Software Engineering Group Project

Project Crawl Self Assessment Report

Team: Team Eagle
Team Members: Luke Dixon, Iona Ryder, Marco Costa, Nasim Ali, and Natalie Aitcheson

For each of the following requirements, indicate (YES/NO) whether your submission meets the requirement.

Deployment requirements: Does the entire application compile from source, without error, by executing the following command from the root directory of the source code: javac MainApp.java **(YES)**

Does the application start and run without uncaught exceptions when executing the following command: java MainApp **(YES)**

Must-have requirements:
1. The submitted source code must meet the requirements specified in the section on Project Crawl deliverables. **(YES)**
2. The submitted source code must compile and execute using the commands specified above**.(YES)**
3. The application must allow two players sharing a single computer to play the game:
3.1 In two player mode, the game selects a random player to make the first move. **(YES)**
3.2 The application must be able to display the initial board. **(YES)**
3.3 The game must display the number of seeds in each house, either using images or numbers. It is not required to hide the number of seeds in a house from an opponent. **(YES)**
3.4 The application must be able to redistribute seeks according to the rules of the game. When selecting a house with 12 or more seeds, the starting house should be skipped as specified in the rules. **(YES)**
3.5 The application must implement the capture rule correctly, including the prohibition on capturing all of the opponent's seeds. Capture moves are not applicable to the player's own houses. **(YES)**
3.6 The application must recognise when one player has won the game. **(YES)**
3.7 The application must implement the rule that if the opponent has no seeds, and one or more moves is/are possible that gives the opponent seeds, then only such moves are permitted**.(YES)**

Should/May-have requirements:
1. The application should include a graphical user interface that visualises the board, perhaps using sprites or other visual representations of the board. (**YES**)
2. The application should include a basic single player mode. A basic single player mode is very easy to implement: just get the computer player to make random moves. (**YES**)
3. The application should include animations visualising the process of redistribution. Be careful if you decide to attempt this as developing animations can potentially be very time consuming if you are not experienced with this. Also note that this would not affect your mark much. (**YES**)
4. The application may include an AI player, using a basic search algorithm that explores all combinations of possible future moves up to a very limited number of moves, and identifies the best possible move for the AI player assuming the opponent also makes optimal moves for the opponent. It is extremely challenging to add this to the project in a single week and it does not affect your mark by much. (**NO**) Are there any limitations you wish to report? *insert any limitations you are aware of here* Are there any particular strengths you wish to report? *insert optional comments here*