3. WRITTEN RESPONSES

3 a.

3.a.i.

This program was created in Scratch MIT. The overall purpose of the program is to test critical thinking skills and challenge users to guess 8 letter words.

3.a.ii.

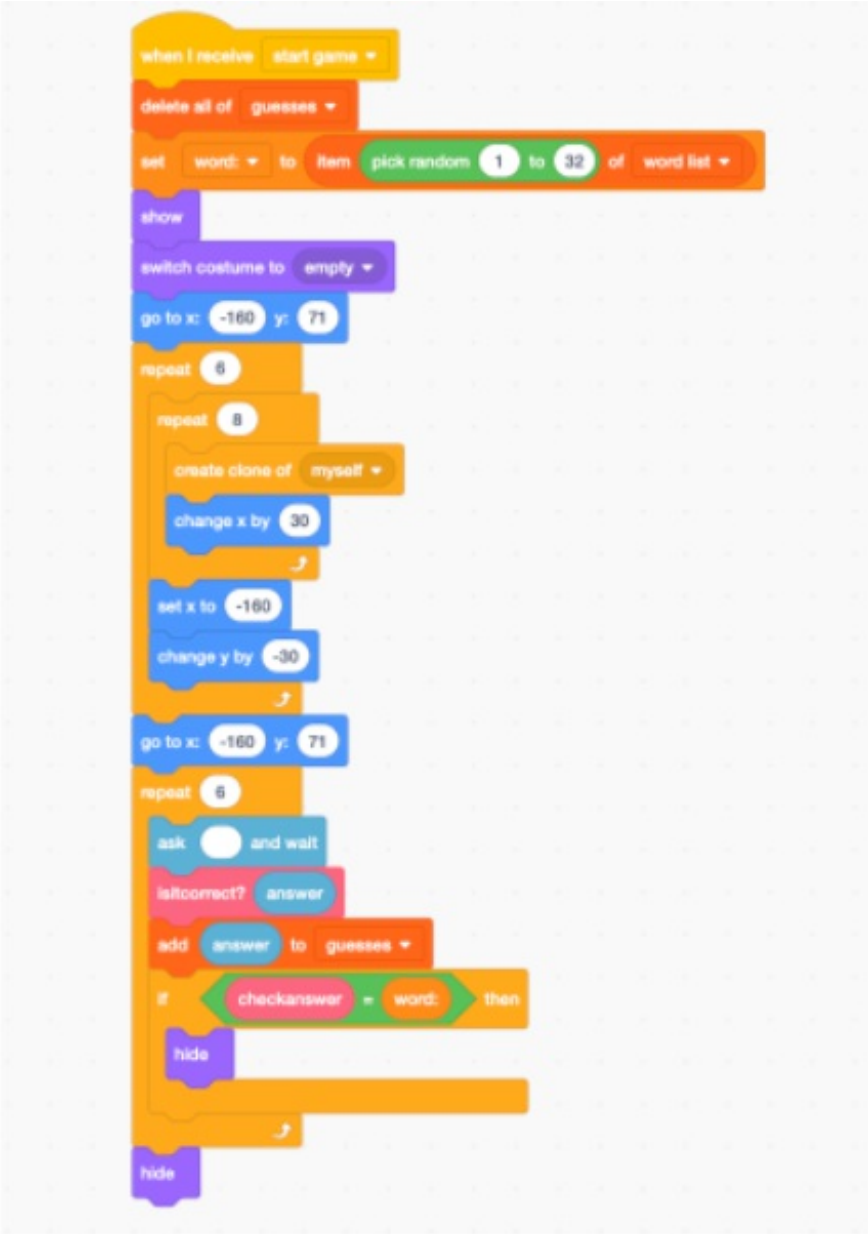
In the video, the user initially clicks the “help” button to learn how to play. Then, it shows the user trying to guess the word by inputting an 8-letter-word into the textbox, clicking enter, then having the color scheme of the corresponding letters be outputted. The user correctly answered the first word “touching” in 4 tries so the program changed to the “You Win” screen. The user clicked play again, inputting new words but couldn't guess the correct word (“sandwich”) in 6 tries.

3.a.iii.

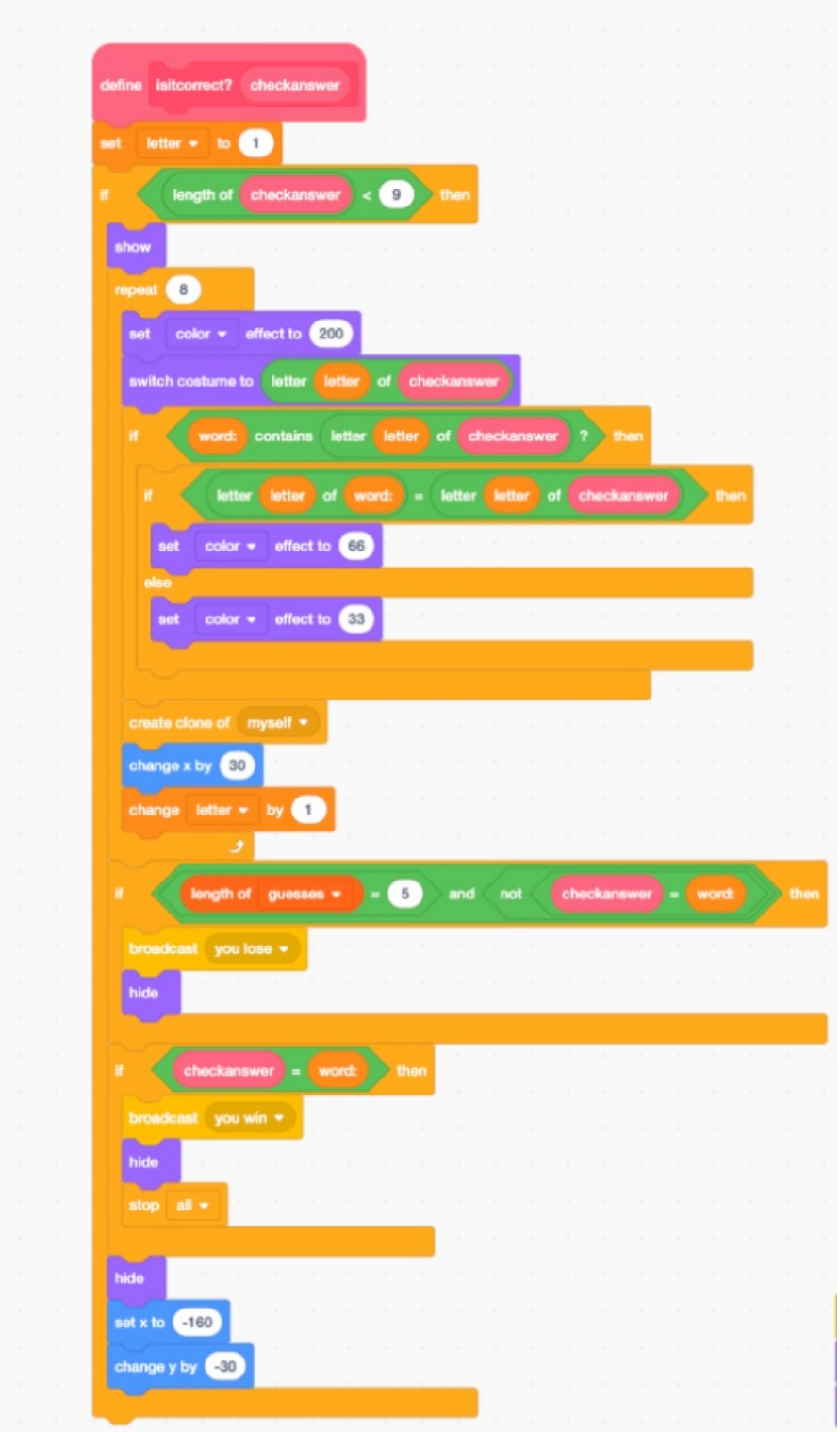
In the video, the user inputs the word “fraction.” The output displays the colors that correspond to the letters imputed. This created an output of the letters F, R, and A being in red, indicating that those letters weren’t in the word. The letters C and I were displayed in red, indicating that those letters were in the word and in the right place. The letters T, O, and N were displayed in yellow, indicating that those letters were in the word but in the wrong place.

3 b.

3.b.i.



3.b.ii.



3.b.iii.

The name of the list is “guesses.”

3.b.iv.

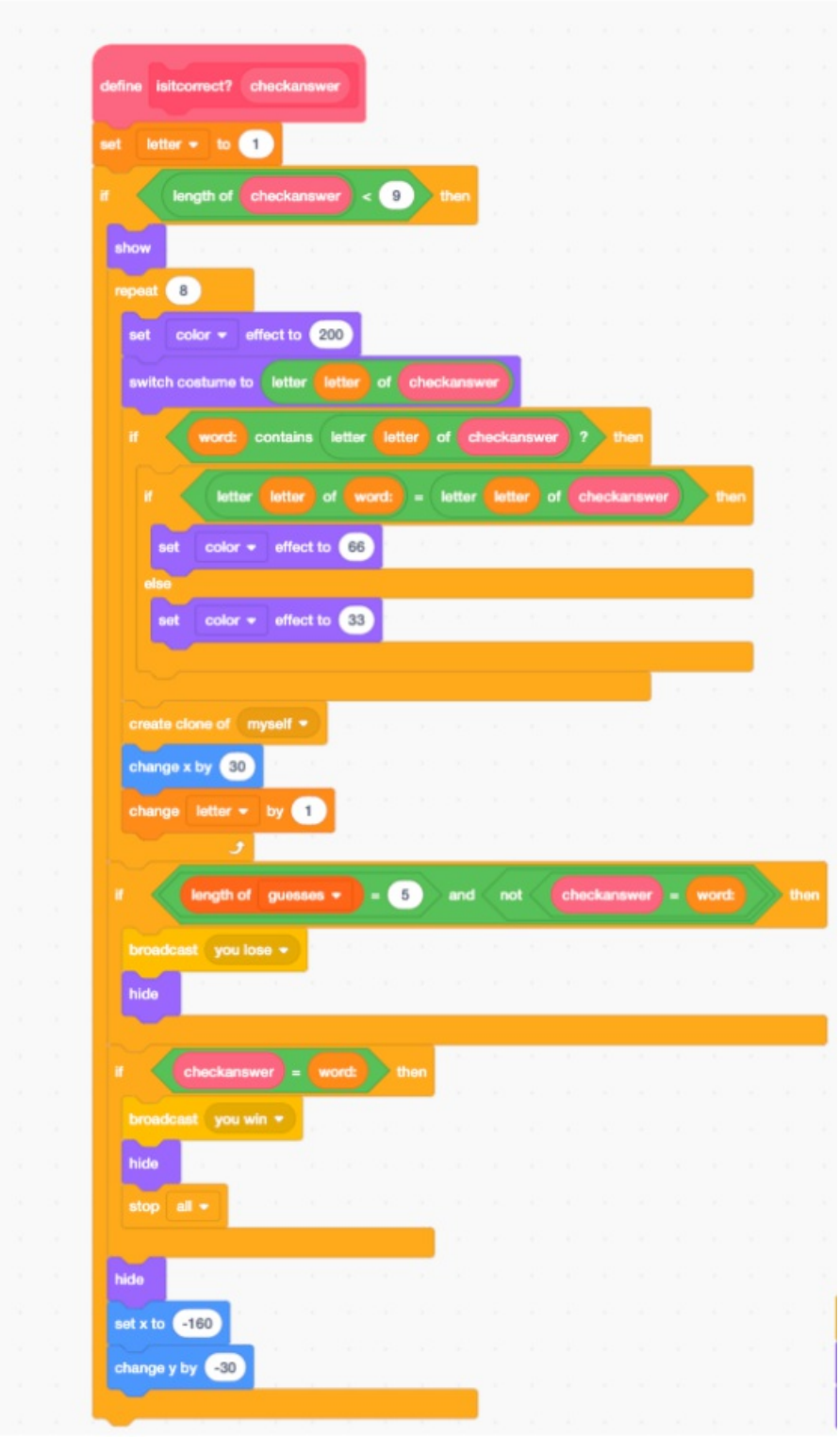
The data contained in the list represents all of the user’s word inputs. When the user inputs a response, their answer is added to the list “guesses” and this process is repeated until either the user guessed the word or the user has guessed 6 times (when the length equals 6).

3.b.v.

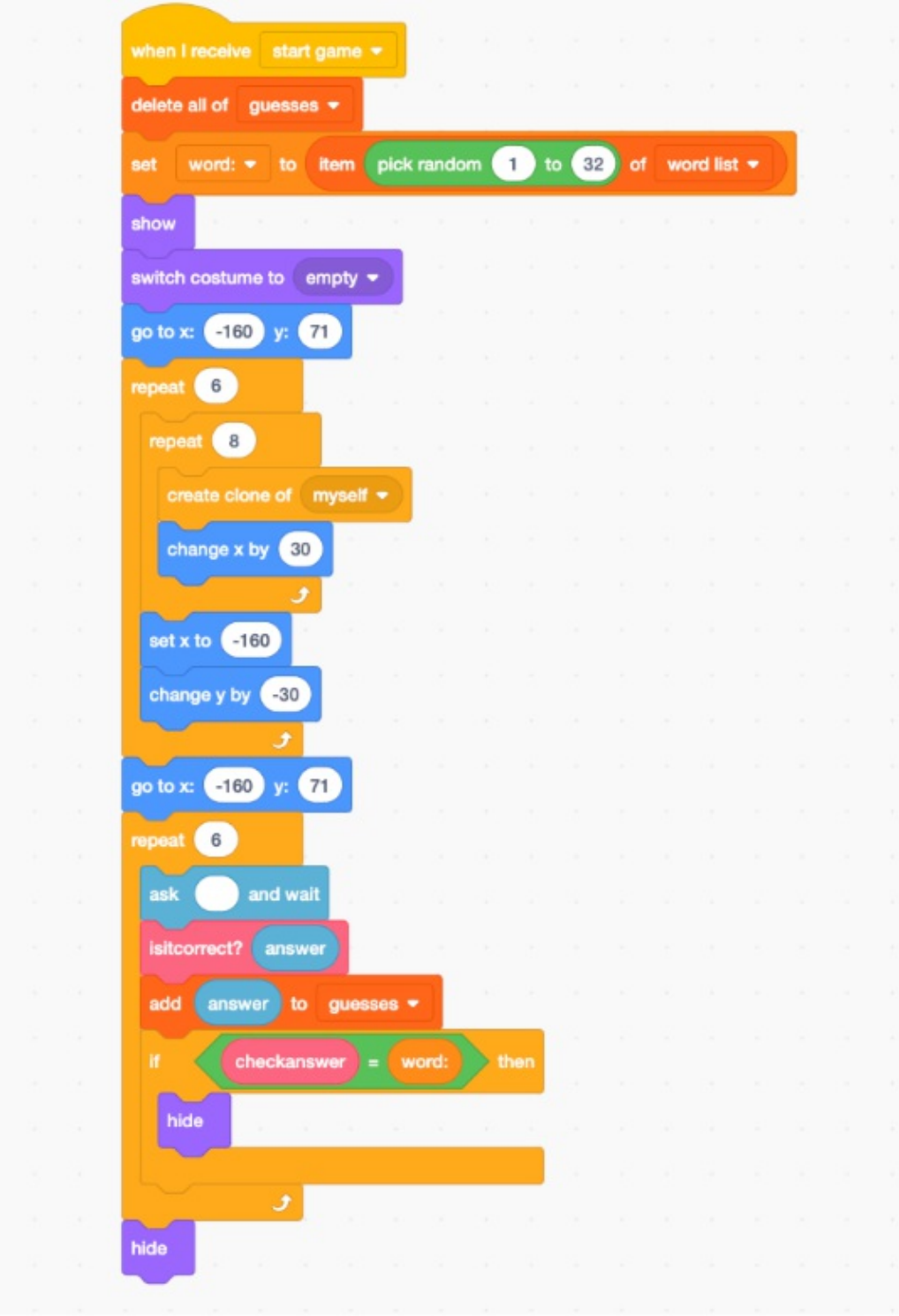
For the program to function without the list “word list”, the variable “words” would have to be set to each individual word at different times in the game meaning, a lot more code will be added, causing overlaps and complexity in the program. Additionally, without the list “guesses,” the game would never end as the user would have infinite guesses. The point of this list is to stop the game when the user has guessed 6 times and without this, the user wouldn’t be challenged by the game.

3 c.

3.c.i.



3.c.ii.



3.c.iii.

The procedure of “checkanswer” helps to contribute to the overall functionality of the program by checking the answer and seeing if it matches the correct word or letter positions. The algorithm is needed every time a user inputs their guess. Therefore, this procedure helps with the efficiency and accuracy of the program by having the code be applicable and accessible whenever it is called.

3.c.iv.

When a user inputs an answer, it calls the algorithm “check answer.” If the input is less than 9 letters, the program will not output anything. However, if the input is 8 letters, the letters will be checked for position and accuracy. If the letter is in the right position, the output letter will turn green. If the letter is in the wrong position but in the word, the output letter will turn yellow while if the letter isn’t present in the word, the output letter will turn red. This part of the algorithm uses iteration as this process is repeated 8 times for the 8 letters imputed. This algorithm also determines when the “You Win” or “You Lose” screen will show up. This action demonstrates selection. If you guessed the word or ran out of guesses, the program will switch screens. This procedure is able to take the parameters and users input to output them in the correct color and position on the screen. This action demonstrates selection. This procedure also includes sequencing as the program is able to do the tasks in order from the code.

3 d.

3.d.i.

First call:

The first call is right letter, right position.

Second call:

The second call is right letter, wrong position.

3 d.ii.

Condition(s) tested by first call:

The conditions tested by the first call are whether or not the user inputs the right letter in the right position. If the user imputes the right letter in the right position, those corresponding letters will turn green.

Condition(s) tested by second call:

The conditions tested by the second call are whether or not the user inputs the right letter in the wrong position. If the user imputes the right letter in the wrong position, those corresponding letters will turn yellow.

3.d.iii.

Results of the first call:

The result of the first call is the letter color changing to green, indicating that the user is closer to solving the word.

Results of the second call:

The result of the second call is the letter color changing to yellow, indicating that the user will have to move the letter position around to solve the word.