By Majd Abdulqadir

Road Crossing Game Template

Game documentation and HowTo guide.

FROGGY
CROSSES THE ROAD!
HELP FROGGY THE FROG FILL HIS BELLY WITH YUMMY YUMMY COINS, BUT BE CAREFUL WHEN YOU CROSS THE ROAD!
CLICK TO START

This document contains:

Package Description and features

Road Crossing Game is a full Unity template ready for release. It is compatible with mobile as well as standalone and webplayer. It is similar in genre to classic arcade games, as well as modern infinite runners.

How to Play?

Use the keyboard arrows or the mouse to move around, collect coins and avoid being killed by the many deadly obstacles. The game controls also fit mobile and console without need for further coding.

Try the webplayer

Current version 1.18

Update history

(1.18) 05.03.2015

- Added a shop where you can buy and unlock new characters to play as. The coins you collect during gameplay are stored and can be used to buy new cute animals.
- You can toggle swipe controls for mobile devices. Swipe in a direction to move the player.
- You can toggle the random move direction for the objects on a lane.
- You can also set the width of each lane, so you can now have lanes of various widths.
- Locked movement of player to the grid.

(1.1) 12.02.2015

- Entire project ported to C# thanks to Jordan Swapp. You now get the project in both JS and C#.
- Improved performance and reduced drawcalls.

(1.03) 09.02.2015

- You can limit the speed of the Death Line.
- You can set the minimum gap between moving objects.
- MAX models are now included in the package
- Updated tutorial to cover reskinning 3D models, creating new models, and creating new tiles.

Credits

The sounds are courtesy of the free sound project.

Intro music is a clip from Vivacity by Incompitech

Credits go to these authors for their great sound samples: **mattj99, anechoix, joelaudio, tramppa34, fenrirfangs, zerolagtime, atomwrath, jc144940**

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 RCGAssets folder using Unity3D 4.6 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
- **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.
- **UI:** Holds all the canvases the game which are used to hold buttons and other UI elements.

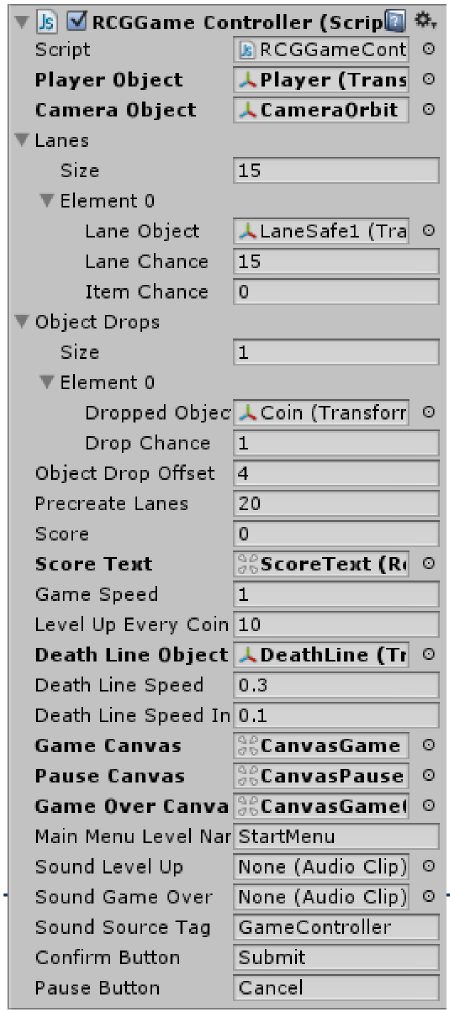
Customization Guide

Getting started

Road Crossing Game (RCG) 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 RCG 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 lanes, and also makes the camera chase the player. The Game Controller is also used to increase the difficulty of the game after collecting a set amount of coins.



RCGGame Controller (Scrip

Script	RCGGameCont
Player Object	**Player (Trans**
Camera Object	**CameraOrbit**
Lanes	
Size	15
Element 0	
Lane Object	LaneSafe1 (Tra
Lane Chance	15
Item Chance	0
Object Drops	
Size	1
Element 0	
Dropped Objec	Coin (Transforr
Drop Chance	1
Object Drop Offset	4
Precreate Lanes	20
Score	0
Score Text	**ScoreText (R**
Game Speed	1
Level Up Every Coin	10
Death Line Object	**DeathLine (Tr**
Death Line Speed	0.3
Death Line Speed In	0.1
Game Canvas	**CanvasGame**
Pause Canvas	**CanvasPause**
Game Over Canva	**CanvasGame(**
Main Menu Level Nar	StartMenu
Sound Level Up	None (Audio Clip)
Sound Game Over	None (Audio Clip)
Sound Source Tag	GameController
Confirm Button	Submit
Pause Button	Cancel

Player/Camera Object – These two must be assigned from the scene to make the game playable. The GameController makes the camera follow the position of the player as it moves around.

Lanes – This list contains all the different lanes that are created as the player goes forward in the game. You can set the chance for this lane to appear, as well as the chance of an item to appear on this lane.

Object Drops – This list contains all the items that may drop in a lane. You can set the chance of a certain item to appear.

Object Drop Offset – This is how far along a lane an item can be dropped.

Precreate Lanes – How many lanes to

create at the start of the game.

Changing Game Difficulty

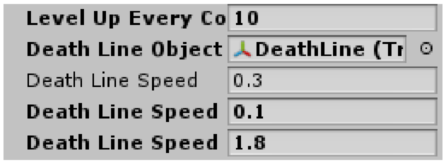
The GameController also allows you to change the difficulty of the game through several variables, making the death fog line get gradually faster with as you rise in the levels.

The Death Line:

The death line is the object (The fog) that constantly chases the player as he advances through the level.

You can control how fast it advances toward the player by changing the **Death Line Speed** field, and you can also set how much faster it becomes **(Death Line Speed Increase)** after collecting a number of coins **(Level Up Every Coins)**.

So based on the settings below, the speed of the death line increases by 0.1 after collecting 10 coins, and keeps doing so until it reaches the maximum speed of 1.8.



Level Up Every Co	10
Death Line Object	DeathLine (Tr
Death Line Speed	0.3
Death Line Speed	0.1
Death Line Speed	1.8

Level Up Every Coins – Is the number of coins you need to collect to level up.

Death Line Object – Is the object that keeps chasing the player.

Death Line Speed – The movement speed of the death line.

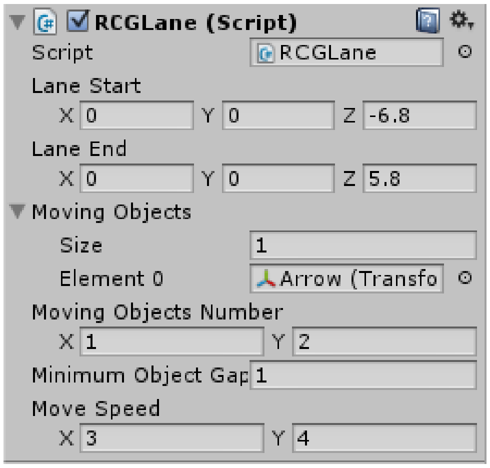
Death Line Speed Increase – The increase in the speed of the death line.

Death Line Speed Max – The maximum speed of the death line.

Notice that the death line will always catch up to the player whenever he is advancing in the level. This is a game design decision I made. If you think it should be made otherwise, tell me.

If you don't want the pressure of fog behind you, you can simply clear the **Death Line Object** field. Of course you can have all kinds of death lines instead of the fog, like a set of spinning spikes for example, or a horde of blobs. Just ideas!

Another way to increase overall difficulty in the game is to change the attributes of the lanes. Here you can change the number of moving objects that appear in a lane, and the movement speed of the objects.



RCGLane (Script)
Script: RCGLane
Lane Start
X 0 Y 0 Z -6.8
Lane End
X 0 Y 0 Z 5.8
Moving Objects
Size: 1
Element 0: Arrow (Transfo
Moving Objects Number
X 1 Y 2
Minimum Object Gap: 1
Move Speed
X 3 Y 4

Lane Start/End – These are the points that objects move through.

Moving Objects – This is a list of the objects types that will be created in a lane.

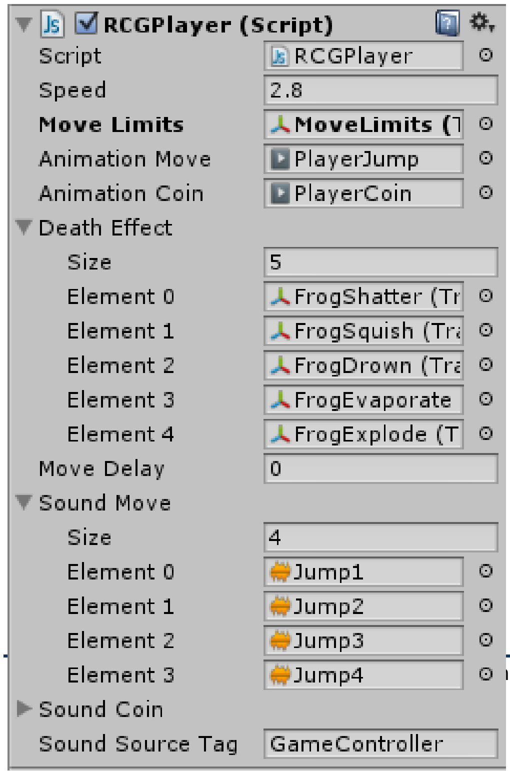
Moving Objects Number – A random range for the number of objects that will be creates in the lane. The objects are scattered along the lane with some randomization in position.

Minimum Object Gap – The minimum distance between two moving objects in a lane. This is used to prevent any objects from overlapping each other.

Move Speed – A random number between these two values is chosen as the speed of all the moving objects in this lane.

Editing the Player

The player object has several variables that can change how it behaves, mainly the move speed death effects.



RCGPlayer (Script)
Script: RCGPlayer
Speed: 2.8
Move Limits: **MoveLimits (T**
Animation Move: PlayerJump
Animation Coin: PlayerCoin
Death Effect
Size: 5
Element 0: FrogShatter (Tr
Element 1: FrogSquish (Tra
Element 2: FrogDrown (Tra
Element 3: FrogEvaporate
Element 4: FrogExplode (T
Move Delay: 0
Sound Move
Size: 4
Element 0: Jump1
Element 1: Jump2
Element 2: Jump3
Element 3: Jump4
Sound Coin
Sound Source Tag: GameController

Speed – The movement speed of the player.

Move Limits – This object is assigned from the scene, and it has colliders that blocks the player's movement to the sides by bouncing it back.

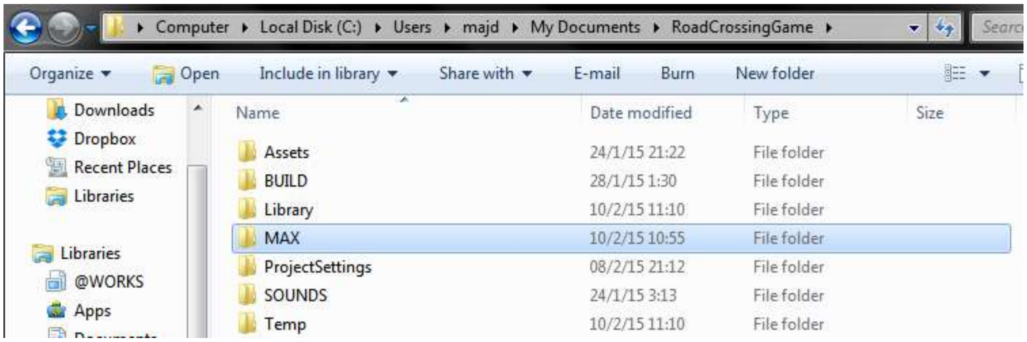
Death Effect – A list of effect objects that are created after the player object is removed. This is decided by the object that kills the player.

Sound Move – A list of sounds that are played randomly when the player moves.

Sound Coin – A list of sounds that area played randomly when the player gets a coin.

Changing Graphics

Graphics in Road Crossing Game are 3D. They are modelled in 3ds Max and then exported to FBX to be used in Unity. I included the MAX files in the package; you can find it in ExtraAssets.rar. The reason for having it inside an archive and not in a simple folder in the project is because unity has trouble dealing directly with MAX files, so when you have any MAX models in unity the software will hang up for a while in order to accommodate the new objects. So my recommendation is to keep these files just outside of your project folder, like this:



Name	Date modified	Type	Size
Assets	24/1/15 21:22	File folder	
BUILD	28/1/15 1:30	File folder	
Library	10/2/15 11:10	File folder	
MAX	10/2/15 10:55	File folder	
ProjectSettings	08/2/15 21:12	File folder	
SOUNDS	24/1/15 3:13	File folder	
Temp	10/2/15 11:10	File folder	

The textures are made using Flash CS3, and then exported to PNG. I do this because Flash allows me to control the quality of the textures easily since it is vector based graphics.

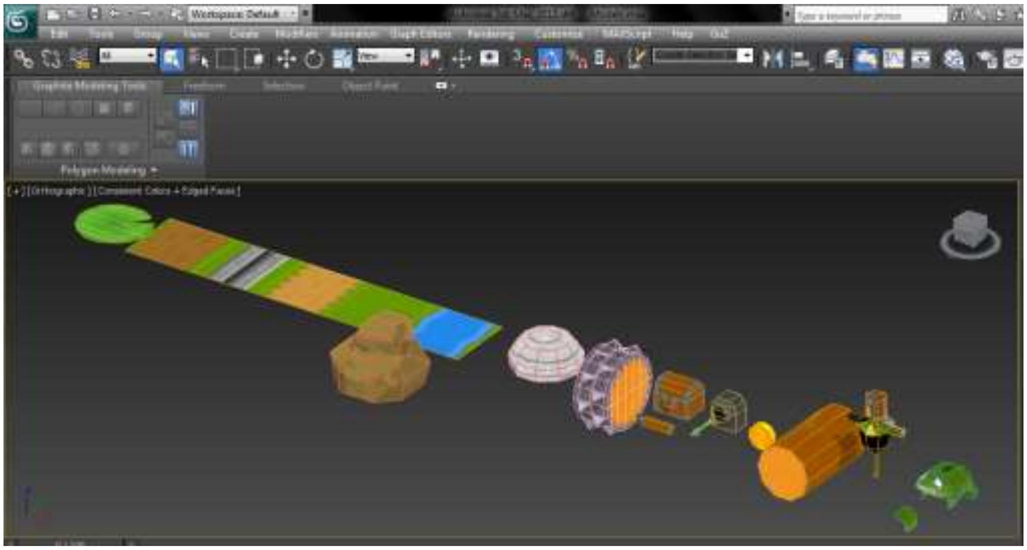
The various animations are made in unity itself and are assigned to generically named parts of the model, so that you can easily switch models without while maintaining the animation on the object.

Reskinning 3D Models

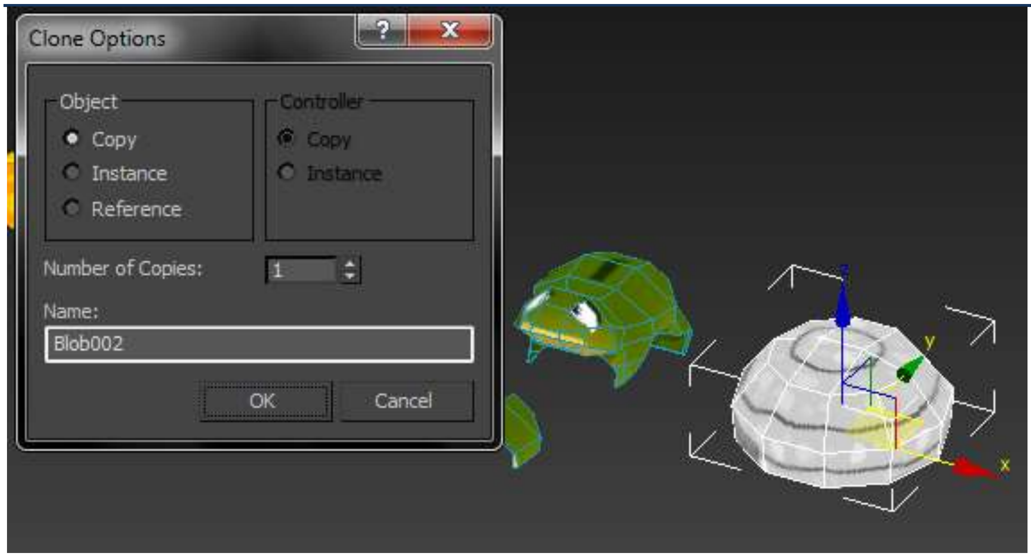
Here's how you can put your 3D models instead of the original ones, using the "reskin" method. Using this method you can simply "reskin" your model with another model without having to worry about creating new prefabs and assigning meshes.

- The package includes a MAX folder that contains all the models in the game. You can open this in 3DS Max 2014 or above. Open the file Models.max. This contains all the models in the game.

- Now you can edit each model individually, and check how it scales relative to other objects. Once you are done, you can export the whole file as FBX to be used in unity.



- For our specific example I will replace the frog model with a blob model. First I shift+drag the blob object to create another copy of it.



Clone Options

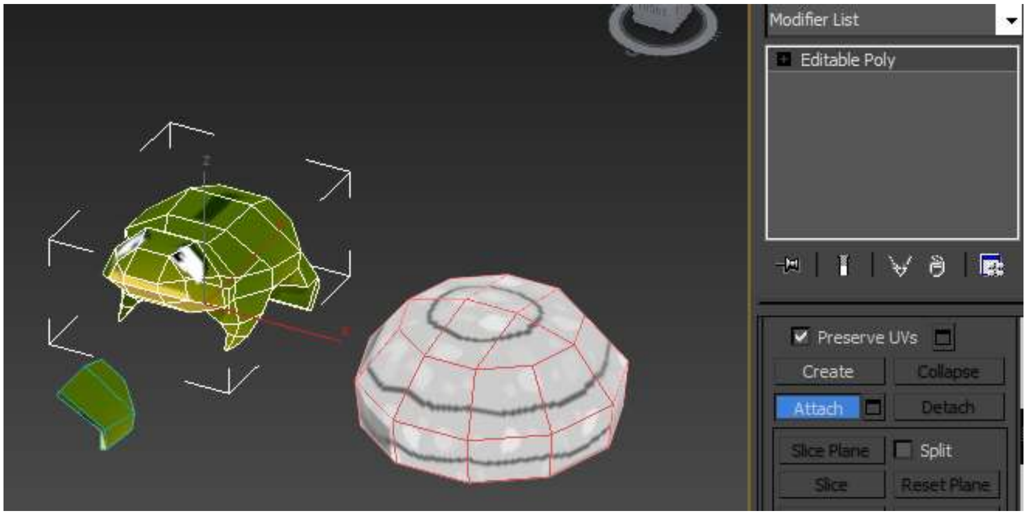
Object
● Copy
○ Instance
○ Reference

Controller
● Copy
○ Instance

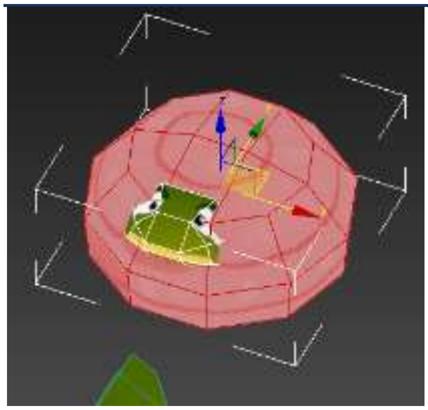
Number of Copies: 1

Name: Blob002

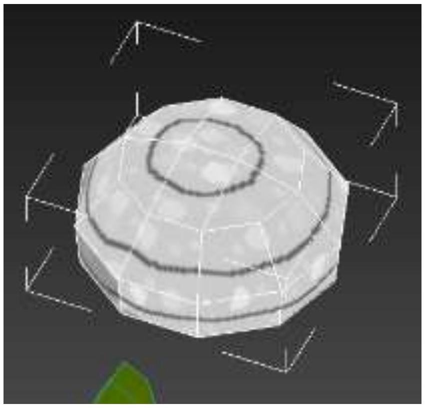
- Then I select the frog model and **Attach** to it the blob model. Click the frog, click attach, and then click the blob. Now the blob is part of the frog model.



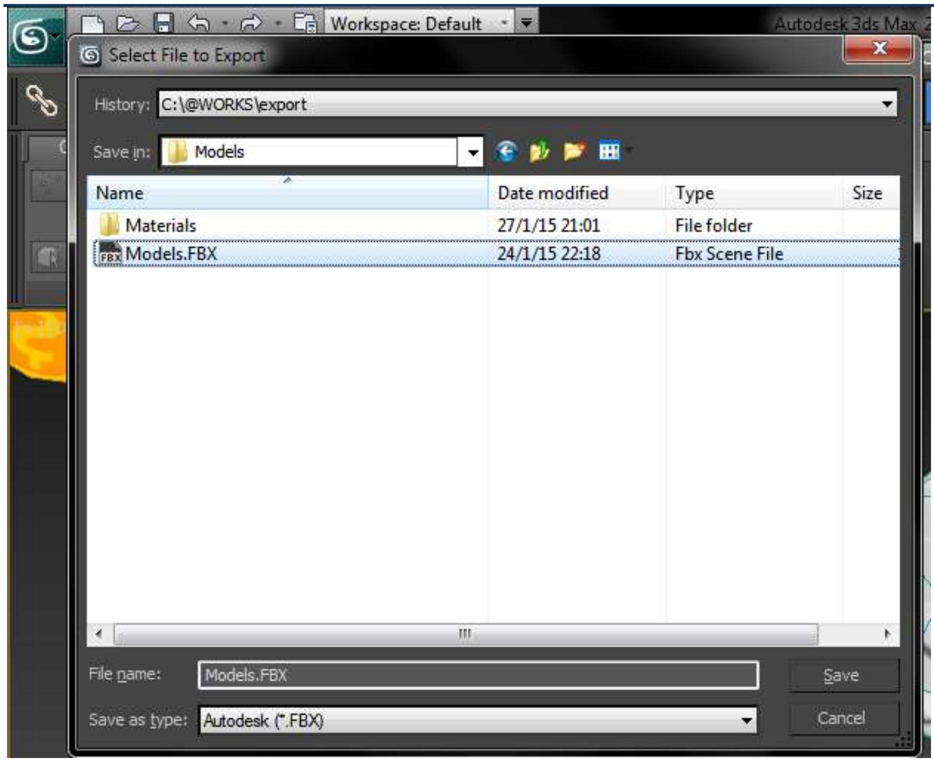
- Now switch to element mode, select and move the blob to stand right over the position of the frog.



- Next we select the frog element and delete it. Now we are left only with the blob.



- To complete our work we will export our models into FBX for use in Unity.
- Click **Export** from the menu, go to the folder **Models** inside your game project and overwrite the file **Models.FBX**



Select File to Export

History: C:\@WORKS\export

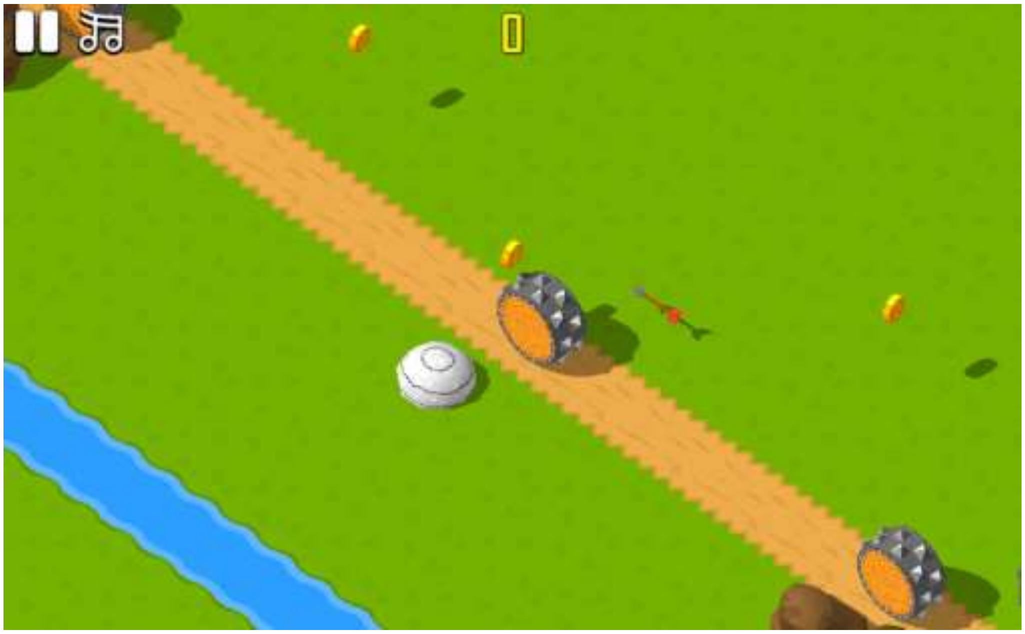
Save in: Models

Name	Date modified	Type	Size
Materials	27/1/15 21:01	File folder	
Models.FBX	24/1/15 22:18	Fbx Scene File	

File name: Models.FBX

Save as type: Autodesk (*.FBX)

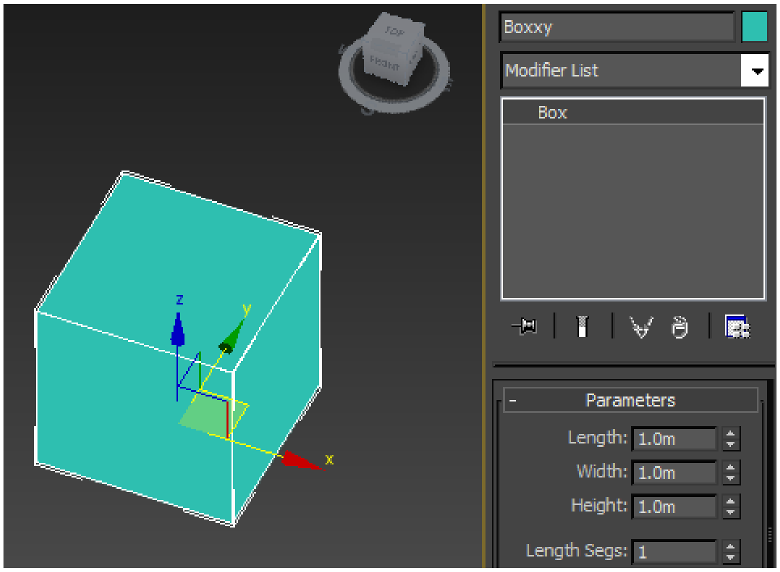
- And here is the result:



You'll notice the "blob" player has the same attributes of the "frog" as before because we only replace the mesh and nothing else. This is the most straightforward method for reskins, but you can also of course create new models and use them in the game without having to replace a current model.

Creating a new model

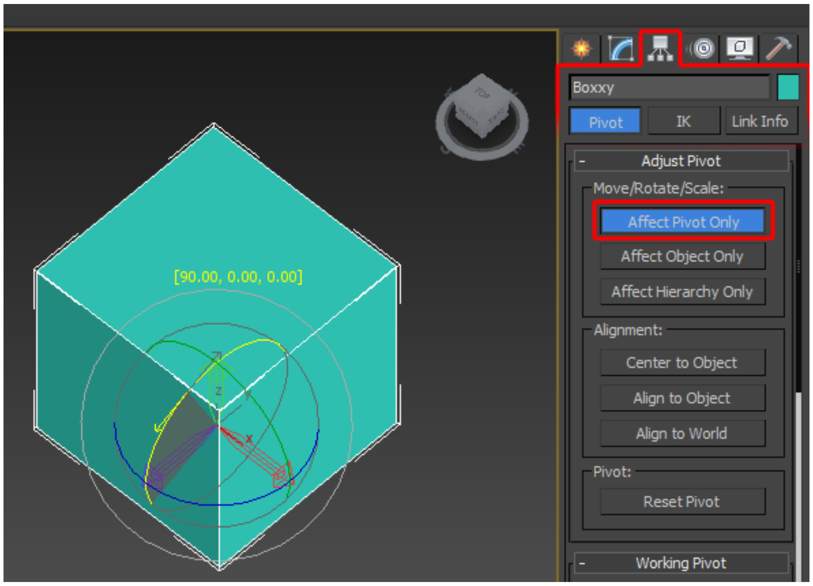
To create an entirely new model, just model it in 3ds max inside **Models.MAX**, give it a unique name and then export to **Models.FBX** again.



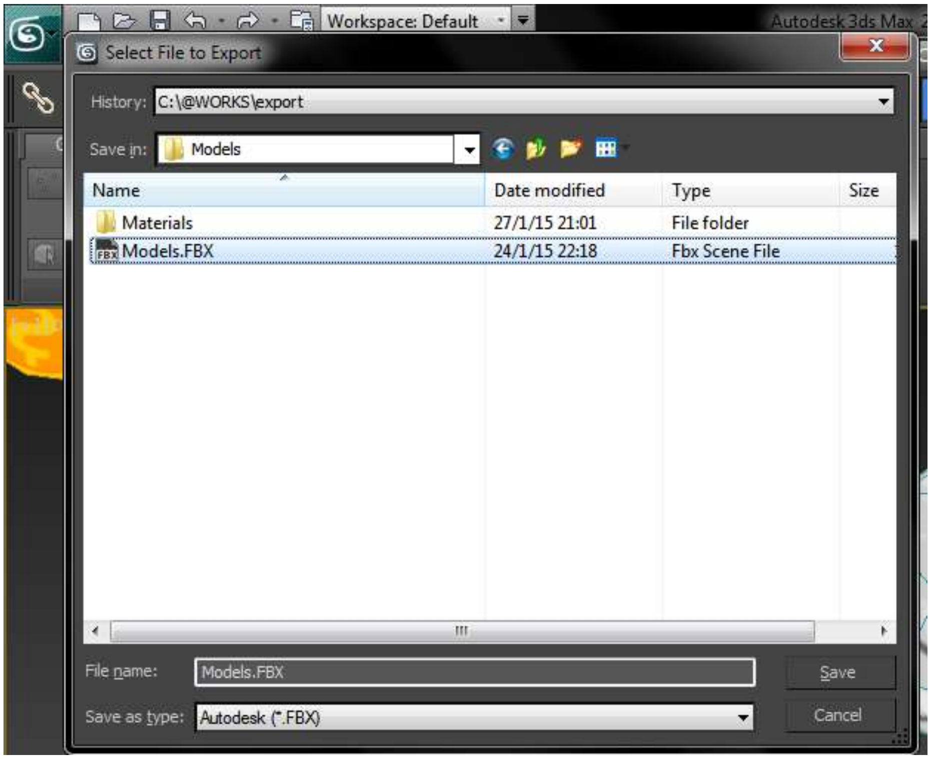
Length: 1.0m
Width: 1.0m
Height: 1.0m
Length Segs: 1

- Note: When creating a model in 3ds max for unity, make sure the Z (blue) axis is looking at the front and the Y (green) axis is looking up. This will align the model properly in unity.

- To do this, select your model and go to the Hierarchy pane, then choose Affect Pivot Only, and rotate the axis 90 degress so it faces the proper direction.



- Click **Export** from the menu, go to the folder **Models** inside your game project and overwrite the file **Models.FBX**



History: C:\@WORKS\export

Save in: Models

Name	Date modified	Type	Size
Materials	27/1/15 21:01	File folder	
Models.FBX	24/1/15 22:18	Fbx Scene File	

File name: Models.FBX

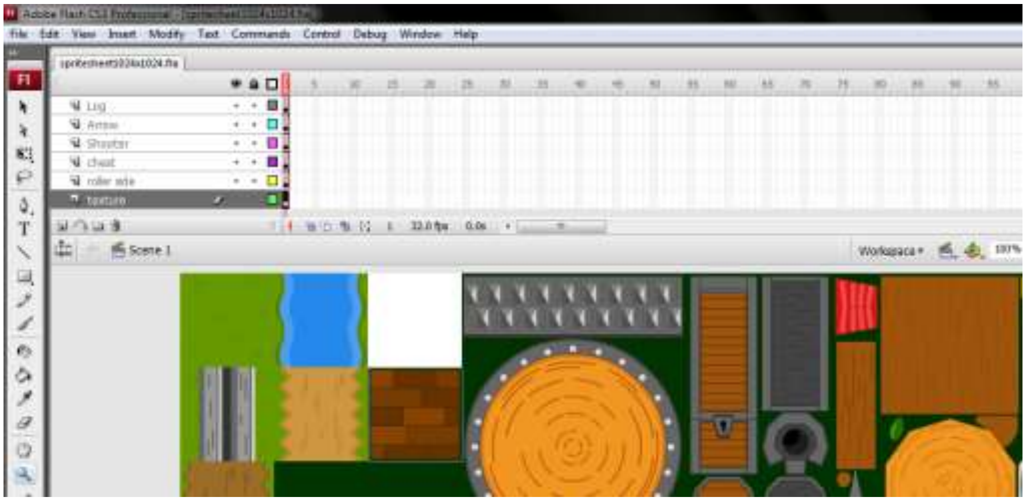
Save as type: Autodesk (*.FBX)

- Now when you go to Unity and check the file **Models.FBX** inside **Models** folder, you'll find a new model named **Boxxy** which you can use in your game!

Creating a new tile

Here's the quickest way to make your own quad:

- For our example I want to make a new tile with a brick texture. I keep all the textures in the same sprite sheet to keep size down.

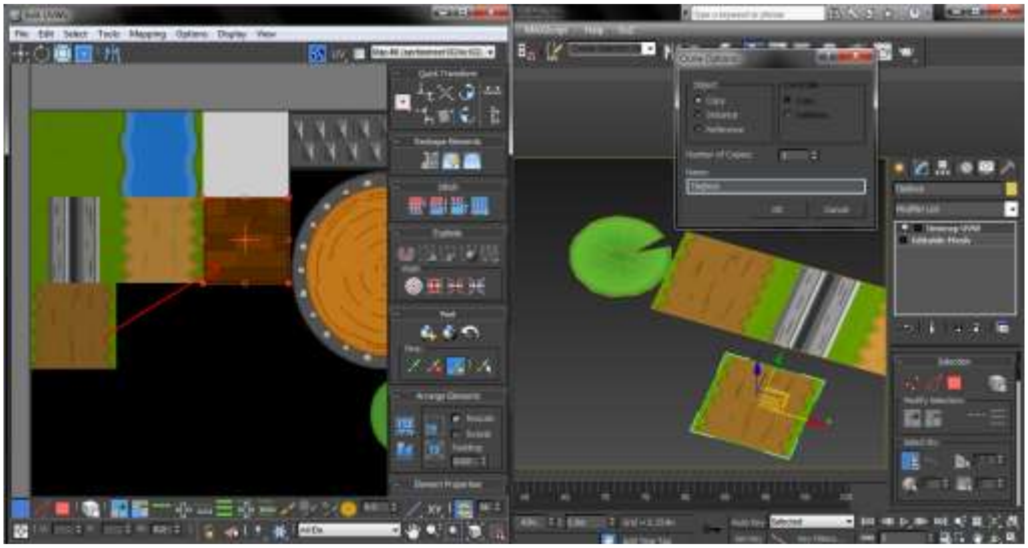


File Edit View Insert Modify Text Commands Control Debug Window Help
Arrow
Shooter
chest
roller side
texture
Scene 1
Workspace

- Note: The package includes a MAX folder that contains all the models in the game. You can open this in 3DS Max 2014 or above. Open the file Models.max.

This contains all the models in the game.

- In 3ds max I duplicate one of the tiles and rename it, then edit it UVWs, by just dragging them from the previous tile to this new one's texture. If you can't see the texture, choose it from the menu on the top right (where it says Map#0 (spritesheet1024x1024)).



Now just export and overwrite the models.FBX file in your project.

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.

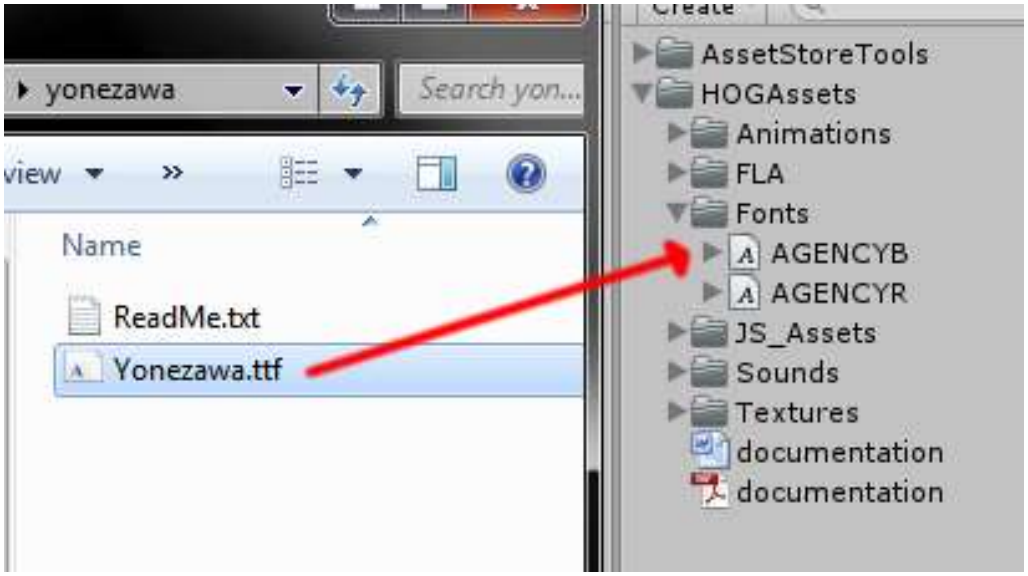
To solve this we must change the texture compression format for iOS. Follow these steps:

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

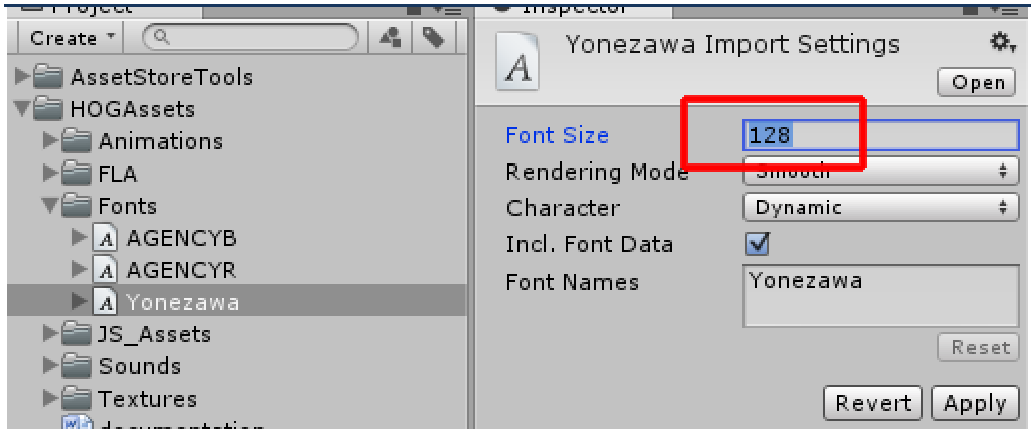
How to change font in the game?

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.



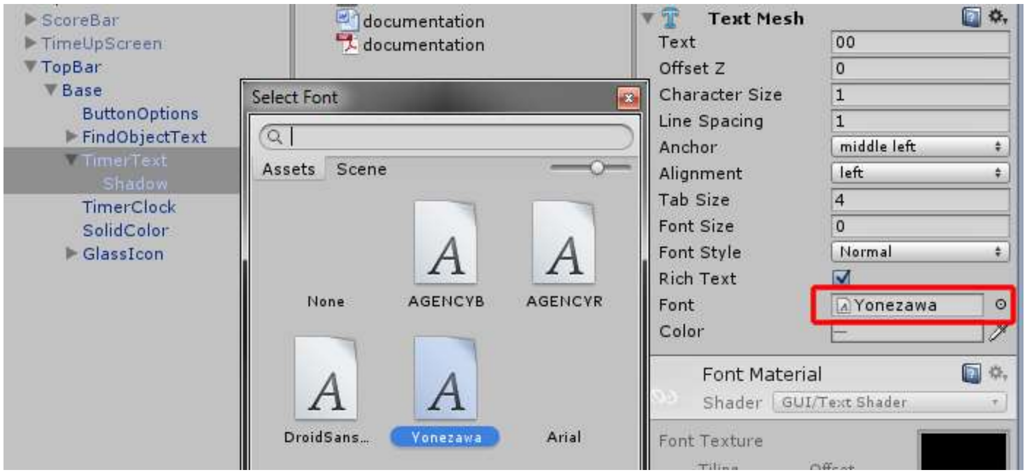
Click on the font you added and edit its attributes. I personally set all my fonts to a high number (and then scale the text object down) so that they look crisper in-game.



Yonezawa Import Settings

Font Size: 128
Rendering Mode: Smooth
Character: Dynamic
Incl. Font Data: ☑
Font Names: Yonezawa

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.



Text Mesh

Text: 00
Offset Z: 0
Character Size: 1
Line Spacing: 1
Anchor: middle left
Alignment: left
Tab Size: 4
Font Size: 0
Font Style: Normal
Rich Text: ☑
Font: Yonezawa
Color:

Font Material
Shader: GUI/Text Shader
Font Texture

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!