

Beadandó II

Magyar Tamás | RNYR2F

(1. Feladat)

Feladat Leírása

Potyogós amőba Készítsünk programot, amellyel a következő két személyes játékot lehet játszani. Adott egy $n \times m$ mezőből álló tábla (n az oszlopok, m a sorok száma), amelyre a játékosok X, illetve O jeleket potyogtatnak (azaz egy adott oszlopban a karakter mindig „leesik” a legalsó üres sorba, függetlenül attól, melyik sorban helyeztük le).

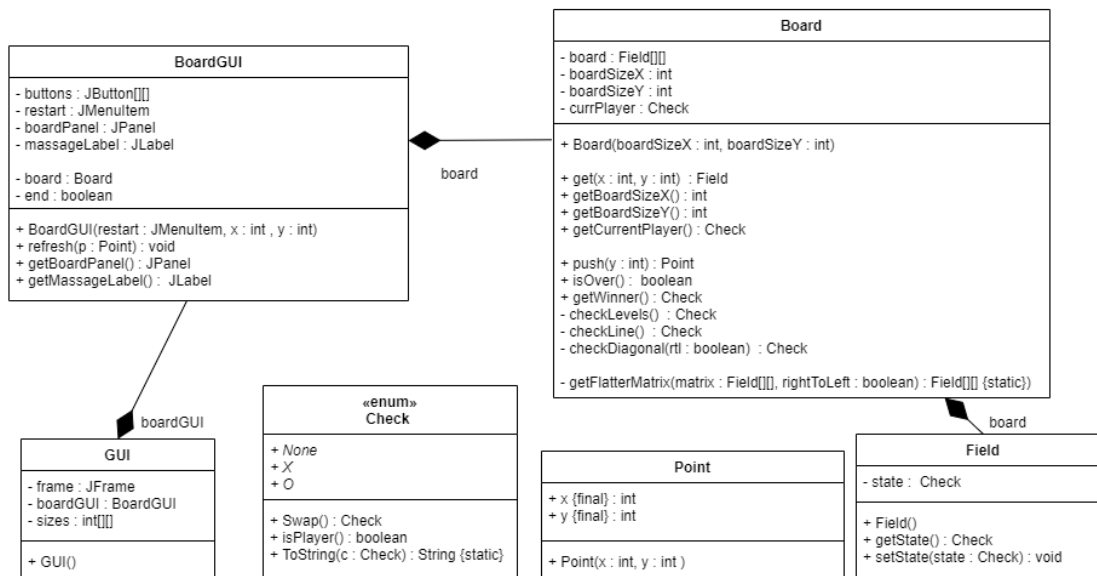
A játékosok felváltva lépnek, és egy oszlopban csak akkor helyezhetnek el új jelet, ha az még nem telt meg. A játékot az nyeri, aki előbb elhelyez vízszintesen, vagy átlósan négy szomszédos jelet. A játék döntetlennel ér véget, ha betelik a tábla. A program biztosítson lehetőséget új játék kezdésére a táblaméret megadásával (8×5 , 10×6 , 12×7), és ismerje fel, ha vége a játéknak. Ekkor jelenítse meg, hogy melyik játékos győzött (ha nem lett döntetlen), majd kezdjen automatikusan új játékot.

Feladat Elemzése

A feladat egy $n \times m$ -es játéktér megvalósítása ahol az adott (x, y) -on koordinátán kattintva befolyásolhatjuk a játékot, a kattintásnál csak az y koordinátáját kell figyelembe vennünk mivel a x tengelyen töltjük fel a játéktér. Mivel a játéktér változó méretű ezért általános algoritmussal kell megkeresnünk az egymás mellett és keresztben álló jelzőket.

Feladat Megvalósítása

UML Diagram



Esemény/Eseménykezelő

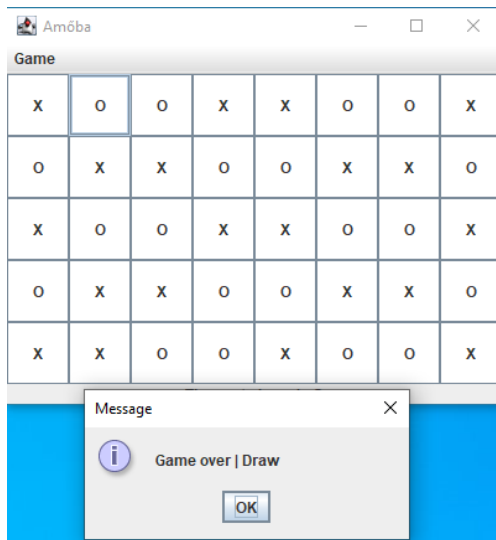
- Indítás -> új játék készítése
- Játék indító gombok > Action eseményére -> új játék indítása
- Játék mező gombok > Action eseményére -> egy jelző elhelyezése a játéktéren vagy új játék indítása

Tevékenységek

- **BoardGUI::refresh(p : Point)** – frissíti a megadott ponton a grafikus felület elemét, kiírja a jelenleg következő lépést
- **Board::push(y : int)** – az y koordinátán megpróbál lehelyezni a jelenlegi játékos számára egy jelzőt
- **Board::isOver()** – ellenőrzi hogy a játéknak vége van-e
- **Board::getWinner()** – lekérdezi a győztest ha van ha nincs None-al tér vissza
- **Board::checkLevels()** – ellenőrzi hogy van-e vízszintesen nyertes játékos
- **Board::checkDiagonal()** – ellenőrzi hogy van-e átlósan nyertes játékos
- **Board::checkLine(line : Field[])** – segéd függvény, ellenőrzi a kapott tömbben hogy van-e egymás követő 4 ugyanolyan jelző.
- **Board::getFlatterMatrix(matrix, rightToLeft)** - a mátrix átlóiból sorokat csinál

Tesztelés

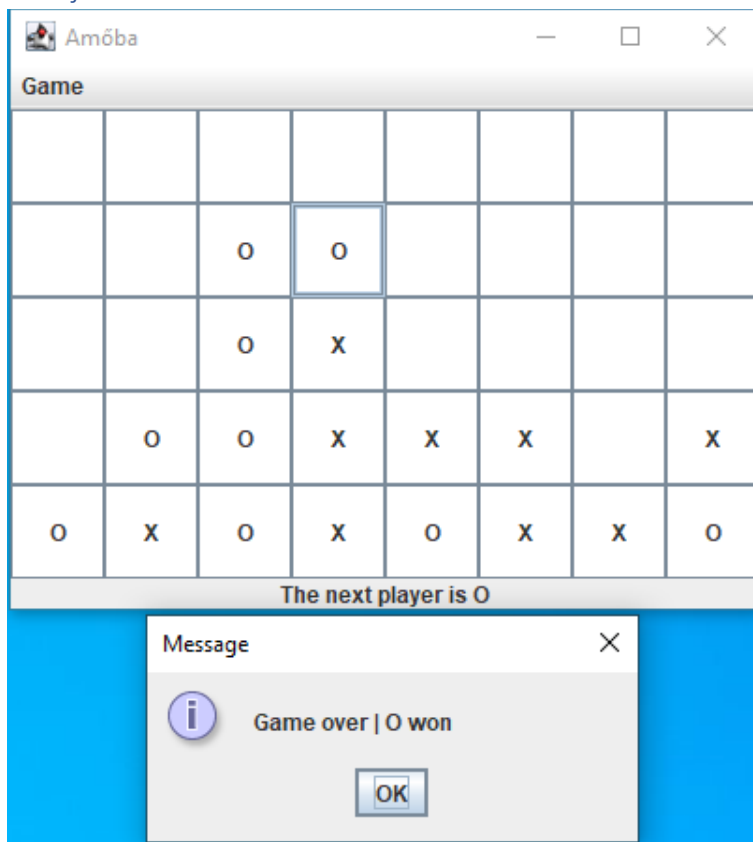
Döntetlen



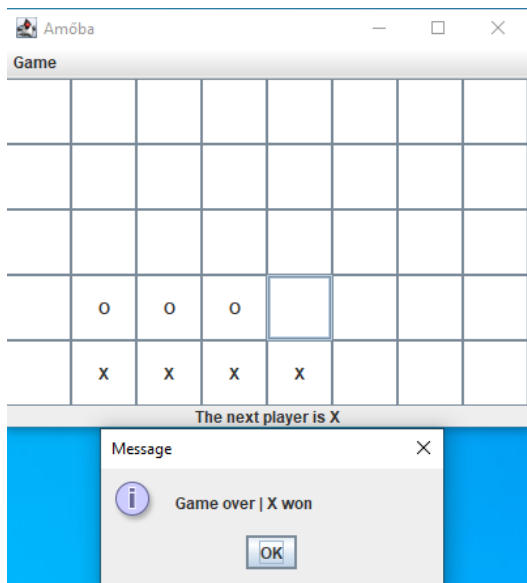
Átló balról-jobbra



Átló jobbrol-balra



Vízszintes



Nagyobb tábla

