Austin Matel
Mr. Ettlin
AP Computer Science Principles, Period 1
21 November 2019

2a.) This project was coded in p5js and is intended to serve as entertainment. To start, the player just needs to move in whichever direction that they want using the arrow keys. Try to touch as many red apples that you can. The apple will move to a random location on the screen when touched and the length of the snake will increase. Don't get tangle because if the head of the snake touches any part of its body, the game ends. Make sure to stay on the screen, hitting the edge of the screen will also result in a loss. After losing, press space to restart. The score is shown in the top right of the screen and try to increase your high score.

2b.) I first started with the idea to create the snake game and then decided to begin by making the food show up on the canvas. Then I tried to make the snake appear on the screen. Then I tried to make the snake move but found a problem. I didn't know how to make the keys register. I then went to the p5js reference and I found out that I needed to use "keycode" to use the arrow keys. Then, once The snake was moving, I coded for the collisions so that the food would move to a random part of the screen and the score would increase. Then I tried to make the snake grow. This was another problem. The snake body parts were appearing on the screen but not following the head. With collaboration with fellow students and the teacher, I found that the snake body could change its location to the location of the body part ahead of it in the list of snake segments. I then did several tests. When I was done, I added the restart feature and the high score feature. Then I made it add 3 segments after eating one apple.

2c.)

```
tangle(){//you lose the game if you
collide with your body parts
    if(this.body.length > 3){
      for(var i = 0; i < this.body.length;
i++){
          if (this.loc.x ===
this.body[i].x && this.loc.y ===
this.body[i].y){
            gameOver = 1;
          }
        }
      }
    }
```

This algorithm is important because it makes sure that the snake head is not colliding with the other segments of its body. If the head is touching the body, it tells the losing algorithm and the game ends. This creates the challenge of the game and it would not be fun without it.

2d.)

```
class Snake{
  constructor(x,y){
    this.loc = createVector(x,y);
    this.w = cubeWidth;
    this.clr = color(random(0,255), random(0,255), random(0,255));
    this.vel = createVector(0, 0);
    this.body = [];
  }
```

This abstraction allows the snake to have its function and display on the screen. This abstraction made the code less complex by compiling all of the information about the snake. Instead of running this code many times, I could call it once and it would display multiple.

```
//  Austin Matel
//   10/31/19
//  This is a comment
//  The setup function function is called once when your program
begins
var cubeWidth, food, direction, temp;
var highscore = 0;
var score = 0;
var columns;
var rows;
var snake;
var oneCube = 0;
var gameOver = 0;
var snakeNum = 1;
function setup() {
  var cnv = createCanvas(800, 800);
  cnv.position((windowWidth-width)/2, 30);
  background(5, 5, 5);
  fill(200, 30, 150);
  cubeWidth = 20;
  columns = width / cubeWidth;
  rows = height / cubeWidth;
  snake = new Snake(columns, rows);
  food = new Food(cubeWidth * int(random(0,800/cubeWidth)),cubeWidth *
int(random(0,800/cubeWidth)));
}
function runObjects(){//runs the snake and food
  snake.run();
  food.run();
}
function keyPressed(){//detects when the arrow keys are pressed
  if(keyCode === RIGHT_ARROW && direction !== 2){
    snake.vel.y = 0;
    snake.vel.x = cubeWidth;
    direction = 1;
  }
  if(keyCode === LEFT_ARROW && direction !== 1){
    snake.vel.y = 0;
    snake.vel.x = -cubeWidth;
    direction = 2;
  }
  if(keyCode === UP_ARROW && direction !== 4){
    snake.vel.x = 0;
```

```
    snake.vel.y = -cubeWidth;
    direction = 3;
  }
  if(keyCode === DOWN_ARROW && direction !== 3){
      snake.vel.x = 0;
      snake.vel.y = cubeWidth;
      direction = 4;
  }
}
//  The draw function is called @ 30 fps
function draw() {//pauses and ends game when snake dies
  if(gameOver === 0){
    background(5,5,5);
    runObjects();
    keyPressed();
    textSize(20);
    text("Score = "+score,10,20);
    frameRate(15);
    text("Highscore = "+highscore, 660, 20);
  }
  if(gameOver === 1){
    fill(255);
    textSize(90);
    text("Game Over!!!", 100, 100);
    textSize(40);
    text("Press Spacesbar", 220, 600);
    if(score > highscore){
      highscore = score;
    }
    if(keyCode === 32){
      gameOver = 0;
      snake = new Snake(columns, rows);
      food = new Food(cubeWidth *
int(random(0,800/cubeWidth)),cubeWidth *
int(random(0,800/cubeWidth)));
      snake.body = [];
    }
    score = 0;
  }
}
//Austin Matel
//10/31/19
class Snake{
```

```
  constructor(x,y){
    this.loc = createVector(x,y);
    this.w = cubeWidth;
    this.clr = color(random(0,255), random(0,255), random(0,255));
    this.vel = createVector(0, 0);
    this.body = [];
  }
  run(){
    this.render();
    this.loadSegments();
    this.update();
    this.checkEdges();
    this.tangle();
  }
  render(){//displays the head and segments of the snake
    //render head
    fill(this.clr);
    rect(this.loc.x, this.loc.y, this.w, this.w);
    //render body
    for(var i = 0; i < this.body.length; i++){
      rect(this.body[i].x, this.body[i].y, this.w, this.w);
    }
  }
  update(){//makes sure the head moves and the segments follow the
head
    for(var i = this.body.length - 1; i > 0; i--){
      this.body[i].y = this.body[i - 1].y;
      this.body[i].x = this.body[i - 1].x;
    }
    if(this.body.length > 0){
      this.body[0].x = this.loc.x;
      this.body[0].y = this.loc.y;
    }
    this.loc.add(this.vel);
  }
  loadSegments(){//fills the list of segments with vectors
    if(this.body.length / 3 < score){
      this.body.push(createVector(this.loc.x, this.loc.y));
      }
    }
  checkEdges(){//makes you lose when the snake head hits the edges
    if(this.loc.x > width - this.w || (this.loc.x < 0) || (this.loc.y
> height - this.w) || (this.loc.y < 0)){
```

```
      gameOver = 1;
    }
  }
  tangle(){
    if(this.body.length > 3){
      for(var i = 0; i < this.body.length; i++){
          if (this.loc.x === this.body[i].x && this.loc.y ===
this.body[i].y){
            gameOver = 1;
          }
        }
      }
    }
}
//Austin Matel
//10/31/19
class Food{
  constructor(x,y){
    this.loc = createVector(x,y);
    this.clr = color(255,0,0);
    this.w = cubeWidth;
  }
  run(){
    this.render();
    this.touchingSnake();
  }
  render(){//shows the red food
    fill(this.clr);
    rect(this.loc.x, this.loc.y, this.w, this.w);
  }
  touchingSnake(){
    if(snake.loc.x === this.loc.x && snake.loc.y ===
this.loc.y){//puts the food in a random place

      this.loc.x = cubeWidth * int(random(0,800/cubeWidth));
      this.loc.y = cubeWidth * int(random(0,800/cubeWidth));
      for(var i = 0; i < snake.body.length; i++){
        if(this.loc.x === snake.body[i].x){
          this.loc.x = cubeWidth * int(random(0,800/cubeWidth));
        }
        if(this.loc.y === snake.body[i].y){
          this.loc.y = cubeWidth * int(random(0,800/cubeWidth));
        }
```

```
      }
      score = score + 1;
    }
  }
}
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Snake Game</title>
    <script src="libraries/p5.js" type="text/javascript"></script>
    <script src="libraries/p5.dom.js" type="text/javascript"></script>
    <script src="libraries/p5.sound.js"
type="text/javascript"></script>
    <script src="sketch.js" type="text/javascript"></script>
    <script src="snake.js" type="text/javascript"></script>
    <script src="food.js" type="text/javascript"></script>
    <style> body {padding: 0; margin: 0;} canvas {vertical-align:
top;} </style>
  </head>


  <body>
  </body>
</html>
```