

Use case: StartFromMainMenu

Primary actor: Player

Goal in context: To start the game from the main menu.

Preconditions: The player is at the main menu window, and therefore the game has not been started.

Trigger: The player clicks on the “Start” icon with their mouse.

Scenario:

1. Player: observes the Main menu, decides they want to start the game.
2. Player: clicks on the “Start” icon
3. Player: observes the game starting

Exceptions: N/A

Priority: Not essential.

When available: Third increment.

Frequency of use: Once per game.

Channel to actor: mouse.

Secondary actors: N/A

Channels to secondary actors: N/A

Open issues:

1. Should there be a countdown before the game starts?

Use case: GetHelp

Primary actor: Player

Goal in context: To open the Help window.

Preconditions: The game has not started or is paused.

Trigger: The player clicks on the “How to Play” icon in the Main menu or on the “?” icon in the Pause window with their mouse.

Scenario:

1. Player: observes the Main menu or Pause window, decides they want to get help.
2. Player: clicks on the “How to Play” or “?” icon.
3. Player: observes the Help window.
4. Player: clicks on the “X” icon to go back to the previous window.

Exceptions: N/A

Priority: Not essential.

When available: Third increment.

Frequency of use: 0 to several times per game.

Channel to actor: mouse.

Secondary actors: N/A

Channels to secondary actors: N/A

Open issues: N/A

Use case: QuitProgram

Primary actor: Player

Goal in context: To quit the program.

Preconditions: The game has not started and therefore is at the Main menu.

Trigger: The player clicks on the “Quit” icon with their mouse.

Scenario:

1. Player: observes the Main menu, decides they want to quit the program
2. Player: clicks on the “Quit” icon.
3. Player: observes the program stop running and the game closing.

Exceptions: N/A

Priority: Not essential.

When available: Third increment.

Frequency of use: Once per game.

Channel to actor: mouse.

Secondary actors: N/A

Channels to secondary actors: N/A

Open issues:

1. Should we have text that clearly explains the difference between quitting and exiting?
2. Should we be able to quit from the Pause, Win, or Lose windows?

Use case: MoveUp

Primary actor: Player

Goal in context: To move Rock up on the game board by one cell per tick.

Preconditions: The game has been started, is not on the Pause window, the player has not won or lost the game, and the game has been programmed to recognize “W” as an input.

Trigger: The player presses “W” on their keyboard.

Scenario:

1. Player: observes the game board, decides they want to move Rock up.
2. Player: enters “W” on their keyboard
3. Player: observes Rock move up the game board by one cell per tick.

Exceptions:

1. The cell above Rock is filled with a barrier or wall; do not move Rock.

Priority: Essential; must be implemented

When available: First increment

Frequency of use: $\sim 1/4$ of total ticks per game.

Channel to actor: keyboard

Secondary actors: N/A

Channels to secondary actors: N/A

Open issues: N/A

Use case: MoveDown

Primary actor: Player

Goal in context: To move Rock down on the game board by one cell per tick.

Preconditions: The game has been started, is not on the Pause window, the player has not won or lost the game, and the game has been programmed to recognize “S” as an input.

Trigger: The player presses “S” on their keyboard.

Scenario:

1. Player: observes the game board, decides they want to move Rock down.
2. Player: enters “S” on their keyboard
3. Player: observes Rock move down the game board by one cell per tick.

Exceptions:

1. The game is on the main menu, Help, Pause, Win, or Lose window; do not move Rock.

Priority: Essential; must be implemented

When available: First increment

Frequency of use: $\sim 1/4$ of total ticks per game.

Channel to actor: keyboard

Secondary actors: N/A

Channels to secondary actors: N/A

Open issues: N/A

Use case: MoveLeft

Primary actor: Player

Goal in context: To move Rock left on the game board by one cell per tick.

Preconditions: The game has been started, is not on the Pause window, the player has not won or lost the game, and the game has been programmed to recognize “A” as an input.

Trigger: The player presses “A” on their keyboard.

Scenario:

1. Player: observes the game board, decides they want to move Rock left.
2. Player: enters “A” on their keyboard
3. Player: observes Rock move left on the game board by one cell per tick.

Exceptions:

1. The cell to the left of Rock is filled with a barrier or wall; do not move Rock.

Priority: Essential; must be implemented

When available: First increment

Frequency of use: $\sim 1/4$ of total ticks per game.

Channel to actor: keyboard

Secondary actors: N/A

Channels to secondary actors: N/A

Open issues: N/A

Use case: MoveRight

Primary actor: Player

Goal in context: To move Rock right on the game board by one cell per tick.

Preconditions: The game has been started, is not on the Pause window, the player has not won or lost the game, and the game has been programmed to recognize “D” as an input.

Trigger: The player presses “D” on their keyboard.

Scenario:

1. Player: observes the game board, decides they want to move Rock right.
2. Player: enters “D” on their keyboard
3. Player: observes Rock move right on the game board by one cell per tick.

Exceptions:

1. The cell to the right of Rock is filled with a barrier or wall; do not move Rock.

Priority: Essential; must be implemented

When available: First increment.

Frequency of use: $\sim 1/4$ of total ticks per game.

Channel to actor: keyboard

Secondary actors: N/A

Channels to secondary actors: N/A

Open issues: N/A

Use case: PauseGame

Primary actor: Player

Goal in context: To open the Pause window.

Preconditions: The game has started, the player has not won or lost yet, and the game is not paused.

Trigger: The player clicks on the “Pause” icon or presses “P” on their keyboard.

Scenario:

1. Player: observes the game board, decides they want to pause the game.
2. Player: clicks on the “Pause” icon or presses the “P” key.
3. Player: observes the Pause window.
4. Player: clicks on the “Resume” icon to un-pause the game, “?” to go to the Help window, “Restart” to restart the game, or “Exit” to go to the Main menu.

Exceptions: N/A

Priority: Important but not essential; implement if at all possible

When available: Second increment.

Frequency of use: 0 to several times per game.

Channel to actor: mouse and keyboard.

Secondary actors: N/A

Channels to secondary actors: N/A

Open issues: N/A

Use case: ResumeGame

Primary actor: Player

Goal in context: To resume a paused game.

Preconditions: The game has already started, the player has not won or lost the game, and the game is paused.

Trigger: The player clicks the “Resume” icon with their mouse

Scenario:

1. Player: observes the Pause window, decides they want to resume the game.
2. Player: clicks the “Resume” icon with their mouse.
3. Player: observes the game resume.

Exceptions: N/A

Priority: Important but not essential; implement if at all possible.

When available: Second increment.

Frequency of use: 0 to several times per game.

Channel to actor: mouse.

Secondary actors: N/A

Channels to secondary actors: N/A

Open issues: N/A

Use case: GetReward

Primary actor: Player

Goal in context: To increase the player’s score.

Preconditions: The game has already started, the player has not won or lost the game, and the game is not paused.

Trigger: The player uses their keyboard to move Rock into a cell occupied by a reward.

Scenario:

1. Player: observes the game window, decides they want to move Rock into a cell occupied by a reward.
2. Player: moves Rock as detailed in MoveUp, MoveDown, MoveLeft, and MoveRight use cases.
3. Player: observes their score increasing and the reward disappearing.

Exceptions: N/A

Priority: Essential; must be implemented.

When available: First increment.

Frequency of use: 0 to X times per game.

Channel of actor: keyboard

Secondary actors: N/A

Channels to secondary actors: N/A

Open issues: N/A

Use case: GetPunishment

Primary actor: Player

Goal in context: To decrease the player's score.

Preconditions: The game has already started, the player has not won or lost the game, and the game is not paused.

Trigger: The player uses their keyboard to move Rock into a cell occupied by a punishment.

Scenario:

1. Player: observes the game window, accidentally (or maybe on purpose) decide they want to move Rock into a cell occupied by a punishment.
2. Player: moves Rock as detailed in MoveUp, MoveDown, MoveLeft, and MoveRight use cases.
3. Player: observes their score decreasing and the punishment disappearing.

Exceptions: N/A

Priority: Essential; must be implemented

When available: First increment.

Frequency of use: 0 to X times per game.

Channel of actor: keyboard

Secondary actors: N/A

Channels to secondary actors: N/A

Open issues: N/A

Use case: WinGame

Primary actor: Player

Goal in context: to win the game?

Preconditions: The game has already started, the player has collected all of the regular rewards, they have moved Rock to the end cell, Rock has not died, and the player's score is greater than or equal to 0.

Trigger: The player uses their keyboard and moves Rock into the end cell.

Scenario:

1. Player: observes the game window, decides they want to move Rock into the end cell.
2. Player: moves Rock as detailed in MoveUp, MoveDown, MoveLeft, and MoveRight use cases.
3. Player: observes their game ending and opening the Win window.

Exceptions:

1. An enemy moves into the end cell on the same tick that the player moves Rock into the end cell: Rock dies and the player loses the game.

Priority: Essential; must be implemented

When available: First increment.

Frequency of use: 0-1 times per game.

Channel of actor: keyboard

Secondary actors: N/A

Channels to secondary actors: N/A

Open issues: N/A

Use case: LoseGame

Primary actor: Player

Goal in context: to lose the game?

Preconditions: The game has already started, and Rock has died or the player's score is less than zero.

Trigger: The player uses their keyboard and moves Rock into an enemy or punishment.

Scenario:

1. Player: observes the game window, decides they want to move Rock into a cell occupied by a punishment or enemy.
2. Player: moves Rock as detailed in MoveUp, MoveDown, MoveLeft, and MoveRight use cases.
3. Player: observes their score decreasing or Rock dying, their game ending, and the Lose window opening.

Exceptions:

1. An enemy moves into the same cell the player moves Rock into on the same tick: Rock dies.

Priority: Essential; must be implemented

When available: First increment.

Frequency of use: 0-1 times per game.

Channel of actor: keyboard

Secondary actors: N/A

Channels to secondary actors: N/A

Open issues: N/A

Use case: RestartGame

Primary actor: Player

Goal in context: to restart the game.

Preconditions: the player is on the Pause, Lose, or Win window.

Trigger: the player clicks on the "Restart" icon in the Pause window, the "Try Again" icon in the Lose window, or the "Continue" icon in the Win window with their mouse.

Scenario:

1. Player: observes the Pause, Lose, or Win window, decides they want to restart the game.
2. Player: clicks on the "Restart" icon in the Pause window, the "Try Again" icon in the Lose window, or the "Continue" icon in the Win window with their mouse.
3. Player: observes the game restarting.

Exceptions: N/A

Priority: Important but not essential; implement if at all possible.

When available: Second increment.

Frequency of use: 0 to several times per program.

Channel to actor: Mouse.

Secondary actors: N/A

Channels to secondary actors: N/A

Open issues: N/A

Use case: ExitGame

Primary actor: Player

Goal in context: to exit the game.

Preconditions: the player is on the Pause, Lose, or Win window.

Trigger: the player clicks on the “Exit” icon in the Pause, Lose, or Win window with their mouse.

Scenario:

1. Player: observes the Pause, Lose, or Win window, decides they want to exit the game.
2. Player: clicks on the “Exit” icon in the Pause, Lose, or Win window with their mouse.
3. Player: observes the game returning to the Main menu.

Exceptions: N/A

Priority: Not essential.

When available: Third increment.

Frequency of use: 0 to several times per program.

Channel to actor: Mouse.

Secondary actors: N/A

Channels to secondary actors: N/A

Open issues:

1. Should we have text that clearly explains the difference between quitting and exiting?