# Készítette:

**Kis Gergely Domonkos (VMT982)**

E-mail: [tianarath30@gmail.com](mailto:tianarath30@gmail.com) , [vmt982@inf.elte.hu](mailto:vmt982@inf.elte.hu)

Csoportszám: **6**

# Feladat

**4-es játék**

Készítsünk programot, amellyel a következőkét személyes játékot lehet játszani. Adott egy 5×5mezőből álló tábla, amelynek mezői 0 és 4 közötti értékeket tartalmaznak. Kezdetben minden mezőn a 0 érték van. Ha a soron következő játékos a tábla egy tetszőleges mezőjét kiválasztja, akkor az adott mezőn és a szomszédos négy mezőn az aktuális érték eggyel nő felfelé, ha az még kisebb, mint 4. Aki a lépésével egy, vagy több mező értékét 4-re állítja, annyi pontot kap, ahány mezővel ezt megtette. A játékosok pontjait folyamatosan számoljuk, és a játékmezőn eltérő színnel jelezzük, hogy azt melyik játékos billentette 4-esre. A játék akkor ér véget, amikor minden mező értéke 4-et mutat. Az győz, akinek ekkor több pontja van. A program ismerje fel, ha vége a játéknak. Ekkor jelenítse meg, melyik játékos győzött, majd automatikusan kezdjen új játékot.

# A feladat értelmezése

A játékban két játékos vesz részt, de az alkalmazás szempontjából mindig csak egy felhasználó van. Az alkalmazás indítását követően az alábbi tevékenységeket végezheti a felhasználó:

*Főmenü*

Főmenü tevékenyégek

***Játékos***

Játéktér tevékenységek

***Játékos***

*Játéktér*

A felhasználói tevékenységek során az alábbi esetek következhetnek be.

*Főmenü*

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **1** | Az alkalmazás indítása | GIVEN | Az telepítve van. |
| WHEN | Az alkalmazás indítása. |
| THEN | Megjelenik a főablak a főmenüvel. |
| **2** | Kilépés | GIVEN | Az alkalmazás fut |
| WHEN | Az alkalmazás ablakának lezáró ikonjára kattintunk |
| THEN | Az alkalmazás bezáródik |
| **3** | Játék indítása | GIVEN | A játékosadatok helyesek:   * Nincs egyező játékosnév * Nincs egyező játékos szín |
| WHEN | A „Play” feliratú gombra kattintáskor |
| THEN | A menü eltűnik és létrejön a játéktér. |
| **4a** | Játékosszín választás (1) | GIVEN | A menü látható |
| WHEN | Az **első játékoshoz** tartozó „color” feliratú gombra kattintáskor |
| THEN | Felugrik egy színválasztó ablak, amelynek elfogadásával beállítódik az **első játékoshoz** tartozó játékos szín. |
| **4b** | Játékosszín választás (2) | GIVEN | A menü látható |
| WHEN | Az **második játékoshoz** tartozó „color” feliratú gombra kattintáskor |
| THEN | Felugrik egy színválasztó ablak, amelynek elfogadásával beállítódik az **második játékoshoz** tartozó játékos szín. |

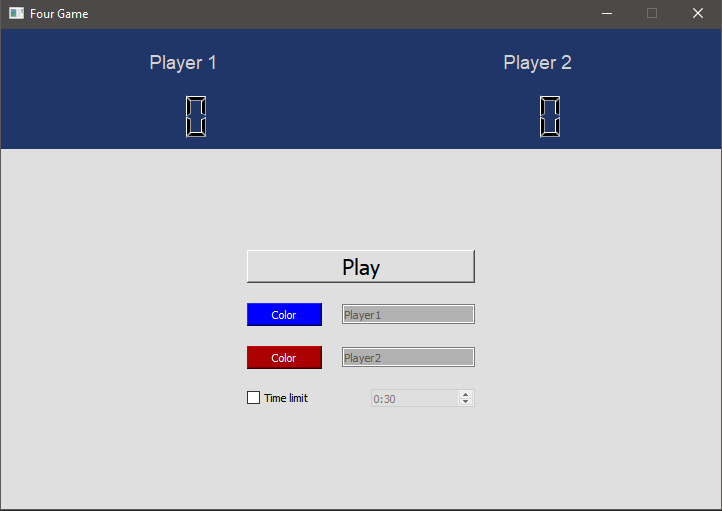
|  |  |  |  |
| --- | --- | --- | --- |
| **5a** | Játékosnév választás (1) | GIVEN | A menü látható |
| WHEN | Az első játékoshoz tartozó szövegmezőbe írunk |
| THEN | A szövegmezőbe írt szöveg lesz az **első játékos** játékosneve. |
| **5b** | Játékosnév választás (2) | GIVEN | A menü látható |
| WHEN | A második játékoshoz |
| THEN | A szövegmezőbe írt szöveg lesz a **második játékos** játékosneve. |
| **6** | Köridő engedélyezése | GIVEN | A menü látható |
| WHEN | A „Time limit” címkével ellátott jelölődobozt bepipáljuk |
| THEN | Elérhetővé válik a mellette lévő időválasztó, amellyel egy játékos körének idejét maximalizálhatjuk. |
| **7** | Köridő választás | GIVEN | * A menü látható * A „Time limit” jelölődoboz be van pipálva |
| WHEN | Változtatunk az időválasztó értékén |
| THEN | Változik a maximális köridő |

A felhasználói tevékenységek során az alábbi esetek következhetnek be.

*Játéktér*

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **1** | Kilépés | GIVEN | Az alkalmazás fut  A játéktér aktív |
| WHEN | Az alkalmazás ablakának lezáró ikonjára kattintunk |
| THEN | Az alkalmazás bezáródik. |
| **2a** | Lépés | GIVEN | A játéktér aktív |
| WHEN | Az aktív játékos egy szabad (4-nél kisebb értékű) mezőre kattint, és marad kattintható mező |
| THEN | A mező és a körülötte elhelyezkedő másik 4 mező, amelyeknek szintén 4-nél kisebb értéke van (ha ezek léteznek) értékük eggyel nő, és az aktív játékos a másik játékos lesz.  Azok a mezők, amelyek elérték a 4-es értéket lezáródnak és nem lesznek kattinthatóak, valamint az utolsó növelést elvégző játékos játékosszínére színeződik a hátterük. |
| **2b** | Lépés | GIVEN | A játéktér aktív |
| WHEN | Az aktív játékos egy szabad (4-nél kisebb értékű) mezőre kattint, és nem marad kattintható mező |
| THEN | A játék véget ér és megjelenik egy felugró ablak, amely közli melyik játékos nyert és hogy alakultak a pontok a játék során. Az OK gombra kattintva új játék indul ugyanezekkel a paraméterekkel. |

# Felhasználói felület terve

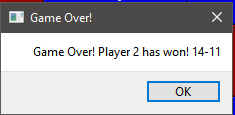
Az alkalmazás egyetlen fixméretű ablakból áll, amely megnyitás után egy állapotsávot (játékosok nevei és pontjai), alatta pedig egy menüt tartalmaz.

Főmenü ablak

A menü 3 gombból (Indító és színválasztók), 2 szövegbeviteli mezőből (játékosnevek), egy jelölődobozból (legyen-e köridő limit), és egy időválasztóból (milyen hosszú köridő legyen) áll.

Játéktér ablak

A játék elindítását követően jelenik meg a menü helyén a játéktér, amely 5x5 nyomógombot tartalmaz, azokon pedig külön-külön az aktuális mező értéke.

Az éppen aktuális játékos nevének színe pedig sárga az állapotsávon.

Játék végét jelző üzenetablak

# Osztálydiagram

Az alkalmazást 3 osztállyal írjuk le. Külön osztályokba lett szervezve a felhasználói felület, a játéklogika, és a játékos leírása.

## Felhasználói felület

QWidget

|  |
| --- |
| GameLogic |
| … |

|  |
| --- |
| Ui\_FourGameWindow |
|  |

|  |
| --- |
| Player |
| … |

|  |
| --- |
| FourGameWindow |
|  |
| + FourGameWindow() |
| + ~FourGameWindow() |
| - initGameField() : **void** |
| - incrementButton(row:**int**,col:**int**) : **void** |
| - playReady() : **bool** |
| **<<slots>>** |
| - startNewGame() : **void** |
| - updateTime() : **void** |
| - setPlayerColor(btn:**QPushButton**) : **void** |
| - on\_scored(row:**int**,col:**int**) : **void** |
| - on\_playerChanged() : **void** |
| - gameButtonClick(row:**int**,col:**int**) : **void** |
| - gameEnded() : **void** |

QGridLayout

QPushButton

QLineEdit

QTimeEdit

QCheckBox

QLCDNumber

QLabel

1

1

3

1

2

2

2

1

2

1

## Játéklogika

|  |
| --- |
| GameLogic |
| + rowCount : **int**  + colCount : **int**  + maxFieldValue : **int** = 4  - gameTimer : **QTimer**  - players : **Player[]**  - activePlayer : **Player**  - activePInd : **int**  - fields : **int[,]** |
| + GameLogic() |
| + ~GameLogic() |
| + getFieldValue(row:**int**,col:**int**) : **int** |
| + addPoint(row:**int**,col:**int**) : **int** |
| + isValidField(row:**int**,col:**int**) : **bool** |
| + checkWinCond() : **void** |
| + getPlayer(n:**int**) : **Player** |
| + getActivePlayer() : **Player** |
| + changeActivePlayer() : **void** |
| **<<signals>>** |
| + scored(row:**int**,col:**int**) : **void** |
| + playerChanged() : **void** |
| + endGame() : **void** |

QObject

|  |
| --- |
| QTimer |
|  |

|  |
| --- |
| Player |
| … |

## Játékos

|  |
| --- |
| Player |
| - playerName : **QString**  - startTime : **QTime**  - color : **QColor**  - gameTime : **QTime**  - score : **int** |
| + Player() |
| + ~Player() |
| + getPlayerName() : **QString** |
| + getGameTime() : **QTime** |
| + subTime() : **QTime** |
| + resetTime() : **void** |
| + getScore() : **int** |
| + addScore() : **int** |
| + getColor() : **QColor** |
| **<<signals>>** |
| + timeEnded() : **void** |

QObject

|  |
| --- |
| QTime |
|  |

|  |
| --- |
| QColor |
|  |

# Vezérlők és adattagok

## Felhasználói felület

### Vezérlők és adattagok

Az osztály a QWidget-ből származik majd, és a vezérlőit és elrendezőit a Ui\_FourGameWindow típusú adattagjában tárolja. A felhasználói felület a Qt Designerrel készült, így a további módosításokat a *fourgamewindow.ui* nevű fájlban kell eszközölni.

Az osztály tartalmaz továbbá egy játéklogikát leíró objektumot (*game*), és a játékosok adatait tartalmazó QVector objektumot (*players*);

### Metódusok

A legfontosabb eseménykezelő metódus a ***startNewGame***() ,amely a „Play” gomb megnyomására, továbbá a játék végét követő üzenetablak bezárása után fut le. Ez példányosít egy GameLogic objektumot, és elvégzi a játéklogika, játékosadatok, és a játéktér beállítását.

További metódusok és feladataik:

|  |  |
| --- | --- |
| *initGameField*(): | Létrehozza a gombokból álló játékteret. |
| *incrementButton(…):* | A gomb feliratát frissíti a megnövelt pontra. |
| *playReady():* | Ellenőrzi hogy ütköznek-e a játékosadatok. |

### Eseménykezelés

A apróbb eseményeket kezelő kódrészletek névtelen (*lambda*) eseménykezelőket kaptak, így azok ebben a listában nem szerepelnek.

|  |  |  |  |
| --- | --- | --- | --- |
| *updateTime()* | | A köridő megjelenítését frissíti másodpercenként. | |
| ***sender***:  game->gameTimer | ***signal***:  timeout() | ***receiver***:  FourGameWindow | ***slot***:  *updateTime()* |
| *setPlayerColor(…)* | | Gomb megnyomás hatására színválasztó dialógusablakot nyit meg és frissíti a játékos színt. | |
| ***sender***:  Ui->colorBtn | ***signal:***  clicked() | ***receiver:***  FourGameWindow | ***slot:***  setPlayerColor() |
| *on\_scored()* | | Frissíti a pontszámot megjelenítő displayt. | |
| ***sender:***  game | ***signal:***  scored() | ***receiver:***  FourGameWindow | ***slot:***  on\_scored() |
| *on\_playerChanged()* | | Frissíti az aktuális játékost jelző címkét. | |
| ***sender***:  game | ***signal:***  playerChanged() | ***receiver:***  FourGameWindow | ***slot:***  on\_playerChanged() |
| *gameButtonClick(…):* | | Meghívja a megfelelő gombok növelését végző metódust, játékost cserél, és ellenőrzi hogy vége van-e a játéknak. | |
| ***sender:***  Ui->playBtn[i,j] | ***signal:***  clicked() | ***receiver:***  FourGameWindow | ***slot:***  gameButtonClick() |
| *gameEnded()* | | Egy dialógusablakban megjeleníti a győztest. | |
| ***sender:*** game | ***signal:*** endGame() | ***receiver:*** FourGameW | ***slot:*** gameEnded() |

## Játéklogika

### Adattagok

Az osztály eltárolja a játékosok adatait, az aktív játékost, egy időzítőt ami méri a köridőt, valamint a tábla méreteit és a maximum mezőértéket.

Továbbá a mezők értékét egy nxn-es Mátrixban (fields) tárolja.

### Metódusok

A getter setter metódusokon kívül három érdembeli metódus van. Az addPoint() növeli a mező értéket és signált küld (scored()) ha pontot szerzett az aktív játékos. A changeActivePlayer() megcseréli az aktív játékost, a checkWinCond() pedig ellenőrzi, hogy vége van-e a játéknak.

### Eseménykezelés

Ennek az osztálynak egyetlen eseménykezelő metódusa van, amely kicseréli az aktív játékost ha annak az aktuális körideje lejárt.

Az osztály az alábbi signálokat küldheti:

scored() – pontszerzés esetén

playerChanged() – játékos csere történt

endGame() – játék vége

Eseménykezelők

|  |  |  |  |
| --- | --- | --- | --- |
| ***sender***:  Player | ***signal***:  timeEnded() | ***receiver:***  GameLogic | ***slot:***  ChangeActivePlayer() |

## Játékos

Ez az osztály egy egyszerű adatleíró osztály, amely nem szorul magyarázatra az osztályleírás a fenti UML diagrammon látható.