3. beadandó feladat dokumentáció

# Készítette:

Mikus Márk (CM6TSV)

E-mail: [kyussfia@gmail.com](mailto:kyussfia@gmail.com)

# Feladat:

Kiszúrós amőba Készítsünk programot, amellyel a közismert amőba játék következő változatát játszhatjuk. Adott egy -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 kiszúrás 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 hármasból), ha 4 egymással szomszédos jelet ér el, akkor pedig kettőt. A program biztosítson lehetőséget új játék kezdésére a táblaméret megadásával ( 6x6, 10x10, 14x14), valamint az aktuális játék mentésére és egy korábban elmentett játék betöltésére. Ismerje fel, ha vége a játéknak, és jelenítse meg, melyik játékos győzött, illetve azt is, ha döntetlen lett a vége, majd automatikusan kezdjen új játékot.

# Elemzés:

Egy olyan háromrétegű alkalmazás megvalósítása a cél, amelyben a nézet tartalmazza a menüfunkciókat kiváltó gombokat (Új játék, mentés, töltés, kilépés) az aktuális játékost megjelenítő feliratból, és magából a játéktérből áll. (Gombok, amelyek először üresek, s ha az egyik játékos egy jelet rak oda, akkor az a gomb a játékos szimbólumát veszi fel, továbbá az aktuális játékost megjelenítő címke-kijelzőn megjelenik a soron következő játékos).

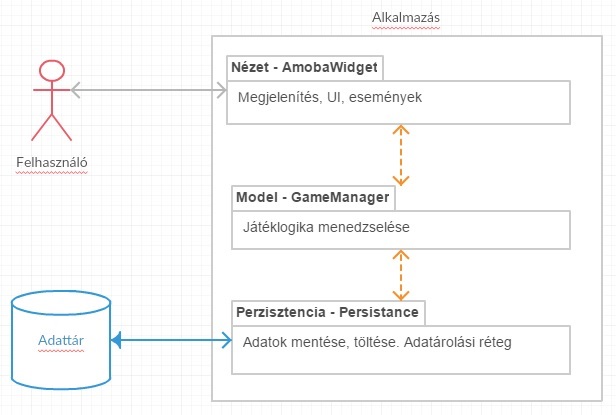
A játéklogikába, a szokásos szabályok mellett (Új játék funkció, Kilépés funkció) implementálása kerülnek, a kiszúrást segítő szabályok is.

A játéklogika alatti rétegbe a perzisztenciába, a mentést illetve a betöltést kell implementálni.

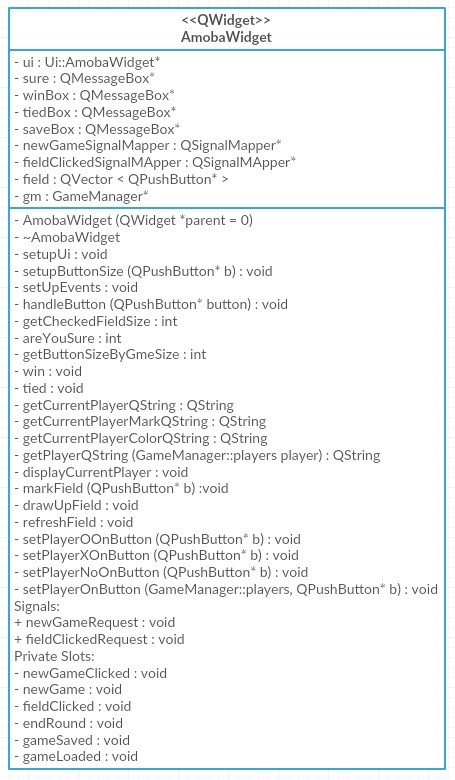
A perzisztencia a Modelbe injektált osztály (Persistance). A model (GameManager) eseményekkel kommunikál a nézettel (AmobaWidget). Az egyes eseményleírókat később taglaljuk.

# Használati eset:

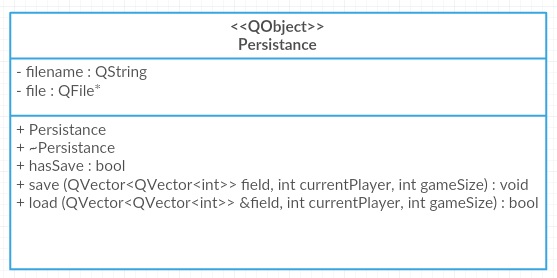
# Architektúra:

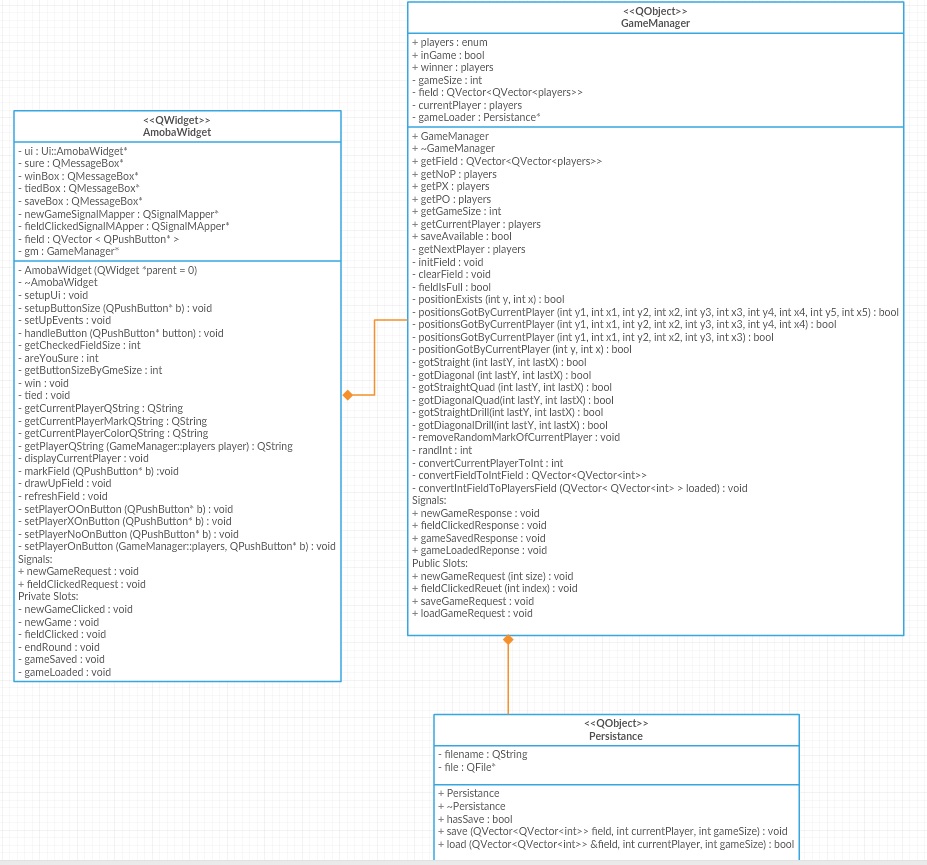


# Osztályok:









|  |  |  |  |
| --- | --- | --- | --- |
|  | Felhasználói eset | Leírás | |
| 1 | Alkalmazás indítása | Given | Alkalmazás futtatható állománya a célszámítógépen található. |
| When | Alkalmazás indítása |
| Then | Megjelenik a ui, a menü és az üres játéktér. |
| 2 | Alkalmazás bezárása | Given | Az alkalmazás fut. |
| When | Alkalmazás bezárása |
| Then | Az alkalmazás leáll, az ablakai bezárulnak. |
| 3 | Kilépés | Given | Az alkalmazás fut. |
| When | Kilépés az alkalmazásból. |
| Then | Az alkalmazás leáll. |
| 4 | Új játék | Given | Az alkalmazás elindítottuk és fut. |
| When | Új játék gombra kattintva |
| Then | új játék indul, adott méretben, frissül a tábla. |
| 5 | Jelöl | Given | Játék közben, azaz ha új játék közbe vagyunk |
| When | Adott játékmezőbeli gombra kattintva. |
| Then | Megjelölődik az adott játékos jelével a mező, ill működébe lépnek a szabályok. |
| 6 | Mentés | Given | Ha játék közben vagyunk. |
| When | Mentés gombra kattintva, |
| Then | Lementősik az aktuális játékállás, a játékhoz tartozó mentésfájlba. |
| 7 | Betöltés | Given | Ha van korábban mentett állás, az adatbázisban (file) |
| When | Játék töltése gombra kattintva. |
| Then | Új játéktábla töltődik, be s így a mentett állás jelenítjük meg. |

# Eseményvezérlés:

A játékfelületen lévő clicked események triviálisan az adott gombhoz tartoznak, az egyeskomponensek-beli események a következők:

* AmobaWidget
  + Signal:
    - newGameRequest : Új játék indításakor kiváltott esemény, amellye a GameManagernek üzen a nézet.
    - fieldClickedRequest: Játék közben, amikor egy mezőre kattint egy játékos ez az esemény váltódik ki.
  + Slot:
    - newGameClicked: Új játék gomb megnyomása fut be ide.
    - newGame: A GameManager newGameResponse jele váltja ki.
    - fieldClicked: A mezőgombok megnyomása fut be ide.
    - endRound: AGameManager fieldClickedResponse jele váltja ki.
    - gameSaved: A GameManager gameSavedResponse jele váltja ki.
    - gameLoaded: A GameManager gameLoadResponse jele váltja ki.
* GameManager
  + Signal:
    - newGameResponse: Új játék kezdése, után váltódik ki.
    - fieldClickedResponse: Egy játékos lépésének kiértékelése után váltódik ki.
    - gameSavedResponse: Játék mentésekor választ küldünk a nézetnek.
    - gameLoadedReponse: Játék töltésekor választ küldünk a nézetnek.
  + Slot:
    - newGameRequest(int size) : Új játék kezdésének logikája, majd newGameResponse.
    - fieldClickedRequest(int index) : Egy játékos egy lépésének hatása, logikája, majd fieldClickedResponse.
    - saveGameRequest : Játék mentésének kérelmezése, majd gameSavedResponse.
    - loadGameRequest : Játék töltésének kérelmezése, majd gameLoadedResponse.

# Tesztesetek:

* Új játék kezdése első szinten
* Új játék kezdése második szinten
* Új játék kezdése harmadik szinten
* Mező megjelölése
* Szabályok ellenőrzése
* Jelölés-Eltűnés vizsgálata
* 3-as Mezők jelölése
* 4-es Mezők jelölése
* 5-ös Mezők jelölése
* Dupla Új játék kezdése