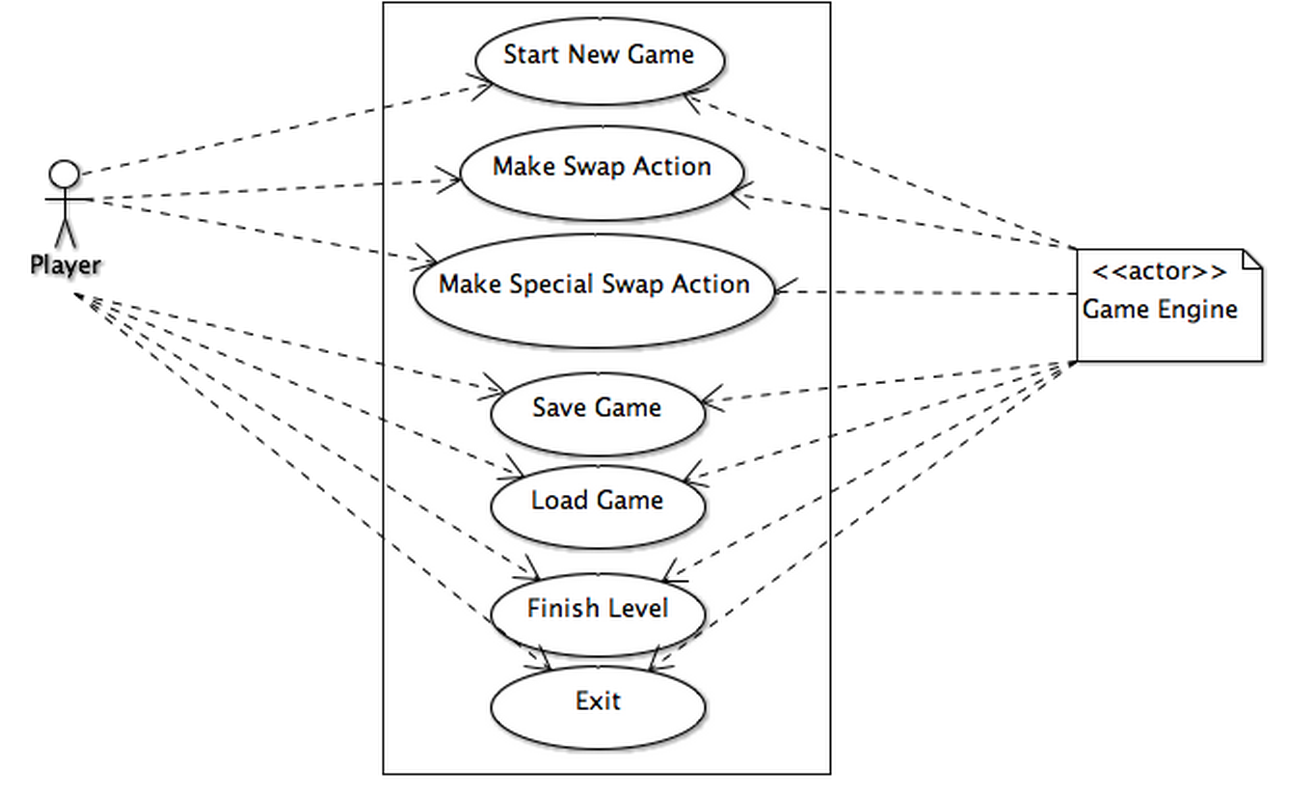
****

**Use Case:** Start New Game

**Scope:** Chewy Lokum Legend Application

**Primary Actor:** Player

**Stakeholders and Interests:**

**- Player:** Wants to start a new game

**- Game Engine:** Wants to instantiate and start a new game with zero score and without any errors

**Preconditions:** None

**Success Guarantee:** Game is created and started on the screen. Player is able to interact with the game.

**Main Success Scenario:**

1. Player clicks on "Start New Game" button

2. Game Engine creates the board with randomly generated lokums and an empty score calculation data

3. Game Monitor displays the screen with board and score table

**Extensions:**

\*a. At any time user may close the game or the game may crush.

1. The game is closed.

**Frequency of occurrence:** Almost once in every execution

**Use Case:** Make Swap Action

**Scope:** Chewy Lokum Legend Game

**Primary Actor:** Player

**Stakeholders and Interests:**

**- Player:** Wants to swap the lokums and also see the action and updated score on the screen

**- Game Engine:** Wants to do the necessary lokum moves until the board sets on stable screen and show it. Also wants to calculate the score due to action and update it.

**Preconditions:** Game is started and running

**Success Guarantee:** Action is made. Screen and the score table is updated.

**Main Success Scenario:**

1. User swaps one of the Lokums with another one and create a lokum combo.

2. Game Engine removes combo lokums and shows the next state of the board.

3. Game Engine updates the score.

4. Steps two and three are repeated until the board is stable.

**Extensions:**

\*a. At any time user may close the game or the game may crush.

1. Game Engine saves the game state.

2. Game Engine closes the game.

2a. The swap does not create a combo.

1. Game Engine swaps the lokums back to their previous locations.

2b. The swap creates special lokum combo.

1. Game Engine removes combo lokums.

2. Game Engine puts the special lokum into the swap destination.

2c. One of the removed lokums is a special lokum.

1. Game Engine explodes the special lokum.

2d. One of the removed lokums is a Time Lokum.

1. Game Engine adds extra time to remaining time.

**Frequency of occurrence:** Very frequent

**Use Case:** Make Merge Swap Action

**Scope:** Chewy Lokum Legend Game

**Primary Actor:** Player

**Stakeholders and Interests:**

**- Player:** Wants to swap the special lokums and also see the action and updated score on the screen.

**- Game Engine:** Wants to do the necessary lokum moves until the board sets on stable screen and show it. Also wants to calculate the score due to action and update it.

**Preconditions:** Game is started and running

**Success Guarantee:** Action is made. Screen and the score table is updated.

**Main Success Scenario:**

1. User swaps one of the Lokums with another one and create a special lokum combo.

2. Game Engine removes combo lokums and any lokum destroyed by the special combo.

3. Game Engine updates the score.

*Steps two and three are repeated until the board is stable.*

**Extensions:**

\*a. At any time user may close the game or the game may crush.

1. Game Engine saves the game state.

2. Game Engine closes the game.

2a.The result might depend on the combo made.

1a. The one of special lokums is a bomb lokum.

1a. The other swapped lokum is a normal lokum.

1. Game Engine removes all the lokums having same color with the normal lokum.

1b. The other swapped lokum is a striped or wrapped lokum.

1. Game Engine turnes all the lokums with the same color is turned in to the same special lokum.

2. Game Engine explodes all these special lokums.

1c. Other swapped lokum is a bomb lokum.

1. Game Engine removes all the lokums on the board.

1b. Two swapped lokums are special lokums but not bomb lokum.

1a. Two swapped lokums are striped lokum

1. Game Engine removes row and column lokums.

1b. One of the lokums is striped and other lokum is wrapped

1. Game Engine removes three rows and three coloumns.

1c. Two swapped lokums are wrapped lokums

1. Game Engine removes lokums extra points.

2b. The removed lokums can be special lokums.

1. Game Engine explodes the special lokums to be removed.

**Frequency of occurrence:** Moderately frequent

**Use Case:** Make Special Swap Action

**Scope:** Chewy Lokum Legend Game

**Primary Actor:** Player

**Stakeholders and Interests:**

**- Player:** Wants to swap any two lokums and also see the action and updated score on the screen.

**- Game Engine:** Wants to do the necessary lokum moves until the board sets on stable screen and show it. Also wants to calculate the score due to action and update it.

**Preconditions:** Game is started and running

**Success Guarantee:** Action is made. Screen and the score table is updated.

**Main Success Scenario:**

1. User swaps one of the Lokums with another one and create a special lokum swap.

2. Game Engine removes combo lokums and any lokum destroyed by the special swap.

3. Game Engine updates the score.

*Steps two and three are repeated until the board is stable.*

**Use Case:** Save Game

**Scope:** Chevy Lokum Legend Game

**Primary Actor:** Player

**Stakeholders and Interests:**

**-Player:** Wants to save his game.

**-Game Engine:** Wants to save game to a XML file state such that it may be recovered at a later time.

**Preconditions:** Chevy Lokum Legend game is in progress.

**Success Guarantee:** Game state is correctly saved.

**Main Success Scenario:**

1. Player clicks on Save Game button.

2. Game Engine saves the game state.

**Alternative Scenarios:**

1a. System is unable to save the game state for some reason.

1. Game Engine displayes an error message to the Player.

**Frequency of Occurrence:** Never to several times per execution

**Use Case:** Load Game

**Scope:** Chevy Lokum Legend Game

**Primary Actor:** Player

**Stakeholders and Interests:**

**-Player:** Wants to load his game from a previous state.

**-Game Engine:** Wants to read an XML file and load the game from the previous state.

**Preconditions:** Chevy Lokum Legend software is up and running. Xml file from a previous game state exists.

**Success Guarantee:** Game state is correctly recovered.

**Main Success Scenario:**

1. Player clicks on Load Game button.

2. Game Engine loads the last game state.

**Alternative Scenarios:**

1a. System is unable to load the game state for some reason.

1. Game Engine displayes an error message to the Player.

**Frequency of Occurrence:** Never to several times per execution

**Use Case:** Finish Level

**Scope:** Chevy Lokum Legend Game

**Primary Actor:** Player

**Stakeholders and Interests:**

**-Player:** Wants to observe the current level’s end and observe the level end actions such as forming and destroying of striped lokums as many as the user’s remaining lives.

**-Game Engine:** Wants to perform level end actions mentioned in the Player stakeholder.

**Preconditions:** Chevy Lokum Legend game is in progress. Level score suggests moving on to next level.

**Success Guarantee:** Level end actions are performed and next level has been reached.

**Main Success Scenario:**

1. Player reaches to the level score which suggests moving on to next level.

2. Game Engine calculates the score.

3. Game Engine removes the current level’s board.

4. Game Engine creates the next level’s board and initializes it with random lokums.

**Alternative Scenarios:**

\*a. Game Engine is unable to perform level end actions because of a failure.

1. Game Engine saves the state of the board.

2. Game Engine closes the application.

1a. Time is up and player loses the game.

**Frequency of Occurrence:** Intermediate.

**Use Case:** Exit Game

**Scope:** Chevy Lokum Legend Game

**Primary Actor:** Player

**Stakeholders and Interests:**

**-Player:** Wants to exit the game.

**-Game Engine:** Wants to terminate ongoing game.

**Preconditions:** Chevy Lokum Legend software is up and running.

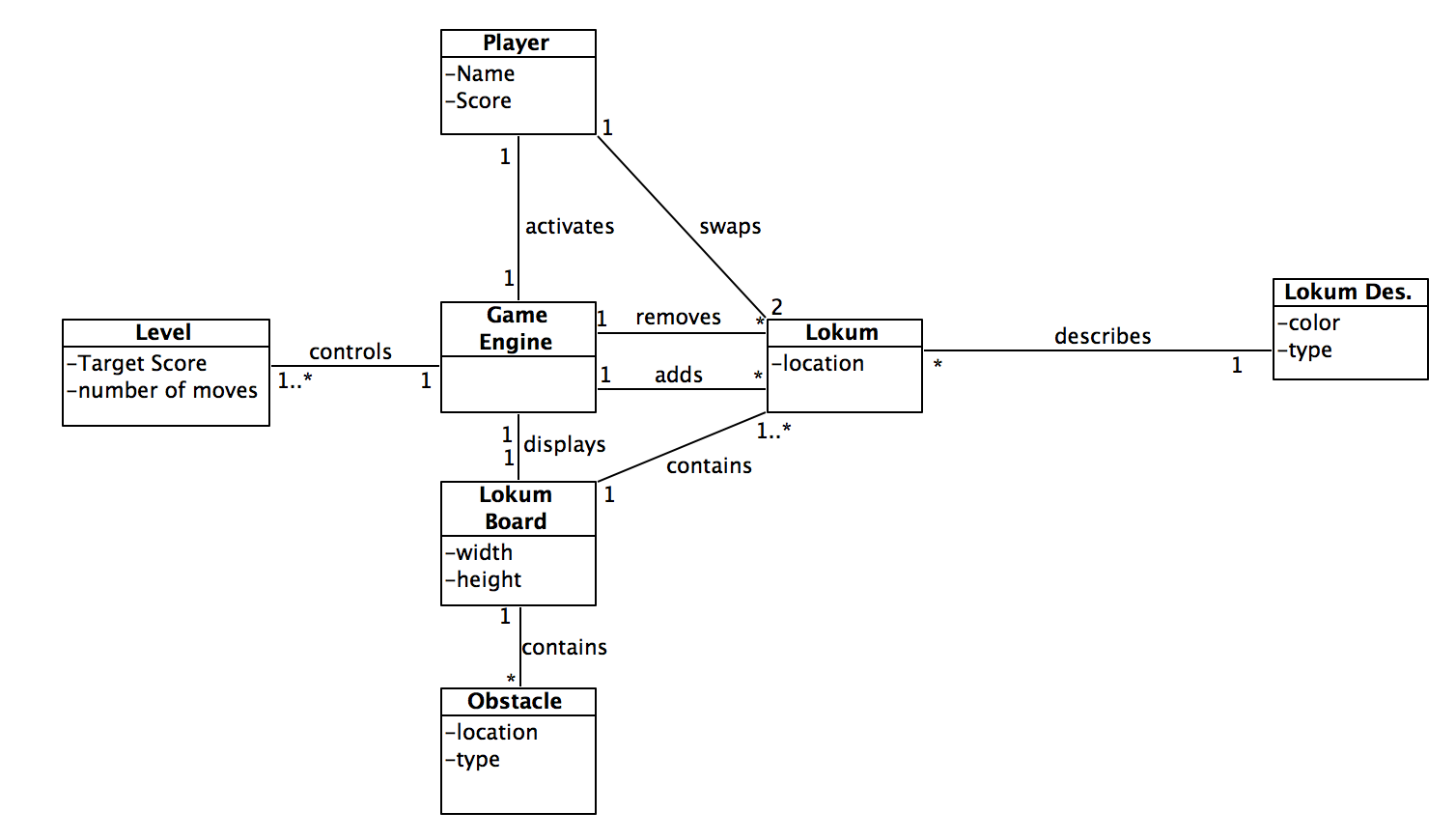
**Success Guarantee:** Application is closed

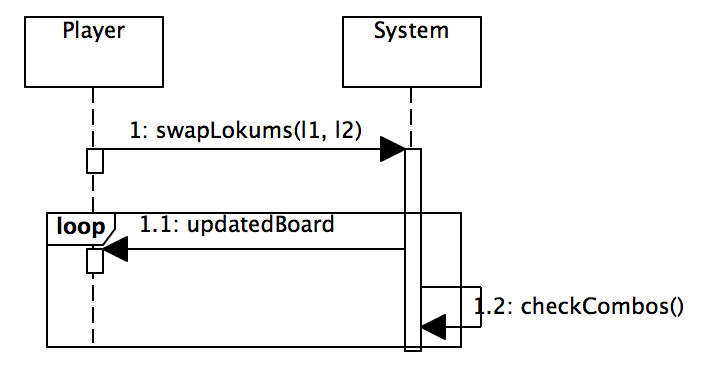
**Main Success Scenario:**

1. Player clicks on Exit Game button.

2. Game Engine saves the level progress of the Player.

3. Game Engine closes the application.

**Frequency of Occurrence:** Once per execution

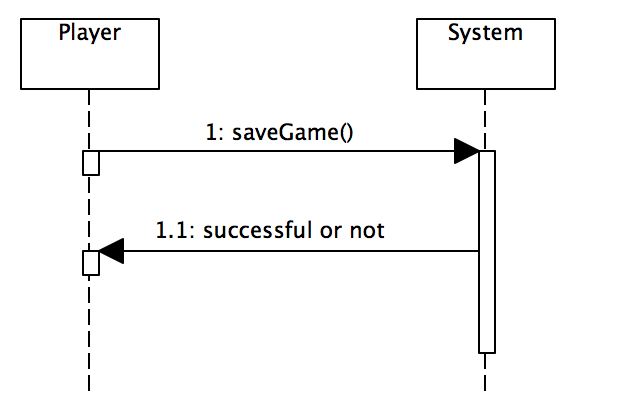
****

**Operation:** swapLokums(firstLokum: lokum, secondLokum: lokum)

**Cross References:** Use Cases: Make Swap Action

**Preconditions:** Game is initialized and running

**Postconditions:** Necessary blocks are removed. Special lokums are created. Board and score is updated.



**Operation:** saveGame(game: Game)

**Cross References:** Use Cases: Save Game

**Preconditions:** Game is initialized and running

**Postconditions:** Current state and level progress of the game is saved to an XML file.

**Operation:** checkCombos(lokumBoard: board)

**Cross References:** Use Cases: Make Swap Action

**Preconditions:** Game is initialized and running

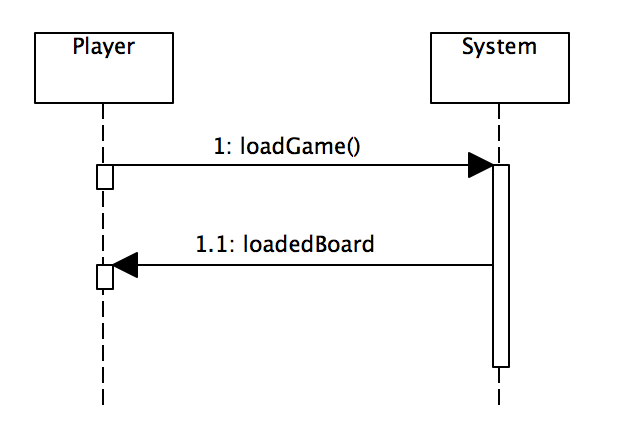
**Postconditions:** Checks if the board is stable(no combos remaining).

**Operation:** finishLevel(score: Score)

**Cross References:** Use Cases: Finish Level

**Preconditions:** Game is initialized and running. Player reached the necessary score.

**Postconditions:** Level is completed. Player starts new level. New level's board is created and displayed.

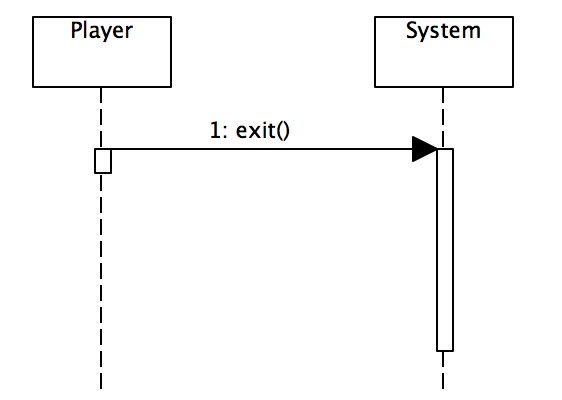


**Operation:** loadGame()

**Cross References:** Use Cases: Load Game

**Preconditions:**Software is open. User requests to load a game.

**Postconditions:** Running game, if present, is terminated and the selected game is loaded.

****

**Operation:** exitGame()

**Cross References:** Use Cases: Exit

**Preconditions:**Game is in progress. User requests to exit.

**Postconditions:**Running game is saved. Running game is terminated and program exits.