

# App Overview & Purpose

## Quiz app

**(No more than 150 words for the 3 answers combined)**

Describe the overall purpose of your App:

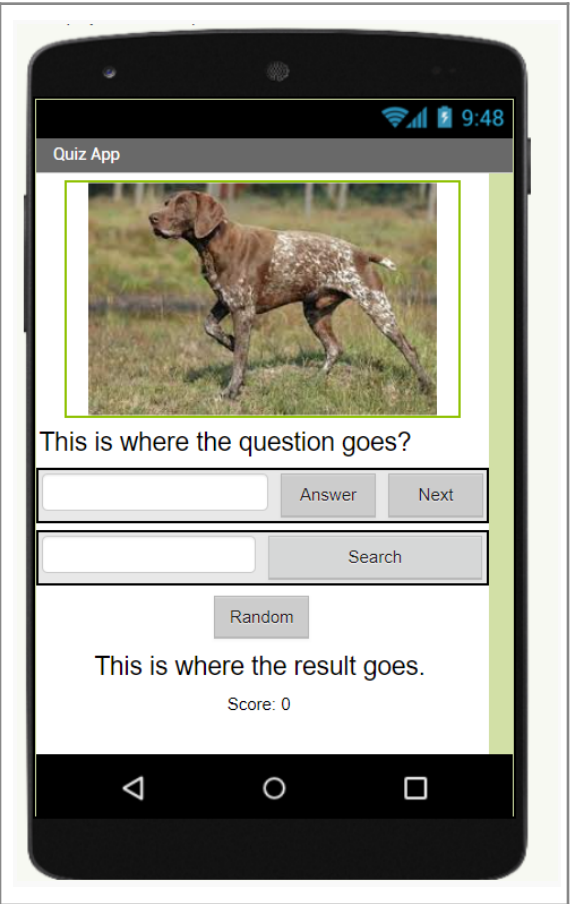
The purpose of my app is to quiz you on four different questions, and see how well you do.

Describe the function of your App as displayed in your video

The function is to answer questions you are given and you can search for question if you want to

Describe the input and the output of the app as displayed in the video:

The app's input and output are the answers the user gives and the response they receive, or the search.



## Digital Distribution (optional):

Your teacher may optionally ask you to upload the .aia and .apk files as well as [generate the QR code for App Inventor Projects](#). If use the Add files option on the bottom of your portfolio page.

How to Install my App: (Scan Here)



## Viewer & Components

Copy and paste the Viewer and Components of your app here

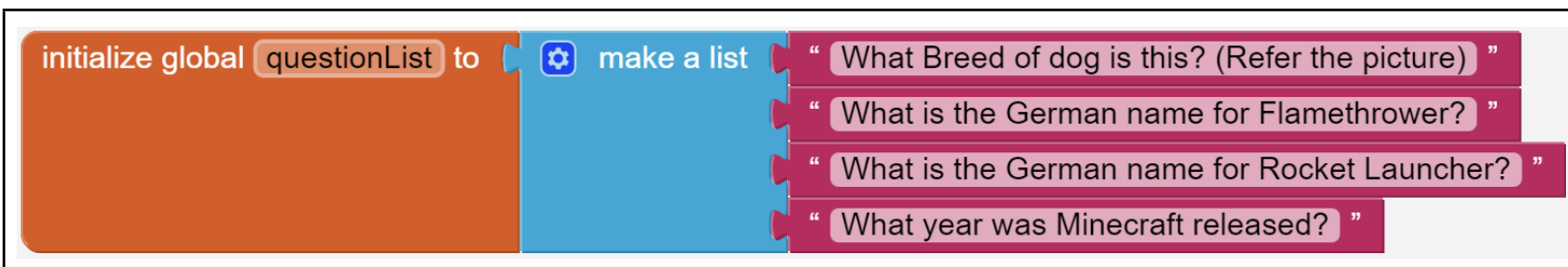
# Copy/Paste your Viewer & Components windows here.



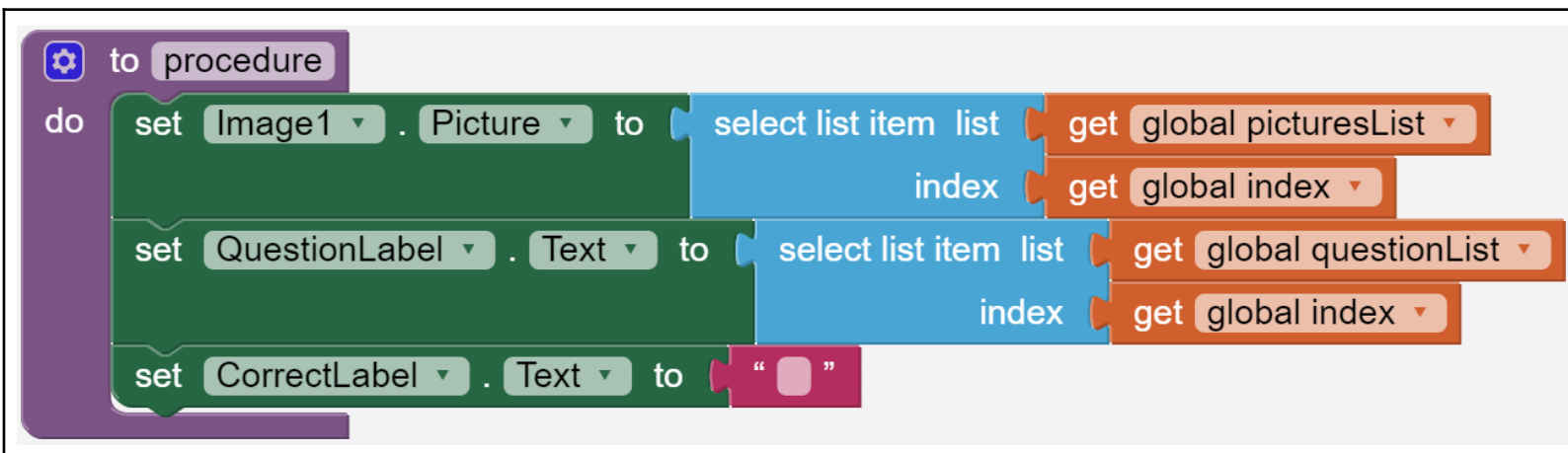
## 3b. List

Capture and paste two program code segments you developed during the administration of this task that contain a list (or other collection type) being used to manage complexity in your program. **Approx. 200 words (for all subparts of 3b combined, exclusive of program code)**

i. The first program code segment must show how data has been stored in the list.



ii. The second program code segment must show the data in the same list being used, such as creating new data from the existing data or accessing multiple elements in the list, as part of fulfilling the program's purpose.



Then, provide a written response that does all three of the following:

iii. Identifies the name of the list being used in this response

The name of the list is questionList, so it can be easily identified as the list holding the questions.

iv. Describes what the data contained in the list represent in your program

The list data represents the questions asked to the user, that will then be answered by them.

v. Explains how the selected list manages complexity in your program code by explaining why your program code could not be written, or how it would be written differently, if you did not use the list.

AP sidebar note: The data abstraction must make the program easier to develop (alternatives would be more complex) or easier to maintain (future changes to the size of the list would otherwise require significant modifications to the program code).

The program would be almost impossible to write if you didn't use a list due to the fact that you won't be able to easily access the answers or photos for the program to use.

## Design & Enhancements:

Be sure to provide **screenshots** along with your **explanations** for each of the **enhancements** that you made.

The code block is a Scratch script starting with an 'if' block. The condition is 'select list item list' with 'index' and 'get global answered' compared to 'false'. The 'then' block contains: 'replace list item list' with 'index' and 'replacement' set to 'true'; 'set global score to' 'get global score' plus '1'; and 'set ScoreLabel . Text to' 'join' of 'Score: ' and 'get global score'.

This code Gives the user a point if the answer is correct, but only once per question

```

initialize global picturesList to [make a list]
  DOG.jpg
  Flam.jpg
  rock.jpg
  block.jpg

initialize global answerList to [make a list]
  " German shorthaired pointer "
  " flammenwerfer "
  " raketwerper "
  " 2011 "

initialize global questionList to [make a list]
  " What Breed of dog is this? (Refer the picture) "
  " What is the German name for Flamethrower? "
  " What is the German name for Rocket Launcher? "
  " What year was Minecraft released? "

```

These are the new question, answer, and image lists to be used later

```

to displayQuestion index
do
  set global index to get index
  call procedure

when SearchButton .Click
do
  call searchProcedure
  keyword Search_box . Text

to searchProcedure keyword
do
  for each i from 1
  to length of list list get global questionList
  by 1
  do
    if contains text upcase select list item list get global questionList
    piece upcase get keyword index get i
  then
    call displayQuestion index get i

```

These blocks work with the search bar to find and bring up a question

**Note:** The above section may repeat based upon how many large Event Handlers you have.

# Complete Code

- Put the code screenshots in the following order
  - Variables
  - Procedures
  - Event handlers

```
when Screen1.Initialize do call procedure
```

```
to procedure do set Image1.Picture to select list item list get global picturesList index get global index set QuestionLabel.Text to select list item list get global questionList index get global index set CorrectLabel.Text to ""
```

```
to displayQuestion index do set global index to get index call procedure
```

```
when SearchButton.Click do call searchProcedure keyword Search_box.Text
```

```
to searchProcedure keyword do for each i from 1 to length of list list get global questionList by 1 do if contains text upcase select list item list get global questionList index get i piece upcase get keyword then call displayQuestion index get i
```

```
initialize global answered to make a list false false false false
```

```
initialize global questionList to make a list "What Breed of dog is this? (Refer the picture)" "What is the German name for Flamethrower?" "What is the German name for Rocket Launcher?" "What year was Minecraft released?"
```

```
initialize global answerList to make a list "German shorthaired pointer" "flammenwerfer" "raketenwerper" "2011"
```

```
initialize global picturesList to make a list "DOG.jpg" "Flam.jpg" "rock.jpg" "block.jpg"
```

```
initialize global score to 0
```

```
initialize global index to 1
```

```
when RandomButton.Click do set global index to random integer from 1 to length of list list get global questionList call procedure
```

```
when NextButton.Click do set global index to get global index + 1 if get global index > length of list list get global questionList then set global index to 1 call procedure
```

```
when AnswerButton.Click do if compare texts upcase AnswerTextBox.Text upcase select list item list get global answerList index get global index then set CorrectLabel.Text to "Good, that is correct" if select list item list get global answered index get global index = false then replace list item list get global answered index get global index replacement true set global score to get global score + 1 set ScoreLabel.Text to join Score get global score else set CorrectLabel.Text to join "Sorry, that is incorrect. The correct answer is" select list item list get global answerList index get global index
```