# Prototípus koncepciója

## Változás hatása a modellre

### Változás leírása:

A változás lényege, hogy a játékosok nem minden esetben képesek eltolni a ládákat, mert lehetséges, hogy nem elég erősek hozzá. Minden játékosnak van egy meghatározott ereje, a ládáknak pedig tapadási súrlódási erejük van.

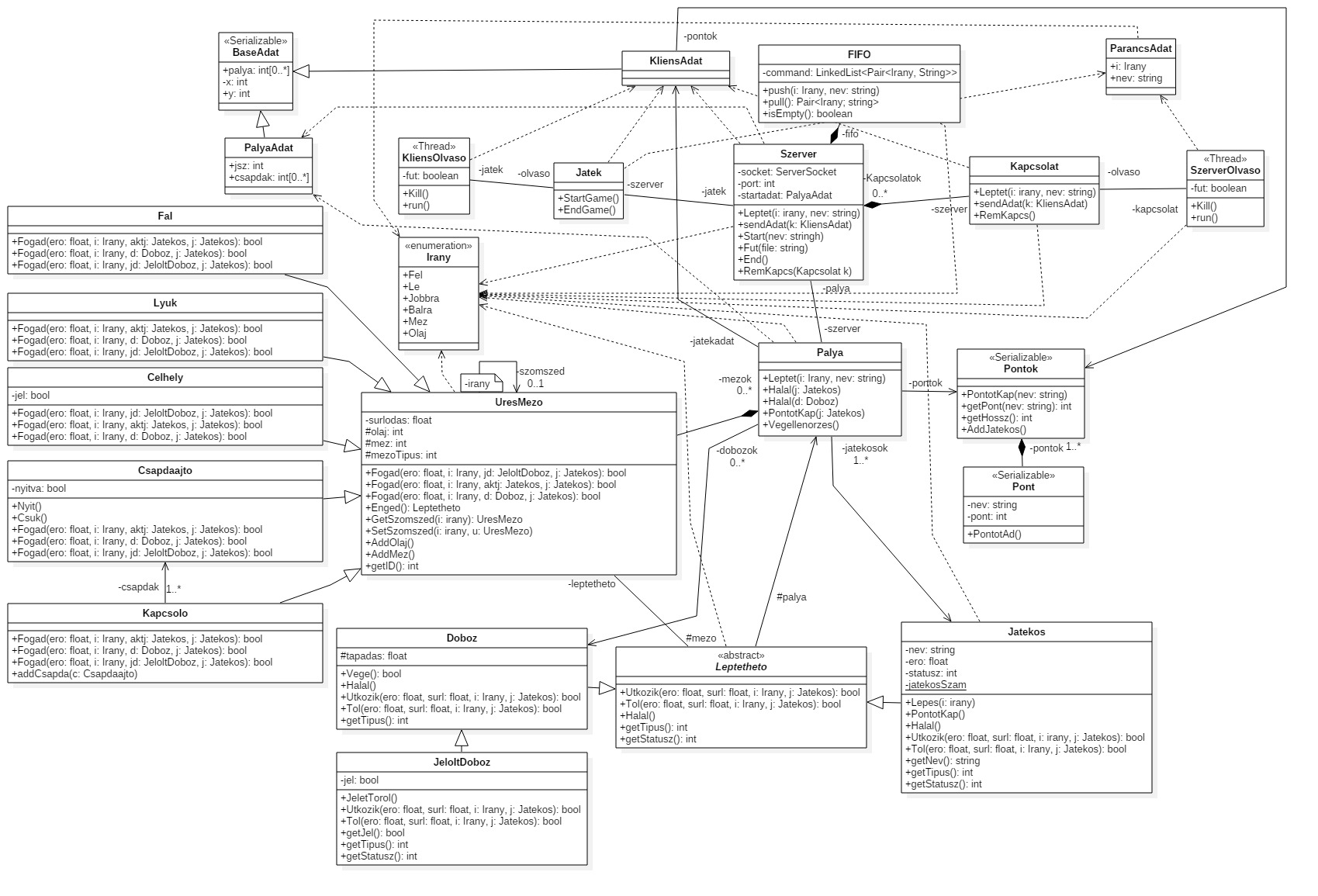
Ahhoz, hogy egy játékos el tudjon tolni egy ládát, az erejének nagyobbnak kell lenni, mint a láda súrlódási ereje. Ha több ládát próbál egyszerre eltolni, akkor a ládák tapadásai összeadódnak, és a játékos erejének az együttes erőt kell meghaladnia a sikeres toláshoz.

A mezőknek a súrlódása alapból fix szám, befolyásolják a ládák súrlódását, azonban ezt a játékosok megváltoztathatják. Egy játékos olajat vagy mézet tehet arra a mezőre, amin éppen áll. Az olaj csúszósabbá teszi a mezőt, csökkentve a rajta lévő láda tapadását, a méz pedig ragacsossá teszi, növelve a tapadási erőt. Már olajos vagy mézes mezőre újabb anyagot téve a régi nem tűnik el. Minden mezőre, ahova lépni lehet, tehetünk mézet és olajat is. Egy mezőn egyszerre több olaj, vagy méz is lehet.

**Új funkcionális követelmények**:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Azonosító** | **Leírás** | **Ellenőrzés** | **Prioritás** | **Forrás** | **Use-case** | **Komment** |
| REQ-100 | A játékosok képesek mézet elhelyezni. | Méz lehelyezése. | 7 | Rendelői követelmény (feladatkiírás) | Mozgás |  |
| REQ-101 | A játékosok képesek olajat elhelyezni. | Olaj lehelyezése | 7 | Rendelői követelmény (feladatkiírás) | Mozgás |  |
| REQ-102 | A mézes mező csökkenti a rajta lévő doboz súrlódását. | Mézes mezőn lévő doboz eltolása. | 7 | Rendelői követelmény (feladatkiírás) | DobozTolás |  |
| REQ-103 | Az olajos mező növeli a rajta lévő doboz súrlódását. | Olajos mezőn lévő doboz eltolása. | 7 | Rendelői követelmény (feladatkiírás) | DobozTolás |  |
| REQ-104 | Egy adott tolásnál a dobozok súrlódási erői összeadódnak. | Több doboz megtolása. | 7 | Rendelői követelmény (feladatkiírás) | DobozTolás |  |
| REQ-105 | Ha a játékosnak nincs elég ereje, az elmozdulás sikertelen. | Elmozdulási kísérlet kevés erővel. | 7 | Rendelői követelmény (feladatkiírás) | DobozTolás |  |
| REQ-105 | Egy mező mézezése, nem tünteti el az olajat. | Mézet teszünk olajos mezőre. | 7 | Csapat | Mozgás |  |
| REQ-106 | Egy mező olajozása, nem tünteti el az mézet. | Olajat teszünk mézes mezőre. | 7 | Csapat | Mozgás |  |
| REQ-107 | A mező képes tartalmazni több folyadékot. | Több olajat és mézet is tehetünk egy mezőre. | 7 | Csapat | Mozgás |  |

### Módosult osztálydiagram



### A diagram új elemei

#### Új vagy megváltozó osztályok

Új osztályok:

##### Szerver

* + - * **Felelősség**

A játékosok parancsainak továbbítása a pálya felé, valamint a játék létrehozása a felelőssége.

Mindenféleképpen egy localhost szerver socketen hozza létre a játékosztálytól átvett porton a szervert. Ide tudnak majd a kliensek kapcsolódni. Tartalmazza a kapcsolatokat a kliensekkel.

Tartalmaz egy szálat, ami a parancsokat, egy FIFO csatornából olvassa ki, ahova az osztály a beérkező parancsokat teszi, majd végrehajtja azokat a játékban.

* + - * **Attribútumok**
* **FIFO fifo:** A fifo csatorna amiben a parancsokat tároljuk.
* **Kapcsolat Kapcsolatok:** A kliens-kapcsolatok listája.
* **Jatek jatek:** referencia a játékra.
* **Palya palya:** referencia a pályára.
* **Socket socket:** socket amin a szerver fut.
* **Int Port:** A port, amin a szerverhez majd kapcsolódni lehet.
* **PalyaAdat startadat:** A pálya létrehozásához szükséget Pálya adatok.
  + - * **Metódusok**
* **void Leptet(Irany i, String nev)**: A léptetési parancsot továbbítja a játék felé egy külön szálon.
* **void SendAdat(KliensAdat k)**: Elküldi a megjelenítéshez szükséges adatokat a játékosokhoz.
* **void Start(String nev):** A szerver játékosának neve.
* **void Fut(String file):** A pálya fájljának neve, amit beolvastatni szeretnénk.
* **void End():** Véget vet a játéknak.
* **void RemKapcs(Kapcsolat k):** Törli az adott kapcsolatot a listából.

##### Kapcsolat

* + - * **Felelősség**

A szerver egy klienssel való kapcsolatát reprezentálja. Az adatok küldéséért felelős, valamintazok beolvasásáért egy szerverolvasón keresztül.

* + - * **Metódusok**
* **void Leptet(Irany i, String nev)**: A léptetési parancsot továbbítja a szerver felé.
* **void SendAdat(KliensAdat k)**: Elküldi a megjelenítéshez szükséges adatokat a játékosokhoz.
* **void RemKapcs(Kapcsolat k):** Törli az adott kapcsolatot a listából.

##### SzerverOlvaso

* + - * **Felelősség**

A játékosok parancsainak kiolvasása az objektumfolyamból, és azok továbbítása.

* + - * **Attribútumok**
* **boolean fut:** A szál futásának állapota.
  + - * **Metódusok**
* **void Kill()**: Megöli, azaz megállítja a szálat.
* **void run()**: A szál futásának metódusa.

##### ParancsAdat

* + - * **Felelősség**

A játékosok parancsainak tárolása, amíg azok az objektumfolyamban haladnak.

* + - * **Attribútumok**
* **Irany i:** A parancs amit a játékos végre szeretne hajtani.
* **String nev:** A jatekos neve.

##### FIFO

* + - * **Felelősség**

A játékosok parancsainak tárolása, amíg azokat végre nem hajtják.

* + - * **Attribútumok**
* **LinkedList<Pair<Irany, String>> command:** A parancsok listája pair-ekben tárolva.
  + - * **Metódusok**
* **void push(Irany i, String nev)**: Egy parancsot tesz a fifo csatornába.
* **Pair<Irany, String> pull()**: Kiolvas egy parancsot a csatornából.
* **boolean isEmpty():** Megnézi üres-e.

##### KliensAdat

* + - * **Felelősség**

A játékosok számára megjelenítendő adatokat tartalmazza.

* + - * **Ősosztályok**

BaseAdat → KliensAdat

* + - * **Attribútumok**
* **Pontok pontok:** A játékosok pontjai.

##### KliensOlvaso

* + - * **Felelősség**

A kliensek számára szükséges adatokat olvassa ki az objektumfolyamból.

* + - * **Attribútumok**
* **boolean fut:** A szál futásának állapota.
  + - * **Metódusok**
* **void Kill()**: Megöli, azaz megállítja a szálat.
* **void run()**: A szál futásának metódusa.

##### PalyaAdat

* + - * **Felelősség**

A pályák mentésére használt adatokat tárolja.

* + - * **Ősosztályok**

BaseAdat → KliensAdat

* + - * **Attribútumok**
* **int jsz:** A játékosok száma, akik a pályán játszhatnak.
* **int[] csapdak:** A pálya kapcsolóinak és csapdaajtóinak összerendelését tartalmazza.

##### BaseAdat

* + - * **Felelősség**

A Kliens és a Pálya adatok ősosztálya. A kettő közös attribútumait tárolja.

* + - * **Attribútumok**
* **int[] palya:** A pályát tárolja a mezőinek ID kódjaival.
* **int x:** A pálya szélessége.
* **int y:** A pálya magassága.

#### Új vagy megváltozó metódusok

Megváltozott metódusok:

UresMezo osztályban(és a leszármazottjaiban):

**bool Fogad(ero: float, i: Irany, jd: JeloltDoboz, j: Jatekos)**

**bool Fogad(ero: float, i: Irany, aktj: Jatekos, j: Jatekos)**

**bool Fogad(ero: float, i: Irany, d: Doboz, j: Jatekos)**

A “Fogad” függvény kezeli a léptethető objektumok mezőre lépését, kezeli a mezőtípusnak megfelelő történéseket és vizsgálja, hogy van-e másik léptethető objektum az adott mezőn. Az újdonság, hogy továbbadjuk az erőt is az eddigi paraméterek mellett, amit a “Fogad” által meghívott “Utkozik” vagy “Tol” függvény is megkap (amennyiben van az aktuális mezőn léptethető objektum).

Leptetheto osztályban(és a leszármazottjaiban):

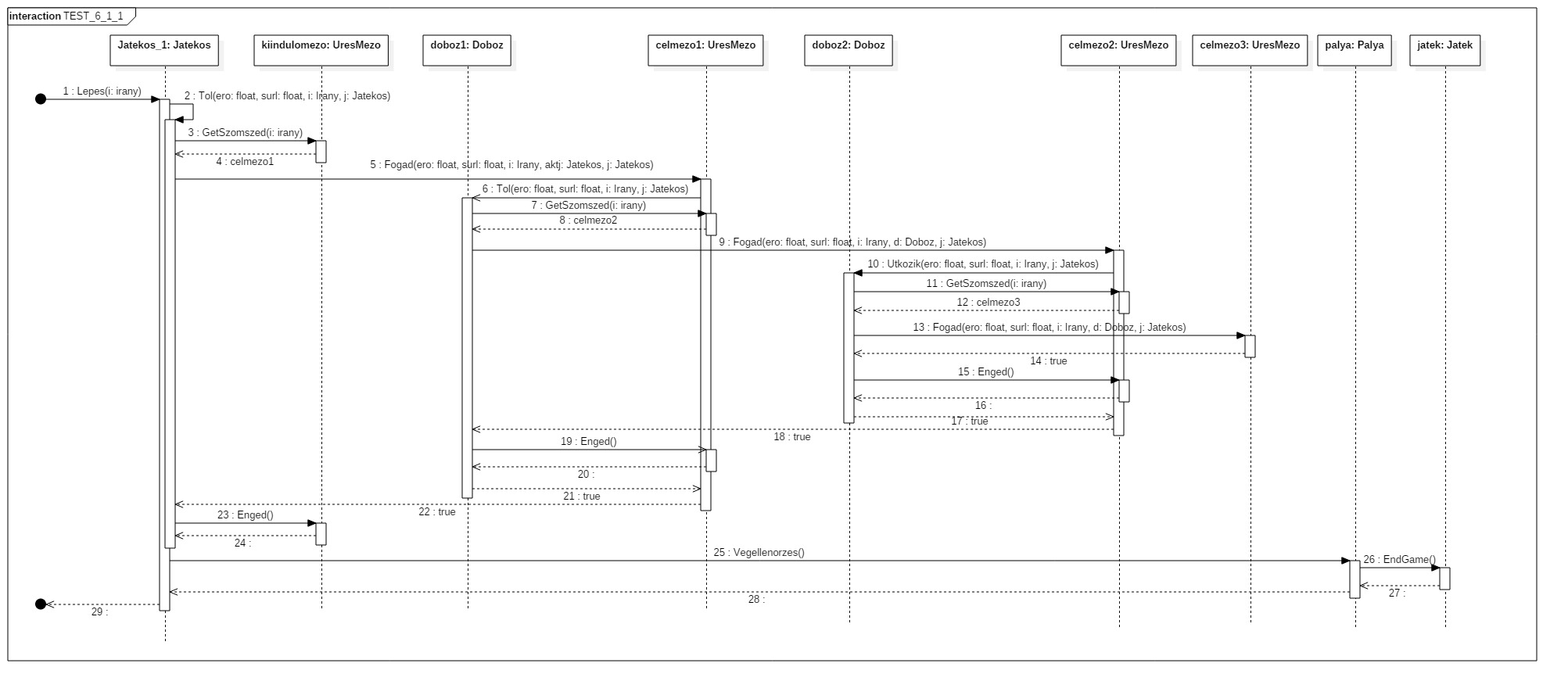
**bool Utkozik(ero: float, surl: float, i: Irany, j: Jatekos)**

**bool Tol(ero: float, surl: float, i: Irany, j: Jatekos)**

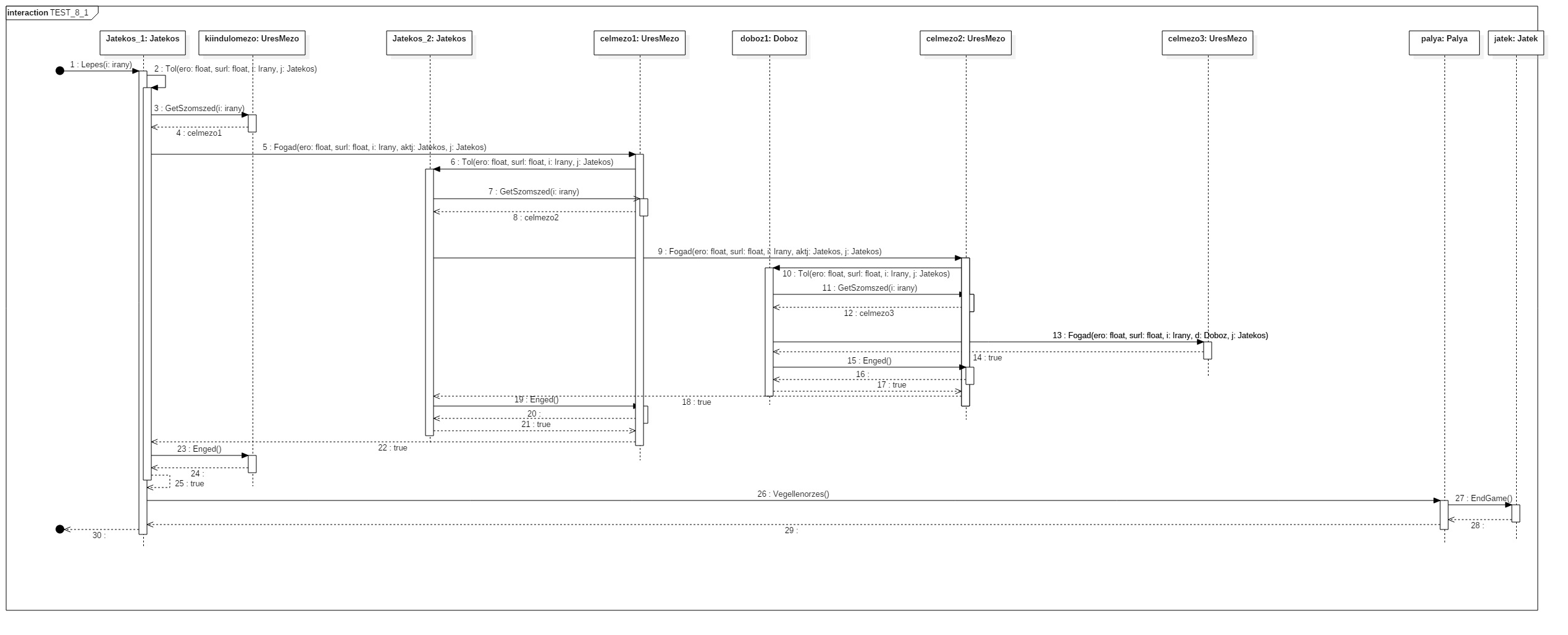
Ez a két függvény azokat az eseteket kezeli, amikor valami nekimegy az adott léptethető objektumnak. Ha ez a valami egy doboz, akkor az “Utkozik”, ha egy játékos, akkor a “Tol” függvény hívódik meg. Az újdonság, hogy átadunk két új parametert, a játékos erejét, aki a tolást kezdeményezte, és a mező súrlódását, amin a léptethető objektum van.   
Ha a léptethető objektum egy láda, akkor lehetséges, hogy már nincs elég erő eltolni az objektumot, itt történik az ellenőrzés: ha a doboz tapadása és a mező súrlódásának szorzata nagyobb, mint az átadott erő, akkor false a visszatérési érték és az elmozdulás nem történhet meg.

### Szekvencia-diagramok

Munkás-láda-láda:



Munkás-láda-munkás-láda:



## Prototípus interface-definíciója

### Az interfész általános leírása

A szabványos bemeneten és kimeneten tudunk kommunikálni a programmal. A bemeneten parancsokat adhatunk meg, a válasz/eredmény pedig megjelenik minden végrehajtáskor a kimeneten. A program képes betölteni egy adott (például korábbi) játékállapotot.

### Bemeneti nyelv

A játék kezdeti állapota egy menü. Itt választhatjuk ki a játék módjait. A programban ezután két állapot lehetséges, az egyik, amikor a játékot kliens módban futtatjuk. Ekkor egy megadott címem csatlakozni próbál egy szerverre, amivel kommunikálva játszhatunk vele. A másik, amikor szerverként fut. Ekkor a játékhoz kliensek is csatlakozhatnak.

Parancsok a kezdeti állapotban (menü):

**<szerver mód>**

**Leírás:** Szerver mód kiválasztása.

**Opciók:** <s/S>

**<kliens mód>**

**Leírás:** Kliens mód kiválasztása.

**Opciók:** <k/K>

**<kilépés>**

**Leírás:** Kilépés.

**Opciók:** <q/Q>

Parancsok a kezdeti állapotban (szerver):

**<Név>**

**Leírás:** A játékos neve.

**Opciók:** Bármilyen stringet elfogad egy sorban.

**<Port>**

**Leírás:** A játék számára kiválasztott hálózati port.

**Opciók:** Bármilyen szabad port.

**<Fájl útvonal>**

**Leírás:** A pálya fájl elérési útvonala a „ .mocsi ” kiterjesztés nélkül.

**Opciók:** bármilyen elérési útvonal, ahol pálya elérhető.

**<Indítás>**

**Leírás:** A játék elindítása az adott pályán az eddig csatlakozott kliensekkel.

**Opciók: <start>**

Parancsok a kezdeti állapotban (kliens):

**<Név>**

**Leírás:** A játékos neve.

**Opciók:** Bármilyen stringet elfogad egy sorban.

**<IP>**

**Leírás:** A hoszt IP címe.

**Opciók:** Bármilyen IP-t elfogad. Bizonyos szavakat is, pl: „localhost”.

**<Port>**

**Leírás:** A játék számára kiválasztott hálózati port.

**Opciók:** Bármilyen szabad port.

Parancsok a játék alatt (kliens és szerver):

**<parancs>**

**Leírás:** A végrehajtandó parancs, amit a játékos tenni szeretne.

**Opciók:** <w/a/s/d>/<W/A/S/D> lépésirányok, <m,o>/<M/O> méz és olaj használata.

**<kilépés>**

**Leírás:** játék megszakítása, kilépés a főmenübe.

**Opciók:** <exit>

### Kimeneti nyelv

Kezdetben megjelenő kimenet:

*[MENU]: Opciok:*

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

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

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

Nem létező parancs esetén megjelenő kimenet:

*[MENU]: Hibas parancs!*

Az adott parancsokra megjelenő lehetséges kimenetek:

**<szerver mód>**

[SERVER]:

[SERVER]: SZERVER MOD [SERVER]:

[SERVER]: Jatekos neve:

**<kliens mód>**

[Client]:

[Client]: KLIENS MOD [Client]:

[Client]: Jatekos neve:

Parancsok a kezdeti állapotban (szerver):

**<Név>**

[SERVER]:

[SERVER]: Port:

**<Port>**

[SERVER]:

[SERVER]: File path:

**<Fájl útvonal>**

[SERVER]: Lehetséges játékésok száma: X

[SERVER]:

[SERVER]: Type „start” to start the game:

Ahol X a pályához engedélyezett játékosok mennyisége.

**<Indítás>**

**[kirajzolja a pályát. Későbbiekben lesz róla szó.]**

Parancsok a kezdeti állapotban (kliens):

**<Név>**

[Client]:

[Client]: Host IP:

**<IP>**

[Client]:

[Client]: Port:

**<Port>**

[Client]: Connecting to server...

[Client]: Connected to server!

**[kirajzolja a pályát. Későbbiekben lesz róla szó.]**

Parancsok a játék alatt (kliens):

**<parancs>**

**[kirajzolja a pályát. Későbbiekben lesz róla szó.]**

**<kilépés>**

**[Visszatér a főmenübe, mintha frissen indítottuk volna a játékot.]**

Kimenet a játék alatt:

(Ha a játék véget ér, akkor ugyanaz történik, mint a „exit” parancs hatására.)

Szerver esetén egy koordináta halmaz előzi meg a pálya első kirajzolását. Ez a kapcsolók és csapdaajtók összerendelésének listája. Mintája az alábbi:

[3;2]: [30;2]; [11;20]; [20;3];

[12;1]: [36;40];

[34;10]: [23;12]; [10;2];

[50;20]: [3;21];

Minden sorban az első koordinátapár jelöli a kapcsolót, a rákövetkezők vele egy sorban pedig a rákötött csapdákat jelölik.

Mindig láthatjuk a pálya reprezentációját, ami minden lépéskor frissül. Ennek a szintaktikája a következő:

<szám> <szám> <szám> <szám> <szám> <szám>

<szám> <szám> <szám> <szám> <szám> <szám>

<szám> <szám> <szám> <szám> <szám> <szám>

<szám> <szám> <szám> <szám> <szám> <szám>

Ahol minden szám egy mező ID-ja, amit minden lépés után generálunk. Ez egy 8 jegyű szám, amelyből kiderül a mező aktuális állapota.

Felépítése:

|  |  |  |  |
| --- | --- | --- | --- |
| Neve | Számjegy sorszáma | Tulajdonság | Opciók |
| Léptethető típus | 1 | A léptethető objektum típusát azonosítja. | 0: üres, 1: játékos, 2: doboz, 3: jelölt doboz |
| Léptethető státusz | 2-3 | A léptethető objektumok egyéb adatai. | doboz lelölése (0-1), játékos azonosítója (szám) |
| Mező típus | 4-5 | A mező objektum típusát azonosítja. | 0: üres, 1: fal, 2: lyuk, 3: celhely, 4: csapda, 5: kapcsolo |
| Mező státusz | 6 | A mező objektumok egyéb adatai. | Csapda nyitva-csukva(0-1),  Celhely jel (0-1) |
| Méz | 7 | A mezőn lévő méz mennyiségének száma. (max 9) |  |
| Olaj | 8 | A mezőn lévő olaj mennyiségének száma. (max 9) |  |

Szerver esetén ezt követi a függvényhívások kimenete. Minta:

*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)*

*Mocsi1 Tol(Irany, Jatekos)*

*UresMezo GetSzomszed(Irany)*

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

*Lyuk Fogad(Irany, Jatekos, Jatekos)*

*Mocsi1 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*

## Összes részletes use-case

|  |  |
| --- | --- |
| **Use-case neve** | Pálya betöltése |
| **Rövid leírás** | Betöltünk egy adott játékállást. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | A megadott fájlba elmentett pálya rajza jelenik meg a kimeneten, ezután az adott játékállással játszatunk. |

|  |  |
| --- | --- |
| **Use-case neve** | Pálya mentése |
| **Rövid leírás** | Elmentünk egy adott játékállást. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | A megadott fájlba elmentett pálya rajza jelenik meg a kimeneten, ezután az adott játékállással játszatunk. |

|  |  |
| --- | --- |
| **Use-case neve** | Játékos mozgatása |
| **Rövid leírás** | Lépést végzünk egy adott játékossal az aktív játékmódban. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | A játékos mozog és a változás és a részletek megjelennek a kimeneten. |

|  |  |
| --- | --- |
| **Use-case neve** | Méz lerakása |
| **Rövid leírás** | Lehelyezünk mézet. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | A játékos mézesíti a mezőt, ez megjelenik a pályán és a függvényhívások között. |

|  |  |
| --- | --- |
| **Use-case neve** | Olaj lerakása |
| **Rövid leírás** | Lehelyezünk olajat. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | A játékos olajosítja a mezőt, ez megjelenik a pályán és a függvényhívások között. |

|  |  |
| --- | --- |
| **Use-case neve** | Játék befejezése |
| **Rövid leírás** | Befejezzük a játékot. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | Az aktív játékmód véget ér, kiértékelődik az addigi állás és kezdeti módba váltunk. |

## Tesztelési terv

|  |  |
| --- | --- |
| **Teszt-eset neve** | Pályakezelés |
| **Rövid leírás** | Pálya betöltése egy adott fájlból. |
| **Teszt célja** | Annak ellenőrzése, hogy megfelelően működik-e a betöltés egy adott fájlból. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Játékos mozgatása |
| **Rövid leírás** | Léptetjük a játékost. |
| **Teszt célja** | Annak ellenőrzése, hogy a megfelelő játékos a megfelelő irányba lép-e és megfelelő következmények történnek-e, illetve a pálya megfelelően frissül-e a mozgás parancs hatására. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Objektum mozgatása |
| **Rövid leírás** | A játékos mozgat egy másik objektumot. |
| **Teszt célja** | Annak ellenőrzése, hogy ha a játékos egy másik objektumot akar mozgatni, akkor az elvárt viselkedés történik-e. Mozgatható objektumok mozognak-e, illetve a pálya megfelelően frissül a mozgatás hatására. |

## Tesztelést támogató segéd- és fordítóprogramok specifikálása

A tesztelés alapvetően a parancssorban zajlik, itt jelenik meg a tesztelés kimenetele és hogy sikerrel lefutott-e. Ahhoz, hogy ezt minden esetben könnyen megállapíthassuk, egy segédprogram tartalmazza az összes teszt elvárt kimenetét. Mikor egy teszt helyességét ellenőrizni szeretnénk, csak össze kell hasonlítani az aktuális (fájlba töltött) kimenetet a segédprogram megfelelő kimenetet tartalmazó fájljával. Ha a két tartalom megegyezik, akkor a teszt sikeres volt.

## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.03.23. 21:00 | 2 óra | Zsíros | Tevékenység: programkód javítása a követelmények megváltozásának megfelelően, tesztesetek javítása, kommentek javítása |
| 2018.03.24. 13:00 | 1 óra | Mocsári | Tevékenység: Tesztelési szekvenciadiagramok javítása |
| 2018.03.24. 13:00 | 2 óra | Bottlik | Tevékenység: Tesztelési szekvenciadiagramok javítása; a korábbi osztályleírások és kezdeti szekvenciadiagramok javítása, kiegészítése |
| 2018.03.25. 01:00 | 2 óra | Bottlik | Tevékenység: Dokumentáció megírása |
| 2018.03.25. 12:00 | 1 óra | Holub | Tevékenység:  Tesztelési szekvenciadiagramok javítása |
| 2018.03.25. 18:00 | 1 óra | Horváth  Bottlik | Tevékenység:  Dokumentáció kiegészítése, véglegesítés |