



## QuestionnaireToolkit

v1.1

### Documentation

© Pascal Jansen, 25.10.2020

## Table of Contents

1. Setup.....	2
1.1 Setup Questionnaire in a Scene .....	2
1.2 Setup Multiple Questionnaires in a Scene .....	3
1.3 Setup VR Device and Interaction.....	3
2. Usage .....	4
3. QuestionnaireManager .....	4
3.1 Properties .....	4
3.2 Public Methods.....	6
4. QuestionPageManager.....	7
5. Question Items .....	8
5.1 Slider .....	8
5.2 Dropdown.....	8
5.3 Linear Scale.....	9
5.4 Multiple-Choice .....	9
5.5 Multiple-Choice Grid .....	9
5.6 Checkboxes.....	10
5.7 Checkboxes Grid .....	10
5.8 Text Input .....	10
5.9 Text.....	10
5.10 Image .....	11
5.11 Video.....	11
.....	11

5.12 Button.....	11
6. Questionnaire Results .....	12
Academic Citation .....	13

## Introduction

The QuestionnaireToolkit (QT) is an all-in-one tool to create questionnaires, surveys, and simple UI forms for VR, desktop, and mobile applications inside Unity. It eases the implementation of sophisticated questionnaires and accelerates the overall development process of an application as developers can focus on other important app-specific features instead. The toolkit provides 8 different customizable question items as well as full VR support and a VR keyboard. Additionally, a Comma-Separated-Values (CSV) results file can be automatically generated and saved to the filesystem. The UI elements used are based on the standard Unity UI system (UUI), which makes the questionnaire fully compatible with regular UUI coding. Because all creation and editing functionalities are accessible through the Unity Editor, no coding is required to create and run a custom questionnaire in a scene.

## 1. Setup

The QuestionnaireToolkit requires **TextMeshPro**. Make sure to install it and import the **TMP Essential Resources** (Window > TextMeshPro > Import TMP Essential Resources).

### 1.1 Setup Questionnaire in a Scene

To setup a new questionnaire in a scene, follow these steps:

#### Step 1:

Locate the **QuestionnaireManager** prefab inside the prefabs folder (/QuestionnaireToolkit/Prefabs/).

#### Step 2:

Drag and drop the QuestionnaireManager prefab in the scene hierarchy. In case this was done before the TMP Essential Resources were imported, delete the QuestionnaireManager from the hierarchy and add it again to ensure correct TMP rendering.

The Questionnaire Manager will be spawned at world root of the scene. If there is not already an EventSystem, a new one will be automatically generated by the prefab. Additionally, a new empty question page will be created and added to the QuestionnaireManager.

#### Step 3:

Click on the QuestionnaireManager gameObject in the hierarchy to inspect the available settings.

*Note: It may be the case that at first the QuestionnaireManager gameObject is not visible in the scene view. To change that, click on the gameObject in the Hierarchy and press F to focus the scene view on the object.*

## 1.2 Setup Multiple Questionnaires in a Scene

### Step 1:

Locate the **QTManager** prefab inside the prefabs folder (/QuestionnaireToolkit/Prefabs/)

### Step 2:

Click on the QTManager gameObject and use the dedicated questionnaire management UI to add, delete, copy, or edit questionnaires.

*Note: (VR mode only) In case of simultaneous active questionnaires make sure that each has a distinct position in world space. (Any other mode) Only one questionnaire should be visible at a time.*

## 1.3 Setup VR Device and Interaction

In this version of the QuestionnaireToolkit the supported VR device types are: Vive and Oculus. To setup a VR device and to enable the interaction with the questionnaire, follow these steps:

### Vive:

#### Step 1:

Make sure to import the **ViveInputUtility** (VIU) from the Unity Asset Store.

#### Step 2:

Replace the main camera of the scene with the **ViveCameraRig** prefab from the VIU (/HTC.Plugin/ViveInputUtility/Prefabs/).

#### Step 3:

Add the **VivePointers** prefab to the scene (/HTC.Plugin/ViveInputUtility/Prefabs/).

#### Step 4:

Click on the QuestionnaireManager gameObject in the hierarchy and select **VR** from the display mode dropdown and set **Vive** as device type.

#### Step 5:

Click on the **VivePointers** gameObject and unfold its children until the **Reticle** gameObject is visible. Inspect it and set **Auto Scale Reticle** to true. Repeat this for the other controller if you want to use both in your scene.

### Oculus:

#### Step 1:

Make sure to import the **OculusIntegration** from the Unity Asset Store, if not already done.

#### Step 2:

Setup a VR compatible scene with the tools provided by the OculusIntegration package and add the QuestionnaireManager prefab.

#### Step 3:

Click on the QuestionnaireManager gameObject in the hierarchy and select **VR** from the display mode dropdown and set **Oculus** as device type.

### Other:

This setting should be used if the target VR device is not a Vive or an Oculus device. In this case, no automatic UI conversion will happen. If it is necessary, you must do this by yourself.

*Note: Try to follow any official instruction on how to convert a UUI to work with the VR device.*

## 2. Usage

By setting the **StartWithScene** bool field of a **QuestionnaireManager** to **true**, it is possible to let a questionnaire automatically start when the scene is loaded. To manually start a questionnaire this bool has to be set to **false** and the **StartQuestionnaire()** method of a **QuestionnaireManager** needs to be called. If there are multiple questionnaires, the desired questionnaire must be retrieved first.

### Usage Examples:

```
1 usage
private static void SingleQuestionnaireExample()
{
    //Get the QuestionnaireManager script.
    //This only works if there is only one active QuestionnaireManager in the scene!
    var questionnaire = GameObject.FindWithTag("QTQuestionnaireManager").GetComponent<QTQuestionnaireManager>();

    //Start this questionnaire
    questionnaire.StartQuestionnaire();
}

1 usage
private static void MultipleQuestionnairesExample()
{
    //Get the QTManager
    var qtManager = GameObject.FindWithTag("QTManager").GetComponent<QTManager>();

    //Get the desired questionnaire script from the questionnaires list, e.g.: the first one (index is 0)
    var questionnaire = qtManager.questionnaires[0];

    //Start this questionnaire
    questionnaire.StartQuestionnaire();
}
```

## 3. QuestionnaireManager

### 3.1 Properties

The **QTQuestionnaireManager** script is responsible for question page visualization, management, and other settings. It also handles the creation of results files, which are used to document submissions of questionnaire users. In the following each available inspector property is described.

#### Display Settings:

Name	Description	Values	Type
<b>Start With Scene</b>	If true, the questionnaire starts when scene is loaded. Otherwise the questionnaire needs to be started manually.		Bool
<b>Display Mode</b>	Set the display mode of the questionnaire according to the target platform. <b>Desktop/Mobile</b> : ScreenSpace-Overlay Canvas. <b>VR</b> : WorldSpace Canvas.	Desktop Mobile VR	Enum
<b>Device Type</b>	(VR only) Determines which VR config needs to be run.	Vive Oculus	Enum
<b>Orientation</b>	Orientation of a question page. Horizontal aspect ratio: 16:9, Vertical aspect ratio: 10:16.	Horizontal Vertical	Enum
<b>Dynamic Page Height</b>	If true, then each page will always be as high as its content and will not be scrollable anymore.		Bool
<b>Page Height</b>	The height for all pages in pixels. <b>Note: This value is NOT in meters.</b>	>= 900 (Desktop)	Float

	<b>Desktop mode:</b> Keep in mind, that the greater the value, the smaller the UI gets, and it will be difficult to read and click something.	& Mobile Mode)	
<b>Use Custom Transform</b>	(VR only) If true, then you can set the position/rotation of the questionnaire in world space. Otherwise the distance below will be used, and the questionnaire will be positioned in the center of the view.		Bool
<b>Distance To Camera</b>	(VR only) Distance to main camera in meters.	> 0	Float
<b>Page Scale Factor</b>	(VR only) Determines the size of a question page in meters.	> 0	Float
<b>Color Scheme</b>	Sets the color scheme for all pages.	Default Red Green Blue Custom	Enum
<b>Background Transparency</b>	Sets the alpha value of the page background. 0 renders the background opaque and 100 renders it transparent.	0 to 100	Float
<b>Show Bottom Panel</b>	If the bottom panel is deactivated the questionnaire cannot be finished! Use only for basic UI elements!		Bool
<b>Show Page Number</b>	Toggles the visibility of page numbers.		Bool
<b>Show Prev Button</b>	Enables the ability to go back to the previous page.		Bool

### Import/Export Questionnaire:

Allows to save or load entire questionnaires with arbitrary number of question pages and items in **JSON** format.

**Import from:** Select the file you want the questionnaire to be imported from (needs to be in **JSON**). Importing a questionnaire removes all existing question pages from the current selected questionnaire.

**Export to:** Select the location and filename to export the selected questionnaire to.

### Page Management:

The question pages list contains all pages that are currently active inside the questionnaire.

**Reorder:** Drag and drop an element to its new position in the page list.

**Delete:** Deletes the current selected element from the list.

**Copy:** Copies the current selected page with all its items and appends it to the list.

### Results File Settings:

Name	Description	Values	Type
<b>Generate Results File</b>	If true, then the results of a questionnaire run will be saved to a results file.		Bool
<b>Results Save Path</b>	A valid path to an existing folder in the filesystem		String
<b>Results File Name</b>	A custom name of the results file		String
<b>Results File Format</b>	The file format the results file will be saved to.	CSV	Enum
<b>New File Each Start</b>	If true, then a new results file will be created whenever a questionnaire is started. The Results File Name will be used with an additional counter for the file name.		Bool

## Results Visualization:

Name	Description	Values	Type
Runs Per User	Indicates how many times a user should answer this questionnaire.	>= 1	Int
Generate Start Timestamp	Include the timestamp of when the questionnaire was started.		Bool
Generate Finish Timestamp	Include the timestamp of when the questionnaire was finished.		Bool
Overwrite Results Header Items	If true, then the Results Header Items list can be edited. <b>Note:</b> <i>It is strongly recommended to NOT overwrite this list because the list will no longer be updated automatically.</i>		Bool
Results Header Items	A list, which serves as an overview of all question item header names that will be added to the top of the csv file when the results will be recorded.		List<string>

## Additional CSV Items:

In case a questionnaire should log any additional items (e.g. custom integer or string values from other scripts and gameObjects) alongside a questionnaire results file, they must be specified in this list. The order in which these items will be added at the end of the results file is determined by the list item order. Such an item consists of a “Header Name” string and a “Item Value”. The “Header Name” string will be added automatically to the header line. To set the item value, a desired field or property of a gameObject’s script must be selected from the dropdown menu.

## OnQuestionnaireFinished():

Whenever a Questionnaire is finished and submitted by clicking the “Finish” button, each method given in this list will be invoked. This callback can be used to automatically trigger custom application logic or to directly call another questionnaire.

## 3.2 Public Methods

Only such methods are listed that are meant to be accessible by a class outside the QuestionnaireToolkit namespace. The other public methods: CreatePage(), DeletePage(), CopyPage(), ReorderPages(), PrevPage(), NextPage() can still be used, but it is not recommended because there can be unforeseen behaviour.

Name	Description
<b>bool StartQuestionnaire()</b>	Starts the questionnaire with the settings specified in the inspector. Returns false, if there are no question pages. Generates a results file with a csv header line if requested.
<b>bool StartQuestionnaire(Vector3 position, Quaternion rotation, bool restart)</b>	Starts the questionnaire at given position and rotation. If restart ( <i>default value is true</i> ) set to false, the questionnaire will only be made visible but not reinitialized.
<b>void ResetQuestionnaire()</b>	Resets all question items to allow another questionnaire run without restarting the application.

<b>bool AnswersPending()</b>	Checks if there are any answers pending which are marked as mandatory. Returns true if any item is still pending.
<b>void ShowPage(int pageNumber)</b>	Shows the desired page with the given page number ( <b>starting at 0</b> ). Hides all other pages.
<b>void ShowQuestionnaire()</b>	Shows the currently active page of a questionnaire.
<b>void HideQuestionnaire()</b>	Hides the current questionnaire. Can be used to pause a questionnaire run. The answers already made in the current run will remain.

## 4. QuestionPageManager

The **QTQuestionPageManager** script manages all items of a question page. In the following each available inspector property is described.

Name	Description	Values	Type
<b>Show Fullscreen Text</b>	Toggles the visibility of the fullscreen text item that covers the whole page.		Bool
<b>Fullscreen Text</b>	The text that is displayed in the fullscreen area.		String
<b>Show Instruction Panel</b>	Toggles the visibility of the top panel on a question page.		Bool
<b>Instruction Text</b>	The text that is displayed in the top panel area.		String

### Question Items:

The list contains all question items of the page.

**Reorder:** Drag and drop an element to its new position in the item list.

**Delete:** Select an element from the list and press the “-” button in the bottom right of the list.

**Copy:** Copies the selected item including its configuration and appends it to the list.

**Move Item:** Copies and adds the selected item to the desired page. The given page index must exist.

### Question Item Creation:

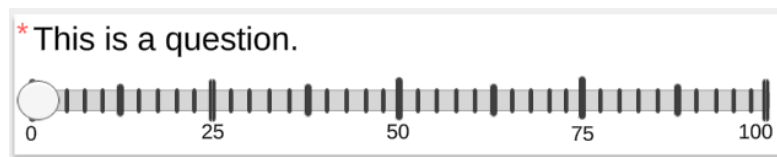
Name	Description	Values	Type
<b>Type</b>	Determines which item will be added when the Create Item button is pressed.	See question items	Enum
<b>Header Name</b>	The name of the item which will appear in the results file as header entry.		String
<b>Question</b>	The displayed question of the item to be created.		String
<b>Automatic Fill</b>	If true, then the newly created item will be automatically filled with some example options.		Bool

## 5. Question Items

There are a few Question Item properties which can always be edited (except for Text, Image, and Button items). For clarity, these properties are listed here.

Name	Description	Values	Type
<b>Answer Mandatory</b>	Bool to determine, if this question items must be answered. If true, then a red star will appear on the left of the question text.		Bool
<b>Header Name</b>	The name of the item which will appear in the results file as header entry.		String
<b>Question</b>	The displayed question of the item to be created.		String

### 5.1 Slider



#### Slider Settings:

Name	Description	Values	Type
<b>Min Value</b>	The min value at which the slider will begin.		Int
<b>Max Value</b>	The max value at which the slider reaches 100% of the slider area.		Int
<b>Whole Numbers</b>	Determines if the slider should snap to whole numbers.		Bool

#### Slider Visuals:

Name	Description	Values	Type
<b>Show Panels</b>	If true, then the lines at 0, 25, 50, 75, and 100 percent are displayed.		Bool
<b>Show Intermediate Panels</b>	If true, then the intermediate lines between 0-25, 25-50, 50.75, and 75-100 percent are displayed.		Bool
<b>Label X</b>	Can be used to have a custom label text for 0, 25, 50, 75, and 100 percent.		String

### 5.2 Dropdown



#### Options:

The options list contains all dropdown options. An option consists of a text and an optional image. The image will be displayed on the left side of the text when the dropdown is clicked at runtime and all the available options are shown to the user.



### 5.3 Linear Scale

\* This is a question.

<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Option 1	Option 2	Option 3	Option 4	Option 5

#### Options:

The options list contains all linear scale answer options. Each time a new option is added, the item resizes all options to fit the available item width. An option can be edited by selecting and edit its values in the **Answer Option** and **Answer Value** fields.

### 5.4 Multiple-Choice

\* This is a question.

- ☐ Option 1
- ☐ Option 2
- ☐ Option 3
- ☐

#### Include Other Option:

Bool, to determine if an “Other” option should be appended to the end of this item. The “Other” option provides a text input where the user can specify his/her choice. But it is also possible to only check the “Other” option without specifying.

#### Options:

The options list contains all multiple-choice answer options. Each time a new option is added, the item increases its vertical height to fit all options. An option can be edited by selecting and edit its values in the **Answer Option** and **Answer Value** fields.

### 5.5 Multiple-Choice Grid

\* This is a question.

	Column 1	Column 2	Column 3	Column 4	Column 5
Row 1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Row 2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Row 3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

#### Grid Content:

**Row Texts:** A list of strings that represents the rows of the grid. A row name can be edited by simply overwriting the string in the list at the right position.

**Column Texts:** A list of string that represents the columns of the grid. A column name can be edited by simply overwriting the string in the list at the right position.

## 5.6 Checkboxes

\* This is a question.

☐ Option 1

☐ Option 2

☐ Option 3

☐

**See 5.4 Multiple-Choice.** The difference is that multiple options can be checked instead of just one.

## 5.7 Checkboxes Grid

\* This is a question.

	Column 1	Column 2	Column 3	Column 4	Column 5
Row 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Row 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Row 3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**See 5.5 Multiple-Choice Grid.** The Difference is that multiple columns inside a row can be checked instead of just one.

## 5.8 Text Input

\* This is a question.

### Placeholder Text:

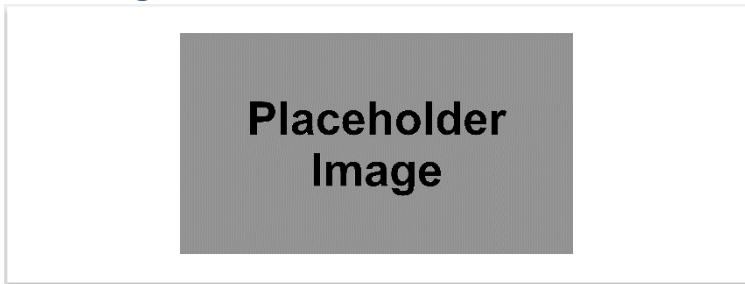
The string which will be displayed inside the text input field, whenever there is no input.

## 5.9 Text

This is a text.

To edit the text and overall style, click on its child gameObject (**TextTMP**) in the hierarchy. All **Text Mesh Pro** settings are available there, which can be used to edit the text appearance. In case the vertical item space is not enough to fit the text, it is possible to set a custom height in the **RectTransform** component of the item.

## 5.10 Image



To edit the displayed image, replace the source image of the Image component in the item child (**ImageArea**). At first the image will not be optimally scaled. For this reason, it may be necessary to set the **Left** and **Right** offset in the **Rect Transform** of the **ImageArea** gameObject.

## 5.11 Video



To set the displayed video, add a video to the video clip field of the Video Player component in the item child (**VideoArea**). To scale the video, edit the **Left**, **Right**, **Top**, and **Bottom** offset in the **Rect Transform** of the **VideoArea** component.

## 5.12 Button



The Button item allows the use of custom UI buttons in a page. With the “Add Button” button in the **QTButton** script, it is possible to add additional buttons. The buttons will always be placed inside a Horizontal Layout Group. In order to set the displayed text and to define **OnClick()** callbacks, click on a button child and edit its settings. To delete any button, it is necessary to remove it from the gameObject Hierarchy (Right click > Delete).

## 6. Questionnaire Results

The results are exported in CSV format and saved to the specified path in the file system. There are some entries which will or can be automatically added to the results file.

### **Response\_id:**

Unique id which is always added as first item to identify the results of a specific user (in this case a row). The id that will be added next is stored in the QTMetaData scriptable object in the Scripts folder.

### **User\_id (optional):**

A custom user\_id to identify a user based on specific properties, research conditions, and to encode a treatment order, like e.g. "12ABC" or "13BCA". Must be entered by hand.

*Note: If used in play mode, user\_id will be set to an empty string after a user completed all his runs.*

### **Run (optional):**

Counter starting at 1 which identifies a single run of a user (in case there are more than one). The response\_id will stay the same until the run counter is equals the specified count in the Questionnaire Manager.

### **Started (optional):**

Timestamp indicating the time the questionnaire run was started (UTC).

### **Finished (optional):**

Timestamp indicating the time the questionnaire run was finished (UTC).

Hereinafter, the custom items are added to the header row. Each header item name is escaped by quotation marks, e.g. "Item 4" to allow the use of a comma inside a header item name. This applies for any answer which can contain text. Answers which must be numbers are added without quotation marks. In the following the result type of each question item is listed.

Name	CSV Value
Slider	Number at which the slider was left at questionnaire submission
Dropdown	The selected option name (quoted)
Linear Scale	Index of the selected option (starting at 1)
Multiple-Choice	Index of the selected option (starting at 1)
Multiple-Choice Grid	Index of the selected column within a row (quoted, starting at 1)
Checkboxes	Index of the selected option (starting at 1). In case of multiple checks, they are concatenated, e.g. 1;2;4
Checkboxes Grid	Index of the selected column within a row (quoted, starting at 1)
Text Input	The entered text (quoted)

### **Results Save Path:**

The results save path depends on the used target platform. For more information see:

<https://docs.unity3d.com/ScriptReference/Application-persistentDataPath.html>

## Academic Citation

If the QuestionnaireToolkit is used for academic purposes, feel free to cite it in your paper:

```
@software{questionnaire_toolkit,  
  author = {Pascal Jansen and Fabian Fischbach},  
  title = {QuestionnaireToolkit for Unity},  
  url = {https://zefwih.com},  
  version = {1.0},  
  date = {2019-12-15},  
}
```