

## Programozás 3 nagy házi feladat

### Sokoban

#### 1. Az alkalmazás leírása:

A Sokoban egy logikai játék, ahol a játékosnak minél kevesebb lépésből kell a karakterével(csirke) odatolni a végpontra (piros négyzet fekete karikával) a dobozt(úrlényt).

A játék indítása után egy menü jelenik meg ahonnan el lehet kezdeni a játékot (ez esetben 6 pálya közül választhat a játékos a 6. pálya átdolgozható), a menüben még található a 6. pályának a szerkesztésére egy gomb, ami oda irányít át, hogy szerkessze a pályát a felhasználó, illetve a menüben még van egy Dicsőséglista gomb, ahol meg lehet tekinteni, hogy az első 5 pályát ki vitte végig a legkevesebb lépésből pályánként lebontva. A menüben megadható a játékos neve is.

Mindenhonnan vissza lehet lépni a menübe.

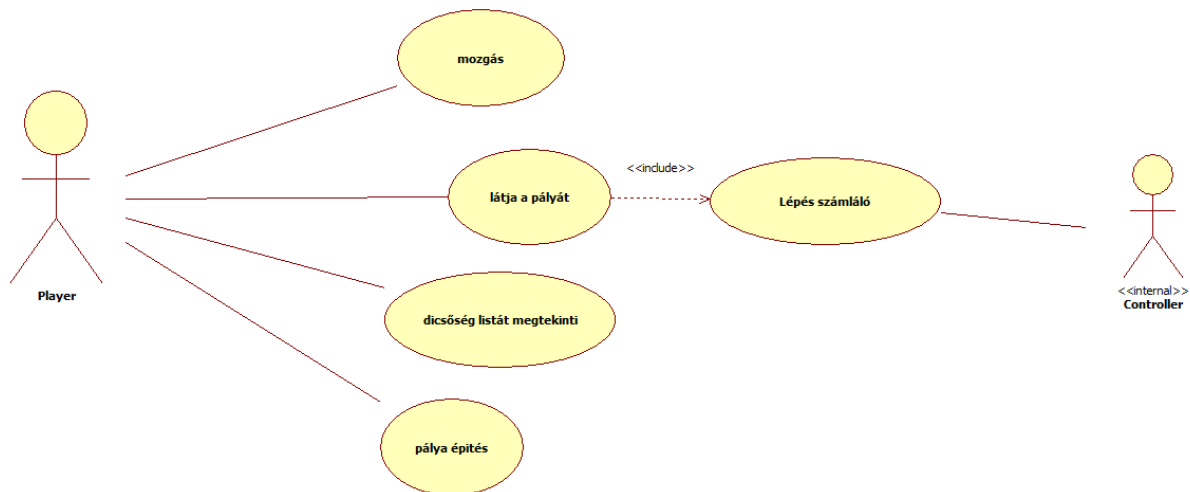
Amikor elindul a játék feljön egy menü, ahol lehet választani (6) pálya közül a játék gomb megnyomása után, vagy megnézni a dicsőséglistát vagy egy új pályát készíteni.

A játék arról szól, hogy van egy karakter(csirke), amit a játékos irányít nyilakkal/"AWSD"-vel és egy zárt pályán belül dobozt (úrlényt) kell tolnia, megadott négyzetekre, ha ezt megteszi akkor nyer/teljesíti a pályát.

A játék közben a játékos lépéseit fogja mutatni egy számláló az adott pályán. Illetve az eddigi rekordot az adott pályán.

A játékos a dobozt magától csak elfele tudja tolni, szóval húzni nem tudja, ebből kifolyóan, ha pl egy doboz bekerül egy sarokba akkor az már ott marad. Lesz egy restart gomb vagy ha megnyomja a játékos az "R" betűt akkor újakezdi a pályát.

#### 2. Use-case-ek



Mozgás: A player a játékon belül mozog vagy mozog és dobozt tol el így.

Látja a pályát: Látja a pályát és a lépés számlálót.

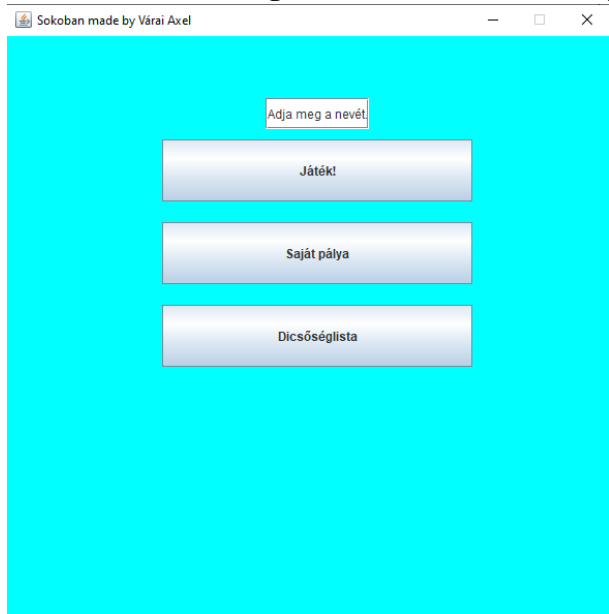
Dicsőség listát megtekint: Megnézi a rekord lépéseket.

Pálya építés: Pályát épít, amit ki tud utána próbálni.

Lépés számláló: számolja a lépéseket egy pályán, illetve beleírja a dicsőséglistába, ha új rekord.

### 3. User Manual

A menüben **3 gomb** található, illetve itt tudja a játékos megadni a nevét is



El tudja indítani a játékot “Játék!”, ahol 6 szint közül választhat itt a 6. pályát szerkeszteni is tudja a “Saját pálya” gombra kattintva a menüben. A menüben még található a Dicsőséglista gomb is, ami az első 5 pályához a rekord, azaz legkevesebb lépésből teljesítő ember nevét írja ki a lépésszámmal együtt.

A játékos irányítása:

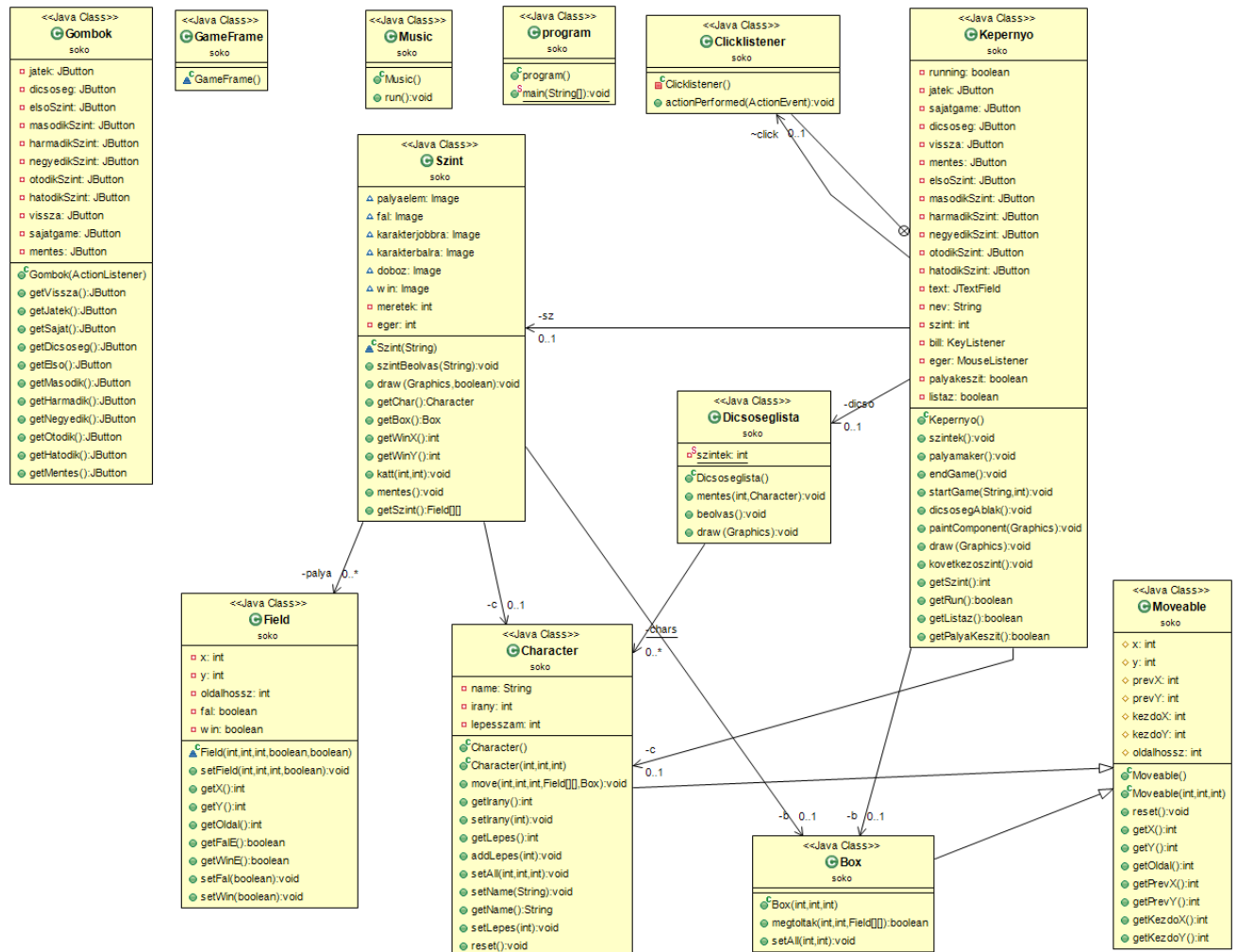
- A jobbra mozog
- W felfelé mozog
- S lefelé mozog
- D jobbra mozog
- R újra kezdő állapotba kerülnek az elemek a pályán.

Pályakészítő módban, ha a pálya mellé kattint az egyik elemre a játékos olyan elemet fog kirajzolni utána a pályára ahová kattint, karakter, doboz és nyertes mezőből csak 1 lehet a pályán. Ezek után a mentés gombra kattintva elmenti a pályát a játék és ezt ki lehet aztán próbálni a többi szintnél.



#### 4. Az osztályok:

##### 4.0.1 Osztály diagram



##### 4.1 Kepernyo

##### Felelősségek

Itt valósul meg a menü és a játék is.

##### Attribútumok

-running :boolean	Ezzel jelzem, hogy játék megy-e (valamelyik szinten van) eszerint rajzol.
-jatek :JButton -sajatgame :JButton -dicsoseg :JButton -vissza :JButton -mentes :JButton -elsoSztint :JButton -masodikSztint :JButton -harmadikSztint :JButton	Ezek a gombok amiket a játék és a menü során használok.

-negyedikSzint :JButton -otodikSzint :JButton -hatodikSzint :JButton -text :JButton	
-nev :String	Ide mentem ki azt a szöveget amit beír a felhasználó, ez lesz a neve.
-szint :int	A pálya számát jelöli.
-bill :KeyListener	A billentyűzet által való karakter irányítás miatt kell.
-eger :MouseListener	Az egér által való irányítás miatt kell a pálya készítő részénél.
-palyakeszit :boolean	Azt jelzi, hogy a pálya készítő részen van-e a felhasználó.
-listaz :boolean	Azt jelzi, hogy a dicsőség lista oldalon van-e a felhasználó

**Metódusok**

+Kepernyo()	Konstruktor.
+szintek()	A szintek képernyője.
+palyamaker()	A pálya készítő rész ablaka.
+ endGame()	Ha vége a játéknak a menüt rajzolja ki újra.
+ startGame(fileszint : String,level :int)	Elindul az egyik pálya level.
+ dicsosegAblak()	Dicsőség listának az ablaka.
+paintComponent(g : Graphics)	Rajzolásért felel.
+ draw(g : Graphics)	Rajzolásért felel.
+ kovetkezoszint()	Következő szintre lépést kezeli.

□ **4.2 ClickListener****Felelősségek**

A gombok irányítására a “ClickListener”-t implementáltam, ezek megnyomására a Kepernyo kirajzolja azokat az oldalakat, ahova irányult a felhasználó.

**Attribútumok**

-

**Metódusok**

+actionPerformed(e : ActionEvent)	A különböző gombok megnyomását kezeli.
-----------------------------------	--

□ **4.3 Szint****Felelősségek**

Beolvassa a szintet txtből, a képeket és átadja a képernyőnek, ő kezeli a pályaszerkesztést is. Egy szinten van egy 10x10es Field tömb, illetve a Character meg a Box

**Attribútumok**

-c : Character	A karakter.
-b :Box	A doboz amit a karakter tol.
-palya :Field[][]	A pálya.
-palyaelem :Image	A játékban felhasznált képek.

-fal :Image -karakterjobbra :Image -karakterbalra :Image -doboz :Image -win:Image	
-meretek :int	Fix 50 a pályán a karakter,doboz, pálya "tileok" oldalainak mérete.
-eger :int	Jelzi, hogy melyik képet fogja kirajzolni a pálya készítő módban.

**Metódusok**

+ Szint(file : String)	Konstruktor.
+ szintBeolvas(file: String)	Beolvassa a szintet egy fájlból.
+ draw(g :Graphics,palyakeszit boolean)	Kirajzolja a pályát
+ katt(x :int, y :int)	A kattintást kezeli a pálya készítő módban.
+ mentes()	Elmenti a készített pályát.

☐ **4.4 Moveable**
**Felelősségek**

A mozgatható osztályok őse.

**Attribútumok**

#x :int	Az x koordináta.
#y :int	Az y koordináta.
#prevX :int	Az előző x koordináta.
#prevY :int	Az előző y koordináta.
#kezdőX :int	A kezdő x koordináta.
#kezdőY :int	A kezdő y koordináta.
#oldalhossz :int	Az objektum mérete.

**Metódusok**

+Moveable(xx :int,yy :int,oldalhossz :int)	Konstruktor
+ reset()	A kezdőpontba visszaállít.

☐ **4.5 Character**
**Felelősségek**

A karakter(csirke) amivel a játékos játszik, a Moveable osztály leszármazottja..

**Attribútumok**

#x :int	Az x koordináta.
#y :int	Az y koordináta.
#prevX :int	Az előző x koordináta.
#prevY :int	Az előző y koordináta.
#kezdőX :int	A kezdő x koordináta.
#kezdőY :int	A kezdő y koordináta.
#oldalhossz :int	Az objektum mérete.

**Metódusok**

+Moveable(xx :int,yy :int,oldalhossz :int)	Konstruktor
+ reset()	A kezdőpontba visszaállít.

□ **4.6 Box****Felelősségek**

A doboz amit a játékos tologat, a Moveable osztály leszármazottja.

**Attribútumok**

-

**Metódusok**

+ Box(xx :int,yy :int,oldalhossz :int)	Konstruktor.
+ megtoltak (xx :int,yy :int,palya :Field[[]]) :boolean	Amikor a dobozt megtolják.
+ setAll(int i, int j)	Beállítja a doboz x,y koordinátáit.

□ **4.7 Field****Felelősségek**

A pálya négyzetei, fal/háttér/cél

**Attribútumok**

-x :int	Az x koordináta.
-y :int	Az y koordináta.
-oldalhossz :int	Egy "tile" hossza.
- fal :boolean	Fal-e a "tile".
-win :boolean	Cél-e a "tile". Ebből csak 1 van egy pályán.

**Metódusok**

+ Field(xx :int,yy :int,oldal :int,falE :boolean,winE :boolean)	Konstruktor.
+ setField(xx :int,yy :int,oldal :int,falE : boolean)	Beállít egy "tile"-t.

□ **4.8 GameFrame****Felelősségek**

GameFrame egy JFrame-t terjeszt ki, itt adom meg a kép méreteit, címét, illetve hozzáadok egy JPanel kiterjesztett osztályt is a Kepernyo-t.

**Attribútumok**

-

**Metódusok**

+ GameFrame()	Konstruktor.
---------------	--------------

#### □ 4.9 Gombok

##### Felelősségek

A gombok inicializálása történik itt, amiket majd használok a Képernyőn.

##### Attribútumok

-

##### Metódusok

-

#### □ 4.10 Music

##### Felelősségek

A zenét kezeli.

##### Attribútumok

-

##### Metódusok

+ run()	Elindítja a zenét.
---------	--------------------

#### □ 4.11 program

##### Felelősségek

Itt indul el a játék, itt indítom el a zenét is ami egyszer játszódik le, illetve itt hozok létre egy új GameFramet ami elindítja a játékot.

##### Attribútumok

-

##### Metódusok

-

#### □ 4.12 Dicsoseglista

##### Felelősségek

A dicsőséglistához a mentést és a beolvasást kezeli.

##### Attribútumok

-chars : ArrayList<Character>	Character tömb.
-szintek :int	A szinteket jelöli, 5, mert az első 5 szinthez számolom a lépéseket amit el is mentek.

##### Metódusok

+ Dicsoseglista()	Konstruktor.
-------------------	--------------



+ mentes(szing :int,c :Character)	Fájlba mentés.
+ beolvas()	Fájlból beolvassa a szerializált dicsőség listát ha még nincs ilyen akkor generál egyet értékekkel.
+ draw(g :Graphics)	Rajzolás.

**További dokumentáció JavaDoc-ként a mappában.**

#### **5. Fájlok szerkezete:**

A pályáimat 6db txt fájlban tárolom és ezeket olvasom be, mindegyik mezőt egy karakter jelöli 'h' a hátteret, 'f' a falat, 'b' a boxot(úrlény), 'c' a karaktert, 'w' a nyertes mezőt.

A dicsőséglistát szerializálva tárolom.