Allison Smith
Mr. Ettlin
AP Comp Sci, period 1
November 20, 2019

1. Provide a written response for your video that:
    - identifies the programming language;
    - identifies the purpose of your program; and
    - explains what the video illustrates.

My program is written in p5js and is used for entertainment. The video illustrates how the code runs and how to play the snake game. The goal of the snake game is to become as big of a snake as you can without going off the play screen or touching other parts of the body. In order to move the snake, you use the arrow keys for up, down, left, and right. The video shows the different screen of gameplay; start screen, play screen, and the end screen. You can replay the game by pressing the "r" key on your keyboard.

2. Describe the incremental and iterative development process of your program, focusing on two distinct points in that process. Describe the difficulties and/or opportunities you encountered and how they were resolved or incorporated. In your description clearly indicate whether the development described was collaborative or independent. At least one of these points must refer to independent program development.

The development process of my program was independent. First, I started coding the food to make it appear on the screen. Then, I created a snake and added velocity so the snake would move. I struggled to make the snake and food collide, as I needed them to be at the same location in order for my code to work. I realized my snake and food weren't exactly in the same position and that was why my food didn't move to a new location when the snake touched it. I fixed the location problem by using math floor. With this, I was able to guarantee the food and snake were on the same "grid", the snake would move as if it were on a checkerboard.

Another problem I faced was coding the body for the snake. I struggled to add new body pieces to the array. I originally had code that said to add a piece to the body when the array was equal to 1. I realized later that my snake never got a body because the array could never add a body piece because of my code. I fixed my problem by with a for loop.

3. Capture and paste the program code segment that implements an algorithm (marked with an oval) that is fundamental for your program to achieve its intended purpose. Your code segment must include an algorithm that integrates other algorithms and integrates mathematical and/or logical concepts. Describe how each algorithm within your selected algorithm functions independently, as well as in combination with others, to form a new

algorithm that helps to achieve the intended purpose of the program. *(Approximately 200 words)*

*Scroll down for the code

My if statement functions independently because it checks if the snake and any other part of the snake are in the same location. The code allows me to stop the game because my rules state that the snake can't touch itself. The for loop checks all parts of the snake. The code works in combination with other because in order to lose the game, there must be code detecting when the snake hits itself.

4. Capture and paste the program code segment that contains an abstraction you developed (marked with a rectangle in section 3 below). Your abstraction should integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program.

*Scroll down for the code

My abstraction is the code for my snake. It creates the snake, allows it to move, and adds body parts to itself. This method uses math by using an if statement to check if the snake is in the same location as the food. If it is, the snake adds a body piece to the array of the body. My snake code makes my game complex because without it there would be no game. Because of this abstraction, I did not have to re-write the snake code three times for each game mode.

5. Capture and paste your entire program code in this section.
    - Mark with an oval the segment of program code that implements the algorithm and integrates mathematical and /or logical concepts.
    - Mark with a rectangle the segment of program code that represents an abstraction you developed.
    - Include comments or citations for program code that has been written by someone else.
    -

```
//  Allison Smith
//  November 15, 2019

var gameState = 1
var w = 10

function setup() {
  var cnv = createCanvas(800, 800);
  cnv.position((windowWidth-width)/2, 30);
```

```
  // slow down snake movement
  frameRate(10);
}

//creates snake and food
function loadThings(){
    snake = new Snake(400, 400);
    food = new Food(Math.floor(random(0,800/w))*w,Math.floor(random(0,800/w))*w);
}

function checkEdges(){
  //checks edges
    if(snake.head.x<0||snake.head.x>width||snake.head.y<0 || snake.head.y>height){
      gameState = 3
    }
            //checks snake hitting snake
              for(var i = snake.body.length-1; i >= 0; i--)
                if(snake.head.x === snake.body[i].x &&
                  snake.head.y === snake.body[i].y){
                    gameState = 3
                  }
          }

//runs food and snake code
function runThings(){
  food.run();
  snake.run();
}

//each gamestate start code
function draw(){
  background(5,5,5);
    if(gameState === 1){
      startGame()
    }
    else if(gameState === 2){
        playGame()
    }
    else if(gameState === 3){
        loseGame()
    }
```

```
  }

//opening screen
  function startGame(){
     background(20,20,20);
  //title
     fill(250, 250, 250);
     textSize(100);
     text("Snake Game", 100, 300);
  //easy
       fill(0, 250, 0);
       textSize(20);
       text("Press E for Easy", 100, 500);
        if (keyCode === 69) {
          frameRate(10);
          w=40
          loadThings();
          gameState = gameState + 1
    }
  //medium
       fill(255,165,0);
       textSize(20);
       text("Press M for Medium", 325, 500);
       if (keyCode === 77) {
          frameRate(15);
          w=20
         loadThings();
         gameState = gameState + 1
      }
  //hard
       fill(255,0 ,0 );
       textSize(20);
       text("Press H for Hard", 600, 500);
        if (keyCode === 72) {
          frameRate(20);
          w=10
          loadThings();
          gameState = gameState + 1
     }
  //how to play
     fill(255, 100 ,0);
```

```
    textSize(20);
    text("The goal of the game is to become as long of a snake as you can.", 100, 625);
    text( "If you hit one part of your snake with anohter, the game ends.", 100, 650);
    text( "Beat your own highscore to win the game.", 100, 675);
 }

//gameplay screen
 function playGame(){
    background(20,20,20);
    runThings();
    checkEdges();
  }

//lose screen
function loseGame(){
  background(250,0,0);
//score
  fill(0, 250, 0);
  textSize(32);
  text("Score: " + snake.body.length, 650, 50);
//lose
  fill(250, 250, 250);
  textSize(100);
  text("Try Again", 175, 300);
//restart
  fill(250, 250, 250);
  textSize(50);
  text("To restart, press R", 200, 600);
  if (keyCode === 82) {
    gameState = 1
  }
 }
```

```
class Snake{
  constructor(x,y,dx,dy){
    this.head = createVector(x,y);
    this.vel = createVector(dx,dy);
    this.body = [];
  }
```

```
  loadSegment(){
    //load new body segment
    this.body.push(createVector(-100,-100));
  }

  run(){
    //run update, checkEdges, and render functions
    this.checkEdges();
    this.update();
    this.render();
  }

  checkEdges(){
    this.vel.x = 0;
    this.vel.y = 0;
//up arrow
    if(keyCode===38){
      this.vel.x = 0;
      this.vel.y = -w;
    }
//down arrow
    else if (keyCode===40) {
      this.vel.x = 0;
      this.vel.y = w;
    }
//left arrow
    else if (keyCode===37) {
      this.vel.x = -w;
      this.vel.y = 0;
    }
//right arrow
    else if (keyCode===39) {
      this.vel.x = w;
      this.vel.y = 0;
    }
  }

  update(){
    // update the body
    for(var i = this.body.length-1; i >= 0; i--){
```

```
      if(i===0){
        this.body[0].x = this.head.x;
        this.body[0].y = this.head.y;
        }
      else{
        this.body[i].x = this.body[i-1].x;
        this.body[i].y = this.body[i-1].y;
         }
       }
    // update the head
    this.head.add(this.vel);
    // add segment
    if(this.head.x === food.food.x &&
      this.head.y === food.food.y){
         this.loadSegment();
      }
}

  render(){
  // render head
  fill (0,250,0);
  rect(this.head.x, this.head.y, w, w);
  // render the body
  for(var i = 0; i < this.body.length; i++){
    rect(this.body[i].x, this.body[i].y, w, w);
    }
  }
}
```

```
class Food{
  constructor(x,y){
    this.food = createVector(x,y);
  }

run(){
  //run update and render functions
  this.update();
  this.render();
}
```

```
update(){
  //change location if snake and food touch
  if(snake.head.x === this.food.x &&
     snake.head.y === this.food.y){
      if(w==10){
          this.food.x = Math.floor(random(0,79))*w;
          this.food.y = Math.floor(random(0,79))*w;
      }
      if(w===20){
        this.food.x = Math.floor(random(0, 39))*w;
        this.food.y = Math.floor(random(0,39))*w;
      }
      if(w===40){
        this.food.x = Math.floor(random(0,19))*w;
        this.food.y = Math.floor(random(0,19))*w;
      }
    }
}

render(){
  // render the food
  fill(random(255), random(255), random(255));
  rect(this.food.x, this.food.y, w, w);
  }
}
```