

# Programozás alapjai 3. - NHF

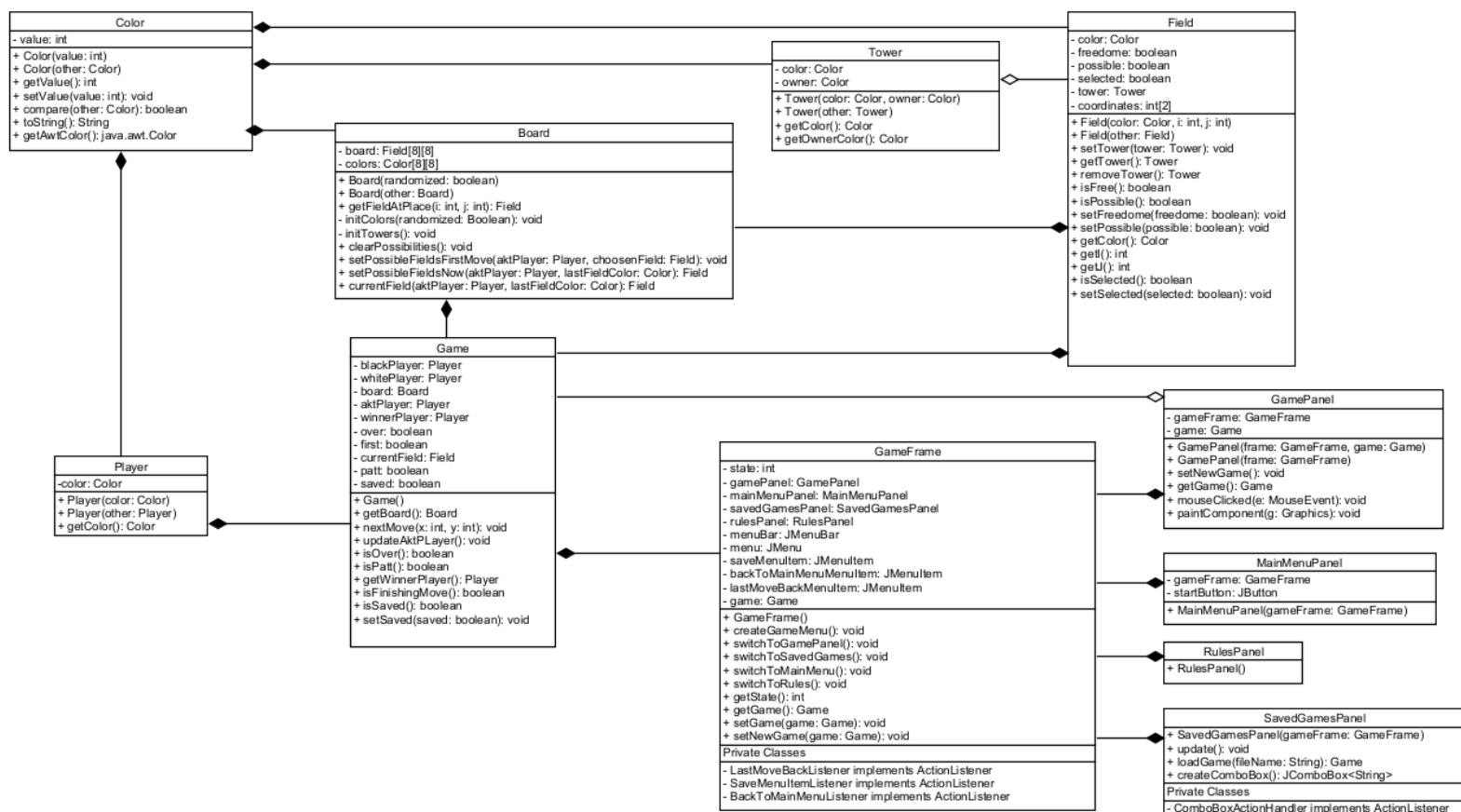
## Dokumentáció

### Kurely Mózés - F0QEL1

# Kamisado

## Megvalósítás

A játékot az UML diagrammon látható, és ezután bemutatott osztályok segítségével valósítom meg.



Az **Elements.Board** osztály felelős a Kamisado játék táblájának kezeléséért. Az osztály implementálja a **Serializable** interfészt, lehetővé téve a játék állapotának mentését és betöltését.

Az osztály fontosabb adatai:

- **board**: A táblát reprezentáló mezők tömbje.
- **colors**: A mezők színeit tároló tömb.

Fontosabb metódusok:

- **getFieldAtPlace(int i, int j)**: Visszaadja a tábla adott pozícióján lévő mezőt.
- **setPossibleFieldsNow(Player aktPlayer, Color lastFieldColor)**: Megjelöli azokat a mezőket, ahova a játékos léphet a jelenlegi állásban.
- **clearPossibilities()**: Eltávolítja az összes lehetséges lépést a táblán.
- **currentField(Player aktPlayer, Color lastFieldColor)**: Visszaadja azt a mezőt, ahol az aktuális játékosnak az a tornya található, amivel lépnie kell.

Az **Elements.Color** osztály felelős a színek kezeléséért a játékban. A színeket egész számok reprezentálják, amelyek az osztály konstansai. Az osztály implementálja a **Serializable** interfészt, lehetővé téve a színobjektumok állapotának mentését és betöltését.

Az osztály fontosabb adatai:

- **value**: Az egész szám érték, amely a színt reprezentálja.

Fontosabb metódusok:

- **compare(Color other)**: Összehasonlítja ezt a színt egy másik színnel.
- **toString()**: Visszaadja a szín szöveges reprezentációját.
- **getAwtColor()**: Visszaadja a szín java.awt.Color reprezentációját.

Az **Elements.Field** osztály felelős a Kamisado játéktábla egy mezőjének kezeléséért. Egy mező egy adott színnel és a játéktáblán való helyzetével van azonosítva. A mező szabad, amikor nincs rajta torony. Az osztály implementálja a **Serializable** interfészt, lehetővé téve a mezők állapotának mentését és betöltését.

Az osztály fontosabb adattagjai:

- **color**: A mező színét adja meg
- **freedome**: Megadja, hogy a mező szabad vagy foglalt.
- **possible**: Megadja, hogy a mező lehetséges lépési helynek ki van-e jelölve.
- **selected**: Megadja, hogy a mező ki van-e választva lépésre.
- **tower**: A mezőn álló torony
- **coordinates**: A mező a táblán való pozícióját tároló tömb, ahol az első elem a sorindex, a második pedig az oszlopindex.

Fontosabb metódusok:

- **setTower(Tower tower)**: Beállítja a mező tornyát, ha a mező szabad.
- **removeTower()**: Eltávolítja a tornyot a mezőről és visszaadja azt.
- **isFree()**: Megadja, hogy a mező szabad-e.
- **isPossible()**: Megadja, hogy a mező lehetséges lépési hely-e.
- **isSelected()**: Megadja, hogy a mező ki van-e választva.

Az **Elements.Tower** osztály felelős a tornyok kezeléséért a játékban. Minden toronyhoz tartozik egy szín (**color**), amely a tornyot reprezentálja, és a torony tulajdonosának a színe (**owner**). Az osztály implementálja a **Serializable** interfészt, lehetővé téve a tornyok állapotának mentését és betöltését.

Az osztály fontosabb adattagjai:

- **color**: A torony színe.
- **owner**: A torony tulajdonosának a színe.

Fontosabb metódusok:

- **getColor()**: Visszaadja a torony színét.
- **getOwnerColor()**: Visszaadja a torony tulajdonosának színét.

Az **Elements.Player** osztály felelős a játékosok kezeléséért a játékban. A játékosokat a színük inicializálja. Az osztály implementálja a **Serializable** interfészt, lehetővé téve a játékosobjektumok állapotának mentését és betöltését.

Az osztály fontosabb adattagjai:

- **color**: A játékos színét reprezentáló Color osztály példánya.

Fontosabb metódusok:

- **getColor()**: Visszaadja a játékos színét.

A **GameControl.Game** osztály felelős a Kamisado játék logikájának kezeléséért. A játék állapotát tárolja, vezérli a játékmenetet, ellenőrzi a lépések helyességét, valamint elvégzi a szükséges frissítéseket a játék során. Az osztály implementálja a **Serializable** interfészt, lehetővé téve a játék állapotának mentését és betöltését.

Az osztály fontosabb adatai:

- **blackPlayer**: A fekete játékos.
- **whitePlayer**: A fehér játékos.
- **board**: A játéktábla.
- **aktPlayer**: Az aktuális játékos.
- **winnerPlayer**: A győztes játékos.
- **over**: Igaz, ha a játék véget ért, különben hamis.
- **first**: Igaz, ha ez az első lépés a játékban, különben hamis.
- **currentField**: Az aktuális mező, ahonnan lépünk.
- **patt**: Igaz, ha a játék pattban van, különben hamis.
- **previousState**: A játék előző állapota, amelyet visszalépés esetén használunk.
- **saved**: Igaz, ha a játék állapota el van mentve, különben hamis.

Fontosabb metódusok:

- **nextMove(int x, int y)**: Elvégzi a következő lépést a játékban, amennyiben az érvényes
- **backOneMove()**: Visszalép egy lépést a játékban
- **isOver()**: Ellenőrzi, hogy a játék véget ért-e.
- **isPatt()**: Ellenőrzi, hogy a játék patthelyzetben van-e.
- **getWinnerPlayer()**: Visszaadja a győztes játékost.

A **GameControl.JKamisado** osztály futtatja a **main** függvényt amely elindítja magát a játékot.

A **View.GameFrame** osztály felelős a játék ablak kezeléséért és megjelenítéséért. A keret tartalmazza a játékhoz tartozó panelokat, valamint a menüsávot.

Fontosabb metódusok:

- **switchToGamePanel():** A keretben megjelenő panelt átváltja a játék panelére.
- **switchToSavedGames():** A keretben megjelenő panelt átváltja a mentett játékok panelére.
- **switchToMainMenu():** A keretben megjelenő panelt átváltja a főmenü panelére.
- **switchToRules():** A keretben megjelenő panelt átváltja a játékszabályzat panelére.
- **getState():** Visszaadja az aktuális állapotot (0 - MainMenu; 1 - Game; 2 - SavedGames; 3 - Rules).
- **setNewGame(Game game):** Új játékot állít be.

A belső osztályok felelősek a menüsáv eseménykezeléséért. (**SaveMenuItemListener, BackToMainMenuListener, LastMoveBackListener**)

A **SavedGamesPanel** osztály felelős a mentett játékok kezeléséért és megjelenítéséért. Az osztály a **JPanel** osztályból származik, és tartalmazza a mentett játékok listáját.

Az osztály adattagjai:

- **gameFrame:** A **GameFrame** osztály példánya, amely tartalmazza ezt a panelt.
- **comboBox:** A legördülő menü a mentett játékok kiválasztásához.
- **savedGames:** Egy térkép, amely a mentett játékok nevét és a hozzájuk tartozó Game objektumokat tartalmazza.

Fontosabb metódusok:

- **update():** Frissíti a legördülő menüt a mentett játékok alapján.
- **loadGame(String fileName):** Betölti a megadott fájlnev alapján a mentett játékot.
- **createComboBox():** Létrehozza a legördülő menüt a mentett játékok listájával.

A belső osztály felelős a comboBox eseménykezeléséért. (**ComboBoxActionHandler**)

A **View.GamePanel** osztály felelős a játéktábla grafikus megjelenítéséért. Az osztály a **JPanel** osztályból származik, és implementálja a **MouseListener** interfészt az egér események kezeléséhez.

Fontosabb metódusok:

- **mouseClicked(MouseEvent e)**: A felhasználó egérekattintása esetén hívódik meg, ahol meghatározzuk a játékbeli lépést, és frissítjük a grafikus megjelenítést.
- **paintComponent(Graphics g)**: A JPanel által meghívott metódus, ahol megrajzoljuk a játékot magát.

Az osztály adatai:

- **gameFrame**: A **GameFrame** osztály példánya, amely tartalmazza ezt a panelt.
- **game**: A játék állapotát tároló **Game** osztály példánya.

A **View.MainMenuPanel** osztály felelős a főmenü grafikus megjelenítéséért és az abban található gombok eseményeinek kezeléséért. Az osztály a **JPanel** osztályból származik, és tartalmazza a főmenü megjelenítéséhez szükséges elemeket.

Az osztály adatai:

- **gameFrame**: A GameFrame osztály példánya, amely tartalmazza ezt a panelt.

A belső osztályok felelősek a menüpontokat irányító gombok eseménykezeléséért. (**StartButtonListener**, **LoadButtonListener**, **RulesButtonListener**)

A **View.RulesPanel** osztály felelős a játékszabályzat grafikus megjelenítéséért. Az osztály a **JPanel** osztályból származik, és tartalmazza a játékszabályzat szövegét.