**Beadandó feladat dokumentáció**

**Készítette:**

Németh Dominik Kristóf

E-mail: [nemethdominik02@gmail.com](mailto:nemethdominik02@gmail.com)

**Feladat:**

Készítsünk programot, amellyel a következő két személyes játékot játszhatjuk. Adott egy n × n mezőből álló tábla, amelyre a játékosok 2 × 1-es méretű téglalapokat helyezhetnek el (vízszintesen, vagy függőlegesen). A játékosok felváltva léphetnek. A játék célja, hogy a téglalapokkal elhatároljuk a tábla egy részét (teljesen körbevéve téglalapokkal), amelyben így minden mező a játékosé lesz (beleértve az ellenfél által korábban elfoglalt mezőket is). A program külön jelölje meg a lehelyezett téglalapokat, illetve az elfoglalt területeket, és játék közben folyamatosan jelenítse meg az elfoglalt terület méretét játékosonként. A játék akkor ér véget, ha már további lépés nem végezhető a pályán. A program biztosítson lehetőséget új játék kezdésére a pályaméret megadásával (6 × 6, 8 × 8, 10 × 10), valamint játék mentésére és betöltésére. Ismerje fel, ha vége a játéknak, és jelenítse meg, melyik játékos győzött. Tipp: a bekerítés ellenőrzéséhez használható például az elárasztásos kitöltés (flood fill) algoritmusa.

**Elemzés:**

* A játékot három táblamérettel játszhatjuk: kicsi (6x6), közepes (8x8), nagy(10x10). A program indításakor közepes táblaméretet állít be, és automatikusan új játékot indít.
* A feladatot egyablakos asztali alkalmazásként Windows Forms grafikus felülettel valósítjuk meg.
* Az ablakban elhelyezünk egy menüt a következő menüpontokkal: File (Új játék, Játék betöltése, Játék mentése, Kilépés), Beállítások (6x6, 8x8, 10x10). Az ablak alján megjelenítünk egy státuszsort, amely a kék illetve a piros játékos által elfoglalt mezők számát jelzi.
* A játéktáblát egy nyomógombokból álló rács reprezentálja. A nyomógomb egérkattintás hatására kiszínezzük, újboli kattintás hatására, a kattintások és a környezet lehetőségeitől függően egy szomszédos gombot is kiszínezhetünk. a Space gomb megnyomásával véglegesíthetjük a döntésünk.
* A játék automatikusan feldob egy dialógusablakot, amikor vége a játéknak. (Nem végezhető több lépés). Szintén dialógusablakokkal végezzük el a mentést, illetve betöltést.

**Felhasználói estek diagramja**

A képen diagram, szöveg, kör, képernyőkép látható

Automatikusan generált leírás

**Tervezés:**

* Programszerkezet:
  + A programot háromrétegű architektúrában valósíjuk meg. A megjelenítés a **View**, a modell a **Model**, míg a perzisztencia a **Persistence** névtérben helyezkedik el. A program csomagszerkezete a 2. ábrán látható.
  + A program szerkezetét két projektre osztjuk implementációs megfontolásból: a **Persistence** és **Model** csomagok a program felületfüggetlen projektjében, míg a **View** csomag a Windows Formstól függő projektjében kap helyet.
* Perzisztencia:
  + Az adatkezelés feladata a táblával kapcsolatos információk tárolása, valamint a betöltés/mentés biztosítása.
  + A **BekeritesTable** egy érvényes játéktáblát biztosít (azaz mindig ellenőrzi a beállított értékeket), ahol minden mezőre ismert az értéke (**Player**). A tábla alapértelmezés szerint 8x8-as, de ez GameMode választásával állítható. A tábla lehetőséget ad az állapotok lekérdezésére (**IsFilled**, **GetValue**, **Size**, **CurrentPlayer**), valamint szabályos léptetésre (**StepValue**), illetve direkt beállítás (**SetValue**) elvégzésére.
  + A hosszú távú adattárolás lehetőségeit az **IBekeritesDataAccess** interfész adja meg, amely lehetőséget ad a tábla betöltésére (LoadAsync), valamint mentésére (SaveAsync). A műveleteket hatékonysági okokból aszinkron módon valósítjuk meg.
  + Az interfészt szöveges fájl alapú adatkezelésre a **BekeritesFileDataAccess** osztály valósítja meg. A fájlkezelés során fellépő hibákat a **BekeritesDataException** kivétel jelzi.
  + A program az adatokat szöveges fájlként tudja eltárolni, melyek az stl kiterjesztést kapják. Ezeket az adatokat a programban bármikor be lehet tölteni, illetve ki lehet menteni az aktuális állást
  + A fájl első sora megadja a tábla méretét, valamint a jelenlegi játékost (**CurrentPlayer**) (ami alapértelmezés szerint 8x8 és piros játékos. A fájl többi része izomorf leképezése a játéktáblának, amely soronkra és oszlopokra bontva számokat tartalmaz szóközökkel választva. A számok 0-3 közöttiek lehetnek, ahol 0 reprezentálja a NoPlayert, 1-2 pedig a piros és a kék játékos.
  + A képen szöveg, képernyőkép, Betűtípus, diagram látható

    Automatikusan generált leírás
* Modell:
  + A modell lényegi részét a **BekeritesGameModel** osztály valósítja meg, amely szabályozza a tábla tevékenységeit. A típus lehetőséget ad új játék kezdésére (NewGame), valamint lépésre (StepGame, DealWithSurrounded, SecondBlock).
  + A mezők állapotváltozásáról a **RotatedSecondBlock** esemény tájékoztat. Az esemény argumentuma (**FieldChangedEventArgs**) tárolja megváltozott mező pozícióját.
  + A játékállapot megváltozásáról (elfoglalt terület) a **GameAdvanced** esemény, míg a játék végéről döntetlen esetében a **GameOver** ha van nyertes akkor a **GameWon** esemény tájékoztat. Az események argumentuma (**GameWonEventArgs**) tárolja a győztest.
  + A modell példányosításakor megkapja az adatkezelés felületét, amelynek segítségével lehetőséget ad betöltésre (**LoadGameAsync**) és mentésre (**SaveGameAsync**)
  + A játékmódot a **GameMode-on** át kezeljük, a játékosokat pedig a **Player** felsorolási típuson át kezeljük.
* Nézet:
  + A nézetet a **GameForm** osztály biztosítja, amely tárolja a modell egy példányát (\_**model**), valamint az adatelérés konkrét példányát (\_**dataAccess**).
  + A játéktáblát egy dinamikusan létrehozott gombmező (\_**buttonGrid**) reprezentálja. A felületen létrehozzuk a megfelelő menüpontokat, illetve státuszsort, valamint dialógusablakokat, és a hozzájuk tartozó eseménykezelőket. A játéktábla generálását (**GenerateTable**), illetve az értékek beállítását (**SetupTable**) külön metódusok végzik.
  + A képen szöveg, képernyőkép, Betűtípus, Párhuzamos látható

    Automatikusan generált leírás
* **Tesztelés:**
  + A modell funcionalítása egységtesztek segítségével lett ellenőrizve a **BekeritesGameModelTest** osztályban.
  + Az alábbi tesztesetek kerültek megvalósításra:
    - **BekeritesGameModelNewGameSixTest,**
    - **BekeritesGameModelNewGameEightTest**,
    - **BekeritesGameModelNewGameTenTest**: Új játék indítása, adott méretű táblával.
* **BekeritesGameModelStepTest:** Játékbeli lépés hatásainak ellenőrzése, lépés végrehajtás, mező foglalás.
* **BekeritesGameModelLoadTest:** A játék modell betöltésének tesztelése mockolt perzisztencia réteggel.