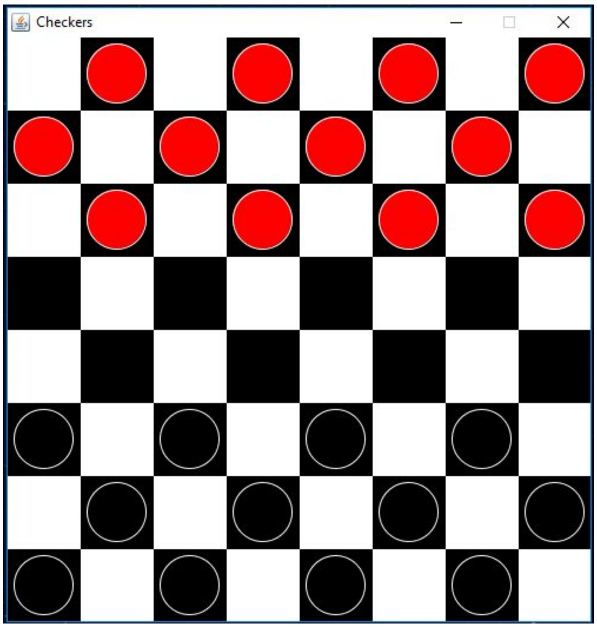
GCI-CS-CHECKERS v1.0.0

An implementation of Checkers written in Java, using the Swing library.

Written by John Andrew Oss

How To Play:



Setup:

There are two players, Red Team and Black Team.
Each team starts with 12 checkers.
These checkers are placed on a 8x8 board, with 32 spaces being available to move to.

The goal is to capture or “jump” all of the opponent's pieces, so they have none.

Black moves first, then the players alternate turns moving.

Movement:

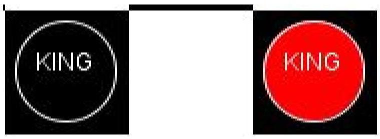
Basic movement is to move a checker one space diagonally forward. You can move two spaces diagonally forward but only when capturing an opponent’s piece.
You can not move a checker backwards until it becomes a King.

Jumping:

If one of your opponent’s checkers is on a forward diagonal next to one of your checkers, and the next space beyond the opponent’s checker is empty, then your checker can jump the opponent’s checker and land in the space beyond. Your opponent’s checker is captured and removed from the board.

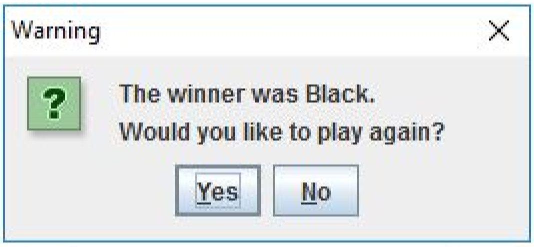
Becoming a King:

When one of your checkers reaches the opposite side of the board, it becomes a King.
Kings can move in either direction, instead of merely forward, they can move backwards as well.
Kings look like normal checkers, but they have the word "KING" drawn in the centre of the checker as seen in the image to the right.



Winning:

When all of yours or your opponent's checkers are captured, there will be a prompt asking you if you'd like to restart the game. To continue playing a new game, click the "Yes" button in the opened dialog. The game will then reset to the beginning board state.


Warning

The winner was Black.
Would you like to play again?

This covers all basic rules outlined in the v1.0.0 release of GCI-CS-CHECKERS. Potential rule changes may occur, and this will be updated to reflect those changes. Good luck, and have fun!