

Том и Джери

изготвил: Георги Павлов, ФН: 80950, 7 група

Проектът използва само стандартната библиотека на C++. Примерите са подготвени в текстов файл в директорията на проекта.

Избрах да използвам лист за да реализирам полето понеже нормален масив не може да се инициализира по дадения начин (поне аз не се сетих как). Всяко квадратче от полето ми представлява структура с координати, символ, казващ какво има на квадратчето, булева променлива дали е посетено квадратчето и променлива за завоите. Завоите се пазят във всяко квадратче защото трябваше да се знае колко са били завоите на предното квадратче и нямаше друг лесен начин за достъпване. Реших да направя рекурсията за търсенето на пътя с итератори сочещи към определените полета, а не с изцяло нови квадратчета. Запомням правилните пътища във вектор от листи от пътища, а завоите в друг вектор. Когато приключи рекурсията спрямо вектора от завои гледам кой лист от пътища има най-малко завои и също най-малко квадратчета. Когато го намеря използвам итераторите за да променя символа на квадратчетата.

Структури

Проектът е съставен от един основен клас Дъска, съдържащ цялата логика, и една помощна структура Квадрат, нямаща никаква функционалност.

= **class Board** - Реализира логиката на програмата.

Член-данни:

- итератори, сочещи към Том и Джери.
- лист от квадратчета служещ за дъска
- променлива, в която се пази броя на завоите, за да може да се използва във функцията за записване на резултата.

Публични функции:

- **read(ifstream&)** - чете данните за една ,сценка‘, създава дъската и препятствията.
- **write(ofstream&)** – записва изходните данни във файл.
- **findPath()** – вика вътрешните функции, намиращи пътя.

Вътрешни функции:

- **void makeBoard(int, int, const int, const int, const int, const int)** – приема координати на дъската, Том и Джери и създава съответните елементи.

Създадена за да не трябва да се вика вътрешно конструктор.

- **makeObstacle(int*)** – приема масивче с 4 координати и създава препятствие по тях.

- **bool findSquare(const int, const int, I&)** – намира квадратче по координатите му и насочва дадения итератор към него, връщайки вярно при правилно извършена операция.

- **makeStep(I&, Direction&, list<I>&, vector<int>&, vector<list<I> >&)** – приема итератор към квадратче, посоката на последното извършено движение, лист със текущ път, вектор от завои и вектор от листи с пътища. Извиква последователно функцията за движение във всички посоки.

- **move(Direction&, I&, Direction&, list<I>&, vector<int>&, vector<list<I> >&);** - приема същите променливи като предната функция плюс посока, в която да се опита да продължи. В началото създава нов итератор сочещ към квадратчето в дадената посока (ако съществува такава), при възможност да се продължи по него добавя предното квадратче в листа с текущия път. Пренася завоите, направени до предното квадратче на новото, и ако за него е направен завои ги увеличава с едно. Прави проверка дали новото квадратче не е Джери и добавя завоите и текущия път в определените им вектори. Ако не е Джери прави стъпка.
- **getNewCoordinates(Direction&, I&, int&, int&)** – помощна функция, намираща координатите на следващото квадратче спрямо зададената посока.

= **struct Square** – създава квадратче с координати, символ, булева променлива, казваща дали е посетено квадратчето и променлива паеща броя направени завои. Конструктор инициализира квадратчетата със зададени стойности.