Fully Dressed Use Case 3: Rolling a Dice

Primary Actor : Player (Human or Computer)

Stakeholders and Interests:

Stakeholders	Interests
Player	Wants to roll dice to get fair combinations that allow placement on high-scoring squares, Wants clear feedback on dice results, Wants to understand available combinations to place a stone on the board.
Other Player(Human and/or Computer)	Want to see dice results for fairness, Want to know what combinations are available to the current player.
Developers	Wants the dice rolling mechanism to be fair and random, Ensuring generation of random dice values, Accurately determine valid combinations, Enforce rules.
Distributors	Reaching the target gaming audience, ensuring the availability of the product and its updates.
Customer Support	Handling game failures or dice failure if any, Assisting with any game queries, Handling complaints about enforcing rules if anty.

Preconditions:

- The game board is set up.
- It is the player's turn.
- Player has at least one stone remaining.
- The player has not yet rolled the dice in this turn.

Success Guarantee (Postconditions):

- The dice have been rolled.
- The players can see the result of the dice roll.
- The system calculates available combinations and displays them.

- Player is ready to proceed to the stone placement decision.

Main Success Scenario:

1. The system indicates it is the player's turn.[Alt1: *Computer Player (Easy Difficulty),* Alt2: *Computer Player (Hard Difficulty)]*
2. The system prompts the player to roll dice.
3. The player chooses to roll the dice.
4. The system rolls all six dice and generates six random dice values(1-6).
5. The system displays the dice results to all the players.
6. The system analyzes the dice values and calculates all valid combinations available from this roll. [Alt3: *Invalid Combination*]
7. The system displays the valid combinations to the player.[Alt4: *Five of a Kind or 6 Straight,* Alt5: *Six of a kind*]
8. The system highlights squares on the board where placement is possible based on the combinations.
9. The system informs the player to place a stone or save the game.[Alt6: *Player chooses to save the game*] [Use Case Ends]

Alternative Flows:

Alt1: *Computer Player (Easy Difficulty)*

1. The system automatically selects the highest-scoring valid combination.
2. Proceed to the "Placing a Stone" use case.
3. Flow resumes at Step 9.

Alt2: *Computer Player (Hard Difficulty)*

1. The system selects a combination that blocks opponents' high scoring moves.
2. Proceed to the "Placing the Stone" use case.
3. Flow resumes at Step 9.

Alt3: *Invalid Combination*

1. The system informs the player that the combination is invalid.
2. The player decides to roll again.

3. Flow resumes at Step 1.

Alt4: *Five of a Kind or 6 Straight*

1. The player may place a stone on any free square.
2. The system highlights every single free square on board
3. Flow resumes at Step 9.

Alt5: *Six of a kind*

1. Players can place the stone on any free or occupied square.
2. System moves the existing stone to a new location chosen by the player, if the selected square is occupied.
3. Flow resumes at Step 9.

Alt6: *Player chooses to save the game*

1. The player confirms he wants to save the game.
2. The system saves the current game state, including board configuration, player information, and current turn.
3. The system confirms successful saving of the game.
4. The use case ends.
5. Flow resumes at Step 5.

Exceptions:

- If the system fails to roll the dice or display the result, it prompts the player, logs the error details, and ends the current turn so that the player may try again. The use case continues.
- If the system cannot identify valid combinations or highlight available squares, the system informs the player of the error and attempts to restart the dice rolling process.

Special Requirements:

- The dice rolling mechanism must be random and fair.
- Colour vision deficiency support should be implemented in displaying results.
- Dice roll results must be displayed within 3 seconds.

- The game should allow saving and resuming sessions.

Open Issues:

- How does the computer player’s hard mode prioritizes moves?
- What if the player rolls an invalid combination multiple times in a row?
- How should special combinations (five of a kind, straight of six, six of a kind) be visually distinguished?