Use Case: Start Game

Primary Actor: Player

Goal in Context: To begin a new game session where the player controls the character to navigate through a maze, collect rewards, and avoid ghosts and traps (lava and water).

Preconditions: The game application is loaded, and the main menu is displayed to the player.

Trigger: The player selects the option to start a new game from the main menu.

Scenario:
1. Player: Observe the main menu.
2. Player: Selects the "Start Game" option.
3. Player: Watches as the game board loads with the character, ghosts, rewards, punishments, and barriers placed in their initial positions.
4. Player: Begins controlling the character using the keyboard or mouse

Exceptions:
1. Game fails to load: Player is prompted to check their system or restart the game application.
2. Configuration error (e.g., controls not set): Player is directed to the settings menu to configure controls.

Priority: Essential, must be implemented

When Available: First release

Frequency of Use: Every time a player starts the game

Channel to Actor: Via game user interface (keyboard, mouse)

Secondary Actors: None

Channels to Secondary Actors: None

Open Issues:
1. Should there be different difficulty levels available from the start?
2. How can players save their progress or high scores?

Use Case: Collect treasure

Primary Actor: Player

Goal in Context: To increase the score by navigating the character to collect treasures while avoiding ghosts and obstacles (e.g lava, spikes and water).

Preconditions: Game is in progress, and treasures are placed on the game board.

Trigger: The character moves onto the cell containing a treasure.

Scenario:
1. Player: Moves the character around the maze.
2. Player: Guides character to a cell that contains a treasure.
3. System: Automatically collects the treasure , increases the player's score, and removes the reward from the board.

Exceptions:
1. Treasure disappears before being collected (if applicable, e.g., bonus rewards): Player must find another treasure.
2. The character is caught by a ghost, touches lava or spikes before collecting the reward: The game ends or a life is lost, depending on the game rules.

Priority: High, essential for gameplay

When Available: First release

Frequency of Use: Repeatedly throughout the game

Channel to Actor: Direct interaction through game controls

Secondary Actors: None

Channels to Secondary Actors: None

Open Issues:
1. How should the game balance the number and value of rewards?
2. Should special rewards offer temporary abilities or protections?

Use Case: Collect Skull

Primary Actor: Player

Goal in Context: To finish the level by navigating the character to collect all three skulls while avoiding ghosts and obstacles (e.g lava, spikes and water).

Preconditions: Game is in progress, and skulls are placed on the game board.

Trigger: The character moves onto the cell containing a skull.

Scenario:
1. Player: Moves the character around the maze.
2. Player: Guides character to a cell that contains a skull.
3. System: Automatically collects the skull, increases the player's score, and removes the reward from the board.

Exceptions:
1. The character is caught by a ghost, touches lava or spikes before collecting the reward: The game ends or a life is lost, depending on the game rules.

Priority: High, essential for gameplay

When Available: First release

Frequency of Use: Repeatedly throughout the game

Channel to Actor: Direct interaction through game controls

Secondary Actors: None

Channels to Secondary Actors: None

Open Issues: None

Use Case: Collect Skull or Treasure in the middle

Primary Actor: Player

Goal in Context: To finish the level by navigating the character to collect the skulls and treasure in the middle map while avoiding ghosts and obstacles (e.g lava, spikes and water).

Preconditions: Game is in progress, and skulls, treasures are placed on the middle square of the game board.

Trigger: The character moves onto the cell containing a skull or treasure inside the middle box area.

Scenario:
1. Player: Moves the character around the maze and into the middle box.
2. Player: Guides character to a cell that contains a skull or treasure.
3. System: Automatically collects the skull / treasure, increases the player's score, and removes the reward from the board.
4. System: Automatically make the ghost (red eyes), chase after the character at an increased speed for a few seconds

Exceptions:
1. The character is caught by a ghost, touches lava or spikes before collecting the reward: The game ends or a life is lost, depending on the game rules.

Priority: High, essential for gameplay

When Available: First release

Frequency of Use: Repeatedly throughout the game

Channel to Actor: Direct interaction through game controls

Secondary Actors: None

Channels to Secondary Actors: None

Open Issues: None

Use Case: Picking Up the Light

Primary Actor: Player

Goal in Context: To temporarily empower the character by making ghosts flee, enhancing the player's ability to navigate the maze safely.

Preconditions: The game is in progress, and light power-ups are placed on the game board.

Trigger: The character moves onto a cell containing the light power-up.

Scenario:
1. Player maneuvers the character to collect the light power-up.
2. System activates the power-up, causing ghosts to run away from the character for a few seconds, even if they are in an enraged state.
3. Player takes advantage of this temporary state to navigate through the maze more freely, collecting rewards and avoiding previously aggressive ghosts.

Exceptions:
1. The effect of the light power-up expires before the player can fully utilize its benefits.
2. The character encounters a ghost just as the power-up effect begins or ends, resulting in confusion about the ghost's behavior.

Priority: Medium

When Available: From the first release or upon introducing light power-ups

Frequency of Use: Depends on the availability of light power-ups in the game levels

Channel to Actor: Direct interaction through game controls

Secondary Actors: Ghosts (affected by the power-up)

Channels to Secondary Actors: Game mechanics that govern ghost behavior

Open Issues:
1. Determining the exact duration for which the ghosts will flee from the characterafter picking up the light.
2. Ensuring the change in ghost behavior is visually apparent to the player, so they can strategize accordingly.

Use Case: Using Green Portals

Primary Actor: Player

Goal in Context: To strategically teleport the character across the maze instantly using green portals, facilitating navigation and evasion of ghosts.

Preconditions: The game is in progress, and green portals are placed at the middle left and right areas on the game board.

Trigger: The character enters / touches a cell with a green portal.

Scenario:
1. Player identifies the location of green portals on the maze.
2. Player navigates the character to enter a green portal.
3. System teleports the character instantly to the corresponding green portal on the other side of the maze.
4. Player continues to navigate the character from the new location, leveraging the strategic advantage.

Exceptions:
1. Player attempts to use a green portal that is too close to a ghost, risking capture upon exit.
2. Technical issues prevent the portal from functioning as expected.

Priority: Medium

When Available: From the first release or upon introducing green portals into the game

Frequency of Use: As often as players encounter green portals during gameplay

Channel to Actor: Direct interaction through game controls

Secondary Actors: None (Ghosts cannot use portals)

Channels to Secondary Actors: None

Open Issues:
1. Clarification on whether all green portals are paired or if some lead to unique, one-way destinations.
2. Potential for adding visual or audio cues to indicate active portals.

Use Case: Completing the Level

Primary Actor: Player

Goal in Context: To successfully complete the level by collecting all loose treasures and all three skulls, enabling progression to the next level or completion of the game.

Preconditions: The game level is in progress with loose treasures and three skulls placed throughout the maze.

Trigger: The player collects the last piece of loose treasure and the third skull.

Scenario:
1. Player navigates the character through the maze, identifying and collecting loose treasures and skulls.
2. After collecting all designated loose treasures and skulls, the system triggers a level completion event.
3. The game transitions to the next level or displays a victory screen if it's the final level.

Exceptions:
1. The player misses a loose treasure or skull, requiring further exploration.
2. Technical issues prevent level completion even after all items are collected.

Priority: High

When Available: First release

Frequency of Use: Once per level that includes this completion criterion

Channel to Actor: Direct interaction through game controls

Secondary Actors: None

Channels to Secondary Actors: None

Open Issues: None

Use Case: Losing the Game

Primary Actor: Player

Goal in Context: To avoid losing the game by managing the character's interactions with ghosts or lava or traps

Preconditions: Game is in progress.

Trigger: The character's life count reaches zero or specific game-ending conditions are met (e.g., touching a ghost, touching lava if it results in an instant loss).

Scenario:
1. Player navigates the maze with the character.
2. The character interacts with a game-ending hazard (e.g., ghost, lava).
3. System ends the game, displaying the score and offering options to restart or exit.
4. System restarts the game with the score resetted to 0

Exceptions:
1. If the character has extra lives, losing one does not end the game but respawns the character.
2. Technical issues prevent game over condition from properly triggering.

Priority: High

When Available: First release

Frequency of Use: Occurs at the end of every lost game

Channel to Actor: Game interface (visual feedback and control inputs)

Secondary Actors: None

Channels to Secondary Actors: None

Open Issues:
1. Clarification on conditions that lead to losing the game aside from ghost interactions.
2. Mechanisms for granting the character extra lives or respawns.

Use Case: Select Difficulty

Primary Actor: Player

Goal in Context: To choose a game difficulty level that matches the player's skill and desired challenge level.

Preconditions: Game is not in progress and is at the main menu or settings menu.

Trigger: Player navigates to the options menu and selects "Difficulty."

Scenario:
1. Player opens the game's options/settings menu.
2. Player selects the difficulty setting option.
3. Player chooses a difficulty level (Easy, Medium, Hard).
4. System applies the selected difficulty settings to the game.

Exceptions:
1. Player attempts to change difficulty during a game, which is either not allowed or requires restarting.

Priority: Medium

When Available: First release

Frequency of Use: Once per game session or as desired by the player

Channel to Actor: Through the game's menu interface

Secondary Actors: None

Channels to Secondary Actors: None

Open Issues:
1. Whether difficulty level changes should be allowed mid-game.
2. How difficulty levels affect game dynamics (e.g., ghost speed, number of enemies).

Use Case: Touching Water

Primary Actor: Player

Goal in Context: To manage the consequences of the character touching water during gameplay.

Preconditions: Water hazards are present on the game board.

Trigger: The character moves into a cell containing water.

Scenario:
1. Player navigates Pacman towards or into a water cell.
2. System responds to the interaction (e.g., slowing movement).
3. Player adjusts strategy based on the effect of water on the character.

Exceptions: None

Priority: Medium

When Available: From the first release or upon introducing water hazards

Frequency of Use: As often as Pacman encounters water during gameplay

Channel to Actor: Direct interaction through game controls

Secondary Actors: None

Channels to Secondary Actors: None

Open Issues:
1. Strategies for players to mitigate or avoid the negative effects of water.

Use Case: Touching Lava

Primary Actor: Player

Goal in Context: To effectively manage the impact of the character touching Lava cells during the game.

Preconditions: Lava cells are present on the game board.

Trigger: Pacman moves onto a cell containing Java.

Scenario:
1. Player directs the character to a lava cell.
2. System activates the effect of lava on the character (e.g instant loss)
3. The game ends

Exceptions: None

Priority: Medium

When Available: From the first release or upon introducing lava cells

Frequency of Use: As often as the character encounters lava cells during gameplay

Channel to Actor: Direct interaction through game controls

Secondary Actors: None

Channels to Secondary Actors: None

Open Issues:
1. Lava immunity after picking up items with special protection effect