

# Trivia Template

Documentation | 25-07-2022

SINCE 2010



# Table of Contents

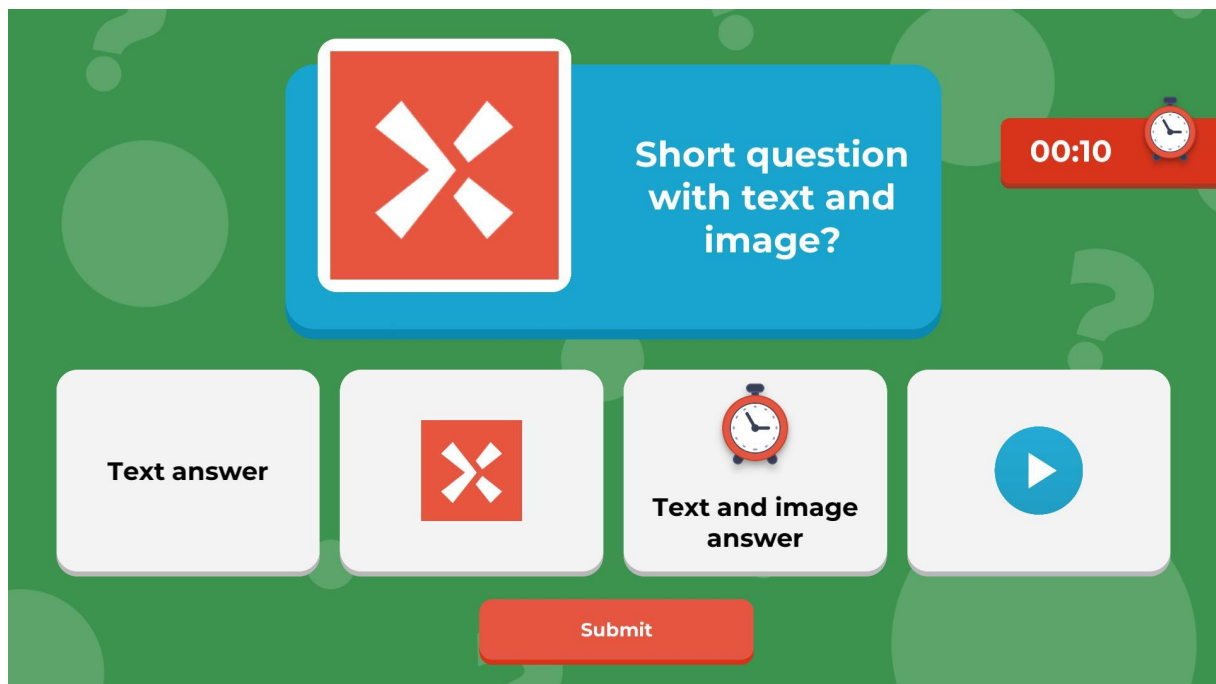
1. Get started quickly	3
2. Introduction	4
3. Set-Up	5
4. Editor	6
QuestionEditor	6
QuestionManagerEditor	8
5. API	10
GameManager	10
QuestionManager	11
6. Known Limitations	12
7. Support and feedback	13

# 1. Get started quickly

1. Create a new **Question** by right-clicking in the project window and selecting *'Create/DTT/MiniGame/Question.'* Or, If you would like to have a question with an image select *'Create/DTT/MiniGame/ImageQuestion.'*
2. Configure the **Question** details and the list of **Answers**. The process is described in detail at [5. Editor](#).
3. Create a **QuestionManager** by right-clicking in the project window and selecting *'Create/DTT/MiniGame/QuestionManager.'*
4. Configure the **QuestionManager** details. All **Questions** are loaded automatically into the list at the bottom. Select which **Questions** you would like to have in your game. You can then see an overview of the selected **Questions** in the ReorderableList in the middle. The process is described in detail at [5. Editor](#).
5. Select the **QuestionManager** you would like to use in your game in the **GameManager** script.

## 2. Introduction

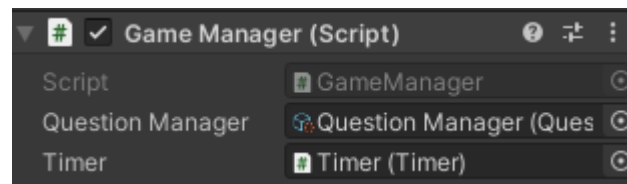
**DTT Trivia** is a Unity asset, which aims to provide you with the foundation needed to create a trivia game. The asset includes a demo scene and UI classes, which give you an example of how you can implement the template in your game. Using **ScriptableObjects** for the **Questions** and the **QuestionManager** allows you to easily configure the game to suit your needs.



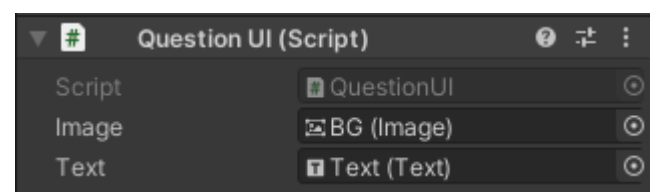
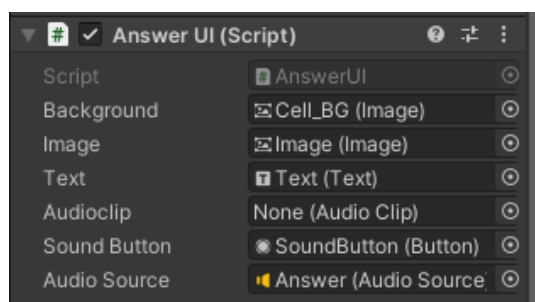
## 3. Set-Up

This section will guide you through the process of implementing a custom trivia game

1. Open a scene in Unity. Create an empty **GameObject** that you are going to use as a game manager. Add the **GameManager** class to it. Now add the required references.
  - a. **Timer** is optional but if you want to have one in your game you can create your own and attach the **Timer** component to it. You can also use the one from the default setup.
  - b. Reference the **QuestionManager** that you are going to use. You can find more details about it at [5. Editor](#).



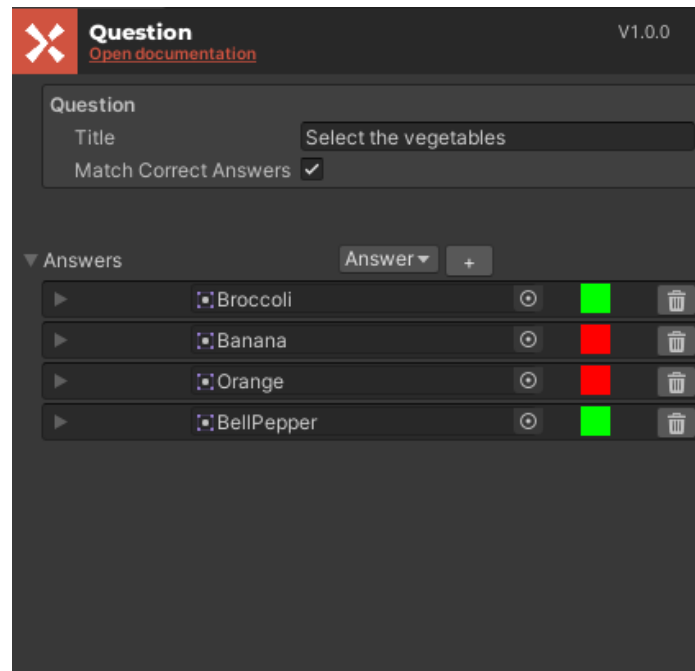
2. Create UI classes for the questions and answers of the quiz. They are going to make use of the data in **Answer** and **Question** to display it in-game. See the scripts **AnswerUI** and **QuestionUI** for an example.



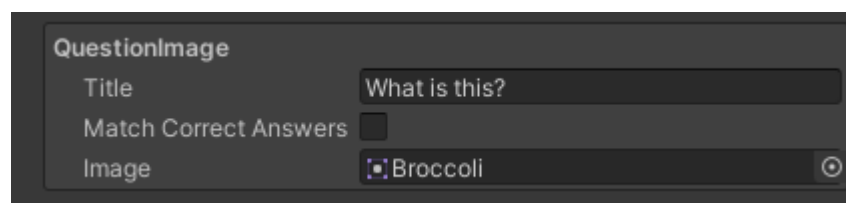
3. Create a UI manager class that will generate the layout you want to have and use the **GameManager** API to control the game. You can find an example of this in **TriviaGameUI**.

## 4. Editor

### QuestionEditor

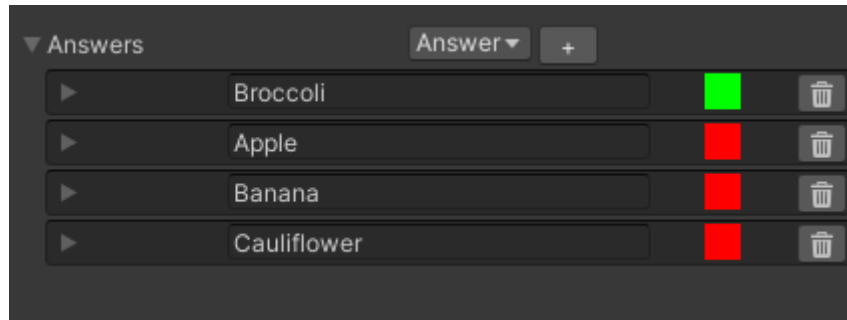


The **Question** ScriptableObject can be used to create and configure new questions for your quiz.



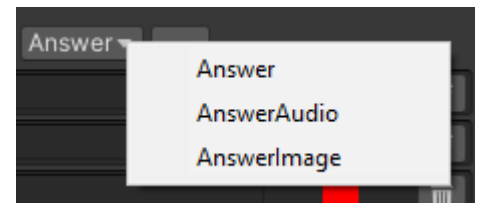
The top panel of the editor is used to fill in the question details.

1. **Title** - this is the text content of the question.
2. **Match Correct Answers** - This check is used in the evaluation of guesses. If you have multiple correct answers to the question, checking this box will force the **GameManager** to only invoke the **CorrectGuess** event if all correct answers have been selected by the user.
3. **Image** - This is where you place your **Sprite** if the question is a **QuestionImage**.



The next editor panel is used to manage the question's answers.

1. On the top, you have a dropdown that you can use to select the type of **Answer** you would like to add.

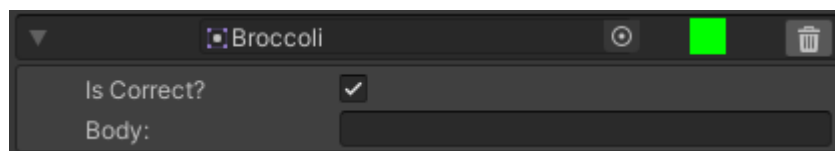


2. Click on the "+" button to create a new **Answer**.
3. Depending on which type you have selected, the **Answer** element will be collapsed and will only show the most important field.

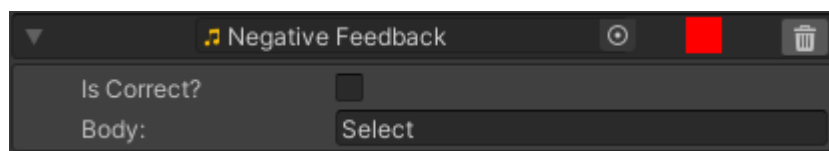
a. **Answer** - Text



b. **AnswerImage** - Sprite

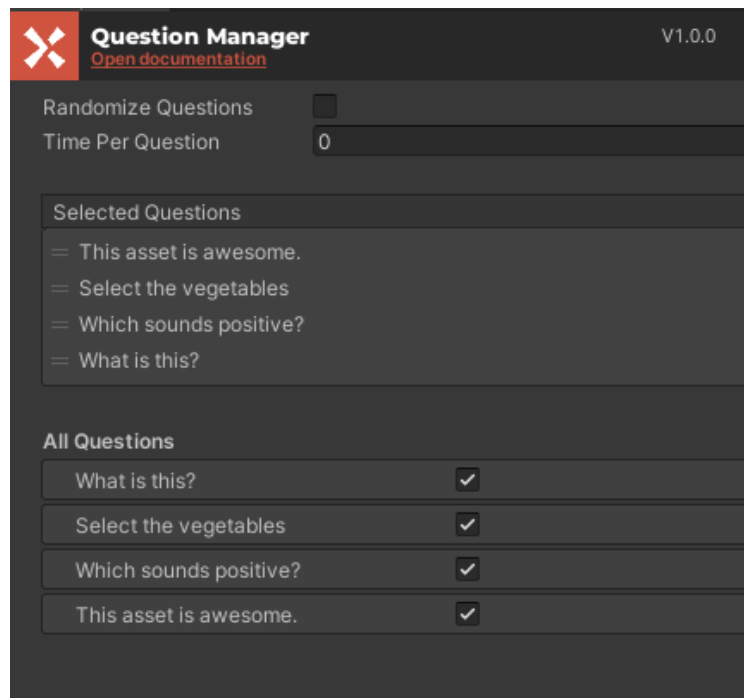


c. **AnswerAudio** - Audioclip

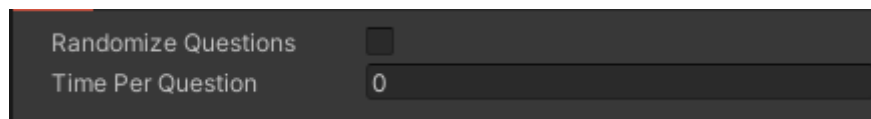


4. The green indicator means that the answer is marked as correct.
5. The trash bin button removes the answer from the list.
6. When expanded, the answers show an "**IsCorrect?**" toggle. When toggled, the square indicator will turn green and the answer will be marked as correct.
7. **AnswerImage** and **AnswerAudio** also have an optional **Body** text field.

# QuestionManagerEditor



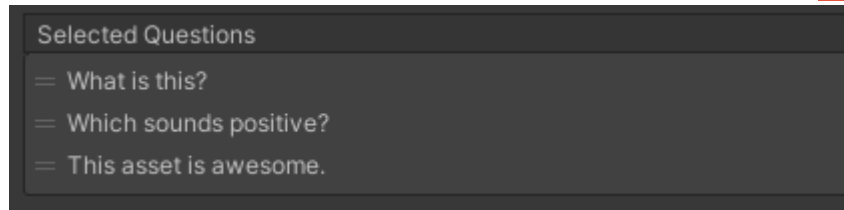
The **QuestionManager** **ScriptableObject** can be used to create and configure new quiz settings.



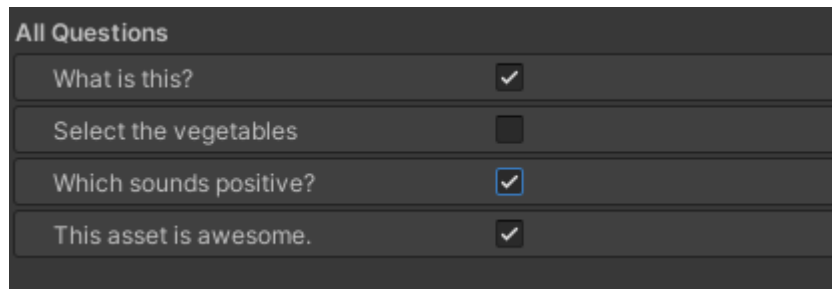
The top panel of the editor:

1. **Randomize Questions** – If this is toggled, the order in which questions are being presented will be random. Otherwise, the flow of questions will be determined by the next panel.
2. **Time Per Question** – The amount of time the player has to answer before the **TimeLimitReached** event is invoked.





The middle panel of the editor is a **ReorderableList**, the order of which is used to present the questions in the quiz. This order is used only when “**Randomize Questions**” is not checked.



The last panel is an overview of all **Question ScriptableObject**s that are found in your project. It is automatically updated. The title of each question is shown with a checkbox next to it that controls whether it is part of the current **QuestionManager** quiz configuration.

## 5. API

### GameManager

*Controls the game flow and handles QuestionManager communication.*

Property Name	Type	Description
GameResults	GameResults	The results controller.
QuestionManager	QuestionManager	Reference to the question manager.
IsPaused	bool	Is true when the game is paused.
IsGameActive	bool	Is true when the game has started and hasn't finished.
CorrectGuess	event	Event invoked when a correct guess has been made.
IncorrectGuess	event	Event invoked when an incorrect guess has been made.
NoQuestionsRemaining	event	Event invoked when there are no questions remaining to be answered.
TimeLimitReached	event	Event invoked when the timer reaches the set limit.
Finish	event	Event invoked when the game has finished.
Started	event	Event invoked when the game has started.

Method name	Return Type	Parameters	Description
StartGame	void		Starts the game using the default game settings.
StartGame	void	QuestionManager questionManager	Starts the game using custom game settings.
Pause	void		Pauses the game.
Continue	void		Continues the game.
ForceFinish	void		Is called when the user wants to force finish the game.
EvaluateGuess	void	List<Answer> guesses	Used to invoke the evaluation events after comparing the guess to the correct answers.
NextQuestion	void		Goes to the next question.

## QuestionManager

*Contains useful information about the current quiz.*

Property Name	Type	Description
TimePerQuestion	float	The time limit for each question.
RandomizeQuestions	bool	Whether to randomize the question order.
AllQuestions	List<Question>	Collection with all questions that are part of this quiz.
AvailableQuestions	List<Question>	Tracks all questions that have not been answered yet.
CurrentQuestion	Question	The current round's question.

## 6. Known Limitations

- The package has to be updated for new types of **Answers** and **Questions** to be possible.

## 7. Support and feedback

If you have any questions regarding the use of this asset, we are happy to help you out.

Always feel free to contact us at:

[unity-support@d-tt.nl](mailto:unity-support@d-tt.nl)

*(We typically respond within 1-2 business days)*

We are actively developing this asset, with many future updates and extensions already planned. We are eager to include feedback from our users in future updates, be they 'quality of life' improvements, new features, bug fixes or anything else that can help you improve your experience with this asset. You can reach us at the email above.

Reviews and ratings are very much appreciated as they help us raise awareness and to improve our assets.

### **DTT stands for Doing Things Together**

DTT is an app, web and game development agency based in the centre of Amsterdam. Established in 2010, DTT has over a decade of experience in mobile, game, and web based technology.

Our game department primarily works in Unity where we put significant emphasis on the development of internal packages, allowing us to efficiently reuse code between projects. To support the Unity community, we are publishing a selection of our internal packages on the Asset Store, including this one.

More information about DTT (including our clients, projects and vacancies) can be found here:

<https://www.d-tt.nl/en/>