Introduction: I decided to do the default Tetris game for my CS101 final project. My goal was to create a fully functional Tetris game. Overall, I felt that the project went pretty well as I was able to successfully complete all of the goals I set out to achieve, except that two of the extensions I implemented did not work out as well as I had hoped. In order to create a fully functional Tetris game I needed to create several functions that performed certain tasks. I successfully created a function that makes the tetrads fall at certain speeds, a function that creates a new tetrad when the old tetrad hits the bottom of the game panel or another tetrad, a function that clears rows of permanent tetrads and moves the other tetrads down, a function that terminates the game by stopping the main while loop, and extensions including a function that creates the scoring method, a function that draws the scoreboard, a function that updates the scores and levels, a function that creates and prints out the “GAME OVER” message and finally, a function that changes the speed of the tetrads.

Main Functions:

1. **Falling Tetrad Function:** The first function I created was the function that makes the tetrads fall at a certain speed. In order to accomplish this task I created an “if statement” that called the moveDown function from the Tetromino class and created a counter that incremented the moveDown function every 20 milliseconds if the player is on level 1. I then put this “if statement” in the while loop of the main method to implement it.

   Code Segment:

   ```java
   if (level == 1 && cn > 20) //Start speed of tetrads falling at slow
   {
   tetris.tetrad.moveDown();
   cn = 0;
   }
   cn++;
   ```

2. **Create New Tetrad Function:** Next, I created a function that creates a new tetrad when another tetrad hits the bottom of the game panel or another tetrad. In order to do this I created a method with an “if statement” that called the stay( ) method from the Tetromino class and the createNewTetrad( ) method from the Tetris class only if the tetrad could not move down anymore. This method worked very well.
Code Segment:

```java
public void callNew()
{
    if (!tetrad.canMoveDown())
    {
        tetrad.stay();
        createNewTetrad();
    }
}
```

3. **Clear Horizontal Rows Function:** Next, I successfully created the function that clears rows of permanent tetrads and shifts the other tetrads down; the clearLine() function. To accomplish this task I first created a series of 10 if statements that searched each of the 10 columns for permanent tetrad id numbers and then called the shift row function to shift down each tetrad block:

```java
public void clearLine()
{
    //Check the horizontal lines for the permanent tetrad id numbers
    for(int i = 0; i < BH; i++)
    {
        if(board[i][0] >= 1 && board[i][0] <= 7)
            ...
        if(board[i][9] >= 1 && board[i][9] <= 7)
            //Call the shiftRow function
            shiftRow(i);
    }
}
```

In order to shift down the tetrad blocks, I needed to figure out how the board array was arranged. I realized that it was arranged in this way:

Bottom row: 19,0 19,1 19,2 19,3... 19,9 and so on up to the top row: 0,0 0,1 0,2 0,3... 0,9. At first I tried to use System.arraycopy() to shift down the array but these tactic caused significant glitches and so I just decided to just create a function with two for loops. Since the array was arranged in this particular way I knew that in the shift row function I had to subtract 1 from the “i” array values because the “i” values represented all of the rows in the array. I then created the actual shift row function using two for loops and assigning the board array a new value. I made the first for loop count down from whatever value “i” was to zero because I only wanted to shift down the tetrad blocks that were above the row that was cleared.

```java
public void shiftRow(int row)
{
    for(int i = row; i > 0; i--)/Loop for all the full rows from 0 to 20
    for(int j = 0; j < BW; j++)
        board[i][j] = board[i-1][j];/Shift down the rows by 1
    increaseScore();//Each row shifted down will increase the score
}
```
4. **End Game Function**: I then created a function that terminated the Tetris game; the 
endGame() function. In order to accomplish this task I created a boolean method that 
returned true if the tetrad at the top of the game panel was a permanent tetrad. This 
method then called the gameOverMessage() function to print out “GAME OVER” 
on the panel and I also put an “if statement” in the while loop that said if the 
endGame() function was true then a break statement would stop the while loop. That 
is how I was able to stop the tetrads from being created when the game was over. 
The function also plays a buzzer sound as the “GAME OVER” message is displayed.

**Code:**

```java
public boolean endGame(){

    boolean end = false;
    for (int j = 0; j < BW; j++)
        if (board[1][j] > 0 && board[1][j] < 8) //check 2nd row to top row for id#
            end = true;
    gameOverMessage();
    StdAudio.play("buzzer.au");
    return end;
}
```

**Extensions:**

1. **Mini-Scoreboard Function, Scoring Function, and Sound Effects**:

   In order to make the game more interesting I added a function that kept the score 
of the game and that printed the updated score and level on a mini-scoreboard on 
top of the game panel. I created the scoreboard using the draw class and below I 
am showing the score function that converts the updated score and level and 
converts them to strings and then prints them on the top of the game panel.

   **Code:**

   ```java
   public void score()
   {
       //draw the score
       int c = score;
       String b = "" + c; //create empty string to convert from int to string
       draw.setPenColor(Color.white);
       draw.text(3.4, 20.4, b);

       //draw the level
       int d = level;
       String e = "" + d;
   }
   ```
In order to create the scoring function that updates the score and level I used two “if statements” and I called this increaseScore() function in the shiftRow() function so that the score was increased by 100 every time one row was cleared and made the level increase by one whenever the player got another 500 points. It also plays a “cha-ching” sound effect using the StdAudio class whenever the player goes up one level.

Code:

```java
public void increaseScore() {
    if (score >= 0) {
        score = score + 100;
    }

    if (score % 500 == 0) // Increases Level every 500 points
    {
        level = level + 1;
        StdAudio.play("cash_register.au");
    }
}
```

2. Game-Over Message Function:

I used the draw class in order to create a message that says “GAME OVER” in orange color and I then called the gameOverMessage( ) function in the endGame( ) method in order to have the GAME OVER print out on the game panel when the game is terminated.

Code:

```java
public void gameOverMessage() {

    draw.setPenColor(Color.ORANGE);
    draw.setFont(new Font("Helvetica", Font.BOLD, 43));
    draw.text(5, 10, "GAME OVER");
}
```

3. Changing Tetrad Speed Function:

My final extension was to change the speed at which tetrads fell once the player hit a certain level. In order to accomplish this task I created two more “if statements” in the while loop of the main method. These if statements would check to see which level the player was on and then implement the moveDown( ) function at a certain speed depending on the level. This function worked well but I wish I could have come up with a function that was in the Tetris class and then just call it from the main method while loop. This can be seen in the main method code below.
The Main Method Code and the Tetrad Speed “If Statements:”
- Here is the main method of my Tetris program.

Main Method Code:

```java
public static void main(String[] args) {
    Tetris tetris = new Tetris();
tetris.createNewTetrad();
    int cn = 0;

    while(true) {
        tetris.clearLine();
tetris.callNew();
tetris.clearBoard();
tetris.drawBorder();
tetris.tetrad.move();
tetris.tetrad.show(10);
tetris.draw();
        if (level == 1 && cn > 25)//Start speed of tetrads falling at slow
        {
            tetris.tetrad.moveDown();
            cn = 0;
        }
        cn++;
        if (level >= 2 && level < 5 && cn > 25)//Increase speed to medium after 2nd level
        {
            tetris.tetrad.moveDown();
            cn = 3;
        }
        cn++;
        if (level >= 5 && cn > 25)//Increase speed to fast after 5th level
        {
            tetris.tetrad.moveDown();
            cn = 10;
        }
        cn++;
    if(tetris.endGame()){
        break;
    }
}
```

**Conclusion:** I had a lot of fun working on this project and I learned a lot about object-oriented programming. In the future as I learn more about programming I hope to add some more extensions to the final game including a much better scoreboard that shows the next tetrad that will come down.
Screenshots of the Game:

A
This screenshot shows how the createNew( ) function calls a new tetrad when another hits bottom of the game panel.

B
This screenshot shows how the level and score are updated and how completed rows are cleared.

C
This screenshot shows that the gameOver( ) function works.