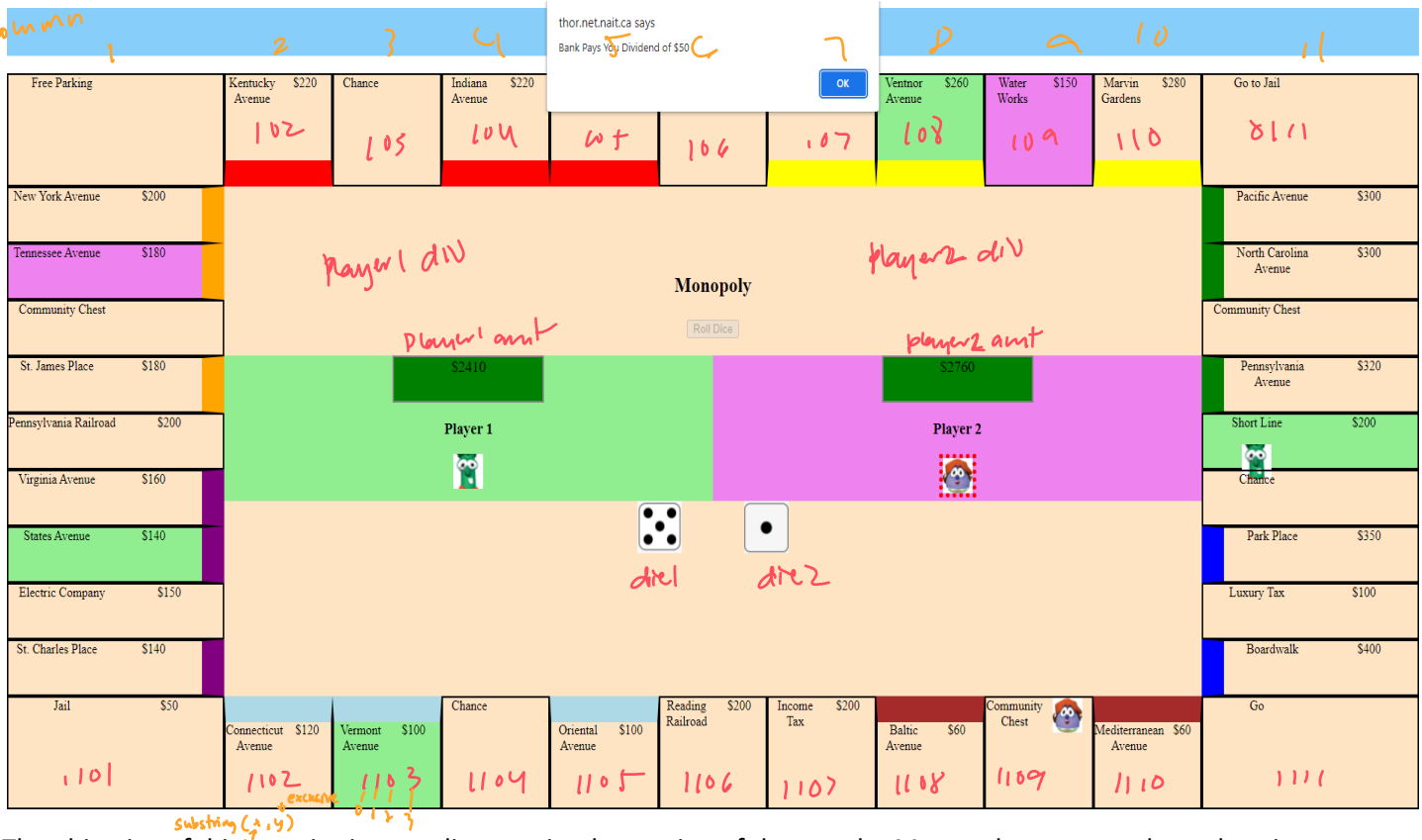
School of Applied Sciences And Technology
CMPE2000 - Lab 2 - Implementing Monopoly Game using Javascript



The objective of this exercise is to replicate a simpler version of the popular Monopoly game, on the web, using JavaScript!

You will be provided a starting HTML file. It is encouraged to make no changes to this file (other than your name). If you are stuck and you find that updating this file will help, you must comment your changes with an explanation.

Note that this file contains some helpful items for your use. These are:

gameboard	Contains each square of the gameboard in order of GO (at position 0) through Boardwalk (at position 39), plus the middle square. All the squares that are involved in dice rolls are tagged as SECTION – SECTION is not used for anything else. TIP: strategic use of document.querySelectorAll with the provided ordering will prove helpful!
suite	Each game square includes a suite attribute that can be accessed using JavaScript. It contains the ROW and COLUMN coordinates of each square for use with a grid (RRCC). Parse them out and use them to place everything in the correct place.
class	Each square contains class attributes that indicate if the square is a corner or regular, as well as the type of square. Use these in your JavaScript code and CSS file to set up the grid accordingly

While coding the solution to this exercise, using JavaScript, you are free to add/remove/update attributes and classes of any cell as required, and add elements.

Val = $ amount to be subtracted

Rules to Implement

Community Chest/Chance

Declare these arrays in your JavaScript and randomly select one of these rules whenever a player lands on either of these squares

```
const takeAChanceText = ["Second Place in Beauty Contest: $10", "Bank Pays You Dividend of $50","Repair your Properties. You owe $250", "Speeding Fine: $15", "Holiday Fund Matures: Receive $100", "Pay Hospital Fees: $100"];
const takeAChanceMoney = [10, 50, -250, -15, 100, -100];
```

Go

When the player passes by, or lands on, Go, add $200 to his balance.

Tax

When the player lands on any tax square, deduct the amount specified in that squares “val” attribute.

Go to Jail

Immediately send the player to Jail without passing Go

Jail

Pay $50 immediately and next turn continues per normal

Properties that can be purchased

If the player lands on a property that hasn’t been purchased, it will be automatically purchased. The property’s background color will change to indicate which player owns it.

If the player lands on a property that was already purchased by the other player, rent will automatically be paid according to these rules:

- Utilities: Rent will be 5x the value shown on the dice at the time
- Railroads: Rent will be $25 per RR owned by the other player
- Coloured properties: Rent starts out at 10% of the purchase price, and every time rent is paid out, the rent goes up by 20% of the current rent amount (always in integer increments). For example, if New York is purchased, initial rent is $20, then $24, then $29, etc.

Alerts should be shown to the user when landing on Community Chest, Chance, Tax, Go to Jail and Jail to advise what is happening. The dollar amounts for each player shall automatically update at the end of each turn.

Roll Dice

The game will display whose turn it is visually (your choice on how). In the example, Player2 is currently indicated with the red dotted border. Pressing the “Roll Dice” button will randomly roll 2 dice and display the results visually on the screen using images. (All images must be **preloaded** at page load time). The player’s marker will walk across the board stopping briefly at each square on its way around the board. Implement this using a timer. The “Roll Dice” button shall be disabled while the marker is moving.

If the player runs out of money during a turn, an alert “Player X loses” must be displayed.

Players will start with $3000 each.

If the two dice have the same value (doubles) the current player gets another turn, otherwise the next turn goes to the other player.

Note:
This lab is quite long. You are advised to proceed incrementally.

Testing:

Testing each possible outcome based purely on random occurrences is quite challenging. Instead, you are required to create a function that can have a player and a number of slots as parameters. It causes the player to move the required number of slots (stepwise on a timer) and invokes the required code to obtain the desired outcome. This function will be called by the JS code upon dice roll, but it can also be called from the console for easy testing.

Rubric:

Item	Max Marks	Penalty
Game Setup- Proper positioning of cells, players and dice	20	Wrong positioning of cells : -20 Players not positioned: -8 Cells colors missing: -5 Cell texts and values missing: -20
Dice Roll and display	8	
Correct player marker moves using timer	8	Player reaches destination but does not move using timer: -4 Incorrect player moves: -4
Doubles gives turn to same player	4	
Go Jo Jail- Immediately jumps to jail	3	
Landing on Jail- Penalty applied	2	
Tax – Penalty applied	5	
Player purchasing property- Amount deducted as required and property changes color properly	10	Amount not deducted from player amount: -5 Property not changing color as required: -5
Utilities- Rent= 5x value on dice	5	
Railroads- Rent $25x RR owned by other player	10	
Colored properties- Rent starts at 10% of property value and goes up by 10% of current rent amount	10	
Alerts for Community Chest, Chance, Tax, Go to Jail and Jail	5	
Separate method that allows easy testing	10	