

Quiz Game Documentation

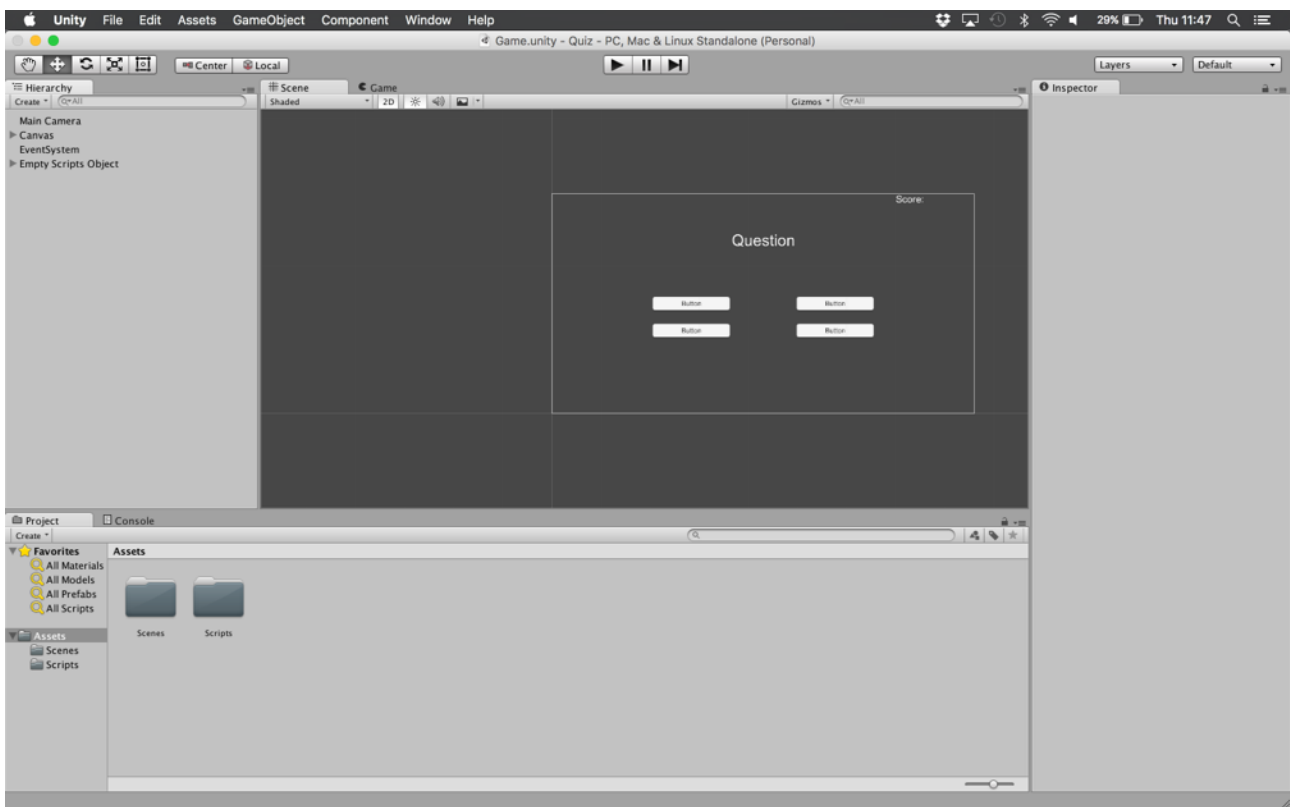
1. Opening the project

Supported Versions - Tested on Unity 5.0+

The project folder consists of all the skeletal requirements to get the game up and running. To open the project follow the below instructions:

Start Unity, menu File/Open Project ..., click on the button "Open Other..." and browse to the top folder of the project which contains Assets and Library as subfolders

Unity should open something to this effect:



Note:

Incase you don't see the scene populated with the buttons and the question text, in your project tab, under the Assets folder, navigate into Scenes and double click on Game. This should open the Game scene.

2. Running The Game

The game can be run by pressing the play button on the top. Once pressed, Unity switches from the Scene tab to the Game tab where the game actually plays out.

In this case, one way to distinguish between them is by looking at the text. In the scene tab, the text on the top says "Question" and the buttons have placeholder names (Button). But once the

game is run and switches to the game tab, The “Question” text is replaced with the actual question - "Which year was CMU founded?"

3. Working Of The Game

The game skeleton is quite straightforward. Once the game is run, the question appears with 4 options to choose from. The player can click on any 4 of the options. The selected option turns yellow.

There is a wait of 2 seconds before the answer is revealed. The right answer is marked in green. Incase you selected the wrong answer, your selection will be in red.

2 seconds after the answer is revealed, the game reloads the same scene (with the same question)

Scoring System:

Right answers fetch you 10 points which is shown on top of the screen. Initially the score is 0 and as you start to answer correctly, your score increments by 10.

What these scripts do:

Theses scripts consists of a skeleton to display your questions, answers and how the UI reacts to the right and wrong options. The logic for the right and wrong option has been implemented. You'll have to move this logic around as you implement the cloud modules.

What these scripts do not do:

These scripts **does not** include any cloud features. All it does is implement how the UI reacts to the player's choices. As of now, the scene is restarted and the same question is loaded.

ScoreManager.cs and UIManager.cs are attached to two empty GameObjects under “Empty Scripts Object” in the Hierarchy.

QuestionAndOptions class is called internally from these scripts.

Clicking on _UIManager Game Object under “Empty Scripts Object” reveals the public properties that can be changed.

In the Inspector tab, you will notice a component by the name of “UI Manager(Script)” which contains different fields like Script, Question Text Field etc.

The property “Score Incrementer” is used to increment the score on the top right. By defaults, its been set to 10. A logic to decrement the score isn't available.

More information about each script is discussed in section 4.

Note:

This is just a framework for the game. You have complete liberty to use the same code or change how to game works entirely. That is your choice.

4. Structure Of The Game/Code

In terms of the code-base, the game consists of 3 scripts.

- QuestionAndOptions.cs
- ScoreManager.cs
- UIManager.cs

QuestionAndOptions.cs

This code consists of the logic and implementation to get and set questions, options and the right option. It also consists of methods to get the answer count.

Setters:

- void setQuestion(string question)
- void setOptions(string option1, string option2, string option3, string option4)
- void setRightOption(int rightOption)

Getters:

- string question ()
- ArrayList options ()
- int rightOption ()
- int optionsCount ()
- string optionAtIndex (int index)

ScoreManager.cs

The score manager keeps track of the score on the top right of the game. It contains two important methods.

- int score () - Used to get the current score
- int incrementScoreBy (int increment) - Used to increment the score by a number and return the new score

UIManager.cs

This is the script where we make use of the above two scripts and actually display it during the game. The other two scripts will not need a lot of changes. But your game have to make use of parts of this script while you implement the cloud features for the game.

Feel free to use parts of this code in your other modules as you develop the cloud based features of the game