

Zombie Drive Game Template

Game documentation and HowTo guide.



This document contains:

Package Description and features	2
Try the demo	2
Credits.....	2
Overview of the game's library contents	3
Customization Guide.....	4
Getting started.....	4
The Game Controller	4
Obstacles, Zombies, and Pickups	6
UnityAds Integration	7
Frequently Asked Questions	9
Does this package work on mobile?	9
My sprites are not showing on iOS	9
How to change font in the game?	9
More games by Puppeteer	11

Package Description and features

Survive the apocalypse in your armored van, collect repairs and fuel to keep going, smash through hordes of zombies and avoid deadly obstacles! 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, keyboard, gamepad, or touch controls.

How to Play?

Tap to change direction, avoid obstacles, hit zombies, and collect fuel and repair.

Try the demo

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 1.0

1.04 (19.10.2016)

- Added a speed increase based on how many points we got, so the level of challenge increases.
- Added gaps to spawns so now you can create a whole section and set a gap between it and the next spawn. There is an example for this in the Zombie Horde spawn.
- Fixed gamepad support so now you can navigate the UI and play the game with the gamepad.
- Fixed the gib physics when the zombie explodes. Now its head fly up correctly.
- Fixed the pick objects being chosen randomly from the list, so now they appear consecutively as they should, for example the repair item appears, then the fuel, then the repair again, and so on.
- Fixed the music button not muting correctly.

Credits

The sounds are courtesy of [the free sound project](#).

Music is Intrusion by Severed Personality (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 ZDGAssets folder using Unity3D 5.4.1 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 that 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.

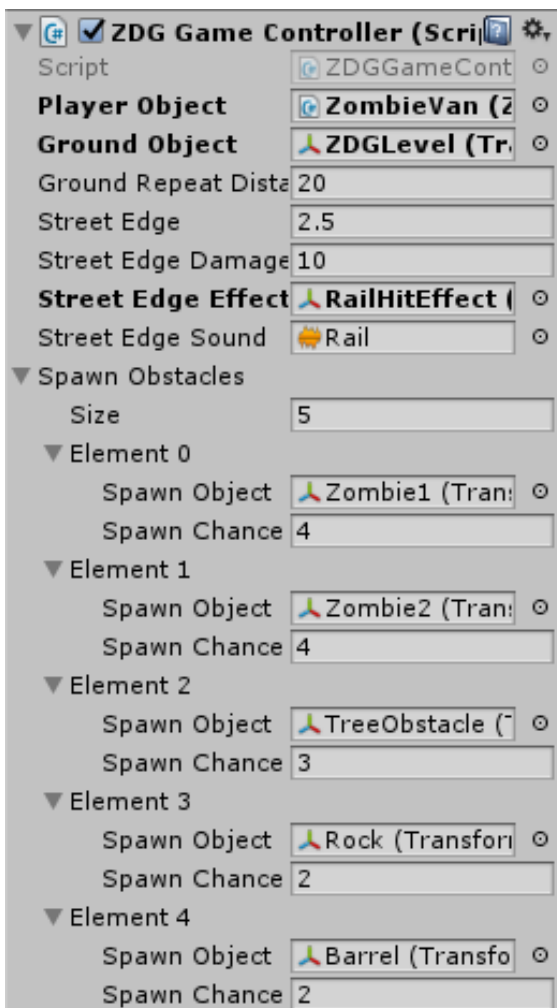
Customization Guide

Getting started

Zombie Drive Game (ZDG) 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 ZDG 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 game over condition.



chance for it to appear. The higher the number the higher the chance compared to the rest of chances in the list.

Player Object – The player object assigned from the scene.

Ground Object – The ground object that repeats under the player while he is moving.

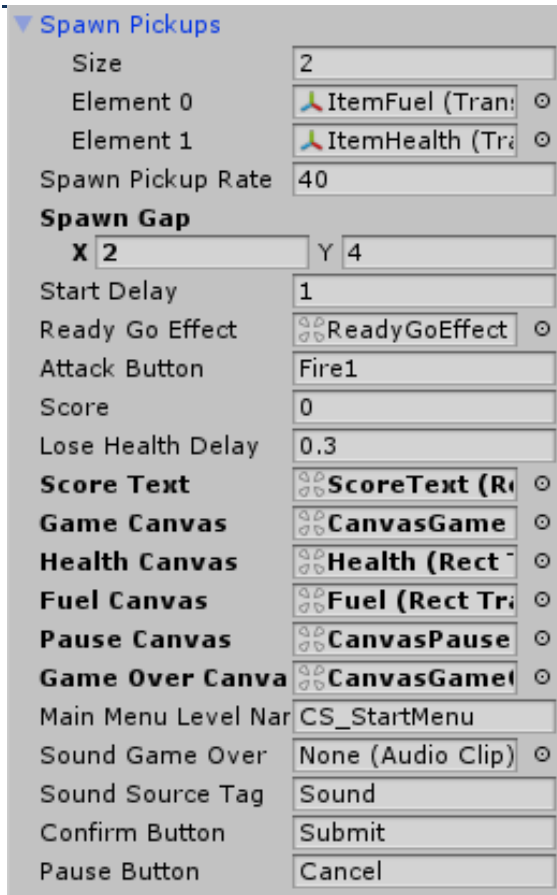
Ground Repeat Distance – How far should the player move before the ground object repeats.

Street Edge - The edge of the street where there is a railing that bounces the player back.

Street Edge Damage - How much damage touching the railing causes to the player.

Street Edge Sound - The sound that plays when hitting the rail.

Spawn Obstacles – These are the obstacles that appear in the street and can hurt the player, or be killed by the player. You can set each spawn object and the



Spawn Pickups – The items you can pick up which give you fuel and health/repair. These only appear every certain number of obstacles.

Spawn Gap – How far apart are obstacles and pickups created? This is a min/max range.

Start Delay – How long to wait before starting the game. Ready/GO! Time.

ReadyGoEffect – The effect displayed before starting the game.

Turn Button – The button that changes the player direction.

Lose Health Delay – A delay to prevent the player from too much health at once. If you lose

health, you will not lose more health for some time.

Score - The score of the player.

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

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.

Obstacles, Zombies, and Pickups

Obstacles, zombies, and pickups all use the same component to achieve different effects. This is the ZDGTouchable which basically reacts to the player touching it and triggers a custom function. In each of those objects the function that is called is different. First let's look at the general format of the component:



Touch Target Tag – The tag of the object that can touch this block.

Touch Function – The function that runs when this object is touched by the target. Here you can enter any function that exists in the target script, such as ChangeScore, ChangeHealth, and ChangeFuel.

Function Parameter – The parameter that will be passed with the function.

Function Target – The target object that the function will be run from.

Touch Effect - The effect that is created at the location of this object when it is touched.

Rotation Range - A random rotation given to the object only on the Y axis.

In the case of the Zombie we set the function to **ChangeScore** and the parameter to the score we want to give the player.

In the case of the Fuel pickup we set the function to **ChangeFuel** and the parameter to the fuel we want to add to the player.

In the case of the Health/Repair pickup we set the function to **ChangeHealth** and the parameter to a positive value that will be added to the player's health.

In the case of the other Obstacles we set the function to **ChangeHealth** and the parameter to a negative value that will be reduced from the player's health.

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>

Frequently Asked Questions

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.

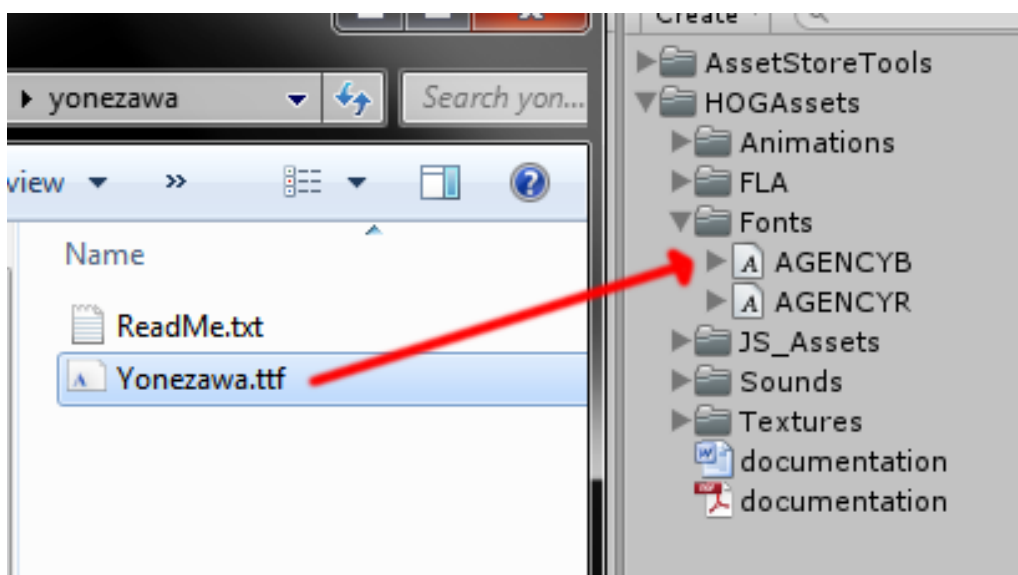
To solve this we must change the texture compression format for iOS. Follow these steps:

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

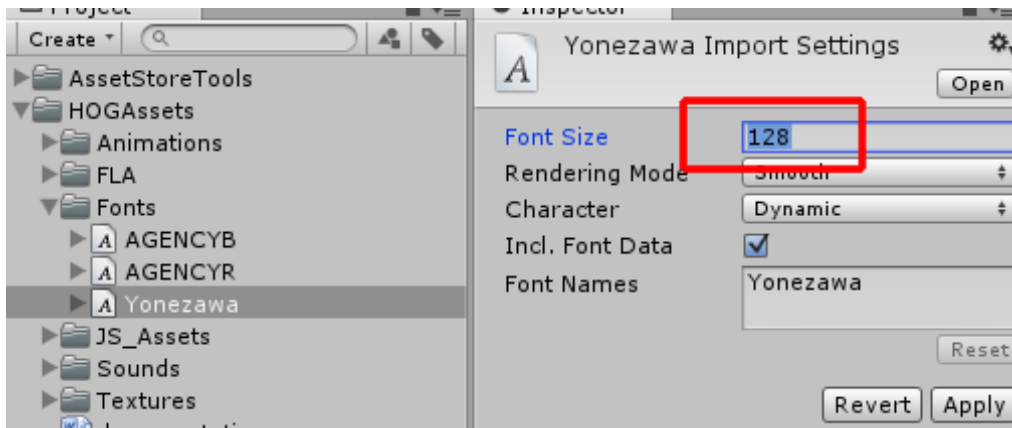
How to change font in the game?

To change a font in the game do the following:

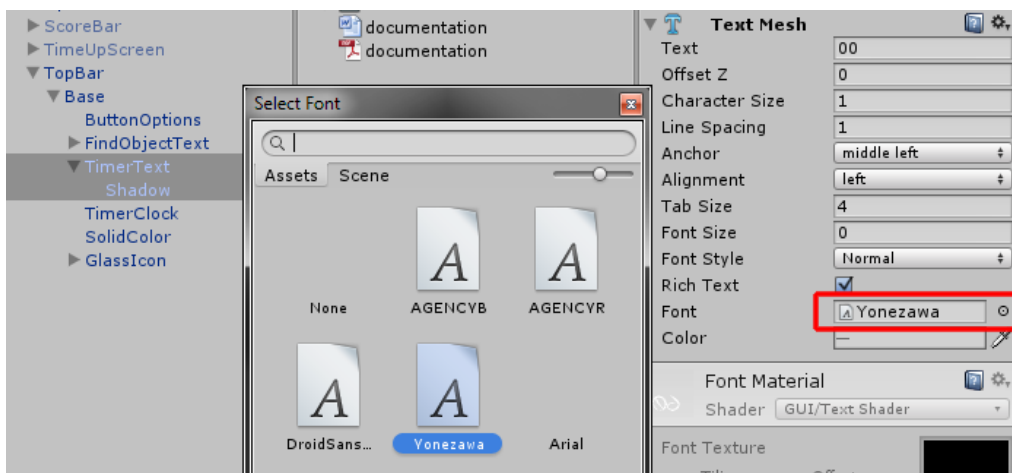
Find a font you like and drag the .ttf file over to the Fonts folder in your game.



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!