Creating the scratch program

Initializing steps:

1. Create variable “round” and initialise it to zero.
2. Create variable “best_score” and initialise it to zero.
3. Create array “score” and initialise it to zero.
4. Create array “vowels” and let it contain a,e,i,o,u. (consonants will be from the else statement).
5. Create variable “best_word” and set its length to zero.
6. Create array “user_guess”. (This will contain the length of the user's word).
7. Create variable “random_num_gen”and initialize it to zero.
8. Create variable “counter” and initialize it to zero.
9. Create variable “counter2” and initialize it to zero.
10. Create variable “Counter3” and initialize it to zero.
11. Create variable user_letter.
12. Create array “letters(99)”.
13. Create array “letters(1-9)”.
14. Assignment of “letters(99)”is show at the end.
15. Create variable “Letter_checker”.

Running the program steps:

1. Display the rules of the game to the user the rules of the game.
2. Create a loop which states: Until Round == five the program will continue.
3. Create a loop which will continue until “random_num_gen” is equal to nine.
4. While true it will pick random numbers between one and ninety-nine (each representing a letter from the letters array) and assign them to letter1-9 array.
5. Display the random random letters(1-9).
6. Insert the user’s answer into the “user_guess” variable and also assign their answer to the “user_letter” variable.
7. Create a loop which states: that until the counter is greater than the length of “user_guess” the loop will continue.
8. The two points below are error checkers and the second one splits into a complicated two part if, else. For clarity I have split them from this block of instructions.

Error Checking:

1. Check if “letters(1-9)” contains item (counter) of “user_guess”. If not, reset the score and the user’s “best_word” to zero. Display a message informing the user that they have made a mistake and the penalties involved.
2. Check if the “user_guess” contains vowels. If it does display block of Code A as described below. If not display block of code B as described below.

Block of code A:

1. Create a loop which will be true unless counter is greater than one.
2. If item (counter) of “user_guess” is equal to item (counter2) of letters(1-9) replace the item (counter2) of “letters(1-9) with ✔ , set “letter_checker” to one and change “counter2” by one.
3. If “letter_checker” is equal to one, then change “counter3” to one and set “counter2” to zero.
4. Display message “Point earned for a vowel!”, and increase their score by one.
5. Check if the “score_counter” is greater than the “best_score” variable. If true reset “best_word” to “user_letter” and the “best_score” variable to the “score_counter” variable.

Block of code B:

1. Create a loop which will be true unless counter is greater than one.
2. If item (counter) of “user_guess” is equal to item (counter2) of letters(1-9) replace the item (counter2) of “letters(1-9) with ✔ , set “letter_checker” to one and change “counter2” by one.
3. If “letter_checker” is equal to one, then change “counter3” to one and set “counter2” to zero.
4. Display message “Two point earned for a consonant!”, and increase their score by two.
5. Check if the “score_counter” is greater than the “best_score” variable. If true reset “best_word” to “user_letter” and the “best_score” variable to the “score_counter” variable.

Running the program steps:

9. Finally display to the user their “best_word” and “score”.

Interacting with the scratch project:

1. The user is presented with nine letters randomly generated each round.
2. The aim of the game is to get as large a score as possible, score is calculated by assigning each vowel in the word one point and each consonant two points.
3. The user is given five rounds (i.e. five different attempts to make words from the nine random letters). After five rounds the program will end and the user will be notified of their longest word.
4. An actual word dictionary is not used in the scratch program and therefore it is assumed that the user will play fairly and not input random letters from the list.
5. If the user enters invalid letters they are punished by bringing their score down to zero and setting their best word to “0”. This is a harsh punishment, but the implementation of the game mechanic makes the user type very carefully and adds to the tense atmosphere when playing the game. I found that this added to the game as a whole.

The ninety-nine letters are assigned as follows:

12 letters:e
9 letters:a,i
8 letters:o
6 letters:n,r,t
4 letters:l,s,u,d
3 letters:g
2 letters:b,c,m,p,f,h,v,w,y,k
1 letters:j,x,q,z

Image sources: vocabulary.co.il