Use Cases

Arda Cifci, Nayeong Lee, Ellie Neufeld, Cole Thacker

Use case: Start Game

Primary actor: Player

Goal in context: To start the game and display the playable map.

Preconditions: The Player has access to or is currently on to the main menu.

Trigger: Player presses the start button on the main menu interface and chooses a difficulty.

Scenario :

1. Player: launches the video game.
2. Player: The game launches into the main menu which has multiple different buttons.
3. Player: The Player presses the start game button.
4. Player: The Player chooses a difficulty by pressing a button.
5. Player: The game's system/engine changes the game's state from "idle" to "running" and starts the game.

Exceptions:

1. Player could not launch the video game.
2. The game's state is stuck in "running" from a previous game start so a new game cannot be started.

Priority: Essential, must be implemented.

When available: Earliest possible time in development, first increment/version.

Frequency of use: Every time the player wants to play the game.

Channel to actor: Via the main menu interface.

Secondary actors: Computer hardware (Mouse, Keyboard, Etc).

Channels to secondary actors: Wires/Cables to computer software.

Open issues:

1. Should there be another start game button before switching to the "running" state of the game or should the game auto start after selecting a difficulty?
2. Should there be a quick start game option that saves the previous difficulty?

Use case: Display Help interface

Primary actor: Player

Goal in context: To display the instructions on how to play the video game to the player.

Preconditions: The Player has access to or is currently on to the main menu.

Trigger: The Player presses the Help button the main menu interface.

Scenario :

1. Player: Launches the video game.
2. Player: The game launches into the main menu which has multiple different buttons.
3. Player: Presses the “Help” button in the main menu interface.

Exceptions:

1. Player could not launch the video game.

Priority: Low, Can be implemented later in development.

When available: Near end of development, first increment/version.

Frequency of use: A few times per player in total.

Channel to actor: Via the main menu interface.

Secondary actors: Computer hardware (Mouse, Keyboard, Etc).

Channels to secondary actors: Wires/Cables to computer software.

Open issues:

1. Do we need to have this interface or can we display the help information in the main menu or another part of the game instead?

Use case: Exit Game

Primary actor: Player

Goal in context: To exit out of the game.

Preconditions: The game has launched and is in the main menu interface.

Trigger: The player presses the exit button in the main menu interface.

Scenario :
1. Player: launches the video game.
2. Player: The game launches into the main menu which has multiple different buttons.
3. Player: Presses the Exit button in the main menu interface.

Exceptions:
1. Player could not launch the video game.

Priority: Essential, must be implemented.

When available: Earliest possible time in development, first increment/version.

Frequency of use: Everytime the game is opened.

Channel to actor: Via the main menu interface.

Secondary actors: Computer hardware (Mouse, Keyboard, Etc).

Channels to secondary actors: Wires/Cables to computer software.

Open issues:
1. Do we need an exit button in the main menu interface or should it be placed in the end game interface (Win/ Game Over).

Use case: Choose game difficulty.

Primary actor: Player

Goal in context: To choose the games "running" state difficulty.

Preconditions: The Player has access to or is in the games main menu interface.

Trigger: Player presses one of three difficulty button options.

Scenario :

1. Player: launches the video game.
2. Player: The game launches into the main menu which has multiple different buttons.
3. Player: Chooses the start game button.
4. Player: Picks a difficulty and presses the corresponding button.

Exceptions:

1. Player picks a different button than the start game in the main menu interface.
2. Player is already in the “running” state of the game, thus can’t pick a difficulty.

Priority: Medium, After all essential parts have been implemented.

When available: After implementing the essentials and a working (non difficulty changing) version is implemented**,** first increment/version.

Frequency of use: Everytime the game starts (turns to running state).

Channel to actor: Via the choose difficulty interface.

Secondary actors: Computer hardware (Mouse, Keyboard, Etc).

Channels to secondary actors: Wires/Cables to computer software.

Open issues:

1. Should there be another start game button before switching to the “running” state of the game or should the game auto start after selecting a difficulty?

Use case: Pause Game

Primary actor: Player

Goal in context: To pause the game when playing the game.

Preconditions: The game is in the “running” (actively playing) state.

Trigger: The player presses the pause button.

Scenario :

1. Player: launches the video game.
2. Player: The game launches into the main menu which has multiple different buttons.
3. Player: The Player presses the start game button.
4. Player: The Player chooses a difficulty by pressing a button.
5. Player: The game's system/engine changes the game’s state from “idle” to “running” and starts the game.
6. Player: Moves the mouse cursor over to the pause button.
7. Player: Presses the pause button.

Exceptions:

1. The player doesn’t start the game (is in the main menu), thus can’t pause the game.
2. Player couldn't launch the game.

Priority: Low, Can be implemented later in development.

When available: Near end of development but before the “Help” related stuff has been implemented, first increment/version.

Frequency of use: A few to no times per game session.

Channel to actor: Via in game interface while game is in “running” state.

Secondary actors: Computer hardware (Mouse, Keyboard, Etc).

Channels to secondary actors: Wires/Cables to computer software.

Open issues:

1. Do we also implement a pause key bind so the player does not have to press the pause button with their mouse?

Use case: Display Help interface via pause interface.

Primary actor: Player

Goal in context: To display the instructions on how to play the video game to the player.

Preconditions: The player is in the paused interface.

Trigger: The player presses the “Help” button in the paused interface.

Scenario :

1. Player: launches the video game.
2. Player: The game launches into the main menu which has multiple different buttons.
3. Player: The Player presses the start game button.
4. Player: The Player chooses a difficulty by pressing a button.
5. Player: The game's system/engine changes the game’s state from “idle” to “running” and starts the game.
6. Player: Presses the pause game button.
7. Player: Presses the Help button.

Exceptions:

1. Player couldn't launch the game.
2. Player never starts the game (game in “running” state), thus can’t pause and then resume.

Priority: Low, Can be implemented at the end of development.

When available: After all Medium and essential priority functions have been added, first increment/version.

Frequency of use: A few times per player in total.

Channel to actor: Via the pause game interface.

Secondary actors: Computer hardware (Mouse, Keyboard, Etc).

Channels to secondary actors: Wires/Cables to computer software.

Open issues:

1. Do we need to put another Help me button or should we let the player return to the main menu to press the Help me button.

Use case: Return to Main Menu via pause interface.

Primary actor: Player

Goal in context: To return to the main menu.

Preconditions: The game is in the “running” state and the player pauses.

Trigger: The player presses the Main menu button in the paused interface.

Scenario :

1. Player: launches the video game.
2. Player: The game launches into the main menu which has multiple different buttons.
3. Player: The Player presses the start game button.
4. Player: The Player chooses a difficulty by pressing a button.
5. Player: The game's system/engine changes the game’s state from “idle” to “running” and starts the game.
6. Player: Presses the pause game button.
7. Player: Presses the Main Menu button.

Exceptions:

1. Player couldn't launch the game.
2. Player never starts the game (game in “running” state), thus can’t pause and then press the main menu button.

Priority: Low, Can be implemented later in development.

When available: After all Medium and essential priority functions have been added, first increment/version.

Frequency of use: few times per game session.

Channel to actor: Via the pause game interface.

Secondary actors: Computer hardware (Mouse, Keyboard, Etc).

Channels to secondary actors: Wires/Cables to computer software.

Open issues:

1. Do we put a return to the main menu in the pause interface or on the board interface?

Use case: Resume Game

Primary actor: Player

Goal in context: To resume the game from a paused state.

Preconditions: The game is in the “running” (actively playing) state and is paused.

Trigger: The player presses the Resume button.

Scenario :
1. Player: is actively playing the game in the “running state”.
2. Player: Moves the mouse cursor over to the pause button.
3. Player: Presses the pause button.
4. Player: Presses the resume button.

Exceptions:
1. The player doesn’t start the game (is in the main menu), thus can’t pause the game.
2. Player couldn't launch the game.

Priority: Low, Can be implemented later in development.

When available: Near end of development, first increment/version.

Frequency of use: A few to no times per game.

Channel to actor: Via in game interface while game is in “running” state.

Secondary actors: Computer hardware (Mouse, Keyboard, Etc).

Channels to secondary actors: Wires/Cables to computer software.

Open issues:
1. Do we also implement a resume key bind so the player does not have to press the resume button with their mouse?

Use case: Player Movement (WASD)

Primary actor: Player

Goal in context: To move the player into a new cell.

Preconditions: The Player is playing the game in the “running” state.

Trigger: The Player presses the W,A,S, or D key.

Scenario :
1. Player: Launches game.
2. Player: Presses start game on the Main Menu.
3. Player: Chooses a difficulty and presses the corresponding button.
4. Player: Enters the “running” states of the game.
5. Player: Presses a movement key to move into a new cell.

Exceptions:
1. The player tries to move into a wall (barrier) cell.
2. The player does not have a keyboard or similar software emulating a keyboard.
3. Game is paused.

Priority: Essential, must be implemented.

When available: Early in development, First version/Increment

Frequency of use: Many times per game. Very frequent.

Channel to actor: Via computer's Keyboard (WASD key).

Secondary actors: Computer hardware (Except keyboard)..

Channels to secondary actors: Wires/Cables to computer software.

Open issues:
1. Do we want W,A,S, and D to be our movement or do we want to move with arrow keys?

Use case: Return to Main Menu via end game interfaces.

Primary actor: Player

Goal in context: To return to the main menu.

Preconditions: The game is in the “running” state and the player wins or loses.

Trigger: The player presses the Main menu button in the Game Over / Win interface.

Scenario :

1. Player: Launches the video game.
2. Player: The game launches into the main menu which has multiple different buttons.
3. Player: The Player presses the start game button.
4. Player: The Player chooses a difficulty by pressing a button.
5. Player: The game's system/engine changes the game’s state from “idle” to “running” and starts the game.
6. Player: Plays the game and wins or loses, resulting in one of the end game interfaces displaying.
7. Player: Presses the Main Menu button.

Exceptions:

1. Player couldn't launch the game.
2. Player stays paused and never wins or loses thus can’t get to the end game screen.

Priority: Medium, Can be implemented later in development but before Low priority implementations.

When available: Implemented first in the end game interface, first increment/version.

Frequency of use: few times per game session.

Channel to actor: Via an end game interface

Secondary actors: Computer hardware (Mouse, Keyboard, Etc).

Channels to secondary actors: Wires/Cables to computer software.

Open issues:

1. Do we put a return to the main menu in the end game interfaces or on the board interface?

Use case: Replay Game

Primary actor: The Player

Goal in context: To restart the game's "running" state from the beginning.

Preconditions: The Player is in one of the end game interfaces.

Trigger: The player presses the restart game button.

Scenario :

1. Player: launches the video game.
2. Player: The game launches into the main menu which has multiple different buttons.
3. Player: The Player presses the start game button.
4. Player: The Player chooses a difficulty by pressing a button.
5. Player: The game's system/engine changes the game's state from "idle" to "running" and starts the game.
6. Player: Plays the game and wins or loses, resulting in one of the end game interfaces displaying.
7. Player: Presses the replay button.

Exceptions:

1. The Player presses the exit button instead.
2. The Player never wins or loses and can't reach the replay button.
3. Player never starts to play the game in the "running" state.

Priority: Medium, can be implemented later in development.

When available: After the Main menu button in the interface has been added.

Frequency of use: Few to many times per session. Every time the player wants to restart the level.

Channel to actor: Via an end game interface (Win/Game Over).

Secondary actors: Computer hardware (Mouse, Keyboard, Etc).

Channels to secondary actors: Wires/Cables to computer software.

Open issues:

1. Should the difficulty selector screen be displayed again after pressing the replay button?

Use case: Display ‘How to Play’ interface via one of the end game interfaces.

Primary actor: Player

Goal in context: To display the instructions on how to play the video game to the player.

Preconditions: The player is in one of the end game interfaces (Win / Game Over)

Trigger: The player presses the “Help” button in the Game Over / Win interface.

Scenario :

1. Player: launches the video game.
2. Player: The game launches into the main menu which has multiple different buttons.
3. Player: The Player presses the start game button.
4. Player: The Player chooses a difficulty by pressing a button.
5. Player: The game's system/engine changes the game’s state from “idle” to “running” and starts the game.
6. Player: Plays the game and wins or loses, resulting in one of the end game interfaces displaying.
7. Player: Presses the Help button.

Exceptions:

1. Player couldn't launch the game.
2. Player stays paused and never wins or loses.
3. Player presses the Replay or Menu button instead.

Priority: Low, Can be implemented at the end of development.

When available: After all Medium and essential priority functions have been added, first increment/version.

Frequency of use: A few times per player in total.

Channel to actor: Via an end game (Win / Game Over) interface.

Secondary actors: Computer hardware (Mouse, Keyboard, Etc).

Channels to secondary actors: Wires/Cables to computer software.

Open issues:

1. Do we need to put another Help me button or should we let the player return to the main menu to press the Help me button.