

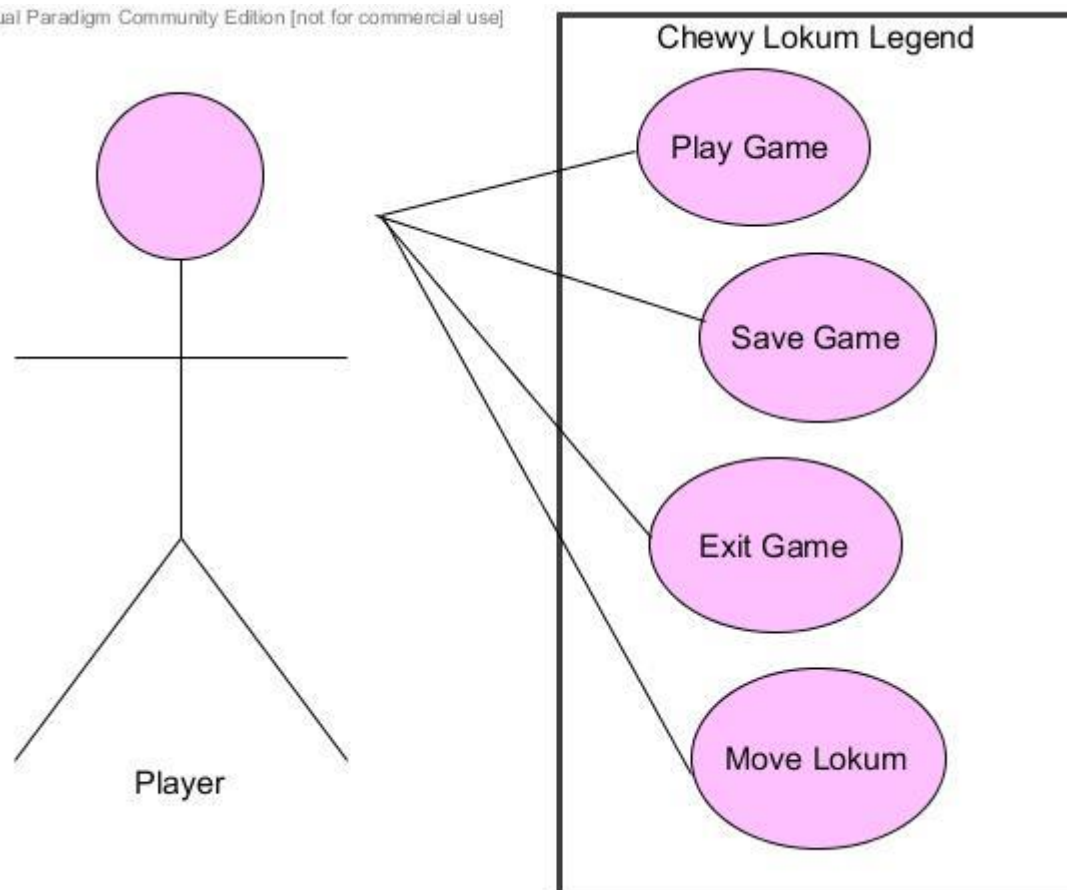
Requirements of the Chewy Lokum Legend

Phase 2

Naz Battal

- Use Case Diagram

Visual Paradigm Community Edition [not for commercial use]



• Use Case Narratives

Use Case UC1: Play Game

Primary Actor: Player

Stakeholders and Interest:

- Player: Wants to start playing the game without any errors.

Preconditions: The game is ready to start.

Success Guarantee (Postconditions): Game is loaded.

Main Success Scenario (Basic Flow):

1. The player opens the game.
2. System opens a frame which includes "new game" and "continue" buttons.
3. The player clicks the new game.
4. System loads the game with given state.

Extensions (Alternative Flows):

3a. Player clicks the “continue” button to continue playing from the last saved state of the game.

1. System loads the saved score, board state and remaining moves for the level.

Frequency of Occurrence: As frequently as the player wants.

Use Case UC2: Save Game

Primary Actor: Player

Stakeholders and Interests:

- Player: Wants to save the state of a game at any level without any errors.

Preconditions: Game is already started.

Success Guarantee (Postconditions): Relevant game state is saved.

Main Success Scenario (Basic Flow):

1. User clicks the save button for saving his current state.
2. System saves the current score, current board state and remaining moves.
3. System returns to the main screen.

Special Requirements:

- XML schema to record game state.

Frequency of Occurrence: As frequently as the player wants.

Use Case UC3: Exit Game

Primary Actor: Player

Stakeholders and Interests:

- Player: Wants to stop playing the game.

Preconditions: Game is already started.

Success Guarantee (Postconditions): Game is closed with/without saving game.

Main Success Scenario (Basic Flow):

1. Player clicks the exit game button to quit the game.
2. System sends an alert whether user wants to save the game or not.
3. Player clicks “don’t save” button.
4. System returns to the main screen.

Extensions (Alternative Flows):

3a. Include “Save Game”

Frequency of Occurrence: As frequently as the player wants.

USE CASE UC4: Move Lokum

Primary Actor: Player

Stakeholders and Interest:

- Player: Wants to move the lokums while playing the game.

Preconditions: The game is already started.

Success Guarantee (Postconditions): The chosen lokums’ positions are swapped positions.

Main Success Guarantee:

1. Player clicks on any lokum on the game screen.
2. System saves that clicked lokum.
3. Player clicks another lokum on the board.

4. System saves that clicked lokum.
5. System swaps the coordinates of chosen lokums.
6. Three or more lokums of the same color is in line vertically or horizontally so they are destroyed.
7. The lokums above are fallen needed number of square to fill in the empty space by the system.
8. System fills the empty space which is open up at the top of the board because of the falling pieces with randomly selected lokums.

System repeats iterations after step 5 (according to lokums conditions' the system can continue along alternative flows).

Extensions (Alternative Flows):

5a: The chosen lokums are not adjacent to each other, so the system doesn't swap the coordinates of lokums.

6a: Three or more same colored lokums are not in a line vertically or horizontally so the lokums are not destroyed.

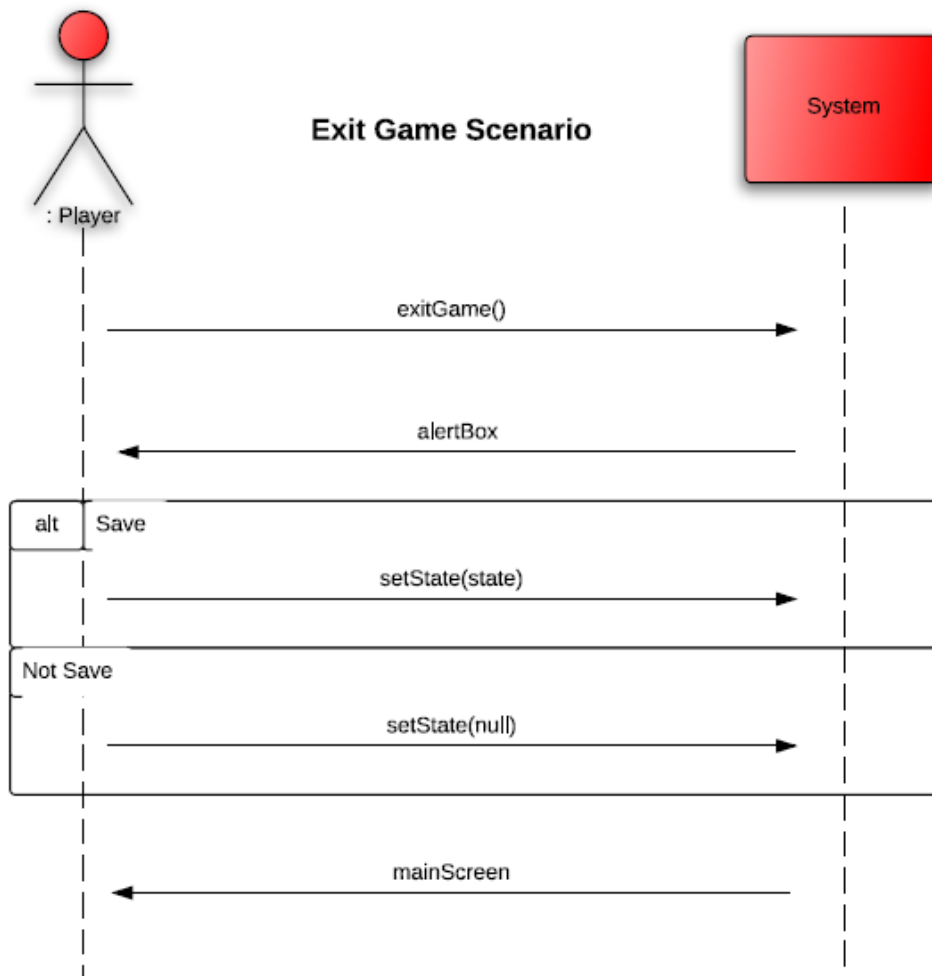
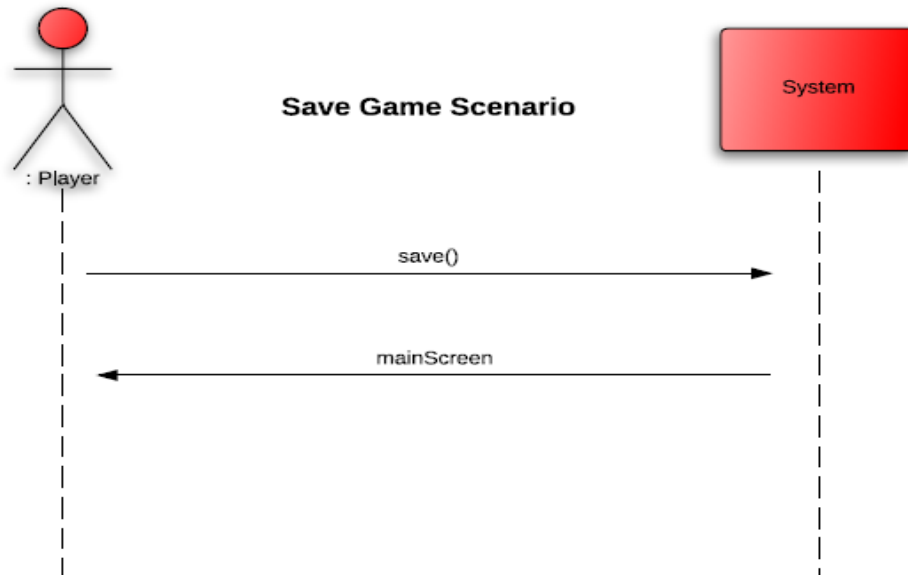
1. Swapped lokums go back to their first places.

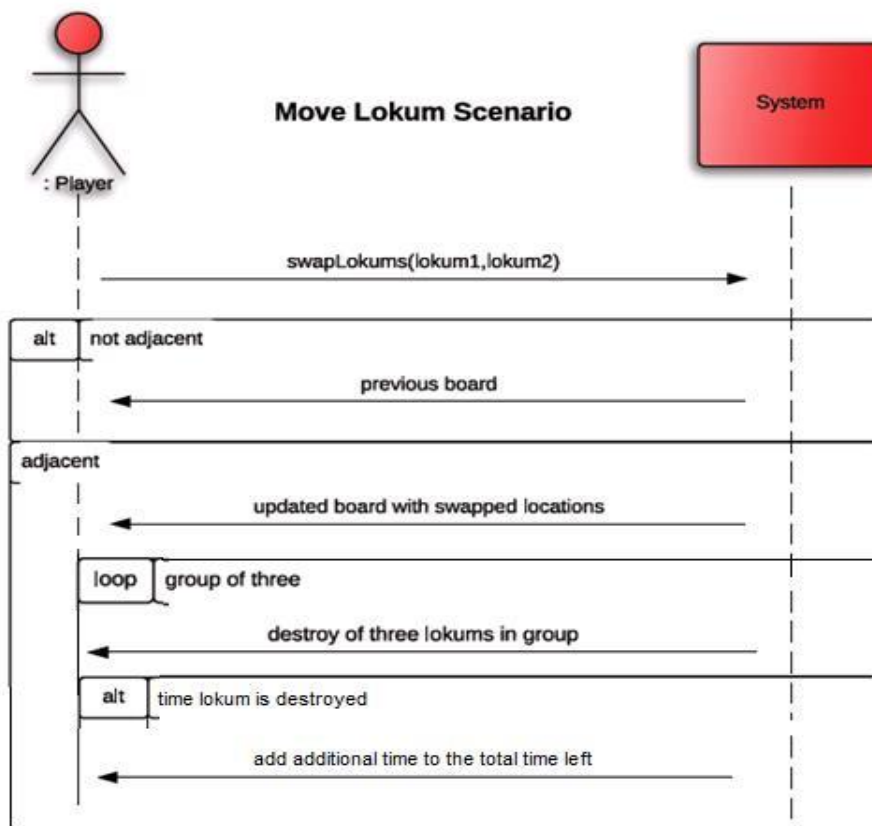
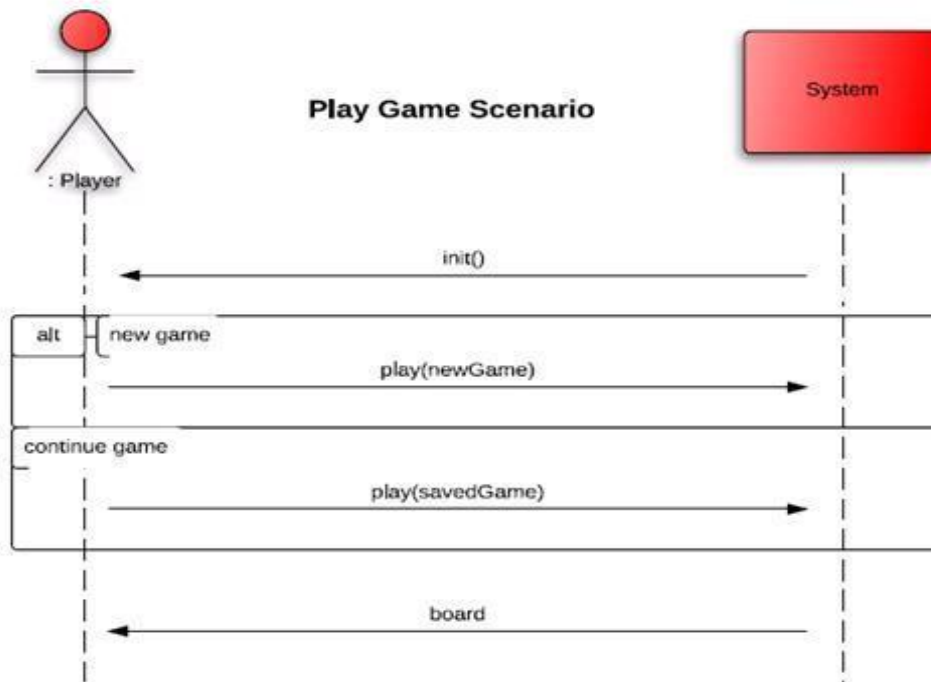
6b: Three or more same colored lokums are in a line vertically or horizontally so they are destroyed.

1. The line consists time lokum so when they are destroyed, additional time is included to the total time left for player to finish the game.
2. The lokums above are fallen needed number of square to fill in the empty space by the system.
3. System fills the empty space which is open up at the top of the board because of the falling pieces with randomly selected lokums.

Frequency of Occurrence: As frequently as the player wants.

- System Sequence Diagrams





- ## Operation Contracts

- Contract CO1: play

Operation: play

References: Use Cases: Play Game

Preconditions: Game is opened.

Postconditions: currentboard, currentScore and remaining moves are transferred to the board.
Game is ready to play.

- Contract CO2: swapLokums

Operation: swapLokums(Lokum: lokum1, Lokum: lokum2)

References: Use Cases: Move Lokum

Preconditions: Game is already played.

Postconditions: Positions of the lokums change.

- Contract CO3: save

Operation: save()

References: Use Cases: Save Game

Preconditions: Game is already being played.

Postconditions: State instance is created. currentScore, remaining moves and current board are saved.

- Domain Model

