

3. WRITTEN RESPONSES

3 a.

3.a.i.

The overall purpose of the program is to entertain users with a win or lose game of click-on-me.

3.a.ii.

The program functionality shows the program monkey sprite welcoming the user to the game and asking if they want an easy or hard game. We then see the user playing a game of Click On Me with "Easy" settings", winning the game by clicking most times on the monkey. We then see the sprite congratulating the user for winning and showing how many times they clicked on it and the elephant. The monkey sprite then asks if the user wishes to play again. The user says yes and chooses a "Hard" setting, this time losing the game by clicking too many times on the elephant. The monkey sprite tells the user they have lost and shows them how many times they clicked on it and the elephant. When asked if they want to play again, the user types "No" and the game says thank you for playing.

3.a.iii.

The input shown is the user typing "1" or "2" for an easy or hard game, typing "yes" or "no" to play again or not, and clicking on the elephant and monkey sprites. The output we see in the video is the sprite introducing itself, asking the user if the user would like to play an easy or hard game, explaining the instructions, displaying whether they won or lost along with the list of times they clicked on either of the sprites, asking if the user wants to play again, and saying thank you for playing when the user is done.

3 b.

3.b.i.



```

+ play + game + EasyHard +
if EasyHard = 1
  broadcast beginMoving
  repeat until length of ElephantOrChimp > 0
  go to random position
  wait 3 secs
  broadcast stopMoving
  go to x: 0 y: 0
  if winTimes > loseTimes
    say Congrats you win! for 2 secs
  else
    say Oh too bad you lost. for 2 secs
  say These are all the times you clicked me and my elephant friend for 2 secs
  say ElephantOrChimp for 4 secs
else
  if EasyHard = 2
    broadcast beginMoving2
    repeat until length of ElephantOrChimp > 0
    go to random position
    wait 1 secs
    broadcast stopMoving
    go to x: 0 y: 0
    if winTimes > loseTimes
      say Congrats you win! for 2 secs
    else
      say Oh too bad you lost. for 2 secs
    say These are all the times you clicked me and my elephant friend for 2 secs
    say ElephantOrChimp for 4 secs
  else
    say Oh no! you clicked on a wrong key, please type in 1 or 2. for 2 secs
    ask Press 1 for an EASY game and press 2 for a HARD game! and wait
    set difficulty to answer
    play game difficulty
  
```

3.b.iii.

The name of the list being processed in this response is “ElephantOrChimp”.

3.b.iv.

The data in “ElephantOrChimp” is a list of strings that contains the words “elephant” and “chimp” based on which animal the user clicks on.

3.b.v.

The list manages complexity in the program because it stores the number of elephant and chimp clicks and the pattern that the user clicks on them to show the user at the end. Without the list there would not be a way of showing the user the pattern in which the chimp and elephant were clicked.

3 c.

3.c.i.

```

+ play + game + EasyHard +
if EasyHard = 1
  broadcast beginMoving
  repeat until length of ElephantOrChimp > 9
    go to random position
    wait 3 secs
  broadcast stopMoving
  go to x: 0 y: 0
  if winTimes > loseTimes
    say Congrats you win! for 2 secs
  else
    say Oh too bad you lost. for 2 secs
  say These are all the times you clicked me and my elephant friend for 2 secs
  say ElephantOrChimp for 4 secs
else
  if EasyHard = 2
    broadcast beginMoving2
    repeat until length of ElephantOrChimp > 9
      go to random position
      wait 1 secs
    broadcast stopMoving
    go to x: 0 y: 0
    if winTimes > loseTimes
      say Congrats you win! for 2 secs
    else
      say Oh too bad you lost. for 2 secs
    say These are all the times you clicked me and my elephant friend for 2 secs
    say ElephantOrChimp for 4 secs
  else
    say Oh no! you clicked on a wrong key, please type in 1 or 2. for 2 secs
    ask Press 1 for an EASY game and press 2 for a HARD game! and wait
    set difficulty to answer
    play game difficulty
  
```

3.c.ii.

```

when clicked
  set winTimes to 0
  set loseTimes to 0
  set ElephantOrChimp to (list)
  go to x: 0 y: 0
  say Hello! My name is Chimp! Welcome to my game! for 2 secs
  say Let's play catch me. I'll move and you click to catch. for 2 secs
  say Everytime you click me you get a Win Point for 3 secs
  say Also if you click on my elephant friend, you get a lose point for 3 secs
  say Okay here we go! for 1 secs
  ask Press 1 for an EASY game and press 2 for a HARD game! and wait
  set difficulty to answer
  play game difficulty
  ask Would you like to play again? and wait
  set playAgain to answer
  repeat until (playAgain = no)
  set winTimes to 0
  set loseTimes to 0
  set ElephantOrChimp to (list)
  ask Press 1 for an EASY game and press 2 for a HARD game! and wait
  set difficulty to answer
  play game difficulty
  ask Would you like to play again? and wait
  set playAgain to answer
  say Thank you for playing my game! for 2 secs

```

3.c.iii.

The procedure itself is the click-on-me game and, based on the user input, plays the game on an easy or difficult setting. It contributes to the overall functionality of the program because it is the primary source of using the program which is to play the click-on-me game and is called each time the user wishes to play.

3.c.iv.

The procedure has a parameter called EasyHard. If EasyHard is equal to 1, broadcast beginMoving initiates. Then the sprites will move to a random position and wait three seconds before jumping to a new random position. This repeats until length of the list ElephantOrChimp is greater than nine. After that, broadcast stopMoving initiates, the sprite moves to x position 0 and y position 0. If winTimes is greater than loseTimes the sprite will say, "Congrats you win!" for two seconds. If not the sprite will say, "Oh too bad, you lost!" for two seconds. Finally, the sprite will say the times clicked on ElephantOrChimp. If EasyHard is equal to 2, the exact same thing happens except the difficulty increases to hard because there will only be a 1 second wait before the sprites jump to next random position. Finally, if easyHard is anything other than 1 or 2, it will say that they clicked a wrong key, ask the user to type in a 1 or 2, and call the procedure again with the new input.

3 d.

3.d.i.

First call:

The first call is with the number 1.

Second call:

The second call is with the number 2.

3 d.ii.

Condition(s) tested by first call:

Testing to see if the first IF section executes which will cause the game to start at an easy mode having the sprites jump to random positions every 3 seconds.

Condition(s) tested by second call:

Testing to see if the first IF section executes which will cause the game to start at a hard mode having the sprites jump to random positions every 1 second.

3.d.iii.

Results of the first call:

The game is initiated with a difficulty of "easy" as the chimp and elephant jump to random positions at a pace of three seconds per move.

Results of the second call:

The game is initiated with a difficulty of "hard" as the chimp and elephant jump to random positions at a pace of one second per move.