Po-Han Chen
Github: eric915c

ID: 923446482
CSC317 Web Development

Assignment 3 - Javascript Individual Game Assignment

Description:
This assignment involves creating a memory card game using JavaScript and HTML Canvas. The game consists of 8 cards, with pairs of matching numbers (1, 2, 3, 4) represented by different colors. Players must flip over cards to find matching pairs, and upon successfully matching all pairs, they win the game. Using the basic code of JavaScript and Canvas API to complete.
Approach / What I Did:
Identified key tasks such as creating a memory card game, using HTML Canvas, implementing card flipping logic, and providing win condition feedback.

Planning create HTML structure with Canvas element and message container. And write the Javascript functions for card shuffling, drawing cards on the canvas, handling card clicks, checking for matches, and displaying win message. Then implement card flipping logic. Add the function that can reset the game.

Separating tasks were created for each JavaScript function needed to handle game logic.

Testing different situations, such as flipping cards, matching pairs, and winning the game.
Issues and Resolutions:

My first issue was the cards were not being shuffled properly, leading to the predicted cards can't show up. And I asked ChatGPT to give me some ideas and I borrowed the idea from ChatGPT . I already noted in my code that this function uses ChatGPT.

javascript Copy code

```javascript
function shuffle(array) {
    return array.sort(() => Math.random() - 0.5);
}
```

This version of the shuffle function utilizes the `sort` method to sort the array. The sorting criterion is based on a random number subtracted by 0.5. Since the `sort` method is stable, the relative order of elements will not change when the comparison function returns the same value. Therefore, by using the result of `Math.random() - 0.5` as the comparison function, the order of array elements will be randomized, achieving the shuffling effect.

I resolved it by checking the shuffle function, I found The previous approach was not sufficiently random, leading to predictable shuffling. To solve this problem, I use the `Array.sort()` method along with a custom comparison function. Through sorting the array with a comparison

function that generates random values between -0.5 and 0.5, the order of elements in the array is effectively randomized.

The second issue is that I had trouble with the handle click part, so I asked ChatGPT. I've noted in my code that I borrowed this idea.

javascript Copy code

```javascript
function handleClick(event) {
    const rect = mycanvas.getBoundingClientRect();
    const mouseX = event.clientX - rect.left;
    const mouseY = event.clientY - rect.top;

    const c = Math.floor(mouseX / (cardWidth + padding));
    const r = Math.floor(mouseY / (cardHeight + padding));
    const index = r * 4 + c;

    if (!flippedCards.includes(index)) {
        flippedCards.push(index);
        drawCard(c * (cardWidth + padding) + padding, r * (cardHeight + padding) + pa
        if (flippedCards.length === 2) {
            setTimeout(() => {
                const [card1, card2] = flippedCards;
                const id1 = cards[card1].id;
                const id2 = cards[card2].id;
                if (pairsToMatch.some(pair => pair[0] === id1 && pair[1] === id2)) {
                    matchedPairs++;
                    checkWin();
                } else {
                    drawCard(card1 % 4 * (cardWidth + padding) + padding, Math.floor(
                    drawCard(card2 % 4 * (cardWidth + padding) + padding, Math.floor(
```

```
    if (!flippedCards.includes(index)) {
        flippedCards.push(index);
        drawCard(c * (cardWidth + padding) + padding, r * (cardHeight + padding) + pa
        if (flippedCards.length === 2) {
            setTimeout(() => {
                const [card1, card2] = flippedCards;
                const id1 = cards[card1].id;
                const id2 = cards[card2].id;
                if (pairsToMatch.some(pair => pair[0] === id1 && pair[1] === id2)) {
                    matchedPairs++;
                    checkWin();
                } else {
                    drawCard(card1 % 4 * (cardWidth + padding) + padding, Math.floor(
                    drawCard(card2 % 4 * (cardWidth + padding) + padding, Math.floor(
                }
                flippedCards = [];
            }, 1000);
        }
    }
}
```

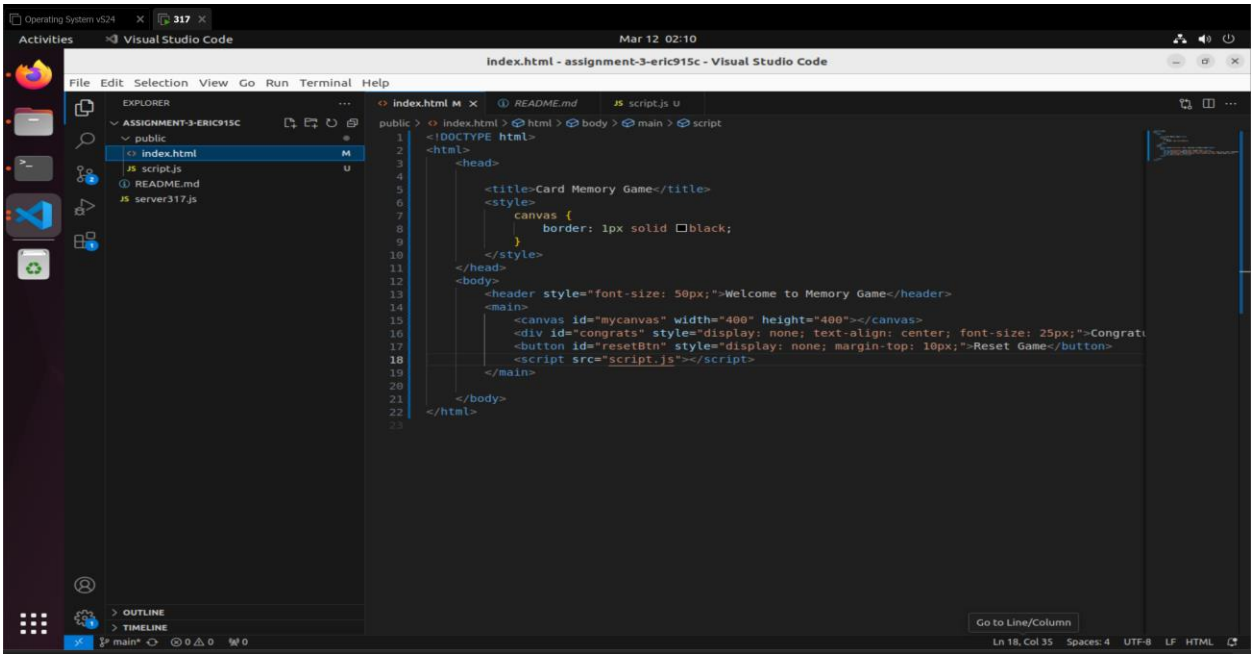
I resolved by following the idea from ChatGPT, it gave me a handleClick function. And to make the game smoother, I changed the delay time from 1000 milliseconds to 100 milliseconds. After testing, I think this will make the game smoother for players.

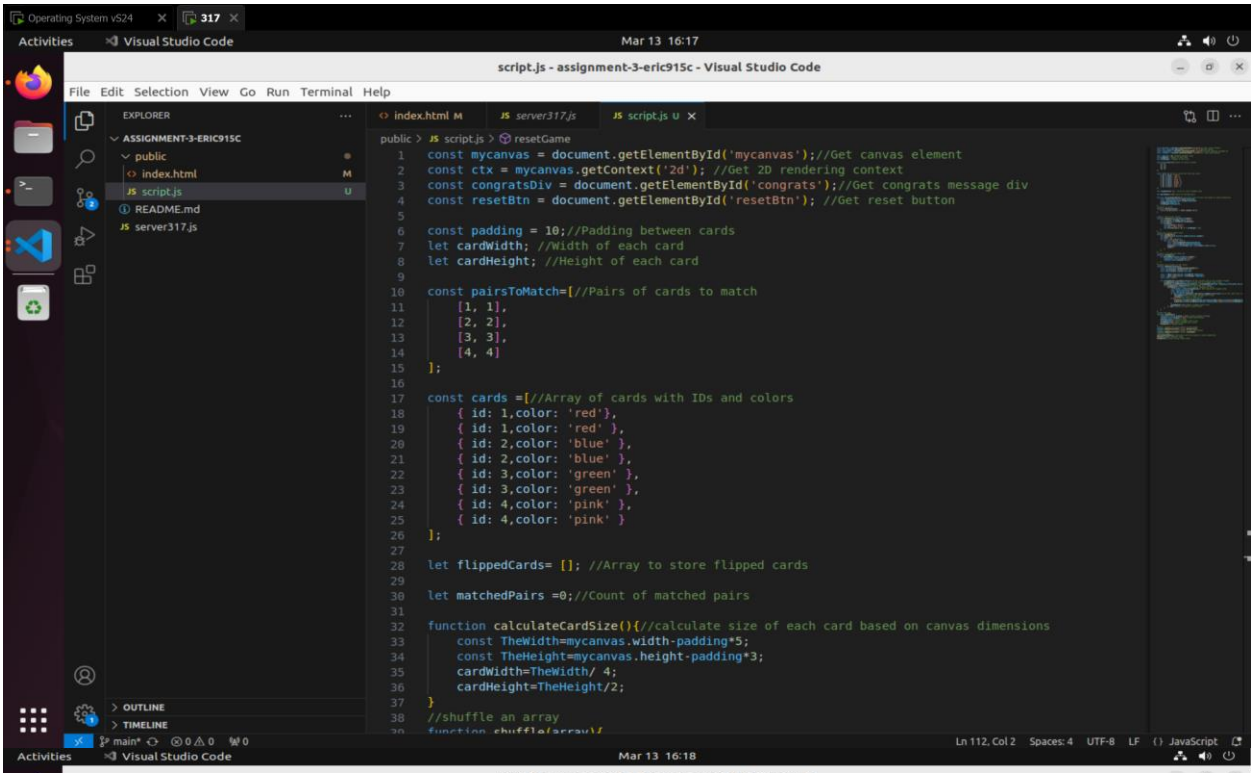
Analysis: (Reflect on what you have learned and how decisions made could impact users.)

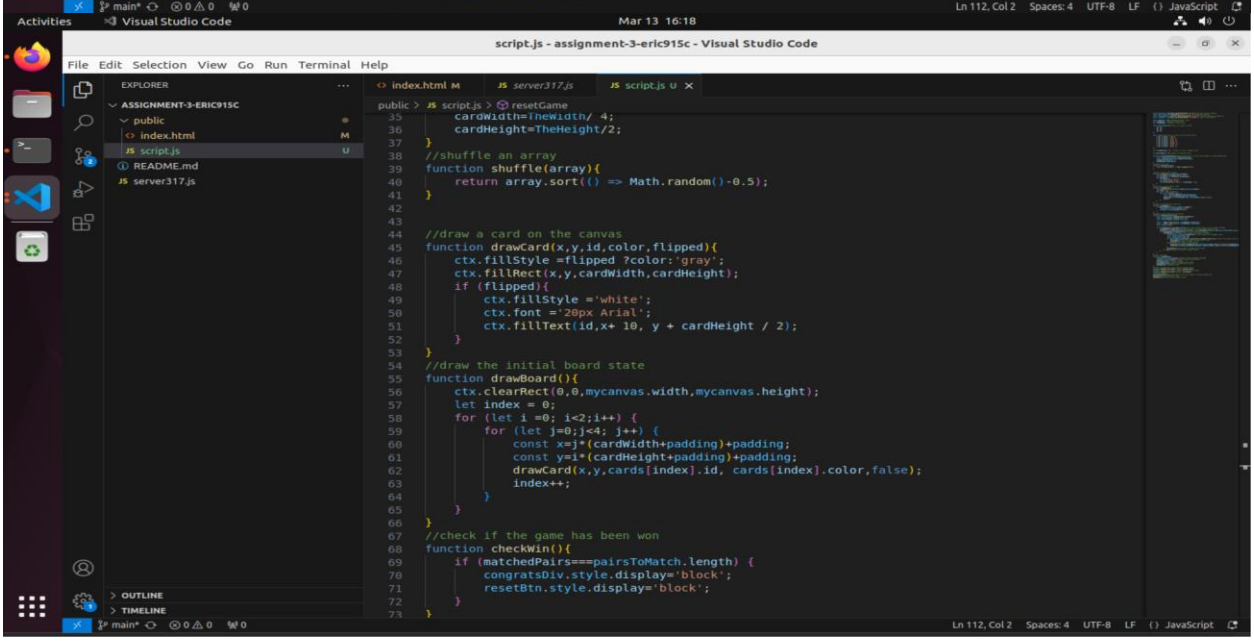
This assignment helped me deepen my understanding of game logic implementation, particularly in a web-based environment. I learned how to structure code to handle game mechanics such as card flipping, matching, and win conditions.

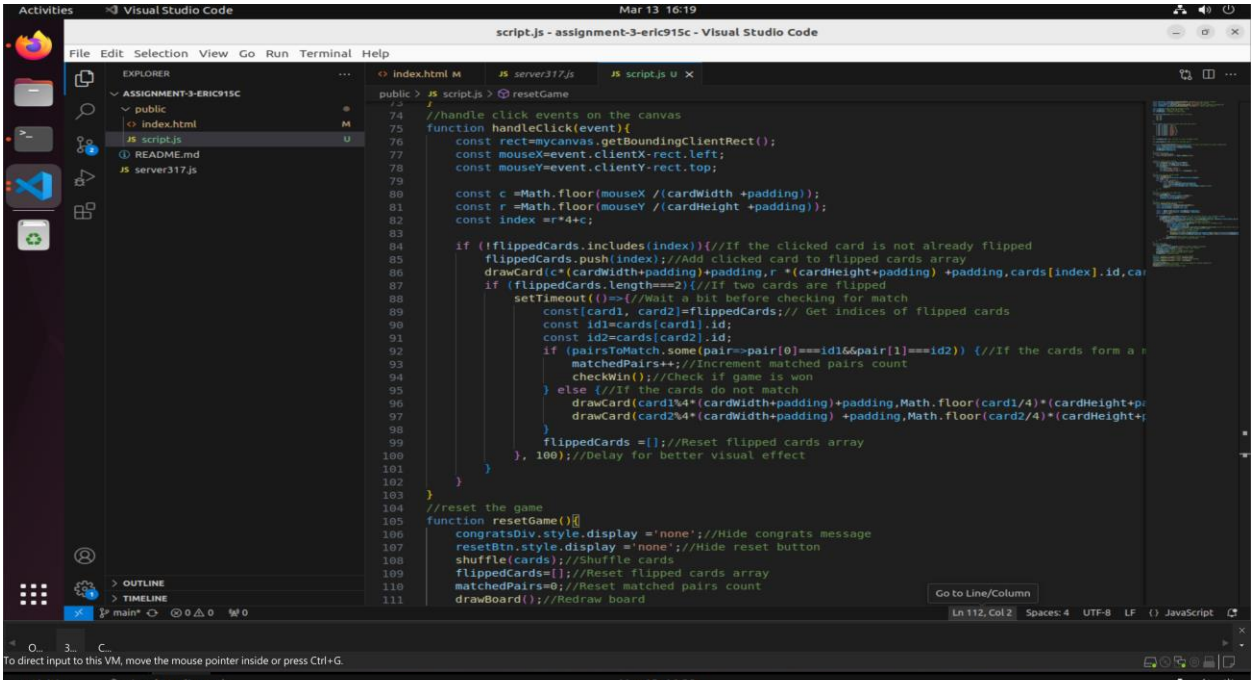
I think that an effective shuffling algorithm enhances replayability and excitement, and the "Reset Game" button improve usability and player engagement.

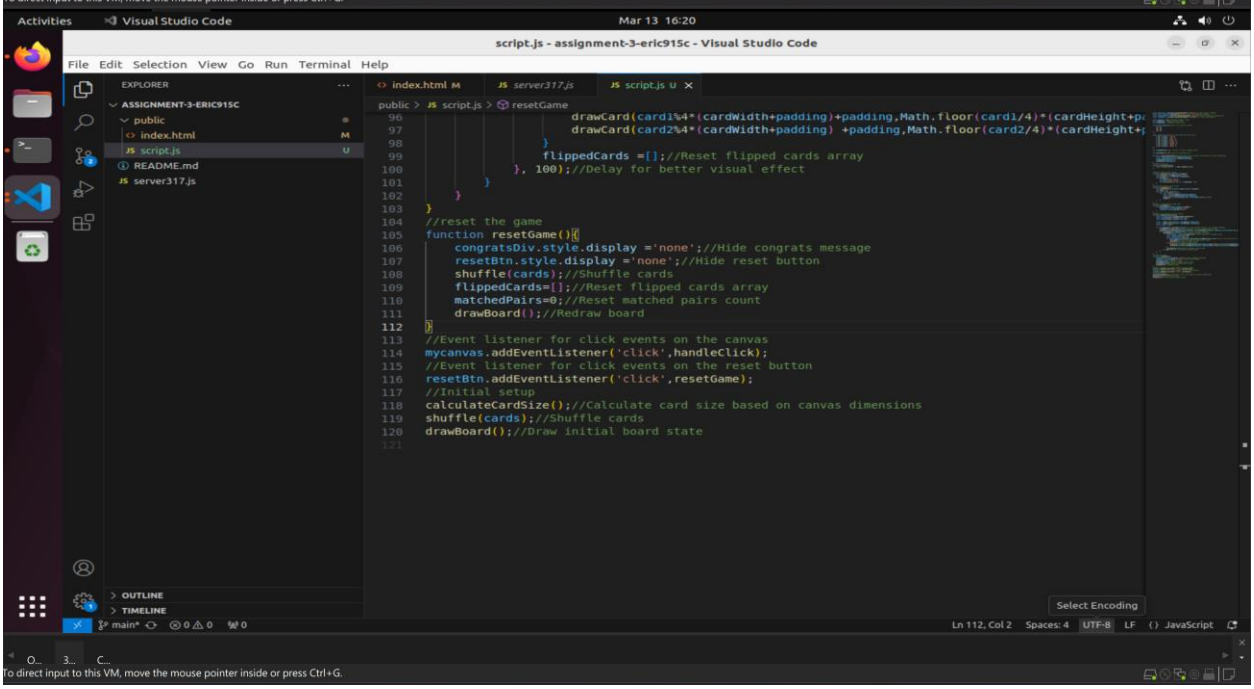
Screen shots:

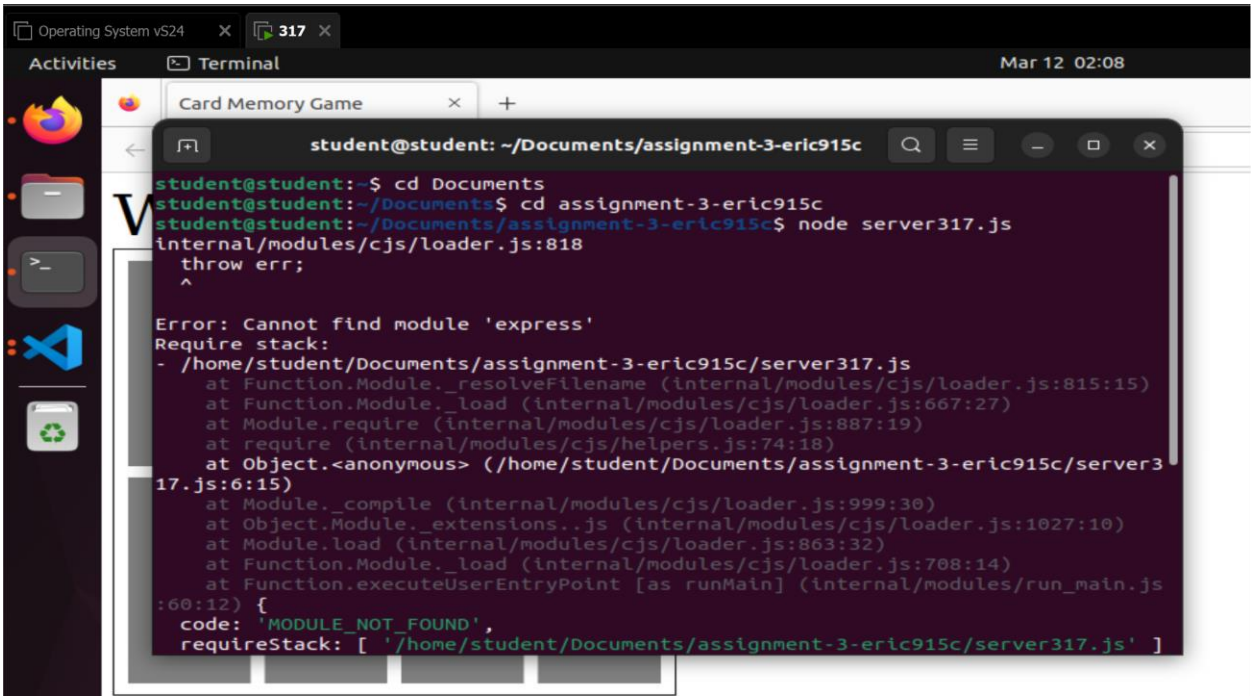

index.html - assignment-3-eric915c - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
ASSIGNMENT-3-ERIC915C
public
index.html
script.js
README.md
server317.js
index.html M
README.md
script.js U
public > index.html > html > body > main > script
<!DOCTYPE html>
<html>
<head>
<title>Card Memory Game</title>
<style>
canvas {
border: 1px solid black;
}
</style>
</head>
<body>
<header style="font-size: 50px;">Welcome to Memory Game</header>
<main>
<canvas id="mycanvas" width="400" height="400"></canvas>
Congratu
<button id="resetBtn" style="display: none; margin-top: 10px;">Reset Game</button>
<script src="script.js"></script>
</main>
</body>
</html>
OUTLINE
TIMELINE
Go to Line/Column
Ln 18, Col 35 Spaces: 4 UTF-8 LF HTML

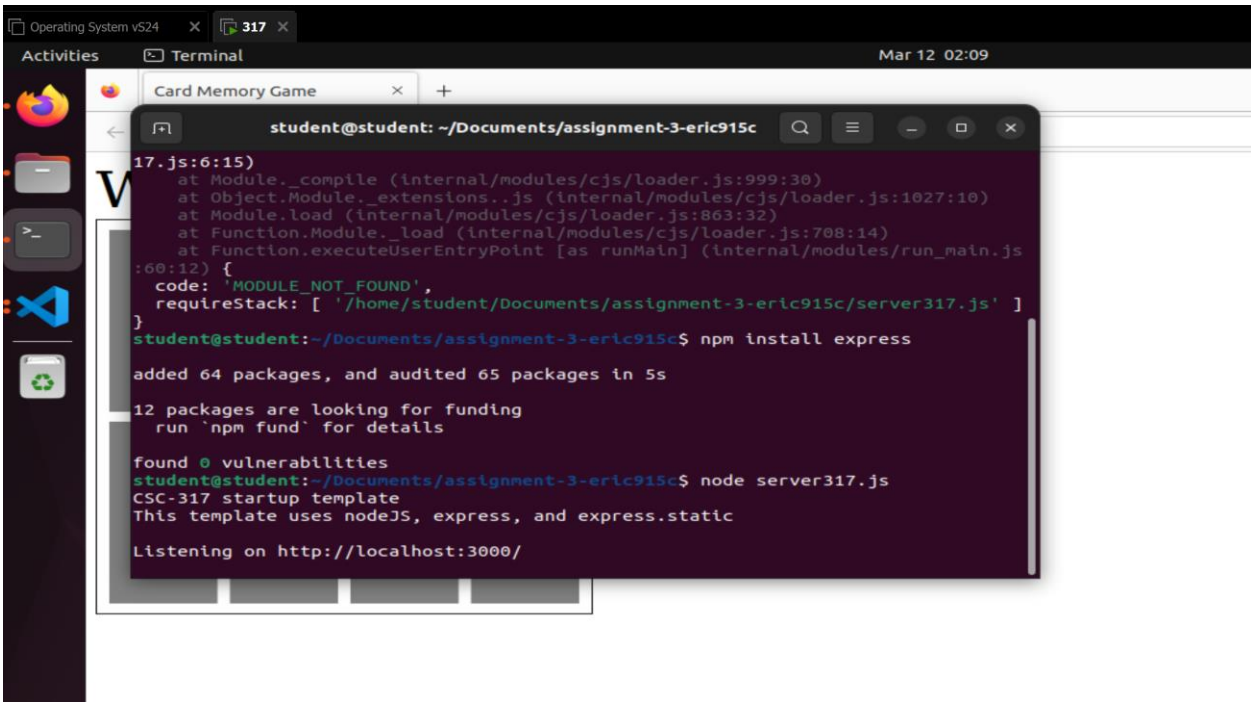

Activities
Visual Studio Code
Mar 13 16:17
script.js - assignment-3-eric915c - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
ASSIGNMENT-3-ERIC915C
public
index.html
script.js
README.md
server317.js
OUTLINE
TIMELINE
index.html M
server317.js
script.js U
public > script.js > resetGame
const mycanvas = document.getElementById('mycanvas');//Get canvas element
const ctx = mycanvas.getContext('2d'); //Get 2D rendering context
const congratsDiv = document.getElementById('congrats');//Get congrats message div
const resetBtn = document.getElementById('resetBtn'); //Get reset button
const padding = 10;//Padding between cards
let cardWidth; //Width of each card
let cardHeight; //Height of each card
const pairsToMatch=[//Pairs of cards to match
[1, 1],
[2, 2],
[3, 3],
[4, 4]
];
const cards =[//Array of cards with IDs and colors
{ id: 1,color: 'red'},
{ id: 1,color: 'red' },
{ id: 2,color: 'blue' },
{ id: 2,color: 'blue' },
{ id: 3,color: 'green' },
{ id: 3,color: 'green' },
{ id: 4,color: 'pink' },
{ id: 4,color: 'pink' }
];
let flippedCards= []; //Array to store flipped cards
let matchedPairs =0;//Count of matched pairs
function calculateCardSize(){//calculate size of each card based on canvas dimensions
const TheWidth=mycanvas.width-padding*5;
const TheHeight=mycanvas.height-padding*3;
cardWidth=TheWidth/ 4;
cardHeight=TheHeight/2;
}
//shuffle an array
main*
Ln 112, Col 2
Spaces: 4
UTF-8
LF
JavaScript


Activities
Visual Studio Code
Mar 13 16:18
script.js - assignment-3-eric915c - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
ASSIGNMENT-3-ERIC915C
public
index.html
script.js
README.md
server317.js
OUTLINE
TIMELINE
index.html M
server317.js
script.js U
public > script.js > resetGame
cardWidth=TheWidth/ 4;
cardHeight=TheHeight/2;
}
//shuffle an array
function shuffle(array){
return array.sort(() => Math.random()-0.5);
}
//draw a card on the canvas
function drawCard(x,y,id,color,flipped){
ctx.fillStyle =flipped ?color:'gray';
ctx.fillRect(x,y,cardWidth,cardHeight);
if (flipped){
ctx.fillStyle ='white';
ctx.font ='20px Arial';
ctx.fillText(id,x+ 10, y + cardHeight / 2);
}
}
//draw the initial board state
function drawBoard(){
ctx.clearRect(0,0,mycanvas.width,mycanvas.height);
let index = 0;
for (let i =0; i<2;i++) {
for (let j=0;j<4; j++) {
const x=j*(cardWidth+padding)+padding;
const y=i*(cardHeight+padding)+padding;
drawCard(x,y,cards[index].id, cards[index].color,false);
index++;
}
}
}
//check if the game has been won
function checkWin(){
if (matchedPairs===pairsToMatch.length) {
congratsDiv.style.display='block';
resetBtn.style.display='block';
}
}
main*
Ln 112, Col 2
Spaces: 4
UTF-8
LF
JavaScript

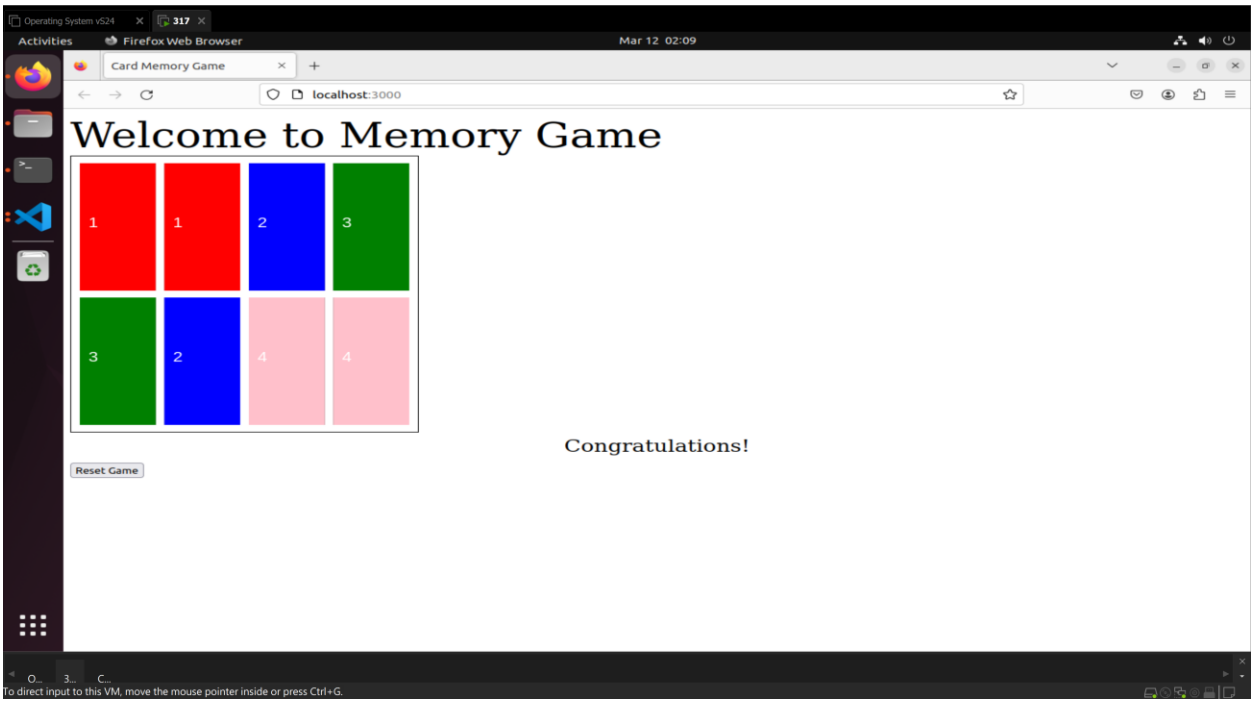

Activities
Visual Studio Code
Mar 13 16:19
script.js - assignment-3-eric915c - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
ASSIGNMENT-3-ERIC915C
public
index.html
script.js
README.md
server317.js
OUTLINE
TIMELINE
index.html M
server317.js
script.js U
public > script.js > resetGame
//handle click events on the canvas
function handleClick(event){
const rect=mycanvas.getBoundingClientRect();
const mouseX=event.clientX-rect.left;
const mouseY=event.clientY-rect.top;
const c =Math.floor(mouseX /(cardWidth +padding));
const r =Math.floor(mouseY /(cardHeight +padding));
const index =r*4+c;
if (!flippedCards.includes(index)){//If the clicked card is not already flipped
flippedCards.push(index);//Add clicked card to flipped cards array
drawCard(c*(cardWidth+padding)+padding,r *(cardHeight+padding) +padding,cards[index].id,car
if (flippedCards.length===2){//If two cards are flipped
setTimeout(()=>{//Wait a bit before checking for match
const[card1, card2]=flippedCards;// Get indices of flipped cards
const id1=cards[card1].id;
const id2=cards[card2].id;
if (pairsToMatch.some(pair=>pair[0]===id1&&pair[1]===id2)) {//If the cards form a
matchedPairs++;//Increment matched pairs count
checkWin();//Check if game is won
} else {//If the cards do not match
drawCard(card1%4*(cardWidth+padding)+padding,Math.floor(card1/4)*(cardHeight+pa
drawCard(card2%4*(cardWidth+padding) +padding,Math.floor(card2/4)*(cardHeight+
}
flippedCards =[];//Reset flipped cards array
}, 100);//Delay for better visual effect
}
}
}
//reset the game
function resetGame(){
congratsDiv.style.display ='none';//Hide congrats message
resetBtn.style.display ='none';//Hide reset button
shuffle(cards);//Shuffle cards
flippedCards=[];//Reset flipped cards array
matchedPairs=0;//Reset matched pairs count
drawBoard();//Redraw board
Go to Line/Column
Ln 112, Col 2
Spaces: 4
UTF-8
LF
JavaScript
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.


Activities
Visual Studio Code
Mar 13 16:20
script.js - assignment-3-eric915c - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
ASSIGNMENT-3-ERIC915C
public
index.html
script.js
README.md
server317.js
OUTLINE
TIMELINE
index.html M
server317.js
script.js U
public > script.js > resetGame
drawCard(card1%4*(cardWidth+padding)+padding,Math.floor(card1/4)*(cardHeight+pa
drawCard(card2%4*(cardWidth+padding) +padding,Math.floor(card2/4)*(cardHeight+
}
flippedCards =[];//Reset flipped cards array
}, 100);//Delay for better visual effect
}
}
}
//reset the game
function resetGame(){
congratsDiv.style.display ='none';//Hide congrats message
resetBtn.style.display ='none';//Hide reset button
shuffle(cards);//Shuffle cards
flippedCards=[];//Reset flipped cards array
matchedPairs=0;//Reset matched pairs count
drawBoard();//Redraw board
}
//Event listener for click events on the canvas
mycanvas.addEventListener('click',handleClick);
//Event listener for click events on the reset button
resetBtn.addEventListener('click',resetGame);
//Initial setup
calculateCardSize();//Calculate card size based on canvas dimensions
shuffle(cards);//Shuffle cards
drawBoard();//Draw initial board state
Select Encoding
Ln 112, Col 2
Spaces: 4
UTF-8
LF
JavaScript
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.


Operating System vS24
317
Activities
Terminal
Mar 12 02:08
Card Memory Game
student@student: ~/Documents/assignment-3-eric915c
student@student:~$ cd Documents
student@student:~/Documents$ cd assignment-3-eric915c
student@student:~/Documents/assignment-3-eric915c$ node server317.js
internal/modules/cjs/loader.js:818
throw err;
^
Error: Cannot find module 'express'
Require stack:
- /home/student/Documents/assignment-3-eric915c/server317.js
at Function.Module._resolveFilename (internal/modules/cjs/loader.js:815:15)
at Function.Module._load (internal/modules/cjs/loader.js:667:27)
at Module.require (internal/modules/cjs/loader.js:887:19)
at require (internal/modules/cjs/helpers.js:74:18)
at Object.<anonymous> (/home/student/Documents/assignment-3-eric915c/server317.js:6:15)
at Module._compile (internal/modules/cjs/loader.js:999:30)
at Object.Module._extensions..js (internal/modules/cjs/loader.js:1027:10)
at Module.load (internal/modules/cjs/loader.js:863:32)
at Function.Module._load (internal/modules/cjs/loader.js:708:14)
at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:60:12) {
code: 'MODULE_NOT_FOUND',
requireStack: ['/home/student/Documents/assignment-3-eric915c/server317.js']


Operating System vS24
317
Activities
Terminal
Mar 12 02:09
Card Memory Game
student@student: ~/Documents/assignment-3-eric915c
17.js:6:15)
at Module._compile (internal/modules/cjs/loader.js:999:30)
at Object.Module._extensions..js (internal/modules/cjs/loader.js:1027:10)
at Module.load (internal/modules/cjs/loader.js:863:32)
at Function.Module._load (internal/modules/cjs/loader.js:708:14)
at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:60:12) {
code: 'MODULE_NOT_FOUND',
requireStack: ['/home/student/Documents/assignment-3-eric915c/server317.js']
}
student@student:~/Documents/assignment-3-eric915c$ npm install express
added 64 packages, and audited 65 packages in 5s
12 packages are looking for funding
run `npm fund` for details
found 0 vulnerabilities
student@student:~/Documents/assignment-3-eric915c$ node server317.js
CSC-317 startup template
This template uses nodeJS, express, and express.static
Listening on http://localhost:3000/


Card Memory Game
localhost:3000
Welcome to Memory Game


Card Memory Game
localhost:3000
Welcome to Memory Game
1
1
2
3
3
2
4
4
Congratulations!
Reset Game