

**Viczián Rita****2. beadandó/4. feladat**

2020.11.12.

Y2PBRT

[y2pbrt@inf.elte.hu](mailto:y2pbrt@inf.elte.hu)

7.csoport

**Feladat leírása:**

Kiszúrók amőba

Készítsünk programot, amellyel a közismert amőba játékkövetkező változatát játszhatjuk. Adott egy  $n \times n$ -es tábla, amelyen a két játékos felváltva X, illetve O jeleket helyez el. Csak olyan mezőre tehetünk jelet, amely még üres. A játék akkor ér véget, ha betelik a tábla (döntetlen), vagy valamelyik játékos kirak 5 egymással szomszédos jelet vízszintesen, függőlegesen vagy átlósan. A program minden lépésnél jelezze, hogy melyik játékos következik, és a tábla egy üres mezőjét kijelölve helyezhessük el a megfelelő jelet.

A kiszúrák a játékban az, hogy ha egy játékos eléri a 3 egymással szomszédos jelet, akkor a program automatikusan törli egy jelét egy véletlenszerűen kiválasztott pozícióról (nem biztos, hogy a hármastól), ha pedig 4 egymással szomszédos jelet ér el, akkor pedig kettőt.

A program biztosítson lehetőséget új játék kezdésére a táblaméret megadásával ( $6 \times 6$ ,  $10 \times 10$ ,  $14 \times 14$ ), 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:**

Először érdemes elkészíteni a feladat logikáát. Szükségem lesz játékosokra. Egy játékos lehet X vagy O illetve az üres mezők miatt SENKI. A Player osztály így egy enum (Player.java).

Majd elkészítem a játék modelljét (Model class). Először is szükség lesz egy táblaméretre (size), az aktuális játékosra (actualPlayer) és a játékos táblára (table). A tábla egy játékosokat tartalmazó kétdimenziós tömb. Ezekkel el lehet készíteni a játék alap logikáját.

A játékosok tudnak lépni (step()). Ez a metódus megkapja a koordinátákat, és ha nincs a játékos táblán senki, akkor az aktuális játékosra átírja a tábla megadott mezőjét. Rögtön ezután pedig az aktuális játékost az ellenkezőjére állítja, visszatér az aktuális játékoshoz.

A mozgás után mindig meg kell nézni, hogy az aktuális játékos megnyerte-e a játszmát (findWinner()). A metódus folyamatosan vizsgálja, hogy kijött-e az 5 egyforma jel egymás mellett, alatt vagy átlósan és visszatér egy játékoshoz. Abban az esetben, ha nem nyer senki SENKI-vel, ellenkező esetben pedig X-el vagy O-val.

Ezen a ponton el lehet kezdeni a grafikus felület elkészítését. Először is létre kell hozni egy alap ablakot (BaseWindow class). Ez lekezelet az alap ablakbeállításokat, amik minden továbbira érvényesek lesznek.

A BaseWindow osztályból származtatunk egy MainWindow osztályt, ami a kezdő ablakot hozza létre. Itt lehet kiválasztani, hogy milyen játékos tábla mérettel szeretnénk elkezdni játszani.

A BaseWindow osztályból származik a Window osztály is. Miután kiválasztottuk a MainWindow-n a táblaméretet. Ez alapján a Window megalkotja a megfelelő játékos táblát. Létrehoz megfelelő számú gombot, amelyekre kattintva lehet jelölni az X-eket és az O-kat. Folyamatosan mutatja a következő játékost az ablak tetején és figyeli, hogy ki nyerte meg a játékot, vagy döntetlen született-e. Játék közben is lehetőséget ad egy új kezdésre ugyan azon a táblán. Miután pedig született egy nyertes vagy döntetlen új játékot kezd.

Miután az alap mechanika működik visszatérhetünk a Model osztályhoz és megcsinálhatjuk a „kiszúrós” részt. Ehhez az osztályban létrehoztam négy listát a játékosok jelenlegi X és Y koordinátáinak. A step() metódus belerakja mindig az aktuális játékos koordinátáit. Majd a minusPieces() figyeli a legutolsó lerakott jel-et és megnézi, hogy egymás mellett, alatt vagy átlósan van-e 3 vagy 4 egyforma szimbólum. Ha talál a számnak megfelelően a changeThree() vagy changeFour() metódussal és az eltárolt koordináták alapján random kiválaszt egy elhelyezkedést és SENKI-re állítja, majd visszaadja a táblát.

A Window folyamatosan figyeli, hogy az aktuális játékosnak lett-e egymás mellett 3 vagy 4 jele és frissíti a táblát.

### **A feladat megoldásához létrehozott osztályok listája:**

1. Player –játékosok típusai, szerepelnek benne
2. Model – a játék logikája
3. BaseWindow – alap ablak beállítások, bezárás kezelése, alap méret
4. MainWindow –az első ablak megjelenése, ahol ki lehet választani a játékos táblát
5. Window az aktuális játékos tábla
6. TicTacToe – main függvény

### **Esemény-eseménykezelő párosítások és eseményeik rövid leírása**

*MainWindow osztályban:*

- getActionListener(size, ablakméret) metódus.
  - 6x6-os gombra kattintva létrehozza a játékos táblát, 400x400-as méretben.
  - 10x10-es gombra kattintva létrehozza a játékos táblát, 500x500-as méretben.
  - 14x14-es gombra kattintva létrehozza a játékos táblát, 700x700-as méretben

*Window osztályban:*

- Új játék gomhoz(NewGameButton): kattintásra új játék kezdődik, az aktuális játékos táblán.
- updateButton(x koordináta,y koordináta) metódus:
  - A játékos táblán lévő gombokat figyeli. Ha valamelyikre rákattintunk, az aktuális játékosnak megfelelő jelet ráírja a gombra. Ha három vagy négy egymással szomszédos egyforma jel lesz egymás mellett a kattintással végrehajtja a megfelelő

metódust. Minden kattintásnál vizsgálja, hogy született-e nyertes vagy döntetlen, ha igen kiírja egy felnyíló ablakban.

### **Tesztelési terv:**

- Létrehozza megfelelően az alap ablakot
- Az alap ablakban a 6x6-os gombra kattintva létrehozza a 6x6-os táblát
- Az alap ablakban a 10x10-es gomra kattintva létrehozza a 10x10-es táblát
- Az alap ablakban a 14x14-es gombra kattintva létrehozza a 14x14-es táblát
- Külön-külön a 6x6-os , 10x10-es, 14x14-es táblán:
  1. megtalálja vízszintesen minden sorban a nyertes játékost(5 egyforma jel)
  2. megtalálja függőlegesen minden oszlopban a nyertes játékos(5 egyforma jel)
  3. megtalálja átlósan mind a két irányban minden lehetséges sorban és oszlopban a nyertes játékost (5 egyforma jel)
  4. A nyertes megtalálása után a felugró ablak megmondja a nyertes játékos típusát és új játékot kezd
  5. Érzékeli a döntetlent, a felugró ablak után új játékot kezd
  6. Érzékeli minden sorban, a három egymással szomszédos jelet, kitöröl egyet a táblán lévő aktuális játékos jelei közül
  7. Érzékeli minden oszlopban a három egymással szomszédos jelet, kitöröl egyet a táblán lévő aktuális játékos jelei közül
  8. Érzékeli átlósan mindkét irányban, a három egymással szomszédos jelet, kitöröl egyet a táblán lévő aktuális jelei közül
  9. Érzékeli minden sorban, a négy egymással szomszédos jelet, kitöröl kettőt a táblán lévő aktuális játékos jelei közül
  10. Érzékeli minden oszlopban a négy egymással szomszédos jelet, kitöröl kettőt a táblán lévő aktuális játékos jelei közül
  11. Érzékeli átlósan mindkét irányban, a négy egymással szomszédos jelet, kitöröl kettőt a táblán lévő aktuális jelei közül
  12. Ezek után, ha az 5. jelet rakom be még mindig megtalálja a nyertest, és nem töröl ki jelet
  13. Megfelelően mutatja mindig az aktuális játékost
  14. A törölt elemek helyén üres lesz a gomb, újra rákattintva az aktuális játékos jele rá kerül
  15. Az ablak bezárásakor leáll a program

