

Programozási technológia

2. Beadandó

Név: Magyar Viktor

Neptun kód: O9WEJJ

Dátum: 2024.04.01.

Feladat száma: 1

1. Feladat leírása:.....	3
2. Megoldási terv:.....	4
2.1 Megoldás leírása.....	4
2.2 Model komponens.....	4
2.2.1 tictactoe.table package:	4
2.2.2 tictactoe.gamecontrol package:	5
2.2.3 tictactoe.gamemenu package:	6
2.2.4 tictactoe.ObjectCompositionUtils	6
2.4 Controller komponens.....	7
2.4.1 tictactoe.applicationstates package	7
2.4.2 tictactoe.exceptions:	7
2.5 View komponens	8
2.5.1 tictactoe.views package osztályai:.....	8
2.5.2 tictactoe.views.mainmenu package:	9
2.5.3 tictactoe.views.initializegameview:	10
2.5.4 tictactoe.views.tictactoeview:	11
Esemény-eseménykezelő párok	12
Tesztelési Terv.....	12

1. Feladat leírása:

Készítsünk programot, amellyel a következő két személyes játékot lehet játszani. Adott egy $n \times m$ mezőből álló tábla (n az oszlopok, m a sorok száma), amelyre a játékosok X, illetve O jeleket potyogtatnak (azaz egy adott oszlopban a karakter mindig „leesik” a legalsó üres sorba, függetlenül attól, melyik sorban helyeztük le). A játékosok felváltva lépnek, és egy oszlopban csak akkor helyezhetnek el új jelet, ha az még nem telt meg. A játékot az nyeri, aki előbb elhelyez vízszintesen, vagy átlósan négy szomszédos jelet. A játék döntetlennel ér véget, ha betelik a tábla. A program biztosítson lehetőséget új játék kezdésére a táblaméret megadásával (8×5 , 10×6 , 12×7), és ismerje fel, ha vége a játéknak. Ekkor jelenítse meg, hogy melyik játékos győzött (ha nem lett döntetlen), majd kezdjen automatikusan új játékot.

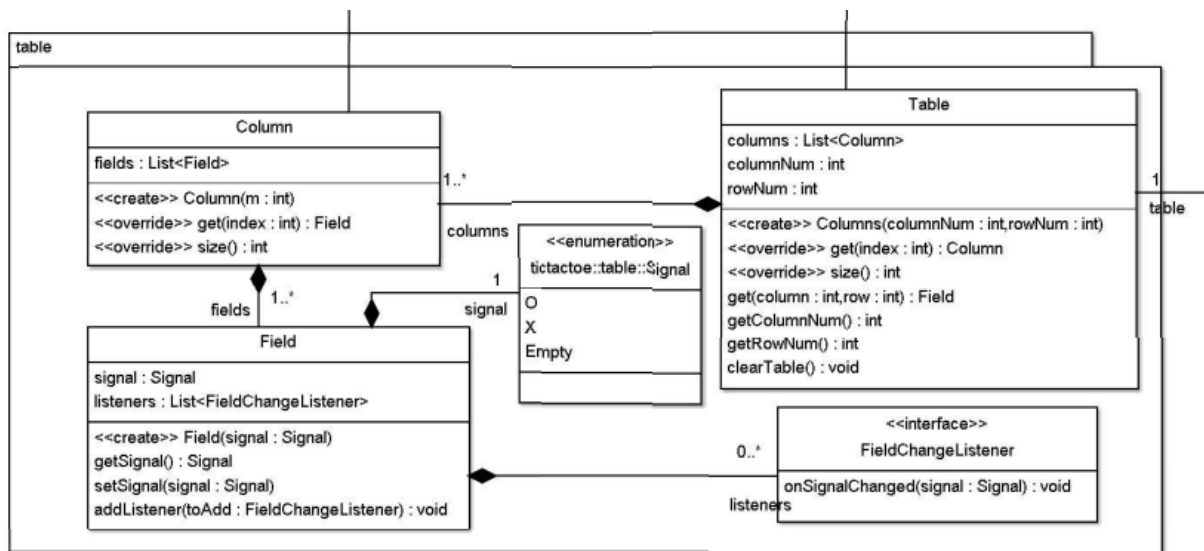
2. Megoldási terv:

2.1 Megoldás leírása

A feladat megoldását az MVC architektúra alapján készítettem el. Ami azt jelenti, hogy a program szétválasztható három komponensre, ami a Model, a View és a Controller. Részletesen bemutatom a következő fejezetekben, hogy mely package-ek, illetve osztályok teszik ki a három komponenst.

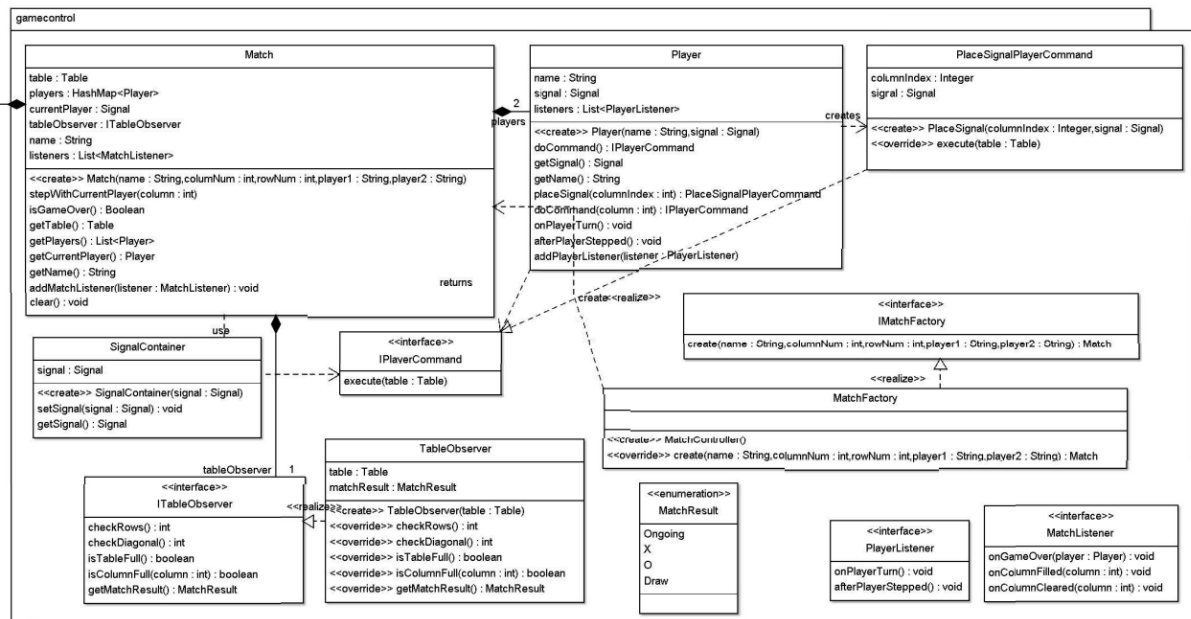
2.2 Model komponens

2.2.1 tictactoe.table package:



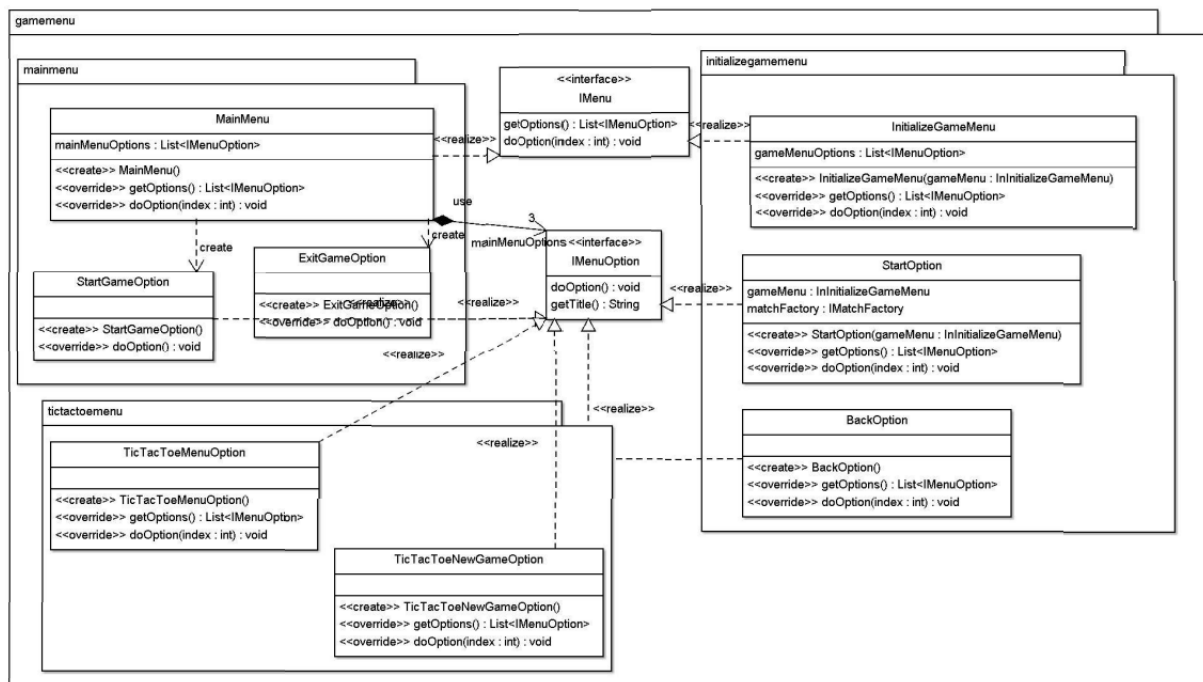
A model komponens table csomagja tartalmazza a pálya logikáját. A Table és a Column osztály egy `AbstractList` generikus osztályból leszármazó osztály. Azért választottam ezt a lista fajtát, mert így a pálya nem módosítható, mivel csak a `get` és a `size` metódusait írtam felül a két osztályban. A Field osztály egy mezőt reprezentál és tartalmazza azt, hogy az üres-e, X, illetve O jel van-e rakva. Ennek van gettere, illetve settere is. A `FieldChangeListener` osztály egy eseményt reprezentál, amire fel lehet iratkozni. A feliratkozott példányok `onSignalChanged` metódusa, akkor hívódik meg, amikor a `Field.setSignal` metódus meghívódik.

2.2.2 tictactoe.gamecontrol package:



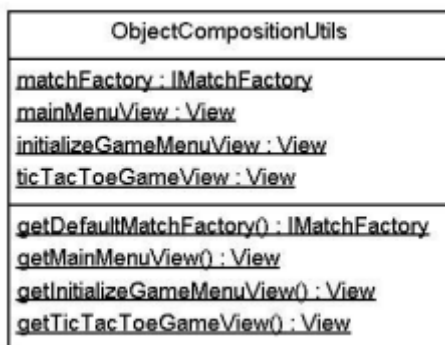
Ez a játékmenet leveznélésért felelő csomag. A játékosok „utasításait” command pattern segítségével implementáltam. A Player osztály felelős a játékos adatainak lekéréséért és az utasítások elkészítéséért. A TableObserver osztály implementálja az ITableObserver interfészt. Ez az osztály ellenőrzi a játékmenet tábláját a játék közben, illetve állítja a MatchResult attribútumának értékét, amit le tudnak kérni kívülről. A Match osztály pedig tartalmazza a táblát, játékosokat, a egy TableObserver példányt, illetve ezek segítségével vezényli a játékot. Két féle eseményfigyelő interfész van ebben a csomagban. Egy, ami a játékosok állapotváltozását figyeli, illetve egy, ami a játékmenet változásait figyeli.

2.2.3 tictactoe.gamemenu package:



A gamemenu csomag a különböző menüelemek logikáját valósítja meg. Az `IMenuOption` interfész `doOption` metódus implementálása adja meg, hogy az adott menü elem az mit fog csinálni, amikor megnyomják vagy kiválasztják, illetve a `getTitle` metódus, pedig a View által megjelenítendő szöveggel tér vissza.

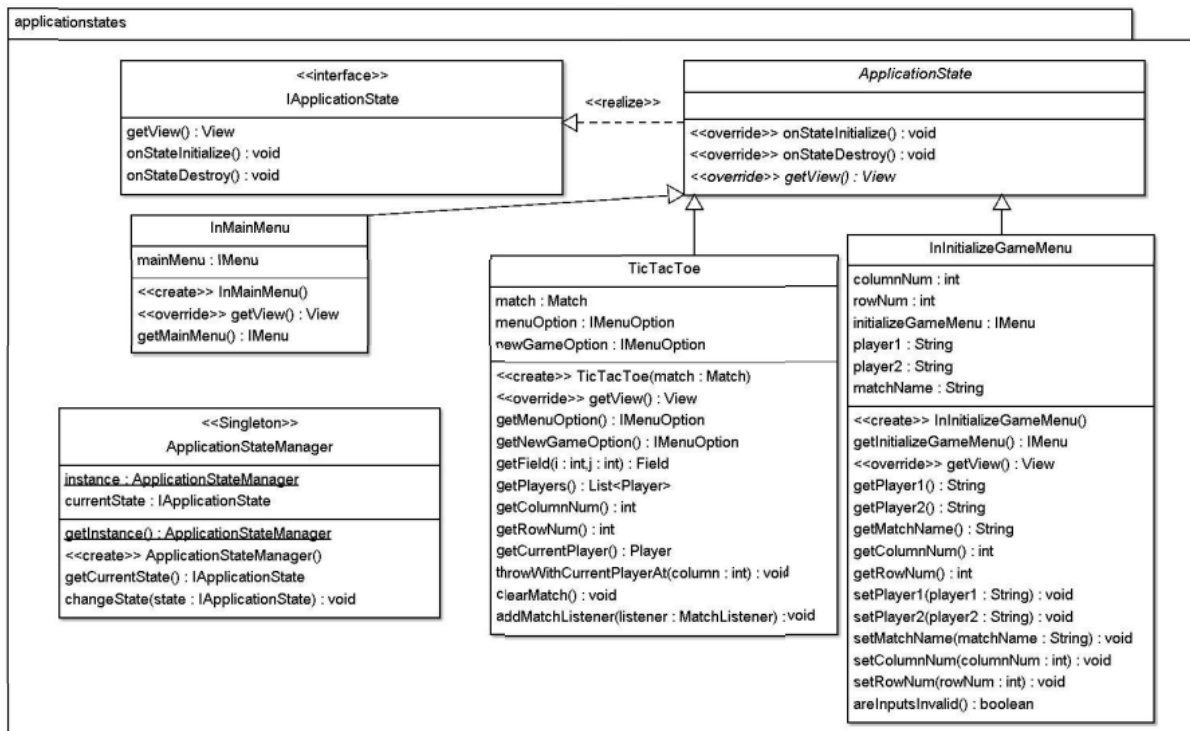
2.2.4 tictactoe.ObjectCompositionUtils



Ez az osztály különböző Singleton objektumokat tartalmaz. Tartalmazza a különböző `View`-okat, illetve az alapértelmezett `IMatchFactory`-t implementáló osztály egy példányát.

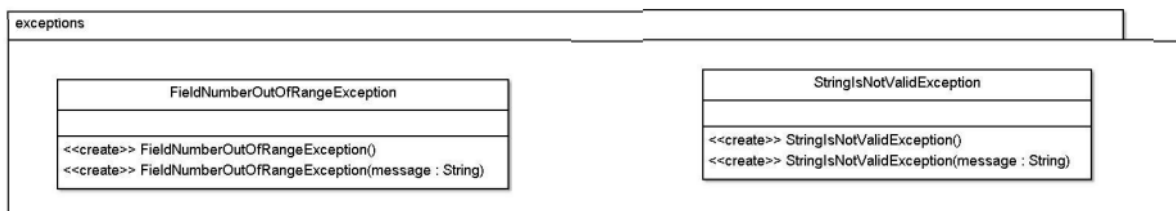
2.4 Controller komponens

2.4.1 tictactoe.applicationstates package



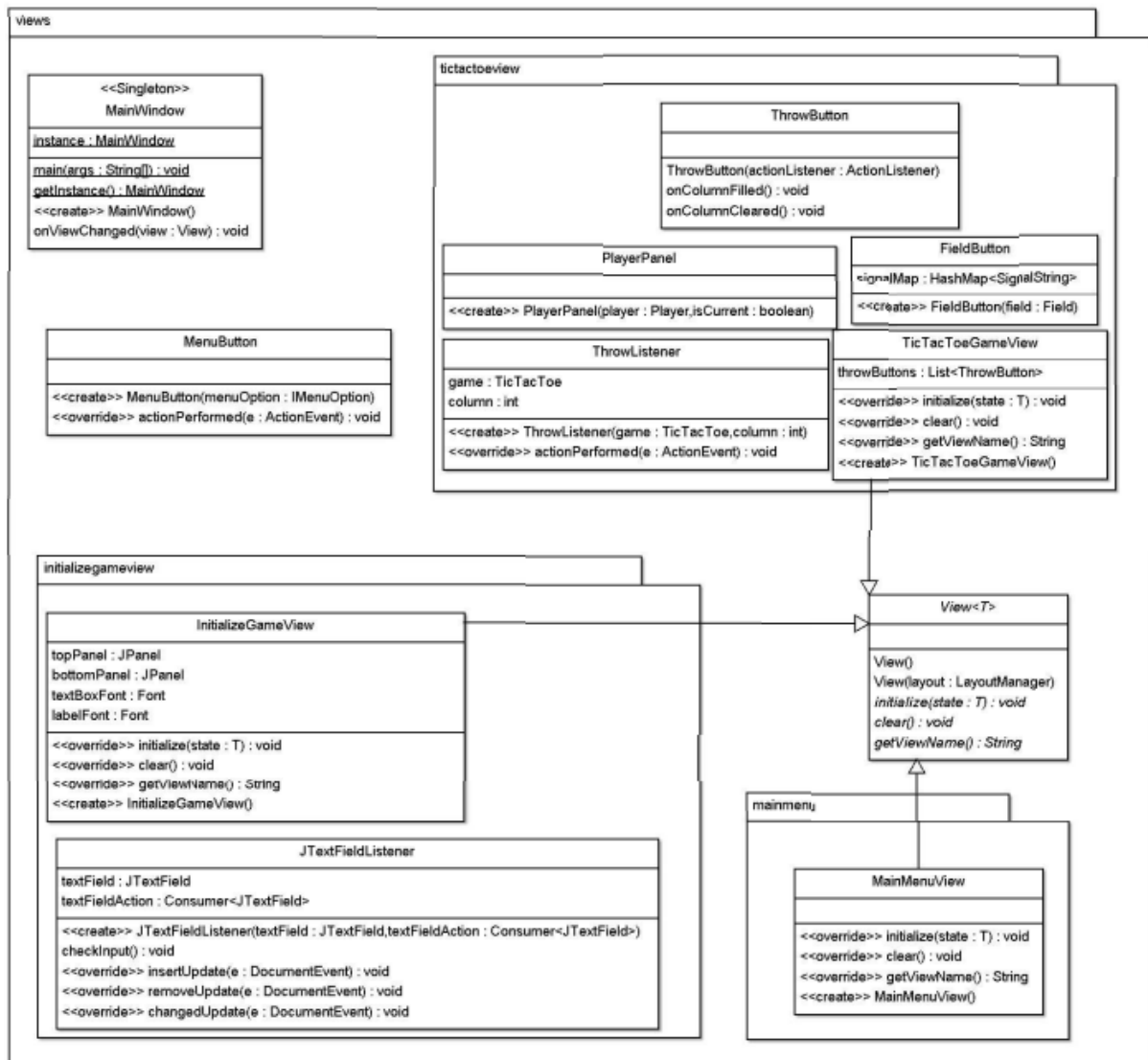
Ez a csomag reprezentálja a „Controller” komponenst, tehát ezen keresztül kommunikál a view a model-el. A programnál minden állapothoz egy nézet-et, vagyis egy oldalt rendeltem, ezért van három féle ApplicationState, hiszen 3 féle nézet létezik. Az ApplicationStateManager osztály felelős azért, hogy az átmenetet megvalósítja a különböző alkalmazás állapotok között. Amikor az alkalmazás állapota változik, akkor a nézet is változik a megfelelő nézetre.

2.4.2 tictactoe.exceptions:



Ez a csomag tartalmazza a kivételeket, amiket az InInitializeGameMenu osztályban dobunk.

2.5 View komponens



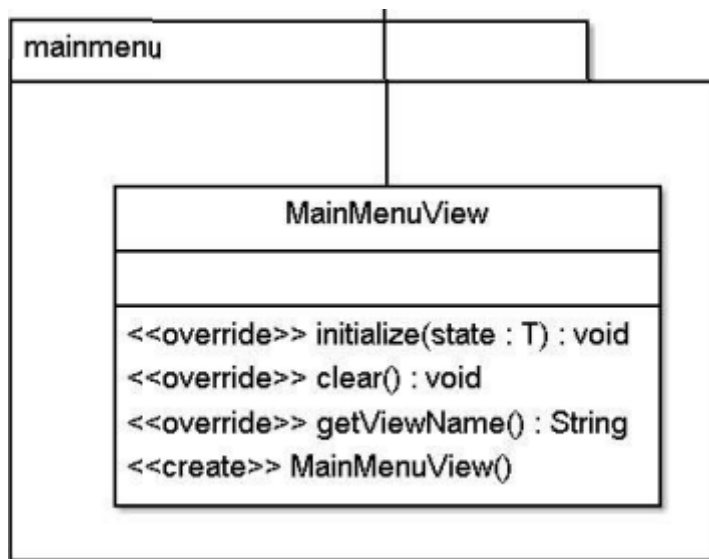
2.5.1 tictactoe.views package osztályai:

A MainWindow osztály a JFrame osztályból származik és innen indul el az alkalmazás. Ez egy Singleton osztály. A main metódusban a MainWindow.getInstance metódusát hívjuk csak meg. A megfelelő View osztályokat egy CardLayout-ba töltjük be a konstruktorban, illetve az onViewChanged metódus segítségével válthatunk a nézet-ek között. Ezt a metódust az ApplicationState.onStateInitialize metódusa hívja meg.

A MenuButton általánosan a menü gombokat reprezentáló osztály, ami egy IMenuOption-t kap meg és a doOption metódusát hívja meg a gomb megnyomásakor.

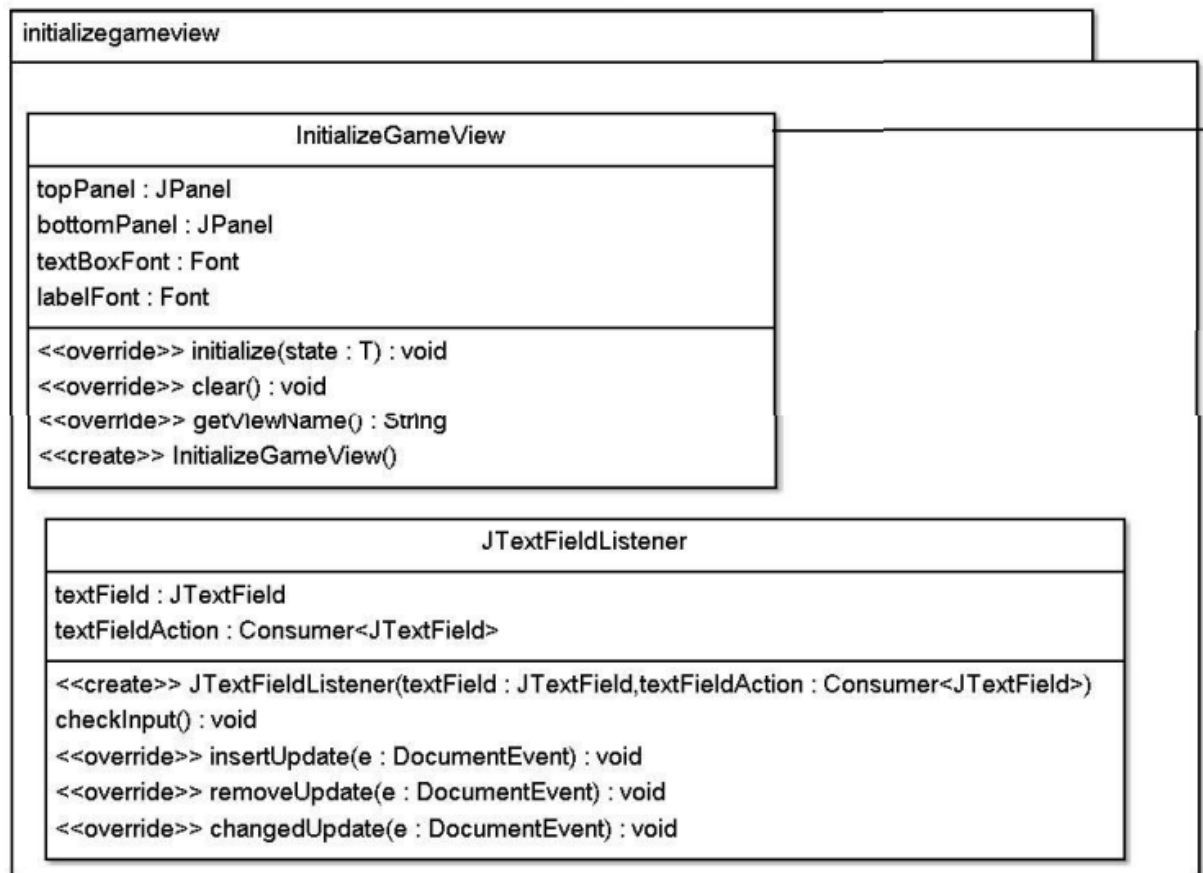
A View osztály egy generikus osztály, ami a különböző nézetek ősoosztálya.

2.5.2 tictactoe.views.mainmenu package:



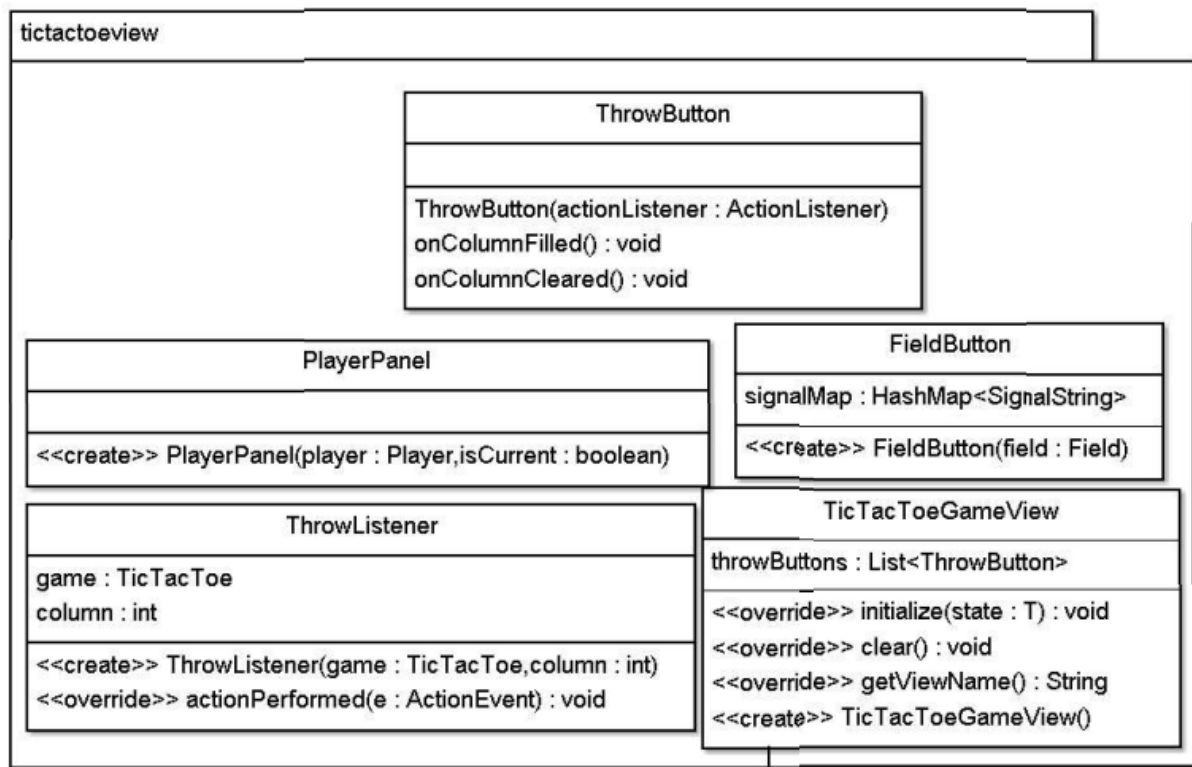
A `MainMenuView` a `View<InMainMenu>` osztályából származik le, így az `initialize` egy `InMainMenu` példányt kap. Ez alapján lekéri tőle az `IMenu` attribútumát, majd `MenuButton` gombokat helyez el egymás alá középre.

2.5.3 tictactoe.views.initializegameview:



A `InitializeGameView` a `View<InInitializeGameMenu>` osztályából származik le, így az `initialize` egy `InInitializeGameMenu` példányt kap. Ez alapján inicializálja a különböző vezérlőket, illetve a `JTextFieldListener` osztály segítségével a `JTextField`-ekhez hozzárendeli az `InInitializeGameMenu` különböző settereit. Tehát, amikor a text field-ek értéke változik, akkor a controller példány értéke is változik, ami kivételt dob, ha valamilyen input nem megfelelő. Ekkor piros keretet kap a text field.

2.5.4 tictactoe.views.tictactoeview:



Ez a csomag tartalmazza a játékmenet nézetéhez szükséges osztályokat. A `FieldButton` reprezentálja a mezőket, ami egy `field`-et kap a konstruktorában és feliratkozik a `FieldChangeListener.onSignalChanged` eseményére. A `signalMap` attribútum segítségével változtatja át `String` típusúra az adott `Signal` objektumot és azt állítja be a saját szövegének, amikor változik a `Signal` attribútum. A `PlayerPanel` a játékos releváns adatait megjelenítő panel, a `ThrowButton`, pedig a pálya feletti gombokat reprezentálja, amit megnyomva a játékos ledobja az adott oszlopba a saját jelét. A `TicTacToeGameView` ezen osztályok segítségével készíti el a játék nézetét.

Esemény-eseménykezelő párok

Esemény	Eseménykezelő
A játékos az egyik oszlopban elhelyezi a saját jelét	ThrowListener
A játékos megnyom egy menü gombot	MenuButton
A játékos beír valamit az egyik szövegmezőbe	TextFieldListener
Egy mező jele megváltozik a pályán	FieldChangeListener
Játékosok váltakozása a nézetben játékmenet közben	PlayerListener
Játék vége	MatchListener
A pályán egy oszlop betelik	MatchListener
A pályát alaphelyzetbe állítják	MatchListener

Tesztelési Terv

Név	Lépések
Játék kezdés teszt	<ul style="list-style-type: none">Indítsd el a programotNyomj a Start Game gombraNyomj a Start gombra <p>Elvárt működés: A játékmenet elindul a player1 nevezetű játékos zöld színű, tehát ő a kezdő játékos, a tábla üres, illetve a tábla felett lévő gombok zöld kerettel rendelkeznek és megnyomhatóak. Bal felső saroknál két menüelem látszódik.</p>
Játék vége teszt1	<ul style="list-style-type: none">Indítsd el a programotNyomj a Start Game gombraNyomj a Start gombraNyomd meg az 1. oszlop gombjátNyomd meg a 2. oszlop gombjátNyomd meg a 2. oszlop gombjátNyomd meg a 3. oszlop gombjátNyomd meg a 3. oszlop gombjátNyomd meg az 1. oszlop gombjátNyomd meg a 3. oszlop gombjátNyomd meg a 4. oszlop gombjátNyomd meg a 4. oszlop gombjátNyomd meg a 4. oszlop gombját <p>Elvárt működés: A lépések sikeresek és az utolsó lépés után jelzi a program a felhasználónak, hogy a player1 nevű játékos nyert 4 átlósan kirakott jellel.</p>
Játék vége teszt2	<ul style="list-style-type: none">Indítsd el a programotNyomj a Start Game gombra

	<ul style="list-style-type: none">• Nyomj a Start gombra• Nyomd meg az 1. oszlop gombját• Nyomd meg a 1. oszlop gombját• Nyomd meg a 2. oszlop gombját• Nyomd meg a 2. oszlop gombját• Nyomd meg a 3. oszlop gombját• Nyomd meg az 3. oszlop gombját• Nyomd meg a 1. oszlop gombját• Nyomd meg a 4. oszlop gombját• Nyomd meg a 2. oszlop gombját• Nyomd meg a 4. oszlop gombját <p>Elvárt működés: A lépések sikeresek és az utolsó lépés után jelzi a program a felhasználónak, hogy a player2 nevű játékos nyert 4 darab sorban kirakott jellel.</p>
Játék vége teszt3	<ul style="list-style-type: none">• Indítsd el a programot• Nyomj a Start Game gombra• Nyomj a Start gombra• Nyomd meg az 1. oszlop gombját• Nyomd meg a 2. oszlop gombját• Nyomd meg a 1. oszlop gombját• Nyomd meg a 2. oszlop gombját• Nyomd meg a 1. oszlop gombját• Nyomd meg az 2. oszlop gombját• Nyomd meg a 1. oszlop gombját• Nyomd meg a 2. oszlop gombját• Nyomd meg a 3. oszlop gombját• Nyomd meg a 4. oszlop gombját• Nyomd meg a 3. oszlop gombját• Nyomd meg a 4. oszlop gombját• Nyomd meg a 3. oszlop gombját• Nyomd meg a 4. oszlop gombját• Nyomd meg a 3. oszlop gombját• Nyomd meg a 4. oszlop gombját <p>Elvárt működés: A lépések sikeresek és az utolsó lépés után jelzi a program a felhasználónak, hogy a meccs döntetlen és látszódik, hogy a tábla teljesen betelt az X és az O jelekkel.</p>
Kilépés a játékból teszt	<ul style="list-style-type: none">• Indítsd el a programot• Nyomj az Exit Game gombra <p>Elvárt működés: A program bezáródik, amikor megnyomják az Exit Game gombot.</p>
Vissza a menübe teszt	<ul style="list-style-type: none">• Indítsd el a programot• Nyomj a Start Game gombra• Nyomj a Start gombra• Nyomj a Menu gombra <p>Elvárt működés:</p>

	A játékmenetben a Menu gombra kattintva visszatér a program a főmenübe.
Vissza teszt	<ul style="list-style-type: none">• Indítsd el a programot• Nyomj a Start Game gombra• Nyomj a Back gombra Elvárt működés: A meccs adatainak megadásakor a Back gombra kattintva visszatér a program a főmenübe.
Új játék kezdése teszt	<ul style="list-style-type: none">• Indítsd el a programot• Nyomj a Start Game gombra• Nyomj a Start gombra• Nyomj a New Game gombra• Írd be a Column Number X Row Number mezőbe a 8x5 értéket• Nyomj a Start gombra Elvárt működés: Az első start gombra nyomáskor egy 4x4-es tábla jelenik meg, a második start gombra nyomáskor, pedig egy 8x5-ös tábla jelenik meg.