## 3. WRITTEN RESPONSES

### 3 a.

### 3.a.i.

This program, made in Scratch MIT, has an overall purpose to train a user's mouse reflex skills if coming from the older generation or simply someone who may have a slow time clicking on links.

### 3.a.ii.

This program functionality is based on two features. One of the features is a difficulty system that modifies the size and wait time of three different types of sprite:target. The second feature is the clicking mechanic based on a mouse cursor clicking on a target to earn points.

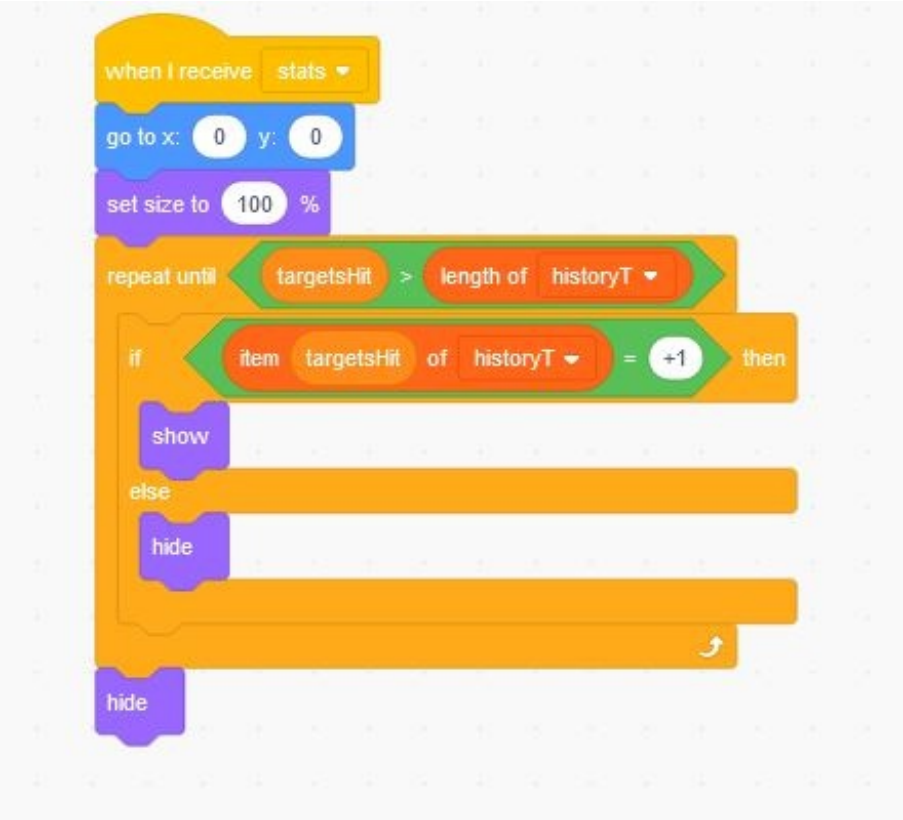
### 3.a.iii.

The user inputs a value greater than 1 for the output:difficulty to be set on hard mode or less than 1 for the output:difficulty to be set on easy mode. The mouse clicks are inputs that will work only if clicked on a sprite:target which the output:score and output:targetsHit will be updated correspondingly to a target's point value. At the very end of the program, the user will be granted his or her history, in chronological order, of all types of targets that were clicked.

### 3 b.

### 3.b.i.

### 3.b.ii.




### 3.b.iii.

This list named “historyT” is a list of all targets that were clicked by representing them by their corresponding values from three different targets: +1, +2, and +3. The code shown above corresponds to one of the targets with a value of +1 when clicked.

### 3.b.iv.

The data contained in this list represents the specific targets that were hit in chronological order.

### 3.b.v.

Therefore, the lists manage complexity in the program because at the very end, the user can see each target that shows up on the screen to know what targets were hit throughout the ten seconds. This is accomplished using the list’s position numbers for the code to track each position’s value which correspond for a sprite:target to appear. Without this list, it would be difficult because the variable block would only track the amount but not order and say and ask/answer blocks would require some extra code and may get overwhelmed by the large inputs of multiple target values.

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


define diffMod difficulty value
set drag mode not draggable
if difficulty > 1 then
repeat until timer > 10
set size to 50 %
set x to pick random -200 to 200
set y to pick random -140 to 140
show
wait 0.7 seconds
hide
wait 0.3 seconds
else
repeat until timer > 10
set size to 70 %
set x to pick random -185 to 185
set y to pick random -130 to 130
show
wait 1 seconds
hide
wait 0.3 seconds

### 3.c.ii.


### 3.c.iii.

The identified procedure above accounts for the difficulty modifier that is accomplished by a parameter that accepts values based on a ask/answer block where one target can change in size or the wait time with specific boundaries to prevent a sprite to be at the edge. This program contributes to the overall functionality of the program because without the program the output:score and output: targetsHit cannot be updated because the two inputs mouse click and difficulty cannot be established.

### 3.c.iv.

The procedure's parameter accepts values greater or less than 1 when called based on a ask/answer block. The procedure has selection where a value greater than 1 enables the harder difficulty. Both outcomes are similar with some numbers adjusted. There is a timer paired with a repeat until loop and an operator block for the game to continue running for a set amount of time. Sequencing begins when size is included to increase difficulty. Then, the x & y coordinate blocks have a random operator block with a fixed value for the targets to spawn in various locations and not the edge. Finally, once the target has moved, it will appear on screen for a set amount of seconds so it can disappear. After that, it hides and waits while the second wait block gets enabled to control the spawn rates of the targets. If the second wait time is lower than the first, the targets may disappear faster than normal.

## 3 d.

### 3.d.i.

First call:

The first call is hard difficulty.

Second call:

The second call is easy difficulty

### 3 d.ii.

Condition(s) tested by first call:

If the user inputs the number 2, then the procedure will run the "then" portion of the command and not the "else" portion because the value is greater than 1. Hard Difficulty is enabled.

Condition(s) tested by second call:

If the user inputs the number 0, then the procedure will run the "else" portion of the command and not the "then" portion because the number is less than 1. Easy Difficulty is enabled.

### 3.d.iii.

Results of the first call:

Includes a smaller size for targets, a greater boundary to spawn since the sprites are smaller, and a smaller initial wait time for how long a target will stay on screen.

## Results of the second call:

Includes a larger size for targets, a smaller boundary to spawn since the sprites are bigger, and a bigger initial wait time for how long a target will stay on screen.