Ammar Almeher
GitHub: akalmeher
ID: 921564498
CSC317 Web Development

Individual Assignment 3 – JavaScript Game

Description:

This assignment is to create a game using JavaScript for a web browser without frameworks or stacks.

Approach:

I decided to make an endless runner game, like the dinosaur game on the Chrome browser when you have no internet. The game seemed simple enough for me to add my own twist, and manipulate the mechanics to my liking. At first, I watched some videos on YouTube, to draw inspiration from (Making a Game in JavaScript with No Experience, JavaScript Game Development Course for Beginners, and Endless Runner Game in HTML5 and JavaScript with Source Code). I drew up a sketch of how I wanted it to look like at the end. I started with the HTML and CSS code and set up the site in which the game canvas would be centered. I asked around for ideas and I ended up with a cute "cupcake" runner game design and started working on the character designs. I drew a cupcake using a free online pixel art creator: PiskelApp. For the background, I used DALL-E to create the perfect looping background that goes with my idea of the game.

The player would stay in a fixed position horizontally, only being able to jump. The background would move from right to left, for which I had to make a looping mechanism.
I drew up two different frames for the player, that would switch while in the air.

Issues and Resolutions:

When trying to figure out how to make the game more interesting, I thought of adding platforms where the player could land on, and add cherries and cookies that the player could collect to get more points (like coins in other platform games). The code was getting too complicated, so I decided against these ideas since there was not much time left for submission.

I proceeded to make the jumps limited to two, instead of limitless. This gave the player more restrictions in terms of movement and made the game a bit more challenging.

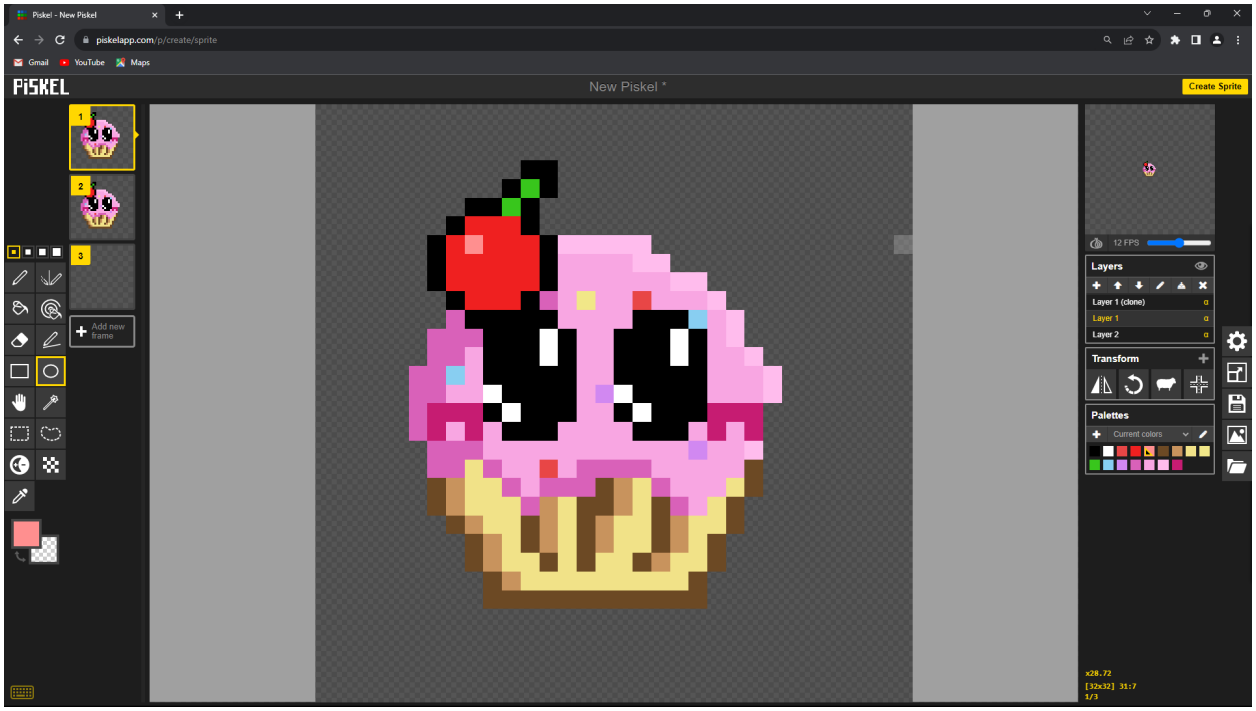
Ammar Almeher
GitHub: akalmeher

ID: 921564498
CSC317 Web Development

Analysis:

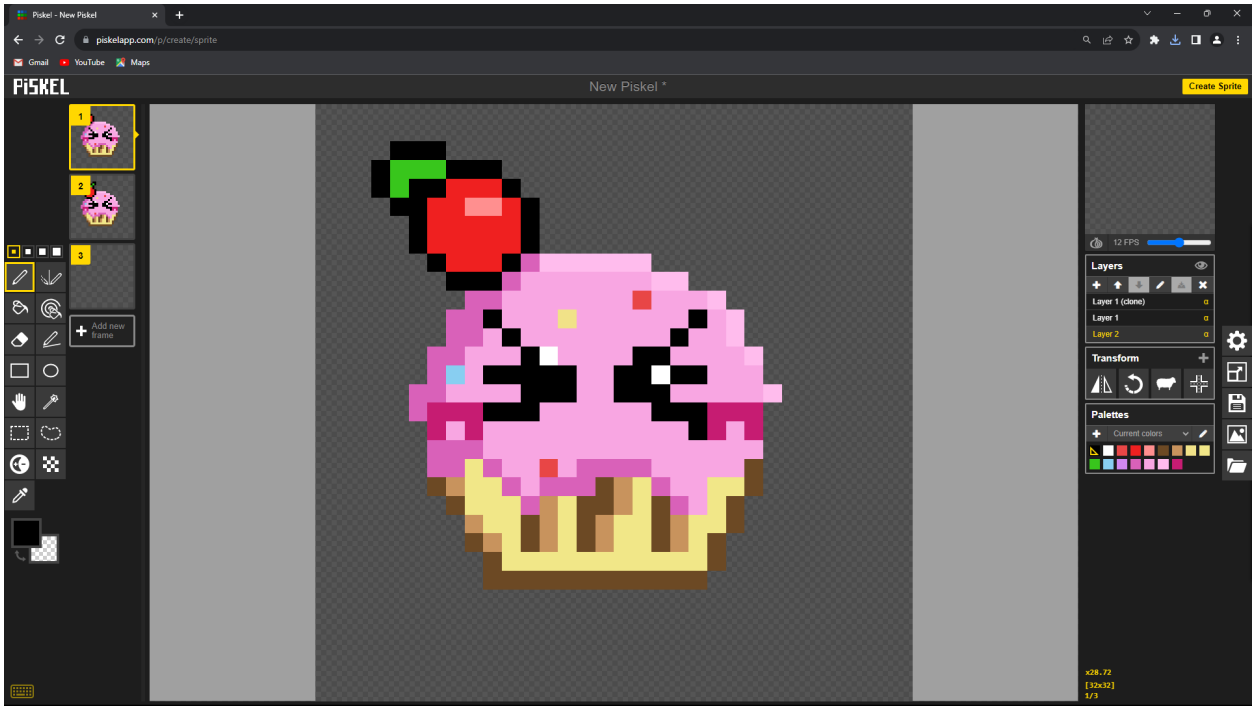
It takes a lot of time and patience for this kind of project, as I have come to realize. There is a lot of trial and error since the users aren't just reading from pages, they are INTERACTING with the elements/objects. After building the skeletal code, I needed to keep both windows for the code and the page open, so I could go back and forth to see what improvements I could make. I loved drawing the objects, even though I could not use most of them. These drawings made the game unique to my style, and I am very happy with the result. I am planning to keep building on this project after submission as well!

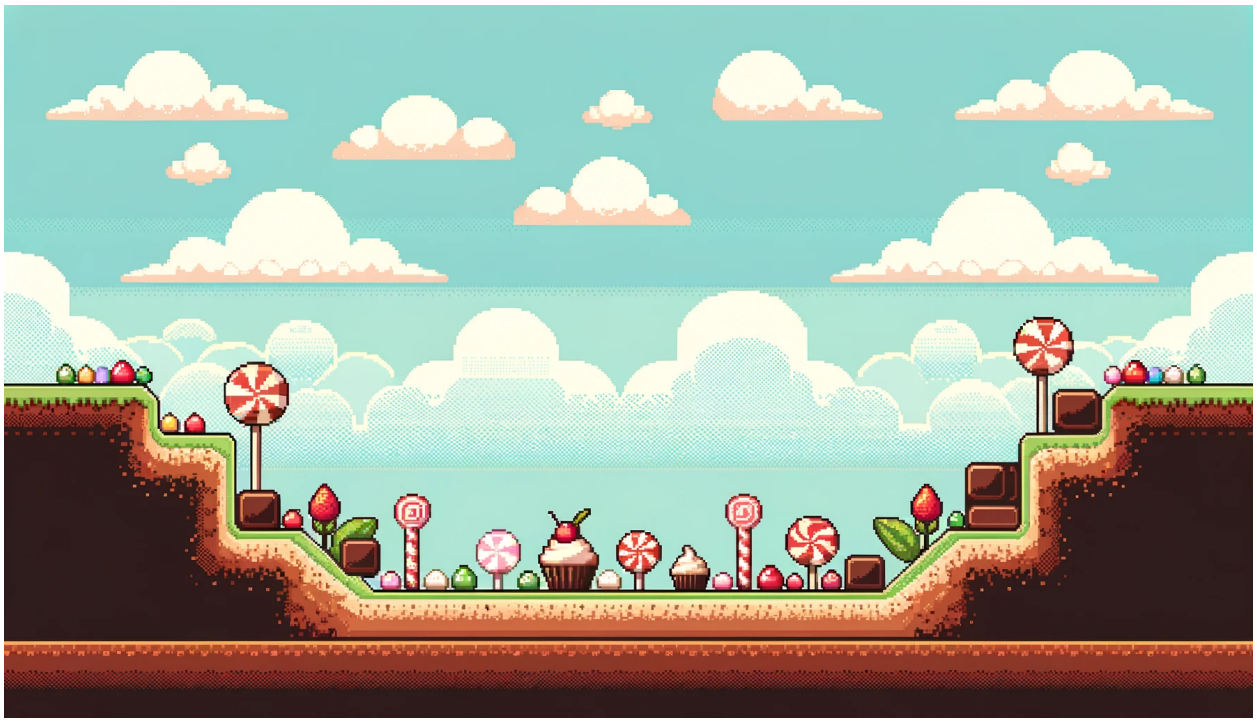
Screenshots:



Ammar Almeher
GitHub: akalmeher

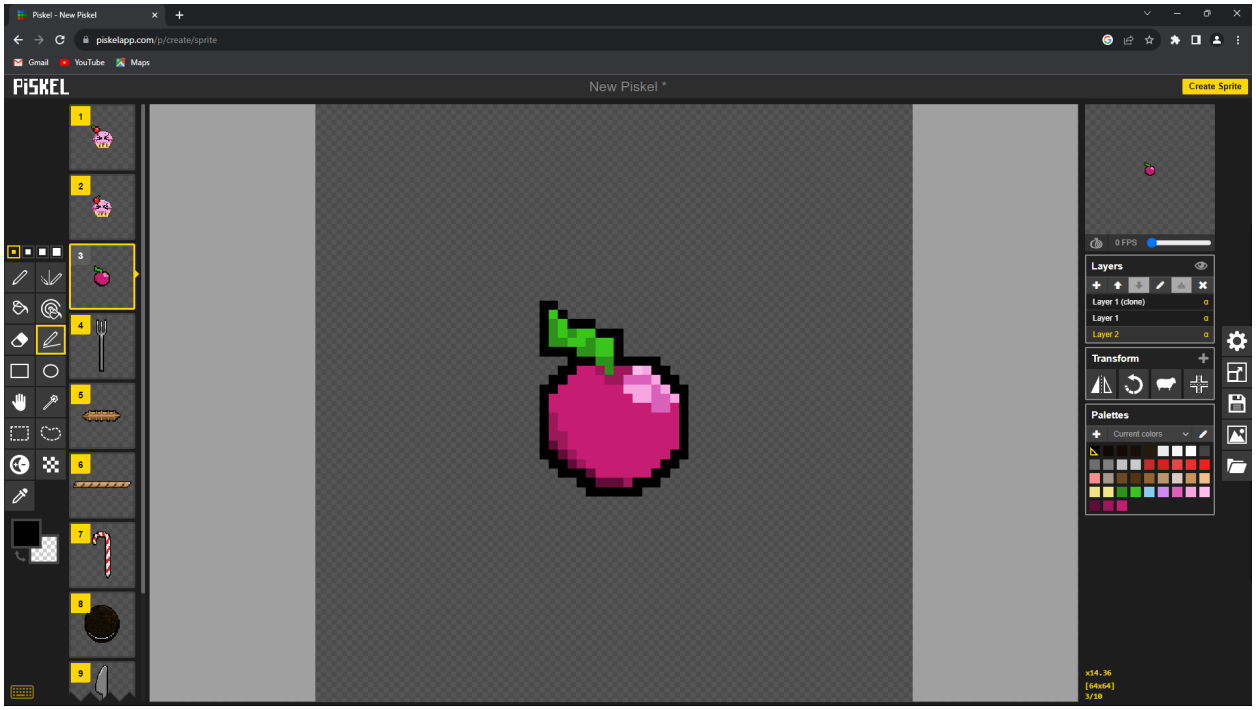
ID: 921564498
CSC317 Web Development

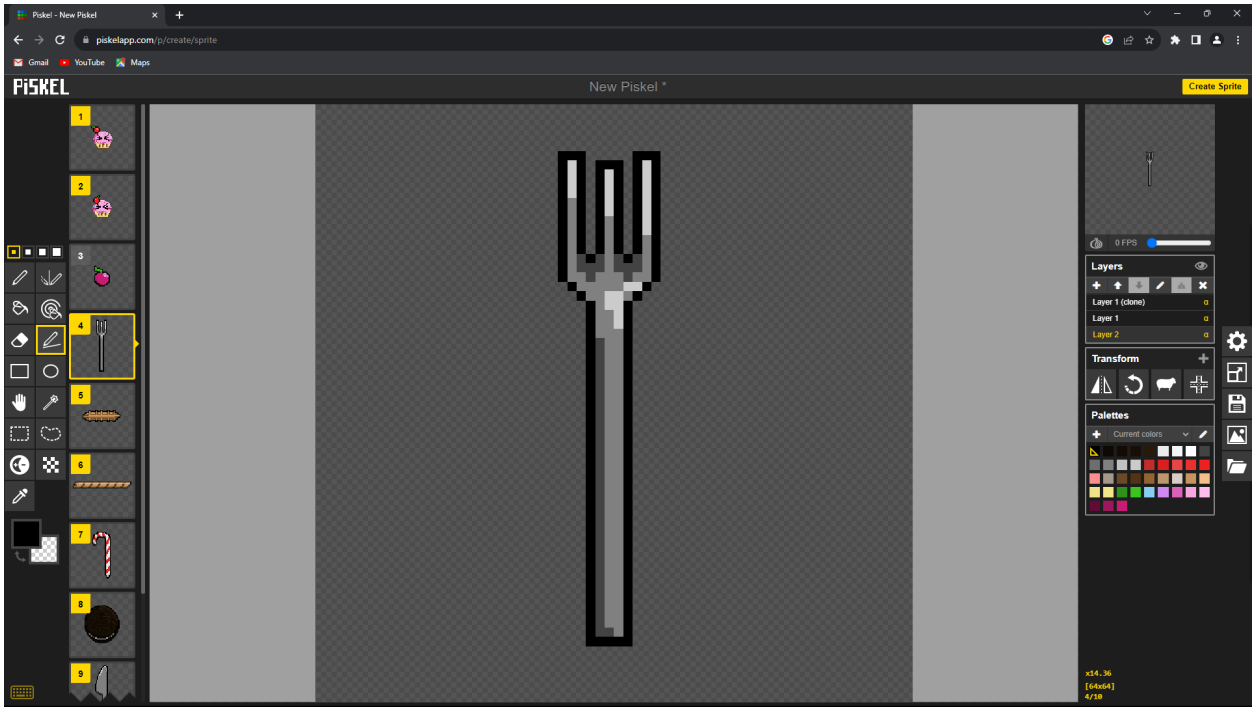




Ammar Almeher
GitHub: akalmeher

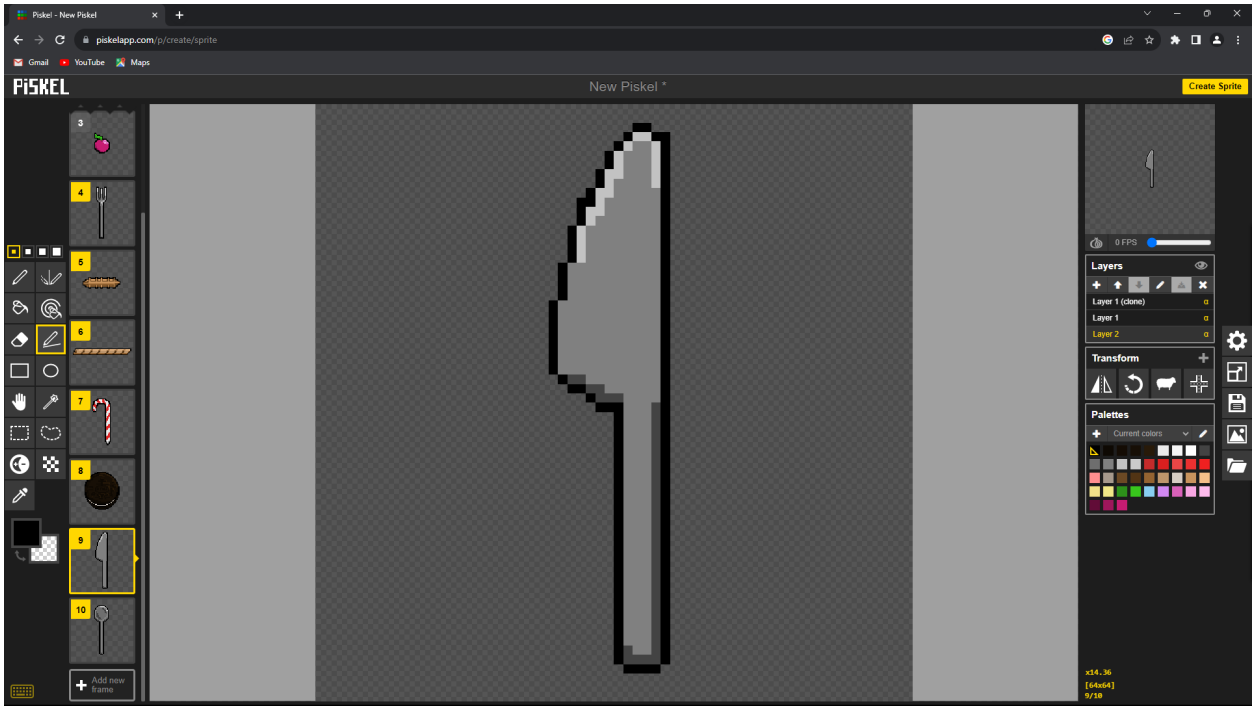
ID: 921564498
CSC317 Web Development

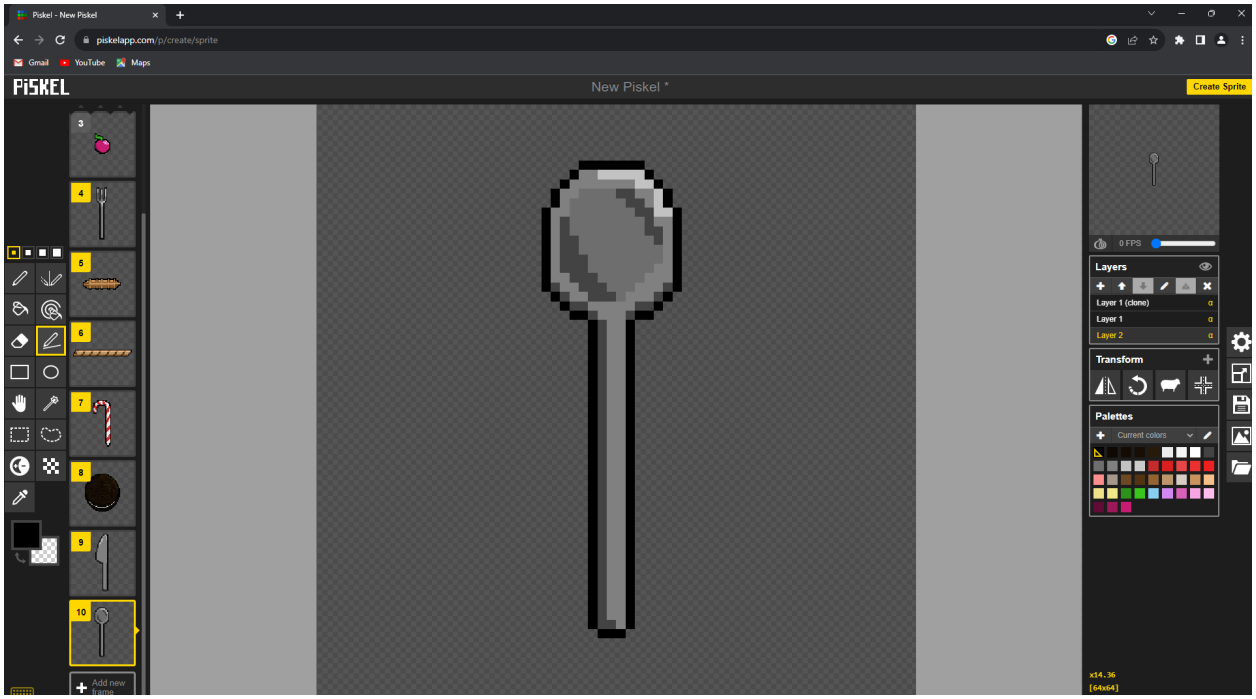




Ammar Almeher
GitHub: akalmeher

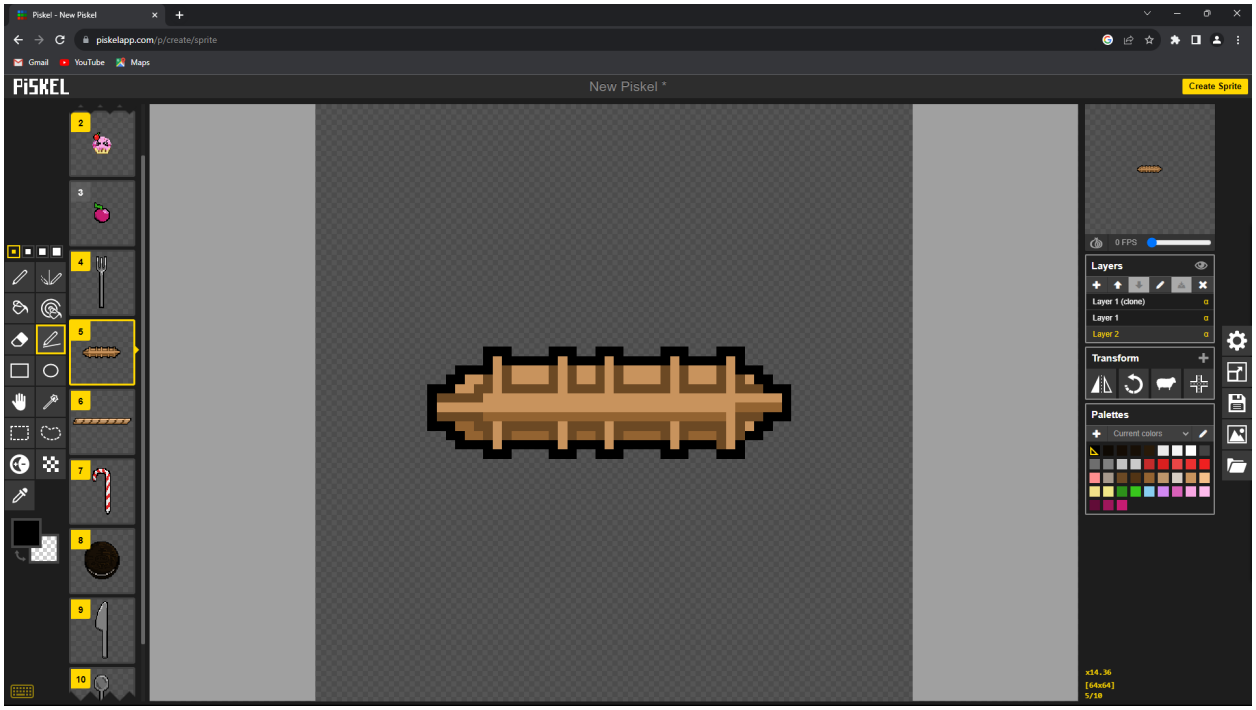
ID: 921564498
CSC317 Web Development

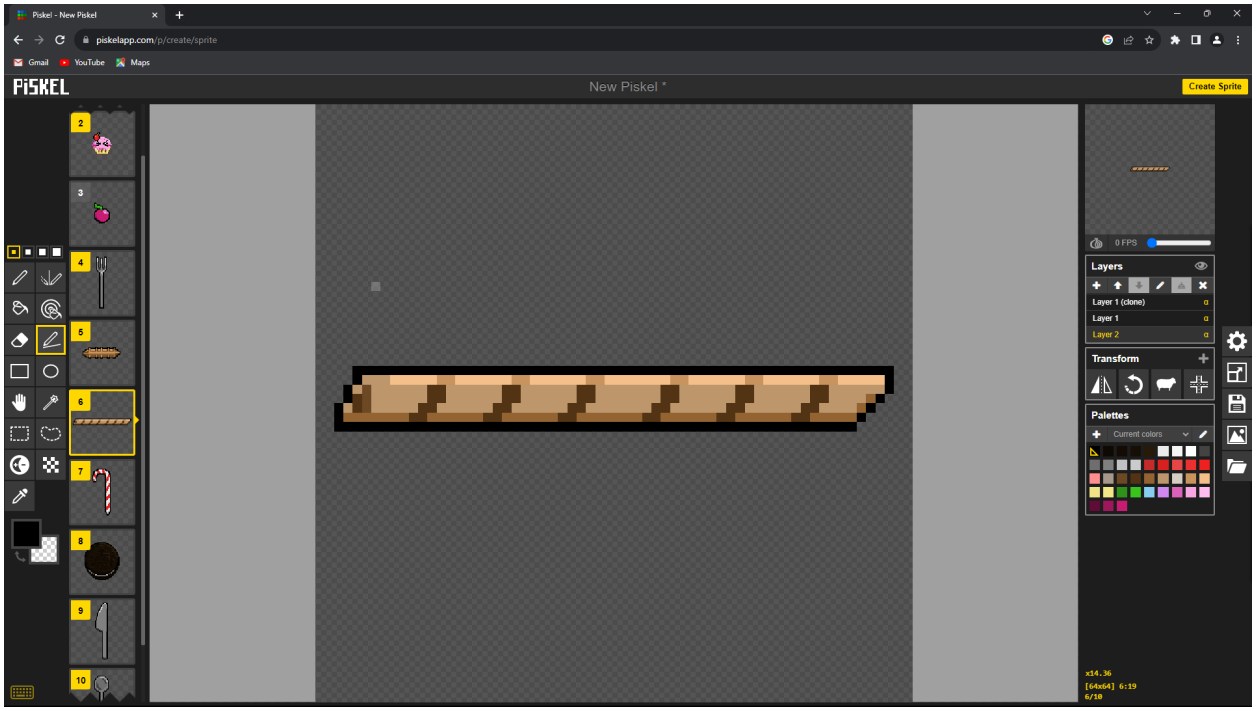




Ammar Almeher
GitHub: akalmeher

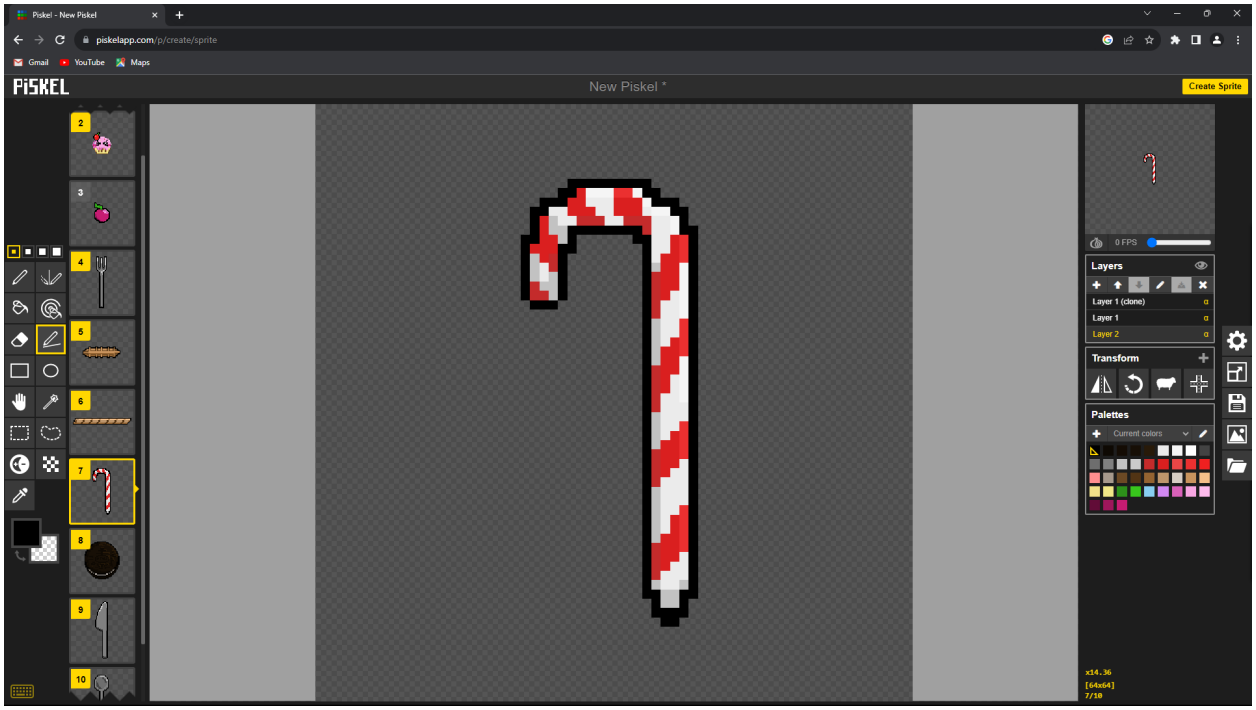
ID: 921564498
CSC317 Web Development

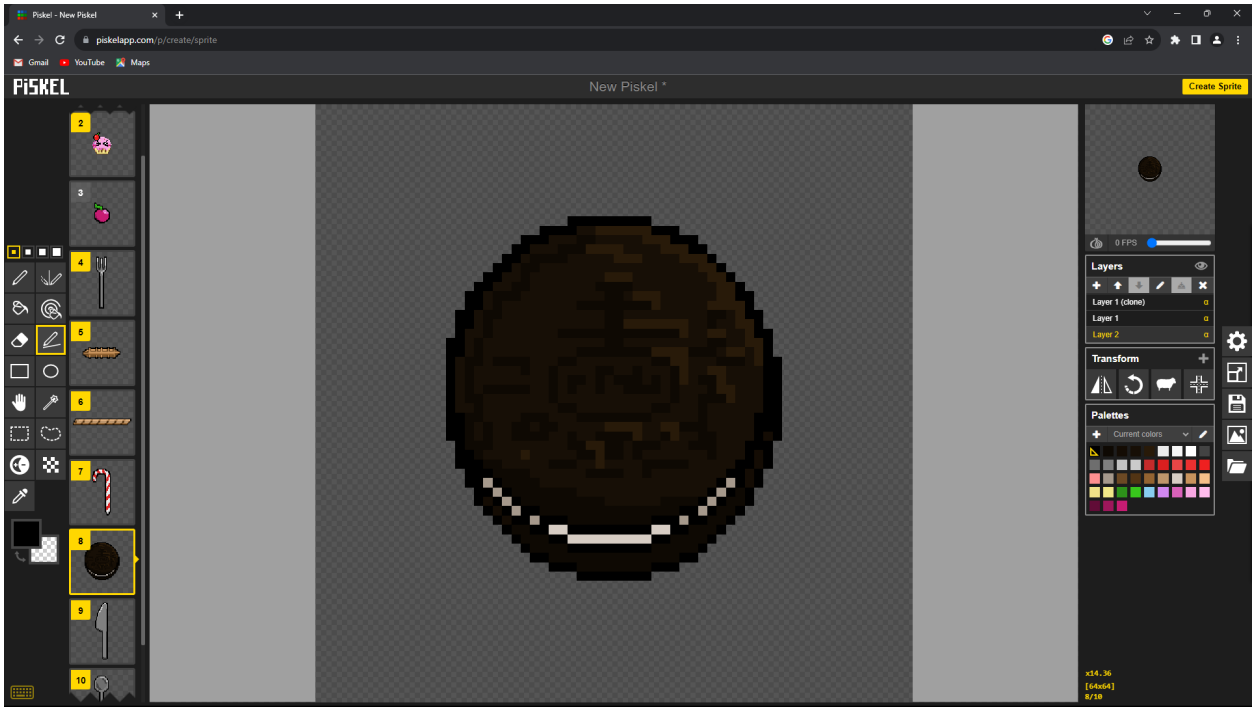




Ammar Almeher
GitHub: akalmeher

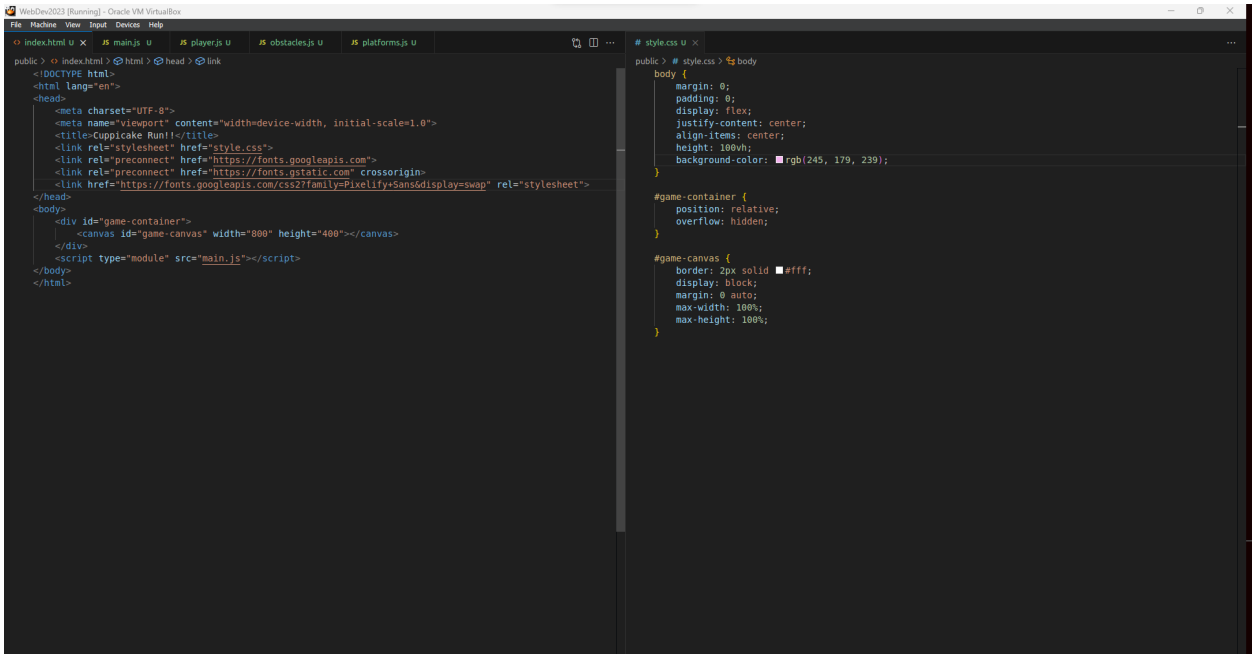
ID: 921564498
CSC317 Web Development





Ammar Almeher
GitHub: akalmeher

ID: 921564498
CSC317 Web Development

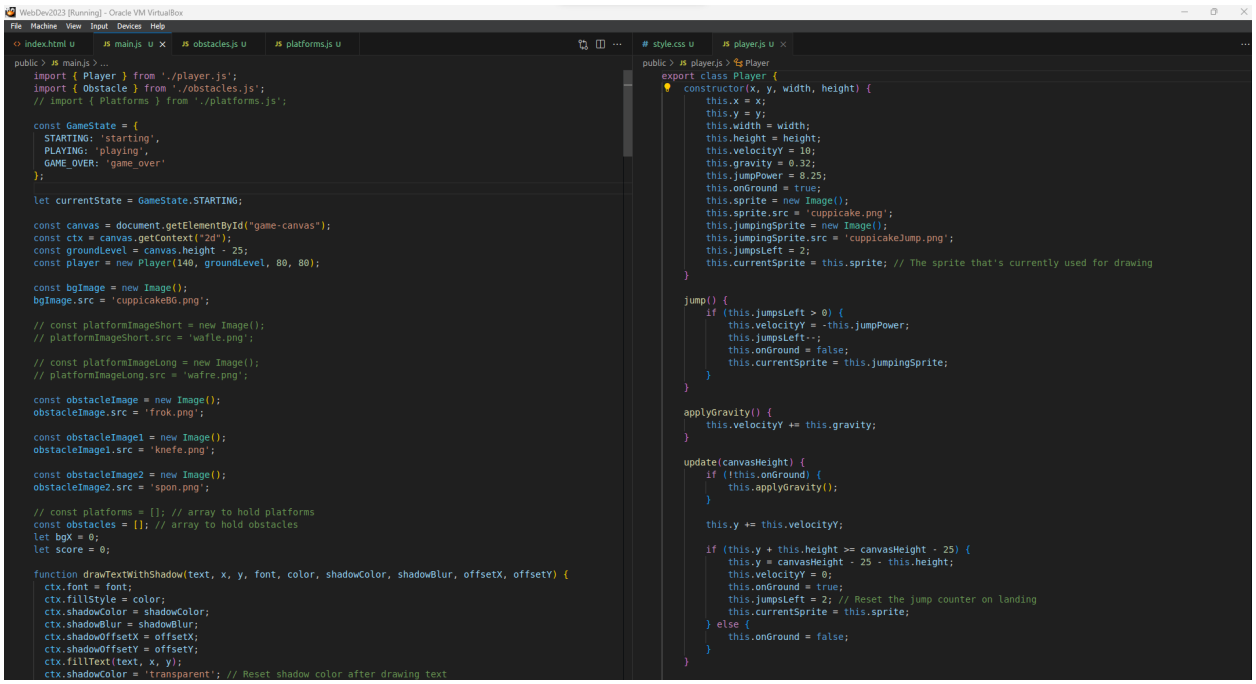


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cuppicake Run!!</title>
    <link rel="stylesheet" href="style.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Pixelify+Sans&display=swap" rel="stylesheet">
</head>
<body>
    <div id="game-container">
        <canvas id="game-canvas" width="800" height="400"></canvas>
    </div>
    <script type="module" src="main.js"></script>
</body>
</html>
```

```
body {
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background-color: rgb(245, 179, 239);
}

#game-container {
    position: relative;
    overflow: hidden;
}

#game-canvas {
    border: 2px solid #fff;
    display: block;
    margin: 0 auto;
    max-width: 100%;
    max-height: 100%;
}
```



```
import { Player } from './player.js';
import { Obstacle } from './obstacles.js';
// import { Platforms } from './platforms.js';

const GameState = {
  STARTING: 'starting',
  PLAYING: 'playing',
  GAME_OVER: 'game_over'
};

let currentState = GameState.STARTING;

const canvas = document.getElementById("game-canvas");
const ctx = canvas.getContext("2d");
const groundLevel = canvas.height - 25;
const player = new Player(140, groundLevel, 80, 80);

const bgImage = new Image();
bgImage.src = 'cuppicakeBG.png';

// const platformImageShort = new Image();
// platformImageShort.src = 'wafle.png';

// const platformImageLong = new Image();
// platformImageLong.src = 'wafre.png';

const obstacleImage = new Image();
obstacleImage.src = 'frok.png';

const obstacleImage1 = new Image();
obstacleImage1.src = 'knefe.png';

const obstacleImage2 = new Image();
obstacleImage2.src = 'spon.png';

// const platforms = []; // array to hold platforms
const obstacles = []; // array to hold obstacles
let bgX = 0;
let score = 0;

function drawTextWithShadow(text, x, y, font, color, shadowColor, shadowBlur, offsetX, offsetY) {
  ctx.font = font;
  ctx.fillStyle = color;
  ctx.shadowColor = shadowColor;
  ctx.shadowBlur = shadowBlur;
  ctx.shadowOffsetX = offsetX;
  ctx.shadowOffsetY = offsetY;
  ctx.fillText(text, x, y);
  ctx.shadowColor = 'transparent'; // Reset shadow color after drawing text
```

```
export class Player {
    constructor(x, y, width, height) {
        this.x = x;
        this.y = y;
        this.width = width;
        this.height = height;
        this.velocityY = 10;
        this.gravity = 0.32;
        this.jumpPower = 8.25;
        this.onGround = true;
        this.sprite = new Image();
        this.sprite.src = 'cuppicake.png';
        this.jumpingSprite = new Image();
        this.jumpingSprite.src = 'cuppicakeJump.png';
        this.jumpsLeft = 2;
        this.currentSprite = this.sprite; // The sprite that's currently used for drawing
    }

    jump() {
        if (this.jumpsLeft > 0) {
            this.velocityY = -this.jumpPower;
            this.jumpsLeft--;
            this.onGround = false;
            this.currentSprite = this.jumpingSprite;
        }
    }

    applyGravity() {
        this.velocityY += this.gravity;
    }

    update(canvasHeight) {
        if (!this.onGround) {
            this.applyGravity();
        }

        this.y += this.velocityY;

        if (this.y + this.height >= canvasHeight - 25) {
            this.y = canvasHeight - 25 - this.height;
            this.velocityY = 0;
            this.onGround = true;
            this.jumpsLeft = 2; // Reset the jump counter on landing
            this.currentSprite = this.sprite;
        } else {
            this.onGround = false;
        }
    }
```

Ammar Almeher
GitHub: akalmeher

ID: 921564498
CSC317 Web Development