COS126: Final Project Report Template

Fall 2022

**Instructions:** Make a copy of this Google doc and respond to all required questions. Then, download this Google doc as a PDF and change its name to report.pdf. Finally, upload report.pdf to [TigerFile](https://tigerfile.cs.princeton.edu/COS126_F2022/Final_Project_Implementation).

**Note:** Unlike the proposal template, this template is just a suggestion of the minimum information you should include. You can and should feel free to modify this template to suit the needs of your project, for example, by adding sections or questions.

# 1. Basic Information

**Name #1: Brandon Lee**

**NetID #1: bl0428**

Name #2:

NetID #2:

**Project Advisor:**

**Link to YouTube video presentation:**

# 2. Project/Course Feedback

***Before moving on****, fill up the following form (*[*https://forms.gle/MSQC4cswCJ3PBszY8*](https://forms.gle/MSQC4cswCJ3PBszY8)*) to provide feedback on the project and on the course.*

**We filled up the feedback form: (X)**

# 3. Project Description

**What is your project title? Chess Replay**

**Please summarize your project in 1 paragraph. Only describe things you implemented.**

**My project replays a game of chess given an initial setup file and a game file. I did this by implementing a piece data type that serves as the pieces and a board data type that takes an array of pieces. Therefore, one can start the game in any position if the squares and moves are legal. I also implemented a display that shows the pieces captured, as well as an argument to change the replay speed for better viewing experience.**

**Please list all the relevant files (.java files, datasets, .jar libraries) from your project. Include a brief explanation of their contents.**

**Piece.java: piece data type class that simulates a piece and displays it on StdDraw**

**Board.java: board data type class that takes in an array of pieces and visualizes the board on StdDraw. Has methods such as checking if a piece is at a certain square on the board or returning the piece at the square on the board.**

**Game.java: client that takes the piece file and move file and plays the game on StdDraw. I implemented all the rules such as castling, pawn promotion, and en passant. I also implemented some helper methods for maintainability.**

**Describe instructions on how to compile and run your project (e.g. compilation commands).**

**1. Compile using javac-introcs Piece.java, javac-introcs Board.java, javac-introcs Game.java**

**2. Run Game.java using the command java-introcs Game [pieces] [game] [speed (optional)]**

**[pieces]: file of the starting position you want**

**[game]: file of the moves of the game you want to replay**

**[speed]: replay speed, default is 1s in between each move if no argument is given**

**example: java Game pieces.txt game1.txt**

**this will play game1.txt at 1x speed**

**pieces.txt is the starting position for a game of chess**

# 4. Features and Project Requirements

*For each of the following, mention the .java files, classes, and methods that implement the things you are describing. If you used any external libraries, mention those as well.*

*Feature #1:* Piece and Board Data Type

**In 2-3 sentences, please describe your feature. Only describe things you implemented.**

I implemented this in Piece.Java and Board.java. Together, they form the pieces and board, and can simulate a chess board. Piece stores the position of a piece and its image, while board stores an array of pieces which are the pieces on the board. The board class can initialize the board drawing, fill squares to hide pieces, and check if a piece is at a square and what piece is there. The piece class can place and move pieces and change the image of a piece for promotion

**In 1-2 sentences, please describe how you tested your feature.**

I used the main methods of each class to test all the functions, such as placing and moving pieces, creating a new board, and checking the piece at a certain square. I also implemented toString() methods for these classes to make it easier.

**What was the agreed category of your feature using the buckets paradigm?**

( ) Standard

(X) Sprinkle

( ) Sparkle

*Feature #2: Game Replayer*

**In 2-3 sentences, please describe your feature. Only describe things you implemented.**

I implemented a chess game player that includes all the rules of chess, such as captures, castling, pawn promotion, and en passant. The client assumes that all the pieces are in valid squares and all the moves in the input file are valid and legal. It takes 3 command line arguments: the piece starting position file, the moves file, and an optional replay speed double to change the speed.

**In 1-2 sentences, please describe how you tested your feature.**

**I made some test positions and games that tested all the features of this feature, such as moving all the castling and en passant. The test files are moveTest.txt, which moves all the pieces using pieces.txt as the piece file, and pawnTest.txt, which tests en passant and pawn promotion using testPosition.txt as a piece file, which places only pawns on the board**

**What was the agreed category of your feature using the buckets paradigm?**

( ) Standard

( ) Sprinkle

(X) Sparkle

*Feature #3: Captured Piece Display*

**In 2-3 sentences, please describe your feature. Only describe things you implemented.**

The captured piece display is in the main method of Game.java.

**In 1-2 sentences, please describe how you tested your feature.**

**What was the agreed category of your feature using the buckets paradigm?**

( ) Standard

(X) Sprinkle

( ) Sparkle

Other Project Information:

What would you try to add to your project if you had more time to implement extra features?

What did you learn by working on this project?

What advice would you give to students next year, to help them have a better experience working on their projects?

# 5. Extra credit

*You should use this section to mention any of the extra credit items you did. Please provide justification for each one (for example, for the version control item you should have a link to your project on a website like GitHub)*