Tetris

For the computer science final project, I chose to replicate the classic video game Tetris. The object of the game is to rotate the falling blocks in order to create completely filled lines. The main goals for this project were to:

- complete a code that will allow users to play the classic form of Tetris.
- I wanted for the pieces to fall one by one, and when they could not fall any further, a new Tetrad would be formed and start to drop.
- I wanted there to be an “End Game” portion for when no more pieces could fall.
- There needed to be a way to tally points
- In addition to the points tally, every time a row was filled, the row needed to get deleted.
  - This would indicate that points should be added to the score
- And hopefully, the “End Game” and point tally would open in their own window.
- Finally, I wanted music to play in the background while each person played Tetris.

However, I ran into several complications. First I had to develop a method that would get the blocks to fall freely on their own. Then, I had to develop methods that would form new tetrads when the previous tetrad came to a stop. I also needed to create a method that would clear a row when it was completed (it also became the same method that kept track of points). Finally, I needed to create an end game system as well.

To get the tetrads to fall freely on their own, I used an if statement along with a counter that every 30 milliseconds would let the tetrad fall down. It looked like this:

```java
while(true)
{
    tetris.newTetrad();
    tetris.clearBoard();
    tetris.drawBorder();
    if(cn==30)
    {
        tetris.tetrad.moveDown();
        cn = 0;
    }
    tetris.tetrad.move();
    tetris.draw();
    tetris.draw.show(10);
    cn++;
}
```

Once it came to a stop, I had a new problem, which was to get a new tetrad to fall.

The new tetrad formation proved to be challenging at first, but I was able to get it in the end. The code looked like this:

```java
public void newTetrad()
```
if(!tetrad.canMoveDown()){
if(topFull()){
    System.out.println("Game Over");
    System.exit(0);
}
tetrad.stay();
createNewTetrad();
for(int i = BH-1; i > -1; i--){
    checkRow(i);
}
}
}

This simply says that once the previous tetrad has stopped moving, the code checks the Row to see if it is possible for a new Tetrad to move down. If that is possible, a new tetrad is formed. But if any part of the top row is full and a new Tetrad cannot move down, then the program quits and “End Game” appears in the Terminal.

Still the most difficult problem I encountered was to delete a full row and have the blocks fall down once the full row was cleared.

public void checkRow(int r){
    boolean isfull = true;
    for(int j = 0; j<BW; j++){
        if(board[r][j] == 0){
            isfull = false;
        }
    }
    if(isfull){
        points += 100;
        System.out.println("You now have + " + points);
        for(int i = r; i > 0; i--){
            for(int j = 0; j < BW; j++){
                board[i][j] = board[i-1][j];
            }
        }
        for(int j = 0; j < BW; j++){
            board[0][j] = 0;
        }
    }
}

Some of the issues I ran into with the checkRow method was that initially, I was checking from the top-down instead of from the bottom-up. The problem with this is that when I was playing the Tetris game, if a block fell and completed a row, the block would not stop. Instead the block would complete the row, the row would delete, but the blocks would keep falling. The result was that one Tetrad could hypothetically delete more than 4 rows at a time because it would complete the row and continue to fall (which is not allowed). However, once this problem was solved, I was able to add a counter to the
method that would allow me to keep track of the number of rows deleted by assigning a value of 100 points to each row that was deleted.

Finally, I wanted the game to have music. However, the StdAudio did not work for me for some reason. Thus, I contacted Professor Jiang in hopes he could help. However, the cliplayer program I got from him still did not work. I worked for two days after our presentations to get the program to work for me, but was unsuccessful. Thus, the game will not have music which I really disappointing.

Another problem was the “End Game” I had initially hoped for. I wanted to open up a completely new window and have a jframe that would contain not only the score, but also say end game, if the end game argument was met. However, jframe’s do not update themselves, thus I did not decide to try this. However, having seen what some of my classmates had done, I wish I had just thought to put the two on the board itself. I could have had the points and the status of the game above the board itself if I had just used StdDraw. In hindsight, I would have liked to have done that.

Thus, if I had more time I would have liked to do a few more things. With more time I would have liked to polish the game a little more. I would like to implement a method that after a certain period of time would make the Tetrad pieces fall faster (like levels). I would also like to try and have the game pause for a second or two when it was deleting a row rather than just delete them quickly. Because it is hard to tell how many rows just got deleted and it makes the game difficult to follow while you are playing. One last thing I would have liked would be to have some way of notifying the player which new Tetrad was coming up next. This would make the game much easier. It would be a lengthy task, but I believe the game would benefit quite a bit from this feature.

In the end, I believe this Tetris project was a success. I have had fun both making and playing the game myself. I think that this fully functional Tetris game would be a great start to future projects. I am proud of the work I put in and the product that resulted from my hard work. I am sure you will enjoy it too.