#include "Ant.hpp"

#include "critter.hpp"

#include <iostream>

using std::cout;

using std::endl;

void Ant::move(int row, int col, Critter\*\*\* gameBoardPtr, int numberOfRows, int numberOfColumns)

{

CritterDirection newDirection;

newDirection = nextDirection(); // This returns a random enum

if (newDirection == Up && gameBoardPtr[row - 1][col]->getBugLetter() == ' '

&& row != 1 && gameBoardPtr[row][col]->getAlreadyMoved() == false)

{

delete gameBoardPtr[row - 1][col]; // Delete the blank generic Critter there

gameBoardPtr[row][col]->setAlreadyMoved(true);

gameBoardPtr[row - 1][col] = gameBoardPtr[row][col]; // Our Ant moves up one row

gameBoardPtr[row][col] = new Critter(' ');

cout << "Ant moved Up!" << endl;

}

else if (newDirection == Down && gameBoardPtr[row + 1][col]->getBugLetter() == ' '

&& row != numberOfRows && gameBoardPtr[row][col]->getAlreadyMoved() == false)

{

delete gameBoardPtr[row + 1][col];

gameBoardPtr[row][col]->setAlreadyMoved(true);

gameBoardPtr[row + 1][col] = gameBoardPtr[row][col];

gameBoardPtr[row][col] = new Critter(' ');

cout << "Ant moved Down!" << endl;

}

else if (newDirection == Left && gameBoardPtr[row][col - 1]->getBugLetter() == ' '

&& col != 1 && gameBoardPtr[row][col]->getAlreadyMoved() == false)

{

delete gameBoardPtr[row][col - 1];

gameBoardPtr[row][col]->setAlreadyMoved(true);

gameBoardPtr[row][col - 1] = gameBoardPtr[row][col];

gameBoardPtr[row][col] = new Critter(' ');

cout << "Ant moved Left!" << endl;

}

else if (newDirection == Right && gameBoardPtr[row][col + 1]->getBugLetter() == ' '

&& col != numberOfColumns && gameBoardPtr[row][col]->getAlreadyMoved() == false)

{

delete gameBoardPtr[row][col + 1];

gameBoardPtr[row][col]->setAlreadyMoved(true);

gameBoardPtr[row][col + 1] = gameBoardPtr[row][col];

gameBoardPtr[row][col] = new Critter(' ');

cout << "Ant moved Right!" << endl;

}

else

{

cout << "Ant stayed put!" << endl;

}

// If none of these conditions met, the Ant will not move

}