Create – Applications From Ideas
Written Response Submission Template

Please see Assessment Overview and Performance Task Directions for Student for the task directions and recommended word counts.

Program Purpose and Development

2a)

The programming language I used was javaScript and the Code.org app lab. The purpose of my game is an entertaining clicker game that is centered around birds. A user has to choose a character, name it, and then click that character only among many other birds. When the bird is clicked, it moves to a random position on the screen. If the user clicks their character 10 times then they win. If they lose 3 lives, or run out of time then they lose the game. Some stats such as score, name, and character chosen are displayed on the last page. If the user wishes, they can replay the game by clicking a button.

2b)

I made this game without collaboration. I remembered one of my previous projects for a clicker game that wasn't quite finished, so I decided to build upon it. I made a list of all the things I wanted my game to accomplish; a scoring system, lives, a timer, character selection/customization, and showing 'stats'. Some of the scoring was already in the previous project so I made some minor modifications, but I had to do everything else from scratch. I thought that the game wasn't developed so I wanted to add a timer. When it runs out, it is supposed to move to the 'lose screen'. This was difficult because I had to wait 60 seconds for the timer to run out so I could see the effect it had every time. Another difficulty I had was with the character customization. I forgot about setImageURL so I was having trouble with making the bird what the player chose. I went through some of my previous projects and I remembered it, so I was able to change the image. In order to solve both of these problems I had to research a little online.

2c)

```
13  //set bird for game + store bird name
14  onEvent("bird1", "click", function() {
15    setName(bird1);
16    setBirdImage(bird1Image);
17    timeFunction();
18  });
```

```
83
84  //define bird
85  function setName(whichBird) {
86    setScreen("game_screen");
87    chosenBird = whichBird;
88    var birdName = getText("nameInput");
89    setText("birdNameLose", birdName);
90    setText("birdNameWin", birdName);
91  }
92
93  //create timer
94  function timeFunction() {
95    var time = 60;
96    setText("numberTime", time);
97  timedLoop(1000, function() {
98    time--;
99    setText("numberTime", time);
100   if(time === 0) {
101     stopTimedLoop();
102     setScreen("lose_screen");
103   }
104 });
105 }
```

The algorithm I wrote is the onEvent function at the top (lines 14-18). It contains two child functions called setName (85-91) and timeFunction (93-104). The setName function is part of the character customization. It has one parameter called whichBird that is different for each algorithm. The player types a name in the nameInput box, and on line 82 it gets the string from the user input. In order to show the name at the end of the game, it sets birdNameLose and birdNameWin to birdName. In the timeFunction, there is a variable called time. At first, time is equal to 60, but the timedLoop decreases the time by 1 every 1000 milliseconds. Then it updates the value of time. There is an if-statement that says if the time is equal to 0 (time runs out) then the timer will stop and it will change to lose_screen. Both of these functions need to occur regardless of which character is chosen, so it is included for each parent onEvent function.

2d)

```
6  var chosenBird;
7  var bird1Image = "Screenshot-2020-03-05-at-2.19.44-PM.png";
8  var bird2Image = "Screenshot-2020-03-05-at-2.18.10-PM.png";
9  var bird3Image = "Screenshot-2020-03-05-at-2.18.59-PM.png";
10 var bird4Image = "Screenshot-2020-03-05-at-2.19.17-PM.png";
11
12
13 //set bird for game + store bird name
14 onEvent("bird1", "click", function() {
15   setName(bird1);
16   setBirdImage(bird1Image);
17   timeFunction();
18 });
```

```
75
76 //FUNCTION LIST
77 //sets bird images + abstraction
78 function setBirdImage(birdImage) {
79   setImageURL("chosenBird", birdImage);
80   setImageURL("chosenBirdLose", birdImage);
81   setImageURL("chosenBirdWin", birdImage);
82 }
```

setBirdImage (lines 78-82) is an abstraction that I developed on my own in order to manage the timing system. This function helps to manage complexity in my program because I didn't want to repeat the setImageURL line 3 times for each bird for a total of 12 times. I decided to use a parameter birdImage, and four variables bird1Image, bird2Image, etc. Each of the variables stores an image corresponding to each bird. Using the parameter in setBirdImage, I didn't have to include the lengthy image URL for each of the lines from 79-81. The use of functions in the onEvent block makes the code easier to read and debug. This specific function does not contain a mathematical or logical concept, but the other two functions have them as well as some of the other sections in my code. Without the setBirdImage function, it would be difficult to read, and the use of parameters helps to keep everything organized.