USE CASE UC1: PLAYING GAME

* Primary Actor: Player
* Preconditions: Game must be downloaded before requests of playing game
* Success Guarantee (Post-conditions): Game board appears on the screen by game engine. Mission, score, star component and level are arranged. Lokums randomly fall down to the screen
* Main Success Scenario (or Basic Flow):

1. Player clicks to “Play” button
2. Game engine creates the empty game board on the screen.
3. System shows level number which is set before the last playing game if player don’t choose the completed previous level, mission box which has inside randomly allocated duties such as statements like “get 1500 points”, score board that is initialized as 0 at the beginning and also moving number box which shows assigned specific movement number and also this number decreases in every lokum combinations by players. The last box shown in the game console is star component which has 3 combined stars and its color
4. After setting this information, randomly selected lokums appear on the game console by system.

* Extensions (or Alternative Flows) :
* If battery finishes during game:

Game will be closed but game is saved by game engine. Thanks to this property, player can continue his game when battery is charged.

USE CASE UC2: COMPLETING CERTAIN LEVEL

* Primary Actor: Player
* Preconditions: Game must be opened and played. Game Missions must be completed. There must be moving number on moving number box. Targets must be completed according to game rules.
* Success Guarantee (Post-conditions): Next level screen appears on the screen.
* Main Success Scenario (or Basic Flow):

1. Player completes missions that are given randomly by game(in my assumption goal ıs the eliminating of the 50 red rose lokums )
2. System checks whether mission is actually completed or not. Then, if it is so, system control moving number box.
3. When there are any moving number, system updates level number.
4. New level appears on the screen with different target, moving number etc.
5. System gives information screen such that player has to repeat level again because player didn’t complete given mission.

* Extensions (or Alternative Flows) : NONE

USE CASE UC3: CHOOSING COMPLETED LEVEL

* Primary Actor: Player
* Pre-conditions: Player should see all completed level and these level games must be done at least one time.
* Success Guarantee (Post-conditions): Information of desired level appears on the screen.
* Main Success Scenario (or Basic Flow):

1. Player wants to play any old level and choose one of them.
2. Levels and saved games are remembered by system.
3. When player choose a certain level, information such as missions and moving number about that level calls to screen.
4. Target will be updated according to completed level.

* Extensions (or Alternative Flows) :
* If there isn’t enough new life (oynama hakkı)

Game engine gives to player waiting time for new life.

USE CASE UC4: MOVE SPECIAL LOKUMS

* Primary Actor: Player
* Preconditions: Game must be opened and played. According to game levels, player must get these special lokums during game.
* Success Guarantee (Post-conditions): After special lokums, some score points are given to player. Moving number will change.
* Main Success Scenario (or Basic Flow):

1. Player faces striped+ stripped lokum combos vertically or horizontally.
2. System will update score box as if player makes 2\* individual stripped lokums.
3. Moving number box will decrease one movement number.
4. At that time, crushed lokums will disappear; lokums that are in the higher rows will fall down due to disappeared lokums.
5. If there is any crush and combination respectively during the falling down, process will be repeated.
6. During crushes and combinations, score component also changes according to game rules.
7. Player faces striped+ wrapped lokum combos vertically or horizontally.
8. System will update score box as if player makes3x individual horizontal Striped lokums and 3x individual vertical Striped lokums.
9. At that time, crushed lokums will disappear; lokums that are in the higher rows will fall down due to disappeared lokums.
10. , randomly lokums will fall down from the top of the board. If there is any crush and combination respectively during the falling down, process will be repeated.
11. During crushes and combinations, score component also changes according to game rules.
12. Player faces wrapped + wrapped lokum combos vertically or horizontally.
13. System will update score box as if player makes 3600 points.
14. Moving number box will decrease one movement number. At that time, crushed lokums will disappear; lokums that are in the higher rows will fall down due to disappeared lokums.
15. Randomly lokums will fall down from the top of the board. . If there is any crush and combination respectively during the falling down, process will be repeated.
16. During crushes and combinations, score component also changes according to game rules.
17. Player faces striped+ color bomb lokum combos vertically or horizontally.
18. System will update score box as if player makes n\*individual striped lokums where n is number of the matching striped lokum with the same color lokum.
19. Moving number box will decrease one movement number. At that time, crushed lokums will disappear; lokums that are in the higher rows will fall down due to disappeared lokums.
20. Randomly lokums will fall down from the top of the board. If there is any crush and combination respectively during the falling down, process will be repeated.
21. During crushes and combinations, score component also changes according to game rules.
22. Player faces wrapped +color bomb lokum combos vertically or horizontally.
23. System will update score box as if player makes 2\* individual color bomb lokums.
24. Moving number box will decrease one movement number. At that time, crushed lokums will disappear; lokums that are in the higher rows will fall down due to disappeared lokums.
25. Randomly lokums will fall down from the top of the board. If there is any crush and combination respectively during the falling down, process will be repeated.
26. During crushes and combinations, score component also changes according to game rules.
27. Player faces color +color bomb lokum combos vertically or horizontally.
28. System will update score box as if player makes 2^n\*100 points using the entire board.
29. At that time, crushed lokums will disappear; lokums that are in the higher rows will fall down due to disappeared lokums.
30. Randomly lokums will fall down from the top of the board. If there is any crush and combination respectively during the falling down, process will be repeated.
31. During crushes and combinations, score component also changes according to game rules.

USE CASE UC5: MOVE LOKUMS

* Primary Actor: Player
* Preconditions: Game must be opened and played. According to game levels, player must get at least 3 lokums together.
* Success Guarantee (Post-conditions): After combinational lokums, some score points are given to player. Moving number will change.
* Main Success Scenario (or Basic Flow):

1. Player aligns same 4 lokums shape vertically or horizontally (forming striped lokum).
2. Score box is updated according to game rules. System gives to player 120 points if 4 lokums which has same color and shape are obtained.
3. Using this special striped lokum, system will additionally give to player extra n\*60 point where n is number of lokum that is cleaned by this special lokum. At the end of these movements, moving number box also is updated and decreased.
4. Player aligns same 5 lokums shape which form “T” view (wrapped lokum).
5. Score box is again updated according to game rules.
6. System gives 200 points to player for obtaining wrapping lokum.
7. When the player uses this special lokums in the game (in our assumption, this is like bomb, when it is used), player is awarded 1080 points.
8. Player aligns same 5 lokums shape which form vertical or horizontal alignment (color bomb lokum).
9. This time, system gives to player 200 point towards to color bomb lokum shape.
10. When player uses this special lokum, according to game rules, score box is updated by 2^n \*60 where n is number of lokum that is bomb by color bomb lokum. At the end of these movements, moving number box also is updated and decreased.
11. Player aligns standard combination of lokums(3 lokums with horizontally or vertically).
12. Systems give to player 60 points.
13. After 3 lokums crush, if crush continues, player will be awarded m\*2 point where m is the number of the combination.
14. Yet another crush happens respectively, player will be awarded m\*4 point where m is the number of the combination.

USE CASE UC6: SAVE GAME

* Primary Actor: Player
* Preconditions: Game must be opened. Player must click to “save” button.
* Success Guarantee (Post-conditions): All game information is remembered. Player can continue with the saved game.
* Main Success Scenario (or Basic Flow):

1. Player should be in the game or continuing game.
2. Player wants to save current game by using save button.
3. System saves all information about current game.

* Extensions (or Alternative Flows) :
* If battery runs out during game:

In my assumption, player plays third times this level. When the game is closed because of the external situations (battery issue), game cannot save for third game instead of second game saving.

USE CASE UC7: EXIT GAME

* Primary Actor: Player
* Preconditions: Game must be played. Player must click to “exit” button.
* Success Guarantee (Post-conditions): Player gets back to menu with saving game or not.
* Main Success Scenario (or Basic Flow):

1. Player wants to exit from the game.
2. Player should push the exit button to back to menu.
3. Game engine asks whether all game information should be remembered or not.
4. Player wants to save current game with all information.
5. Player clicks to “save” button.
6. System saves to player’s game.
7. Then player comes back to menu.

**PHASE 2**

USE CASE UC8: MAKING SPECIAL SWAP

-Primary Actor: Player

-Preconditions: Player should start new game. Player must have at least one right to make special swap.

- Success Guarantee (Post-conditions): If it is available for swap, crush happens and then score box will updated.

-Main Success Scenario (or Basic Flow):

1. Player wants to play new game.

2. Current level must be updated.

3. According to game rules, player has 4 rights for level 4. If level is 5, then player will have 5 special swap rights.

4. When player has at least one right to swap, he can swap any type two lokums in board. Otherwise he cannot.

5. If swapping is proper, then board will be updated. After crush or crushes will happen then player will get score according to type of movement.

USE CASE UC9: COMPLETING TIME BASE LEVEL

-Primary Actor: Player

-Preconditions: Player should start a new game and current level is 4. It has timer which is specific. Player should reach target before time finishes.

- Success Guarantee (Post-conditions): When the play is completed by player, game will be saved then the new level will come for playing with updates.

-Main Success Scenario (or Basic Flow):

1. Player wants to play new game.

2.Current level must be 4.

3.Player has specific time and should finish game before this time.

4.When target is reached, level is completed.

5.Game engine will save this current level and level number increased with new updates.

USE CASE UC10: GETTING TIMER LOKUM

-Primary Actor: Player

-Preconditions: Player wants to play new game. Current level must be level 4. During game, player must randomly be faced with timer lokum.

- Success Guarantee (Post-conditions): Time will increase by 5 minutes and score will increase.

-Main Success Scenario (or Basic Flow):

1. Player wants to start new game.

2. Current level must be level 4.

3.When player will make a crush during game, player may face with timer lokum besides basic lokums.

4. According to game rules, since player is in the time base level, time increase by 5 minutes. And also score box will be updated with proper score point.