2 fully dressed use-cases (iteration 1)

Set Up a Game

Primary Actor:
Player(s)

Stakeholders and Interests:
Player: should be able to start a new game or load up a saved game.

Preconditions:
The player that is creating the game is identified and authenticated by system.

Postconditions:
A game setup, ready for the player(s) to play.

Main Success Scenario:
1. The system provides the user with the opportunity to select a new game or a saved game.
2. The user elects to start a new game. [Alt1: user picks a saved game]
3. The system provides the user with multiple themes to choose from.
4. The user picks the default theme.
5. The system provides the user with an option of playing a game with 2 players or 4 players.
6. The user picks a 4-player game. [Alt2: user picks a 2-player game]
7. The system provides the user with the opportunity to select a single-player or a multiplayer game.
8. The user elects to play a single-player game. [Alt3: user picks a multiplayer game]
9. The system requests the user to enter a name and choose a colour for their king figure/castle tiles.
10. The user chooses a name and the colour for the player.
11. The system provides the user with an option of playing on easy mode or hard mode.
12. The user opts for the easy mode. [Alt4: user picks hard mode]
13. The system assigns respective “easy mode” computer player(s).
14. The system places the castle tiles onto the game board.
15. The system retrieves 4 dominos from the deck and displays them to the user in ascending order.
16. The system determines the initial order of play.
17. The system uses this order to inform which player’s turn it is. [Alt 5: Computer player’s turn]
18. The user chooses a domino to place their king-figure.
19. The system checks if the domino selected by the user is available. [Alt6: User chooses an unavailable domino]
20. The system places the king-figure on the domino.
21. Steps 17-20 are repeated until all the dominos are occupied by king-figures.
22. Use case ends.

Alternative Flows:
Alt1: user picks a saved game

1. The system loads a previously saved game. Use case ends

Alt2: user picks a 2-player game

1. The system removes half of the dominos from the deck.
2. Flow resumes at Main Success Scenario Step 7.

Alt3: user picks a multiplayer game

1. The system requests the user to enter names and choose colours for their king figures.
2. The user chooses names and the colours for the players.
3. Flow resumes at Main Success Scenario Step 11.

Alt4: user picks hard mode

1. The system assigns respective “hard mode” computer player(s).
2. Flow resumes at Main Success Scenario Step 14.

Alt 5: Computer player’s turn

1. The system places the computer player’s king-figure on an available domino.
2. Flow resumes at Main Success Scenario Step 21.

Alt6: User chooses an unavailable domino

1. The system informs the user that an unavailable domino was picked.
2. Flow resumes at Main Success Scenario Step 18.

Exceptions:

If at any time the system is unable to retrieve, record or provide details then the system informs the user of the problem, and the use case ends.

Special Requirements:

Open Issues:

Can we implement computer players (AI) into the game on time?
What type of themes do we have to implement (common colour vision deficiency)?

Take a Turn

Primary Actor:

Player(s).

Stakeholders and Interests:

Player(s): should be able to take turns in the game.

Preconditions:

A game must be setup among players/players and computer players.

Postconditions:

Each player in the game must be able to take their turns.

Main Success Scenario:

1. The system retrieves 4 dominos from the deck and displays them to the user in ascending order. [Alt1: empty deck of dominos]
2. The system determines which king-figure is on the smallest domino and informs the corresponding player that it is their turn. [Alt2: Computer player's turn]
3. The user chooses a new domino to place their king-figure. [Alt3: User chooses an unavailable domino]
4. The system checks if the new domino selected by the user is available.
5. The system places the king-figure on the new domino.
6. The user then picks a position to place the old domino on the game board. [Alt4: No position available]
7. The system checks if the domino is placed in a valid position. [Alt5: User places domino in an invalid position]
8. The system places the old domino in the position.
9. Steps 2-8 are repeated until all the king-figures are on the new dominos or all the king-figures are removed from the old dominos.
10. Use case ends.

Alternative Flows:

Alt1: empty deck of dominos

1. The system determines which king-figure is on the smallest domino and informs the corresponding player that it is their turn. [Alt2: Computer player's turn]
2. The system removes the king-figure from the domino
3. Flow resumes at Main Success Scenario Step 6.

Alt2: Computer player's turn

1. The system removes computer player's king-figure and places it on an available domino (if any).
2. The system places the old dominos in a valid position
3. Flow resumes at Main Success Scenario Step 9.

Alt3: User choose an unavailable domino

1. The system informs the user that an unavailable domino was picked.
2. Flow resumes at Main Success Scenario Step 3.

Alt4: No position available

1. The system informs the user that there are no possible positions to place the domino.
2. The system discards the domino and informs the user that the domino has been discarded.
3. Flow resumes at Main Success Scenario Step 9.

Alt5: User places domino in an invalid position

1. The system informs the user that position selected is an invalid position.
2. Flow resumes at Main Success Scenario Step 6.

Exceptions:

If at any time the system is unable to retrieve, record or provide details then the system informs the user of the problem, and the use case ends.

Special Requirements:

If the user does not respond to their turn within 4 minutes, the system informs that the player's turn was skipped and gives the next player their turn.

Open Issues:

The implementation/functionality of the computer player will affect the way the system interacts with the user.