Gunslinger Game Template

Game documentation and HowTo guide.

This document contains:

Package Description and features

Gunslinger Game is a full Unity template ready for release. It is compatible with mobile as well as standalone and webplayer.

How to Play?

Shoot targets and keep them bouncing up in the air. Balance more than one target for extra bonus, and don't let any of them fall!

Try the webplayer

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.
- UnityAds support with integration guide.

Current version 1.17

Update history

1.17 (19.07.2016)
- Support for Unity 5.3 and higher versions.
- Better support for UnityAds 5.2 and above.

1.15 (02.02.2016)
- Fixed first-tap error on android.

1.14 (22.01.2016)
- Support for UnityAds along with an integration guide.
- Uploaded packages for Unity 4.6.9, 5.1, 5.2, and 5.3
- Support for SceneManager.

1.11 (10.08.2015)
- Added a side-throw chance. Some objects will be thrown from the side of the screen across the game area.
- Added more special targets, 2X bonus, enlarge targets, and shrink targets.
- Fixed first-tap error on mobile devices.

1.06 (10.08.2015)
- Added gamepad/keyboard support. Detection of the gamepad/keyboard/mouse is automatic.
- Added a progress wheel that shows your progress to the next level and the current level.
- Added more variations to the targets.

1.03 (14.07.2015)
- Changed code to bypass a Unity3D bug where buttons don't respond if built on iOS 64bit.
- Improved target throwing algorithm, and added a chance for faster thrown targets.
- Improved UI size, color, and features.

1.0 (14.07.2015)
- Initial version

Credits

The font used is Saddlebag by Dieter Steffmann

The sounds are courtesy of the free sound project.

Music is a clip from Born Barnstomers by Brian Boyko

Credits go to these authors for their great sound samples: **eliasheuninck, simon-rue, cgeffex, daboy291, twohoursago, aumguy, lemudcrab, fins, altfuture, lmr9, lolamadeus, spookymodem, dkudos, diboz, lukeupf, blukotek, chemicatz**

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 GSGAssets folder using Unity3D 4.6.7 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:

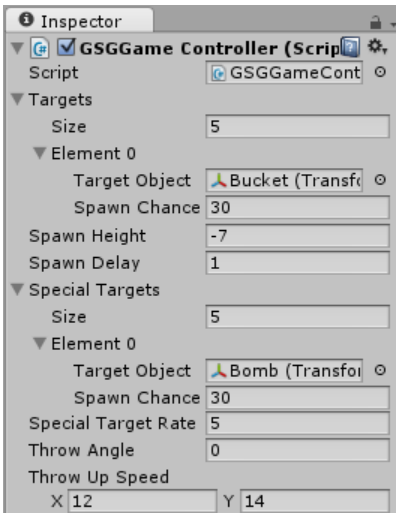
- **Animations:** Holds the animation clips made with Unity's built-in animation system.
- **FLA:** Holds the object graphics made with Flash CS3. These are vector graphics than can be easily scaled without loss of quality and then exported as PNG to be used in Unity.
- **Fonts:** Holds the font used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

Getting started

Gunslinger Game Template (GSG) 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 GSG 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 and throws targets at the player. The Game Controller is also used to calculate the bonus the player gets when hitting a target.



Targets – A list of targets that may be thrown at up into the game area. These are bottle, buckets, etc. In this list you must set the **Target Object** and the **Spawn Chance**. A higher **Spawn Chance** means that the target will appear more often.

Spawn Height – The height (Y position) at which targets are created before being thrown up.

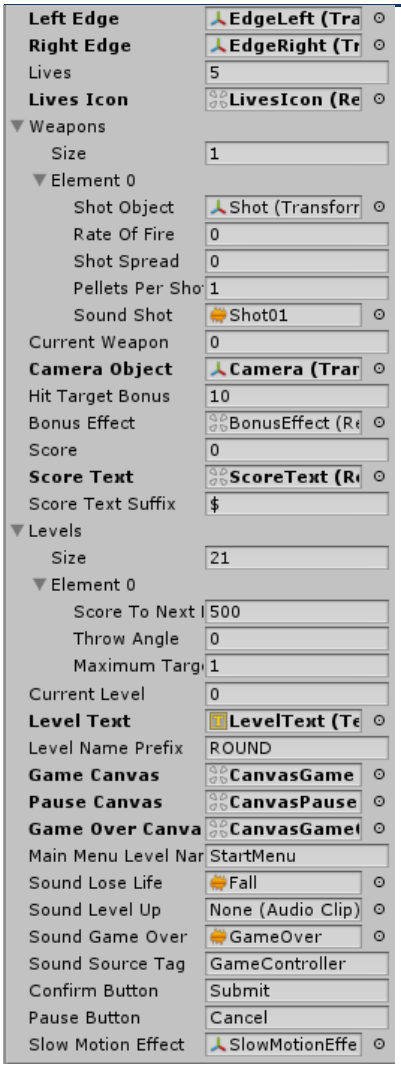
Spawn Delay – How many seconds to wait before throwing the next target.

Special Targets – A list of special targets that may be thrown up into the game area. These may be bad targets like bombs and dynamite, or good targets like the badge and extra life. In this list you must set the **Target Object** and the **Spawn Chance**. A higher **Spawn Chance** means that the target will appear more often.

Special Target Rate – This decides how many normal targets need to be created before a special target is created. **5** means that every 5 normal targets, a special one is created.

Throw Angle – This is basically the sideways speed a target is thrown at. A higher number means a bigger angle range.

Throw Up Speed – This is the speed in which a target is thrown upwards.



Left Edge/ Eight Edge – These collider walls bounce the targets back.

Lives/ Lives Icon – The number of lives the player has, and the icon that displays them.

Weapons – A list of weapons the player can have. Each weapon has a **Shot Object** (The bullet), a **Rate of Fire** (How fast it shoots), **Spread** (How far away from center the shots are created), **Pellets** (How many bullets per shot are created, like a shotgun), and a sound.

Current Weapon – The currently used weapon

Camera Object – The camera is assigned so that we can have effects like the camera flipping when shooting the Mirror special target.

Hit Target Bonus – How many points we get when hitting a target. The bonus is multiplied by the number of targets on-screen.

Bonus Effect – The effects that shows when hitting or destroying a target.

Score / Text / Suffix – The current score of the player, and the text object that displays the score. Also the suffix that is added to the score.

Levels – A list of all the levels in the game. Each level has **Score To Next Level** (How many points we need to win this level), **Throw Angle** (How fast sideways targets are thrown), **Maximum Targets** (The maximum number of targets that can be on-screen at the same time).

Current Level – The current level we are on.

Level Text / Prefix – The text object that shows the number of level we are on, and the prefix text added before the number.

Canvases – These are canvas UI screens. **Game Canvas** appears during gameplay, **Pause Canvas** appears when the game is paused and at the start of the game, **Game Over Canvas** appears at the end of the game when the player dies.

Main Menu Level Name – The name of the level that will be loaded if we choose to quit after Game Over.

Sounds – Various sounds for leveling up, losing a life, and game over.

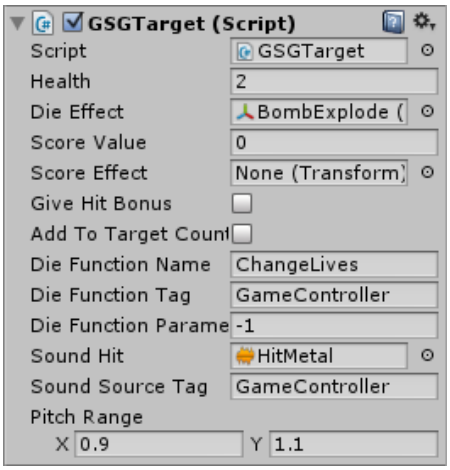
Sound Source Tag – The audio source from which the Game Over sound plays.

Confirm & Pause Button – These are the Keyboard/Gamepad equivalents to the regular UI buttons. If you press Confirm on Game Over you restart, and if you press Pause you quit the level.

Slow Motion Effect – The slow motion effect that appears when destroying a badge.

Editing Targets

There are two kinds of targets in the game; regular ones like the bucket or bottle, and special targets like the badge and bomb. Each of these has its own target list (In the Game Controller) , but they share the same basics.



Health – How much damage this target takes before it's destroyed.

Die Effect – The effect that appears when this target is destroyed.

Score Value – How many points we get when destroying the target.

Score Effect – The score text that is displayed when destroying the target.

Give Hit Bonus – If true, hitting this target will give a bonus.

Add To Target Count – If true, this target is counted towards the maximum number of allowed targets on screen.

Die Function – This is a function that runs when the target is destroyed. You can set the name of the function and the object tag is runs on. In the image above we reduce from the number of lives when the bomb is destroyed.

Sound Hit – The sound that play when this target is destroyed.

Sound Source Tag – The audio source from which sounds play.

Pitch Range – The pitch range is used to make the sound more varied.

UnityAds Integration (Unity 5.2 +)

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.**

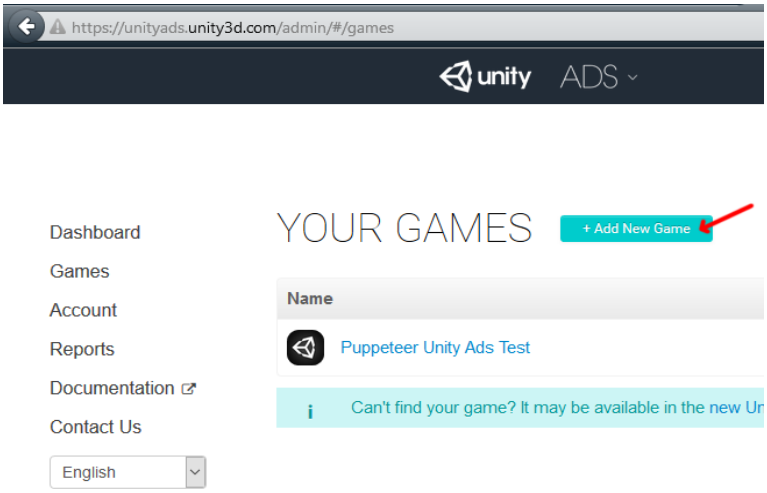
For more info about integrating UnityAds read this:

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

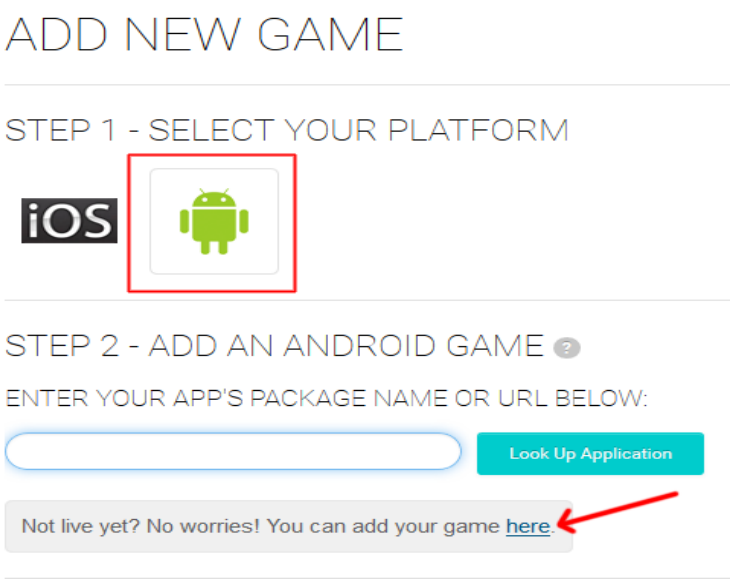
Integrating UnityAds into your project

Adding support for UnityAds into your current project is simple and shouldn't take you more than 5 minutes. Let's start:

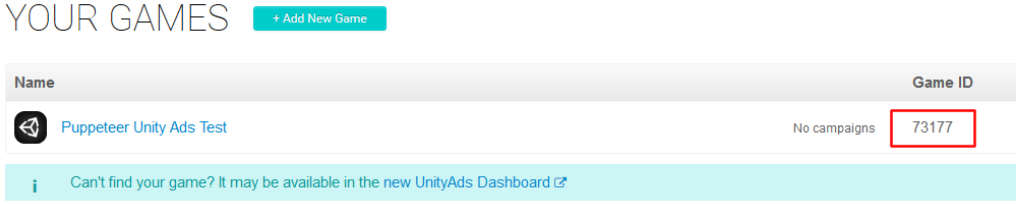
First we need to create our game entry on the UnityAds website. Go to https://unity3d.com/services/ads and create a new game. If you already have your app set and your GameID noted, just skip this part and go straight to importing the UnityAds package into the game.



Now we need to choose the platform. The process is similar for both iOS and Android but for the purpose of this tutorial we'll choose Android. If you have an app on Android, enter its name to find it. If you don't have an app, click below where the red arrow points in order to enter the name of the app that has not been added to the store yet. This way you can test the app before it goes live.



After you created your app in the website, make note of the Game ID that appears. This will be used to link the ads to your app.

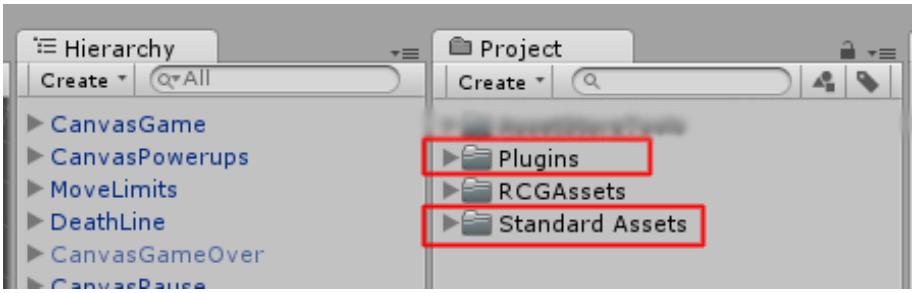


In Unity Editor

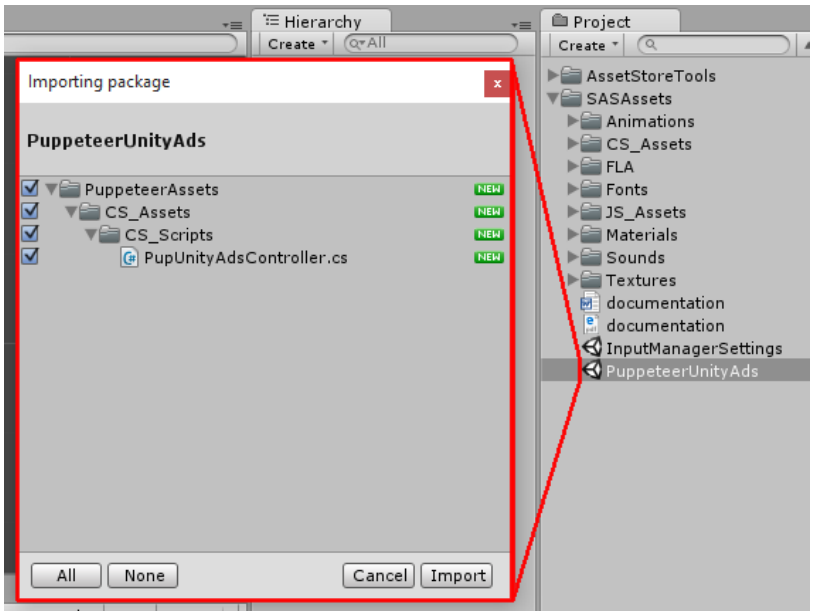
Now we need to import the UnityAds package. Open the Unity Asset Store and download the UnityAds package. Import it into your project.

(https://www.assetstore.unity3d.com/en/#!/content/21027)

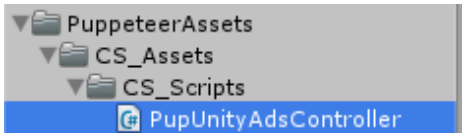
After import you should have two additional folders in your project.

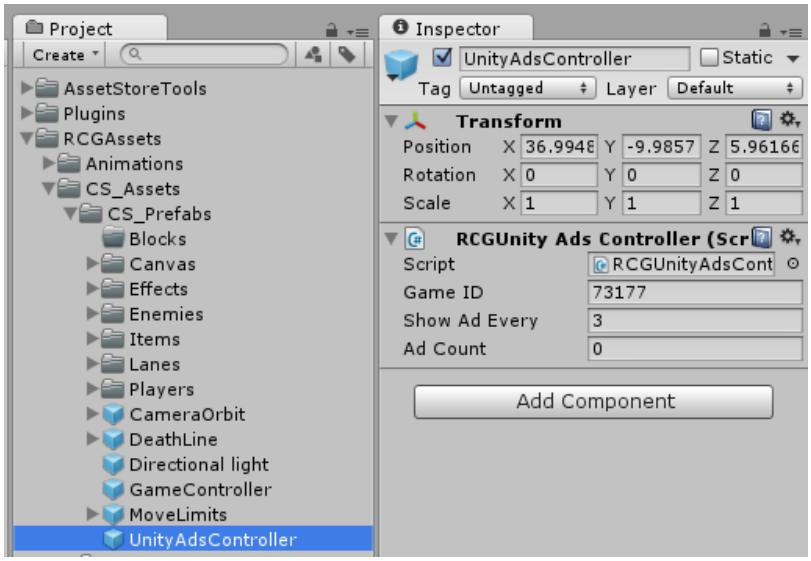


Now we need to bring in the code that integrates the ads into our game. Click on the **PuppeteerUnityAds** package in your project to import it into the game, or choose **Assets > Import Package > Custom Package…** from the top menu and navigate to the **PuppeteerUnityAds** package in your project to import it.

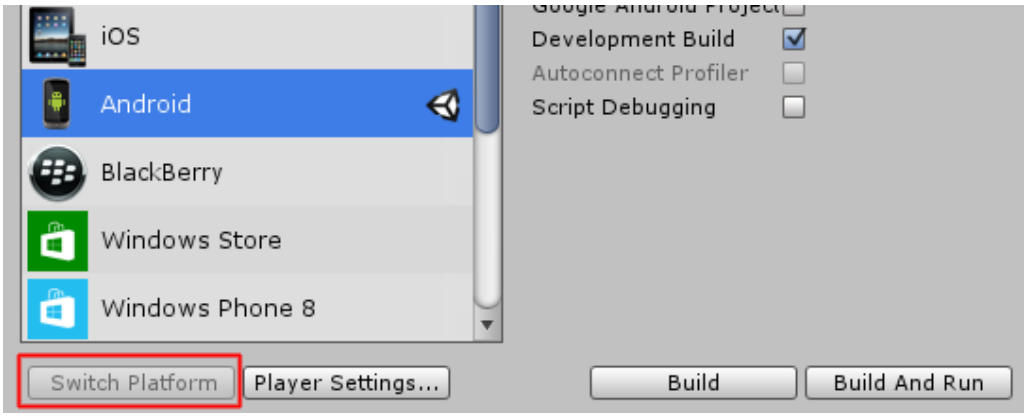


PupUnityAdsController.cs is the main script that links your app to the unityads system. Drag it into your game controller. Now when you look at it you see you can set the GameID of your app, and how often the ads appear. The ad is checked when the level is loaded. "**Show Ad Every**" decides how many times the level needs to be loaded before an ad appears.





In order to test the ads, we need to switch to the Android platform.



That's it! Now start a level and restart it 3 times, then you should see a blue screen showing the ad system has been activated correctly. If you build to Android you should see an actual video ad appear after 3 level loads.

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.

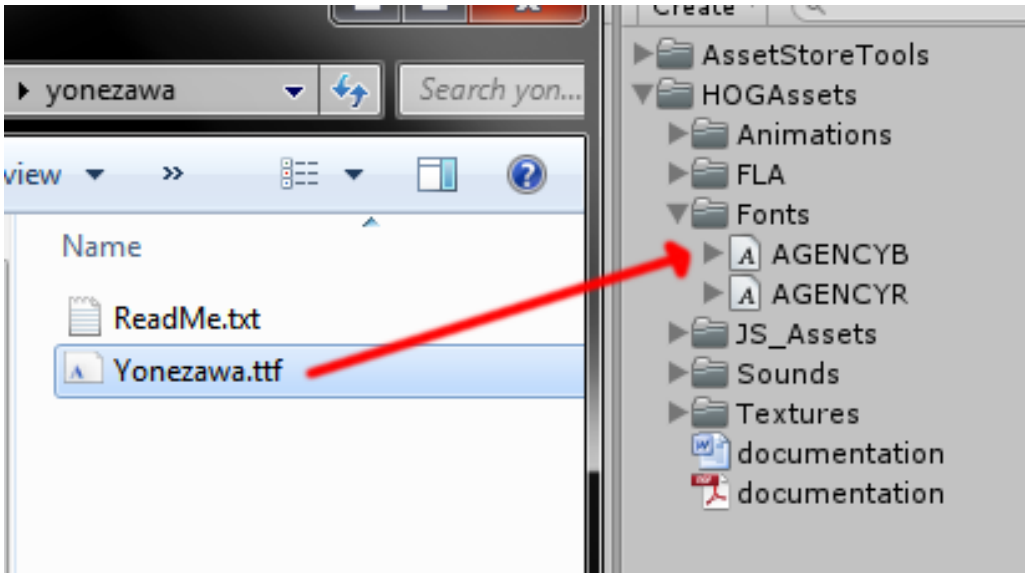
To solve this we must change the texture compression format for iOS. Follow these steps:

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

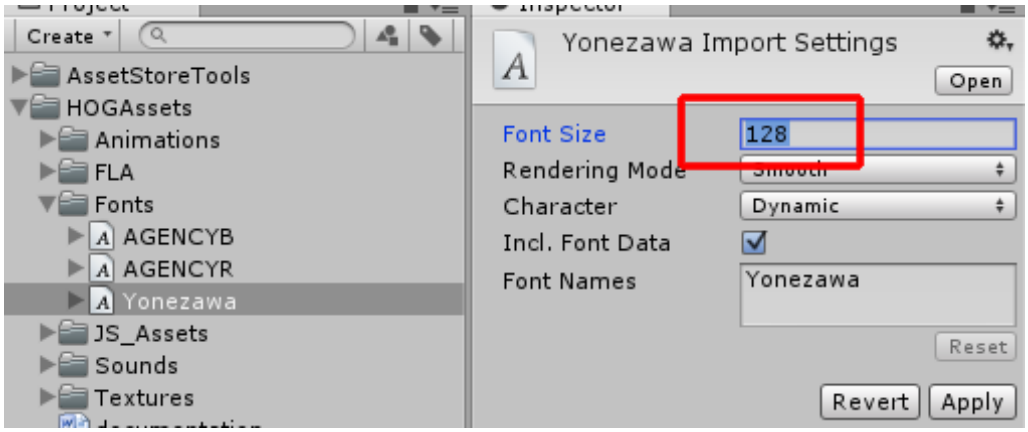
How to change font in the game?

To change a font in the game do the following:

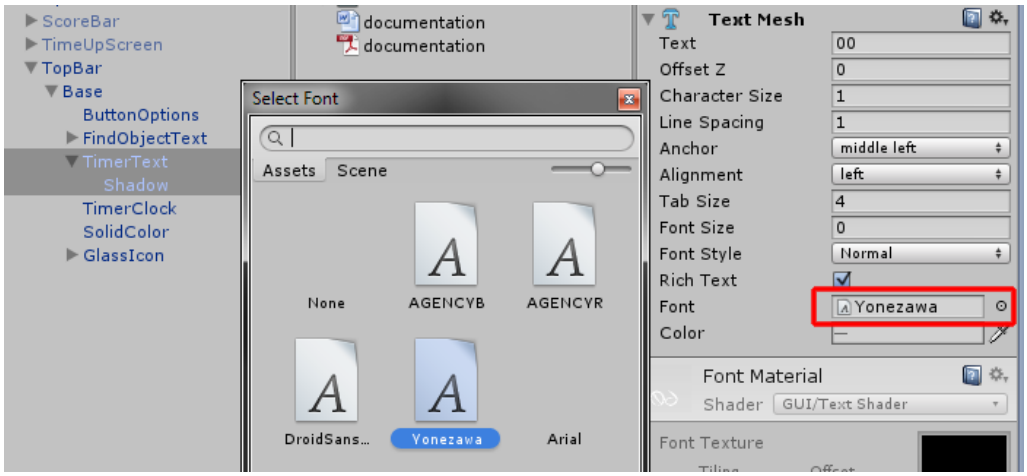
Find a font you like and drag the .ttf file over to the Fonts folder in your game.



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.

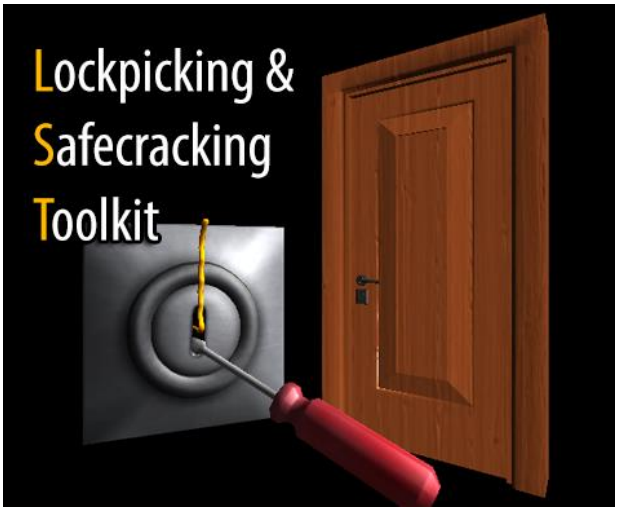


Select any text object in the game and change its font to the new font you have. Sometimes the text might disappear, but it's normal. Just write something in the text box above and it will refresh. Also, make sure you change the text for the shadow; you can select both the main text and its shadow and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