Use Case

Team: 06

Use Case 1: Start New Game

Iteration: 1

Primary Actor: Player
Goal in Context: To initialize a new game session with all elements (map, player, enemies, rewards) loaded and displayed.
Preconditions: Game application is running; player is on the main menu.
Trigger: Player selects “Start Game”.

Scenario:
1. Player selects Start Game.
2. System loads map, walls, rewards, punishments, and start/exit cells.
3. System places player and enemies.
4. System initializes score and timer.
5. HUD appears and the game loop starts.

Exceptions:
1. Map file missing → error and return to menu.
2. Initialization error → terminate game.

Priority: High
When Available: First increment
Frequency: Once per session
Channel: Keyboard and Display
Secondary Actors: GameController, Board, GameTimer

Use Case 2: Move Main Character

Iteration: 1

Primary Actor: Player
Goal in Context: To move the player one cell in a valid direction each tick.
Preconditions: Game running.
Trigger: Player presses arrow key.

Scenario:
1. Player presses arrow key.
2. System checks if target cell is valid.
3. If valid, move character.
4. System checks for collisions with objects.
5. Update screen and tick.

Exceptions:
1. Invalid move → no movement.
2. Game paused → input ignored.

Priority: High
When Available: First increment
Frequency: Continuous
Channel: Keyboard
Secondary Actors: Board, GameController

Use Case 3: Collect Regular Reward

Iteration: 1

Primary Actor: Player
Goal in Context: To collect a regular reward and increase score.
Preconditions: Player moves onto reward cell.
Trigger: Collision with Regular Reward.

Scenario:
1. System detects collision.
2. Remove reward from map.
3. Increase player score.
4. Update HUD and rewards left.
5. Check for win condition.

Exceptions:
1. Reward missing → ignore.
2. Enemy collision same tick → trigger lose.

Priority: High
When Available: First increment
Frequency: Frequent
Channel: Automatic
Secondary Actors: Reward, GameController

Use Case 4: Collect Bonus Reward

Iteration: 1

Primary Actor: Player
Goal in Context: To collect temporary bonus reward and gain extra score.
Preconditions: Bonus Reward active.
Trigger: Player moves onto bonus cell.

Scenario:
1. System spawns Bonus Reward.
2. Player reaches bonus cell.
3. Add reward to score and remove from map.
4. Update HUD.
5. Remove reward if timer expires.

Exceptions:
1. Reward expired → no effect.

2. Pause → timer stops.

Priority: Medium
When Available: Second increment
Frequency: Occasional
Channel: Automatic
Secondary Actors: BonusReward, GameController

Use Case 5: Apply Punishment

Iteration: 1

Primary Actor: Player
Goal in Context: To decrease score when stepping on punishment cell.
Preconditions: Player moves onto punishment.
Trigger: Collision detected.

Scenario:
1. Detect punishment cell.
2. Subtract value from score.
3. Remove punishment.
4. Update HUD.
5. If score < 0 → trigger loss.

Exceptions:
1. Missing punishment data → ignore.
2. Score calc error → revert.

Priority: High
When Available: First increment
Frequency: Occasional
Channel: Automatic
Secondary Actors: GameController, MainCharacter

Use Case 6: Enemy Movement and Collision

Iteration: 1

Primary Actor: System (GameController)
Goal in Context: To move enemies and check for collisions each tick.
Preconditions: Game tick active.
Trigger: End of player move.

Scenario:
1. Move each enemy one cell closer to player.
2. Avoid barriers and walls.
3. Check collision with player.
4. If collision → trigger loss.
5. Continue next tick.

Exceptions:

1. Path blocked → enemy waits.
2. Movement error → skip tick.

Priority: High
When Available: Second increment
Frequency: Continuous
Channel: Automatic
Secondary Actors: Enemy, Board

Use Case 7: Evaluate Game State (Win or Lose)

Iteration: 1

Primary Actor: System
Goal in Context: To determine if the player wins or loses after each action.
Preconditions: Game running.
Trigger: End of movement or score update.

Scenario:
1. System checks score and player/enemy positions.
2. If player on exit and all rewards collected → win.
3. If score < 0 or enemy collision → lose.
4. Stop timer and display result.
5. Offer restart or quit.

Exceptions:
1. False collision → ignore.
2. Data mismatch → re-evaluate.

Priority: High
When Available: First increment
Frequency: Once per game
Channel: Automatic
Secondary Actors: GameController, Timer, Board