# Operation Contracts

## Contract CO1: startNewGame()

**Operation:** startNewGame()

**Cross References:** Use Cases: Start New Game

**Preconditions:**  -Player executes Chewy Lokum program

**Postconditions:** -Player is navigated to screen so he can choose level or load game

## Contract CO2: chooseLevel(x)

**Operation:** chooseLevel(x)

**Cross References:** Use Cases: Choose Level

**Preconditions:**  -player started new game

**Postconditions:** -selected level x is started in the board

-lokums are generated in board

-board state is set ready to swap

## Contract CO3: loadGame()

**Operation:** loadGame()

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

**Preconditions:**  -player saved a game while playing the game

-player started the game

**Postconditions:** -game is loaded from saved file to the board

-board state is set ready to swap

## Contract CO4: saveGame()

**Operation:** saveGame()

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

**Preconditions:**  -Player paused the game while playing

**Postconditions:** -current instance of board, scores, number of remaining moves are saved at a file

## Contract CO5: pauseGame()

**Operation:** pauseGame ()

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

**Preconditions:**  - Player is playing the game in a board

**Postconditions:** -game is paused

-player is navigated in to pause screen

## Contract CO6: continueGame()

**Operation:** continueGame ()

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

**Preconditions:**  - game is paused

**Postconditions:** -board state is set ready to swap

## Contract CO6: selectLokum(x, y)

**Operation:** selectLokum(x, y)

**Cross References:** Use Cases: Play Game

**Preconditions:**  - board state is ready to swap

**Postconditions:** -lokum at (x,y) is selected if selectedLokum1 is empty(null) and selectedLokum1 is set to (x,y)

-if selectedLokum1 is not empty, selectedLokum2 is set to (x,y) .

-if selectedLokum1 and selectedLokum2 are not adjacent unselectLokums() sets them back to null and set board state ready to swap.

-if they are adjacent check if they are swappable( checks if lokums are matched into necessary combination of lokums to be destroyed(3 or more lokums are matched or special lokums are swapped)).

-if false unselectLokums() if true swapLokums(selectedLokum1, selectedLokum2)

-number of remaining moves is decreased by one

-destroyLokums() destroys matched lokums or special lokums accordingly

-score is updated

-lokums fall from the above to the destroyed lokums places

-generateLokum() generates lokum at the top of board if the first line is empty

-loops until no lokums can be destroyed from destoryLokums()

-board is ready to swap

## Contract CO8: exitCurrentGame()

**Operation:** exitCurrentGame()

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

**Preconditions:**  -game is paused

**Postconditions:** -player exits the current game

-player is navigated back to main screen

## Contract CO9: activeSpecialSwap()

**Operation:** activeSpecialSwap()

**Cross References:** Use Cases: Play Game

**Preconditions:**  -player has chance for special swap. Special Swap button has to be clicked.

**Postconditions:** -The location of any two lokums are changed. The chance is decreased by one at a time.