

Use case: Setup a game

Primary Actor: Player

Stakeholders and interests:

- Player: Want to play a match of Kingdomino with or without other players present, want to not have to carry the physical board game around and want a smooth bugless experience.
- Debugger Staff: Keep the game updated and bug-free, fix any bugs that are present and maintain the functionality of the application.

Preconditions:

The game has compiled properly and at least 1 player is present.

Success Guarantee (Postconditions):

The game begins. The user and other players are aware that the game has begun and have claimed their initial domino. Computer-player (AI) have been initialized, have claimed their initial domino, and are waiting their turns.

Main Success Scenario:

1. The user requests a list of game-modes that are currently available.
[Alternate 1: No game-modes available]
2. The system retrieves a list of game-modes available and displays them to the user.
3. The system provides the user with the opportunity to select a game-mode or to exit the use case. [Use Case Ends].
4. The user selects the game-mode they want to play.
5. The system asks the user of how many human players will be present.
[Alternate 2: too many players]
6. The system retrieves the details for the chosen game-mode, including the maximum number of dominos, number of human controlled 'meeples', number of computer (AI) controlled 'meeples' along with their difficulty

setting, and displays the details to the user.

[Alternate 3: AI players present]

7. The user confirms that they want to begin the game with their selected settings. [Alternate 4: User declines to begin]
8. The system begins the game and displays 4 randomly selected dominos with their terrain side visible from the overall deck available and removes them from the deck.
9. The system randomly selects one of the players and provides them with the opportunity to make the first move and claim their initial domino.
10. The system checks whether all players (Humans and AI) have claimed their initial domino [Alternative 5: Players haven't claimed a domino]
11. The system displays 4 randomly selected dominos with their blank side hidden and removes them from the deck. [Use Case Ends].

Alternative Flows:

Alternative 1: No game-modes available

1. The system informs the user that no game-modes are currently available to be played. Use case ends.

Alternative 2: Too many players

1. The system informs the user that too many players are present and there must be a maximum of 4 players present.
2. Flow resumes at main success step 4.

Alternative 3: AI players present

1. The system requests the user to set the difficulty for the AI controlled 'meeples'.
2. Flow resumes at main success step 7.

Alternative 4: User declines to begin

1. Flow resumes at main success step 2.

Alternative 5: Players haven't claimed a domino

1. The system randomly selects one of the players that has not claimed their initial domino and provides them with the opportunity to claim an unclaimed domino.
2. Flow resumes at main success step 10.

Exceptions:

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

Special Requirements:

Colors and sizes of text fonts used must provide - or be able to provide - for the visually impaired (e.g., color blindness, weak-sighted).

Open Issues:

- Is the AI too difficult or too easy to play against?
- Can the game have more than 4 players?
- Do the user(s) know the rules and how to play the game?