

Use Cases

Use case #1: Start Game - Initialize Board

Primary Actor: Player

Goal:

Preconditions: The game is running and the map has been loaded.

Trigger: The player decides to play the game.

Scenario:

1. Player: Launches the game.
2. Player: Observes the board to see that the game has started successfully.

Use-case #2: End Game - Win

Primary Actor: Player

Goal: Display the final score and time when the player wins.

Preconditions: Game is running and has been programmed to recognize when the player has collected all regular rewards.

Trigger: Player reaches the exit.

Scenario:

1. Player:
2. Player: The player observes the final score and time which indicates that the game has been won.

Exception:

1. Player has not collected all regular rewards: player collects all remaining regular rewards and returns to the exit

priority : Essential, must be implemented after the main character movement

When available: first increment

Frequency of use: frequent every time a player wins the game.

Channel to actor: Via in game button

Use-case #3: Pause game

Actor: Player

Goal: pause every process in the game for a short time period

Preconditions: game must be ongoing

Trigger: player decide to pause the game for a while so that it can be continued later in a short time period

Scenario:

1. The player reach and press the pause button
2. The system show pause interface or game has been freeze
3. Player press the pause button again to continue

Exception:

1. Player exit while in the pause state

priority : Essential, must be implemented after the basic gameflow is created (start to end game)
When available: first increment
Frequency of use: infrequent considering the game will be short.
Channel to actor: Via keyboard control

Use-case #4: Player Movement

Primary Actor: Player

Goal: To update the main character's location when the player presses a valid key.

Preconditions: Board has been initialized and the game has been programmed to accept player input, game has started.

Trigger: The player decides to move.

Scenario:

1. The main character can move up/down/left/right to an adjacent cell if there are no barriers on the target cell.
2. The player can move at most one cell at each "tick" of the game.
3. The main character will not move if the player doesn't press a valid key, or if the move is not valid (i.e., a move to a cell that is blocked by a wall or a barrier).

Exceptions:

1. The player presses an invalid key or makes an invalid move: the player's location does not change
2. There can be at most one moving character (main character or moving enemy) on each cell at a time

priority : Essential, must be implemented after the game_init is implemented.
When available: first increment
Frequency of use: every time the player presses the moving button.
Channel to actor: Via keyboard control

Use-case #5: Display Player's change in score

Primary actor: Player

Goal: To view the effect of Player moves to cells with rewards, punishment, and moving enemies.

Preconditions: Board is initialized with enemies, punishments, and regular rewards.
Enemies will move toward the position of player.

Trigger: The player decides to move on board.

Scenario:

1. The player is on a cell of the board within the walls (barriers).
2. The player passes a cell with regular reward - score displayed on screen increases by 1.
3. The player passes a cell with a bonus reward - score displayed on screen increases by 2.
4. The player passes a cell with punishment (4x punishment) - score displayed on screen decreases by 4.

5. Total score of the player is below zero, the player loses and the game ends.

Exceptions:

1. Player/Moving enemies cannot go through walls/barriers
2. Moving enemies passing through cells with rewards or punishments does not affect - does not change the overall score of the player.
3. If the player moves to a cell containing a moving enemy, or the enemy catches up to the player, the game ends and the player loses.
4. Different punishments contain different amounts of scores to deduct.
5. Regular reward scores 1 point.
6. Bonus reward contains more points, but it appears randomly and will disappear after a few seconds.

Priority: Essential, must be implemented after basic movement functions are implemented.

When available: First increment

Frequency of use: Many times per game

Channel to actor: Via keyboard control