

# Tartalomjegyzék

0.1. Készítette . . . . .	1
<b>1. Feladat elemzése</b>	<b>1</b>
1.1. Feladat . . . . .	1
1.2. Elemzés . . . . .	1
<b>2. Felhasználói esetek</b>	<b>2</b>
2.1. Eset táblázat . . . . .	2
2.2. Felületi terv . . . . .	4
<b>3. Logikai terv</b>	<b>5</b>
3.1. Rétegek . . . . .	5
3.1.1. Perzisztencia réteg . . . . .	5
3.1.2. Modell réteg . . . . .	6
3.1.3. Nézet réteg . . . . .	7
3.1.4. Modell réteg teljes osztályhierarchiája . . . . .	8
3.1.5. Eseménykezelés . . . . .	8

## 0.1. Készítette

Pilipár Botond

Email-cím: pilipar.botond@gmail.com

Neptun-kód: KD417D

Csoportszám: 12

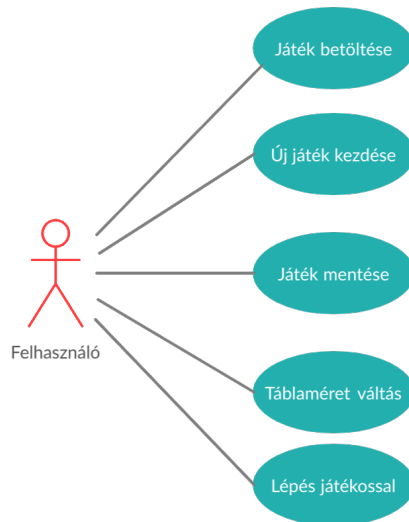
## 1. Feladat elemzése

### 1.1. Feladat

Készítsünk programot, amellyel a következő két személyes játékot lehet játszani. Adott egy  $n \times n$  mezőből álló tábla, ahol egy menekülő és egy támadó játékos helyezkedik el. Kezdetben a menekülő játékos figurája középen van, míg a támadó figurái a négy sarokban helyezkednek el. A játékosok felváltva lépnek. A figurák vízszintesen, illetve függőlegesen mozoghatnak 1-1 mezőt, de egymásra nem léphetnek. A támadó játékos célja, hogy adott lépésszámon (4) belül bekerítse a menekülő figurát, azaz a menekülő ne tudjon lépni. A program biztosítson lehetőséget új játék kezdésére a táblaméret (3x3, 5x5, 7x7) és így a lépésszám (12, 20, 28) megadásával, folyamatosan jelenítse meg a lépések számát, 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.

### 1.2. Elemzés

A játékot két személy játssza, azonban a játék szempontjából csak váltakozó lépések vannak. Minden játék lépését a vadász fél kezdi, ezzel is növelve a bekerítés esélyét a nagyobb táblákon. Esetdiagramm nézőpontjából az egyféle játékos az alábbi cselekedeteket végezhet:



## 2. Felhasználói esetek

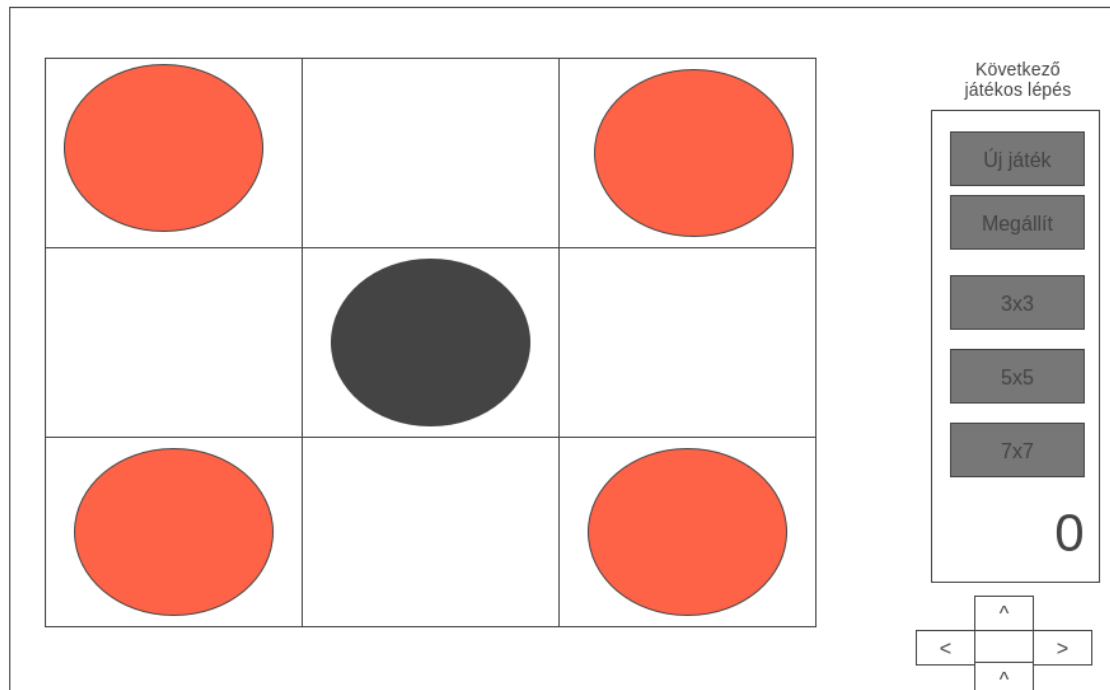
### 2.1. Esettáblázat

1.	Alkalmazás elindítása	GIVEN:	Az alkalmazás feltelepült
		WHEN:	Futtatható alkalmazás elindítása
		THEN:	Megjelenik a játéktábla, az alapértelmezett 5x5 méretben és alapvető játékhelyzetben.
2.	Kiválasztás	GIVEN:	A játék aktív állapotban van, megjelenik a tábla és a rajta található 5 játékos
		WHEN:	A felhasználó a tábla valamelyik mezőjére kattint, kiválasztva ezzel a rajta várakozó játékost valamelyikét, vagy egy üres mezőt.
		THEN:	A tábla jobb alsó sarkában található 4 gomb aktiválódik, jelezve ezzel a lépés lehetőségét
3.	Lépés	GIVEN:	A játékos kiválasztotta a tábla egy mezőjét jelezve, hogy az ott található (vagy nem található) játékosal szeretne lépni
		WHEN:	A felhasználó egy újabb kattintással kiválasztotta a célmezőt vagy a "W A S D" gombok lenyomásának valamelyikével, vagy a játéktáblán található aktiválódott gombok segítségével egy adott irányú lépést tett olyan irányba, ami nem engedélyezett, mert a mező már foglalt, vagy mert a játéktáblán kívül esik.
		THEN:	A játéktáblán nem történik semmi, az irányító gombok inaktiválódnak
4.	Lépés	GIVEN:	A játékos kiválasztotta a tábla egy mezőjét jelezve, hogy az ott található (vagy nem található) játékosal szeretne lépni

		WHEN:	A felhasználó egy újabb kattintással kiválasztotta a célmezőt vagy a "W A S D" gombok lenyomásának valamelyikével, vagy a játéktáblán található aktiválódott gombok segítségével egy adott irányú lépést tett olyan irányba, amely engedélyezett (nem foglalt és a játéktáblán helyezkedik el)
		THEN:	A táblán lévő a játékos az először kiválasztott helyről a másodszorra kiválasztott helyre lép. Abban az esetben, ha a lépéssel bekerítésre került sor, megjelenik a "Játék vége" felirat kijelevze, hogy melyik játékos hány lépésből nyert.
5.	Lépés	GIVEN:	A játékos kiválasztotta a tábla egy mezőjét jelezve, hogy az ott található (vagy nem található) játékosal szeretne lépni
		WHEN:	A felhasználó egy újabb kattintással kiválasztotta a célmezőt vagy a "W A S D" gombok lenyomásának valamelyikével, vagy a játéktáblán található aktiválódott gombok segítségével egy adott irányú lépést tett olyan irányba, amely engedélyezett (nem foglalt és a játéktáblán helyezkedik el), de olyan játékosal aki épp nem következik.
		THEN:	A játéktáblán nem történik semmi, az irányító gombok inaktíválódnak
6.	Játék mentése	GIVEN:	A játéktábla aktív állapotban van
		WHEN:	Felhasználó a bal felső sarokban található Fájl -> Mentés gombra kattint
		THEN:	Megjelenik a fájlválasztó ablak, lehetőséget adva a felhasználónak a játékállapot elmentésére.
7.	Játék mentése	GIVEN:	A játéktábla aktív állapotban van, továbbá a felhasználó a bal felső sarokban található Fájl -> Mentés gombra kattintott
		WHEN:	A felhasználó bezárja az ablakot fájl kiválasztása nélkül.
		THEN:	A játék ekkor is mentésre kerül, egy automatikusan generált fájlnev szerint.
7.	Játék mentése	GIVEN:	A játéktábla aktív állapotban van, továbbá a felhasználó a bal felső sarokban található Fájl -> Mentés gombra kattintott
		WHEN:	A felhasználó kiválaszt egy fájlt.
		THEN:	A játék ekkor mentésre kerül, a kiválasztott fájlnev szerint.

## 2.2. Felületi terv

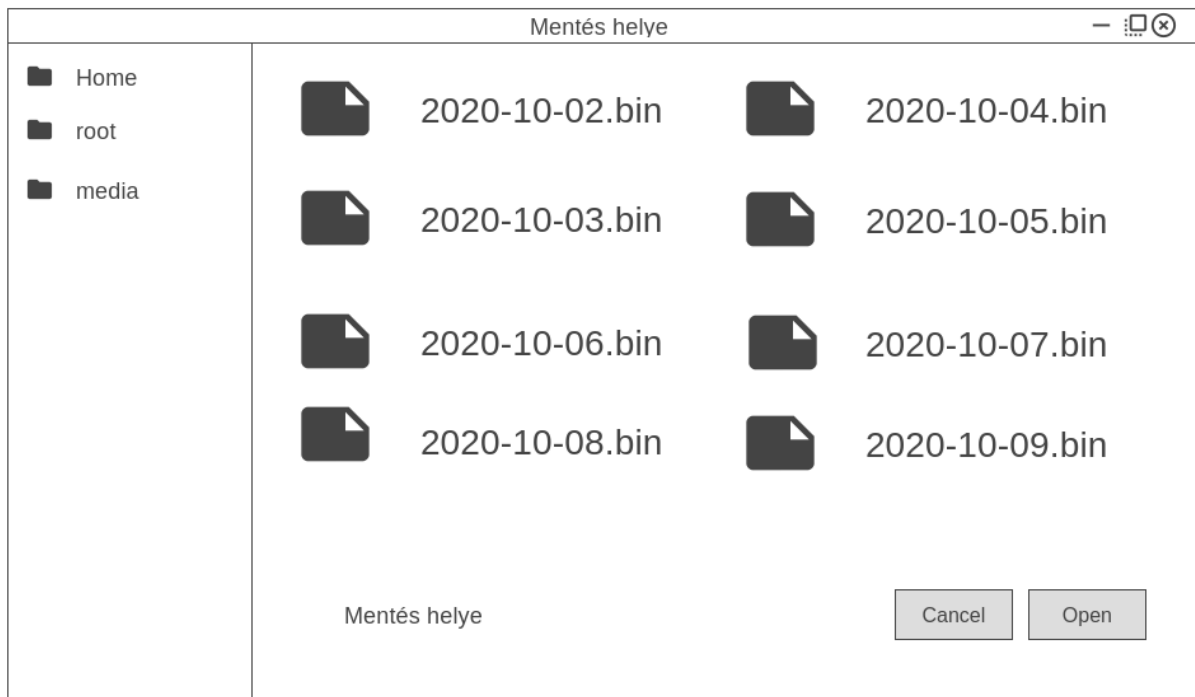
A játék egy kirajzolt  $n \times n$  táblázat, melyben különböző színű körökkel jelöljük a különböző játékosokat



Ha valamelyik fél megnyerte a játékot, arról egy információs ablak tájékoztatja, melyen szerepel a győztes fél neve és lépésszám, ami a kezdés óta eltelt.



A játék bármely pillanatában (kivéve ha az információs doboz aktív) van lehetőség a játék mentésére. A Fájl -> Mentés vagy Fájl -> Betöltés gombra kattintva megjelenik a mentés ablak, ahol a felhasználó kiválaszthatja a fájl nevét, illetve akár törölhet is korábbi fájlokat, ha úgy gondolja, már túl sok mentés van az adott mappában.

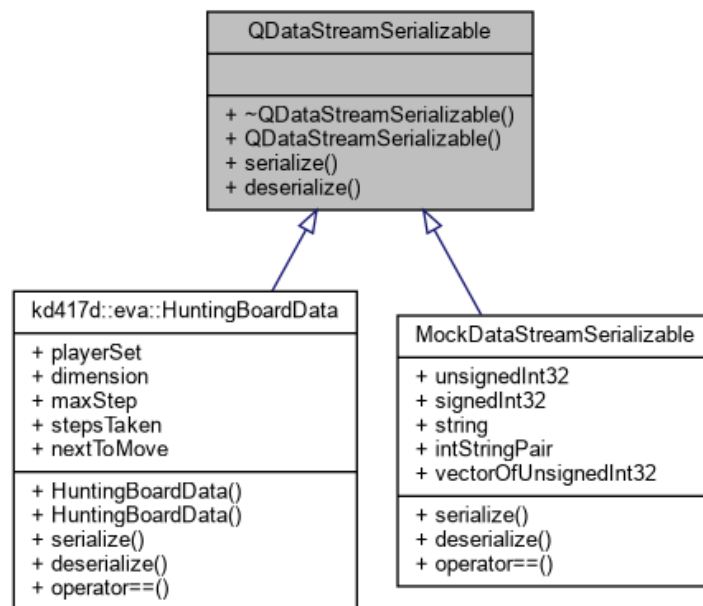


### 3. Logikai terv

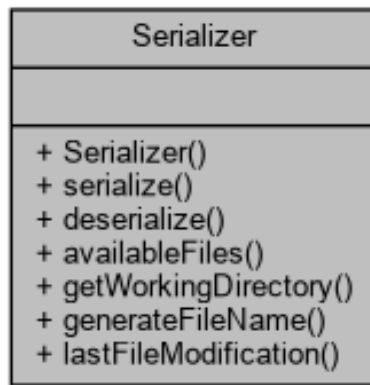
#### 3.1. Rétegek

##### 3.1.1. Perzisztencia réteg

Legelső réteg a perzisztencia, ahol a fájlok kezelése, mentése és betöltése történik. Mivel a mentés osztály-hierarchiába van szervezve, egy osztálynak elég megvalósítania a *QDataStreamSerializable* interfészt ahhoz, hogy a perzisztencia osztály kezelni tudja.

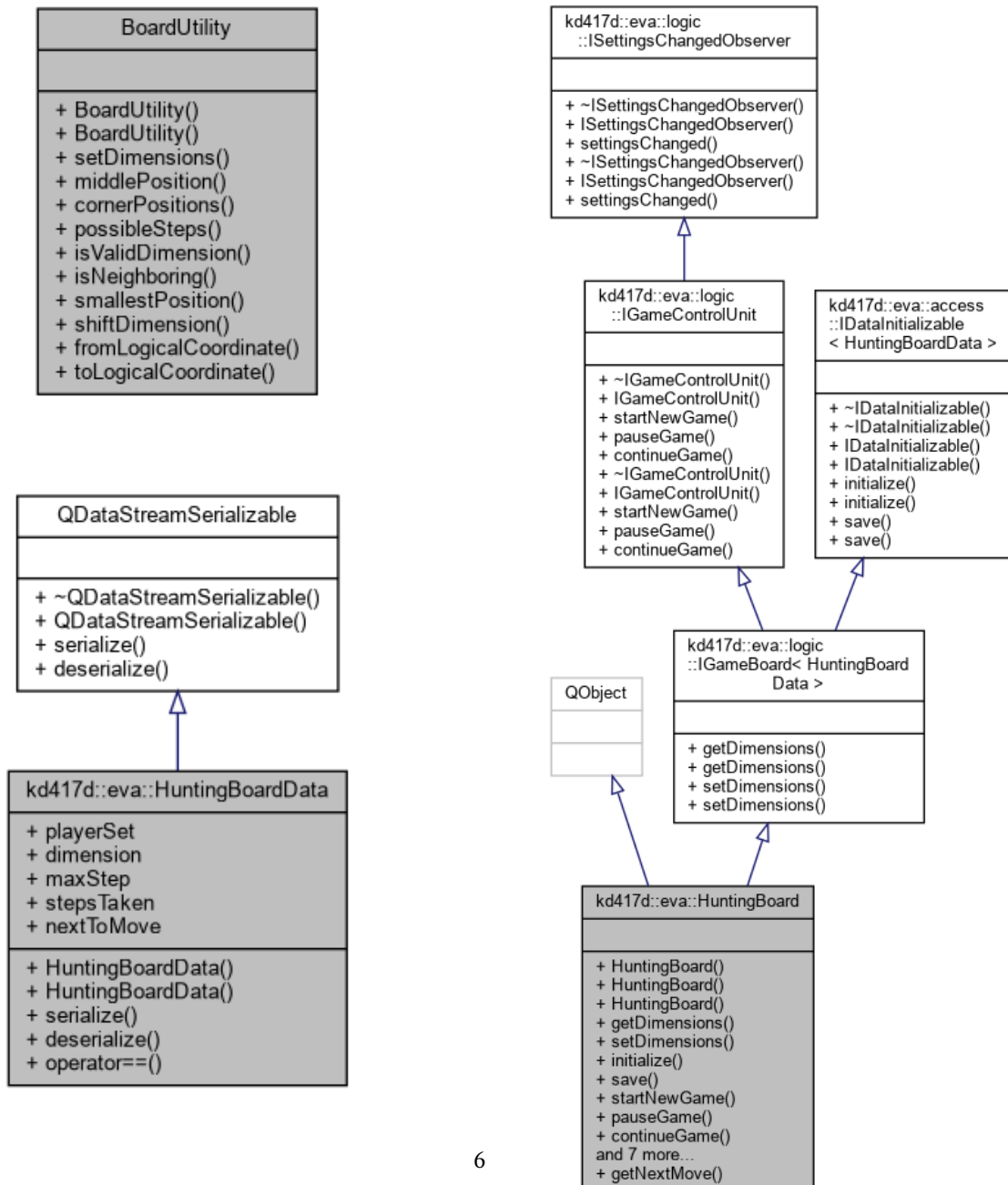


A *Serializer* osztálynak már csak a fájlkezelést kell végeznie, mivel a szerializációs logikát egy osztály korábban megvalósította.



### 3.1.2. Modell réteg

A modell réteg egy metódusban elfedve kezeli a játék logikai részét. Ehhez több "Utility" és egy főosztályba vannak szervezve a releváns metódusok. A modell osztály logikai koordinátákkal dolgozik, melyeket 1-től indexelünk. Minden adat, ami a modellben található, menthető egy adatosztályba, ami aztán szerializálható a perzisztencia rétegben.



## Metódusok

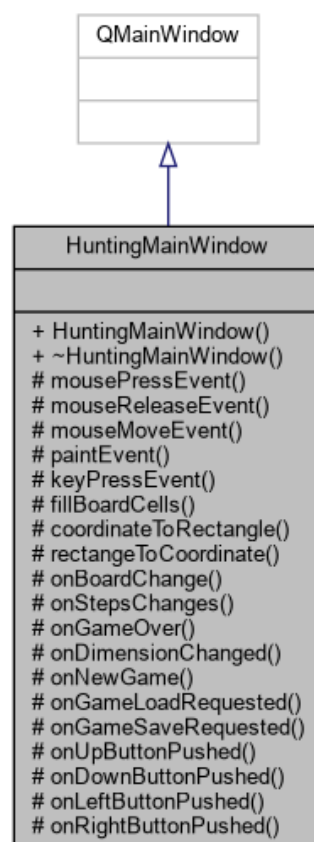
A *movePlayer* metódus segítségével kommunikál a nézet réteg a modell réteggel. Ha egy játékos mozgatása sikeres, arról esemény ad tájékoztatást.

## Adattagok

A *mPlayerMap* adattagban vannak a játékosok, akik helyzete változhat, játékos típusuk azonban soha.

### 3.1.3. Nézet réteg

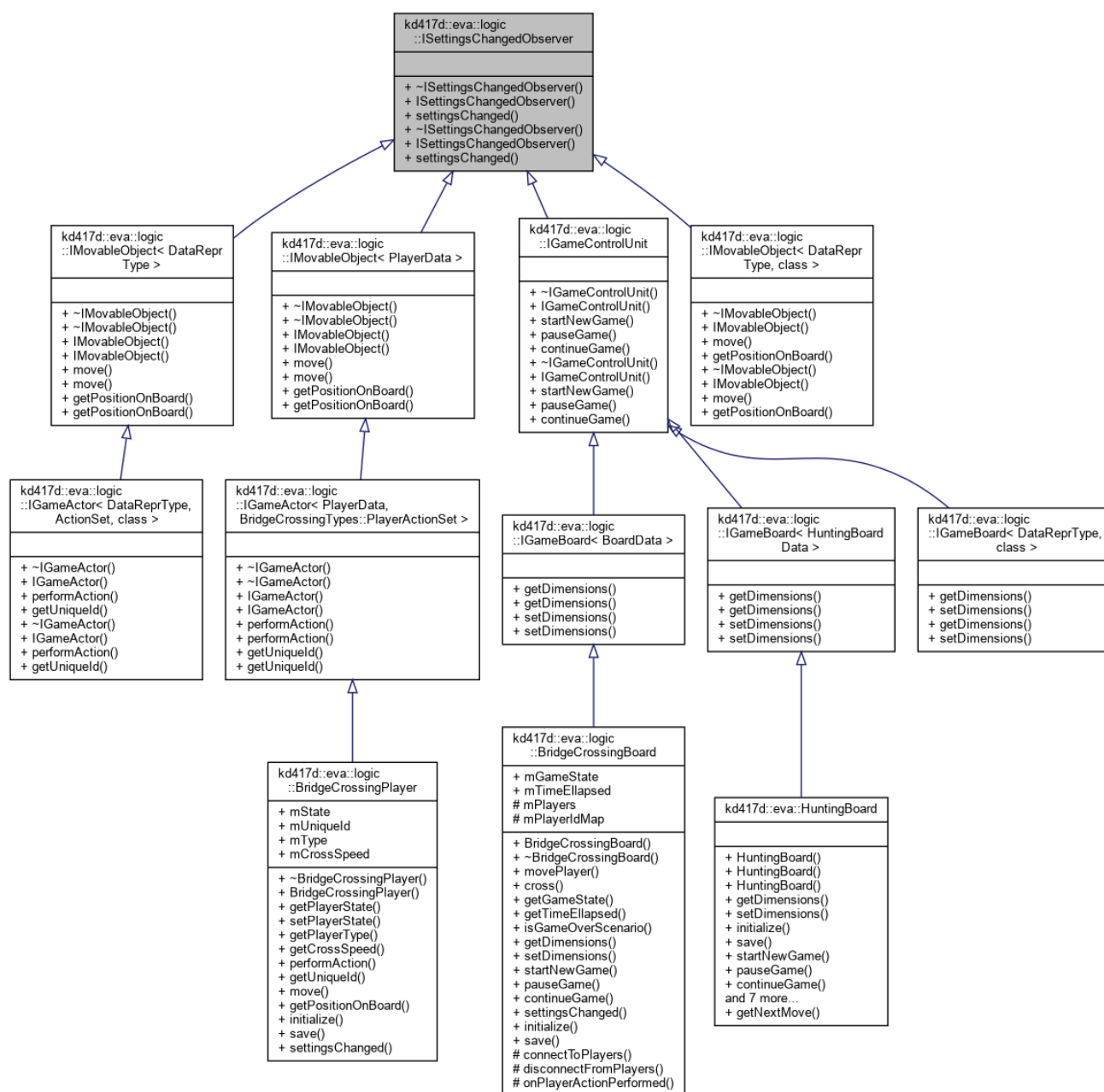
A nézet réteg felelős a tábla eseményeinek megjelenítéséért. A tábla logikáját kirajzolva, egy táblázatként jeleníti meg, ami minden lépésnél frissülés hatására újra kirajzolódik. Ez a réteg felelős továbbá a felhasználó által mozgatni kívánt bábuk regisztrálásáért és továbbításáért a modell réteg felé. Ezt az egér követésével valósítja meg.



## Adattagok

A metódus segítségével kommunikál a nézet réteg a modell réteggel. Ha egy játékos mozgatása sikeres, arról esemény ad tájékoztatást.

### 3.1.4. Modell réteg teljes osztályhierarchiája



### 3.1.5. Eseménykezelés

Eseménykezelés megvalósul a nézet-tábla között. A nézet által kezelt nyomógombok a tábla bizonyos metódushívásait idézik elő. Ezeket a hívásokat a tábla továbbítja a táblának, majd eseményben tájékoztatást kap annak sikerességéről. Természetesen az is lehet, hogy a tábla által észlelt lépés engedélyezett, de a jelenlegi játékkállapot ezt nem teszi lehetővé (például az adott pozíció már foglalt).



sender	signal	reciever	slot
HuntingBoard	newGameSignal	HuntingMainWindow	onNewGameStarted
HuntingBoard	boardChangedSignal	HuntingMainWindow	onBoardChanged
HuntingBoard	scoredPointChangedSignal	HuntingMainWindow	onScoredPointChanged
HuntingBoard	gameOverSignal	HuntingMainWindow	onGameOver