Json Quiz Editor

User Manual
Version 1.0

April 11, 2020

Email support: luisrevcoding@gmail.com
Website: www.luiscoding.com

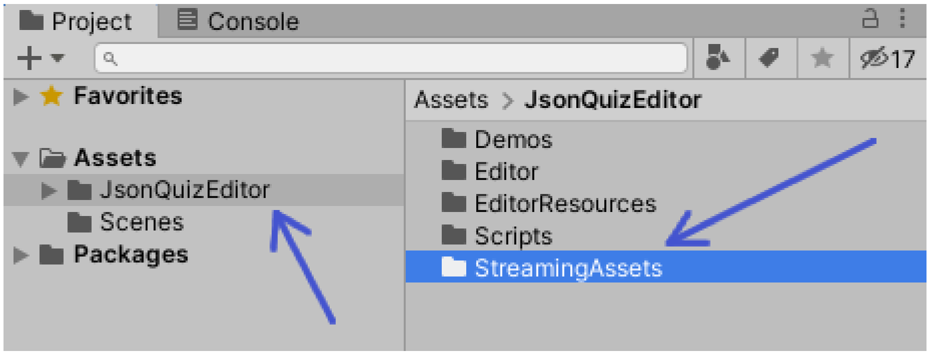
CONTENTS

INTRODUCTION

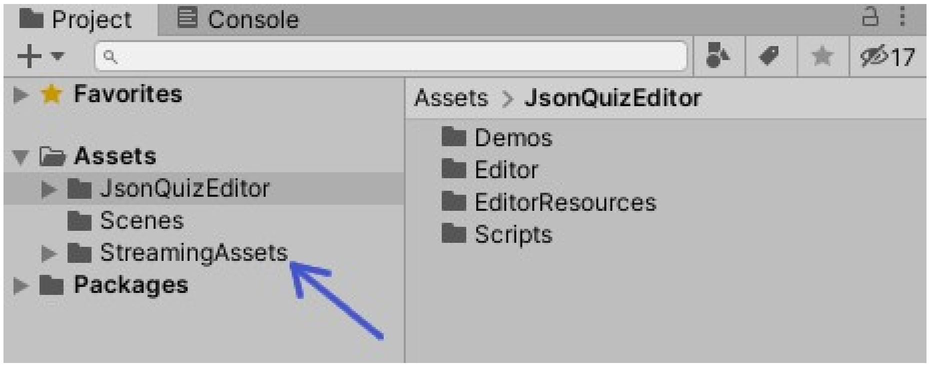
What's the Json Quiz Editor?

It is a customized editor that allows you to create questions and answers.
The information that you have created is saved in a json format file.
You can integrate it into any question apps.
The project has two demos that show how integration is done.
The Quiz Editor has support for Windows and Linux.

Important:
You have to move the StreamingAssets folder to the root, follow these steps:
- The StreamingAssets folder is located inside the JsonQuizEditor folder.
  Assets/JsonQuizEditor/StreamingAssets/



- The StreamingAssets folder has to be moved to the root.
  Assets/StreamingAssets/

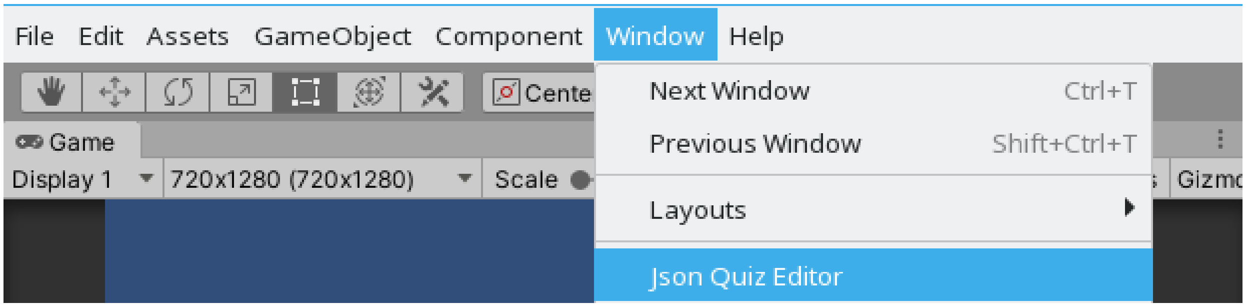


EDITOR

In the Editor, you can create, update, delete the questions and configure the location of the json file and images.

Open Editor

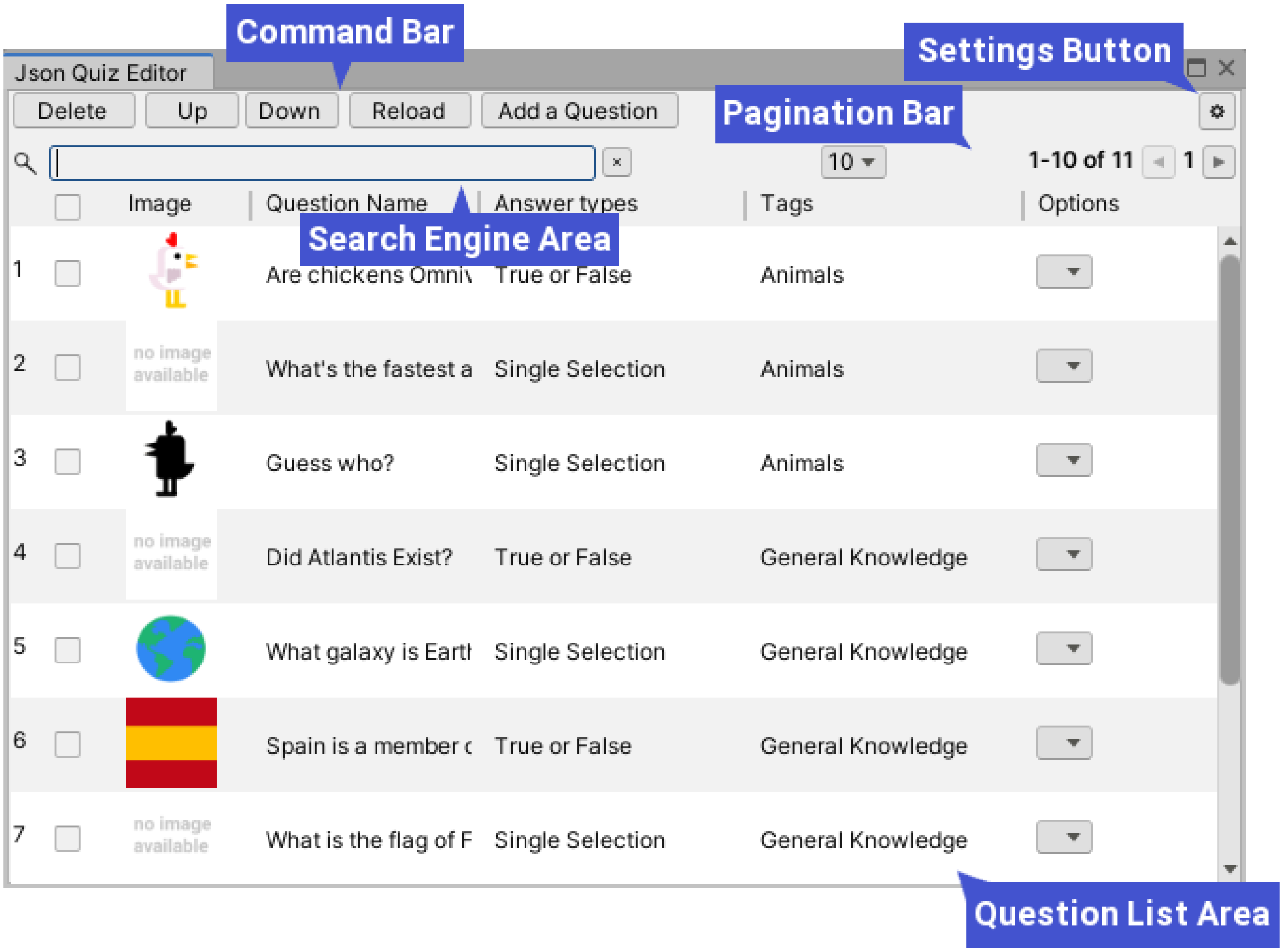
Go to Window menu → Json Quiz Editor



Parts of Quiz Editor

Main Window

In this window, the list of questions is displayed and allows access to the different options of the editor.
The main window may be divided into the following parts:



	□	Image	Question Name	Answer types	Tags	Options
1	□		Are chickens Omniv	True or False	Animals	▾
2	□	no image available	What's the fastest a	Single Selection	Animals	▾
3	□		Guess who?	Single Selection	Animals	▾
4	□	no image available	Did Atlantis Exist?	True or False	General Knowledge	▾
5	□		What galaxy is Eartl	Single Selection	General Knowledge	▾
6	□		Spain is a member c	True or False	General Knowledge	▾
7	□	no image available	What is the flag of F	Single Selection	General Knowledge	▾

Command Bar – a command bar located at the top of the main window.
Comprises the following buttons:
Delete – delete the selected questions.
Up – move up the selected questions..
Down – move down the selected questions..
Reload – reload questions from the json file.
Add a Question – open the edit window.

Settings Button – this button opens the settings window.

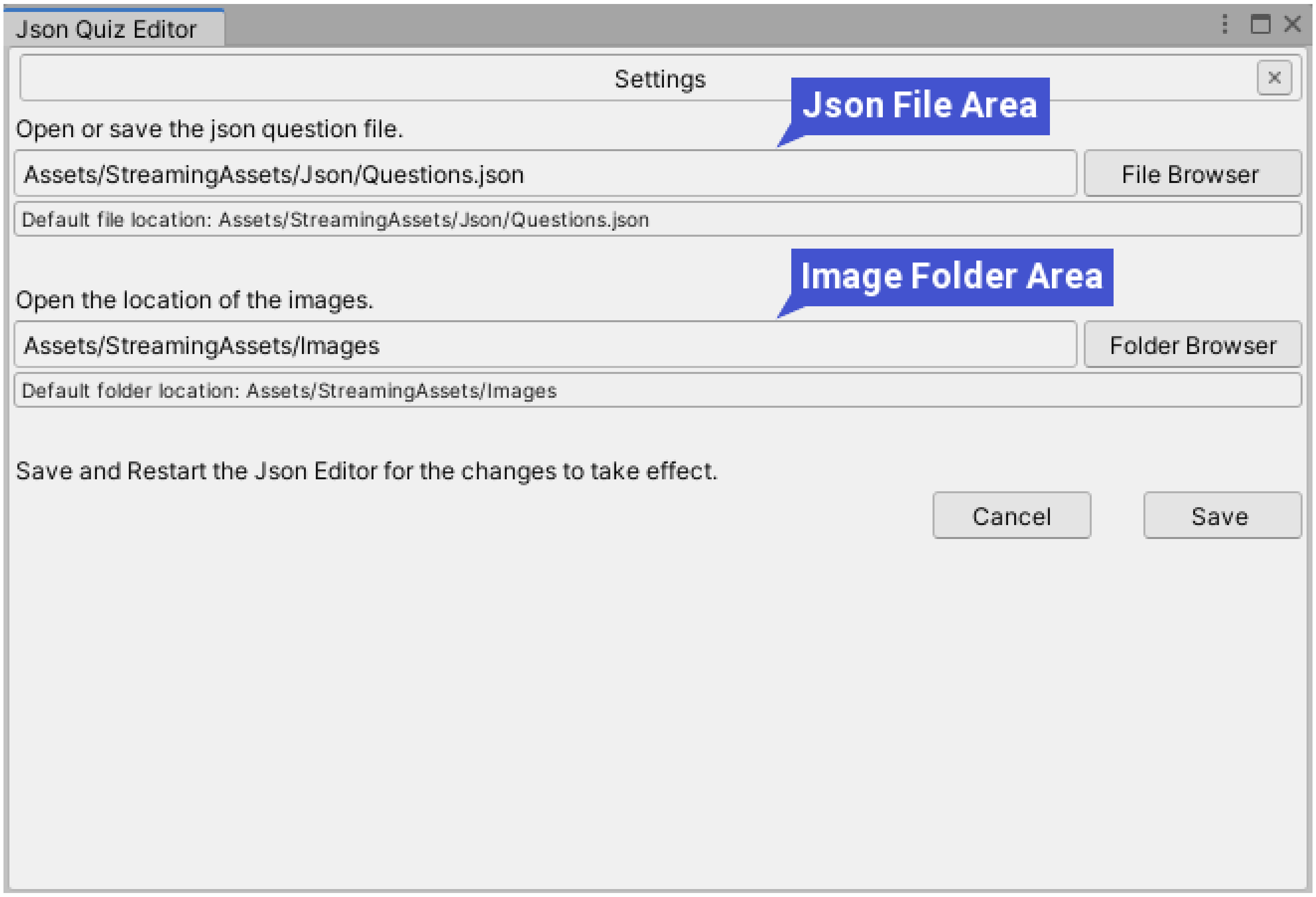
Pagination Bar – in this bar, the selector for the number of questions and the pagination buttons are located.

Search Engine Area – the area where the search engine is located.

Question List Area – the area where the list of questions will be displayed with the image, text, question type, tags and editing option.

Settings Window

In this window, the json file location and images are set.
The json file and images must be inside the project.
Settings window may be divided into the following parts:



Open or save the json question file.

Assets/StreamingAssets/Json/Questions.json

Default file location: Assets/StreamingAssets/Json/Questions.json

Open the location of the images.

Assets/StreamingAssets/Images

Default folder location: Assets/StreamingAssets/Images

Save and Restart the Json Editor for the changes to take effect.

Json File Area – In this area, the path of the json file that will contain the questions is added.

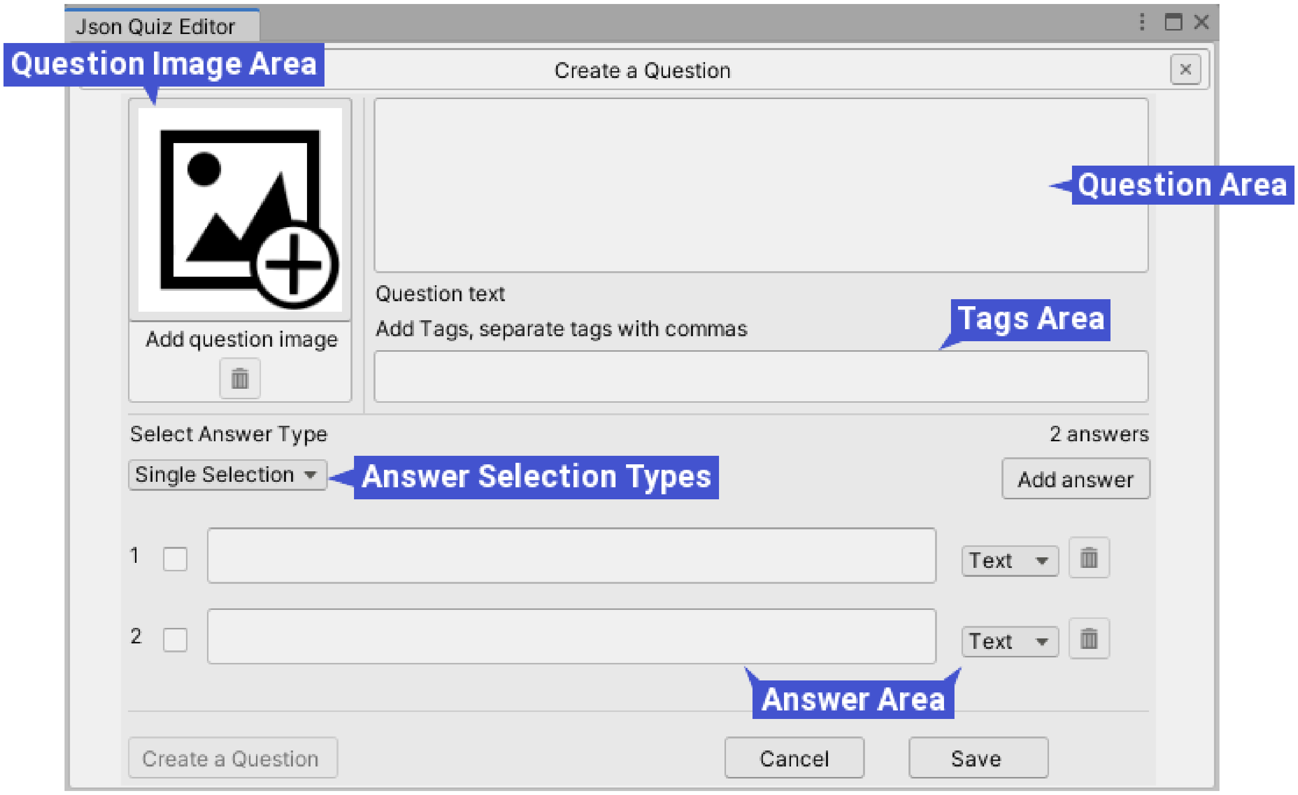
Important:
The json file must be located in Unity project.

Image Folder Area – in this area, the folder path of the images that will be used for the questions and answers is added.

Important:
- The image folder must be located in Unity project.
- Images with png and jpg extension are allowed.
- The location of the json file and the images folder depend on the Windows, Linux or Android platform.

Editor Window

In this window, the questions and answers are added and updated.
Editor window may be divided into the following parts:



Question Area – in this area, you can add the name of the question.

Question Image Area – in this area, you can add or remove an image.
🖼 – this button opens the image gallery.

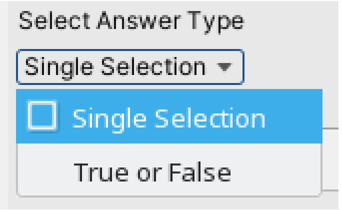
🗑 – delete the image.

Important:
You can add a question that contains:
- Text only.
- Image only.
- Image and text.

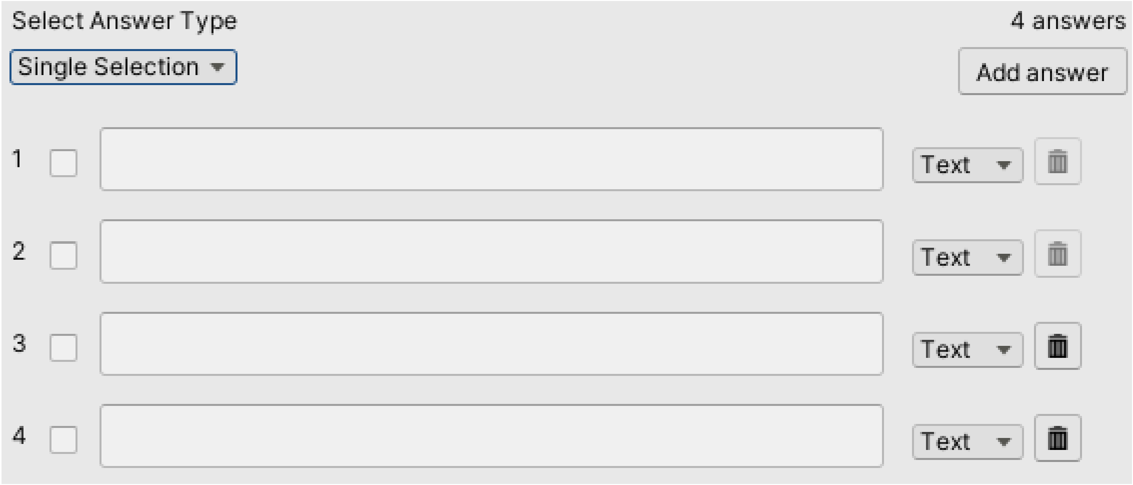
Tags Area – in this area, you can add tags separated by commas.

Answer Area – in this area, you add or remove the answers to the question.

Answer Selection Types – the editor supports two types of answers:



*Single Selection Type – y*ou can add multiple answers, but you can only choose one answer as valid.



Select Answer Type

Single Selection ▾

4 answers

Add answer – this button adds an answer.

🗑 – delete an answer.

Text option – it allows you to add a text string as an answer.

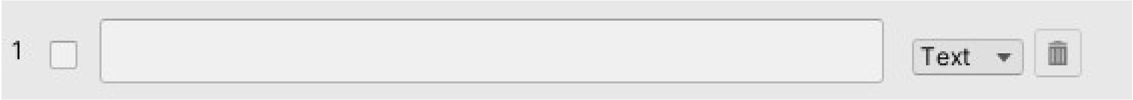
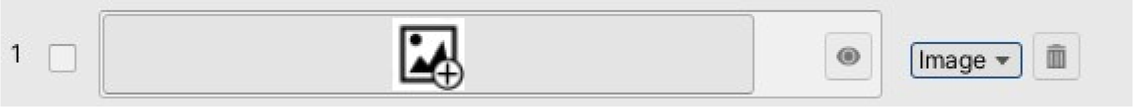


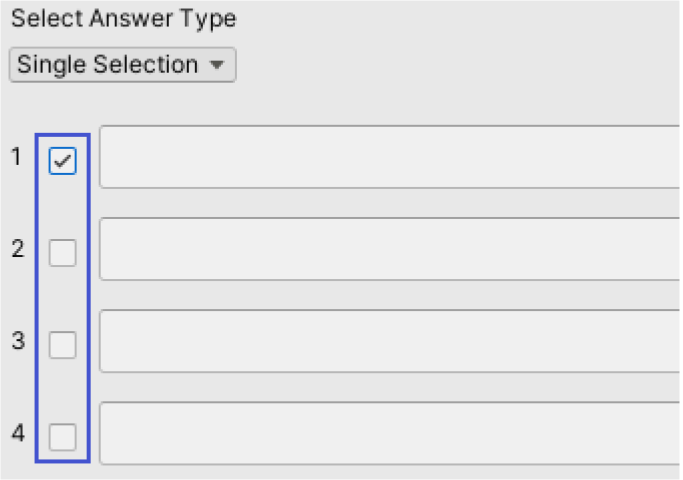
Image option – it allows you to add an image as an answer.



🖼 – this button opens the image gallery.

👁 – this button opens the image viewer.

Select the correct answer – you have to choose the correct answer.



Select Answer Type

Single Selection ▾

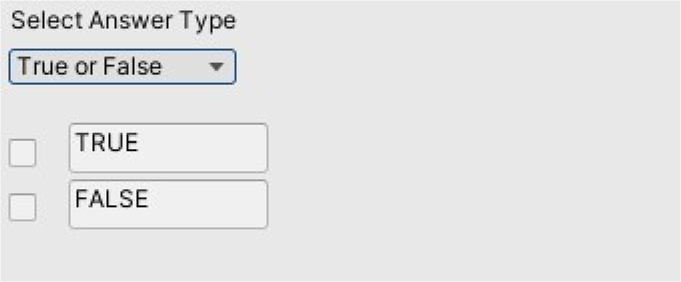
1 ☑

2 □

3 □

4 □

True or False Type – choose the answer which can be true or false.



Select Answer Type

True or False ▾

□ TRUE

□ FALSE

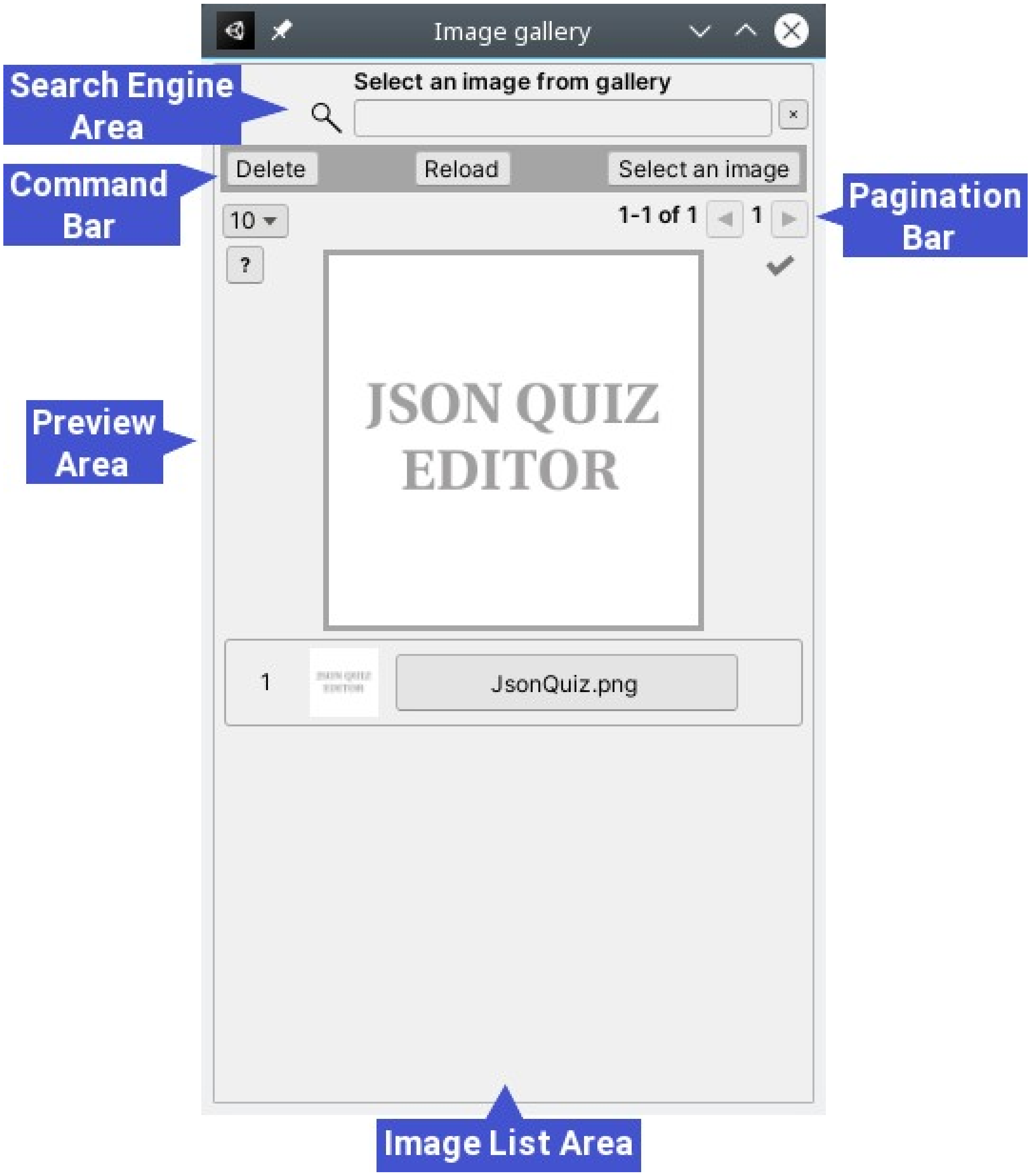
IMAGES GALLERY

The image gallery allows you to choose images that will be added to the questions or answers.

Parts of Images Gallery

Main Window

The main window may be divided into the following parts:



Search Engine Area – the area where the search engine is located.

Command Bar – a command bar located at the top of the main window.
Comprises the following buttons:
Delete – delete an image.
Reload – reload the images.
Select an image – when you choose an image, by clicking on this button, the image will be added to the question or answer that you are editing.

Pagination Bar – The image number selector and paging options are located in this bar.

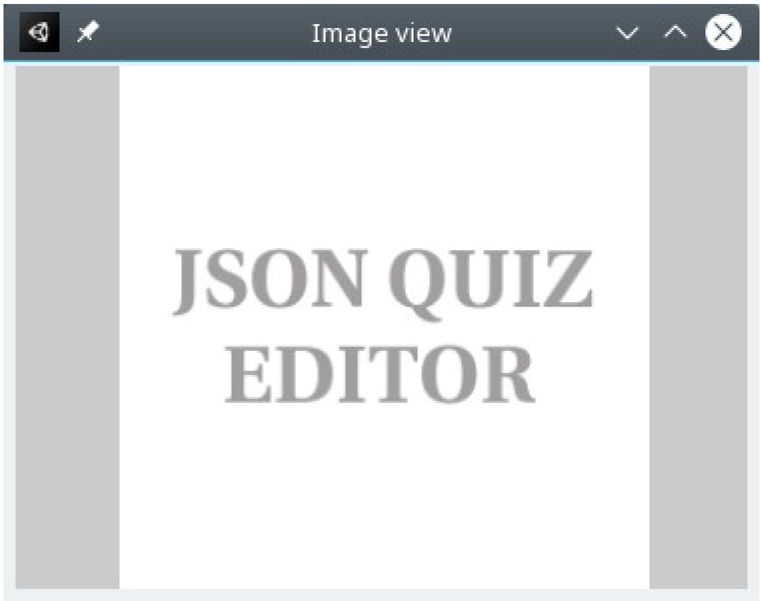
Preview Area – the chosen image will be displayed in this area.
? – this button allows displaying the image data such as name, resolution and size.

✓ – it is not a button; you must place the cursor over the image to see the number of questions and answers the chosen image is using.

Image List Area – the area where all the images that are located in the folder that you chose in the setting window will be displayed.

IMAGE VIEWER

The image viewer shows the image of the answers.



Open the Image viewer

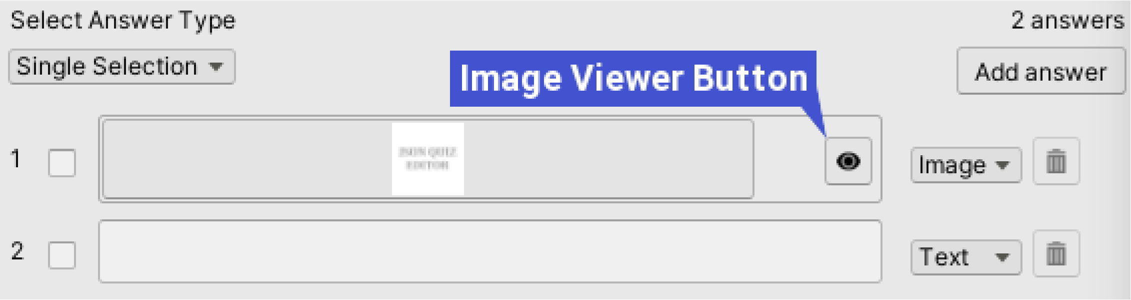
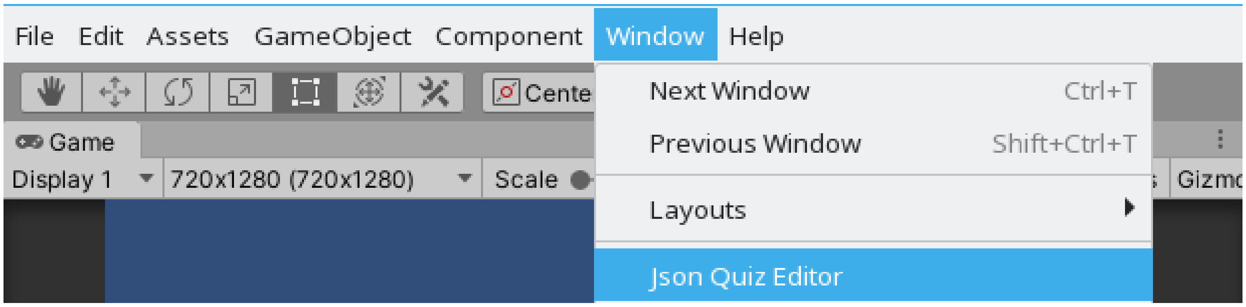


Image Viewer Button – Click this button to see the image that was added in the answer.

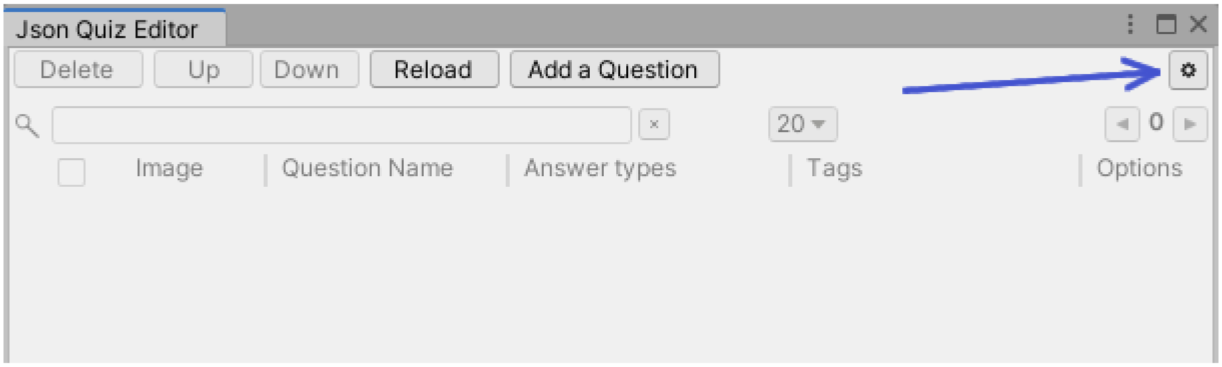
HOW TO ADD QUESTIONS AND ANSWERS

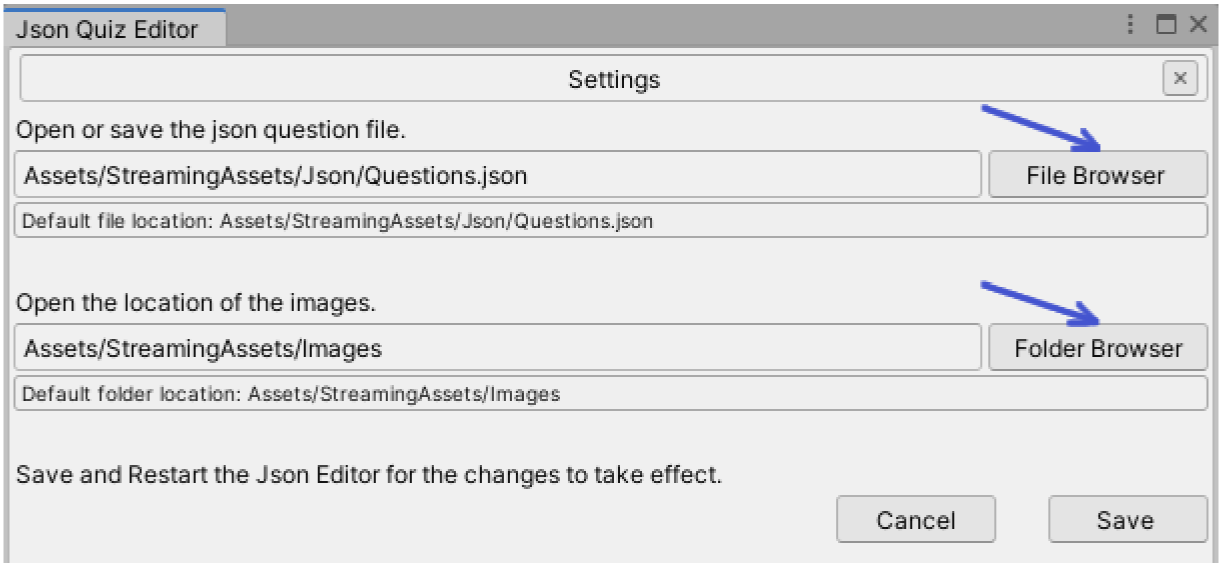
To add questions and answers, follow the next steps.

- Open the editor, go to menu Window → Json Quiz Editor.



- Go to the settings window and chose the json file location and the folder where the images will be depends on the Windows, Linux or Android platform.





Open or save the json question file.

Assets/StreamingAssets/Json/Questions.json

Default file location: Assets/StreamingAssets/Json/Questions.json

Open the location of the images.

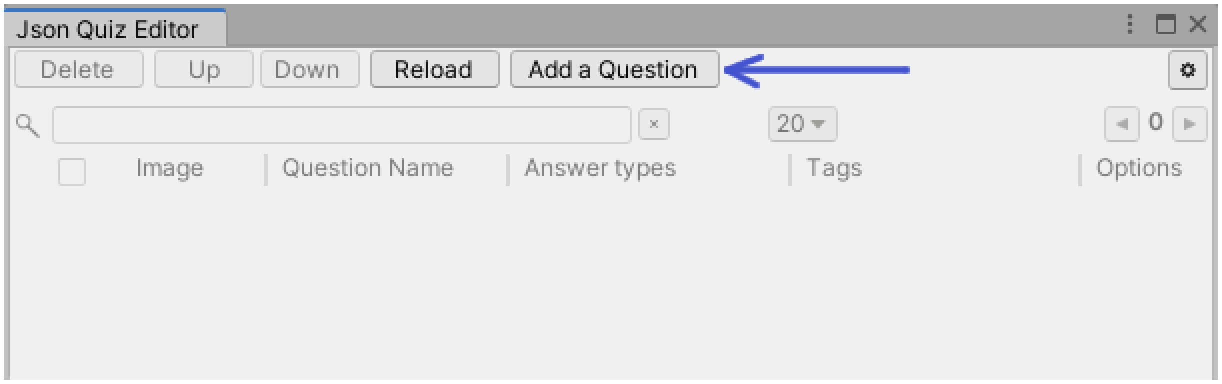
Assets/StreamingAssets/Images

Default folder location: Assets/StreamingAssets/Images

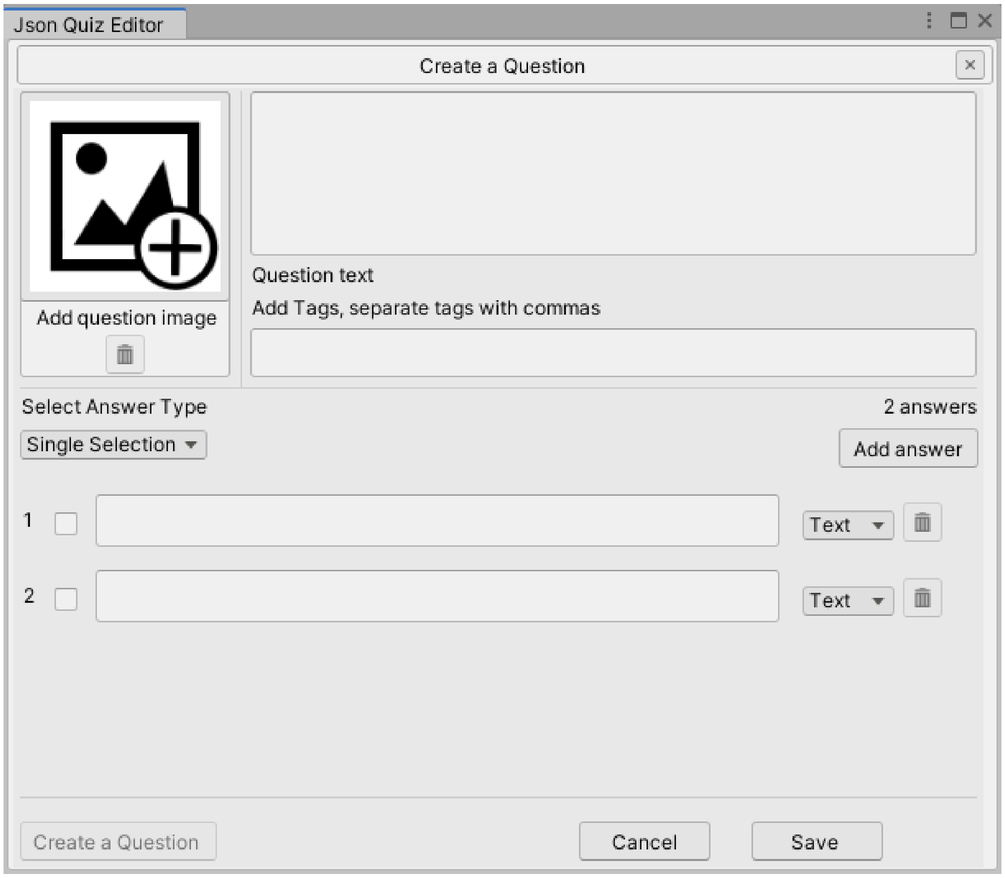
Save and Restart the Json Editor for the changes to take effect.

- In the images folder, add your images that will be used in the application. Only png and jpg formats are supported.

- Go to the main window and add a question by clicking on the “Add a Question” button.



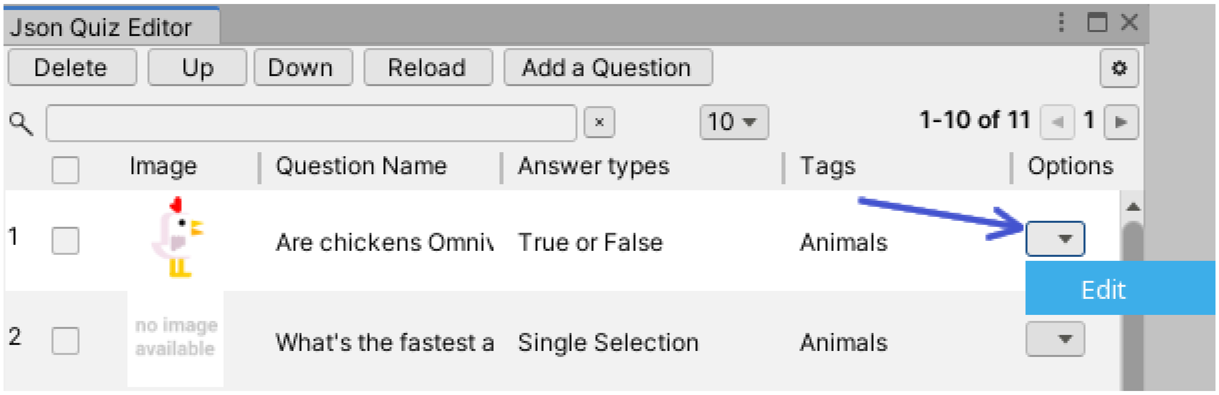
- In the edit window, you can add a question as text and/or image.



- Choose the type answer and add the required information.

- Save question.

To edit a question in the options column, deploy the menu and click on the edit option.



	□	Image	Question Name	Answer types	Tags	Options
1	□		Are chickens Omniv	True or False	Animals	▾
2	□	no image available	What's the fastest a	Single Selection	Animals	▾

JSON FILE STRUCTURE

The questions and answers are saved in a json file.

The json file has the following structure.

```
{"Questions":
        [
                {
                        "imageName":"ChickenMask.png",
                        "questionName":"Guess who?",
                        "tags":"Animals",
                        "typeAnswer":0,
                        "answers":["Bird","Rabbit","Shark","Chicken"],
                        "textOrImage":[0,0,0,0],
                        "selectedAnswers":[false,false,false,true],
                        "trueOrFalse":false,
                        "key":"80252.3252458"
                }
        ]
}
```

imageName – image name of a question.

questionName – question name.

tags – you can add tags, must be separated by commas.

typeAnswer – you have two types of answers:
Single Selection and True or False.
It has the following values:
0 = Single Selection
1 = True or False

answers – it is the list of answers; it can be text or image name.

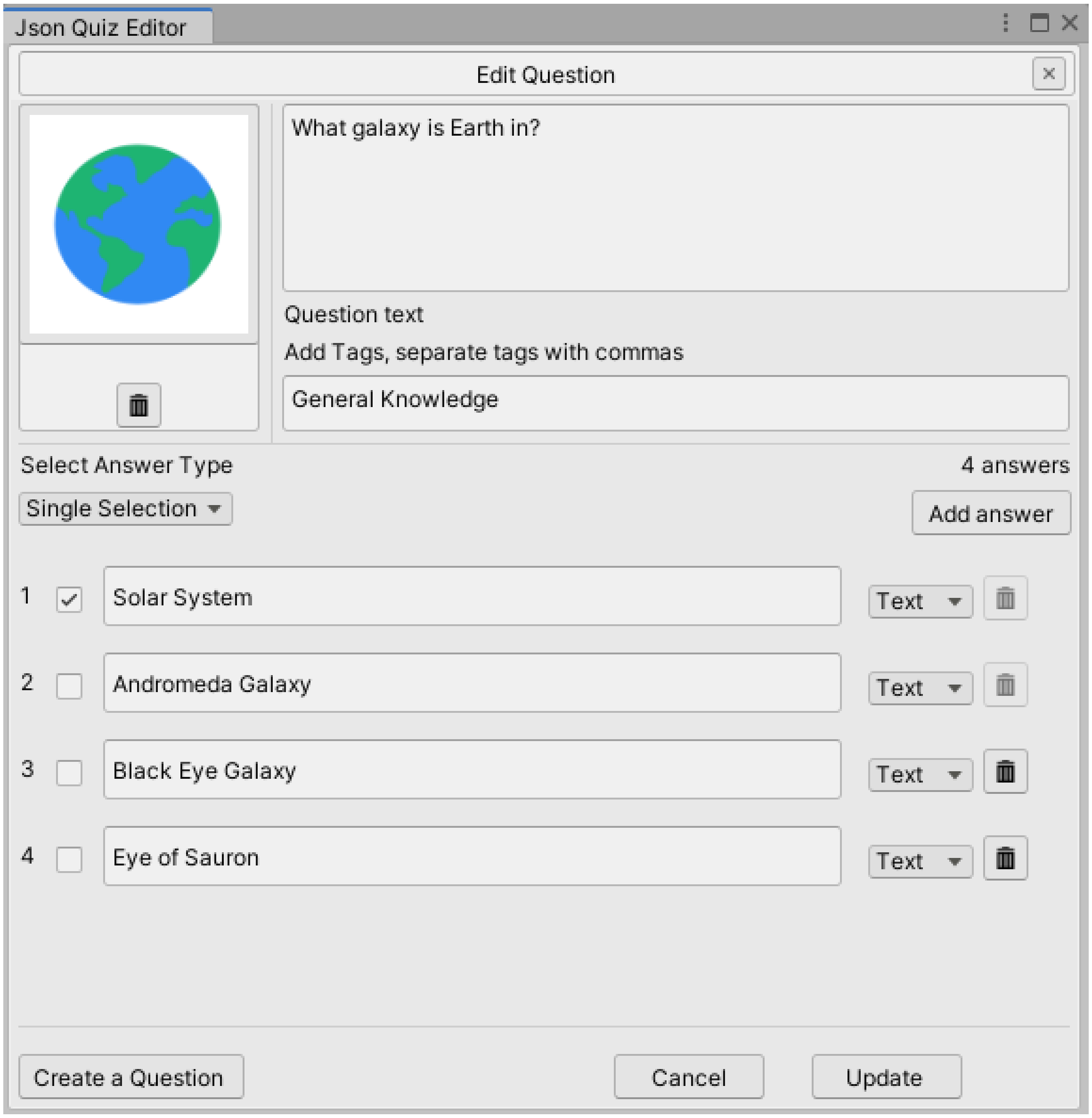
textOrImage – answers can be text or an image name.
It has the following values.:
0 = Text
1 = Image

selectedAnswers – if I choose the answer type "Simple Selection" in this parameter, a list that contains the valid answer (marked as true) is added.

trueOrFalse – if I chose the true or false answer type, in this parameter, the value that corresponds to true or false is added.

key – key question.

Examples



What galaxy is Earth in?

Question text

Add Tags, separate tags with commas

General Knowledge

Select Answer Type: Single Selection

4 answers

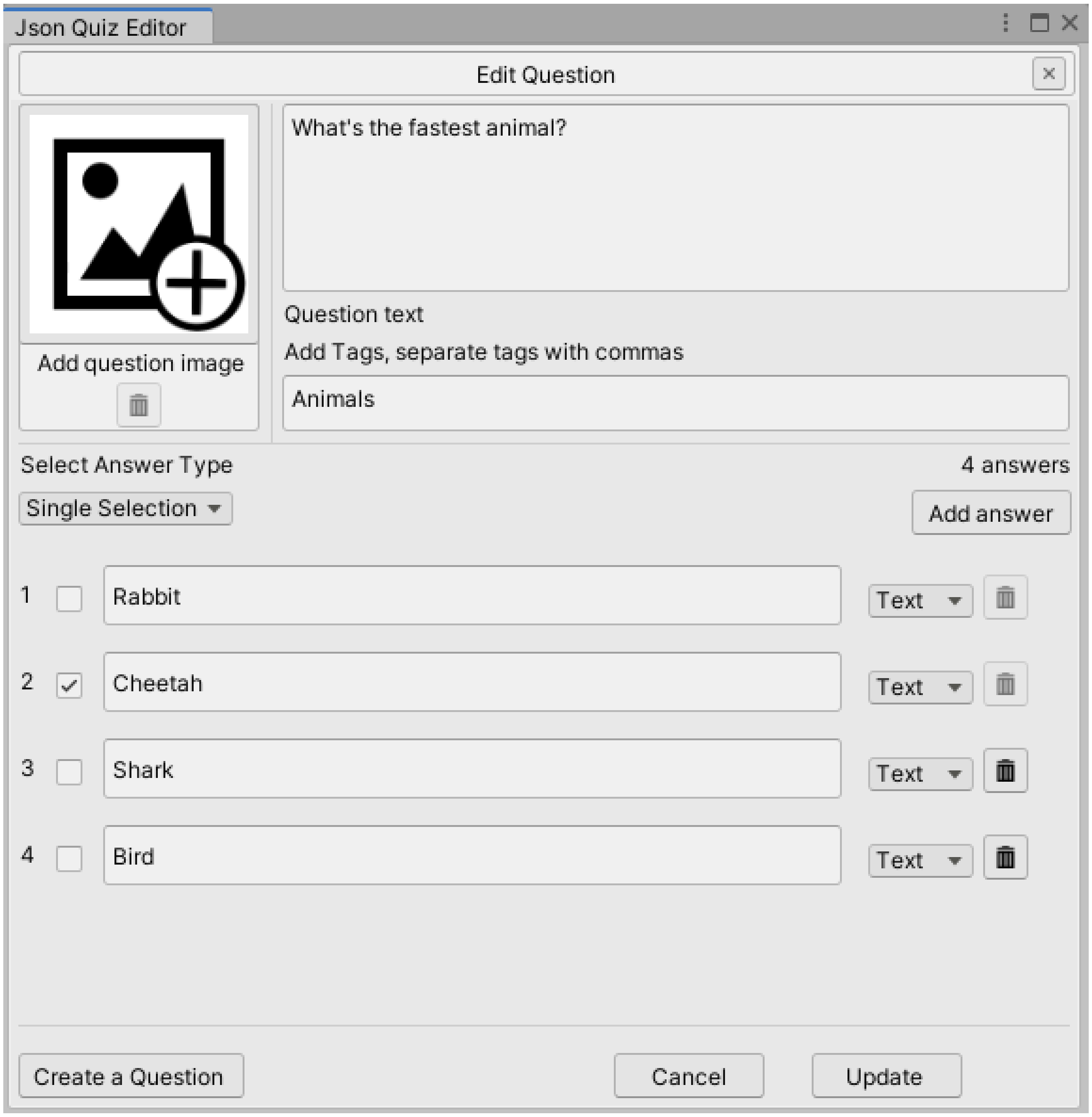
1 ☑ Solar System — Text

2 ☐ Andromeda Galaxy — Text

3 ☐ Black Eye Galaxy — Text

4 ☐ Eye of Sauron — Text

"**imageName**":"PlanetEarth.png",
"**questionName**":"What galaxy is Earth in?",
"**tags**":"General Knowledge",
"**typeAnswer**":0,
"**answers**":
["Solar System","Andromeda Galaxy","Black Eye Galaxy","Eye of Sauron"],
"**textOrImage**":[0,0,0,0],
"**selectedAnswers**":[true,false,false,false],
"**trueOrFalse**":true,
"**key**":"45022.8234647"



What's the fastest animal?

Animals

Single Selection

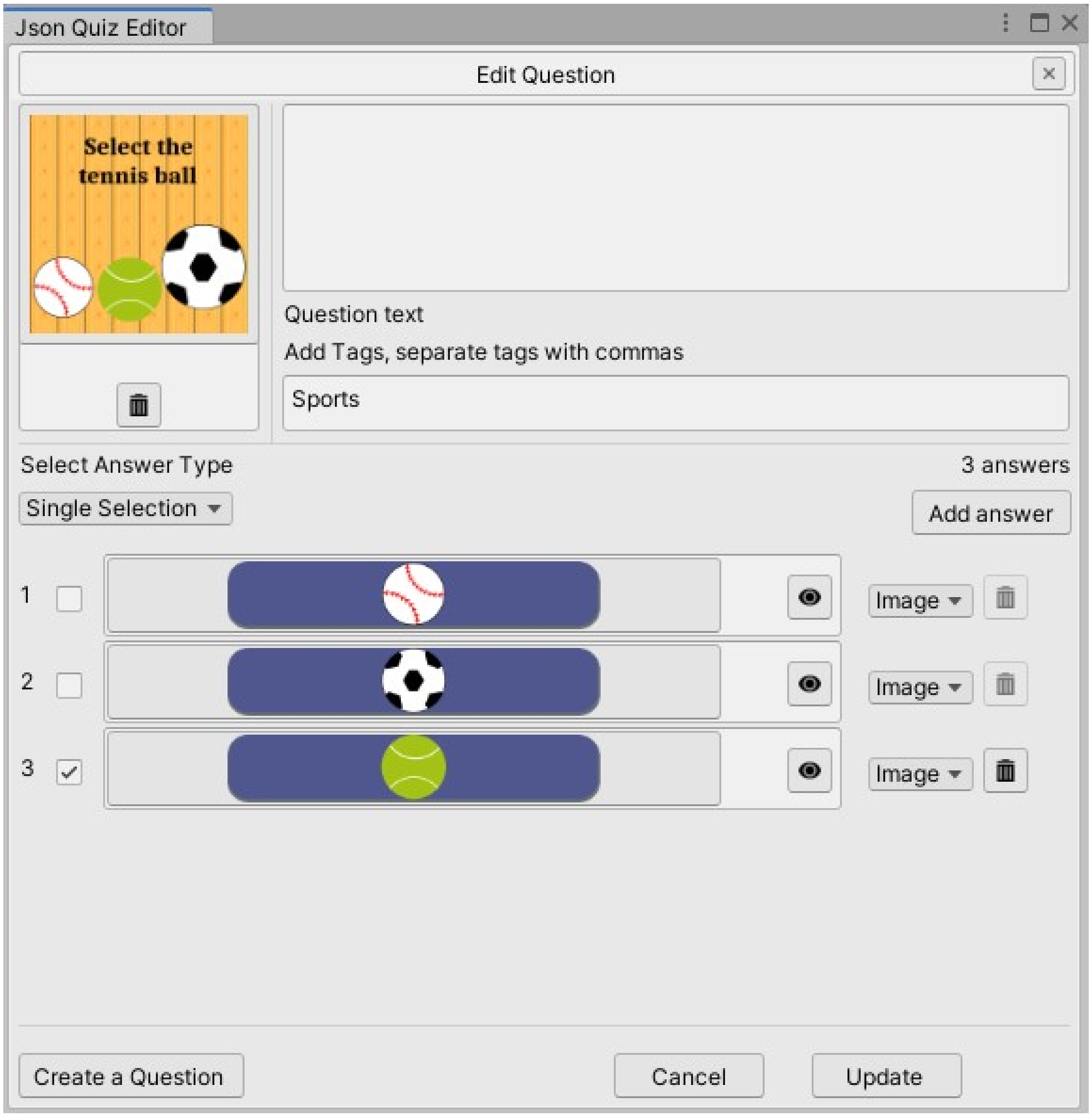
1 ☐ Rabbit

2 ☑ Cheetah

3 ☐ Shark

4 ☐ Bird

"**imageName**":"",
"**questionName**":"What's the fastest animal?",
"**tags**":"Animals",
"**typeAnswer**":0,
"**answers**":["Rabbit","Cheetah","Shark","Bird"],
"**textOrImage**":[0,0,0,0],
"**selectedAnswers**":[false,true,false,false],
"**trueOrFalse**":false,
"**key**":"74042.2280342"



Question text

Add Tags, separate tags with commas

Sports

Select Answer Type

Single Selection

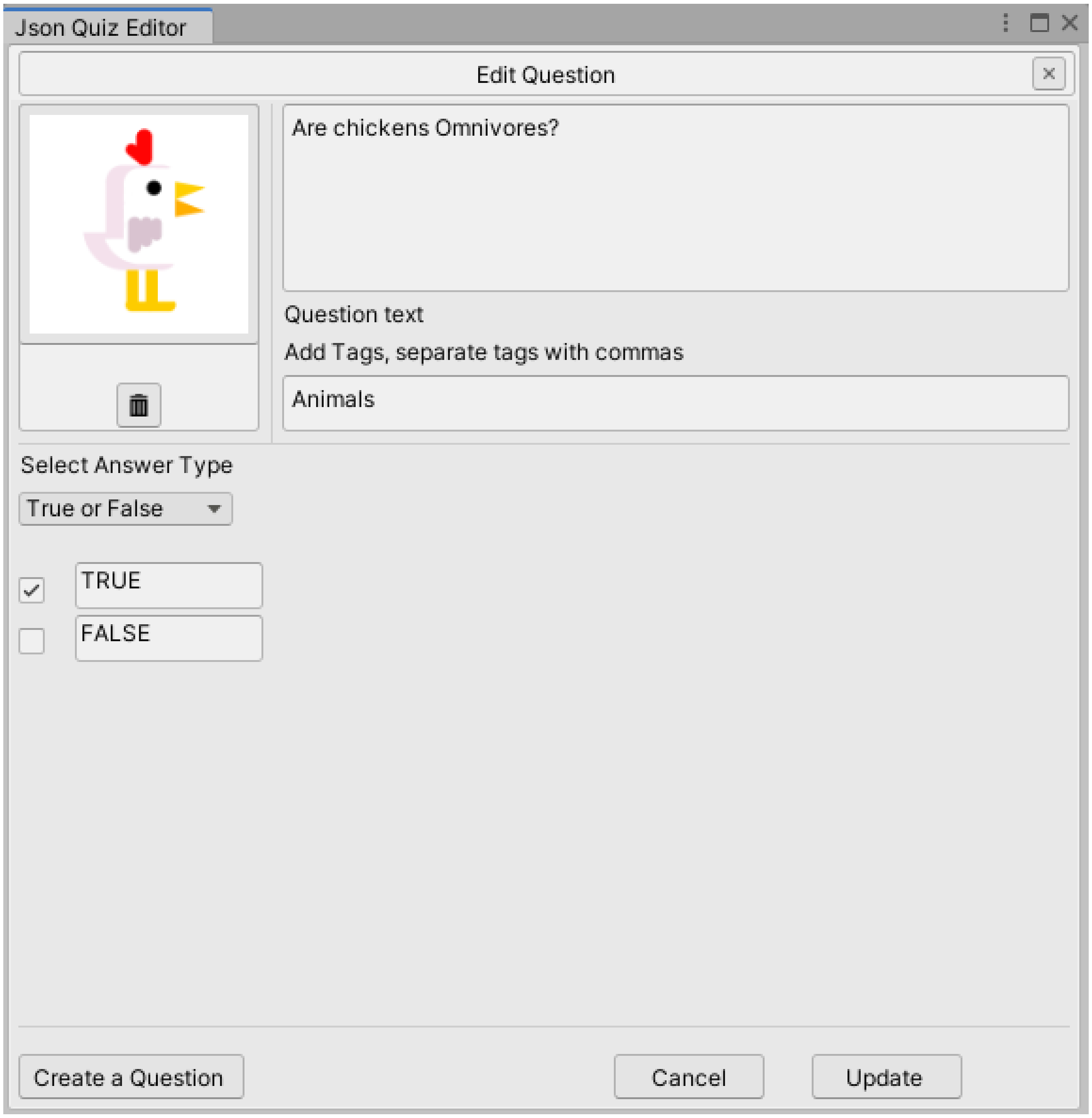
3 answers

1 □ Image

2 □ Image

3 ☑ Image

"**imageName**":"SelectTenisBall.png",
"**questionName**":"",
"**tags**":"Sports",
"**typeAnswer**":0,
"**answers**":["Baseball_Answer.png","SoccerBall_Answer.png","TennisBall_Answer.png"],
"**textOrImage**":[1,1,1],
"**selectedAnswers**":[false,false,true],
"**trueOrFalse**":false,
"**key**":"96942.2282244"



Are chickens Omnivores?

Question text

Add Tags, separate tags with commas

Animals

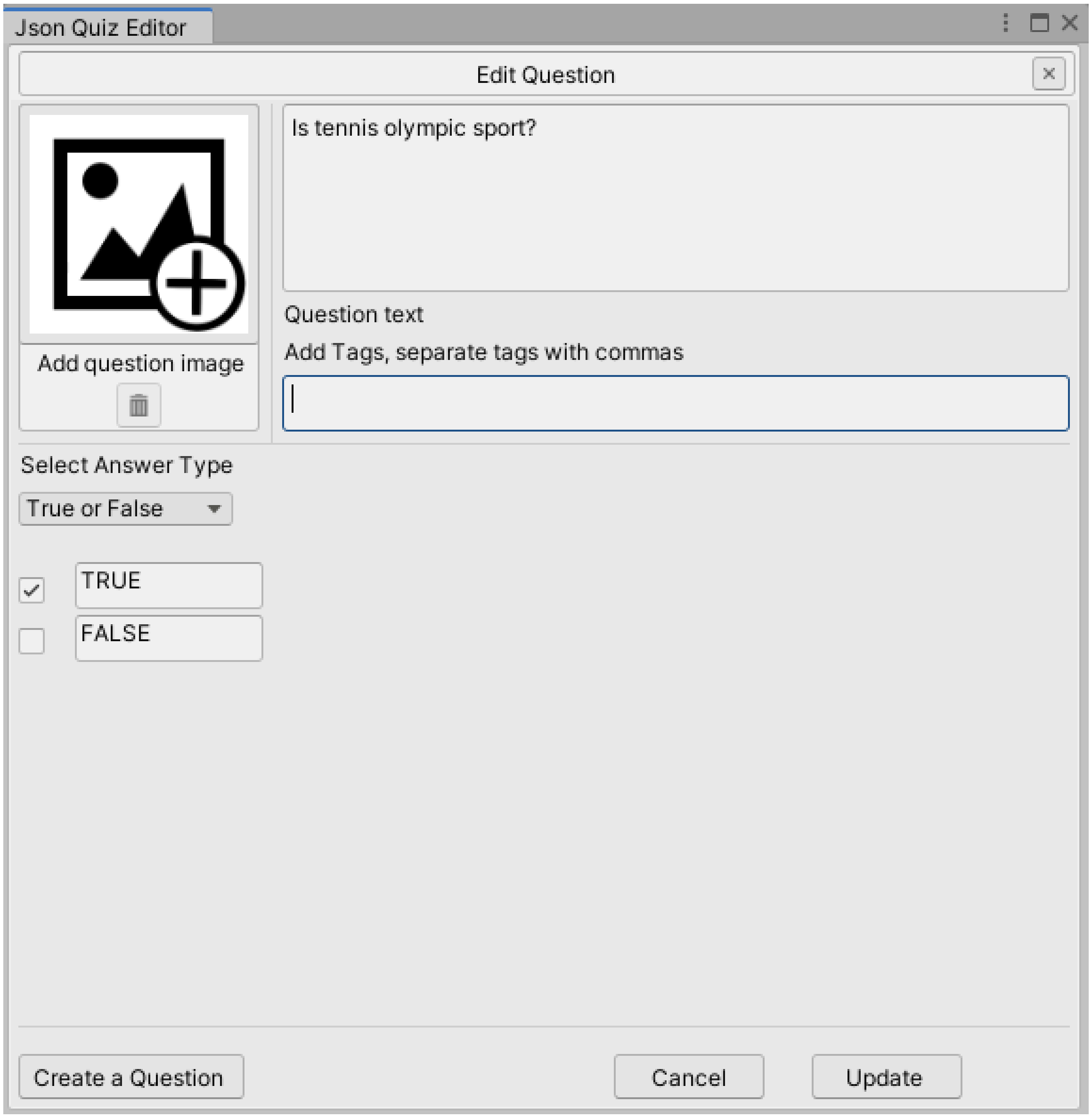
Select Answer Type

True or False

☑ TRUE

☐ FALSE

"**imageName**":"Chicken.png",
"**questionName**":"Are chickens Omnivores?",
"**tags**":"Animals",
"**typeAnswer**":1,
"**answers**":["",""],
"**textOrImage**":[0,0],
"**selectedAnswers**":[true,false],
"**trueOrFalse**":true,
"**key**":"62002.1290797"


Json Quiz Editor

Edit Question

Is tennis olympic sport?

Question text

Add Tags, separate tags with commas

Add question image

Select Answer Type

True or False

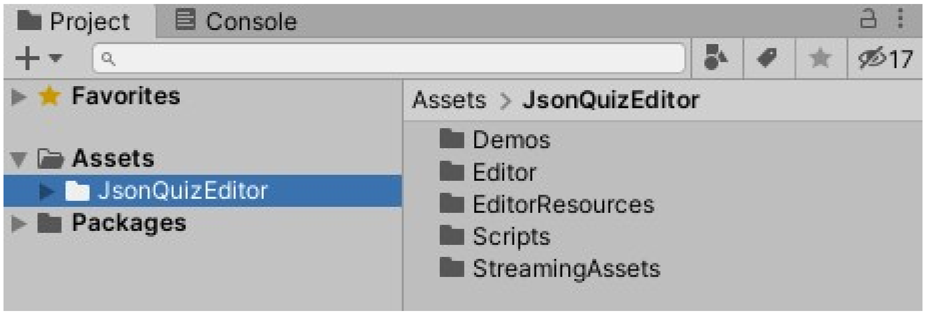
☑ TRUE

□ FALSE

"**imageName**":"",
"**questionName**":"Is tennis olympic sport?",
"**tags**":"",
"**typeAnswer**":1,
"**answers**":["",""],
"**textOrImage**":[0,0],
"**selectedAnswers**":[true,false],
"**trueOrFalse**":true,
"**key**":"13130.1297024"

PACKAGE STRUCTURE

The package is organized by folders intended for the Editor and Demos.



The folders for the Editor are as follows:
JsonQuizEditor/Editor/ → contains the scripts.
JsonQuizEditor/EditorResources/ → contains the images and configuration files.

The folders for the Demos are as follows:
JsonQuizEditor/Demos/QuizDemo/ → contains scripts, scene, prefabs and sprites.
JsonQuizEditor/Demos/SimpleDemo/ → contains the scripts and scene.
JsonQuizEditor/StreamingAssets/Demo → contains images and the json file.

Assets/JsonQuizEditor/Scripts/QuestionsFromJson.cs → the main class that allows obtaining the questions and answers from the json file is used by the demos or integrating it into a project.

HOW TO INTEGRATE IT INTO YOUR PROJECTS

If you have experience in reading and writing json files, you will have no problem to integrate it into your projects.

Scripting API

The different classes and methods are shown.

QuestionsFromJson class

Description
It is the base class that allows you to obtain the information from the json file and integrates it into your project.
File location path Assets/JsonQuizEditor/Scripts/QuestionsFromJson.cs

Constructor
public QuestionsFromJson (string jsonFileFolder, string jsonFilePath)
It allows the creation of an object with access to the information in the json file.
Parameters:
string jsonFileFolder – path of the folder where the json file is located.
string jsonFilePath – name of the json file.
Returns:
It returns the object with the information from the questions and answers.

Public Methods
GetTags – returns the list of tags.
GetQuestionsKeysByTag – it returns the keys to the questions that are associated with a tag.
GetOnlyOneQuestion – It returns a question chosen by the key.

Method Detail
GetTags
public List<string> GetTags ()
Returns:
It returns the tag list for all questions.

GetQuestionsKeysByTag
public List<string> GetQuestionsKeysByTag (string tag)
Parameters:
string tag – tag name.
Returns:
It returns the question's keys list associated with a tag.

GetOnlyOneQuestion
public OnlyOneQuestion GetOnlyOneQuestion (string key)
A question is obtained with its corresponding answers.
Parameters:
string key – key question.
Returns:
It returns an object of the "OnlyOneQuestion" class that contains the information of the question.

OnlyOneQuestion class

Description
It is the class that contains the properties of the question.
Variables
string imageName – name of image.
string questionName – question.
string tags – question tags separated by commas.
int typeAnswer – type of answer, single selection = 0 , true/false = 1
AllAnswers allAnswers – list of answers.
bool trueOrFalse – if I choose the type of answer "true or false" in this parameter, the corresponding value is added (true or false).
string key – key question.

AllAnswers class

Description
It is the class that contains the answers' properties.
Variables
List<string> answers – list of answers to a question, it can be text or the image name.
List<int> textOrImage – Answers' type of content, 0 = text or 1 = image.
List<bool> selectedAnswers – it indicates the location of the correct answer; the true value is added.

Examples

Basic Code

A basic code example.
It prints the list of questions, answers, and tags in the console window.

```
using System.Collections.Generic;
using UnityEngine;

public class Demo : MonoBehaviour
{
      QuestionsFromJson questionsFromJson;

      void Start () {
            ReadJsonQuestionFile ();
      }

      void ReadJsonQuestionFile () {
            questionsFromJson = new
                  QuestionsFromJson (
                        "/StreamingAssets/Demo/Json/", "Questions.json"
                  );
            ReadTags ();
      }

      void ReadTags () {
            List<string> tagList = questionsFromJson.GetTags ();
            for (int indexTag = 0; indexTag < tagList.Count; indexTag++) {
                  string tag = tagList [indexTag];
                  Debug.Log ("TAG = " + tag);
                  ReadQuestions (tag);
            }
      }

      void ReadQuestions (string tag) {
            List<string> keyQuestions =
                  questionsFromJson.GetQuestionsKeysByTag (tag);
            for (int indexKey = 0; indexKey < keyQuestions.Count; indexKey++) {
                  Debug.Log ("-> QUESTION = " + (indexKey + 1));
                  string key = keyQuestions [indexKey];
                  OnlyOneQuestion question =
                        questionsFromJson.GetOnlyOneQuestion (key);
                   // Name of the image.
                  string imageName = question.imageName;
                  Debug.Log ("--> Image Name = " + imageName);
                  // Question in text format.
                  string questionName = question.questionName;
                  Debug.Log ("--> Question Name = " + questionName);
            ReadAnswers (question);
        }
    }

    void ReadAnswers (OnlyOneQuestion question) {
        switch (question.typeAnswer) {
        case 0: // Type of answer: Simple selection.
            Debug.Log ("---> TYPE ANSWER = Single Selection");
            AllAnswers allAnswers = question.allAnswers;
            for (int index = 0; index < allAnswers.answers.Count; index++) {
                if (allAnswers.selectedAnswers [index]) {
                    // Check if the answer is correct.
                }
                if (allAnswers.textOrImage [index] == 0) {
                    // The answer is only text.
                    string text = allAnswers.answers [index];
                    Debug.Log ("----> Text = " +
                        allAnswers.answers [index]);
                }
                if (allAnswers.textOrImage [index] == 1) {
                    // The answer is only image.
                    string image = allAnswers.answers [index];
                    Debug.Log ("----> Image = " +
                        allAnswers.answers [index]);
                }
            }
        break;
        case 1: // Type of answer: True or False.
            Debug.Log ("---> TYPE ANSWER = True or False");
            if (question.trueOrFalse) {
                // correct answer = true;
                Debug.Log ("----> True or False = true");

            } else {
                // correct answer = false;
                Debug.Log ("----> True or False = false");
            }
        break;
        }
    }
}
```

Demos

The project includes two demos "SimpleDemo" and "QuizDemo", both use the "GlobalDemoVariables" class located at:
Assets/JsonQuizEditor/Demos/Scripts/GlobalDemoVariables.cs

This class encloses global variables, such as the location of the json file and the images folder for Windows, Linux and Android. If you want to change the location of these files, modify these variables.

```
// File name json.
public const string jsonQuestionFileName = "Questions.json";
// Folder containing json file. Works only on Editor, Windows and Linux.
// Add trailing slash (/) to start and end.
public const string jsonQuestionFolderPC = "/StreamingAssets/Demo/Json/";
// Folder containing json file. Works only on Android.
// Add trailing slash (/) to start and end.
public const string jsonQuestionFolderAndroid = "/Demo/Json/";
// Folder containing images. Works only on Editor, Windows and Linux.
// Add trailing slash (/) to start and end.
public const string imagesFolderPC = "/StreamingAssets/Demo/Images/";
// Folder containing images. Works only on Android.
// Add trailing slash (/) to start and end.
public const string imagesFolderAndroid = "/Demo/Images/";
```

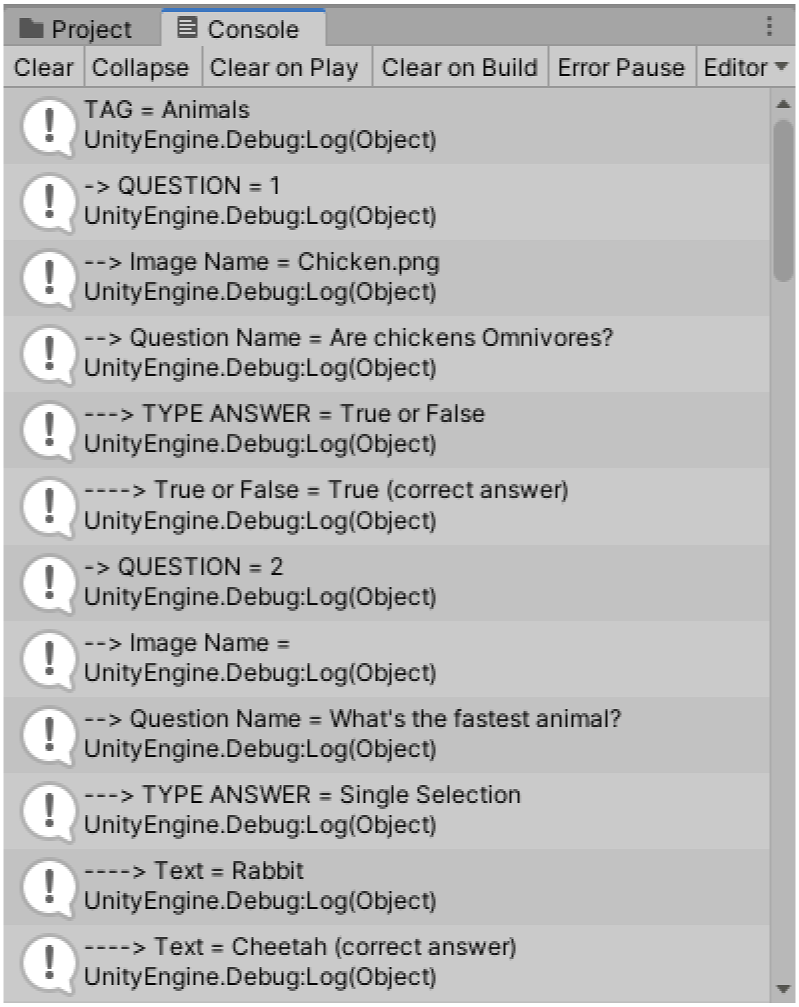
They also use the "QuestionFromJson" class, which allows getting the questions and answers from the json file, this class is located at:
Assets/JsonQuizEditor/Scripts/QuestionsFromJson.cs

The source code of the demos has comments so that you can adapt it to your projects.

Simple Demo

Description

The demo shows the use of APIs in an easy and simple way.
Prints in the console window the list of questions, answers and tags.
The json file is located at
Assets/StreamingAssets/Demo/Json/Questions.json
The images are located in the folder
Assets/StreamingAssets/Demo/Images
The scene file is located at
Assets/JsonQuizEditor/Demos/SimpleDemo/Scenes/SimpleDemo.unity
The script file is located at
Assets/JsonQuizEditor/Demos/SimpleDemo/Scripts/SimpleDemo.cs
Put the game in play mode.



```
TAG = Animals
UnityEngine.Debug:Log(Object)
-> QUESTION = 1
UnityEngine.Debug:Log(Object)
--> Image Name = Chicken.png
UnityEngine.Debug:Log(Object)
--> Question Name = Are chickens Omnivores?
UnityEngine.Debug:Log(Object)
---> TYPE ANSWER = True or False
UnityEngine.Debug:Log(Object)
----> True or False = True (correct answer)
UnityEngine.Debug:Log(Object)
-> QUESTION = 2
UnityEngine.Debug:Log(Object)
--> Image Name =
UnityEngine.Debug:Log(Object)
--> Question Name = What's the fastest animal?
UnityEngine.Debug:Log(Object)
---> TYPE ANSWER = Single Selection
UnityEngine.Debug:Log(Object)
----> Text = Rabbit
UnityEngine.Debug:Log(Object)
----> Text = Cheetah (correct answer)
UnityEngine.Debug:Log(Object)
```

You can add or delete questions in the editor, access the configuration option and select the location of the json file and the images folder.

Quiz Demo

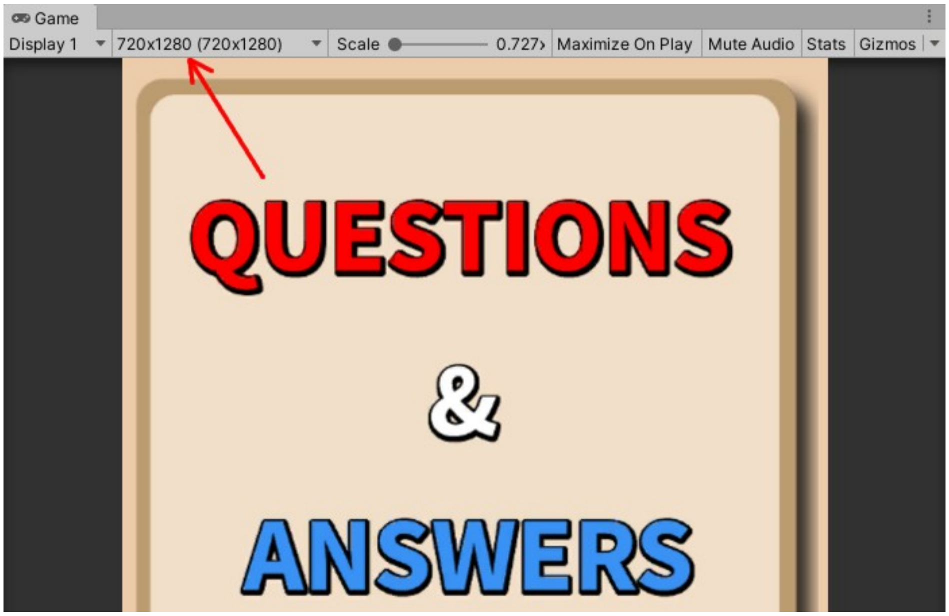
Description

The demo has different menus: main, categories, level, pause and end of the game.
The demo can be exported to Windows, Linux and Android platforms.
When exporting to Windows or Linux, you must set the resolution to 720x1280 in the "Player Settings" and "Resolution and Presentation" option, on Android change the orientation in portrait.

The json file is located at
Assets/StreamingAssets/Demo/Json/Questions.json
The images are located in the folder
Assets/StreamingAssets/Demo/Images
The scene file is located at
Assets/JsonQuizEditor/Demos/QuizDemo/Scenes/QuizDemo.unity
The scripts file is located at
Assets/JsonQuizEditor/Demos/QuizDemo/Scripts

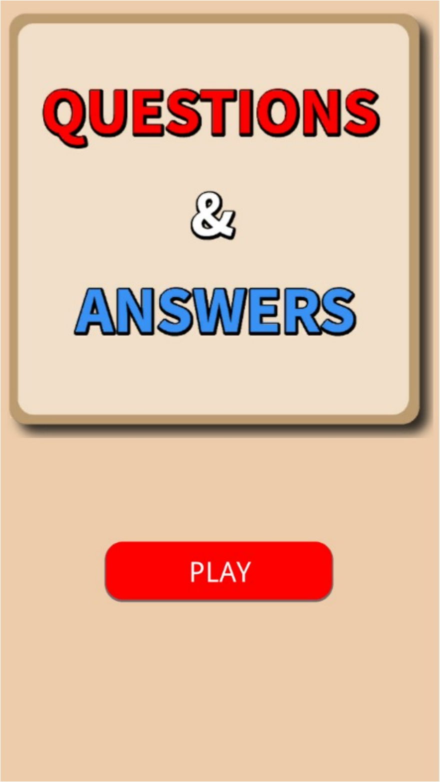
You can add or delete questions in the editor, access the configuration option and select the location of the json file and the images folder.

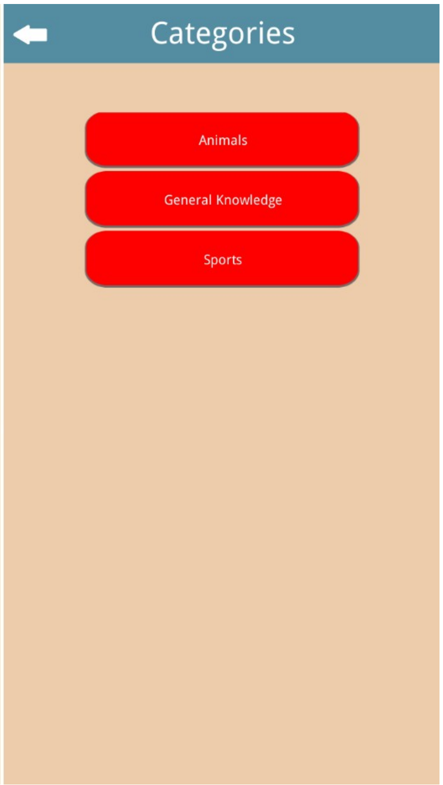
This demo is created specifically for portrait mode, change aspect ratio to 720x1280.

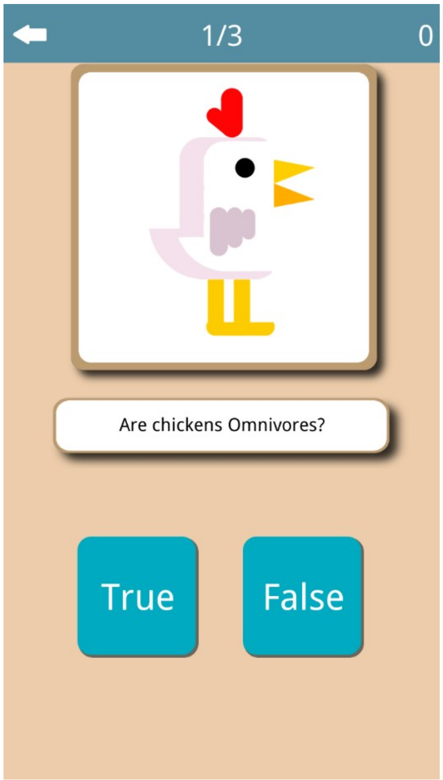


Display 1 | 720x1280 (720x1280) | Scale 0.727x

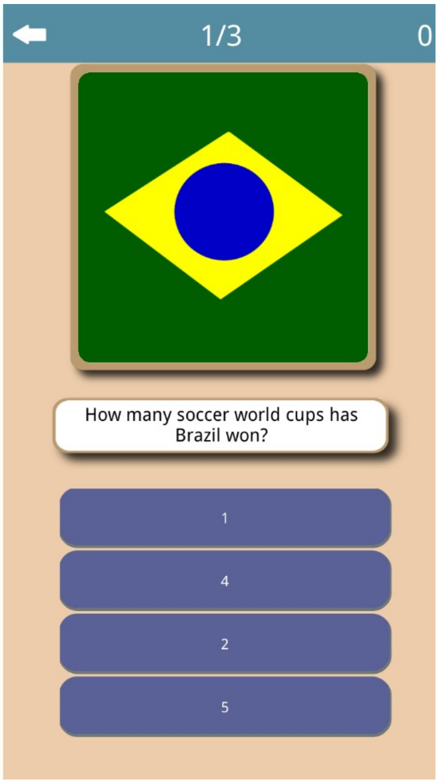
Below are some images of the demo.



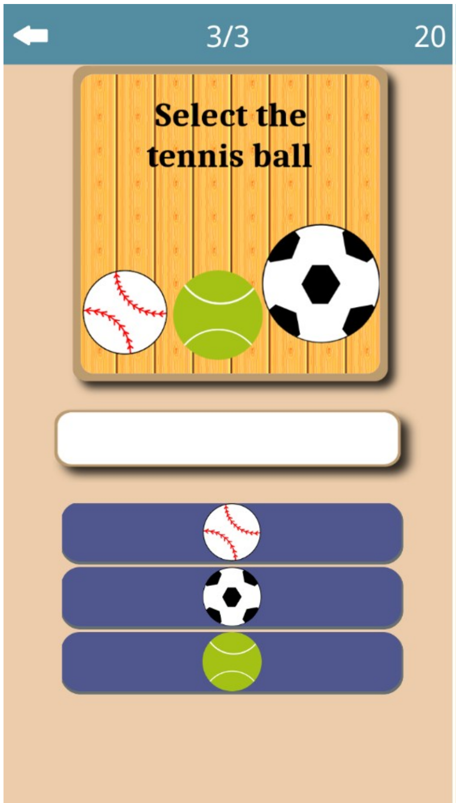




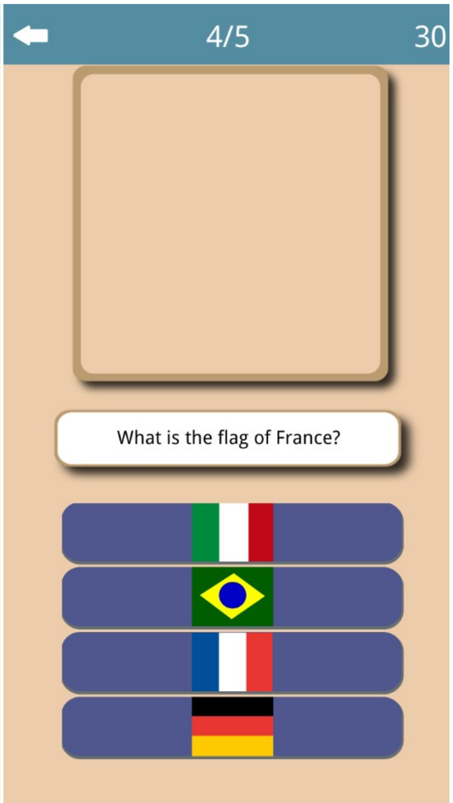
Are chickens Omnivores?



How many soccer world cups has Brazil won?



Select the tennis ball



What is the flag of France?