Neon Unicorns
Alvin Ye, Zane Wang, Ahnaf Hasan

APCS pd1
1/4/18

Final Project Proposal

Tic-tac-toe in the Terminal

Woo.java will print an empty tic-tac-toe board in the terminal with each run, and start the game, which will run until the game ends with a tie or a win.

The terminal will accept a row and a column input as a coordinate for where to put the user's "x".

It will be a one-player program where there will be 3 levels of AI the user can choose from.

Whoever goes first will be randomly determined.
(For the easy and medium mode, the AI will randomly put an "o" on the tic tac toe board to start off the game)

The "easy" mode will allow the user to go first, and the AI will randomly put an "o" next to any "x."

The "medium" mode will have the AI put an "o" next to an "x" and check for instances of 2 "o"s in a row, column, or diagonal to block off the win.

The "hard" mode will have the AI will check for instances of 2 "x"s in one row, column or diagonal to block off the win. If the AI goes first it will always place an "o" in the center of the board. It will implement strategy (like 2-way death traps).

We plan to use 2D arrays to represent a tic-tac-toe board and we will need to compare elements of the list horizontally, vertically, and hardcode situations where there is a diagonal of 3 matching "o"s or "x"s.

We will implement Keyboard.java to make it easier to accept and read terminal inputs.

Instance Variables:

- **String[][] Board** is the 2D array which houses the “o”s and “x”s and is the tic tac toe board

Methods include:

- **toString()** will return the string representation of the 2D array
- **startGame()** which starts a new session (if AI is first, the board starts with one “x” already in place)
- **isWin()** which checks horizontally or vertically for 3-in-a-row
- **isWinDiag()** which checks to see if either diagonal has 3-in-a-row
- **input()** which places an x or an o in the tic-tac-toe board
- **isEmpty()** which checks to see if a spot in the tic-tac-toe board is occupied
- **checkTwo()** which checks to see if there is a scenario with 2 “x”s or 2 “o”s in a column or row or diagonal (used by the “medium” and “hard” AI)
- **isNextToO()** which checks if the value horizontally or vertically adjacent to the index is an “o” (used by the “easy” AI)
-