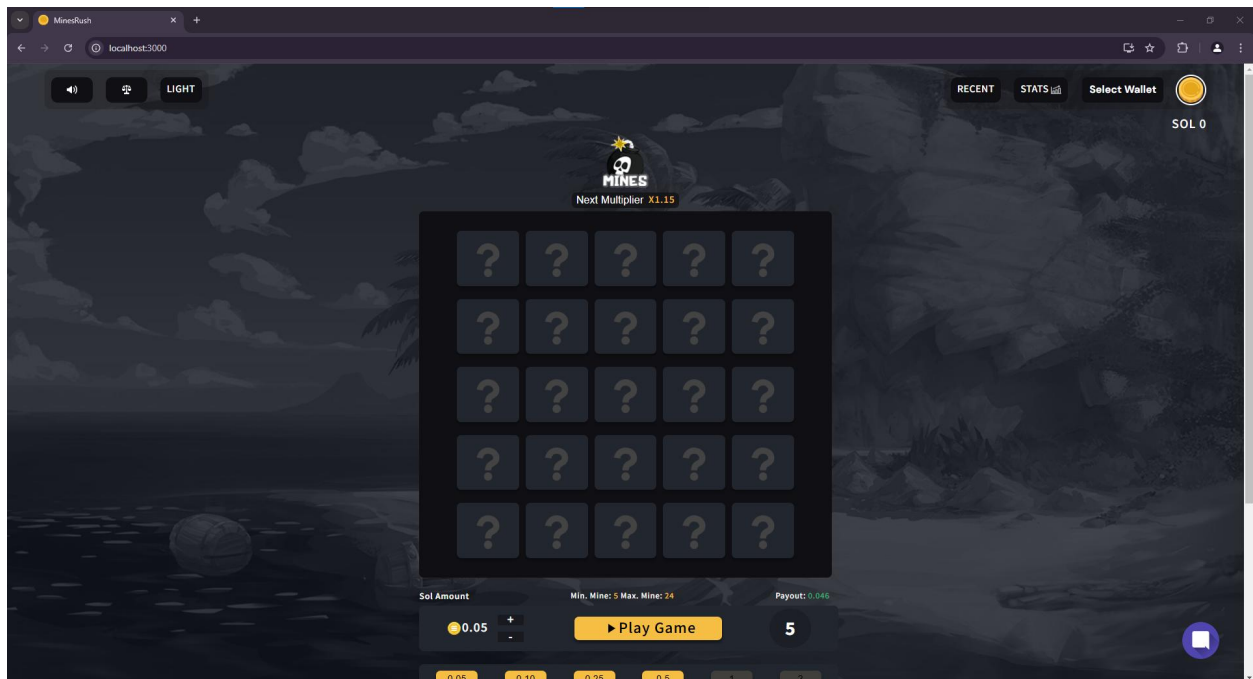


# Game Overview

This game is a digital "Minesweeper-inspired" gambling game that leverages the popularity of chance-based gaming with cryptocurrency integration, creating a unique combination of risk, reward, and strategic decision-making. The game invites players to test their luck and nerves as they reveal tiles on a 5x5 grid, aiming to avoid hidden "mines" and increase their winnings.

Players wager cryptocurrency (in this case, Solana, or "SOL") and attempt to uncover safe tiles to increase their payout multiplier. With each safe tile revealed, the potential winnings grow, but hitting a mine results in an instant loss of the wager. The player can cash out at any time, adding a strategic layer of risk management as they decide whether to press on or secure their current earnings.

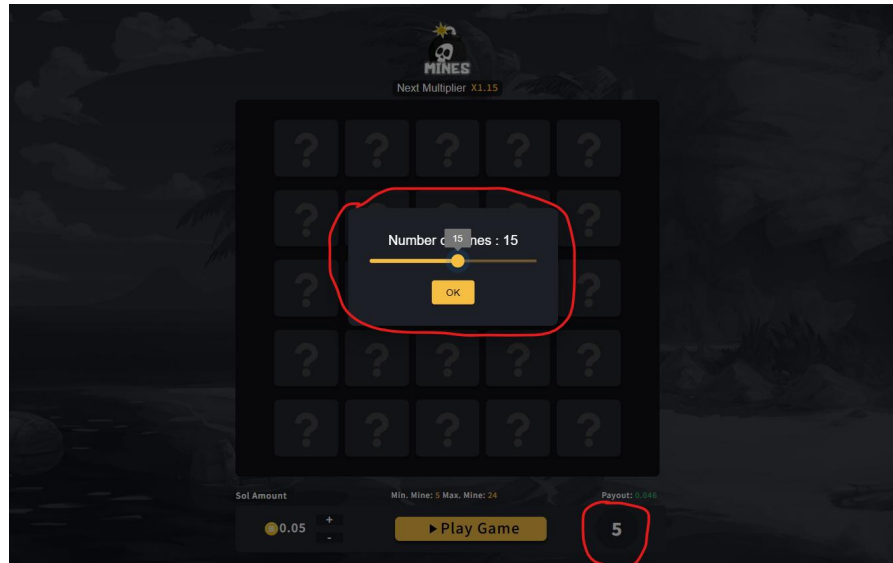


## 1. Functional Requirements

### Game Logic and Mechanics

#### 1. Grid Generation:

- The game must display a 5x5 grid containing 25 clickable tiles, each with a hidden state that can either be safe or contain a mine.
- Mines should be randomly distributed across the grid based on the player's selected difficulty (number of mines).



## 2. Mine Configuration:

- Allow the player to select the number of mines on the grid before starting (e.g., between 5 and 24 mines).
- The number of mines impacts the risk level and potential reward, increasing with higher mine counts.

## 3. Safe Tile Mechanic:

- Each revealed safe tile should incrementally increase a multiplier, which boosts the potential payout.
- Provide real-time feedback on the updated payout as more safe tiles are revealed.

## 4. Game Outcomes:

- If a player clicks a tile containing a mine, they instantly lose the game and forfeit their wager.
- Players can choose to “cash out” their winnings at any time based on the current multiplier, ending the game with their current payout.

## 5. Game Start and Flow:

- The game begins upon pressing the “Play Game” button, at which point the grid activates.
- Each subsequent click on a tile either reveals a safe space or a mine, with updated multipliers shown after each successful reveal.

## Wagering and Cash-Out System

### 1. Wager Input:

- Players should be able to set a wager amount in SOL (Solana cryptocurrency) via "+" and "-" buttons or select from predefined wager buttons (e.g., 0.05, 0.10, 0.25, 0.5 SOL).

**2. Cash-Out Functionality:**

- Players should have the option to cash out at any time during gameplay, capturing their current multiplier and payout.
- Display the cash-out option prominently, especially after each safe tile reveal.

**3. Payout Calculation:**

- Calculate payout in real time based on the wager amount, number of mines, and the current multiplier.
- Show projected winnings dynamically in a clear, visible format.

## **Wallet and Cryptocurrency Integration**

**1. Wallet Selection and Connectivity:**

- Allow players to connect a Solana-compatible wallet to deposit and withdraw funds.
- Ensure seamless wallet connectivity to facilitate gameplay without interruptions.

**2. Transaction Management:**

- Support wagering, payout, and cash-out transactions in SOL cryptocurrency.
- Display the player's current SOL balance prominently.

## **2. Interface (UI/UX) Requirements**

### **Main Game Interface**

**1. Game Title and Branding:**

- Display a distinctive game title ("MINES") with an associated bomb icon to convey the theme of risk.
- Show the current multiplier (starting from x1.15) below the title and update it dynamically with each successful tile reveal.

**2. Grid Design:**

- Display a 5x5 grid with 25 clickable tiles, each initially marked with a "?" symbol.
- Upon clicking, tiles should reveal either a mine (game over) or a safe space (incremental multiplier).

**3. Top Control Panel:**

- Provide icons for sound and language settings in the top-left corner to enhance accessibility for global players.
- Include a toggle for Light/Dark modes, allowing players to switch themes based on personal preference.

#### 4. **Player Stats and Wallet Options:**

- Display “RECENT” and “STATS” buttons to allow players to view recent games and detailed statistics (e.g., average multiplier, success rate).
- Include a “Select Wallet” option to facilitate wallet connection for depositing and withdrawing funds.

### **Bottom Panel (Wager Settings)**

#### 1. **Wager Settings:**

- Allow players to set their wager in SOL, with controls to increment or decrement the amount by small units.
- Display the selected wager amount prominently, along with quick-select buttons for common wager amounts.

#### 2. **Mine Count Selection:**

- Provide options to select the number of mines, impacting risk and reward.
- Display a minimum and maximum mine range (e.g., between 5 and 24 mines) and ensure selected mine count is visible.

#### 3. **Projected Winnings Display:**

- Show the real-time projected winnings in green, updating with each successful safe tile reveal.
- Ensure visibility of this display as it’s a key motivator for players to continue or cash out.

### **Game Interaction Controls**

#### 1. **Play Game Button:**

- Provide a prominent “Play Game” button to start gameplay with the selected wager and mine count settings.
- After pressing “Play Game,” enable tile selection and display the cash-out button.

#### 2. **Real-Time Feedback:**

- Update the projected winnings display dynamically based on the multiplier and number of mines remaining.
- Highlight the cash-out option to make it easy for players to exit the game and collect their winnings.

### 3. Backend and Technical Requirements

#### Game Mechanics and Randomization

##### 1. Randomized Mine Placement:

- Ensure that mines are randomly placed in the grid each time a game starts.
- Implement a secure random number generator to ensure fair distribution of mines, maintaining game integrity.

##### 2. Multiplier Calculation Logic:

- Use an algorithm to calculate the payout multiplier, increasing it with each safe tile revealed.
- Adjust the multiplier based on the number of mines and grid size for balanced risk-reward dynamics.

#### User Account and Wallet Management

##### 1. Wallet Integration:

- Integrate Solana wallet functionality to facilitate transactions in cryptocurrency (SOL).
- Ensure secure wallet connections and manage deposits, withdrawals, and cash-outs directly via blockchain protocols.

##### 2. Transaction Logging and Verification:

- Record each transaction, including wagers, payouts, and cash-outs, to ensure accountability.
- Verify each transaction on the Solana blockchain for transparency and security.

#### Data Handling and Storage

##### 1. Player Stats and Game Records:

- Track player performance data, including recent games, success rates, and multipliers achieved.
- Store and display real-time stats to give players insights into their performance.

##### 2. Game State Management:

- Manage game state securely to prevent tampering, especially in cases where a player cashes out mid-game.
- Use efficient data handling for real-time feedback and immediate game state updates.

#### Security and Compliance

### 1. **Anti-Cheat Mechanisms:**

- Implement measures to detect and prevent cheating, ensuring fair gameplay for all players.
- Regularly audit randomization and game logic to prevent exploitation.

### 2. **Compliance with Cryptocurrency Regulations:**

- Ensure the platform meets regulatory requirements for cryptocurrency gaming in applicable jurisdictions.
- Implement responsible gaming features, such as allowing users to set wager limits or cooling-off periods.

## 4. **Optional Social and Interactive Features**

### **Live Chat and Community**

#### 1. **In-Game Chat:**

- Provide an optional chat feature where players can interact with each other or access support.
- Moderation controls should be implemented to maintain a positive player community.

#### 2. **Leaderboard and Competitive Stats:**

- Implement a leaderboard for players to see top scores or highest multipliers achieved.
- Update leaderboards in real-time to foster competition and enhance player engagement.

## 5. **Testing and Quality Assurance**

### **Testing Requirements**

#### 1. **Functional Testing:**

- Test all game functions, including random mine placements, wager settings, and cash-out features, to ensure they operate smoothly.

#### 2. **Security Testing:**

- Ensure robust security for wallet connections, data handling, and transaction integrity.

#### 3. **User Interface Testing:**

- Validate UI elements across various devices and screen sizes to ensure consistent and accessible gameplay.

#### 4. Load Testing:

- Test the platform's ability to handle high volumes of concurrent players, especially during peak hours or events.

**Developer's Role:** You are tasked with building a **mine-sweeper-style blockchain-based betting game** where players wager cryptocurrency (e.g., **Solana**) to uncover safe tiles on a grid while avoiding hidden mines. Players can adjust the number of mines and bet amounts, with successful tile reveals increasing the payout multiplier. The interface will include features for managing bets, selecting wallets, and viewing recent game plays.