Appendix

HTML File 1

```
<html>
   <head>
      <title>Snake Math Game</title>
      <link rel = "stylesheet" href = "Snake.css">
      <style>
         canvas{
            display: block;
            margin: 0 auto;
         }
      </style>
   </head>
   <body>
      <canvas id="snake" style =
"position:absolute;top:32px;width:608px;height:608px;left:32px; border:2px  solid
black;"width ="608" height="608"></canvas>
      <canvas id="secondcanvas" style =
"position:absolute;top:120px;width:550px;height:490px;left:62px; border:2px  solid
black;"></canvas>
      <script src="Snakemathgame.js"></script>
      <button type="submit" style = "position: absolute;top:620px; width:100px;
height:px;left:530px; "onClick="refreshPage()">New Game</button>
      <script>
         function refreshPage(){
         window.location.reload();
         }
      </script>
   </body>
</html>
```

HTML File 2

```
<html>
   <head>
      <title>Snake Menu</title>
      <link rel = "stylesheet" href = "Snake.css">
   </head>
   <body>
      <h1>Snake Math Game</h1>
      <h2>Instruction</h2>
      <p>Instructions</p>
      <p>To play the game please use the arrow keys on your keyboard</p>
      <img src = "arrowkeys.png" alt="Image of Arrow Keys"
style="width:300px;height:300px;">
      <p>The snake math game educates young scholars by implementing math concepts
into a simple game</p>
```

```
        <p>Once you access the game the coordinates will appear</p>
        <p>Please pay close attention to the coordinates</p>
        <p>This is the fourth quater where x numbers are positive and y numbers are
negative</p>
        <p>If you feel ready to enjoy the magic, please click the button PLAY below</p>
        <p><a href="Snake.html">PLAY</a></p>
    </body>
</html>
```

Javascript file 1

```
var cvs = document.getElementById("snake");
var ctx = cvs.getContext("2d");

var cvs1 = document.getElementById("secondcanvas");
var ctx1 = cvs1.getContext("2d");
let eating = false;
cvs1.style.visibility = 'hidden'
// create the constant variable with 32 pixels cause one square a = 32 pixels
const sq = 32;
paused = false;
var idInterval;

//load images from the folder in the Sublime

const green = new Image();
green.src = "img/Full.png";

const taco = new Image();
taco.src = "img/taco.png";

//call draw function every 100m
var game = setInterval(draw,100);

//create the snake out of the squares with x postion of 9 and y postiton of10
//snake is a 2d array as it has x and y coordinates
//snake[0] is the head

var snake = [];
snake[0] = {
          x : 288,
          y : 320
}

let snack = sq

generateRandom();
```

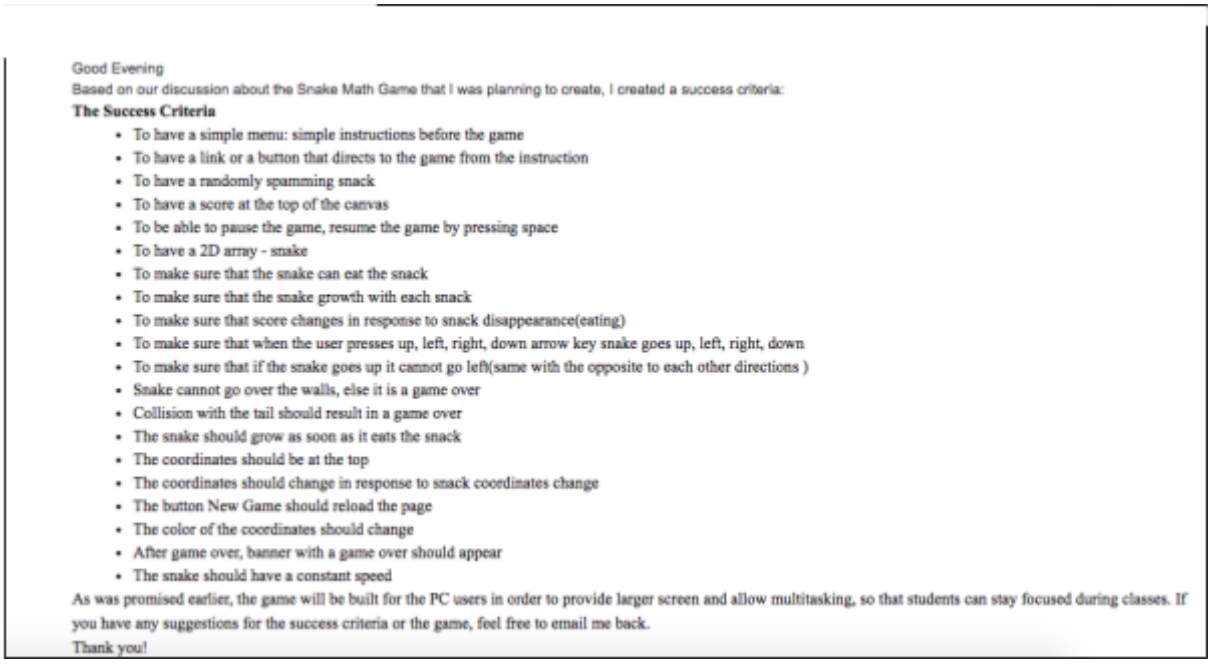
```
function hidecanvas(){
  if (cvs1.style.visibility === 'hidden') {
    cvs1.style.visibility = 'visible';
  }else{
    cvs1.style.visibility = 'hidden';
  }
}

function generateRandom(){
        snack = {
                x : Math.floor(Math.random()*17+1) * sq,
                y : Math.floor(Math.random()*15+3) * sq
        }
        console.log(snack);
}

print()
//create tGE score var

var score = 0

//pause function

//cordinates recording


//control the snake as it m-oves

var m;

document.addEventListener('keydown',m);

function direction(event){

}

//pause the snake game if the player presses space
function Pause(){
    if (!paused){
       paused = true;
       game = clearInterval(idInterval)
    }else if (paused){
       paused= false;
       idInterval = setInterval(game, 100)
    }
```

```
}
function print(){
	if(!eating){
	ctx.fillStyle = "RandomColor";
	ctx.font = "40px Comic Sans";
	ctx.fillText("x",7*sq,1*sq);
	ctx.fillText(snack.x/32,6.7*sq,2.4*sq);
	ctx.fillStyle = "RandomColor";
	ctx.font = "40px Comic Sans";
	ctx.fillText("yi",11*sq,1*sq);
	ctx.fillText(-snack.y/32,10.7*sq,2.4*sq);
	}
}

function RandomColor() {
  let letters = '0123456789ABCDEF';
  let color = '#';
  for (let c = 0; c < 6; c++) {
    color += letters[Math.floor(Math.random() * 16)];
  }
  return color;
}



function setRandomColor() {
  $("#colorpad").css("background-color", RandomColor());
}

window.addEventListener('keydown', function (e){
	var key = e.keyCode;
	if (key === 32){
    Pause();
	}

	console.log(m);
	if(key == 37 && m != "RIGHT"){
		m = "LEFT";
	}else if(key == 38 && m != "DOWN"){
		m = "UP";
	}else if(key == 39 && m != "LEFT"){
		m = "RIGHT";
	}else if(key == 40 && m != "UP"){
		m = "DOWN";
	}
	console.log(m);
```

```
});

//snakeX, snakeY, snakeX+sq, snakeY+sq, snack.x, snack.y, snack.x+sq, snack.y+sq
function overlap(lx1, ly1, rx1, ry1, lx2, ly2, rx2, ry2) {
/*	if (lx1 > rx2 || lx2 > rx1){
		return false
	}
	else if (ly1 > ry2 || ly2 > ry1){
		return false
	}else{
		return true
	}*/


	if(lx1==lx2 && ly1==ly2 && rx1==rx2 && ry1==ry2){
		return true;
	}else{
		return false;
	}
}


//check collision snake
function intersection(head, array){
	for(var k=1; k<array.length; k++){
		if(head.x === array[k].x && head.y ===array[k].y){
			return true;
		}else{
			return false;
		}
	}
}

//draw everything to the canvas
// draw is called everytime
//press space bar to resume the game after eating a snack
function draw(){

	ctx.drawImage(green,0,0);

	ctx1.fillStyle ="green";
	ctx1.fillRect(0, 0, 17*sq, 15*sq);
	ctx1.fillStyle = "black";
	ctx1.font = "40px Impact";
	ctx1.textAlign = "center";
	ctx1.fillText("Game Over", 4.5*sq, 2.5*sq);
```

```
	ctx1.fillStyle = "black";
	ctx1.font = "12px Comic Sans";
	ctx1.textAlign = "center";
	ctx1.fillText("Please click New Game to restart the Snake game", 4.5*sq, 3*sq);

	for(var i = 0; i < snake.length; i++){
		ctx.fillStyle = ( i==0 )? "coral":"black";//condition if  fillstyle is green and i == 0 and coral - the snake head is coral the rest is black
		ctx.fillRect(snake[i].x,snake[i].y,sq,sq);

		ctx.strokeStyle = "lightsalmon";
		ctx.strokeRect(snake[i].x,snake[i].y,sq,sq);

	}
	changeMode();
	ctx.drawImage(taco, snack.x, snack.y, sq, sq);

	//old head postition
	var newHeadX = snake[0].x;
	var newHeadY = snake[0].y;


	//which direction
	if(m == "LEFT") newHeadX -= sq;
	if(m == "UP") newHeadY -= sq;
	if(m == "RIGHT") newHeadX += sq;
	if(m == "DOWN") newHeadY += sq;

	//Lunch timemeemememem
	if(overlap(newHeadX, newHeadY, newHeadX+sq, newHeadY+sq, snack.x, snack.y, snack.x+sq, snack.y+sq)){
		eating = true;
	}
	else {
		eating = false;
		print();
	}
	if(eating){
		score++
		/*ctx.fillStyle ="white";
		ctx.font = "40px Comic Sans"
		ctx.fillText(coordinates, 6*sq,1.7*sq);*/
		//create a function that switches between the canvases and point out hte coordinates of the snake

		generateRandom();
```

```
	}else{
		//remove the tail as it is moving
		snake.pop();
	}

	function changeMode(){
		if (score%10 == 0 && score != 0){

			ctx.fillStyle =RandomColor();
			ctx.fillRect(sq, 3*sq, 17*sq, 15*sq);

			for(var i = 0; i < snake.length; i++){
			ctx.fillStyle = ( i==0 )? "lightpink":"aliceblue";//condition if  fillstyle is
green and i == 0 and coral - the snake head is coral the rest is black
			ctx.fillRect(snake[i].x,snake[i].y,sq,sq);

			ctx.strokeStyle = "mediumpurple";
			ctx.strokeRect(snake[i].x,snake[i].y,sq,sq);
			}
		}
	}
	//add new Head

	let newHead = {
		x : newHeadX,
		y : newHeadY
	}

	//gameover
	if((newHeadX < (0.8*sq)) || (newHeadX > (17 * sq)) || (newHeadY < (3 * sq)) ||
(newHeadY > (17 *sq)||intersection(newHead,snake))){
		clearInterval(game);
		hidecanvas()
		console.log("clearInterval");

	}


	snake.unshift(newHead);

	ctx.fillStyle = "white";
	ctx.font = "40px Comic Sans";
	ctx.fillText(score,3.7*sq,1.7*sq);
}
```



Good Evening
Based on our discussion about the Snake Math Game that I was planning to create, I created a success criteria:

The Success Criteria

- To have a simple menu: simple instructions before the game
- To have a link or a button that directs to the game from the instruction
- To have a randomly spamming snack
- To have a score at the top of the canvas
- To be able to pause the game, resume the game by pressing space
- To have a 2D array - snake
- To make sure that the snake can eat the snack
- To make sure that the snake growth with each snack
- To make sure that score changes in response to snack disappearance(eating)
- To make sure that when the user presses up, left, right, down arrow key snake goes up, left, right, down
- To make sure that if the snake goes up it cannot go left(same with the opposite to each other directions)
- Snake cannot go over the walls, else it is a game over
- Collision with the tail should result in a game over
- The snake should grow as soon as it eats the snack
- The coordinates should be at the top
- The coordinates should change in response to snack coordinates change
- The button New Game should reload the page
- The color of the coordinates should change
- After game over, banner with a game over should appear
- The snake should have a constant speed

As was promised earlier, the game will be built for the PC users in order to provide larger screen and allow multitasking, so that students can stay focused during classes. If you have any suggestions for the success criteria or the game, feel free to email me back.
Thank you!

Helic

It sounds good to me. I have a couple ideas towards your future game:
- the ability to change the color of the canvas and the snake when the score is 10
-the ability to make the snake transparent so that it could go through itself. Although it goes against the classical version of the snake I believe that this will allow the game to become easier for students so that they could play longer. In the end, students will face and remember more coordinates.

The IA discussion with my client that was executed when I showed and gave my final product to the client:
(Greetings, formalities...)
Me: Can we please discuss each aspect of the educational game separately. Let's start with the instruction page: what do you think about the page?
The client: I think that the page is well structured and is easy to understand. The instructions are simple and the simple idea behind IA is introduced. Moreover, it is easy to go the actual game: in one click on the PLAY button, I can start playing the game.
Me: What can you say about the movement of the snake?
The client: The snake moves properly - without any delays or bugs. For the snake design, however, I would recommend creating a snake out of the squares that you have.
Me: Could you please describe if the snack spams correctly: randomly with no delays or bugs?
The client: Firstly, I thought that there is a special pattern made to spawn the snack, however when playing the game several times I understood that the position of the snack is random and programmed in a way as it should be.
Me: Do you have any concerns or further improvements for the snake part of the game?
The client: The snake game works properly and I am satisfied with it. I would suggest earlier mentioned design and I would want to make an ability for students to play the game with each other. It will create more interest in the game.
Me: Yes, thanks I also thought about doing that. Were there any problems of a snake eating the snack - maybe the score count didn't update or snake went through the snack.

The client: There were no major concerns about the snake itself, however, I would mention that sometimes snacks would spawn on the snake itself and I guess in the further coding process you should consider it not spamming on the snake itself.
Me: Now, regarding the pause of the game.
The client: The pause is one of the flaws that your code had. Although, the pause was not the major function it is nice to have to allow usability to the students.
Me: What about displaying the coordinates?
The client: The coordinate display worked well, however, I would suggest making the game more educational. Ad couple modes with different math topic focus, so that the program can cover most of the basic topics in math.
Me: Could you please talk about the game over the condition and the banner?
The client: Well, I would not call the banner basic, but I would like more color to it as you did with the snake and canvas when score heats 10. I can not really comment on the conditions of the game over. I was satisfied that you have considered making the snake transparent so that the players could go through the snake and have more chances of staying alive and thus scoring.
Me: Yes, thank you. The design question was supposed to be next, but you have already answered it.
The client: Overall, I really liked the program, but if you can consider the comments and further suggestions that I gave your IA can become very successful in the math field. Also, think about the google extension so that students could do better at multitasking. I would also suggest doing the level mode, where the snake game can be divided into several modes.
Me: Would you suggest anything else?
The client: I would also add some sound effects, for instance, when a snake eats the snack, that would act an extra effect on the game and drag more people to play it. These are the extra features that I want you to consider for further investigation. I liked the game and I find it very helpful when educating my students. Thanks for working on it for me. Great job!
Me: Thank you for giving me a feedback! It was a pleasure working on this project!
(goodbye formalities)

Bibliography:

Tsolova, Iana. “How Do I Pause Execution in JavaScript?” *Telerik Blogs*, Telerik, 10 Oct. 2019, https://www.telerik.com/blogs/how-do-i-pause-execution-in-javascript.

“Style Visibility Property.” *HTML DOM Style Visibility Property*, https://www.w3schools.com/jsref/prop_style_visibility.asp.

“Math.random().” *MDN Web Docs*, https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random.

“Find If Two Rectangles Overlap.” *GeeksforGeeks*, https://www.geeksforgeeks.org/find-two-rectangles-overlap/.

KeyboardEvent Key Property, https://www.w3schools.com/jsref/event_key_key.asp.

JavaScript - Move Object with Arrow Keys Using JavaScript Function, https://www.includehelp.com/code-snippets/move-object-with-arrow-keys-using-javascript-function.aspx.