been simplified, here's how you can have full screen video ads in your game.

This video shows a quick process of integrating UnityAds into your project. In the example we used one of my templates, but it works on all my other templates too.

https://www.youtube.com/watch?v=EQNTgfV35DU

Here is what we did in the process:

1. Sign in to your Unity account in order to allow Unity Services such as UnityAds to be activated.
2. Open Build Settings and switch the platform to one of the supported ones (iOS, Android).
3. Download Puppeteer's UnityAds package from: puppeteerinteractive.com/freebies/PUPUnityAds.unitypackage
4. Drag the downloaded package into your Unity project, and import it. This UnityAds prefab can be used to display ads every several minutes.
5. Drag the prefab into any scene where you want ads to be shown. Make sure to save changes.
6. The time check is shared between all prefabs in all scenes, so you will never show too many ads.
7. The final step is to activate UnityAds services and get your unique project ID.
8. Open the services window and choose your organization, then click create.
9. Choose UnityAds from the list and turn it On.
10. Choose age group for your project (Will affect the nature of ads shown), and save changes.

11. While working on your project keep Test Mode activated. But when you are ready to release the final project, switch Test Mode off.

12. That's it! Now when you start the game, an ad will be shown after 3 minutes. The ad will never appear during gameplay or post-game screen. Instead, it will wait until the next level load (restart, main menu, etc) and then show the ad.

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.

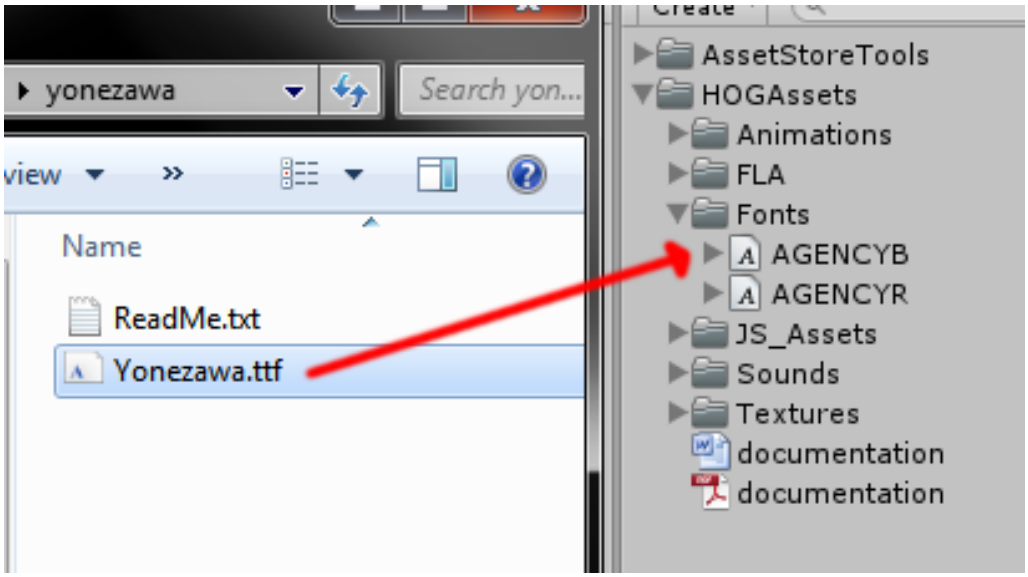
To solve this we must change the texture compression format for iOS. Follow these steps:

1. Click on a texture in the project view.
2. Click on the override for iPhone button on the right side.
3. Change the format to 16bit.
4. Click Apply.

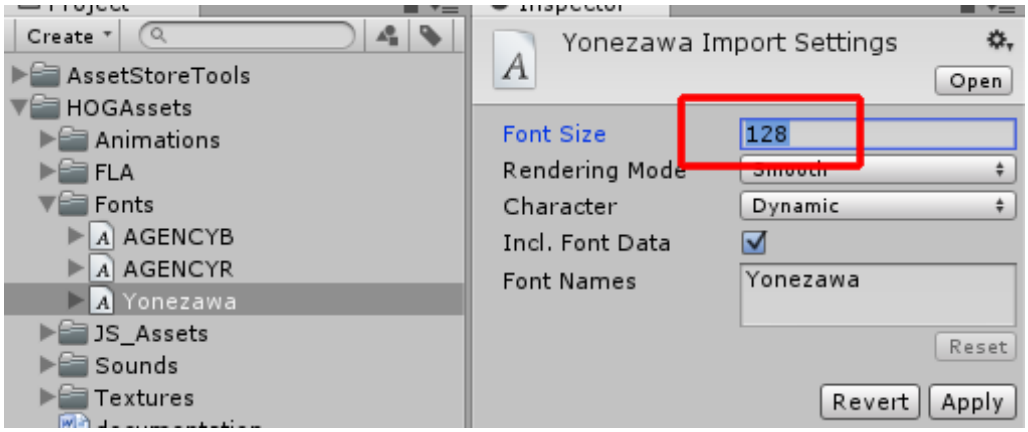
How to change font in the game?

To change a font in the game do the following:

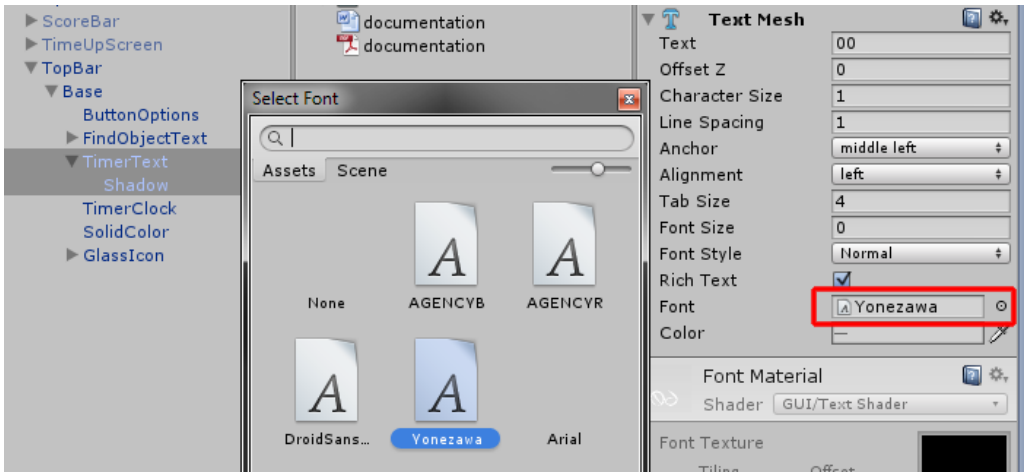
Find a font you like and drag the .ttf file over to the Fonts folder in your game.



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.

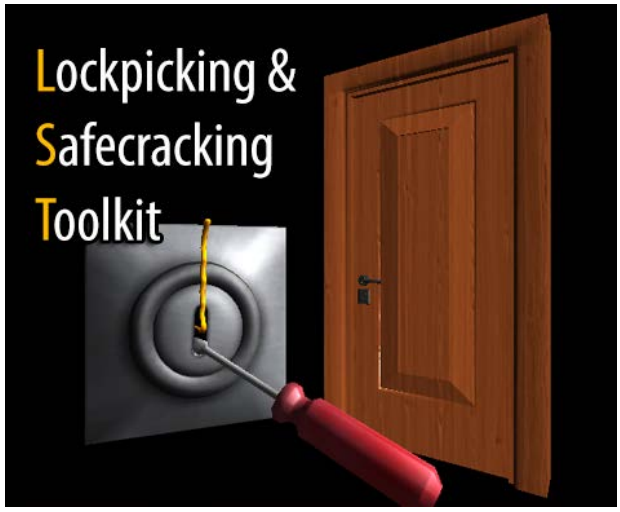


Select any text object in the game and change its font to the new font you have. Sometimes the text might disappear, but it's normal. Just write something in the text box above and it will refresh. Also, make sure you change the text for the shadow; you can select both the main text and its shadow and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



More games by Puppeteer

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!