

A PROGRAMOZÁS ALAPJAI 2.

HÁZI FELADAT DOKUMENTÁCIÓ

SUDOKU JÁTÉK

KÉSZÍTETTE: FEKETE ÁKOS, N6Z4HP
akosroland25@gmail.com

KÉSZÍTÉS FÉLÉVE: 2018/19/2

TARTALOMJEGYZÉK

Felhasználói dokumentáció.....	3
Osztályok statikus leírása.....	4
Class_Sudoku.....	4
Felelőssége.....	4
Attribútumok.....	4
Metódusok.....	4
UML osztálydiagramm.....	5
Összegzés.....	6
Mit sikerült és mit nem sikerült megvalósítani a specifikációból?.....	6
Mit tanultál a megvalósítás során?.....	6
Továbbfejlesztési lehetőségek.....	6
Képernyőképek a futó alkalmazásról.....	7

Felhasználói dokumentáció

Miután elindult a program egyből egy menüben találjuk magunkat, ahol három választási lehetőségünk van. Itt tudjuk kiválasztani azt a nehézségi szintet amin szeretnénk játszani, ezek a következők: Easy – Könnyű, Medium – Közepes, Hard – Nehéz. Miután ez megtörtént egyből egy Sudoku rejtvénnel találjuk magunkat szemben. A pályától balra található két gomb, a felső körbe mutató nyíllal új pályát kezdhünk a nulláról az elején kiválasztott nehézségi szinten, míg az alsóval kiléphetünk a programból, ezeket kattintásra lehet aktiválni. A kilépés még az ESC billentyű lenyomásával is elérhető. A pályán az üres helyekre úgy tudjuk az általunk helyesnek vélt számot felvinni, ha először bal-klikkel kiválasztjuk a kívánt mezőt, majd beütjük a kívánt számot. Ilyenkor ez a szám kékes-zölden megjelenik a pályán, függetlenül annak helyességétől. Ha úgy véljük, hogy az általunk felvitt szám rossz, azt egy jobb-klikkel törölni tudjuk. Ha minden üresen hagyott mezőt kitöltünk, két dolog következhet:

1. A megoldás helyes, ilyenkor kékre vált a pálya rácsa, illetve az eddig felvitt kékes-zöld számok feketére váltanak. Ilyenkor lehetőség van új pályának kezdésére, illetve a játék abba hagyására.
2. A megoldás helytelen, ilyenkor a pálya rácsa pirosra vált, majd arra vár a program, hogy az eddigi számokból töröljünk ki, és oldjuk meg a rejtvényt. Ha ilyenkor kitörlünk egy számot, a pályának a rácsa visszaáll az eredeti fekete színbe, és a játék folytatódik tovább.

Ezt minden egyes alkalommal meg fogja tenni, amikor elfogyott a felhasználó számára kitöltendő mező, és csak ez a kettő kimenetele lehet ilyenkor. Természetesen a két baloldalsó gombot bármikor használhatjuk, ilyenkor is. Eközben mindenről tájékoztat minket a konzol, ami a játék indításával ugyanúgy elindul. Itt a játék helyzetéről kapunk információkat, angolul, ilyen információk a következők:

- Melyik mezőt sikerült kiválasztani – (oszlop, sor)
- Melyik mezőn sikerült elvégezni a törlést (oszlop, sor) //A pálya alap tagjait nem lehet törölni!
- Melyik nehézséget választottuk
- Mi volt az érvényes bemenet,
- Sikerült-e a pályát helyesen legenerálni, ha igen, ezt hány milliszekundum alatt sikerült
- Minden egyes számbeírásnál/törlésnél, az üres helyek számát
- Nyertünk-e?
- Ha vesztettünk kiírja hogy töröljünk le egy már beírt számot

Osztályok statikus leírása

Class_Sudoku

Felelőssége

A Sudoku pályát tartalmazza, a Sudokuban fellelhető üres helyek számát, illetve a játék futásához szükséges állapotokat, STATE-ket is itt tartja számon. Ezeknek a beállításához szükséges függvények is itt találhatóak. Lehet egy adott elemet a pályáról lekérni, beállítani, sorokat/oszlopokat megcserélni, egész pályát lemásolni, vagy egy egész példányt.

Attribútumok

Privát

`char board[COL][ROW];`

Ez maga a pálya, ez azért érdemes char-ba tárolnunk, mivel a felhasználó is a billentyűzetéről char típusú elemeket ad be. Ez igazából egy 2D-s szöveget tartalmazó mátrix.

`int state;`

A játékhoz szükséges állapotot tartalmazza.

`int start_null;`

A pályán lévő nullák/üres helyeket tartja számon.

Metódusok

Publikus

`Class_Sudoku();`

A Sudoku alapját hozza létre

`Class_Sudoku(const char * file);`

A Sudoku alapját hozza létre egy szöveges fájlból

`~Class_Sudoku();`

Destruktor, kilépésnél automatikusan meghívódik és törlni az osztálypéldányt

`void copyClass(const Class_Sudoku);`

Lemásolja a beadott Sudoku-t az adott osztálypéldányba

`void copyClass(const Class_Sudoku*);`

Lemásolja az adott pointerként beadott Sudoku-t az az adott osztálypéldányba

`void setDefault();`

Beállít minden kezdetlegi értéket 0-ra, vagy pályánál üres helyre

`void setBoard(const char*);`

Feltölti a Sudoku pályáját egy szöveges fájlból

`void setBoardElement(int,int,char);`

Egy adott elemét állítja be a Sudoku pályának

`void setState(int STATE);`

Beállítja a játék állapotát

`void setNull(int);`

Egy adott értékre állítja be a nullák/üres helyek számát

`void setNullPlus();`

Eggyel növeli nullák/üres helyek számát

`void setNullMinus();`

Eggyel nullák/üres helyek számát

`void copyBoard(const Class_Sudoku);`

Lemásolja a beadott Sudoku pályáját osztálypéldány pályájába

`void copyBoard(const Class_Sudoku*);`

Lemásolja a pointerként beadott Sudoku pályáját osztálypéldány pályájába

`char getBoardElement(int,int) const;`

Megadja az adott helyen lévő elemet a pályáról

`int getState() const;`

Megadja a játék állapotát

`int getNull() const;`

Megadja a nullák/üres helyek számát

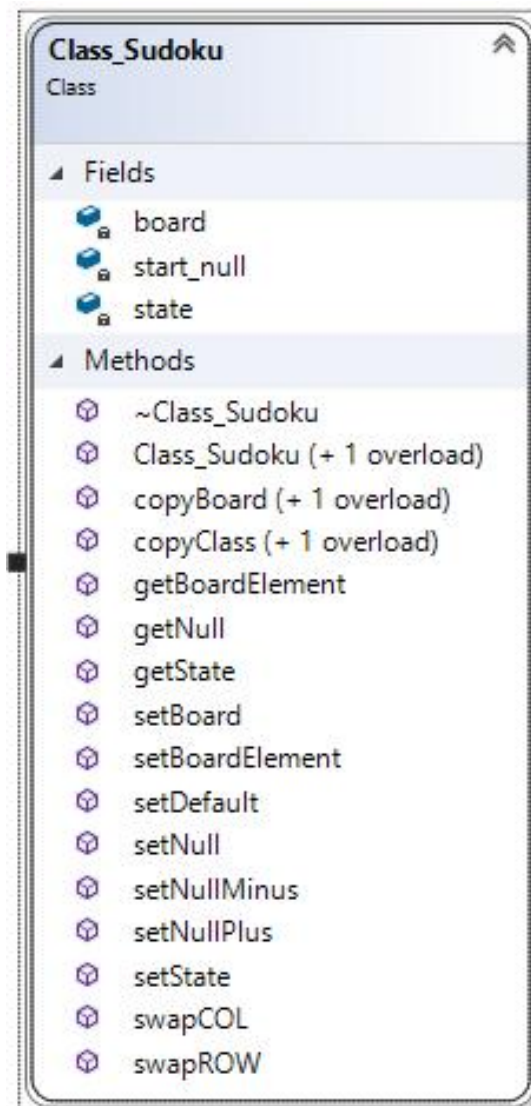
`void swapROW(int, int);`

Megcserél kettő megadott sort

`void swapCOL(int, int);`

Megcserél kettő megadott oszlopot

UML osztálydiagramm



Összegzés

Mit sikerült és mit nem sikerült megvalósítani a specifikációból?

Mindent sikerült megvalósítanom, amit a specifikációba megírtam.

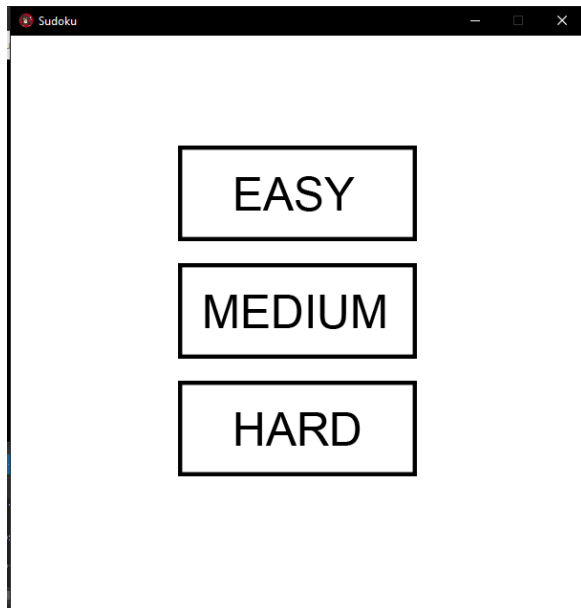
Mit tanultál a megvalósítás során?

Nehéz volt a pályagenerálás és maga az egész SDL2 is teljesen új volt. Nagy nehézség volt különösen az SDL2-nek utána járni, mivel magyarul nem találtam jól magyarázó leírásokat, csak angolul, illetve egy hatékony algoritmust találni arra, hogy a pályát randomra generáljam.

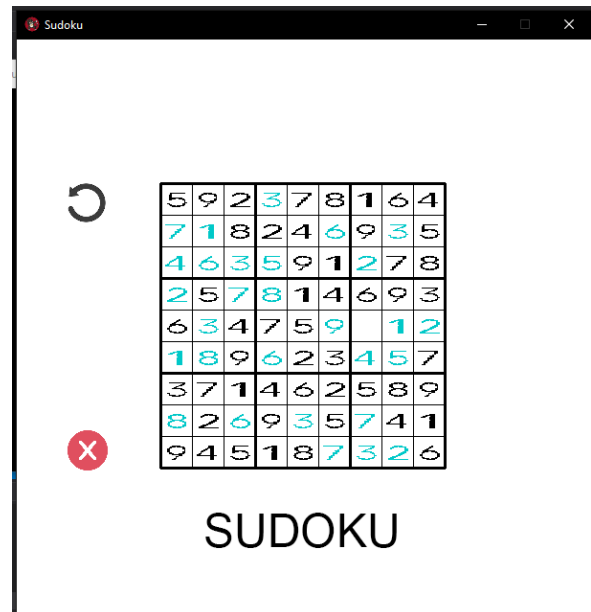
Továbbfejlesztési lehetőségek

Lehetne egy plusz gomb, ami leellenőrzi a számokat, és azokat átválthatná pirosra, jelezve hogy mit nem töltöttünk ki jól, illetve lehetne egy olyan új funkció, ami az biztosítaná, hogy egy adott mezőhöz a felhasználó az arra a mezőre tippelt számait felvihesse, a mező egyik felső sarkába, így segítve ezzel a játéknak az átláthatóságát.

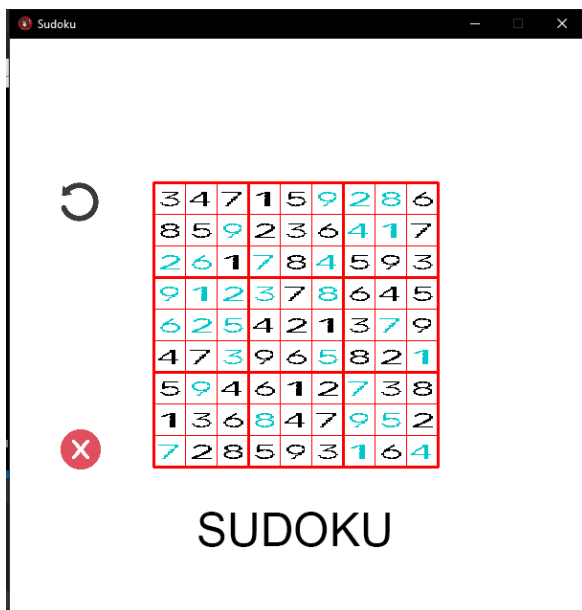
Képernyőképek a futó alkalmazásról



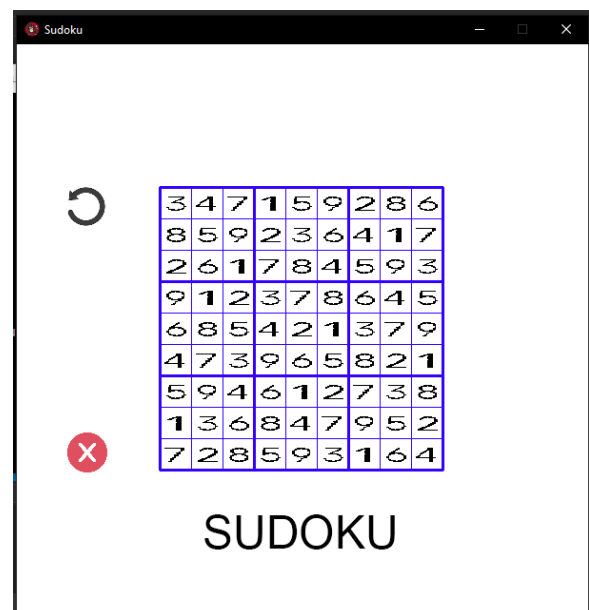
1. ábra Nehézségválasztás menü



2. ábra Játék közben



3. ábra Ha rossz a megoldás



4. ábra Nyert a játékos