

## Treasure Hunt

### Core Concepts:

- **Grid-based Exploration:** The game is played on a 10x10 grid where each cell can either contain a treasure, a trap, or be empty.
- **Player Movement:** Players can move up, down, left, or right across the grid using buttons on the webpage.
- **Interactive Elements:** Cells in the grid may contain treasures, which increase the player's score, or traps, which decrease the number of moves left or affect the score negatively.
- **Finite Moves:** The player has a limited number of moves (20 moves) to maximize their score by collecting as many treasures as possible.

### Educational Objectives:

- **Programming Practice:** This game provides a practical application of JavaScript, focusing on functions, loops, conditional statements, and basic DOM manipulation.
- **Algorithmic Thinking:** Players must think strategically about how to navigate the grid to maximize their score while minimizing penalties.
- **Problem Solving:** The setup encourages problem-solving as players decide their movement strategy based on their current position and remaining moves.

### Gameplay Mechanics:

- **Initialization:** At the start, the grid is initialized with a random placement of treasures and traps.
- **Interaction:** As the player moves to different cells, they interact with the contents of the cells, triggering gains (treasures) or losses (traps).
- **Visibility and Updates:** The grid updates visually on the webpage with each move, showing the player's current position, treasures collected, and traps encountered.
- **Scoring and Moves:** The score and remaining moves are displayed on the webpage, updating dynamically with each action.

### Technical Implementation:

- **HTML and CSS:** Used for structuring and styling the game interface.
- **JavaScript:** Handles the game logic, DOM updates, and user interactions.