

Rátkai Balázs  
ZLBM9Q  
[zlbm9q@inf.elte.hu](mailto:zlbm9q@inf.elte.hu)  
5.csoport

2.beadandó  
6.feladat  
**4-es játék**

2021.11.07.

## Feladat leírása

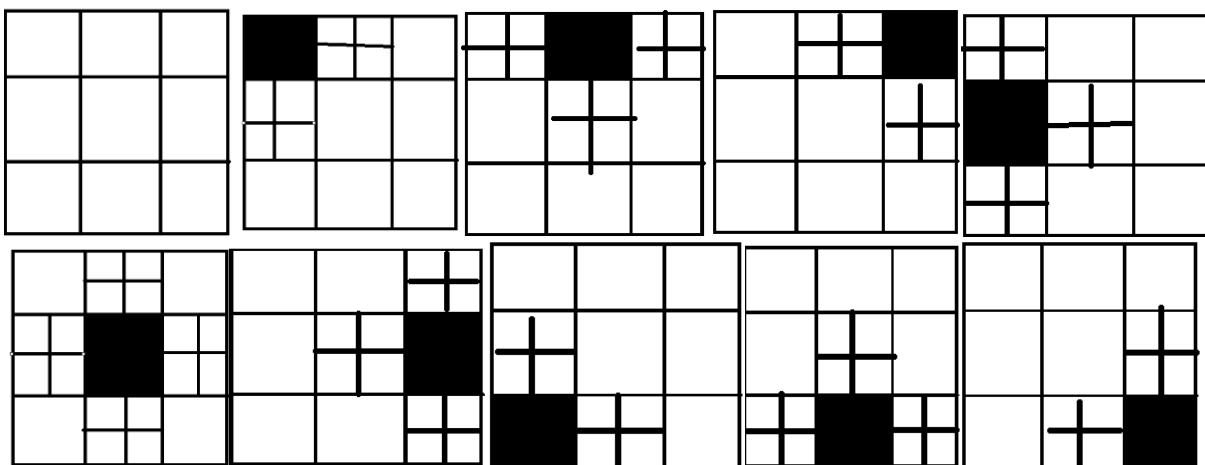
Készítsünk programot, amellyel a következő két személyes játékot lehet játszani. Adott egy  $n \times n$  mezőből álló tábla, amelynek mezői 0 és 4 közötti értékeket tartalmaznak. Kezdetben minden mezőn a 0 érték van. Ha a soron következő játékos a tábla egy tetszőleges mezőjét kiválasztja, akkor az adott mezőn és a szomszédos négy mezőn az aktuális érték eggyel nő felfelé, ha az még kisebb, mint 4. Aki a lépésével egy, vagy több mező értékét 4-re állítja, annyi pontot kap, ahány mezővel ezt megtette. A játékosok pontjait folyamatosan számoljuk, és a játékmezőn eltérő színnel jelezzük, hogy azt melyik játékos billentette 4-esre. A játék akkor ér véget, amikor minden mező értéke 4-et mutat. Az győz, akinek ekkor több pontja van.

A program biztosítson lehetőséget új játék kezdésére a táblaméret megadásával ( $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$ ), és ismerje fel, ha vége a játéknak. Ekkor jelenítse meg, melyik játékos győzött, majd automatikusan kezdjen új játékot.

## Terv

Először létrehozzuk az alap ablakot a játéknak melyen megtalálható lesz 3 gomb. Ezekkel a gombokkal lehet beállítani hogy mekkora pályán akarunk játszani(3x3,5x5,7x7). Majd létrehozzuk a pályákat.

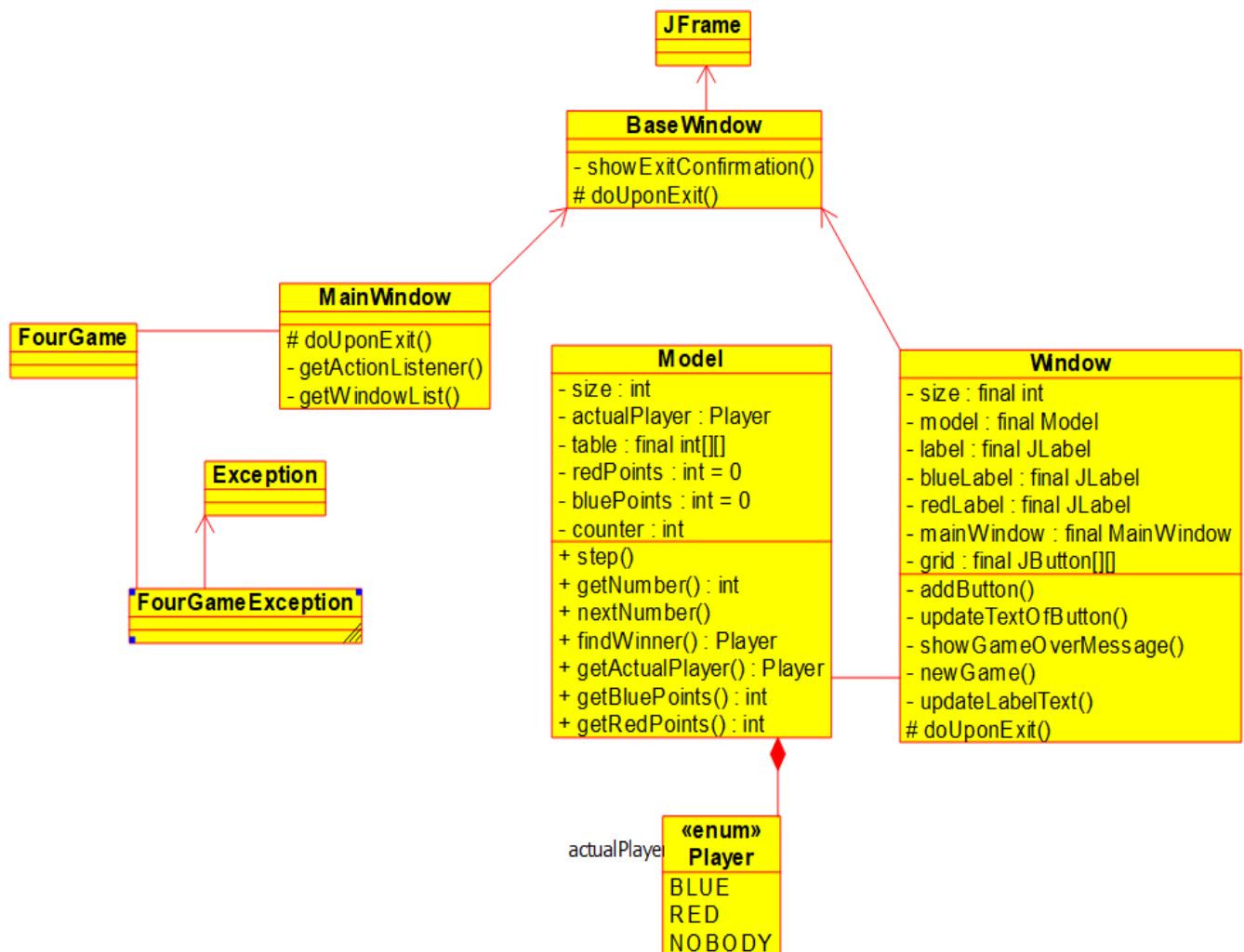
Minden pályára készítünk egy „új játék” gombot, amivel új játékot lehet indítani, egy label-t ami az aktuális játékost fogja kiírni és még két label-t ami a játékosok pontszámát fogja mutatni. Majd létrehozzuk a gombokból álló táblázatot. A gombok értékét alapból 0-ra állítjuk. Készítünk egy olyan függvényt ami a gombok értékét fogja változtatni (nextNumber) és egy olyan függvényt amely segít feltüntetni a gombok értékeit. Ezek után bővíteni lehet a nextNumber függvényt, hogy csak 4ig növelje a számokat és ha valamelyik játékos emelete a gombon lévő értéket négyre akkor az a játékos kapjon pontot. Ezt követően már csak a gomb átsínezését kell megoldanunk melyet az updateTextOfButton metódusban tudunk megtenni.



(Itt a kiszínezett négyzet jelenti azt a gombot amire rányomtunk és a + jelrel ellátott négyzetek a szomszédos gombokat jelölnek.

Az a négyzet ami ki van színezve vagy + jel van rajta annak az értékét kell majd 1-el növelni.)

# UML osztálydiagram



## Leírás

A JFrame nevű alaposztályból származik le a BaseWindow osztály. Ezzel az osztállyal fogjuk létrehozni az ablakok alap vázát, illetve hogy mi legyen az ablak neve, mit csináljon ha ki akarunk lépni a programból. Ebből két kisebb osztály származik le, a MainWindow és a Window. A MainWindow osztály fogja megvalósítani a program futtatásakor kapott ablakot. Itt lehet kiválasztani hogy mekkora pályán szeretnénk játszani. A Window osztály a kiválasztott pálya ablakát fogja megvalósítani és a felhasználó által látható elemeket például: gomb, címke. Az addButton metódus fogja létrehozni azt a gombokból álló táblázatot mellyel a játékok fogjuk majd játszani. Az updateTextofButton a gombokon lévő értékeket fogja frissíteni míg az updateLabelText a címkék értékét. Ha vége a játéknak a showGameOverMessage függvény megy végbe amely kiírja a győztes játékos nevét majd meghívja a newGame metódust. A newGame metódus új játékot indít. A Model egy publikus osztály mely tartalmazza a gombok értékeit és kiszámítja hogy mely gomboknak kell megváltoztatni az értékét hogyha rájuk nyomunk. Az utóbbit a nextNumber függvény fogja végrehajtani. Ezen felül a függvény megnézi hogyha egy gomb elérte a 4-es értéket akkor annak a játékosnak fog pontot adni aki ezt megtette. A step függvény fogja változtatgatni a játékosok sorrendjét. A findWinner metódus dönti el hogy melyik játékos lett a győztes. Ezt úgy érti el hogy a counter-t megnézi hogy egyenlő-e 0-val. Ha igen akkor megnézi hogy melyik játékosnak van több pontja és amelyiknek több azt adja vissza.