

**3 a.****3.a.i.**

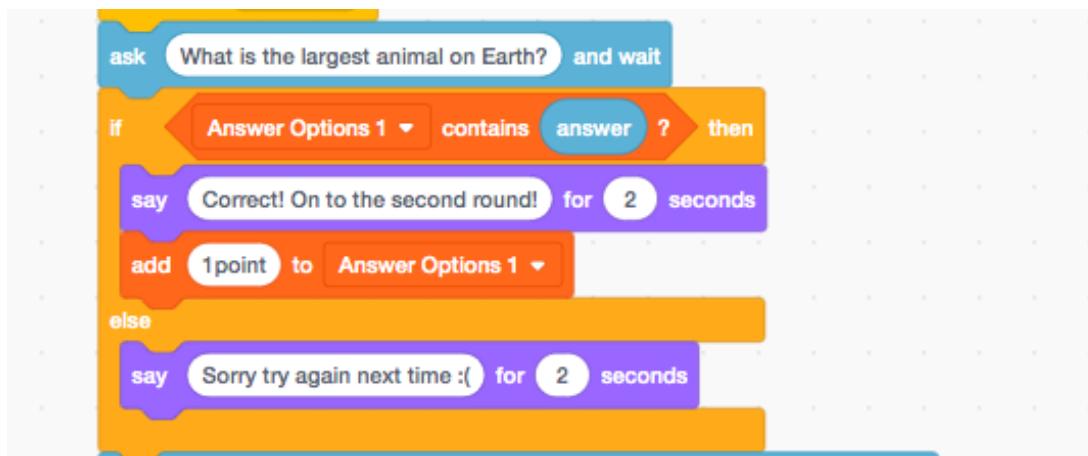
For my create performance task I chose to create a questionnaire where questions are asked at random to the user and they are prompted to answer it.

**3.a.ii.**

Within the video you can see the questions being asked have two different things that can happen depending on the different answers the user may provide. For example, if the user were to give a correct answer they would move on and a point will be added to their score, but if they get the answer wrong the point will not be rewarded to the user. At the end of the questionnaire, the user is shown their score.

**3.a.iii.**

Input and output are used constantly in the video. Input coming from the answers being given by the user and output when being told overall score/progression in the game.

**3 b.****3.b.i.****3.b.ii.****3.b.iii.**

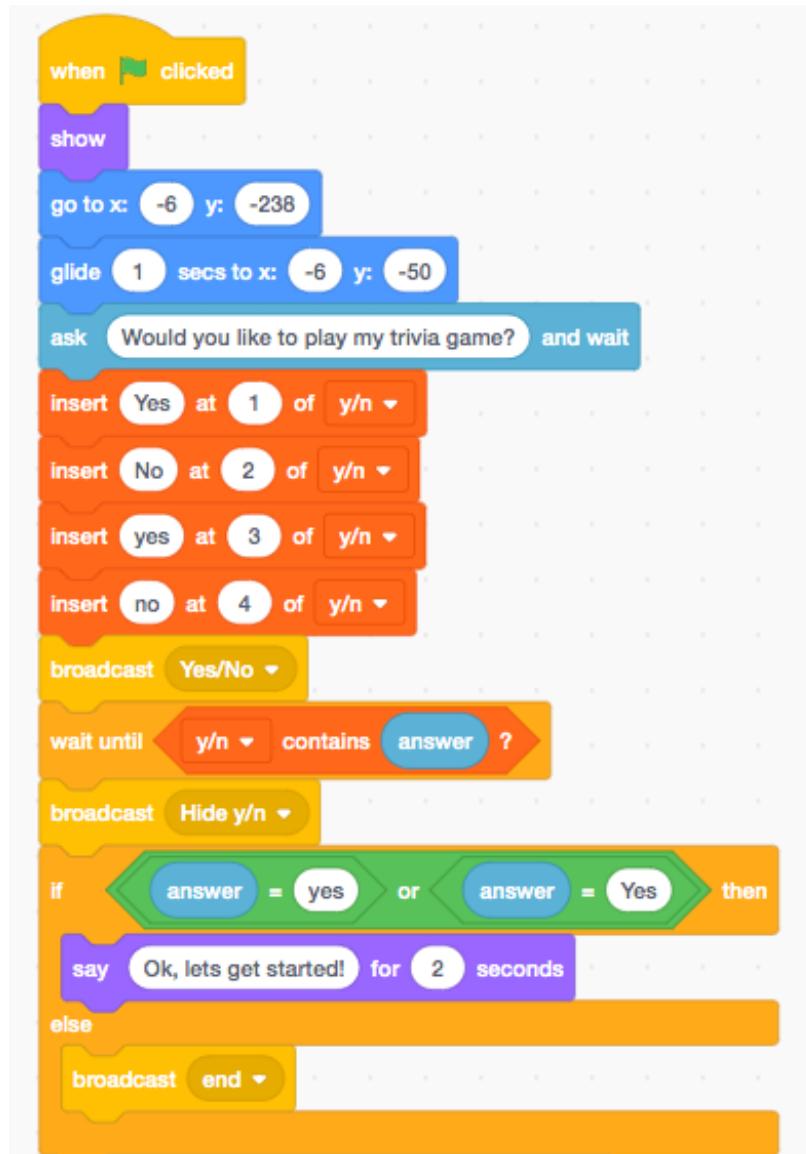
In the photo above these variables are being added to the "Answers Option 1" list.

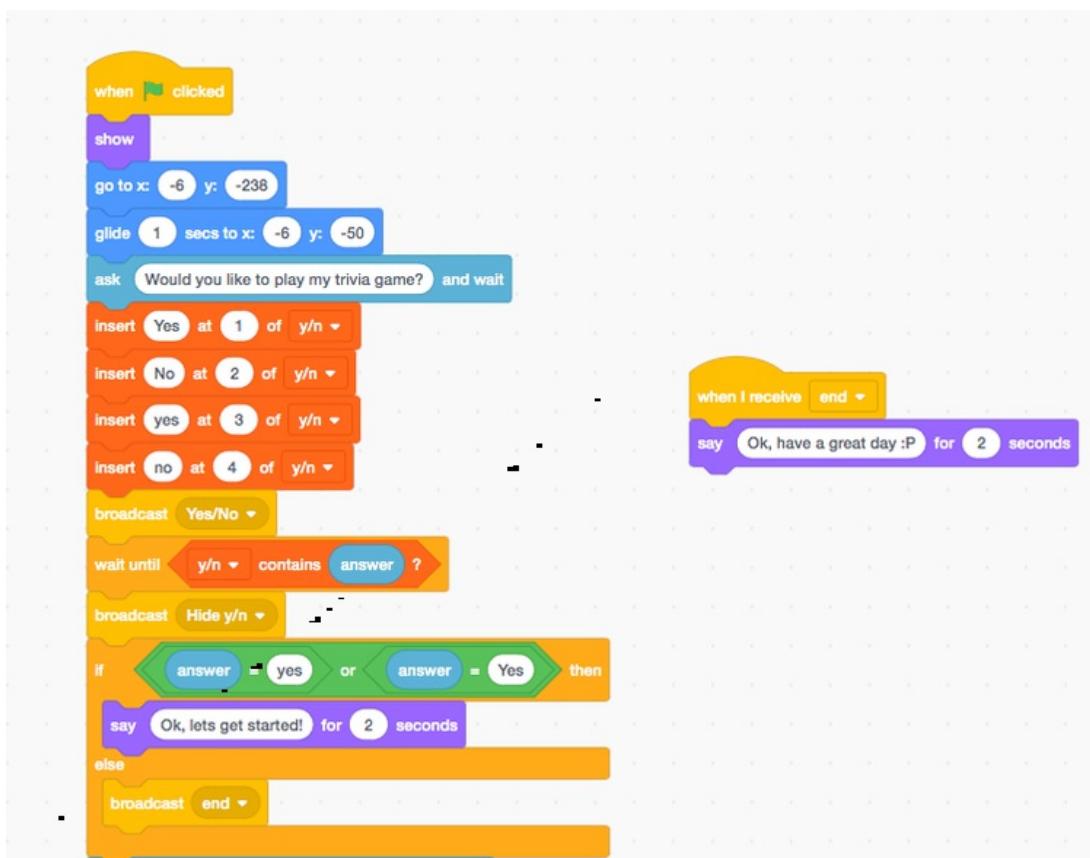
**3.b.iv.**

The data within the "Answer Option 1" list is being used to allow only those answers to modify the points awarded to the user after they input one of the answers that are on that same list. If the same answer is not input when the question is asked they are not rewarded the point and it affects their final score.

**3.b.v.**

Within this code, it was important for me to use the list to reduce further complexity. I used a list in order to take as many answer possibilities as possible while still maintaining a balance of right and wrong. This is inclusive of spaces in between words and capitalization in the program. When using scratch I saw that the only way that multiple answers can be accepted in this manner is if you use the list to assign the answer to multiple variables.

**3 c.****3.c.i.**

**3.c.iii.**

The procedure identified is the decision of the user to participate in the trivia game or not. This was important to implement into the code because I wanted the user to have the option even though it is more likely than not that the user will want to participate in the game.

**3.c.iv.**

Within the code, I used the different variables of yes and no to determine the user's participation in the game. I set 4 different versions of yes and no including the capitalized versions of the words so they will still be accepted if typed. Then with these variables, I was able to dictate through code whether or not the user wanted to play by only allowing them to move on if they typed, "Yes" or "yes". Alternatively, if the user were to answer "No or "no" the code would broadcast to end the procedure with the message "Ok, have a great day :P".

**3 d.****3.d.i.**

**First call:**

In the code, the first call is the option to type "Yes" or "yes" both options are accepted as answers.

**Second call:**

In the code, the second call is the option to type "No" or "no" both options are accepted as answers.

**3 d.ii.**

**Condition(s) tested by first call:**

By using a list I applied the condition of whether the user will type "yes" or "Yes"

**Condition(s) tested by second call:**

Using this same list the condition of typing "no" or "No" is tested.

**3.d.iii.**

**Results of the first call:**

If the first call was acted upon it would trigger the beginning of the game for the user to answer the five preset questions.

**Results of the second call:**

If the user typed either variation of no the code would have ended with a short message after their response.