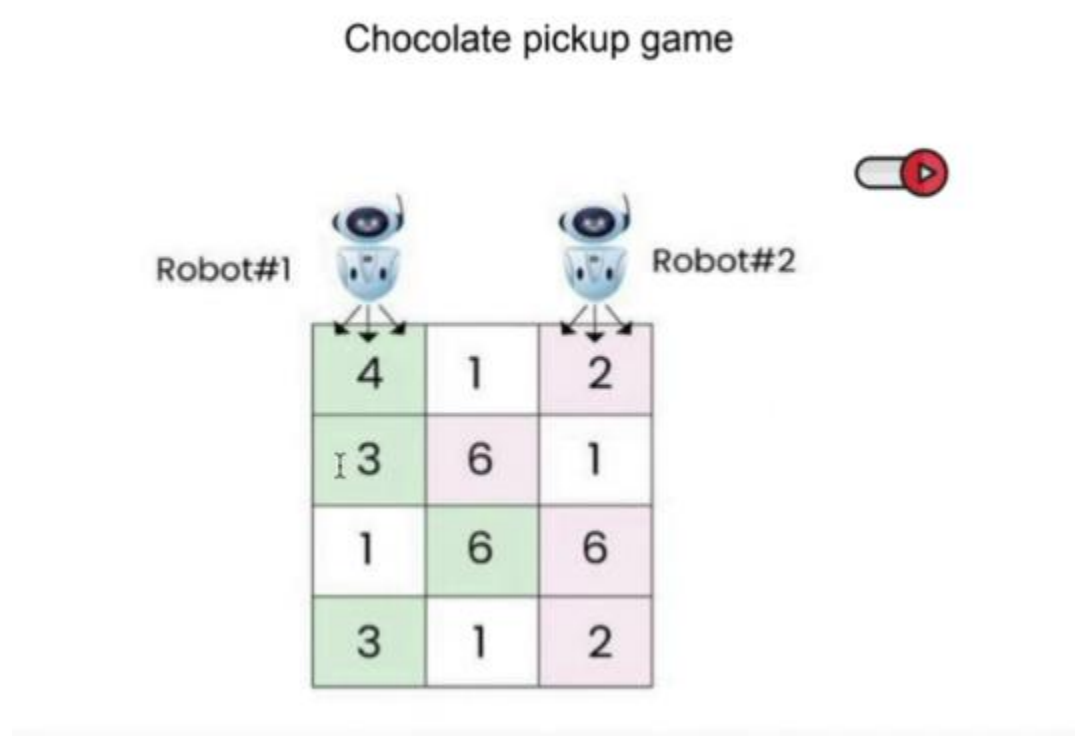





Chocolate pickup game

Problem Statement






Build a board game where the objective is to collect chocolates from a $N \times N$ grid by using two robots where the robots can only move vertically down, diagonally left down and diagonally right down.

The goal is to collect as many chocolates as possible. The robots can collect chocolates from the same cell if they both land on it, but the chocolates are only counted once.

To help the robots, you can build keyboard shortcuts   and  to move down, diagonally left down and diagonally right down respectively. Player should be able to use the “tab” key to toggle between two robots.

There should be an auto play mode which will show the player the optimal chocolate collection to maximize the collected chocolates.

LIST OF REQUIREMENTS:

1. Render a matrix grid representing the chocolate board, where each cell shows the number of chocolates.
2. The grid should be adjustable for different sizes (N rows x N columns).
3. The chocolates in each grid cell will be between 1 -10.
4. Allow for resetting the game at any point in time.
5. The grid should be filled with non-negative chocolates to begin with.
6. Visually distinct markers for both Robot 1 and Robot 2, highlighting their current positions on the grid Robot 1 should be initially positioned at (0,0) and Robot2 at (0, N-1)
7. Initially Robot 1 would be ready to make its move which should be visibly distinct.
8. Player can make a move with Robot1 to the cell using keys  /  /  to move down, diagonally left down and diagonally right down respectively
9. After Robot1 makes its move, Robot2 should be ready to make its move automatically. However
10. However Player can press “tab” to toggle between Robot 1 and Robot 2 at any point in time
11. Highlight the paths both robots take as they move down the grid. Different colors or patterns for Robot 1 and Robot 2’s paths. Indicate cells where both robots have landed (if they coincide on the same cell).
12. A real-time counter showing the total chocolates collected by both Robots. Separate counters for individual robots collections as well.
13. The Game ends when both robots reach the last row of the board. At the end of the game (when both robots reach the last row), display the total chocolates collected.
14. Allow the Player to manually select moves or toggle an “Auto-Play” mode that shows an optimal chocolate collecting path to collect maximum chocolates from the board.

Points to consider:

- Focus on the modularity of code and design of the solution. Keep performance of the application in mind.
- The solution should be coded in native / vanilla JavaScript and CSS. Please refrain from using any plugins / libraries like jQuery / React / Angular or css frameworks like Bootstrap.

However, you can use tooling such as webpack, grunt, gulp, etc. ● The final solution should work without errors.

● Do not completely ignore the layout / visual design. A minimalist visual design / layout must be followed, it is also important for us to evaluate your CSS knowledge.

You will be evaluated based on:

1. Correctness and completeness of the solution.
2. Code design and quality.
3. Optimal data structure.
4. Visual aesthetics (the UI should be as close as possible to the given design).
5. Technology choices (e.g. ES 6/7 over ES 5 - avoid mixing of ES 5/6/7).
6. Your understanding of the problem statement.
7. Explanation of the principle and the approach taken for solution.