//Zhihao Jin

// maze.cpp

#include "maze.h"

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

Maze::Maze (std::string file\_name)

{

std::ifstream input\_file{ file\_name };

if (input\_file)

{

std::string line;

getline (input\_file, line);

while (input\_file)

{

maze.push\_back (line);

getline (input\_file, line);

}

}

else

{

std::cout << "ERROR: cannot open file " << file\_name << std::endl;

}

}

void Maze::print\_maze (void)

{

for (auto line : this->maze)

{

std:: cout << line << std::endl;

}

std::cout << std::endl;

}

void Maze::write\_maze (std::ofstream &results\_file)

{

for (auto line : this->maze)

{

results\_file << line << std::endl;

}

results\_file << std::endl;

}

bool Maze::find\_start (size\_t &row, size\_t & column)

{

for(size\_t i = 0; i < maze.size(); i ++)

{

for(size\_t n = 0; n < maze[i].length(); n++)

{

if(maze[i][n] == START)

{

row = i;

column = n;

return true;

}

}

}

return false;

}

bool Maze::find\_path (size\_t row, size\_t column)

{

if(maze[row][column] == EXIT )

return true;

if(maze[row][column] == WALL || maze[row][column] == EXPLORED|| maze[row][column] == EXPLORED || row == maze[0].length() - 1 || column == maze.size() - 1 )

return false;

else

{

maze[row][column] = EXPLORED;

if(find\_path(row, column+1))

{

maze[row][column] = PATH;

return true;

}

if(find\_path(row+1, column))

{

maze[row][column] = PATH;

return true;

}

if(find\_path(row, column-1))

{

maze[row][column] = PATH;

return true;

}

if(find\_path(row-1, column))

{

maze[row][column] = PATH;

return true;

}

}

return false;

}