

## Use Case UC3: Load Game

**Scope:** Chewy Lokum Legend Game

**Level:** User-goal

**Primary Actor:** Player

**Stakeholders and Interests:**

- Player: able to load the state of a saved game at any level, and is able to resume from this saved state.

**Preconditions:** Player is in the main menu and wants to load a saved game.

**Success Guarantee (or Post conditions):** The state of the game is returned with the saved state of the board, number of remaining moves and the score.

**Main Success Scenario (or Basic Flow):**

1. Player clicks the 'Load' button.
2. A pop-up window appears for selecting an XML file.
3. User selects an XML file.
4. State of the board, number of remaining moves and the score are loaded and the game is ready to be played.

**Alternative Flows:**

- \*a. If a device dependent problem occurs, and game is not loaded,
  1. System signals error to player,
  2. Pop-up windows appear again for selecting the XML.
- 3a. User selects an invalid file:
  1. System signals error to player,
  2. Pop-up windows appear again for selecting the XML.

**Special Requirements:**

- While selecting an XML file, files extensions other than .XML should be disabled, so the user only can select XML files.

**Technology and Data Variations List:**

XML schema will be used to for the validation of the file.

**Frequency of Occurrence:** Depends on user, could be frequently

**Open Issues:**

## Contract C01: loadGame

Operation: Cross

loadGame(File)

References:

Use Cases: Load Game

Preconditions:

- Player is in the main menu and clicks load button.

Post conditions:

- State of the board, number of remaining moves and the score are loaded.

## **Contract C02: crushTimeLokum**

Operation: Cross	crushTimeLokum(TimeLokum)
References:	Use Case: Play New Game
Preconditions:	<ul style="list-style-type: none"><li>- Game is active.</li><li>- A group is formed with Time Lokum type of lokum.</li></ul>
Post conditions:	<ul style="list-style-type: none"><li>- The group is crushed.</li><li>- And additional time is added to the remaining time of the level.</li></ul>

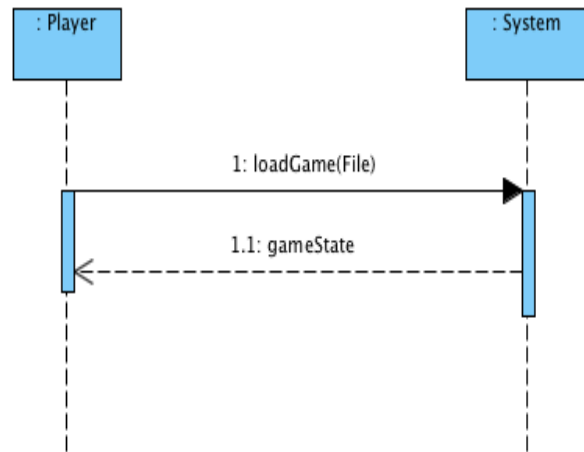
## **Contract C03: decreaseTime**

Operation: Cross	decreaseTime()
References:	Use Case: Play New Game
Preconditions:	<ul style="list-style-type: none"><li>- Game is active.</li><li>- The player is playing a time-based level.</li></ul>
Post conditions:	<ul style="list-style-type: none"><li>- Time is decreased.</li><li>- If the remaining time is 0, the game stops and the score is checked if it exceeded the goal score.</li></ul>

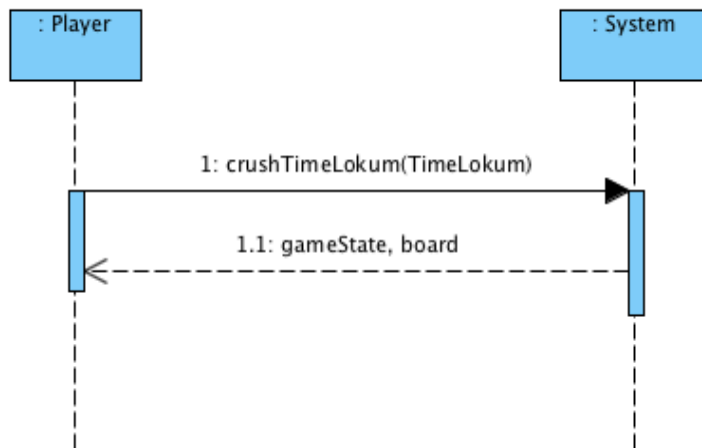
## **Contract C04: specialSwap**

Operation: Cross	specialSwap(Lokum1, Lokum2)
References:	Use Case: Play New Game
Preconditions:	<ul style="list-style-type: none"><li>- Game is active.</li><li>- Two lokums that are not adjacent selected.</li><li>- Number of special swaps is greater than zero.</li></ul>
Post conditions:	<ul style="list-style-type: none"><li>- Lokums are swapped.</li><li>- Number of special swaps, number of moves, score are updated.</li></ul>

**sd LoadGame**



**d crushTimeLokum**



**sd** crushTimeLokum

