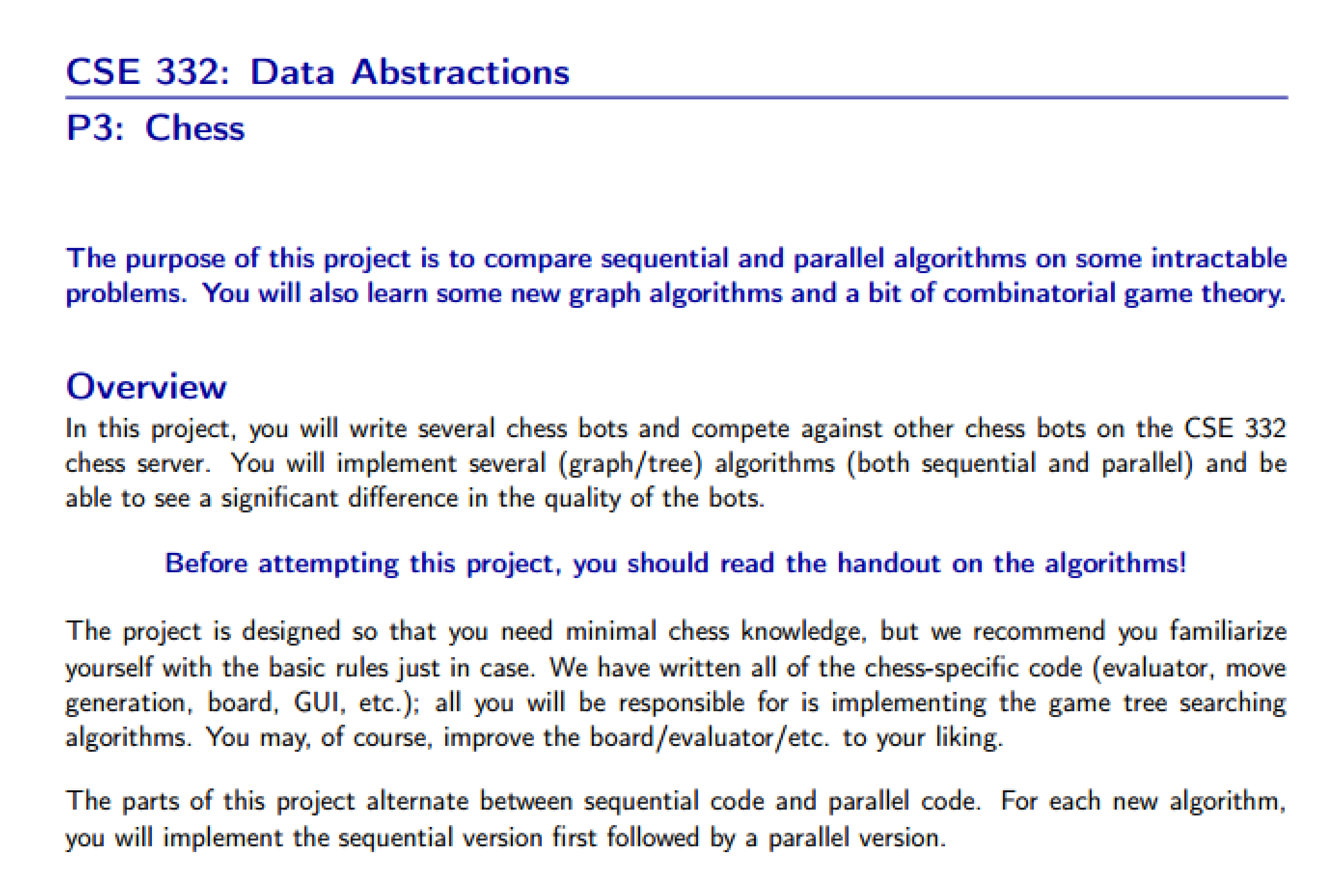
Project 3: Parallel Game Search for Chess

CSE 332

Spring 2016

Project 3, Due June 1, 2016



CSE 332: Data Abstractions

P3: Chess

The purpose of this project is to compare sequential and parallel algorithms on some intractable problems. You will also learn some new graph algorithms and a bit of combinatorial game theory.

Overview

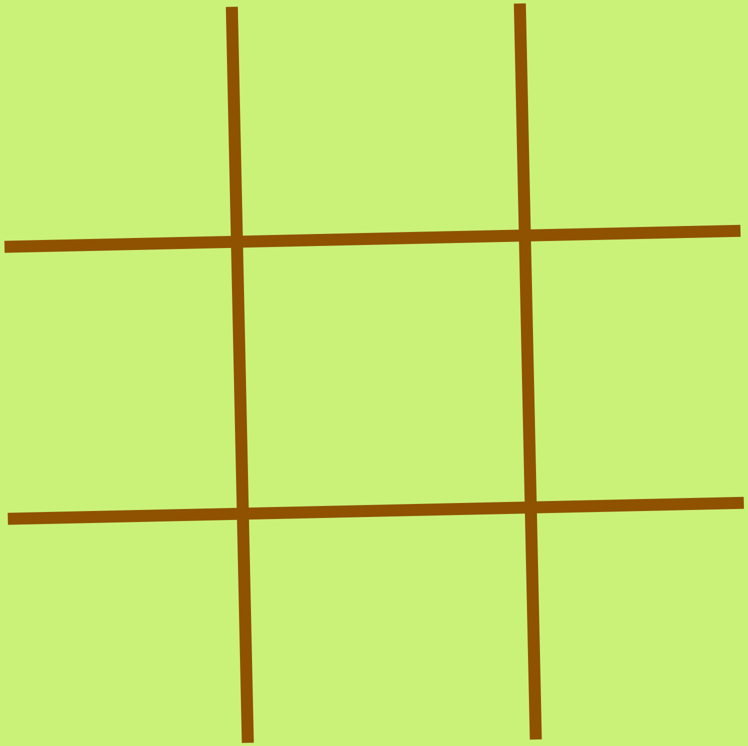
In this project, you will write several chess bots and compete against other chess bots on the CSE 332 chess server. You will implement several (graph/tree) algorithms (both sequential and parallel) and be able to see a significant difference in the quality of the bots.

Before attempting this project, you should read the handout on the algorithms!

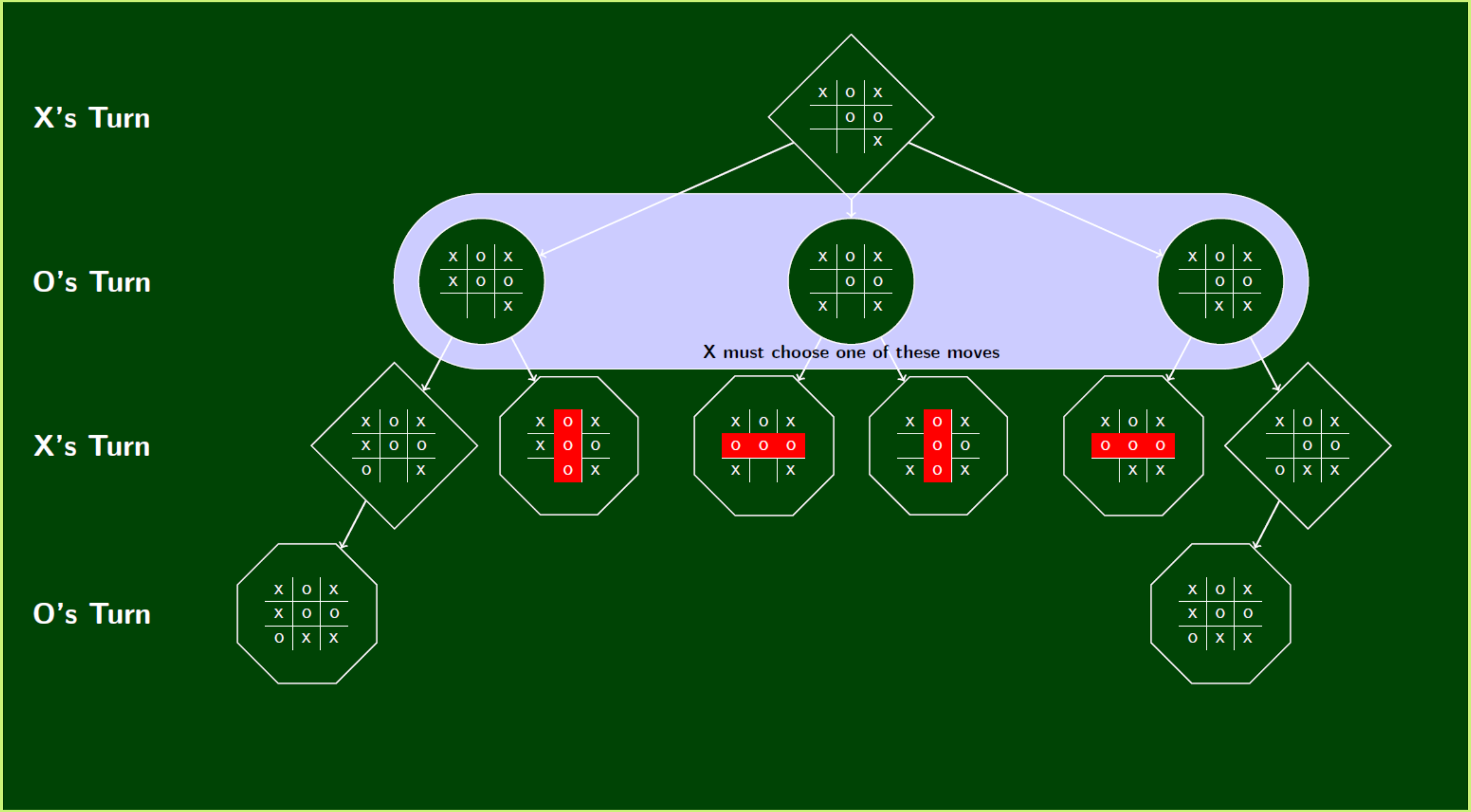
The project is designed so that you need minimal chess knowledge, but we recommend you familiarize yourself with the basic rules just in case. We have written all of the chess-specific code (evaluator, move generation, board, GUI, etc.); all you will be responsible for is implementing the game tree searching algorithms. You may, of course, improve the board/evaluator/etc. to your liking.

The parts of this project alternate between sequential code and parallel code. For each new algorithm, you will implement the sequential version first followed by a parallel version.

Tic-Tac-Toe



Game Tree


X's Turn

O's Turn

X's Turn

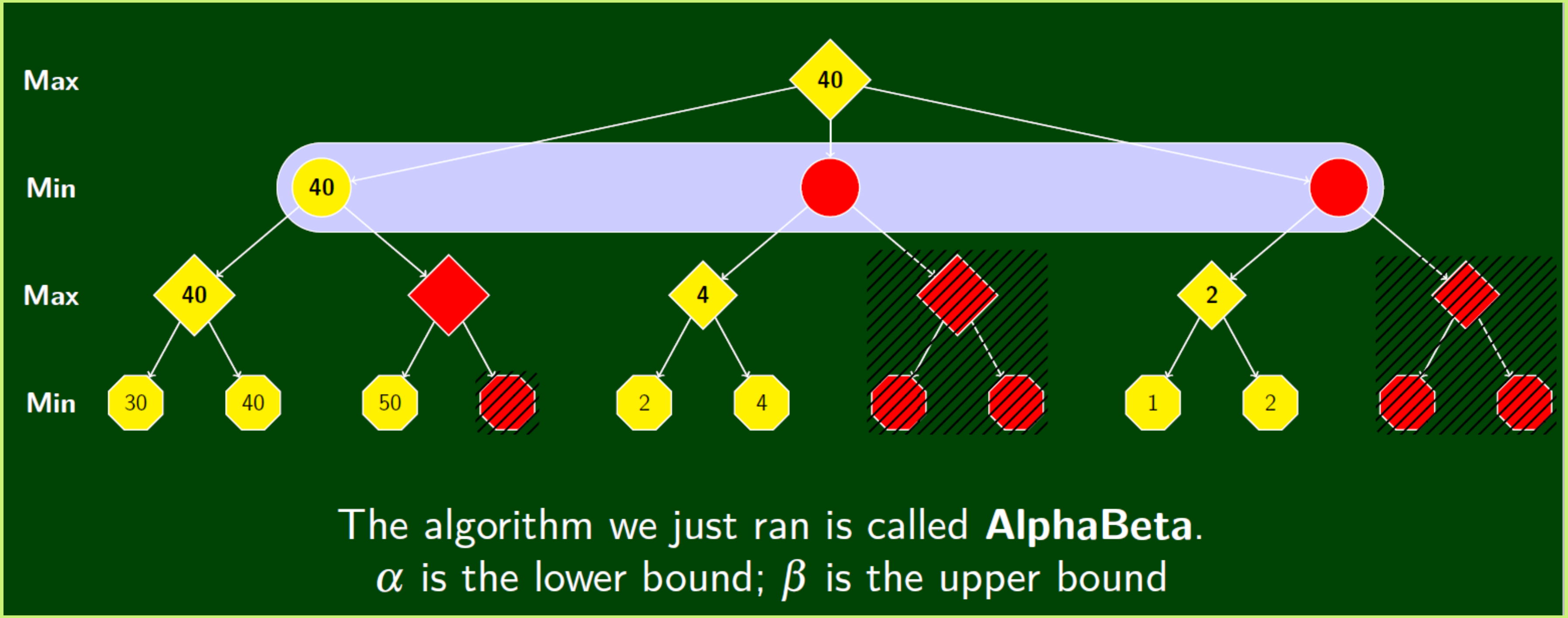
O's Turn

X must choose one of these moves

Min-Max search

- Tree with numeric outcomes at leaves
  - Player 1 wants to maximize score
  - Player 2 wants to minimize score
- Computing values
  - Leaves return values
  - Max levels return max of children
  - Min levels return min of children

Alpha beta Pruning



The algorithm we just ran is called **AlphaBeta**.

α is the lower bound; β is the upper bound