Using JSwing in Java, we created a checkers and chess game. This is the classic versions of two great games. This semester long project helped us with programming design, planning and implementation.

Checkers & Chess

PROG 3300
Integrated Project for Programming
TEAM JOE DINGUS
Winter 2025

Jeremy, Peter, Brandon

Contents

Introduction

What we intend to create with our skills is a few board-based games using Java and it's Swing libraries. We will make recreations of the famous board games, Chess and Checkers. In terms of what problems, we hope to solve with this project is the simple problem of boredom. We shall use IntelliJ as our IDE, as well as other technologies like Microsoft Teams, Discord, and GitHub for communications and code sharing.

Project Background

We're interested in this project because we want to give our own take on these classic games and have the ability to mess around with them in ways that you cannot with standard digital versions of Chess and Checkers. What if we want to have a game of all pawns? You can't do that online, but we'll be able to have the freedom to do whatever our hearts desire by making these games ourselves.

Project Deliverables

MILESTONE 1

The very first milestone for our project is simply to get a graphical interface up and running with 2 major elements. The first being the board, as it will be the same between both the Checkers and Chess games, as well as a popup menu of some sort to choose between the different games.

MILESTONE 2

Milestone two will be the game of Checkers itself, as the simplicity of the game will let us focus on the game functions, rather than the game so to speak. We will be able to focus on letting the user make moves, which then will later be able to be adapted into chess. The same goes for the pieces, since Checkers has two (the starting pieces and the "queen" pieces) which can also then be adapted to chess. As a result, this will largely focus on movement and rules.

MILESTONE 3

Milestone three will involve the first aspects of the Chess portion of this project. Since Chess is a more complex game, this third milestone will focus on the pieces themselves. Each piece moves differently from every other piece, so we will try to implement that logic for this milestone.

MILESTONE 4

This final milestone will be the latter half of the Chess portion of this project. It will involve more of the subtle complexities of chess, including the non piece specific rules, pieces being able to take other pieces, alternative win conditions (i.e. Checkmate) and potentially other elements like a move timer.

Roles & Responsibilities

Brandon MacCallum:
Roles:

>Programmer
>Stakeholder
>Technology Supervisor (In charge of Git, merge conflicts, etc.)

Peter Vaughan:
Roles:

>Programmer
>Stakeholder
>Chief Technology Officer (In change of design, mock-ups, etc.)

Jeremy Whitenect:
Roles:

>Programmer
>Stakeholder
>Scrum Master (in charge of project management)

Risks

1. Not meeting deliverables on time.

   This will be managed through our bi-weekly meetings, us giving updates, concerns, ideas and suggestions on how to keep our project running on time. Reflecting on what has been done as well as our meeting minutes will help with this as well.
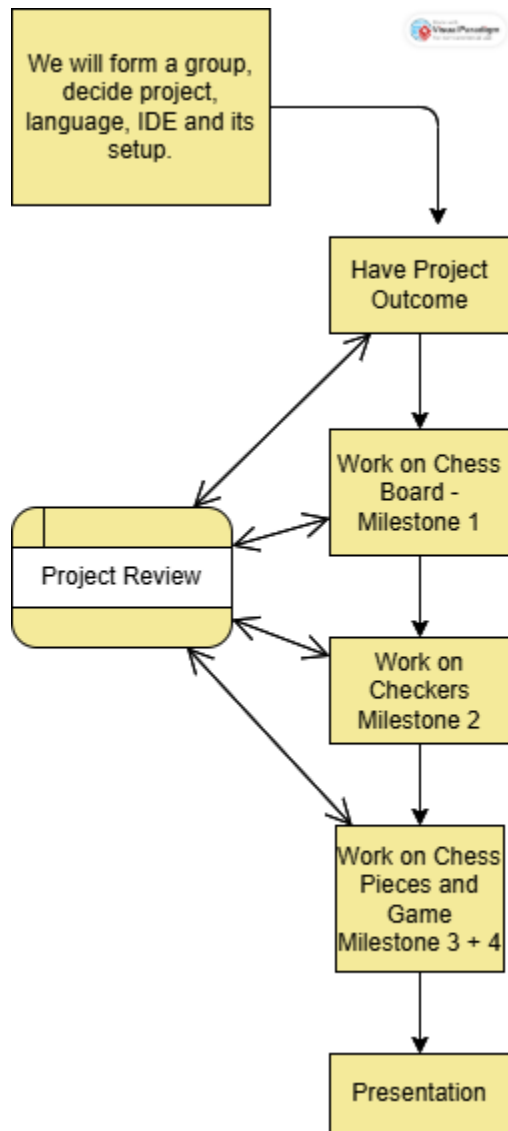
2. Loss of code/work.

   Through our uses of GitHub, as well as each of us having local copies of our project, no work should be lost. But to prevent this further, branches will be used for potentially dangerous changes, so they are not made on the main branch.

Investment/Running Costs

Description	Cost
IntelliJ (assuming non student/corporate plan)	$550 CAD
Outside Contractor (Teacher a.k.a Cost of Tuition)	$2500 CAD
Total Cost of Outside Contractor (Cost of Entire Course):	$10,000 CAD
Total Cost	$10,550 CAD

Workflow Diagram

```
┌─────────────────────┐
│ We will form a group,│
│ decide project,      │          Visual Paradigm
│ language, IDE and its│
│ setup.               │
└─────────────────────┘
                          │
                          ▼
                 ┌──────────────────┐
                 │ Have Project     │
                 │ Outcome          │
                 └──────────────────┘
                          │
              ▲           ▼
              │  ┌──────────────────┐
   ┌──────────┐ │ Work on Chess     │
   │          │ │ Board -           │
   │ Project  │◄┤ Milestone 1       │
   │ Review   │ └──────────────────┘
   │          │          │
   │          │◄─┐       ▼
   └──────────┘  │┌──────────────────┐
              ▲  ││ Work on          │
              │  ┤│ Checkers          │
              │   │ Milestone 2      │
              │   └──────────────────┘
              │           │
              │           ▼
              │   ┌──────────────────┐
              └──►│ Work on Chess     │
                  │ Pieces and       │
                  │ Game             │
                  │ Milestone 3 + 4  │
                  └──────────────────┘
                          │
                          ▼
                  ┌──────────────────┐
                  │ Presentation     │
                  └──────────────────┘
```

Lesson Learned

All of our team members learned a lot with this project. We explored JSwing and its components. There are ways to expand these components into interesting code and how it is graphically represented. We had issues with program while running into bugs but we knew how to deal with these bugs. We used concepts like logs to troubleshoot code. Also, we learned about UI elements and what we can do with these elements.

We were constantly learning. Refactoring helped us to program better in Object-Oriented Programming. This will help us in industry.

Project Performance Summary

We have weekly reports to keep this project on track. We were on track on this project and completed it on time.

Final Product
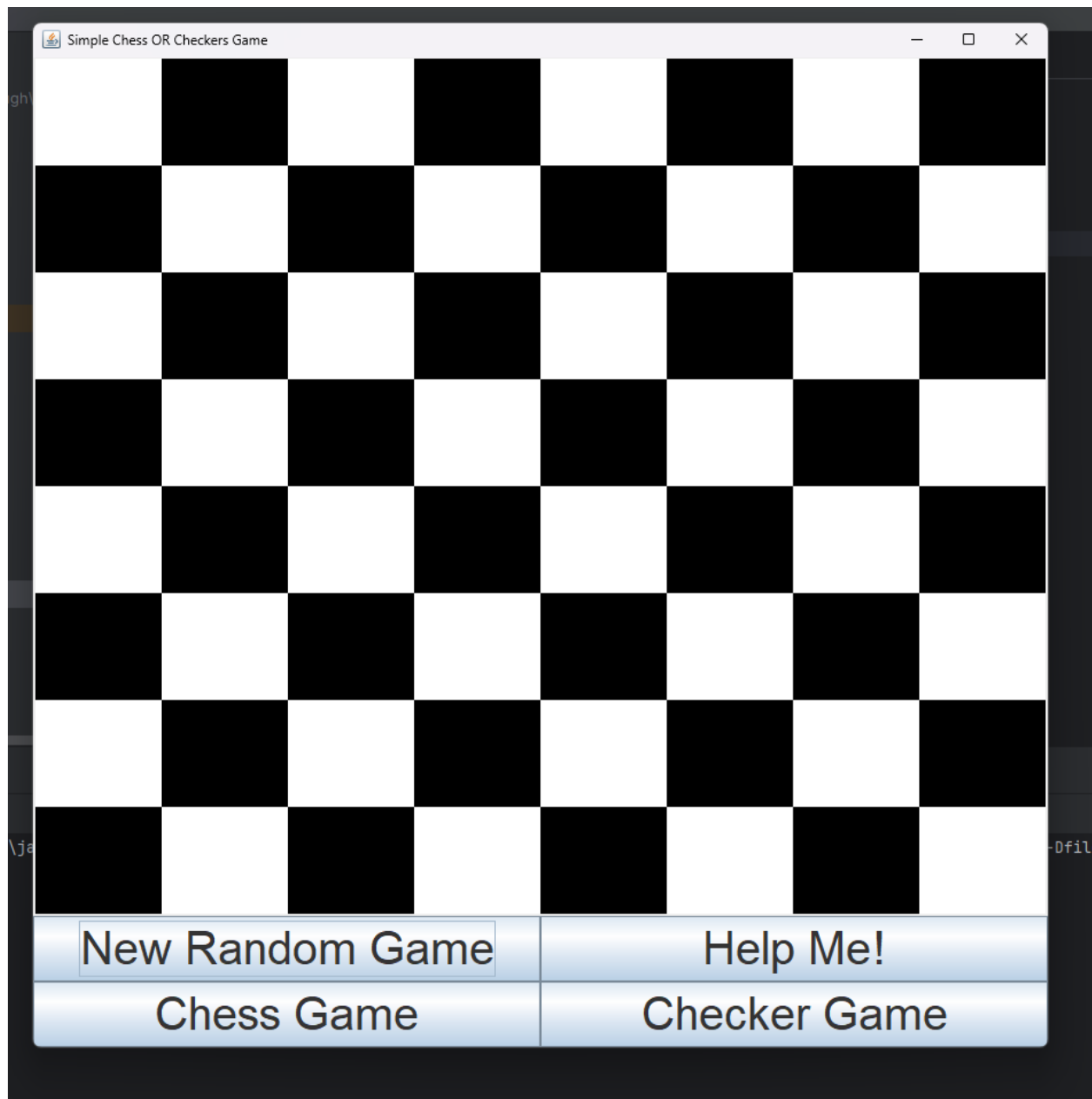
The board can play checkers or chess. Users can select the game or they can choose random. The board has a help me button, which send user to two websites. One for checkers gameplay help. The other for chess gameplay help. We have a folder for more images.

Checkers

This is the classic game of checkers. Red goes first and then alternates turns, moving diagonally on the board. The goal is to capture all of the opponent's pieces or block them so they cannot make a legal move. You can move your pieces one step diagonally forward, but if an opponent's piece is in an adjacent diagonal space and the space beyond it is open, you can "jump" and capture their piece. We have a folder for more images.

Chess

While there are variations on chess, this is the classical chess as the two-player strategy board game. The goal is to checkmate your opponent's king, meaning the king is in a position where it cannot escape capture. We have a folder for more images.

Pieces and their movements:

King: Moves one square in any direction. The most important piece — losing your king means you lose the game. However, you cannot put yourself in check or checkmate.

Queen: Moves diagonally, horizontally, or vertically any number of squares. The most powerful piece.

Rooks: Move horizontally or vertically any number of squares within the board or it comes into contact of another piece.

Bishops: Move diagonally any number of squares within the board or it comes into contact of another piece.

Knights: Move in an "L" shape: two squares in one direction and then one square perpendicular. They can jump over other pieces and "lands" on a square, potentially with an opposing piece.

Pawns: Move forward one square, but capture diagonally. On their first move, they can move forward two squares. If they reach the opponent's back row, they can be promoted to any piece (except a king), usually a queen.

Basic rules and variations:

- Players take turns, moving one piece at a time.
- You cannot move a piece to a square occupied by one of your own pieces.
- You can capture an opponent's piece by moving to its square (except for pawns, which capture diagonally).
- A game can end in checkmate (win), stalemate (draw), or a variety of other conditions such as timed games.

Special moves, which could be implemented:

- **Castling**: A move involving the king and one of the rooks that helps protect the king and develop the rooks. The king moves two squares toward the rook, and the rook moves to the other side of the king.
- **En passant**: A special pawn capture that can occur if a pawn moves two squares forward from its starting position and lands beside an opponent's pawn. The opponent can capture it as if it had only moved one square.
- **Pawn promotion**: When a pawn reaches the opponent's back row, it is promoted to a queen, rook, bishop, or knight.

The game is won by checkmating the opponent's king, where the king is under attack and cannot escape capture. While we did not implement every possible scenario in this game overview but a draw can occur through stalemate, insufficient pieces or a repetition of moves.

The Buttons

The "New Random Game" button goes to a random new game between the checker game and the chess game. If we wanted to add more games to this, we could. The "Help Me!" button goes to two websites for more help about checkers and chess. The "Chess Game" goes to a new chess game while the "Checker Game" goes to a new checkers game.

Recommendations for Improvement

Improvements to add

- While this project is OOP, it could have been better with OOP. This will come with more experience.
- There could more games to our board. This could include custom games such as an all-pawn chess game.
- We could have more rules to our games.
- We can do suggested moves as well.
- We moved pieces by moving JLabels but there might be better options to moving pieces.

Improvements to us

- We could be learning more OOP.
- We could explore the details with project management.

Improvement to the supervisor/teacher

- We should more structured week to week reports and expectations.
- We should have more feedback than "Ack".

Future directions

We can do many more improvements if we have the time and resources to do such tasks. This would be fun to explore more if we have the time to do so.

- We could use a different library than JSwing.
- We can make this program online for online play.
  - We could explore ways to host games.
- We can create more games (custom or not).
- In checkers, we can jump multiple pieces.
- We could have a scoreboard for players.
  - We could have more of a generic one for white or red side vs black side.

Submission

GitHub Repo: **https://github.com/JermNet/IntegratedProgrammingProject**

README File

Video Recording

In-Class Presentation

PDF