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)

My program is called “Nyan Cat Clicker”. The programming language that was used was JavaScript in a program called AppLab. This program is a clicker game, in which the user clicks “Start” and it changes screen to the “PlayScreen”. This is where a timer will start, the user has 30 seconds to click on the Nyan Cat multiple times, in which every click counts as a score of 1. The goal of this program is to click the Nyan Cat as many times as possible to get a higher score everytime in 30 seconds. Once 30 seconds are over, it will switch the screen to the “GameOverScreen”. This screen shows that the game is over and how many times you clicked the Nyan Cat, along with the highest score the user received from all the games played. The app also has a restart button and pause button.

2b)

There were several problems that were presented in the code. One problem was setting the timer to countdown 30 seconds and stopping the time when the pause button was clicked, then resuming the time again. This was an independent development. I had to find out how to resume the game, such that the time that was paused did not restart again after clicking the resume button. To solve this, I made a function called Time and called it in the event handler of the resume button, which in result would change just the time in intervals of 1000 milliseconds, but not move back to 30 seconds. Another problem, also an independent development, was updating the high score the user received every time the score was higher. The code would update the high score, but to the score the user received, regardless if it was higher or not. I decided to add in an if statement, such that if the score the user received was greater than the score from other games, it would update the high score value to that score.

2c)

If your captured code segment is an image, click the Picture icon to browse to the location of your saved image. If your captured code segment is text, include it with the rest of your 2c written response below.

```
onEvent("btnPlay", "click", function(event) {
 setScreen("playScreen");
 Time();
 });

onEvent("Catimage", "click", function(event) {
  setPosition("Catimage",randomNumber(0,200),randomNumber(0,395));
 getText("Score", Score);
 Score = Score + 1;
 setText("Score", Score);
   });

 onEvent("btnRestart", "click", function(event) {
 setScreen("playScreen");
 seconds = 30;
 Score = 0;
 setText("Time", seconds);
 setText("Score", Score);
 Time();
   });

onEvent("btnPause", "click", function(event) {
setScreen("PauseScreen");
clearInterval(myInterval);
```

```
      });

onEvent("btnResume", "click", function(event) {
 setScreen("playScreen");
  Time();
   });

onEvent("btnRestart1", "click", function(event) {
 setScreen("playScreen");
 seconds = 30;
 Score = 0;
 setText("Time", seconds);
 setText("Score", Score);
 Time();
});
```

I used this algorithm in order to set the code in an organized way, such that it would work efficiently and when the a certain button is pressed, it would execute the code written for the button in the event handler. For example when the event handler “Catimage” was called, it would execute the code in that event handler when the Nyan Cat image was clicked, then adding adding 1 point to the score and setting the text of the score to update everytime the Nyan Cat was clicked such that the score would keep increasing. Something else to notice is the Time function being called in several event handlers. This was done in order to set the timer to restart or to start again depending on the event handler that is executed.

2d)

If your captured code segment is an image, click the Picture icon to browse to the location of your saved image. If your captured code segment is text, include it with the rest of your 2d written response below.

```
function Time() {
myInterval = setInterval(function() {
  seconds = seconds - 1;
  setText("Time",seconds);

  if (seconds === 0){

   if (Score > HighScore) {
     HighScore = Score;
    setKeyValue("HighScore", Score, function () {
      setText("HighScore", HighScore);
    });
  }
    setText("txtScore","You have received a score of " + Score + " Nyan Cats that
you clicked!!");
    setScreen("GameOverScreen");
    clearInterval(myInterval);
  }
}, 1000);
}
```

In order to use the , such as certain lines of code are not repeated, I made my code use abstraction is by using this Time() function, and implementing it into the event handlers so the code is reduced and not being repeated on certain event handlers. My first thought for my Time() function was using the setInterval() function and the if statement and applying it to certain event handlers over again, such that the 30 second interval would run, and the screen would change when the 30 seconds would end. Using abstraction helped me implement the setInterval() function and the if statement such that it would be unnecessary to repeat code and have extra lines of code, adding less complexity to the algorithms that I used into making this code.