

Use Case ID: UC1

Use Case Name: Start of game

Iteration : 1 last modification: Oct 24 by K.Singh, A.Mondair, K.Khubbar, P.Singh

Actors: Player

Goal in context: To start the game and menu is displayed and all characters, maps, keys, barriers are initialized

Preconditions: Game starts

Trigger: The player chooses to press “Start Game” option from main menu

Scenario:

1. Player chooses to launch game application
2. The system displays the main menu with “Start game: and “Exit” options
3. The player selects “Start Game”
4. The game loads up the themed map (Hotel Lobby)
5. The system puts the player, NPC, enemies in their starting position
6. The player starts with \$0 and 5 lives
7. The system enables the gameboard and enables movement controls
8. Money(score), life counter, and map UI appear on screen

Exceptions:

1. Player chooses Exit (Start game and Exit)
2. Missing asset or map file

Priority: High (Starts the game)

When available: First Increment

Frequency of use: Frequent

Chanel to Actor: Via keyboard and monitor

Secondary Actor: System hardware and game engine

Chanel to Secondary Actors: Game engine: software initialization

Hardware: file loading

Use Case ID: UC2

Use Case Name: Move Main Character

Iteration: 1 last modification: Oct 24 by K.Singh, A.Mondair, K.Khubbar, P.Singh

Actors: Player

Goal in context: To move player around hotels rooms/hallways (depends on keyboard input)

Preconditions: Game should be started, Game should not be paused, Player should be on gameboard

Trigger: Player presses an arrow key or (WASD)

Scenario:

1. The player presses arrow key (up down left right) or WASD
2. The system takes in keyboard response
3. System checks if the following room is in range
4. System checks if there is any boundary
5. If everything valid the player gets updated to the new room
6. System checks for any keys on that following room
7. Ghost moves closer when player moves
8. If none interactions happen the game continues

Exceptions:

1. If player presses any other key – input ignored
2. If target room blocked – input ignored again

Priority: High – Core game function

When available – First increment

Frequency: Continuous (throughout the game)

Chanel to actor: Keyboard input from players device

Secondary actor: ghost, game board

Channels to secondary actor:

Game board: updates players location/coordinates

Ghost: updates own location when player moves

Use Case ID: UC3

Use Case Name: Collect Key

Iteration: 1 last modification: Oct 24 by K.Singh, A.Mondair, K.Khubbar, P.Singh

Actors: Player

Goal in context: To collect keys when ghost hunter interacts with the room containing the key and updating the score or unlocking new areas.

Preconditions:

Game should be running

Player should be on the gameboard

At least one key on screen

Trigger: The player walks to the room containing the key

Scenario:

1. The player moves to the room containing the key
2. System knows if player interacted with the key
3. Player gets a point, and money goes up
4. Key gets removed from the screen so not collectable again
5. Score updates and game continues

Exceptions:

1. If room does not have key score wont update
2. If key is already collected (same one can't be collected again)

Priority: High – Needed to win the game

When available – First increment

Frequency: multiple times (throughout the game)

Chanel to actor: Interface displayed on players screen

Secondary actor: game board, score

Channels to secondary actor:

Game board: updates and removes key from board

score: updates and goes up 1

Use Case ID: UC4

Use Case Name: Hit Punishment

Iteration: 1 last modification: Oct 24 by K.Singh, A.Mondair, K.Khubbar, P.Singh

Actors: Player

Goal in context: To penalize player when they hit the cursed room, and it decreases their health or score ending the game

Preconditions: Game should be started, and player is alive on the gameboard

Trigger: The player moves onto a cursed room

Scenario:

1. The player moves to the room that contains punishment
2. System will detect punishment
3. Score will decrease or can end the game depending on health
4. System will remove the punishment from the board
5. System detects if health under 0
6. Score updates and game will continue if not dead

Exceptions:

1. If room doesn't have punishment (no deduction)
2. If punishment already has been collected
3. If punishment interface doesn't load

Priority: Medium – affects gameplay and risks players lives

When available – after key and game functionality

Frequency: Occasional, whenever stepped on cursed room

Chanel to actor: Game screen/gameboard

Secondary actor: Game board and scoring

Channels to secondary actor:

Gameboard – Punishment interface goes away

Scoring – score goes down

Use Case ID: UC5

Use Case Name: Ghost collision

Iteration: 1 last modification: Oct 24 by K.Singh, A.Mondair, K.Khubbar, P.Singh

Actors: Player

Goal in context: To detect if player's characters collides with an ghost and trigger a loss and ending the game

Preconditions: Game should be started, and player is alive on the gameboard and so is the ghost

Trigger: An ghost moves onto a player's room or the player moves onto a room containing the ghost

Scenario:

1. The player or ghost move s to a new room
2. System checks if player and ghost are on the same room
3. If both characters match it results in damage
4. If health < 0 "Game over" displays

Exceptions:

1. If player and ghost rooms are diagonal (adjacent) no Collison

Priority: High – changes the whole outcome of the game

When available – after game functionality and key

Frequency: Rare, whenever player and ghost step on the same room

Chanel to actor: Game interface

Secondary actor: Game board and scoring

Channels to secondary actor:

Gameboard – validates room Collison

Scoring – score goes down

Use Case ID: UC6

Use Case Name: Win game

Iteration: 1 last modification: Oct 24 by K.Singh, A.Mondair, K.Khubbar, P.Singh

Actors: Player

Goal in context: To complete game by defeating the boss ghost in the basement after collecting the 2 required keys

Preconditions: Game should be started, basement door is unlocked, boss is active

Trigger: Player defeats the boss by bringing the ghost boss to 0

Scenario:

1. The player collects both keys from earlier levels
2. The system unlocks basement door
3. Player enters basement and fights
4. Player attacks until boss health is 0
5. System stops the game and initializes boss's defeat
6. The game displays "You win"
7. Player returns to main menu and prompted to restart

Exceptions:

1. If player doesn't have 2 keys access to basement denied
2. If boss defeats player you lose

Priority: High – motive of the game

When available – Final increment

Frequency: Once per successful game session

Chanel to actor: Game display and input

Secondary actor: Boss and scoring

Channels to secondary actor:

Boss – Handles fighting logic

Scoring – Win game

Use Case ID: UC7

Use Case Name: Lose game

Iteration: 1 last modification: Oct 24 by K.Singh, A.Mondair, K.Khubbar, P.Singh

Actors: Player

Goal in context: To handle the scenario where player loses the game by getting defeated by the boss ghost in the basement

Preconditions: Game should be started, player needs at least 1 live

Trigger: Player lives count reaches 0

Scenario:

1. Player is hit by a ghost or punishment
2. The system decreases the life counter
3. If 0 lives player dies
4. System displays a “Game Over” screen
5. Player can choose to restart

Exceptions:

1. If player has more than 0 lives (still alive)

Priority: High – game resets and you lose

When available – Core (Whole game) from first increment

Frequency: Once per failed game session

Channel to actor: Game screen

Secondary actor: Game engine and scoring

Channels to secondary actor:

Game engine – Engine freezes and you lose

Scoring – Scoring shows final results