project[0],

due 9/28/2020, midnight (3x3 Tic-Tac-Toe)

Purpose: *The purpose of this assignment is to provide a basic programming experience with an algorithm commonly-used in AI. You should be able to demonstrate the ability to utilize a recursive algorithm in a game-playing scenario.*

Skills demonstrated:

* using a github repository to download code
* c++ programming
  + recursive functions
  + working with an established API
  + review of basic concepts from earlier programming classes
* adversarial search
* game-playing

Please write a program to play 3x3 tic-tac-toe. The program should use minimax search.

You do not need to play the entire game, you just need to make one move. For example, the program should be executed like:

$ ./tictactoe <infile> <outfile>

infile: file to be loaded with a board state

output: file to be saved with board state after a move is made

Your program should make a single move (x,y) which will be written to a file <outfile>, and exit. If there is an immediate win in the moves available, make that move. Otherwise, do minimax on all available moves.

The github repo: <https://github.com/UNR-RoboticsResearchLab/cs482-fall2020.git> has some example code (compilation instructions are provided in the readme file).

The provided code has a main file that reads in the command-line arguments described above and has the following implemented functions:

/\*\*

load\_board: loads a tic-tac-toe board state from a file

args:

std::string filename: file to read the board from

int [][3] board: 3x3 array of ints representing the board state

0: empty

1: x

-1: o

returns (int):

0: if file loaded OK

-1: if file doesn't exist

\*\*/

int load\_board( std::string filename, int board[][3] );

/\*\*

save\_board: saves a tic-tac-toe board state to a file

args:

std::string filename: file to write the board to

int [][3] board: 3x3 array of ints representing the board state

0: empty

1: x

-1: o

returns (int):

0: if file opened OK

-1: if file can't be opened

\*\*/

int save\_board( std::string filename, int board[][3] );

/\*\*

make\_move: takes a board state and makes a legal

(hopefully optimal) move

args:

int [][3] board: 3x3 array of ints representing the

board state. The values of board are altered based

on the move

0: empty

1: x

-1: o

returns (int):

the number of steps it took to choose the best move

(returns 1 by default, 0 if no move made)

\*\*/

int make\_move( int board[][3] );

You'll be wanting to edit the make\_move function. All code needed to make this work needs to be contained in the tictac\_turnin.cc file for the autograder to work.

**Graduate Student Extra Assignment:**

Please also write a program to conduct a full game (including saying who is the winner). This assignment will not be autograded.

**Code should be submitted using the submit script:**

(!!!do only once per computer!!!) first, download the submit script:

$ cd ~

$ wget http://www.cse.unr.edu/~newellz2/submit

$ chmod +x submit

then, from your project directory (1-tictactoe):

$ ~/submit

this will submit all files from the current directory

***(login: netid / password: passwd for your netid)***

**Your turned in directory should have the following files:**

.

+-- CMakeLists.txt

+-- include

+ +-- tictac\_support.h

+-- src

+-- tictac\_support.cc

+-- tictac.cc

+-- tictac\_turnin.cc (all other code files

needed to build your programs)

it is acceptable to turn in files that are not in this list, but these have to be turned in.

// must build executable named:

// tictactoe: 3x3 tictactoe player for a single move