Tetris

Goal

We are planning to implement Tetris. In Tetris (the “marathon” mode), the player will start with a blank map. When the player decides to start the game, blocks of various shapes, called “tetriminos” will begin falling, one at a time, from the top of the map. The player must move and rotate each tetrimino in order to fill up a line. As each line fills up, the line clears and the score will increase. If the player clears several lines at once, they will score a bonus in addition to the sore associated with the line cleared. Note that if a tetrimino falls on some existing tetrimino where there is an empty space below the existing block, the empty space cannot be filled. Each tetrimino has an area of four and there are seven types of tetriminos, in play. As the game progresses, the tetriminos will fall faster thereby increasing the difficulty of the game. The game ends when the map fills up and the player is unable to place more blocks. Since there is no way to really win the game, the goal would be to get as high of a score as possible.

Features

The following are the features included in the marathon mode of tetris

- Basic clear map
- Score board
- Falling blocks
- Line clears after it fills up
- Controls over falling blocks (movement left and right as well as rotation by 90 degrees)
- Ending the game with the player tops out of the map

If we have extra time, we will implement these additional features:

- Extra features: Map mode of tetris (start with a map), multi-player versions

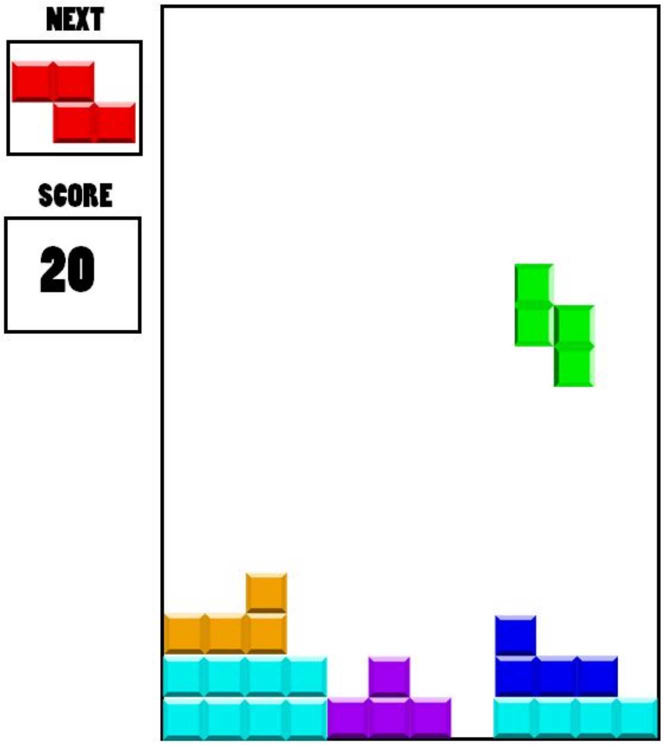
Design Patterns

The following design patterns will be used:

- The over architectural pattern used will be MVC. The model will handle most the game logic. For instance, the model will know about the placement of the blocks, the score, and the general layout of the map. The view will be the general graphics of the game which the user interacts with. It is currently unclear as to whether a controller will be necessary for mediating between the view and model. We expect to determine whether or not to include controller after experimentation.

- Observer-Subject: Since we are implementing the MVC architectural pattern, the GUI of the game will be an observer of the model, which handles all of the game logic (position of the blocks, relations between the blocks, line clears, etc).
- Composite: The GUI uses JavaFX which in turn uses the composite pattern in order to display elements on the screen.

Sketch



-
- The user uses the left and right arrow keys to shift the current tetrimino to the left or right. The user uses the up arrow key to rotate the current tetrimino clockwise by 90 degree.

Team Roles

Overal:

- View & Controller: Qi and Sam
- Model: Weijia and James

Details:

- James
    - Implement the behavior of the tetriminos
        - falling
        - rotating
        - being blocked by existing tetriminos
        - landing
- Weijia
    - Implement clearing lines
    - Update the score
    - Determine the end of the game
- Qi
    - The display of the board
    - Respond to user’s keyboard input
- Sam
    - The display of the existing tetriminos
    - The display of the next tetriminos