Team 4 – Project 2

Backlog, Initial Burndown Chart, & Sprints

Members: Patrick Casey, Matt Hacker, and Megan Kerins

CSCE 315-501

Checkpoint #1

First Deadline

For our project, our "SCRUM" master/project manager will be Megan Kerins as she has the most experience with Java out of the group.

<u>PRODUCT BACKLOG:</u>

- User launches program a window opens with the title of the game and creators.
- Next the display switches to the actual interface
- There will be buttons on the top of the board reading "New Game", "Board State", and "Undo"
- New Game will simply reset the pieces to how they were when the game was started. No Titles or creator names will display
- Board State will either open a separate window of the *current* board state, or simply pause the game and display it over the screen
- If the board state is displayed over the screen, a button will be in the upper right corner that says "Back" which takes you back to the game of play
- A box to input the board size will be available before the game begins
- An option to play locally against an AI will be displayed
- An option to play locally against another person will be available
- An option to play against a remote opponent (regardless of AI/person) will be available
- The initial board state is displayed if the Board State button is pushed before game play
- "Undo" undoes the previous move, this is only possible for the Human player
- Before the game can be started the AI intelligence must be chosen (possibly before interface is displayed)
- Three Buttons will be listed somewhere on the window or this will be a choice from the command linne
- Create a utility function
- Generate a minimax tree
- Evaluate the tree with alpha-beta pruning one level at a time
- The AI will use the minimax tree, the difficulty level will determine how far ahead it looks
- There might be a "Tree" button for the player to see the minimax tree as well
- Easy will make the AI look ahead 1 move
- Medium will allow the AI to look ahead 2 moves
- Hard will allow the AI to look ahead 3 moves
- Our board will have a graphical interface, not command line
- P1 will move by a click-and-move process
- User gets response of invalid if proposing an invalid move
- User also gets a response if they still have valid moves to initiate
- Whether the move is valid or invalid, either a green/red box or simply text (saying P1 or invalid) will display at bottom of screen alerting the player
- After the turn is over, the box will turn red or the text will switch over to AI
- The user sees a display of the current board state, again to exit the button "Back" must be clicked

- Board state is checked to determine if either player has won, and is checked after every move
- Game is checked to determine if maximum number of moves has been exceeded. This too is done after every move.
- Possibly a counter will be displayed counting down the moves until they are exceeded
- Now it is the computer's move (AI)
- Computer identifies all valid moves it can make
- AI then checks the board state (it does not appear across the screen this is done internally)
- Computer looks ahead in tree (again, done internally. How far ahead depends on its skill level)
- After every move, Human or Computer, the Minimax tree in the Computer's class is updated for its best possible move
- Computer can ONLY select valid moves
- Like the user, after turn is over the board state is updated and encouraged to be clicked
- "Back" must be selected to return to game
- The game is again checked to determine if maximum number of moves has been exceeded.
- Iterative deepening implemented
- Board evaluation function works for determining winner/loser
- At the end of the game the victor's name will be displayed
- The board will have no pieces on it and the option to play again will be offered

<u>BURNDOWN CHART:</u>

Main priority is to actually have an interface for the game with all of the pieces on it in the correct positions and correct colors execute when the program is run. This includes having a window pop-up displaying the name of the game and team member names, then after say 5 seconds transition to the actual interface of the game that will be played. Being human, we have the initiative and are allowed to start the game.

All the buttons should at least be there, whether they work or not is not an issue. For the first sprint we want the game to have all its pieces ready to implement.

Time Estimation:
Total initial graphic Interface: 28 Hours
    Startup Scenario-18 Hours
    New Game Options- 5 Hours
    Board States- 5 Hours
Game Implementation: 35 Hours
    Minimax Tree- 8 hours
    Piece Movement- 5 hours
    Board State updating-7 hours

- Game State checking- 10 hours
- AI difficulty level- 5 hours

Client Server Interface: 20 hours

- Client end- 10 hours
- Server end- 10 hours

General debugging and fixes: 32 hours

Total Estimation: 125 hours