# Részletes tervek

## Osztályok és metódusok tervei.

### Celhely

#### Felelősség

A játékban el van helyezve a jelölt dobozoknak megfelelő számú célhely. Ez az osztály reprezentálja a célhelyeket. Ezek a célhelyek egy jellel vannak ellátva, hogy meg tudjuk különböztetni a többi mezőtől. Ha egy jelölt dobozt sikerül erre a mezőre tolni, akkor a célhely elveszti a megkülönböztetésre szolgáló jelzést, és többé nem kaphat játékos pontot, ha erre a mezőre egy másik jelölt dobozt tolna.

#### Ősosztályok

Uresmezo → Celhely

#### Attribútumok

* **-bool jel**: Ha a mezőre jelölt láda kerül, elveszti a jelölését. Ez az attribútum ezt jelöli.

#### Metódusok

* **bool Fogad(float ero, Irany i, JeloltDoboz jd, Jatekos j)**: Kezeli azt az esetet, amikor valaki jelölt dobozt tol a célhelyre. A lépés sikerességétől függően bool értékkel tér vissza és siker esetén pontot ad az adott játékosnak, aki rátolta a dobozt.
* **bool Fogad(float ero, Irany i, Jatekos aktj, Jatekos j)**: Játékos lép a mezőre. Ha sikeres a lépés, igaz értékkel tér vissza.
* **bool Fogad(float ero, Irany i, Doboz d):** Kezeli azt az esetet, amikor valaki dobozt tol a célhelyre. A lépés sikerességétől függően bool értékkel tér vissza

### Csapdaajto

#### Felelősség

A pályákon csapdaajtók vannak elhelyezve. Ez az osztály reprezentálja ezeket a mezőket. Két állapota lehet –nyitott, csukott. Nyitott állapotban Lyuk-ként viselkedik - a rá kerülő objektumok „beleesnek” a lyukba, és eltűnnek a pályáról. Zárt állapotban az objektumok rákerülnek a mezőre, mintha egy normális mezőre lépnének.

#### Ősosztályok

Uresmezo → Csapdaajto

#### Attribútumok

* **-bool nyitva**: A csapdaajtó állapota, ami megmondja, hogy nyitva van-e vagy csukva.

#### Metódusok

* **Nyit()**: A jel értékét igazra állítja, amivel az ajtót nyitott állapotba teszi.
* **Csuk()**: A jel értékét hamisra állítja, amivel az ajtót csukott állapotba teszi.
* **bool Fogad(float ero, Irany i, Jatekos aktj, Jatekos j)**: Ha a csapdaajtó állapota nyitott, akkor a rákerülő játékos meghal (és igazzal tér vissza), ha az állapota csukott, akkor úgy viselkedik, mint az ősosztálya.
* **bool Fogad(float ero, Irany i, Doboz d, Jatekos j)**: Ha a csapdaajtó állapota nyitott, akkor a rákerülő doboz eltűnik (és igazzal tér vissza), ha az állapota csukott, akkor úgy viselkedik, mint az ősosztálya.
* **bool Fogad(float ero, Irany i, JeloltDoboz jd, Jatekos j)**: Ha a csapdaajtó állapota nyitott, akkor a rákerülő jelölt doboz eltűnik (és igazzal tér vissza), ha az állapota csukott, akkor úgy viselkedik, mint az ősosztálya.

### Doboz

#### Felelősség

A játékban számos Doboz található. Ezeket kell a játékosoknak eltolniuk annak érdekében, hogy elérjék céljaikat. Egy doboz eltolásánál meg kell vizsgálnunk, hogy hova akarjuk tolni a dobozt, lehet-e az adott mezőre tolni, vagy sem. Ha egy másik léptethető objektum van a tolási irányban levő mezőn, akkor rekurzívan megvizsgáljuk a célmező mellett levő mező(ke)t, amíg meg nem tudjuk, hogy elvégezhető-e a mozgatás. Minden doboznak van egy „tapadas” értéke, melynek segítségével kiszámíthatjuk, hogy a Doboz tolható-e. Ha egy dobozt lyukba tolunk, az osztály felelőssége lekezelni a Doboz eltűntetését a pályáról. Az osztály lekérdezhető, hogy a Doboz tud-e még mozogni, vagy sem, ez a játék végének ellenőrzésére ad lehetőséget.

#### Ősosztályok

Doboz → Leptetheto

#### Attribútumok

* **#float tapadas**: A doboz tapadásának értéke a mezőhöz.

#### Metódusok

* **bool Vege()**: Ellenőrzi, hogy a doboz képes-e még mozogni, vagy már nem tudjuk eltolni. Ennek megfelelően tér vissza egy bool értékkel.
* **Halal()**: Eltünteti a dobozt a játékból.
* **bool Utkozik(float ero, float surl, Irany i, Jatekos j):**Ha egy másik doboznak tolunk egy dobozt, ellenőrizzük, hogy a kapott erő és a mező súrlódása alapján el tudjuk-e mozgatni, vagy sem.
* **bool Tol(float ero, float surl, Irany i, Jatekos j):** Ez a függvény hívódik meg, ha egy játékos tolja meg dobozt.
* **int getTipus():** A hálózat működéséhez kell, visszaadja az elem típusát, egy 2-est.

### Fal

#### Felelősség

A pályánk egy mezője. A Fal egy akadály a pályán, ami meggátolja az adott mezőre egy léptethető objektum rákerülését.

#### Ősosztályok

Uresmezo → Fal

#### Metódusok

* **bool Fogad(float ero, Irany i, Jatekos aktj, Jatekos j)**: Az UresMezo Fogad metódusát írja felül. Elutasítja az érkeztetési kérést (hamissal tér vissza).
* **bool Fogad(float ero, Irany i, Doboz d, Jatekos j)**: Az UresMezo Fogad metódusát írja felül. Elutasítja az érkeztetési kérést (hamissal tér vissza).
* **bool Fogad(float ero, Irany i, JeloltDoboz jd, Jatekos j)**: Az UresMezo Fogad metódusát írja felül. Elutasítja az érkeztetési kérést (hamissal tér vissza).

### Irany <<enumeration>>

#### Felelősség

Ez egy Enum osztály, amely a lehetséges mozgási irányokat tárolja. Ennek felhasználásával tudjuk az objektumainkat léptetni a pályán, és lekérdezni egy adott mező szomszédjait.

#### Attribútumok

* **Fel**
* **Le**
* **Jobbra**
* **Balra**
* **Mez**
* **Olaj**

### Jatek

#### Felelősség

Ez az osztály felelős egy játék elindításáért, és befejezéséért.

#### Attribútumok

* **-KliensOlvaso olvaso:** Referencia a KliensOlvasóra.
* -**Szerver szerver:** Referencia a szerverre.

#### Metódusok

* **Startgame()**: Elindítja a játékot
* **Endgame()**: Befejezi a játékot

### Jatekos

#### Felelősség

A játékosokat reprezentáló osztály. Itt tudjuk nyilván tartani, hogy kik a játékosok, és hogy mekkora erővel rendelkeznek. Az erő szükséges ahhoz, hogy a dobozokat el tudjuk tolni. Az osztály felelős a játékosok léptetéséért is. Ha a játékos a megfelelő helyre tol egy jelölt dobozt, akkor itt tudunk a játékosnak pontot adni. Továbbá ez az osztály kezeli le azokat az eseteket is, amikor megtolódik egy másik léptethető objektum által.

#### Ősosztályok

Leptetheto → Jatekos

#### Attribútumok

* **-string nev:** A felhasználó azonosítója, aki a játékost irányítja.
* **-float ero:** A játékos ereje, amivel tolni képes.
* **-int status**: A játékos státusza.
* **-static int jatekosSzam**: Megadja az aktuális játékosok számát.

#### Metódusok

* **Lepes(Irany i):** A pálya hívja meg, hogy léptesse a játékost, ami meghívja a mező Fogad függvényét.
* **PontotKap():** A jelölt ládák célhelyre tolásakor a játékos pontot kap. Ez a függvény felelős érte.
* **Halal():** Megöli a játékost.
* **bool Utkozik(float ero, float surl, irany i, Jatekos j)**: A Leptethető interfész-béli függvény felüldefiniálása, az az eset, amikor egy doboz megtol egy játékost. Ilyenkor, ha a játékos mögött nincs már üres mező, a játékos nem továbbtolódik, hanem meghal. Az „ero” az az erő, amivel a sort megtoló játékos tolt, „surl” pedig a mező súrlódása.
* **bool Tol(float ero, float surl, Irany i, Jatekos j):** Ha a játékost játékos tolja meg, akkor ez a függvény hívódik meg. A játékos referencia ahhoz kell, amikor egy sor megtolt doboz végén egy jelölt doboz pontot akar adni a játékosnak, tehát a lépést kezdeményező játékost jelöli. Az „ero” az az erő, amivel a sort megtoló játékos tolt, „surl” pedig a mező súrlódása.
* **string getNev():** Visszaadja a játékos nevét.
* **int getTipus():** Visszaadja a játékos típusát.
* **int getStatusz():** Visszaadja a játékos státuszát.

### JeloltDoboz

#### Felelősség

A játékban több JeloltDoboz is található. Ezeket kell a játékosoknak eljuttatni a célmezőkre. A jelölt dobozoknak van egy megkülönbözetésre szolgáló jele. Működése hasonló a Doboz osztályéhoz, tehát eltolásnál meg kell vizsgálnunk, hogy hova akarjuk tolni a dobozt, lehet-e az adott mezőre tolni, vagy sem. Ha egy másik léptethető objektum van a tolási irányban levő mezőn, akkor rekurzívan megvizsgáljuk a célmező mellett levő mező(ke)t, amíg meg nem tudjuk, hogy elvégezhető-e a mozgatás. Minden jelölt doboznak van egy „tapadas” értéke, melynek segítségével kiszámíthatjuk a JeloltDoboz tolható-e. Ha egy jelölt dobozt lyukba tolunk, eltűnik a pályáról. Ha egy jelölt doboz egy célmezőre jut, akkor elveszti a jelölését.

#### Ősosztályok

* Leptetheto → Doboz → JeloltDoboz

#### Attribútumok

* **-bool jel:** A doboz jelölése. Ha a célmezőre jelölt láda kerül, a JeloltDoboz elveszti a jelölését.

#### Metódusok

* **JeletTorol():** Törli a jelölést a dobozról.
* **bool Utkozik(float ero, float surl, Irany i, Jatekos j)**: Akkor hívódik ez a függvény, ha a dobozunkat megtolja egy másik doboz. Paraméterként meg van adva a lépést kezdeményező játékos és a tolás iránya. Az „ero” az az erő, amivel a sort megtoló játékos tolt, „surl” pedig a mező súrlódása.
* **bool Tol(float ero, float surl, Irany i, Jatekos j)**: Akkor hívódik ez a függvény, ha a dobozunkat megtolja egy másik doboz. Paraméterként meg van adva a lépést kezdeményező játékos és a tolás iránya. Az „ero” az az erő, amivel a sort megtoló játékos tolt, „surl” pedig a mező súrlódása.
* **bool getJel()**: Visszaadja a jelölést.
* **int getTipus():** Visszaadja a jelölt doboz típusát, a 3-at.
* **int getStatusz():** Visszaadja a jelölt doboz státuszát a jel értékétől függően (ha a „jel” igaz, akkor 1-el, egyébként 0-val tér vissza).

### Kapcsolo

#### Felelősség

A pálya Kapcsolo típusú mezőjét kezeli, figyeli a Kapcsolo-ra kerülő objektumokat és végrehajtja az ennek következményeként kialakuló szükséges módosításokat. Ha egy játékos egy ládát tol a kapcsolóra, akkor az kinyitja a hozzákapcsolt csapdaajtókat, ha ő lép a kapcsolóra, akkor bezárja.

#### Ősosztályok

Uresmezo → Kapcsolo

#### Attribútumok

* **Csapdaajto[ ] csapdak:** Azoknak a Csapdaajto-knak a listája, amiket az adott kapcsolóval átállíthatunk**.**

#### Metódusok

* **bool Fogad(float ero, Irany i, Jatekos aktj, Jatekos j)**: Ha játékos lép rá akkor becsukja a csapdaajtókat.
* **bool Fogad(float ero, Irany i, Doboz d, Jatekos j)**: Ha a csapdaajtó állapota nyitott, akkor a rákerülő doboz meghal (és igazzal tér vissza), ha az állapota csukott, akkor úgy viselkedik, mint az ősosztálya.
* **bool Fogad(float ero, Irany i, JeloltDoboz jd, Jatekos j)**: Ha a csapdaajtó állapota nyitott, akkor a rákerülő doboz meghal (és igazzal tér vissza), ha az állapota csukott, akkor úgy viselkedik, mint az ősosztálya.
* **addCsapda(Csapdaajto c):** Hozzáadja a paraméterként megadott csapdaajtót a kapcsoló által kapcsolható csapdákhoz.

### Leptetheto <<abstract>>

#### Felelősség

Abstract osztály, ami a léptethető objektumokat kezeli. Segítségével a játékosokat és a dobozokat is képesek vagyunk a mezők azonos attribútumában tárolni.

#### Attribútumok

* **#UresMezo mezo:** Az aktuális mezőt tárolja, amin az objektum van.
* **#Palya palya:** Referencia a pályára.

#### Metódusok

* **bool Utkozik(float ero, float surl, Irany i, Jatekos j):** Ha egy mozgatható objektumot doboz tol meg, akkor ez hívódik meg. A játékos referencia ahhoz kell, amikor egy sor megtolt doboz végén egy jelölt doboz pontot akar adni a játékosnak.
* **bool Tol(float ero, float surl, irany i, Jatekos j):** Ha egy mozgatható objektumot játékos tol meg, akkor ez a függvény hívódik meg. A játékos referencia ahhoz kell, amikor egy sor megtolt doboz végén egy jelölt doboz pontot akar adni a játékosnak.
* **Halal():** Absztrakt függvény a leszármazottak halálához.
* **int getTipus():** Visszaadja a Léptethető típusát (alapesetben 0-t).
* **int getStatusz():** Visszaadja a Léptethető státuszát (alapesetben 0-t).

### Lyuk

#### Felelősség

A pálya egy mezője. Ha egy léptethető objektum kerül rá, az „meghal”, és eltűnik a pályáról. Ennek megfelelően meghívja az adott objektum megfelelő függvényét.

#### Ősosztályok

Uresmezo → Lyuk

#### Metódusok

* **bool Fogad(float ero, Irany i, Jatekos aktj, Jatekos j)**: Meghívja a játékos halál függvényét, és igazzal tér vissza.
* **bool Fogad(float ero, Irany i, Doboz d, Jatekos j)**: Meghívja a doboz halál függvényét, és igazzal tér vissza.
* **bool Fogad(float ero, Irany i, JeloltDoboz jd, Jatekos j)**: Meghívja a jelölt doboz halál függvényét, és igazzal tér vissza.

### Palya

#### Felelősség

Ez a játéktér. Tárolja a játék elemeit. Összeköti a játék különálló részeit, és előállítja a kirajzoláshoz szükséges adatokat. Egy játékos lépését ez az osztály kezdeményezi, és ha egy objektum „meghal”, akkor törli őket a pályáról. Pontszerzés esetén jóváírja a megfelelő játékosnak. Lépések után ellenőrzi, hogy van-e a pályán tolható doboz, ha nincs, akkor befejezi a játékot.

#### Attribútumok

* **-Uresmezo[ ] mezok:** Eltárolja a pályán lévő mezőket egy egydimenziós tömbben. tömbben.
* **-KliensAdat jatekadat**: Referencia a KliensAdatra.
* **-Szerver szerver:** Referencia a szerverre.
* **-Jatekos[ ] jatekosok:** Referencia a játékban lévő játékosokra.
* **-Doboz[ ] dobozok:** Referencia a dobozokra.
* **-Pontok pontok:** A játékosok pontjait tárolja.

#### Metódusok

* **Leptet(Irany i, string nev):** Elvégez egy lépést a játékban.
* **Halal(Jatekos j):** Törli az adott játékosra mutató referenciát.
* **Halál(Doboz d):** Törli az adott dobozra mutató referenciát.
* **PontotKap(Jatekos j):** Pontot ad az adott játékosnak.
* **Vegellenorzes():** Ellenőrzi, hogy van-e még doboz, ami képes lépni.

### Pont

#### Felelősség

Ez az osztály reprezentálja a játékban szerezhető pontokat. Ha egy játékos pontot szerez, akkor hozzáírja a már meglevő pontjaihoz.

#### Attribútumok

* **-string nev:** A pont tulajdonosának neve.
* **-int pont:** A játékos pontszáma.

#### Metódusok

* **PontotAd():** Pontot ad a játékosnak.

### Pontok

#### Felelősség

Ez az osztály felelős a pontok kiosztásáért, és tárolásáért.

#### Attribútumok

* **-Pont[ ] pontok:** A játékosok pontszámai.

#### Metódusok

* **PontotKap(Jatekos j):** Az adott játékosnak ad pontot.
* **Pont getPont(int i):** Visszaadja az i-edik pontot Pont objektumként.
* **int getHossz():** Visszaadja a pontok számát.
* **AddJatekos():** Új játékost adunk hozzá, hogy az ő pontjait is nyilvántarthassuk.

### UresMezo

#### Felelősség

A pályánk egy mezője, ezekből épül fel a pálya. Minden mező ismeri a szomszédos mezőket, és hogy mi található rajta. Továbbá minden mezőnek van egy „surlodas” attribútuma, ami a léptethető objektumok mozgatásához nyújt információt. Egy mezőre a játékosok elhelyezhetnek olajt, és mézet, ami hatással van a léptethető objektumokra. Ez az osztály felelős még az objektumok mozgatásáért is. Megmondja, hogy lehet-e rá objektumot mozgatni, vagy sem.

#### Attribútumok

* **-UresMezo szomszed[Irany]:** Tárolja a szomszéd mezőket a megadott irányokban.
* **-Leptetheto leptetheto:** Referencia a mezőn lévő léptethető objektumra
* **-surlodas:** A mezőn lévő súrlódás mértéke.
* **#bool olaj:** Megmondja van-e olaj a mezőn.
* **#bool mez:** Megmondja van-e méz a mezőn.
* **#int MezoTipus:** Megadja a mező típusának az értékét.

#### Metódusok

* **bool Fogad(ero: float, i: Irany, jd: JeloltDoboz, j: Jatekos)**: Játékos lép a mezőre. Ha sikeres a lépés, igaz értékkel tér vissza.
* **bool Fogad(float ero, Irany i, Jatekos aktj, Jatekos j)**: Játékos került a mezőre. A lépést kezdeményező játékos az „aktj”, a ténylegesen a mezőre kerülő játékos a „j”.
* **bool Fogad(float ero, Irany i, Doboz jd, Jatekos j)**: Doboz kerül a mezőre. A játékos az, aki a dobozt megtolta. Ha sikeres a lépés, igaz értékkel tér vissza.
* **Leptetheto Enged():** Elengedi a léptethető objektumot, ha az más mezőre kerül.
* **UresMezo GetSzomszed(Irany i):** Visszaadja a szomszéd mezőt az adott irányban.
* **UresMezo SetSzomszed(Irany i, UresMezo u):** Beállítja a megadott mezőt szomszédnak a megadott irányban.
* **SetOlaj():** Beállítja az „olaj” értékét.
* **SetMez():**Beállítja a „mez” értékét.
* **int getID():** Visszaadja a mező ID értékét.

A hálózati működéshez kellő osztályok:

### BaseAdat

#### Felelősség

Alapadatokat tartalmaz a pályáról, szerializált.

#### Interface-ek

#### Serializable

#### Attribútumok

* **-int[] palya:** Tartalmazza a pálya mezőinek ID-jait.
* **-int x:** x koordináta
* **-int y**: y koordináta

### FIFO

#### Felelősség

A kliensektől érkező parancsokat tartja nyilván egy fifo-ban, hogy a szerver a megfelelő sorrendben tudja őket kiszolgálni.

#### Attribútumok

* **-LinkedList<Pair<Irany, String>> commands**: lista a parancsokról a lépés irányának és a játékos nevének megadásával

#### Metódusok

* **push(Irany i, String nev):** Új parancs berakása a fifo-ba.
* **Pair<Irany, String> pull():** Parancs kivétele a fifo-ból.
* **boolean isEmpty():** Vizsgálat, hogy üres-e a fifo.

### Kapcsolat

#### Felelősség

Ez az osztály biztosítja, hogy létrejöjjön a kapcsolat a szerver és egy kliens között.

#### Attribútumok

* **-SzerverOlvaso olvaso:** Referencia a SzerverOlvasóra.
* **-Szerver szerver:** Referencia a szerverre.

#### Metódusok

* **Leptet(Irany i, String nev):** A kliens (játékos) lépéseit továbbítja a szerver felé az irány és a játékos nevének megadásával.
* **SendAdat(KliensAdat k):** Továbbítja az adatokat a kliens felé. (Probléma esetén kiveteti a kapcsolatot a szerver kapcsolat-listájából.)
* **remKapcs():** A kliens kapcsolatának megszakadásakor eltávolítja azt a szerver listájából.

### KliensAdat

#### Felelősség

A kliensek adatait nyilvántartó osztály.

#### Ősosztályok

KliensAdat -> BaseAdat

#### Attribútumok

* **-Pontok pontok:** A játékosok pontszámait tartja nyilván.

### KliensOlvaso

#### Felelősség

Szál egy adott kliens számára, hogy ezen keresztül továbbítsa az adatokat a játék felé.

#### Ősosztályok:

KliensOlvaso -> Thread

#### Attribútumok

* **-bool fut:** Megadja, hogy éppen fut-e a szál.
* **-Jatek jatek**: Referencia az aktuális játékra.

#### Metódusok

* **Kill():** Leállítja az osztály futását, ezzel véget vetve a szálnak.
* **run():** A szál futtatása.

### PalyaAdat

#### Felelősség

A parancsok kezeléséért felelős osztály.

#### Ősosztályok

PalyaAdat -> BaseAdat

#### Attribútumok

* **-int jsz:** Megadja a játékosok számát.
* **-int csapdak[]**: Megadja a csapdák számát.

### ParancsAdat

#### Felelősség

A parancsok kezeléséért felelős osztály.

#### Attribútumok

* **-Irany i:** Megadja a lépés irányát.
* **-string nev**: Megadja a lépést kezdeményező játékos nevét.

### Szerver

#### Felelősség

Maga a szerver, ami kezeli a hálózatot és a létrejövő kapcsolatokat. Közvetít maga a játék és a hálózat között.

#### Attribútumok

* **-Jatek jatek:** Referencia az aktuális játékra.
* **-Kapcsolat[] Kapcsolatok:** Az aktuális kapcsolatok tömbje.
* **-Palya palya:** Referencia a pályára.
* **-ServerSocket socket:** Referencia a socket-re.
* **-int port**: Referencia a portra.
* **-PalyaAdat startadat**: Referencia a kezdőállapotra.
* **-FIFO fifo**: A parancsokat tartalmazó fifo.

#### Metódusok

* **Leptet(Irany i, Jatekos j):** Továbbítja a játékosok kérését a kliensek és a szerver játékosától, a léptetésben játszik szerepet.
* **sendAdat(KliensAdat k):** Az adatokat továbbítja a szerver játékosa a kliensek felé.
* **Start():** Ez a függvény indítja el a játékot (és bezárja a lobbit).
* **Fut():** Elindítja a lobbit, ami alatt a kliensek csatlakozni tudnak.
* **End():** Lezárja a játékot és az információt a pályától a játék felé továbbítja.
* **RemKapcs():** A kliensek kapcsolatának megszakadásakor eltávolítja őket a listából.

### SzerverOlvaso

#### Felelősség

Szál a szerver számára, hogy ezen keresztül továbbítsa az adatokat a játék felé.

#### Ősosztályok

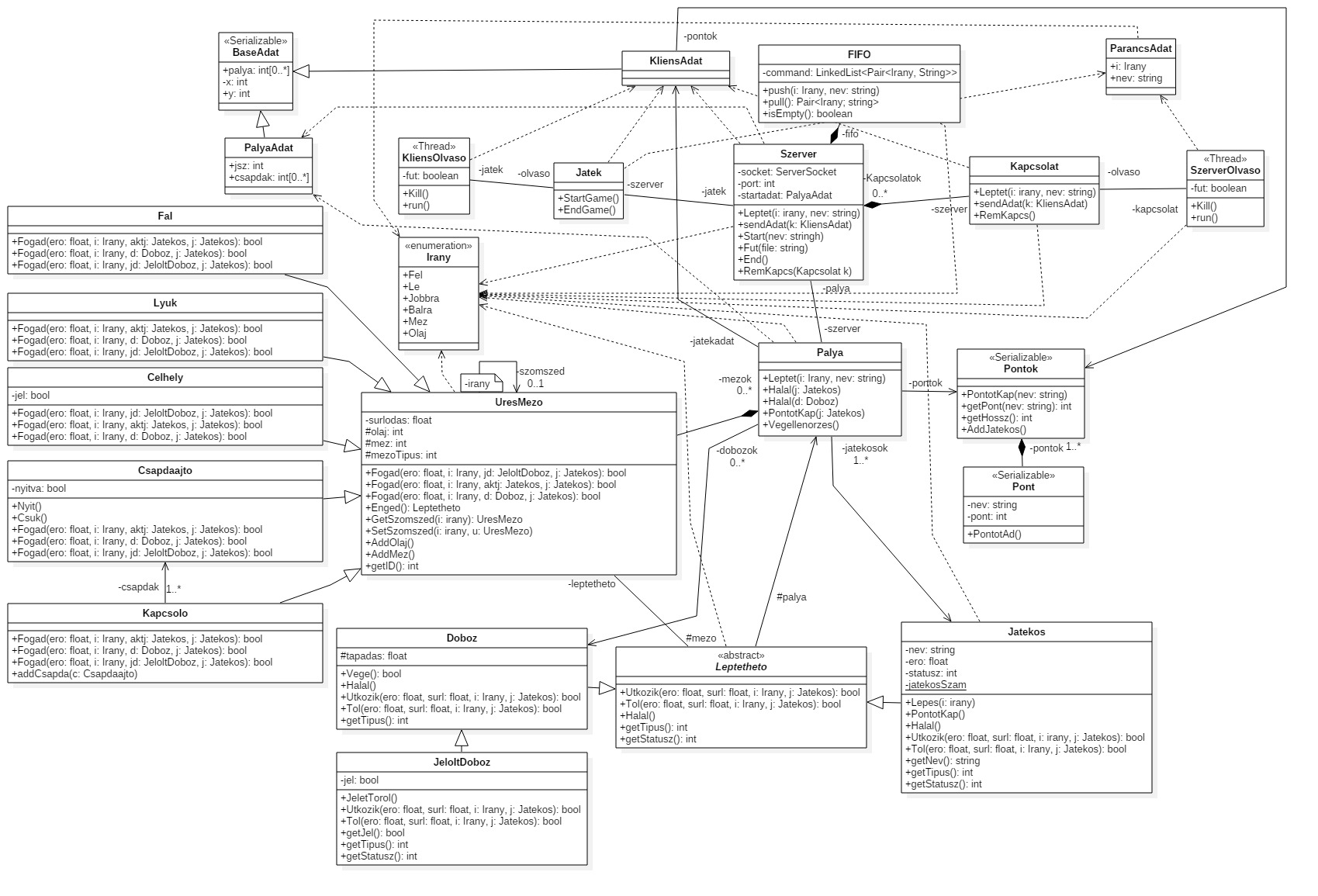
SzerverOlvaso -> Thread

#### Attribútumok

* **-bool fut:** Megadja, hogy éppen fut-e a szál.
* **-Kapcsolat kapcsolat**: Referencia a kapcsolatra.

#### Metódusok

* **Kill():** Leállítja az osztály futását, ezzel véget vetve a szálnak.
* **run():** A szál futtatása.



## A tesztek részletes tervei, leírásuk a teszt nyelvén

### Teszteset1

* **Leírás**

Egy játékos egy másik játékos belelök egy lyukba. A teszt eredményeként a második játékos meghal amint a lyukra ér.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Játékos léptetése, kliens-szerver kapcsolat felépítése, kliens-szerver adattovábbítás, játékos halála, játékos tolása

* **Bemenet**

***Szerver oldalon:***

1. s
2. Nova
3. 1111
4. test1

***Kliens oldalon:***

1. k
2. Mocsi
3. localhost
4. 1111

***Szerver oldalon:***

1. start
2. d

* **Elvárt kimenet**

***Kliens oldalon:***

*[MENU]: Opciok:*

*[MENU]: 1, Start szerver <S>*

*[MENU]: 2, Kapcsolodas <K>*

*[MENU]: 3, Quit <Q>*

*[MENU]: k*

*[Client]:*

*[Client]: KLIENS MOD [Client]:*

*[Client]: Jatekos neve: Mocsi*

*[Client]:*

*[Client]: Host IP: localhost*

*[Client]:*

*[Client]: Port: 1111*

*[Client]: Connecting to server...*

*[Client]: Connected to server!*

*00001000 00001000 00001000 00001000 00001000*

*00001000 10000000 10100000 00002000 00001000*

*00001000 00001000 00001000 00001000 00001000*

*00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 10000000 00002000 00001000*

*00001000 00001000 00001000 00001000 00001000*

***Szerver oldalon:***

*[MENU]: Opciok:*

*[MENU]: 1, Start szerver <S>*

*[MENU]: 2, Kapcsolodas <K>*

*[MENU]: 3, Quit <Q>*

*[MENU]: s*

*[SERVER]:*

*[SERVER]: SZERVER MOD [SERVER]:*

*[SERVER]: Jatekos neve: Nova*

*[SERVER]:*

*[SERVER]: Port: 1111*

*[SERVER]:*

*[SERVER]: File path: test1*

*[SERVER]: Lehetséges játékésok száma: 2*

*[SERVER]:*

*[SERVER]: Any input to start the game: [SERVER]: Client connected!*

*start*

*00001000 00001000 00001000 00001000 00001000*

*00001000 10000000 10100000 00002000 00001000*

*00001000 00001000 00001000 00001000 00001000*

*d*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@cb7a4a5*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Mocsi Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.Lyuk@2704b43*

*Lyuk Fogad(Irany, Jatekos, Jatekos)*

*Mocsi Halal()*

*Palya Halal(Jatekos)*

*Return*

*Return*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 10000000 00002000 00001000*

*00001000 00001000 00001000 00001000 00001000*

### Teszteset2

* **Leírás**

Egy játékos egy ládát belelök egy lyukba. A teszt eredményeként a láda meghal amint a lyukra ér.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Játékos léptetése, kliens-szerver kapcsolat felépítése, kliens-szerver adattovábbítás, doboz halála, doboz tolása

* **Bemenet**

***Szerver oldalon:***

1. s
2. Nova
3. 1111
4. test2

***Kliens oldalon:***

1. k
2. Mocsi
3. localhost
4. 1111

***Szerver oldalon:***

1. start
2. d

* **Elvárt kimenet**

***Kliens oldalon:***

*[MENU]: Opciok:*

*[MENU]: 1, Start szerver <S>*

*[MENU]: 2, Kapcsolodas <K>*

*[MENU]: 3, Quit <Q>*

*[MENU]: k*

*[Client]:*

*[Client]: KLIENS MOD [Client]:*

*[Client]: Jatekos neve: Mocsi*

*[Client]:*

*[Client]: Host IP: localhost*

*[Client]:*

*[Client]: Port: 1111*

*[Client]: Connecting to server...*

*[Client]: Connected to server!*

*00001000 00001000 00001000 00001000 00001000*

*00001000 10000000 20000000 00002000 00001000*

*00001000 00001000 00001000 00001000 00001000*

*00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 10000000 00002000 00001000*

*00001000 00001000 00001000 00001000 00001000*

***Szerver oldalon:***

*[MENU]: Opciok:*

*[MENU]: 1, Start szerver <S>*

*[MENU]: 2, Kapcsolodas <K>*

*[MENU]: 3, Quit <Q>*

*[MENU]: s*

*[SERVER]:*

*[SERVER]: SZERVER MOD [SERVER]:*

*[SERVER]: Jatekos neve: Nova*

*[SERVER]:*

*[SERVER]: Port: 1111*

*[SERVER]:*

*[SERVER]: File path: test2*

*[SERVER]: Lehetséges játékésok száma: 2*

*[SERVER]:*

*[SERVER]: Type "start" to start the game: [SERVER]: Client connected!*

*start*

*00001000 00001000 00001000 00001000 00001000*

*00001000 10000000 20000000 00002000 00001000*

*00001000 00001000 00001000 00001000 00001000*

*d*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@1ea8b57*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Doboz Tol(float, float, Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.Lyuk@12d755a*

*Lyuk Fogad(Irany, Doboz,Jatekos)*

*Doboz Halal()*

*Palya Halal(Doboz)*

*Return*

*Return*

*Return true*

*UresMezo Enged()*

*Return*

*Return: true*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 10000000 00002000 00001000*

*00001000 00001000 00001000 00001000 00001000*

### Teszteset3

* **Leírás**

Játékos egy ládát tol meg, ami egy másik játékost a falnak tol, így az meghak.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Játékos léptetése, kliens-szerver kapcsolat felépítése, kliens-szerver adattovábbítás, játékos halála, doboz tolása, játékos ütközése

* **Bemenet**

***Szerver oldalon:***

1. s
2. Nova
3. 1111
4. test3

***Kliens oldalon:***

1. k
2. Mocsi
3. localhost
4. 1111

***Szerver oldalon:***

1. start
2. d

* **Elvárt kimenet**

***Kliens oldalon:***

*[MENU]: Opciok:*

*[MENU]: 1, Start szerver <S>*

*[MENU]: 2, Kapcsolodas <K>*

*[MENU]: 3, Quit <Q>*

*[MENU]: k*

*[Client]:*

*[Client]: KLIENS MOD [Client]:*

*[Client]: Jatekos neve: Mocsi*

*[Client]:*

*[Client]: Host IP: localhost*

*[Client]:*

*[Client]: Port: 1111*

*[Client]: Connecting to server...*

*[Client]: Connected to server!*

*00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00001000*

*00001000 10000000 20000000 10100000 00001000*

*00001000 00000000 00000000 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000*

*00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00001000*

*00001000 00000000 10000000 20000000 00001000*

*00001000 00000000 00000000 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000*

***Szerver oldalon:***

*[MENU]: Opciok:*

*[MENU]: 1, Start szerver <S>*

*[MENU]: 2, Kapcsolodas <K>*

*[MENU]: 3, Quit <Q>*

*[MENU]: s*

*[SERVER]:*

*[SERVER]: SZERVER MOD [SERVER]:*

*[SERVER]: Jatekos neve: Nova*

*[SERVER]:*

*[SERVER]: Port: 1111*

*[SERVER]:*

*[SERVER]: File path: Test3*

*[SERVER]: Lehetséges játékésok száma: 2*

*[SERVER]:*

*[SERVER]: Type "start" to start the game: [SERVER]: Client connected!*

*start*

*00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00001000*

*00001000 10000000 20000000 10100000 00001000*

*00001000 00000000 00000000 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000*

*d*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@e5da54*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Doboz Tol(float, float, Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@223178*

*UresMezo Fogad(Irany, Doboz, Jatekos)*

*Mocsi Utkozik(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.Fal@1db124*

*Fal Fogad(Irany, Jatekos, Jatekos)*

*Return: false*

*Mocsi Halal()*

*Palya Halal(Jatekos)*

*Return*

*Return*

*UresMezo Enged()*

*Return*

*Return: true*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00001000*

*00001000 00000000 10000000 20000000 00001000*

*00001000 00000000 00000000 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000*

### Teszteset4

* **Leírás**

Egy játékos rátol egy ládát egy kapcsolóra úgy, hogy ő közben rálép a kapcsolóhoz rendelt csapdaajtóra. Ennek eredményeképpen a játékos meghal.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Játékos léptetése, kliens-szerver kapcsolat felépítése, kliens-szerver adattovábbítás, játékos halála, doboz tolása, csapdaajtó nyitása

* **Bemenet**

***Szerver oldalon:***

1. s
2. Nova
3. 1111
4. test4
5. start
6. d

* **Elvárt kimenet**

***Szerver oldalon:***

*[MENU]: Opciok:*

*[MENU]: 1, Start szerver <S>*

*[MENU]: 2, Kapcsolodas <K>*

*[MENU]: 3, Quit <Q>*

*[MENU]: s*

*[SERVER]:*

*[SERVER]: SZERVER MOD [SERVER]:*

*[SERVER]: Jatekos neve: Nova*

*[SERVER]:*

*[SERVER]: Port: 1111*

*[SERVER]:*

*[SERVER]: File path: test4*

*[SERVER]: Lehetséges játékésok száma: 2*

*[SERVER]:*

*[SERVER]: Type "start" to start the game: start*

*Kapcsolo, csapdaajto összerendelések*

*[4;2]: [3;2];*

*00001000 00001000 00001000 00001000 00001000*

*00001000 10000000 20004000 00005000 00001000*

*00001000 00001000 00001000 00001000 00001000*

*d*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.Csapdaajto@b5a469*

*Csapdaajto Fogad(Irany, Jatekos, Jatekos)*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Doboz Tol(float, float, Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.Kapcsolo@72514f*

*Kapcsolo Fogad(Irany, Doboz,Jatekos)*

*UresMezo Fogad(Irany, Doboz, Jatekos)*

*Return: true*

*Csapdaajto Nyit()*

*UresMezo Enged()*

*Return*

*Return*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Return: true*

*Nova Halal()*

*Palya Halal(Jatekos)*

*Return*

*Return*

*UresMezo Enged()*

*Return*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00004100 20005000 00001000*

*00001000 00001000 00001000 00001000 00001000*

### Teszteset5

* **Leírás**

A játékos mézet helyez egy mezőre, majd megpróbálja eltolni a ládát arról a mezőről. Ez sikertelenségben végződik.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Játékos léptetése, kliens-szerver kapcsolat felépítése, kliens-szerver adattovábbítás, doboz tolása, méz elhelyezése, súrlódás-erő számítás

* **Bemenet**

***Szerver oldalon:***

1. s
2. Nova
3. 1111
4. test5
5. start
6. w
7. d
8. d
9. s
10. m
11. m
12. m
13. m
14. m
15. m
16. m
17. m
18. m
19. w
20. a
21. a
22. s
23. d
24. d

* **Elvárt kimenet**

***Szerver oldalon:***

*[MENU]: Opciok:*

*[MENU]: 1, Start szerver <S>*

*[MENU]: 2, Kapcsolodas <K>*

*[MENU]: 3, Quit <Q>*

*[MENU]: [SERVER]:*

*[SERVER]: SZERVER MOD [SERVER]:*

*[SERVER]: Jatekos neve: [SERVER]:*

*[SERVER]: Port: [SERVER]:*

*[SERVER]: File path: [SERVER]: Lehetséges játékésok száma: 2*

*[SERVER]:*

*[SERVER]: Type "start" to start the game:*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 10000000 20000000 00000000 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@df9c2c*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 10000000 00000000 00000000 00000000 00001000*

*00001000 00000000 20000000 00000000 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@1869c7*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 10000000 00000000 00000000 00001000*

*00001000 00000000 20000000 00000000 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@27f75c*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 10000000 00000000 00001000*

*00001000 00000000 20000000 00000000 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@12563fc*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 00000000 20000000 10000000 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*AddMez()*

*return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 00000000 20000000 10000010 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*AddMez()*

*return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 00000000 20000000 10000020 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*AddMez()*

*return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 00000000 20000000 10000030 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*AddMez()*

*return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 00000000 20000000 10000040 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*AddMez()*

*return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 00000000 20000000 10000050 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*AddMez()*

*return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 00000000 20000000 10000060 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*AddMez()*

*return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 00000000 20000000 10000070 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*AddMez()*

*return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 00000000 20000000 10000080 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*AddMez()*

*return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 00000000 20000000 10000090 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@27f75c*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 10000000 00000000 00001000*

*00001000 00000000 20000000 00000090 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@1869c7*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 10000000 00000000 00000000 00001000*

*00001000 00000000 20000000 00000090 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@df9c2c*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 10000000 00000000 00000000 00000000 00001000*

*00001000 00000000 20000000 00000090 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@1bbb348*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 10000000 20000000 00000090 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@fcb2a9*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Doboz Tol(float, float, Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@12563fc*

*UresMezo Fogad(Irany, Doboz, Jatekos)*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 00000000 10000000 20000090 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@12563fc*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Doboz Tol(float, float, Irany, Jatekos)*

*Return: false*

*Return: false*

*Return: false*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 00000000 10000000 20000090 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

### Teszteset6

* **Leírás**

A játékos többször rátol egy jelölt dobozt egy célhelyre, de csak egyszer kap pontot érte.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Játékos léptetése, kliens-szerver kapcsolat felépítése, kliens-szerver adattovábbítás, doboz tolása, jelöltdoboz célhelyre érkeztetése, pont kiosztása

* **Bemenet**

***Szerver oldalon:***

1. s
2. Nova
3. 1111
4. test6
5. start
6. d
7. w
8. d
9. d
10. s
11. a
12. w
13. a
14. a
15. s
16. d

* **Elvárt kimenet**

***Szerver oldalon:***

*[MENU]: Opciok:*

*[MENU]: 1, Start szerver <S>*

*[MENU]: 2, Kapcsolodas <K>*

*[MENU]: 3, Quit <Q>*

*[MENU]: [SERVER]:*

*[SERVER]: SZERVER MOD [SERVER]:*

*[SERVER]: Jatekos neve: [SERVER]:*

*[SERVER]: Port: [SERVER]:*

*[SERVER]: File path: [SERVER]: Lehetséges játékésok száma: 2*

*[SERVER]:*

*[SERVER]: Type "start" to start the game:*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 10000000 30100000 00003100 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Pontok getHossz()*

*Return: 1*

*Pontok getPont(int)*

*Return: pontok.get(0)*

*Pont getNev()*

*Return: Nova*

*Pont getPont()*

*Return: 0*

*Nova: 0 Pontok getHossz()*

*Return: 1*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@11d04d7*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*JeloltDoboz Tol(float, float, Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.Celhely@1dafa84*

*Celhely Fogad(Irany,JeloltDoboz,Jatekos)*

*UresMezo Fogad(Irany, JeloltDoboz, Jatekos)*

*Return: true*

*JeloltDoboz getJel()*

*Return true*

*JeloltDoboz JeletTorol()*

*Return*

*Nova PontotKap()*

*Palya PontotKap(Jatekos)*

*Nova getNev()*

*Return: Nova*

*Pontok PontotKap(String)*

*Pont getNev()*

*Return: Nova*

*Pont PontotAd()*

*Return*

*Return*

*Return*

*Return*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 00000000 10000000 30003100 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Pontok getHossz()*

*Return: 1*

*Pontok getPont(int)*

*Return: pontok.get(0)*

*Pont getNev()*

*Return: Nova*

*Pont getPont()*

*Return: 1*

*Nova: 1 Pontok getHossz()*

*Return: 1*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@1e56fa7*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 10000000 00000000 00000000 00001000*

*00001000 00000000 00000000 30003100 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Pontok getHossz()*

*Return: 1*

*Pontok getPont(int)*

*Return: pontok.get(0)*

*Pont getNev()*

*Return: Nova*

*Pont getPont()*

*Return: 1*

*Nova: 1 Pontok getHossz()*

*Return: 1*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@e7f0bf*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 10000000 00000000 00001000*

*00001000 00000000 00000000 30003100 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Pontok getHossz()*

*Return: 1*

*Pontok getPont(int)*

*Return: pontok.get(0)*

*Pont getNev()*

*Return: Nova*

*Pont getPont()*

*Return: 1*

*Nova: 1 Pontok getHossz()*

*Return: 1*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@1356e08*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 10000000 00001000*

*00001000 00000000 00000000 30003100 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Pontok getHossz()*

*Return: 1*

*Pontok getPont(int)*

*Return: pontok.get(0)*

*Pont getNev()*

*Return: Nova*

*Pont getPont()*

*Return: 1*

*Nova: 1 Pontok getHossz()*

*Return: 1*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@163beb0*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 00000000 00000000 30003100 10000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Pontok getHossz()*

*Return: 1*

*Pontok getPont(int)*

*Return: pontok.get(0)*

*Pont getNev()*

*Return: Nova*

*Pont getPont()*

*Return: 1*

*Nova: 1 Pontok getHossz()*

*Return: 1*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.Celhely@1dafa84*

*Celhely Fogad(Irany, Jatekos, Jatekos)*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*JeloltDoboz Tol(float, float, Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@11d04d7*

*UresMezo Fogad(Irany, JeloltDoboz, Jatekos)*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Return: true*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 00000000 30000000 10003100 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Pontok getHossz()*

*Return: 1*

*Pontok getPont(int)*

*Return: pontok.get(0)*

*Pont getNev()*

*Return: Nova*

*Pont getPont()*

*Return: 1*

*Nova: 1 Pontok getHossz()*

*Return: 1*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@e7f0bf*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 10000000 00000000 00001000*

*00001000 00000000 30000000 00003100 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Pontok getHossz()*

*Return: 1*

*Pontok getPont(int)*

*Return: pontok.get(0)*

*Pont getNev()*

*Return: Nova*

*Pont getPont()*

*Return: 1*

*Nova: 1 Pontok getHossz()*

*Return: 1*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@1e56fa7*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 10000000 00000000 00000000 00001000*

*00001000 00000000 30000000 00003100 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Pontok getHossz()*

*Return: 1*

*Pontok getPont(int)*

*Return: pontok.get(0)*

*Pont getNev()*

*Return: Nova*

*Pont getPont()*

*Return: 1*

*Nova: 1 Pontok getHossz()*

*Return: 1*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@15b7eac*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 10000000 00000000 00000000 00000000 00001000*

*00001000 00000000 30000000 00003100 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Pontok getHossz()*

*Return: 1*

*Pontok getPont(int)*

*Return: pontok.get(0)*

*Pont getNev()*

*Return: Nova*

*Pont getPont()*

*Return: 1*

*Nova: 1 Pontok getHossz()*

*Return: 1*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@c80ff0*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 10000000 30000000 00003100 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Pontok getHossz()*

*Return: 1*

*Pontok getPont(int)*

*Return: pontok.get(0)*

*Pont getNev()*

*Return: Nova*

*Pont getPont()*

*Return: 1*

*Nova: 1 Pontok getHossz()*

*Return: 1*

*Palya Leptet(Irany, String)*

*Nova getNev()*

*Return: Nova*

*Nova Lepes(Irany)*

*Nova Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.UresMezo@11d04d7*

*UresMezo Fogad(Irany, Jatekos, Jatekos)*

*JeloltDoboz Tol(float, float, Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

*Return: sokoban.mezok.Celhely@1dafa84*

*Celhely Fogad(Irany,JeloltDoboz,Jatekos)*

*UresMezo Fogad(Irany, JeloltDoboz, Jatekos)*

*Return: true*

*JeloltDoboz getJel()*

*Return false*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Return: true*

*UresMezo Enged()*

*Return*

*Return: true*

*Palya Vegellenorzes()*

*Doboz Vege()*

*Return: true*

*Return*

*Return*

*Return*

*00001000 00001000 00001000 00001000 00001000 00001000*

*00001000 00000000 00000000 00000000 00000000 00001000*

*00001000 00000000 10000000 30003100 00000000 00001000*

*00001000 00001000 00001000 00001000 00001000 00001000*

*Pontok getHossz()*

*Return: 1*

*Pontok getPont(int)*

*Return: pontok.get(0)*

*Pont getNev()*

*Return: Nova*

*Pont getPont()*

*Return: 1*

*Nova: 1 Pontok getHossz()*

*Return: 1*

## A tesztelést támogató programok tervei

nem használtunk támogató segédprogramokat

## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.03.30. 13:30 | 10 óra | Mocsári | kód javítás, kiegészítés |
| 2018.03.30. 13:30 | 16 óra | Zsíros | kód javítás, kiegészítés |
| 2018.04.07. 12:00 | 5 óra | Horváth | Az osztályleírások elkészítése |
| 2018.04.07. 21:00 | 3 óra | Bottlik | Az osztályleírások kiegészítése, a kód és az osztálydiagram átnézése |
| 2018.04.08. 21:00 | 7.5 óra | Mocsári  Zsíros | tesztelés tervezése, előző dokumentáció javítása/cseréje  kódba maradék funkciók implementálása |
| 2018.04.09. 7:00 | 2 óra | Bottlik | Az osztályleírások kiegészítése a módosítások alapján |