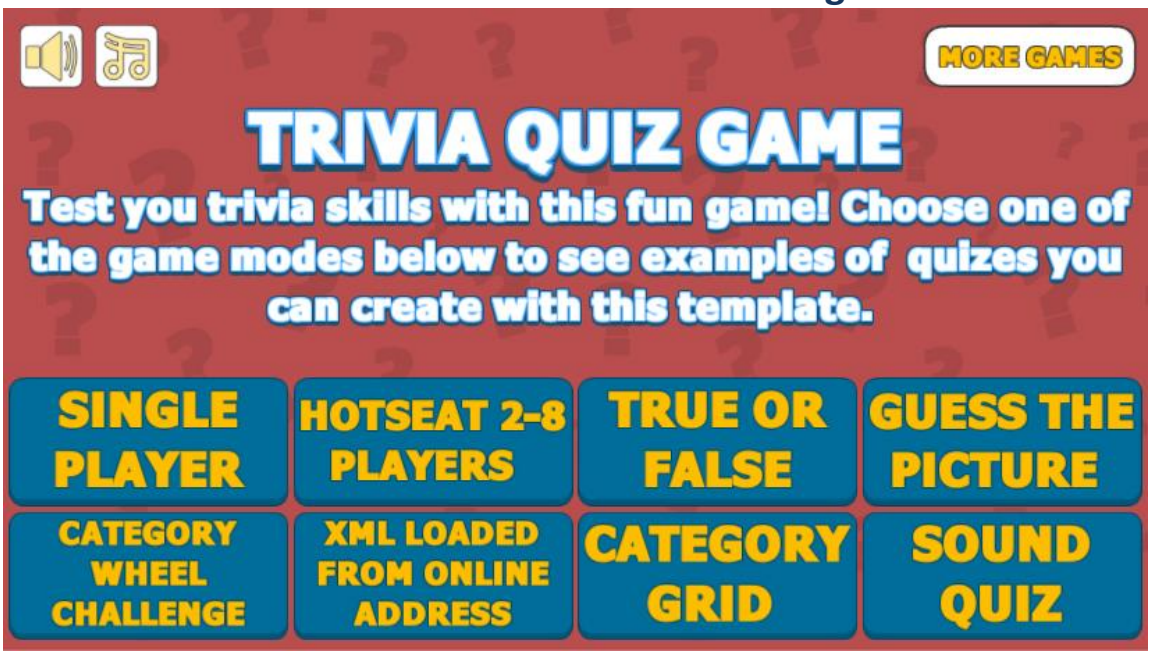
Trivia Quiz Game Template

Game documentation and HowTo guide.



This document contains:

Package Description and features

Trivia Quiz Game is a full Unity template that gives you a quick and easy way to create a trivia game, along with a timer, lives, varying bonuses, and some nice audio-visual feedback.

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.
- Includes UnityAds support with integration guide.

Current version 1.65

Update history

1.65 (10.08.2016)
Game Modes:
- Added game mode example for a math quiz with simple arithmetic questions.

Improvements:
- You can set a global time value, which will override any time set locally per question. The global timer does not reset between questions like the local timer.
- Option to lose timer when answering incorrectly. This fits well with the global timer feature.
- Option to add bonus time when answering correctly, which runs only if we use the global timer.
- You can attach a specialized Math Quiz component which will fill up any quiz with math related questions.
- Removed button definition for ShowLargerImage() function, it is now auto assigned in the code.
- Continue button is now automatically assigned to the question box when showing a followup text, so there is no need to add a continue button manually.

Fixes:
- Error when trying to use video in WebGL (Video is not supported in WebGL).
- When you set the address in a Dynamic XML component, the changes are not saved sometimes.

1.59 (22.07.2016)
- Hotfix: When switching to a mobile platform (Android, iOS), an error message appears regarding video support (Video is not supported on mobile). Also a warning related to SceneManager in some versions of Unity.

1.58 (17.07.2016)

Game Modes:
- Added game mode example for a quiz loaded from a local XML file (on your computer).

Improvements:
- A new Dynamic XML component allows you to load questions from an online or local (from computer) address. If no questions are loaded, the quiz reverts back to the built-in question list in the gamecontroller.
- The XML menu allows you to also append questions to an existing quiz, and even export questions from a quiz to an XML file.
- You can override a quiz by attaching a category component to the gamecontroller. The gamecontroller will take the questions from the category component.
- Sound questions are added to the XML format list, so you can import them too.

Fixes:
- If the player finishes all questions in a quiz before getting to question limit, victory screen activates twice.

1.54 (03.07.2016)

Game Modes:
- Added game mode example for sound-based questions. Listen to a sound and answer.
- Added game mode example for a category grid. Choose a category and play it.

Improvements:
- You can now assign sounds to questions. A question cannot have a sound and a video/image at the same time.
- Share by mail: You can send your score to a predetermined email address. This is good for instructors who want to quiz their students and get the results directly to their mail.
- Option to have images in the answers too, also through XML.
- You can define a score target to win. Reach this score to win the quiz, regardless of how many questions you answered.

Fixes:
- Event system was being referenced before it was assigned, causing problems for the Android build.

1.51 (04.06.2016)

Game Modes:
- Added another version of the standard game mode, with the UI elements aligned to fit a portrait mode (vertical).

Improvements:
- Option to toggle whether we show the correct answer or not at the end of a question. (When you choose an answer, the correct one is highlighted).
- Added Dynamic Grid script which lets you set grid sizes for both horizontal and vertical aspect ratios.
- Switch the sound source to an independent prefab similar to how the music source is set up.
- Lives Object can also display text. If you display text you can’t use the lives bar.
- Improved support for Unity version 5.4 and above.
- Improved UnityAds support with new integration video.

Fixes:
- In HotSeat mode, the game skips the next player after making 2 mistakes.

- If the game starts while the category wheel is enabled, sometimes it shows a one-time null reference.
- When using a gamepad, if you press the button on an image to enlarge it, you press the button again and the enlarged image doesn't close.
- In Unity 5 the last selected button stays highlighted after displaying the next question.
- Some of the UI animations are missing.

1.43 (22.02.2016)

New Game Modes:

- Category Wheel Challenge: 6 categories, 5 questions each, pass 3 random categories successfully to win the challenge.
-True/False: 10 questions with progress tabs and a horizontal timer bar.
- Guess the picture**:** Guess the zoomed-in picture of an animal, then read a follow-up about it.

Options:

- Options to randomize the question list, sort it from low to high bonus, and randomize the answers too. This affects how questions are presented.
- You can set the number of the first question. This can be used to start from a higher question number, for example if you want to start from a higher difficulty in the question list (if we have it sorted by bonus groups from easy to difficult).
- You can set the total number of questions asked, regardless of whether we answer correctly or not. This is used when you want a strict number of questions to be asked, otherwise the game will try to go through all available question groups.
- You can assign multiple categories (or question lists) to the quiz, and switch between them with the Category Selector.

Game Elements:

- Added categories which are defined by Name, Color, Icon, and a list of questions associated with it.
- Added a category wheel which chooses a random list of questions for the quiz.
- Added Progress Object which shows the current question we are on and the total number of questions before we finish. Has a tab mode and a text mode.
- Added follow-up text to the quiz, which appears after you answer a question.
- Added several more tab graphics, round, blurry, straight, etc.

Fixes:

- Improved Gamepad and Amazon TV controller support. Credits to Jason Bentley for testing that out. Also gamepad buttons are more responsive now, and you can select and zoom into images/videos with the gamepad.
- Added WebGL to MovieTexture (video) exception list. Not sure about this one yet, but I think WebGL can't play video.
- Fixed Image/Video/Text order when displaying a question.
- Reorganized UI screens (victory, defeat, results, player selection, etc.) to better fit both landscape and portrait modes.
- Minor changes and bug fixes.

1.32 (25.01.2016)

- Expanded the HotSeat mode to allow for 2-8 players (and potentially more). Added a slider to choose number of players before starting. Current player is now highlighted. Improved end of game results screen with a highscore table and an icon for the winners.

- Added the option to load text-based quizzes from an online web address. This allows you to change the quiz frequently without having to upload the game again.
- Added the option to load the questions from another quiz by simply assigning it in the current quiz gamecontroller.
- Redesigned the timer object, you no longer need to assign it in the gamecontroller, and it has the option for a text timer too.
- The list of players now includes colors, which will be updated in the game screen and in the results screen for the HotSeat mode.
- Added support for UnityAds along with an integration guide. This will allow you to monetize your game with fullscreen video advertisements.
- Minor changes and bug fixes.

1.26 (22.12.2015)

- Fix an error when trying to build for mobile using video questions. **Unity does not support video playback on mobile.**
- Added warning when importing video question while on a mobile platform.
- Removed example videos from package to avoid potential conflict. The package still supports video questions for Unity 4 pro and Unity 5.

1.24 (12.12.2015)

- Added support for video based questions. You can now assign a video as a question, with or without a text caption. The video can also be enlarged to take a better look at it. Video questions are supported by Unity 4 pro, Unity 5 free and Unity 5 pro. **Unity 4 free does not support video.**
- Added support for Unity 5.3 SceneManager.
- Minor changes and fixes.

1.2 (14.11.2015)

- Added the ability to load a list of questions and answers from an external XML file using the Tools menu in the Unity editor. XML example provided with documentation.
- Reorganized the format of the answers, allowing you to set more than one correct answer. **The new format is not compatible with older formats, so you won't retain the answers from previous versions.**
- Minor changes and fixes.

1.14 (01.11.2015)

- Added a new local multiplayer game mode, hot-seat, which allows you to play in against several players, answering questions in turn for top points. Also added a screen to choose how many players want to participate in the match.
- Reorganized the GameController and added comments on each component element. Hover over the value in the component to read the comment.
- Fixed double tap bug (Thanks to Miguel Paolino for the find!).
- Minor changes and fixes.

1.09 (19.10.2015)

- Support for image based questions. You can now assign an image as a question, with or without a text caption. The image can also be enlarged to take a better look at it.

1.06 (29.09.2015)

- Support for Keyboard and Gamepad controls, detected automatically.
- Support for dynamic answer numbers (2 answers, 3 answers, etc in the same trivia level)

1.0 (06.09.2015)
- Initial version

Credits

The font used is Fava Black by Themnific

The sounds are courtesy of the free sound project.

Music is River Meditation by Jason Shaw (Public Domain)

Credits go to these authors for their great sound samples: **xyzr-kx, isaac200000, harris85, speedygonzo, wagna, jimhancock, sforsman**

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 TQGAssets folder using Unity 4.6.8 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, Icons, 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. Win, Lose, etc
- **Textures:** Holds all the textures used in the game which are used as sprites in Unity.

Getting started

Trivia Quiz Game Template (TQG) 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 TQG 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 questions for the player, and keeps track of answers, timer, and lives. The Game Controller is also used to calculate the bonus the player gets when answering.

Click here to read more about Players.

Players – Holds a list of the players in the game. Each player can be assigned a name, a score text, lives and lives bar. You must have at least one player in the list in order to play the game. You don't need to assign all fields. For example, a player may have a name with no lives bar and it will work fine.

Question Object – Holds the question object. The question object displays the text of a question. This is assigned from the scene.

Answer Objects – Holds all the answer objects. The answer objects display the text of an answer. These are also assigned from the scene.

Questions – Holds all the possible questions in the level. Each question has a correct answer and several wrong answers, as well as a bonus value and time. Click here to read more about how questions are made.

Questions per Group – The minimum number of question to be asked from this bonus group. Click here to read more about how questions are made.

Maximum Mistakes – How many mistakes we can make before losing a life.

Lives – The number of lives we have left. If lives reach 0, it’s game over.

Lives Bar – The lives bar that displays how many lives we have left (The heart icons). This is assigned from the scene.

Lives Bar Width – The width of a single life in the lives bar (A single heart).

Bonus Object – The bonus object that displays our potential bonus value. This is displayed above the score text, and is assigned from the scene.

Bonus Loss – The percentage we lose from our potential bonus if we answer a question wrongly. For example 0.5 makes us lose half the bonus if we answer wrongly once, and ¾ of the bonus if we answer twice incorrectly.

Score/ Score Text – The score we currently have, and the text displaying it. The text is assigned from the scene.

Canvases – These are canvas UI screens. **Timer Canvas –** Shows the current time and the progress to 0. **Game Over Canvas** and **Victory Canvas** appear at the end of the game when we lose or win.

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

Animations – Holds a list of animations for the question and answers.

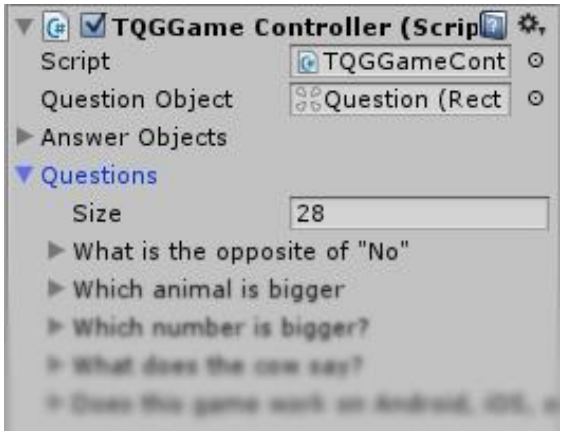
Sounds – Various sounds for leveling up, losing a life, and game over.

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

Confirm & Cancel Buttons – 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.

Questions & Answers

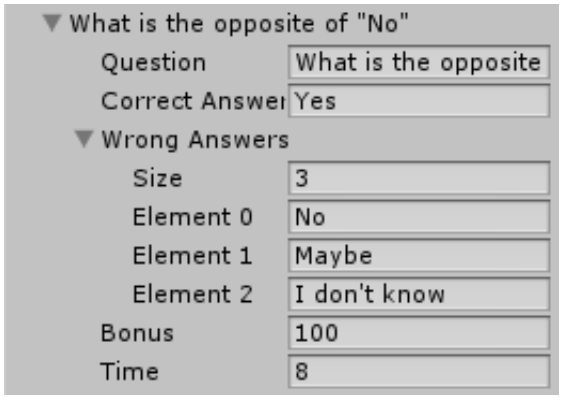
The most important part of this package is of course the question list. Let's take a look at it and see how we can modify it. Click on the game controller and take a look at the component inspector.



The questions list holds all the possible questions in the game. You can have as many questions as you like in this list. At the start of the game the list is shuffled, and then sorted based on the bonus value of the question. This results in a list of questions that starts with the low bonuses and rises up to higher bonuses as you progress through the game. Based on this logic your game design should take into consideration the difficulty of the question you want to ask and then assign a proper bonus to it. For example, a very simple question would have a bonus of 100, while a more

difficult one could have a bonus of 5000. This will ensure a nice flow to your game progress from easy to hard.

Now let's see what a single question is made of. Click on the first question in the list to view it.



First we wrote the text of the question, and the correct answer for it. Next we wrote the wrong answers. In total we have 4 possible answers that correspond with the 4 answer objects we have.

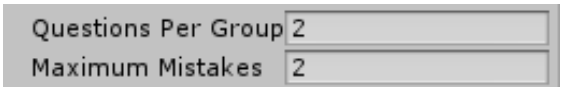
Finally we wrote the bonus and time we have for this question.

Questions per Group

You can control the length of your game by changing the value of Questions per Group. This value decides how many questions of the same bonus group will be asked (at least). For example if you set it to 2, the game will choose 2 random questions from each bonus group and then move on to the next higher bonus group, so you'll get 2 questions for 100 bonus, and then 2 questions for 250 bonus, and so on.

If you answer the question incorrectly, you will not pass to the next group, but another random question from the same group will be chosen. So in the case above if we answer the first question correctly, and then answer the second question incorrectly in the 100 bonus group, we will get another question from the same group.

If all the questions in a group are asked, the game will move on to the next bonus group regardless of whether we answered correct or wrong.



Mistakes

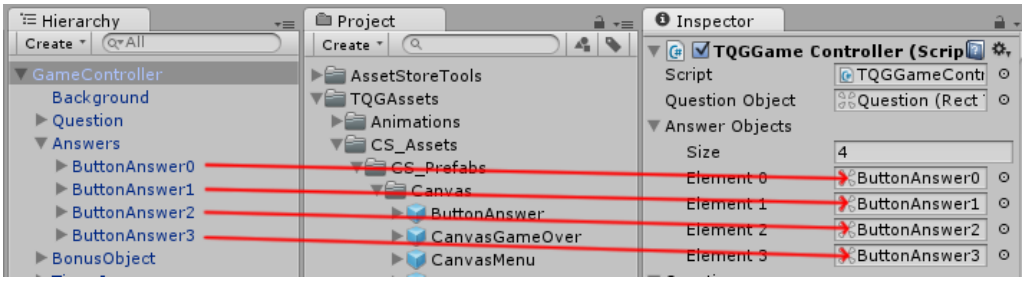
Mistakes are another way to control the difficulty of the game. By default they are set to 2, which means that if we answer incorrectly once, we get a chance

to answer again. If we make 2 mistakes we lose the question bonus and move on to the next question.

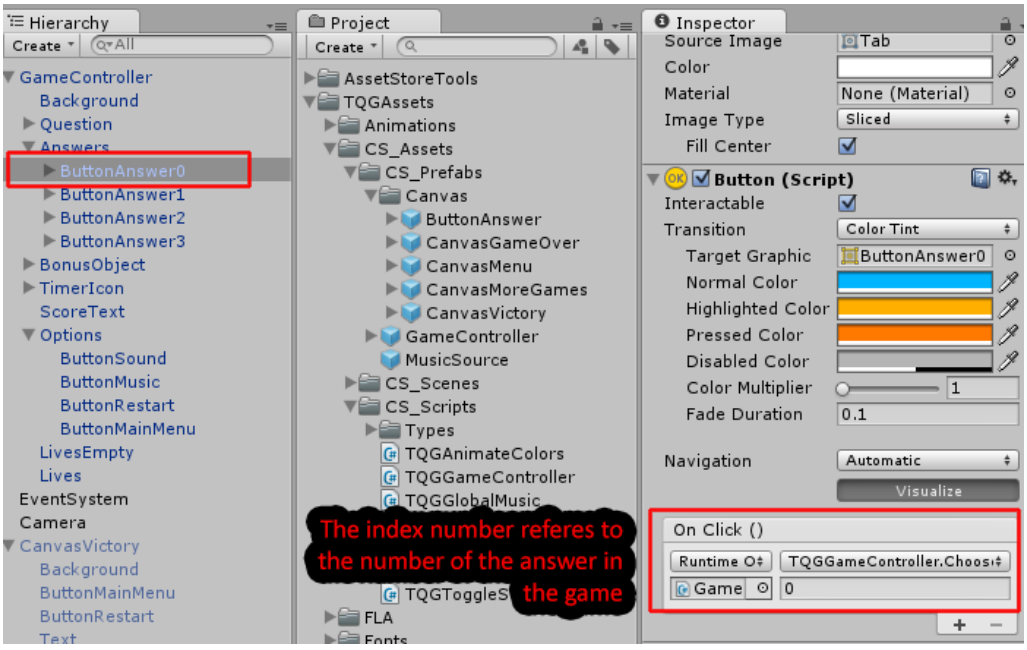
Dynamic answer count

In this package you can also have dynamic answer counts. For example you can have a question with 4 answer, and another question with 2 or 3 answers, all in the same level.

In order to use this feature, first we need to setup the **maximum** number of possible answers and assign them in the game controller. Select the game controller in your scene and create the number of answer buttons you would like to use. This will be the maximum number of answers in the level.



Now select each of the button answer objects and set the index number correctly in the OnClick function. This will be the reference in the game to the number of the answer.



So for example **ButtonAnswer0** has the index 0, and **ButtonAnswer1** has the index 1, so in the game when you click **ButtonAnswer0** the first answer is chosen and when you click **ButtonAnswer1** the second question is chosen.

Finally, in the game controller you can set any number of answers to each of your questions **as long as you don't set more than the maximum number of answers. In our case we set up 4 answers so we can't have more than 4.**

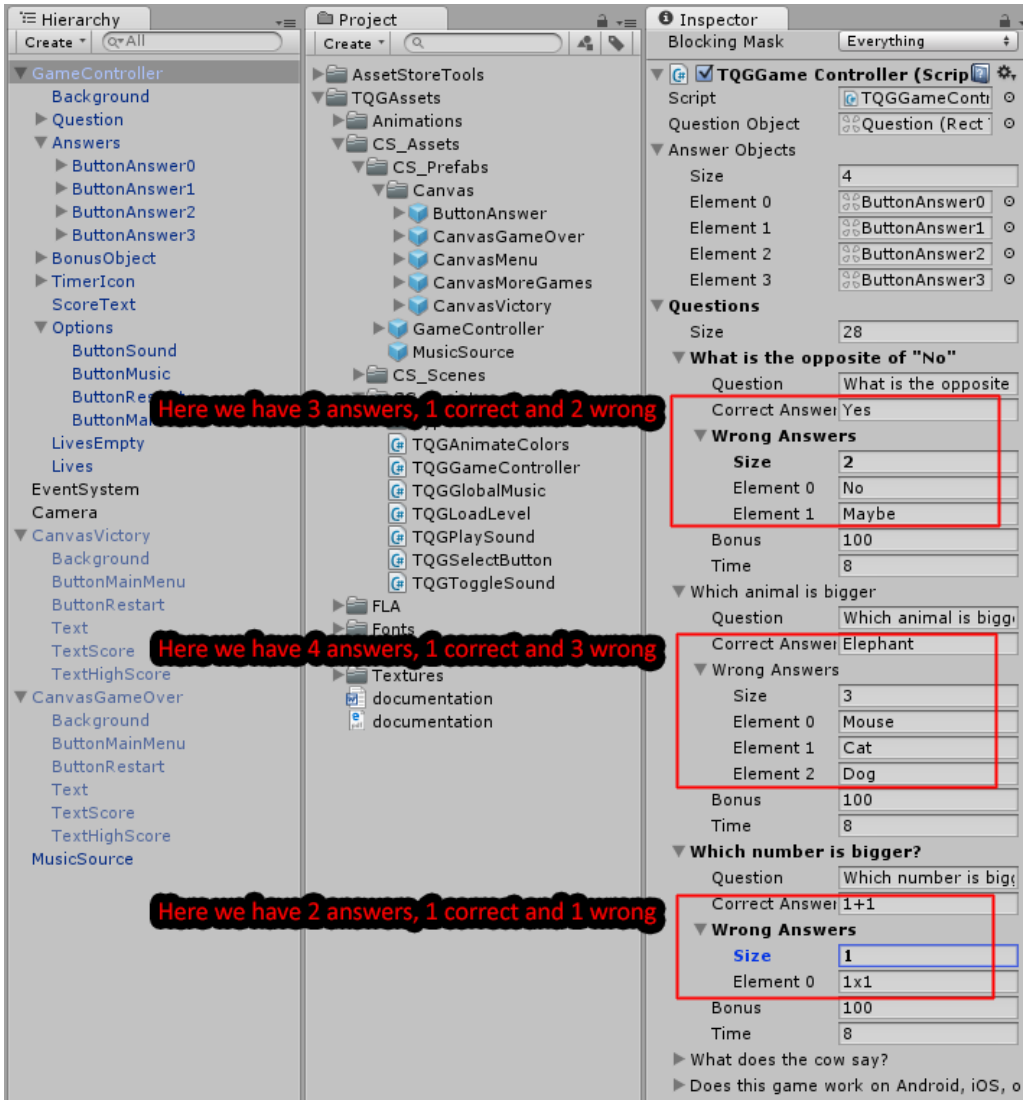
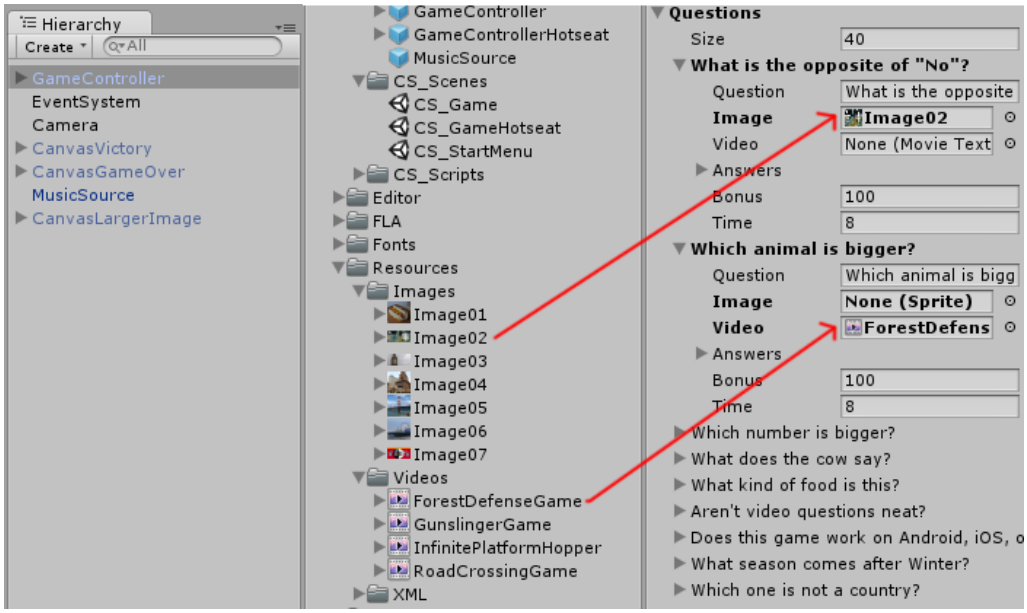


Image and Video Based Questions

Since update 1.09 you can also add an image to the questions. You can have questions with text, images, or both. To assign an image to a question simply drag it to the Image slot in the game controller.



Update 1.24 also added support for video questions, similar to how images are used. If you assign a video to a question, it will take priority over an image assigned to the same question. You cannot have an image and a video assigned to the same question. The free version of Unity 4 does not support video playback, so you need to have Unity 4 Pro, or Unity 5 (free and pro) in order to use video questions. Unity does not support video questions on mobile devices (iPhone/iOS, Android, etc)

The chosen image/video is masked by the question object, and the text appears above it. If you click on the question tab, the image/video is enlarged to its full size. Another click on the enlarged image/video will remove it.

Importing Questions from XML

Since update 1.2 you can also import the list of questions and answers from an XML file. The default format of the XML file is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<data-set xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <record>
    <Question>Which animal is this?</Question>
    <Image>Cat</Image>
    <Video>Cat</Video>
    <Answers>
      <Answer correct="true">Cat</Answer>
      <Answer correct="false">Lion</Answer>
      <Answer correct="false">Cheetah</Answer>
    </Answers>
    <Followup>The cat is a small domesticated carnivorous mammal with soft fur, a sho
  </record>
```

You should generally follow this format when creating your questions, but you can change some of the elements based on your needs; For example you can have 4 answers for a question, and then 3 answers for another question. Each question has an attribute to choose if it's correct or not (true/false), so a question may have more than one correct answer if you want to.

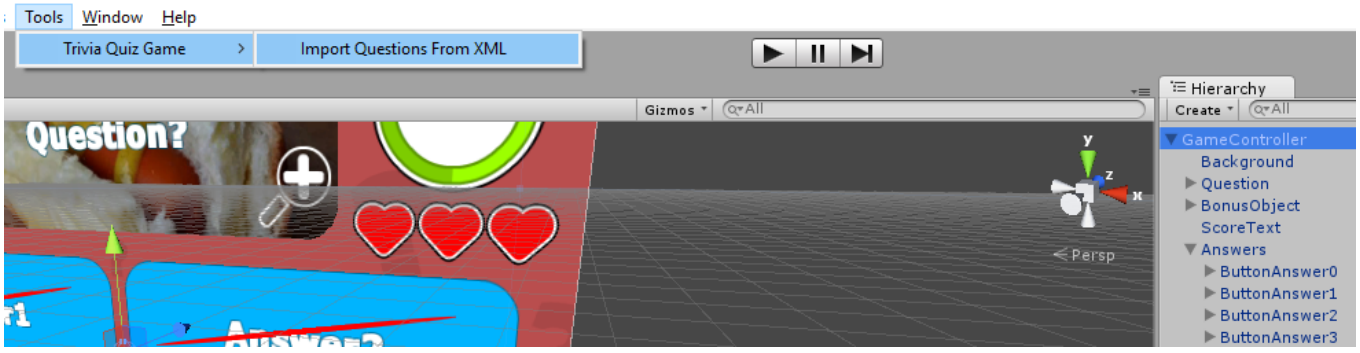
You can also remove some of the fields you will not be using, for example if your quiz does not contain any images or videos you can simply omit them from the format and unity will ignore those when importing.

Notice that the number of answers for each question should not be more than the number of Answer Objects you have in the game. If you try to import an XML with extra answers you'll get a warning asking you to increase the number of Answer Objects and assign them in the game controller. In a future update this will be automated, allowing you to choose one Answers Object and the answers will be created dynamically based on the number of answers loaded.

(Read here how you can set Answer Objects in the game controller)

You can also choose an image for each question. The images need to be stored in the Unity project at **Resources > Images**, like in the screenshot below. The name of the image you want to include should be written without the extension (png,jpg), for example <Image>**Image01**<Image> is correct, while <Image>**Image01.png**<Image> is wrong.

To import a list of questions, simply select an object which can have questions (any object that contains a TQGGameController component or a Category component), and then choose from the top Unity menu **Tools > Trivia Quiz Game > Import**

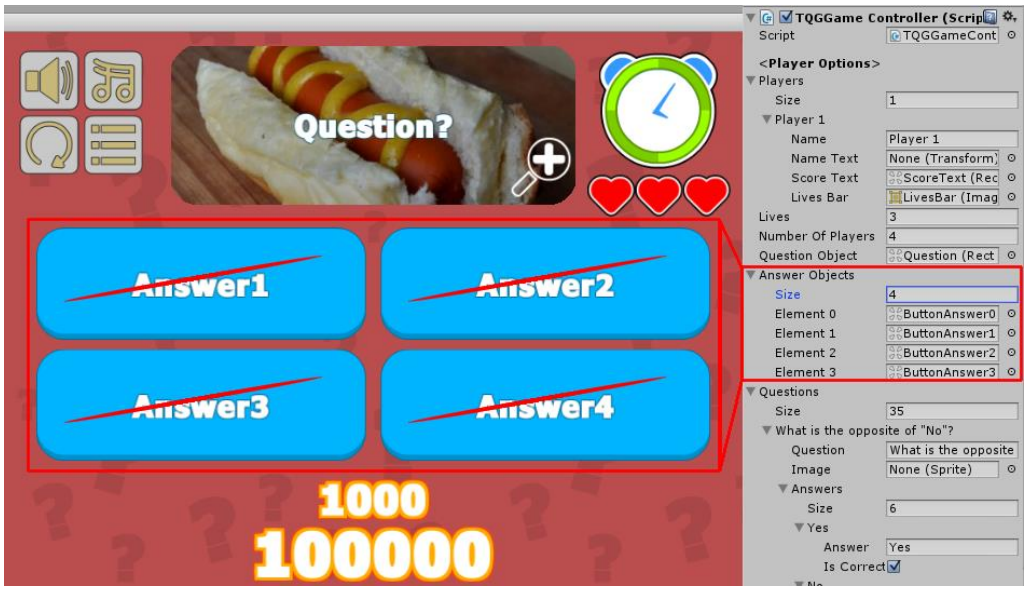


Now choose the XML file from your computer and open it. If everything worked as intended you should see the list of questions in the game controller.

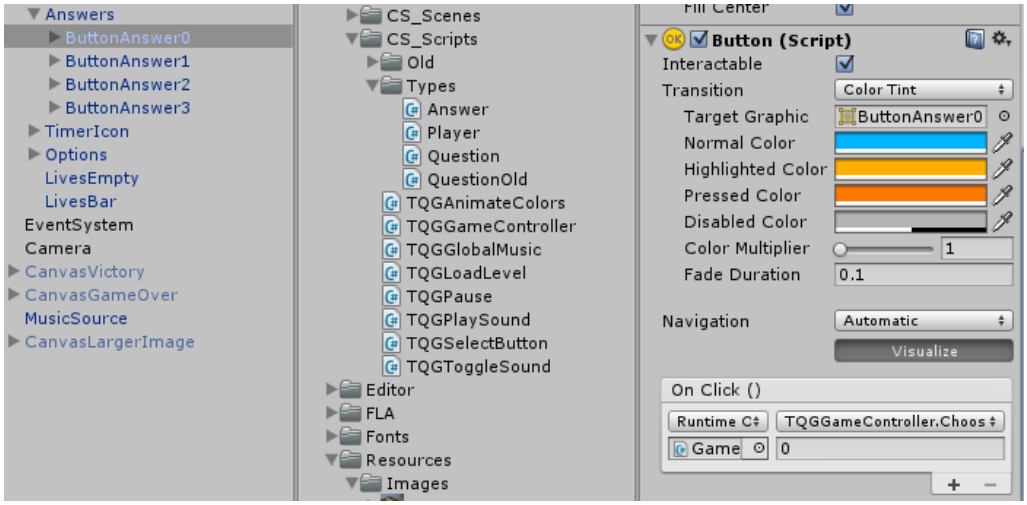
That's it!

Answer Objects

Answer Objects are the buttons the display the answers for each question, and that you can click no to choose an answer.



Each answer object has a button function in it with a reference to the index of the answer we want to choose when clicking the button. For example **ButtonAnswer0** chooses the answer with index 0 (the first answer).

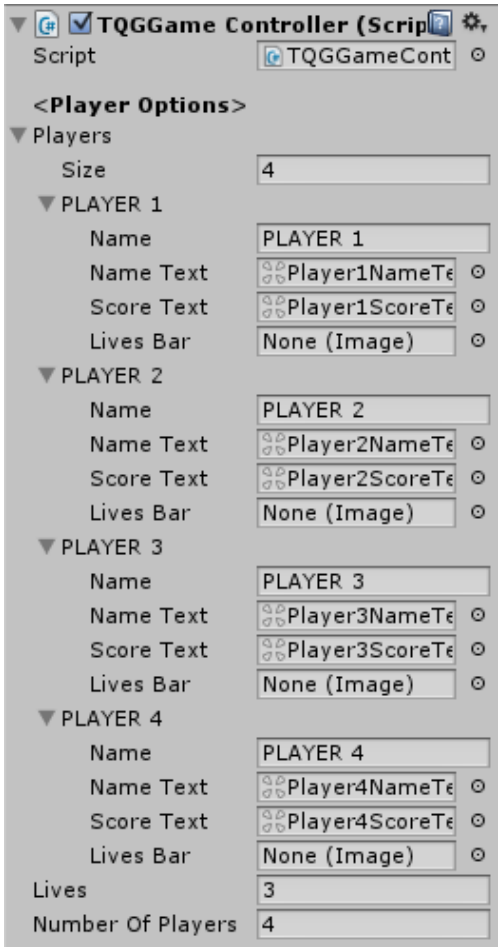


In order to increase the number of Answer Objects, first increase the list of Answer Objects from 4 to 5, and then duplicate one of the answer, and set the index to 4 (which will reference answer number 5).


We increased the list to 5
We duplicated the button object
We set the index to the answer we want to choose
This is the newly duplicated Answer Object
Finally we assigned the new button to the list of buttons by dragging &

Players

You must have at least one player defined in the game controller in order to play the game. You can also have more than one player in the game, which allows you to make hot-seat types of games as well as single player. Each player has a name, score, and lives. Each of these also has a text object or bar that displays the corresponding value for the player.



Tip: You don't have to assign each object for the player. If you don't want to display a name, just don't assign it for the player, and if you don't want to have lives in your game just don't assign them.

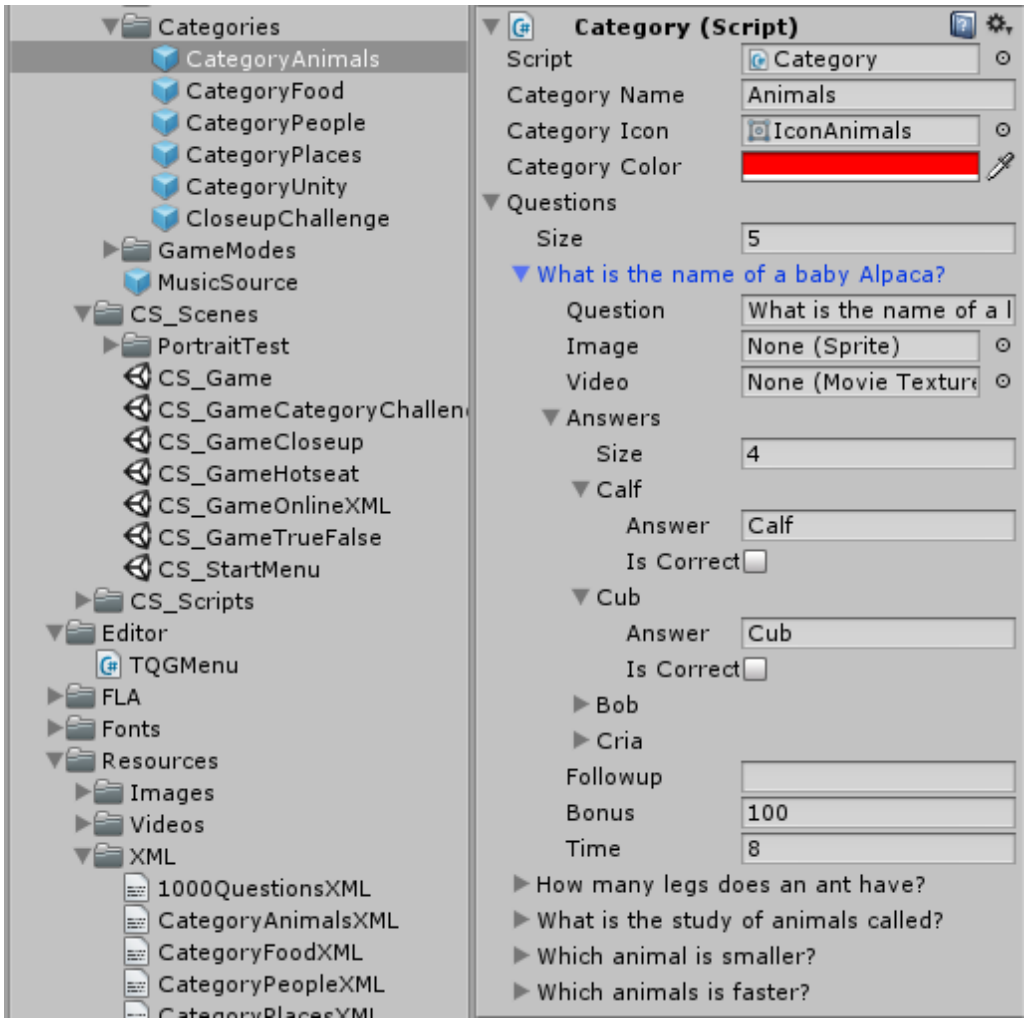
One of the examples provided in the package uses a list of 4 players to create a hot-seat game mode. Each of these players has the name and score assigned to it, but no lives bar.

We also added a special start screen in the game that allows the player to choose between 2 to 4 players to play the hot-seat mode, so even though we defined 4 players we can still allow less players to participate in a match.

Categories

Categories are a new addition to the package introduced in update 1.32. They can be used to display and choose question lists in the game without being attached to the same questions in the quiz.

Each category has a name, a color, an icon, and a list of questions associated with it:

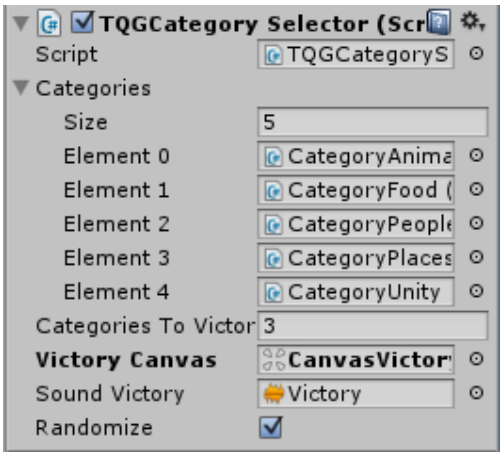


As you can notice above, the question list is identical to that in the gamecontroller question list.

You can also import questions to a category from XML, similar to how questions can be imported into the gamecontroller question list.

Category Selector

The category selector is a component used to display a list of categories and allow the player to choose them or switch between them during a multi-category game.



Here you can add all the category objects you want to use. Each category is a prefab which contains a list of questions, a name, a color, and an icon.

You can also set the number of categories required to win a game, and the victory screen that appears when you pass enough categories. You can also randomize the categories.

One of the implementations of the Category Selector is in the form of a Category Wheel, which is used as part of the Category Wheel Challenge. In this game mode, the categories are displayed in wheel, and chosen randomly. If the player passes enough categories, the game is won.

(Read here about the Category Wheel Challenge game mode)

In a future update the Category Selector will also be able to be used as a grid from which you can choose a category to play in the quiz. This will be another example mode in the package.

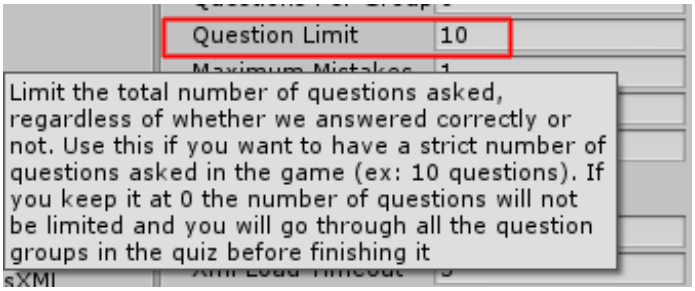
Game Modes

Several new game mode examples have been added to the package in update 1.32, each of them presents several unique features which can make your game much more engaging to players. Although

True False

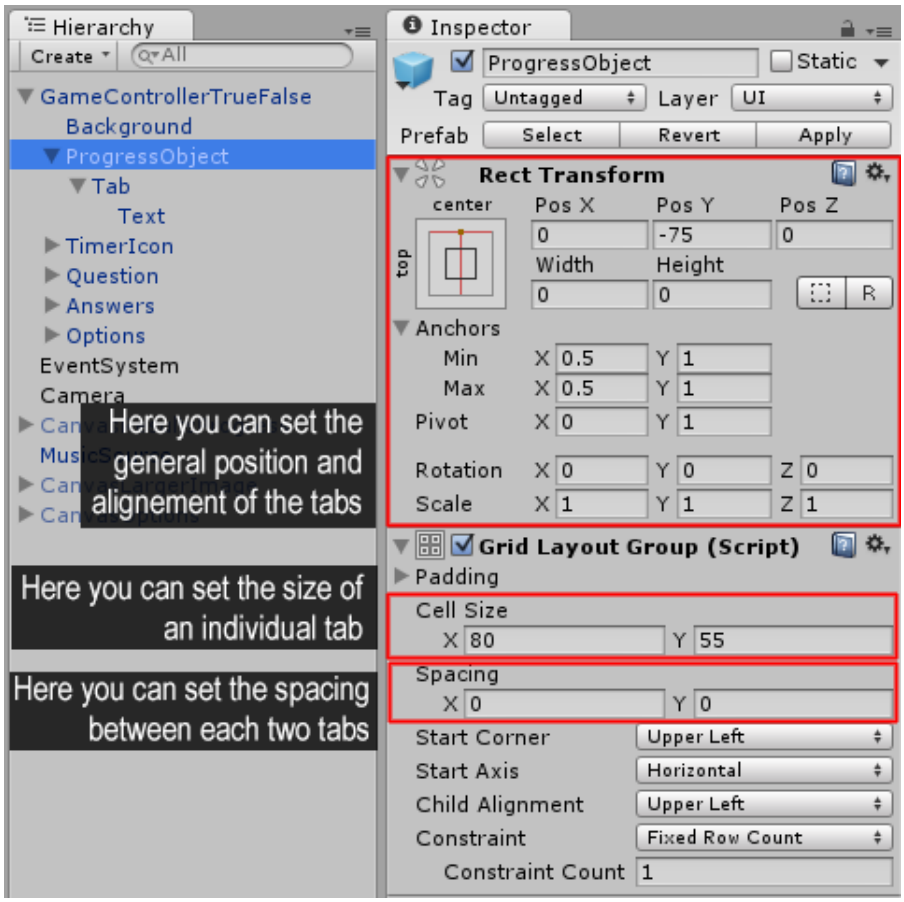
This game mode presents the player with 10 random questions from a larger pool of questions, to which the answer is either “true” or “false”. Each question you answer is marked as correct or false in the progress tabs above, and the timer is a horizontal bar instead of the usual circular clock with text.

The questions are limited using the **Question Limit** value, which can be set in the game controller. Read about it in the comment box below.



Another unique feature of this game mode is the progress tabs. These show you the current question being asked, and the state of previously asked questions (green if we answered correctly, and red if we answered incorrectly).

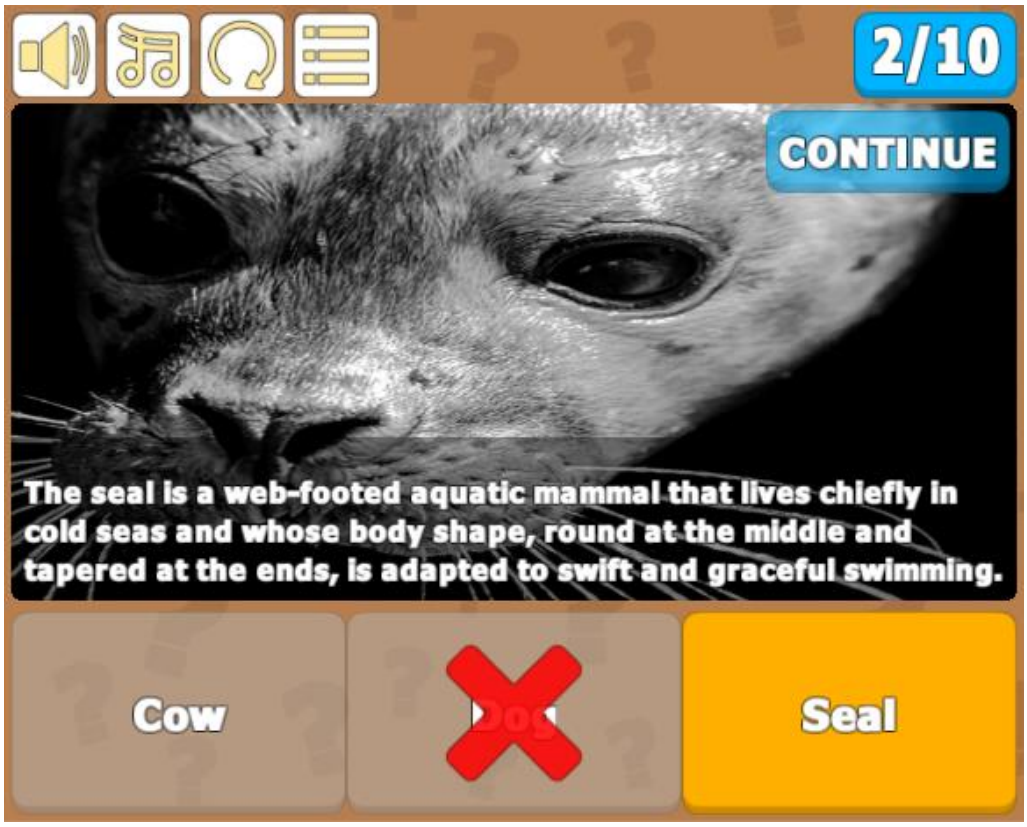
You can edit the way tabs are displayed by changing the values in the **Progress Object** in the game controller.



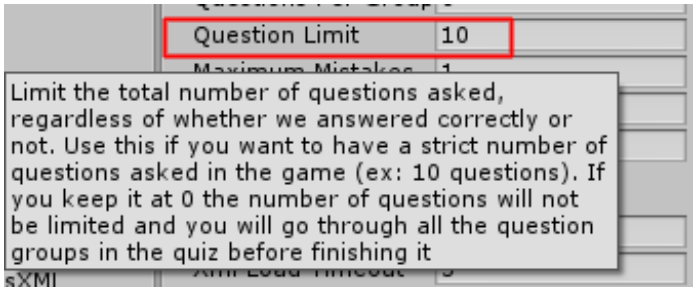
You can also edit the shape of an individual tab by editing the **Tab** object and the **Text** object inside it.

Guess the Picture

This game mode presents the player with 10 zoomed in images and 3 possible answers for each. You must guess which animal the image belongs to. After you answer, the full image is revealed and some follow-up text about the answer is presented. Progress is presented in text form above the image.

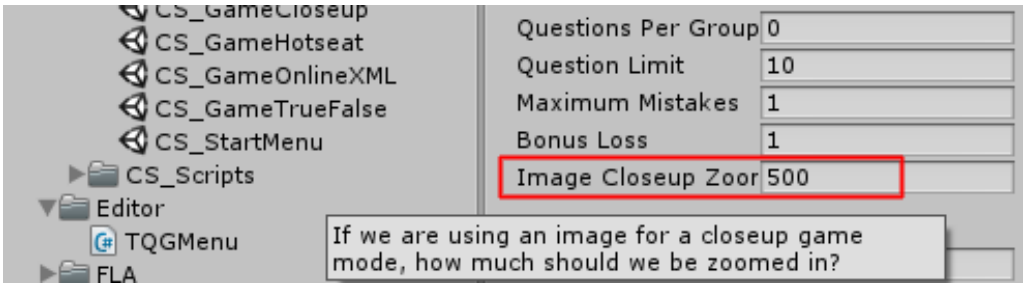


The questions are limited using the **Question Limit** value, which can be set in the game controller. Read about it in the comment box below.

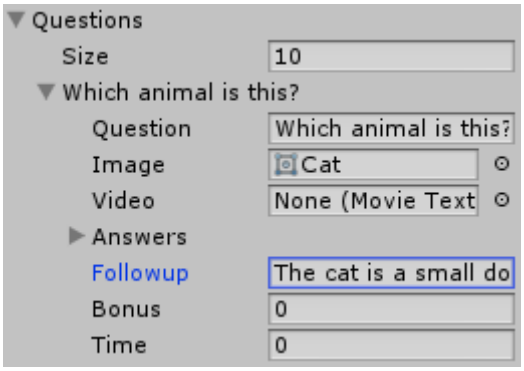


The progress tab is a simple text object which shows you the current question being asked, and the total number of questions in the quiz. You can edit the shape of the tab by editing the **ProgressObject** and the **Text** object inside it.

The unique feature of this game mode is the zoomed-in image, and the follow-up text which appears after you answer. You can set the zoom value in the gamecontroller.



Another feature of this mode is the follow-up text. This is displayed instead of the question text after we answer. You can set a follow-up for each question in the quiz.



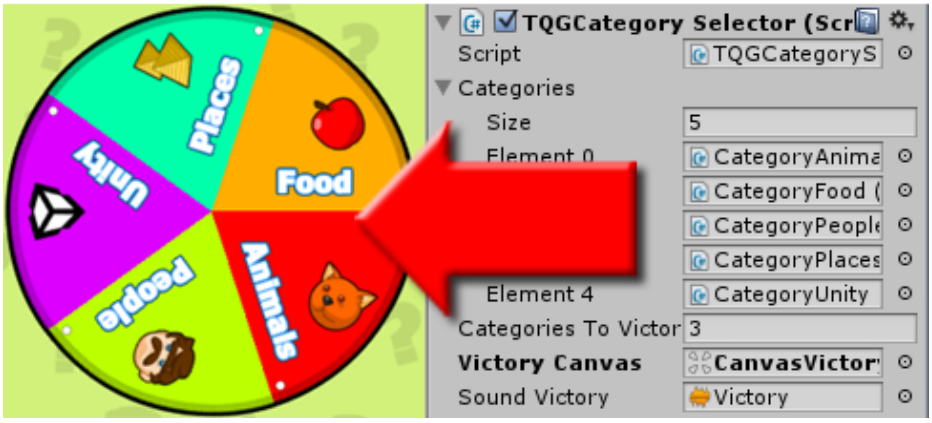
When a follow-up text is displayed, the game is paused. The only way to continue to the next question is using the “continue” button at the top of the image. If your game doesn’t include a continue button your game will be stuck.

In a future update I will add an option to have a time delay for displaying the follow-up and automatically continuing, as an alternative to the current “continue” system.

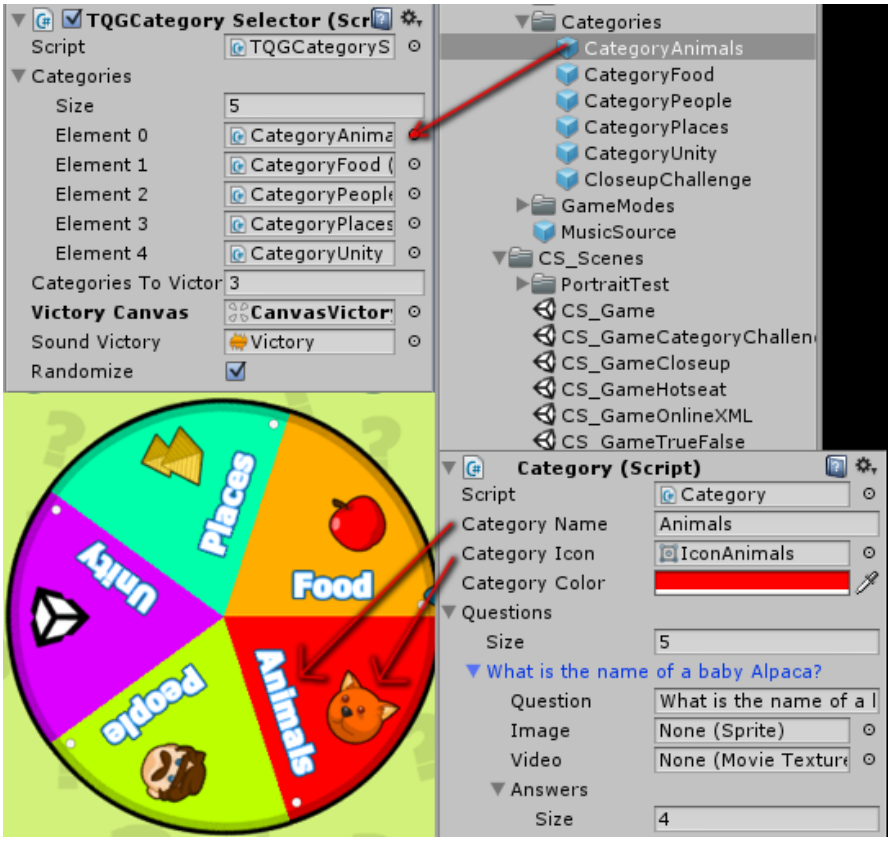
Category Wheel Challenge

This game mode gives the player a random category using a spinning wheel. After you answer 5 questions in a category you are presented again with the category wheel, and given another random category. If you answer 3 categories you win the game.

The main feature of this mode is of course the category wheel which can be filled out with a list of categories, and will automatically display them in a nice circle, along with each category's name, icon, and color.



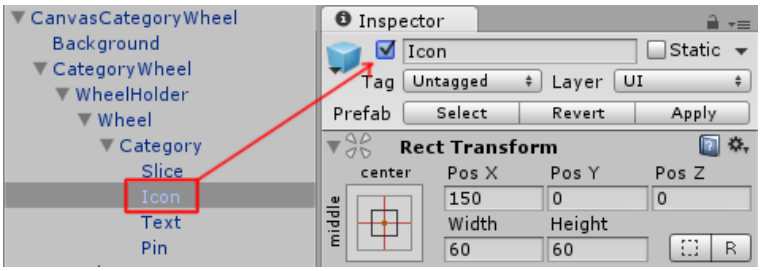
The attributes of each category affect how it looks in the wheel. In the category object you can set the name, icon, and color of the category.

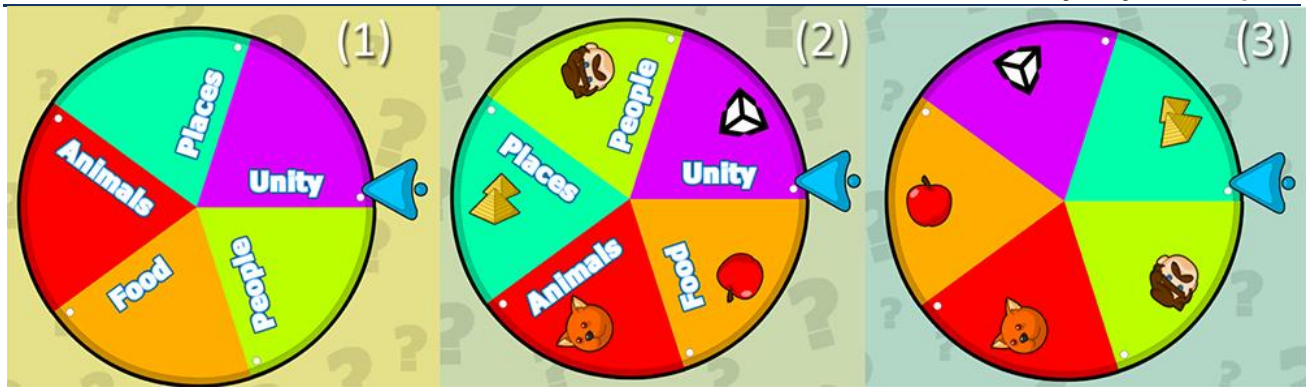


The component that allows the wheel to receive and display categories is the Category Selector.

(Click here to read about the Category Selector)

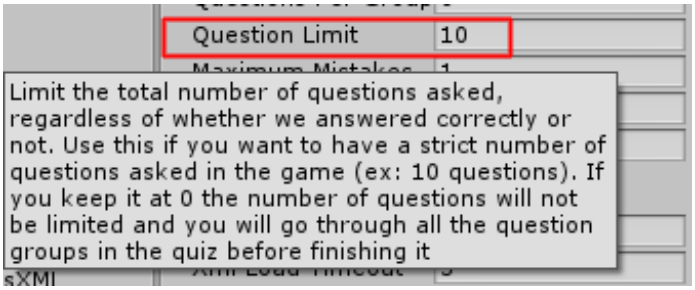
In the wheel itself you have a **Category** object which contains an **Icon** and a **Text** object. If you disable any of them, they will simply not appear in the wheel. This way you can have the wheel look like case (1) with no icons, or like case (2) with both text and icon, or you can disable the text object in the category and just have icons only like in case (3).





The quiz that is displayed for each category is similar to the regular quiz example, except for the hard limit on the number of questions asked. We limited it to 5 questions only.

The questions are limited using the **Question Limit** value, which can be set in the game controller. Read about it in the comment box below.

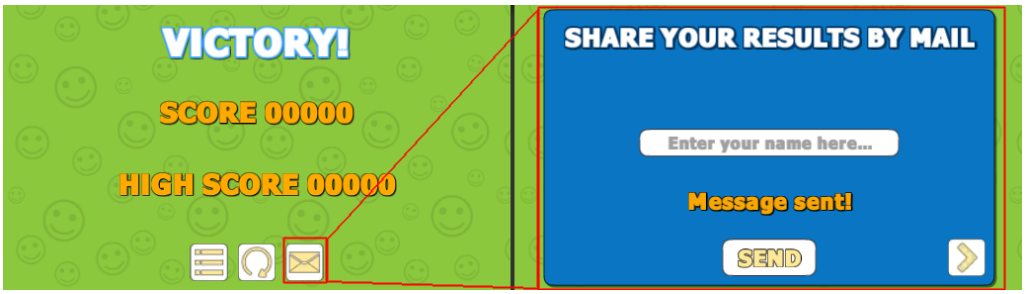


Sharing score by E-Mail

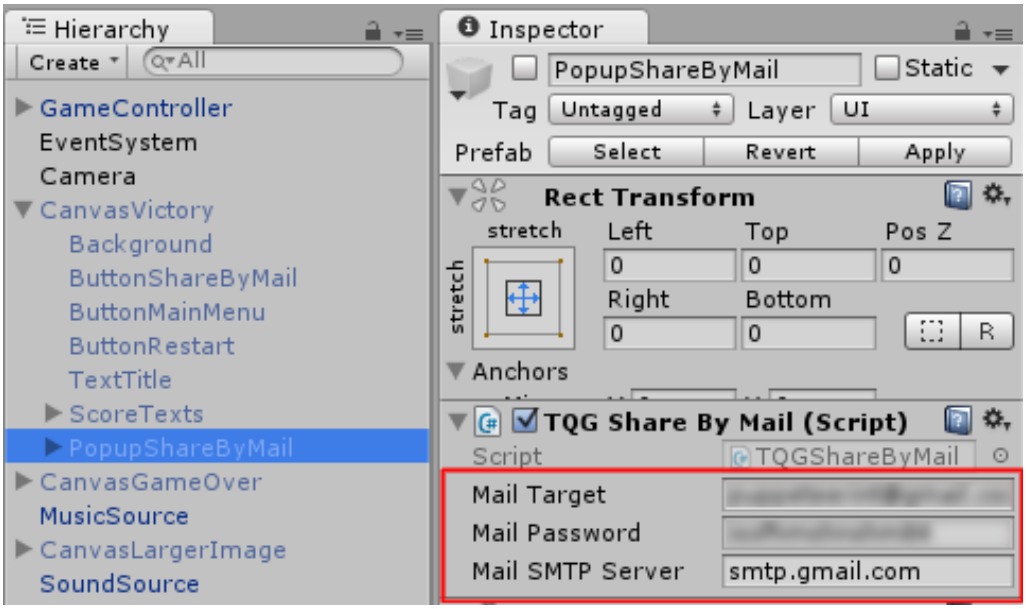
Since version 1.54 a new feature was added which allows users to send their score to a predetermined email address. This is good for instructors who want to quiz their students and get the results directly to their mail.

Limitations: Currently only works in PC, Mac, & Linux Standalone

If you look at the GameOver or Victory screens you will see a new button. Press it and a popup appears, prompting you to enter your name and send the score to the quiz maker.



As the quiz maker, you will need to assign your email address and password in order to receive the results by mail. You can do this in the component **TQGShareByMail** which is attached to the mail form popup.



Depending on the mail type (Gmail, Yahoo) you must also assign the SMTP Server name. For **Gmail** this is **smtp.gmail.com**, and for **Yahoo** it is **smtp.mail.yahoo.com.** Other emails will use other names. In future updates I'll improve the functionality of mail sharing.

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.**

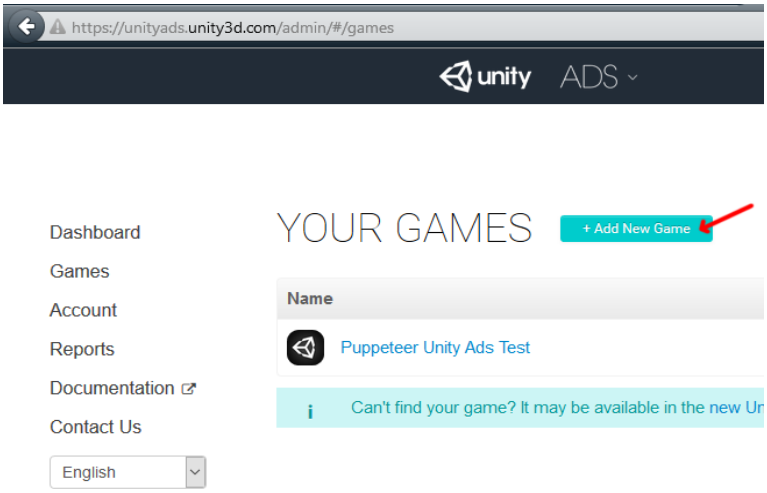
For more info about integrating UnityAds read this:

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

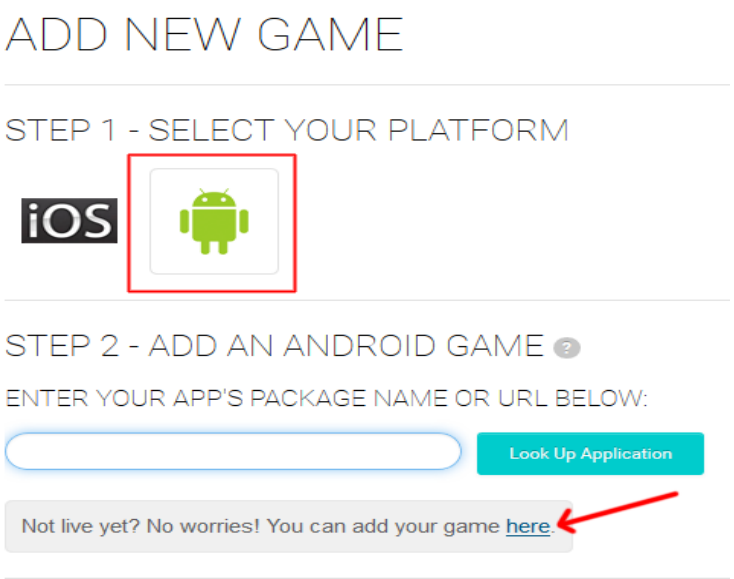
Integrating UnityAds into your project (Unity 4)

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

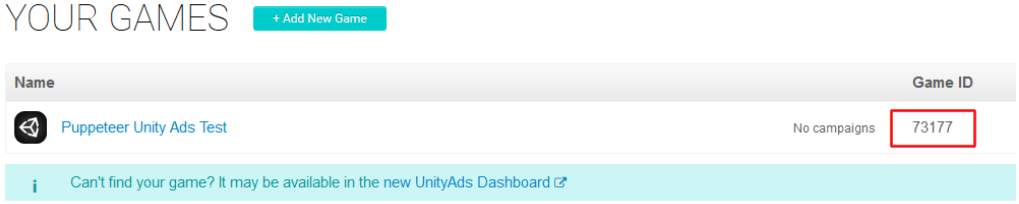
First we need to create our game entry on the UnityAds website. Go to https://unity3d.com/services/ads and create a new game. If you already have your app set and your GameID noted, just skip this part and go straight to importing the UnityAds package into the game.



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.

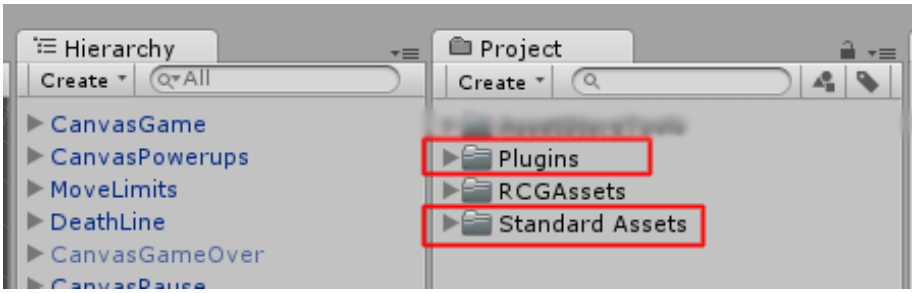


In Unity Editor

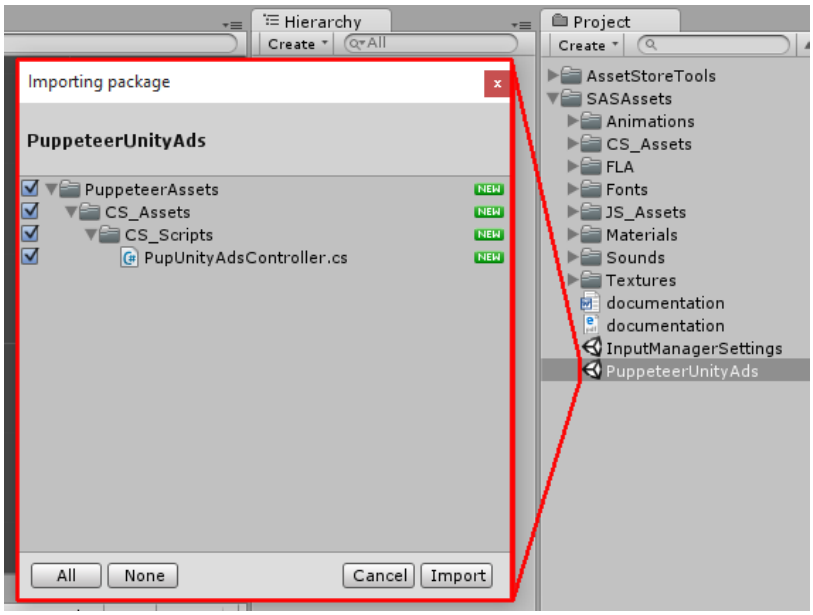
Now we need to import the UnityAds package. Open the Unity Asset Store and download the UnityAds package. Import it into your project.

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

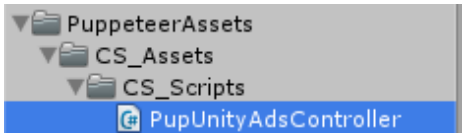
After import you should have two additional folders in your project.

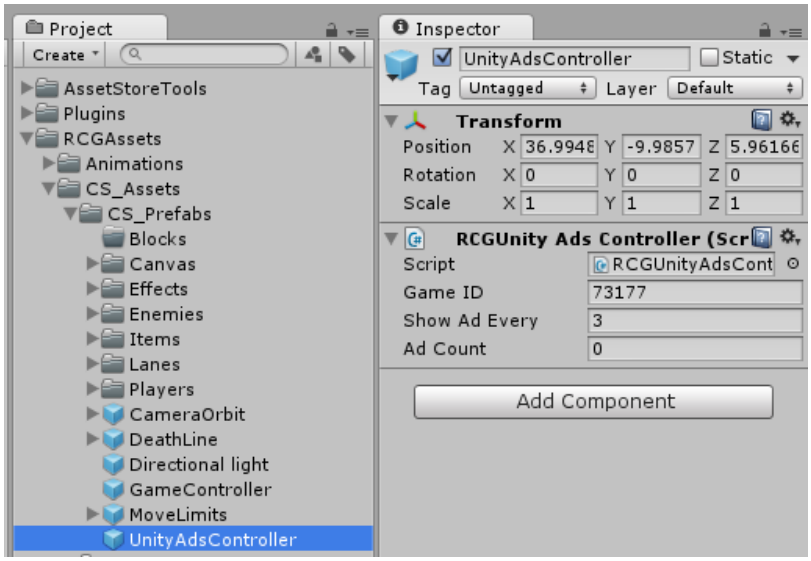


Now we need to bring in the code that integrates the ads into our game. Click on the **PuppeteerUnityAds** package in your project to import it into the game, or choose **Assets > Import Package > Custom Package…** from the top menu and navigate to the **PuppeteerUnityAds** package in your project to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

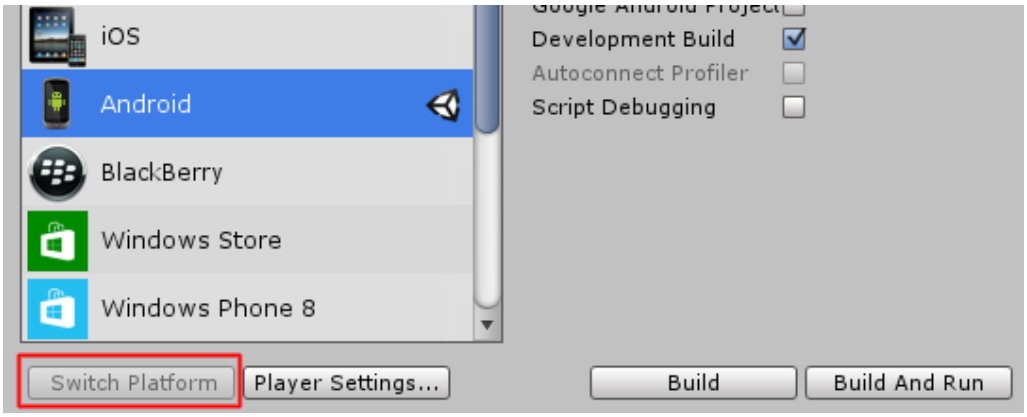


PupUnityAdsController.cs is the main script that links your app to the unityads system. Drag it into your game controller. Now when you look at it you see you can set the GameID of your app, and how often the ads appear. The ad is checked when the level is loaded. "**Show Ad Every**" decides how many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

Can this package be used for commercial projects?

Yes, the basic concept behind the asset store is that you can use assets for your own project, be it commercial or noncommercial. In our case you can use the game template as the basis for your game, or you can take parts of the package (graphics, code, etc.) and put them in your own game.

The only limit is that you may not sell the source (graphics, code, etc) to other people or sell it as a template in a market like Unity Asset Store.

Read more about it here:

http://answers.unity3d.com/questions/342714/use-of-asset-store-assests-in-commercial-game.html

And here are the official Unity Asset Store terms:

https://unity3d.com/legal/as_terms

How do I edit the background, buttons, etc?

The textures in this game, including background, buttons, and other 2D graphical elements are made in PNG format, which can be edited in any graphics/painting software. A quick way to access your textures folder is to right click on the folder and select **Show In Explorer.**

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:

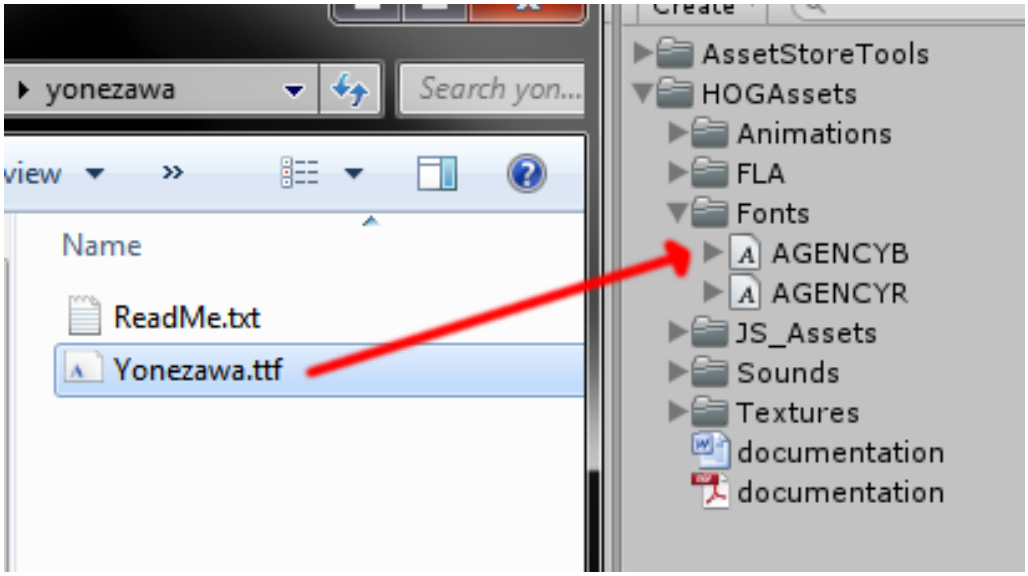
1. Click on a texture in the project view.
2. Click on the override for iPhone button on the right side.
3. Change the format to 16bit.

4. Click Apply.

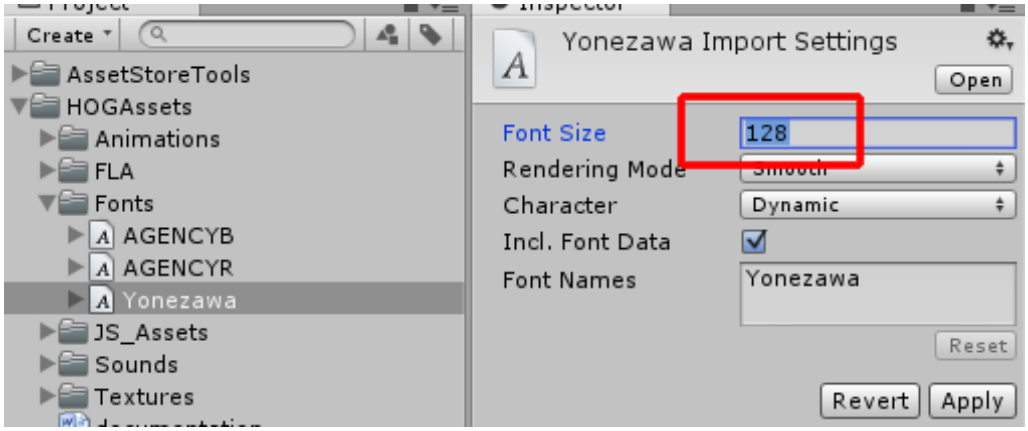
How to change font in the game?

To change a font in the game do the following:

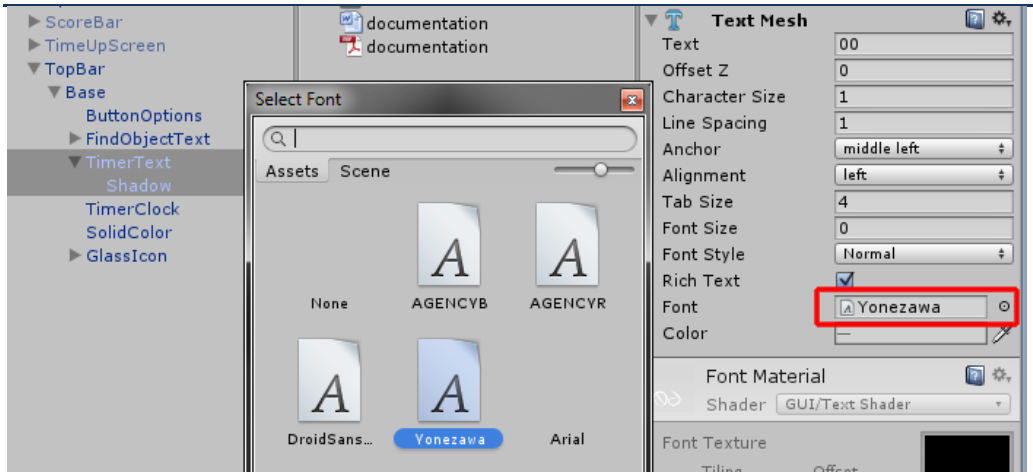
Find a font you like and drag the .ttf file over to the Fonts folder in your game.



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.

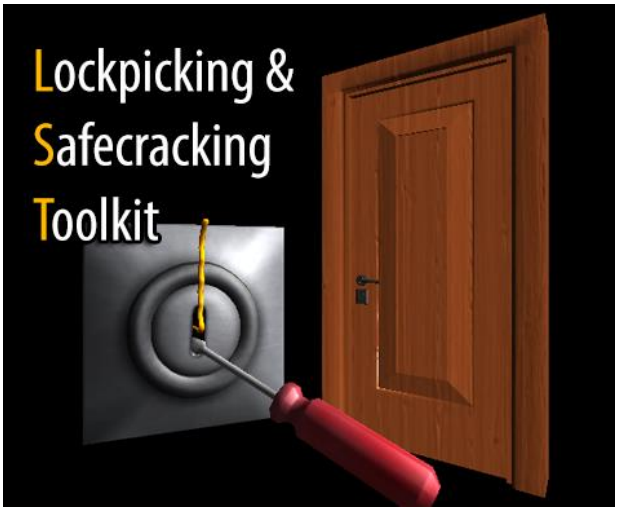


Select any text object in the game and change its font to the new font you have. Sometimes the text might disappear, but it's normal. Just write something in the text box above and it will refresh. Also, make sure you change the text for the shadow; you can select both the main text and its shadow and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


ScoreBar
TimeUpScreen
TopBar
Base
ButtonOptions
FindObjectText
TimerText
Shadow
TimerClock
SolidColor
GlassIcon
documentation
documentation
Select Font
Assets
Scene
None
AGENCYB
AGENCYR
DroidSans...
Yonezawa
Arial
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!