Presentation:

**Team Members:**

Phillip Neidlinger, Noah Schuler, Zackery Kim, Ashley Simpson

**Project Abstract:**

This project will allow two users to play a game of checkers against each other. There will be an eight by eight board, with each person starting with twelve checkers. The users’ checkers will be placed on the same diagonal path, allowing them to jump over, and therefore remove, the other user’s checker. A checker can only move in a diagonal manner, and can only move one space, unless an opponent checker is in the space, then it can jump over the checker. King checkers are made when a user moves their checker to the farthest row from where they started. A king checker can move both directions, while a normal checker can only move away from where each user started individually. The first person who jumps all of the other users checkers will be the winner.

**General Overview of Division of Labor:**

There will be four classes that will make up this project. The first class will be a class called Main. The second class will be a class called CheckersBoard. The third class will be a class called Checker. The fourth and final class will be a class called King.

**Class Descriptions:**

The Main class will be the class that runs the game. A game loop will be created using multiple functions to make the game playable, and Phillip will be responsible for this class. The CheckersBoard class will create a board of objects that will be used as the game board, and Zackery will be responsible for this class. The Checkers class will create an object that will be used as the checker pieces that the user is playing with, and Ashley will be responsible for this class. The King class will be a version of a checker that will be allowed to move both forward and backward on the game board, Noah will be responsible for this class.

**VIDEO**

Errors:

6,2 – 5,2 = INVALID; cannot go straight up

6,2 – 7,2 = INVALID; cannot go straight down

6,2 – 6,3 = INVALID; cannot move to the side

6,1 – 5,2 = INVALID; no checker at location

6,2 – 5,1 = VALID

5,1 – 4,2 = INVALID; PLAYER DID NOT CHANGE

…

4,2 – 6,4 = VALID; jumped a piece \*at 1:05

…

2,2 – 1,1 = VALID; king is made and shown with a capital ‘X’ \*at 3:00

…

1,1 – 3,3 = VALID; king can move backwards and jump over a piece \*at 4:30

…

\*at 5:06 = king is shown moving both ways

…

6,6 – 8,4 = VALID; king is made and shown with a capital ‘O’ \*at 6:17

…

\*game ends and player 2 (o’s) wins