**EVA1 – Harmadik beadandó dokumentációja: Potyogós amőba**

# **Feladat:**

Készítsünk programot, amellyel a következő két személyes játékot lehet játszani. Adott egy 𝑛 × 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), 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:**

* A játékot egy grafikus felületen jelenítjük meg, melyre elemi grafikával helyezzük rá a mezőket, vonalak kirajzolásával.
* Koordináták lekérésével állapítjuk meg, hogy melyik mező területére kattintottunk, majd megjelenítjük az aktuális játékos szimbólumát az adott oszlopban lévő legalsó üres mezőn. Az aktuális játékost minden lépés után váltjuk.
* Ha az adott mezőn van már szimbólum, akkor nem történik semmi.
* A grafikus felületet főwidgetként állítjuk be egy főablakon, melyen elheyezkedik 2 menüpont, melyek segítségével az új játékot kezdhetjük, illetve elmenthetjük az aktuális állást vagy betölthetünk egy előzőt.
* A mentést fájl segítségével végezzük, minden sorba egy adat kerül: az első sor az eddigi lépések száma, a második a következő játékos, a maradékba pedig 0,1 és 2 számok kerülnek, ahol 0 az üres hely, az 1 az X szimbólumú mező, a 2 pedig a O szimbólumot képviseli.
* Lehetőséget adunk játékállás elmentésére (Ctrl+S) és betöltésére (Ctrl+L), ehhez a felhasználó 5 mentési hely közül választhat (egy külön ablakban).
* A játék állását egy egészeket tartalmazó mátrixban tároljuk, ahol az üres mezők 0, az X játékos mezői 1-es, a O játékos mezői 2-es értéket kapnak.
* létrehozunk egy betöltésre és egy mentésre szolgáló ablakot (SaveWidget, LoadWidget), a modellt pedig kiegészítjük a műveletekkel (saveGame, loadGame), valamint a játéklista lekérdezésével (saveGameList).

# **Használati esetek:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sorszám | Felhasználói eset | Leírás | |
| 1. | Alkalmazás indítása | Given | Az alkalmazás telepítve van. |
| When | Alkalmazás indítása. |
| Then | Megjelenik egy főablak, üres játéktáblával. |
| 2. | Kilépés | Given | Játék felület. |
| When | A játék menü Kilépés menüpontra kattintás. |
| Then | Alkalmazás bezárása. |
| 3. | Lépés | Given | A játék fut. |
| When | Rákattintás az egyik mezőre. |
| Then | A megfelelő szimbólum a megfelelő mezőre kerül. |
| 4. | Új játék | Given | A játék menü új játék menüpontjára kattintás, vagy az új játék beállítása menüpontra kattintás. |
| When | A játék megállítása/folytatása gombra kattintás. |
| Then | Az új játék menüpont esetén az aktuális mérettel új, üres játéktábla a jelenik meg, míg beállítás esetén egy dialógus, ahol beállíthatjuk a méretet és a játékosok neveit. |
| 5. | Mentés | Given | A játéktábla aktív. |
| When | CTRL-S, vagy Játék mentése menüpontra kattintás. |
| Then | Egy külön ablakban lehet az aktuális játékot elmenteni. |
| 6. | Betöltés | Given | A játéktábla aktív. |
| When | CTRL-L, vagy Játék betöltése menüpontra kattintás. |
| Then | Egy külön ablakban lehet egy elmentett állást betölteni. |

# **Tervezés:**

A program szerkezetét két rétegre bontjuk a modell/nézet architektúrának megfelelően. A modell eseményeken keresztül kommunikál a nézettel. A modellt a ***GameModel*** osztály valósítja meg, amely a játéklogikát biztosítja.

* Az eseménykezelés megvalósítása érdekében az osztályt a *QObject*- ből származtatjuk.
* A játékot egy időzítő vezérli ( guardstep ), amely a vadőrök léptetéséért felel, valamint ez vezérli ajáték megállítását, illetve elindítását.
* Négy eseménye a gameWon(), gameOver(), updateTime(), illetve a fieldChanged( <Előző koordináta>, <Új koordináta>, <Mező új típusa>), amely paraméterek segítségével adja meg a módosításokat.
* Az őröket, piknikkosarakat, illetve lépésirányt vektorban tároljuk.
* A külső hozzáférést 3 eseménykezelő ( newGame(<méret>, <kosarak>, <őrök>, <akadályok>), stepGuard(), pauseOrPlay() ) biztosítja.
* A játéktáblát egy FieldType ( felsorolási típus, amely tartalmazza a mezők lehetséges fajtáit ) típusú mátrixban tároljuk.
* A játéktábla könnyeb lekérdezéséhez definiálunk egy getField( <x koordináta>, <y koordináta>) függvényt, amely megadja az aktuális indexen lévő mező típusát.

A megjelenítést a ***GameView*** biztosítja, amely a *QWidget* leszármazottja.

* A felület felső sorában elhelyezünk 2 címkét, amely a megszerzett piknikkosarak számát, illetve az eltelt játékidőt mutatja, valamint elhelyezünk 3 nyomógombot, amely a 3 különböző méretű pálya létrehozásáért felelnek, megfelelő eseménykezelővel.
* A megjelenítéshez nyomógombokat használunk, melyeknek hátteret állítunk a mező típusának megfelelően.
* Lekezeljük a játéklogika eseményeit: gameWon() és gameOver() eseménynél leállítjuk a timert és *QMessageBox* ugrik fel.
* A fieldChanged esemény eseménykezelője átszínezi a megfelelő nyomógombokat.

|  |
| --- |
| *QWidget*  GameView |
| -size: int  -smallButton: QPushButton\*  -mediumButton: QPushButton\*  -bigButton: QPushButton\*  -pause: QPushButton\*  -bsk: QLabel\*  -time: QLabel\*  -grid: QGridLayout\*  -main: QVBoxLayout\*  -\_model: GameModel  -buttons: QVector<GameButton\*> |
| +GameView(QWidget\*)  #keyPressEvent(QKeyEvent\*): void  ***<<slots>>***  *-newGame(): void*  *-\_modelGameWon(): void*  *-\_modelGameOver(): void*  *-\_modelUpdateTime(): void*  *-\_modelFieldChanged(prev: Coordinate, cur: Coordinate, type: GameModel::FieldType): void*  *-small(): void*  *-medium(): void*  *-big(): void* |

# **Osztályszerkezet:**

-GameModel

|  |
| --- |
| *QObject*  GameModel |
| +size: int  +baskets: int  +guards: int  +obstacles: int  +guardstep: QTimer\*  +\_time: QTime  +previous: Coordinate  +player: Coordinate  -gameTable: FieldType\*\*  -step: int  -isOver: bool  -\_baskets: QVector<Coordinate>  -\_guards: QVector<Coordinate>  -steps: QVector<int> |
| +GameModel()  +getField(x: int, y: int): FieldType  +newGame(): void  +getBaskets(): void  +stepPlayer(x: int, y: int): void  +checkWin(): bool  -setGuards(): void  -setObstacles(): void  -setBaskets(): void  -setElements(): void  -verticalGuard(index: int): void  -horizontalGuard(index: int): void  ***<<slots>>***  +stepGuard(): void  +pauseOrPlay(): void  ***<<signals>>***  +gameWon(): void  *+*gameOver(): void  +fieldChanged(prev: Coordinate, cur: Coordinate, type: Fieldtype): void  +updateTime(): void |

# **Eseményvezérlés:**

* smallButton.Clicked: Játékméret váltás és új játék kezdése

Forrás: smallButton

Feladat: új játék indítása kis méretű játéktérrel

* mediumButton.Clicked: Játékméret váltás és új játék kezdése

Forrás: mediumButton

Feladat: új játék indítása közepes méretű játéktérrel

* bigButton.Clicked: Játékméret váltás és új játék kezdése

Forrás: bigButton

Feladat: új játék indítása nagy méretű játéktérrel

* pauseButton.Clicked: Játék szüneteltetése illetve elindítása

Forrás: pauseButton

Feladat: a timer megállítása illetve elindítása

* \_modelFieldChanged: Mező megváltoztatása

Forrás: a játék modellje

Feladat: Régi koordinátán lévő mező és új mező megváltoztatása

* \_modelGameWon: Győzelem kiiratása

Forrás: játék modellje

Feladat: QMessageBox felugrása, timer megállítása

* \_modelGameOver: Vereség kiiratása

Forrás: játék modellje

Feladat: QMessageBox felugrása, timer megállítása

* \_modelUpdateTime: Label frissítése

Forrás: játék modellje

Feladat: Label frissítése az eltelt játékidővel

# **Végfelhasználói tesztesetek:**

|  |  |  |
| --- | --- | --- |
| Sorszám | Teszteset | Elvárt hatás |
| 1. | Alkalmazás indítása | Megjelenik a menü, a várt gombokkal. |
| 2. | Játék létrehozása | Létrejön a játéktábla a megadott paraméterekkel, Maci Laci a bal felső sarokban helyezkedik el. |
| 3. | Kilépés folyó játékból | Az alkalmazás leáll. |
| 4. | Kilépés szüneteltetett játékból | Az alkalmazás leáll. |
| 5. | Piknikkosár felvétele | A piknikkosarak száma nő. |
| 6. | Kilépés a pályáról | Maci Laci visszaugrik az előző mezőre. |
| 7. | Szüneteltetés | Folyamatban lévő játék megszakítható: a megszerzett kosarak száma és az eltelt idő továbbra is látható, de az őr nem lép tovább másodpercenként. |
| 8. | Folytatás | Szüneteltetett játék folytatása. |
| 9. | Méretváltoztatás következménye | Szüneteltetett illetve folyó játék esetén is újra rajzolódik a pálya, új helyre kerülnek a mező különféle fajtái, valamint Maci Laci is jó helyre kerül.  Működik kisebbről nagyobbra és nagyobbról kisebbre is. |