Use Case UC1: Starting and ending game

Primary Actor: A user playing the game.
Stakeholders:
- User: Wants to play a game session.
- Developer: Wants to provide the user with a good experience.
Precondition: The application is started.
Success Guarantee (postconditions): The application is shut down.

Basic Flow:
1. The user is presented with a choice between playing a singleplayer game, a multiplayer game, getting help, or exiting the game.
2. The user makes his choice.
3. The game board is shown.
4. The game is played (see UC2 below).
5. Repeat from 1.

Extensions:
2a. The user selects to play a singleplayer game.
2b. The user selects to play a multiplayer game.
    1. The user is presented with a choice between joining or hosting a game.
        1a. The user selects to join a game.
            1. The user is prompted for a nickname.
            2. A list of games is shown.
            3. The user selects a game from the list.
            4. The user waits for the game session to start.
        1b. The user selects to host a game.
            1. The user is prompted for a nickname and the number of players.
            2. The user starts the game.
            3. Data is sent to a central server to update the game list and open the game.
            4. The user waits for the game to get full.
            5. The user starts the game session.
            6. The central server is notified that the game is closed.
2c. The user is presented with a help menu
    1. The user leaves the menu; repeat from 1 in the basic flow.
2d. The user exits the game.
    1. Shut down the application.
    2. Terminate process.
3-4. A user leaves the game.
    1. The remaining players are asked to decide if they want to continue on a smaller board or stop the current session.
    2. The game continues or stops based on a majority decision.

Special requirements:
For multiplayer gameplay, access to a computer network is required.

Technology and data variation:
None.

Frequency of occurrence:
Unspecified.

Trigger:
Starting the application.

Use Case UC2: Playing game

Primary Actor: A user playing the game.
Stakeholders:
- Users: Wants to place pieces on the board so as not to reach the top.
- Developers: Wants to make sure the game is tactically challenging.
Precondition: A game session is active.
Success Guarantee (postconditions): The game session is quit and the user is presented with a total score.

Basic Flow:
1. The user recieves a piece.
2. The user navigates the piece by rotating it and/or moving it by pressing keys on his keyboard.
3. The piece gets fixed, and a result depending on the placement of the piece is given.
4. A score bonus is calculated.
5. If the piece is not at the top of the board, repeat from 1.
6. The session ends.
7. The user is presented with a total score.
8. The session is quit.

Extensions:
2a. The user uses a powerup.
    1. The powerup takes effect, altering gameplay or the game board.
2b. In a multiplayer session, the user's piece collides with another player's piece.
    1a. The pieces are removed.
        1. Points are subtracted from the total score.
    1b. The pieces move apart.
        1. Nothing happens.
3a. The piece does not complete a row.
3b. The piece completes a row.
    1. The row is removed.
    2. If there are any powerup bricks[1] in the given row, add them to the player's powerup list.
    3. Pick one brick randomly, and replace it with a powerup brick.
4a. A row was removed.
    1. Add a certain score, X, to the total score.
4b. A row was not removed.
    2. Add a certain score, Y, to the total score.

Special requirements:
For multiplayer gameplay, access to a computer network is required.

Technology and data variation:
2. The user uses a game controller instead of a keyboard.

Frequency of occurrence:
Continuous.

Trigger:
Starting a game session.

1 Each piece consists of four square parts, here referred to as bricks.

Non-functional requirements

- The game will have to be properly documented so that the game can be understood by the users.

- The game should be reliable, so that network games are not interrupted by internal desynchronization or differences between different players' clients.

- We are limited to using the Open Graphics Library via the Lightweight Java Game Library wrapper (see the Project Overview Document for further details) for developing the game; however, we have no constraints on other developing tools.