MocsiSweeper

I. Feladat ismertetése

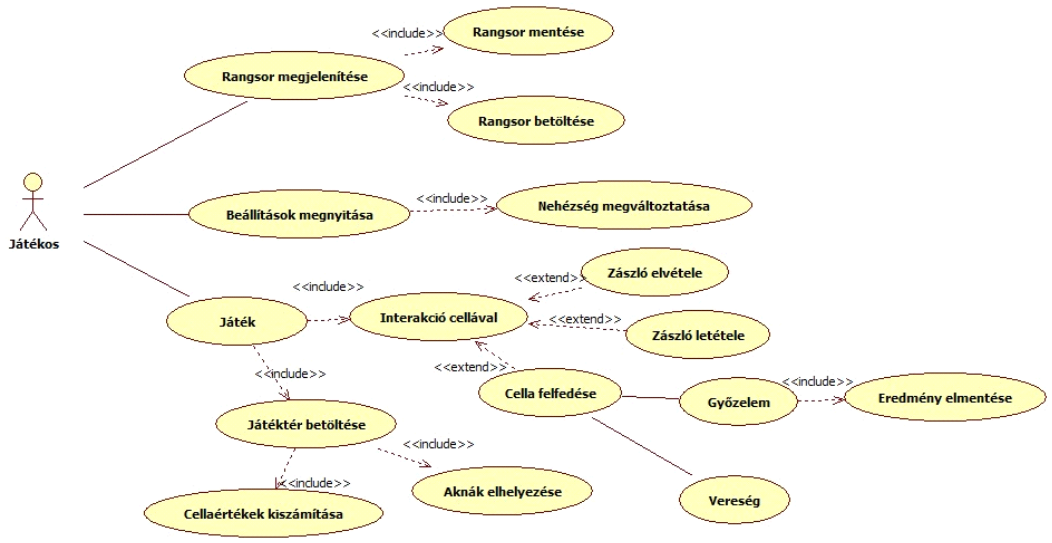
A MocsiSweeper egy játék, mely során a játékos feladata, hogy felfedje az összes biztonságos cellát a játéktéren, miközben elkerüli a veszélyes cellákat, melyek alatt aknák találhatók.

Bal egérkattintással tud a játékos felfedni egy cellát, melyben, ha nem akna található, akkor egy szám jelzi, hogy hány szomszédos cella veszélyes. Abban az esetben, ha egyik szomszédos cella sem tartalmaz aknát, akkor nincs benne sem szám, se akna. Természetesen, ha aknára lép a játékos, akkor veszít, a játék véget ér, és visszakrül a főmenübe.

A játék háromféle nehézségi szintet különböztet meg, melyek a pálya méretét és az áknák számát befolyásolják. Ezek közül a beállítások menüben lehet választani.

A játékok pontozását idő alapján teszi a rendszer, minél rövidebb ideig tartott a győzelmet elérni, annál jobb. Erről a rendszer egy ranglistát tart, melyet az adott nehézségi szintre ki lehet kérni.

II. Use case-k



III. Megoldási ötlet

A grafikus felhasználói felületet **JavaFX**-ben készítem el. 4 különböző ablakot tartalmaz, egy főmenü-t, egy beállítások ablakot, egy rangsor ablakot és egy játékteret.

A főmenü és a beállítások menü kinézetét egy statikus **fxml** file tárolja, melyből azok könnyen beölthetők, megjeleníthetők. A főmenüben **Button**-k találhatók, melyek segítségével a felhasználó a többi ablakra navigálhat. A Beállítások menüben egy **combobox** segítségével választhat a felhasználó nehézségi szintet, majd a 2 **Button** segítségével visszaléphet a főmenübe vagy elmentve, vagy elvetve a beállítások változtatásait.

A játéktér vázát szintén egy **fxml** file tárolja, melyben egy **BorderPane** helyezkedik el. Ennek jobb oldalán egy másik **fxml** file-ból beöltött **StackPane** látható, mely a jelenlegi állást (hátralévő aknák - letehető zászlók - számát, valamint az eltelt időt mutatja **Label**-k segítségével). A **BorderPane** maradék részében maga a játéktér látható, mely egy **board** típusú objektum, mely a **GridPane** osztályt extend-eli, mely több **Button**-nel van feltöltve. Ezeket futási időben készíti el a program, és feltölti értékekkel (0-8 attól függően, hogy hány akna van körülötte, vagy 9, ha akna található az adott cellában).

A gombkezelésre egy saját osztályt, a **MyButton**-osztályt használom, mely az eredeti **Button** osztályt egészíti ki a megfelelő metódusokkal és field-ekkel.

Az **fxml** file-oknak közös **Controller** osztályuk van, mely az **fxml** metódusokat tartalmazza.

A **HighScore** osztály tartalmazza azoknak a játékosoknak a neveit és idejét, akik megnyerték a játékot. Ezt egy **ArrayList**-ekkel oldja meg, mely **Score** típusú objektumokat tárol. A **Score** egy osztály, mely egy **String**-et és egy **int** értéket párosít (a játékos neve és ideje).

A játék idejét a **Timer** osztály kezeli, mely periodikusan update-l egy **Label**-t a kijelzőn, mutatva, hogy mennyi időnél jár éppen a játékos.

A fontosabb, gyakran előkerülő értékeket egy **Consts** osztály tárolja.