



Worksheet 1

Google Account & Connecting to AppInventor

Activity: You will create a Google Account and use it to login to the AppInventor site.

Time: 15 Minutes

App Inventor was developed by Google and is now owned and developed by the Massachusetts Institute of Technology (MIT) and can be accessed through this website: <http://appinventor.mit.edu/explore/>.

You need three things to use App Inventor:

- *A computer*—Windows, Macintosh, or GNU/Linux.
- *An internet connection and browser*—Chrome, Firefox, or Safari. (Internet Explorer support is planned for the future.)
- *A Google account*—Free, and available at <https://accounts.google.com/signup>. Note that to set up your own account, you have to be age 13 or older in most countries. If you’re under 13, a parent can sign up for an account and work alongside you. (We know—the kids will teach the grownups!)
- Android phone or tablet

(Source: Book “Hello App Inventor!” Beer P., Simmons C., Publications *Manning Shelter Island*)

TASK	DONE?
<p>Google account</p> <p>Should you already have a Google Account, consider helping out your fellow students who don’t. Otherwise, open a browser and type in https://accounts.google.com/SignUp .</p>	<input type="checkbox"/>
<ul style="list-style-type: none"> • Fill in the required information. 	<input type="checkbox"/>
<ul style="list-style-type: none"> • You should also have an active phone number so as to verify your account. 	<input type="checkbox"/>
<ul style="list-style-type: none"> • Write down your username and password for future reference. 	<input type="checkbox"/>
<p>Connecting to AppInventor</p> <ul style="list-style-type: none"> • In the address bar of your browser type http://ai2.appinventor.mit.edu . 	<input type="checkbox"/>
<ul style="list-style-type: none"> • After this you will be presented with the environment of AppInventor. 	<input type="checkbox"/>



Worksheet 2 (part A)

AppInventor Environment

Activity: You will be introduced to the main parts of AppInventor.

Time: 10 Minutes

Sign in with your Google account at <http://ai2.appinventor.mit.edu>

In all the App Inventor apps you'll build, you'll use the same three steps with three different screens:

1. **Design** the app screen by using the *app Designer*.
2. **Tell** the app **what to do** by programming the *Blocks Editor*.
3. **Test** the program using your *phone* or *emulator*.

Designer

The AppInventor Designer lets you

- a) *Create* a new project,
- b) *Add* components and
- c) *design* the look of the app.

Blocks Editor

- The Blocks Editor lets you *control* how the app works by using *programming blocks*. (You can toggle between Designer and Blocks as you program.)

Testing^(*)

- Finally, you'll test the program on your phone.
- You can see if the app looks and works OK.

^(*) Testing on your smartphone

- You will need an Android smartphone to download from Google Play the "MIT AI2 Companion"





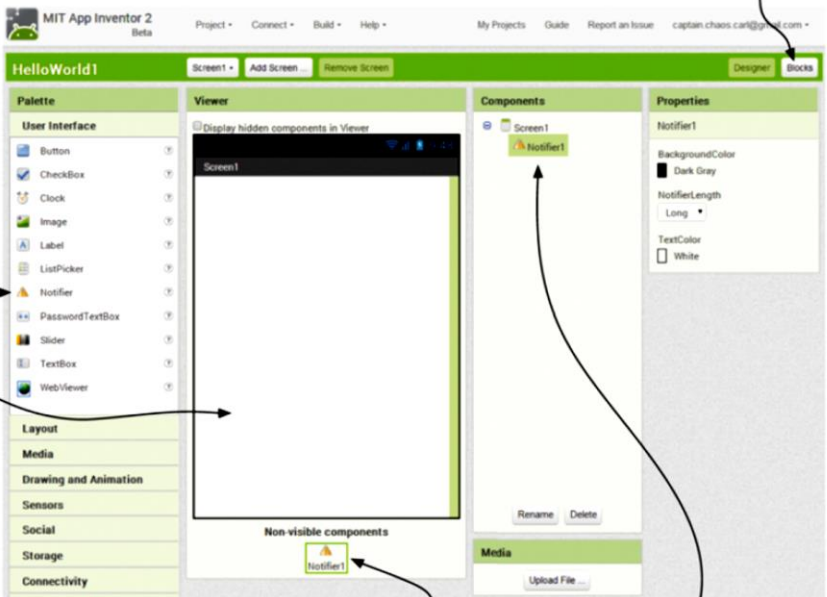
Worksheet 2 (part B)

My first app: Hello World!

Activity: You will create an app that displays a pop-up message saying “Hello World!”

Time: 30 Minutes

*Follow your facilitator and complete the tasks below (put a mark if completed).
Don't hesitate to ask if there is something you are not sure of.*

TASK	DONE?
<p>Starting a new project</p> <ol style="list-style-type: none"> From the menu click “Project” and then choose “Start New Project”. Enter the project name, <code>HelloWorld</code>, and then click OK. 	
<p>TIP:</p> <ul style="list-style-type: none"> No spaces are allowed in project names or component names. One way to separate words is to use <i>camel case</i>, i.e. capital letters to indicate the start of each word. App Inventor projects are automatically saved every few seconds as you work on them. 	
<p>Adding a Notifier component to the project</p> <p>You can use the useful <code>Notifier</code> component for all kinds of pop-up messages and warnings.</p> <p>3. You're finished in the Designer, so open the Blocks Editor by clicking the Blocks button.</p>  <p>1. From the Palette, drag the <code>Notifier</code> component onto <code>Screen1</code> here.</p> <p>Tip: If you can't spot <code>Notifier</code> in the Palette, make sure you're looking in the User Interface section.</p> <p>2. You should see the <code>Notifier</code> icon appear in the Components list above and also down here.</p>	



Writing the program using blocks

1. Click Notifier1.

2. Drag the Notifier1.ShowMessageDialog block from the sidebar into the main work area.

Now you join the blocks

To join the blocks, drag the purple Notifier block into the space in the Screen1.Initialize block.

Have you noticed the empty jigsaw slots in the Notifier block? These are three extra pieces of information that the notifier needs before it will work. Here you can see how they affect the phone's pop-up notifier.

The message, title, and button text that the notifier needs are all pieces of text—in programming, we call them *text strings*. To make text strings, you use a Built-in Text block.

1. Click the empty Text block from the Built-in blocks.
2. Drag three empty Text blocks into the three empty Notifier1 slots.

You can set a text string by clicking between the quote marks.

Change the three text strings to what is shown here.



TIP:

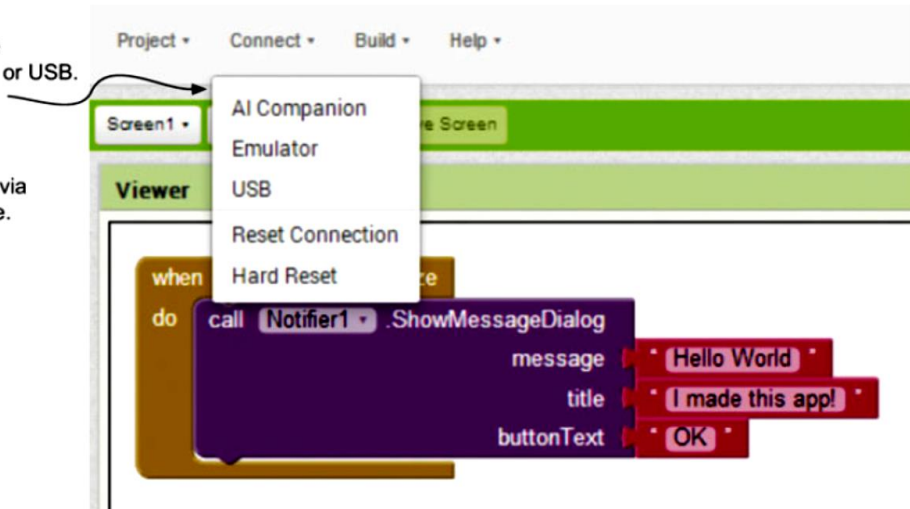
Before testing the app you should download the “MIT AI2 Companion” from Google Play.

Testing the app

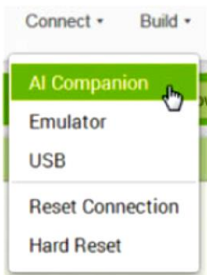
Connect your smartphone or tablet from the Connect menu at the top of your screen.

1. Click Connect. Choose AI Companion, Emulator, or USB.

2. You can be connected via only one method at a time.



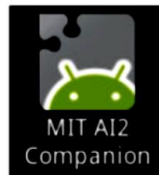
- Connect your smartphone on the Wi-Fi network.
- Choose AI Companion. A code will appear on your computer screen. You can enter or scan this code into your phone by running the AI Companion phone app.



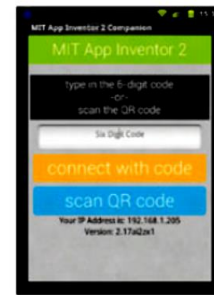
1. Choose AI Companion from the Connect menu.



2. A unique app code appears in both QR and text form.



3. On your phone, start the MIT AI2 Companion app.



4. Either type the app code and click Connect with Code or click Scan the QR Code and point your phone at the QR code on your computer screen.

(Source: Book “Hello App Inventor!” Beer P., Simmons C., Publications Manning Shelter Island)



Worksheet 2 (part C)

Adding functionality: Hello World! App, version 2

Activity: You will add a button to a new copy of your app.

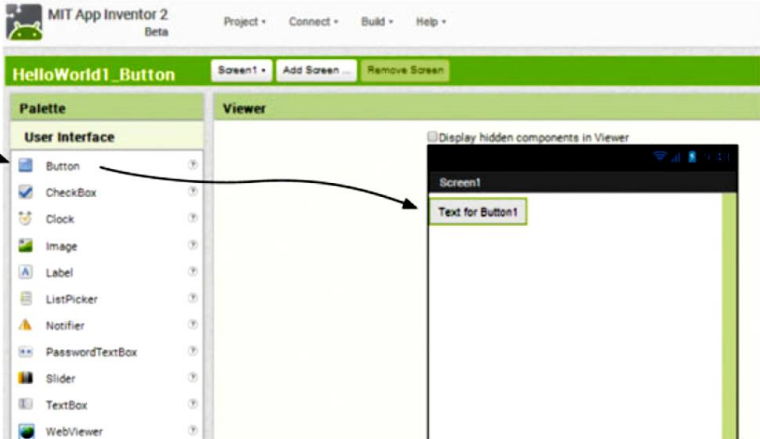
Time: 15 Minutes

The problem in the Hello World! App is that the `Screen1.Initialize` event handler you use to trigger the app only happens once: when the app first runs.

It would be better to have an event like a *button press* that the user can touch to trigger the event handler whenever they want to see the message.

Follow your facilitator and complete the tasks below (put a mark if completed).

Don't hesitate to ask if there is something you are not sure of.

TASK	DONE?
<p>1. Saving a new copy of Hello World!</p> <p>You call the new project <code>HelloWorld1_Button</code> so that when you look at the list of projects, you know what makes this version different from any others. Giving items sensible names will help you later when things get more complicated and you've created lots of different projects.</p> <ol style="list-style-type: none"> 1. Menu Project → Save As... 2. Call your new project <code>HelloWorld1_button</code> 3. Click OK. 	
<p>2. Adding a Button component</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1; padding-right: 20px;"> <p>In the Designer screen, click and drag a button from the Palette onto your phone screen.</p> </div> <div style="flex: 2;">  </div> </div> <p>You need to tell the user what to do with the button. Change the button to say <code>Click Me!</code></p>	



You can change the text that appears on the button by changing its `Text` property here.

3. Programming the blocks

1. Click Button1.

2. Drag a Button1.Click event block into the work area under your Screen1.Initialize code block.

Now drag the **Notifier** block's code from Screen1.Initialize into Button1.Click:

1. Drag the purple Notifier1 block from Screen1.Initialize into the new Button1.Click event.

2. You don't need the empty Screen1.Initialize event any more, so to clean up, drag it to the trash can.



4. Testing your app Connect your smartphone or tablet by choosing from the Connect menu “AI Companion” at the top of your screen.	
Taking it further <ol style="list-style-type: none">1. Add a new button and notifier to your app. Make the notifier say “Goodbye World” if the user clicks the second button.2. Try changing the color properties of Screen1 and Button1. Also change the Font and Size properties of <code>Button1</code>.	
Don't forget to Sign out from AppInventor once you're finished with your work.	

(Source: Book “Hello App Inventor!” Beer P., Simmons C., Publications *Manning Shelter Island*)



Worksheet 3

Designing the User Interface (UI)

Activity: You will design drafts of the Quiz User Interface.

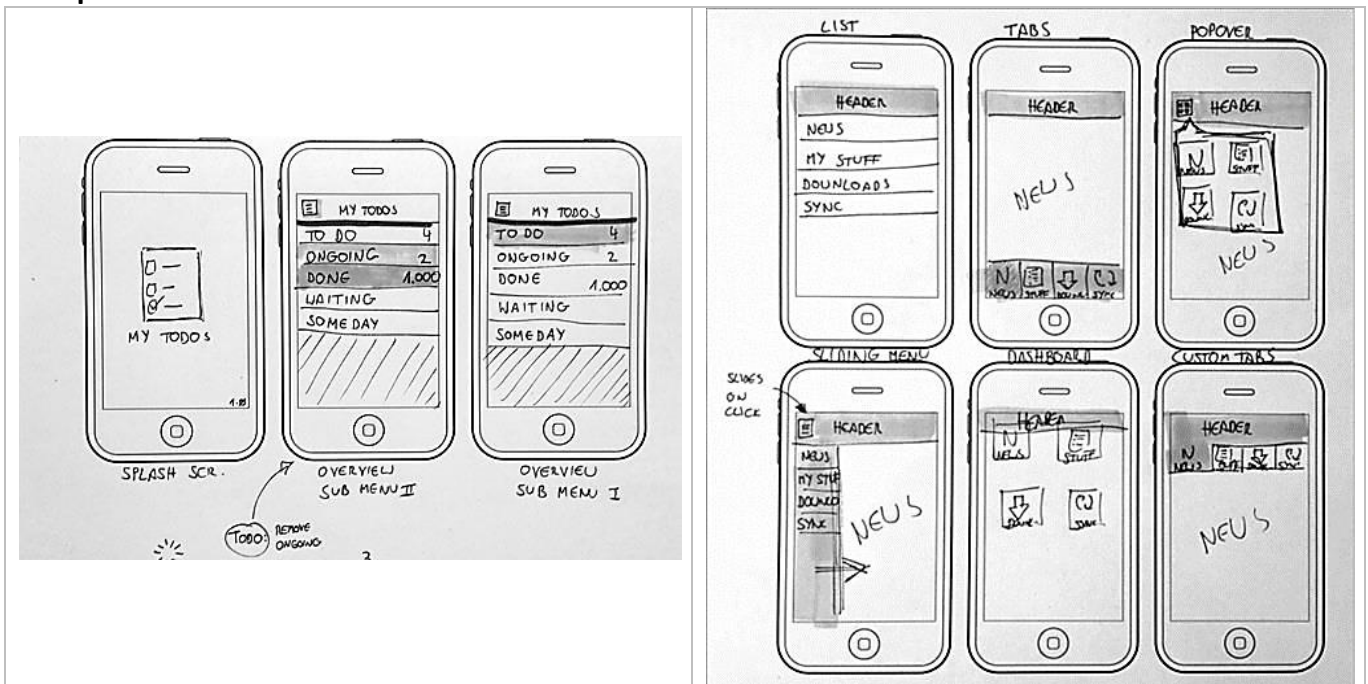
Time: 30 Minutes

The User Interface (UI) is about

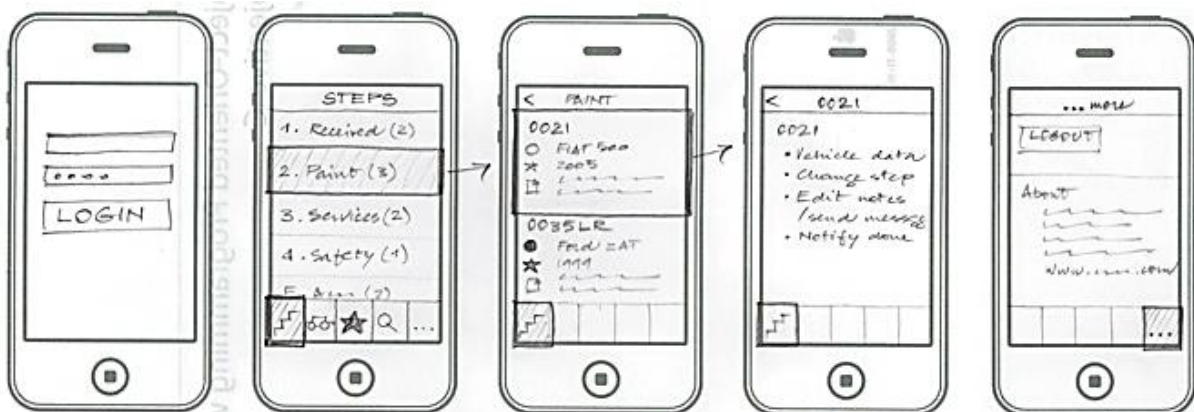
- How the app looks – graphic design, typography (fonts), and color
- How people tell the app what they want to do
- How the app gives information back to the user through the screen, speakers, or *haptic* feedback (buzzing the phone’s little motor)

(Source: Book “Hello App Inventor!” Beer P., Simmons C., Publications Manning Shelter Island)

Examples



(Source: <https://www.smashingmagazine.com/2013/06/sketching-for-better-mobile-experiences/>)



(Source: <http://lrcommunicationdesign.com/RapidRecon-web-application-design-example.html>)



*Follow your facilitator and complete the tasks below (put a mark if completed).
Don't hesitate to ask if there is something you are not sure of.*

TASK	DONE?
<p>Assigning Groups</p> <ul style="list-style-type: none"> • You will be assigned in intercultural-mixed groups. • Think of a <u>name</u> for your team. • Choose <u>who</u> will present your work. 	
<p>User Interface of the app</p> <ul style="list-style-type: none"> • How should the <u>first screen</u> of the Quiz app look like? • What functions should it have? • Brainstorm your ideas inspired from other smartphone applications. • Make drafts on the template provided. Ask for more copies if you need. 	
<p>Present your results</p> <ul style="list-style-type: none"> • Present your ideas to the other group members. • Explain your thoughts. 	





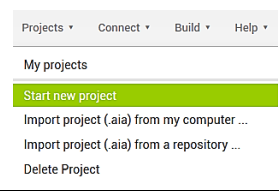
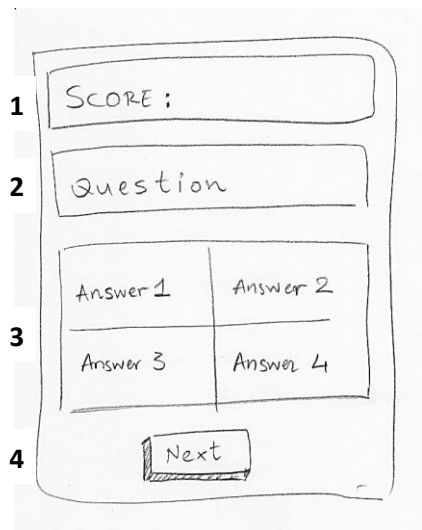
Worksheet 4 (Part A)

Creating a simple Quiz

Activity: You will create a simple quiz with two questions.
First, you will **design the interface** of the app screen in AppInventor.

Time: 35 Minutes

*Follow your facilitator and complete the tasks below (put a mark if completed).
Don't hesitate to ask if there is something you are not sure of.*

TASK	DONE?
<p>Assigning Teams</p> <ul style="list-style-type: none"> To accomplish this task you have to team up with your host/guest fellow student. 	
<p>Connecting to AppInventor</p> <ul style="list-style-type: none"> Open a browser, e.g. Firefox, Chrome, MS Edge, Safari. In the address bar of your browser type http://ai2.appinventor.mit.edu. Sign in with your Google Account. After this you're presented with the AppInventor environment. 	
<p>Starting a new project</p> <ol style="list-style-type: none"> From the menu click "Projects" and then choose "Start New Project". Enter the project name, SimpleQuiz, and then click OK. 	
<p>Designing the interface</p> <ul style="list-style-type: none"> Make a simple draft of your Quiz interface on paper. <p><i>How do you want it to look like? What components should it include?</i></p> <ul style="list-style-type: none"> We will use the "Layout" tools to arrange all the components on the screen (points 1, 2, 3 and 4). <p>Choose and drag the components from the Palette category "Layout" in the following order:</p> <p align="center">point 1: HorizontalArrangement point 2: HorizontalArrangement point 3: TableArrangement point 4: HorizontalArrangement</p>	



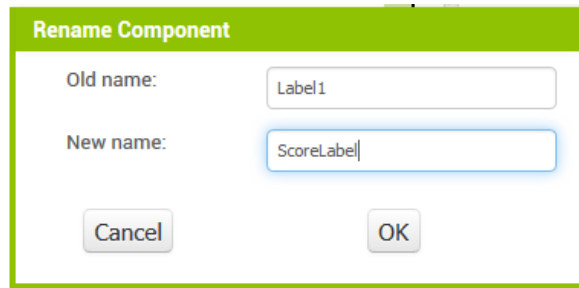
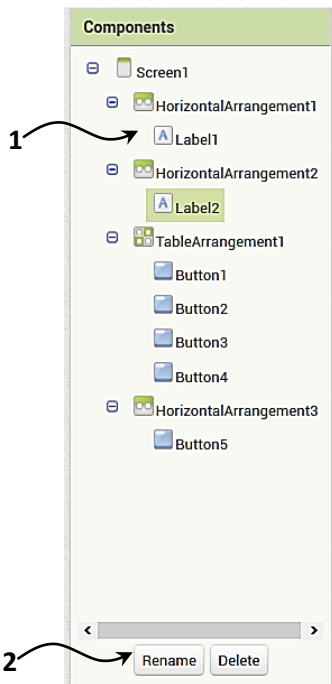
- For every component drawn on the screen change the “Width” to **Fill Parent**.



What do you notice?

- We need “Labels” for text output, such as the *score* and the *question*. Then, we want the user to “click on” the answers while choosing, so we decide to draw buttons for our 4 possible answers. We also need a “Button” to click in order to proceed to the next question.
- Drag from the Palette the appropriate components to the screen, either **Label1** or **Button**, as shown below.

- Change the names of the labels and the buttons in “Components” by clicking first the name and then “Rename” as shown on the next page.



Repeat this for the label and button components as follows:

Label1 → **ScoreLabel**

Label2 → **QuestionLabel**,

Button1 → **AnswerButton1**,

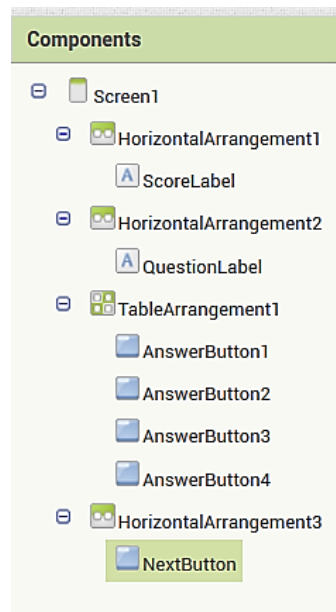
Button2 → **AnswerButton2**,

Button3 → **AnswerButton3**,

Button4 → **AnswerButton4**,

Button5 → **NextButton**

- At the end it should look like this



Changing Properties

General Rule: First click on the component, then change/edit the properties of it.

- Change the “FontSize” of the **QuestionLabel** to **18.0**
- Change the “TextColor” for **every AnswerButton** to **blue**
- Change the “AlignHorizontal” of the **HorizontalArrangement3** to **Center**.



- Change the “Text” property of every component to look like as below

Screen1	
Score:	
Question	
answer1	answer2
answer3	answer4
NEXT	





Worksheet 4 (Part B)

Creating a simple Quiz

Activity: You will program the components of your app “to tell them what to do”.

Time: 35 Minutes

*Follow your facilitator and complete the tasks below (put a mark if completed).
Don't hesitate to ask if there is something you are not sure of.*

TASK	DONE?
<div style="display: flex; align-items: center; margin-bottom: 10px;">  <p>“What should the app do?”</p> </div> <p>First, we have to think about how we want the app to “behave”.</p> <p>How should the components interact? For example,</p> <ul style="list-style-type: none"> When the screen initializes (the app starts) the first question and 4 possible answers should be shown. Then the user clicks on an answer. The app evaluates if it's correct and ...gives feedback by changing the color of the correct answer button to GREEN Maybe all the other buttons should be deactivated to prevent the user on trying different choices. If the answer is correct the score should be updated “+1”. <p>Finally, the user can move to the next question.</p>	
<p>Blocks editor</p> <p>Switch to the BLOCKS editor (upper right corner).</p> <p>Create/Initialize 3 variables: the score, a questionList and an answerList</p> <div style="display: flex; justify-content: space-around; margin: 10px 0;"> <div style="border: 1px solid #ccc; padding: 5px; width: 45%;"> <p style="font-size: small; margin: 0;">Blocks</p> <ul style="list-style-type: none"> Built-in Control Logic Math Text Lists Colors <li style="background-color: #e0e0e0;">Variables Procedures Screen1 HorizontalArrangemen </div> <div style="border: 1px solid #ccc; padding: 5px; width: 45%;"> <p style="font-size: small; margin: 0;">Viewer</p>  </div> </div> <ul style="list-style-type: none"> When the app starts, we initialize our score to 0, ...create a list with one question “What is the capital of Spain?” and create a second list within a list of possible answers. 	



Follow the instructions as shown below.

```

initialize global score to 0
initialize global questionList to make a list " What is the capital of Spain? "
initialize global answerList to make a list make a list " Sevilla " " Madrid " " Cordoba " " Alicante " 2
    
```



Why did we create the *answerList* as "a list within a list"?



What does the number "2" represent in the list of possible answers?

Now you have to **set** the texts of the **QuestionLabel** and all the **AnswerButton**'s.

Follow the instructions as shown below.

```

when Screen1 .Initialize
do
  set QuestionLabel .Text to select list item list get global questionList index 1
  set AnswerButton1 .Text to select list item list select list item list get global answerList index 1 index 1
  set AnswerButton2 .Text to select list item list select list item list get global answerList index 1 index 2
  set AnswerButton3 .Text to select list item list select list item list get global answerList index 1 index 3
  set AnswerButton4 .Text to select list item list select list item list get global answerList index 1 index 4
    
```



Selecting a list item

We select a list item with the use of an “index number”.

Example:

The `questionList` has only one item.

This item holds the text “*What is the capital of Spain?*”

If you want to use the text of this item (1st in the list) you have to use the following programming block with the index number 1.

```
initialize global questionList to make a list "What is the capital of Spain?"
```

```
set QuestionLabel Text to select list item list index 1 get global questionList
```



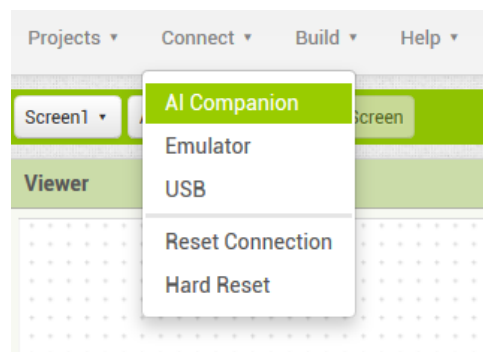
What is the index number for the item “Madrid”? How do you refer to it?

Testing the app

TIP: Before testing the app you should download the “MIT AI2 Companion” from Google Play.

Connect your smartphone or tablet from the Connect menu at the top of your screen.

1. Connect your smartphone on the Wi-Fi network.
2. Choose AI Companion from the Connect menu
3. A unique app code appears in both QR and text form.
4. On your phone start the MIT AI2 Companion app.
5. Either type the app code and click “Connect with Code” or click “Scan the QR Code” and point your phone at the QR code on your computer screen.



Try your app by pressing the buttons.



Do the buttons have some functionality?





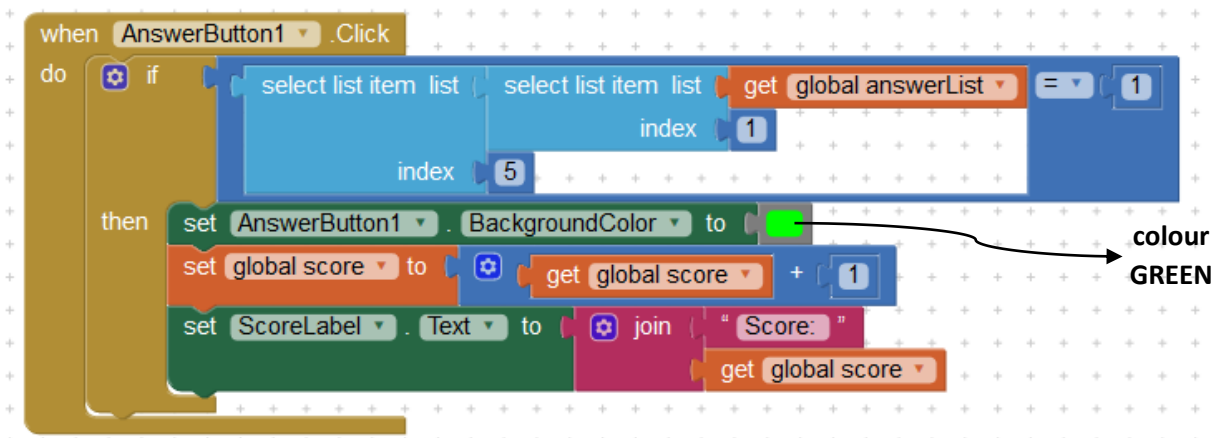
Worksheet 4 (Part C)

Creating a simple Quiz

Activity: Adding functionality to the buttons.

Time: 15 Minutes

Follow your facilitator and complete the tasks below (put a mark if completed).
 Don't hesitate to ask if there is something you are not sure of.

TASK	DONE?
<div data-bbox="124 678 226 757" data-label="Image">  </div> <p data-bbox="231 728 651 763">“What should the buttons do?”</p> <p data-bbox="124 784 1302 819">If the user clicks on an Answer button, then the answer should be evaluated if it's correct.</p> <p data-bbox="218 840 1441 875">The correct answer is given by the number (in this example 2) in the list of possible answers.</p> <div data-bbox="231 884 574 1086" data-label="Image">  </div> <ul data-bbox="172 1075 1358 1220" style="list-style-type: none"> • • If the user clicks on the correct Answer button, <ul data-bbox="268 1142 1358 1178" style="list-style-type: none"> ◦ then a) it should change its background to green and b) update the score to +1. • If the user clicks the wrong button, then do nothing. 	
<p data-bbox="124 1256 657 1292">Follow the instructions as shown below</p>	
<div data-bbox="124 1344 1340 1780" data-label="Image">  </div>	



<pre> when AnswerButton2 .Click do if select list item list index 5 select list item list index 1 get global answerList = 2 then set AnswerButton2 . BackgroundColor to colour GREEN set global score to get global score + 1 set ScoreLabel . Text to join "Score: " and get global score </pre>	
<p>Repeat the above for the remaining two (2) AnswerButtons.</p> <p>TIP: To save time <u>Right Click</u> on the <u>outer block</u> and <u>duplicate</u> it.</p> <p>Important: Make the necessary changes.</p>	




Worksheet 4 (Part D)

Creating a simple Quiz

Activity: Adding a second question with four possible answers.

Time: 10 Minutes

*Follow your facilitator and complete the tasks below (put a mark if completed).
Don't hesitate to ask if there is something you are not sure of.*

TASK	DONE?
<div style="display: flex; align-items: flex-start;">  <p>“Think of a new question”</p> <p>Write down your question and 4 possible answers.</p> </div>	
<p>Following the previous instructions (as in Worksheet 4 part B) to add your question and answers in the programming blocks.</p>	




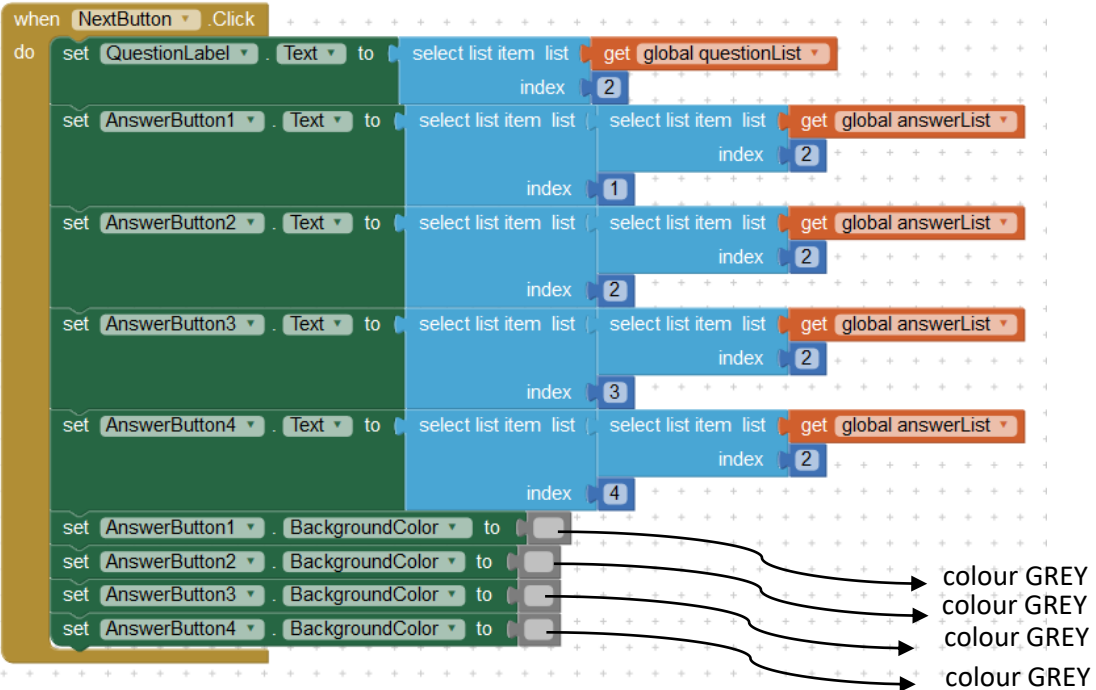

Worksheet 4 (Part E)

Creating a simple Quiz

Activity: Programming the NEXT button.

Time: 15 Minutes

Follow your facilitator and complete the tasks below (put a mark if completed).
Don't hesitate to ask if there is something you are not sure of.

TASK	DONE?
<div data-bbox="124 678 226 757">  </div> <p data-bbox="231 728 1093 766">“What actions should be taken after pressing the NEXT button?”</p> <p data-bbox="124 786 1161 824">To programme the next screen we have first to “initialize” certain components.</p> <ol data-bbox="172 842 1225 958" style="list-style-type: none"> 1. Set/Load the QuestionLabel to the new question 2. Set/Load the new answers into the AnswerButton's. 3. Un-color the green button, i.e. all buttons should have the same grey colour. 	
<p data-bbox="124 992 667 1025">Follow the instructions as shown below.</p>	
 <p>The image shows a Scratch script for the 'NextButton' click event. It starts with a 'do' block containing several 'set' blocks. The first 'set' block sets the 'Text' of 'QuestionLabel' to the 'list item' at 'index 2' of 'global questionList'. The next four 'set' blocks set the 'Text' of 'AnswerButton1' through 'AnswerButton4' to the 'list item' at 'index 2', '1', '2', and '3' respectively of 'global answerList'. The final four 'set' blocks set the 'BackgroundColor' of 'AnswerButton1' through 'AnswerButton4' to 'colour GREY'. Arrows point from the 'colour GREY' text to the 'to' field of each of these four 'set' blocks.</p>	
<p data-bbox="124 1798 418 1832">Test your application.</p>	
<div data-bbox="124 1865 178 1944">  </div> <p data-bbox="183 1917 1037 1955"><i>Does your application function properly? Are there any problems?</i></p> <p data-bbox="124 1973 550 2011"><i>How would you deal with them?</i></p>	





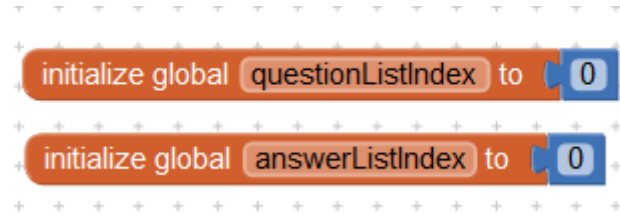
Worksheet 4 (Part F)

Creating a simple Quiz

Activity: Recognizing and resolving problems.

Time: 15 Minutes

*Follow your facilitator and complete the tasks below (put a mark if completed).
Don't hesitate to ask if there is something you are not sure of.*


TASK	DONE?
<p>“Problem identification”</p> <ul style="list-style-type: none"> We move to the next question but the the next set of answers isn't loaded. The app doesn't highlight the correct answer (green button) when clicked. Score is not incremented. 	
<p>“Problem resolving”</p> <ul style="list-style-type: none"> To resolve the above issues we have to make use of variables. 	
<div data-bbox="124 1016 226 1099" data-label="Image">  </div> <p>Variables</p> <p>Variables are “containers” of information that can be manipulated.</p> <p>In AppInventor we use the built-in programming blocks “Initialize”, “Get” and “Set” for this purpose.</p> <div data-bbox="124 1267 593 1532" data-label="Image">  </div>	
<p>We will create a new variable called “questionListIndex” to hold the index number of the questionList and a “answerListIndex” to hold the index number for the answerList.</p> <div data-bbox="124 1659 746 1872" data-label="Image">  </div>	
<p>We also have to <u>edit</u> the programming blocks to accommodate this change.</p> <p>Follow the instructions as shown below.</p>	



<pre> when Screen1.Initialize do set global questionListIndex to get global questionListIndex + 1 set global answerListIndex to get global answerListIndex + 1 set QuestionLabel.Text to select list item list get global questionList index get global questionListIndex set AnswerButton1.Text to select list item list select list item list get global answerList index get global answerListIndex 1 set AnswerButton2.Text to select list item list select list item list get global answerList index get global answerListIndex 2 set AnswerButton3.Text to select list item list select list item list get global answerList index get global answerListIndex 3 set AnswerButton4.Text to select list item list select list item list get global answerList index get global answerListIndex 4 </pre>	<p>Add new instructions</p> <p>Get <u>first</u> question</p> <p>Get <u>first</u> answer choice</p> <p>Get <u>second</u> answer choice</p> <p>Get <u>third</u> answer choice</p> <p>Get <u>fourth</u> answer choice</p>
<pre> when AnswerButton1.Click do if select list item list select list item list get global answerList index get global answerListIndex = 1 then set AnswerButton1.BackgroundColor to green set global score to get global score + 1 set ScoreLabel.Text to join "Score: " get global score </pre>	<p>Get <u>first</u> answer choice</p>

Repeat this for the remaining three (3) answer programming blocks.



<pre> when NextButton Click do set global questionListIndex to (get global questionListIndex + 1) set global answerListIndex to (get global answerListIndex + 1) set QuestionLabel Text to (select list item list (get global questionList) index (get global questionListIndex)) set AnswerButton1 Text to (select list item list (select list item list (get global answerList) index (get global answerListIndex)) index 1) set AnswerButton2 Text to (select list item list (select list item list (get global answerList) index (get global answerListIndex)) index 2) set AnswerButton3 Text to (select list item list (select list item list (get global answerList) index (get global answerListIndex)) index 3) set AnswerButton4 Text to (select list item list (select list item list (get global answerList) index (get global answerListIndex)) index 4) set AnswerButton1 BackgroundColor to grey set AnswerButton2 BackgroundColor to grey set AnswerButton3 BackgroundColor to grey set AnswerButton4 BackgroundColor to grey </pre> <p style="text-align: right;">} Point to NEXT question</p>	
<p>Test your application.</p>	
 <p>Does your application function properly?</p>	



Worksheet 5

Taking it further...

Activity: Making suggestions to improve the quiz.

Time: 10 Minutes

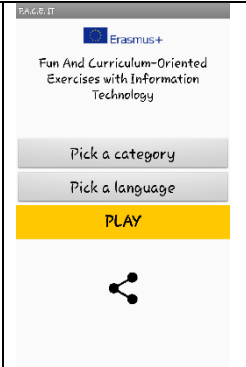


*Follow your facilitator and complete the tasks below (put a mark if completed).
Don't hesitate to ask if there is something you are not sure of.*

TASK	DONE?
<p>“Further ideas”</p> <ul style="list-style-type: none">• What are some drawbacks of this app?• How could you improve them?• What functionality would you add/remove?• Would you change the interface design?• ...	
<p>Discuss your ideas/suggestion in the class.</p> <p>Save a new copy of your application and implement the changes.</p>	



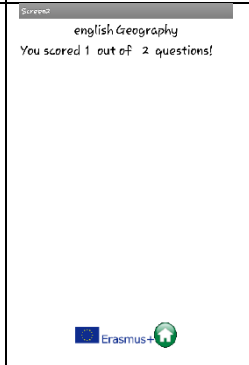


Explaining the concept

You will create the very first version (v1) of the “F.A.C.E. IT” Quiz app(lication).
The user will choose a *category* and a *language* to play the Quiz.
He/She will proceed through the questions by choosing an answer (correct or wrong).
At the end the final score will be shown.

<ul style="list-style-type: none"> • This will be the initial user interface. • At the top there is an “Erasmus+” logo with the title of the Quiz application. • Below, a list picker for categories and one for languages. • ...followed by a “Play” button to start the game. • At the bottom, a Share Application icon. 	
<ul style="list-style-type: none"> • Since we have to keep it simple the Quiz app will be limited to 1 category (Geography) and 1 language (English). • Should someone pick a different set (category, language) a message should inform him/her about the only option available. 	<p>Demo version. Only works for category “Geography” and language “english”. Please try again. Click Home button. Thank you.</p>
<ul style="list-style-type: none"> • The user has to choose the only working combination of “Geography” and “english”. After choosing a language an icon with the country flag will appear next to it. • Pressing the “PLAY” button will start the game. 	
<ul style="list-style-type: none"> • The second screen will appear. • At the top it will inform the user about the current <i>language</i> and <i>category</i>. • The score will be set initially to zero (0). • The question will be in blue color. • The answers will be buttons to click on. • At the bottom there will be an “Erasmus+” logo and a “Home” button to start over. 	



<ul style="list-style-type: none"> • After clicking on an answer, <ul style="list-style-type: none"> ○ ...the correct answer will be highlighted in green, ○ ...the score will be updated with +1 for correct answering or nothing and ○ ...the “NEXT” button will appear at the bottom. 	
<ul style="list-style-type: none"> • We will provide <u>only</u> 2 questions • ...to keep it as simple as possible. 	
<ul style="list-style-type: none"> • At the end the final score will be displayed. • The user will have to click the “Home” button to start over again. 	

Let's start!



Worksheet 5 (Part A)

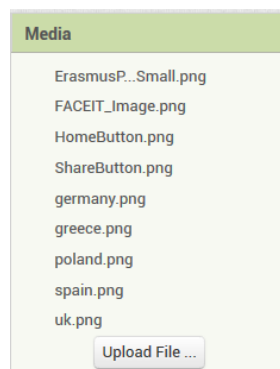
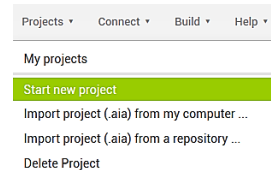
Creating the user interface of the Quiz app

Activity: You will create a simple **user interface** for the Quiz app which will allow the user to choose the **category** of the questions and the **language**.

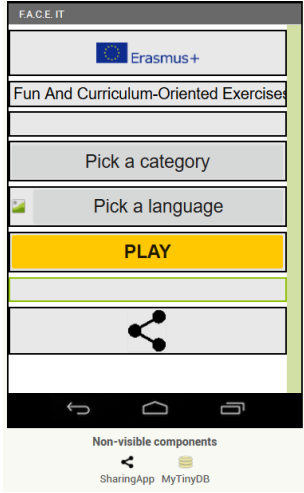
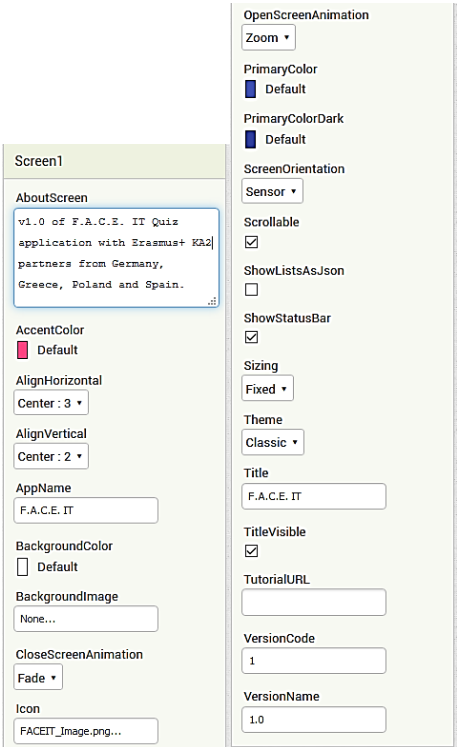
Time: 30 Minutes

Follow your facilitator and complete the tasks below (put a mark if completed).
Don't hesitate to ask if there is something you are not sure of.

TASK	DONE?
<p>Connecting to AppInventor</p> <ul style="list-style-type: none"> Open a browser, e.g. Firefox, Chrome, MS Edge, Safari. In the address bar of your browser type http://ai2.appinventor.mit.edu . Sign in with your Google Account. After this you're presented with the AppInventor environment. 	
<p>Starting a new project</p> <ul style="list-style-type: none"> From the menu click "Projects" and then choose "Start New Project". Enter the project name, FACE_IT_v1, and then click OK. 	
<p>Uploading images to AppInventor</p> <ul style="list-style-type: none"> You will need 9 pictures for this version of the app. The images are in a zipped file format located in the Twinspace Material folder. <ul style="list-style-type: none"> Sign in to your Twinspace account. Follow the path "Material → Files → DE-Workshops" and download the file "QuizAppImages.zip" or directly under http://twinspace.etwinning.net/files/collabspace/1/21/721/44721/files/b3f920b7.zip After downloading unzip the files at your Desktop. Upload each one of the files to your AppInventor project. NOTE: You cannot upload all files at once. After completing your Media section should look like this 	

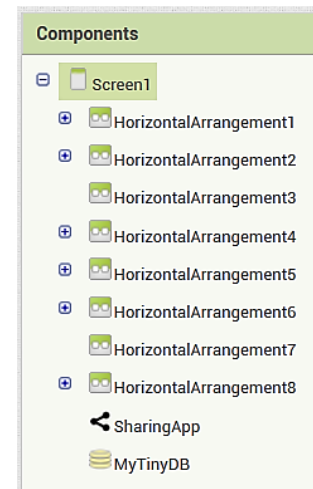




TASK	Screenshot
<ul style="list-style-type: none"> This is a screenshot of the user interface of your Screen1 which you will design. 	
<ul style="list-style-type: none"> Edit the Screen1 properties as shown. <p>NOTE: The screenshots are splitted in two halves.</p> <p>AboutScreen: “v1.0 of F.A.C.E. IT Quiz application with Erasmus+ KA2 partners from Germany, Greece, Poland and Spain.”</p>	



- The interface consists of
8 HorizontalArrangements,
1 SharingApp and
1 TinyDB.



- Check and edit the properties of all **8 HorizontalArrangements** as shown.

<p>HorizontalArrangement1</p> <p>AlignHorizontal Center : 3 ▾</p> <p>AlignVertical Center : 2 ▾</p> <p>BackgroundColor Default</p> <p>Height 50 pixels...</p> <p>Width Fill parent...</p> <p>Image None...</p> <p>Visible <input checked="" type="checkbox"/></p>	<p>HorizontalArrangement2</p> <p>AlignHorizontal Left : 1 ▾</p> <p>AlignVertical Top : 1 ▾</p> <p>BackgroundColor Default</p> <p>Height Automatic...</p> <p>Width Fill parent...</p> <p>Image None...</p> <p>Visible <input checked="" type="checkbox"/></p>	<p>HorizontalArrangement3</p> <p>AlignHorizontal Center : 3 ▾</p> <p>AlignVertical Center : 2 ▾</p> <p>BackgroundColor Default</p> <p>Height 25 pixels...</p> <p>Width Fill parent...</p> <p>Image None...</p> <p>Visible <input checked="" type="checkbox"/></p>	<p>HorizontalArrangement4</p> <p>AlignHorizontal Left : 1 ▾</p> <p>AlignVertical Top : 1 ▾</p> <p>BackgroundColor Default</p> <p>Height Automatic...</p> <p>Width Fill parent...</p> <p>Image None...</p> <p>Visible <input checked="" type="checkbox"/></p>
<p>HorizontalArrangement5</p> <p>AlignHorizontal Left : 1 ▾</p> <p>AlignVertical Top : 1 ▾</p> <p>BackgroundColor Default</p> <p>Height Automatic...</p> <p>Width Fill parent...</p> <p>Image None...</p> <p>Visible <input checked="" type="checkbox"/></p>	<p>HorizontalArrangement6</p> <p>AlignHorizontal Left : 1 ▾</p> <p>AlignVertical Top : 1 ▾</p> <p>BackgroundColor Default</p> <p>Height Automatic...</p> <p>Width Fill parent...</p> <p>Image None...</p> <p>Visible <input checked="" type="checkbox"/></p>	<p>HorizontalArrangement7</p> <p>AlignHorizontal Center : 3 ▾</p> <p>AlignVertical Center : 2 ▾</p> <p>BackgroundColor Default</p> <p>Height 25 pixels...</p> <p>Width Fill parent...</p> <p>Image None...</p> <p>Visible <input checked="" type="checkbox"/></p>	<p>HorizontalArrangement8</p> <p>AlignHorizontal Center : 3 ▾</p> <p>AlignVertical Center : 2 ▾</p> <p>BackgroundColor Default</p> <p>Height Fill parent...</p> <p>Width Fill parent...</p> <p>Image None...</p> <p>Visible <input checked="" type="checkbox"/></p>

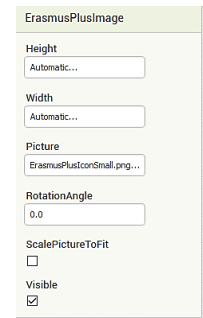


HorizontalArrangement1

- From the Palette “User Interface”, drag the **Image** component onto your **HorizontalArrangement1**.

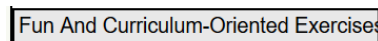


- Rename it to **ErasmusPlusImage** and
- ...edit the properties as shown:



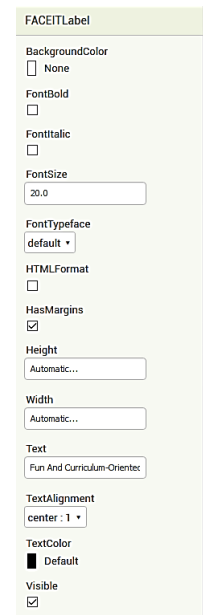
HorizontalArrangement2

- From the Palette “User Interface”, drag the **Label** component onto your **HorizontalArrangement2**.



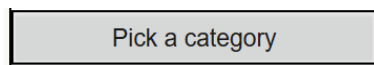
- Rename it to **FACEITLabel** and
- ...edit the properties as shown:

Text property is
“Fun And Curriculum-Oriented Exercises with Information Technology”

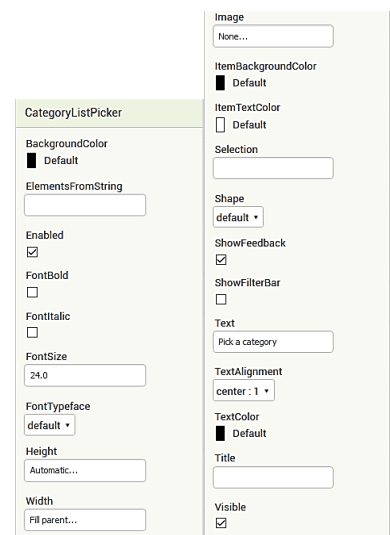


HorizontalArrangement4

- From the Palette “User Interface”, drag the **ListPicker** component onto your **HorizontalArrangement4**.



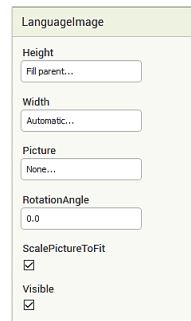
- Rename it to **CategoryListPicker** and
- ...edit the properties as shown:



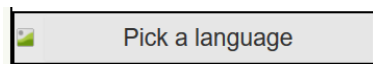


HorizontalArrangement5

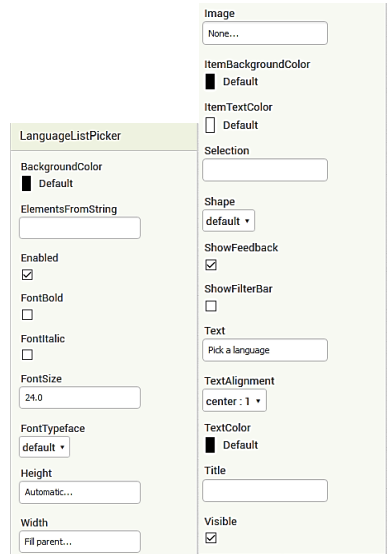
- From the Palette “User Interface”, drag the **Image** component onto your **HorizontalArrangement5**.
- Rename the **Image** to **LanguageImage** and
- ...edit the properties as shown.



- From the Palette “User Interface”, drag the **ListPicker** component onto your **HorizontalArrangement5**.



- Rename the **ListPicker** to **LanguageListPicker** and
- ...edit the properties as shown.

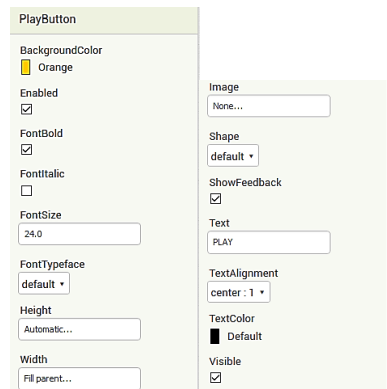


HorizontalArrangement6

- From the Palette “User Interface”, drag the **Button** component onto your **HorizontalArrangement6**.

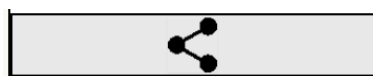


- Rename the **Button** to **PlayButton** and
- ...edit the properties as shown.

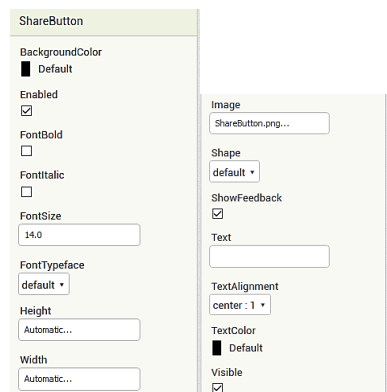


HorizontalArrangement8

- From the Palette “User Interface”, drag the **Button** component onto your **HorizontalArrangement8**.



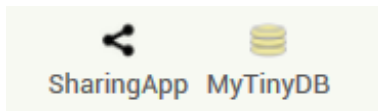
- Rename the **Button** to **ShareButton** and
- ...edit the properties as shown.





Non-visible components

- From the Palette “Social” choose the component **SharingApp** and drag it to the interface.
- From the Palette “Storage” choose the component **TinyDB** and drag it to the interface. Rename to **MyTinyDB**.

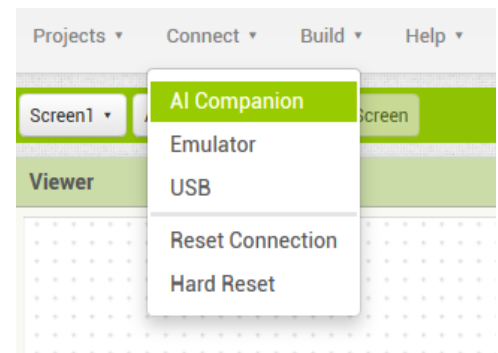


Testing the app

TIP: Before testing the app you should download the “MIT AI2 Companion” from Google Play.

Connect your smartphone or tablet from the Connect menu at the top of your screen.

1. Connect your smartphone on the Wi-Fi network.
2. Choose AI Companion from the Connect menu
3. A unique app code appears in both QR and text form.
4. On your phone start the MIT AI2 Companion app.
5. Either type the app code and click “Connect with Code” or click “Scan the QR Code” and point your phone at the QR code on your computer screen.



Try your app by pressing the buttons.



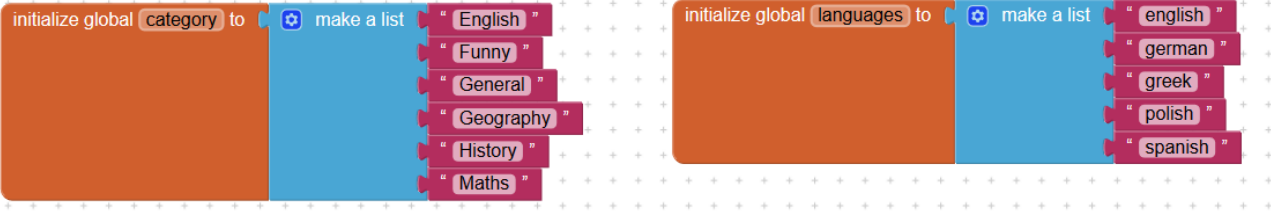
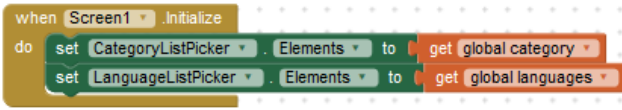
Worksheet 5 (Part B)

Programming the user interface (Screen1)

Activity: You will program the components of the user interface in order to add interactivity to your application.

Time: 20 Minutes

Follow your facilitator and complete the tasks below (put a mark if completed).
 Don't hesitate to ask if there is something you are not sure of.

TASK	Screenshot
<p>Blocks editor</p> <p>Switch to the BLOCKS editor (upper right corner).</p> <p>Initializing</p> <p>We initialize two variables as lists</p> <ul style="list-style-type: none"> • category to accommodate a list with the choices “English”, “Funny”, “General”, “Geography”, “History”, and “Maths” and • language to accommodate a list with the choices “english”, “german”, “greek”, “polish” and “spanish”.  <p>...and when Screen1 is loaded both lists get their elements from their respective variables.</p> 	



Picking elements from lists

- After the user picks a category we change the Text property of **CategoryListPicker** to reflect the user selection.
- Likewise, after picking a language we change the Text property of **LanguageListPicker** to reflect the user selection. Also, we set a picture of the respective country flag by changing the Picture property of **LanguageImage** as shown.

```

when CategoryListPicker.AfterPicking
do
  set CategoryListPicker.Text to CategoryListPicker.Selection

when LanguageListPicker.AfterPicking
do
  set LanguageListPicker.Text to LanguageListPicker.Selection
  if LanguageListPicker.Selection = "english"
  then set LanguageImage.Picture to "uk.png"
  if LanguageListPicker.Selection = "german"
  then set LanguageImage.Picture to "germany.png"
  if LanguageListPicker.Selection = "greek"
  then set LanguageImage.Picture to "greece.png"
  if LanguageListPicker.Selection = "polish"
  then set LanguageImage.Picture to "poland.png"
  if LanguageListPicker.Selection = "spanish"
  then set LanguageImage.Picture to "spain.png"
  
```

Playing

- The user has made a choice of category and language.
- Those values have to be passed on to a new screen to load the appropriate questions.
- For this purpose, we take advantage of the Database component **TinyDB** to share those values between two screens.

```

when PlayButton.Click
do
  call MyTinyDB.StoreValue
  tag "categoryID"
  valueToStore CategoryListPicker.Selection
  call MyTinyDB.StoreValue
  tag "languageID"
  valueToStore LanguageListPicker.Selection
  open another screen screenName "Screen2"
  
```

Sharing the application

- Last, we want to offer the opportunity to share the application through social media.
- At this point, we haven't yet any functional link to Google Play.
- That's why we'll show only a message with the text *"A link will be provided in future to download from Google Play."*

```

when ShareButton.Click
do
  call SharingApp.ShareMessage
  message "A link will be provided in future to download fr..."
  
```

Testing the app

- Up until this point you've created the first screen (**Screen1**) and added functionality to its components. Before proceeding with the next task you should test your app.



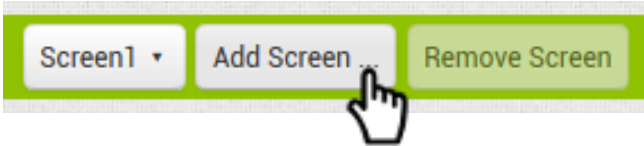
Worksheet 5 (Part C)

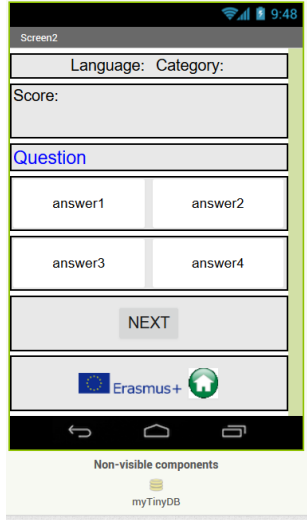
Creating the Question interface

Activity: You will create a second screen for the interface of the questions.

Time: 30 Minutes

Follow your facilitator and complete the tasks below (put a mark if completed).
Don't hesitate to ask if there is something you are not sure of.

TASK	DONE?
<p>Adding a Screen</p> <p>Click on the Add Screen button and add a new screen with the name Screen2.</p> 	

TASK	Screenshot
<ul style="list-style-type: none"> This is a screenshot of the interface of your Screen2 which you will design. 	



- Edit the **Screen2** properties as shown.

Screen2

AboutScreen

AlignHorizontal
Center : 3 ▾

AlignVertical
Center : 2 ▾

BackgroundColor
 Default

BackgroundImage

CloseScreenAnimation
Zoom ▾

OpenScreenAnimation
Zoom ▾

ScreenOrientation
Sensor ▾

Scrollable

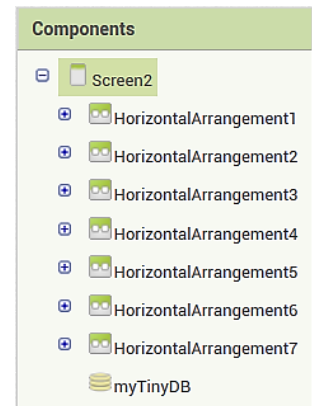
ShowStatusBar

Title

TitleVisible



- The interface consists of **7 HorizontalArrangements** and **1 TinyDB**.



- Check and edit the properties of all **7 HorizontalArrangements** as shown.

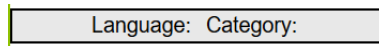
HorizontalArrangement1	HorizontalArrangement2	HorizontalArrangement3	HorizontalArrangement4
AlignHorizontal Center : 3 ▾	AlignHorizontal Left : 1 ▾	AlignHorizontal Left : 1 ▾	AlignHorizontal Left : 1 ▾
AlignVertical Center : 2 ▾	AlignVertical Top : 1 ▾	AlignVertical Top : 1 ▾	AlignVertical Top : 1 ▾
BackgroundColor ■ Default	BackgroundColor ■ Default	BackgroundColor ■ Default	BackgroundColor ■ Default
Height Automatic...	Height Fill parent...	Height Automatic...	Height Fill parent...
Width Fill parent...	Width Fill parent...	Width Fill parent...	Width Fill parent...
Image None...	Image None...	Image None...	Image None...
Visible <input checked="" type="checkbox"/>	Visible <input checked="" type="checkbox"/>	Visible <input checked="" type="checkbox"/>	Visible <input checked="" type="checkbox"/>

HorizontalArrangement5	HorizontalArrangement6	HorizontalArrangement7
AlignHorizontal Left : 1 ▾	AlignHorizontal Center : 3 ▾	AlignHorizontal Center : 3 ▾
AlignVertical Top : 1 ▾	AlignVertical Center : 2 ▾	AlignVertical Center : 2 ▾
BackgroundColor ■ Default	BackgroundColor ■ Default	BackgroundColor ■ Default
Height Fill parent...	Height Fill parent...	Height Fill parent...
Width Fill parent...	Width Fill parent...	Width Fill parent...
Image None...	Image None...	Image None...
Visible <input checked="" type="checkbox"/>	Visible <input checked="" type="checkbox"/>	Visible <input checked="" type="checkbox"/>



HorizontalArrangement1

- From the Palette “User Interface”, drag two **Label** components onto your **HorizontalArrangement1**.



- Rename the first label to **LanguageLabel** and
- ...edit the properties as shown.
- Rename the second label to **CategoryLabel** and
- ...edit the properties as shown.

LanguageLabel	CategoryLabel
BackgroundColor <input type="checkbox"/> None	BackgroundColor <input type="checkbox"/> None
FontBold <input type="checkbox"/>	FontBold <input type="checkbox"/>
FontItalic <input type="checkbox"/>	FontItalic <input type="checkbox"/>
FontSize 20.0	FontSize 20.0
FontTypeface default ▾	FontTypeface default ▾
HTMLFormat <input type="checkbox"/>	HTMLFormat <input type="checkbox"/>
HasMargins <input checked="" type="checkbox"/>	HasMargins <input checked="" type="checkbox"/>
Height Automatic...	Height Automatic...
Width Automatic...	Width Automatic...
Text Language:	Text Category:
TextAlignment left: 0 ▾	TextAlignment left: 0 ▾
TextColor Default	TextColor Default
Visible <input checked="" type="checkbox"/>	Visible <input checked="" type="checkbox"/>

HorizontalArrangement2

- From the Palette “User Interface”, drag a **Label** component onto your **HorizontalArrangement2**.



- Rename the label to **ScoreLabel** and
- ...edit the properties as shown.

ScoreLabel	Properties
BackgroundColor <input type="checkbox"/> None	HasMargins <input checked="" type="checkbox"/>
FontBold <input type="checkbox"/>	Height Automatic...
FontItalic <input type="checkbox"/>	Width Automatic...
FontSize 20.0	Text Score:
FontTypeface default ▾	TextAlignment left: 0 ▾
HTMLFormat <input type="checkbox"/>	TextColor Default
	Visible <input checked="" type="checkbox"/>

HorizontalArrangement3

- From the Palette “User Interface”, drag a **Label** component onto your **HorizontalArrangement3**.



- Rename the label to **QuestionLabel** and
- ...edit the properties as shown.

QuestionLabel	Properties
BackgroundColor <input type="checkbox"/> None	HasMargins <input checked="" type="checkbox"/>
FontBold <input type="checkbox"/>	Height Automatic...
FontItalic <input type="checkbox"/>	Width Automatic...
FontSize 24.0	Text Question
FontTypeface default ▾	TextAlignment left: 0 ▾
HTMLFormat <input type="checkbox"/>	TextColor Blue
	Visible <input checked="" type="checkbox"/>

HorizontalArrangement4

- From the Palette “User Interface”, drag two **Button** components onto your **HorizontalArrangement4**.



- Rename the first button to **AnswerButton1** and
- ...the second button to **AnswerButton2**.
- Edit the properties as shown.

AnswerButton1	Properties
BackgroundColor <input type="checkbox"/> White	Image None...
Enabled <input checked="" type="checkbox"/>	Shape default ▾
FontBold <input type="checkbox"/>	ShowFeedback <input checked="" type="checkbox"/>
FontItalic <input type="checkbox"/>	Text answer1
FontSize 18.0	TextAlignment center: 1 ▾
FontTypeface default ▾	TextColor Default
Height Fill parent...	Visible <input checked="" type="checkbox"/>
Width Fill parent...	

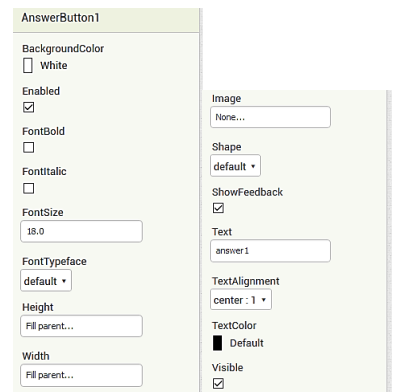


HorizontalArrangement5

- From the Palette "User Interface", drag two **Button** components onto your **HorizontalArrangement5**.



- Rename the first button to **AnswerButton3** and
- ...the second button to **AnswerButton4**.
- Edit the properties as shown.

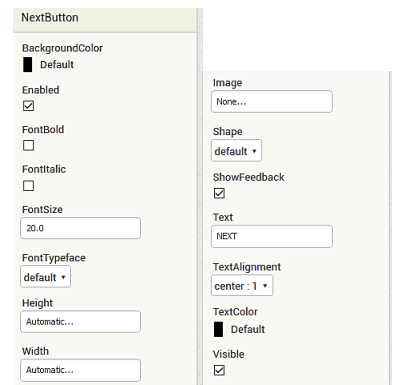


HorizontalArrangement6

- From the Palette "User Interface", drag a **Button** component onto your **HorizontalArrangement6**.

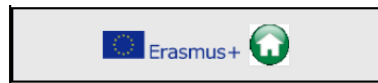


- Rename the button to **NextButton** and
- ...edit the properties as shown.

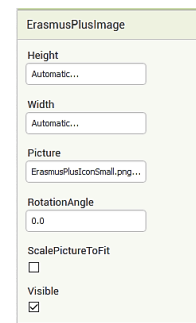


HorizontalArrangement7

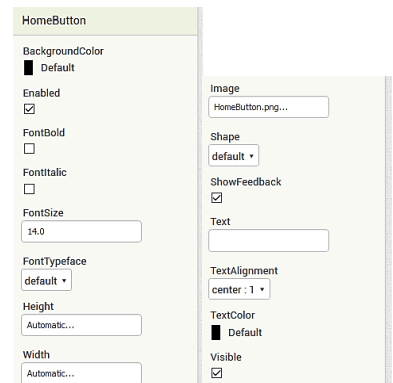
- From the Palette "User Interface", drag an **Image** and a **Button** component onto your **HorizontalArrangement7**.



- Rename the image to **ErasmusPlusImage** and
- ...edit the properties as shown.

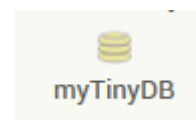


- Rename the button to **HomeButton** and
- ...edit the properties as shown.



Non-visible components

- From the Palette "Storage" choose the component **TinyDB** and drag it to the interface. Rename to **MyTinyDB**.





Testing the app

- Up until this point you've created successfully the design of the second screen (**Screen2**). Before proceeding with the next task you should test your app.




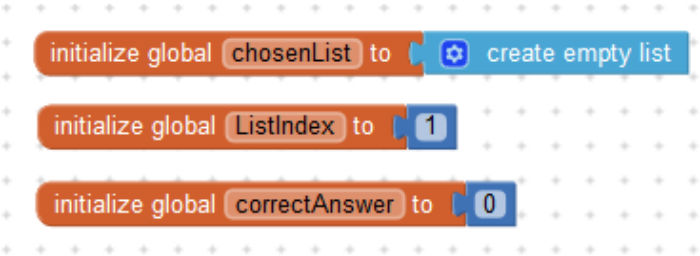

Worksheet 5 (Part D)

Programming the Question interface (Screen2)

Activity: You will program the components of the Question interface in order to pick the right list of user's choice of category and language. Then to proceed through the questions, to evaluate the user's answer, to update the score and finally to show a message of the final score.

Time: 45 Minutes

Follow your facilitator and complete the tasks below (put a mark if completed).
 Don't hesitate to ask if there is something you are not sure of.

TASK	Screenshot
<p>Blocks editor</p> <p>Switch to the BLOCKS editor (upper right corner).</p> <p>Initializing</p> <p>We will use a set of variables which have to be initialized with a proper value before using.</p> <ul style="list-style-type: none"> Variables category and language will hold the shared values of the user's choice of category and language from Screen1. Initially they will hold an empty text string.  <ul style="list-style-type: none"> The variable chosenList will be used to load the appropriate Question-Answers of the user's choice. Initially it will hold an empty list. The variable ListIndex will point to the elements in the list and the variable correctAnswer will hold the correct answer for every question.  <ul style="list-style-type: none"> The variable score will hold the current score of the user. Initially it is set to 0. The variable userAnswer will be used to compare user's choice with the correctAnswer. Both of them are initially set to 0. 	



Initializing the Question interface

- The Question interface (**Screen2**) gets the values of the **category** and the **language** from the shared Database **myTinyDB**.
- Also, sets the labels of **CategoryLabel** and **LanguageLabel** respectively to inform the user during the game.

IMPORTANT: In order to keep it simple we will create a quiz with only 2 questions and for 1 category in 1 language. For this purpose we have chosen *Geography* and *english*.

- If the criteria of “Geography” and “english” are met, the **chosenList** is loaded with the appropriate list from a csv (comma separated value) table.
- You have to fill the data of the csv table in the text field as follows
*What is the capital of Germany?, Hamburg, Munich, Berlin, Hannover, 3\n
 Which is the largest river in Germany?, Elbe, Danube, Rhine, Amazon, 2\n*

Pay attention to the punctuation, such as the comma (,) and the question mark (?). Each question ends with a new line character “\n”.

IMPORTANT: At this point the procedure **initializeQA** must be called. Since you haven’t created it yet you will add this programming block later on once you’ve built it.

- If the criteria are not met, a message will inform the user to choose “Geography” and “english”. The message will be
Demo version.\n Only works for category "Geography" and language "english".\nPlease try again. Click Home button.\nThank you.

```

when Screen2.Initialize
do
  set global category to call myTinyDB.GetValue
                        tag "categoryID"
                        valueIfTagNotThere ""
  set global language to call myTinyDB.GetValue
                        tag "languageID"
                        valueIfTagNotThere ""
  set CategoryLabel.Text to get global category
  set LanguageLabel.Text to get global language
  if
    get global category = "Geography" and get global language = "english"
  then
    set global chosenList to list from csv table text "What is the capital of Germany?, Hamburg, Munich..."
    call initializeQA
  else
    set QuestionLabel.Text to "Demo version.\n Only works for category "Geograp..."
    set NextButton.Visible to false
  
```



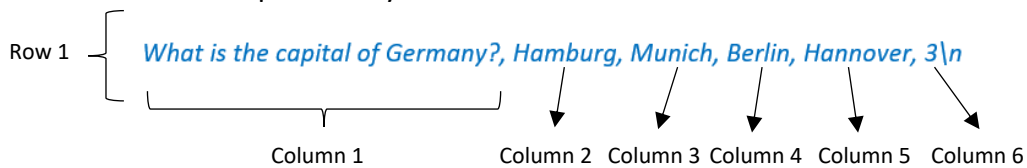

Procedure initializeQA

- This procedure will be called each time to set the labels of **Screen2** with the content of the next question, the possible answers, and the score respectively.
- In order to avoid an “empty” response from the user the **NextButton** is hidden (set to false). It will only be shown once the user makes a choice, i.e. clicks on an **AnswerButton**.

```

to initializeQA
do
  set NextButton . Visible to false
  set ScoreLabel . Text to join " Score: "
  get global score
  set QuestionLabel . Text to select list item list
  select list item list index 1
  get global chosenList
  get global ListIndex
  set AnswerButton1 . Text to select list item list
  select list item list index 2
  get global chosenList
  get global ListIndex
  set AnswerButton2 . Text to select list item list
  select list item list index 3
  get global chosenList
  get global ListIndex
  set AnswerButton3 . Text to select list item list
  select list item list index 4
  get global chosenList
  get global ListIndex
  set AnswerButton4 . Text to select list item list
  select list item list index 5
  get global chosenList
  get global ListIndex
  set global correctAnswer to select list item list
  select list item list index 6
  get global chosenList
  get global ListIndex
  
```

NOTE: We use a table for storing our questions, answers and an index to the correct answer. The table consists of rows and columns separated by commas. Each row ends with a new line character “\n”.



Column1 holds the question. Columns 2 to 5 the four possible answers. Column 6 the number for the correct answer which for the above example is 3, i.e. “Berlin”.

```

select list item list index 1
select list item list index
get global chosenList
get global ListIndex
  
```

The above programming blocks translate to “Choose from the current list (variable chosenList) and the row (variable ListIndex) the value of the first column (number 1)”, i.e. the question.



Adding functionality to the AnswerButtonX

- Once the user clicks on an **AnswerButton** his/her choice is marked (**userAnswer** set to 1, 2, 3 or 4),
- ...all the other buttons are deactivated (procedure **deactivateButtons**) to prevent the user from trying many times, and
- ...the score is updated accordingly (procedure **updateScore**) with +1 if the user chose correctly.

IMPORTANT: At this point the procedures **deactivateButtons** and **updateScore** must be called. Since you haven't created them yet you will add those programming block later on once you've built them.

```

when AnswerButton1 .Click
do
  set global userAnswer to 1
  call deactivateButtons
  call updateScore

when AnswerButton2 .Click
do
  set global userAnswer to 2
  call deactivateButtons
  call updateScore

when AnswerButton3 .Click
do
  set global userAnswer to 3
  call deactivateButtons
  call updateScore

when AnswerButton4 .Click
do
  set global userAnswer to 4
  call deactivateButtons
  call updateScore
  
```

Procedure deactivateButtons

To deactivate the buttons we set the **Enabled** property to **false**.

```

to deactivateButtons
do
  set AnswerButton1 . Enabled to false
  set AnswerButton2 . Enabled to false
  set AnswerButton3 . Enabled to false
  set AnswerButton4 . Enabled to false
  
```



Procedure updateScore

- The user's answer is checked against the correct answer. If it is the same (true) the score is updated +1, otherwise not.
- The **BackgroundColor** of the correct answer is highlighted in green (procedure **highlightCorrectAnswer**).
- The value of the new score is stored in our Database **TinyDB**.
- The **NextButton** is shown, i.e. set **Visible** to **true**.

```

to updateScore
do
  if (get global userAnswer = get global correctAnswer)
  then
    set global score to (get global score + 1)
  call highlightCorrectAnswer
  set ScoreLabel . Text to (join " Score: " (get global score))
  call myTinyDB . StoreValue
    tag "userScore"
    valueToStore (get global score)
  set NextButton . Visible to true
  
```

Procedure highlightCorrectAnswer

The **BackgroundColor** of the correct **AnswerButton** is set to green.

```

to highlightCorrectAnswer
do
  if (get global correctAnswer = 1)
  then
    set AnswerButton1 . BackgroundColor to green
  else if (get global correctAnswer = 2)
  then
    set AnswerButton2 . BackgroundColor to green
  else if (get global correctAnswer = 3)
  then
    set AnswerButton3 . BackgroundColor to green
  else
    set AnswerButton4 . BackgroundColor to green
  
```



Moving to next question

- When the user clicks on the **NextButton** the **ListIndex** points to the next question,
- ...and if it hasn't reached the end of the question list,
- ...activates all the buttons (procedure **activateButtons**),
- ...changes the **BackgroundColor** of all the buttons (procedure **whiteButtons**),
- ...and sets the text of the new question with its respective choices of answers.
- Otherwise, the final score is shown (procedure **showFinalScore**).

```

when NextButton .Click
do
  set global ListIndex to (get global ListIndex + 1)
  if (get global ListIndex ≤ length of list list) (get global chosenList)
  then
    call activateButtons
    call whiteButtons
    call initializeQA
  else
    call showFinalScore
  
```

```

to activateButtons
do
  set AnswerButton1 . Enabled to true
  set AnswerButton2 . Enabled to true
  set AnswerButton3 . Enabled to true
  set AnswerButton4 . Enabled to true

```

```

to whiteButtons
do
  set AnswerButton1 . BackgroundColor to white
  set AnswerButton2 . BackgroundColor to white
  set AnswerButton3 . BackgroundColor to white
  set AnswerButton4 . BackgroundColor to white

```



Show final score

- After completing the quiz challenge for a certain category the user is shown the final score.
- For this purpose, we hide some of the components such as the **QuestionLabel**, the **AnswerButtonsX** and the **NextButton** as shown below (procedure **hideComponents**).
- The message is “You scored X out of Y questions!” where X is the user’s score and Y the total number of questions.

```

to showFinalScore
do
  call hideComponents
  set ScoreLabel . Text to join
    " You scored "
    get global score
    " out of "
    length of list list
    get global chosenList
    " questions!"
  
```

```

to hideComponents
do
  set QuestionLabel . Visible to false
  set AnswerButton1 . Visible to false
  set AnswerButton2 . Visible to false
  set AnswerButton3 . Visible to false
  set AnswerButton4 . Visible to false
  set NextButton . Visible to false
  
```

Testing the app

- At this point you’ve created two screens and added proper functionality to its componets. It’s time to test your app check its behaviour. Make adjustments if necessary.



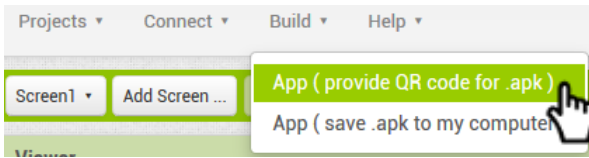
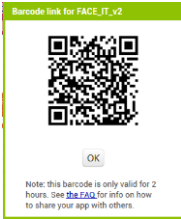
Worksheet 5 (Part E)

Downloading and installing the Application

Activity: You will download and install your recently-created Quiz App to your smartphone.

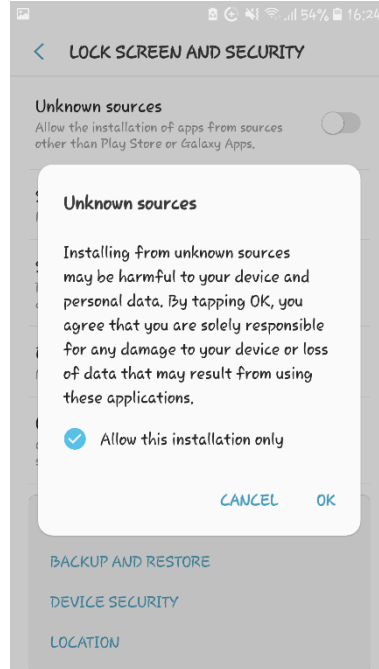
Time: 10 Minutes

*Follow your facilitator and complete the tasks below (put a mark if completed).
Don't hesitate to ask if there is something you are not sure of.*

TASK	DONE?
<p>Before downloading</p> <ul style="list-style-type: none"> AppInventor will bundle all your components and programming blocks of your Quiz app into an APK file (Android Package Kit). This package file format is used by the Android operating system for distribution and installation of mobile apps. Just like Windows systems use an .exe file for installing software. There are two options to install the app to your smartphone: ...either through a download link provided by a QR code ...or by saving it first on the computer desktop. We will go for the first option (QR code). Follow the screenshot instructions below <div style="text-align: center; margin: 10px 0;">  </div> <ul style="list-style-type: none"> After selecting the above choice, AppInventor will compile all parts, build the APK file (you will notice the progress bars) and provide the QR Code to download the file like shown below. <div style="text-align: center; margin: 10px 0;">  </div> <p>NOTE: In order to read the above QR code you have to have a “QR Reader”. If you don't have any on your smartphone or tablet download one of your choice from Google Play.</p>	



- Your security settings of your smartphone may prevent you from “installing from unknown sources”. For this time only, allow the installation since it’s your app you’ve built.



- After installation you are ready to try your application.




EXAMPLE for Worksheet 6

Software Documentation

Activity: Students study in detail and document the different parts of the Quiz Application.

Time: 30 Minutes

DESCRIPTION
<div style="display: flex; align-items: flex-start;">  <p>“What is software documentation?”</p> </div> <p>Software documentation is written text or illustration that accompanies computer software or is embedded in the source code. It either explains how it operates or how to use it.</p> <p style="text-align: right;">(Wikipedia)</p> <p>Documentation is often divided into the following categories:</p> <ul style="list-style-type: none"> • Installation: describes how to install a programme or device • Reference: detailed descriptions of particular items presented in alphabetical order. • Tutorial: teaches a user how to use the product. <p style="text-align: right;">(www.webopedia.com)</p>
PREPARATION
<p>Building working groups</p> <ul style="list-style-type: none"> • Students get in 8 intercultural mixed groups. • Each group gets a designated letter from A to H (i.e. 8 groups are formed). • Once you are in your group choose a catchy name, e.g. <i>App Rangers, Lions</i>, etc. • Each group gets a specific task of the Quiz documentation (see next page).
<p>Connecting to AppInventor (<u>Greek students only</u>)</p> <ul style="list-style-type: none"> • Open a browser, e.g. Chrome, Firefox. • In the address bar of your browser type the URL http://ai2.appinventor.mit.edu . • Sign in with your Google Account. • After this you’re presented with the AppInventor environment.
<p>Load the Quiz file</p> <ul style="list-style-type: none"> • A final version of the Quiz file has been sent to the Greek students’ e-mail accounts. • Import this file into your AppInventor account. • You are now ready to study the components.



GROUP X

Group Name: _____

Students: _____

TASK	DONE?
<p>Documenting</p> <ul style="list-style-type: none">• Read your group assignment carefully.• Each group gets a small part of the entire Quiz application in form of screenshots.• In order to understand your assignment you may load the app on your smartphone, experiment with it, and browse through the different screens in Blocks or Designer View.• Write your comments around the screenshot as shown in the example.• After completion you are invited to present your results in the plenary session.	



HelpScreen (Blocks View)

When the Home button is clicked it returns to the Home Screen (Screen 1)

```
when HomeButton .Click  
do open another screen screenName "Screen1"
```

These are buttons to the webpages of each partner school.

Activity Starter is used to realize the opening of the web browser of the smartphone or tablet.

```
when GermanSchoolButton .Click  
do set ActivityStarter1 .DataUri to "https://www.wirtschaftsschule-dinkelsbuehl.de/"  
call ActivityStarter1 .StartActivity
```

```
when GreekSchoolButton .Click  
do set ActivityStarter1 .DataUri to "https://www.facebook.com/epal.lechaina/"  
call ActivityStarter1 .StartActivity
```

```
when PolishSchoolButton .Click  
do set ActivityStarter1 .DataUri to "http://zsokluczbork.pl/"  
call ActivityStarter1 .StartActivity
```

```
when SpanishSchoolButton .Click  
do set ActivityStarter1 .DataUri to "http://iescolonial.es/"  
call ActivityStarter1 .StartActivity
```



Worksheet 6

Software Documentation

Activity: Students study in detail and document the different parts of the Quiz Application.

Time: 30 Minutes

DESCRIPTION
<div style="display: flex; align-items: flex-start;"> <p>“What is software documentation?”</p> <p>Software documentation is written text or illustration that accompanies computer software or is embedded in the source code. It either explains how it operates or how to use it.</p> <p style="text-align: right;">(Wikipedia)</p> <p>Documentation is often divided into the following categories:</p> <ul style="list-style-type: none"> Installation: describes how to install a programme or device Reference: detailed descriptions of particular items presented in alphabetical order. Tutorial: teaches a user how to use the product. <p style="text-align: right;">(www.webopedia.com)</p> </div>
PREPARATION
<p>Building working groups</p> <ul style="list-style-type: none"> Students get in 8 intercultural mixed groups. Each group gets a designated letter from A to H (i.e. 8 groups are formed). Once you are in your group choose a catchy name, e.g. <i>App Rangers, Lions</i>, etc. Each group gets a specific task of the Quiz documentation (see next page).
<p>Connecting to AppInventor (Greek students only)</p> <ul style="list-style-type: none"> Open a browser, e.g. Chrome, Firefox. In the address bar of your browser type the URL http://ai2.appinventor.mit.edu . Sign in with your Google Account. After this you’re presented with the AppInventor environment.
<p>Load the Quiz file</p> <ul style="list-style-type: none"> A final version of the Quiz file has been sent to the Greek students’ e-mail accounts. Import this file into your AppInventor account. You are now ready to study the components.



GROUP A

Group Name: _____

Students: _____

TASK	DONE?
<p>Documenting</p> <ul style="list-style-type: none">• Read your group assignment carefully.• Each group gets a small part of the entire Quiz application in form of screenshots.• In order to understand your assignment you may load the app on your smartphone, experiment with it, and browse through the different screens in Blocks or Designer View.• Write your comments around the screenshot as shown in the example.• After completion you are invited to present your results in the plenary session.	



Screen 1 (Designer View)

F.A.C.E. IT

Co-funded by the
Erasmus+ Programme of
the European Union  Erasmus+

Fun And Curriculum-Oriented Exercises

Pick a category

 Pick a language

FACE IT!



Worksheet 6

Software Documentation

Activity: Students study in detail and document the different parts of the Quiz Application.

Time: 30 Minutes

DESCRIPTION
<div style="display: flex; align-items: flex-start;"> <p>“What is software documentation?”</p> </div> <p>Software documentation is written text or illustration that accompanies computer software or is embedded in the source code. It either explains how it operates or how to use it.</p> <p style="text-align: right;">(Wikipedia)</p> <p>Documentation is often divided into the following categories:</p> <ul style="list-style-type: none"> • Installation: describes how to install a programme or device • Reference: detailed descriptions of particular items presented in alphabetical order. • Tutorial: teaches a user how to use the product. <p style="text-align: right;">(www.webopedia.com)</p>
PREPARATION
<p>Building working groups</p> <ul style="list-style-type: none"> • Students get in 8 intercultural mixed groups. • Each group gets a designated letter from A to H (i.e. 8 groups are formed). • Once you are in your group choose a catchy name, e.g. <i>App Rangers, Lions</i>, etc. • Each group gets a specific task of the Quiz documentation (see next page).
<p>Connecting to AppInventor (Greek students only)</p> <ul style="list-style-type: none"> • Open a browser, e.g. Chrome, Firefox. • In the address bar of your browser type the URL http://ai2.appinventor.mit.edu . • Sign in with your Google Account. • After this you’re presented with the AppInventor environment.
<p>Load the Quiz file</p> <ul style="list-style-type: none"> • A final version of the Quiz file has already been sent to the Greek students’ e-mail account as an attachment. • Import this file into your AppInventor account. • You are now ready to study the components.



GROUP B

Group Name: _____

Students: _____

TASK	DONE?
<p>Documenting</p> <ul style="list-style-type: none">• Read your group assignment carefully.• Each group gets a small part of the entire Quiz application in form of screenshots.• In order to understand your assignment you may load the app on your smartphone, experiment with it, and browse through the different screens in Blocks or Designer View.• Write your comments around the screenshot as shown in the example.• After completion you are invited to present your results in the plenary session.	



Screen 1 (Blocks View)

```
initialize global category to [make a list] ["English "] ["General "] ["Geography "] ["History "] ["Maths "]
```

```
initialize global languages to [make a list] ["english "] ["german "] ["greek "] ["polish "] ["spanish "]
```

```
when Screen1.Initialize do [set CategoryListPicker.Elements to get global category] [set LanguageListPicker.Elements to get global languages]
```






Screen 2 (Blocks View)

```
when HomeButton .Click
do open another screen screenName "Screen1"

when HelpButton .Click
do open another screen screenName "HelpScreen"
```




Screen 2 (Designer View)

FACE IT!	
 	
Language: Category:	
Score:	
#. Question	
answer1	answer2
answer3	answer4
NEXT	







HelpScreen (Designer View)

HelpScreen



FACE IT or *Fun And Curriculum-Oriented Exercises with Information Technology* is a Quiz application for Android smartphones and tablets implementing over 220 questions from 5 categories (English, General, History, Geography, Maths) and translated into 5 languages (English, German, Greek, Polish and Spanish). This application is a common effort of an Erasmus+ KA2 project during 2017-2019 involving the following partner schools:

1. Staatliche Wirtschaftsschule Dinkelsbühl
(Germany)



Worksheet 6

Software Documentation

Activity: Students study in detail and document the different parts of the Quiz Application.

Time: 30 Minutes

DESCRIPTION
<div style="display: flex; align-items: flex-start;"> <p>“What is software documentation?”</p> <p>Software documentation is written text or illustration that accompanies computer software or is embedded in the source code. It either explains how it operates or how to use it.</p> <p style="text-align: right;">(Wikipedia)</p> <p>Documentation is often divided into the following categories:</p> <ul style="list-style-type: none"> Installation: describes how to install a programme or device Reference: detailed descriptions of particular items presented in alphabetical order. Tutorial: teaches a user how to use the product. <p style="text-align: right;">(www.webopedia.com)</p> </div>
PREPARATION
<p>Building working groups</p> <ul style="list-style-type: none"> Students get in 8 intercultural mixed groups. Each group gets a designated letter from A to H (i.e. 8 groups are formed). Once you are in your group choose a catchy name, e.g. <i>App Rangers, Lions</i>, etc. Each group gets a specific task of the Quiz documentation (see next page).
<p>Connecting to AppInventor (Greek students only)</p> <ul style="list-style-type: none"> Open a browser, e.g. Chrome, Firefox. In the address bar of your browser type the URL http://ai2.appinventor.mit.edu . Sign in with your Google Account. After this you’re presented with the AppInventor environment.
<p>Load the Quiz file</p> <ul style="list-style-type: none"> A final version of the Quiz file has already been sent to the Greek students’ e-mail account as an attachment. Import this file into your AppInventor account. You are now ready to study the components.



GROUP C

Group Name: _____

Students: _____

TASK	DONE?
<p>Documenting</p> <ul style="list-style-type: none">• Read your group assignment carefully.• Each group gets a small part of the entire Quiz application in form of screenshots.• In order to understand your assignment you may load the app on your smartphone, experiment with it, and browse through the different screens in Blocks or Designer View.• Write your comments around the screenshot as shown in the example.• After completion you are invited to present your results in the plenary session.	



Screen 1 (Blocks View)

```
when CategoryListPicker .AfterPicking  
do set CategoryListPicker . Text to CategoryListPicker . Selection
```

```
when LanguageListPicker .AfterPicking  
do set LanguageListPicker . Text to LanguageListPicker . Selection  
  if LanguageListPicker . Selection = " english "  
  then set LanguageImage . Picture to " uk.png "  
  if LanguageListPicker . Selection = " german "  
  then set LanguageImage . Picture to " germany.png "  
  if LanguageListPicker . Selection = " greek "  
  then set LanguageImage . Picture to " greece.png "  
  if LanguageListPicker . Selection = " polish "  
  then set LanguageImage . Picture to " poland.png "  
  if LanguageListPicker . Selection = " spanish "  
  then set LanguageImage . Picture to " spain.png "
```



Worksheet 6

Software Documentation

Activity: Students study in detail and document the different parts of the Quiz Application.

Time: 30 Minutes

DESCRIPTION
<div style="display: flex; align-items: flex-start;"> <p>“What is software documentation?”</p> </div> <p>Software documentation is written text or illustration that accompanies computer software or is embedded in the source code. It either explains how it operates or how to use it.</p> <p style="text-align: right;">(Wikipedia)</p> <p>Documentation is often divided into the following categories:</p> <ul style="list-style-type: none"> • Installation: describes how to install a programme or device • Reference: detailed descriptions of particular items presented in alphabetical order. • Tutorial: teaches a user how to use the product. <p style="text-align: right;">(www.webopedia.com)</p>
PREPARATION
<p>Building working groups</p> <ul style="list-style-type: none"> • Students get in 8 intercultural mixed groups. • Each group gets a designated letter from A to H (i.e. 8 groups are formed). • Once you are in your group choose a catchy name, e.g. <i>App Rangers, Lions</i>, etc. • Each group gets a specific task of the Quiz documentation (see next page).
<p>Connecting to AppInventor (Greek students only)</p> <ul style="list-style-type: none"> • Open a browser, e.g. Chrome, Firefox. • In the address bar of your browser type the URL http://ai2.appinventor.mit.edu . • Sign in with your Google Account. • After this you’re presented with the AppInventor environment.
<p>Load the Quiz file</p> <ul style="list-style-type: none"> • A final version of the Quiz file has already been sent to the Greek students’ e-mail account as an attachment. • Import this file into your AppInventor account. • You are now ready to study the components.



GROUP D

Group Name: _____

Students: _____

TASK	DONE?
<p>Documenting</p> <ul style="list-style-type: none">• Read your group assignment carefully.• Each group gets a small part of the entire Quiz application in form of screenshots.• In order to understand your assignment you may load the app on your smartphone, experiment with it, and browse through the different screens in Blocks or Designer View.• Write your comments around the screenshot as shown in the example.• After completion you are invited to present your results in the plenary session.	



Screen 1 (Blocks View)

```
when HelpButton .Click  
do open another screen screenName " HelpScreen "
```

```
when FacebookButton .Click  
do set WebPageOpenActivityStarter . DataUri to " https://www.facebook.com/FunandCurriculumoriente... "  
call WebPageOpenActivityStarter .StartActivity
```

```
when PadletButton .Click  
do set WebPageOpenActivityStarter . DataUri to " https://padlet.com/inmathenglishteacher/x4kolh0t... "  
call WebPageOpenActivityStarter .StartActivity
```

```
when YoutubeButton .Click  
do set WebPageOpenActivityStarter . DataUri to " https://www.youtube.com/channel/UCeLNcsVy0woSQGU... "  
call WebPageOpenActivityStarter .StartActivity
```

```
when eTwinningButton .Click  
do set WebPageOpenActivityStarter . DataUri to " https://twinspace.etwinning.net/44721/home "  
call WebPageOpenActivityStarter .StartActivity
```

```
when BlogButton .Click  
do set WebPageOpenActivityStarter . DataUri to " https://faceit.blog/ "  
call WebPageOpenActivityStarter .StartActivity
```

```
when WebButton .Click  
do set WebPageOpenActivityStarter . DataUri to " https://agatamurzynska.wixsite.com/faceit "  
call WebPageOpenActivityStarter .StartActivity
```



Worksheet 6

Software Documentation

Activity: Students study in detail and document the different parts of the Quiz Application.

Time: 30 Minutes

DESCRIPTION
<div style="display: flex; align-items: flex-start;"> <p>“What is software documentation?”</p> </div> <p>Software documentation is written text or illustration that accompanies computer software or is embedded in the source code. It either explains how it operates or how to use it.</p> <p style="text-align: right;">(Wikipedia)</p> <p>Documentation is often divided into the following categories:</p> <ul style="list-style-type: none"> • Installation: describes how to install a programme or device • Reference: detailed descriptions of particular items presented in alphabetical order. • Tutorial: teaches a user how to use the product. <p style="text-align: right;">(www.webopedia.com)</p>
PREPARATION
<p>Building working groups</p> <ul style="list-style-type: none"> • Students get in 8 intercultural mixed groups. • Each group gets a designated letter from A to H (i.e. 8 groups are formed). • Once you are in your group choose a catchy name, e.g. <i>App Rangers, Lions</i>, etc. • Each group gets a specific task of the Quiz documentation (see next page).
<p>Connecting to AppInventor (Greek students only)</p> <ul style="list-style-type: none"> • Open a browser, e.g. Chrome, Firefox. • In the address bar of your browser type the URL http://ai2.appinventor.mit.edu . • Sign in with your Google Account. • After this you’re presented with the AppInventor environment.
<p>Load the Quiz file</p> <ul style="list-style-type: none"> • A final version of the Quiz file has already been sent to the Greek students’ e-mail account as an attachment. • Import this file into your AppInventor account. • You are now ready to study the components.



GROUP E

Group Name: _____

Students: _____

TASK	DONE?
<p>Documenting</p> <ul style="list-style-type: none">• Read your group assignment carefully.• Each group gets a small part of the entire Quiz application in form of screenshots.• In order to understand your assignment you may load the app on your smartphone, experiment with it, and browse through the different screens in Blocks or Designer View.• Write your comments around the screenshot as shown in the example.• After completion you are invited to present your results in the plenary session.	



Screen 2 (Blocks View)

```
initialize global chosenList to create empty list
initialize global ListIndex to 1
initialize global userAnswer to 0
initialize global correctAnswer to 0
initialize global score to 0

initialize global language to ''
initialize global category to ''
```



Screen 2 (Blocks View)

```

when Screen2.Initialize
do
  set global category to call myTinyDB.GetValue
  tag categoryID
  valueIfTagNotThere ""
  set global language to call myTinyDB.GetValue
  tag languageID
  valueIfTagNotThere ""
  set CategoryLabel.Text to get global category
  set LanguageLabel.Text to get global language
  if get global category = "English"
  then set global chosenList to list from csv table text "After all that fresh air today, I'm sure I'll s..."
  else if get global category = "General"
  then
    if get global language = "english"
    then set global chosenList to list from csv table text "___ are the main characters in Myths and Legend..."
    else if get global language = "german"
    then set global chosenList to list from csv table text "___ sind die Hauptfiguren in Mythen und Legende..."
    else if get global language = "greek"
    then set global chosenList to list from csv table text "___ είναι οι κεντρικοί χαρακτήρες των Μύθων κ..."
    else if get global language = "polish"
    then set global chosenList to list from csv table text "___ są głównymi postaciami w Mitach i Legendach..."
    else if get global language = "spanish"
    then set global chosenList to list from csv table text "___ son los principales personajes en mitos y l..."
  
```



Screen 2 (Blocks View)

```

else if
  get global category = "Geography"
then
  if
    get global language = "english"
  then
    set global chosenList to list from csv table text "Which is the capital of Germany?,Hamburg,Munic..."
  else if
    get global language = "german"
  then
    set global chosenList to list from csv table text "Wie heißt die Hauptstadt von Deutschland?,Hambur..."
  else if
    get global language = "greek"
  then
    set global chosenList to list from csv table text "Ποια είναι η πρωτεύουσα της Γερμανίας;,Αμβούργο,..."
  else if
    get global language = "polish"
  then
    set global chosenList to list from csv table text "Które miasto jest stolicą Niemiec? ,Hamburg,Mona..."
  else if
    get global language = "spanish"
  then
    set global chosenList to list from csv table text "¿Cuál es la capital de Alemania?,Hamburgo,Múnich..."
else if
  get global category = "History"
then
  if
    get global language = "english"
  then
    set global chosenList to list from csv table text "When did the First World War end?", 8 November ..."
  else if
    get global language = "german"
  then
    set global chosenList to list from csv table text "Wann schrieb Shakespeare seine Werke?,Im Mittela..."
  else if
    get global language = "greek"
  then
    set global chosenList to list from csv table text "Πότε έγραψε ο William Shakespeare τα δραματικά τ..."
  else if
    get global language = "polish"
  then
    set global chosenList to list from csv table text "W jakiej epoce William Szekspir pisał swoje dram..."
  else if
    get global language = "spanish"
  then
    set global chosenList to list from csv table text "¿Cuándo escribió William Shakespeare sus obras d..."
  
```



Screen 2 (Blocks View)

```
else if
  get global category = "Maths"
then
  if
    get global language = "english"
  then
    set global chosenList to list from csv table text "How much is 2/3 + 4/9 ?, 1 1/3, 1 1/9, 2 2/3 , 1 4/9 ..."
  else if
    get global language = "german"
  then
    set global chosenList to list from csv table text "Berechne: 2:3+(4:9)=x, 1 1/3, 1 1/9, 2 2/3 , 1 4/9 2..."
  else if
    get global language = "greek"
  then
    set global chosenList to list from csv table text "Πόσο κάνει 2/3 + 4/9 = ;, 1 1/3, 1 1/9, 2 2/3 , 1 ..."
  else if
    get global language = "polish"
  then
    set global chosenList to list from csv table text "Ile wynosi 2/3 + 4/9 ?", 1 1/3, 1 1/9, 2 2/3 , 1 4/9 ..."
  else if
    get global language = "spanish"
  then
    set global chosenList to list from csv table text "¿Cuánto es 2/3 + 4/9 = ?", 1 1/3, 1 1/9, 2 2/3, 1 4/9, ..."
  call initializeQA
```



Worksheet 6

Software Documentation

Activity: Students study in detail and document the different parts of the Quiz Application.

Time: 30 Minutes

DESCRIPTION
<div style="display: flex; align-items: flex-start;"> <p>“What is software documentation?”</p> </div> <p>Software documentation is written text or illustration that accompanies computer software or is embedded in the source code. It either explains how it operates or how to use it.</p> <p style="text-align: right;">(Wikipedia)</p> <p>Documentation is often divided into the following categories:</p> <ul style="list-style-type: none"> • Installation: describes how to install a programme or device • Reference: detailed descriptions of particular items presented in alphabetical order. • Tutorial: teaches a user how to use the product. <p style="text-align: right;">(www.webopedia.com)</p>
PREPARATION
<p>Building working groups</p> <ul style="list-style-type: none"> • Students get in 8 intercultural mixed groups. • Each group gets a designated letter from A to H (i.e. 8 groups are formed). • Once you are in your group choose a catchy name, e.g. <i>App Rangers, Lions</i>, etc. • Each group gets a specific task of the Quiz documentation (see next page).
<p>Connecting to AppInventor (Greek students only)</p> <ul style="list-style-type: none"> • Open a browser, e.g. Chrome, Firefox. • In the address bar of your browser type the URL http://ai2.appinventor.mit.edu . • Sign in with your Google Account. • After this you’re presented with the AppInventor environment.
<p>Load the Quiz file</p> <ul style="list-style-type: none"> • A final version of the Quiz file has already been sent to the Greek students’ e-mail account as an attachment. • Import this file into your AppInventor account. • You are now ready to study the components.



GROUP F

Group Name: _____

Students: _____

TASK	DONE?
<p>Documenting</p> <ul style="list-style-type: none">• Read your group assignment carefully.• Each group gets a small part of the entire Quiz application in form of screenshots.• In order to understand your assignment you may load the app on your smartphone, experiment with it, and browse through the different screens in Blocks or Designer View.• Write your comments around the screenshot as shown in the example.• After completion you are invited to present your results in the plenary session.	



Screen 2 (Blocks View)

```
to initializeQA
do
  set NextButton . Visible to false
  set ScoreLabel . Text to join " Score: "
  get global score
  set IndexLabel . Text to join get global ListIndex
  " )"
  set QuestionLabel . Text to select list item list
  select list item list index get global chosenList
  get global ListIndex
  index 1
  set AnswerButton1 . Text to select list item list
  select list item list index get global chosenList
  get global ListIndex
  index 2
  set AnswerButton2 . Text to select list item list
  select list item list index get global chosenList
  get global ListIndex
  index 3
  set AnswerButton3 . Text to select list item list
  select list item list index get global chosenList
  get global ListIndex
  index 4
  set AnswerButton4 . Text to select list item list
  select list item list index get global chosenList
  get global ListIndex
  index 5
  set global correctAnswer to select list item list
  select list item list index get global chosenList
  get global ListIndex
  index 6
```



Worksheet 6

Software Documentation

Activity: Students study in detail and document the different parts of the Quiz Application.

Time: 30 Minutes

DESCRIPTION



“What is software documentation?”

Software documentation is written text or illustration that accompanies computer software or is embedded in the source code. It either explains how it operates or how to use it.

(Wikipedia)

Documentation is often divided into the following categories:

- Installation: describes how to install a programme or device
- Reference: detailed descriptions of particular items presented in alphabetical order.
- Tutorial: teaches a user how to use the product.

(www.webopedia.com)

PREPARATION

Building working groups

- Students get in 8 intercultural mixed groups.
- Each group gets a designated letter from A to H (i.e. 8 groups are formed).
- Once you are in your group choose a catchy name, e.g. *App Rangers, Lions*, etc.
- Each group gets a specific task of the Quiz documentation (see next page).

Connecting to AppInventor (Greek students only)

- Open a browser, e.g. Chrome, Firefox.
- In the address bar of your browser type the URL <http://ai2.appinventor.mit.edu> .
- Sign in with your Google Account.
- After this you're presented with the AppInventor environment.

Load the Quiz file

- A final version of the Quiz file has already been sent to the Greek students' e-mail account as an attachment.
- Import this file into your AppInventor account.
- You are now ready to study the components.



GROUP G

Group Name: _____

Students: _____

TASK	DONE?
<p>Documenting</p> <ul style="list-style-type: none">• Read your group assignment carefully.• Each group gets a small part of the entire Quiz application in form of screenshots.• In order to understand your assignment you may load the app on your smartphone, experiment with it, and browse through the different screens in Blocks or Designer View.• Write your comments around the screenshot as shown in the example.• After completion you are invited to present your results in the plenary session.	



Screen 2 (Blocks View)

```
when AnswerButton1 .Click
do
  set global userAnswer to 1
  call deactivateButtons
  call updateScore
```

```
when AnswerButton2 .Click
do
  set global userAnswer to 2
  call deactivateButtons
  call updateScore
```

```
when AnswerButton3 .Click
do
  set global userAnswer to 3
  call deactivateButtons
  call updateScore
```

```
when AnswerButton4 .Click
do
  set global userAnswer to 4
  call deactivateButtons
  call updateScore
```

```
to deactivateButtons
do
  set AnswerButton1 . Enabled to false
  set AnswerButton2 . Enabled to false
  set AnswerButton3 . Enabled to false
  set AnswerButton4 . Enabled to false
```



Worksheet 6

Software Documentation

Activity: Students study in detail and document the different parts of the Quiz Application.

Time: 30 Minutes

DESCRIPTION



“What is software documentation?”

Software documentation is written text or illustration that accompanies computer software or is embedded in the source code. It either explains how it operates or how to use it.

(Wikipedia)

Documentation is often divided into the following categories:

- Installation: describes how to install a programme or device
- Reference: detailed descriptions of particular items presented in alphabetical order.
- Tutorial: teaches a user how to use the product.

(www.webopedia.com)

PREPARATION

Building working groups

- Students get in 8 intercultural mixed groups.
- Each group gets a designated letter from A to H (i.e. 8 groups are formed).
- Once you are in your group choose a catchy name, e.g. *App Rangers, Lions*, etc.
- Each group gets a specific task of the Quiz documentation (see next page).

Connecting to AppInventor (Greek students only)

- Open a browser, e.g. Chrome, Firefox.
- In the address bar of your browser type the URL <http://ai2.appinventor.mit.edu> .
- Sign in with your Google Account.
- After this you're presented with the AppInventor environment.

Load the Quiz file

- A final version of the Quiz file has already been sent to the Greek students' e-mail account as an attachment.
- Import this file into your AppInventor account.
- You are now ready to study the components.



GROUP H

Group Name: _____

Students: _____

TASK	DONE?
<p>Documenting</p> <ul style="list-style-type: none">• Read your group assignment carefully.• Each group gets a small part of the entire Quiz application in form of screenshots.• In order to understand your assignment you may load the app on your smartphone, experiment with it, and browse through the different screens in Blocks or Designer View.• Write your comments around the screenshot as shown in the example.• After completion you are invited to present your results in the plenary session.	



Screen 2 (Blocks View)

```
when NextButton .Click
do
  set global ListIndex to (get global ListIndex + 1)
  if (get global ListIndex ≤ length of list list) (get global chosenList)
  then
    call activateButtons
    call whiteButtons
    call initializeQA
  else
    call showFinalScore
```

```
to activateButtons
do
  set AnswerButton1 . Enabled to true
  set AnswerButton2 . Enabled to true
  set AnswerButton3 . Enabled to true
  set AnswerButton4 . Enabled to true
```

```
to whiteButtons
do
  set AnswerButton1 . BackgroundColor to white
  set AnswerButton2 . BackgroundColor to white
  set AnswerButton3 . BackgroundColor to white
  set AnswerButton4 . BackgroundColor to white
```




Screen 2 (Blocks View)

```
to showFinalScore
do
  call hideComponents
  set ScoreLabel . FontSize to 26
  set ScoreLabel . Text to join
    " You scored "
    get global score
    " out of "
    length of list list
    get global chosenList
    " questions! "
```

```
to hideComponents
do
  set IndexLabel . Visible to false
  set QuestionLabel . Visible to false
  set AnswerButton1 . Visible to false
  set AnswerButton2 . Visible to false
  set AnswerButton3 . Visible to false
  set AnswerButton4 . Visible to false
  set NextButton . Visible to false
```



Screen 2 (Blocks View)

```
to updateScore
do
  if (get global userAnswer = get global correctAnswer)
  then
    set global score to (get global score + 1)
  call highlightCorrectAnswer
  set ScoreLabel.Text to (join (" Score: " (get global score)))
  call myTinyDB.StoreValue
    tag "userScore"
    valueToStore (get global score)
  set NextButton.Visible to true
```

```
to highlightCorrectAnswer
do
  if (get global correctAnswer = 1)
  then set AnswerButton1.BackgroundColor to green
  else if (get global correctAnswer = 2)
  then set AnswerButton2.BackgroundColor to green
  else if (get global correctAnswer = 3)
  then set AnswerButton3.BackgroundColor to green
  else set AnswerButton4.BackgroundColor to green
```



Screen 1 (Blocks View)

```
when PlayButton .Click
do
  call MyTinyDB .StoreValue
  tag "categoryID"
  valueToStore CategoryListPicker . Selection
  call MyTinyDB .StoreValue
  tag "languageID"
  valueToStore LanguageListPicker . Selection
  open another screen screenName "Screen2"
```



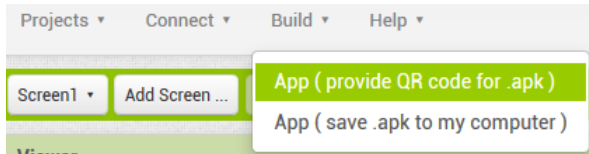
Worksheet 7

Testing the QuizApp

Activity: Students test the Quiz App and comment on positive attributes as well as weaknesses.

Time: 15 Minutes

*Follow your facilitator and complete the tasks below (put a mark if completed).
Don't hesitate to ask if there is something you are not sure of.*

PREPARATION
<p>Building working groups</p> <ul style="list-style-type: none"> You stay in the same group you worked previously (Worksheet 6).
<p>Download on smartphone/tablet</p> <ul style="list-style-type: none"> Download the Android Package Kit (.apk file) to your smartphone or tablet as it's shown below <div style="text-align: center; margin-top: 10px;">  </div>

TASK	DONE?
<p>Why do we need testing?</p> <ul style="list-style-type: none"> Brainstorm your ideas in the group. Present it to the plenary session. 	
<p>What should be tested?</p> <ul style="list-style-type: none"> Discuss in the plenary session. 	
<p>Strong/Weak attributes</p> <ul style="list-style-type: none"> Discuss in your group: <ul style="list-style-type: none"> ➤ How does the Quiz App behave? ➤ What are its strong attributes? ➤ Are there any weaknesses? ➤ What do you like most? ➤ What is missing or at fault? Write down your observations in the respective columns of the following table: 	



STRONG	WEAK