Lab 12

Tetris

Announcements

Project 3A (Phase I) due Monday, 11/13 at 11:59 pm.

- This is for world generation, where a submission is required on Gradescope.

Project 3B & 3C due Monday, 11/27 at 11:59 pm.

- This is for interactivity, where a submission is also required on Gradescope.

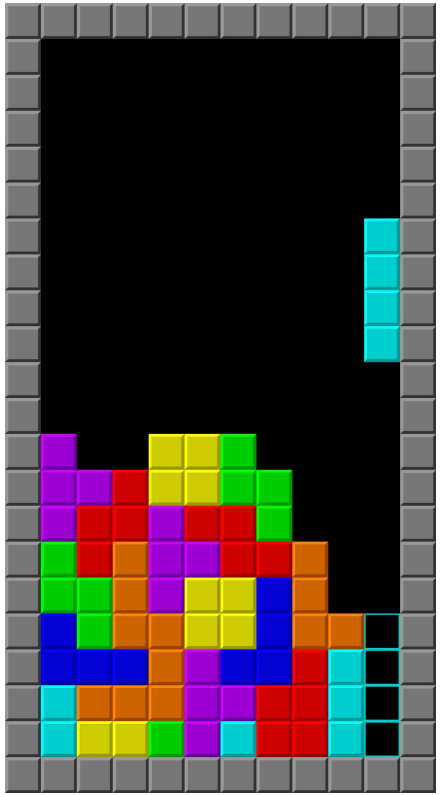
There are no extensions for Project 3B and 3C.

Tetris

Tetris

If you aren't too familiar with Tetris, it's a puzzle video game where players "complete" lines with differently shaped pieces. These pieces are tetrominoes.

For this lab, we'll be implementing a working Tetris game, with a more simple functionality.



Demo

Methods Overview

Methods Overview

The methods you'll have to implement are described more in spec, but here's a brief breakdown:

Methods Overview

The methods you'll have to implement are described more in spec, but here's a brief breakdown:

- **`updateBoard`:** where you take in user input and move the pieces according to the input

Methods Overview

The methods you'll have to implement are described more in spec, but here's a brief breakdown:

- **`updateBoard`:** where you take in user input and move the pieces according to the input
- **`incrementScore`:** updates the score based on the **number of lines cleared**

Methods Overview

The methods you'll have to implement are described more in spec, but here's a brief breakdown:

- **`updateBoard`:** where you take in user input and move the pieces according to the input
- **`incrementScore`:** updates the score based on the **number of lines cleared**
- **`clearLines`:** clears lines after a piece is placed and checks how many lines are completed

Methods Overview

The methods you'll have to implement are described more in spec, but here's a brief breakdown:

- **`updateBoard`:** where you take in user input and move the pieces according to the input
- **`incrementScore`:** updates the score based on the **number of lines cleared**
- **`clearLines`:** clears lines after a piece is placed and checks how many lines are completed
- **`runGame`:** where game logic takes place (specific details in the spec)

Methods Overview

The methods you'll have to implement are described more in spec, but here's a brief breakdown:

- **`updateBoard`:** where you take in user input and move the pieces according to the input
- **`incrementScore`:** updates the score based on the **number of lines cleared**
- **`clearLines`:** clears lines after a piece is placed and checks how many lines are completed
- **`runGame`:** where game logic takes place (specific details in the spec)
- **`renderScore`:** displays the current score

Methods Overview

The methods you'll have to implement are described more in spec, but here's a brief breakdown:

- **`updateBoard`:** where you take in user input and move the pieces according to the input
- **`incrementScore`:** updates the score based on the **number of lines cleared**
- **`clearLines`:** clears lines after a piece is placed and checks how many lines are completed
- **`runGame`:** where game logic takes place (specific details in the spec)
- **`renderScore`:** displays the current score

We'll briefly go over only some of the methods (**`updateBoard`**, **`clearLines`** and **`runGame`**).

Tetris Mechanics

Tetris Mechanics (updateBoard)

When playing the game, the user is able to control the current Tetromino in the following ways:

- `a`: left by one tile
- `s`: down by one tile
- `d`: right by one tile
- `q`: rotate to the left 90 degrees
- `w`: rotate right 90 degrees

Tetris Mechanics (updateBoard)

When playing the game, the user is able to control the current Tetromino in the following ways:

- `a`: left by one tile
- `s`: down by one tile
- `d`: right by one tile
- `q`: rotate to the left 90 degrees
- `w`: rotate right 90 degrees

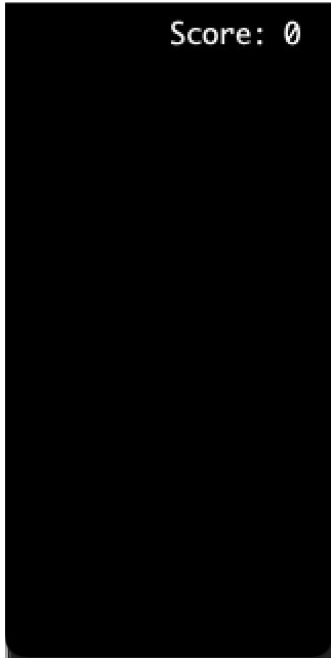
Take a look at the `StdDraw` library (in the spec as well) to see how you might get user input! Make sure to also read through the `Movement.java` class to see the helper methods for moving and rotating.

Completing Lines

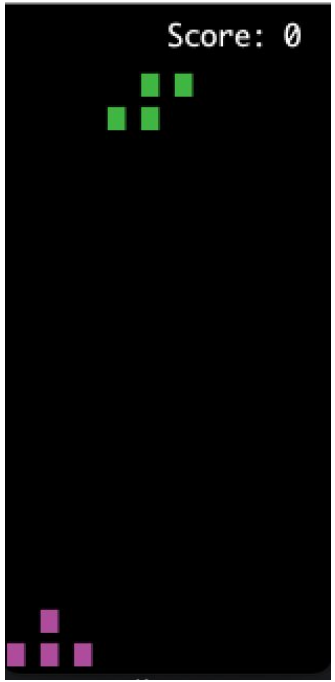
Completing Lines

Earlier we mentioned that we update our score based on the number of lines cleared, which results from when those lines are completed.

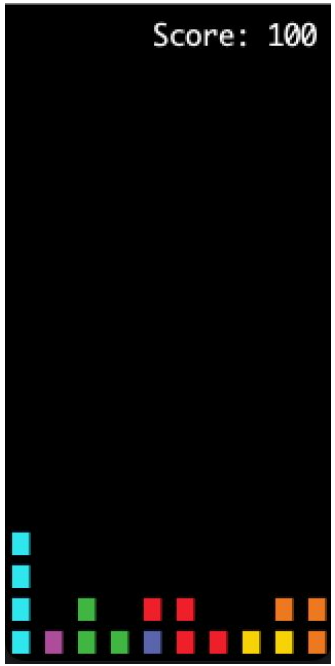
So, what does it mean to complete a line?



When you first start your game, you'll see an empty screen.

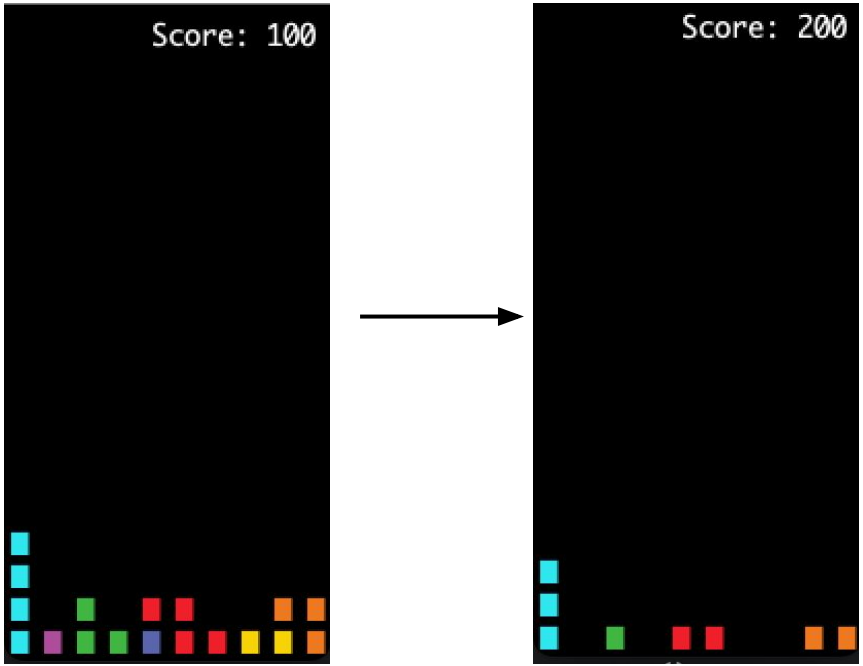


As the game continues, pieces start falling from the top.

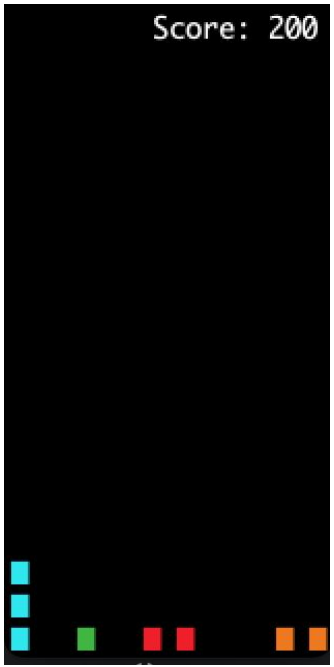


Completed row.

The goal of the player is to assemble the tetrominoes (the pieces) to form one or more rows of blocks, spanning across the width of the board. This completes the row.



Since that row is completed, we clear it and move all the rows above it down. Notice that our score increases after.



So how do we check if a row is completed? Let's think of the opposite: how do we know when a row *isn't completed?* Think about what tile that can be used to check for incompleteness.

clearLines

When we place a single Tetromino, the number of lines cleared can be at minimum of 0 and maximum of 4.

- That means that after every piece is placed, the game will need to check the entire board to see how many lines are cleared.

Running the Game

Running the Game (runGame)

More details are provided in the spec, but one thing to keep in mind is that you'll need your game to continuously run until it reaches a game over state.

- Make sure your game does this! There are helper methods provided, so make sure to use them to help implement the logic for running the game.

Lab Overview

An Overview

Lab 12 is due Wednesday, 11/15 at 11:59 pm. Make sure to plan ahead of time to get checked off!

Deliverables:

- Complete the methods in **`Tetris.java`.**
  - `updateBoard`
  - `incrementScore`
  - `clearLines`
  - `runGame`
  - `renderScore`
- **Make sure to read through the files to know what helper methods are accessible to you!**
  Please read through the spec!

Lab Notes

Some key takeaways for Project 3 (and for checkoff):

- How might you translate what you implemented in this lab to Project 3?
- Interactivity? User input? What about the game logic? How might Tetris resemble the game you're making in Project 3?
- How might you use the `StdDraw` library in your project?

runGame pseudocode

```
public void runGame() {
    resetActionTimer();
    resetFrameTimer();

    // Call on helper method to spawn a piece.

    while (// check for if the game is over) {
        if (shouldRenderNewFrame()) {
            // Once a piece is set to null, it can no longer move
            if (currentTetromino == null) {
               // We then want to call on a helper method to check
               // for rows completed  and clear the lines.

               // As well as spawn a new piece.

            }
            // Call on helper method to take user interactivity.

            // Call on helper method to render the board.

        }
    }
}
```