

89 – megegyezesalatt

Konzulens:
Tóth Márton

Csapattagok

Bottlik Judit
Holub Csongor
Horváth Gergő
Mocsári András
Zsíros Bálint

HZY2TO	bojuu@citromail.hu
UD8CML	holub.csongor@gmail.com
GMN2QN	h.gery94@gmail.com
HGURZI	amocsari@gmail.com
FLNS3I	nova.terra94@gmail.com

2018. május 14.

2. Követelmény, projekt, funkcionálitás

2.1 Bevezetés

2.1.1 Cél

A Killer Sokoban játék részletes specifikációjának elkészítése, melyben minden információt megtalálhatunk a program készítéséről, felépítéséről, működéséről és a hozzá kapcsolódó történések ről, körülmenyekről.

2.1.2 Szakterület

A Killer Sokoban egy egyszerű számítógépes játék, a felhasználók szórakoztatására, számukra pozitív élmény szerzésére való.

2.1.3 Definíciók, rövidítések

HSZK: Hallgatói Számítógép Központ

JRE: Java Runtime Environment

2.1.4 Hivatkozások

A feladat honlapja: <https://www.iit.bme.hu/targyak/BMEVIIIA02>

2.1.5 Összefoglalás

A dokumentum bemutatja a játékot, először távolabbról egy áttekintést ad a programról, majd fokozatosan belemegy a részletekbe is és bemutatja a teljes program működését is.

A dokumentum eddigi részei:

2.1: Bevezetés, alapinformációk

2.2: Áttekintés, rövid leírások a játékról

2.3: Követelmények a programhoz

2.4: Use-case-ek felsorolása és jellemzése

2.5: Szótár, vagyis a programhoz tartozó fogalmak magyarázata

2.6: Projekt terv, információk a kivitelezésről

2.7: Napló

2.2 Áttekintés

2.2.1 Általános áttekintés

Legfontosabb alrendserek:

grafikus felület

főmenü

pályabetöltő

a léptető mechanizmus

hálózati kommunikáció

A játék első alrendsere, amivel a felhasználó találkozik, az a főmenü. Ebben a játékos megmondhatja a program további futásának tényezőit. A program legfelső szintű alrendsere a grafikus felület működése és kialakítása. Ez megjeleníti a felhasználók számára a játékteret, a menüt és lehetővé teszi, hogy a játékosok kommunikálhassanak

a játékkal. Az itt létrejövő események hatására lép működésbe a léptető mechanizmus. Az új játék elinduláskor a pályabetöltő gondoskodik arról, hogy megjelenjen a megfelelő játéktér, illetve amíg tart az „aktív játék”, addig a léptető mechanizmus jut szerephez. A program alapvető eseménye, amikor lép egy játékos, ez az alapja a működésének is. minden „lépés” hatására a program, elvégzi a szükséges módosításokat, figyeli az újonnan létrejött állapotokat és további változtatásokat eszközökkel, ha kell. Az „aktív játék” befejeződése után a program grafikus felülettel foglalkozó része veszi át az irányítást egészen az újabb „aktív játék” kezdetéig.

A játék egyik speciális alrendszere a hálózati kommunikáció. A játék alapvetően egy hálózati kapcsolattal rendelkező számítógépekre van tervezve, így lehetőség van akár sok játékossal is egyszerre játszani.

2.2.2 Funkciók

A játék egy raktárban játszódik. Ennek a raktárnak az alaprajza lesz a pálya, ami egy négyzetekre (mezőkre) felosztott téglalap. A játékosok a raktári munkásokat alkítják, akik a dobozokat (ládákat) tolják egyik mezőből a másikba. A játék célja, hogy a játékos a legtöbb ládát a helyére tolja. A játékban egyszerre több játékos is részt vehet.

A pályák alaprajzai előre meghatározottak, szerver módban, a lobby ideje alatt, választhatóan töltődnek be.

A pályán lévő négyzetek többfélék, minden fajtának megvan a maga funkciója a játékban. Egyszerű lehet egy mező fal, amin nem haladhat át semmi. A pálya széle is végig fallal van körülvéve, így adva egy keretet a játékterületnek, ahonnan nem tudunk kilépni. Ezen kívül bármelyik másik mező is lehet fal, ami akadályként szolgál a játékban. Ha egy fal melletti mezőn lévő játékos egy fal felé próbál elmozdulni, akkor ott marad a helyén, abba az irányba nem tud elmozdulni. Hasonló a helyzet a ládákkal is, a láda sem hatolhat át a falon.

A falon kívül olyan mezők vannak a pályán, amelyre játékos is léphet és láda is kerülhet. Ötféle ilyen mező van.

Az első az egyszerű üres mező, ahova mozoghatunk, ládát tolhatunk.

A második a lyuk, ami elnyeli a játékosat vagy a ládát, ha az odakerül. Az így meghalt játékos már nem kerülhet vissza a játékba (ahogy a láda sem).

A harmadik a csapdaajtó, aminek két lehetséges állapota lehet. Alapból üres mezőnek tűnik, de ha nyitva van, akkor lyukként viselkedik.

A negyedik a kapcsoló, ami a csapdaajtó állapotát változtatja. A csapdaajtók kapcsolói kinyitják az ajtókat, ha dobozt tolunk rájuk, és becsukják őket, ha nincs rajtuk láda. A kapcsoló nem kapcsol be, ha munkás áll rajta.

Az utolsó pedig a megjelölt üres mező, a célmező, ami a megjelölt ládák helyét mutatja, ahová el kell őket helyezni.

A pályán kétféle mozgatható elem van, a már korábban említett játékosok és a ládák. A játékosokat irányítják a felhasználók, akik szabadon mozoghatnak a négy irányba az erre alkalmas négyzeteken, a ládák pedig olyan objektumok, amiket a játékosok képesek

eltolni (ha a körülmények megfelelőek hozzá, lásd később). A játékosok és a ládák minden egyértelműen egy adott mezőn állnak és teljesen kitölthetik azt. Egyszerre egy mezőn csak egy mozgatható elem állhat.

Kétféle láda van, jelöletlen és jelölt. A jelöletlen ládák csupán akadályokként szolgálnak, a jelölt ládákért viszont pontot kaphatunk. Ha egy játékos egy célhelyre tol egy jelölt ládát, akkor kap egy pontot érte. Ezután az a láda jelöletlenné válik, így több pontot nem lehet kapni érte. Ilyenkor a célmező, melyre rátoltuk a ládát egyszerű üres mezővé változik.

Egy játékos úgy tud eltolni egy ládát, hogy amikor a láda melletti mezőn áll, elmozdul a láda irányába. A tolási irány mindenkor az az irány, amerről a játékos megtolta a ládát, tehát az egyenes irányban tolódik tovább, ha tud. Ha van a láda mögött (az előbb említett irányban) olyan mező, amelyre a láda mozoghat, akkor átkerül arra a szomszédos mezőre. Egy játékos képes sok ládát is eltolni, ha azok egymás mögött vannak. Ilyenkor a ládák mindenkor a tolás irányába mozdulnak arrébb egy-egy mezővel.

Ha egy ládát rátolunk egy emberre, akkor az ember is arrébb tolódik, ha van olyan mező mögötte, ami nem fal vagy egy másik láda. Ha nincs, meghal, és eltűnik a pályáról. Ilyenkor a meghalt játékos helyére kerül a láda.

A játékosok egymást is eltolhatják a ládákhoz hasonlóan, de láda nélkül nem nyomhatunk össze játékost.

A játék véget ér, ha az összes ládát a helyére toltuk (vagyis nincs már jelölt láda), ha már nem lehet többet a dobozokat tolni, vagy ha mindenki meghal. Az nyer, aki a játék végére a legtöbb pontot szerezte.

A játék hálózaton keresztül játszható. A játék kétféleképpen képes működni, szerverként, és kliensként.

A szerverként működő játék a játék indítása előtt várakozik a kliensek csatlakozására. Ebben az időben tudja a szerver játékosa kiválasztani a pályát, amit a játék betölt. A játékos akármikor megszakíthatja ezt a várakozást, és elindíthatja a játékot.

A kliensként futó játék egy a felhasználó által megadott szerverhez csatlakozva fog működni. A játékos lépéseiit, interakcióit hálózaton továbbítja a szerver felé, ahonnan az a játék aktuális állását kapja vissza.

2.2.3 Felhasználók

A felhasználók számára nincs semmi megkötés, előképzettséget sem igényel, bárki használhatja a programot, a cél a játékélmény biztosítása. Alapvetően egy vagy több játékos játszhat egyszerre, egymás ellen versenyezve.

2.2.4 Korlátozások

A program megvalósítása Java nyelven történjen, olyan elemeket felhasználva, amik képesek futni a HSZK gépein is.

2.2.5 Feltételezések, kapcsolatok

A feladat honlapja alapján tudjuk, hogy mit hogyan csinálunk.

2.3 Követelmények

2.3.1 Funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Use-case
REQ-001	A játékos minden irányba képes mozogni.	Mozgás minden irányba.	1	Rendelői követelmény (feladatkiírás)	Mozgás
REQ-002	A játékos képes eltolni a látót minden irányba.	Láda képes eltolóni minden irányba.	1	Rendelői követelmény (feladatkiírás)	DobozTolás, Mozgás
REQ-003	A játékos nem képes falon áthaladni.	Minden irányba.	1	Rendelői követelmény (feladatkiírás)	Mozgás
REQ-004	Doboz sem képes áthatolni a falon.	Minden irányba.	1	Rendelői követelmény (feladatkiírás)	DobozTolás, Mozgás
REQ-005	A lyukra lépett játékos meghal.	Minden irányba.	2	Rendelői követelmény (feladatkiírás)	Mozgás
REQ-006	A lyukra tolta doboz eltűnik.	Minden irányba.	2	Rendelői követelmény (feladatkiírás)	DobozTolás, Mozgás
REQ-007	A kapcsolóra tolta doboz, bekapcsolja a kapcsolót.	Rátolni egy látót, letolni róla, majd vissza. (ki- és bekapcsolás is)	2	Rendelői követelmény (feladatkiírás)	DobozTolás, Kapcsolás
REQ-008	A csapdaajtó alapesetben zárva van. (üres mezőként viselkedik)	Játékrendszerek látszik	3	Rendelői követelmény (feladatkiírás)	Csapda
REQ-009	A csapdaajtó nyitva lyukként viselkedik.	Játékos és doboz rátolása esetén is. (REQ-005-006)	3	Rendelői követelmény (feladatkiírás)	DobozTolás, Mozgás, Csapda
REQ-010	A csapdaajtó a rákötött kapcsoló kapcsolásakor van csak nyitva.	Rátolni egy látót, letolni róla, majd vissza. (ki- és bekapcsolás is)	3	Rendelői követelmény (feladatkiírás)	Csapda, Kapcsolás
REQ-011	A célmezők üres mezőként viselkednek jelöletlen látáson.	Játékos és doboz rátolása esetén is. (REQ-001-002)	4	Rendelői követelmény (feladatkiírás)	DobozTolás, Mozgás
REQ-012	Jelölt láták jelöletlen mezők esetén jelöletlen látaként viselkednek.	REQ-002, REQ-004, REQ-006, REQ-007	4	Csapat	DobozTolás, Mozgás
REQ-013	Jelölt láták jelölt helyre tolás után jelöletlen látaként viselkednek.	REQ-002, REQ-004, REQ-006, REQ-007	4	Csapat	DobozTolás, Mozgás
REQ-014	A játékban több	Legalább kettő	5	Rendelői	Mozgás

	játékos is részt tud venni.			követelmény (feladatkiírás)	
REQ-015	Ha egy játékosra dobozt tolunk, az átcsúszik a mögötte lévő üres mezőre.	Több játékos esetén.	5	Rendelői követelmény (feladatkiírás)	DobozTolás
REQ-016	Ha egy játékosra dobozt tolunk, de nincs mögötte üres mező, meghal.	Több játékos esetén.	5	Rendelői követelmény (feladatkiírás)	DobozTolás
REQ-017	Játékosok el tudják egymást tolni.	Több játékos esetén.	5	Csapat	Mozgás
REQ-018	A játék véget ér, ha minden doboz a helyére került.	Minden jelölt doboznak a jelölt mezőkön kell lennie	4	Rendelői követelmény (feladatkiírás)	DobozTolás
REQ-019	A játék véget ér, ha már nem tudunk több jelölt dobozt eltolni.	Ha már nincs mozgatható jelölt doboz	4	Rendelői követelmény (feladatkiírás)	DobozTolás
REQ-028	A játékban egy főmenü kezeli a program fő részeit.	Minden opciót ki kell próbálni.	4	Csapat	Menü
REQ-029	A menünek játék közben is elérhetőnek kell lennie.	Minden opciót ki kell próbálni.	4	Csapat	Menü
REQ-031	A menüben ki lehet választani a játékmódot.	Egy már lejátszott játék után is tudni kell másikat választani.	4	Csapat	Menü
REQ-033	A menüben ki lehet választani a futási módot.	Szerver vagy kliens.	6	Csapat	Hálózat
REQ-034	Több kliens is csatlakozhat egy szerverhez	Több mint egy.	6	Csapat	Hálózat
REQ-035	A szerveren a játékindítást lobby előzi meg.	Még nem indult el a játék, de a szerver már fut.	6	Csapat	Menü
REQ-036	A lobby alatt tudnak csak csatlakozni kliensek.	Játék közben és menü közben nem csatlakozhat kliens.	6	Csapat	Hálózat

2.3.2 Erőforrásokkal kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
REQ-020	JRE legalább 1.6-os verziója	bemutatás	1	Rendelői követelmény (feladatkiírás)	
REQ-021	számítógép, ami képes ezt futtatni	bemutatás	1	Rendelői követelmény (feladatkiírás)	
REQ-022	monitor	bemutatás	1	Rendelői követelmény (feladatkiírás)	
REQ-023	egér	bemutatás	1	Rendelői követelmény (feladatkiírás)	
REQ-024	billentyűzet	bemutatás	1	Rendelői követelmény (feladatkiírás)	
REQ-037	Hálózati hozzáférés többjátékos esetén.	bemutatás	2	Csapat.	Működik nélküle is szerver módban.

2.3.3 Átadással kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
REQ-025	heti dokumentum leadás	leadás	1	Rendelői követelmény (feladatkiírás)	febr 19-től hétfönként
REQ-026	kód bemutatás	bemutatás	1	Rendelői követelmény (feladatkiírás)	márc 21 - szkeleton, ápr 25 - proto, máj 26 – grafikus
REQ-027	a programnak futnia kell a HSZK gépein	bemutatás	1	Rendelői követelmény (feladatkiírás)	

2.3.4 Egyéb nem funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
REQ-032	tesztelés	bemutatás	2	Rendelői követelmény (feladatkiírás)	

2.4 Lényeges use-case-ek

2.4.1 Use-case leírások

Use-case neve	Mozgás
Rövid leírás	A játékos képes mozogni.
Aktorok	Játékos
Főforgatókönyv	1. A játékosok minden irányba képesek mozogni.
Forgatókönyv	1.A.1. A játékos képes eltolni a látót minden irányba.
Forgatókönyv	1.A.2 A játékos nem képes falon áthaladni.
Forgatókönyv	1.A.3 A lyukra lépett játékos meghal.
Forgatókönyv	1.A.4 A nyitott csapdaajtóra lépett játékos meghal.
Forgatókönyv	1.A.5 Játékosok el tudják egymást tolni.
Forgatókönyv	1.B.1. A játékban több játékos is részt tud venni.

Use-case neve	DobozTolás
Rövid leírás	A játékos képes eltolni a látót.
Aktorok	Játékos
Főforgatókönyv	1. A játékos képes eltolni a látót minden irányba.
Forgatókönyv	1.A.1 Doboz sem képes áthatolni a falon.
Forgatókönyv	1.A.2 A lyukra tolta doboz eltűnik.
Forgatókönyv	1.A.3 A kapcsolóra tolta doboz, bekapsolja a kapcsolót.
Forgatókönyv	1.A.4 A nyitott csapdaajtóra tolta doboz eltűnik.
Forgatókönyv	1.B.1. A célmezők üres mezőként viselkednek jelöletlen láda esetén.
Forgatókönyv	1.B.2. Jelölt lánkok jelöletlen mezők esetén jelöletlen lánkoként viselkednek.
Forgatókönyv	1.B.3. Jelölt lánkok jelölt helyre tolás után falként viselkednek.
Forgatókönyv	1.C.1. Ha egy játékosra dobozt tolunk, átcsúszik a mögötte lévő mezőre.
Forgatókönyv	1.C.2. Ha egy játékosra dobozt tolunk, de nincs mögötte üres mező, meghal.
Forgatókönyv	1.D.1. A játék véget ér, ha minden doboz a helyére került.
Forgatókönyv	1.D.2. A játék véget ér, ha már nem tudunk több jelölt dobozt eltolni.

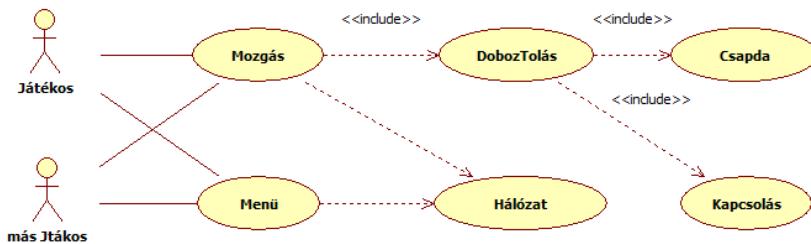
Use-case neve	Kapcsolás
Rövid leírás	A játékos képes a kapcsolókat bekapsolni dobozok kapcsolóira tolásával.
Aktorok	Játékos
Főforgatókönyv	1. A kapcsolóra tolta doboz, bekapsolja a kapcsolót.
Forgatókönyv	1.A.1. A csapdaajtó a rákötött kapcsoló kapcsolásakor van csak nyitva.

Use-case neve	Csapda
Rövid leírás	A csapdák lyukként, és üres mezőként is viselkedhetnek. Ezt a rákötöt kapcsoló befolyásolja.
Aktorok	Játékos
Főforgatókönyv	1. A csapdaajtó a rákötött kapcsoló kapcsolásakor nyitva van.
Forgatókönyv	1.A.1. A csapdaajtó alapesetben üres mezőként viselkedik.
Forgatókönyv	1.A.2. A csapdaajtó nyitva lyukként viselkedik.

Use-case neve	Menü
Rövid leírás	A játék menüjében van a játékosnak lehetősége a játék feltételein változtatni.
Aktorok	Játékos
Főforgatókönyv	1. A játékban egy főmenü kezeli a program fő részeit.
Forgatókönyv	1.A.1. A menünek játék közben is elérhetőnek kell lennie.
Forgatókönyv	1.A.2. A játéknak tudnia kell menteni és betölteni játékállásokat.
Forgatókönyv	1.A.3. A menüben ki lehet választani a játékmódot.
Forgatókönyv	1.A.4. A szerveren a játékindítást lobby előzi meg.

Use-case neve	Hálózat
Rövid leírás	Ki lehet választani a futási módot. Szerver vagy kliens mód lehetséges.
Aktorok	Játékos
Főforgatókönyv	1. Kliensek képesek csatlakozni a szerverhez.
Forgatókönyv	1.A.1. Több kliens is csatlakozhat egy szerverhez.
Forgatókönyv	1.A.2. A lobby alatt tudnak csak csatlakozni kliensek.

2.4.2 Use-case diagram



2.5 Szótár

Célmező	Olyan speciális üres mezők, melyek el vannak látva megkülönböztető jelzéssel. A jelölt lánckat ezekre a mezőkre kell elhelyezniük a játékosoknak.
Csapdaajtó	Olyan speciális mező, amelynek két állapota van. Zárvá üres mezőként viselkedik, nyitva pedig lyukként. Az állapota a hozzákapcsolt kapcsolótól függ.
Fal	Olyan mező, amelyre a játékos nem léphet, és nem tolhat dobozt.
Játékos	A felhasználó által vezérelt raktári munkás. Ő tologatja a dobozokat a raktárban.
Kapcsolás	Ha a játékos dobozot tol a kapcsolóra, akkor az bekapsol. Ha a doboz eltolják róla, akkor kikapsol.
Kapcsoló	A kapcsolóknak két állapota van. Ha van rajta doboz, akkor be van kapcsolva, minden más esetben ki van kapcsolva. Bekapsolt állapotában kinyitja a hozzákapcsolt csapdaajtókat.
Láda	A raktárban megtalálható dobozok. A játékosnak ezeket kell tologatni a pályán.
Láda (jelölt)	Olyan speciális doboz, amit a játékosnak a megjelölt mezőkre kell juttatnia, hogy pontot kapjon.
Lyuk	Olyan speciális mező, amelyre ha doboz kerül, akkor az eltűnik, ha pedig játékos, akkor az meghal.
Meghalni	A játékos, ha meghal, eltűnik a pályáról és veszít.
Mező	A pálya négyzet alakú mezőkre van felosztva. Ezeken tologatja a játékos a dobozokat.
Pálya	A raktár, amiben a játékos a dobozokat tologatja.
Tolni	Olyan folyamat, amiben egy játékos, és egy vagy több láda helye megváltozik, Vagyis ahol egy játékos egy lánccat vagy több lánccat egy neki szomszédos mezőre juttat.
Hálózat	Számítógépek egy hálózata.
Szerver	Olyan játék, amihez a kliensek csatlakoznak. A játék érdemi része ezen fut.
Kliens	Olyan játék, ami a szerverekhez csatlakozik. A parancsok elküldése, és a szervertől érkező adatok alapján a pálya grafikus megjelenítése a feladata.
Lobby	A menü és a szerver játékának indítása között lévő idő. Itt várakozunk a kliensek csatlakozására.
Csatlakozás	A számítógépek hálózatán egy kiépült kapcsolat két számítógép között.

2.6 Projekt terv

Hatóidők:

- 02. 19.: Követelmény, projekt, funkcionálitás – beadás
- 02. 26.: Analízis modell kidolgozása 1. – beadás
- 03. 05.: Analízis modell kidolgozása 2. – beadás
- 03. 12.: Szkeleton tervezése – beadás
- 03. 19.: Szkeleton - beadás és a forráskód herculesre való feltöltése**
- 03. 26.: Prototípus koncepciója – beadás
- 04. 09.: Részletes tervezés – beadás
- 04. 16.: **Prototípus készítése, tesztelése**
- 04. 23.: Prototípus - beadás és a forráskód, a tesztbemenetek és az elvárt kimenetek herculesre való feltöltése**
- 05. 02.: Grafikus felület specifikációja – beadás
- 05. 07.: **Grafikus változat készítése (nem beadás)**
- 05. 14.: Grafikus változat - beadás és a forráskód herculesre való feltöltése**
- 05. 18.: Összefoglalás - beadás és feltöltés

A csapat tagjai:

Bottlik Judit Krisztina (HZY2TO)
Zsíros Bálint György (FLNS3I)
Horváth Gergő Krisztián (GMN2QN)
Mocsári András Attila (HGURZI)
Holub Csongor (UD8CML)

Együttműködés, feladatok:

A csapat tagjainak a feladata előre nincs pontosan meghatározva, mindig az aktuális helyzetben döntjük el, hogy ki mit csinál, tehát nem határozunk meg egyértelmű felelősségeket a tagokhoz. Arra természetesen figyelünk, hogy nagyjából hasonló nehézségű és mennyiségi feladatot végezzen mindenki.

Hetente párszor összejönnünk konzultálni (a hivatalos konzultación kívül, általában szerda reggel és csütörtökön) és itt megbeszéljük együtt az alapokat, és nagyjából tisztázzuk, hogy ki mit fog az adott héten elvégezni. Ez mindig függ a csapattagoktól, hogy mihez értenek, mit szeretnének csinálni és mennyi idejük van. A napi megbeszélés („Daily standup”) többnyire az alábbiak szerint, online történik, így a folyamatos kommunikáció a csapattagok között az értekezletek között is biztosítva van.

A hét további részében elvégezzük a szükséges feladatokat, közben pedig, a napi megbeszéléseken egymással kommunikálunk és tisztázzuk a felmerülő problémákat, tisztázatlan dolgokat. Ehhez a Discord nevű programot használjuk, ahol lehetőség van rá, hogy mind az 5-en beszélgettünk egyszerre. A dokumentumokat is ide tudjuk minden felölteni, így lehetőség van arra, hogy egymást folyamatosan javítgatva, fokozatosan fejlesszük a beadandó anyagokat. A forráskódok megosztása is hasonlóan zajlik majd.

2.7 Napló

Kezdet	Időtartam	Résznevők	Leírás
2018.02.15. 8:00	1,5 óra	Bottlik Holub Mocsári Zsíros	Értekezlet. Döntés: Közösen megírtuk a vázlat egy részét (2.1.1-2 2.2.1-2), Továbbiakban Zsíros írja tovább a következő megbeszélésig.
2018.02.15 15:00	2 óra	Zsíros	Zsíros kiegészíti a meglévő specifikációt. (2.3.1, 2.4, 2.5)
2018.02.17 8:00	2 óra	Bottlik	A specifikáció további kiegészítése, pontosítása (2.1, 2.2, 2.3.2- 3-4, 2.6)
2018.02.17 11:00	1 óra	Zsíros	A specifikáció ellenőrzése, javítása. (2.2.1, 2.3.1, 2.4, 2.6)
2018.02.17 14:00	1 óra	Horváth Zsíros Bottlik	A specifikáció végelesítése, átbeszélése, utolsó pontosítások
2018.02.23. 21:00	0,5 óra	Bottlik	A specifikáció javítása (2.2.2)
2018.02.25. 17:00	0,5 óra	Zsíros	A specifikáció kiegészítése, az új részekkel. (hálózat) (2.2.1-2, 2.3.1-

			2, 2.4, 2.5)
--	--	--	--------------

3. Analízis modell kidolgozása

3.1 Objektum katalógus

3.1.1 Celhely

A pálya egy eleme. Olyan üres mező, amely meg van jelölve. Ha jelölt látát tolnak rá akkor a láda, és a mező is elveszti a jelölést, illetve a játékos, aki a látát tolta pontot kap.

3.1.2 Csapdaajtó

A pálya egy eleme. Ha zárva van, akkor üres mezőként, ha nyitva, akkor lyukként viselkedik. Az állása a hozzá kötött kapcsolótól függ.

3.1.3 Doboz

Mozgatható elem. A játékosok ezeket tologatják a pályán. Alapterülete egy mezőével egyenlő és összenyomhatatlan. A dobozok képesek egymást vagy egy játékost eltolni, ha egy másik játékos rájuk tolja a dobozt. Ha a játékost nincs hova eltolni, azaz nincs mögötte üres mező, akkor a játékos meghal, és a doboz átveszi a helyét.

3.1.4 Fal

A pálya egy eleme. Ez sem tartalmaz semmit. A játékosok nem képesek látát tolni rá, vagy róla le, és nem képesek rálépní, vagy róla le. Ezek képzik majd az akadályokat a pályán.

3.1.5 irány

Egy enum, amivel az irányokat jelöljük majd.

3.1.6 Jatek

A program fő része, ami kezeli, a felhasználó parancsait, és kliens működés esetén kommunikál a szerverrel.

3.1.7 Jatekos

A raktári dolgozók, akik a dobozokat tolják a pályán. A felhasználó ezt irányítva tud játszani. A játék célja, hogy az összes jelölt dobozt a célhelyekre juttassuk.

3.1.8 JeloltDoboz

Olyan doboz, amit a játékosnak a célhelyre kell tolnia. Ha rátolta, akkor a doboz és a célhely is elveszti a jelölést, és doboz valamint üres mező lesz belőlük. minden más esetben dobozként viselkedik.

3.1.9 Kapcsolat

Ez a szerver-kliens kapcsolatokat fogja reprezentálni szerveroldalon. minden bejövő kapcsolat kap egy saját kommunikációs csatornát, amin a szerverrel kommunikálhat.

3.1.10 Kapcsolo

A pálya egy eleme. Ha egy játékos egy látótól a kapcsolóra, akkor az kinyitja a hozzákapcsolt csapdaajtót. Ha játékos áll rajta, akkor nem csinál semmit.

3.1.11 KliensAdat

A kliens számára szükséges adatokat tartalmazza, amik alapján ki tudja rajzolni a képernyőre az aktuális játékállást, pontokat. Ezek alkotják majd a szervertől a kliens felé mutató adatforgalmat.

3.1.12 Leptetheto

Minden olyan objektumot reprezentál amik a pályán mozognak. Ide tartozik a játékos és a dobozok is.

3.1.13 Lyuk

A pálya egy eleme. Olyan mező, amelybe lépve a játékos, vagy láda eltűnik.

3.1.14 OlvasoSz

Ez az objektum fogja a bejövő adatforgalmat kezelní a szerver oldalán. Kiolvassa az adatokat az adatfolyamból.

3.1.15 OlvasoK

Ez az objektum fogja a bejövő adatforgalmat kezelní a kliens oldalán. Kiolvassa az adatokat az adatfolyamból.

3.1.16 Palya

A pálya tartalmazza a mezőket. A játékosok ezen az objektumon végezik el a teendőiket. Ez jegyzi meg a pálya állását, amit a játék a kliensekkel közöl, ez köti össze a játék elemeit. Számon tartja, és lépteti a játékosokat.

3.1.17 ParancsAdat

A szerver számára szükséges adatokat tartalmazza, amik alapján léptetheti a játékost. Ezek alkotják majd a klienstől a szerver felé mutató adatforgalmat.

3.1.18 Pont

A játékosok pontszámait tárolja el a játékosok azonosítóival.

3.1.19 Pontok

A játékosok pontjait tárolja.

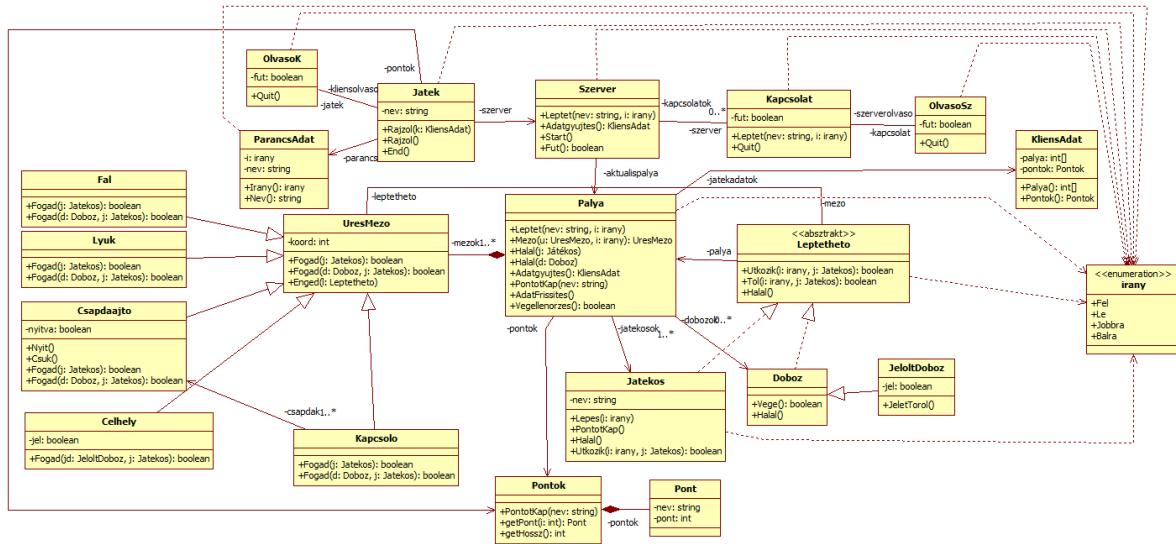
3.1.20 Szerver

A szerver a játék egyik központi eleme. Ezen keresztül történik a kommunikálás a kliensek, és a szerver játékosa felé.

3.1.21 UresMezo

A pálya egy eleme. Ez a legegyszerűbb elem. A játékosok képesek ládát tolni rá, vagy róla le, és képesek rálépni, vagy róla le.

3.2 Statikus struktúra diagramok



3.3 Osztályok leírása

3.3.1 Celhely

Felelősség

A pálya egy eleme. Olyan üres mező, amely meg van jelölve. Ha jelölt látát tolnak rá akkor a láda, és a mező is elveszti a jelölést, illetve a játékos, aki a látát tolta pontot kap.

Ősosztályok

UresMezo → Celhely

Attribútumok

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

Metódusok

- Fogad(jd: JeloltDoboz, jl: Jatekos):** Az üres mező fogad függvényeit egészít ki még egyel, ami jelölt dobozt vár, és a visszatérési értéke a lépés sikeressége. A játékos referencia ahhoz kell, amikor egy sor megtolt doboz végén egy jelölt doboz pontot akar adni a játékosnak.

3.3.2 Csapdaajto

□ Felelősség

A pálya egy eleme. Ha zárva van, akkor üres mezőként, ha nyitva, akkor lyukként viselkedik. Az állása a hozzákötött kapcsolótól függ, de kezdetben minden csapda ajtó zárva van.

□ Ősosztályok

UresMezo → Csapdaajto

□ Attribútumok

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

□ 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.
- **Fogad(j: Jatekos):** ha a csapdaajtó állapota nyitott, akkor a rákerülő játékos meghal (és igazzal tér vissza), ha az állapota csukott, akkor úgy viselkedik, mint az Ősosztálya.
- **Fogad(d: Doboz, j: Jatekos):** ha a csapdaajtó állapota nyitott, akkor a rákerülő doboz meghal (és igazzal tér vissza), ha az állapota csukott, akkor úgy viselkedik, mint az Ősosztálya.

3.3.3 Doboz

□ Felelősség

Mozgatható elem. A Dobozok reprezentációjáért felelős osztály. A játék végén a saját végellenőrzését is ez végzi.

□ Ősosztályok

Leptetheto → Doboz

□ Metódusok

- **Vege():** Ellenőrzi, hogy a doboz képes e még mozogni, vagy már nem lehet semerre sem tolni.
- **Halal():** Eltünteti a dobozt a játékból.

3.3.4 Fal

□ Felelősség

A pálya egy eleme. Ezek képzik az akadályokat a pályán.

□ Ősosztályok

UresMezo → Fal

□ Metódusok

- **Fogad(j: Jatekos):** az UresMezo Fogad metódusát írja felül. Elutasítja az érkeztetési kérést (hamissal tér vissza).
- **Fogad(d: Doboz, j: Jatekos):** az UresMezo Fogad metódusát írja felül. Elutasítja az érkeztetési kérést (hamissal tér vissza)csapdaasadsaad.

3.3.5 irany

□ Felelősség

Enum osztály, a mozgási irányokat tárolja.

□ Attribútumok

- **Fel**
- **Le**
- **Jobbra**
- **Balra**

3.3.6 Jatek

□ Felelősség

Ez az osztály felelős a program és a játékosok kezeléséért.

□ Attribútumok

- **nev:** Az aktuális játékos neve, amivel a játékban azonosítani lehet.
- **pontok:** A játékosok pontjai. A játék vége utáni kiiratáshoz kell.
- **kliensolvaso:** A kliens adatfolyam olvasó szálja, ami a hálózaton figyeli a szervertől érkező adatokat.
- **szerver:** A játék szervere.
- **parancs:** a játék kliensként parancsadatokat küld a szervernek.

□ Metódusok

- **Rajzol(k: KliensAdat):** kliens működés esetén a bejövő adatok alapján rajzolja ki a pályát.
- **Rajzol():** szerver működés esetén a meglévő adatokból rajzolja ki a pályát.
- **End():** A játék végén hívódik meg. Ezzel lehet lezárni a pályát.

3.3.7 Jatekos

□ **Felelősség**

A raktári dolgozók, akik a dobozokat tolják a pályán. A felhasználó ezt irányítva tud játszani. A lépések megvalósításáért felelős.

□ **Ősosztályok**

Leptetheto → Jatekos

□ **Attribútumok**

- **nev:** A felhasználó azonosítója, aki a játékost irányítja.

□ **Metódusok**

- **Lepes(i: irany):** A pálya hívja meg, hogy lépesse a játékest, ami meghívja a mező fogad függvényét.
- **PontotKap():** A jelölt lánkok célhelyre tolásakor a játékos pontot kap. Ez a függvény felelős érte.
- **Halal():** Megöli a játékest.

3.3.8 JeloltDoboz

□ **Felelősség**

Olyan doboz, amit a játékosnak a célhelyre kell tolnia. Ha rátolta, akkor a jelölt doboz elveszti a jelölést.

□ **Ősosztályok**

Leptetheto → Doboz → JeloltDoboz

□ **Attribútumok**

- **jel:** Ha a célmézőre jelölt lánca kerül, a JeloltDoboz elveszti a jelölését. Ez az attribútum ezt jelöli.

□ **Metódusok**

- **JeletTorol():** Törli a jelölést a dobozról.

3.3.9 Kapcsolat

□ **Felelősség**

Ez a szerver-kliens kapcsolatokat fogja reprezentálni szerveroldalon. A kimenő forgalom kezeléséért felelős. Külön szálon fut.

□ **Attribútumok**

- **fut:** A szál futásához van rá szükség. Leállításkor hamis értékre állítjuk.

- **szerver:** A szerver referenciája, amihez a kapcsolat tartozik.
- **szerverolvaso:** A szerver adatfolyam olvasó szálja, ami a hálózaton figyeli a klienstől érkező adatokat.

□ Metódusok

- **Leptet(string nev, irany i):** Továbbítja az olvasótól érkező parancsot.
- **Quit():** Leállítja a szál futását a "fut" hamisra állításával.

3.3.10 Kapcsolo

□ Felelősség

A pálya egy eleme. Ha egy játékos egy látótól tol a kapcsolóra, akkor az kinyitja a hozzákapcsolt csapdaajtókat, ha ő lép a kapcsolóra, akkor bezárja.

□ Ősosztályok

UresMezo → Kapcsolo

□ Attribútumok

- **csapdak:** A kapcsolóhoz tartozó csapdák.

□ Metódusok

- **Fogad(jatekos j):** Ha játékos lép rá akkor becsukja a csapdaajtókat.
- **Fogad(Doboz d, Jatekos j):** Ha doboz lép rá akkor kinyitja azokat.

3.3.11 KliensAdat

□ Felelősség

A kliens adatainak tárolásáért felelős. minden olyan adatot tárol, amire a kliensnek szüksége van, hogy megjelenítendő felületet kirajzolhassa.

□ Attribútumok

- **palya:** Int tömb, ahol egy speciális kódot használunk a mezők tárolására, és ezeket a kódokat használjuk hálózaton való kommunikáláshoz.
- **pontok:** A teljes ponttábla.

□ Metódusok

- **Palya():** Visszaadja az int tömböt, hogy ki lehessen belőle rajzolni a pályát.
- **Pontok():** visszaadja a pontokat.

3.3.12 Leptetheto

□ **Felelősség**

Absztrakt osztály, ami segítségével a játékosokat és a dobozokat is képesek vagyunk a mezők azonos attribútumában tárolni.

□ **Attribútumok**

- **palya:** Referencia a pályára. A mező lekérdezésekhez kell.
- **mezo:** Az aktuális mezőt tárolja, amin az objektum van.

□ **Metódusok**

- **Halal:** absztrakt függvény a leszármazottak halálához.
- **Utkozik(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.
- **Tol(irany i, Jatekos j):** Ha egy mozgatható objektumot játékos 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.

3.3.13 Lyuk

□ **Felelősség**

Olyan mező, amelybe lépve a játékos, vagy láda eltűnik.

□ **Ősosztályok**

UresMezo → Lyuk

□ **Metódusok**

- **Fogad(Jatekos j):** Meghívja a játékos halál függvényét, és igazzal tér vissza.
- **Fogad(Doboz d, Jatekos j):** Meghívja a doboz halál függvényét, és igazzal tér vissza.

3.3.14 OlvasoK

□ **Felelősség**

A hálózaton beérkező adatok olvasása kliens oldalon, és azokban tárolt információk, és adatok továbbítása a feladata.

□ **Attribútumok**

- **fut:** A szál futásához van rá szükség. Leállításkor hamis értékre állítjuk.
- **jatek:** kliensként a játék referenciája, aminek a szervertől kapott adatokat továbbítja.

□ **Metódusok**

- **Quit():** Leállítja a szál futását a “fut” hamisra állításával.

3.3.15 OlvasoSz

□ Felelősség

A hálózaton beértkező adatok olvasása szerver oldalon, és azokban tárolt információk, és adatok továbbítása a feladata.

□ Attribútumok

- **fut:** A szál futásához van rá szükség. Leállításkor hamis értékre állítjuk.
- **kapcsolat:** szerverként a kapcsolat referenciája, ahova az olvaso a klienstől kapott adatokat továbbítja.

□ Metódusok

- **Quit():** Leállítja a szál futását a “fut” hamisra állításával.

3.3.16 Palya

□ Felelősség

Tárolja a mezőket, játékosokat, dobozokat. Összeköti a játék különálló részeit, és előállítja a kirajzoláshoz szükséges adatokat. Ez végzi a játék végellenőrzését.

□ Attribútumok

- **mezok:** Eltárolja a pályán lévő mezőket egy egy-dimenziós tömbben.
- **játékosok:** Referencia a játékban lévő játékosokra.
- **dobozok:** Referencia a dobozokra.
- **pontok:** A játékosok pontjait tárolja.
- **jatekadatok:** Az aktuális játékállás adatai. Hálózati kommunikációra, és a játék kirajzolásához van rá szükség.

□ Metódusok

- **Leptet(String nev, irany i):** A kapott parancsot továbbítva lépteti a játékoszt.
- **Mezo(UresMezo u, irany i):** A játékos számára ad vissza referenciát a meglépni szándékozott mezőről.
- **Halal(Játékos j):** Törli az adott játékosra mutató referenciát.
- **Halál(Doboz d):** Törli az adott dobozra mutató referenciát.
- **Adatgyujtes():** Visszaadja a járék adatait.
- **PontotKap(string nev):** Pontot ad az adott játékosnak.
- **AdatFrissites():** Frissíti a játék adatait a kliense és a játék kirajzolása számára.

- **Vegellenorzes()**: Ellenörzi, hogy van e még doboz ami képes lépni.

3.3.17 ParancsAdat

▫ Felelősség

Játékos mozgásához szükséges adatokat tárolja. hálózati kommunikációnál van rá szükség.

▫ Attribútumok

- **i**: Az irány enum megfelelő értéke.
- **nev**: A játékos neve, aki a parancsot küldte.

▫ Metódusok

- **Irany()**: Visszadja az irányt.
- **Nev()**: Visszaadja a nevet.

3.3.18 Pont

▫ Felelősség

A kiosztandó pont megvalósítása.

▫ Attribútumok

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

3.3.19 Pontok

▫ Felelősség

A pontok kiosztásáért és tárolásáért felelős osztály.

▫ Attribútumok

- **pontok**: A játékosok azonosítói, és a hozzájuk tartozó pontok.

▫ Metódusok

- **PontotKap(nev: string)**: Az adott játékosnak ad pontot.
- **getPont(int i)**: Visszaadja az i-edik pontot Pont objektumként.
- **getHossz()**: Visszaadja a pontok számát.

3.3.20 Szerver

▫ Felelősség

Játék információit továbbítja a megfelelő helyekre.

□ Attribútumok

- **kapcsolatok:** A szerver kliens kapcsolatok listája.
- **aktualispalya:** Az éppen folyó játék pályájának referenciája.

□ Metódusok

- **Leptet(string nev, irany i):** Továbbítja a játékos és a kliensek parancsait.
- **Adatgyujtes():** A kirajzoláshoz szükséges adatokat gyűjti be a pályától.
- **Start(fajl: string):** Elindítja a játékot, a kiválasztott pályával. A lobby miatt van rá szükség.
- **Fut():** Visszaadja, hogy a játék elkezdődött-e már.

3.3.21 UresMezo

□ Felelősség

A játékosok képesek látát tolni rá, vagy róla le, és képesek rálépni, vagy róla le.

□ Attribútumok

- **koord:** A mező pályán lévő koordinája.
- **leptetheto:** A pályán tartózkodó léptethető objektum referenciája.

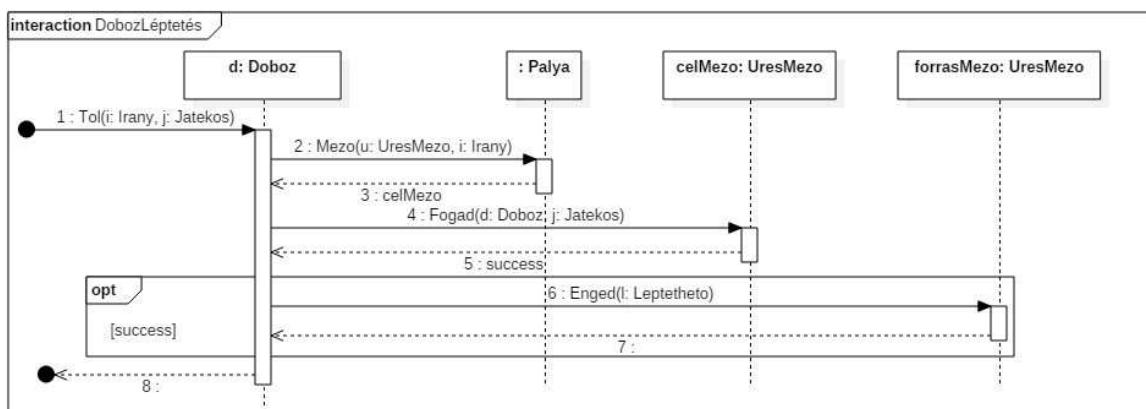
□ Metódusok

- **Fogad(Jatekos j):** Kezeli a játékos mezőre lépését.
- **Fogad(Doboz d, Jatekos j):** Kezeli a doboz mezőre tolását. A játékos referencia ahhoz kell, amikor egy sor megtolt doboz végén egy jelölt doboz pontot akar adni a játékosnak.
- **Enged(Leptetheto l):** Elengedi a léptethető objektumot, ha az más mezőre került.

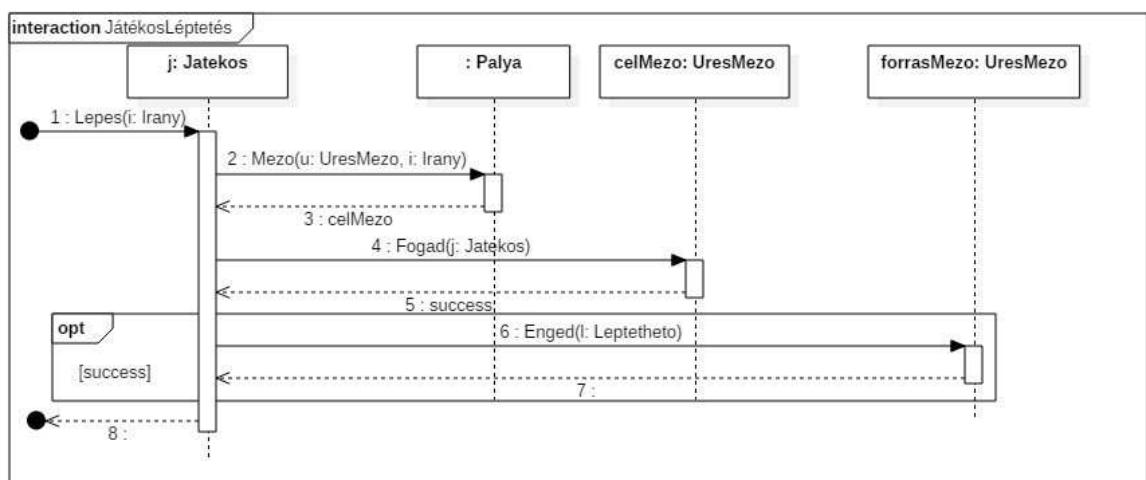
3.4 Szekvencia diagramok

3.4.1 Játék vezérlő logikájának szekvencia diagramjai

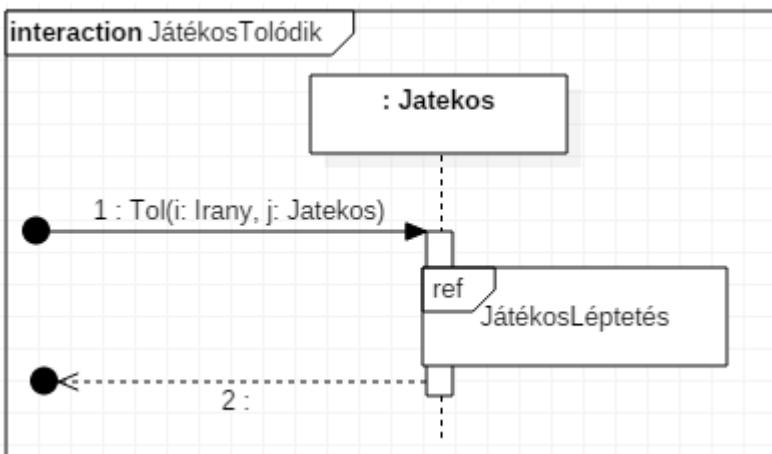
3.4.1.1 Doboz Léptetése



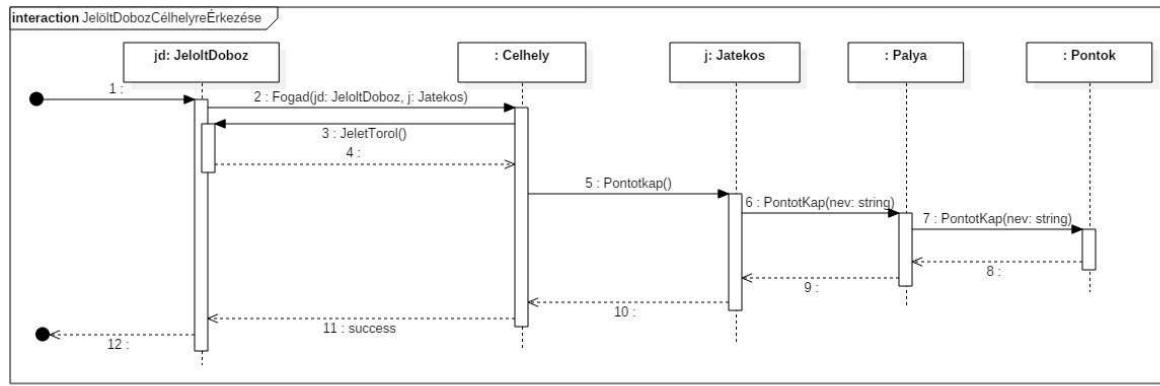
3.4.1.2 Játékos Léptetése



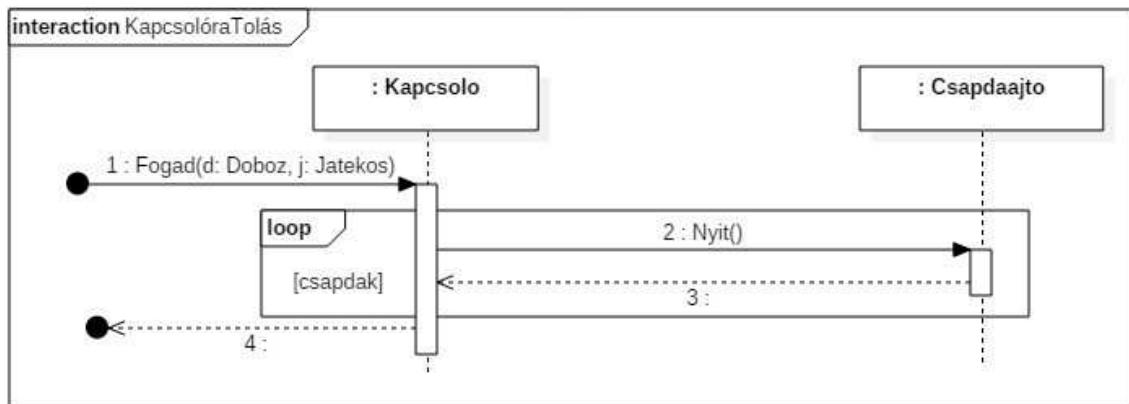
3.4.1.3 Játékos Tolódik



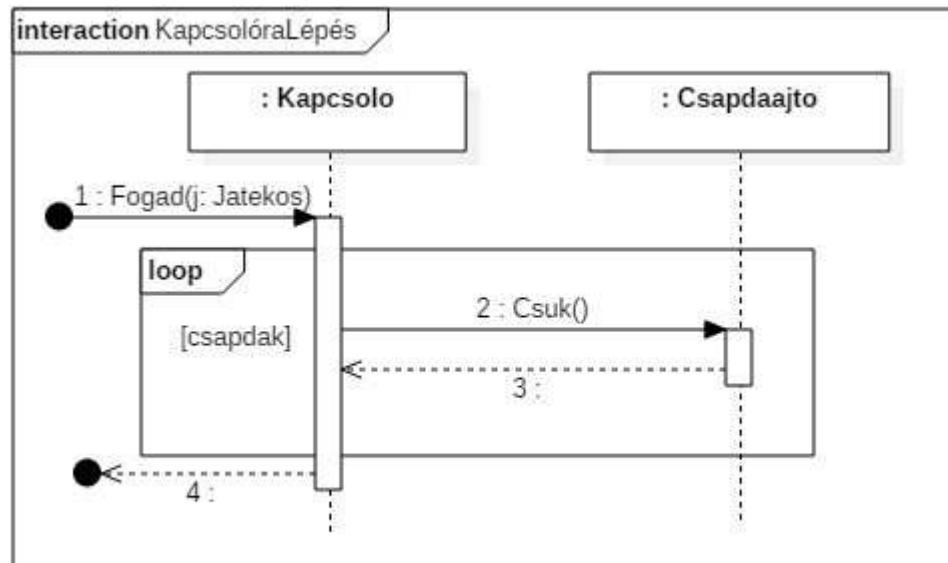
3.4.1.4 Jelölt Doboz Célhelyre Érkezése



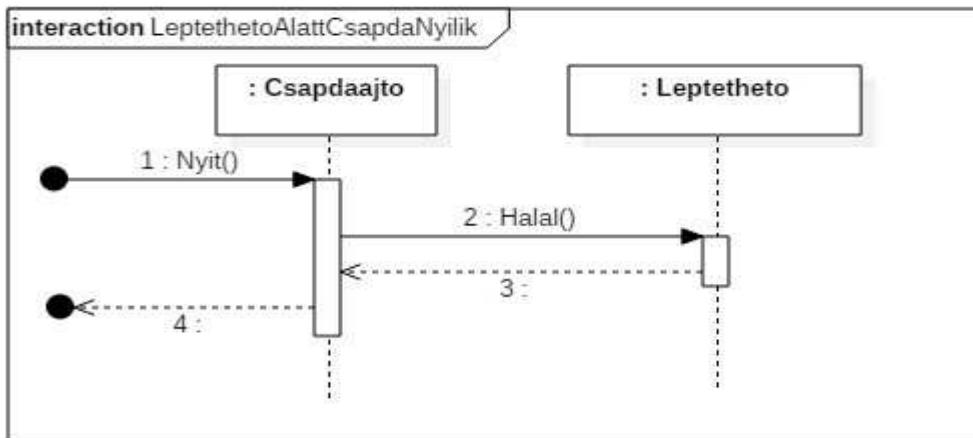
3.4.1.5 Kapcsolóra Tolás



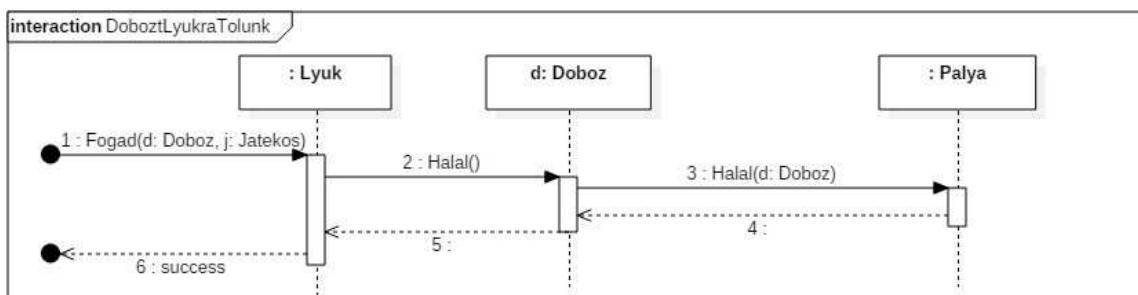
3.4.1.6 Kapcsolóra Lépés



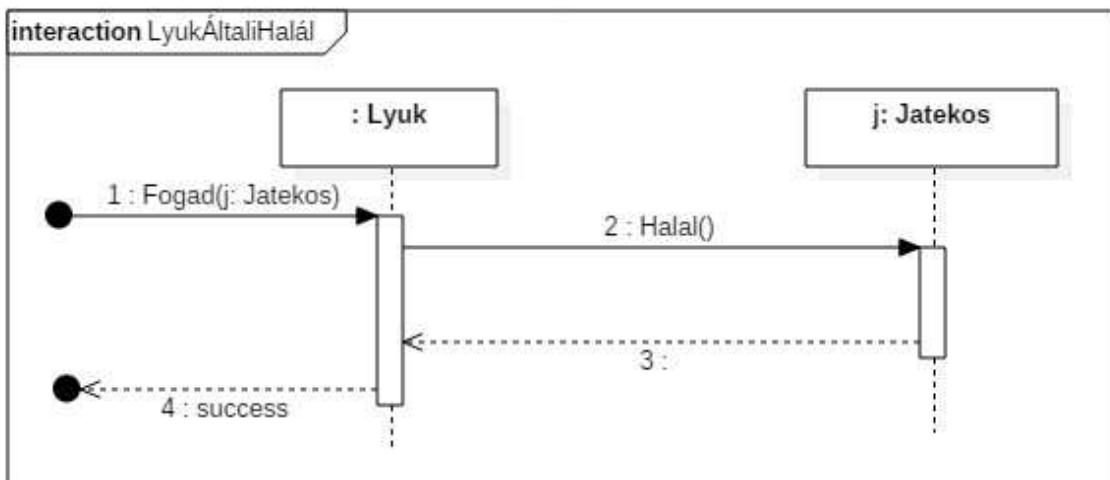
3.4.1.7 Léptethető Alatt Csapda Nyílik



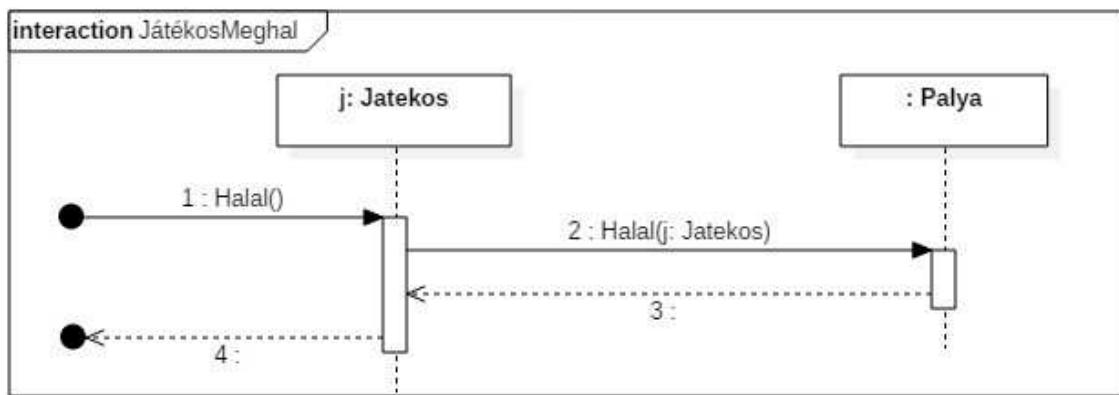
3.4.1.8 Dobozt Lyukra Tolunk



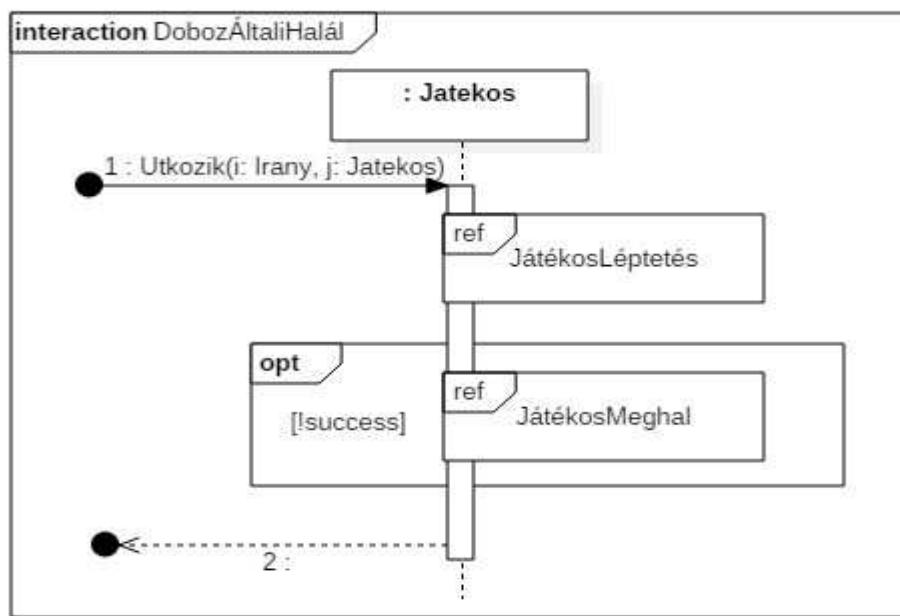
3.4.1.9 Játékos Lyukba Esik



3.4.1.10 Játékos Meghal

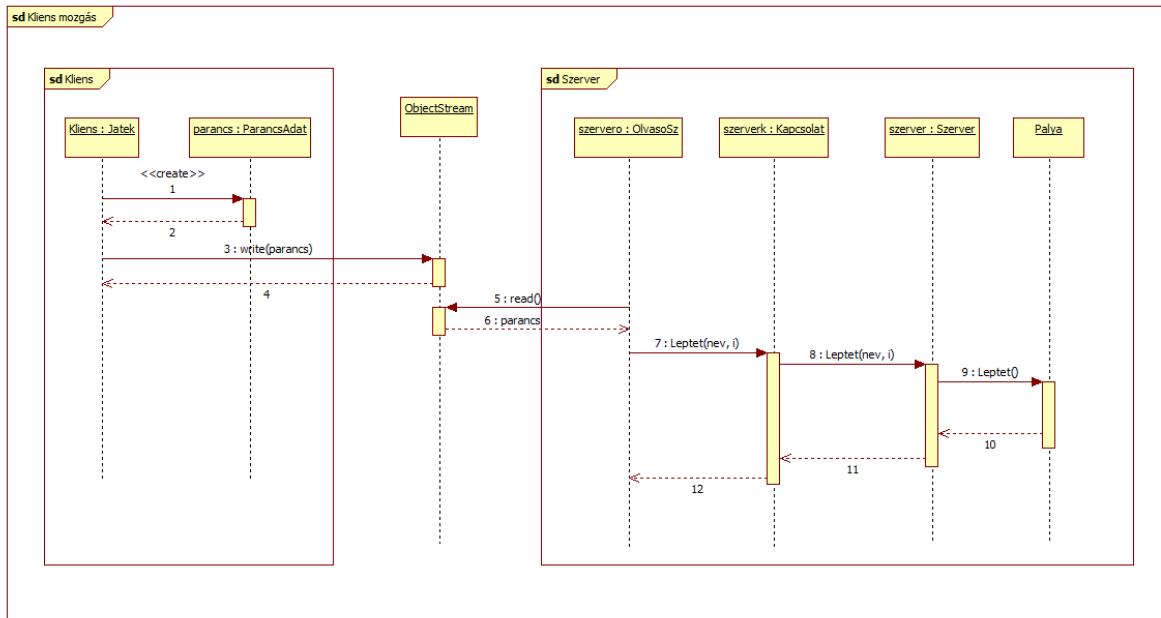


3.4.1.11 Játékost Doboz Össze Nyom

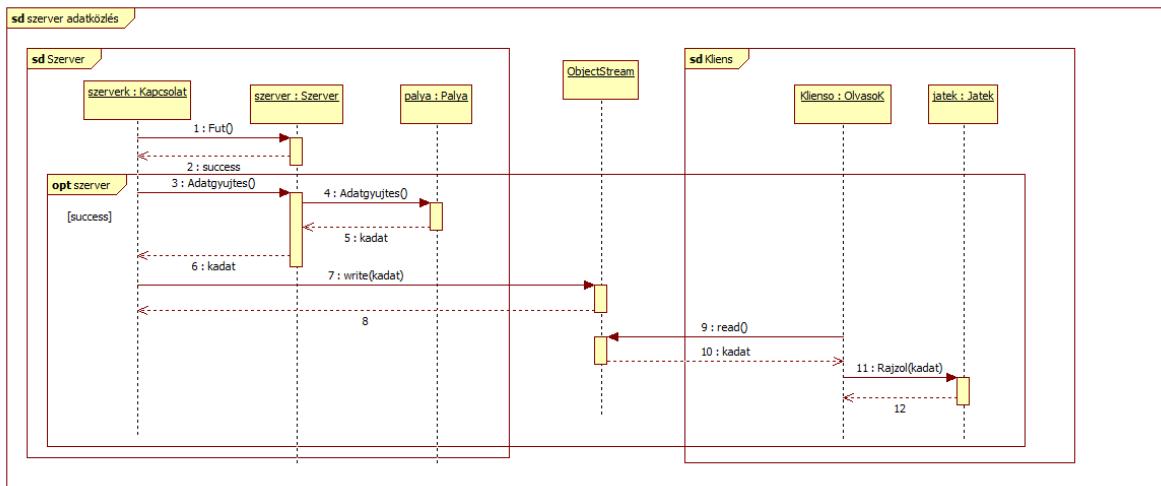


3.4.2 Szerver-Kliens kapcsolatot vezérlő szekvencia diagramok

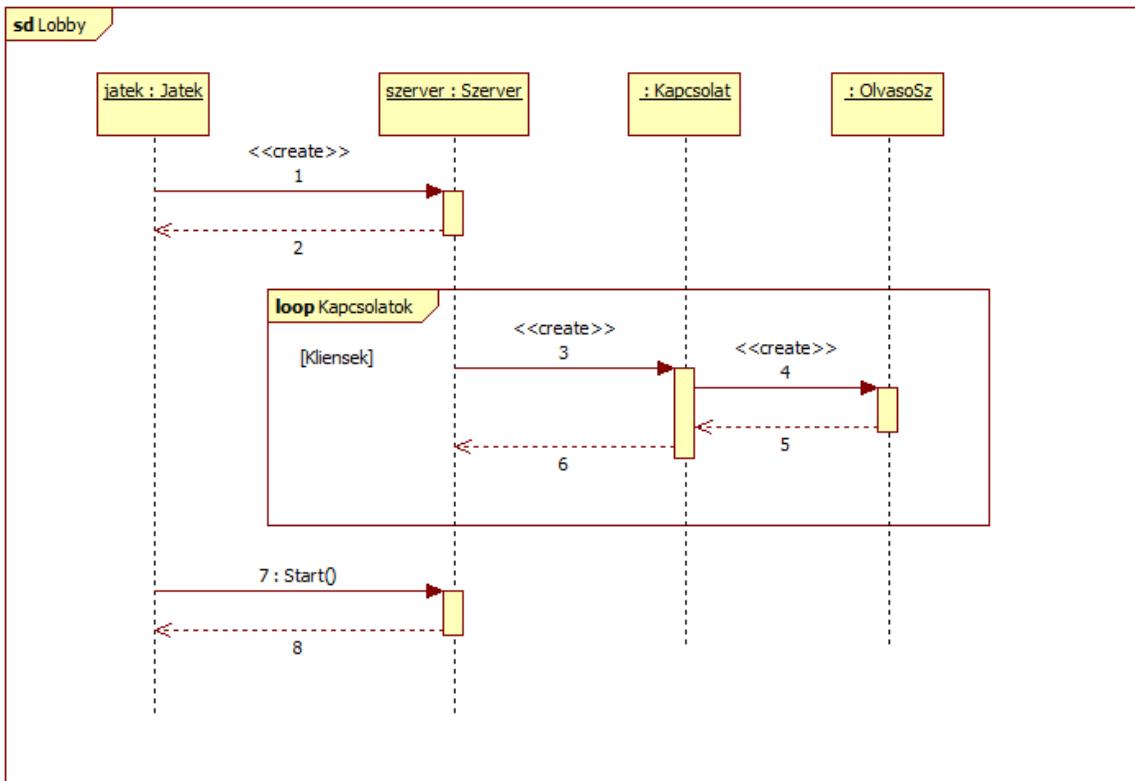
3.4.2.1 Kliens Mozgás



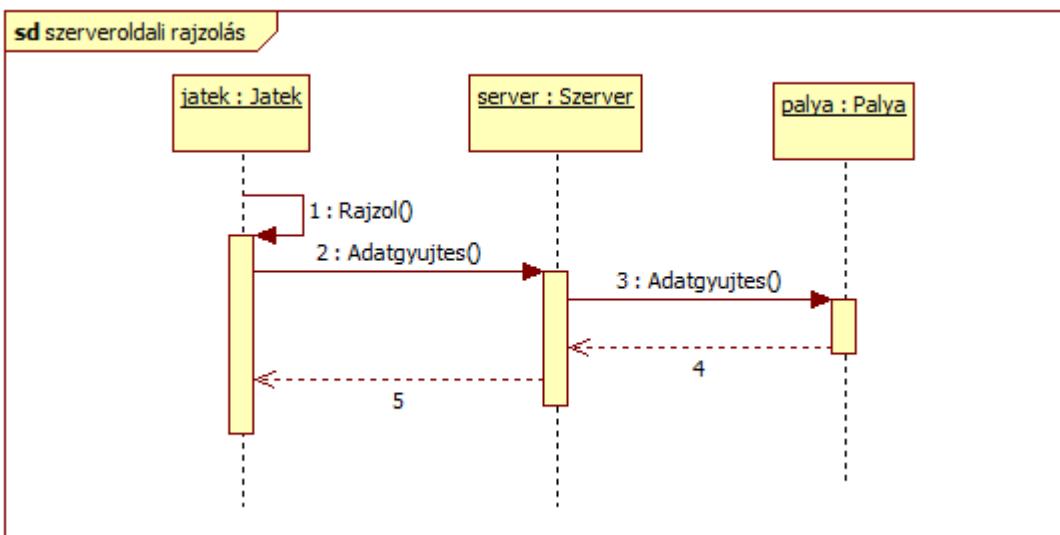
3.4.2.2 Szerver Adatközlés



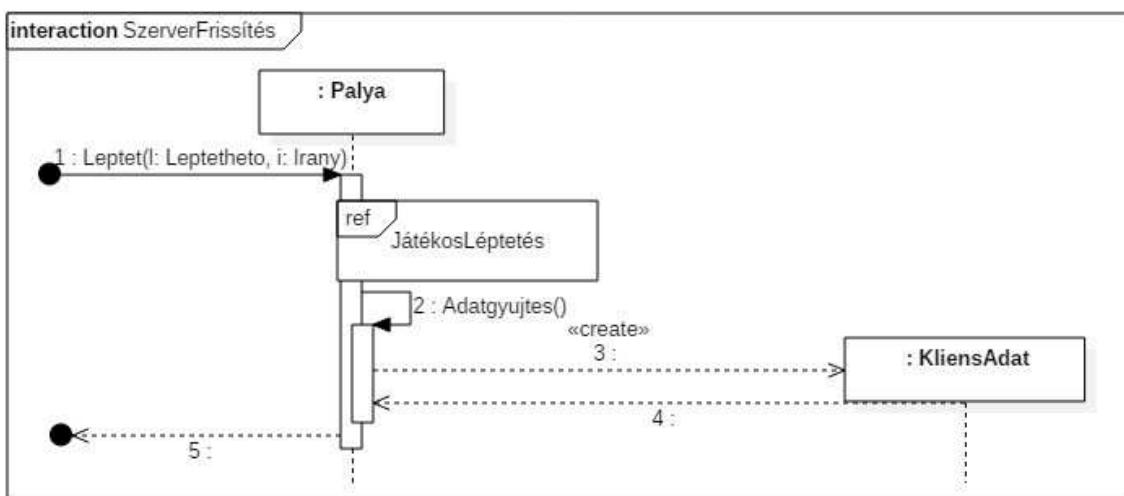
3.4.2.3 Lobby



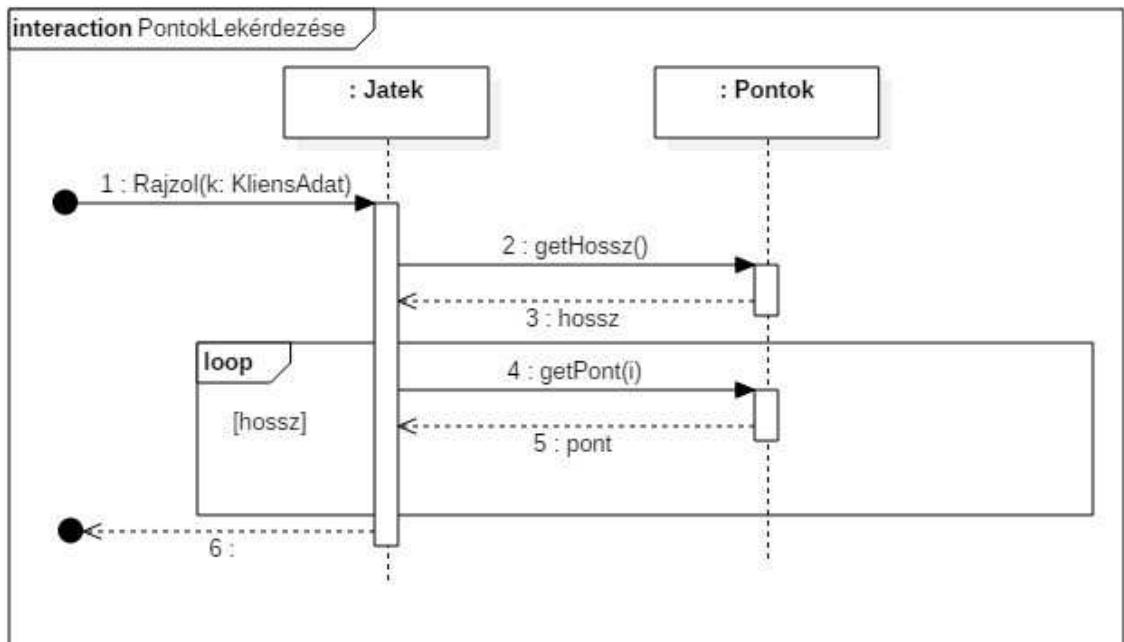
3.4.2.4 Szerveroldali Rajzolás



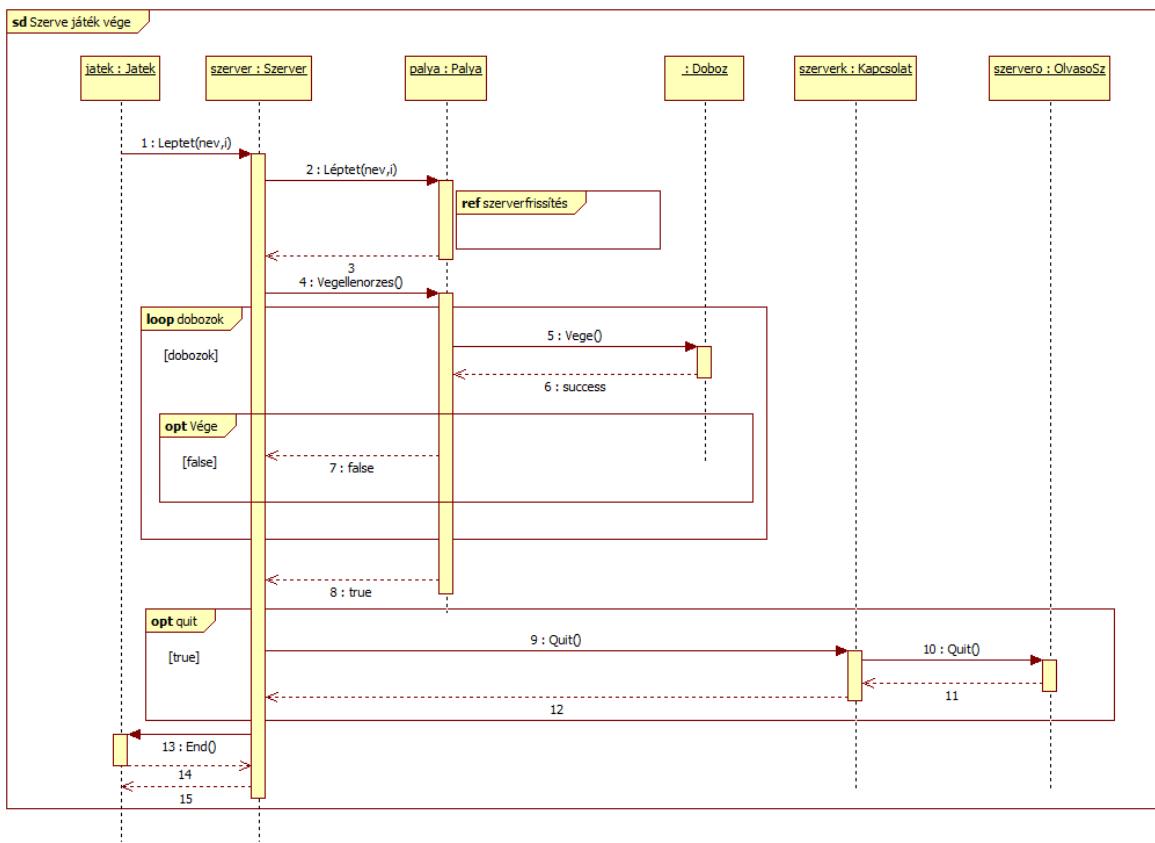
3.4.2.5 Szerver Frissítés



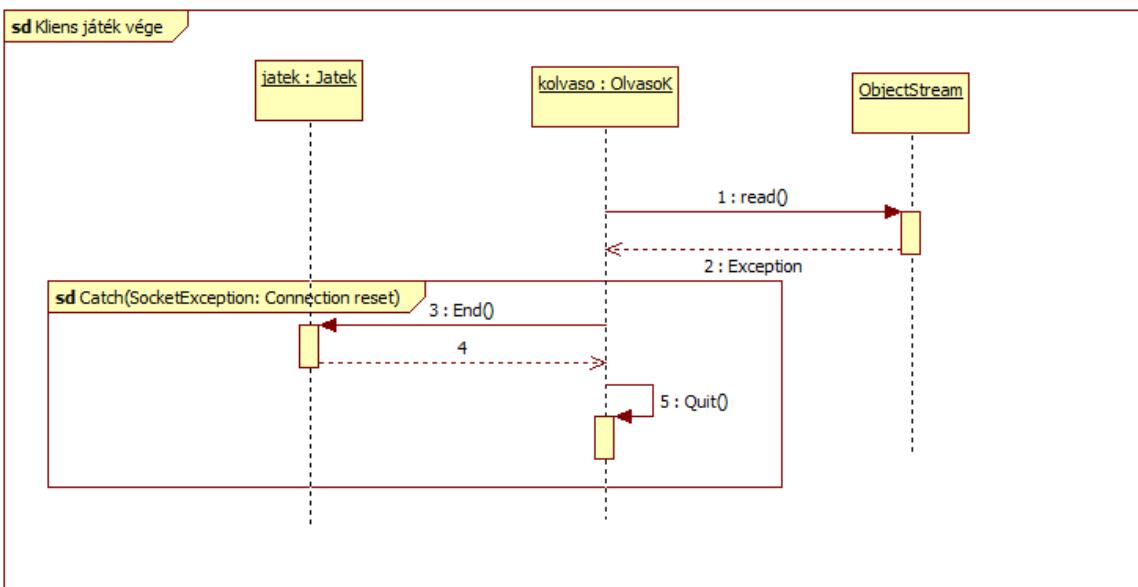
3.4.2.6 Pontok Lekérdezése



3.4.2.7 Szerver Játék Vége

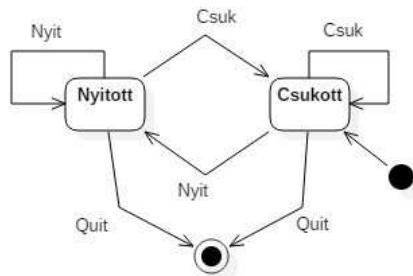


3.4.2.7 Kliens Játék Vége



3.5 State-chartok

3.5.1 Csapdaajtók állása



3.6 Napló

Kezdet	Időtartam	Résznevők	Leírás
2018.02.21. 8:00	1,5 óra	Bottlik Holub Horváth Mocsári Zsíros	Értekezlet. Döntés: Zsíros elkészíti az osztálydiagramot.
2018.02.21. 12:00	3,5 óra	Zsíros	Tevékenység: Zsíros elkészíti az osztálydiagramot, és vázlatos leírást. (3.1-3.3)
2018.02.24. 13:30	2 óra	Mocsári	Szekvencia diagramok elkészítése
2018.02.24. 15:30	11,5 óra	Mocsári Zsíros	Tevékenység: Mocsári és Zsíros újragondolja és elkészíti az osztálydiagramot, leírást, szekvencia diagramokat. (3.1-3.5)
2018.02.25. 12:10	3.5 óra	Bottlik Horváth Mocsári Zsíros	Dokumentum átnézése, kérdéses esetek megvitása

4. Analízis modell kidolgozása

4.1 Objektum katalógus

4.1.1 Celhely

A pálya egy speciális mezője. Olyan üres mező, amely meg van jelölve addig, amíg jelölt látát nem tolnak rá. Ha jelölt láda érkezik rá, sima üres mezővé válik, a rátolt láda is elveszíti jelölését, és a játékos, aki a látát rátolta, pontot kap. A játék célja ezekre tolni a jelölt látákat.

4.1.2 Csapdaajto

A pálya egy olyan mezője, aminek viselkedése egy hozzárendelt kapcsoló állásától függ. Ha zárva van, akkor üres mezőként, ha nyitva, akkor lyukként működik.

4.1.3 Doboz

A pálya mozgatható eleme, ami csak a játékosok interakciója esetén mozdul el. Csak üres mezőkön, és üres mezőként viselkedő mezőkön képes mozogni. Lyukra érve eltűnik. A doboz pontosan egy mezőnyi méretű, és összenyomhatatlan. A dobozok közvetítik a játékosok által kifejtett erőt, azaz ha egy doboz mögött másik doboz(ok) van(nak), vagy másik játékos, azokat is eltolja a játékos az első dobozzal együtt. Ha másik játékos tollánk el egy doboz segítségével, de már nincs a másik játékos mögött olyan mező, amire léphet, akkor az a játékos meghal, és a helyére kerül a doboz.

4.1.4 Fal

A pálya olyan mezője, amire nem lehet lépni vagy dobozt mozgatni. A pályát is fal határolja, hogy a mozgásra képes objektumok ne hagyassák el a pálya területét. A fal képezi az akadályokat is a pályán.

4.1.5 Jatek

A program azon része, ami kezeli a felhasználó utasításait, és továbbítja azokat a megfelelő objektumok felé. Ezzel indíthatunk új játékot, vagy állíthatjuk le a folyamatban lévőt.

4.1.6 Jatekos

A raktári dolgozó, aki a felhasználót reprezentálja a pályán, a pálya egyik eleme. A felhasználó csak ezt képes irányítani, ezzel kell a játékot játszania. Képes mozogni, és dobozokat tolni a pályán üres mezőként viselkedő mezőkön. A feladata, hogy minél több jelölt doboz a célhelyekre juttasson, ezekért pontot kap. Képes meghalni, ekkor veszít az őt irányító felhasználó.

4.1.7 JeloltDoboz

Speciális doboz, amit a játékosoknak a célhelyre kell tolnia. Ugyanúgy működik, mint egy normális doboz, de célhelyre tolva normál dobozzá változik, és az őt odajuttató játékos pontot kap. Ha elfogynak a jelölt dobozok, vagy lehetetlenné válik az összes jelölt doboz a célhelyekre juttatni, akkor a játék véget ér.

4.1.8 Kapcsolo

A pálya egy eleme, amihez tartozik csapdaajtó. Ha egy doboz áll a kapcsolón, akkor a hozzákapcsolt csapdaajtó kinyílik, és lyukként működik. Ha nem áll rajta doboz, akkor zárva van a hozzá tartozó csapdaajtó. Játékosra nem reagál, kizárolag dobozra.

4.1.9 Lyuk

A pálya egy mezője. Ha játékos lép rá, meghal, és veszít. Ha ládát tolunk rá, a láda eltűnik.

4.1.10 Palya

A pálya tartalmazza a játék elemeit. minden játékos, mező és doboz a pálya része. Falak határolják minden irányból. Ez az objektum tárolja az aktuális játékállást és a játékosok pontjait. Összeköti a játék elemeit.

4.1.11 Pont

Egyetlen játékos pontszámát tárolja a játékos nevével együtt. Ha egy játékos pontot kap, ebben az objektumban nő a pontszám értéke.

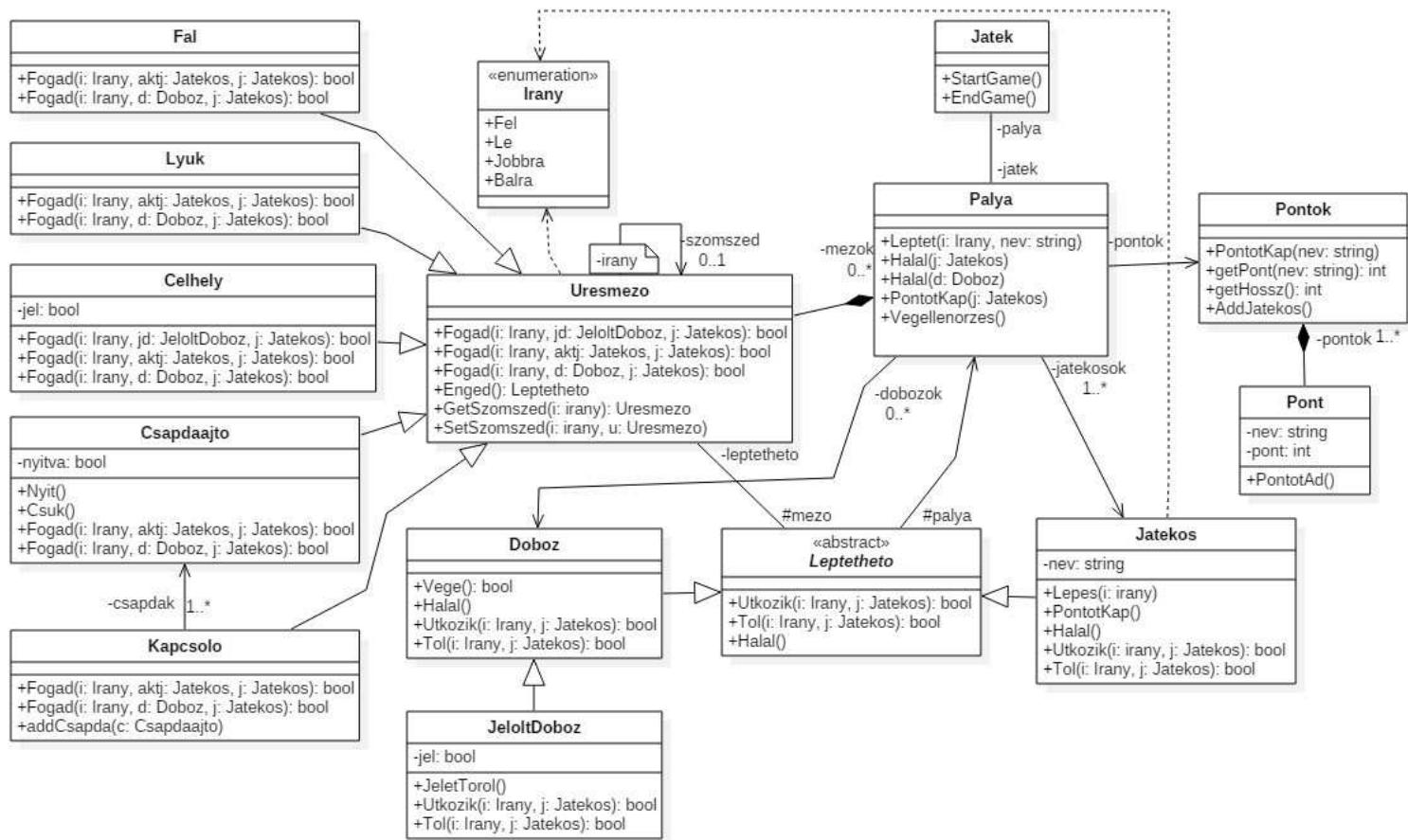
4.1.12 Pontok

Pont objektumokat tárol és kezel, ez tárolja minden játékos pontszámát névvel együtt. Ha egy játékos pontot kap, ez az objektum növeli meg a pontszám értékét a megfelelő helyen.

4.1.13 UresMezo

A pálya egy általános mezője. Ezeken a mezőkön lehetséges mozogni, a játékosok képesek rámenni, ládát rátolni, vagy éppen lemenni róla. Egyetlen üres mezőn egyetlen dolog állhat, vagy 1 játékos, vagy egy doboz, vagy egyik sem, de egyszerre több nem. Ha egy olyan üres mezőre akarunk lépni, amin már doboz áll, akkor a dobozt odébb tolja a játékos, ha képes rá.

4.2 Statikus struktúra diagramok



4.3 Osztályok leírása

4.3.1 Celhely

• Felelősség

A pálya Célhely típusú mezőjét kezeli, figyeli a Célhelyre kerülő objektumokat és végrehajtja az ennek következményeként kialakuló szükséges módosításokat. Ha valaki a Célhelyre tol egy jelölt lánát, pontot kap érte és a Célhely jelöletlenné válik.

• Ősosztályok

Uresmezo → Celhely

• Attribútumok

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

• Metódusok

- **bool Fogad(Irany i, JeloltDoboz jd, Jatekos j)**: Kezeli azt az esetet, amikor valaki jelölt dobozt tol a célhelyre. A lépés sikereségtől függően bool értékkel tér vissza és siker esetén pontot ad az adott játékosnak, aki rátolta a dobozt. Az esetleges továbbtolódás irányát adja meg az „i” paraméter.

- **bool Fogad(Irany i, Jatekos aktj Jatekos j):** Játékos lép a mezőre. A lépést kezdeményező játékos az „aktj”, a ténylegesen a mezőre kerülő játékos a „j”. Ha sikeres a lépés, igaz értékkel tér vissza. Az esetleges továbbtolódás irányát adja meg az „i” paraméter.
- **bool Fogad(Irany i, Doboz d, Jatekos j):** (Jelöletlen) doboz kerül a mezőre. A játékos az, aki a dobozt megtolta. Ha sikeres a lépés, igaz értékkel tér vissza. Az esetleges továbbtolódás irányát adja meg az „i” paraméter.

4.3.2 Csapdaajto

- **Felelősség**

A pálya Csapdaajto típusú mezőjét kezeli, figyeli a Csapdaajto-ra kerülő objektumokat és végrehajtja az ennek következményeként kialakuló szükséges módosításokat. Ha zárva van, akkor üres mezőként, ha nyitva, akkor lyukként viselkedik. Az állása a hozzákötött kapcsolótól függ, de kezdetben minden csapda ajtó zárva van.

- **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(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. A lépést kezdeményező játékos az „aktj”, a ténylegesen a mezőre kerülő játékos a „j”. Az esetleges továbbtolódás irányát adja meg az „i” paraméter.
- **bool Fogad(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. Az esetleges továbbtolódás irányát adja meg az „i” paraméter.

4.3.3 Doboz

- **Felelősség**

A dobozok kezeléséért felelős. Megállapíthatja a mozgathatóságukat, illetve eltüntetheti őket.

- **Ősosztályok**

Leptetheto → Doboz

- **Metódusok**

- **bool Vege():** Ellenörzi, 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(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.

bool Tol(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.

4.3.4 Fal

- **Felelősség**

A pálya Fal típusú mezőjét kezeli. Ez képzi az akadályt a pályán, az osztály figyeli, hogy ne kerülhessen semmi a mezőre.

- **Ősosztályok**

Uresmezo → Fal

- **Metódusok**

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

4.3.5 Irany <>enumeration>

- **Felelősség**

Enum osztály, a mozgási irányokat tárolja.

- **Attribútumok**

- **Fel**
- **Le**
- **Jobbra**
- **Balra**

4.3.6 Jatek

- **Felelősség**

A játék elindításáért és befejeződéséért felelős.

- **Attribútumok**

- **Palya palya**: Referencia a pályára.

- **Metódusok**

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

4.3.7 Jatekos

- **Felelősség**

A játékosok lépéseinak és pontjainak kezeléséért felelős. A felhasználó a játékost irányítva tud játszani, ők a raktári dolgozók, akik a dobozokat tolják a pályán.

- **Ősosztályok**

Leptetheto → Jatekos

- **Attribútumok**

- **string nev:** A felhasználó azonosítója, aki a játékost irányítja.

- **Metódusok**

- **Lepes(Irany i):** A pálya hívja meg, hogy lépesse a játékost, ami meghívja a mező Fogad függvényét.
- **PontotKap():** A jelölt lánkok célhelyre tolásakor a játékos pontot kap. Ez a függvény felelős érte.
- **Halal():** Megöli a játékost.
- **bool Utkozik(irany i, Jatekos j):** A Leptethető interfész-beli 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.
- **bool Tol(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.

4.3.8 JeloltDoboz

- **Felelősség**

A jelölt doboz kezeléséért felelős. A JeloltDoboz-t kell a játékosoknak a célhelyre tolnia, ezután a doboz 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(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.
- **bool Tol(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.

4.3.9 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 látát tol a kapcsolóra, akkor az kinyitja a hozzákapcsolt csapdaajtókat, ha ő lép a kapcsolóra, akkor bezárja.

- **Ősosztályok**

Uresmezo → Kapcsolo

- **Attribútumok**

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

- **Metódusok**

- **bool Fogad(Jatekos j):** Ha játékos lép rá akkor becsukja a csapdaajtókat.
- **bool Fogad(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 ősosztály.
- **addCspda(Csapdaajto c):** Hozzáadja a paraméterként megadott csapdaajtót a kapcsoló által kapcsolható csapdákhoz.

4.3.10 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(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(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, tehát a lépést kezdeményező játékest jelöli.
- **Halal():** Absztrakt függvény a leszármazottak halálához.

4.3.11 Lyuk

- **Felelősség**

A pálya Lyuk típusú mezőjét kezeli, figyeli a Lyuk-ra kerülő objektumokat és végrehajtja az ennek következményeként kialakuló szükséges módosításokat. Olyan mező, amelybe lépve a játékos, vagy láda eltűnik.

- **Ősosztályok**

Uresmezo → Lyuk

- **Metódusok**

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

4.3.12 Palya

- **Felelősség**

Tárolja a mezőket, játékosokat, dobozokat. Összeköti a játék különálló részeit, és előállítja a kirajzoláshoz szükséges adatokat. Ez végzi a játék végellenőrzését is.

- **Attribútumok**

- **Uresmezo[] mezok:** Eltárolja a pályán lévő mezőket egy egydimenziós tömbben.
- **tömbben.**
- **Jatek jatek:** Referencia az aktuális játékra.
- **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.

4.3.13 Pont

- **Felelősség**

A kiosztandó pont megvalósítása egy adott játékosra vonatkozóan.

- **Attribútumok**

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

- **Metódusok**

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

4.3.14 Pontok

- **Felelősség**

A pontok kiosztásáért és tárolásáért felelős osztály.

- **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.

4.3.15 UresMezo

- **Felelősség**

A pálya egy adott mezőjét kezeli, az ottani aktuális történésekre reagál, és elvégzi a szükséges módosításokat, ha kell.

- **Attribútumok**

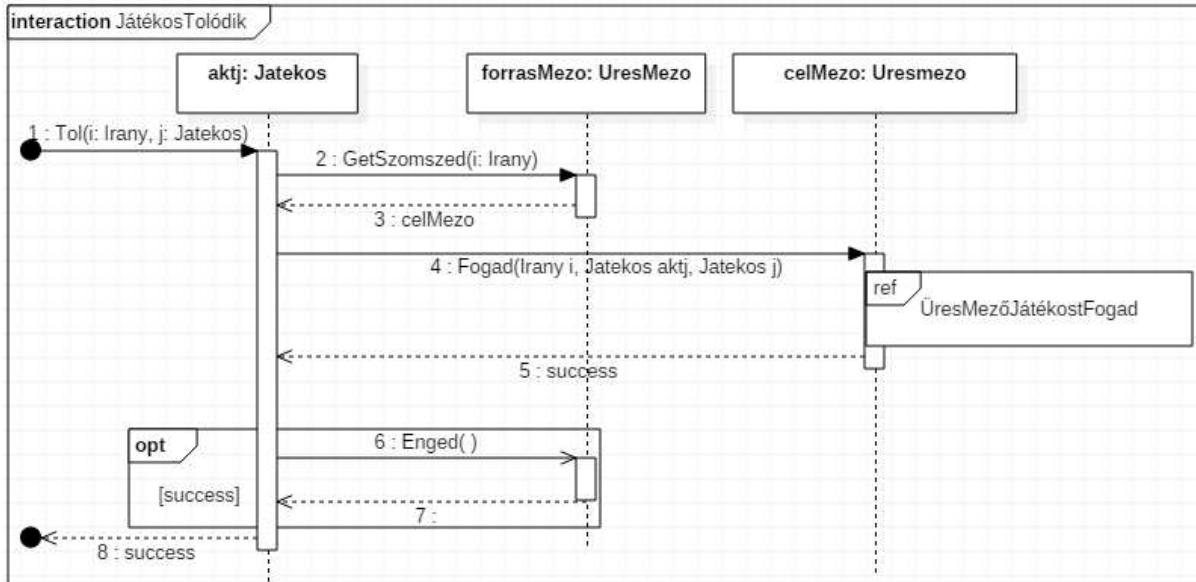
- **UresMezo szomszed[irany]:** Tárolja a szomszéd mezőket a megadott irányokban.
- **Leptetheto leptetheto:** Referencia a mezőn lévő leptethető objektumra

- **Metódusok**

- **bool Fogad(Irany i, JeloltDoboz jd, Jatekos j):** Jelölt doboz tolódik a mezőre a megadott játékos által a megadott irányba. Ha sikeres a lépés, igaz értékkel tér vissza.
- **bool Fogad(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(Irany i, JeloltDoboz 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 és visszaadja a leptethető 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.

4.4 Szekvencia diagramok

4.4.1 JátékosTolódik

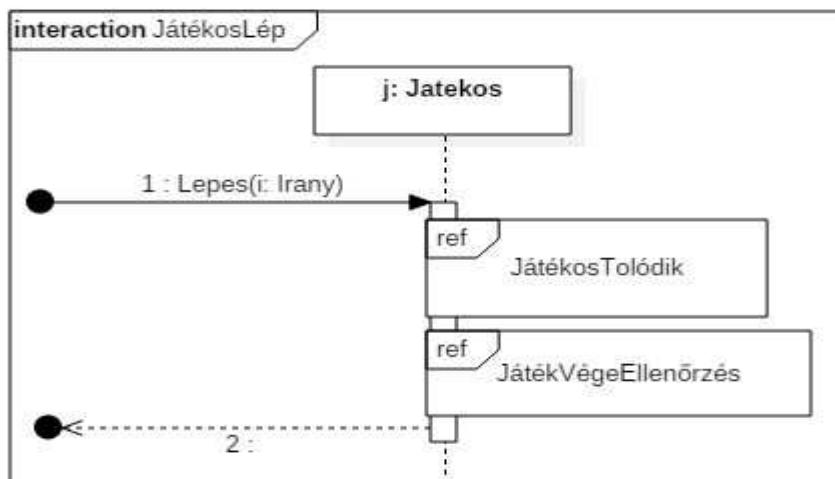


Ezen a szekvencián látható, ahogy egy játékosot egy másik játékos elmozdít. Először lekérdezzük a célmezőt a forrásmezőtől, majd megpróbálunk rá átlépni. Ha ez sikerül, akkor lelépünk az aktuális mezőnkről. A lépésünk sikereségét visszaadjuk a hívónak.

Referált szekvencia diagramok:

ÜresMezőJátékostFogad: 4.4.6

4.4.2 JátékosLép



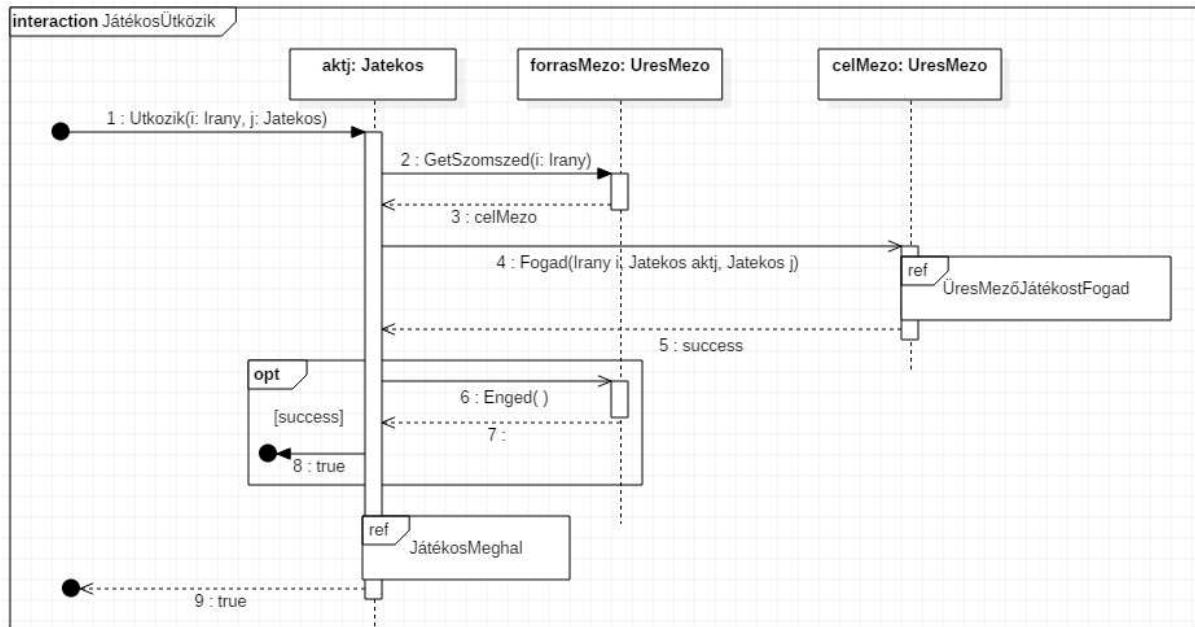
Ezen a szekvencián látható a játékos saját maga által kezdeményezett mozgása. A mozgás része identikus a Tol-lal. Lépés után a játék végének ellenőrzését végezzük.

Referált szekvencia diagramok:

JátékosTolódik: 4.4.1

JátékVégeEllenőrzés: 4.4.17

4.4.3 JátékosÜtközik

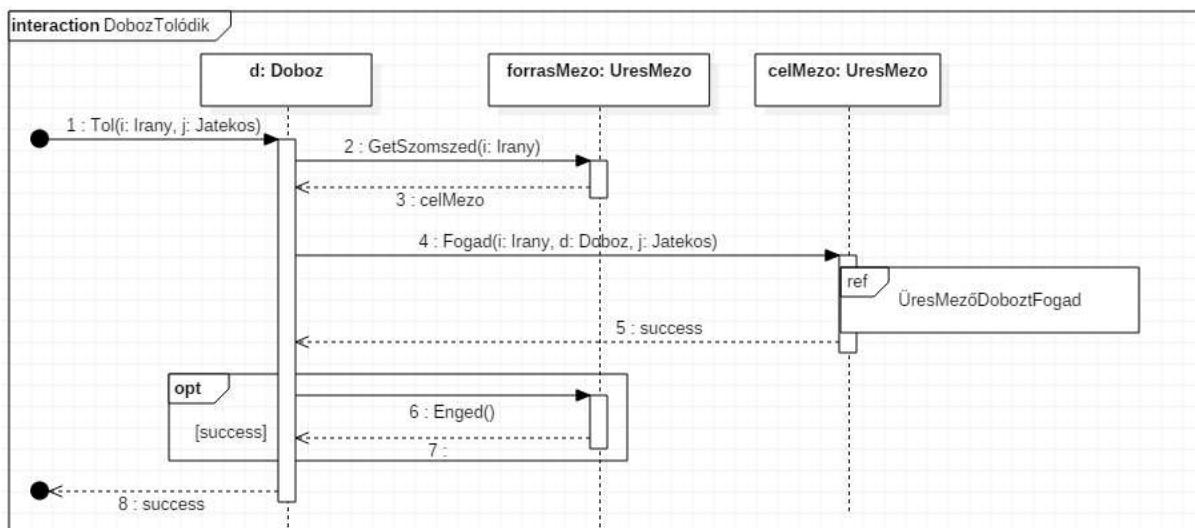


Ezen a szekvencián látható, ahogy egy játékos egy doboz elmozdít. A játékos megpróbáljuk eltolni egyenesen, vagy valamelyik oldalra, ha ez nem sikerül, akkor a játékos meghal.

Referált szekvencia diagramok:

JátékosMeghal: 4.4.15

4.4.4 DobozTolódik

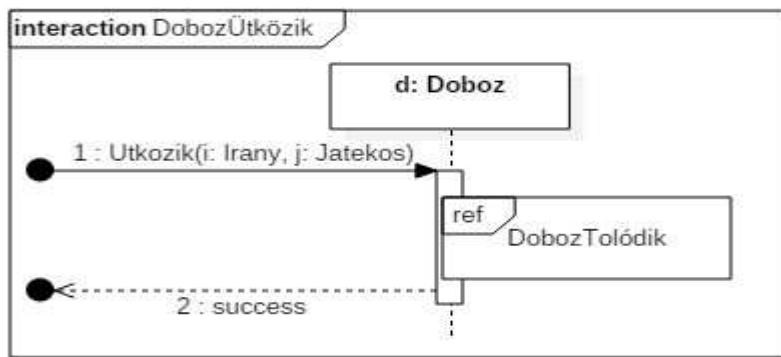


Ebben a szekvenciában látható, ahogy egy dobozt egy játékos elmozdít.

Referált szekvencia diagramok:

ÜresMezőDoboztFogad: 4.4.7

4.4.5 DobozÜtközik

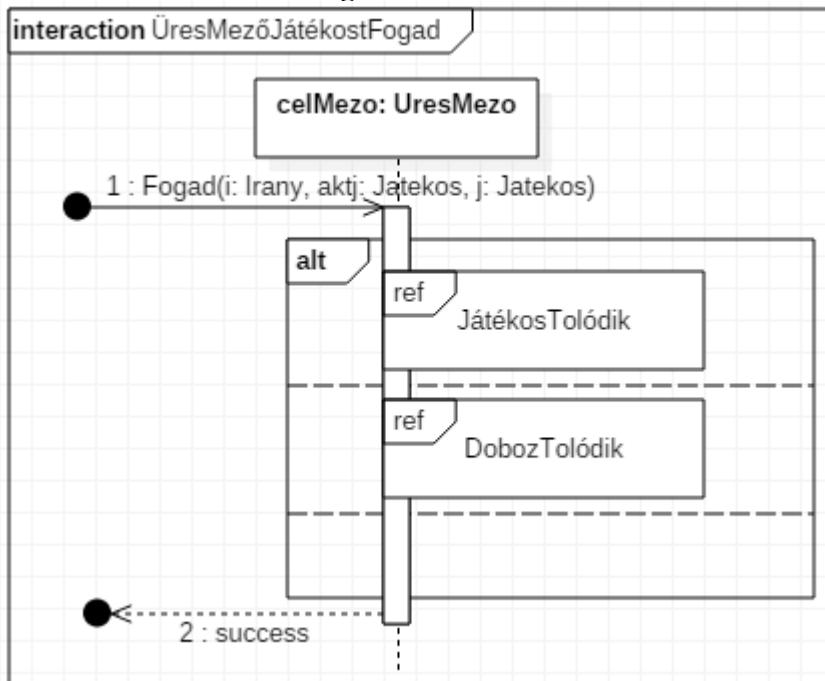


Ezen a szekvencián látható, ahogy egy dobozt egy másik doboz elmozdít. A folyamat identikus azzal, ahogyan egy dobozt játékos elmozdít.

Referált szekvencia diagramok:

DobozTolodik: 4.4.4

4.4.6 ÜresMezőJátékostFogad¹²



Ezen a szekvencián látható az a rész, amely minden mezőre lépéskor lezajlik. Meghívódik a rajta levő Leptetheto Tol metódusa (ha van rajta).

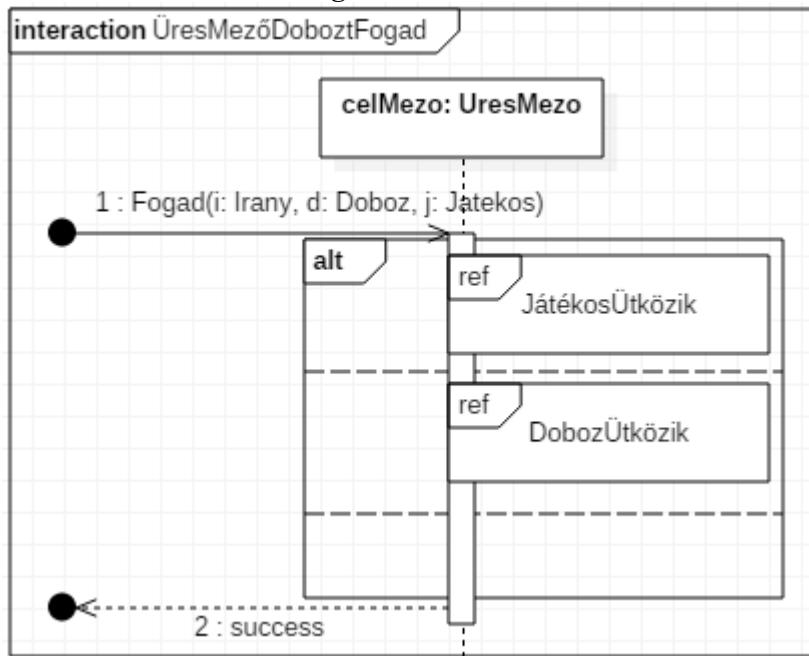
Referált szekvencia diagramok:

JátékosTolodik: 4.4.1

DobozTolodik: 4.4.4

¹¹ Az Jatekos-/Doboz-ra osztás csak szemléltetés végett szerepel. A Fogad metódus egy Leptetheto::Tol metódust hív meg, az elágazás az alapján dől el, hogy az UresMezo által tartalmazott Leptetheto Jatekos, vagy Doboz leszármazott-e.

¹² Azért szerepel még egyszer a Fogad metódus, mivel ez az ősosztály fogad metódusának hívása

4.4.7 ÜresMezőDoboztFogad³⁴

Ezen a szekvencián látható az a rész, amely majdnem minden mezőre lépéskor lezajlik. Meghívódik a rajta levő Leptetheto Utcozik metódusa (ha van rajta).

Referált szekvencia diagramok:

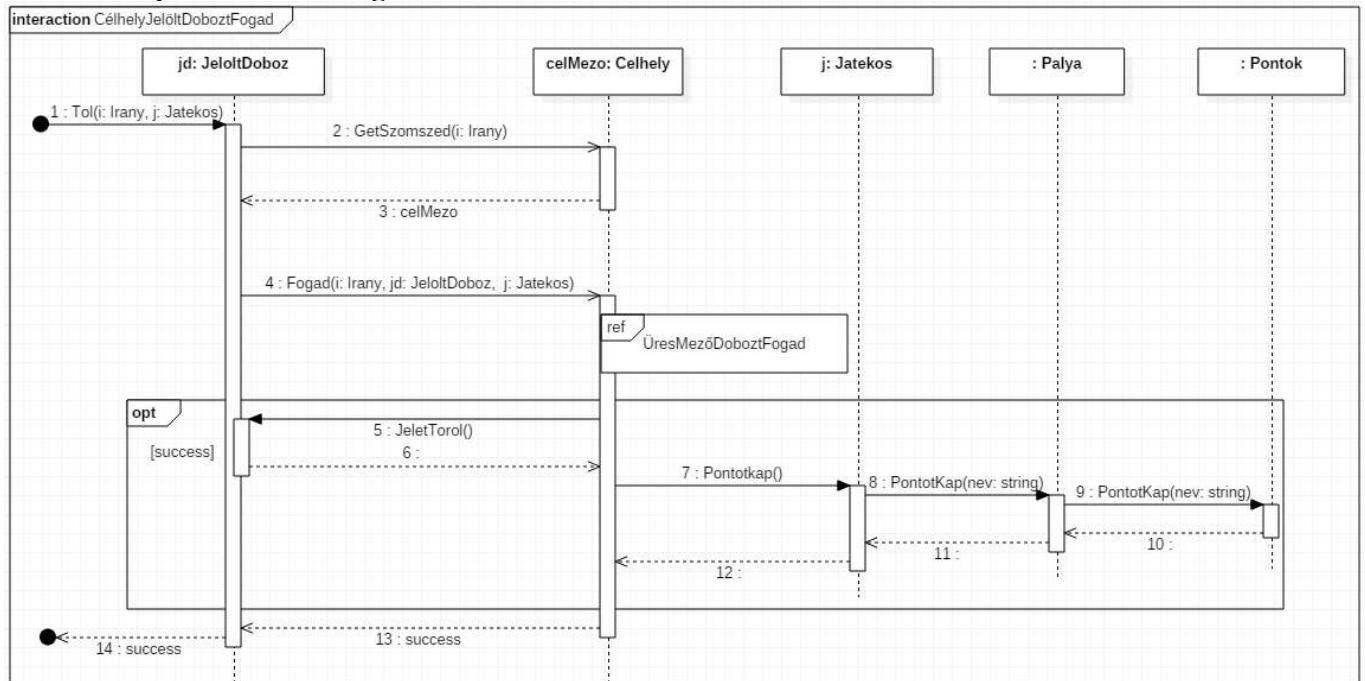
JátékosÜtközik: 4.4.3

DobozÜtközik: 4.4.5

³⁴ Az Jatekos-/Doboz-ra osztás csak szemléltetés végett szerepel, A Fogad metódus egy Leptetheto::Utcozik metódust hív meg, az elágazás az alapján döl el, hogy az UresMezo által tartalmazott Leptetheto Jatekos, vagy Doboz leszármazott-e.

4 Azért szerepel még egyszer a Fogad metódus, mivel ez az ősosztály fogad metódusának hívása

4.4.8 CélhelyJelöltDoboztFogad

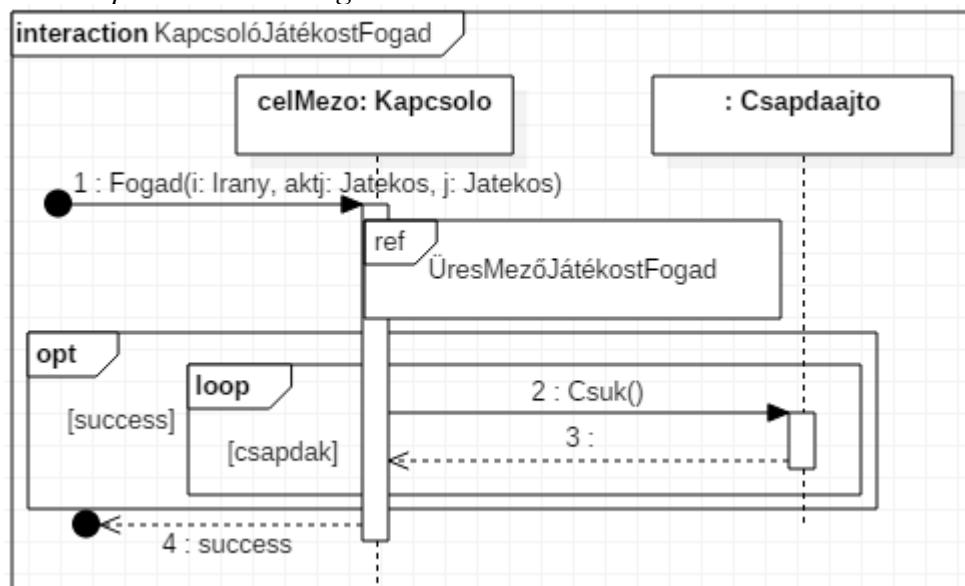


Ezen a szekvencián látható az a rész, amikor egy JelöltDoboz Celhely-re érkezik. Ha az üres mezőkre jellemző műveletek lezajlanak, törlődik a doboz jelölése, és pontot osztunk a megfelelő játékosnak. A sikerességet jelezük a hívónak.

Referált szekvencia diagramok:

ÜresMezőDoboztFogad: 4.4.7

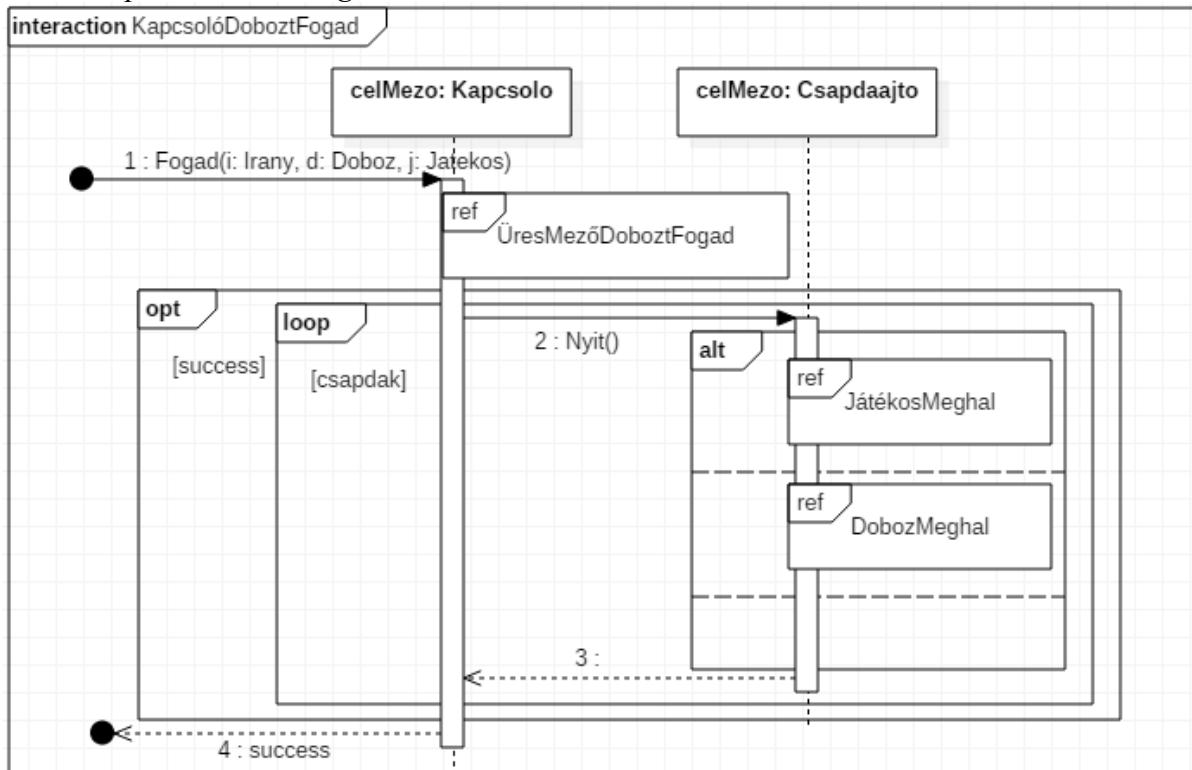
4.4.9 KapcsolóJátékostFogad



Ezen a szekvencián látszik az a folyamat, amikor egy Jatekos egy Kapcsolo-ra érkezik. Ha sikeresen lezajlott az Uresmezo-re általánosan jellemző folyamat, akkor minden, a Kapcsolo-hoz tartozó Csapdaajto Csuk metódusa meghívódik. A sikerességet jelezük a hívónak.

Referált szekvencia diagramok:

ÜresMezőJátékostFogad: 4.4.6

4.4.10 KapcsolóDoboztFogad⁵

Ezen a szekvencián látszik az a folyamat, amikor egy Doboz egy Kapcsolo-ra érkezik. Ha az Uresmezo-re általánosan jellemző folyamat sikeresen lezajlik, akkor minden, a Kapcsolo-hoz tartozó Csapdaajtó Nyit metódusa meghívódik, és ha bármely csapdaajtón található Jatekos, vagy Doboz, azok meghalnak. A sikerességet jelezzük a hívónak.

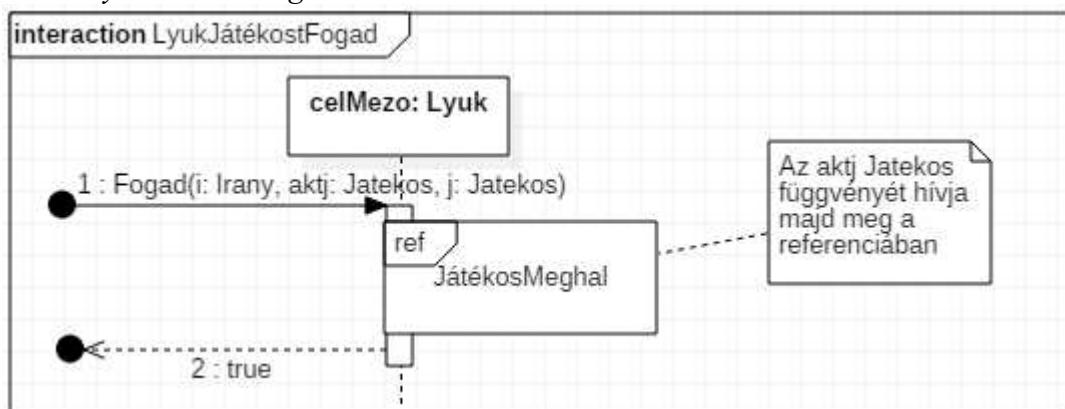
Referált szekvencia diagramok:

ÜresMezőDoboztFogad: 4.4.7

JátékosMeghal: 4.4.15

DobozMeghal: 4.4.16

4.4.11 LyukJátékostFogad



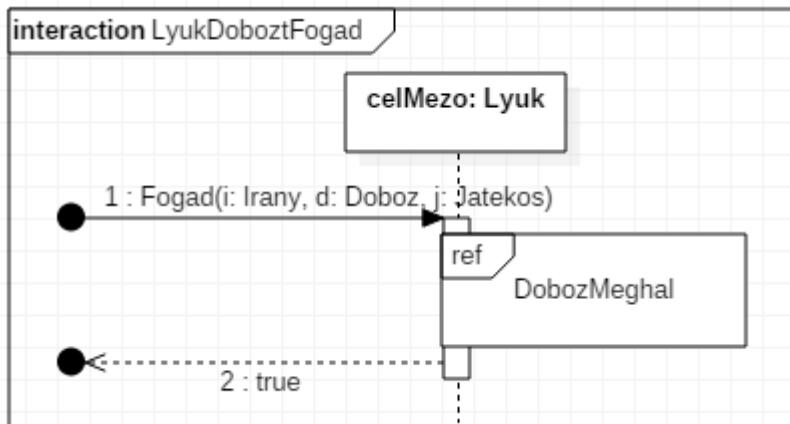
Ezen a szekvencián látszik az a folyamat, amikor egy Jatekos egy Lyuk-ra érkezik. Itt nem hívódik meg az Uresmezo-re általánosan jellemző folyamat, mert minden esetben meghal a Jatekos és true-t jelez vissza a hívónak. A JátékosMeghal szekvenciában az „aktj” játékosra hívjuk a referenciaiban szereplő függvényt, mert ő kerül ténylegesen a lyukba, a „j” játékos csak az, aki a tolást/lépést kezdeményezte.

⁵ Az Jatekos-/Doboz-ra osztás csak szemléltetés végett szerepel. A metódus egy Leptetheto::Halal() metódust hív meg, az elágazás az alapján dől el, hogy az UresMezo által tartalmazott Leptetheto Jatekos, vagy Doboz leszármazott-e.

Referált szekvencia diagramok:

JátékosMeghal: 4.4.15

4.4.12 LyukDoboztFogad

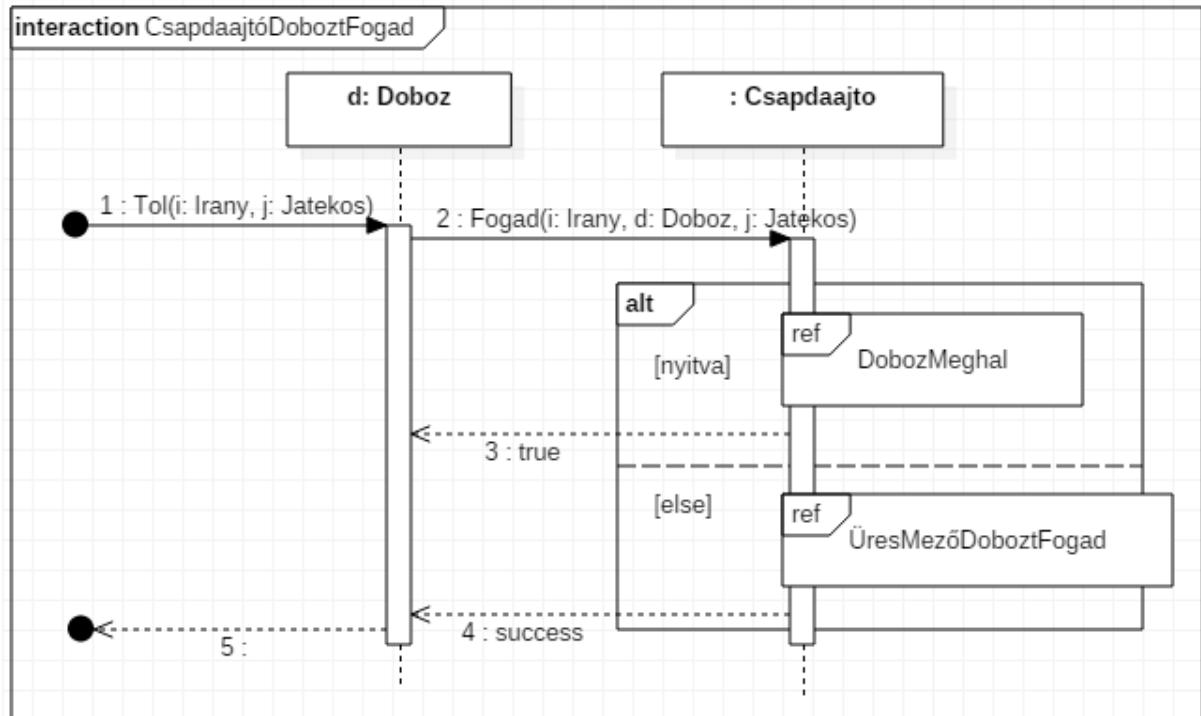


Ezen a szekvencián látszik az a folyamat, amikor egy Doboz egy Lyuk-ra érkezik. Itt nem hívódik meg az Uresmezo-re általánosan jellemző folyamat, mert minden esetben meghal a Doboz és true-t jelez vissza a hívónak.

Referált szekvencia diagramok:

DobozMeghal: 4.4.16

4.4.13 CsapdaajtóDoboztFogad



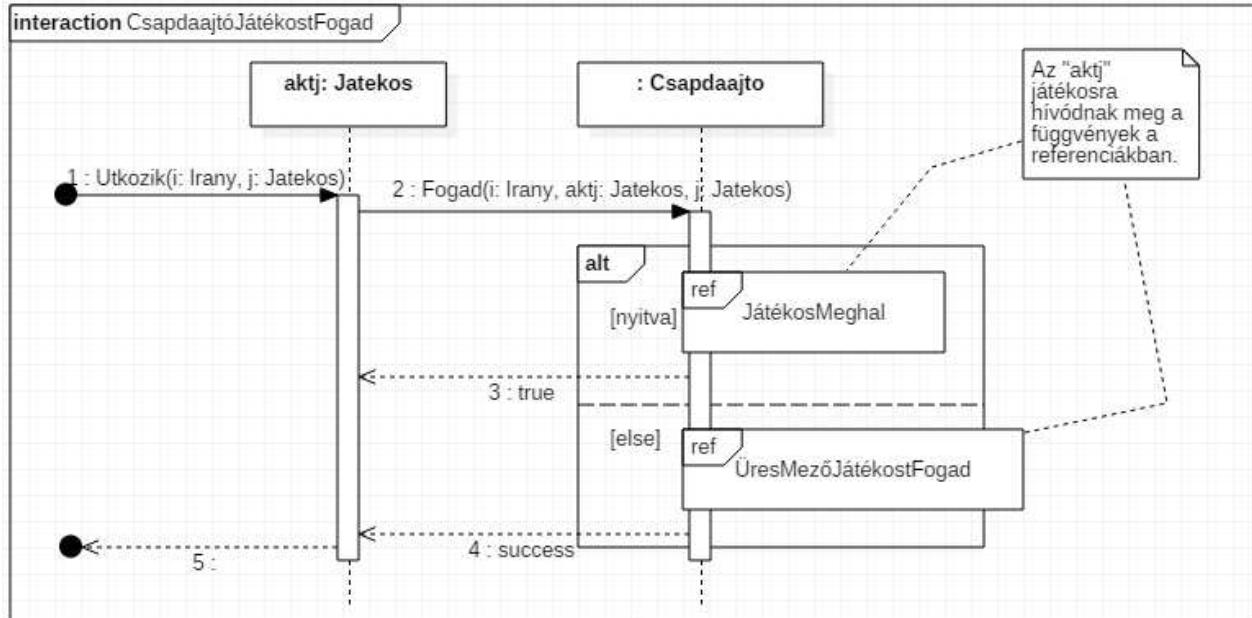
Ezen a szekvencián látszik az a folyamat, amikor egy Doboz Csapdaajtó-ra mozog. Ha nyitva van, a Doboz meghal, és true-val térünk vissza. Csak ha nincs nyitva, értékelődik ki az Uresmezo-re általánosan jellemző folyamat. Ekkor a sikerességet jelezzük a hívónak.

Referált szekvencia diagramok:

DobozMeghal: 4.4.16

ÜresMezőDoboztFogad: 4.4.7

4.4.14 CsapdaajtóJátékostFogad



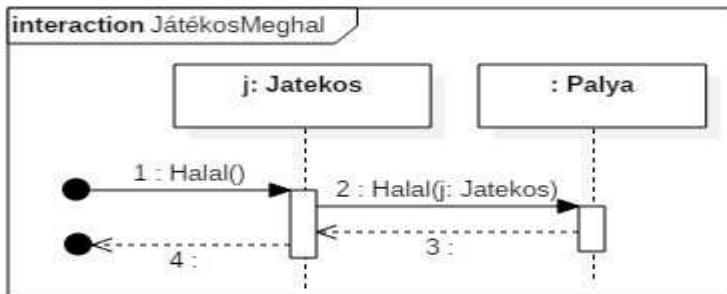
Ezen a szekvencián látszik az a folyamat, amikor egy Doboz Csapdaajtó-ra mozog. Ha nyitva van, a Doboz meghal, és true-val térünk vissza. Csak ha nincs nyitva, értékelődik ki az Uresmezo-re általánosan jellemző folyamat. Ekkor a sikerességet jelezzük a hívónak.

Referált szekvencia diagramok:

JátékosMeghal: 4.4.15

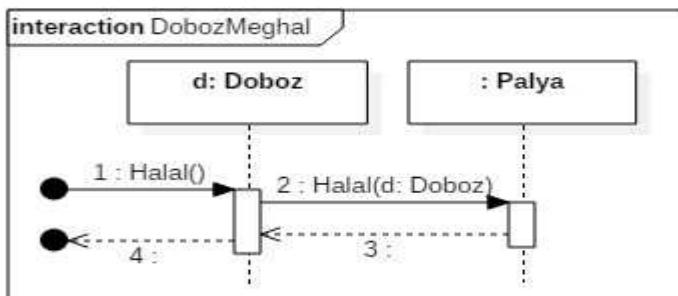
ÜresMezőJátékostFogad: 4.4.6

4.4.15 JátékosMeghal



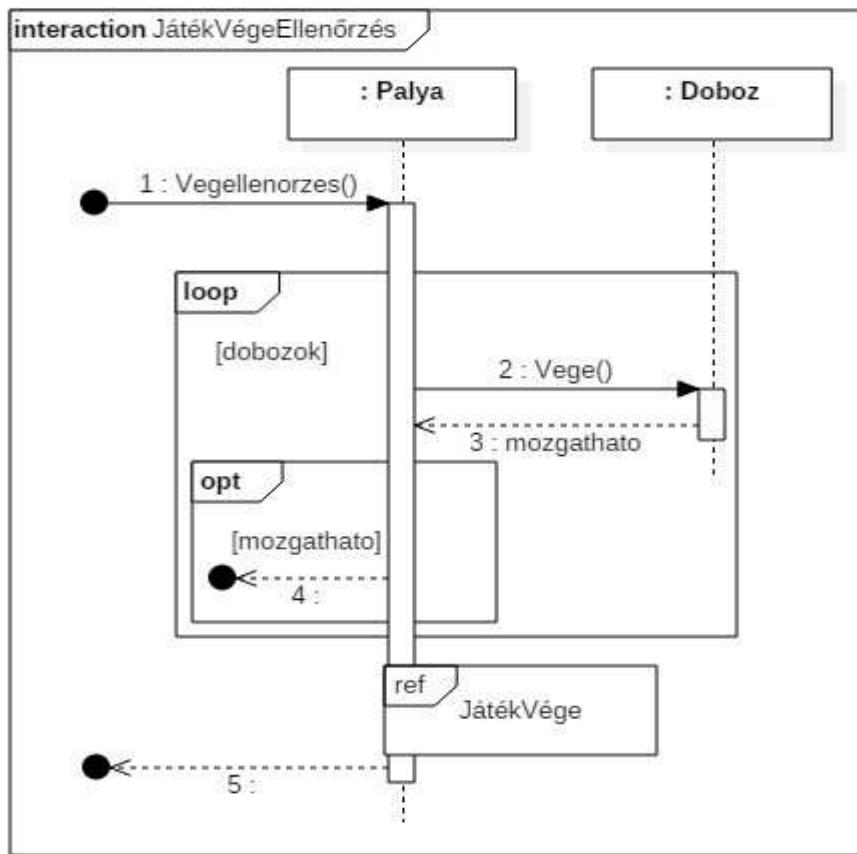
Ezen a szekvencián látszik az a folyamat, amikor egy Jatekos meghal. Jelzi a Palya-nak, hogy meghalt és hívódjon meg a neki megfelelő Halal metódusa.

4.4.16 DobozMeghal



Ezen a szekvencián látszik az a folyamat, amikor egy Doboz meghal. Jelzi a Palya-nak, hogy meghalt és hívódjon meg a neki megfelelő Halal metódusa.

4.4.17 JátékVégeEllenőrzés

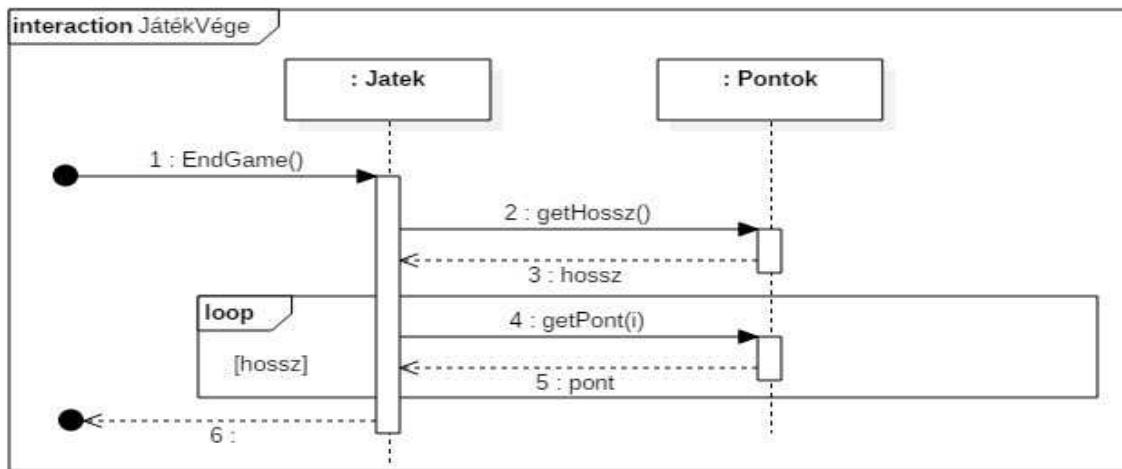


Ezen a szekvencián látszik az a folyamat, amikor a játék végét ellenőrzi a rendszer. A játék végének egyik lehetséges esete, hogy nincs mozgatható doboz. E végett a Palya lekérdezi minden doboztól, hogy mozgatható-e, ha talál mozgatható dobozt, akkor visszatér a többi vizsgálata nélkül. Ha megvizsgálta az összeset és nem talált mozgatható dobozt, akkor a játéknak vége.

Referált szekvencia diagramok:

JátékVége: 4.4.18

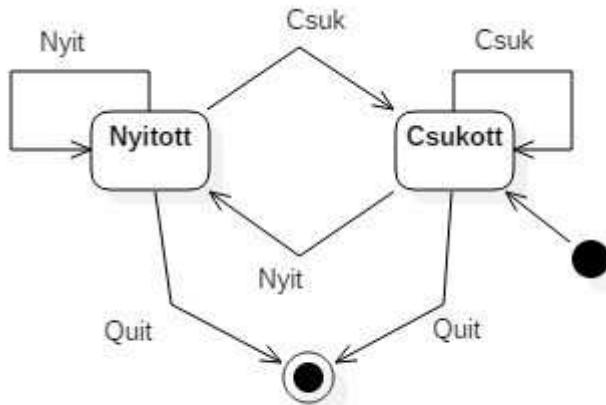
4.4.18 JátékVége



Ez a szekvencia ábrázolja azt a folyamatot, ami a játék végén zajlik le. A Jatek lekérdezi a játékosok szerzett pontjait.

4.5 State-chartok

4.5.1 Csapdaajtók állása



4.6 Napló

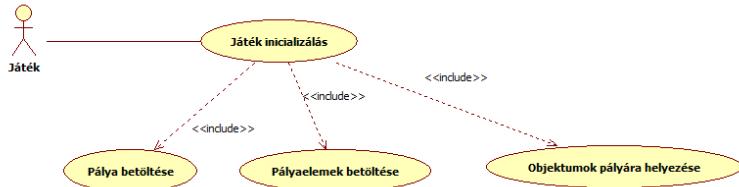
Kezdet	Időtartam	Résztvevők	Leírás
2018.03.02. 11:15	4 óra	Horvath	Tevékenység: Class Diagram áttervezése, és megvalósítása
2018.03.03. 12:30	2 óra	Holub	Holub újra elkészítette az objektumkatalógust a megbeszélt javítások alapján
2018.03.04. 11:00	2 óra	Horvath	Tevékenység: Szekvencia Diagramok átgondolása
2018.03.04. 11:00	4 óra	Mocsári	Tevékenység: Szekvencia Diagramok javítása
2018.03.04. 14:45	3 óra	Bottlik	Tevékenység: Class Diagram leírása
2018.03.04. 22:00	1 óra	Horvath	Tevékenység: Dokumentum megírása, naplózás
2018.03.04. 22:10	1 óra	Holub	Tevékenység: Dokumentum áttekintése
2018.03.05. 7:30	2 óra	Mocsári	Dokumentum végeleges formázása, szekvencia diagram leírások elkészítése

5. Szkeleton tervezése

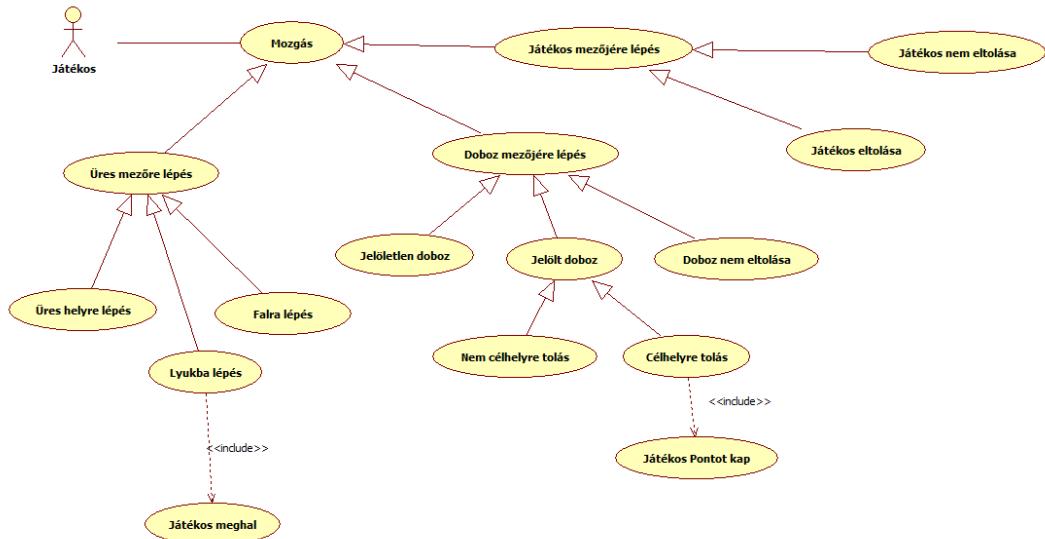
5.1 A szkeleton modell valóságos use-case-ei

5.1.1 Use-case diagram

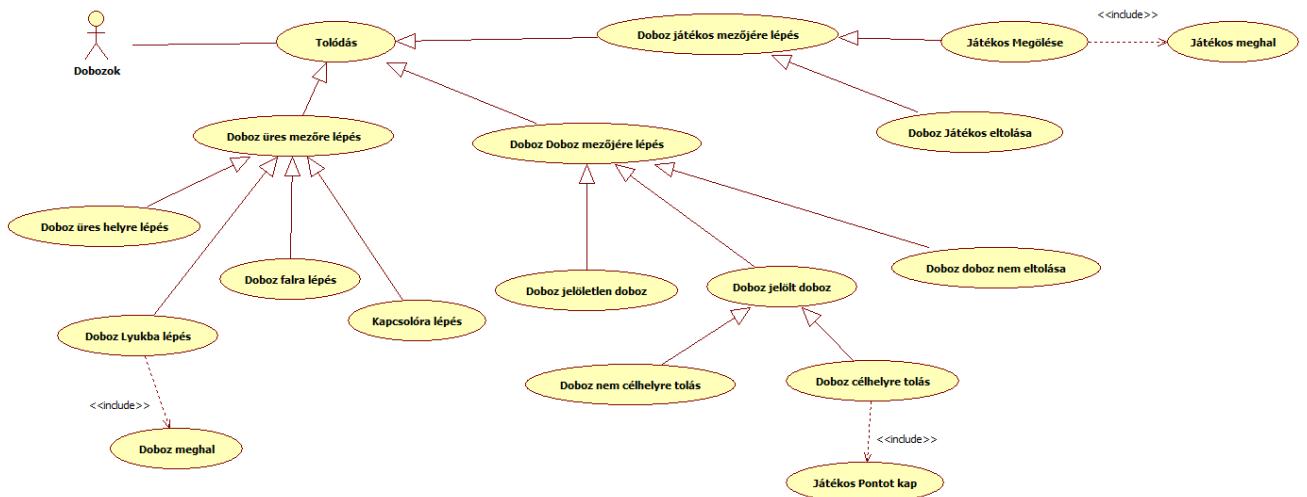
5.1.1.1 Initialization



5.1.1.2 Játékos mozgás



5.1.1.3 Doboz mozgás



5.1.2 Use-case leírások

5.1.2.1 Inicializálás

Use-case neve	Játék inicializálás
Rövid leírás	A játék indításakor a program betölti a játékhoz szükséges információkat.
Aktorok	Játék
Főforgatókönyv	1. A játék indításakor a program betölti a játékhoz szükséges információkat.
Forgatókönyv	1.A.1. A játék betölti az előre elkészített pálya mintát.
Forgatókönyv	1.A.2. A minta alapján létrehozza a játék elemeit és felépíti a játékteret.
Forgatókönyv	1.A.3. Feltölti a játékteret a játékban szereplő elemekkel.

Use-case neve	Pálya betöltése
Rövid leírás	A játék betölti az előre elkészített pálya mintát.
Aktorok	Játék
Főforgatókönyv	1. A játék betölti az előre elkészített pálya mintát.

Use-case neve	Pályaelemek betöltése
Rövid leírás	A minta alapján létrehozza a játék elemeit és felépíti a játékteret.
Aktorok	Játék
Főforgatókönyv	1. A mintának megfelelően létrehozza a pálya mezőit, és felépíti belőlük a pályát.

Use-case neve	Objektumok pályára helyezése
Rövid leírás	Feltölti a játékteret a játékban szereplő elemekkel.
Aktorok	Játék
Főforgatókönyv	1. A dobozok és játékosok elhelyezése a pályán.

5.1.2.2 Játékos mozgása

Use-case neve	Mozgás
Rövid leírás	A játékos képes mozogni.
Aktorok	Játékos
Főforgatókönyv	1. A játékosok minden irányba képesek mozogni.
Forgatókönyv	1.A.1. A játékos képes eltolni látad minden irányba.
Forgatókönyv	1.A.1.A.1 A játékos nem képes eltolni látad falon keresztül.
Forgatókönyv	1.A.1.B.1 A játékos képes eltolni jelölt látad minden irányba.
Forgatókönyv	1.A.1.B.2 A játékos nem képes eltolni jelölt látad falon keresztül.
Forgatókönyv	1.A.1.B.3 A jelölt látad célhelyre toló játékos pontot kap.
Forgatókönyv	1.A.1.B.4 A jelölt látad nem-célhelyre tolásakor a doboz jelöletlen dobozként viselkedik.
Forgatókönyv	1.A.2 A játékos nem képes falon áthaladni.
Forgatókönyv	1.A.3 A lyukra lépett játékos meghal.
Forgatókönyv	1.A.4 A nyitott csapdaajtóra lépett játékos meghal.
Forgatókönyv	1.A.5 játékosok el tudják egymást tolni.
Forgatókönyv	1.B.1. A játékban több játékos is részt tud venni.

Use-case neve	Üres mezőre lépés
Rövid leírás	A játékos olyan mezőre lép, ahol nincs semmilyen más léptethető objektum.
Aktorok	Játékos
Főforgatókönyv	1. A játékosok minden irányba képesek mezőre lépni.
Forgatókönyv	1.A.1 A játékos nem képes falon áthaladni.
Forgatókönyv	1.A.2 A lyukra lépett játékos meghal.
Forgatókönyv	1.A.3 A nyitott csapdaajtóra lépett játékos meghal.

Use-case neve	Üres helyre lépés
Rövid leírás	A játékos olyan üres mezőre lép, ahol nincs semmilyen más léptethető objektum.
Aktorok	Játékos
Főforgatókönyv	1. A játékosok minden irányba képesek mezőre lépni.

Use-case neve	Falra lépés
Rövid leírás	A játékos fal mezőre lép.
Aktorok	Játékos
Főforgatókönyv	1. A játékos nem képes falon áthaladni.

Use-case neve	Lyukra lépés
Rövid leírás	A játékos lyuk mezőre lép.
Aktorok	Játékos
Főforgatókönyv	1. A lyukra lépett játékos meghal.

Use-case neve	Játékos meghal
Rövid leírás	A játékos meghal.
Aktorok	Játékos
Főforgatókönyv	1. A játékos ha meghal eltűnik a pályáról.

Use-case neve	Játékos mezőjére lépés
Rövid leírás	A játékos olyan mezőre lép, ahol más játékos tartózkodik.
Aktorok	Játékos
Főforgatókönyv	1. Játékosok el tudják egymást tolni.
Forgatókönyv	1.A.1 Egy játékos nem tudja a másik játékost olyan irányba tolni, ahol nincs szabad mező.

Use-case neve	Játékos eltolása
Rövid leírás	A játékos olyan mezőre lép, ahol más, eltolható játékos tartózkodik.
Aktorok	Játékos
Főforgatókönyv	1. játékosok el tudják egymást tolni.

Use-case neve	Játékos nem eltolása
Rövid leírás	A játékos olyan mezőre lép, ahol más, nem eltolható játékos tartózkodik.
Aktorok	Játékos
Főforgatókönyv	1. Egy játékos nem tudja a másik játékost olyan irányba tolni, ahol nincs szabad mező.

Use-case neve	Doboz mezőjére lépés
Rövid leírás	A játékos olyan mezőre lép, ahol doboz tartózkodik.
Aktorok	Játékos
Főforgatókönyv	1. A játékos képes eltolni lánát minden irányba.
Forgatókönyv	1.A.1 A játékos nem képes eltolni lánát falon keresztül.
Forgatókönyv	1.B.1 A játékos képes eltolni jelölt lánát minden irányba.
Forgatókönyv	1.B.2 A játékos nem képes eltolni jelölt lánát falon keresztül.
Forgatókönyv	1.B.3 A jelölt lánát célhelyre toló játékos pontot kap.
Forgatókönyv	1.B.4 A jelölt lánát nem-célhelyre tolásakor a doboz jelöletlen dobozként viselkedik.

Use-case neve	Doboz nem eltolása
Rövid leírás	A játékos olyan mezőre lép, ahol nem eltolható doboz tartózkodik.
Aktorok	Játékos
Főforgatókönyv	1. A játékos nem képes eltolni lánát falon keresztül.
Forgatókönyv	1.A.1 A játékos nem képes eltolni jelölt lánát falon keresztül.

Use-case neve	Jelöletlen doboz
Rövid leírás	A játékos olyan mezőre lép, ahol eltolható jelöletlen doboz tartózkodik.
Aktorok	Játékos
Főforgatókönyv	1. A játékos képes eltolni lánát minden irányba.

Use-case neve	Jelölt doboz
Rövid leírás	A játékos olyan mezőre lép, ahol eltolható jelölt doboz tartózkodik.
Aktorok	Játékos
Főforgatókönyv	1. A játékos képes eltolni jelölt lánát minden irányba.
Forgatókönyv	1.A.1 A jelölt lánát célhelyre toló játékos pontot kap.
Forgatókönyv	1.A.2 A jelölt lánát nem-célhelyre tolásakor a doboz jelöletlen dobozként viselkedik.

Use-case neve	Nem célhelyre tolás
Rövid leírás	A játékos nem célhely mezőre tolja a dobozt.
Aktorok	Játékos
Főforgatókönyv	1. A jelölt lánát nem-célhelyre tolásakor a doboz jelöletlen dobozként viselkedik.

Use-case neve	Célhelyre tolás
Rövid leírás	A játékos célhely mezőre tolja a dobozt.
Aktorok	Játékos
Főforgatókönyv	1. A jelölt látadát célhelyre toló játékos pontot kap.

Use-case neve	Játékos pontot kap
Rövid leírás	A jelölt látadát célhelyre toló játékos pontot kap.
Aktorok	Játékos
Főforgatókönyv	1. A játékos pontjainak száma növekszik.

5.1.2.1 Doboz mozgása

Use-case neve	Tolódás
Rövid leírás	A játékos által megtolt láda mozog.
Aktorok	Doboz
Főforgatókönyv	1. A doboz képes eltolódni minden irányba.
Forgatókönyv	1.A.1 A doboz képes eltolni lánát minden irányba.
Forgatókönyv	1.A.1.A.1 A doboz nem képes eltolni lánát falon keresztül.
Forgatókönyv	1.A.1.B.1 A doboz képes eltolni jelölt lánát minden irányba.
Forgatókönyv	1.A.1.B.2 A doboz nem képes eltolni jelölt lánát falon keresztül.
Forgatókönyv	1.A.1.B.3 A jelölt lánát közvetetten célhelyre toló játékos pontot kap.
Forgatókönyv	1.A.1.B.4 A jelölt lánát nem-célhelyre tolásakor a doboz jelöletlen dobozként viselkedik.
Forgatókönyv	1.A.2 Doboz sem képes áthatolni a falon.
Forgatókönyv	1.A.3 A lyukra tolta doboz eltűnik.
Forgatókönyv	1.A.4 A kapcsolóra tolta doboz, bekapcsolja a kapcsolót.
Forgatókönyv	1.B.1. A célmezők üres mezőként viselkednek jelöletlen láná esetén.
Forgatókönyv	1.B.2. Jelölt lánák jelöletlen mezők esetén jelöletlen lánaként viselkednek.
Forgatókönyv	1.B.3. Jelölt lánák jelölt helyre tolás után elveszítik a jelölésüket.
Forgatókönyv	1.C.1. Ha egy játékosra dobozt tolunk, átcsúszik a mögötte lévő mezőre.
Forgatókönyv	1.C.2. Ha egy játékosra dobozt tolunk, de nincs mögötte üres mező, meghal.
Forgatókönyv	1.D.1. A játék véget ér, ha minden doboz a helyére került.
Forgatókönyv	1.D.2. A játék véget ér, ha már nem tudunk több jelölt dobozt eltolni.
Use-case neve	Doboz üres mezőre lépés
Rövid leírás	A doboz olyan mezőre lép, ahol nincs semmilyen más léptethető objektum.
Aktorok	dobozok
Főforgatókönyv	1. A doboz képes eltolódni minden irányba.
Forgatókönyv	1.A.2 Doboz sem képes áthatolni a falon.
Forgatókönyv	1.A.3 A lyukra tolta doboz eltűnik.
Forgatókönyv	1.A.4 A kapcsolóra tolta doboz, bekapcsolja a kapcsolót.

Forgatókönyv	1.B.1. A célmezők üres mezőként viselkednek jelöletlen láda esetén.
Forgatókönyv	1.B.2. Jelölt lánkok jelöletlen mezők esetén jelöletlen lánkoként viselkednek.
Forgatókönyv	1.B.3. Jelölt lánkok jelölt helyre tolás után elveszti a jelölésüket.

Use-case neve	Doboz üres helyre lépés
Rövid leírás	A doboz olyan üres mezőre lép, ahol nincs semmilyen más léptethető objektum.
Aktorok	dobozok
Főforgatókönyv	1. A doboz képes eltolódni minden irányba.

Use-case neve	Doboz falra lépés
Rövid leírás	A doboz fal mezőre lép.
Aktorok	dobozok
Főforgatókönyv	1. Doboz sem képes áthatolni a falon.

Use-case neve	Doboz Lyukba lépés
Rövid leírás	A doboz lyuk mezőre lép.
Aktorok	dobozok
Főforgatókönyv	1. A lyukra tolta doboz eltűnik.

Use-case neve	Doboz meghal
Rövid leírás	A doboz meghal.
Aktorok	dobozok
Főforgatókönyv	1. A doboz ha meghal eltűnik a pályáról.

Use-case neve	Doboz kapcsolóra lépés
Rövid leírás	A doboz mezőre lép.
Aktorok	dobozok
Főforgatókönyv	1. A kapcsolóra tolta doboz, bekapcsolja a kapcsolót.

Use-case neve	Doboz játékos mezőjére lépés
Rövid leírás	A doboz olyan mezőre lép, amelyen játékos tartózkodik.
Aktorok	Doboz
Főforgatókönyv	1. Ha egy játékosra dobozt tolunk, átcsúszik a mögötte lévő mezőre.
Forgatókönyv	1.A.1. Ha egy játékosra dobozt tolunk, de nincs mögötte üres mező, meghal.

Use-case neve	Játékos megölése
Rövid leírás	A doboz olyan mezőre lép, amelyen nem eltolható játékos tartózkodik.
Aktorok	Doboz
Főforgatókönyv	1. Ha egy játékosra dobozt tolunk, de nincs mögötte üres mező, meghal.

Use-case neve	Doboz játékos eltolása
Rövid leírás	A doboz olyan mezőre lép, amelyen eltolható játékos tartózkodik.
Aktorok	Doboz
Főforgatókönyv	1. Ha egy játékosra dobozt tolunk, átcsúszik a mögötte lévő mezőre.

Use-case neve	Doboz Doboz mezőjére lépés
Rövid leírás	A doboz olyan mezőre lép, amelyen másik doboz tartózkodik.
Aktorok	Doboz
Főforgatókönyv	1. A doboz képes eltolni lánát minden irányba.
Forgatókönyv	1.A.1 A doboz nem képes eltolni lánát falon keresztül.
Forgatókönyv	1.B.1 A doboz képes eltolni jelölt lánát minden irányba.
Forgatókönyv	1.B.2 A doboz nem képes eltolni jelölt lánát falon keresztül.
Forgatókönyv	1.B.3 A jelölt lánát közvetetten célhelyre toló játékos pontot kap.
Forgatókönyv	1.B.4 A jelölt lánát nem-célhelyre tolásakor a doboz jelöletlen dobozként viselkedik.

Use-case neve	Doboz doboz nem eltolása
Rövid leírás	A doboz olyan mezőre lép, amelyen másik, nem eltolható doboz tartózkodik.
Aktorok	Doboz
Főforgatókönyv	1. A doboz nem képes eltolni látadat falon keresztül.
Forgatókönyv	1.A.1 A doboz nem képes eltolni jelölt látadat falon keresztül.

Use-case neve	Doboz jelöletlen doboz
Rövid leírás	A doboz olyan mezőre lép, amelyen másik, jelöletlen, eltolható doboz tartózkodik.
Aktorok	Doboz
Főforgatókönyv	1. A doboz képes eltolni látadat minden irányba.

Use-case neve	Doboz jelölt doboz
Rövid leírás	A doboz olyan mezőre lép, amelyen másik, jelölt, eltolható doboz tartózkodik.
Aktorok	Doboz
Főforgatókönyv	1. A doboz képes eltolni jelölt látadat minden irányba.
Forgatókönyv	1.A.1 A jelölt látadat közvetetten célhelyre toló játékos pontot kap.
Forgatókönyv	1.A.2 A jelölt látadat nem-célhelyre tolásakor a doboz jelöletlen dobozként viselkedik.

Use-case neve	Doboz nem célhelyre tolás
Rövid leírás	A doboz nem célhely mezőre tolja a dobozt.
Aktorok	Doboz
Főforgatókönyv	1. A jelölt lánát nem-célhelyre tolásakor a doboz jelöletlen dobozként viselkedik.

Use-case neve	Doboz célhelyre tolás
Rövid leírás	A doboz célhely mezőre tolja a dobozt.
Aktorok	Doboz
Főforgatókönyv	1. A jelölt lánát közvetetten célhelyre toló játékos pontot kap.

5.2 A szkeleton kezelői felületének terve, dialógusok

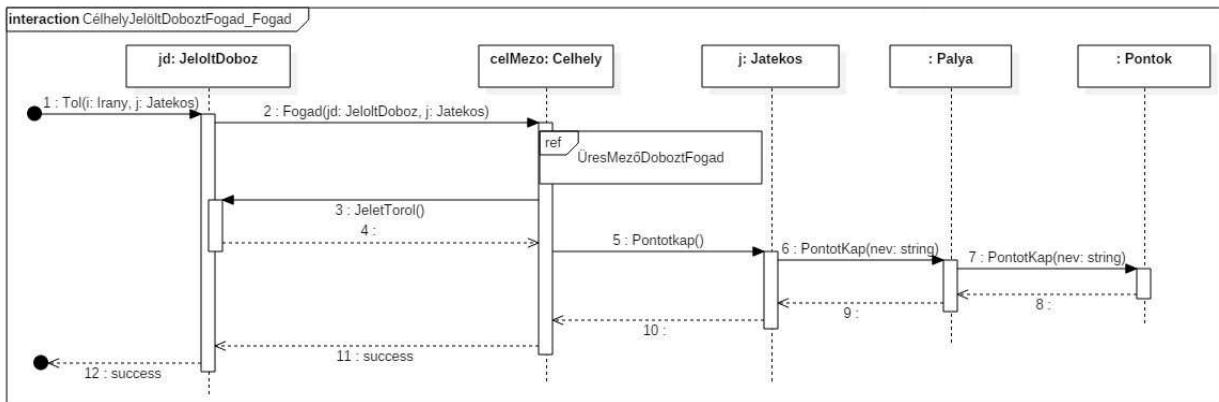
Az alkalmazás indításakor egy menü jelenik meg, melyből a tesztelő a kiválasztott teszteset azonosítójának megadásával lefuttathatja a kiválasztott tesztesetet. A kimenetet egy osztály generálja, mely tabulátorokkal eltoltan kiírja az aktuálisan hívott metódus nevét a standard kimenetre a következő módon:

```
<===== END OF TEST =====>

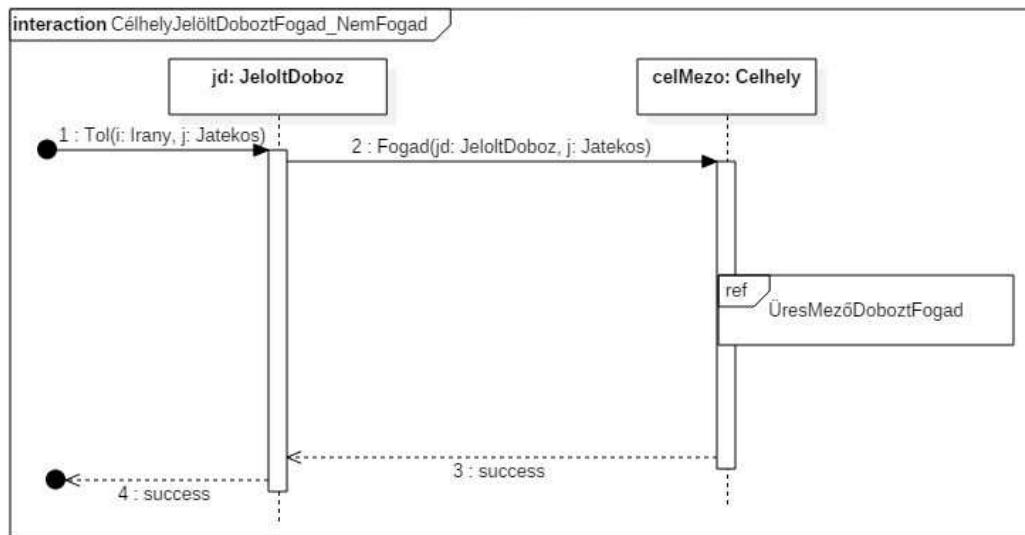
<===== DESCRIPTION =====>
  Specialis eset: Lancolas
  Jatekos - Doboz - Jatekos - Jatekos - Fal
<===== START OF TEST_8_5 =====>
  Jatekos_1  Lepes(Irany)
  Jatekos_1  Tol(Irany, Jatekos)
    UresMezo  GetSzomszed(Irany)
    Return: sokoban.UresMezo@2e5c649
    UresMezo  Fogad(Irany, Jatekos, Jatekos)
    Doboz   Tol(Irany, Jatekos)
      UresMezo  GetSzomszed(Irany)
      Return: sokoban.UresMezo@136432db
      UresMezo  Fogad(Irany, Doboz, Jatekos)
      Jatekos_2  Utkozik(Irany, Jatekos)
        UresMezo  GetSzomszed(Irany)
        Return: sokoban.UresMezo@7382f612
        UresMezo  Fogad(Irany, Jatekos, Jatekos)
        Jatekos_3  Tol(Irany, Jatekos)
          UresMezo  GetSzomszed(Irany)
          Return: sokoban.Fal@1055e4af
          Fal Fogad(Irany, Jatekos, Jatekos)
            Return: false
            Return: false
            Return: false
            Jatekos_2  Halal()
              Palya   Halal(Jatekos)
              Return
              Return
              Return: true
              Return: true
              UresMezo  Enged()
              Return
              Return: true
              Return: true
              UresMezo  Enged()
              Return
              Return: true
              Palya   Vegellenorzes()
                Jatek   EndGame()
                Return
                Return
                Return
                Return
                Return
<===== END OF TEST =====>
```

5.3 Szekvencia diagramok a belső működésre

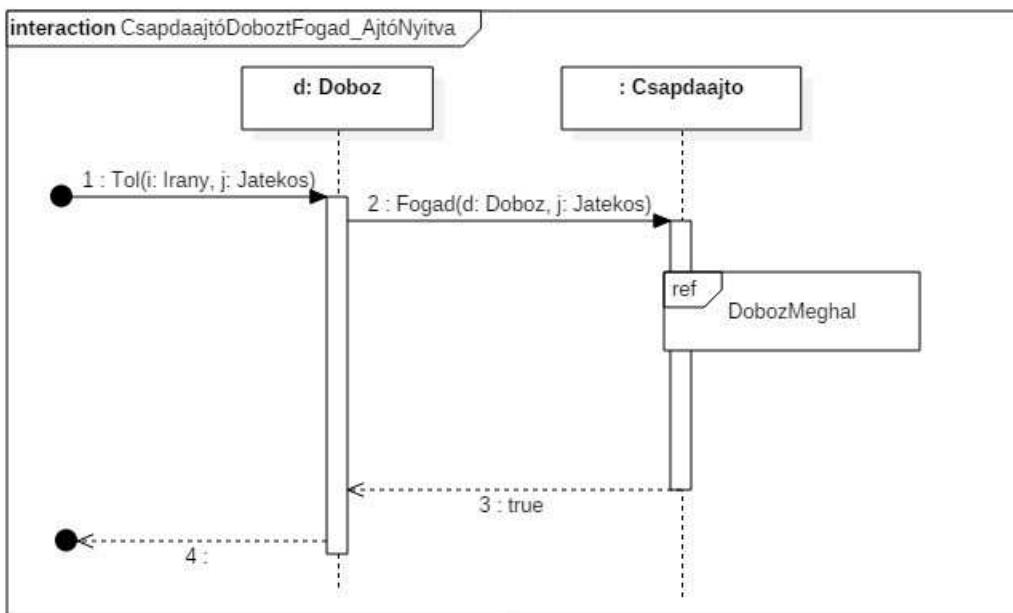
5.3.1 CélhelyJelöltDoboztFogad_Fogad



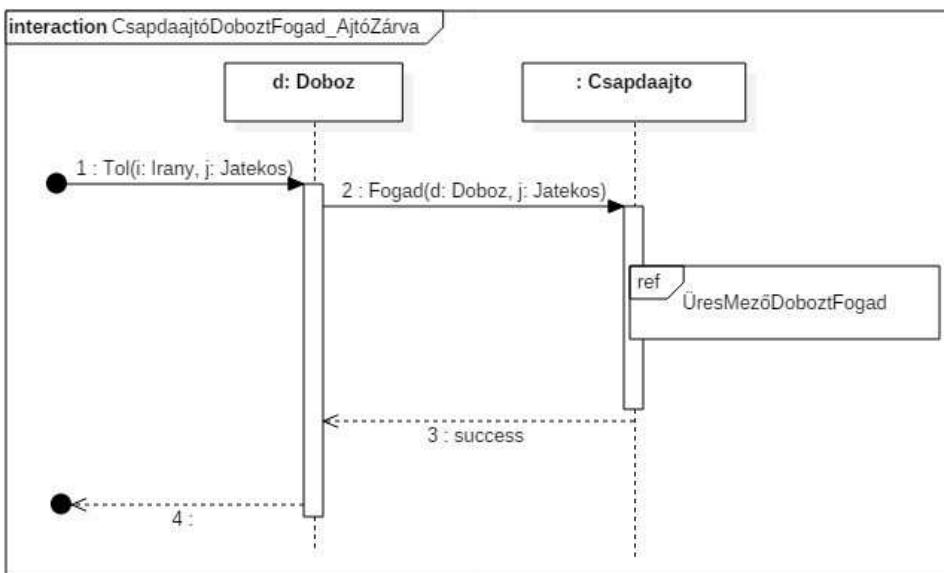
5.3.2 CélhelyJelöltDoboztFogad_NemFogad



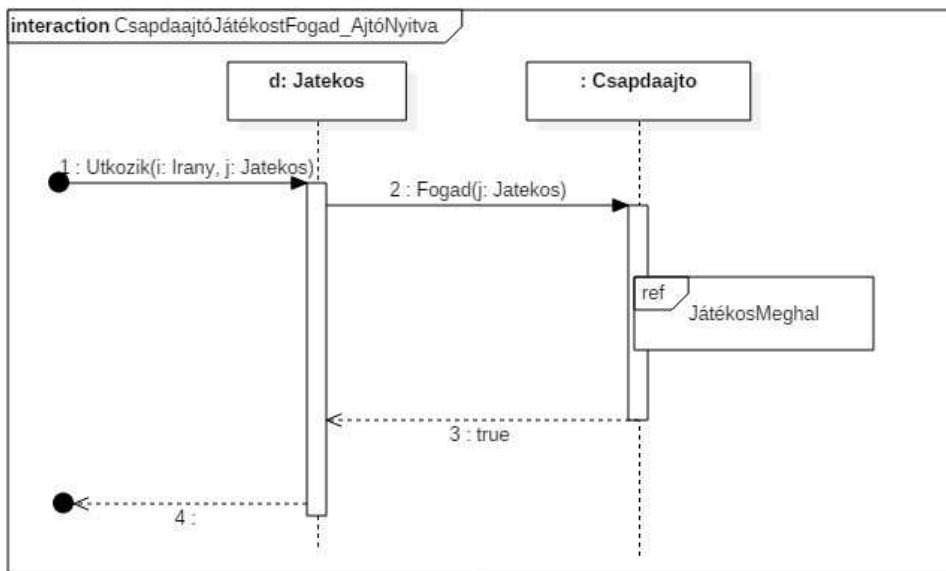
5.3.3 CsapdaajtDoboztFogad_AjtóNyitva



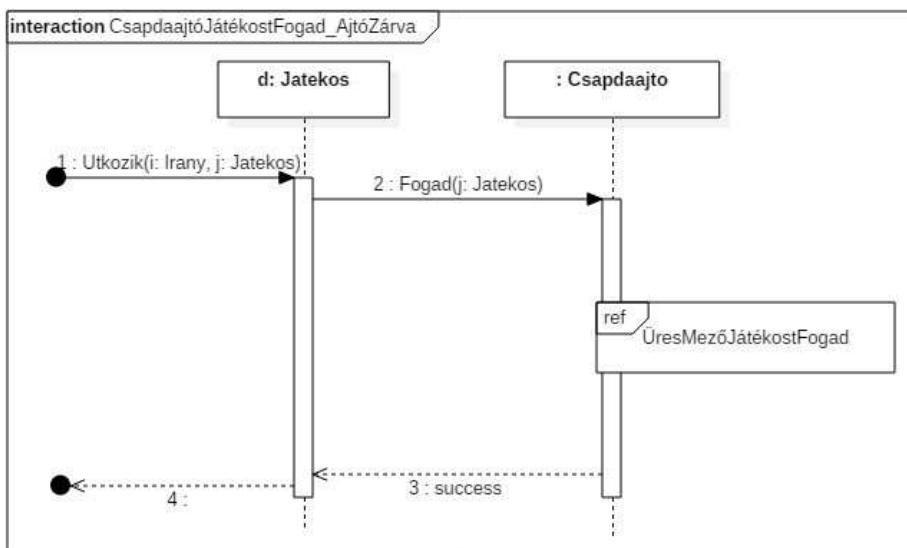
5.3.4 CspadaajtDoboztFogad_AjtóZárva



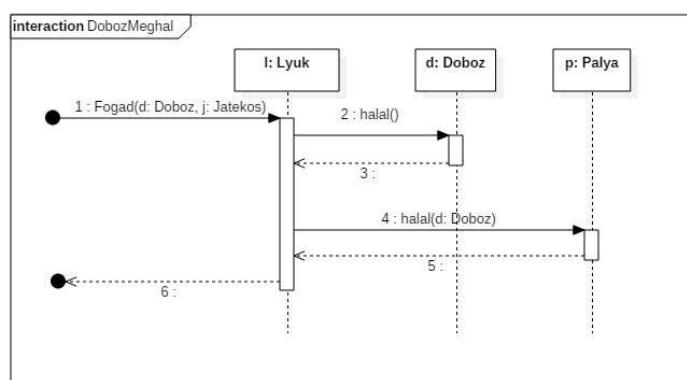
5.3.5 CsapdaajtóJátékostFogad_AjtóNyitva



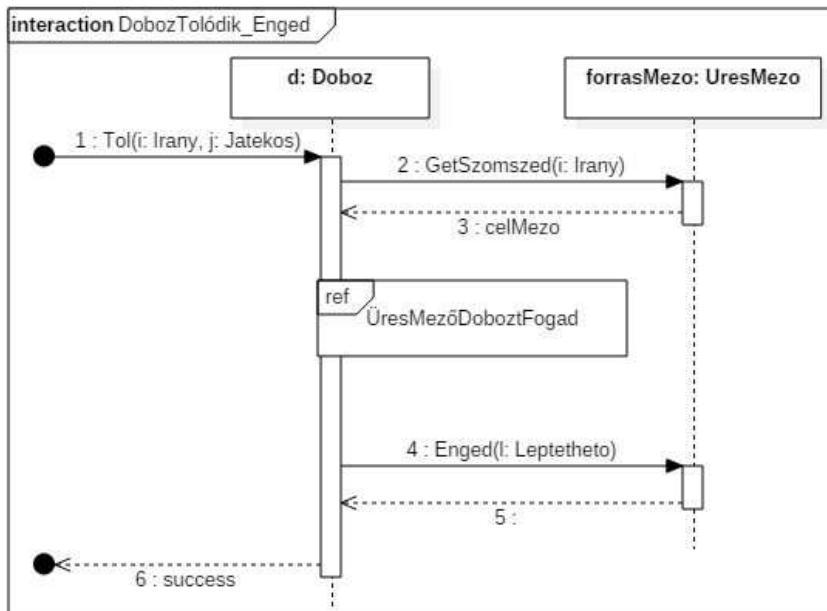
5.3.6 CsapdaajtóJátékostFogad_AjtóZárva



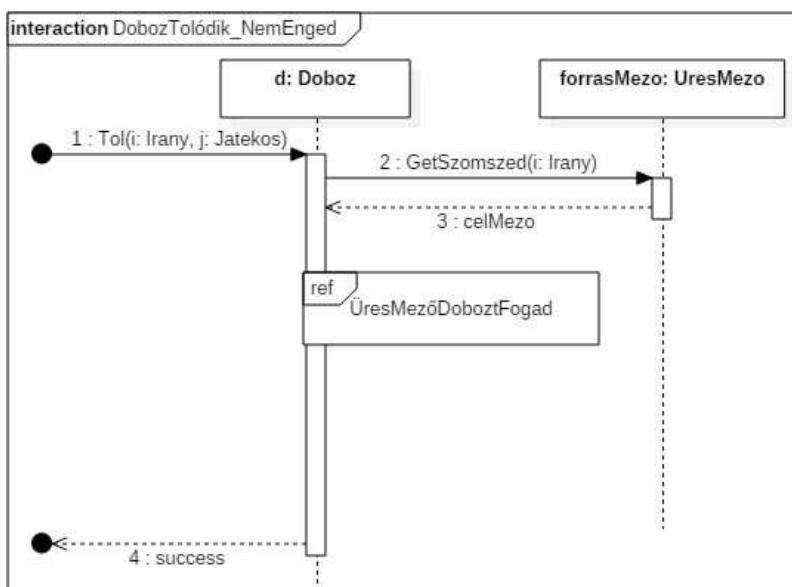
5.3.7 DobozMeghal



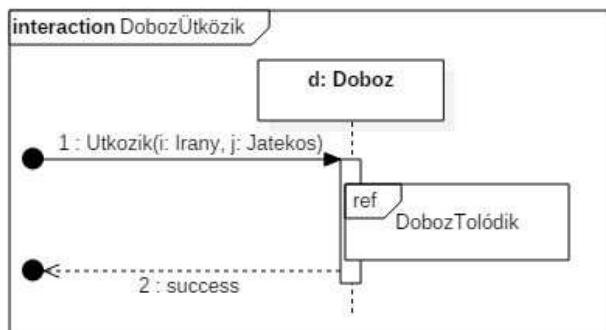
5.3.8 DobozTolódik_Enged



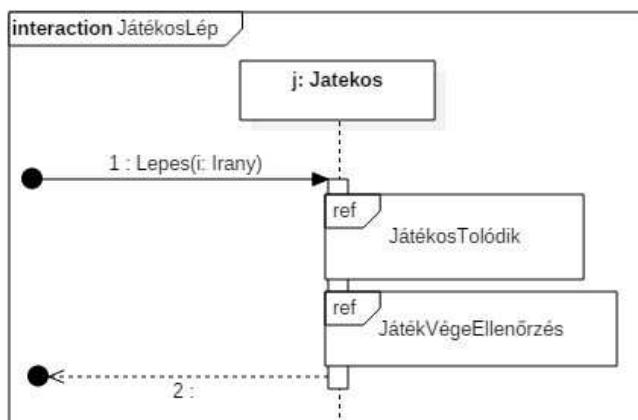
5.3.9 DobozTolodik_NemEnged



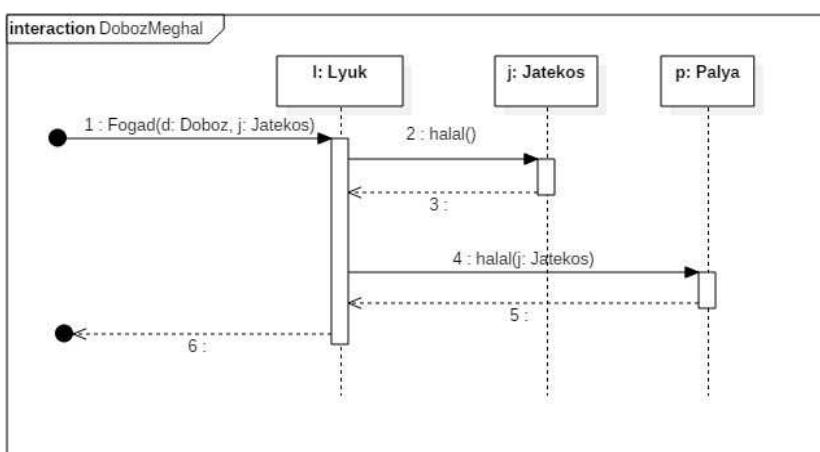
5.3.10 DobozÜtközik



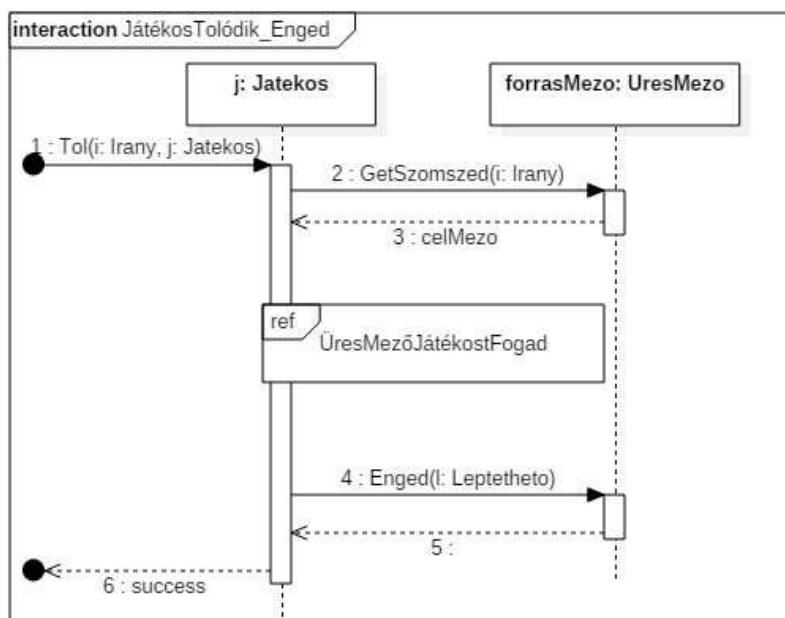
5.3.11 JátékosLép



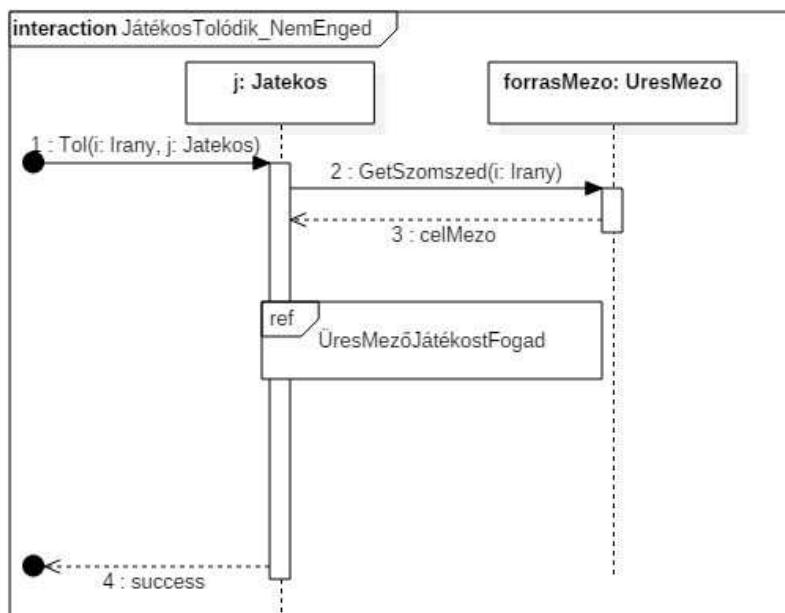
5.3.12 JátékosMeghal



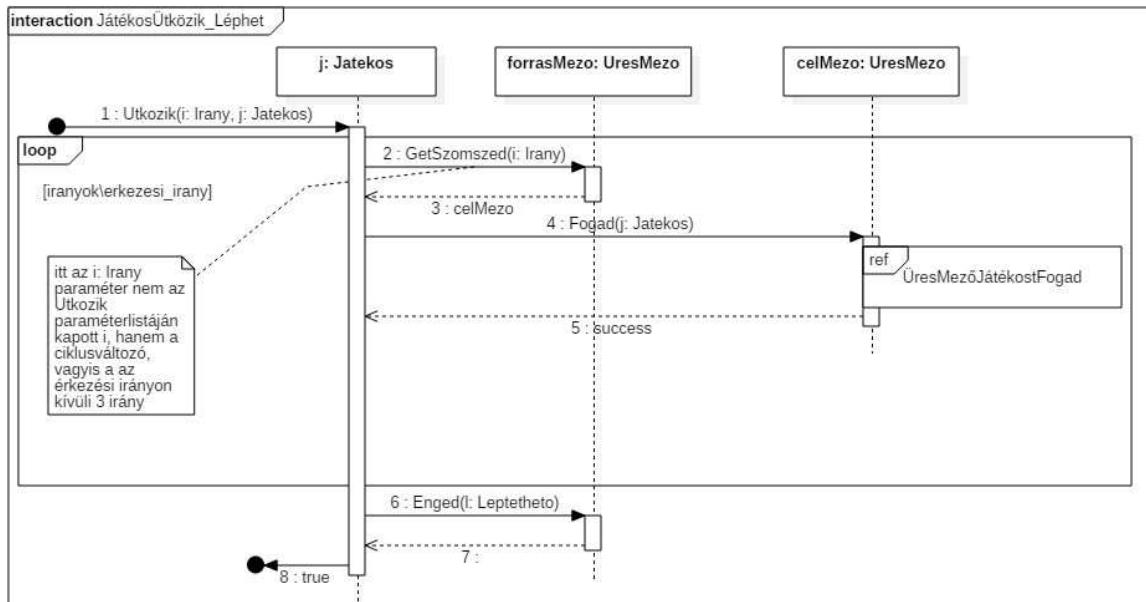
5.3.13 JátékosTolódik_Enged



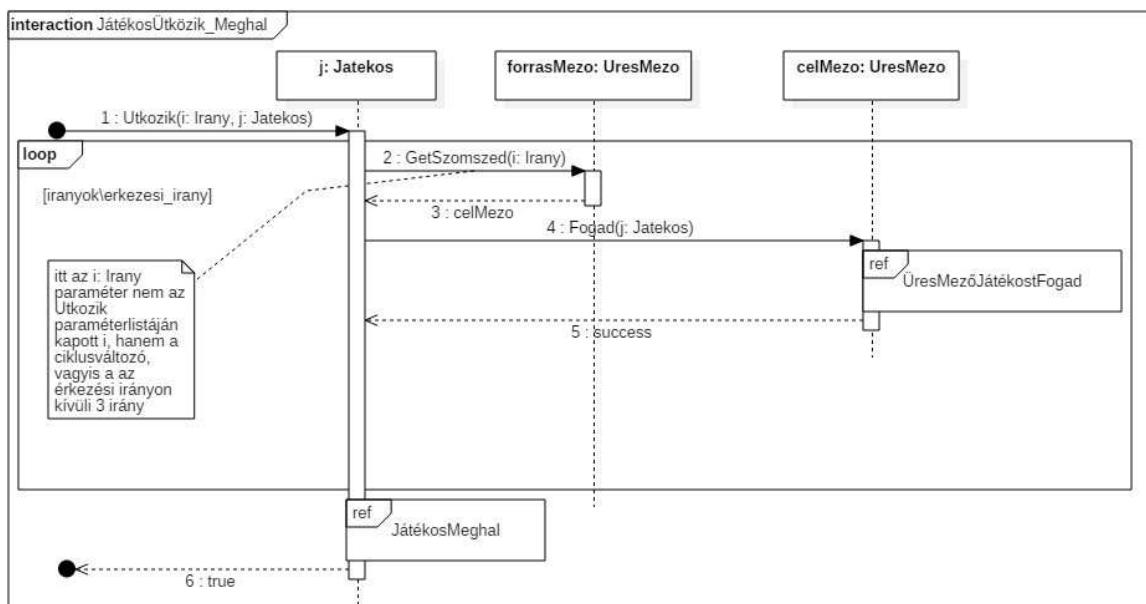
5.3.14 JátékosTolódik_NemEnged



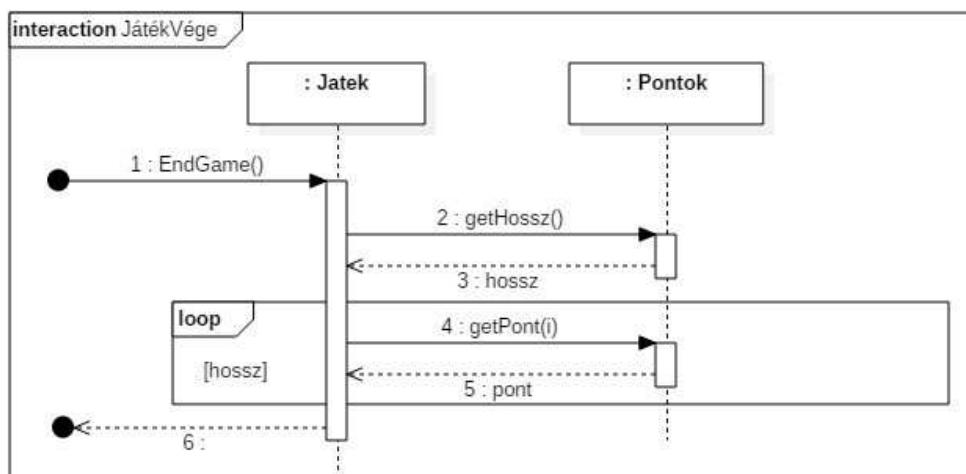
5.3.15 JátékosÜtközik_Léphet



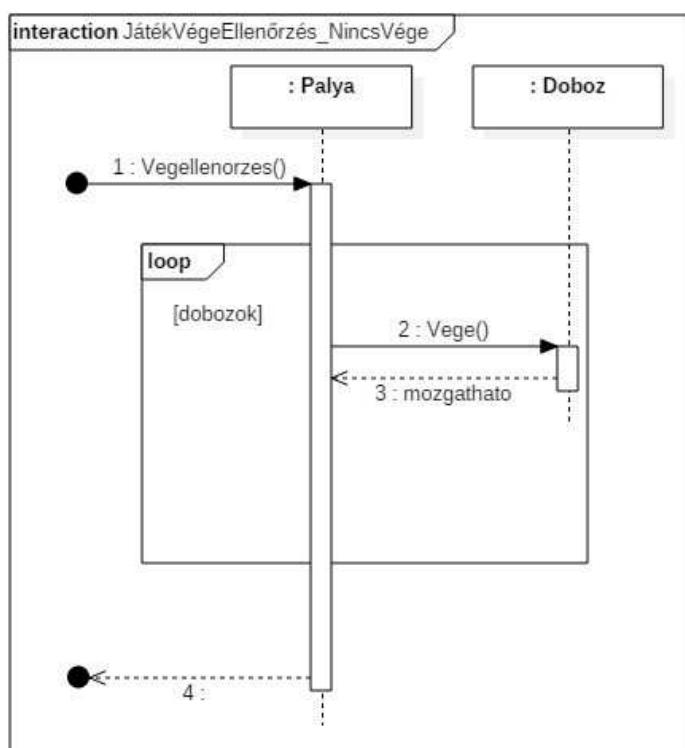
5.3.16 JátékosÜtközik_Meghal



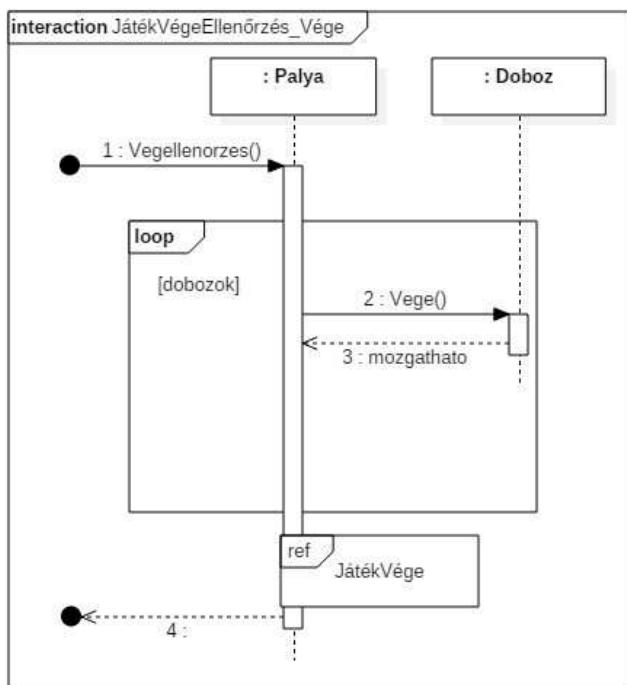
5.3.17 JátékVége



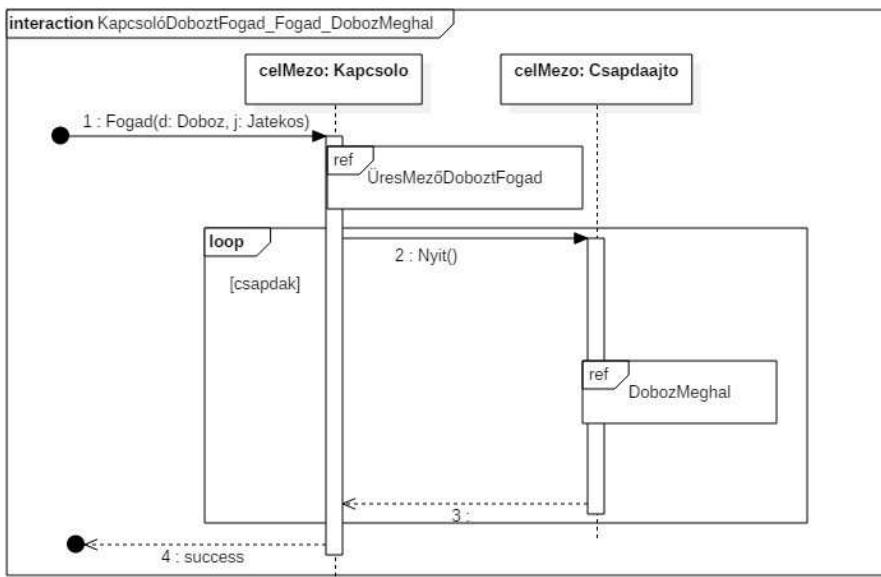
5.3.18 JátékVégeEllenőrzés_NincsVége



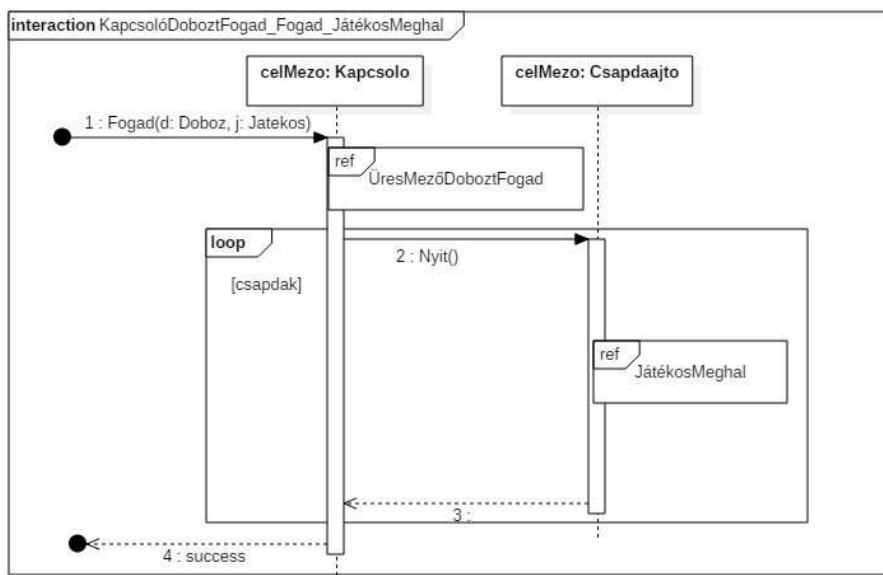
5.3.19 JátékVégeEllenőrzés_Vége



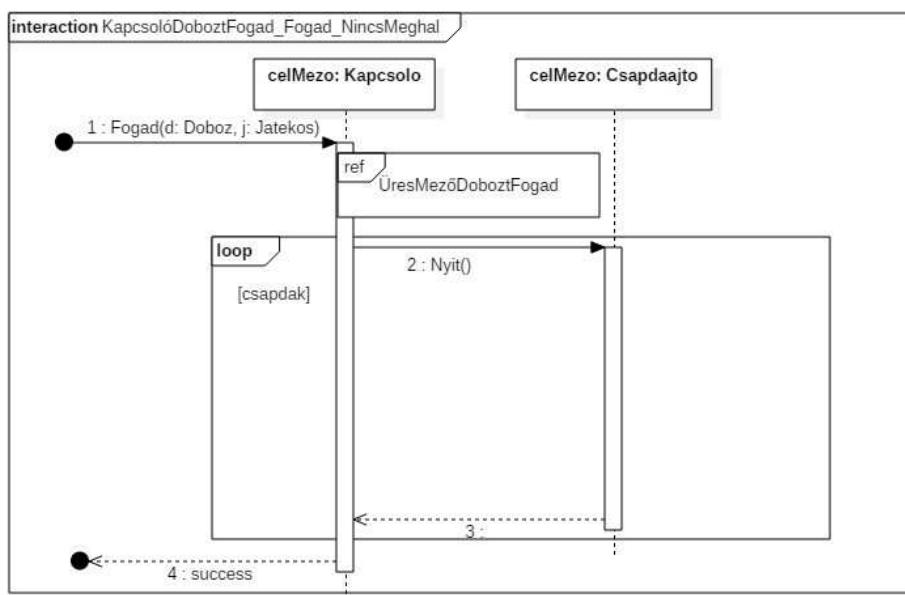
5.3.20 KapcsolóDoboztFogad_Fogad_DobozMeghal



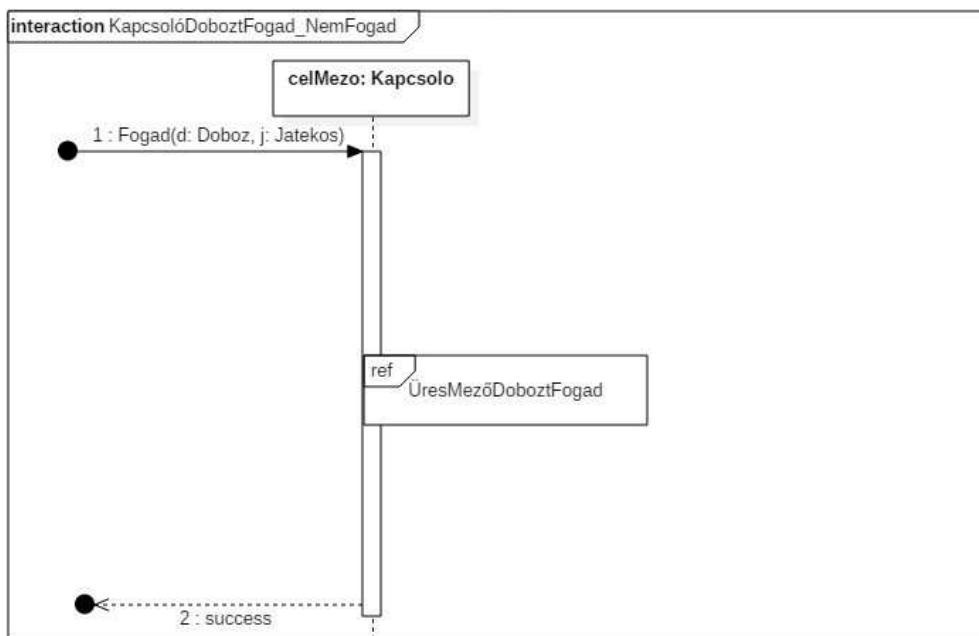
5.3.21 KapcsolóDoboztFogad_Fogad_JátékosMeghal



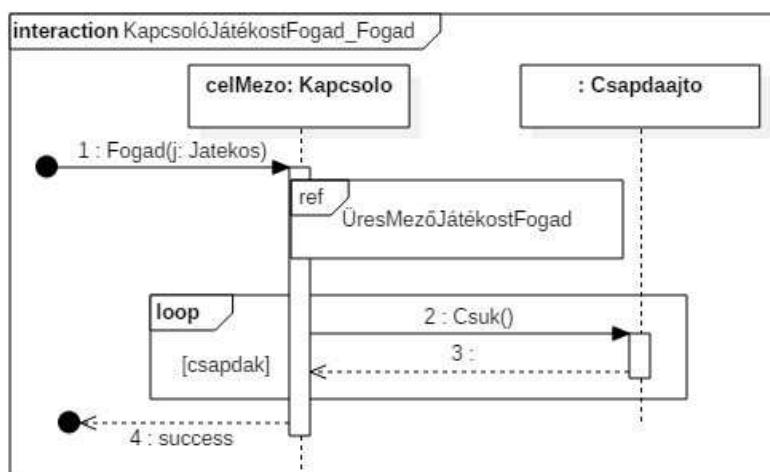
5.3.22 KapcsolóDoboztFogad_Fogad_NincsMeghal



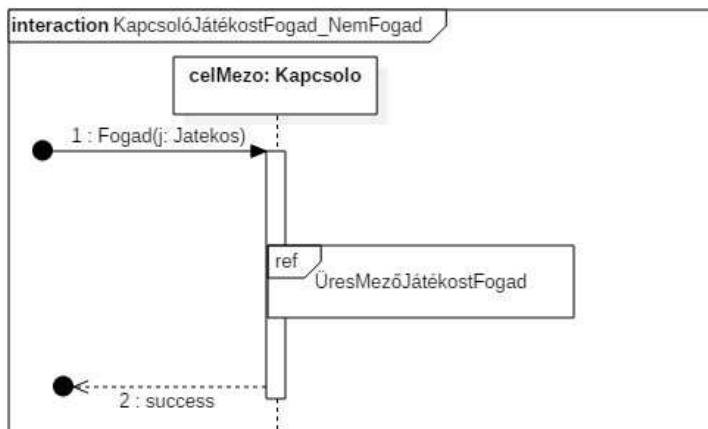
5.3.23 KapcsolóDoboztFogad_NemFogad



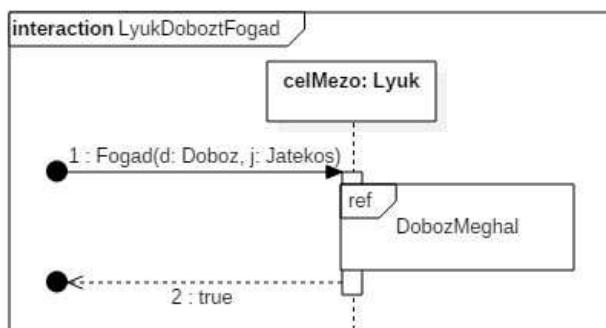
5.3.24 KapcsolóJátékostFogad_Fogad



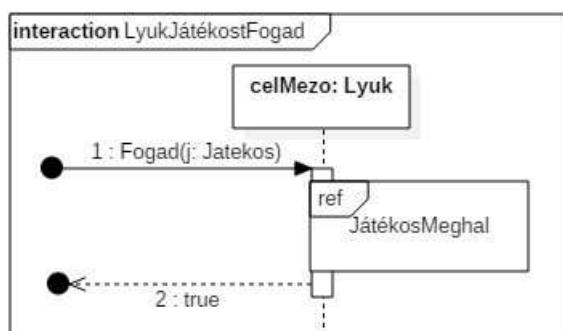
5.3.25 KapcsolóJátékostFogad_NemFogad



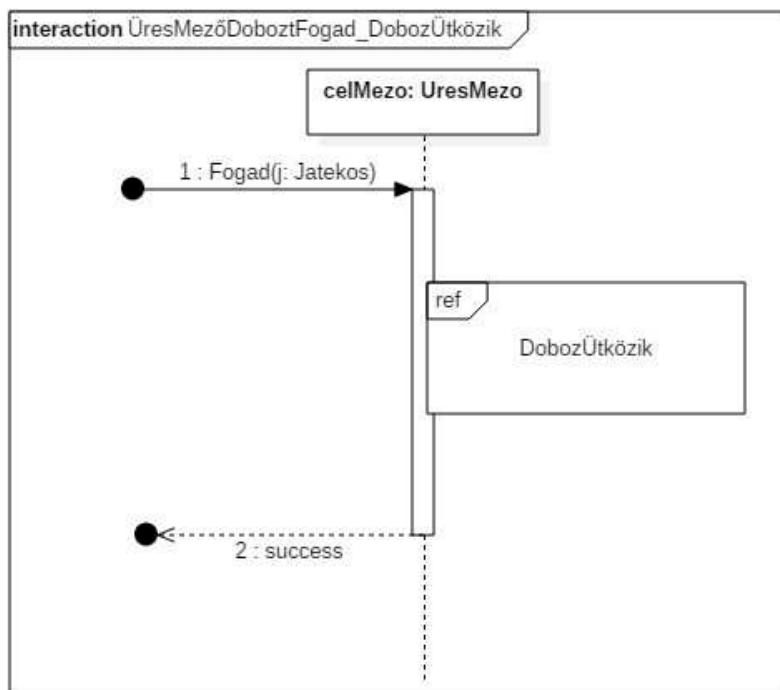
5.3.26 LyukDoboztFogad



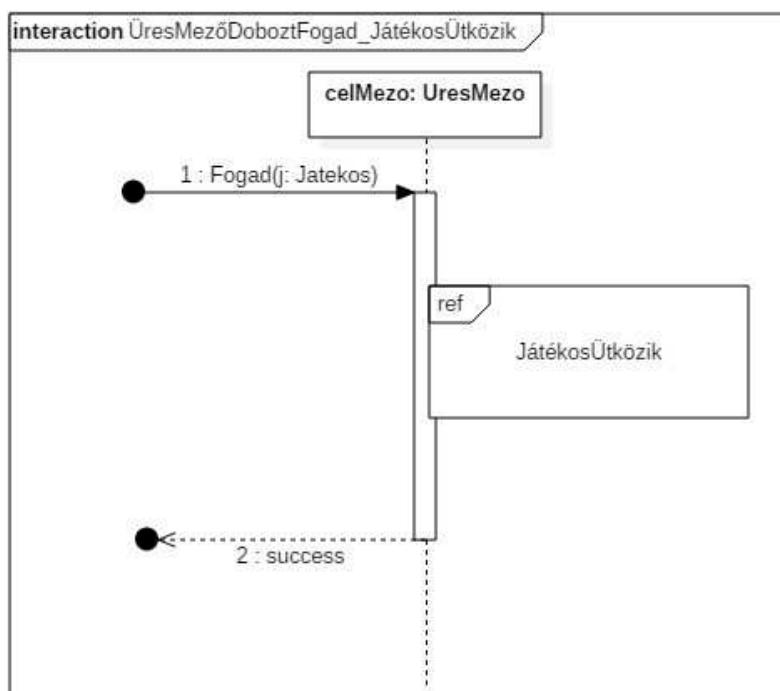
5.3.27 LyukJátékostFogad



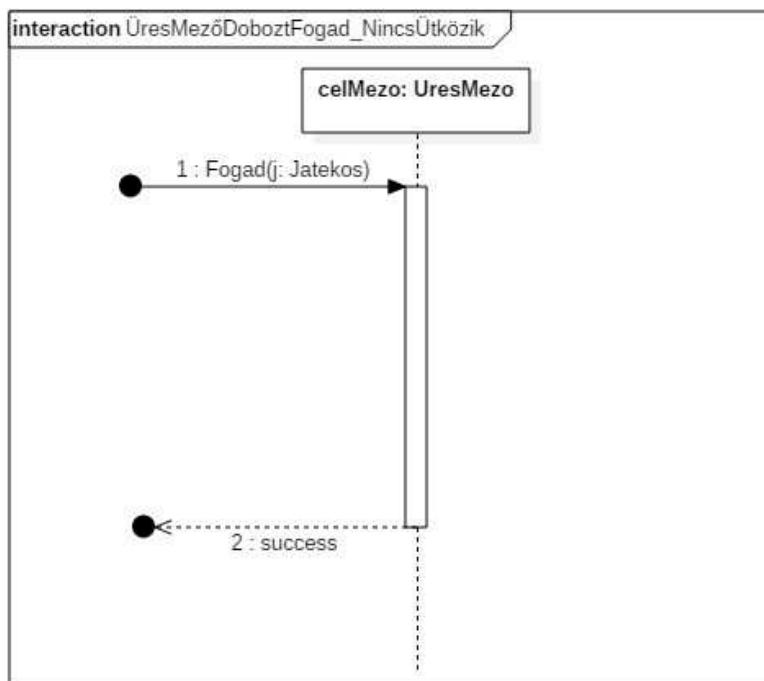
5.3.28 ÜresMezőDoboztFogad_DobozÜtközik



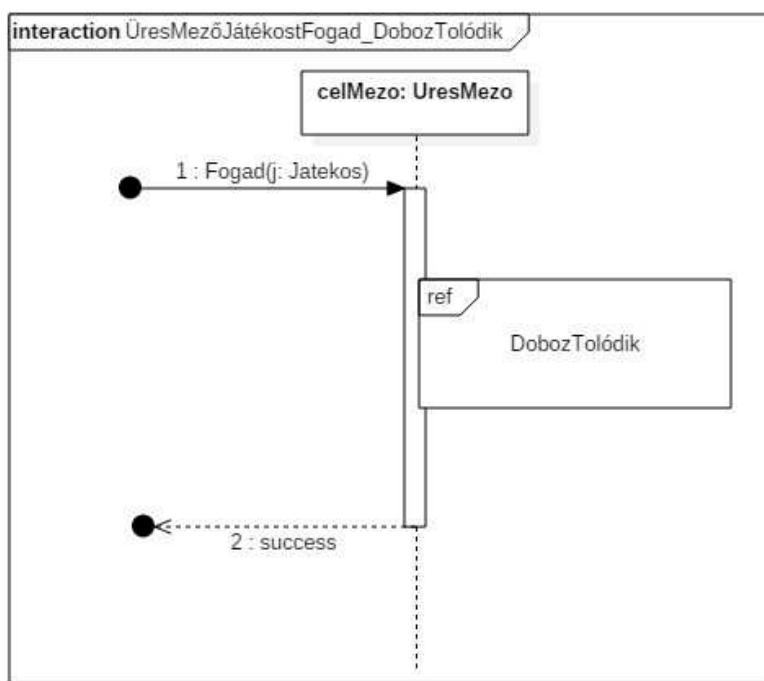
5.3.29 ÜresMezőDoboztFogad_JátékosÜtközik



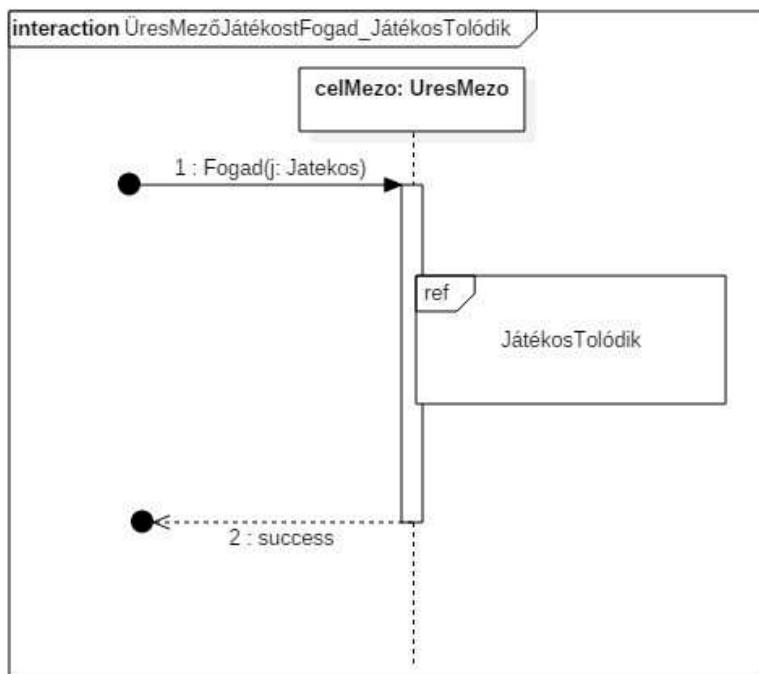
5.3.30 ÜresMezőDoboztFogad_NincsÜtközik



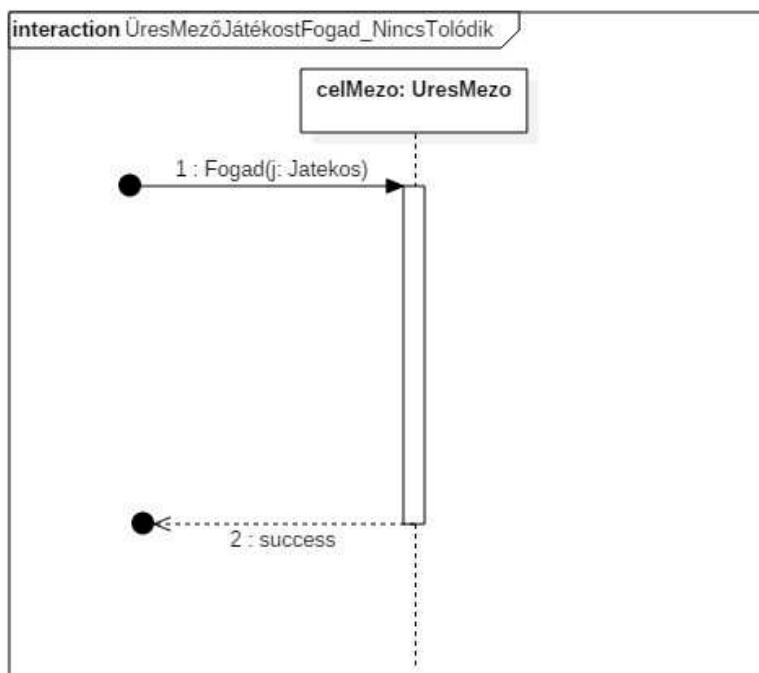
5.3.31 ÜresMezőJátékostFogad_DobozTolódik



5.3.32 ÜresMezőJátékostFogad_JátékosTolódik

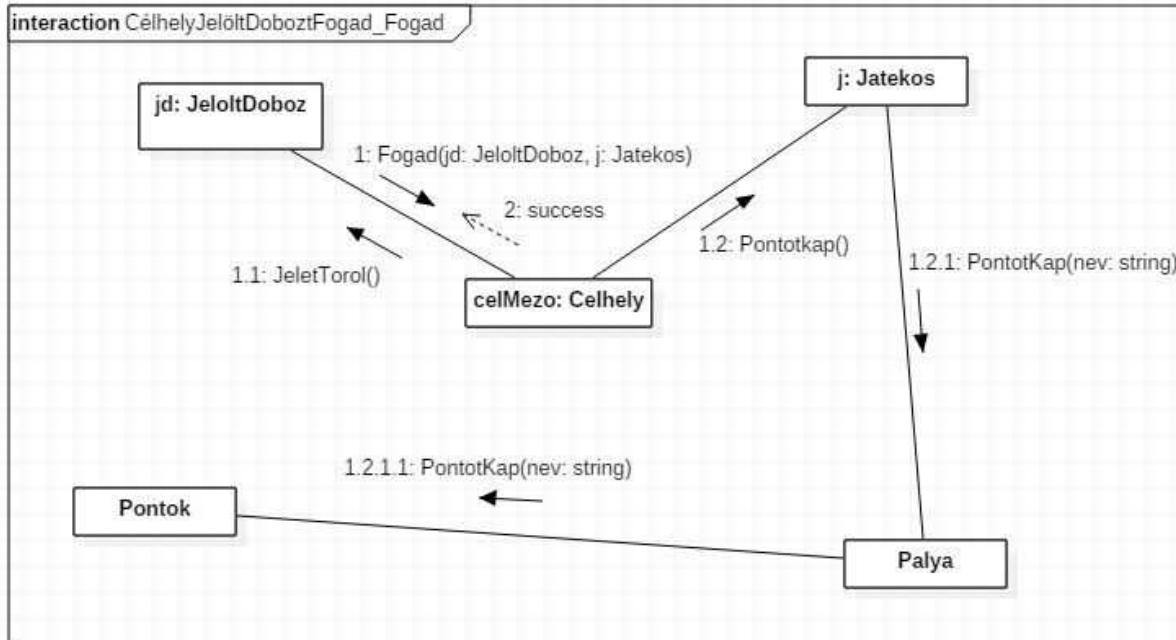


5.3.33 ÜresMezőJátékostFogad_NincsTolódik

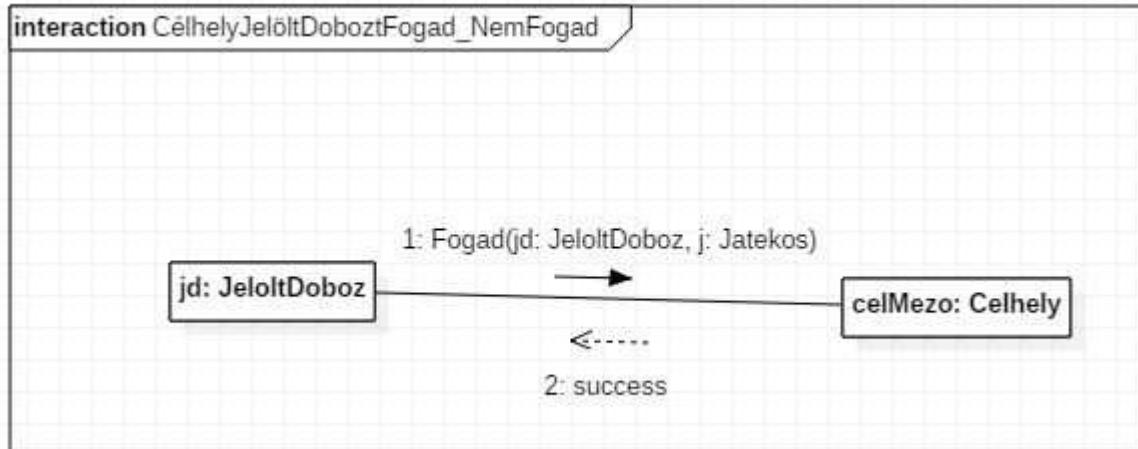


5.4 Kommunikációs diagramok

5.4.1 CélhelyJelöltDoboztFogad_Fogad

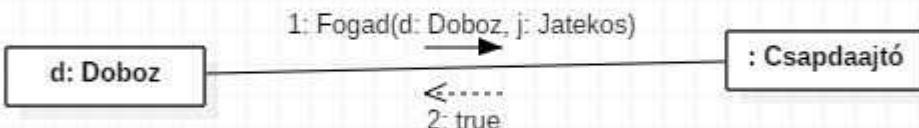


5.4.2 CélhelyJelöltDoboztFogad_NemFogad



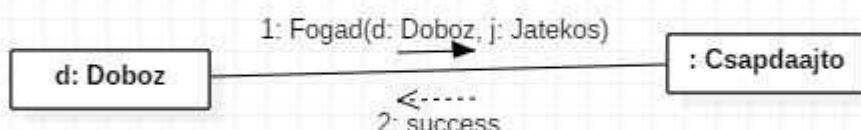
5.4.3 CsapdaajtóDoboztFogad_AjtóNyitva

interaction CsapdaajtóDoboztFogad_AjtóNyitva



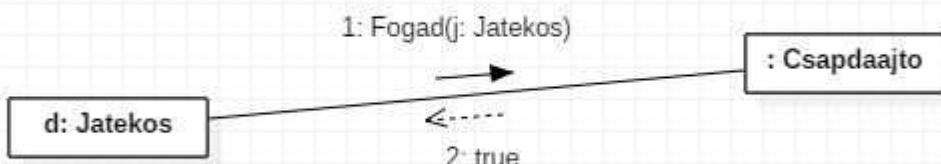
5.4.4 CsapdaajtóDoboztFogad_AjtóZárva

interaction CsapdaajtóDoboztFogad_AjtóZárva



5.4.5 CsapdaajtóJátékostFogad_AjtóNyitva

interaction CsapdaajtoJátékostFogad_AjtóNyitva

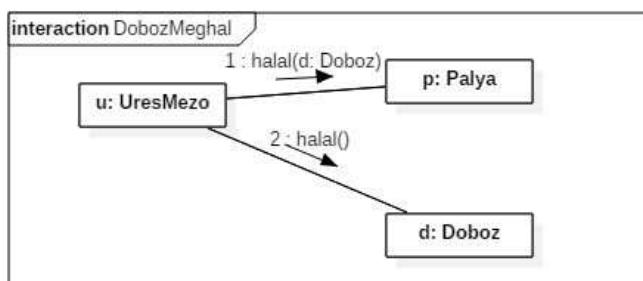


5.4.6 CsapdaajtóJátékostFogad_AjtóZárva

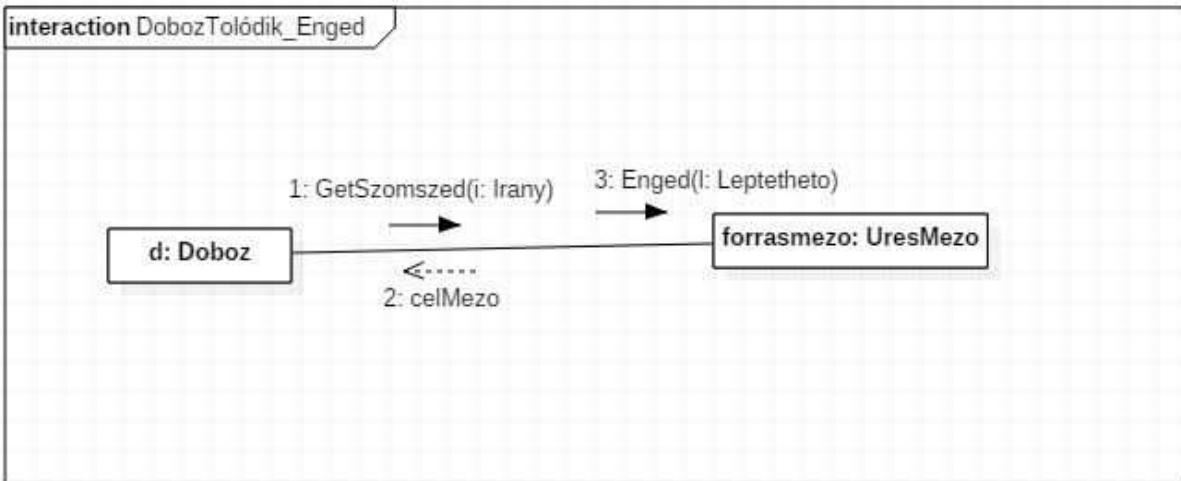
interaction CsapdaajtoJátékostFogad_AjtóZárva



5.4.7 DobozMeghal

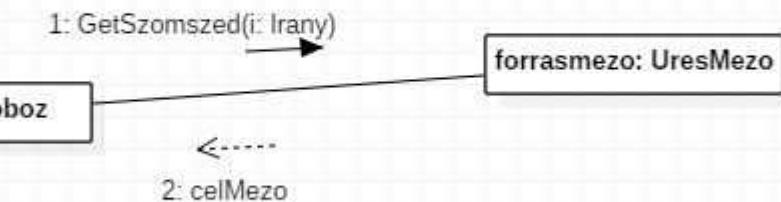


5.4.8 DobozTolódik_Enged

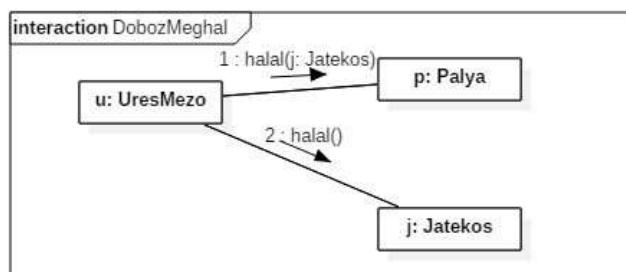


5.4.9 DobozTolódik_NemEnged

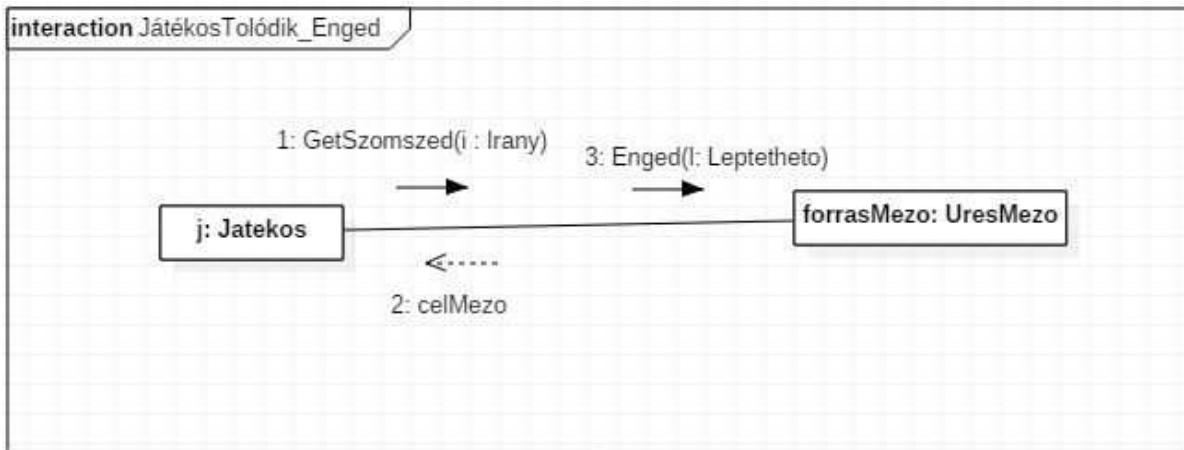
interaction DobozTolodik_NemEnged



5.4.10 JátékosMeghal

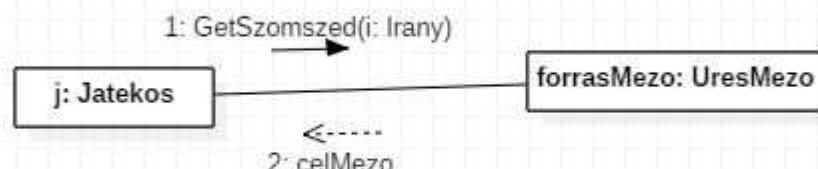


5.4.11 JátékosTolodik_Enged



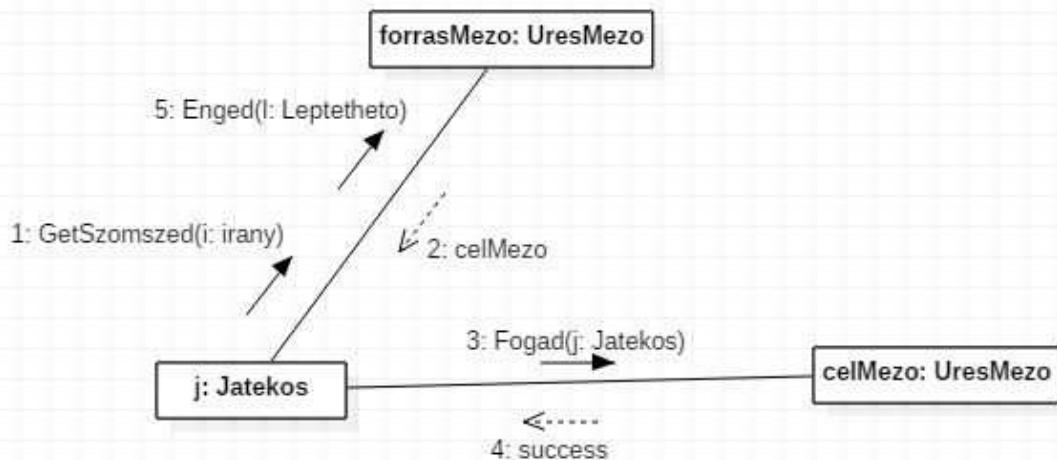
5.4.12 JátékosTolodik_NemEnged

interaction JátékosTolódik_NemEnged



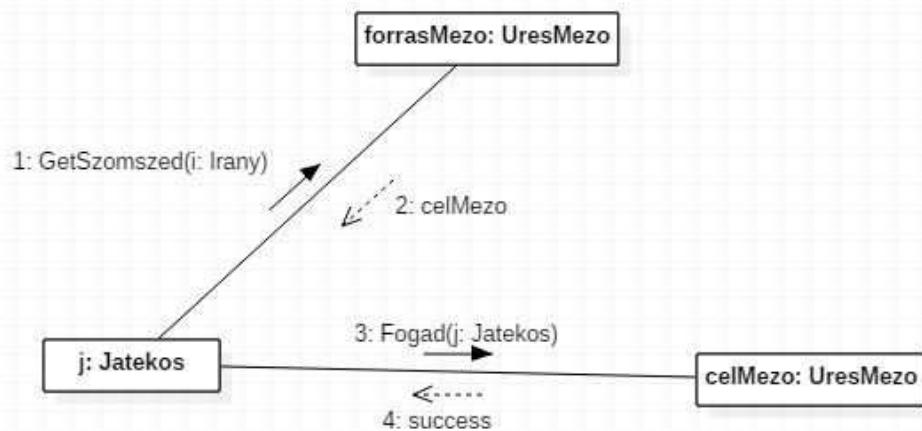
5.4.13 JátékosÜtközik_Léphet

interaction JatekosÜtkozik_Léphet

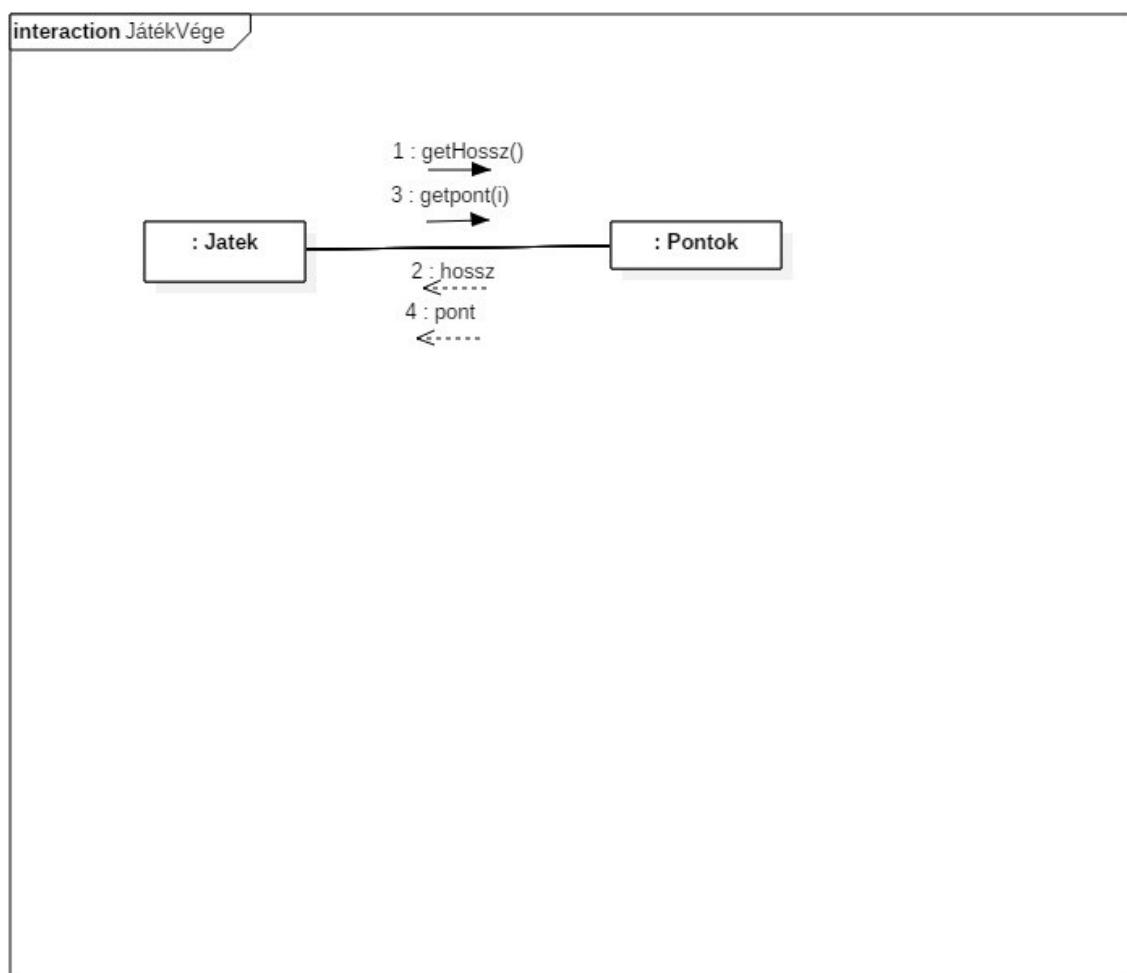


5.4.14 JátékosÜtközik_Meghal

interaction JátékosÜtközik_Meghal



5.4.15 JátékVége

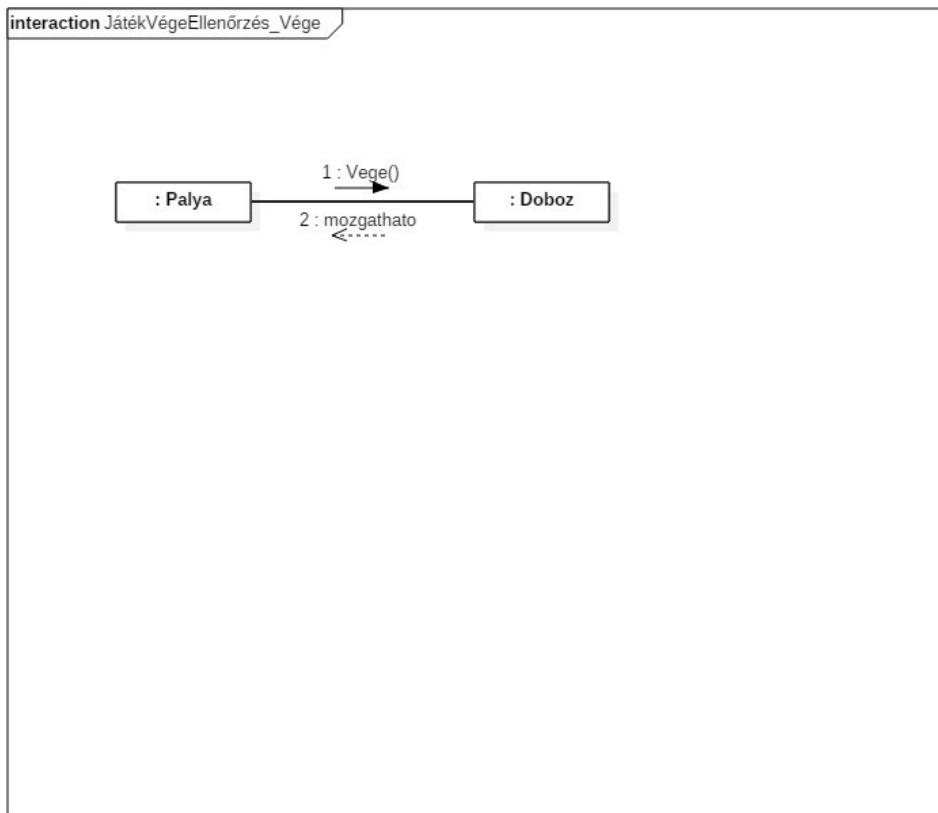


5.4.16 JátékVégeEllenőrzés_NincsVége

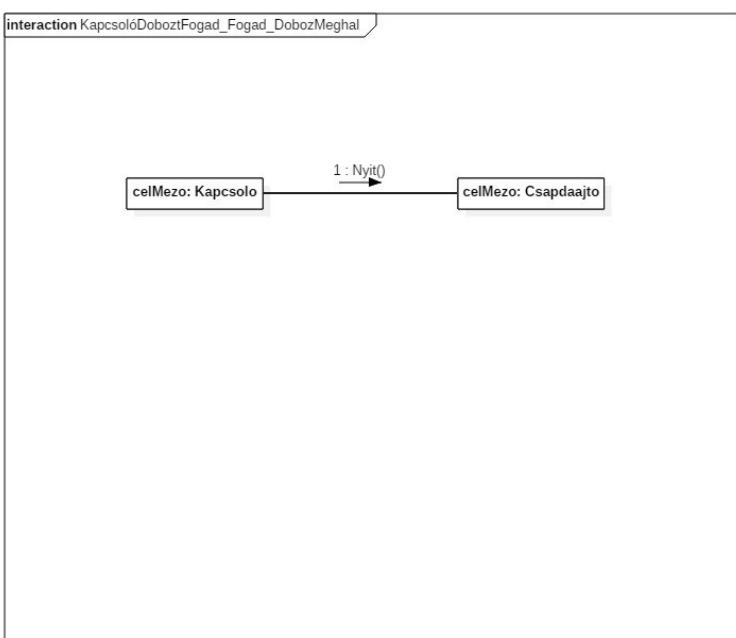
interaction JátékVégeEllenőrzés_NincsVége



5.4.17 JátékVégeEllenőrzés_Vége

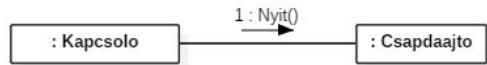


5.4.18 KapcsolóDoboztFogad_Fogad_DobozMeghal

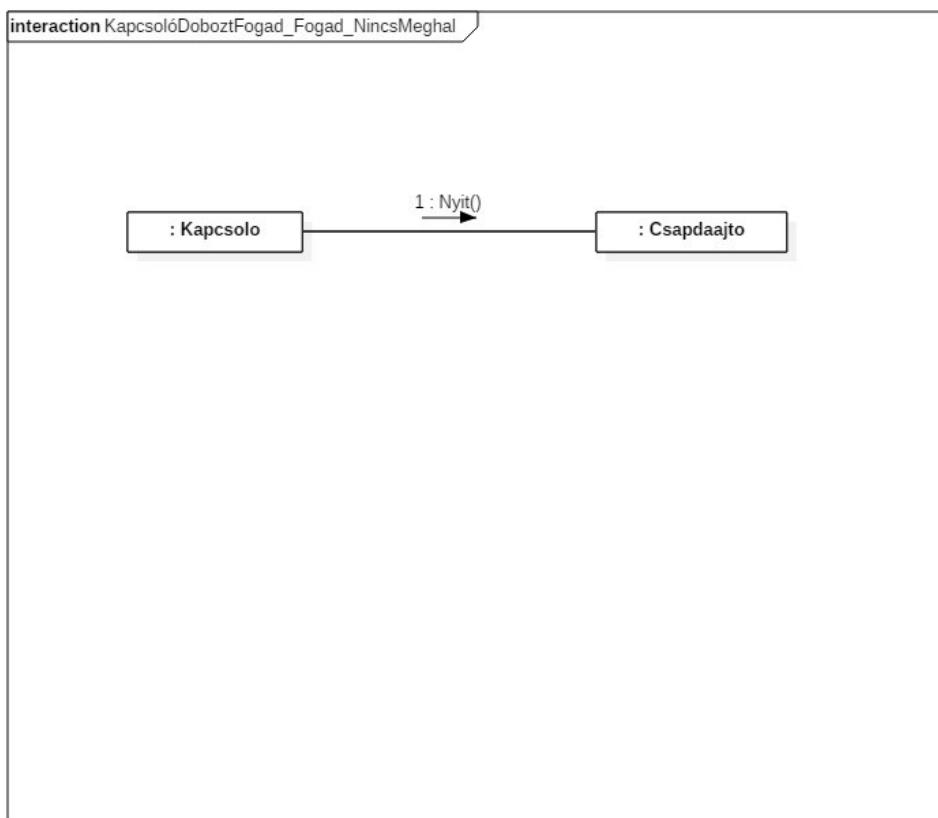


5.4.19 KapcsolóDoboztFogad_Fogad_JátékosMeghal

interaction KapsolóDoboztFogad_Fogad_JátékosMeghal



5.4.20 KapcsolóDoboztFogad_Fogad_NincsMeghal



5.4.21 KapcsolóJátékostFogad_Fogad

interaction KapcsolóJátékostFogad_Fogad



5.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2018.03.10. 17:00	1,5 óra	Holub	Tevékenység: Holub megtervezи a kezelői felületet és dialógusokat (5.2)
2018.03.11. 13:15	4 óra	Horvath	Tevékenység: Szekvencia Diagramok elkészítése. Dokumentum szerkesztése.
2018.03.11 16:30	3 óra	Bottlik	Tevékenység: Kommunikációs diagramok készítése
2018.03.11 17:00	1,5 óra	Holub	Tevékenység: Holub elkészíti a maradék kommunikációs diagramot
2018.03.11 18:30	3 óra	Holub Zsíros Horvath	Tevékenység: Döntés a formai változtatásokról és a dokumentum javítása.
2018.03.11 21:30	5 óra	Zsíros	Tevékenység: Use-case diagramok és azok leírásainak elkészítése. (5.1)

6. Szkeleton beadás

6.1 Fordítási és futtatási útmutató

6.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Celhely.java	4 KB	2018/03/15 17:03	Celhely osztály
Csapdaajto.java	4 KB	2018/03/15 17:13	Csapdaajto osztály
Doboz.java	3 KB	2018/03/15 17:18	Doboz osztály
Fal.java	3 KB	2018/03/15 17:25	Fal osztály
Irany.java	1 KB	2018/03/15 17:31	Irany osztály
Jatek.java	1 KB	2018/03/15 17:33	Jatek osztály, a játék fő osztálya
Jatekos.java	4 KB	2018/03/15 17:35	Jatekos osztály
JeloltDoboz.java	2 KB	2018/03/15 17:37	JeloltDoboz osztály
Kapcsolo.java	3 KB	2018/03/15 17:40	Kapcsolo osztály
Leptetheto.java	2 KB	2018/03/15 17:44	Leptetheto osztály
Lyuk.java	3 KB	2018/03/15 17:46	Lyuk osztály
Main.java	1 KB	2018/03/15 17:47	A főprogram, ami a játékot futtatja
Palya.java	3 KB	2018/03/15 17:50	Palya osztály
Pont.java	2 KB	2018/03/15 17:55	Pont osztály
Pontok.java	2 KB	2018/03/15 17:57	Pontok osztály
Printer.java	1 KB	2018/03/17 21:12	Printer osztály
UresMezo.java	5 KB	2018/03/15 18:02	UresMezo osztály
Test_1.java	4 KB	2018/03/18 17:02	1.1.1 - 1.1.4
Test_2.java	10 KB	2018/03/18 17:12	2.1.1 - 2.2.4
Test_3.java	9 KB	2018/03/18 17:13	3.1.1 - 3.2.4
Test_4.java	10 KB	2018/03/18 17:22	4.1.1 - 4.2.4
Test_5.java	13 KB	2018/03/18 17:40	5.1.1 - 5.2.4
Test_6.java	13 KB	2018/03/18 17:49	6.1.1 - 6.2.4
Test_7.java	13 KB	2018/03/18 17:58	7.1.1 - 7.2.4
Test_8.java	12 KB	2018/03/18 18:20	8.1 - 8.7

6.1.2 Fordítás

A következő parancssori parancccsal:

(a program gyökérmapájában kiadva)

```
javac -cp ".;junit.jar" Tools/Printer.java sokoban/*.java Test/*.java
```

6.1.3 Futtatás

A következő parancccsal:

(a program gyökérmapájában kiadva): java sokoban/Main

6.2 Értékelés

Tag neve	Munka százalékban
Bottlik	20%
Holub	20%
Horváth	20%
Mocsári	20%
Zsíros	20%

6.3 Napló

Kezdet	Időtartam	Résznevők	Leírás
2018.03.15 17:00	4 óra	Holub	Tevékenység: Osztályok felvétele, alapmetódusok lekódolása
2018.03.16 16:00	5 óra	Holub	Tevékenység: Összes kimaradt függvény implementálása
2018.03.16 18:30	9 óra	Zsíros	Tevékenység: kód-kommentezés, ellenőrzés, javítás, osztálydiagram javítása
2018.03.16 16:00	11 óra	Mocsári	Printer osztály implementálása, refaktorálás, tesztesetek tervezése, tesztesetek implementálása, kód átnézése
2018.03.17 15:00	6 óra	Mocsári	Tesztesetek tervezése, tesztesetek implementálása, kód javítása
2018.03.18 17:00	7 óra	Mocsári Zsíros	Tesztesetek tervezése, implementálás
2018.03.18 18:00	1 óra	Holub	Tevékenység: Dokumentumszerkesztés
2018.03.18 22:00	1 óra	Holub	Tevékenység: menü implementálása a tesztekhez
2018.03.18 22:00	2 óra	Bottlik	Korábbi feladatokban a változtatások dokumentálása
2018.03.18 23:00	1 óra	Holub	Tevékenység: Tesztek javítása: írja ki az aktuális teszt jellegét
2018.03.19 0:00	5 óra	Holub Bottlik Zsíros Mocsári Horváth	Tevékenység: Szekvenciadiagramok készítése

7. Prototípus koncepciója

Változás hatása a modellre

Változás leírása:

A változás lényege, hogy a játékosok nem minden esetben képesek eltolni a látákat, mert lehetséges, hogy nem elég erősek hozzá. minden játéknak van egy meghatározott ereje, a látáknak pedig tapadási súrlódási erejük van.

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

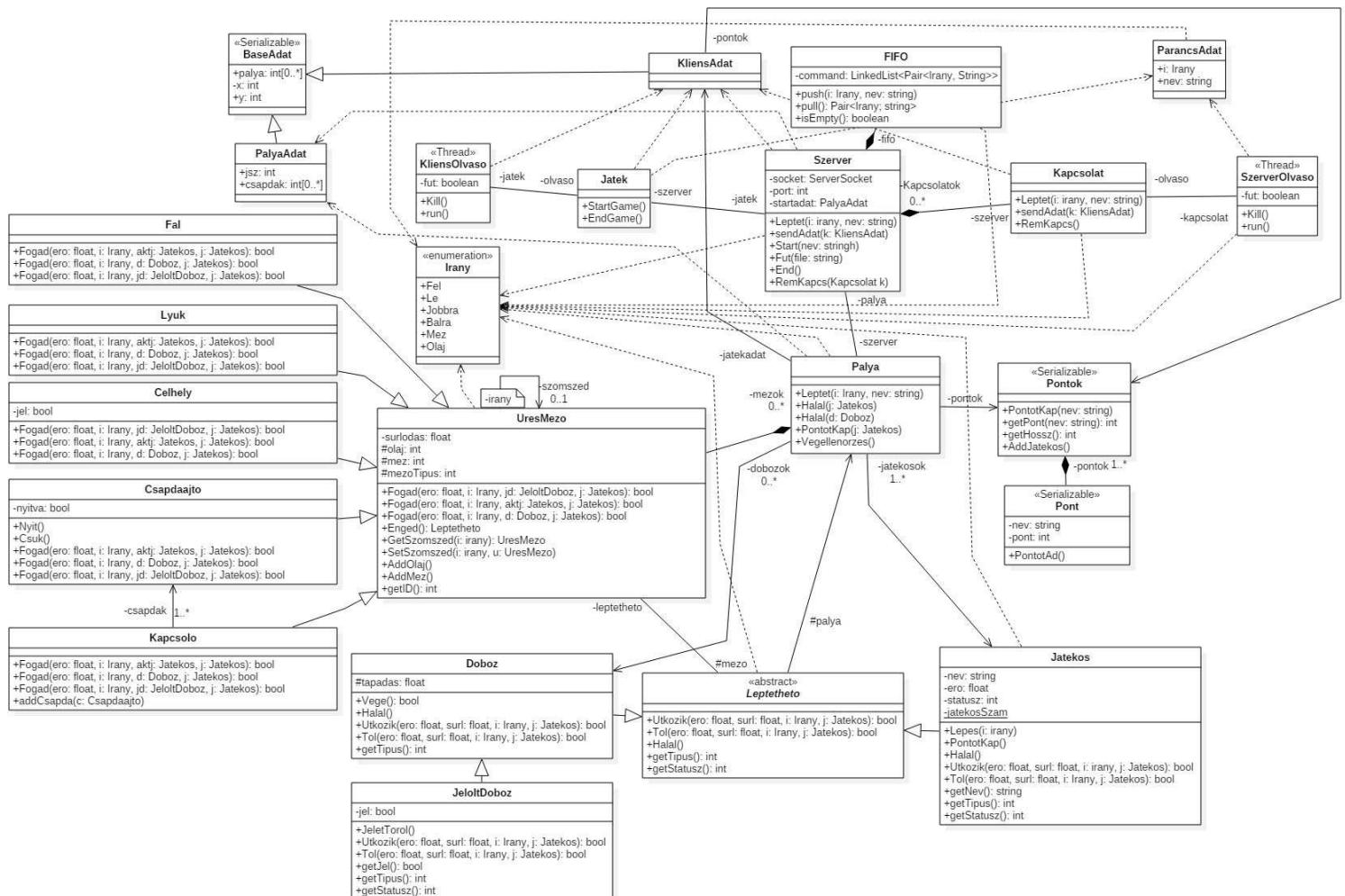
A mezőknek a súrlódása alapból fix szám, befolyásolják a láták súrlódását, azonban ezt a játékosok megváltoztathatják. Egy játékos olajat vagy mézet tehet arra a mezőre, amin éppen áll. Az olaj csúszósabbá teszi a mezőt, csökkentve a rajta lévő látá tapadását, a méz pedig ragacsossá teszi, növelte a 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 játéknak 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	

7. Protótípus koncepciója Módosult osztálydiagram

megegyezés alatt



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 szalon.
- **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 játékos 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).

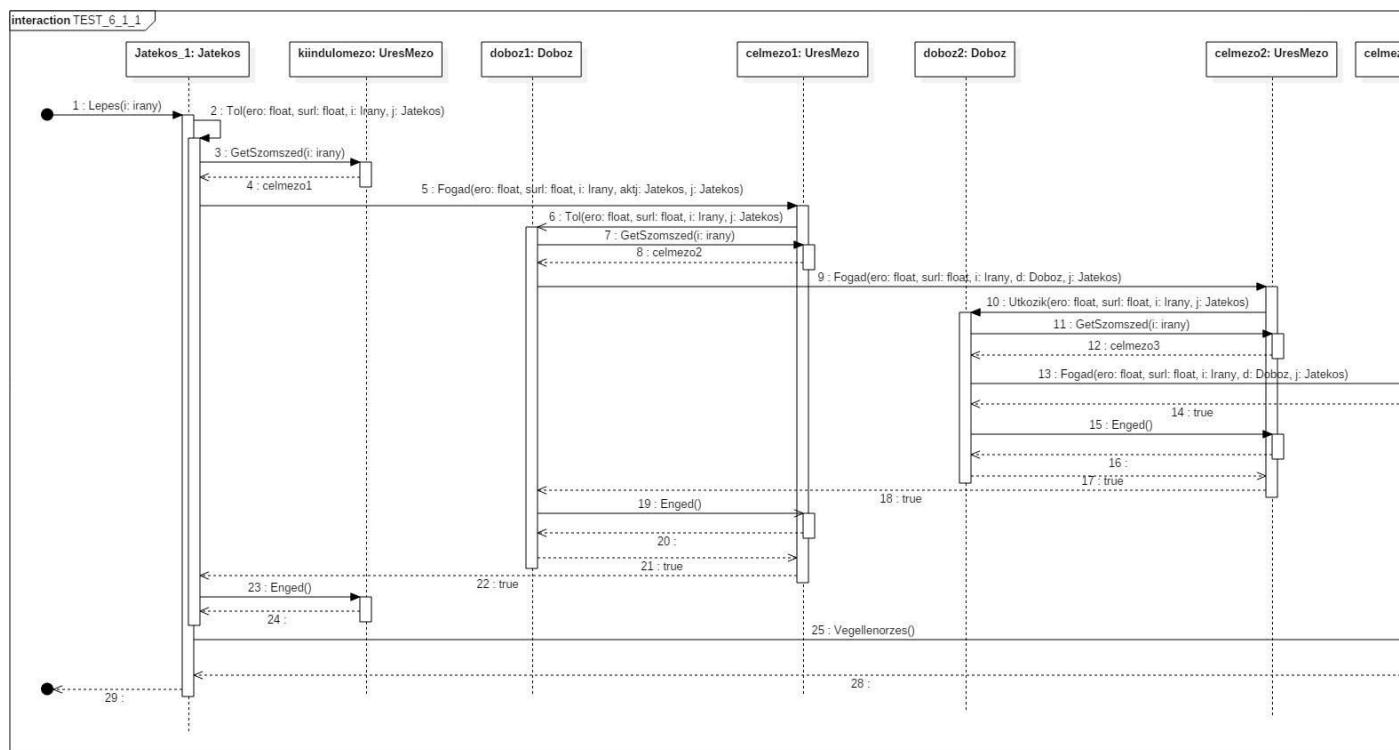
Leptetheto osztályban(és a leszármazottjaiban):

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

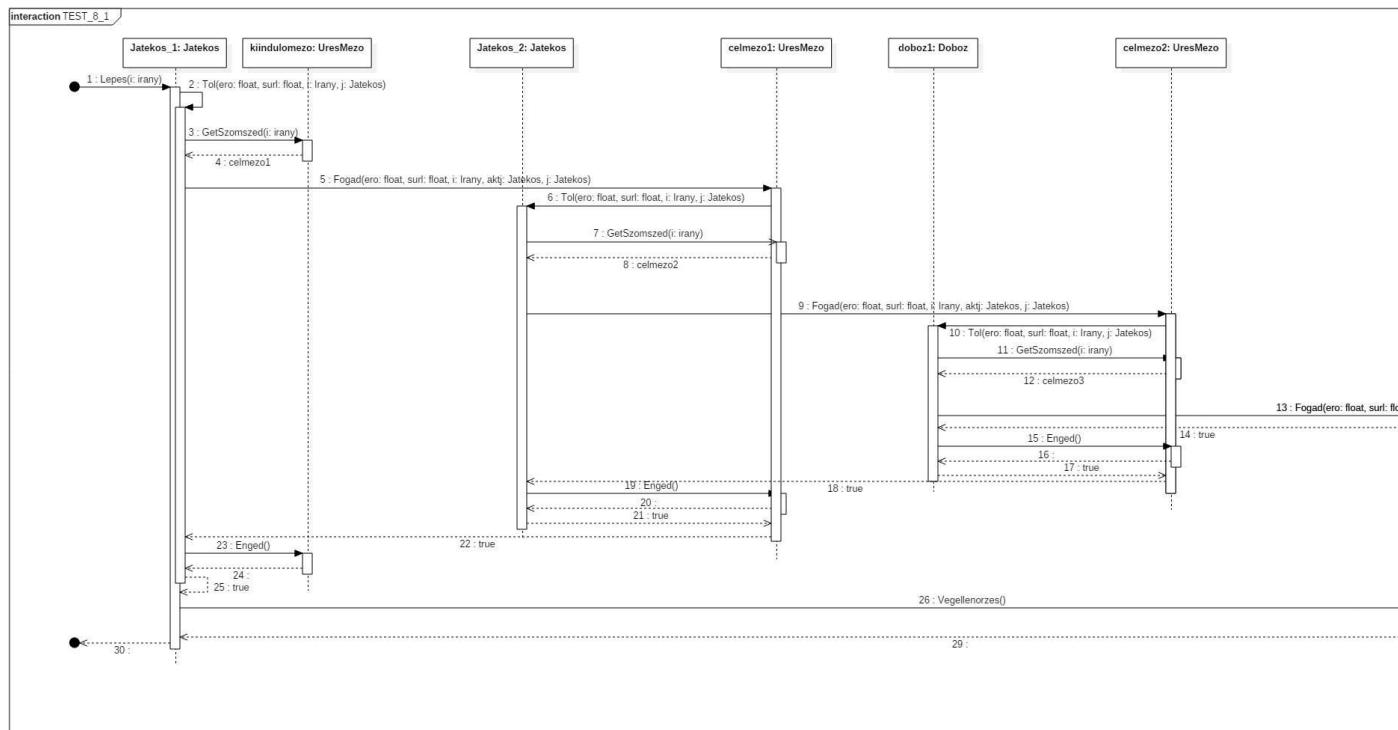
Munkás-láda-láda:



7. Prototípus koncepciója

Munkás-láda-munkás-láda:

megegyezés alatt



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, Kapcsolodás <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.]

7. Prototípus koncepciója Parancsok a kezdeti állapotban (kliens):

megegyezes alatt

<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 fómenübe, 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 halma 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 jelölnek.

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:

7. Prototípus koncepciója

megegyezes alatt

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 mennyiségenek száma. (max 9)	
Olaj	8	A mezőn lévő olaj mennyiségenek száma. (max 9)	

7. Prototípus koncepciója *megegyezesalatt*

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)
Mocsil Tol(Irany, Jatekos)
UresMezo GetSzomszed(Irany)
Return: sokoban.mezok.Lyuk@2704b43
Lyuk Fogad(Irany, Jatekos, Jatekos)
Mocsil 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 sikkerrel 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	Részttvevők	Leírás
2018.03.23. 21:00	2 óra	Zsíros	Tevékenység: programkód javítása a követelmények meg változásának 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, vélegesítés

8. Részletes tervez

8.1 Osztályok és metódusok tervei.

8.1.1 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

8.1.2 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 űrosztálya.

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 űrosztálya.

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 űrosztálya.

8.1.3 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 toltható-e. Ha egy dobozt lyukba tolunk, az osztály felelőssége lekezelni a Doboz eltüntetését a pályáról. Az osztály lekérdezhető, hogy a Doboz tud-e még mozogni, vagy sem, ez a játék végének ellenőrzésére ad lehetőséget.

Őrosztályok

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 értékkal.
- **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.

8.1.4 Fal

Felelősség

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

Őrosztályok

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).

8.1.5 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**

8.1.6 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

8.1.7 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 lánkok 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 Lehetethető interfész-beli 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 tollt, „surl” pedig a mező súrlódása.
- **bool Tol(float ero, float surl, Irany i, Jatekos j):** Ha a játékost játékos tolla 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 tollt, „surl” pedig a mező súrlódása.
- **string getNev():** Visszaadja a játékos nevét.
- **int getTipus():** Visszaadja a játékos típusát.
- **int getStatusz():** Visszaadja a játékos státuszát.

8.1.8 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 lehetséges 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

Lehetethető → 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.

8. Részletes tervezet

megegyezésen alatt

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).

8.1.9 Kapcsolo

Felelősség

A pálya Kapcsolo típusú mezőjét kezeli, figyeli a Kapcsolo-ra kerülő objektumokat és végrehoztja az ennek következményeként kialakuló szükséges módosításokat. Ha egy játékos egy lánát 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 ősosztály.

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 ősosztály.

addCspada(Csapdaajto c): Hozzáadja a paraméterként megadott csapdaajtót a kapcsoló által kapcsolható csapdákhoz.

8.1.10 Leptetheto <<abstract>>

Felelősség

Abstract osztály, ami a leptethető 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).

8.1.11 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.

8.1.12 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 ellenörzi, 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.

-Sserver 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.

8.1.13 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.

8.1.14 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 játékest adunk hozzá, hogy az ő pontjait is nyilvántarthassuk.

8.1.15 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 „surlodás” 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:

8.1.16 BaseAdat

Felelősség

Alapadatokat tartalmaz a pályáról, serializált.

Interface-ek

Serializable

Attribútumok

-int[] palya: Tartalmazza a pálya mezőinek ID-jait.

-int x: x koordináta

-int y: y koordináta

8.1.17 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.

8.1.18 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) lépéseiit 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.

8.1.19 KliensAdat

Felelősség

A kliensek adatait nyilvántartó osztály.

Ősosztályok

KliensAdat -> BaseAdat

Attribútumok

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

8.1.20 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.

8.1.21 PalyaAdat**Felelősség**

A parancsok kezeléséért felelős osztály.

Ósosztályok

PalyaAdat -> BaseAdat

Attribútumok

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

8.1.22 ParancsAdat**Felelősség**

A parancsok kezeléséért felelős osztály.

Attribútumok

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

8.1.23 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.

8.1.24 SzerverOlvaso

Felelősség

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

Ósosztályok

SzerverOlvaso -> Thread

Attribútumok

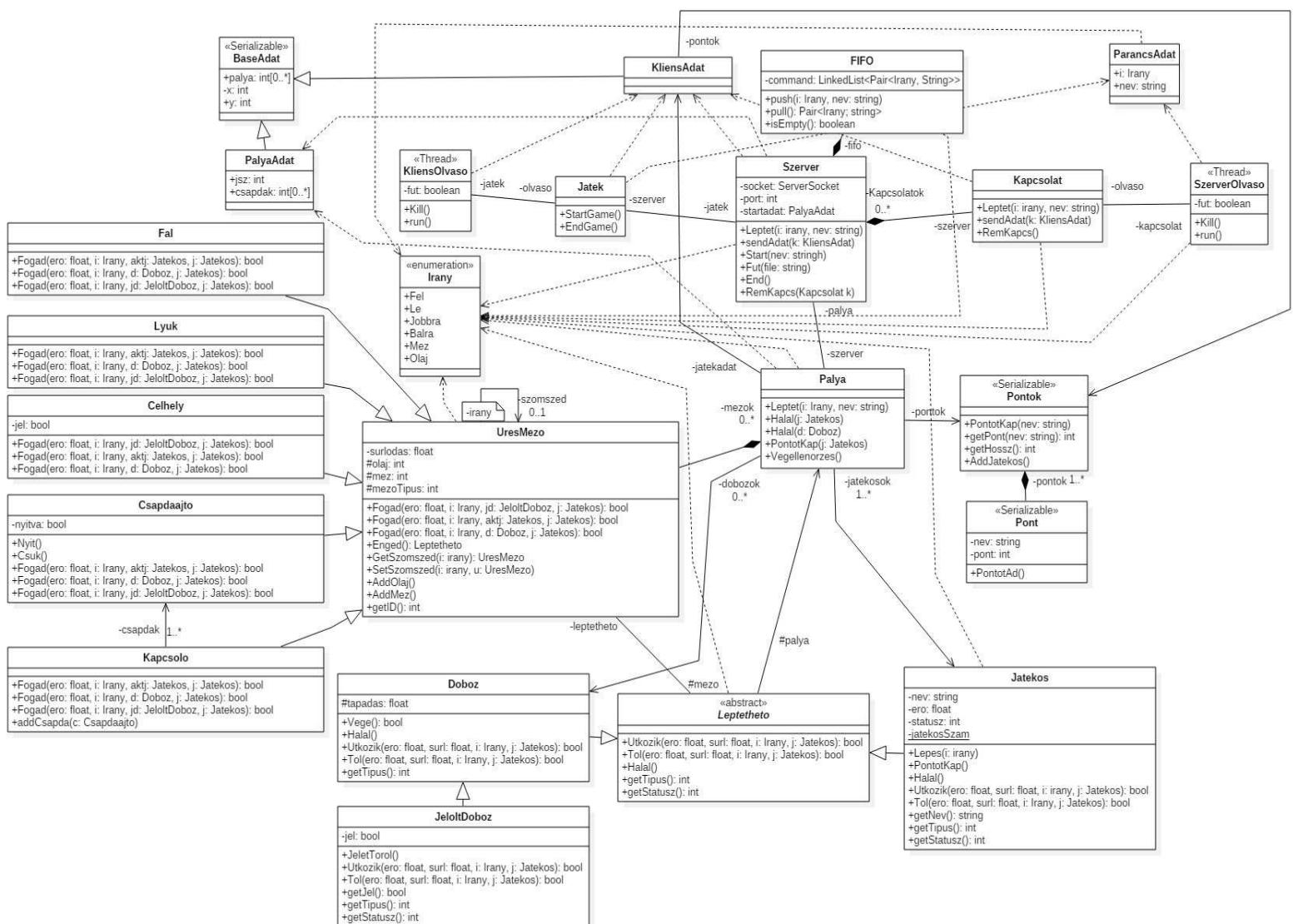
-bool fut: Megadja, hogy éppen fut-e a szál.

-Kapcsolat kapcsolat: Referencia a kapcsolatra.

Metódusok

Kill(): Leállítja az osztály futását, ezzel véget vetve a szálnak.

run(): A szál futtatása.



8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

8.2.1 Teszteset1

Leírás

Egy játékos egy másik játékos belelők 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:

5. k
6. Mocsi
7. localhost
8. 1111

Szerver oldalon:

9. start
10. 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, Kapcsolodás <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 játékészek 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

8.2.2 Teszteset2**Leírás**

8. Részletes tervezet

megegyezés alatt

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

Ellenőrzött funkcionális, 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:

5. k
6. Mocsi
7. localhost
8. 1111

Szerver oldalon:

9. start
10. d

Elvárt kimenet

Kliens oldalon:

[MENU]: Opciók:

[MENU]: 1, Start szerver <S>

[MENU]: 2, Kapcsolódás <K>

[MENU]: 3, Quit <Q>

[MENU]: k

[Client]:

[Client]: KLIENS MOD [Client]:

[Client]: Játékos 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, Kapcsolodás <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 játékészek 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
              Return
  00001000 00001000 00001000 00001000 00001000
  00001000 00000000 10000000 00002000 00001000
  00001000 00001000 00001000 00001000 00001000

```

8.2.3 Teszteset3**Leírás**

8. Részletes tervezek

megegyezesalatt

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

Ellenőrzött funkcionális, 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:

5. k
6. Mocsi
7. localhost
8. 1111

Szerver oldalon:

9. start
10. d

Elvárt kimenet

Kliens oldalon:

```
[MENU]: Opciók:  
[MENU]: 1, Start szerver <S>  
[MENU]: 2, Kapcsolódás <K>  
[MENU]: 3, Quit <Q>  
[MENU]: k  
[Client]:  
[Client]: KLIENS MOD [Client]:  
[Client]: Játékos 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, Kapcsolodás <K>
 [MENU]: 3, Quit <Q>
 [MENU]: s
 [SERVER]:
 [SERVER]: SZERVER MOD [SERVER]:
 [SERVER]: Játékos neve: Nova
 [SERVER]:
 [SERVER]: Port: 1111
 [SERVER]:
 [SERVER]: File path: Test3
 [SERVER]: Lehetséges játékések 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 Lepet(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

8.2.4 Teszteset4

Leírás

Egy játékos rátol egy látát 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]: Opciók:

[MENU]: 1, Start szerver <S>

[MENU]: 2, Kapcsolódás <K>

[MENU]: 3, Quit <Q>

[MENU]: s

[SERVER]:

[SERVER]: SZERVER MOD [SERVER]:

[SERVER]: Játékos neve: Nova

[SERVER]:

[SERVER]: Port: 1111

[SERVER]:

[SERVER]: File path: test4

[SERVER]: Lehetséges játékések száma: 2

[SERVER]:

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

Kapcsolo, csapdaajtó összerendelések

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

00001000 00001000 00001000 00001000 00001000

00001000 10000000 20004000 00005000 00001000

00001000 00001000 00001000 00001000 00001000

d

Palya Lepet(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
```

8.2.5 Teszteset5

Leírás

A játékos mézet helyez egy mezőre, majd megpróbálja eltolni a látót arról a mezőről. Ez sikertelenségen 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]: Opciók:  

[MENU]: 1, Start szerver <S>  

[MENU]: 2, Kapcsolodás <K>  

[MENU]: 3, Quit <Q>  

[MENU]: [SERVER]:  

[SERVER]: SZERVER MOD [SERVER]:  

[SERVER]: Játékos 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 00000000 00000000 00000000 00000000 00001000  

00001000 10000000 20000000 00000000 00000000 00001000  

00001000 00001000 00001000 00001000 00001000 00001000
```

Palya Leptet(Irany, String)
Nova getNev()

8. Részletes tervez

megegyezesalatt

```

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

```

8. Részletes tervezek

meggyezésben

```
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)
```

8. Részletes tervez

meggyezésben

```
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
```

8. Részletes tervezek

00001000 00000000 20000000 10000090 00000000 00001000
 00001000 00001000 00001000 00001000 00001000 00001000

megegyezesalatt

```

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()
```

8. Részletes tervez

megegyezesalatt

```

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  Leptett(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

```

```

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

```

8.2.6 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, jelöldoboz 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]: Opciók:  

[MENU]: 1, Start szerver <S>  

[MENU]: 2, Kapcsolódás <K>  

[MENU]: 3, Quit <Q>  

[MENU]: [SERVER]:  

[SERVER]: SZERVER MOD [SERVER]:  

[SERVER]: Játékos 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)

8. Részletes tervez

meggyezésben

```

    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@1dfa84
                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: true
                                    UresMezo      Enged()
                                    Return
                                    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()

```

8. Részletes tervez

megegyezesalatt

```

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()
```

8. Részletes tervez

megegyezesalatt

```

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()

```

8. Részletes tervez

megegyezesalatt

```

Return: 1
Pontok getPont(int)
Return: pontok.get(0)
Pont getNev()
Return: Nova
Pont getPont()
Return: I
Nova: I Pontok getHossz()
Return: I

Palya Leptet(Irany, String)
Nova getNev()
Return: Nova
Nova Lepes(Irany)
Nova Tol(Irany, Jatekos)
UresMezo GetSzomszed(Irany)
Return: sokoban.mezok.Celhely@1dfa84
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
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: I
Pontok getPont(int)
Return: pontok.get(0)
Pont getNev()
Return: Nova
Pont getPont()
Return: I
Nova: I Pontok getHossz()
Return: I

Palya Leptet(Irany, String)
Nova getNev()
Return: Nova
Nova Lepes(Irany)
Nova Tol(Irany, Jatekos)
UresMezo GetSzomszed(Irany)
Return: sokoban.mezok.UresMezo@e7f0bf

```

8. Részletes tervez

megegyezeselatt

```

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)

```

8. Részletes tervez

meggyezésben

```

    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: I
Pontok  getPont(int)
Return: pontok.get(0)
Pont  getNev()
Return: Nova
Pont  getPont()
Return: I
Nova: 1      Pontok  getHossz()
Return: I

Palya  Leptett(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: I
Pontok  getPont(int)
Return: pontok.get(0)
Pont  getNev()
Return: Nova

```

8. Részletes tervezek

meggyezes alatt

```

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@1dfa84
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

```

8.3 A tesztelést támogató programok tervezése

nem használtunk támogató segédprogramokat

8.4 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

10. Prototípus beadása

10.1 Fordítási és futtatási útmutató

10.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Data.java	158 B	2018/03/31 19:17	Data osztály
palya.mocsi	545 B	2018/03/31 19:17	Egy generált pálya
BaseAdat.java	911 B	2018/04/22 21:26	BaseAdat osztály
FIFO.java	963 B	2018/04/22 21:26	FIFO osztály
Kapcsolat.java	2,3 kB	2018/04/22 21:26	Kapcsolat osztály
KliensAdat.java	890 B	2018/04/22 21:26	KliensAdat osztály
KliensOlvaso.java	1,66 kB	2018/04/22 21:26	KliensOlvaso osztály
PalyaAdat.java	959 B	2018/04/22 21:26	PalyaAdat osztály
ParancsAdat.java	885 B	2018/04/22 21:26	ParancsAdat osztály
Szerver.java	5,41 kB	2018/04/22 21:26	Szerver osztály
SzerverOlvaso.java	1,46 kB	2018/04/22 21:26	SzerverOlvaso osztály
Doboz.java	3,76 kB	2018/03/15 17:18	Doboz osztály
Jatekos.java	4,58 kB	2018/03/15 17:35	Jatekos osztály
JeloltDoboz.java	3,77 kB	2018/03/15 17:37	JeloltDoboz osztály
Leptetheto.java	1,73 kB	2018/03/15 17:44	Leptetheto osztály
Celhely.java	4,1 kB	2018/03/15 17:03	Celhely osztály
Csapdaajto.java	4,85 kB	2018/03/15 17:13	Csapdaajto osztály
Fal.java	3,56 kB	2018/03/15 17:25	Fal osztály
Kapcsolo.java	4,56 kB	2018/03/15 17:40	Kapcsolo osztály
Lyuk.java	3,62 kB	2018/03/15 17:46	Lyuk osztály
UresMezo.java	6,73 kB	2018/03/15 18:02	UresMezo osztály
Irany.java	498 B	2018/03/15 17:31	Irany osztály
Jatek.java	11 kB	2018/03/15 17:33	Jatek osztály, a játék fő osztálya
Main.java	17,3 KB	2018/03/15 17:47	A főprogram, ami a játékot futtatja
Palya.java	6,72 kB	2018/03/15 17:50	Palya osztály
Pont.java	1,55 kB	2018/03/15 17:55	Pont osztály
Pontok.java	2,51 kB	2018/03/15 17:57	Pontok osztály
Printer.java	853 B	2018/03/17 21:12	Printer osztály
Test1.mocsi	1 kB	2018/04/09 21:13	Tesztpállya
Test2.mocsi	1 kB	2018/04/09 21:13	Tesztpállya
Test3.mocsi	1 kB	2018/04/09 21:13	Tesztpállya
Test4.mocsi	1 kB	2018/04/09 21:13	Tesztpállya
Test5.mocsi	1 kB	2018/04/09 21:13	Tesztpállya
Test6.mocsi	1 kB	2018/04/09 21:13	Tesztpállya

10.1.2 Fordítás

A következő parancssori parancccsal:

(a program gyökérmapájában kiadva)

javac data/*.java server/*.java Tools/Printer.java sokoban/*.java

sokoban/leptethetok/*.java sokoban/mezok/*.java

Hint: A hszk gépein a PATH változóban nincs bent mindig a javac és a java

Hint2: A fájlok útvonalainál lehet, hogy kelleni fog az elérési út elő egy „/” jel.

10.1.3 Futtatás

A következő parancssori parancccsal:

(a program gyökérmapájában kiadva)

java sokoban/Main

10.2 Tesztek jegyzőkönyvei

10.2.1 Teszteset1

Tesztelő neve	Mocsári
Teszt időpontja	2018.04.09.

10.2.2 Teszteset2

Tesztelő neve	Mocsári
Teszt időpontja	2018.04.09.

10.2.3 Teszteset3

Tesztelő neve	Mocsári
Teszt időpontja	2018.04.09.

10.2.4 Teszteset4

Tesztelő neve	Mocsári
Teszt időpontja	2018.04.09.

10.2.5 Teszteset5

Tesztelő neve	Mocsári
Teszt időpontja	2018.04.09.

10.2.6 Teszteset6

Tesztelő neve	Mocsári
Teszt időpontja	2018.04.09.

10.3 Értékelés

Tag neve	Munka százalékban
Holub Csongor	20
Horváth Gergő	20
Mocsári András	20
Zsíros Bálint	20
Bottlik Judit	20

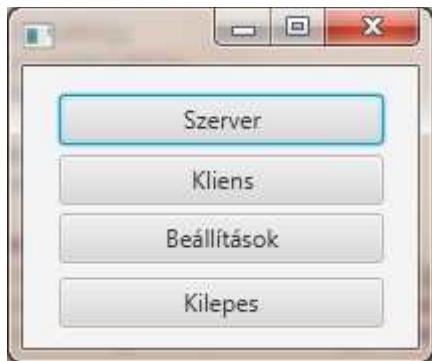
Napló

Kezdet	Időtartam	Részttvevők	Leírás
2018.04.23. 16:30	3,5 óra	Zsíros	Tevékenység: kód, kommentek javítása, kiegészítése
2018.04.23 20:00	1 óra	Bottlik	Dokumentáció készítése
2018.04.23 22:00	3 óra	Holub	Dokumentáció szerkesztése

18.Grafikus felület specifikációja

12.2 A grafikus interfész

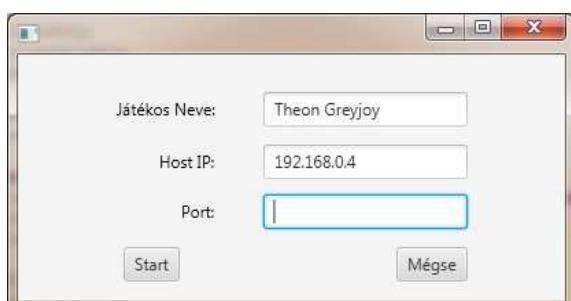
A programot elindítva egy menübe jutunk, ahol kiválaszthatjuk, hogy mit szeretnénk csinálni:



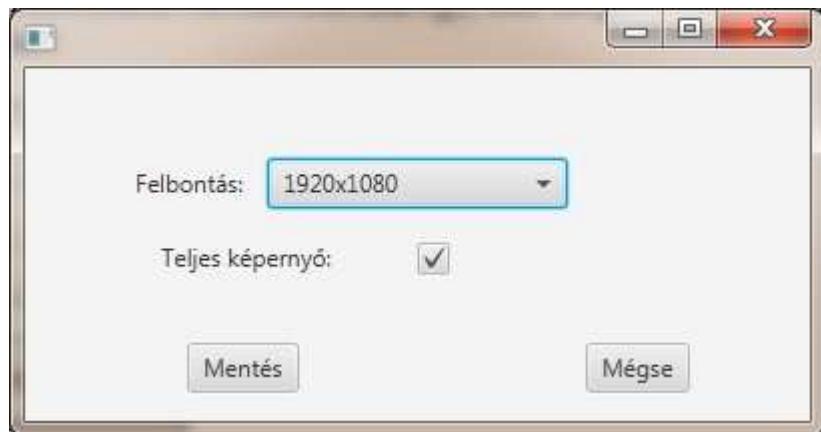
1, A Szerver gomb lenyomásával szervert hozhatunk létre. További kliensek nélkül két játékossal játszatunk. Itt megadhatjuk a játékosneveket, a portot a szerverhez, illetve kiválaszthatjuk a pályát, ami alul meg is jelenik:



2, A Kliens gomb lenyomásával kliensként csatlakozhatunk egy, már meglévő szerverhez, ha beírjuk a megfelelő paramétereket:



3, A Beállítások gombbal beállíthatjuk a felbontást és a teljes képernyő módot:



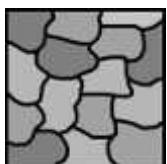
4, A Kilépés gombbal kilépünk a játékból.

Ha ténylegesen elindul a játék, megjelenik a pálya, rajta pedig a játékosok és minden pályaelem. Itt irányíthatják a játékosok a munkásokat a W-A-S-D és I-J-K-L gombok segítségével, illetve a kiléphetnek a menübe, ha rákattintanak a gombra.
Jobboldalon látható a játékosok aktuális pontszáma:

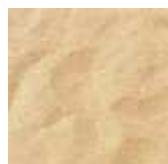


A megjelenő pályaelemek:

Játékosok:



Fal:



Üres mező:



Doboz:



Jelölt doboz:



Célhely:



Lyuk:



Kapcsoló:



Mézes mező:



Olajos mező:



Olajos-mézes mező:

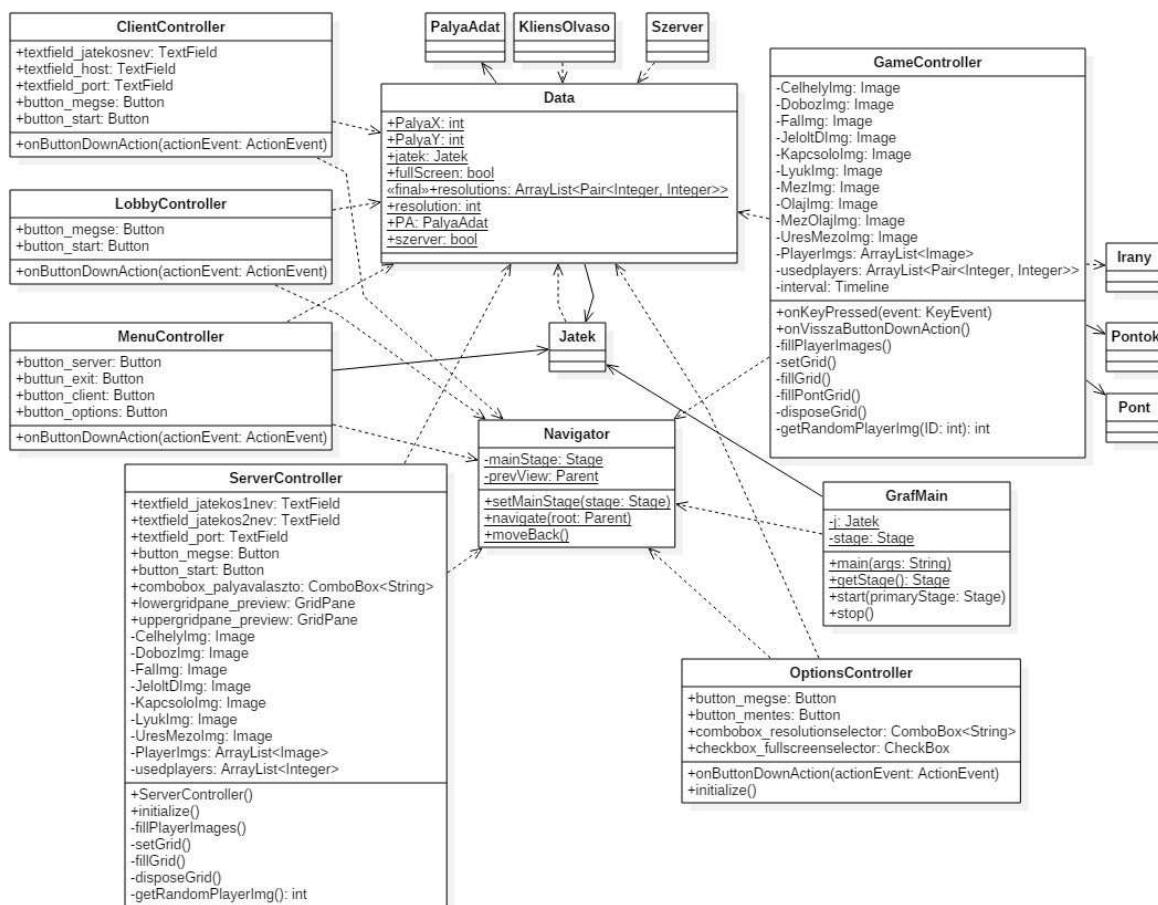
11.2 A grafikus rendszer architektúrája

11.2.1 A felület működési elve

A protoban használt maint felváltja egy grafikus felületet, amely a menünk lesz. Ebből a menüből lehet elindítani a játékot, és grafikusan végrehajtani a már működő proto funkciókat.

A felhasználói interakciókat átadjuk a protoban elkészített programnak, ami elvégzi a megfelelő műveleteket, majd kirajzolja az eredményt a képernyőre

11.2.2 A felület osztály-struktúrája



11.3 A grafikus objektumok felsorolása

Új osztályok:

11.3.1 ClientController

Felelősség

Ez az osztály kezeli az új kliens csatlakozásához kellő grafikus elemeket és megfelelő paraméterek esetén csatlakoztatja a játékost a megadott játékhoz.

Attribútumok

- + **TextField textfield_jatekosnev:** Ide írhatjuk a játékos nevét.
- + **TextField textfield_host:** Ide írhatjuk a host IP-címét, amihez csatlakozni szeretnénk.
- + **TextField textfield_port:** Ide írhatjuk a portot, amihez kapcsolódni szeretnénk.
- + **Button button_megse:** Gomb a visszalépéshez.
- + **Button button_start:** Gomb a játék elindításához.

Metódusok

onButtonDownAction(ActionEvent actionEvent): Kezeli a start és a mégse gombok lenyomását. Ellenőrzést végez, hogy az adatok helyesen vannak-e beírva, majd létrehozza a játékos kapcsolódását, vagy visszalép (a gombtól függően).

11.3.2 GameController

Felelősség

Ez az osztály felelős az aktív játék lebonyolításáért. Folyamatosan kezeli az aktuális változásokat, betölti a képeket, mindenkor megjeleníti a pillanatnyi állapotot.

Attribútumok

- **Image CelhelyImg:** A célhely pályaelem képe.
- **Image DobozImg:** A doboz pályaelem képe.
- **Image FallImg:** A fal pályaelem képe.
- **Image JeloltDImg:** A jelölt doboz pályaelem képe.
- **Image KapcsoloImg:** A kapcsoló pályaelem képe.
- **Image LyukImg:** A lyuk pályaelem képe.
- **Image MezImg:** A mézes mező pályaelem képe.
- **Image OlajImg:** A olajos mező pályaelem képe.
- **Image MezOlajImg:** A mézes-olajos mező pályaelem képe.
- **Image UresMezoImg:** Az üres mező pályaelem képe.
- **ArrayList<Image> PlayerImgs:** A játékosok képei.
- **ArrayList<Pair<Integer, Integer>> usedplayers:** A játékosok pontjait nyilvántartó számpárok halmaza.
- **Timeline interval:** JavaFX Timeline változó a játék szekvenciájának kezelésére.

Metódusok

onKeyPressed(KeyEvent event): Kezeli a játékosok gomblenyomásait és elküldi a szervernek az információt.

onVisszaButtonDownAction(): A „vissza” gomb lenyomásának hatására befejezi a játékot és újra megjelenik a menü.

fillPlayerImages(): Betölti a játékosok képeit.

setGrid(): Aktualizálja az információkat a megjelenítendő pályaképről.

fillGrid(): Megjeleníti az aktuális pályát.

fillPontGrid(): Aktualizálja a játékosok pontjait.

disposeGrid(): Törli a játékállás (aktualitását vesztett) megjelenítését.

int getRandomPlayerImg(int ID): A játékos kap egy random játékos-kinézetet a tárolt játékos-kinézetek közül.

11.3.3 LobbyController

Felelősség

Ez az osztály felelős a szerver indításához tartozó almenü megjelenítéséért és kezeléséért.

Attribútumok

+ **Button button_megse**: Gomb a visszalépéshez.

+ **Button button_start**: Gomb az indításhoz.

Metódusok

onButtonDownAction(ActionEvent actionEvent): Kezeli a start és a mégse gombok lenyomását. „Start” esetén elindítja a megfelelő szervert, „mégse” esetén visszalép.

11.3.4 MenuController

Felelősség

Ez az osztály felelős a (kezdetben is megjelenő) menü megjelenítéséért és kezeléséért.

Attribútumok

+ **Button button_szerver**: Gomb a szerver-módú játékhöz.

+ **Button button_exit**: Gomb a kilépéshez.

+ **Button button_client**: Gomb a kliensként való csatlakozáshoz.

+ **Button button_options**: Gomb a beállítások megjelenítéséhez.

Metódusok

onButtonDownAction(ActionEvent actionEvent): Kezeli a gombok lenyomását az adott gomb funkciója szerint.

11.3.5 OptionsController

Felelősség

Ez az osztály felelős a „beállítások” almenü megjelenítéséért és kezeléséért.

Attribútumok

+ **Button button_megse**: Gomb a visszalépéshez.

+ **Button button_start**: Gomb a mentéshez.

- + **ComboBox<String> combobox_resolutionselector:** ComboBox a felbontások nyilvántartásához.
- + **CheckBox checkbox_fullscreenselector:** CheckBox a teljes képernyős üzemmódhoz.

Metódusok

- onButtonDownAction(ActionEvent actionEvent):** Kezeli a gombok lenyomását az adott gomb funkciója szerint.
- initialize():** Beállítja az alapbeállításokat.

11.3.6 ServerController

Felelősség

Ez az osztály felelős a szerver módban megkezdett játék almenűjének megjelenítéséért és működéséért.

Attribútumok

- + **TextField textfield_jatekos1nev:** TextField az első játékos nevének beírásához.
- + **TextField textfield_jatekos2nev:** TextField a második játékos nevének beírásához.
- + **TextField textfield_port:** TextField a port megadásához.
- + **Button button_megse:** Gomb a visszalépéshez.
- + **Button button_start:** Gomb a játék elindításához.
- + **ComboBox<String> combobox_palyavalaszto:** ComboBox a lehetséges pályákkal.
- + **GridPane lowergridpane_preview:** JavaFX GridPane változó a pályakép megjelenítéséhez.
- + **GridPane uppergridpane_preview:** JavaFX GridPane változó a pályakép megjelenítéséhez.
- **Image CelhelyImg:** A célhely pályaelem képe.
- **Image DobozImg:** A doboz pályaelem képe.
- **Image FallImg:** A fal pályaelem képe.
- **Image JeloltDImg:** A jelölt doboz pályaelem képe.
- **Image KapcsoloImg:** A kapcsoló pályaelem képe.
- **Image LyukImg:** A lyuk pályaelem képe.
- **Image UresMezoImg:** Az üres mező pályaelem képe.
- **ArrayList<Image> PlayerImgs:** A játékosok képei.
- **ArrayList<Pair<Integer, Integer>> usedplayers:** A játékosok pontjait nyilvántartó számpárok halmaza.

Metódusok

- onButtonDownAction():** A „start” és „mégse” gombok lenyomását kezeli.
- initialize():** Beállítja az alapbeállításokat.
- fillPlayerImages():** Betölti a játékosok képeit.
- setGrid():** Aktualizálja az információkat a megjelenítendő pályaképről.
- fillGrid():** Megjeleníti az aktuális pályát.
- fillPontGrid():** Aktualizálja a játékosok pontjait.

disposeGrid(): Törli a játékállás (aktuálisát vesztett) megjelenítését.

int getRandomPlayerImg(int ID): A játékos kap egy random játékos-kinézetet a tárolt játékos-kinézetek közül.

11.3.7 GrafMain

Felelősség

A program belépési pontja, ez az osztály valósítja meg a legmagasabb szintű programkezelést, magát a menüt is ő indítja el, illetve kiléphet a programból.

Őszosztályok

Application → GrafMain

Attribútumok

~**Jatek j**: Referencia a játékra.

- **Stage stage**: Referencia az adott menürészre.

Metódusok

main(): A program belépési pontja, elindítja a kellő függvényeket.

Stage getStage(): Az adott menürészről adja vissza.

start(Stage primaryStage): A menü elindítása a megadott menüréssel.

stop: Kilépés.

11.3.8 Data

Felelősség

Ez az osztály felelős az aktuális játék információinak nyilvántartásáért.

Attribútumok

+ **int PalyaX**: Megadja a pálya szélességét.

+ **int PalyaY**: Megadja a pálya magasságát.

+ **Jatek jatek**: Referencia a játékra.

+ **boolean fullScreen**: Változó, ami megadja, hogy teljes képernyős üzemmódban vagyunk-e.

+ **ArrayList<Pair<Integer, Integer>> resolutions**: Egész számpárok a felbontások nyilvántartásához.

+ **int resolution**: A felbontás nyilvántartása.

+ **PalyaAdat PA**: Referencia a PalyaAdat-ra.

+ **boolean szerver**: Megadja a szerver állapotát.

11.3.9 Navigator

Felelősség

Ez az osztály felelős az aktuális menürész eléréséért.

Attribútumok

- Stage mainStage: Az adott állapot (menürész).
- Parent PrewView: A „szülő” állapot.

Metódusok

- setMainStage(Stage stage): Az állapot beállítása.
- navigate(Parent root): Navigálás az állapotok között.
- moveBack(): Visszalépés a „szülő” állapotba.

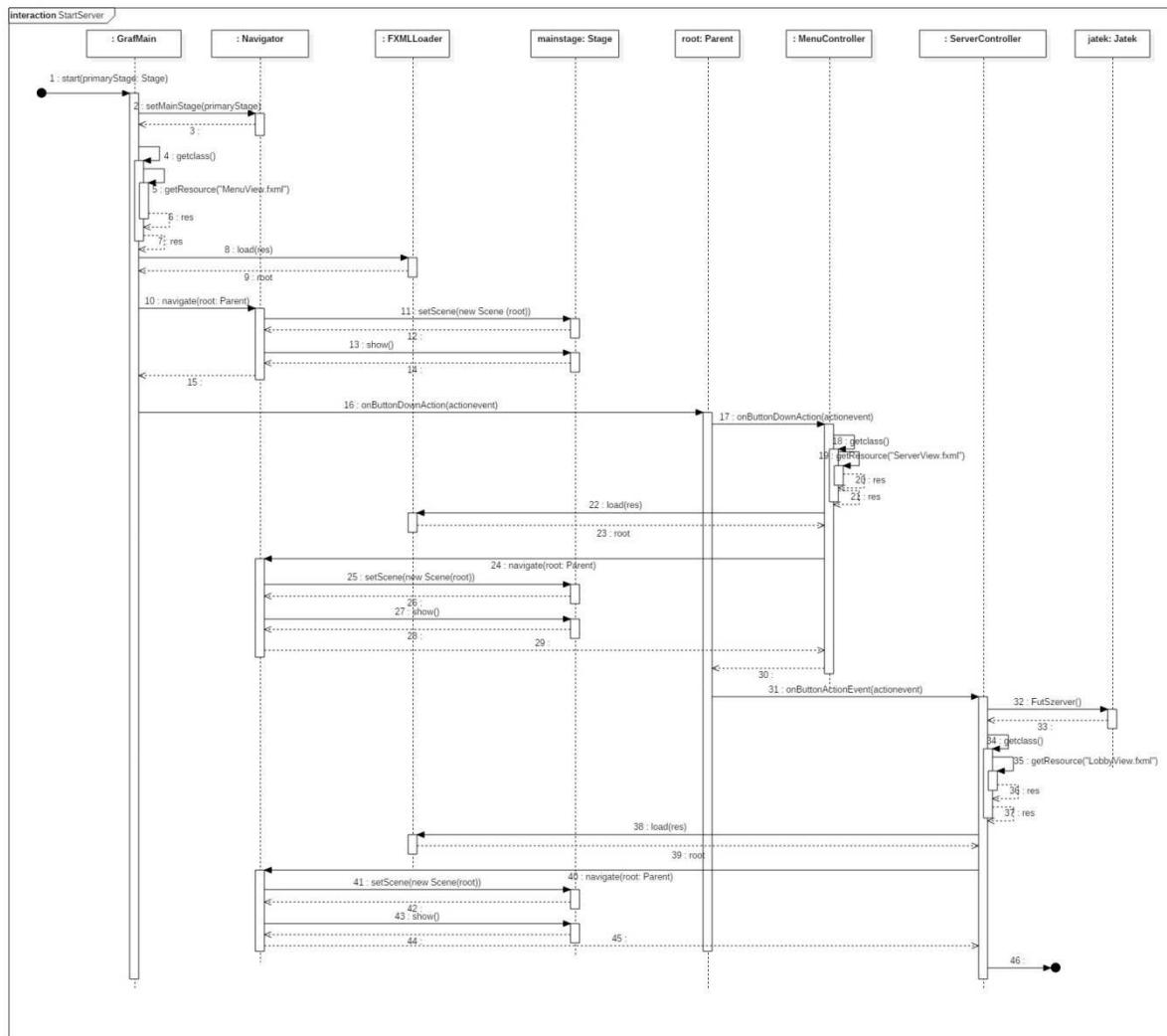
Megváltozott osztályok nincsenek.

11.4 Kapcsolat az alkalmazói rendszerrel

11.4.1 Szerver/pálya indítása

11. Grafikus felület specifikációja

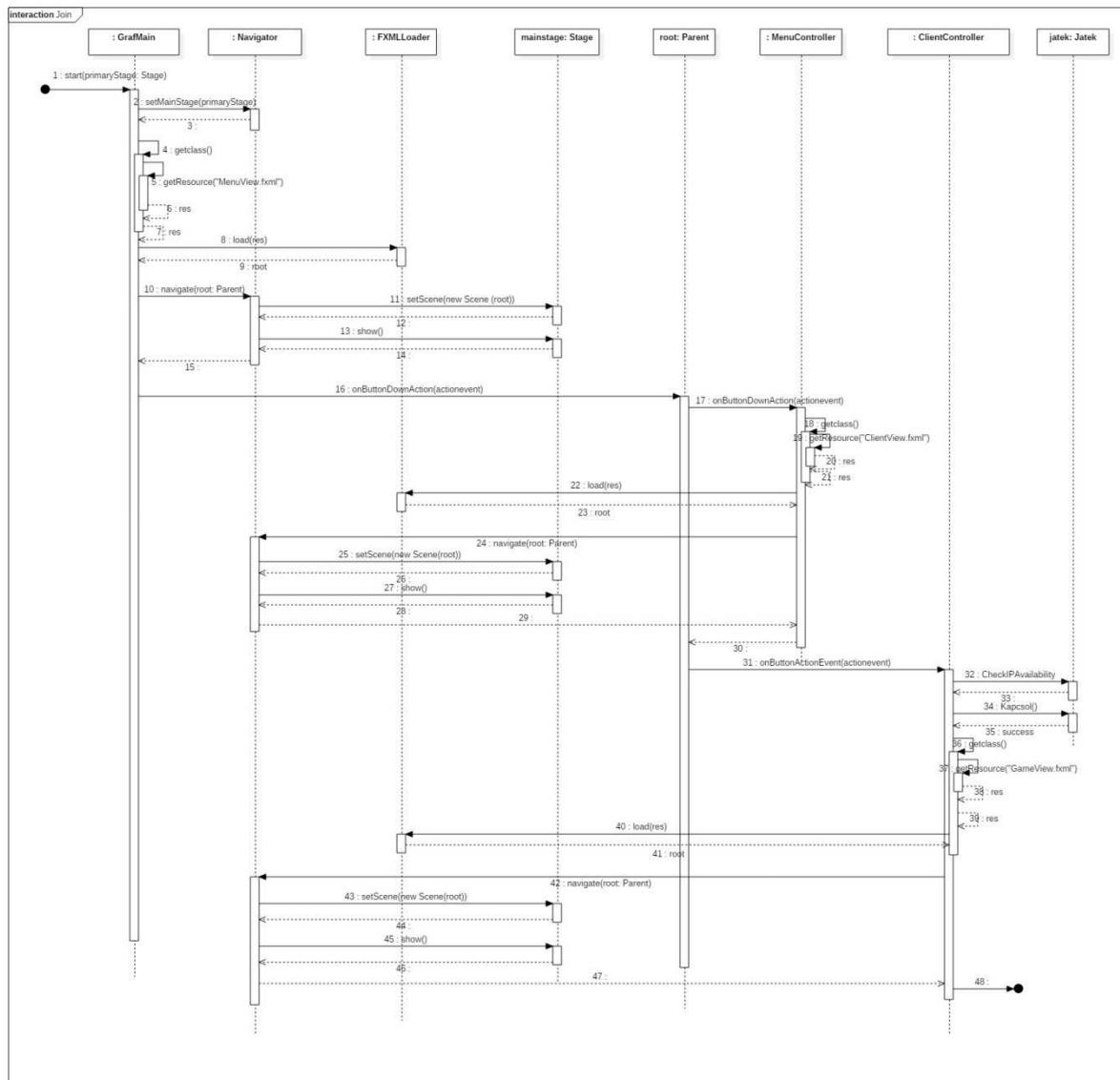
megegyezesalatt



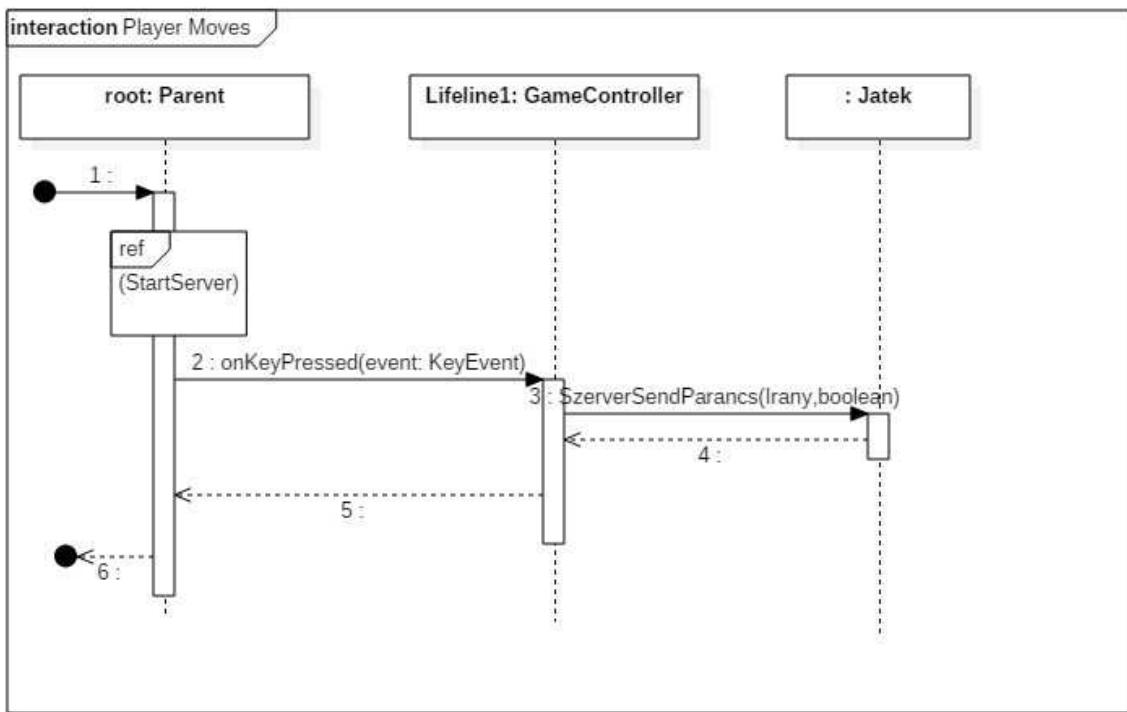
11.4.2 Csatlakozás játékhoz

11. Grafikus felület specifikációja

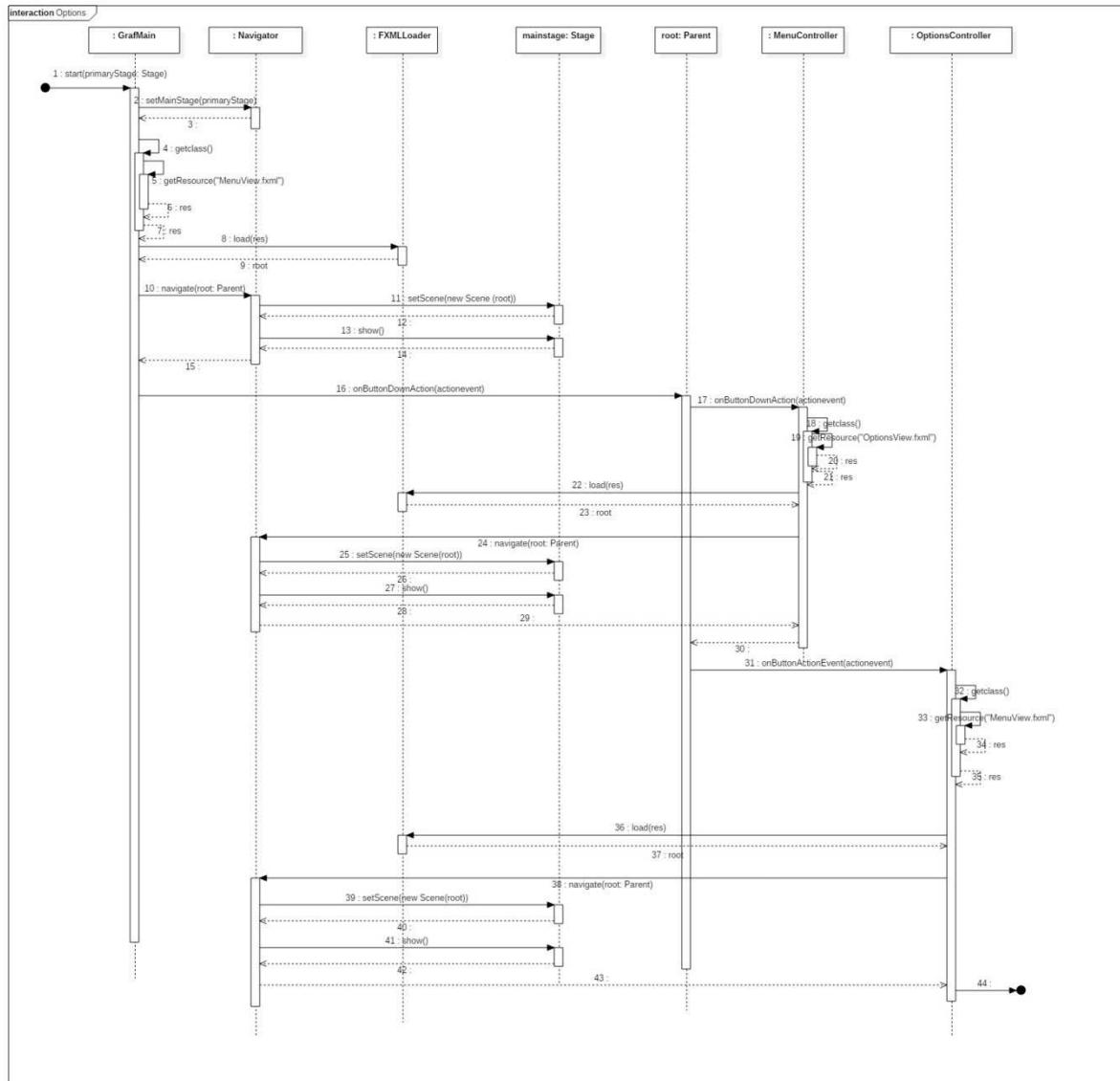
megegyezesalatt



11.4.3 Játékos lép



11.4.4 Beállítások megnyitása



11.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2018.04.26. 13:00	2,5 óra	Holub Horváth	A grafikai megoldás megbeszélése
2018.04.30. 14:00	10 óra	Mocsári	Grafikai interfések implementálása
2018.04.30. 14:00	7 óra	Zsíros	Grafikai interfések implementálása
2018.04.30. 18:00	2 óra	Bottlik	A pálya elemeinek megrajzolása
2018.04.30. 21:00	9 óra	Zsíros	Grafikai interfések implementálása
2018.05.01. 21:30	6 óra	Bottlik	Osztályleírások elkészítése és a grafikus interfész dokumentálása
2018.05.01. 20:30	7,5 óra	Horváth	Osztálydiagram készítése, dokumentumszerkesztés
2018.05.01. 22:30	6 óra	Holub	Szekvenciadiagramok készítése

13. Grafikus változat beadása

Fordítási és futtatási útmutató

Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
GrafMain.java	2 KB	2018/04/13 23:06	A főprogram, ami a játékot futtatja
Irany.java	2 KB	2018/04/29 21:22	Irány osztály
Jatek.java	20 KB	2018/05/13 23:06	Jatek osztály, a játék fő osztálya
Navigator.java	2 KB	2018/05/13 23:06	osztály a menük közötti léptetéshez
Palya.java	8 KB	2018/05/01 22:59	Palya osztály
Pont.java	2 KB	2018/04/29 21:22	Pont osztály
Pontok.java	4 KB	2018/05/13 23:06	Pontok osztály
Doboz.java	4 KB	2018/04/29 21:22	Doboz osztály
Jatekos.java	5 KB	2018/04/29 21:22	Jatekos osztály
JeloltDoboz.java	4 KB	2018/04/29 21:22	JeloltDoboz osztály
Leptetheto.java	3 KB	2018/04/29 21:22	Leptetheto osztály
Celhely.java	5 KB	2018/04/29 21:22	Celhely osztály
Csapdaajto.java	5 KB	2018/04/29 21:22	Csapdaajto osztály
Fal.java	4 KB	2018/04/29 21:22	Fal osztály
Kapcsolo.java	5 KB	2018/04/29 21:22	Kapcsolo osztály
Lyuk.java	4 KB	2018/04/29 21:22	Lyuk osztály
UresMezo.java	8 KB	2018/04/29 21:22	UresMezo osztály
BaseAdat.java	1 KB	2018/04/29 21:22	BaseAdat osztály
FIFO.java	1 KB	2018/04/29 21:22	FIFO osztály
Kapcsolat.java	2 KB	2018/04/29 21:22	Kapcsolat osztály
KliensAdat.java	1 KB	2018/04/29 21:22	KliensAdat osztály
KliensOlvaso.java	2 KB	2018/05/13 23:06	KliensOlvaso osztály
PalyaAdat.java	1 KB	2018/04/29 21:22	PalyaAdat osztály
ParancsAdat.java	1 KB	2018/04/29 21:22	ParancsAdat osztály
Szerver.java	6 KB	2018/04/29 21:22	Szerver osztály
SzerverOlvaso.java	2 KB	2018/04/29 21:22	SzerverOlvaso osztály
ClientController.java	4 KB	2018/05/13 23:06	ClientController osztály
GameController.java	16 KB	2018/05/13 23:06	GameController osztály
LobbyController.java	2 KB	2018/05/13 23:06	LobbyController osztály
MenuController.java	3 KB	2018/05/13 23:06	MenuController osztály
OptionsController.java	3 KB	2018/05/13 23:06	OptionsController osztály
ServerController.java	8 KB	2018/05/13 23:06	ServerController osztály
Data.java	1 KB	2018/05/13 23:06	Data osztály
palya.mocsí	545 B	2018/04/29 21:22	Egy generált pálya
celhely.png	13 KB	2018/04/29 19:26	célhely képe
doboz.png	6 KB	2018/04/29 19:18	doboz képe
fal.png	6 KB	2018/04/29 19:30	fal képe
jeloltdoboz.png	7 KB	2018/04/29 19:18	jelölt doboz képe

kapcsolo.png	14 KB	2018/04/29 20:28	kapcsoló képe
lyuk.png	16 KB	2018/04/29 20:27	lyuk képe
mez.png	8 KB	2018/04/29 21:24	méz képe
mezolaj.png	21 KB	2018/04/29 22:42	mézes-olajos mező képe
olaj.png	7 KB	2018/04/29 21:24	olaj képe
uresmezo.png	13 KB	2018/04/29 19:27	üres mező képe
jo0.png	11 KB	2018/05/01 22:59	játékos képe
jo1.png	12 KB	2018/05/01 22:59	játékos képe
jo2.png	12 KB	2018/05/01 22:59	játékos képe
jo3.png	11 KB	2018/05/01 22:59	játékos képe
jo4.png	11 KB	2018/05/01 22:59	játékos képe
jo5.png	11 KB	2018/05/01 22:59	játékos képe
jo6.png	11 KB	2018/05/01 22:59	játékos képe
jo7.png	11 KB	2018/05/01 22:59	játékos képe
jo8.png	11 KB	2018/05/01 22:59	játékos képe
jo9.png	11 KB	2018/05/01 22:59	játékos képe
jo10.png	11 KB	2018/05/01 22:59	játékos képe
jo11.png	11 KB	2018/05/01 22:59	játékos képe
jo12.png	11 KB	2018/05/01 22:59	játékos képe
jo13.png	11 KB	2018/05/01 22:59	játékos képe
jo14.png	11 KB	2018/05/01 22:59	játékos képe
jo15.png	11 KB	2018/05/01 22:59	játékos képe
jo16.png	11 KB	2018/05/01 22:59	játékos képe
jo17.png	11 KB	2018/05/01 22:59	játékos képe
jo18.png	11 KB	2018/05/01 22:59	játékos képe
jo19.png	11 KB	2018/05/01 22:59	játékos képe
jo20.png	11 KB	2018/05/01 22:59	játékos képe
Test1.mocsi	1 KB	2018/05/01 22:59	Tesztpálya
Test2.mocsi	1 KB	2018/05/01 22:59	Tesztpálya
Test3.mocsi	1 KB	2018/05/01 22:59	Tesztpálya
Test4.mocsi	1 KB	2018/05/01 22:59	Tesztpálya
Test5.mocsi	1 KB	2018/05/01 22:59	Tesztpálya
Test6.mocsi	1 KB	2018/05/01 22:59	Tesztpálya
ujpalya.mocsi	1 KB	2018/05/13 23:06	Tesztpálya
demopalya.mocsi	2 KB	2018/05/14 01:17	Pálya
Printer.java	1 KB	2018/05/11 14:43	Printer osztály

Fordítás

(a program gyökérmapjában kiadva)

javac controller/*.java data/*.java server/*.java sokoban/*.java tools/*.java

Hint: A hszk gépein a PATH változóban nincs bent mindig a javac és a java

*ekkor: „C:\Program Files\Java****\javac” controller/*.java, stb. formában lehet megoldani*

Hint2: A fájlok útvonalainál lehet, hogy kelleni fog az elérési út elő egy „/” jel.

Futtatás

A következő parancssori parancccsal:
 (a program gyökérmapájában kiadva)
java sokoban.GrafMain

Értékelés

Tag neve	Munka százalékban
Holub Csongor	20%
Horváth Gergő	20%
Mocsári András	20%
Zsíros Bálint	20%
Bottlik Judit	20%

Napló

Kezdet	Időtartam	Résztvevők	Leírás
2018.05.12 13:30	5,5 óra	Zsíros	Tevékenység: Kódolás, hibák javítása
2018.05.13. 14:30	10 óra	Mocsári	Tevékenység: Grafikus interfész véglegesítése, hibák javítása
2018.05.13. 20:0	5,5 óra	Zsíros	Tevékenység: Bug javítás, Ellenőrzés
2018.05.13 23:00	1 óra	Holub	Tevékenység: Hibajavítás, tesztelés
2018.05.13. 23:00	1 óra	Bottlik	Tevékenység: Dokumentumszerkesztés
2018.05.14 0:00	1 óra	Holub	Tevékenység: Dokumentumszerkesztés
2018.05.14 12:00	1,5 óra	Bottlik Holub Horváth	Tevékenység: Tesztelés a HSZK gépein

14. Összefoglalás

A projektre fordított összes munkaidő

Tag neve	Munkaidő (óra)
Bottlik	48,5
Holub	45
Horváth	42,5
Mocsári	93,5
Zsíros	110,5
Összesen	340

- A feltöltött programok forrássorainak száma***

Fázis	KódSOROK száma
Szkeleton	122247
Prototípus	2655
Grafikus változat	4284
Összesen	129186

- Projekt összegzés***

Mit tanultak a projektből konkrétan és általában?

A tárgy jó volt arra, hogy kicsit közelebb kerüljünk ahhoz, amilyen jellegű munkákat majd egy cégnél fogunk végezni, így jobban átlátjuk, hogy mi tetszik nekünk a legjobban, és hogy tényleg szeretnénk-e ilyesmivel foglalkozni majd a jövőben.

A csapattagokkal együtt dolgozva, egymás munkáit látva pedig sokkal tisztább lett, hogy melyik az a rész, ami jól megy és mi az, amiben fejlődnünk kell.

Hasznos volt, hogy részegységekre volt bontva a feladat, így a sok határidő biztosította a folyamatos előrehaladást.

Megtapasztaltuk milyen csapatban dolgozni rajta annak minden előnyével, és hátrányával.

Maga a csapatmunka segített abban, hogy egyre jobban tudunk különböző típusú emberekkel együtt dolgozni, és azt, hogy kivel milyen módon érdemes kommunikálni, együttműködni, hogy a munka a legeredményesebb legyen.

Mi volt a legnehezebb és a legkönnyebb?

A csapatban előforduló ellentétes vélemények néha megnehezítik a munkát, ezt sokszor szinte lehetetlen úgy megoldani, hogy a végére mindenki elégedett legyen.

Ezzel szemben, amikor a feladatok jól vannak szétosztva, akkor nagyon gyorsan lehet haladni.

Maga a csapatban dolgozás nagyon meg tudja könnyíteni a munkát, mert ha valaki valamiben elakad, akkor mindenig tud szólni egy csapattársnak, aki esetleg jobban ért hozzá, vagy csak máshogyan látja a dolgokat és tud segíteni. Így gördülékenyebbé vált a munka, mint ha egyedül lett volna.

Összhangban állt-e az idő és a pontszám az elvégzendő feladatokkal?

Nem volt teljesen összhangban, a pontszámok elosztása másképp is törtéhetett volna.

Ha nem, akkor hol okozott ez nehézséget?

A szkeleton több mint a felét elviszi a munkának, és ennek ellenére a végső pontszámba csak 30% súlyal számít bele. A prototípussal feleannyi munka sem volt, mint a szkeletonnal.

Milyen változtatási javaslatuk van?

A jegyszámítás arányain változtatnák, például 40-40-20 vagy akár 50-30-20 százalékra.
(Zsíros javaslata)

Milyen feladatot ajánlanának a projektre?

Esetleg egy gyűjtögetős kártyajáték, vagy valamilyen társasjáték (pl.: Ki nevet a végén).

Egyéb kritika és javaslat

Jó volt, hogy a tárgy kreditértékét felemelték 3-ra, a 2 nagyon kevés volt.

