North Pole Breakout: Use Cases

Use Case 1: Start Game

Primary Actor: Player
Goal in Context: The player initiates a new game of North Pole Breakout by selecting difficulty level and starting the game.
Precondition: The game is launched, and the start screen is displayed, and the player is in a valid starting title in maze.
Trigger: The player selects the "Play" option from the start screen.

Scenario:

1. The player selects "Play" on the start screen.
2. The game presents difficulty level options: Easy, Medium, and Hard.
    a. Refer to [Select Difficulty] use case.
3. The player selects the desired difficulty level.
4. The game loads the selected level and presents the maze with the player character at the start position.
5. The player begins the game with the score counter and present tracker displayed on-screen.

Exceptions:

1. If the player selects "Quit," the game closes.
2. If the player selects "Back" on the difficulty selection screen, they return to the start screen.

Postcondition: The game starts with the selected difficulty level, and the player can now control the character to collect presents and avoid enemies and traps.

Use Case 2: Collect Reward

Primary Actor: Player
Goal in Context: The player collects presents
Precondition: The player is controlling the character within the game. Presents are available in the maze.
Trigger: The player moves the character over a present.

Main Success Scenario:

1. The player navigates the character through the maze.

2. The player successfully moves the character over a present.
3. The game updates the score counter by adding points based on the type of present collected (regular or bonus).
    a. See [Encountering Bonus Reward (Disappearing Rudolph)] use case.
4. The present is removed from the tile.
5. The game updates the present counter to reflect the number of presents collected.
6. The game continues, allowing the player to move again.

Exceptions:

1. The score doesn't update properly when a present is collected.

Postcondition: The player's score and present count are updated in the game UI.

Use Case 3: Avoid Enemy (Elves)

Primary Actor: Player

Goal in Context: The player can avoid enemies and traps

Precondition: The player is controlling the character within the game. Enemies and traps are present.

Trigger: The player navigates near an enemy or trap.

Main Success Scenario:

1. The player moves the character near an enemy or trap towards the player.
2. The system checks if the elf's new position matches the player's position.
3. The player successfully navigates away from the danger without being caught or triggered.
4. The game continues without penalty to the player and the elf continues moving in the player's direction in the next game tick.

Alternate Scenario (Failure Condition):

1. The player moves into an enemy.
2. The game triggers a loss condition (e.g., point loss or game over, depending on the difficulty level and game mechanics).

Exceptions:

1. The game could miss detecting an enemy collision and delay the "Game Over" screen.

Postcondition: The player avoids losing progress or triggering penalties by successfully evading enemies and traps.

Use Case 4: Player Wins

Primary Actor: Player
Goal in Context: The player wins the game by completing all requirements.
Preconditions: Game is functional, being controlled by the player.
Score is greater than 0.
Trigger: The player collects all the presents and moves character to the exit point.

Scenario:

1. The player navigates through the map collects all presents across the map.
2. Once all presents are collected, exit point becomes accessible.
3. The player moves to the exit point successfully, without colliding with an enemy, maintaining the score above 0.
4. The player reached the exit point.
5. A “You Won” screen is shown with the finishing score.

Exceptions:

1. Player collides with an enemy, or Score drops below 0 while navigating through map
    - Game Over screen is presented and the game ends.
    - Game transitions to a different state and the player is presented with options to “Play Again” or “Return to Main Menu”. ￼

Postconditions:

- The game has finished and transitions to a different state.
- The player is presented with options to “Play Again” or “Return to Main Menu”.

Use Case 5: Player Loses

Primary Actor: Player
Goal In Context: The player loses the game
Preconditions: Game is functional, being controlled by the player.
Trigger: The player collides with an enemy entity, or Score drops below 0.

Scenario 1 (Collision with enemy):

1. Player is navigating through the map, being chased by enemies.
2. Player enters a grid containing a moving enemy.
3. Player collides with the enemy
4. Game detects collision and a “Game Over” screen is presented.

Scenario 2 (Score below 0):

1. Player is navigating through the map, being chased by enemies.
2. Player enters a grid containing a trap.
    a. See [Encountering Traps] use case.
3. Score drops below 0.
4. Player loses and a "Game Over" screen is presented.

Exceptions:

1. The game could miss detecting an enemy collision and delay the "Game Over" screen.
2. The score doesn't update properly and doesn't trigger the loss.
3. The "Game Over" screen isn't presented.

Postconditions:

- The game has finished and transitions to a different state.
- The player is presented with options to "Play Again" or "Return to Main Menu".

Use Case 6: Encountering Traps (Puddles and Holes)

Primary Actor: Player
Goal in Context: The player needs to avoid traps to prevent point loss.
Precondition: The game is in progress, and a trap is present on the player's intended tile.
Trigger: The player moves onto a tile containing a trap (puddle or hole).

Scenario:

1. The player moves onto a tile with a trap.
2. The system detects the trap and deducts 5 points from the player's score.
3. The game continues after applying the penalty.

Exceptions:

1. If the player's score drops below zero due to penalties, the system triggers the "Game Over" screen – Refer to use case 5.

Postconditions:

- The player's score is reduced, and the game continues, or the player loses if their score reaches zero.

Use Case 7: Encountering Bonus Reward (Disappearing Rudolph)

Primary Actor: Player
Goal in Context: The player wants to collect the bonus Rudolph reward to increase their score.
Precondition: Rudolph appears randomly for a limited time on a tile near the player.
Trigger: The player moves onto a tile containing Rudolph.

Scenario:

1. Rudolph appears on a random tile for a short period (e.g., 5 seconds).
2. The player navigates towards Rudolph.
3. The player successfully moves onto the tile containing Rudolph before he disappears.
4. The system awards the player 10 points.
5. Rudolph disappears from the grid.

Exceptions:

1. If the player fails to reach Rudolph before he disappears, the bonus opportunity is lost.

Postconditions:

- The player's score increases by for example 10 points, and Rudolph disappears.

Use Case 8: Selecting Difficulty

Primary Actor: Player
Goal in Context: The player wants to adjust the game's difficulty to tailor the experience.
Precondition: The player is on the difficulty selection screen after starting the game.
Trigger: The player selects a difficulty option (Easy, Medium, Hard).

Scenario:

1. The player navigates to the difficulty selection screen from the main menu.
2. The player selects one of the difficulty options.
3. The system adjusts the number of presents, traps, and enemies based on the selected difficulty:
    a. Easy: Fewer enemies, more presents, and fewer traps.
    b. Medium: Moderate number of enemies, traps, and presents.
    c. Hard: More enemies, more traps, and fewer presents.
4. The game begins, and the player starts navigating the maze with the adjusted settings.

Exceptions:

1. If the player selects "Back" instead of a difficulty, they are returned to the main menu without starting the game.

Postconditions:

1. The game environment (presents, enemies, traps) is set up based on the chosen difficulty level.
2. The player is taken to the start of the game and begins navigating the maze with the chosen difficulty in effect.