

Dokumentáció a Programozási technológia 1. tantárgy 2. beadandó feladatához

Készítette: Kósa Réka

Gyakorlatvezető: Nagy Sára

2016/17/2

A feladat szövege

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 (3×3 , 5×5 , 7×7), és ismerje fel, ha vége a játéknak. Ekkor jelenítse meg, melyik játékos győzött, majd automatikusan kezdjen új játékot.

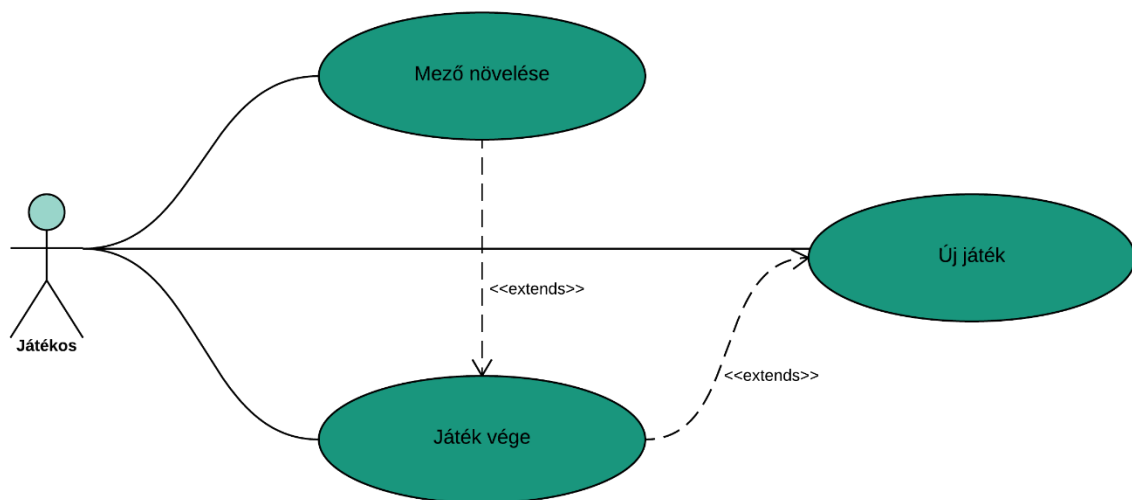
Elemzés

- A szoftvernek Windows operációs rendszeren kell futnia, ablakos felhasználói interfésszel kell rendelkeznie
- A program egy többszemélyes játék, azaz két játékosnak kell felváltva kattintani valamely mezőre egy fix méretű táblán.
- Egy mezőnek értéke egy 0 és 4 közé eső szám lehet, valamint a színe is ezektől az értékektől illetve az aktuális játékosától függ
- Lehetővé kell tenni azt, hogy új játék induljon, akár menet közben is, a pályaméret változhat megadott keretek között a felhasználó döntésétől függően
- A játék folyamán fontos szerepe van annak, hogy mi történik, ha egyik játékos egy adott mezőre lép:
 - megtudjuk, hogy kap-e pontot a játékos (azaz ha az adott mező elérte a maximumát, 4-et)
 - ahhoz, hogy eldönthessük, véget ér-e a játék, szükségünk van arra az információra, hogy az összes mező elérte-e a maximumát. A győztes kiválasztásához még a pontok összehasonlítására van szükség
- Nem igényel adatbázis-kapcsolatot, nincs különösebb idő- és tárigénye

Felhasználói esetek

A játékosok felváltva kattinthatnak, minden egyes kattintás után kiértékelődik annak következménye (nyert-e valamelyik játékos). Ez járhat a játék végével, ami pedig folytatódhat új játék kezdésével. Új játékot menet közben is lehet kezdeni.

A lehetőségeket az alábbi diagramon tudjuk áttekinteni:



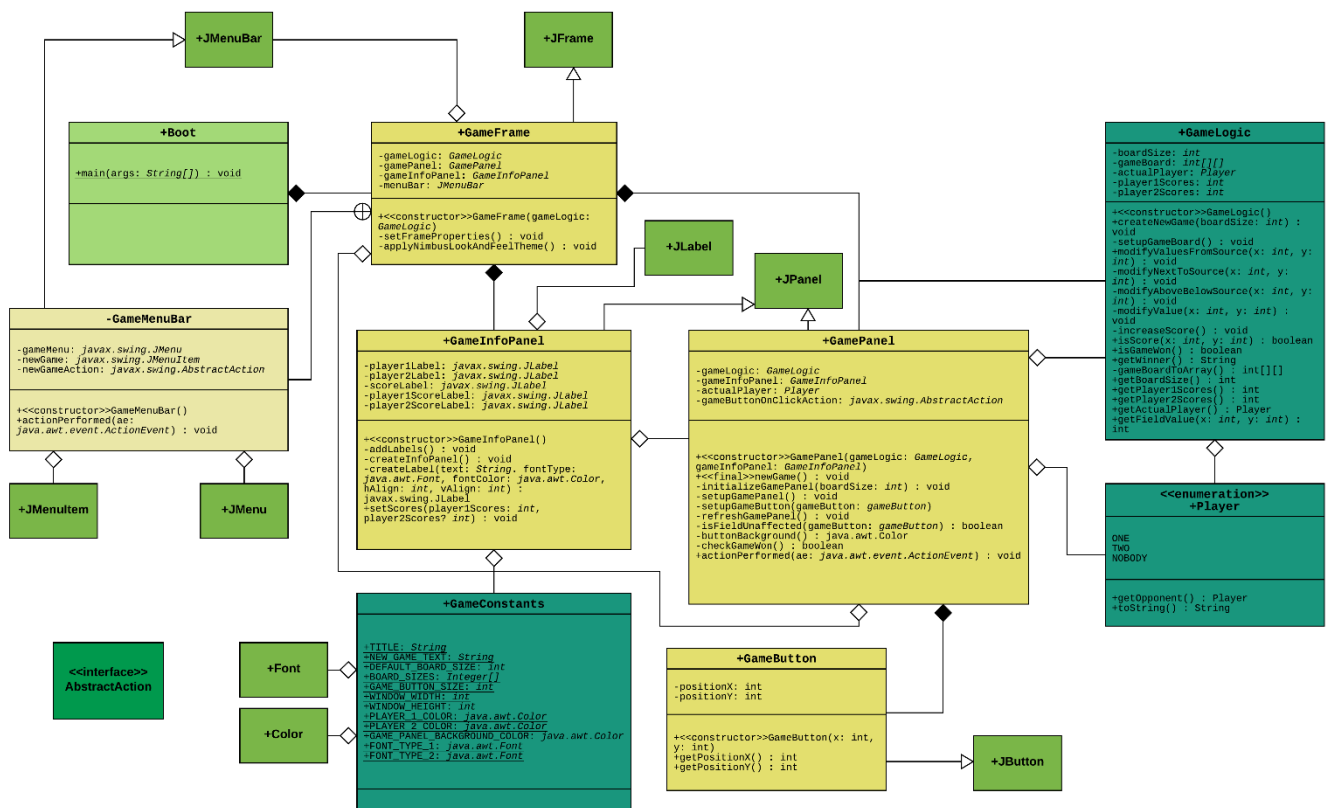
1. ábra: Use Case diagram

Tervezés

Osztályszerkezet

- A Model-View architektúrának megfelelően a projekt osztályai két diszjunkt halmazra bomlanak, az egyiket a `logic`, a másikat a `view` csomag jelöli
- A belépési pont egy `Boot` osztályban van, amely nincs benne sem a `logic`, sem a `view` csomagban. Ez felel a két komponens létrehozásáért, paraméterezéséért és a játék indításáért
- A `view` fő osztálya a `GameFrame`, ez kezdetben kap egy `GameLogic` objektumot paraméterként, ezt továbbítja a `GamePanel`-nek, valamint ez lesz a megjelenítendő ablak, ezért ez egy `JFrame` leszármazott. Két nagy részből áll: egy felső információkat megjelenítendő részből (`GameInfoPanel`) és a központi rácsból, amin a játéktábla van (`GamePanel`), amely gombokból áll (`GameButton`). Tartalmaz még ezenfelül egy menüt is, amin keresztül új játékot indíthatunk. A menü egy belső osztály, a `GameInfoPanel` és a `GamePanel` viszont külön osztályok. Az első kettő példányosítással jön létre, a `GamePanel` pedig a korábban átvett `GameLogic`-ot és a most létrehozott `GamePanel`-t viszi tovább. Az események azonban a panel felé érkeznek, ez kommunikál a `GameLogic`-kal, valamint a `GameMenuBar` is ennek az osztálynak a `newGame()` metódusát használja.
- A `model` fő osztálya a `GameEngine`. Ez tartalmazza az adatok reprezentációját, illetve fogadja a felületről érkező hatásokat és visszaadja azok hatásait. A csomagban található még egy `GameConstants` nevű osztály, ahol a programban használt fontosabb konstansok vannak gyűjtve, tehát különböző méretek, színek, egyéb attribútumok, és van egy `Player` enum is, amelynek a soron következő játékos meghatározásában van szerepe: kezdetben ez `NOBODY`, majd a `createNewGame()` metódus meghívása után `ONE`, aztán `TWO`, és mindig felváltva. A váltogatást a `getOpponent()` függvény végzi.

Az alábbi ábrán az egyes osztályok kapcsolatait láthatjuk.



2. ábra: Osztálydiagram

Esemény-tevékenység párok

ESEMÉNY: MEZŐ ÉRTÉKÉNEK NÖVELÉSE

Kiváltja: kattintás az adott mezőre

Tevékenység: mezőn látható szám megnövelése eggyel

Ezt megvalósító függvények, osztályok: a `GamePanel` észleli a kattintást, ennek hatására a `GameLogic` `modifyValueFromSource()` metódusa meghívódik, és növeli az értéket, amíg az nem lesz 4. Ha eléri a 4-et, a kattintás nem vált ki további tevékenységet. A `GamePanel` ezek után frissül, és az új értéket jeleníti meg.

ESEMÉNY: **MEZŐ ÁTSZÍNEZÉSE**

Kiváltja: kattintás az adott mezőre + kattintás által a mező vagy az alatta-fölötte és mellette található mezők valamelyikének értéke 4 lesz

Tevékenység: mező színének megváltoztatása, aktuális játékos pontszámának növelése

Ezt megvalósító függvények, osztályok: a `GamePanel` észleli a kattintást, ennek hatására a `GameLogic` `modifyValueFromSource()` metódusa meghívódik, és növeli az értéket. Az `isScore()` metódus vizsgálja, hogy elérte-e a 4-et az aktuális mező vagy a felette, alatta, mellette lévő mezők egyike. Ha igen, a `GameLogic`-ban és `GamePanel`-ben is megtalálható `actualPlayer` adattag alapján eldől, hogy milyen színűek legyenek a mezők, ha azok még nem kaptak színt. A megfelelő játékos pontszáma annnyival növekszik, ahány mezőt billentett négyesre.

ESEMÉNY: **ÚJ JÁTÉK INDÍTÁSA**

Kiváltja: a megfelelő menüpont kiválasztása vagy a játék vége

Tevékenység: megjelenik egy felugró ablak egy legördülő menüvel, ahol ki lehet választani a táblaméretet. A következő játék ennek megfelelő dimenziókkal indul, ha kiválasztás után az OK-ra kattintottunk.

Ezt megvalósító függvények, osztályok: a `GamePanel` észleli, ha vége van a játéknak, mégpedig a `GameLogic` `isGameWon()` metódusa keresztül, ami azt vizsgálja, hogy átbillent-e minden mező értéke 4-re. Ha igen, akkor megjelenik egy felugró ablak, majd ha választottunk táblaméretet, ez az adat átadásra kerül a `GameLogic`-nak, és ilyen méretűre inicializálja a következő menetben a táblát, a `GamePanel` pedig új játékot kezdeményez. Ha játék közben a „New game” opciót választjuk, akkor is a `GamePanel` `newGame()` metódusa hívódik meg.