

2. Követelmény, projekt, funkcionálitás

7 – berenyi_kft

Konzulens:

Csapattagok

Kreták Barnabás András

Mesterházi Marcell

Szögi Richárd

Havasi Balázs

Radnai Bálint

2021.02.21

2. Követelmény, projekt, funkcionalitás

2.1 Bevezetés

2.1.1 Cél

Jelen dokumentum célja, hogy bemutassa a Szoftver Projekt Laboratórium tantárgyból a 2020/21 tavaszi szemeszterben a „berenyi_kft” csapat által a félév alatt elkészítendő feladat követelményeinek leírását és a projekt általános áttekintését.

2.1.2 Szakterület

A program egy játék, nem tartozik egy adott szakterülethez sem. Célja csupán a szórakoztatás, a tárgy követelményeinek teljesítése, és csapatban végzett szoftverfejlesztési tapasztalat szerzése.

2.1.3 Definíciók, rövidítések

- **Git:** webes felületű, git rendszerű, elosztott verziókezelő
- **UML:** szabványos, általános célú modellező nyelv
- **Whitestar UML:** UML modellek készítéséhez használható eszköz

2.1.4 Hivatkozások

Szoftvertechnológia tantárgy előadás diái : www.iit.bme.hu/targyak/BMEVIIIA01

Programozás alapai 3 tantárgy diái : www.iit.bme.hu/targyak/BMEVIIIA00/prog3-f%C3%A9l%C3%A9rvi-%C3%BCtemez%C3%A9s%C3%A9s

Feladatkiírás : www.iit.bme.hu/file/11582/feladat

2.1.5 Összefoglalás

A dokumentum tartalmazza az alábbiakat:

2.2 Áttekintés: a szoftver architekturális képe, funkciói, a felhasználói jellemzők

2.3 Követelmények: a programmal szembeni funkcionális, erőforrásokkal és átadással kapcsolatos követelmények

2.4 Lényeges use-case-ek: használati esetek megadása táblázatosan, ábrázolása diagramon.

2.5 Szótár: a feladat készítése során felmerülő hétköznapi szavak/kifejezések körülírása, amelyek a játék világában levő mechanizmusokat írják le.

2.6 Projekt terv: projektütemezés és a csapattagok

2.7 Napló: az előző beadás óta eltelt időszakban elvégzett feladatok és a rájuk szánt idő dokumentálása

2.2 Áttekintés

2.2.1 Általános áttekintés

A játék a megnyitása után egy grafikus felülettel indul, ahol menüből kiválaszthatjuk a játékosok számát (maximum 6 fő), majd az “új játék” gombra kattintva kirajzolódik a játék pályája, azaz az aszteroidamező annyi játékossal, amennyit a menüben megadtunk. Ezen a felületen kommunikálhatunk a játékkal.

A játék körökre osztott. minden körben sorban a következő - még játékban levő - játékos aktív, ő veszi át a kezelőfelületet. Az aktív játékos itt választhatja ki magának a következő lépését (pl. másik aszteroidára mozgás, fúrás, bányászat). A bekövetkező változásokat a grafikus felületünk automatikusan megjeleníti.

Adatbázisunk több adatot is tárol, főként a pálya kirajzolásához szükséges képfájlokat. A programnak nincs szüksége hálózati kapcsolatra, a hozzá tartozó fájlokat a számítógép tárhelyén kell tárolnunk.

2.2.2 Funkciók

Az emberiség elhatározta, hogy kiaknázza az aszteroidabányászatban rejlö lehetőségeket. Ehhez a telepeseknek fel kell építeniük egy ürbázist az aszteroidaövben. Az építéshez szükséges erőforrásokat túlságosan drága lenne odavinni, így azokat az aszteroidákból kell kinyerni. Mindegyik játékos egy-egy telepest irányít. A telepesek ugyanarról az aszteroidáról indulnak, és egyszemélyes űrhajókkal járják az aszteroidákat, megfelelő nyersanyagok után kutatva.

Az aszteroidákat különböző vastagságú sziklaköpenyek borítják. A fontos nyersanyagok (vízjég, vas, szén, urán stb.) az aszteroidák magjában találhatók. A nyersanyagok egy része (például az urán) erősen radioaktív. Vannak olyan aszteroidák is, amelyek magja üreges, bennük nem található nyersanyag. Az aszteroidák magja minden homogén, nem keverednek benne a különböző anyagok.

A játék körökre osztott. minden körben a még életben levő játékosok sorban, egyesével lépnek. Egy telepes egy lépésben egyfajta műveletet hajthat végre. Többfajta művelet is van, például mozgás, fúrás, bányászat, robotépítés, teleportkapupár-építés. Mozgás során a telepes az űrhajójával egy szomszédos aszteroidára megy át (minden aszteroidának néhány, esetenként több száz szomszédja van). Fúrás során a telepes egy egységnivel tudja mélyíteni az aszteroida köpenyébe fűrt lyukat. Bányászat során a telepes kinyeri a fűrt lyukon keresztül az aszteroida magjában található erőforrást, aminek hatására az az aszteroida üregessé válik. Bányászni csak akkor lehetséges, ha az aszteroida köpenyét már sikerült teljesen átfürni. Egy telepésnél legfeljebb 10 egységnyi nyersanyag lehet, ennél több nem fér az űrhajóba (ha éppen 10 egység nyersanyagot tárol, nem bányászhat). Üreges aszteroidába azonban egy műveletként visszahelyezhető egy egységnyi nyersanyag.

A telepesekre veszélyek is leselkednek. Ha egy teljesen megfűrt, radioaktív maggal rendelkező aszteroida éppen napközelben van, akkor az aszteroida felrobban, ami megöli a rajta levő telepeseket, és megsemmisíti az ott található nyersanyagokat is. A radioaktív anyagok tehát csak naptávolban lévő aszteroidából nyerhetők ki. Egy aszteroida Naptól való távolsága a kettejük pozíciói alapján számítható. Veszélyt jelentenek a telepesekre a napviharok is, amelyek időnként elérik az aszteroidaötvet, és végigsöpörnek az összes aszteroidán. A napvihar csak úgy élhető túl, ha a telepes egy már kifűrt üreges aszteroida magjában bújik meg.

A telepes egy egységnyi vas, egy egységnyi szén és egy egységnyi urán felhasználásával mesterséges intelligencia által vezérelt autonóm robotot hozhat létre. Az új robotot egyből útjára is engedi azon az aszteroidán, amelyen létrehozta. Ezek a robotok csak szomszédos aszteroidák közötti mozgásra, illetve fúrásra képesek. A robotok bányászni nem tudnak, és nyersanyagot nem tudnak magukkal vinni. A robotok túlélik a radioaktív robbanást, és ilyenkor valamelyik szomszédos aszteroidán landolnak. A napvíhar azonban a robotokat is tönkreteszi, ha nem bújnak el egy üreges aszteroidában. Az üreges aszteroida magjában egyszerre akármennyi telepes és robot megbújhat.

Két egység vas, egy egység vízjég és egy egység urán felhasználásával a telepes teleportkapupár készíthet. Az egyes kapukat az űrhajós később az éppen meglátogatott aszteroida közelében tudja pályára állítani. A kapupár két tagja összeköttetésben van, az egyikbe belépve a másikban találja magát az utazó (például űrhajós, robot). A frissen legyártott kapukat az űrhajós magával tudja vinni, de egy űrhajónál egyidőben legfeljebb két kapu lehet (tehát amíg van nála kapu, nem készíthet újakat). Ha egy aszteroida felrobban, akkor az összes, közvetlen közelében levő teleportkapu is megsemmisül, a párjaikkal együtt.

A játék kétféleképpen érhet véget. Ha minden telepes meghalt, akkor a játékosok veszítettek. Ha azonban sikerült minden fajta nyersanyagból legalább három egységet kibányászni és egy közös aszteroidára összegyűjteni, akkor a telepesek felépíthetik az ūrbázist és megnyerik a játékot.

2.2.3 Felhasználók

A szoftver játszhatóságához a felhasználóknak nincs szükségük bonyolultabb informatikai ismeretekre, az egér, illetve a billentyűzet használata szükséges. A lehetséges felhasználók száma legalább egy és legfeljebb hat.

2.2.4 Korlátozások

A szoftver futásának folyamatnak kell lennie, nem zavarhatják meg a felhasználói élményt gyakori akadások, esetleges lefagyások. A futtatáshoz Java futtatókörnyezet szükséges, az 1.8-as verziót garantált a helyes működés. A szoftvernek a megfogalmazott követelményeknek megfelelően kell működnie.

2.2.5 Feltételezések, kapcsolatok

Feladatkiírás: leírja, mit kell megvalósítanunk

Szoftvertechnológia tantárgy előadás diájai: útmutatás; a use-case-ek és követelmények megfogalmazásának elméleti megfontolásait írja le

Programozás alapai 3 tantárgy diájai: útmutatást ad a Java nyelv használatához

2.3 Követelmények

2.3.1 Funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Use-case	Komment
R01	A játékosok egy-egy telepest irányítanak az aszteroidaövben.	Bemutatás	alapvető	feladat	Telepes irányítása, Aszteroidaöv megtekintése	
R02	A telepeseknek fel kell építeniük egy ürbázist az aszteroidaövben.	Bemutatás	alapvető	feladat	Telepes irányítása, Aszteroidaöv megtekintése	
R03	Az építéshez szükséges erőforrásokat az aszteroidákból kell kinyerni.	Bemutatás	alapvető	feladat	Nyersanyag bányászata	
R04	A telepesek egyszemélyes űrhajókkal járják az aszteroidákat megfelelő nyersanyagok után kutatva.	Bemutatás	alapvető	feladat	Űrhajóval mozgás, Aszteroidaöv megtekintése	Úgy tekintjük, hogy a telepesek nem válnak el a saját űrhajójuktól.
R05	Az aszteroidákat különböző vastagságú köpeny borítja.	Bemutatás	fontos	feladat	Aszteroidák kezelése	
R06	Az ürbázishoz szükséges nyersanyagok az aszteroidák magjában találhatók.	Bemutatás	alapvető	feladat	Aszteroidák kezelése	
R07	Többféle szükséges nyersanyag van, például: vízjég, vas, szén, urán, stb.	Bemutatás	alapvető	feladat	Aszteroidák kezelése	
R08	A nyersanyagok egy része (például az urán) radioaktív.	Bemutatás	fontos	feladat	Aszteroidák kezelése	
R09	Lehetnek üreges aszteroidák is, amelyek magja üreges, bennük nem található nyersanyag.	Bemutatás	alapvető	feladat	Aszteroidák kezelése	

R10	Egy-egy aszteroida magjában legfeljebb egyfélre nyersanyag található.	Bemutatás	fontos	feladat	Aszteroidák kezelése	
R11	A játék körökre osztott.	Bemutatás	alapvető	csapat	Telepes irányítása, Játék kezelése	
R12	Egy telepes egy lépéssben egyfajta műveletet hajthat végre.	Bemutatás	alapvető	feladat	Telepes irányítása	
R13	Többfajta művelet is van, például mozgás, fúrás, bányászat, robotépítés, teleportkapupár- építés stb.	Bemutatás	alapvető	feladat	Telepes irányítása	A telepes passzolhat is, ilyenkor nem végez műveletet az adott körben.
R14	Minden aszteroidának néhány, esetenként több száz szomszédja van.	Bemutatás, kiértékelés	alapvető	feladat	Aszteroidák kezelése	A játék kezdetén bármely aszteroida mozgásokkal elérhető.
R15	Mozgás során a telepes az ūrhajójával átmegy egy szomszédos aszteroidára.	Bemutatás	alapvető	feladat	Űrhajóval mozgás	
R16	Fúrás során a telepes egy egységnyivel tudja mélyíteni az aszteroida köpenyébe fűrt lyukat.	Bemutatás, kiértékelés	alapvető	feladat	Aszteroida fúrása	
R17	Ha az aszteroida köpenyét már sikerült teljesen átfúrni, akkor bányászat során a telepes kinyerheti a fűrt lyukon keresztül az aszteroida magjában található erőforrást.	Bemutatás	alapvető	feladat	Nyersanyag bányászata	A telepes nem bányászhat, ha 10 egységnyi nyersanyagot tárol, (lásd R19).
R18	Bármelyik aszteroida üregessé válik, ha kibányásszák belőle a nyersanyagot.	Bemutatás, kiértékelés	alapvető	csapat	Nyersanyag bányászata, Aszteroidák kezelése	

R19	Egy telepesnél legfeljebb 10 egységnyi nyersanyag lehet, ennél több nem fér az úrhajóba.	Kiértékelés	alapvető	feladat	Telepes irányítása	
R20	Üreges aszteroidába egy műveletként visszahelyezhető egy egységnyi nyersanyag.	Bemutatás	alapvető	feladat	Nyersanyag visszatöltése	
R21	Az aszteroidák lehetnek napközelben vagy naptávolban, ami az aktuális pozíciójuk alapján számítható.	Bemutatás, kiértékelés	fontos	feladat, csapat	Aszteroidák kezelése	
R22	Ha egy teljesen megfűrt, radioaktív maggal rendelkező aszteroida éppen napközelben van, akkor az aszteroida felrobban.	Bemutatás	alapvető	feladat	Aszteroidák kezelése	
R23	Ha egy aszteroida felrobban, akkor a rajta levő telepesek meghalnak.	Bemutatás	alapvető	feladat	Telepes irányítása, Aszteroidák kezelése	
R24	A telepes halála esetén az összes nála lévő nyersanyag és teleportkapu elvész.	Bemutatás	fontos	csapat	Telepes irányítása	
R25	Ha egy aszteroida felrobban, akkor a magjában található nyersanyag is megsemmisül.	Bemutatás	alapvető	csapat	Aszteroidák kezelése	
R26	Veszélyt jelentenek a telepesekre a napviharok is, amelyek véletlen időközönként elérik az aszteroidaövbbe tartozó összes aszteroidát.	Bemutatás	alapvető	feladat	Napvihar kezelése	A nap-távolban levő aszteroidákat is eléri a napvihar.
R27	A napvihar csak úgy élhető túl, ha a telepes egy üreges aszteroida magjában bújik meg.	Bemutatás	fontos	feladat	Napvihar kezelése	

R28	Ha telepesek, illetve robotok kifúrt üreges aszteroidán tartózkodnak, akkor automatikusan megbújnak a magjában, egyszerre akármennyien.	Bemutatás	fontos	feladat, csapat	Napvihar kezelése	
R29	A telepesek egy egységnnyi vas, egy egységnnyi szén és egy egységnnyi urán felhasználásával mesterséges intelligencia által vezérelt autonóm robotot hozhatnak létre.	Bemutatás, kiértékelés	alapvető	feladat	AI robot építése	Minden űrhajós maga építheti, csak a saját űrhajóján tárolt nyersanyagokból.
R30	Ezek a robotok csak szomszédos aszteroidák közötti mozgásra és fűrészre képesek.	Bemutatás	alapvető	feladat	AI robot cselekvése	
R31	A robotok bányászni nem tudnak, és nyersanyagot nem tudnak magukkal szállítani.	Bemutatás	alapvető	feladat	AI robot cselekvése	
R32	A robotok túlélik a radioaktív robbanást, és ilyenkor valamelyik szomszédos aszteroidán landolnak.	Bemutatás	fontos	feladat	AI robot cselekvése, Aszteroidák kezelése	
R33	A napvihar a robotokat is tönkreteszí, ha nem bújnak el egy üreges aszteroidában.	Bemutatás	alapvető	feladat	AI robot cselekvése, Aszteroidák kezelése	
R34	Két egység vas, egy egység vízjég és egy egység urán felhasználásával teleportkapu-párt lehet készíteni.	Bemutatás, kiértékelés	alapvető	feladat	Teleport-kapu-pár kezelése	Minden űrhajós maga építheti, csak a saját űrhajóján tárolt nyersanyagokból.

R35	Az egyes kapukat az űrhajós később az éppen meglátogatott aszteroida körül tudja pályára állítani.	Bemutatás	fontos	feladat	Teleport-kapu-pár kezelése	
R36	A már pályára állított teleportkapukat az űrhajós később nem veheti fel és nem helyezheti át.	Bemutatás	fontos	csapat	Teleport-kapu-pár kezelése	
R37	A kapupár két tagja összeköttetésben van, az egyikbe belépve a másikban találja magát a karakter (űrhajós, robot stb.).	Bemutatás	alapvető	feladat	Teleport-kapu-pár kezelése	
R38	A frissen legyártott kapukat az űrhajós magával tudja vinni, de egy űrhajósnál egyidőben legfeljebb két kapu lehet.	Bemutatás	fontos	feladat	Teleport-kapu-pár kezelése	
R39	Ha egy aszteroida felrobban, akkor a közvetlenül körülötte keringő teleportkapuk a párjaikkal együtt megsemmisülnek.	Bemutatás	fontos	csapat	Aszteroidák kezelése	
R40	Ha minden telepes meghalt, akkor a játékosok veszítettek, a játék véget ér.	Bemutatás	alapvető	feladat	Játék kezelése	
R41	Ha sikerült minden-fajta nyersanyagból legalább három egységnnyit kibányászni és egy közös aszteroidára összegyűjteni, akkor a telepesek felépíthatik a bázist, és megnyerik a játékot.	Bemutatás	alapvető	feladat	Játék kezelése	
R42	A játékos passzolhat is egy körben.	Bemutatás	fontos	csapat	Telepes irányítása, Játék kezelése	
R43	A robot az adott aszteroida köpenyét addig fúrja, amíg a maghoz le nem ért.	Bemutatás, kiértékelés	opcionális	csapat	AI robot cselekvése	

R44	Ha az aszteroida kérget teljesen átfúrta a robot, akkor egy véletlenszerű szomszédos aszteroidára mozog.	Bemutatás, kiértékelés	opcionális	csapat	AI robot cselekvése	
R45	A játék kezdetekor a telepesek ugyanazon az aszteroidán helyezkednek el.	Bemutatás	opcionális	csapat	Játék kezelése	

2.3.2 Erőforrásokkal kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
R101	A szoftver Java nyelven íródik.	Bemutatás	alapvető	feladat	
R102	A szoftvernek Java JRE 1.8 futtatókörnyezetben futathatónak kell lennie.	Bemutatás, kiértékelés	alapvető	feladat	
R103	A szoftvernek Java JDK 8 fejlesztő környezetben fordíthatónak kell lennie.	Bemutatás, kiértékelés	alapvető	feladat	
R104	A szoftver fejlesztése Eclipse fejlesztő környezetben zajlik.	Bemutatás	opcionális	csapat	
R105	A szoftver tartalmaz grafikus megjelenítést, amelyet az egérrel, illetve a billentyűzettel lehet irányítani.	Bemutatás	alapvető	feladat, csapat	
R106	A grafikus felület megtervezéséhez Photoshopot és Illustratort használunk.	Bemutatás	opcionális	csapat	
R107	A grafikus felület megvalósítása Java Swing eszköztár segítségével történik.	Bemutatás	alapvető	feladat	
R108	A fájlok megosztásához felhasznált környezetek: Github, Teams, Google Drive.	Bemutatás	opcionális	csapat	
R109	Feladatok megbeszéléséhez a Microsoft Teams felületet használunk.	Bemutatás	opcionális	csapat	
R110	A dokumentáció elkészítéséhez MS Word programot részesítettük előnyben.	Bemutatás	opcionális	csapat	

2.3.3 Átadással kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
R201	A program legyen a kari felhő virtuális gépein parancsorból indítható.	Bemutatás, kiértékelés	alapvető	feladat	
R202	A program legyen JAR fájl segítségével indítható.	Bemutatás, kiértékelés	fontos	csapat	
R203	GitHub-ról legyen letölthető az elkészült projekt.	Bemutatás	opcionális	csapat	

2.3.4 Egyéb nem funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
R301	Legyen JUnit tesztelhető a megoldás.	Bemutatás	opcionális	csapat	

2.4 Lényeges use-case-ek

2.4.1 Use-case leírások

Use-case neve	Telepes irányítása
Rövid leírás	A játékos irányítja a telepet
Aktorok	Játékos
Forgatókönyv	A játékos a telepessel mozoghat, fűrhat, bányászhat, robotot építhet, teleportkapu-párt építhet

Use-case neve	Passzolás
Rövid leírás	A játékos dönthet úgy, hogy nem végez érdemi lépést, csak passzol.
Aktorok	Játékos
Forgatókönyv	A játékos a telepessel ilyenkor nem végez érdemi lépést.

Use-case neve	Úrhajóval mozgás
Rövid leírás	A játékos képes a telepes űrhajójával egy másik szomszédos aszteroidára lépni.
Aktorok	Játékos
Forgatókönyv	A játékos az aktuális pozíciójáról egy szomszédos aszteroidára mozog.

Use-case neve	Aszteroida fúrása
Rövid leírás	A játékos egy egységet lefűr az aszteroidán.
Aktorok	Játékos
Forgatókönyv	A játékos egy egységnnyit fűr az aszteroidán, amin épp tartózkodik.

Use-case neve	Nyersanyag visszatöltése
Rövid leírás	A játékos nyersanyagot helyez egy üres aszteroida belsejébe.
Aktorok	Játékos
Forgatókönyv	A játékos az űrhajón tárolt egy egységnnyi nyersanyagot az aszteroida üres magjába tesz.

Use-case neve	Nyersanyag bányászata
Rövid leírás	A játékos kinyeri az erőforrást.
Aktorok	Játékos
Forgatókönyv	A játékos kinyeri a fűrt lyukon keresztül az aszteroida magjában található erőforrást, és az űrhajóján tárolja.

Use-case neve	AI robot építése
Rövid leírás	A játékos létrehoz egy AI robotot.
Aktorok	Játékos
Forgatókönyv	A játékos létrehoz egy AI robotot, amit a mesterséges intelligencia irányít, és ami segít neki a fúrásban.

Use-case neve	Teleportkapu-pár kezelése
Rövid leírás	A telepes képes teleportkapu-párt építeni.
Aktorok	Játékos
Forgatókönyv	A játékos teleportkapu-párt épít és a kapukat aszteroidák körüli pályára állítja.

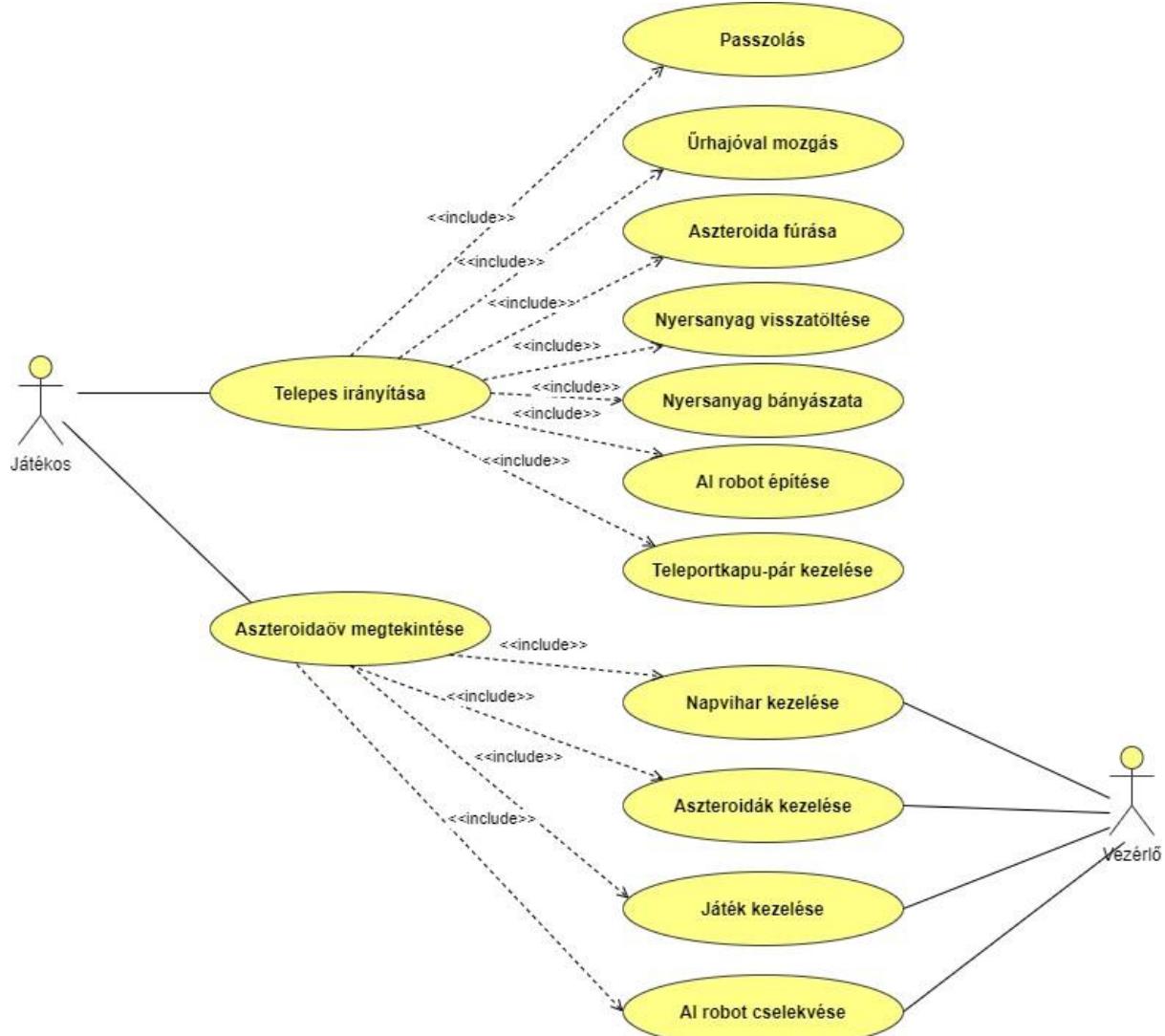
Use-case neve	Napvihar kezelése
Rövid leírás	Az aszteroidaövbén előfordulhatnak napviharok.
Aktorok	Vezérlő
Forgatókönyv	Az egész aszteroidaövbén véletlen időközönként végigsöpörnek napviharok.

Use-case neve	Aszteroidák kezelése
Rövid leírás	A játékban az aszteroidák állapotai változhatnak.
Aktorok	Vezérlő
Forgatókönyv	Az aszteroida köpenye egy egységgel csökkenhet, belseje üressé válhat és fel is robbanhat.

Use-case neve	Játék kezelése
Rövid leírás	A játék irányítása, lefolyásának kezelése, például körök kezelése.
Aktorok	Vezérlő
Forgatókönyv	A Vezérlő kezeli a játékhoz szükséges folyamatokat.

Use-case neve	AI robot cselekvése
Rövid leírás	A játékban létrejöhetnek AI robotok, melyek önállóan lépnek, de a Vezérlő tartja számon és kezeli őket.
Aktorok	Vezérlő
Forgatókönyv	Az AI robotok mesterséges intelligencia segítségével mozgás vagy fúrás műveletet végeznek.

2.4.2 Use-case diagram



2.5 Szótár

AI robot - telepes által létrehozható; egy egységnyi vas, egy egységnyi szén és egy egységnyi urán felhasználásával; csak megadott műveletek végrehajtására képes; a robbanás nem semmisíti meg

AI robotépítés - a telepesek egy egységnyi vas, egy egységnyi szén és egy egységnyi urán felhasználásával robotot hozhatnak létre; egységnyi művelet

aszteroida - olyan tárgy, amelyen tetszőleges számú telepes és robot tartózkodhat; köpeny borítja; maggal rendelkezik

bányászás - az aszteroida magjában található nyersanyag kinyerése lyukon keresztül; egységnyi művelet

felrobban - az aszteroida, a rajta tartózkodó telepesek, a hozzá tartozó teleportkapu és annak pálya megsemmisülnek; a robotok szomszédos aszteroidán landolnak a hatására

fúrás - egy aszteroidát borító köpenybeli lyuk egy egységgel való mélyítése; egységnyi művelet

halál - a telepes és az űrhajója megsemmisül a rajta levő nyersanyagokkal és teleportkapukkal együtt

karakter - a telepesek és a robotok összefoglaló neve

játék - az elkészült, futtatható program; résztvevői a játékosok; körökre osztott

játékos - az aktív játékos, aki éppen soron van; a saját telepesét irányítja

köpeny - az aszteroidát borítja; több, egységnyi vastag sziklarétegből állhat

kör - az összes, még játékban levő játékos egyesével lép egyet a telepesével

lyuk - az aszteroidát borító köpeny vékonyodása fúrás hatására

mag - az aszteroidák belső része, mely egy egységnyi homogén nyersanyagot tartalmazhat, de lehet üres is

megfűrt aszteroida - olyan aszteroida, amelyen teljesen átfűrt köpeny található, így elérhető a magja

mozgás - egy aszteroidáról annak egy szomszédjára való átkelés; egységnyi művelet

művelet - a karakterek által végrehajtható tevékenységek halmazának összefoglaló neve

napközel - egy aszteroida napközelben van, ha a Naptól való távolsága kisebb egy megadott távolságnál

napvihar - véletlenszerű időközönként eléri az aszteroidaötvet, ilyenkor minden aszteroida köpenyén tartózkodó karakter meghal

nyersanyag=erőforrás - az aszteroidák magja tartalmazza; különböző típusai vannak pl.: vízjég, szén, vas, stb.

radioaktív nyersanyag - pl.: urán; csak naptávolban bányászható; uránt tartalmazó megfűrt aszteroida napközelben felrobban

szikla - az aszteroidák felületét borítja tetszőleges egység vastagságban

telepes=űrhajós - a játékos által irányított karakter; műveletek elvégzésére képes; egyszerre egy műveletet csinálhat

teleportkapu-építés - a telepesek két egység vas, egy egység vízjég és egy egység urán felhasználásával teleportkapu-párt tudnak készíteni; egységnyi művelet

teleportkapu-pár - tárgypár; a kapupár két tagja összeköttetésben van: miután a kapukat pályára állították, az egyikbe belépve a másikban találja magát a karakter; a két kapunál levő aszteroidák szomszédosnak számítanak; ha a pár egyik tagja megsemmisül, a teleportkapu-pár nem használható

űrbázis - akkor jön létre, ha egy aszteroidán csoportosul minden nyersanyagból legalább 3 egységnyi; ez a játék megnyerését jelenti

űrhajó - a telepes ezzel közlekedik; ebben tárolja a kibányászott nyersanyagokat és a frissen elkészített teleportkapukat

üreges - az aszteroidamag, ha nem tartalmaz nyersanyagot

2.6 Projekt terv

Csapattagok:

Mesterházi Marcell - csapatvezető

Havasi Balázs

Kreták Barnabás András

Radnai Bálint

Szögi Richárd

Feladatok elvégzése:

Az egyes feladatok elosztásáról értekezleteken döntünk. A csapattagok feladatra kijelölése is itt történik, közös megegyezéssel, ügyelve a minél egyenletesebb elosztásra.

Hatóidők, lépések:

Hatóidő	Feladat
febr. 22.	Követelmény, projekt, funkcionalitás - beadás
márc. 1.	Analízis modell kidolgozása 1. - beadás
márc. 8.	Analízis modell kidolgozása 2. - beadás
márc. 16. (kedd)	Szkeleton tervezése - beadás
márc. 22.	Szkeleton - beadás és a forráskód herculesre való feltöltése
márc. 29.	Prototípus koncepciója - beadás
ápr. 5.	<i>Tavaszi szünet.</i>
ápr. 12.	Részletes tervek - beadás
ápr. 19.	Prototípus - beadás és a forráskód, a tesztbemenetek és az elvárt kimenetek herculesre való feltöltése
ápr. 26.	Grafikus felület specifikációja - beadás
máj. 3.	<i>Grafikus változat készítése 1. Ezen a héten nem kell anyagot beadni.</i>
máj. 10.	Grafikus változat és Összefoglalás - beadás és a forráskód herculesre való feltöltése

Dokumentálásra használt eszközök: Microsoft Word, Google Docs

Kommunikáció: Microsoft Teams, Facebook Messenger

Modellező eszköz: WhiteStarUML, draw.io

Fejlesztőeszköz: Eclipse

Dokumentumok megosztása: Google Drive, Microsoft Teams

Forráskód megosztása, verziókezelése: Git, GitHub

2.7 Napló

Kezdet	Időtartam	Résznevők	Leírás
2021.02.17. 11:00	1 óra	Havasi Kreták Mesterházi Radnai Szögi	Értekezlet. Döntés: Feladatelosztás, ütemezés, tartalommegosztás eszközeinek meghatározása Következő értekezlet tervezett ideje: 2021.02.19. 16:00
2021.02.17. 14:00	2 óra	Szögi	Tevékenység: Bevezetés, kapcsolatok, szótár és projekt terv elkészítése
2021.02.17. 16:00	3 óra	Havasi, Kreták, Radnai	Tevékenység: Követelmények összegyűjtése: 2.3
2021.02.18. 14:30	1,5 óra	Mesterházi	Tevékenység: Use-case-ek összeírása, diagram létrehozása
2021.02.19. 16:00	0,5 óra	Havasi Kreták Mesterházi Radnai Szögi	Értekezlet. Döntés: Eddigi munka megbeszélése, felülbírálata Következő értekezlet ideje: 2021.02.20. 14:00
2021.02.19. 16:30	2,5 óra	Havasi Kreták Mesterházi Radnai Szögi	Tevékenység. Követelmények kiegészítése Use-case-ek tisztázása Use-case-ek követelményekhez rendelése
2021.02.19. 19:30	1 óra	Mesterházi	Tevékenység: Use-case javítások, 2.2.1. megfogalmazása
2021.02.19. 19:00	0,5 óra	Kreták	Tevékenység: 2.2.3, 2.2.4 megfogalmazása
2021.02.20. 14:30	0,5 óra	Havasi Kreták Mesterházi Radnai Szögi	Értekezlet. Döntés: Eddigi munka megbeszélése, felülvizsgálat megbeszélése
2021.02.20. 15:00	0,5 óra	Radnai	Tevékenység: Funkciók leírása, funkcionális követelmények tisztázása: 2.2.2, 2.3.1

3. Analízis modell kidolgozása 1. - beadás

7 – berenyi_kft

Konzulens:

Csapattagok

Kreták Barnabás András
Mesterházi Marcell
Szögi Richárd
Havasi Balázs
Radnai Bálint

2021.02.28

3. Analízis modell kidolgozása

3.1 Objektum katalógus

3.1.1 AlRobot

Telepes által egy egységnnyi vas, egy egységnnyi szén és egy egységnnyi urán felhasználásával létrehozható karakter, amely minden időegységen fúrni vagy szomszédos aszteroidára mozogni tud. Nyilvántartja, hogy melyik aszteroidán tartózkodik épp.

Az aszteroida robbanása nem semmisíti meg, ekkor szomszédos aszteroidára mozog. A napvihar viszont tönkreteszzi.

3.1.2 Asteroid

Az aszteroidákon telepesek és robotok tartózkodhatnak, valamint teleportkapuk keringhetnek körülöttük. Az aszteroida felelőssége a rajta található telepesek és robotok nyilvántartása, ezek mozgatásának biztosítása, illetve a szomszédos aszteroidák listázása is. Tárolja a köpenye aktuális vastagságát, a magjában található nyersanyagot és a hozzá tartozó teleportkapukat.

Robbanás, illetve napvihar esetén a követelményekben leírtak alapján kezeli az ott található nyersanyagot, karaktereket és teleportkapukat. Az aszteroidák szomszédságok által összefüggő csoportja alkotja a telepesek által bejárható pályát.

3.1.3 Coal

A szén nyersanyagot reprezentálja, lehet aszteroida magjában vagy telepesnél tárolva. Robotépítéshez szükséges erőforrás.

3.1.4 Game

A játék eleji inicializálást végzi. Nyilvántartja a játékban levő Napot, az aszteroidákat és a telepeseket.

3.1.5 Ice

A vízjég nyersanyagot reprezentálja, lehet aszteroida magjában vagy telepesnél tárolva. Teleportkapu-építéshez szükséges erőforrás. Ha egy napközelben levő megfűrt aszteroida magjában található, akkor elszublimál.

3.1.6 Iron

A vas nyersanyagot reprezentálja, lehet aszteroida magjában vagy telepesnél tárolva. Teleportkapu- és robotépítéshez szükséges erőforrás.

3.1.7 ResourceCollection

Nyersanyag-egységek halmazát reprezentálja. Felelőssége különféle nyersanyagok tárolása. Képes ellenőrizni a játékban építhető dolgokhoz szükséges nyersanyagok meglétét, illetve eltávolítani a felhasznált nyersanyagokat.

3.1.8 Sun

A játékban levő Napot reprezentálja. A napviharok kiindulópontja, valamint a napközelség számításához szükséges. Nyilvántartja a közvetlen közelében található aszteroidákat.

3.1.9 Settler

A játékos által irányított karakter, amely képes fúrásra, aszteroidák közti mozgásra, nyersanyag kibányászására, valamint visszahelyezésére, továbbá AIRobot és teleportkapu-pár építésre, valamint pályára állítására. Egyszerre egy műveletet csinálhat. Meghal robbanás hatására, illetve akkor is, ha nem tud elbújni a napvihar elől. Nyilvántartja a nála található nyersanyagokat és teleportkapukat, valamint azt, hogy melyik aszteroidán tartózkodik épp.

3.1.10 Timer

Egy órajel jellegű időzítőt reprezentál, amely periodikusan léptet minden léptethető objektumot a játékban.

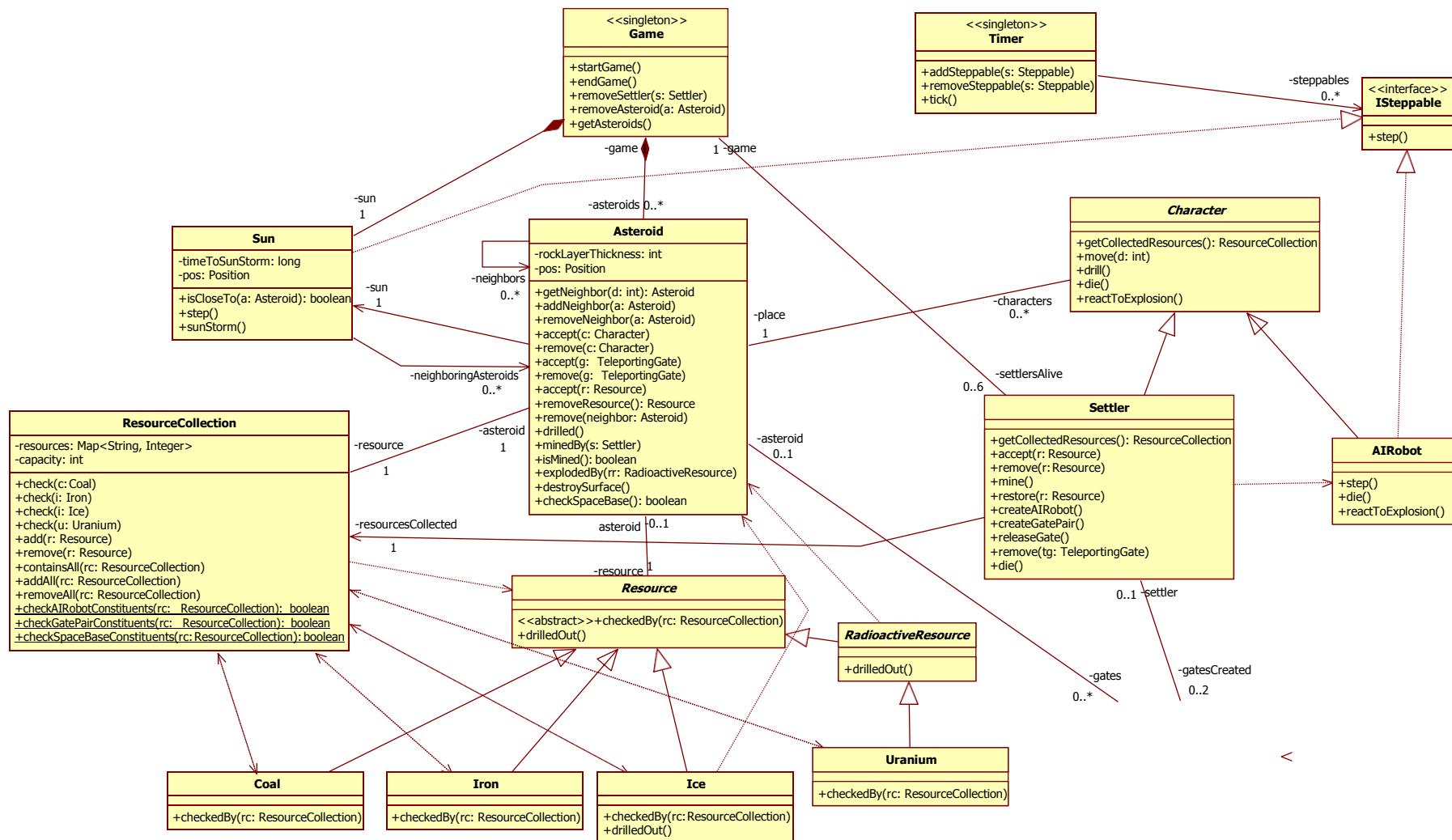
3.1.11 TeleportingGate

A teleportkapuk olyan tárgyak, amelyek párosával teleportkapu-párokat alkotnak, és aszteroidák körül pályára állíthatók. A frissen épített kapuk az építő telepes űrhajóján tárolódnak, aszteroida körüli pályára állítás után pedig az adott aszteroidákhoz tartoznak. minden teleportkapu eltárolja a páját, valamint pályára állítás előtt a telepest, amelyhez tartozik, pályára állítás után az aszteroidát, amely körül kering. Ha a teleportkapu-pár minden tagja egy-egy aszteroida körül kering, akkor ezen aszteroidák között extra, ideiglenes szomszédságot biztosítanak. Ha az egyik kapu megsemmisül, a párja vele együtt automatikusan eltűnik, bárhol is van éppen.

3.1.12 Uranium

Az urán nyersanyagot reprezentálja, lehet aszteroida magjában vagy telepesnél tárolva. Teleporkapu- és robotépítéshez szükséges erőforrás. Ha egy napközelben levő megfűrt aszteroida magjában található, akkor felrobbantja aszteroidáját.

3.2 Statikus struktúra diagramok



A nevesített asszociációvégekhez implicit getterek tartoznak, ezek a diagram átláthatósága érdekében nem minden esetben kerülnek ábrázolásra.

3.3 Osztályok leírása

Az attribútumok alapértelmezett láthatósága private, a metódusoké public. Az örökölt metódusokat csak ott tüntetjük fel újból, ahol felüldefiniálásra kerülnek.

3.3.1 AIRobot

- **Felelősség**

A karakterek egyik fajtája a robot, amely minden időegységben fújni vagy szomszédos aszteroidára mozogni tud.

- **Ősosztályok**

Character

- **Interfészek**

ISteppable

- **Attribútumok**

- **Metódusok**

- **void step()**: A robot megkérdezi az aszteroidától, amelyen áll, hogy mekkora a köpenyvastagsága. Ha az aszteroida nincs teljesen átfürva, akkor a robot drill() műveletet végez, különben valamelyik szomszédos aszteroidára lép (move(int d)).
- **void die()**: A robot megsemmisül: meghívja a Timer removeSteppable(Steppable s) metódusát.
- **void reactToExplosion()**: A robot egy véletlenszerű szomszédos aszteroidán landol: sorsol egy d számot, majd meghívja a move (int d) metódust.

3.3.2 Asteroid

- **Felelősség**

Az aszteroidákon karakterek tartózkodhatnak, valamint teleportkapuk keringhetnek körülöttük. Az aszteroidák felelőssége a rajtuk található karakterek mozgatásának biztosítása, és a szomszédos aszteroidák nyilvántartása is. Tárolja a köpenye aktuális vastagságát és a magjában található nyersanyagot. Robbanás, illetve napvihar esetén a követelményekben leírtak alapján kezeli az ott található nyersanyagot, karaktereket és teleportkapukat.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **int rockLayerThickness:** az aszteroida köpenyvastagsága, vagyis a magot borító sziklarétegek száma
- **Game game:** a játékot reprezentáló osztály
- **Sun sun:** az aszteroidaövbén levő Nap
- **Asteroid[0..*] neighbors:** az aszteroidával szomszédos aszteroidák listája; a teleportkapuk által szomszédossá vált aszteroidákat is magába foglalja
- **Resource resource:** az aszteroida magjában található egységnyi nyersanyag; ha a mag üreges, akkor értéke null
- **Character[0..*] characters:** az aszteroidán tartózkodó karakterek (telepesek, robotok, stb.) kollekciója
- **TeleportingGate[0..*] gates:** a közvetlenül az aszteroida körül keringő teleportkapuk halmaza

- **Metódusok**

- **void accept(Character c):** A c karakter megérkezik az aszteroidára, az aszteroida hozzáadja a characters kollekciójához. Ezután ellenőrzi az űrbázis megépíthetőségét: meghívja az checkSpaceBase() függvényt.
- **void remove(Character c):** A c karakter elhagyja az aszteroidát, az aszteroida eltávolítja a characters kollekciójából.
- **void accept(TeleportingGate tg):** A tg teleportkapu pályára áll az aszteroida körül, az aszteroida hozzáadja a gates kollekciójához.
- **void remove(TeleportingGate tg):** A tg teleportkaput eltávolítja az aszteroida körüli pályáról, az aszteroida törli a gates kollekciójából.
- **void accept(Resource r):** Egy az aszteroidán tartózkodó telepes behelyezi az r nyersanyagot az aszteroida üreges magjába, az aszteroida azt beállítja resource attribútumának. Ha a paraméterül kapott r=null, akkor a függvénynek nincs hatása, a resource attribútumot nem írja felül.
- **Resource removeResource():** Egy az aszteroidán tartózkodó telepes eltávolítja a magban található nyersanyagot. Az aszteroida a resource attribútumát null-ra állítja, visszatérési értékül a nyersanyagot adja. Ha kezdetben resource=null volt, a függvénynek nincs mellékhatása, és null-lal tér vissza.
- **void accept(Asteroid neighbor):** Hozzáadja a neighbor aszteroidát az aszteroida neighbors kollekciójához.
- **void remove(Asteroid neighbor):** Eltávolítja a neighbor aszteroidát a neighbors kollekcióból.
- **void drilled():** Az aszteroida rockLayerThickness attribútumát egygel csökkenti, amennyiben az pozitív volt. Ha ekkor a köpenyvastagság 0, és a resource attribútuma nem null, akkor a sun objektum isCloseTo(Asteroid a) függvényével lekérdezi, hogy napközelben található-e. Ha igen, akkor meghívja a resource nyersanyag drilledOut() függvényét, jelezve, hogy napközeli aszteroidán felszínre került.
- **void minedBy(Settler) s:** Az s telepes bányászik az aszteroidán. Ha az aszteroida köpenyvastagsága nem 0, akkor a függvénynek nincs hatása. Ha a köpenyvastagság 0, és az aszteroida resource attribútuma nem null, akkor eltávolítja azt a magjából (removeResource()), és eltároltatja azt az s telepessel (s.accept(resource)).
- **boolean isMined():** Visszatér annak logikai értékével, hogy az aszteroida megfűrt és üreges, vagyis rockLayerThickness=0 és resource=null.

- **void explodedBy(RadioactiveResource rr)**: A megfúrt, napközelben lévő, radioaktív nyersanyagot tartalmazó aszteroida felrobban. minden rajta tartózkodó karakternek meghívja a reactToExplosion() függvényét. Az aszteroida szomszédain a remove(Asteroid neighbor) függvényt hívja, amivel törli magát a szomszédai szomszédsági listáiból, majd megsemmisíti a körülötte keringő teleportkapukat a párjaikkal együtt. Végül eltávolítja magát a játékból a Game osztály removeAsteroid() függvényét hívva.
- **void destroySurface()**: Az aszteroidán végigsöpör a napvihar. Ha az aszteroida nincs megfúrva (rockLayerThickness > 0), vagy a mag nem üreges (getResource() ≠ null), akkor a függvény meghívja az aszteroidán tartózkodó karakterek die() függvényét.
- **boolean checkSpaceBase()**: Visszatér annak logikai értékével, hogy az adott aszteroidán tartózkodó telepeseknél rendelkezésre áll-e az ūrbázis felépítéshez szükséges nyersanyagmennyiség.
- **Resource getResource()**: Visszaadja az adott aszteroida magjában található nyersanyagot.
- **Sun getSun()**: Visszaadja a játékban levő Napot.
- **Character[0..*] getCharacters()**: Visszaadja az adott aszteroidán tartózkodó karakterek kollekcióját.
- **Asteroid[0..*] getNeighbors()**: Visszaadja az adott aszteroidával szomszédos aszteroidák kollekcióját, beleértve ebbe az aszteroida közvetlen szomszédait (neighbors) és a teleportkapuk (gates) általi szomszédokat is.
- **Asteroid getNeighbor(int d)**: Megadja az aszteroida d-edik szomszédját (az aszteroidához tartozó teleportkapuk általi szomszédai figyelembevételével).
- **TeleportingGate getGate()**: Visszaadja az adott aszteroidához tartozó teleportkapukat.

3.3.3 Character

- **Felelősség**

Ez az osztály absztrakt ōsosztályként szolgál a játék karaktereihez. A karakterek aszteroidákon tartózkodhatnak, köztük mozoghatnak, sziklaköpenyüket képesek fűrni. Különbözöképpen reagálhatnak aszteroidák robbanására.

- **Ósosztályok**

- **Interfészek**

- **Attribútumok**

- **Asteroid place**: az aszterioda, amelyen az adott karakter aktuálisan tartózkodik

- **Metódusok**

- **ResourceCollection getCollectedResources()**: Visszatér a karakter által tárolt nyersanyagok listájával, alapértelmezetten egy üres ResourceCollection listával.
- **void move(int d)**: Átmozog az aktuális aszteroidáról annak egy szomszédjára. Először lekérdezi az aktuális aszteroida d-edik szomszédját (a getNeighbor(int d) függvényel). Ezután eltávolítja magát a jelenlegi aszteroidáról (remove(Character c)), majd a visszakapott szomszéd aszteroidára lép (accept(Character c)).

- **void drill()**: Csökkenti az aktuális aszteroida köpenyvastagságát: meghívja az aszteroida drilled() metódusát.
- **void die()**: A karakter halálát, jelenti: eltávolítja a karaktert az aszteroidájáról (place.remove(Character c)). A leszármazottakban felüldefiniálható.
- **void reactToExplosion()**: A metódus, amit a karakterek akkor kell végrehajtania, ha az aszteroida, amelyen tartózkodik, felrobban. Alapértelmezetten a karakter die() függvényét hívja. A leszármazottak felüldefiniálhatják.

3.3.4 Coal

- **Felelősség**

A szén nyersanyagot reprezentálja.

- **Ősosztályok**

Resource

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **String checkedBy(ResourceCollection rc)**: Meghívja rc check(Coal c) függvényét, paraméterként a saját referenciáját átadva. Ezzel jelzi, hogy szén típusú nyersanyag objektum.

3.3.5 Game

- **Felelősség**

A játék indításáért és befejezéséért felel, valamint a játékban lévő telepesek nyilvántartását végzi.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **Asteroid[0..*] asteroids**: a játékban jelenlévő aszteroidák kollekciója
- **Sun sun**: az aszteroidaövbeli Nap
- **Settler[0..*] settlersAlive**: az adott pillanatban életben levő telepesek listája

- **Metódusok**

- **void startGame()**: Inicializálja az aszteroidákat, a Napot, és elhelyezi a telepeseket valamelyik aszteroidán. Egyes aszteroidák magját üresen hagyja, másokban pedig véletlenszerűen valamilyen nyersanyagot hoz létre. Beállítja az aszteroidák szomszédsági viszonyait is.

- **void endGame()**: Ha van még élő telepes a játékban (settlersAlive nem üres), akkor a telepesek győzelmét írja ki a képernyőre, ellenkező esetben a telepesek vereségéről szóló üzenetet jelenít meg.
- **void removeSettler(Settler s)**: Törli a meghaló s telepest a játékból. Ha nincs több életben levő telepes, akkor meghívja az endGame() függvényt.
- **void removeAsteroid(Asteroid a)**: Törli a megsemmisülő a aszteroidát az asteroids kollekcióból.
- **Asteroid[0..*] getAsteroids()**: Visszaadja a játék összes aszteroidájából álló kollekciót.

3.3.6 Ice

- **Felelősség**

A vízjég nyersanyagot reprezentálja. Ha napközeli, teljesen megfúrt aszteroida magjában található, akkor elszublimál.

- **Ősosztályok**

Resource

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **String checkedBy(ResourceCollection rc)**: Meghívja rc check(Ice i) függvényét, paraméterként a saját referenciáját átadva. Ezzel jelzi, hogy vízjég típusú nyersanyag objektum.
- **void drilledOut()**: Hatására a vízjég elszublimál: meghívja az asteroid.removeResource() metódust a saját aszteroidáján.

3.3.7 Iron

- **Felelősség**

A vas nyersanyagot reprezentálja.

- **Ősosztályok**

Resource

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **String checkedBy(ResourceCollection rc)**: Meghívja rc check(Iron i) függvényét, paraméterként a saját referenciáját átadva. Ezzel jelzi, hogy vas típusú nyersanyag objektum.

3.3.8 RadioactiveResource

- **Felelősség**

A radioaktív nyersanyagtípust reprezentálja, radioaktív nyersanyagok ōsosztályaként szolgál. Ha napközelben lévő, megfűrt aszteroida tartalmazza, akkor felrobbantja az aszteroidáját.

- **Ősosztályok**

Resource

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **void drilledOut()**: Meghívja az aszteroidája explodedBy(RadioactiveResource rr) metódusát (robban).

3.3.9 Resource

- **Felelősség**

A játékban levő nyersanyagok ōsosztálya. Többféle nyersanyag létezik, amelyek lehetnek aszteroida magjában vagy telepesnél tárolva. Ha egy nyersanyag napközelben levő megfűrt aszteroida magjában található, akkor az aszteroidán - a konkrét fajtájától függően - speciális hatást válthat ki.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **Settler settler**: a telepes, akinél az adott nyersanyagegység tárolódik (ha éppen telepesnél található)
- **Asteroid asteroid**: az aszteroida, amelynek magjában az adott nyersanyag elhelyezkedik (ha éppen aszteroida magjában található)

- **Metódusok**

- **void drilledOut()**: Azt a helyzetet kezeli, amikor a nyersanyag éppen egy napközelben levő, megfűrt aszteroida magjában található. A metódus általános esetben nem csinál semmit, de a speciális működésű leszármazott osztályokban felüldefiniálható.
- **abstract String checkedBy(ResourceCollection rc)**: A metódus, amit akkor kell meghívni, amikor egy rc nyersanyaglista objektum szeretné megállapítani a nyersanyag típusát. A Resource osztály minden példányosítható leszármazott osztályának kötelező felüldefiniálnia. A konkrét nyersanyag objektum a saját típusának jelzésére a ResourceCollection osztály megfelelő check(...) függvényét hívja, paraméterül a saját referenciáját átadva.

3.3.10 ResourceCollection

- **Felelősség**

Nyersanyag-egységek halmazát reprezentáló osztály. Felelőssége különféle nyersanyagok tárolása: Resource-objektumok multihalmazát reprezentálja. Képes ellenőrizni a játékban építhető objektumokhoz szükséges nyersanyagok meglétét, illetve eltávolítani a felhasznált nyersanyagokat.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **Map<String, Integer> resources:** asszociatív kollekció, amely a nyersanyagok típusainak azonosítóihoz rendeli, hogy az adott típusú nyersanyagból hány egységet reprezentál a nyersanyagtároló; a resources Map kulcsértékei halmaza a játékban előforduló nyersanyagokat azonosító String-ek halmaza (ezeket csak ezen az osztályon belül használja); a darabszámok nemnegatív egészek
- **int capacity:** a nyersanyag-kollekció által tárolható nyersanyagok számának maximuma; a tárolt nyersanyagok száma a különféle nyersanyag-típusok darabszámainak az összege

- **Metódusok**

- **String check(Coal c):** Visszatér a szén nyersanyagot azonosító egyedi String-gel.
- **String check(Iron i):** Visszatér a vas nyersanyagot azonosító egyedi String-gel.
- **String check(Ice i):** Visszatér a vízjég nyersanyagot azonosító egyedi String-gel.
- **String check(Uranium i):** Visszatér az uránt azonosító egyedi String-gel.
- **void add(Resource r):** Hozzáadja a tárolóhoz az egy egységnyi r nyersanyagot. Ha a tároló már a lépés előtt összesen capacity számú nyersanyagot tartalmazott, akkor a tárolóba nem fér több nyersanyag, és nem történik semmi.
- **void remove(Resource r):** Eltávolít a tárolóból egy r típusával megegyező típusú nyersanyagot, feltéve, hogy van a tárolóban ilyen nyersanyag.
- **boolean containsAll(ResourceCollection rc):** Ellenőrzi, hogy a tároló tartalmazza-e az rc ResourceCollection-ban szereplő összes nyersanyagot legalább olyan darabszámban, mint az rc tároló, és visszatér ennek logikai értékével.
- **void addAll(ResourceCollection rc):** Hozzáadja a nyersanyag-tárolóhoz az rc tároló tartalmát (nyersanyagonként a megfelelő darabszámban). Ha ezzel túllépné, a kapacitáskorlátot, a műveletnek nincs hatása.
- **void removeAll(ResourceCollection rc):** Törli a tárolóból az rc tároló tartalmát. Ha a tároló nem tartalmazza rc minden nyersanyag-egységét (containsAll(rc) hamis), akkor a törlés nem történik meg.
- **boolean checkAIRobotConstituents(ResourceCollection rc):** Ha rc tartalmazza egy AIRobot építéséhez szükséges összes nyersanyagot (legalább egy-egy egység vasat, szenet, illetve uránt), akkor eltávolítja az rc tárolóból ezeket a nyersanyag-egységeket, és igaz értékkal tér vissza. Ha rc-ben nincsenek meg a robotépítéshez szükséges nyersanyagok, akkor hamissal tér vissza, és nem változtat az rc tároló tartalmán.

- **boolean checkGatePairConstituents(ResourceCollection rc)**: Ha rc tartalmazza egy teleportkapu-pár építéséhez szükséges összes nyersanyagot (legalább egy-egy egység vasat, szenet, illetve uránt), akkor eltávolítja az rc tárolóból ezeket a nyersanyag-egységeket, és igaz értékkel tér vissza. Ha rc-ben nincsenek meg a teleportkapu-pár építéséhez szükséges nyersanyagok, akkor hamissal tér vissza, és nem változtat az rc tároló tartalmán.
- **boolean checkSpaceBaseConstituents(ResourceCollection rc)**: Visszatér annak a logikai értékével, hogy az rc nyersanyagtároló tartalmazza-e az ürbázishoz szükséges nyersanyagmennyiséget (a játékban előforduló mindenfajta nyersanyagból legalább 3 egység szükséges az ürbázis megépítéséhez).

3.3.11 Settler

- **Felelősség**

A karakterek egyik fajtája a telepes, amely a Character-ben megfogalmazottakon túl képes egységnnyi nyersanyagot kibányászni vagy visszahelyezni az aszteroidamagba. Emellett építhet robotot és teleportkapu-párt, illetve az elkeszült teleportkapukat pályára állíthatja. Nyilvántartja a nála található nyersanyagokat és teleportkapukat.

- **Ősosztályok**

Character

- **Interfészek**

- **Attribútumok**

- **Game game**: a játékot reprezentáló osztály
- **ResourceCollection resourcesCollected**: a telepes által tárolt nyersanyagok kollekciója; a kollekció kapacitása 10
- **TeleportingGate[0..2] gatesCreated**: a telepes által tárolt teleportkapuk kollekciója

- **Metódusok**

- **ResourceCollection getCollectedResources()**: Visszatér a telepes által összegyűjtött nyersanyagok kollekciójával.
- **void accept(Resource r)**: A telepes eltárolja a kibányászott r nyersanyagot a resourcesCollected kollekciójában (resourcesCollected.add(r)).
- **void remove(Resource r)**: A telepes eltávolítja az r nyersanyagot a resourcesCollected kollekciójából (resourcesCollected.remove(r)).
- **void remove(TeleportingGate: tg)**: A telepes eltávolítja a tg teleportkaput a gatesCreated kollekciójából.
- **void mine()**: A telepes kibányásza az adott megfűrt aszteroida magjában található nyersanyagot. Ehhez meghívja az aszteroida minedBy() metódusát. Ha a bányászat kezdetén az űrhajóban már nincs hely újabb nyersanyagnak, akkor a telepes nem tud bányászni, a függvénynek nincs hatása.
- **void restore(Resource r)**: Meghívja a place aszteroida Accept(Resource r) metódusát. Ha az aszteroida üres, akkor az aszteroida eltárolja az r nyersanyagot, és eltávolítja azt a Settler nyersanyagai közül. Ha az aszteroida magjában már volt nyersanyag, akkor nem történik semmi.

- **void createAIRobot()**: A telepes meghívja a ResourceCollection checkAIRobotConstituents(ResourceCollection rc) függvényét, paraméterként a saját resourcesCollected nyersanyagkészletét átadva. Ha ez a függvény igazzal tért vissza, az építés lehetséges; ekkor létrehoz egy új AIRobot-ot, és hozzáadja a saját aszteroidájához (accept(Character c)). Ha a függvény hamissal tért vissza, akkor az építés nem lehetséges, nem történik semmi.
- **void createGatePair()**: Ha a telepésnél van még elkészült teleportkapu (gatesCreated nem üres), akkor a függvénynek nincs hatása. Különben a telepes meghívja a ResourceCollection checkGatePairConstituents(ResourceCollection rc) függvényét, paraméterként a saját resourcesCollected nyersanyagkészletét átadva. Ha ez a függvény igazzal tért vissza, akkor az építés lehetséges: a telepes létrehoz két új TeleportingGate-et, ezeket beállítja egymás párnainak (setPair()), és az ūrhajóján a gatesCreated kollekcióban eltárolja őket. Ha a checkGatePairConstituents() függvény hamissal tért vissza, akkor az építés nem lehetséges, és nem változik semmi.
- **void releaseGate()**: A telepes a gatesCreated kollekcióból kiválaszt egy teleportkaput, majd az aktuális aszteroidája körüli pályára állítja az Asteroid accept(TeleportingGate tg) függvényével. Ha a telepésnél nincs elkészült teleportkapu (gatesCreated üres), akkor nem történik semmi.
- **void die()**: A telepes meghal: megsemmisíti a nála levő teleportkapukat azok die() függvényeivel, eltávolítja magát az aszteroidájáról, végül meghívja a game removeSettler(Settler s) metódusát.

3.3.12 ISteppable

- **Felelősség**

Olyan objektumot jelöl, amely a játékban külső irányítás (játékos aktor beavatkozása) nélkül léptethető.

- **Ősosztályok**

- **Metódusok**

- **void step()**: A művelet, amelyet az objektum az adott lépésben végez.

3.3.13 Sun

- **Felelősség**

A játékban levő Napot reprezentálja. A napviharok kiindulópontja, valamint a napközelség számításához szükséges.

- **Ősosztályok**

- **Interfészek**

ISteppable

- **Attribútumok**

- **long timeToSunStorm**: a következő napvihar bekövetkezéséig hátralévő idő

- **Asteroid[0..*] neighboringAsteroids:** a Nappal közvetlenül szomszédos aszteroidák listája
- **Metódusok**
 - **void step():** Ha a timeToSunStorm attribútum pozitív, akkor egyelőre csökkenti (várakozik). Ha pedig 0, akkor napvihart indít a sunStorm() függvényt hívva, majd a timeToSunStorm-nak beállít egy véletlen egész értéket.
 - **void sunStorm():** Lekérdezi a Game-től az aszteroidák kollekcióját, majd egyesével meghívja mindegyik aszteroida destroySurface() függvényét.
 - **boolean isCloseTo(Asteroid a):** Megállapítja, hogy az a aszteroida napközeli-e, vagyis legfeljebb másodszomszédos-e a Nappal. Ehhez felhasználja a neighboringAsteroids listáját: megvizsgálja a benne szereplő aszteroidákat, valamint azok összes szomszédait is.

3.3.14 TeleportingGate

- **Felelősség**

Két teleportkapu egy teleportkapu-párt alkot, amely aszteroidák közötti összeköttetést biztosít. A frissen épített kapuk az építő telepes űrhajóján tárolódnak. Aszteroida körüli pályára állítás után az adott aszteroidához tartoznak. Ha a kapupár minden tagját aszteroida körüli pályára állították, akkor az aszteroidák, amelyekhez tartoznak, szomszédosnak számítanak. Ha az egyik kapu megsemmisül, a párja vele együtt automatikusan eltűnik, bárhol is van éppen.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **TeleportingGate pair:** az adott teleportkapu párja, amellyel összeköttetésben áll
- **Asteroid asteroid:** az aszteroida, amely körül az adott teleportkapu kering
- **Settler settler:** a telepes, aki tárolja a létrehozott, de még pályára nem állított teleportkaput

- **Metódusok**

- **TeleportingGate getPair():** Visszaadja a teleportkapu pair párját.
- **void setPair(TeleportingGate tg):** Beállítja tg-t a teleportkapu párjaként.
- **Asteroid getAsteroid():** Visszaadja az aszteroidát, amely körül a teleportkapu kering. Ha a kaput még nem állították pályára, akkor null-lal tér vissza.
- **void setAsteroid(Asteroid a):** Beállítja a teleportkapu aszteroidáját, amely körül keringeni fog.
- **void die():** Eltávolítja a párját (pair) annak aszteroidájáról/űrhajójáról attól függően, hogy azt már pályára állították-e (remove(TeleportingGate tg)). Beállítja a párjának a párját null-ra (megszünteti az összeköttetést), végül saját magát is eltávolítja az aszteroidájáról/űrhajójáról. Ha a függvényhívás elején a teleportkapu párja már null, akkor csak magát távolítja el: ekkor a másik teleportkapu semmisült meg előbb, és az már megszüntette ennek a kapunak a kapcsolatait is.

3.3.15 Timer

- **Felelősség**

Egy órajel jellegű időzítőt reprezentál, amely periodikusan léptet minden léptethető objektumot a játékban.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **ISteppable[0..*] steppables:** a játék léptethető objektumainak listája

- **Metódusok**

- **void addStepable(Steppable s):** Hozzáadja s-t a steppables listához.
- **void removeStepable(Steppable s):** Eltávolítja s-t a steppables listából.
- **tick:** minden steppable listabeli objektumot léptet úgy, hogy meghívja a step() függvényét.

3.3.16 Uranium

- **Felelősség**

Az urán nyersanyagot reprezentálja.

- **Ősosztályok**

RadioactiveResource → Resource

- **Interfész**

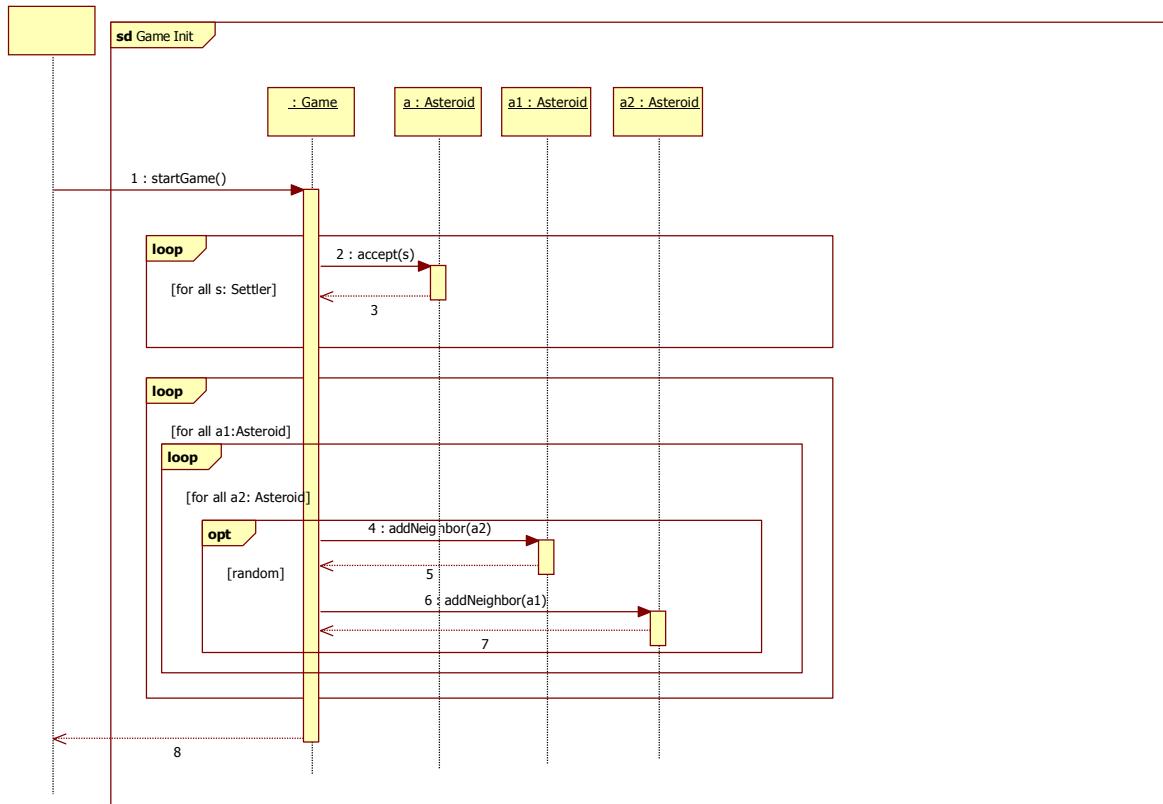
- **Attribútumok**

- **Metódusok**

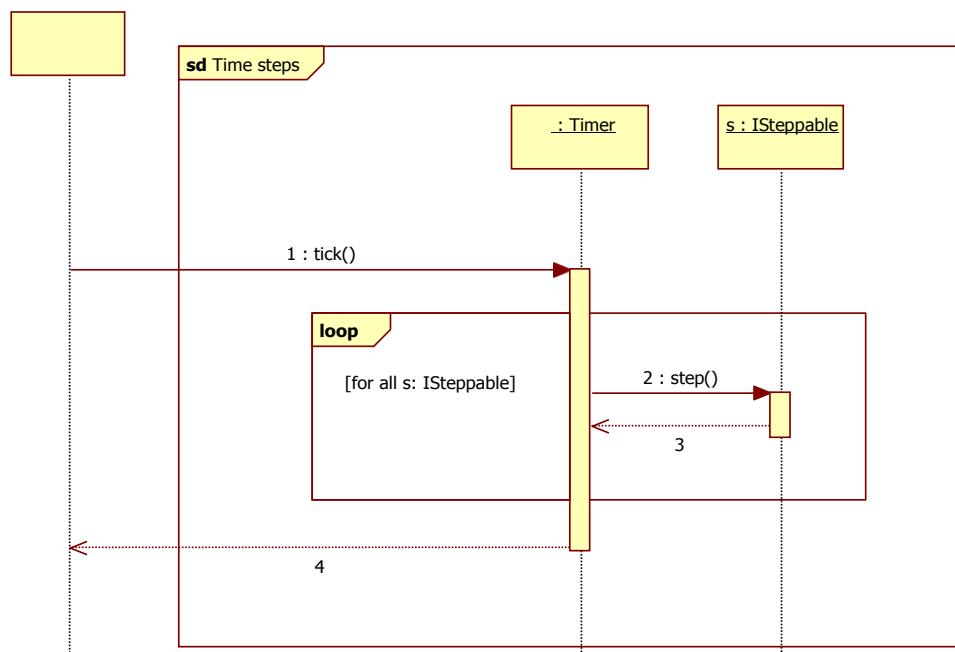
- **String checkedBy(ResourceCollection rc):** Meghívja rc check(Uranium u) függvényét, paraméterként a saját referenciáját átadva. Ezzel jelzi, hogy urán típusú nyersanyag objektum.

3.4 Szekvencia diagramok

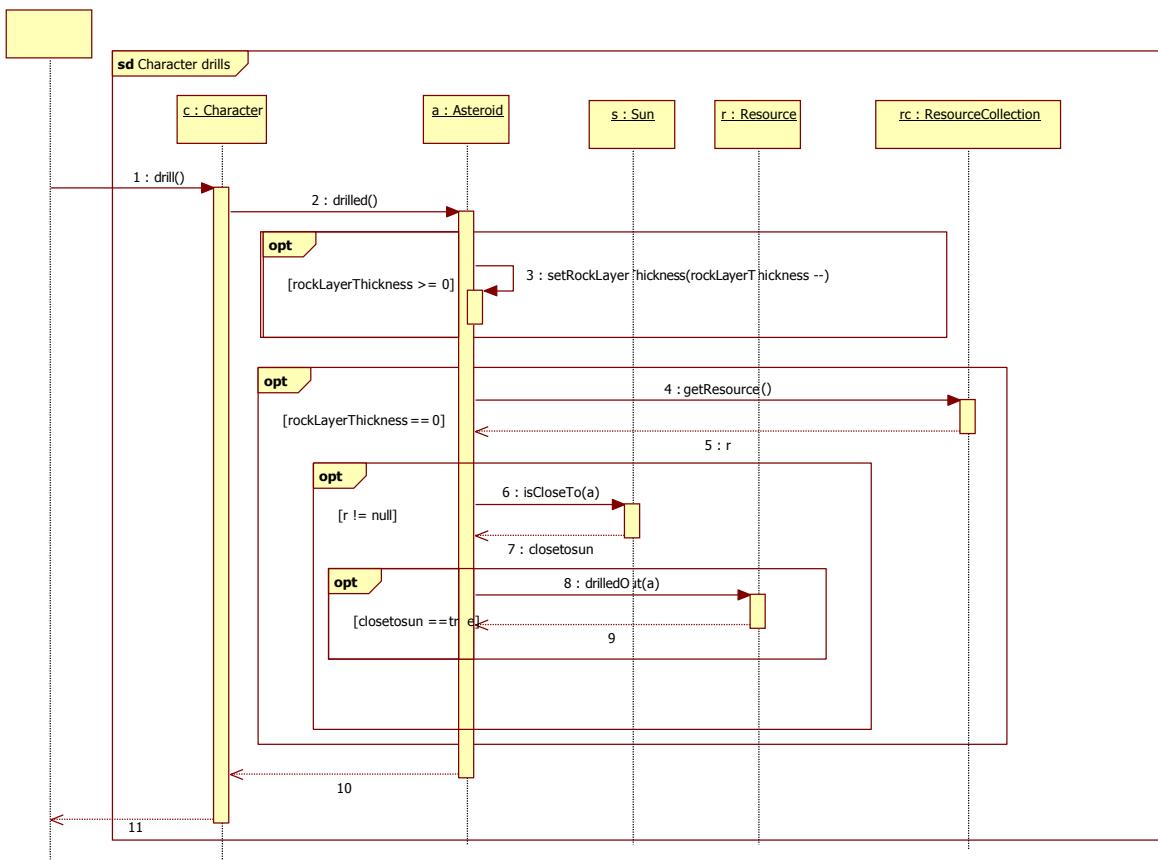
3.4.1 Game init



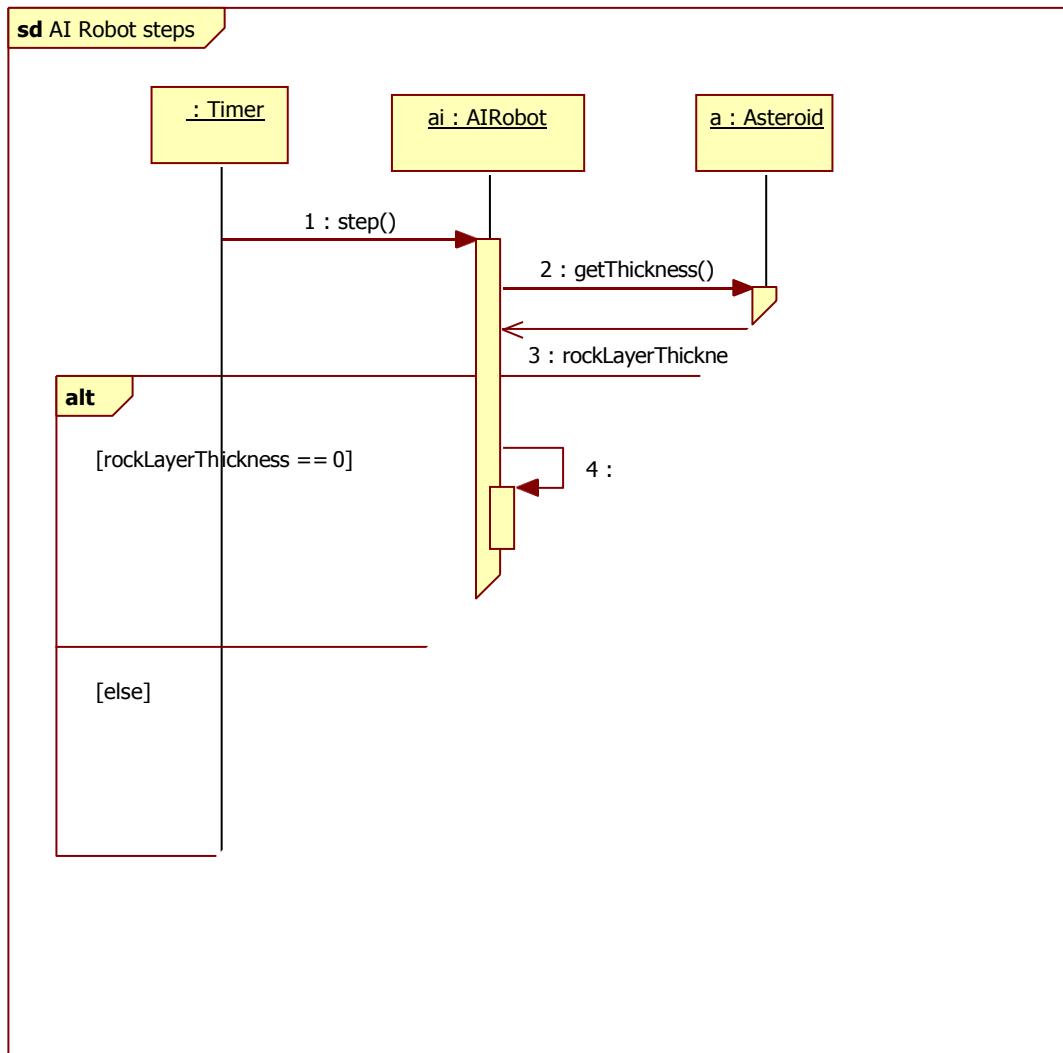
3.4.2 Time steps



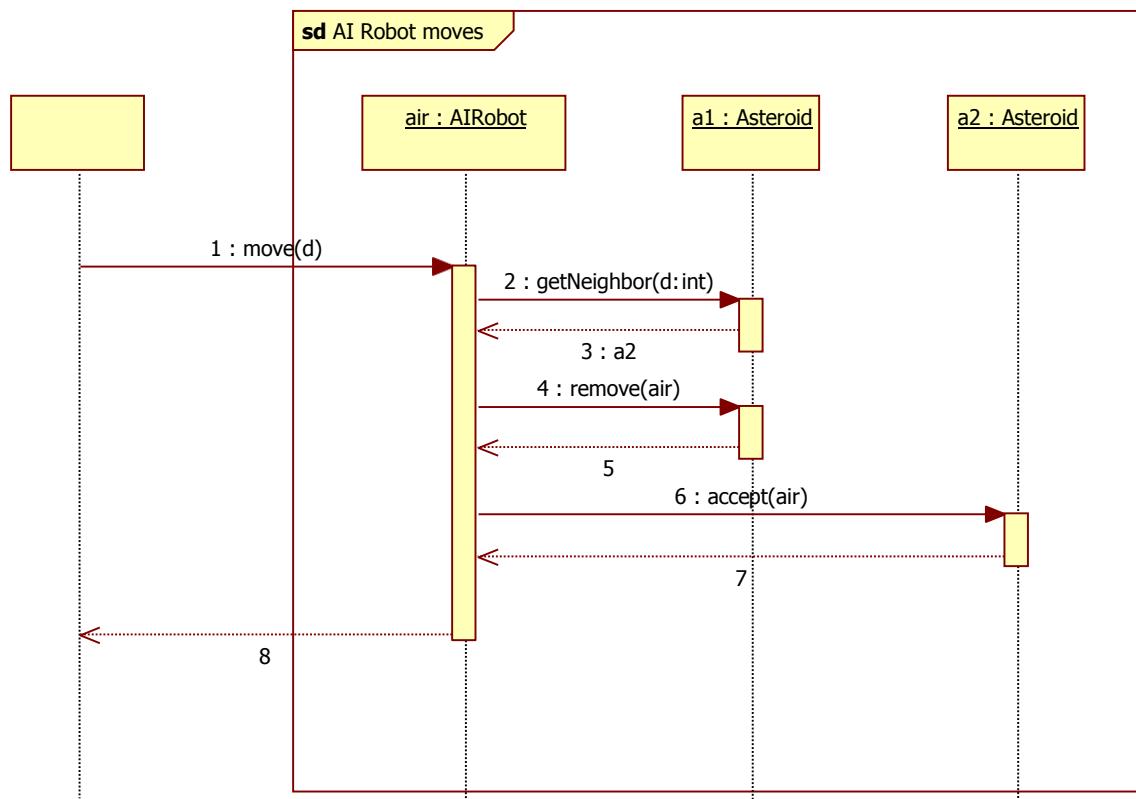
3.4.3 Character drills



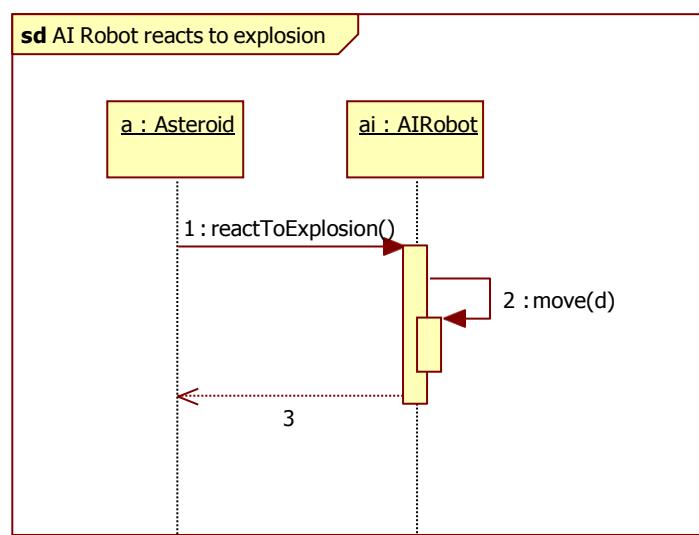
3.4.4 AI Robot steps



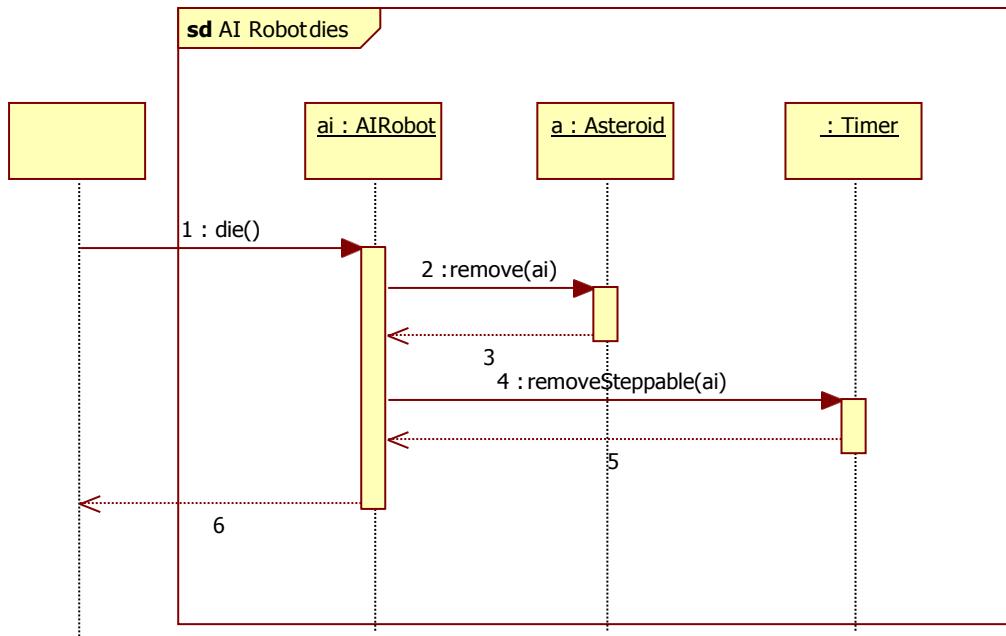
3.4.5 AI Robot moves



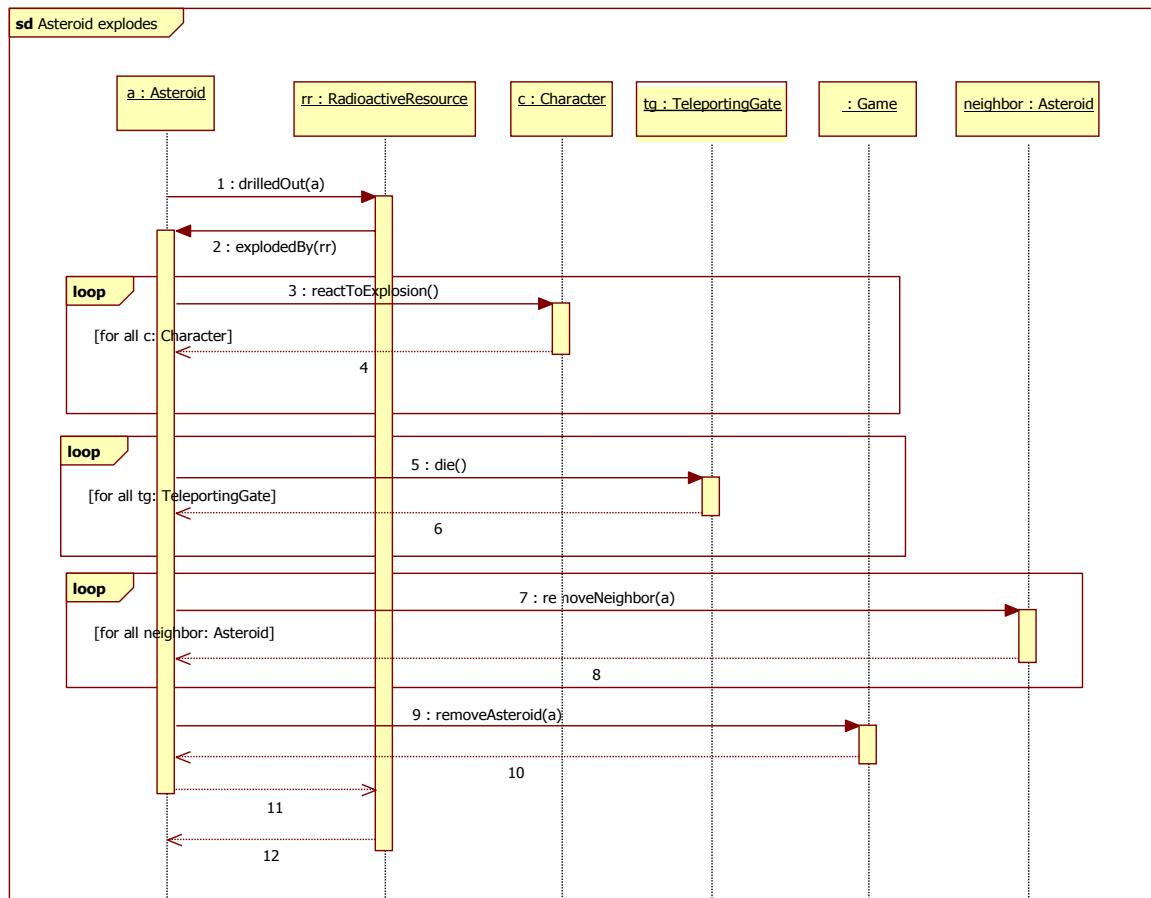
3.4.6 AI Robot reacts to explosion



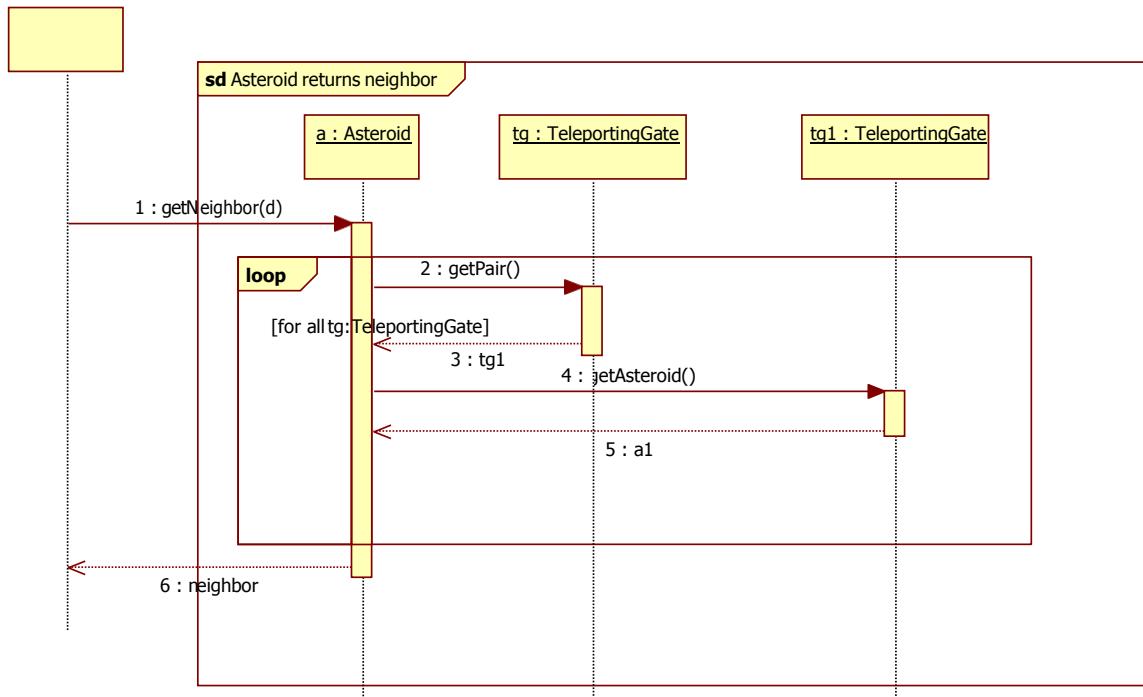
3.4.7 AI Robot dies



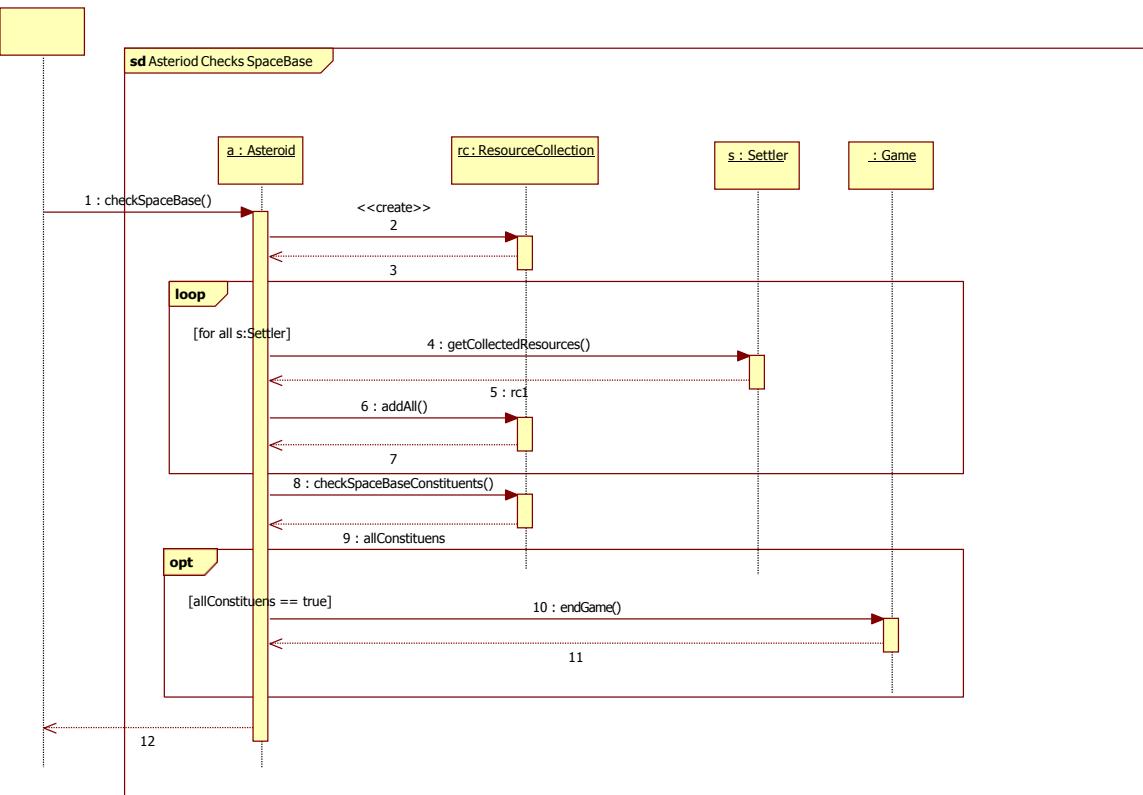
3.4.8 Asteroid explodes



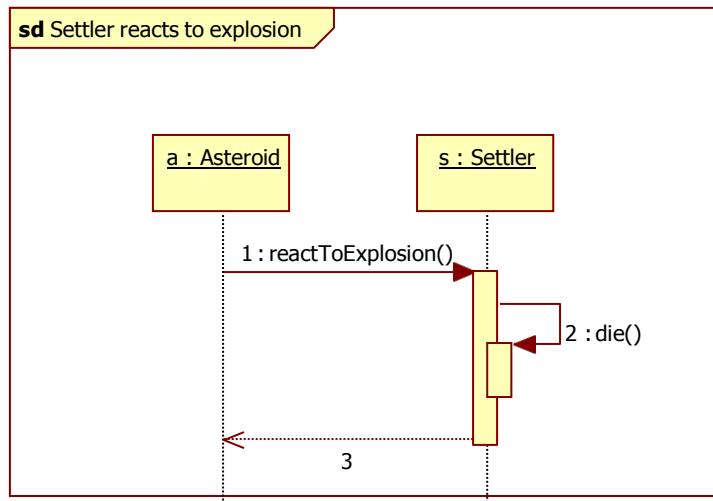
3.4.9 Asteroid returns neighbor



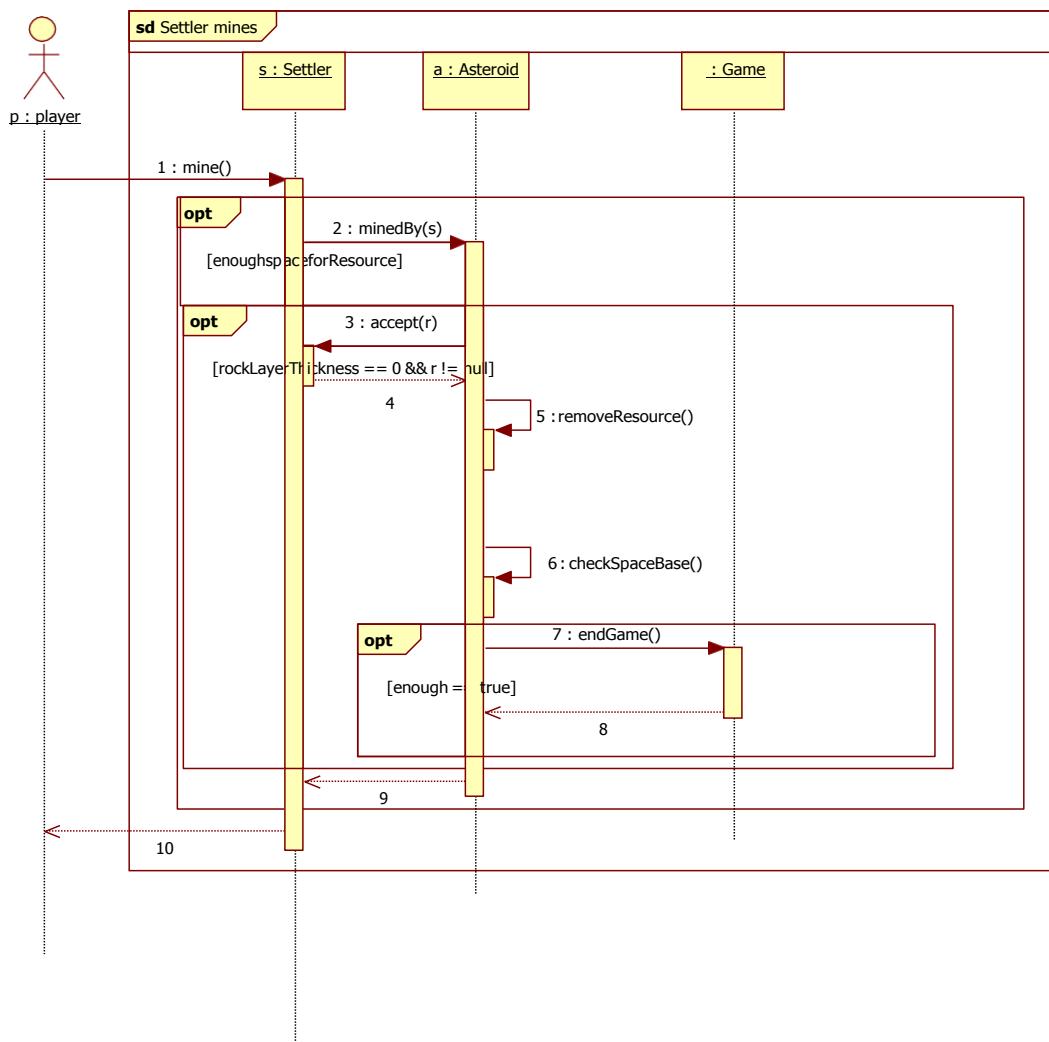
3.4.10 Asteroid Checks Spacebase



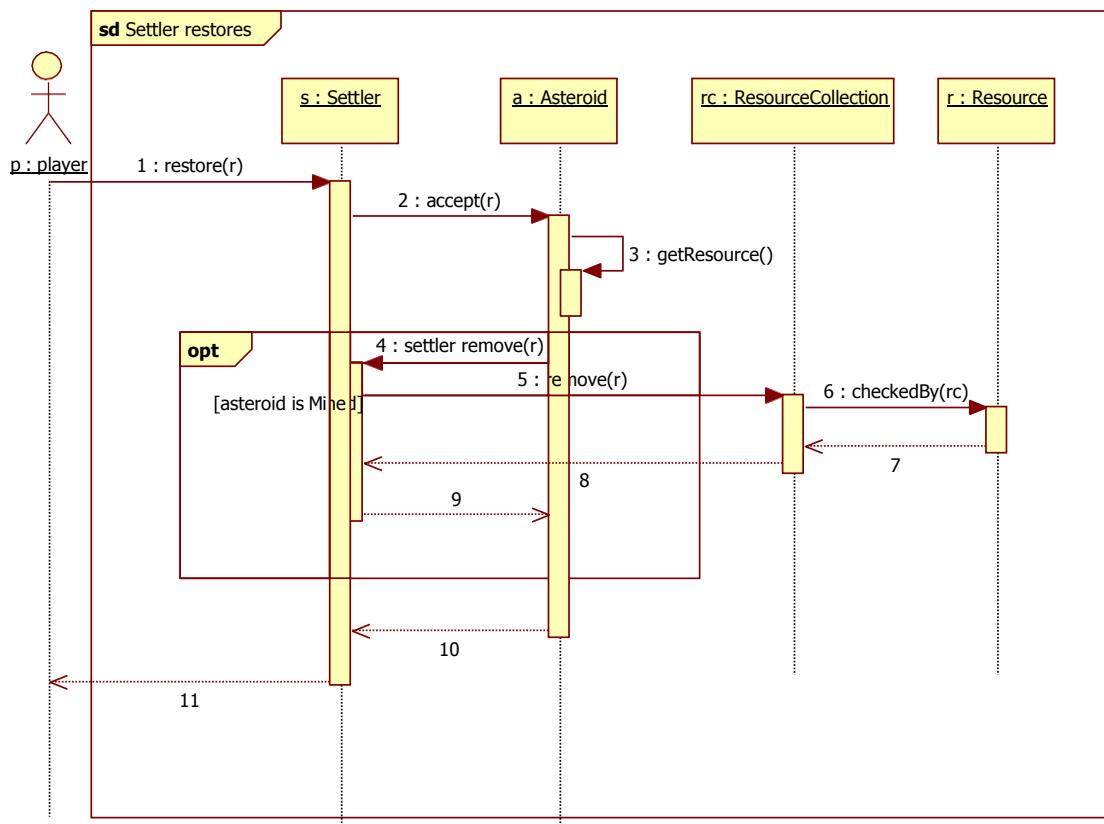
3.4.11 Settler reacts to explosion



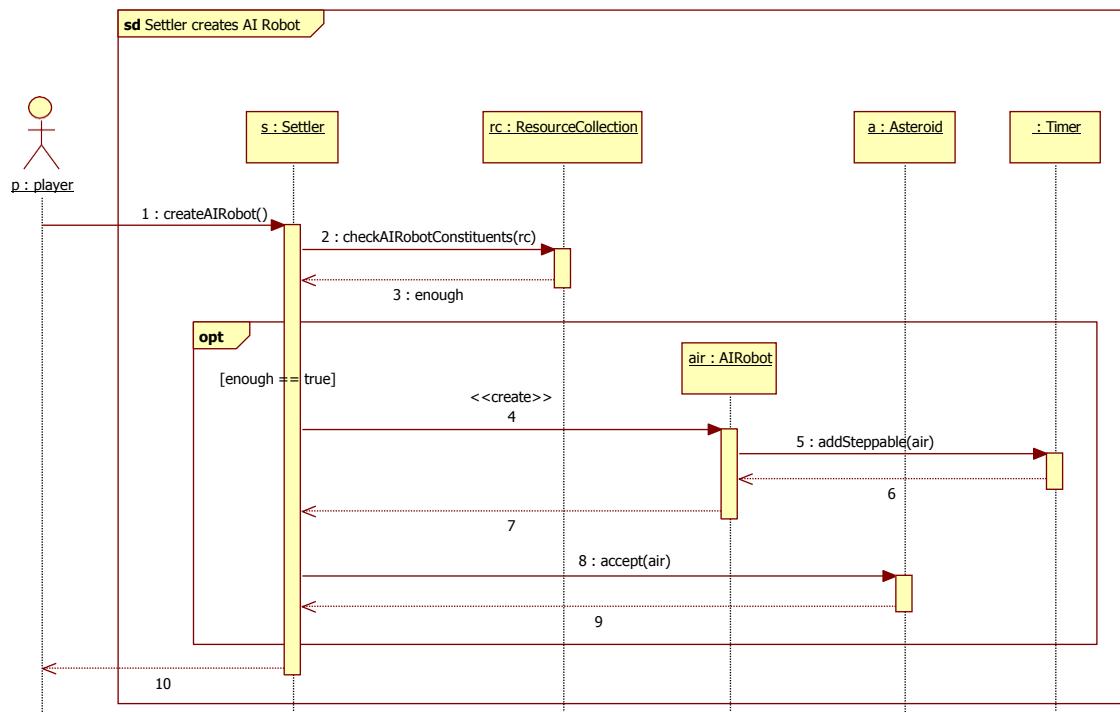
3.4.12 Settler mines



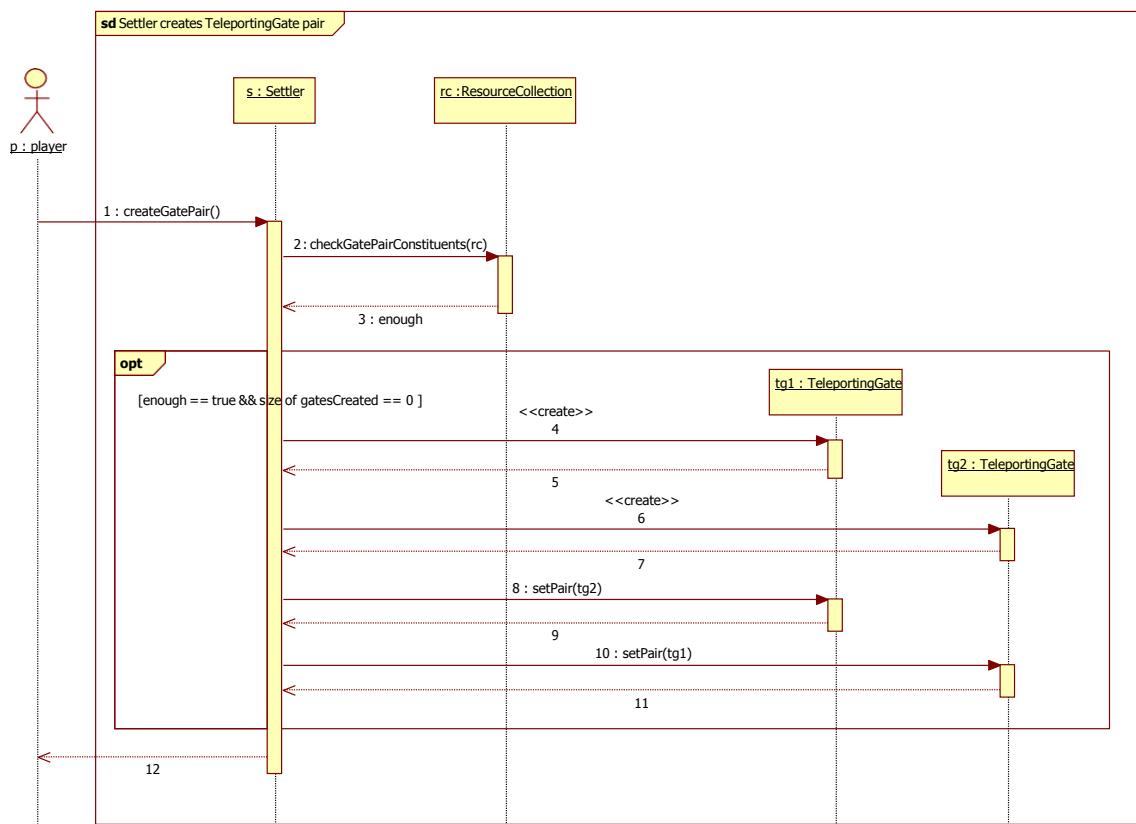
3.4.13 Settler restores



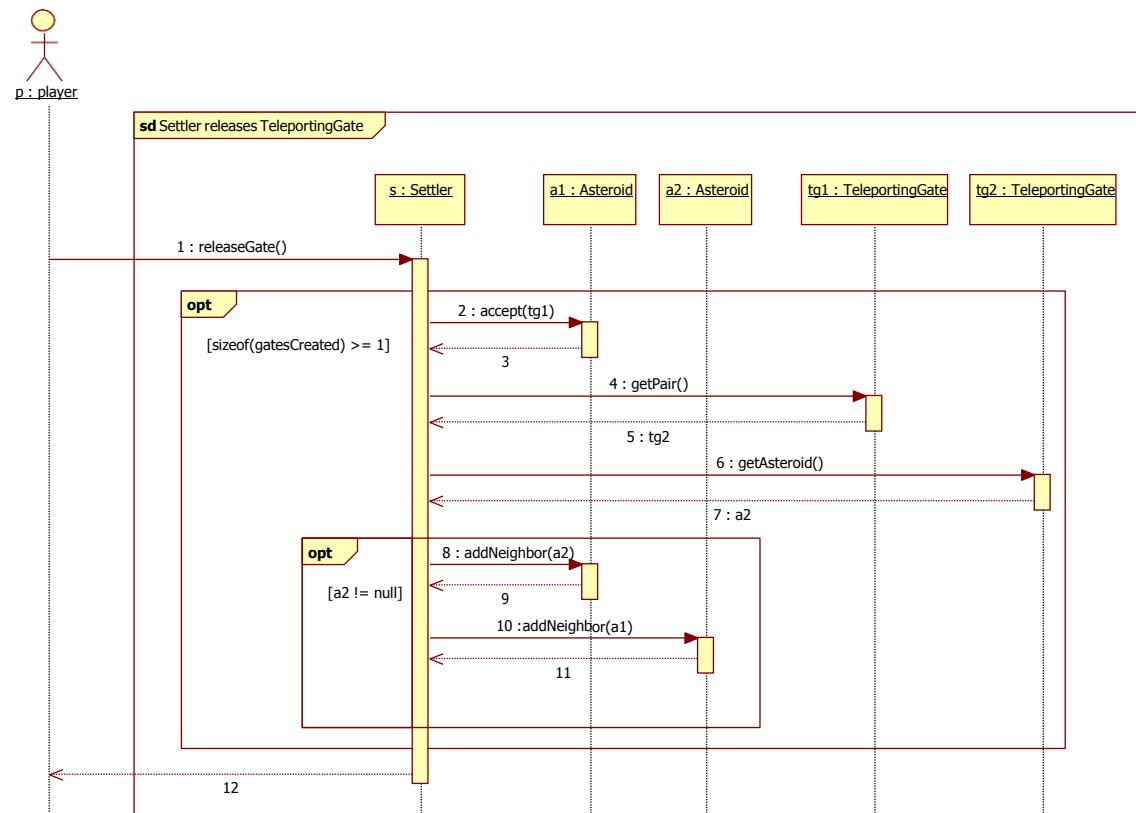
3.4.14 Settler creates AI Robot



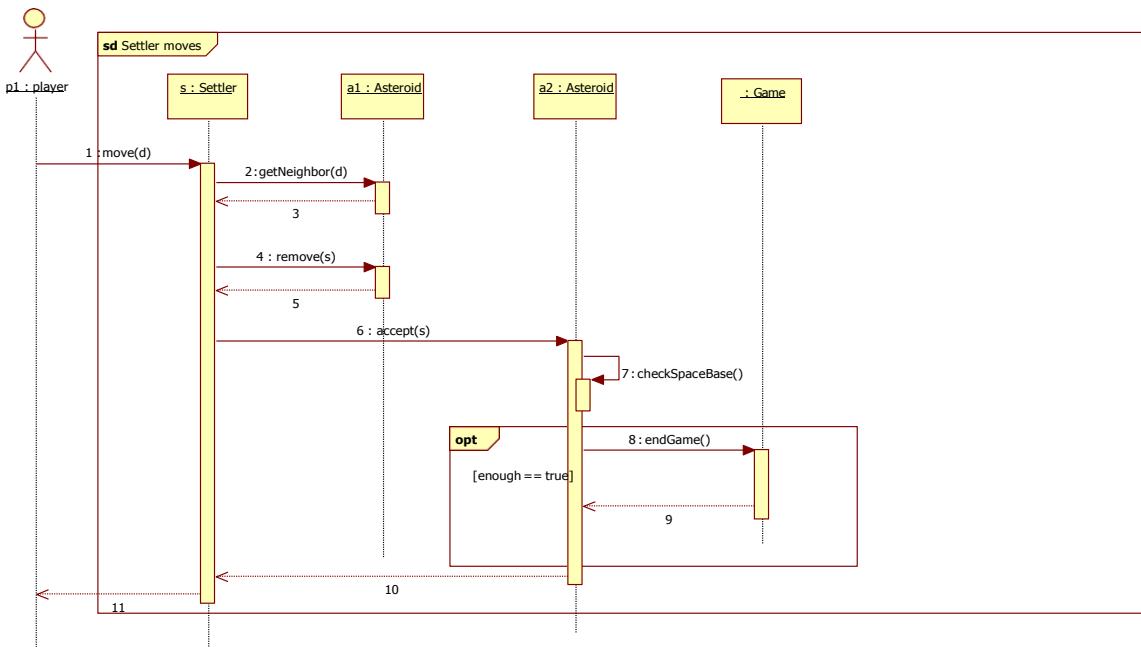
3.4.15 Settler creates TeleportingGate pair



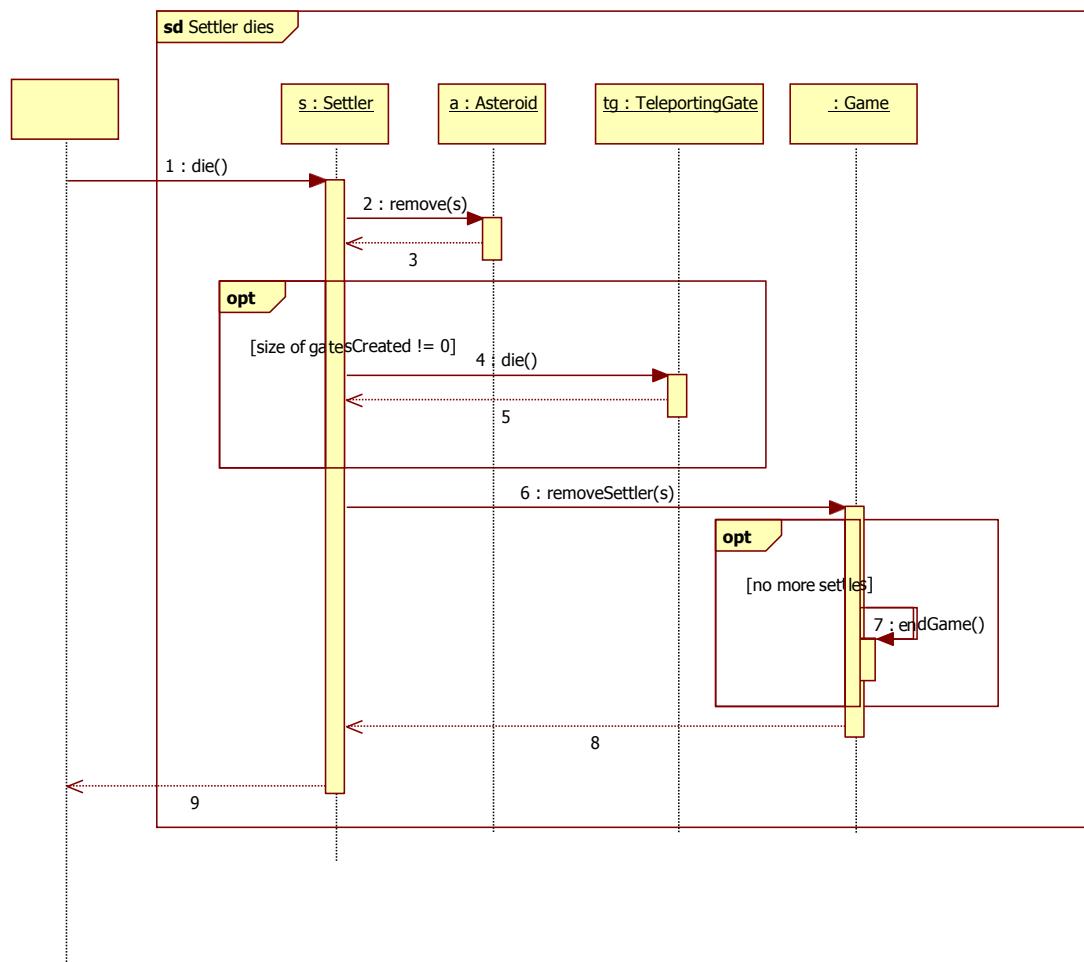
3.4.16 Settler releases TeleportingGate



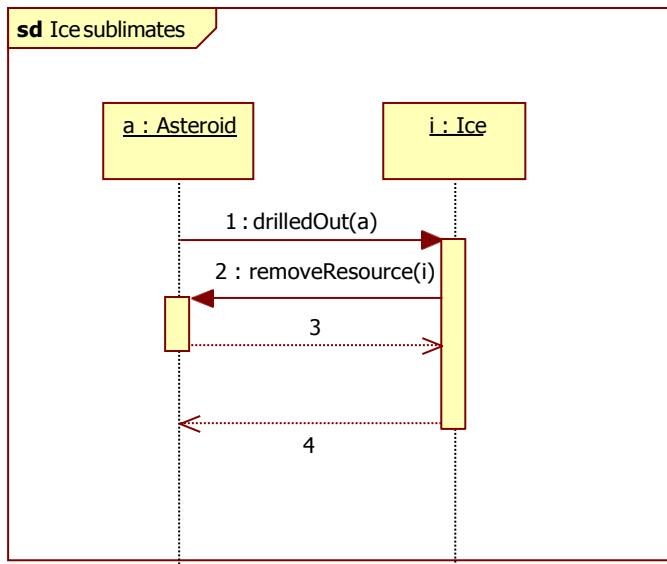
3.4.17 Settler moves



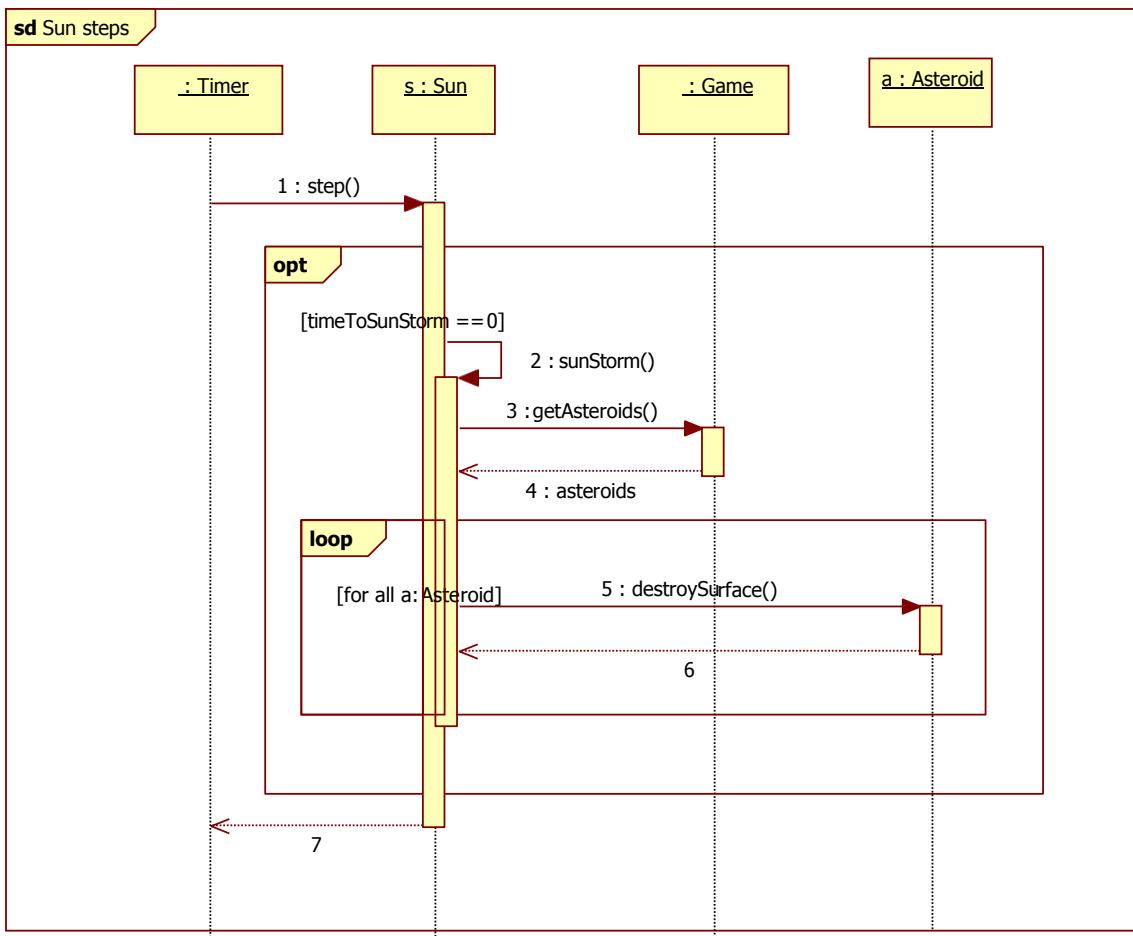
3.4.18 Settler dies



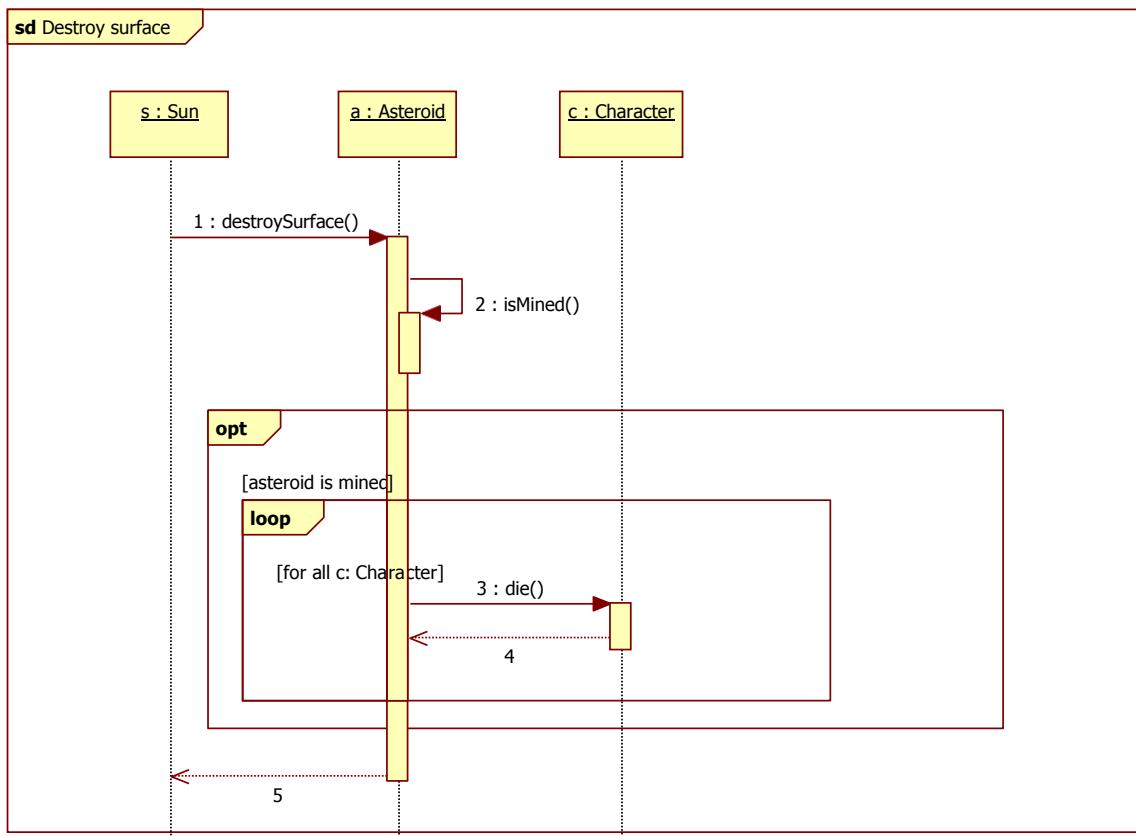
3.4.19 Ice sublimates



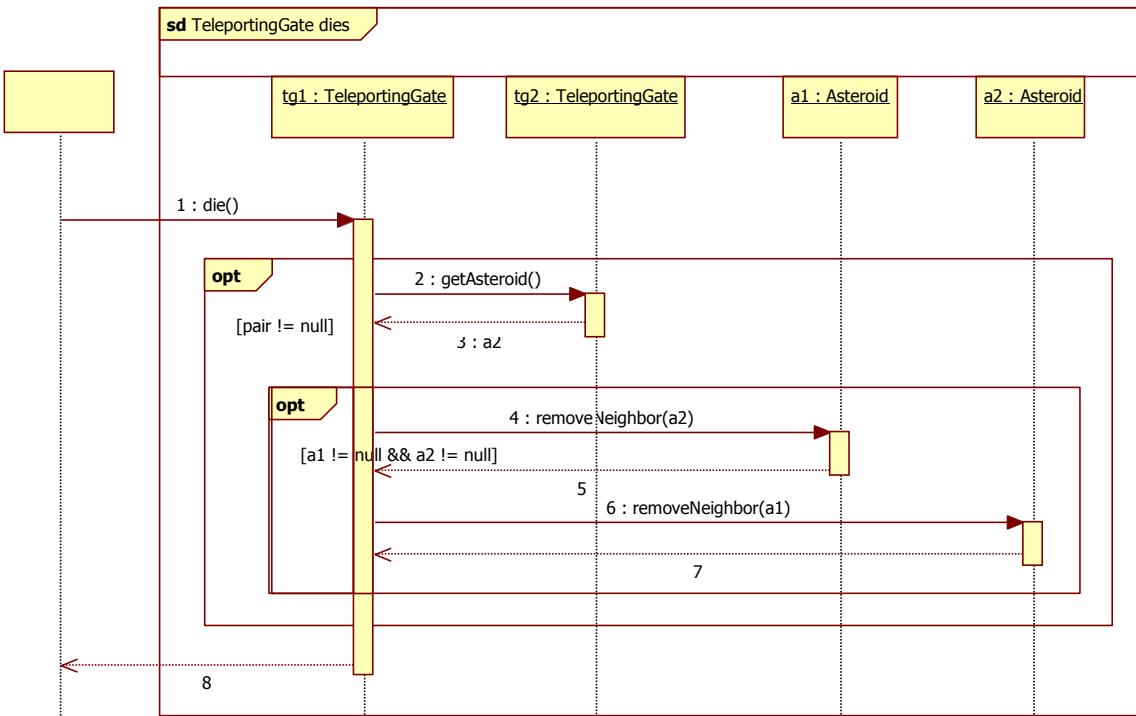
3.4.20 Sun steps



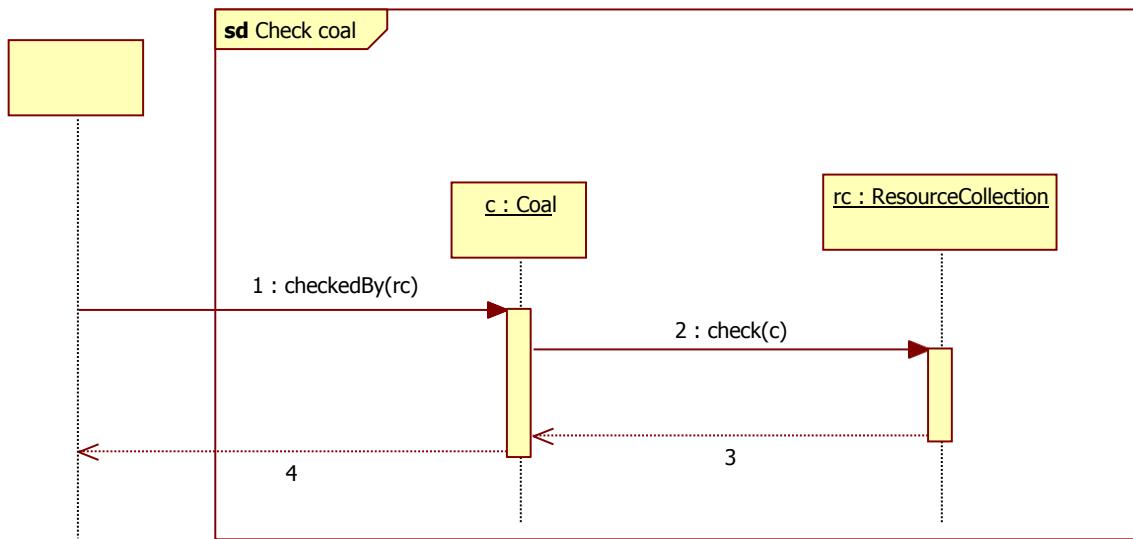
3.4.21 Destroy surface



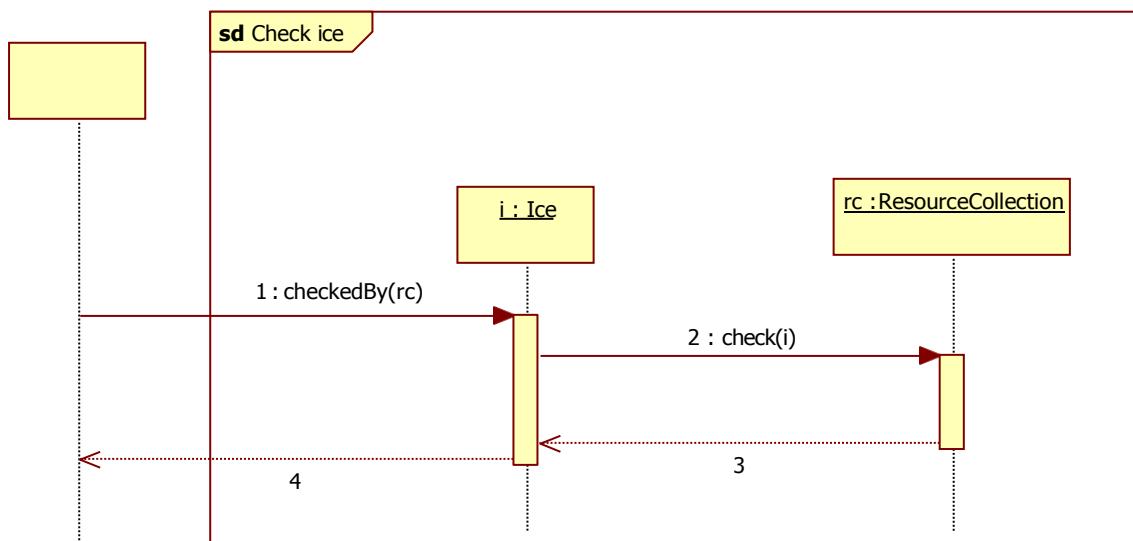
3.4.22 TeleportingGate dies

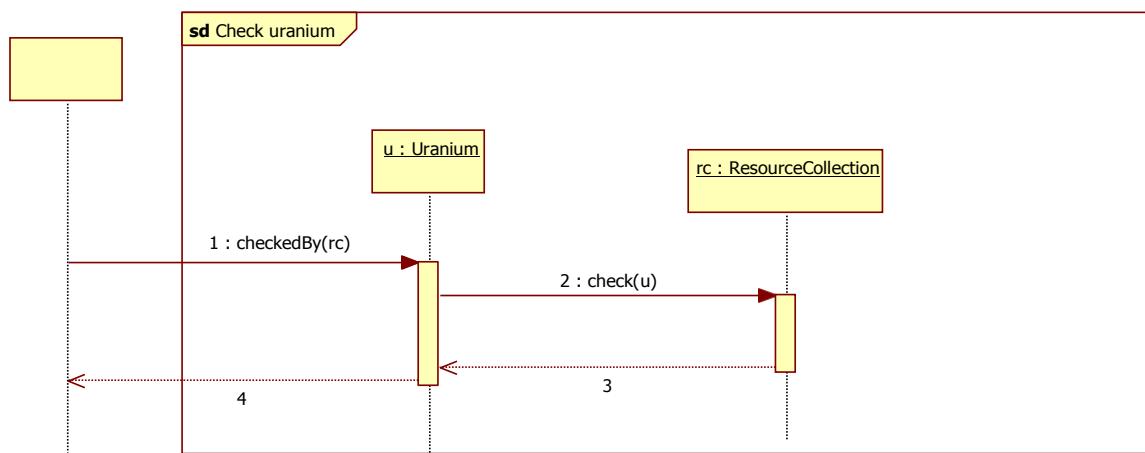


3.4.23 Check Coal



3.4.24 Check Ice



3.4.25 Check Iron**3.4.26 Check Uranium****3.5 State-chartok**

Az analízis modellben nem találtuk szükségesnek állapotgép diagramok rajzolását.

3.6 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2021.02.23. 16:00	1 óra	Radnai	Tevékenység: Az osztálydiagram első verziójának felvázolása.
2021.02.23. 17:00	1,5 óra	Havasi Kreták Radnai Szögi	Értekezlet. Az osztálydiagrammal kapcsolatos kérdések megbeszélése, felelősségek átgondolása, brainstorming. Döntés: A következő értekezlet időpontja: 2021.02.24., konzultáció után
2021.02.24. 10:00	1,5 óra	Havasi Kreták Mesterházi Radnai Szögi	Tevékenység. Az osztálydiagram pontosítása, hiányzó metódusok és osztályok gyűjtése, megbeszélése. Az elkészítendő szekvenciadiagramok összegyűjtése.
2021.02.24. 11:30	0,5 óra	Havasi Kreták Mesterházi Radnai Szögi	Értekezlet. Döntés: Radnai befejezi az osztálydiagramot a megbeszéltek alapján. Radnai és Szögi írja az objektumkatalógust és az osztályleírásokat. Havasi, Kreták és Mesterházi készíti a szekvenciadiagramokat, amelyeket mindenki ellenőriz.
2021.02.24. 17:00	2,5 óra	Radnai Szögi	Tevékenység. Az osztálydiagram javítása Game és Asteroid osztályok leírása.
2021.02.25 10:00	3 óra	Szögi	Tevékenység. Osztályleírások kezdetleges, részleges leírása.
2021.02.26 14:30	1 óra	Radnai Szögi	Tevékenység. Osztálydiagram kiegészítése, osztályleírások frissítése.
2021.02.26 15:00	1,5 óra	Havasi Kreták Mesterházi Radnai	Tevékenység. Osztálydiagram áttekintése, kiegészítése, javítása.
2021.02.26 16:30	3 óra	Havasi Kreták Mesterházi Radnai	Tevékenység. Szekvenciadiagramok rajzolása, kis módosítások az osztálydiagramon.
2021.02.26 20:30	1 óra	Mesterházi	Tevékenység. Szekvenciadiagramok javítása, bővítése.

2021.02.27 15:30	0,5 óra	Kreták	Tevékenység.Szekvenciadiagramok elrendezésének javítása, kisebb javítások.
2021.02.27 17:00	3 óra	Havasi Kreták Mesterházi Radnai Szögi	Tevékenység. A maradék szekvenciadiagramok megrajzolása.
2021.02.28 12:00	1 óra	Szögi	Objektum katalógus megírása, osztályleírások ellenőrzése
2021.02.28 20:00	2 óra	Havasi Kreták Mesterházi Radnai Szögi	Tevékenység. Szekvenciadiagramok véglegesítése, dokumentáció véglegesítése és formázása.

Az adott heti munka csapattagoknál aránya:

Csapattag	Munka aránya százalékban
Havasi	18
Kreták	18
Mesterházi	18
Radnai	25
Szögi	21

4. Analízis modell kidolgozása 2.

7 – berenyi_kft

Konzulens:

Csapattagok

Kreták Barnabás András
Mesterházi Marcell
Szögi Richárd
Havasi Balázs
Radnai Bálint

2021.03.07

4. Analízis modell kidolgozása

4.1 Változások az előző analízis modell óta

A megbeszélések és az értékelés alapján a következő változásokat végeztük el:

- Statechartok készítése - TeleportingGatePair, Asteroid
- Változtatások az aszteroida felelősségein -most már nem az ő felelőssége a nyersanyag tárolása és a character-ek mozgatása
- ResourceCollection helyett OO megoldás – Recipe osztály bevezetése
- A Recipe, Asteroid osztályok miatti változások bevezetése a modellben, osztálydiagramon, szekvenciadiagramokon

4.2 Objektum katalógus

4.2.1 AIRobot

Telepes által egy egységnyi vas, egy egységnyi szén és egy egységnyi urán felhasználásával létrehozható karakter, amely minden időegységben fúrni vagy szomszédos aszteroidára mozogni tud. Nyilvántartja, hogy melyik aszteroidán tartózkodik épp. Az aszteroida robbanása nem semmisíti meg, ekkor szomszédos aszteroidára mozog. A napvihar viszont tönkreteszzi.

4.2.2 Asteroid

Az aszteroidákon telepesek és robotok tartózkodhatnak, valamint teleportkapuk keringhetnek körülöttük. Az aszteroida felelőssége a rajta található telepesek és robotok nyilvántartása, ezek mozgatásának biztosítása, illetve a szomszédos aszteroidák listázása is. Tárolja a köpenye aktuális vastagságát, a magjában található nyersanyagot és a hozzá tartozó teleportkapukat.

Robbanás, illetve napvihar esetén a követelményekben leírtak alapján kezeli az ott található nyersanyagot, karaktereket és teleportkapukat. Az aszteroidák szomszédságok által összefüggő csoportja alkotja a telepesek által bejárható pályát.

4.2.3 Coal

A szén nyersanyagot reprezentálja, lehet aszteroida magjában vagy telepénél tárolva. Robotépítéshez szükséges erőforrás.

4.2.4 Game

A játék eleji inicializálást végzi. Nyilvántartja a játékban levő Napot, az aszteroidákat és a telepeseket.

4.2.5 Ice

A vízjég nyersanyagot reprezentálja, lehet aszteroida magjában vagy telepénél tárolva. Teleporkapu-építéshez szükséges erőforrás. Ha egy napközelben levő megfűrt aszteroida magjában található, akkor elszublimál.

4.2.6 Iron

A vas nyersanyagot reprezentálja, lehet aszteroida magjában vagy telepénél tárolva. Teleporkapu- és robotépítéshez szükséges erőforrás.

4.2.7 Recipe

A játékban felépíthető doleg létrehozásához szükséges nyersanyagok listáját tartalmazza. Az építkezéshez szükséges nyersanyagok meglétének ellenőrzéséhez kell.

4.2.8 Sun

A játékban levő Napot reprezentálja. A napviharok kiindulópontja, valamint a napközelség számításához szükséges. Nyilvántartja a közvetlen közelében található aszteroidákat.

4.2.9 Settler

A játékos által irányított karakter, amely képes fűrésra, aszteroidák közti mozgásra, nyersanyag kibányászására, valamint visszahelyezésére, továbbá AIRobot és teleportkapu-pár építésre, valamint pályára állítására. Egyszerre egy műveletet csinálhat. Meghal robbanás hatására, illetve akkor is, ha nem tud elbújni a napvihar elől. Nyilvántartja a nála található nyersanyagokat és teleportkapukat, valamint azt, hogy melyik aszteroidán tartózkodik épp.

4.2.10 Timer

Egy órajel jellegű időzítőt reprezentál, amely periodikusan léptet minden léptethető objektumot a játékban.

4.2.11 TeleportingGate

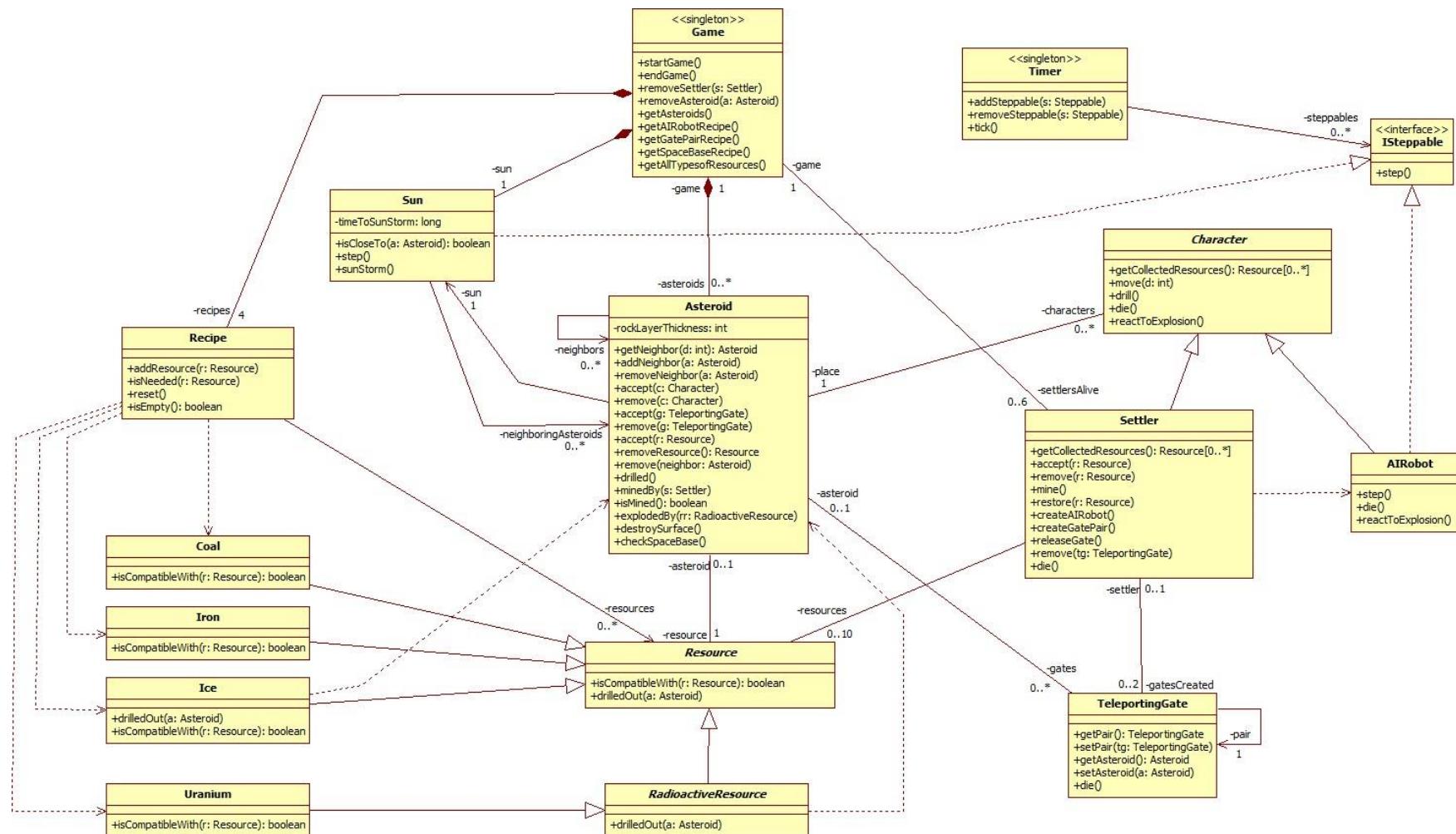
A teleportkapuk olyan tárgyak, amelyek párosával teleportkapu-párokat alkotnak, és aszteroidák körül pályára állíthatók. A frissen épített kapuk az építő telepes űrhajóján tárolódnak, aszteroida körüli pályára állítás után pedig az adott aszteroidákhoz tartoznak. minden teleportkapu eltárolja a párját, valamint pályára állítás előtt a telepest, amelyhez tartozik, pályára állítás után az aszteroidát, amely körül kering.

Ha a teleportkapu-pár minden tagja egy-egy aszteroida körül kering, akkor ezen aszteroidák között extra, ideiglenes szomszédságot biztosítanak. Ha az egyik kapu megsemmisül, a párja vele együtt automatikusan eltűnik, bárhol is van éppen.

4.2.12 Uranium

Az urán nyersanyagot reprezentálja, lehet aszteroida magjában vagy telepesnél tárolva. Teleportkapu- és robotépítéshez szükséges erőforrás. Ha egy napközelben levő megfűrt aszteroida magjában található, akkor felrobbantja aszteroidáját.

4.3 Statikus struktúra diagramok



A nevesített asszociációvégekhez implicit getterek tartoznak, ezek a diagram átláthatósága érdekében nem minden esetben kerülnek ábrázolásra.

4.4 Osztályok leírása

Az attribútumok alapértelmezett láthatósága private, a metódusoké public. Az örökölt metódusokat csak ott tüntetjük fel újból, ahol felüldefiniálásra kerülnek.

4.4.1 AIRobot

- **Felelősség**

A karakterek egyik fajtája a robot, amely minden időegységben fújni vagy szomszédos aszteroidára mozogni tud.

- **Ősosztályok**

Character

- **Interfészek**

ISteppable

- **Attribútumok**

- **Metódusok**

- **void step()**: A robot megkérdezi az aszteroidától, amelyen áll, hogy mekkora a köpenyvastagsága. Ha az aszteroida nincs teljesen átfürva, akkor a robot drill() műveletet végez, különben valamelyik szomszédos aszteroidára lép (move(int d)).
- **void die()**: A robot megsemmisül: meghívja a Timer removeSteppable(Steppable s) metódusát.
- **void reactToExplosion()**: A robot egy véletlenszerű szomszédos aszteroidán landol: sorsol egy d számot, majd meghívja a move (int d) metódust.

4.4.2 Asteroid

- **Felelősség**

Az aszteroidákon karakterek tartózkodhatnak, valamint teleportkapuk keringhetnek körülöttük. Az aszteroidák felelőssége a rajtuk található karakterek mozgatásának biztosítása, és a szomszédos aszteroidák nyilvántartása is. Tárolja a köpenye aktuális vastagságát és a magjában található nyersanyagot. Robbanás, illetve napvihar esetén a követelményekben leírtak alapján kezeli az ott található nyersanyagot, karaktereket és teleportkapukat.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **int rockLayerThickness:** az aszteroida köpenyvastagsága, vagyis a magot borító sziklarétegek száma
- **Game game:** a játékot reprezentáló osztály
- **Sun sun:** az aszteroidaövbén levő Nap
- **Asteroid[0..*] neighbors:** az aszteroidával szomszédos aszteroidák listája; a teleportkapuk által szomszédossá vált aszteroidákat is magába foglalja
- **Resource resource:** az aszteroida magjában található egységnyi nyersanyag; ha a mag üreges, akkor értéke null
- **Character[0..*] characters:** az aszteroidán tartózkodó karakterek (telepesek, robotok, stb.) kollekciója
- **TeleportingGate[0..*] gates:** a közvetlenül az aszteroida körül keringő teleportkapuk halmaza

- **Metódusok**

- **void accept(Character c):** A c karakter megérkezik az aszteroidára, az aszteroida hozzáadja a characters kollekciójához. Ezután ellenőrzi az űrbázis megépíthetőségét: meghívja az checkSpaceBase() függvényt.
- **void remove(Character c):** A c karakter elhagyja az aszteroidát, az aszteroida eltávolítja a characters kollekciójából.
- **void accept(TeleportingGate tg):** A tg teleportkapu pályára áll az aszteroida körül, az aszteroida hozzáadja a gates kollekciójához.
- **void remove(TeleportingGate tg):** A tg teleportkaput eltávolítja az aszteroida körüli pályáról, az aszteroida törli a gates kollekciójából.
- **void accept(Resource r):** Egy az aszteroidán tartózkodó telepes behelyezi az r nyersanyagot az aszteroida üreges magjába, az aszteroida azt beállítja resource attribútumának. Ha a paraméterül kapott r=null, akkor a függvénynek nincs hatása, a resource attribútumot nem írja felül.
- **Resource removeResource():** Egy az aszteroidán tartózkodó telepes eltávolítja a magban található nyersanyagot. Az aszteroida a resource attribútumát null-ra állítja, visszatérési értékül a nyersanyagot adja. Ha kezdetben resource=null volt, a függvénynek nincs mellékhatása, és null-lal tér vissza.
- **void accept(Asteroid neighbor):** Hozzáadja a neighbor aszteroidát az aszteroida neighbors kollekciójához.
- **void remove(Asteroid neighbor):** Eltávolítja a neighbor aszteroidát a neighbors kollekcióból.
- **void drilled():** Az aszteroida rockLayerThickness attribútumát egyelőre csökkenti, amennyiben az pozitív volt. Ha ekkor a köpenyvastagság 0, és a resource attribútuma nem null, akkor a sun objektum isCloseTo(Asteroid a) függvényével lekérdezi, hogy napközelben található-e. Ha igen, akkor meghívja a resource nyersanyag drilledOut() függvényét, jelezve, hogy napközeli aszteroidán felszínre került.
- **void minedBy(Settler) s:** Az s telepes bányászik az aszteroidán. Ha az aszteroida köpenyvastagsága nem 0, akkor a függvénynek nincs hatása. Ha a köpenyvastagság 0, és az aszteroida resource attribútuma nem null, akkor eltávolítja azt a magjából (removeResource()), és eltároltatja azt az s telepessel (s.accept(resource)).
- **boolean isMined():** Visszatér annak logikai értékével, hogy az aszteroida megfűrt és üreges, vagyis rockLayerThickness=0 és resource=null.

- **void explodedBy(RadioactiveResource rr)**: A megfúrt, napközelben lévő, radioaktív nyersanyagot tartalmazó aszteroida felrobban. minden rajta tartózkodó karakternek meghívja a reactToExplosion() függvényét. Az aszteroida szomszédain a remove(Asteroid neighbor) függvényt hívja, amivel törli magát a szomszédai szomszédsági listáiból, majd megsemmisíti a körülötte keringő teleportkapukat a párjaikkal együtt. Végül eltávolítja magát a játékból a Game osztály removeAsteroid() függvényét hívva.
- **void destroySurface()**: Az aszteroidán végigsöpör a napvihar. Ha az aszteroida nincs megfúrva (rockLayerThickness > 0), vagy a mag nem üreges (getResource() ≠ null), akkor a függvény meghívja az aszteroidán tartózkodó karakterek die() függvényét.
- **void checkspaceBase()**: Ellenőrzi, hogy az adott aszteroidán lévő telepeseknél rendelkezésre áll-e az űrbázis felépítéséhez szükséges nyersanyagmennyiség. Ha igen, akkor meghívja a Game endGame() metódusát.
- **Resource getResource()**: Visszaadja az adott aszteroida magjában található nyersanyagot.
- **Sun getSun()**: Visszaadja a játékban levő Napot.
- **Character[0..*] getCharacters()**: Visszaadja az adott aszteroidán tartózkodó karakterek kollekcióját.
- **Asteroid[0..*] getNeighbors()**: Visszaadja az adott aszteroidával szomszédos aszteroidák kollekcióját, beleértve ebbe az aszteroida közvetlen szomszédait (neighbors) és a teleportkapuk (gates) általi szomszédokat is.
- **Asteroid getNeighbor(int d)**: Megadja az aszteroida d-edik szomszédját (az aszteroidához tartozó teleportkapuk általi szomszédai figyelembevételével).
- **TeleportingGate getGate()**: Visszaadja az adott aszteroidához tartozó teleportkapukat.

4.4.3 Character

- **Felelősség**

Ez az osztály absztrakt ōsosztályként szolgál a játék karaktereihez. A karakterek aszteroidákon tartózkodhatnak, köztük mozoghatnak, sziklaköpenyüket képesek fújni. Különbözőképpen reagálhatnak aszteroidák robbanására.

- **Ósosztályok**

- **Interfészek**

- **Attribútumok**

- **Asteroid place**: az aszterioda, amelyen az adott karakter aktuálisan tartózkodik

- **Metódusok**

- **Resource[0..*] getCollectedResources()**: Visszatér a karakter által tárolt nyersanyagok listájával, alapértelmezetten egy üres listával.
- **void move(int d)**: Átmozog az aktuális aszteroidáról annak egy szomszédjára. Először lekérdezi az aktuális aszteroida d-edik szomszédját (a getNeighbor(int d) függvényel). Ezután eltávolítja magát a jelenlegi aszteroidáról (remove(Character c)), majd a visszakapott szomszéd aszteroidára lép (accept(Character c)).

- **void drill()**: Csökkenti az aktuális aszteroida köpenyvastagságát: meghívja az aszteroida drilled() metódusát.
- **void die()**: A karakter halálát, jelenti: eltávolítja a karaktert az aszteroidájáról (place.remove(Character c)). A leszármazottakban felüldefiniálható.
- **void reactToExplosion()**: A metódus, amit a karakterek akkor kell végrehajtania, ha az aszteroida, amelyen tartózkodik, felrobban. Alapértelmezetten a karakter die() függvényét hívja. A leszármazottak felüldefiniálhatják.

4.4.4 Coal

- **Felelősség**

A szén nyersanyagot reprezentálja.

- **Ősosztályok**

Resource

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **boolean isCompatibleWith(r: Resource)**: Összehasonlítja magát a paraméterként kapott nyersanyaggal. Egyezés esetén igaz értékkel tér vissza, ellenkező esetben hamissal.

4.4.5 Game

- **Felelősség**

A játék indításáért és befejezéséért felel, a játékban lévő telepesek, valamint a felépíthető dolgok receptjeinek nyilvántartását végzi.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **Asteroid[0..*] asteroids**: a játékban jelenlévő aszteroidák kollekciója
- **Sun sun**: az aszteroidaövbeli Nap
- **Settler[0..*] settlersAlive**: az adott pillanatban életben levő telepesek listája
- **Recipe[3] recipes**: A játékban felépíthető dolgok receptjeinek a kollekciója.

- **Metódusok**

- **void startGame()**: Inicializálja az aszteroidákat, a Napot, és elhelyezi a telepeseket valamelyik aszteroidán. Egyes aszteroidák magját üresen hagyja, másokban pedig véletlenszerűen valamilyen nyersanyagot hoz létre. Beállítja az aszteroidák szomszédsági viszonyait is.

- **void endGame()**: Ha van még élő telepes a játékban (settlersAlive nem üres), akkor a telepesek győzelmét írja ki a képernyőre, ellenkező esetben a telepesek vereségéről szóló üzenetet jelenít meg.
- **void removeSettler(Settler s)**: Törli a meghaló s telepest a játékból. Ha nincs több életben levő telepes, akkor meghívja az endGame() függvényt.
- **void removeAsteroid(Asteroid a)**: Törli a megsemmisülő a aszteroidát az asteroids kollekcióból.
- **Asteroid[0..*] getAsteroids()**: Visszaadja a játék összes aszteroidájából álló kollekciót.
- **Recipe getGatePairRecipe()**: Visszaadja a teleportkapu-pár felépítéséhez szükséges receptet tároló Recipe objektumot.
- **Recipe getAIRobotRecipe()**: Visszaadja az AIRobot felépítéséhez szükséges receptet tároló Recipe objektumot.
- **Recipe getSpaceBaseRecipe()**: Visszaadja az űrbázis felépítéséhez szükséges receptet tároló Recipe objektumot.

4.4.6 Ice

- **Felelősség**

A vízjég nyersanyagot reprezentálja. Ha napközeli, teljesen megfűrt aszteroida magjában található, akkor elszublimál.

- **Ősosztályok**

Resource

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **boolean isCompatibleWith(r: Resource)**: Összehasonlítja magát a paraméterként kapott nyersanyaggal. Egyezés esetén igaz értékkel tér vissza, ellenkező esetben hamissal.
- **void drilledOut()**: Hatására a vízjég elszublimál: meghívja az asteroid.removeResource() metódust a saját aszteroidáján.

4.4.7 Iron

- **Felelősség**

A vas nyersanyagot reprezentálja.

- **Ősosztályok**

Resource

- **Interfészek**
- **Attribútumok**
- **Metódusok**
 - **boolean isCompatibleWith(r: Resource):** Összehasonlítja magát a paraméterként kapott nyersanyaggal. Egyezés esetén igaz értékkel tér vissza, ellenkező esetben hamissal.

4.4.8 RadioactiveResource

- **Felelősség**
A radioaktív nyersanyagtípust reprezentálja, radioaktív nyersanyagok ōsosztályaként szolgál. Ha napközelben lévő, megfűrt aszteroida tartalmazza, akkor felrobbantja az aszteroidáját.

- **Ősosztályok**

Resource

- **Interfészek**
- **Attribútumok**
- **Metódusok**
 - **void drilledOut():** Meghívja az aszteroidája explodedBy(RadioactiveResource rr) metódusát (robban).

4.4.9 Resource

- **Felelősség**
A játékban levő nyersanyagok ōsosztálya. Többféle nyersanyag létezik, amelyek lehetnek aszteroida magjában vagy telepénél tárolva. Ha egy nyersanyag napközelben levő megfűrt aszteroida magjában található, akkor az aszteroidán - a konkrét fajtájától függően - speciális hatást válthat ki.

- **Ősosztályok**
- **Interfészek**
- **Attribútumok**
 - **Settler settler:** a telepes, akinél az adott nyersanyagegység tárolódik (ha éppen telepénél található)
 - **Asteroid asteroid:** az aszteroida, amelynek magjában az adott nyersanyag elhelyezkedik (ha éppen aszteroida magjában található)
- **Metódusok**
 - **void drilledOut():** Azt a helyzetet kezeli, amikor a nyersanyag éppen egy napközelben levő, megfűrt aszteroida magjában található. A metódus általános esetben

nem csinál semmit, de a speciális működésű leszármazott osztályokban felüldefiniálható.

- **abstract boolean isCompatibleWith(r: Resource):** Összehasonlítja magát a paraméterként kapott nyersanyaggal és egyezés esetén igaz értékkel tér vissza, ellenkező esetben hamissal.

4.4.10 Recipe

- **Felelősség**

A játékban felépíthető dolgok létrehozásához szükséges nyersanyagok listáját tartalmazza. A telepeseknél lévő nyersanyagokat összehasonlítja a listájában lévőkkel. Különböző példányai különböző dolgok megépítéséhez szükséges nyersanyagokat tartalmazzák.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **Resource[0..*] resources:** A szükséges nyersanyagok kollekciója.

- **Metódusok**

- **void addResource(r: Resource):** A paraméterként kapott nyersanyagot hozzáadja a resources kollekcióhoz. A recept módosításához szükséges, ha változnának a feltételek.
- **void isNeeded (r: Resource):** A paraméterként kapott r nyersanyagot összehasonlítja a resources kollekcióban lévő nyersanyagokkal, az isCompatibleWith(r: Resource) metódussal. Amennyiben az isCompatibleWith(r: Resource) függvény igaz értékkel tér vissza valamelyik kollekcióból nyersanyagra, törli az adott nyersanyagot a resources kollekcióból.
- **void reset():** Visszaállítja az eredeti receptet, azaz a resources gyűjtemény tartalmát.
- **boolean isEmpty():** Ha a resources kollekció üres, igaz értékkel tér vissza, ellenkező esetben pedig hamissal.

4.4.11 Settler

- **Felelősség**

A karakterek egyik fajtája a telepes, amely a Character-ben megfogalmazottakon túl képes egységenyi nyersanyagot kibányászni vagy visszahelyezni az aszteroidamagba. Emellett építhet robotot és teleportkapu-párt, illetve az elkészült teleportkapukat pályára állíthatja. Nyilvántartja a nála található nyersanyagokat és teleportkapukat.

- **Ősosztályok**

Character

- **Interfészek**

- **Attribútumok**

- **Game game:** a játékot reprezentáló osztály

- **Resource[0..10] resources:** a telepes által tárolt nyersanyagok kollekciója; a kollekció kapacitása 10
- **TeleportingGate[0..2] gatesCreated:** a telepes által tárolt teleportkapuk kollekciója
- **Metódusok**
 - **Resource[0..*] getCollectedResources():** Visszatér a telepes által összegyűjtött nyersanyagok kollekciójával.
 - **void accept(Resource r):** A telepes eltárolja a kibányászott r nyersanyagot a resources kollekciójában.
 - **void remove(Resource r):** A telepes eltávolítja az r nyersanyagot a resources kollekciójából.
 - **void remove(TeleportingGate: tg):** A telepes eltávolítja a tg teleportkaput a gatesCreated kollekciójából.
 - **void mine():** A telepes kibányássza az adott megfűrt aszteroida magjában található nyersanyagot. Ehhez meghívja az aszteroida minedBy() metódusát. Ha a bányászat kezdetén az űrhajóban már nincs hely újabb nyersanyagnak, akkor a telepes nem tud bányászni, a függvénynek nincs hatása.
 - **void restore(Resource r):** Meghívja a place aszteroida Accept(Resource r) metódusát. Ha az aszteroida üres, akkor az aszteroida eltárolja az r nyersanyagot, és eltávolítja azt a Settler nyersanyagai közül. Ha az aszteroida magjában már volt nyersanyag, akkor nem történik semmi.
 - **void createAIRobot():** A telepes lekérdezi a Game-től az AI robot megépítéséhez szükséges receptet a Recipe getAIRobotRecipe() metódusának meghívásával, majd pedig összehasonlítja a saját nyersanyagait a receptben lévőkkel. Amennyiben rendelkezik a szükséges nyersanyagokkal, létrehoz egy AIRobot példányt, ezután pedig hozzáadja az aktuális aszteroidához az accept(c: Character) metódus meghívásával. Végül meghívja a Recipe reset() függvényét, amelyben ezáltal visszaállítja a recept listájának a tartalmát.
 - **void createGatePair():** A telepes lekérdezi a Game-től a teleportkapu-pár megépítéséhez szükséges receptet a Recipe getGatePairRecipe() metódusának meghívásával, majd pedig összehasonlítja a saját nyersanyagait a receptben lévőkkel. Amennyiben rendelkezik a szükséges nyersanyagokkal és nincs nála elkészített teleportkapu, akkor létrehozza a TeleportingGate két példányát és párba állítja őket a setPair(tg: TeleportingGate) függvények meghívásával, majd az űrhajóján a gatesCreated kollekcióból eltárolja őket. Végül meghívja a Recipe reset() függvényét, amelyben ezáltal visszaállítja a recept listájának a tartalmát.
 - **void releaseGate():** A telepes a gatesCreated kollekcióból kiválaszt egy teleportkaput, majd az aktuális aszteroidája körüli pályára állítja az Asteroid accept(TeleportingGate tg) függvényével. Ha a telepesnél nincs elkészült teleportkapu (gatesCreated üres), akkor nem történik semmi.
 - **void die():** A telepes meghal: megsemmisíti a nála levő teleportkapukat azok die() függvényeivel, eltávolítja magát az aszteroidájáról, végül meghívja a game removeSettler(Settler s) metódusát.

4.4.12 ISteppable

- **Felelősség**

Olyan objektumot jelöl, amely a játékban külső irányítás (játékos aktor beavatkozása) nélkül léptethető.

- **Ősosztályok**

- **Metódusok**

- **void step()**: A művelet, amelyet az objektum az adott lépésben végez.

4.4.13 Sun

- **Felelősség**

A játékban levő Napot reprezentálja. A napviharok kiindulópontja, valamint a napközelség számításához szükséges.

- **Ősosztályok**

- **Interfészek**

ISteppable

- **Attribútumok**

- **long timeToSunStorm**: a következő napvihar bekövetkezéséig hátralévő idő
- **Asteroid[0..*] neighboringAsteroids**: a Nappal közvetlenül szomszédos aszteroidák listája
- **Game game**: a játékot reprezentáló osztály

- **Metódusok**

- **void step()**: Ha a timeToSunStorm attribútum pozitív, akkor eggyel csökkenti (várakozik). Ha pedig 0, akkor napvihart indít a sunStorm() függvényt hívva, majd a timeToSunStorm-nak beállít egy véletlen egész értéket.
- **void sunStorm()**: Lekérdezi a Game-től az aszteroidák kollekcióját, majd egyesével meghívja mindegyik aszteroida destroySurface() függvényét.
- **boolean isCloseTo(Asteroid a)**: Megállapítja, hogy az a aszteroida napközeli-e, vagyis legfeljebb másodszomszédos-e a Nappal. Ehhez felhasználja a neighboringAsteroids listáját: megvizsgálja a benne szereplő aszteroidákat, valamint azok összes szomszédait is.

4.4.14 TeleportingGate

- **Felelősség**

Két teleportkapu egy teleportkapu-párt alkot, amely aszteroidák közötti összeköttetést biztosít. A frissen épített kapuk az építő telepes űrhajóján tárolódnak. Aszteroida körüli pályára állítás után az adott aszteroidához tartoznak. Ha a kapupár minden tagját aszteroida körüli pályára állították, akkor az aszteroidák, amelyekhez tartoznak, szomszédosnak számítanak. Ha az egyik kapu megsemmisül, a párja vele együtt automatikusan eltűnik, bárhol is van éppen.

- **Ősosztályok**
- **Interfészek**
- **Attribútumok**
 - **TeleportingGate pair**: az adott teleportkapu párja, amellyel összeköttetésben áll
 - Asteroid asteroid**: az aszteroida, amely körül az adott teleportkapu kering
 - **Settler settler**: a telepes, aki tárolja a létrehozott, de még pályára nem állított teleportkaput
- **Metódusok**
 - **TeleportingGate getPair()**: Visszaadja a teleportkapu pair párját.
 - **void setPair(TeleportingGate tg)**: Beállítja tg-t a teleportkapu párokaként.
 - **Asteroid getAsteroid()**: Visszaadja az aszteroidát, amely körül a teleportkapu kering. Ha a kaput még nem állították pályára, akkor null-lal tér vissza.
 - **void setAsteroid(Asteroid a)**: Beállítja a teleportkapu aszteroidáját, amely körül keringeni fog.
 - **void die()**: Eltávolítja a párt (pair) annak aszteroidájáról/űrhajójáról attól függően, hogy azt már pályára állították-e (remove(TeleportingGate tg)). Beállítja a pártának a pártat null-ra (megszünteti az összeköttetést), végül saját magát is eltávolítja az aszteroidájáról/űrhajójáról. Ha a függvényhívás elején a teleportkapu pára már null, akkor csak magát távolítja el: ekkor a másik teleportkapu semmisült meg előbb, és az már megszüntette ennek a kapunak a kapcsolatait is.

4.4.15 Timer

- **Felelősség**
Egy órajel jellegű időzítőt reprezentál, amely periodikusan léptet minden léptethető objektumot a játékban.
- **Ősosztályok**
- **Interfészek**
- **Attribútumok**
 - **ISteppable[0..*] steppables**: a játék léptethető objektumainak listája
- **Metódusok**
 - **void addStepable(Steppable s)**: Hozzáadja s-t a steppables listához.
 - **void removeStepable(Steppable s)**: Eltávolítja s-t a steppables listából.
 - **tick**: minden steppable listabeli objektumot léptet úgy, hogy meghívja a step() függvényét.

4.4.16 Uranium

- **Felelősség**

Az urán nyersanyagot reprezentálja.

- **Ősosztályok**

RadioactiveResource → Resource

- **Interfészek**

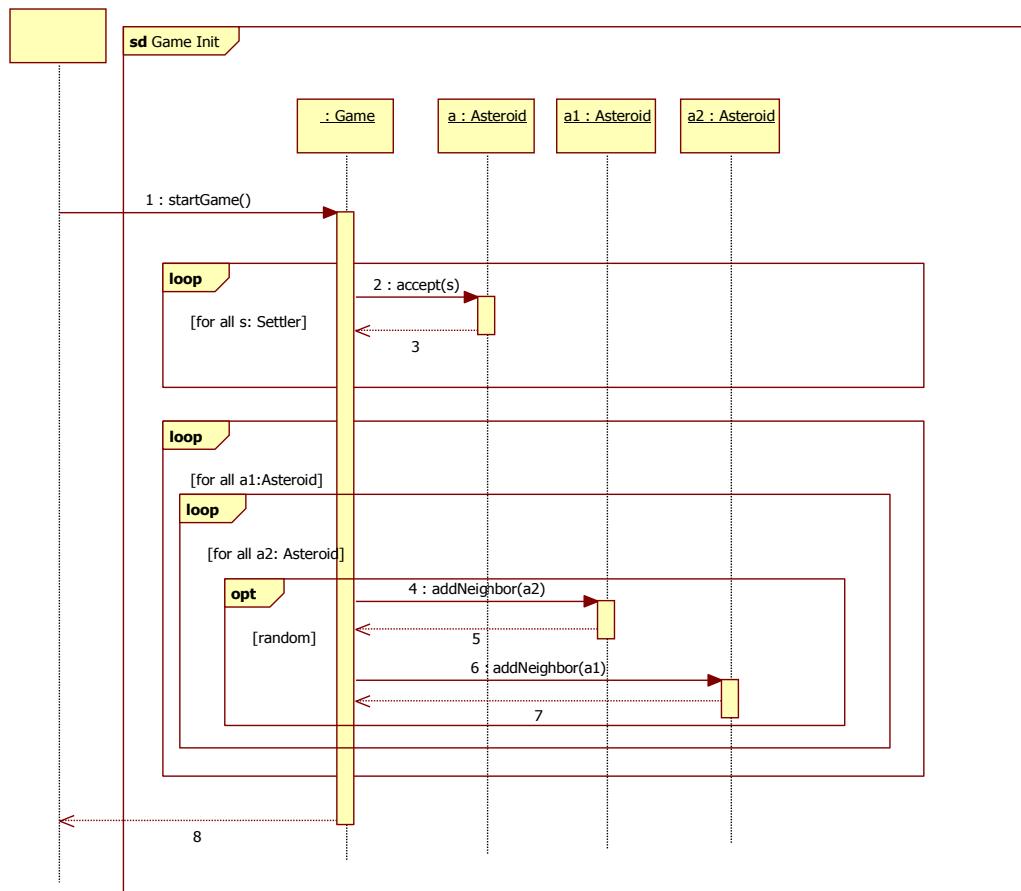
- **Attribútumok**

- **Metódusok**

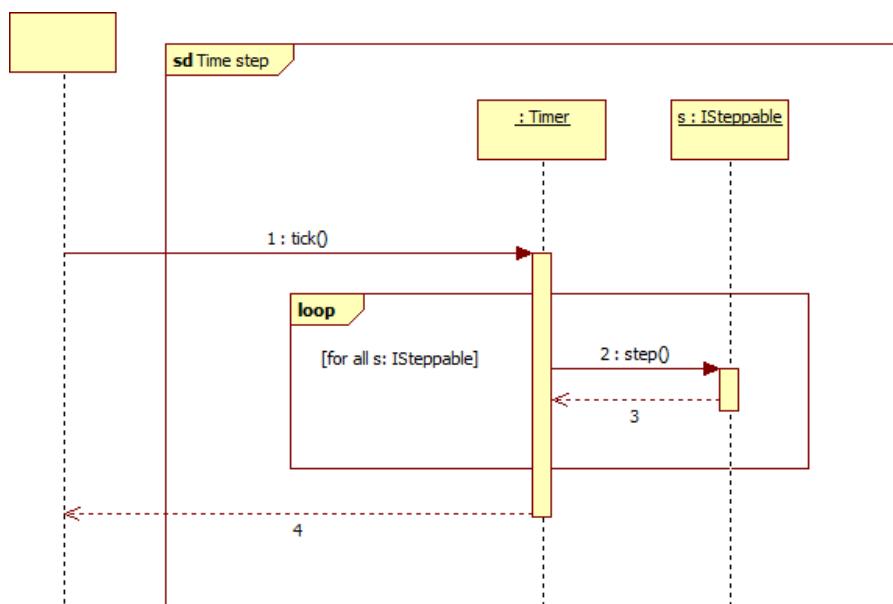
- **boolean isCompatibleWith(r: Resource):** Összehasonlítja magát a paraméterként kapott nyersanyaggal. Egyezés esetén igaz értékkel tér vissza, ellenkező esetben hamissal.

4.5 Szekvencia diagramok

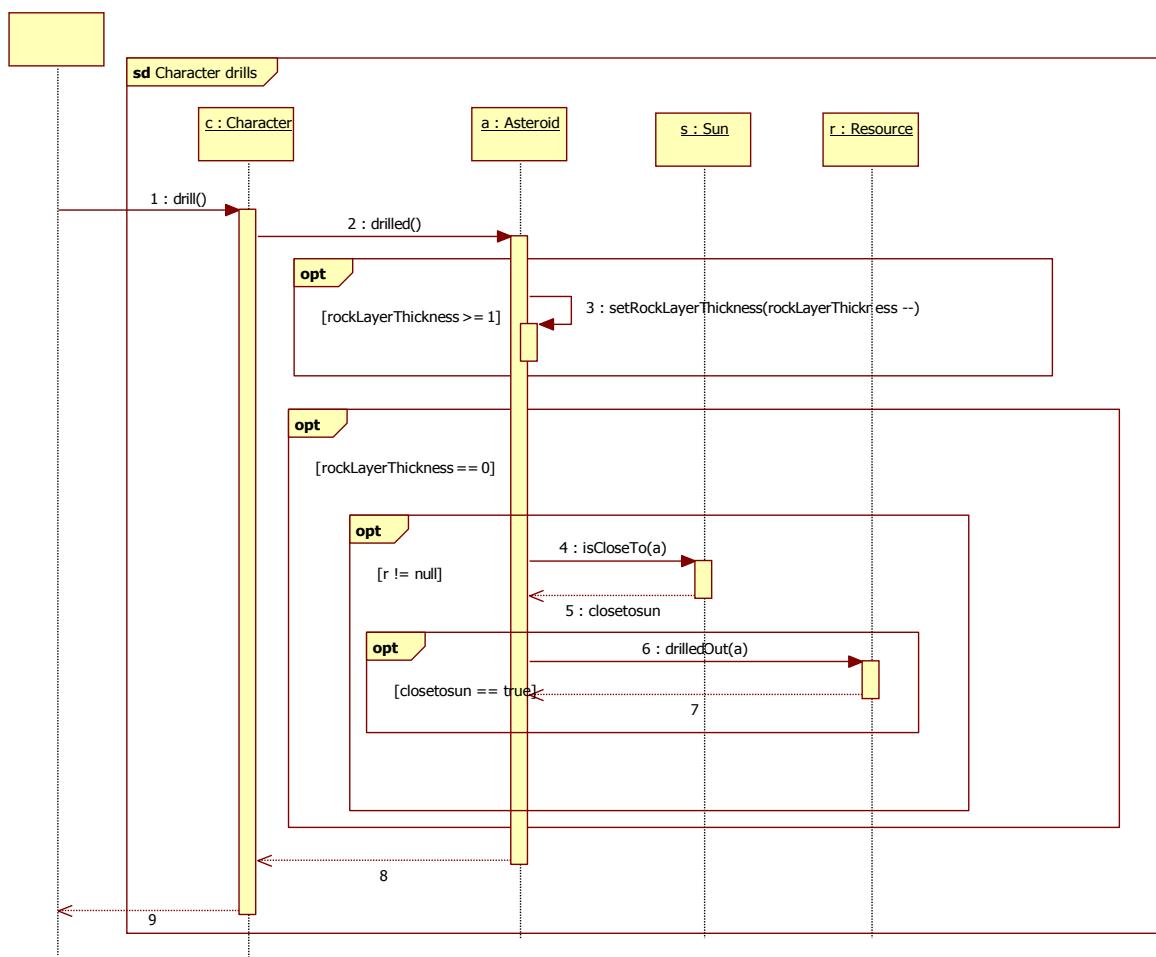
4.5.1 Game init



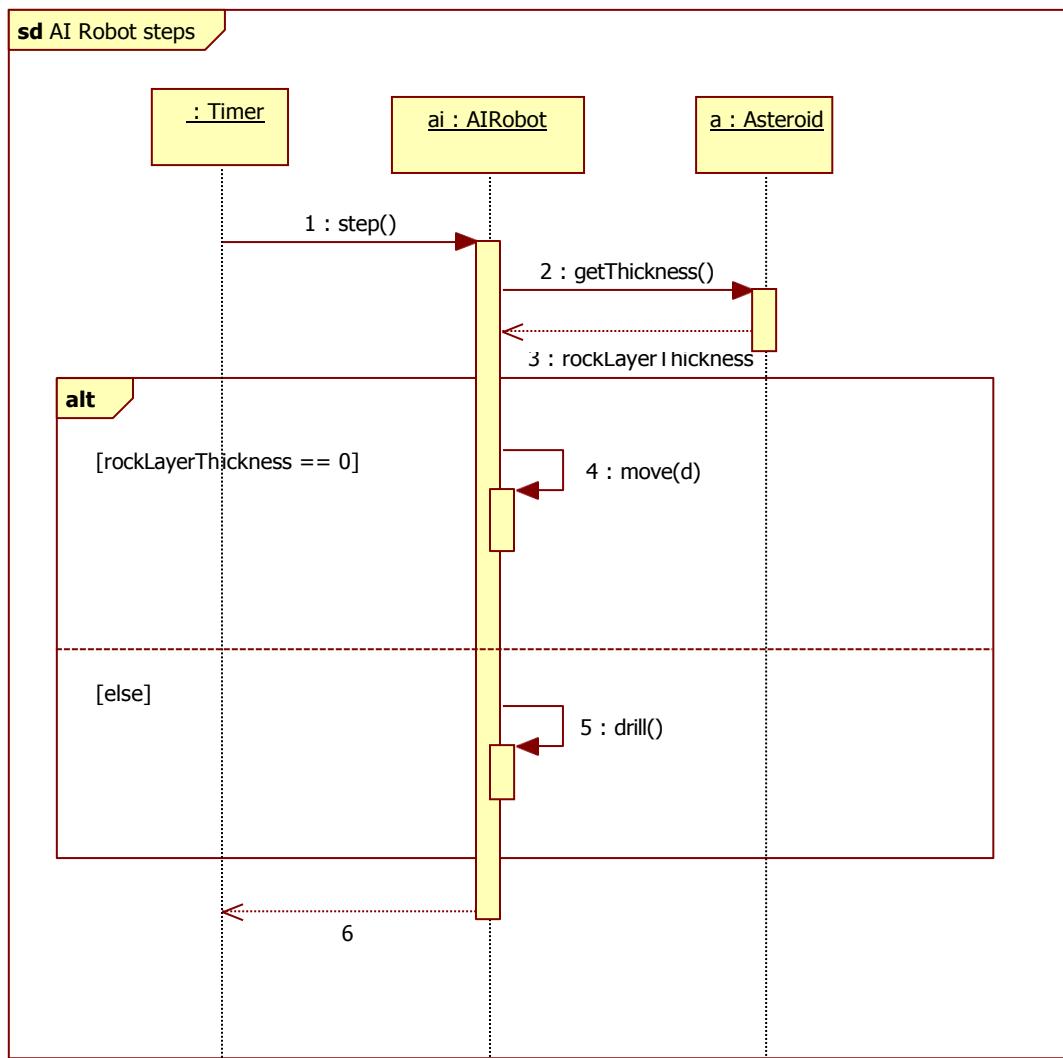
4.5.2 Time step



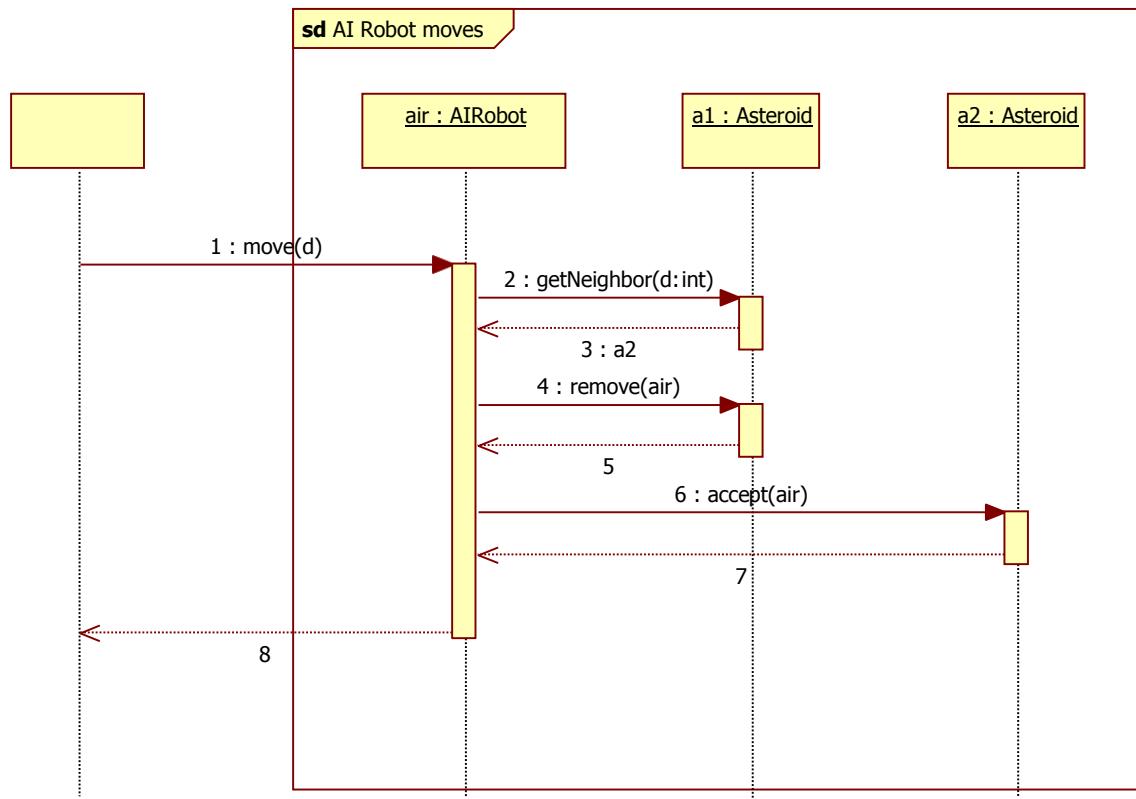
4.5.3 Character drills



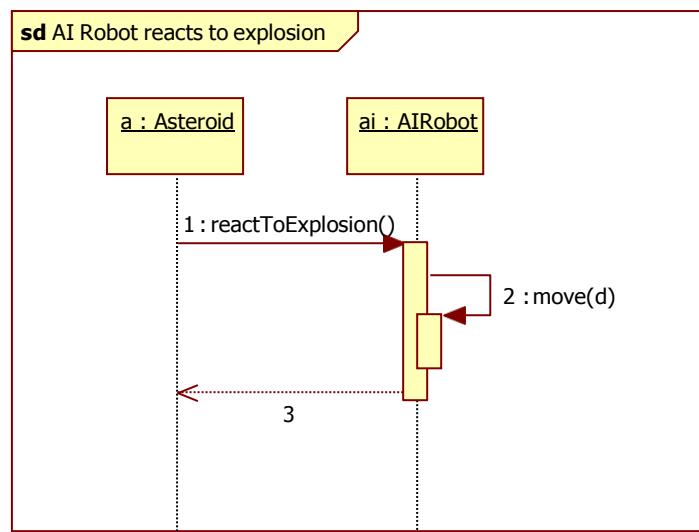
4.5.4 AI Robot steps



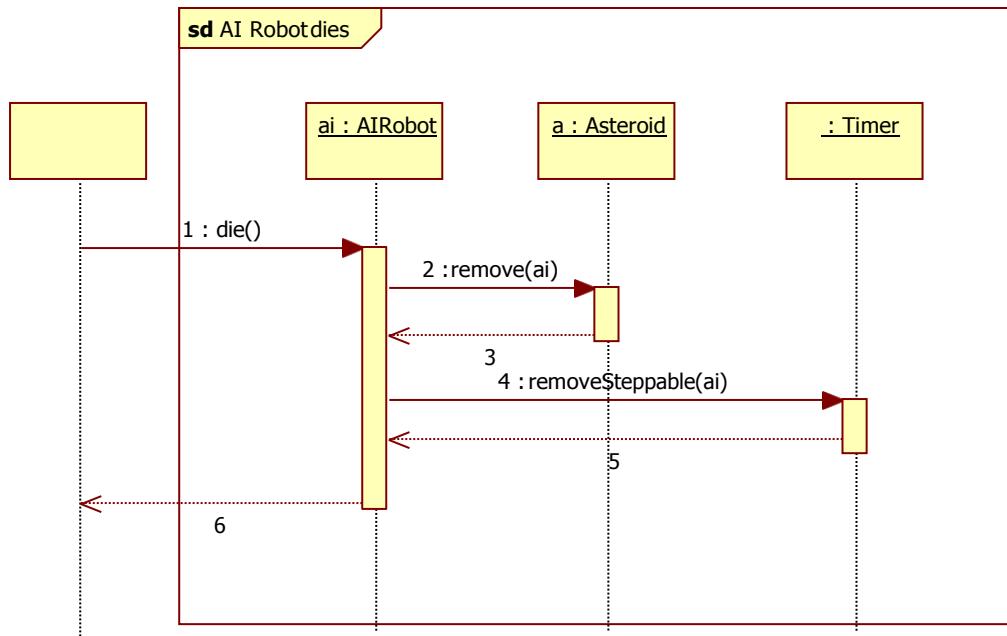
4.5.5 AI Robot moves



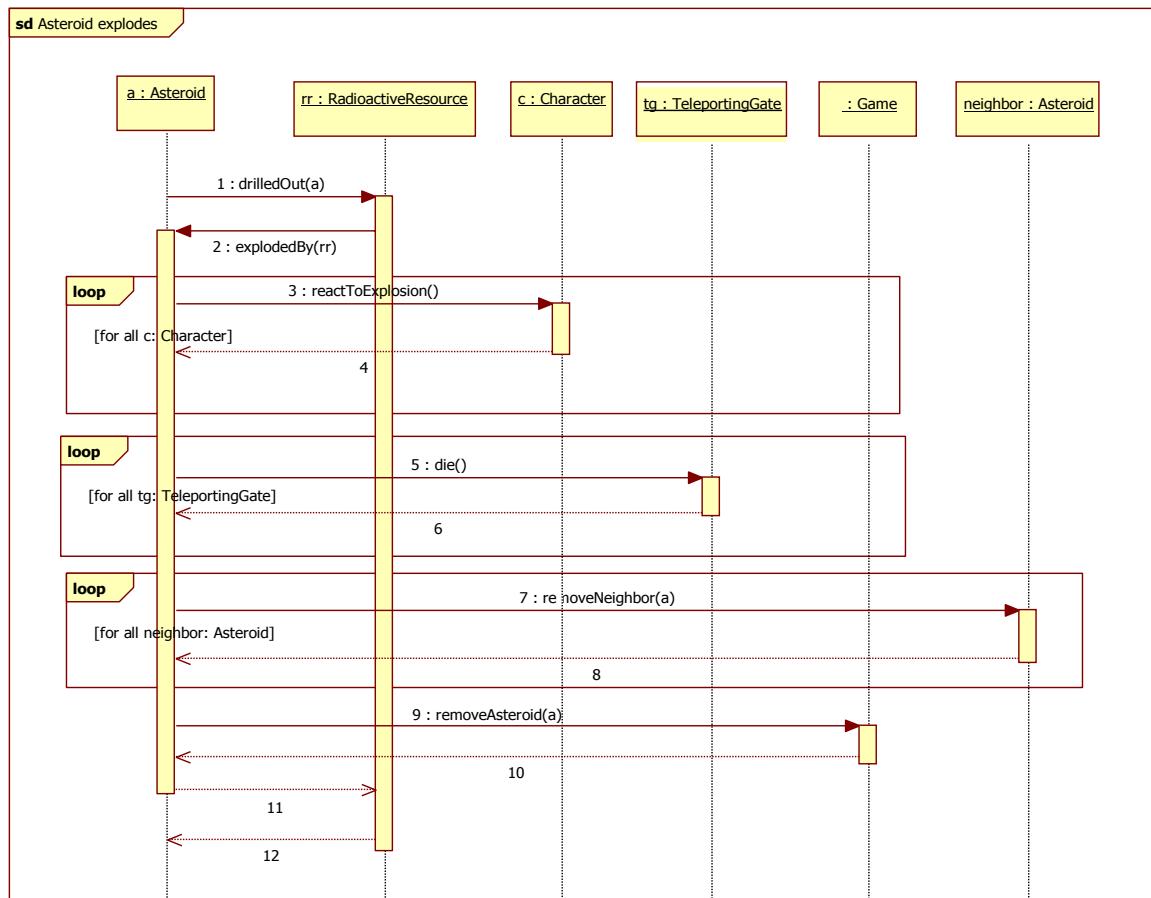
4.5.6 AI Robot reacts to explosion



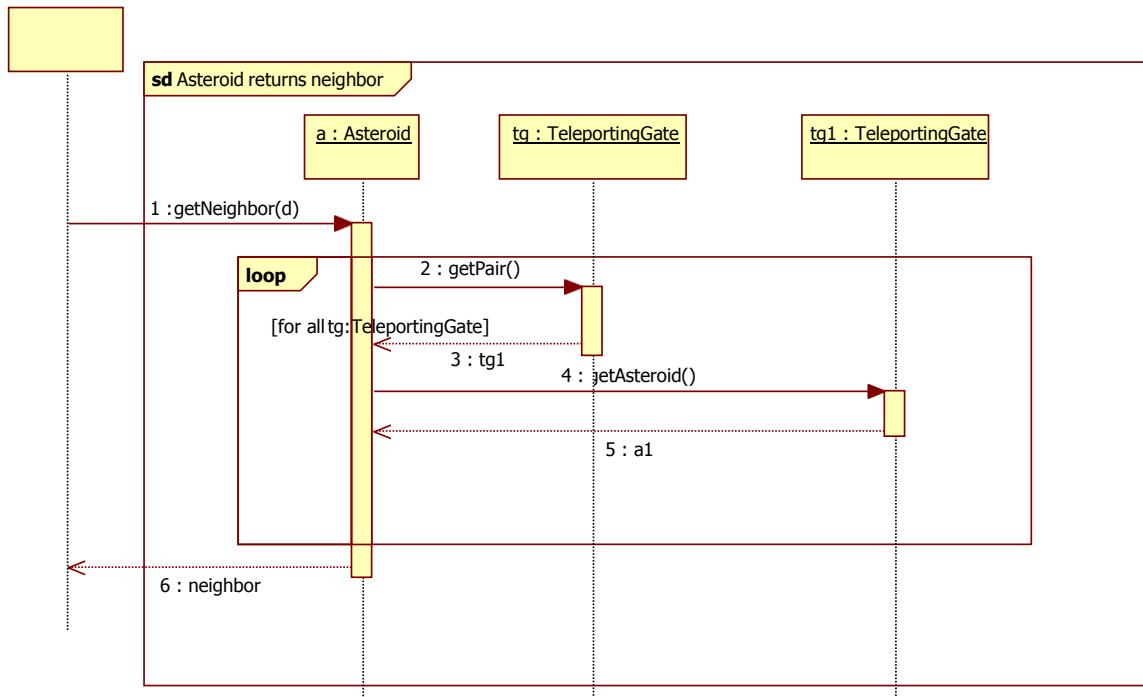
4.5.7 AI Robot dies



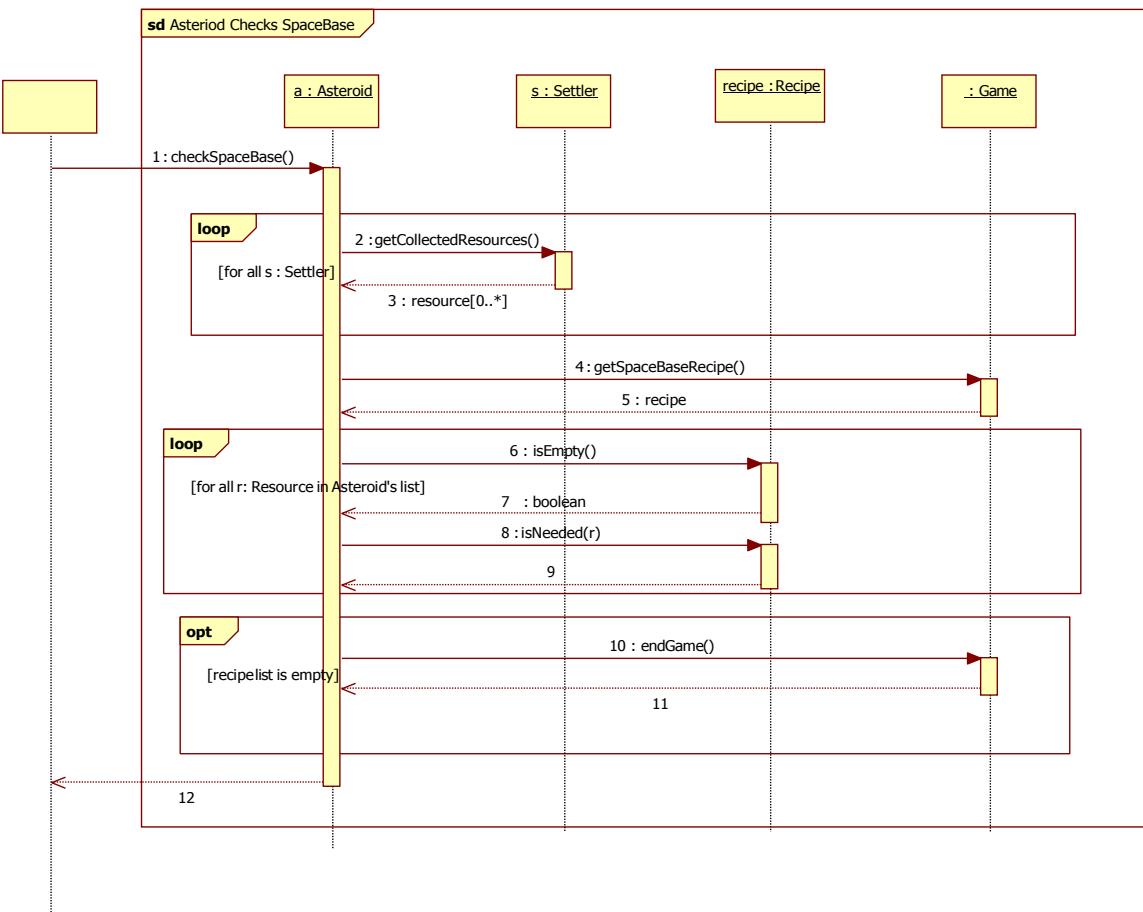
4.5.8 Asteroid explodes



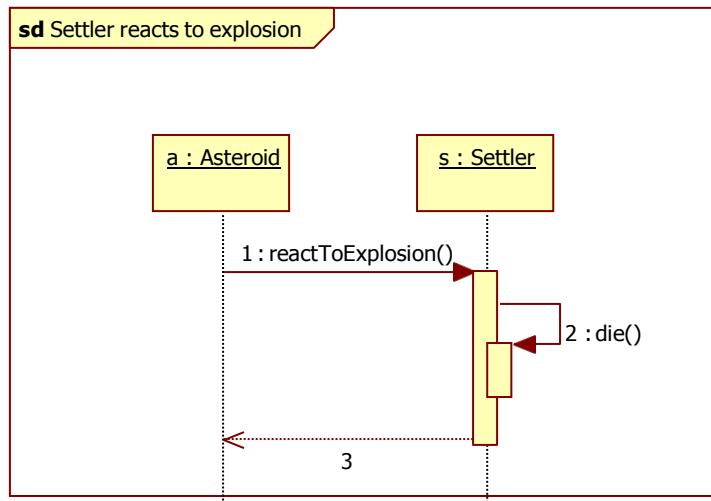
4.5.9 Asteroid returns neighbor



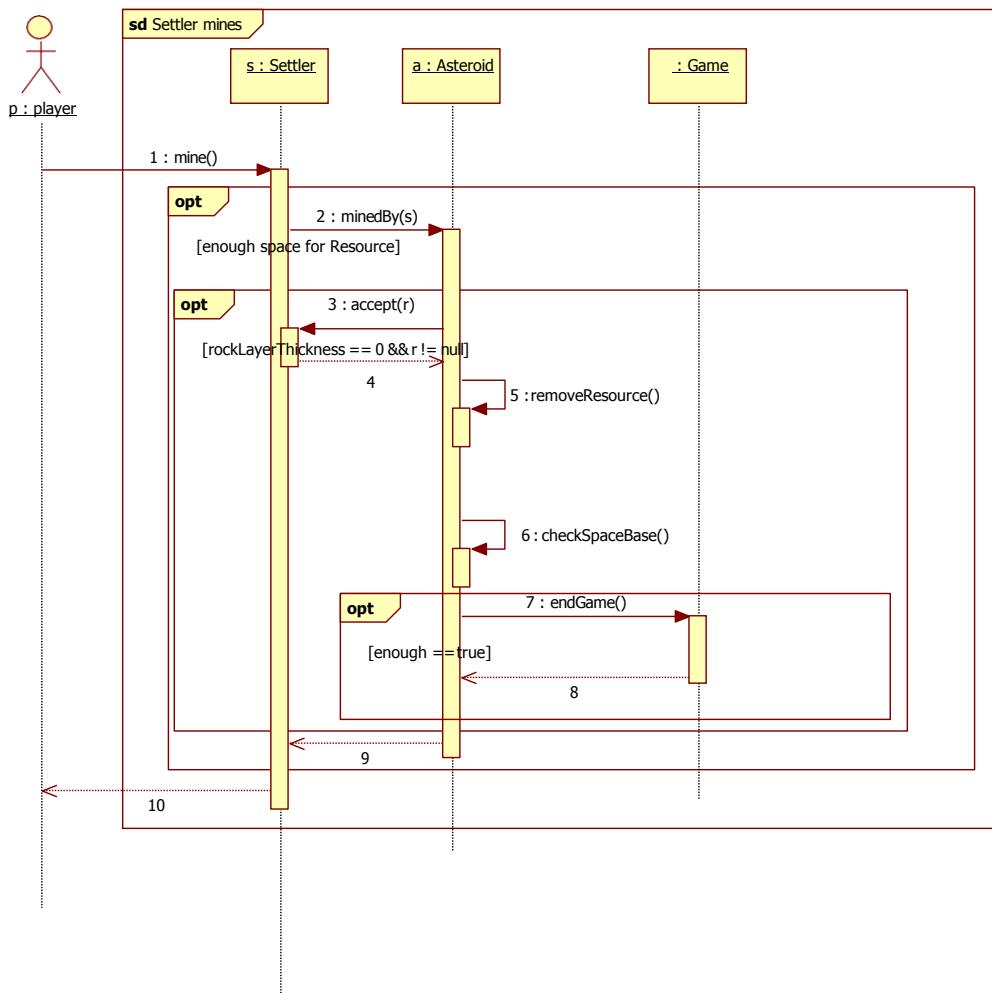
4.5.10 Asteroid checks Spacebase



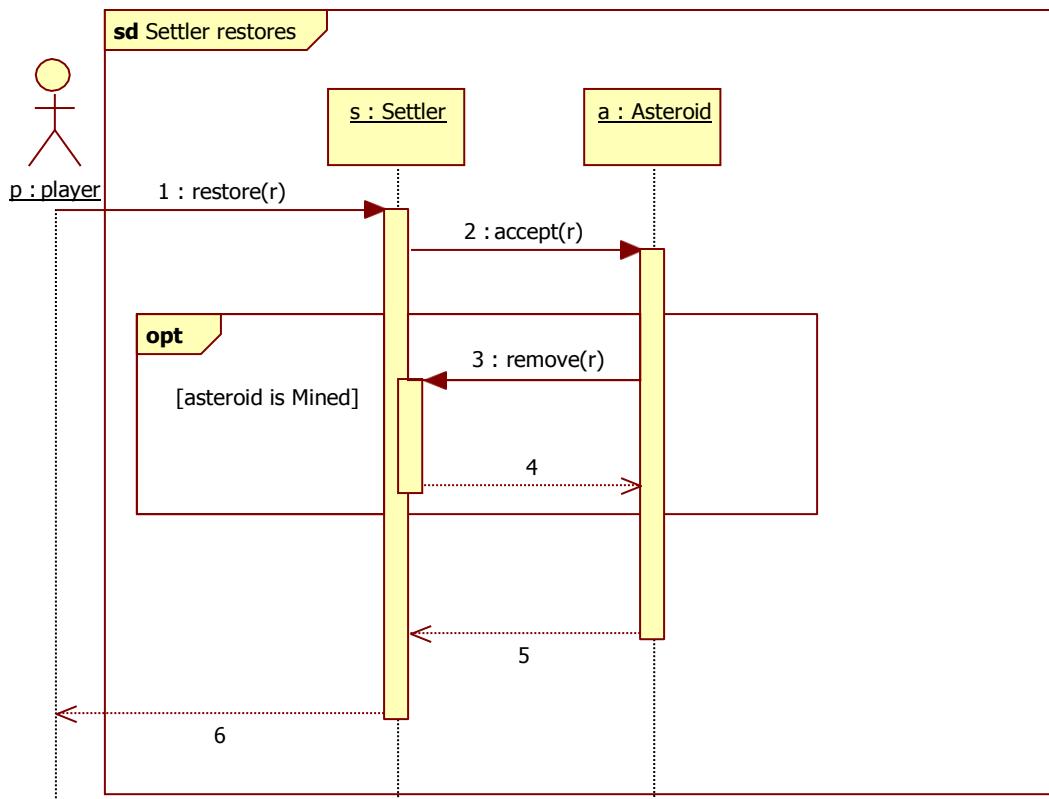
4.5.11 Settler reacts to explosion



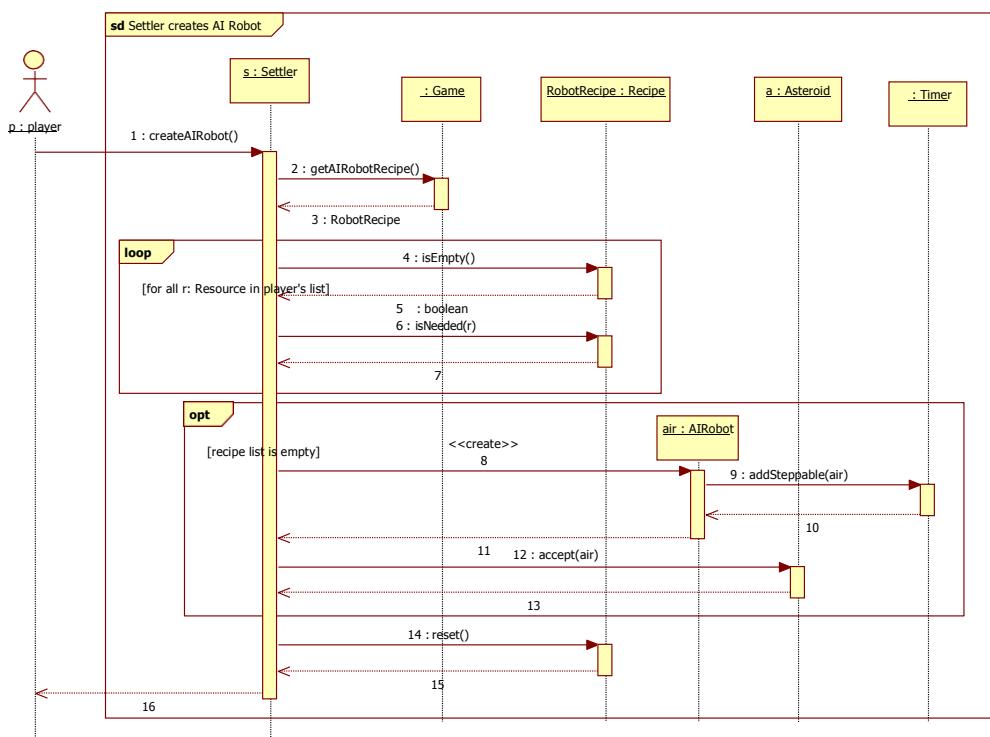
4.5.12 Settler mines



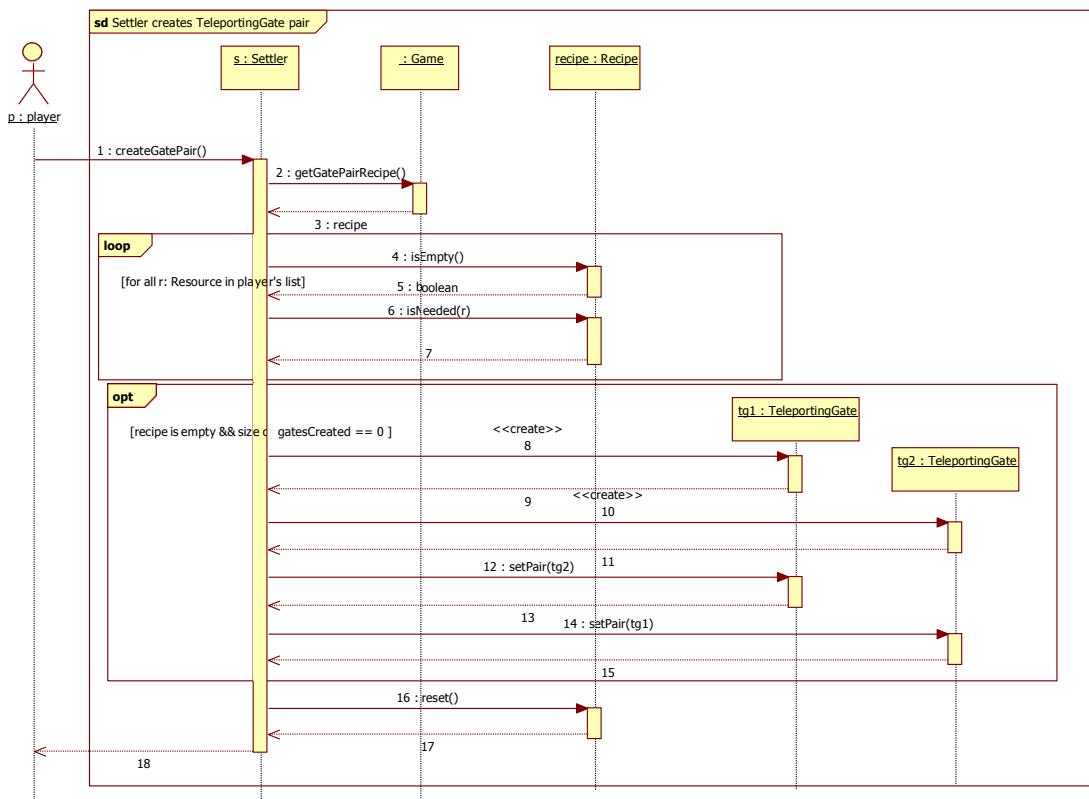
4.5.13 Settler restores



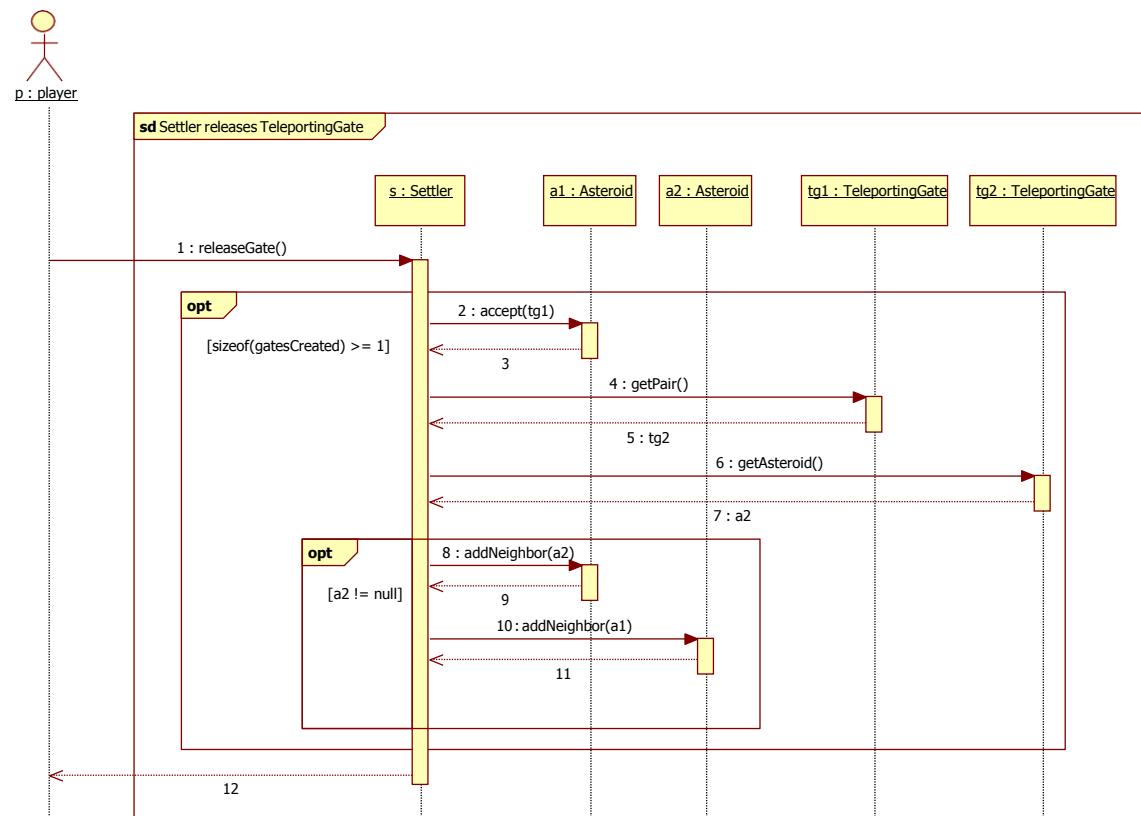
4.5.14 Settler creates AI Robot



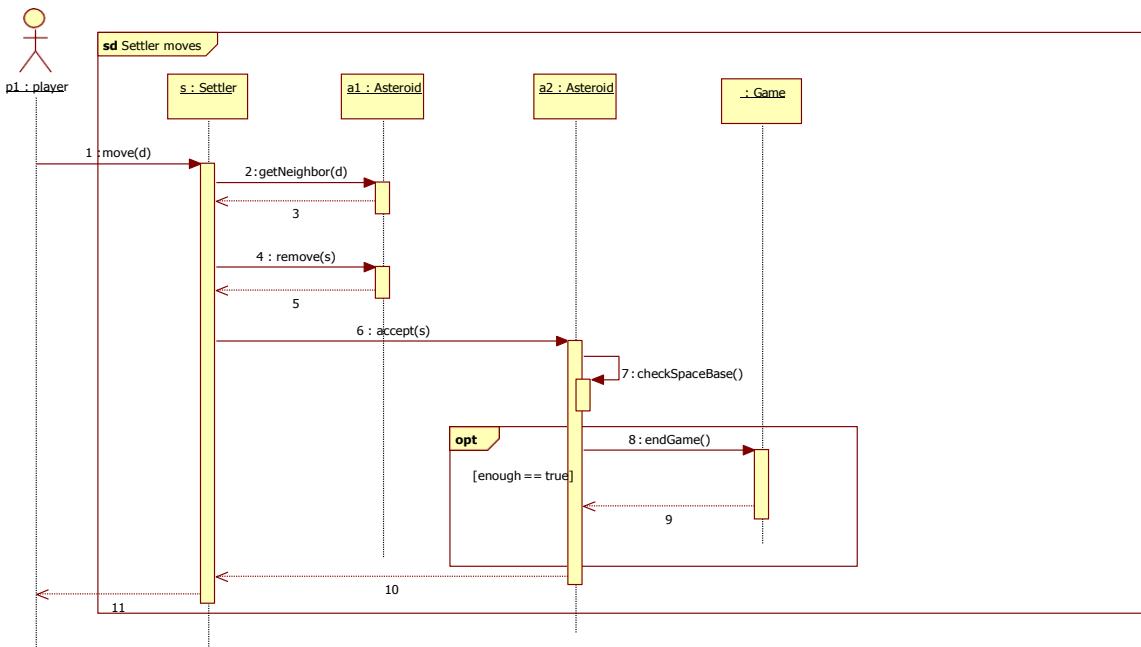
4.5.15 Settler creates TeleportingGate pair



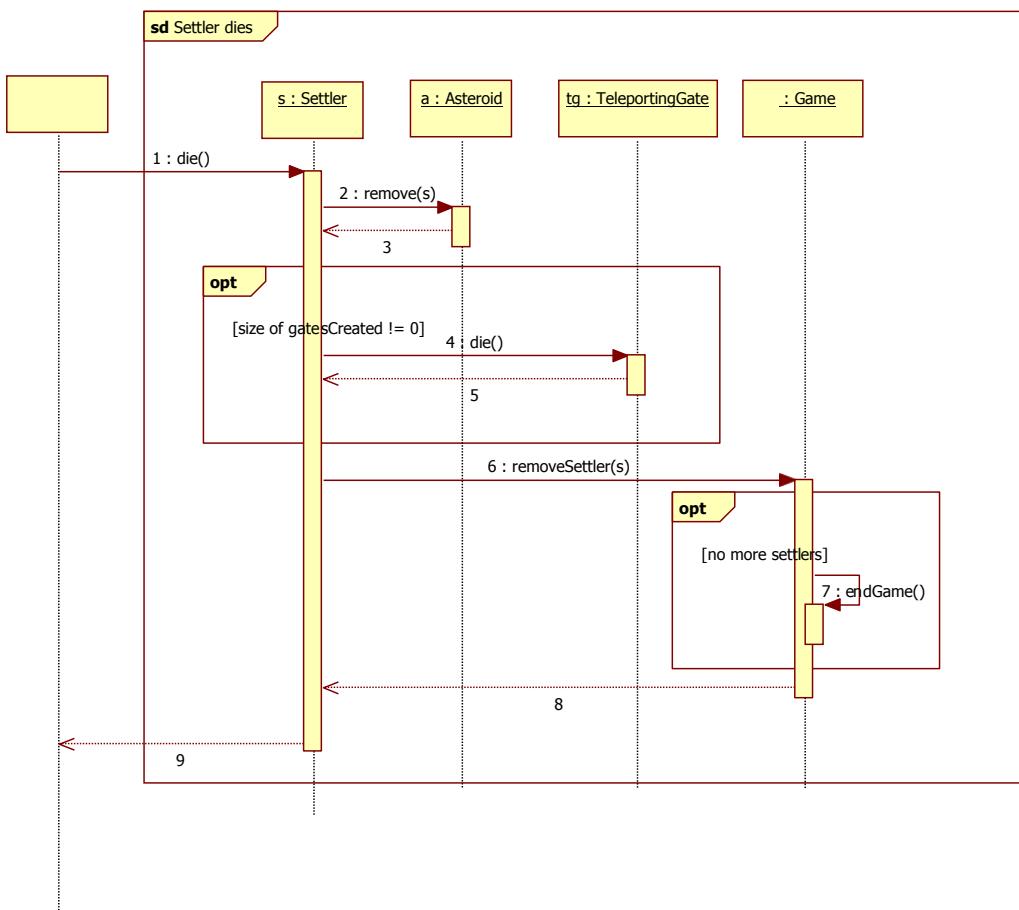
4.5.16 Settler releases TeleportingGate



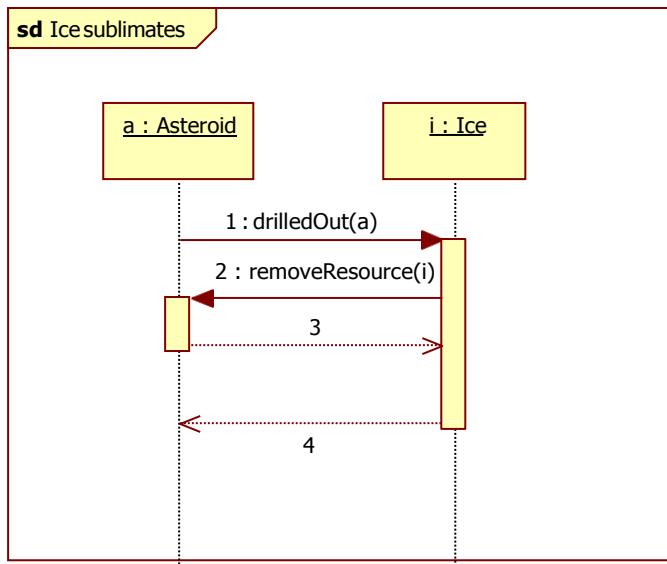
4.5.17 Settler moves



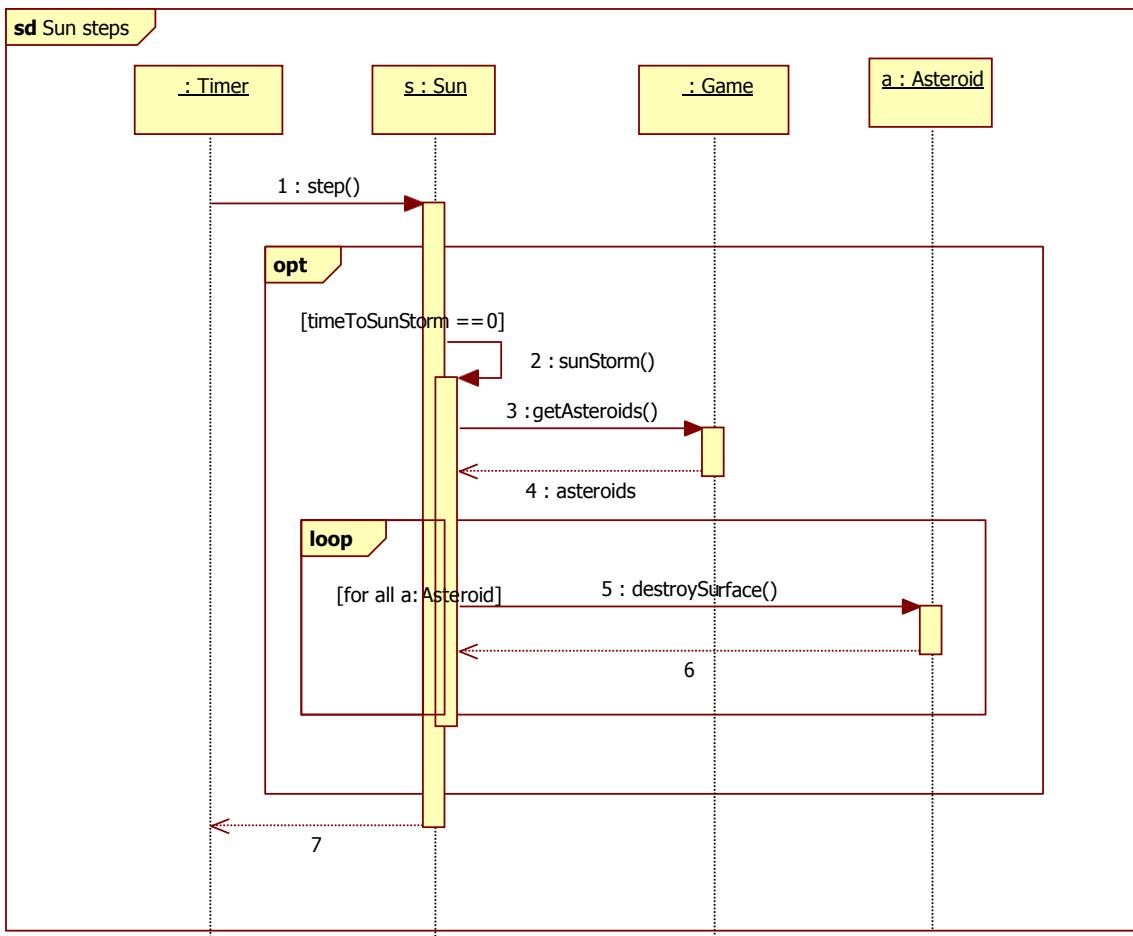
4.5.18 Settler dies



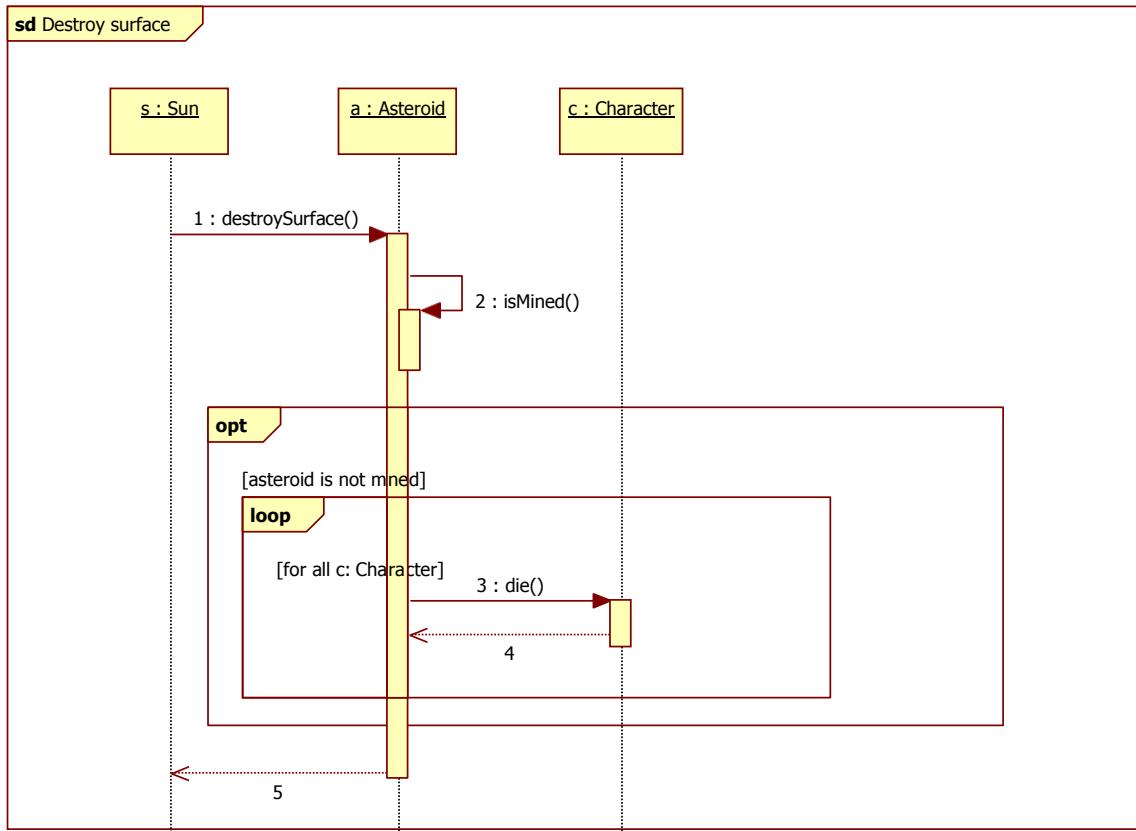
4.5.19 Ice sublimates



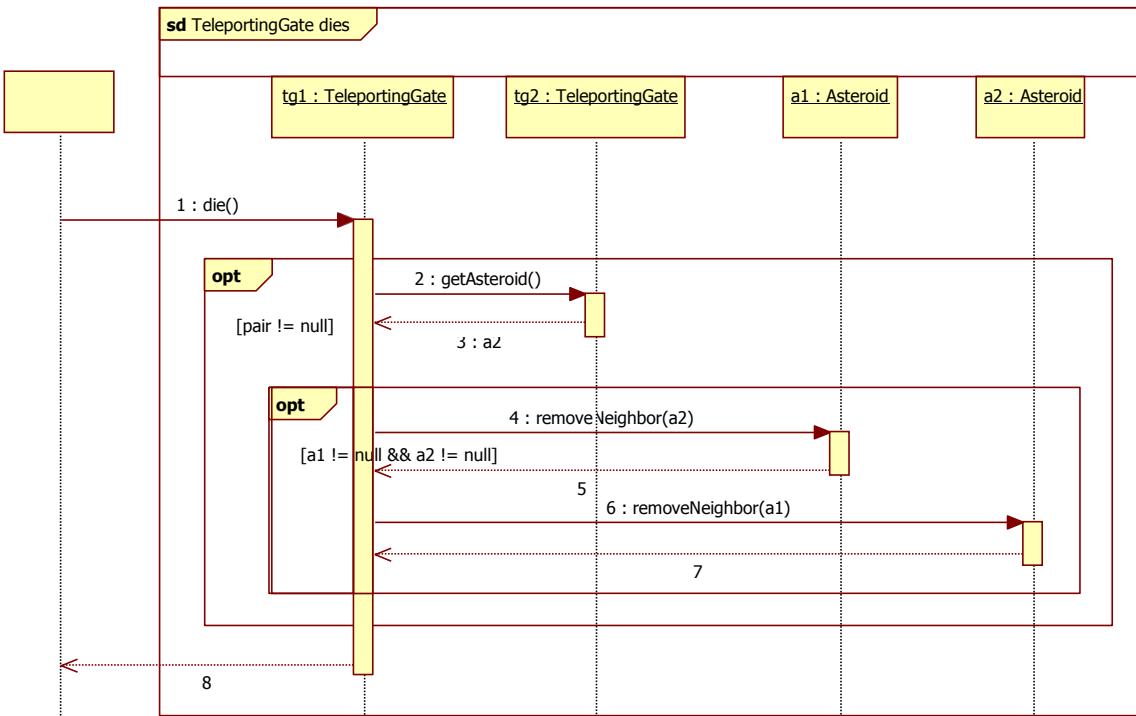
4.5.20 Sun steps



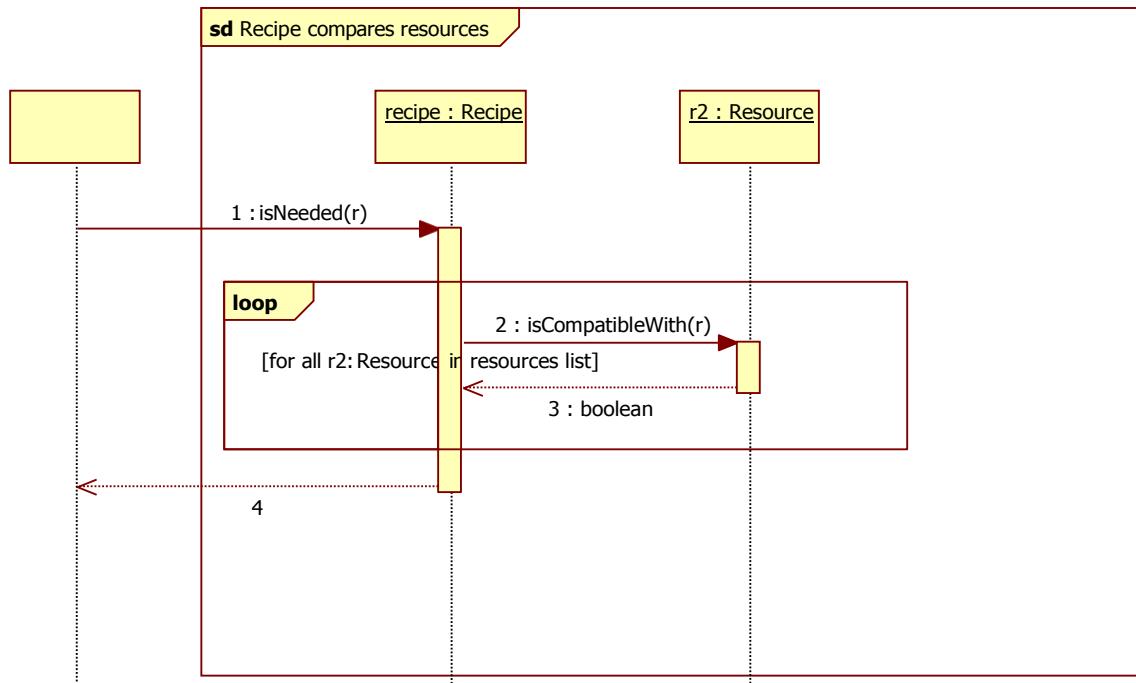
4.5.21 Destroy surface



4.5.22 TeleportingGate dies

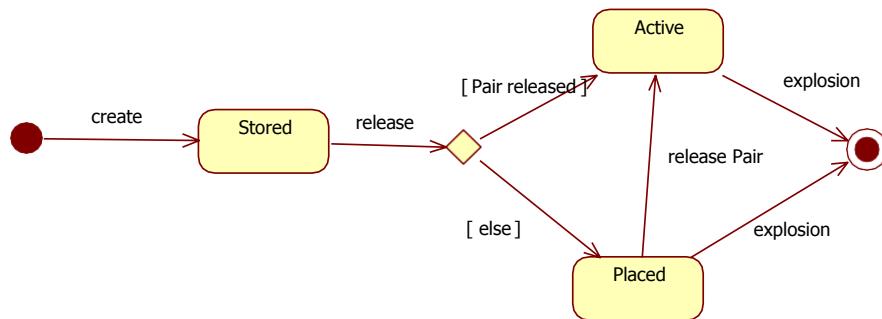


4.5.23 Recipe compares resources

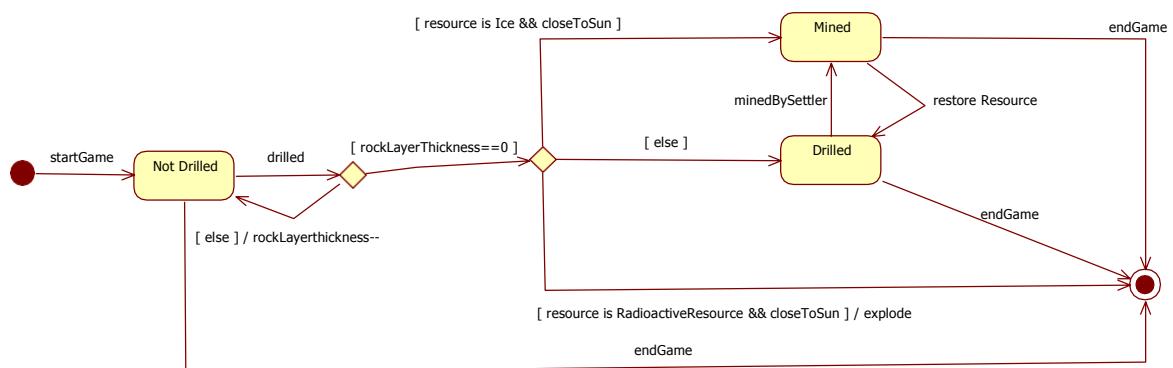


4.6 State-chartok

4.6.1 TeleportingGatePair



4.6.2 Asteroid



4.7 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2021.03.04. 16:00	2,5 óra	Havasi Kreták Mesterházi Radnai Szögi	Értekezlet. Az osztálydiagrammal kapcsolatos felmerült problémák megbeszélése, felelősségek átgondolása, brainstorming. Döntés: ResourceCollection megoldás helyett másik megoldás a nyersanyagok kezelésére (Recipe)
2021.03.05 17:00	1,5 óra	Havasi Kreták Radnai Szögi Mesterházi	Értekezlet. Recipe osztály működésének, függvényeinek megállapítása, megbeszélése. Döntés: Mesterházi készíti a statechartokat, Kreták szekvenciadiagramokat rajzol.
2021.03.05. 19:00	1 óra	Mesterházi	Tevékenység. Statechartok elkészítése, dokumentum javítása.
2021.03.05. 20:00	1 óra	Kreták	Tevékenység. Szekvenciadiagramok javítása, készítése.
2021.03.06. 15:00	0,5 óra	Havasi Kreták Mesterházi Szögi	Értekezlet. Az elkészített Statechartok és szekvenciadiagramok megbeszélése.
2021.03.06 19:00	2 óra	Kreták Szögi	Tevékenység. Az objektumkatalógus és az osztályleírások átnézése és módosítása a változásoknak megfelelően.
2021.03.07 18:00	0,5 óra	Radnai	Tevékenység. Az osztálydiagram formázása, fix, és a dokumentáció átnézése.

Az adott heti munka csapattagonkénti aránya:

Csapattag	Munka aránya százalékban
Havasi	20
Kreták	20
Mesterházi	20
Radnai	20
Szögi	20

5.

Szkeleton tervezése

7 – berenyi_kft

Konzulens:

Csapattagok

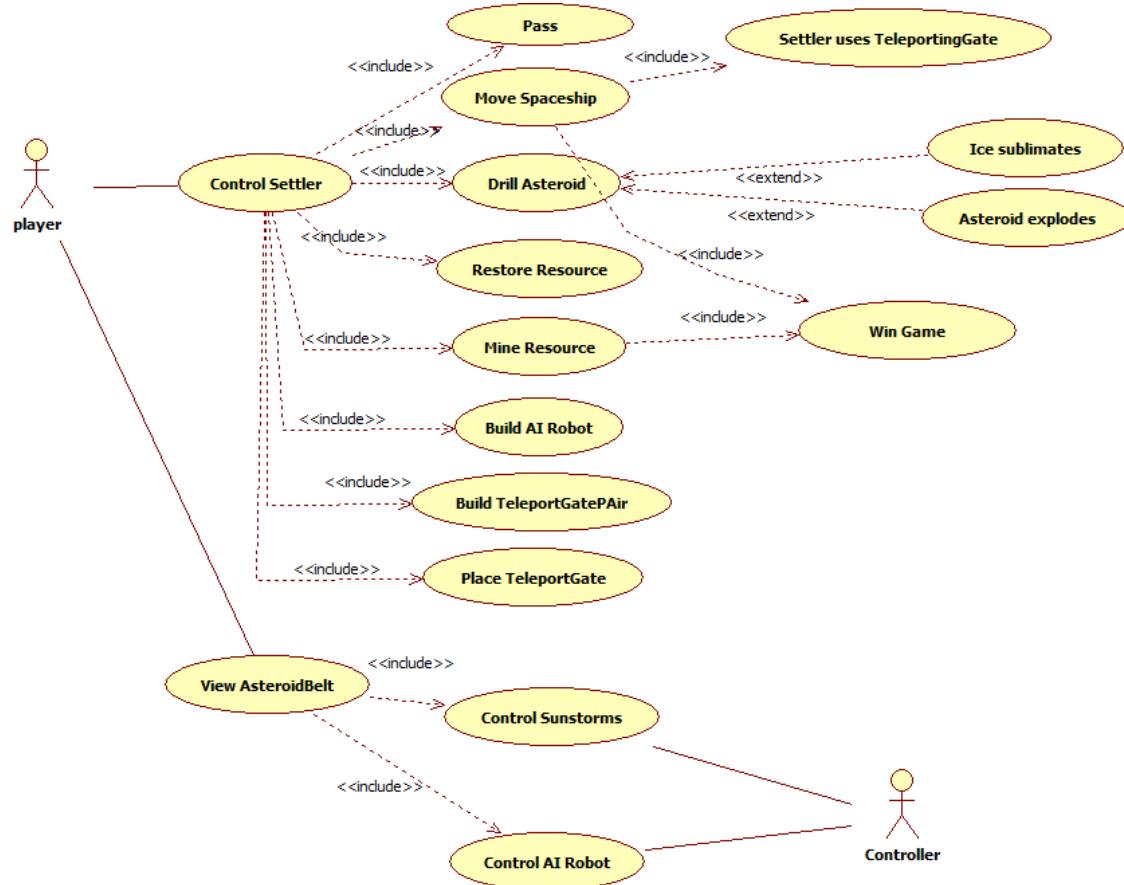
Kreták Barnabás András
Mesterházi Marcell
Szögi Richárd
Havasi Balázs
Radnai Bálint

2021.03.13

5. Szkeleton tervezése

5.1 A szkeleton modell valóságos use-case-ai

5.1.1 Use-case diagram



5.1.2 Use-case leírások

Use-case neve	Control Settler
Rövid leírás	A játékos irányítja a telepest
Aktorok	Player
Forgatókönyv	A játékos a telepessel mozoghat, fűrhet, bányászhat, robotot építhet, teleportkapu-párt építhet

Use-case neve	Pass
Rövid leírás	A játékos dönthet úgy, hogy nem végez érdemi lépést, csak passzol.
Aktorok	Player
Forgatókönyv	A játékos a telepessel ilyenkor nem végez érdemi lépést.

Use-case neve	Move Spaceship
Rövid leírás	A játékos képes a telepes űrhajójával egy másik szomszédos aszteroidára lépni.
Aktorok	Player
Forgatókönyv	1. A játékos az aktuális aszteroidáról egy szomszédos aszteroidára mozog.

Use-case neve	Settler uses TeleportingGate
Rövid leírás	A játékos képes a teleportkapuval a hozzá tartozó másik kapunál található aszteroidára lépni.
Aktorok	Player
Forgatókönyv	1. A játékos az aktuális aszteroidáról teleportkapuk segítségével aszteroidára mozog.
Alternatív forgatókönyv	1.A.1. A játékos nem tudja használni a teleportkaput, mert a kapu párja nincs lerakva.

Use-case neve	Drill Asteroid
Rövid leírás	A játékos egy egységet lefűr az aszteroidán.
Aktorok	Player
Forgatókönyv	1. A játékos egy egységnit fűr az aszteroidán, amin épp tartózkodik.
Alternatív forgatókönyv	1.A.1. A játékos napközelben fűr
Alternatív forgatókönyv	1.A.1.A.1. A játékos napközelben kifűr egy olyan aszteroidát, aminek a közepén Jég van, ami ilyenkor elszublimál.
Alternatív forgatókönyv	1.A.1.A.2. A játékos napközelben kifűr egy olyan aszteroidát, aminek a közepén Urán van, ami ilyenkor felrobban.

Use-case neve	Ice Sublimes
Rövid leírás	A játékos egy átfűr egy aszteroidát, ami ha napközelben van és a benne található nyersanyag vízjég
Aktorok	Player
Forgatókönyv	1. A játékos napközelben kifűr egy olyan aszteroidát, aminek a közepén Jég van, ami ilyenkor elszublimál.

Use-case neve	Asteroid explodes
Rövid leírás	A játékos egy átfűr egy aszteroidát, ami ha napközelben van és a benne található nyersanyag uránium, ami miatt felrobban az egész aszteroida és a rajta található telepesek is.
Aktorok	Player
Forgatókönyv	1. A játékos napközelben kifűr egy olyan aszteroidát, aminek a közepén Urán van, ami ilyenkor felrobban.

Use-case neve	Restore Resource
Rövid leírás	A játékos nyersanyagot helyez egy üres aszteroida belsejébe.
Aktorok	Player
Forgatókönyv	1. A játékos az ūrhajón tárolt egy egységni nyersanyagot az aszteroida üres magjába tesz.
Alternatív forgatókönyv	1.A.1. A játékos nem tudja letenni a nyersanyagot, mert az aszteroida nem üres

Use-case neve	Mine Resource
Rövid leírás	A játékos kinyeri az erőforrást.
Aktorok	Player
Forgatókönyv	1. A játékos kinyeri a fűrt lyukon keresztül az aszteroida magjában található erőforrást, és az ūrhajóján tárolja.
Alternatív forgatókönyv	1.A.1. A telepes nem tudja kinyerni a magból a nyersanyagot, mert az ūrhajó tárolója tele van, nem tud többet beletenni.
Alternatív forgatókönyv	1.A.2. A telepes nem tudja kinyerni a magból a nyersanyagot, mert a köpeny még nem lett átfűrva
Alternatív forgatókönyv	1.A.3. A telepes nem tudja kinyerni a magból a nyersanyagot, mert a mag nem üres.

Use-case neve	Build AI Robot
Rövid leírás	A játékos létrehoz egy AI robotot.
Aktorok	Player
Forgatókönyv	1. A játékos létrehoz egy AI robotot, amit a mesterséges intelligencia irányít, és ami segít neki a fúrában.
Alternatív forgatókönyv	1.A.1. A játékosnak nincs elég anyaga létrehozni robotot.

Use-case neve	Build TeleportGatePair
Rövid leírás	A telepes képes teleportkapu-párt építeni.
Aktorok	Player
Forgatókönyv	1. A játékos teleportkapu-párt épít
Alternatív forgatókönyv	1.A.1. A játékosnak nincs elég anyaga kapupárt létrehozni.
Alternatív forgatókönyv	1.A.2. A játékos nem tud építeni, mert van nála teleportkapu.
Alternatív forgatókönyv	1.B.1. A játékos letesz egy teleportkaput.

Use-case neve	Place TeleportGate
Rövid leírás	A játékos képes egy aszteroida mellé pályára állítani egy teleportkaput.
Aktorok	Player
Forgatókönyv	1. A játékos letesz egy teleportkaput.
Alternatív forgatókönyv	1.A.1. A játékos a második nála lévő kaput teszi le, így a teleportkapu aktív, használható lesz.

Use-case neve	Control Sunstorms
Rövid leírás	Az aszteroidaövbén előfordulhatnak napviharok.
Aktorok	Controller
Forgatókönyv	1. megsemmisíti azokat a telepeseket
Alternatív forgatókönyv	1.A.1. A telepes üreges aszteroidán tartózkodott, ezért nem tudta megsemmisíteni

Use-case neve	Control Asteroids
Rövid leírás	A játékban az aszteroidák állapotai változhatnak.
Aktorok	Controller
Forgatókönyv	Az aszteroida köpenye egy egységgel csökkenhet, belseje üressé válhat és fel is robbanhat.

Use-case neve	Control Game
Rövid leírás	A játék irányítása, lefolyásának kezelése, például körök kezelése.
Aktorok	Controller
Forgatókönyv	A Vezérlő kezeli a játékhöz szükséges folyamatokat.

Use-case neve	Control AI Robot
Rövid leírás	A játékban létrejöhettek AI robotok, melyek önállóan lépnek, de a Vezérlő tartja számon és kezeli őket.
Aktorok	Controller
Forgatókönyv	Az AI robotok mesterséges intelligencia segítségével mozgás vagy fúrás műveletet végeznek.

5.2 A szkeleton kezelői felületének terve, dialógusok

A program indításakor egy konzol fogad minket, ahol az alább megjelenített felsorolás fogad minket. Eleinte 9 tesztből választhatunk, ha megadjuk a választásunk sorszámát, akkor a lehetséges teszteseteket további részeseteit választhatjuk ki tesztelésre. A konstruktorok hívásakor, és a függvények hívásakor az ezeket azonosító szöveg fog megjelenni a konzolon. A végrehajtás sikereségéről a program ugyancsak a konzolon keresztül tájékoztat.

Az áttekintethetőség kevéért indentálunk.

Menü:

1. Win game
2. Move Spaceship
3. Drill Asteroid
4. Restore Resource
5. Mine Resource
6. Build AI Robot
7. Build TeleportingGatePair
8. Place TeleportingGate
9. Call Sunstorm

2. Move Spaceship

- a) Move to asteroid
- b) Move through TeleportingGate
- c) Fail to use TeleportingGate (nincs lerakva a párja)

3. Drill Asteroid

- a) Drill one layer
- b) Drill radioactive asteroid's last layer
- c) Drill icy asteroid's last layer
- d) Fail to drill (nincs több réteg)

4. Restore resource

- a) Restore Resource
- b) Fail to restore (nem üres mag, nincs kifürva)

5. Mine Resource

- a) Mine Resource
- b) Fail to mine (teli tár, üres mag, nincs kifürva)

6. Build AI Robot

- a) Build robot
- b) Fail to build (nincs elég nyersanyaga)

7. Build TeleportingGatePair

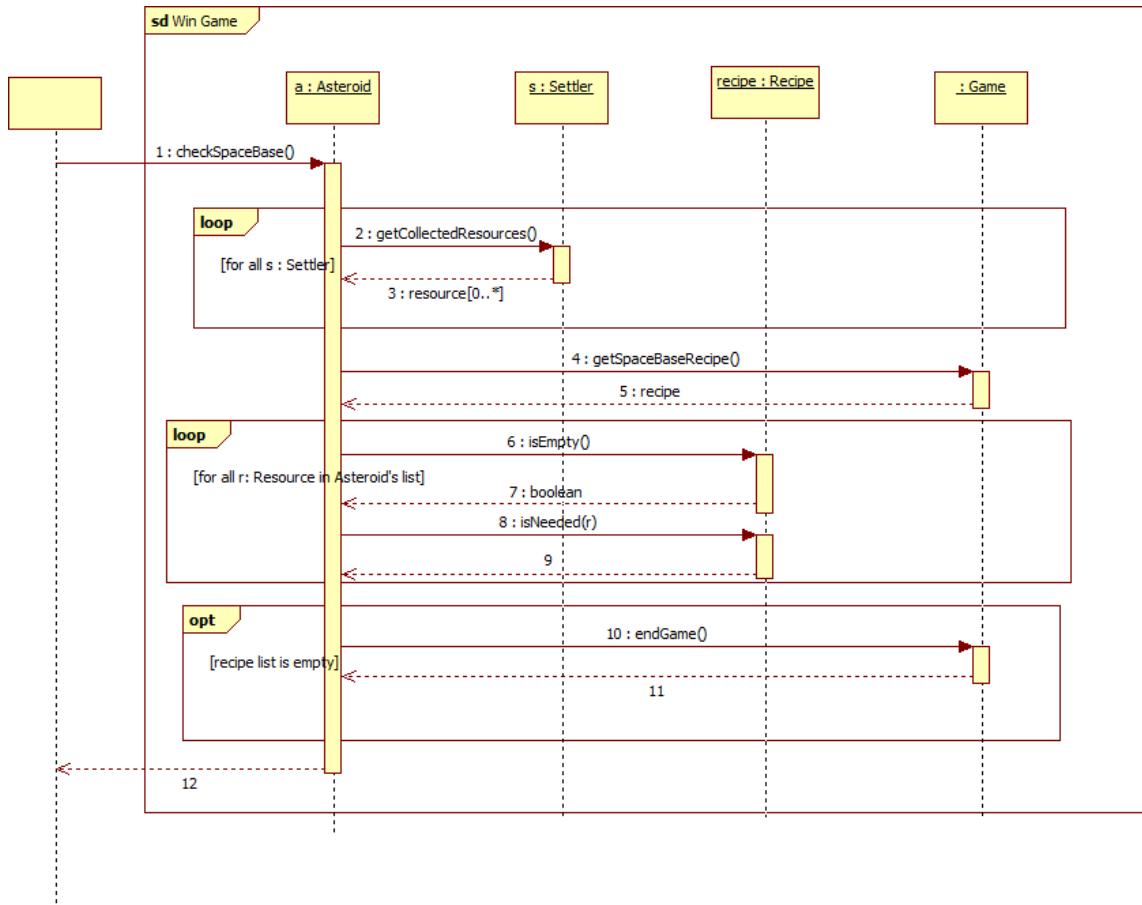
- a) Build TeleportingGatePair
- b) Fail to build (van még tárolt kapu, nincs elég nyersanyaga)

8. Place TeleportingGate

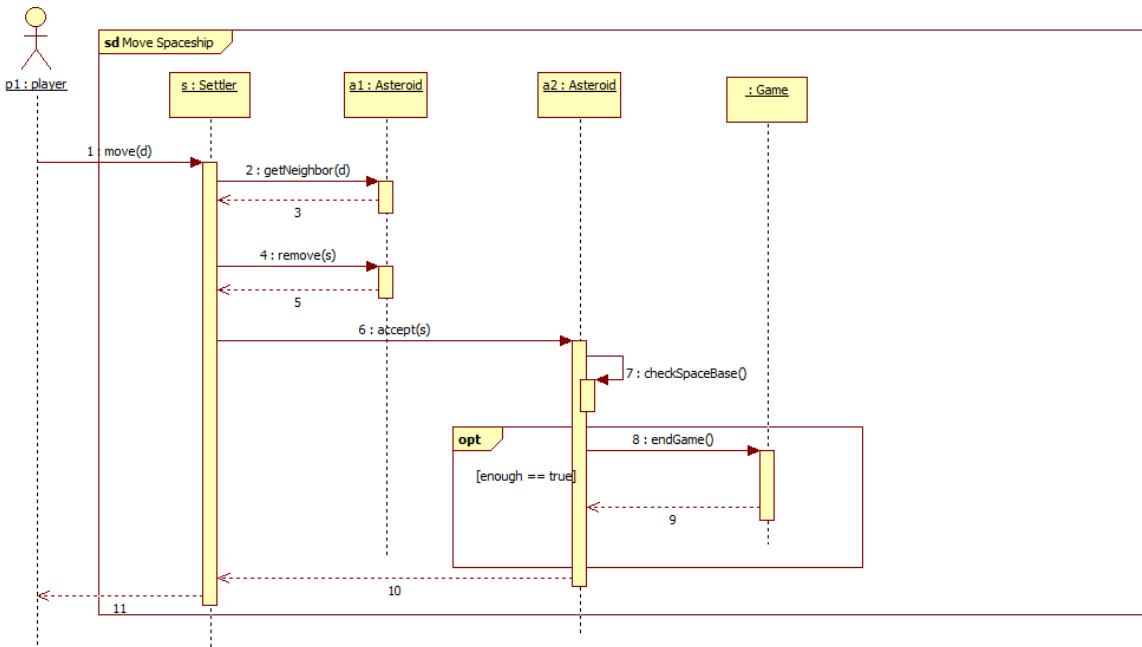
- a) Place TeleportingGate
- b) Fail to place (nincs tárolt kapu)

9. Call Sunstorm

- a) Asteroid is mined
- b) Asteroid is not mined

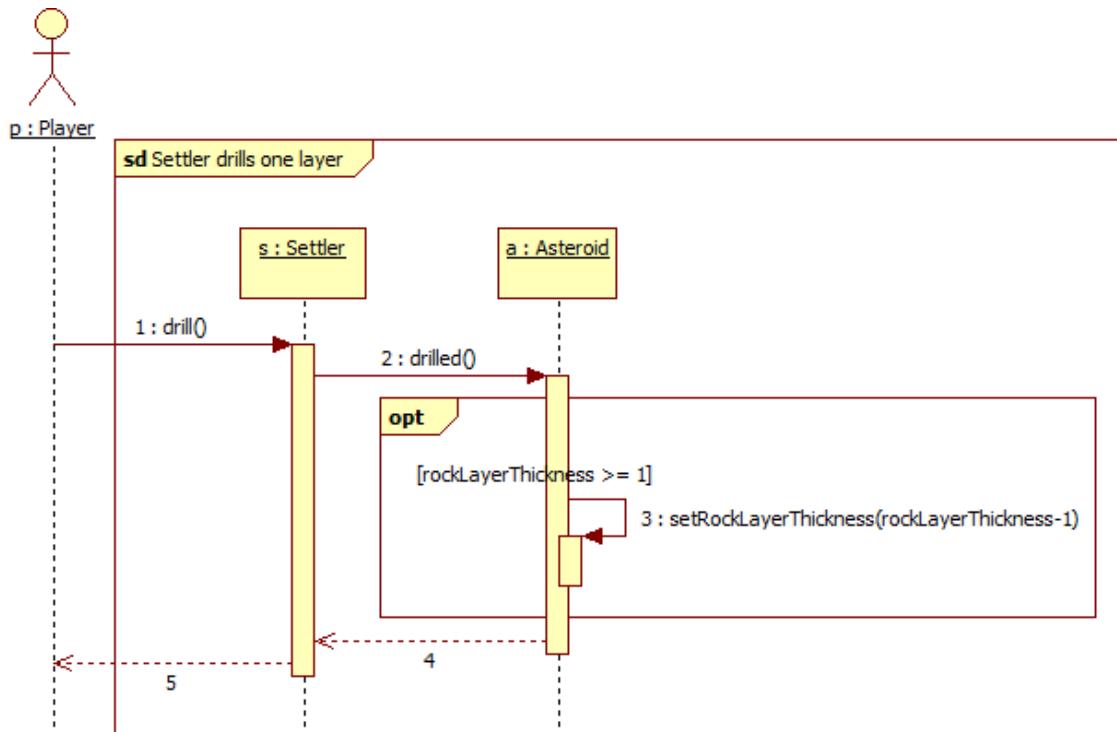
5.3 Szekvencia diagramok a belső működésre**5.3.1 Win Game**

5.3.2 Move Spaceship

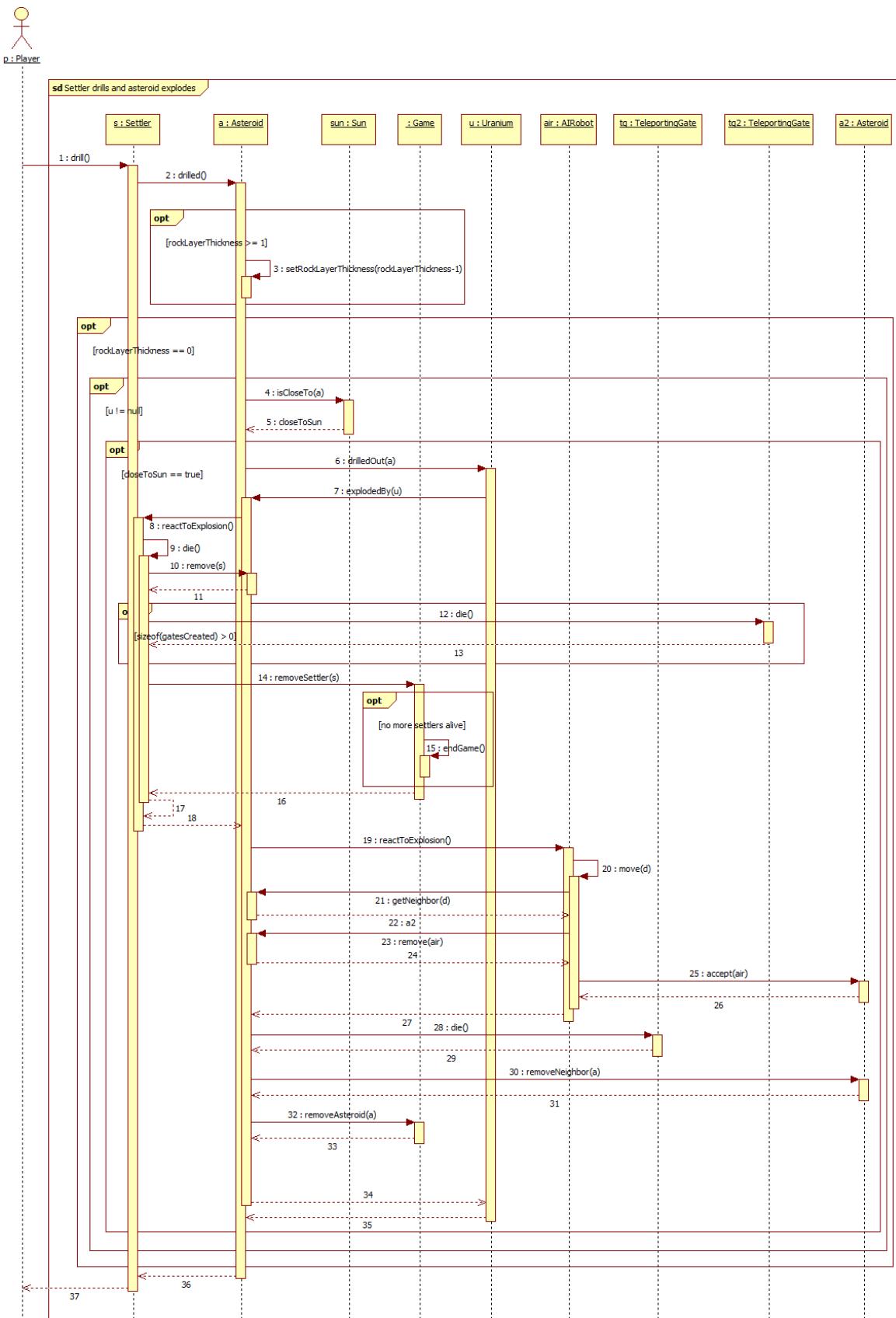


5.3.3 Settler drills

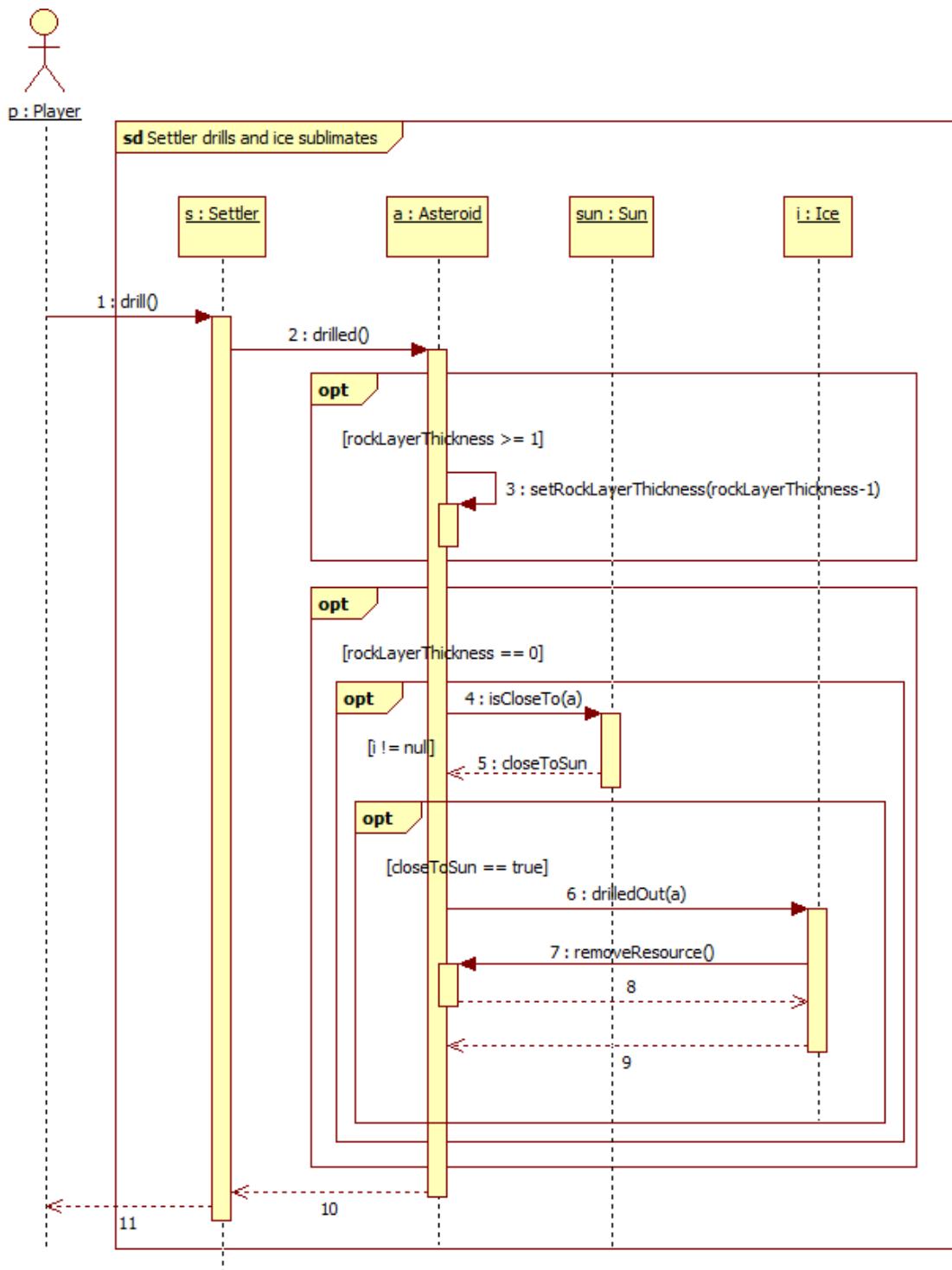
5.3.3.1 Settler drills one layer



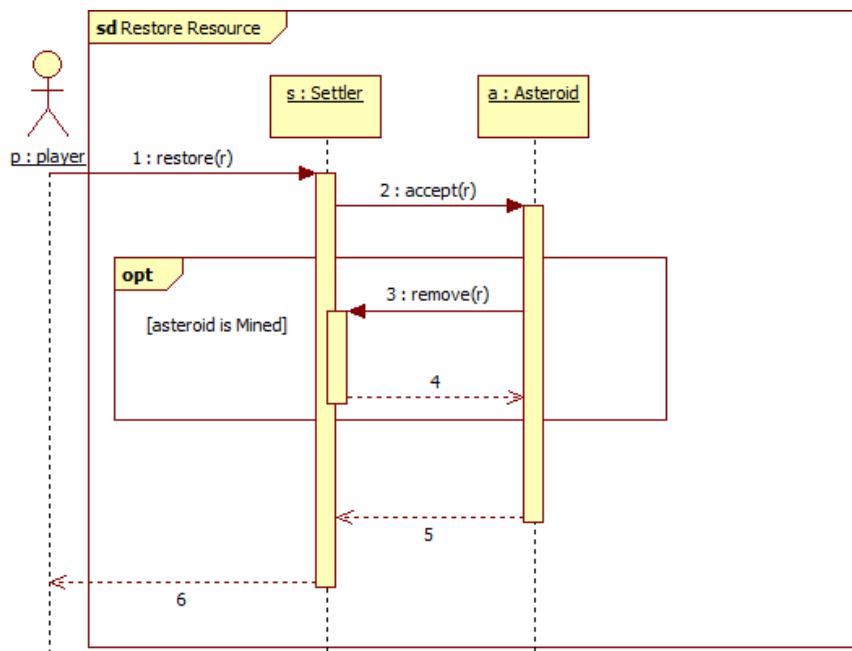
5.3.3.2 Settler drills and asteroid explodes



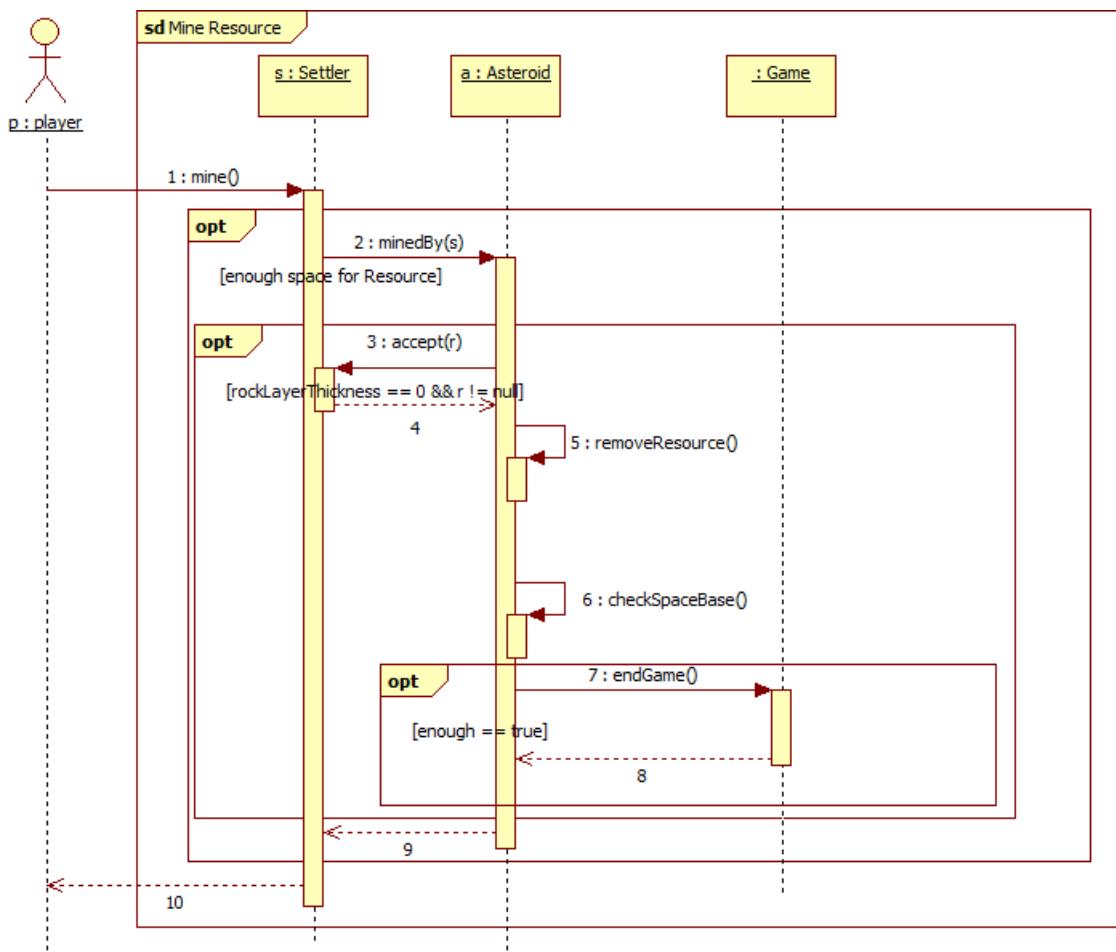
5.3.3.3 Settler drills and ice sublimates



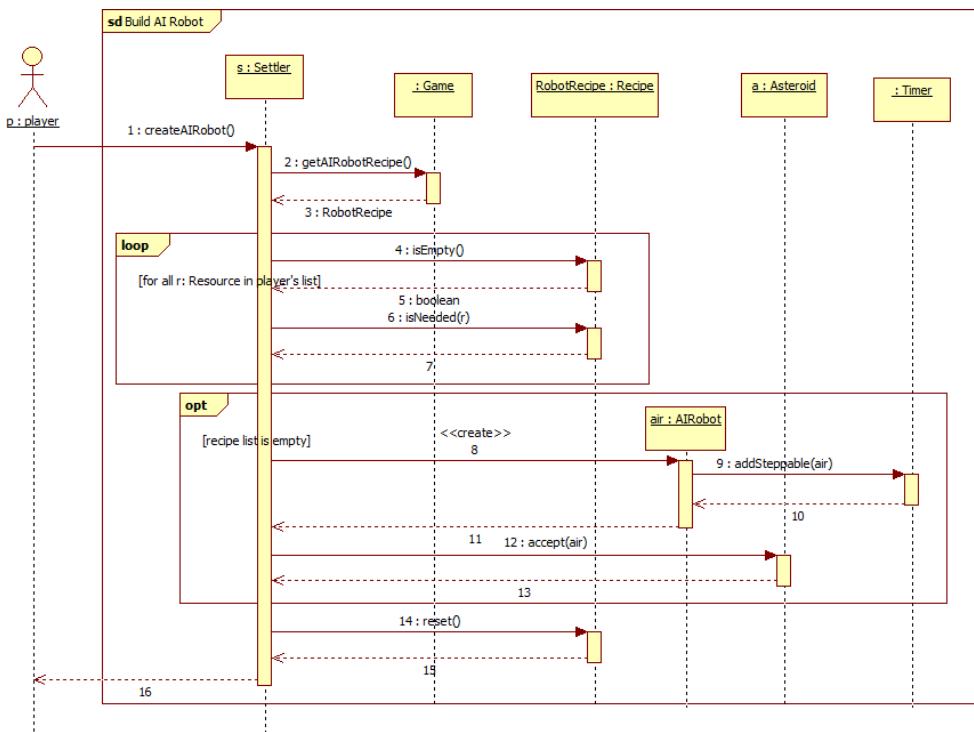
5.3.4 Restore Resource



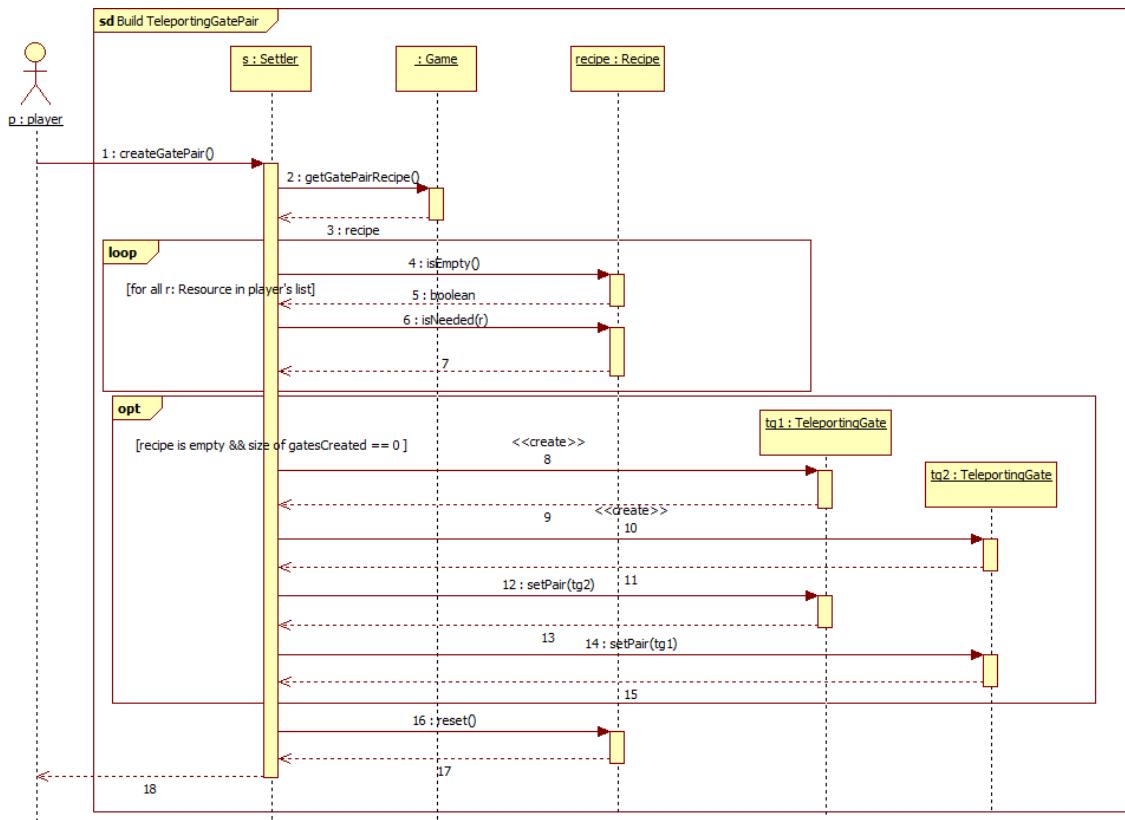
5.3.5 Mine Resource



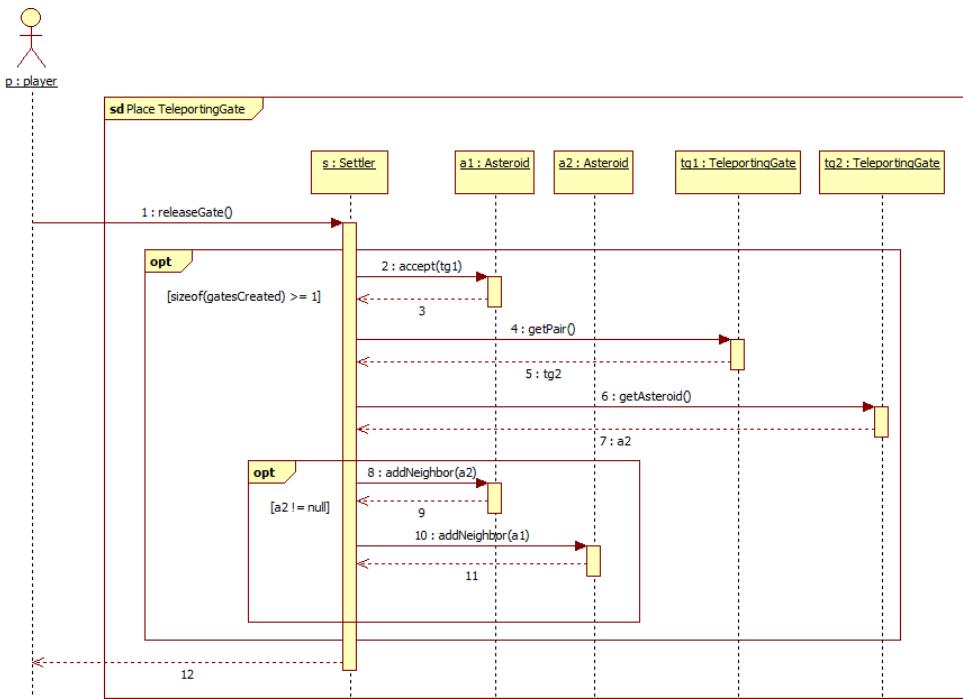
5.3.6 Build AI Robot



5.3.7 Build TeleportingGatePair

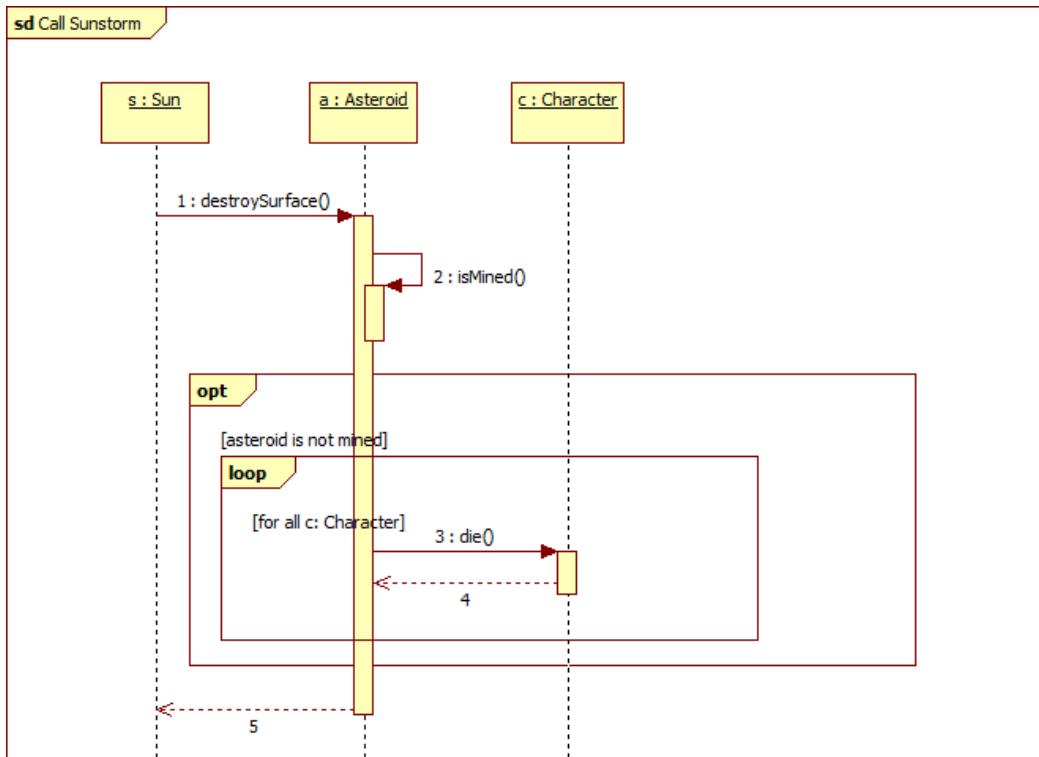


5.3.8 Place TeleportingGate

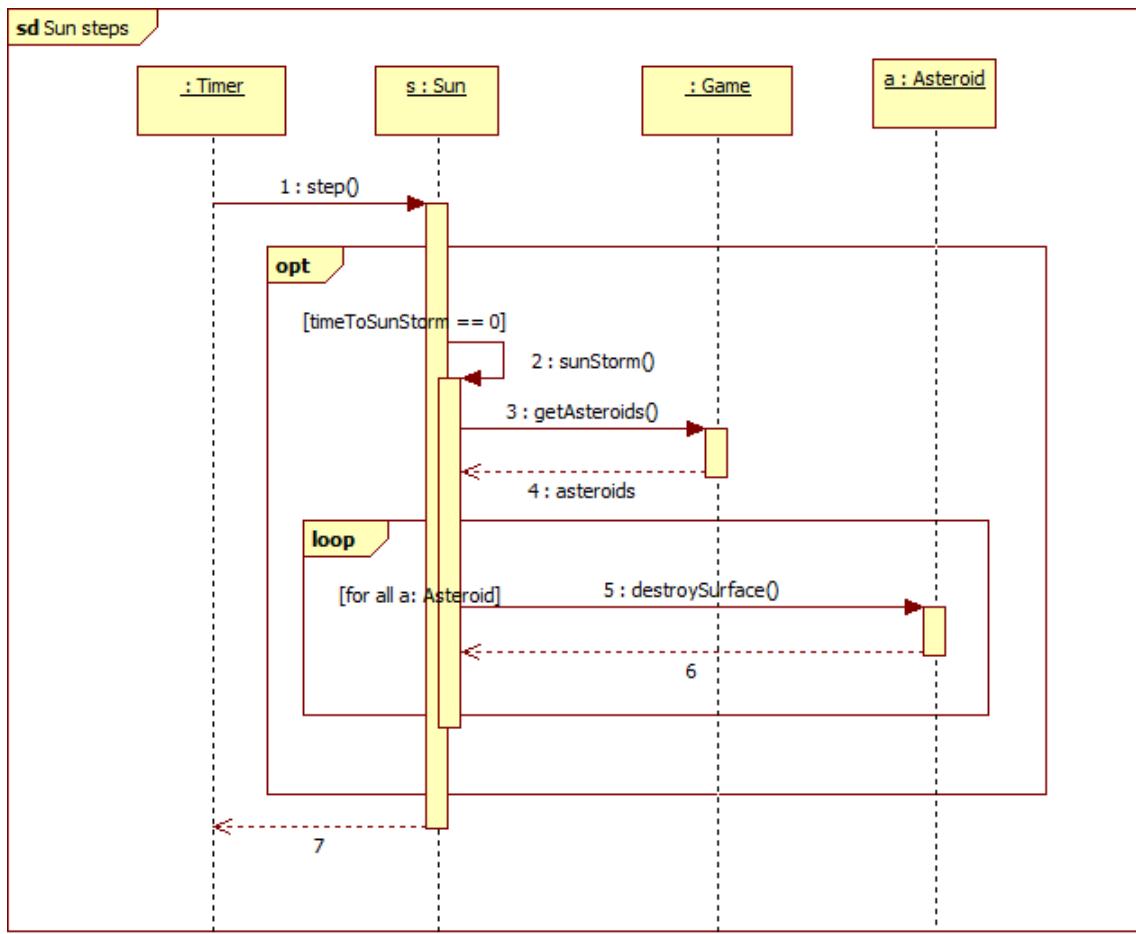


5.3.9 Sunstorm

5.3.9.1 Call Sunstorm

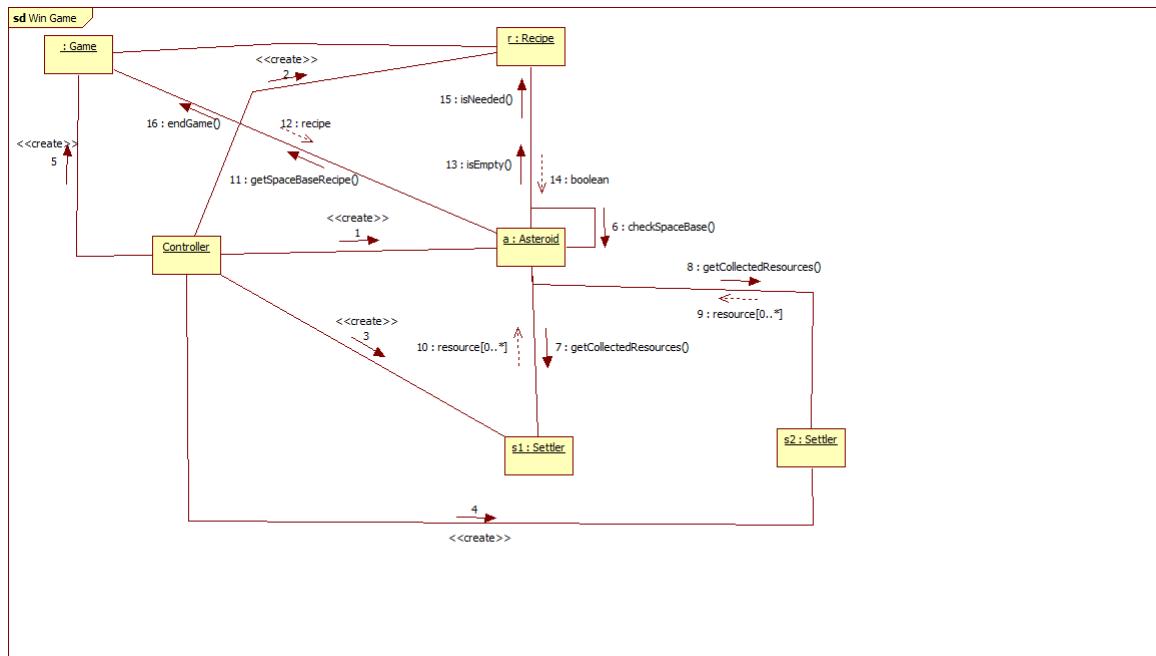


5.3.9.2 Sun steps



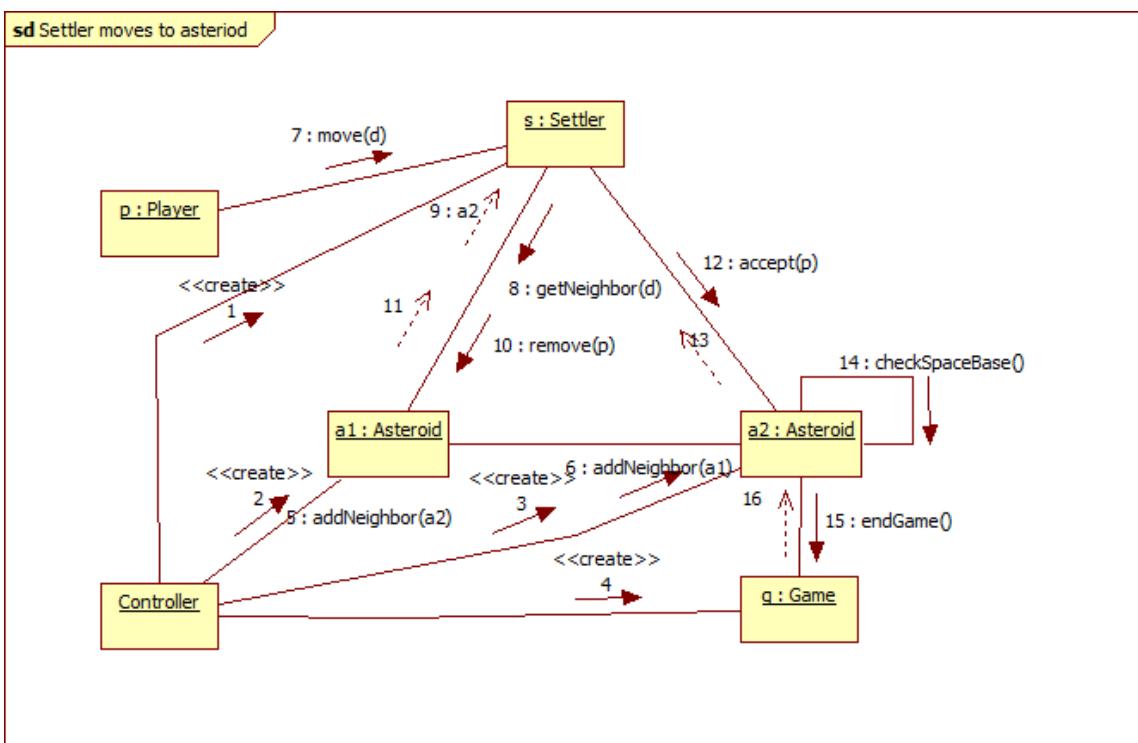
5.4 Kommunikációs diagramok

5.4.1 Win Game

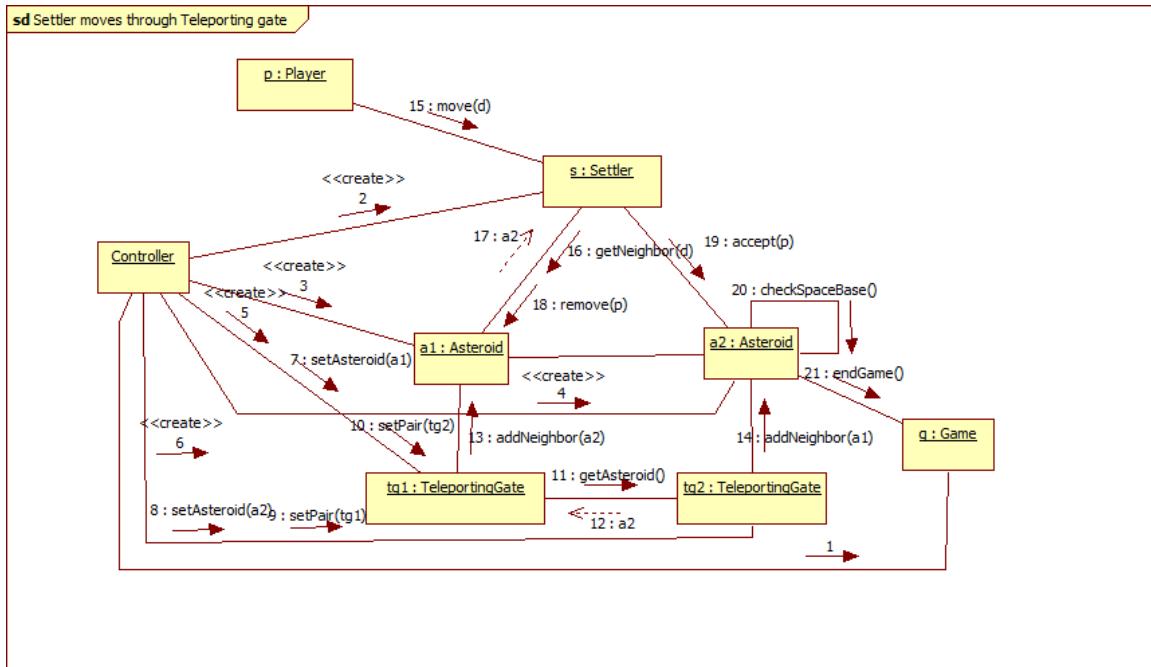


5.4.2 Move Spaceship

5.4.2.1 Settler moves to asteroid

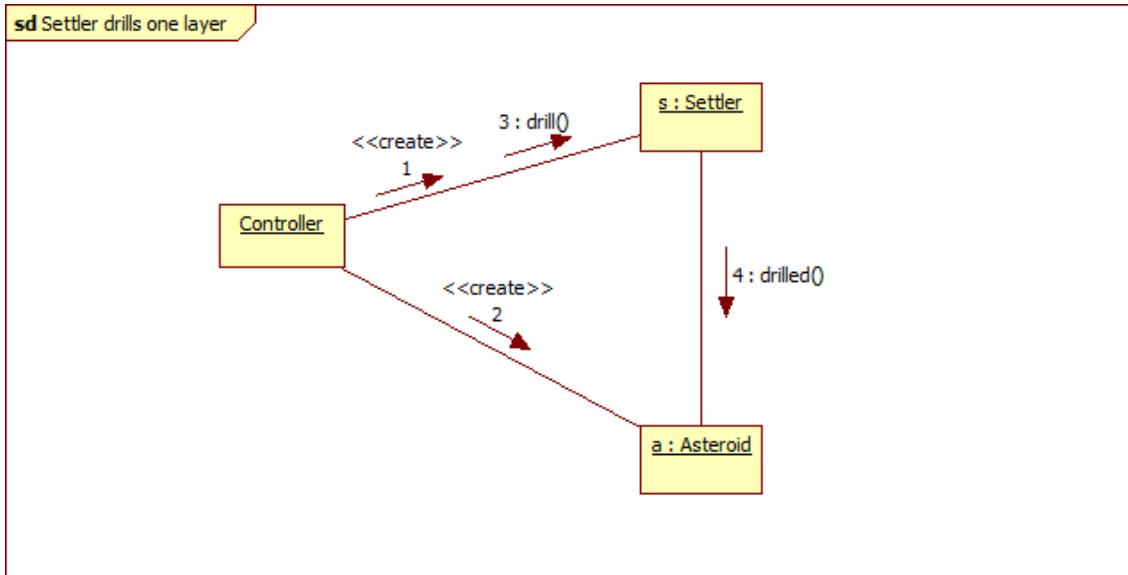


5.4.2.2 Settler moves through TeleportingGate

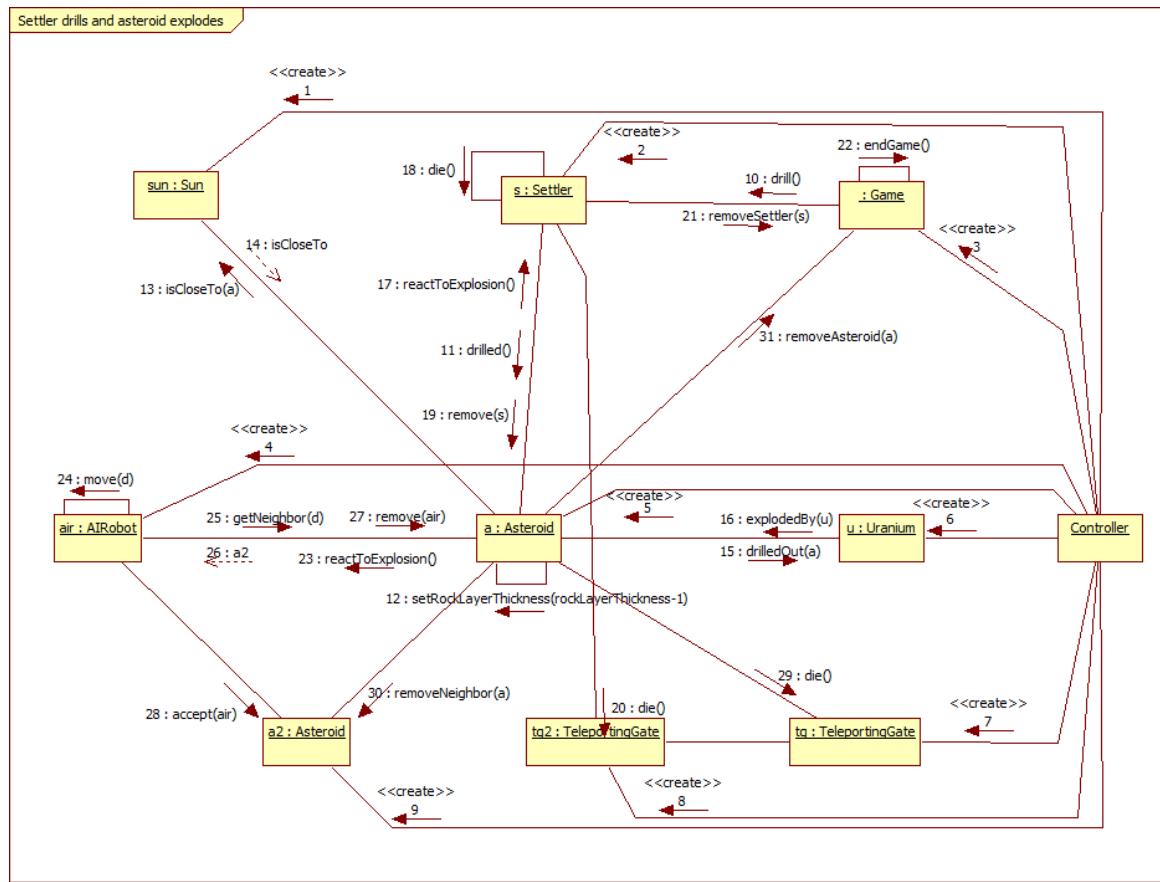


5.4.3 Settler drills

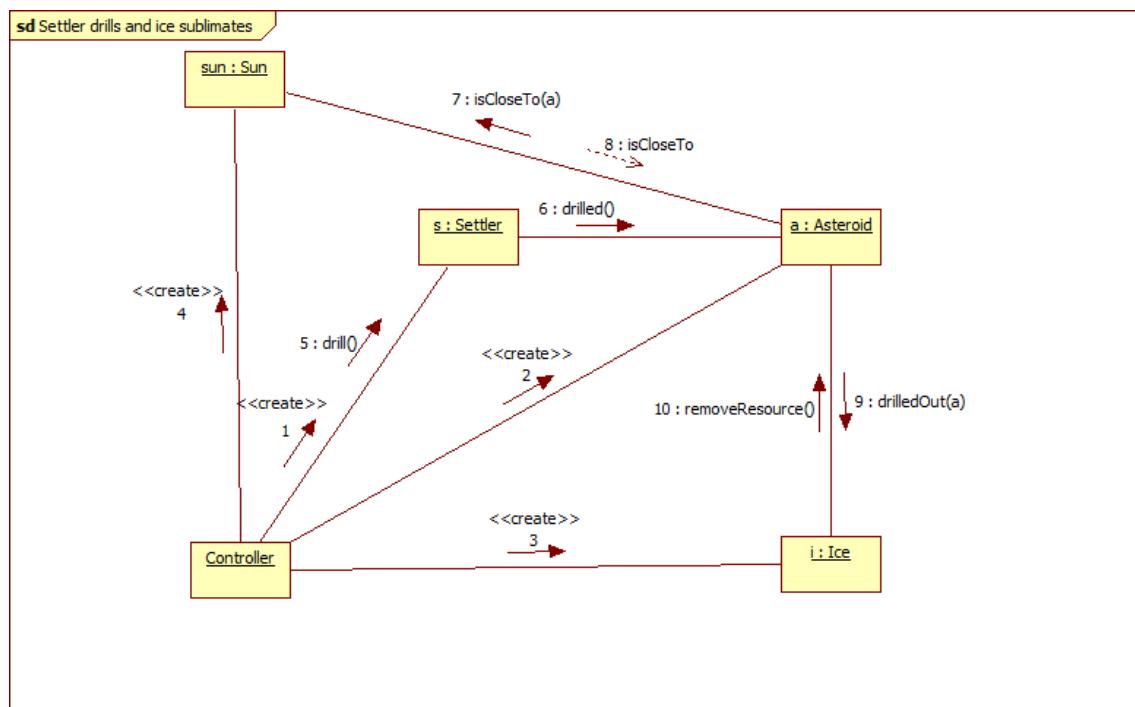
5.4.3.1 Settler drills one layer



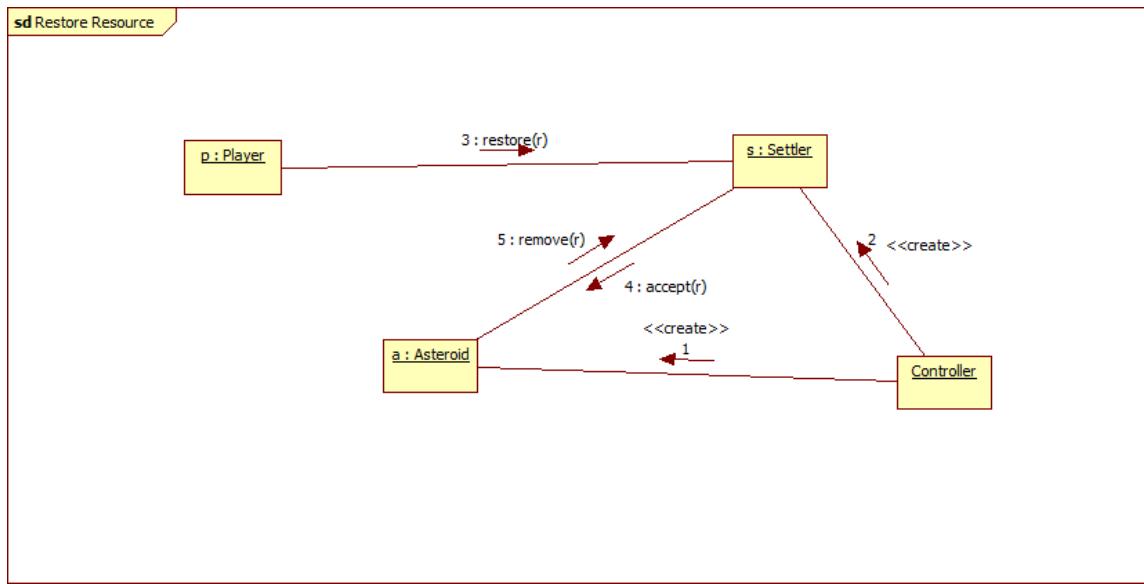
5.4.3.2 Settler drills and asteroid explodes



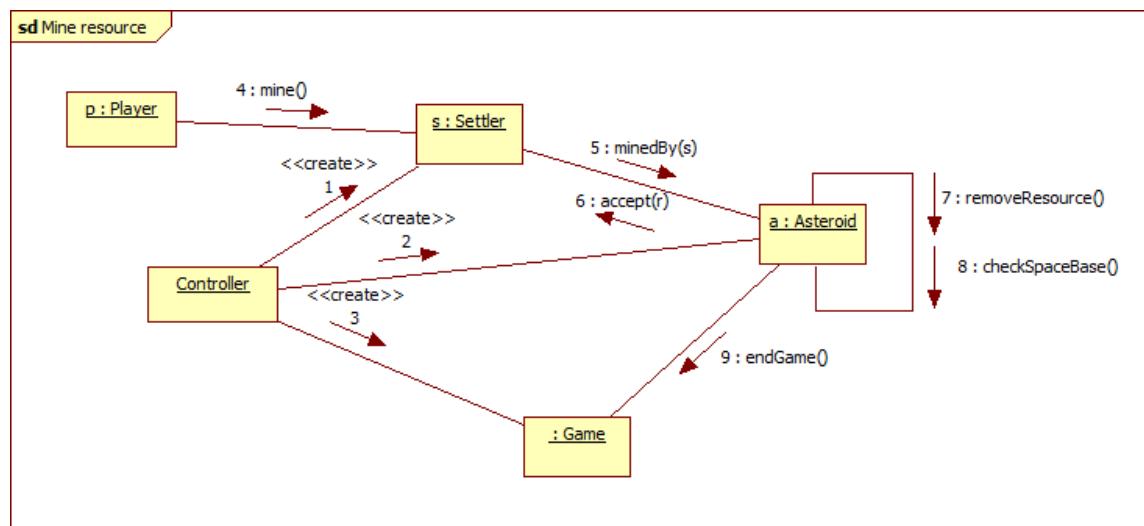
5.4.3.3 Settler drills and ice sublimates



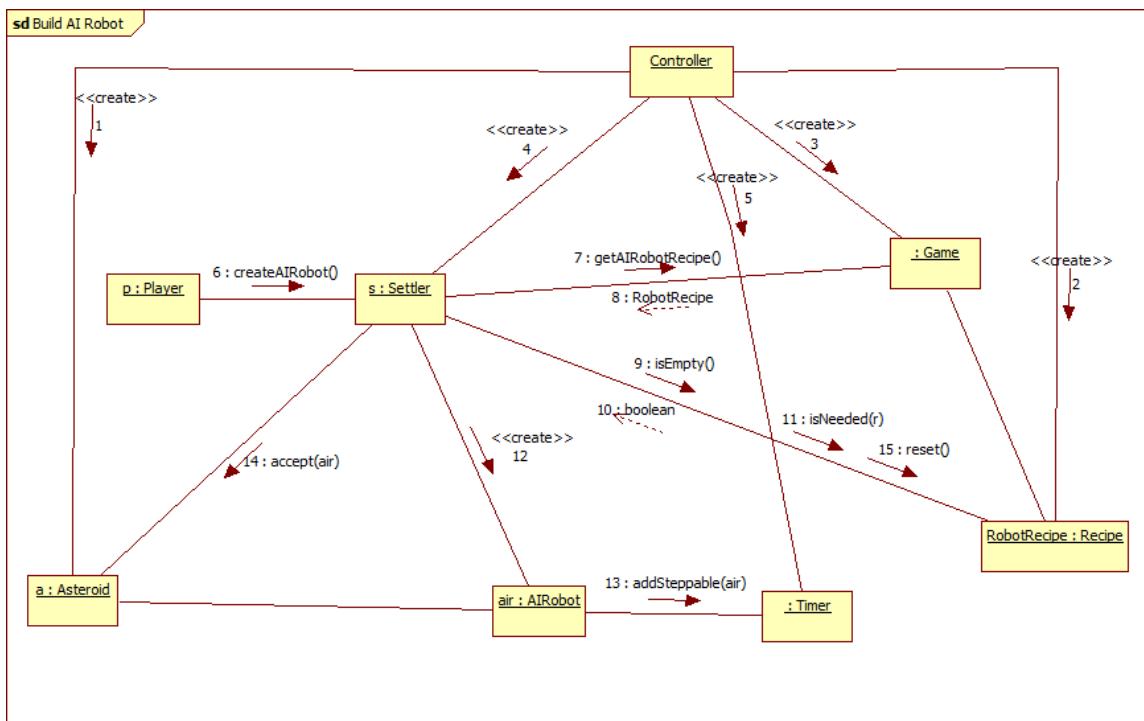
5.4.4 Restore Resource



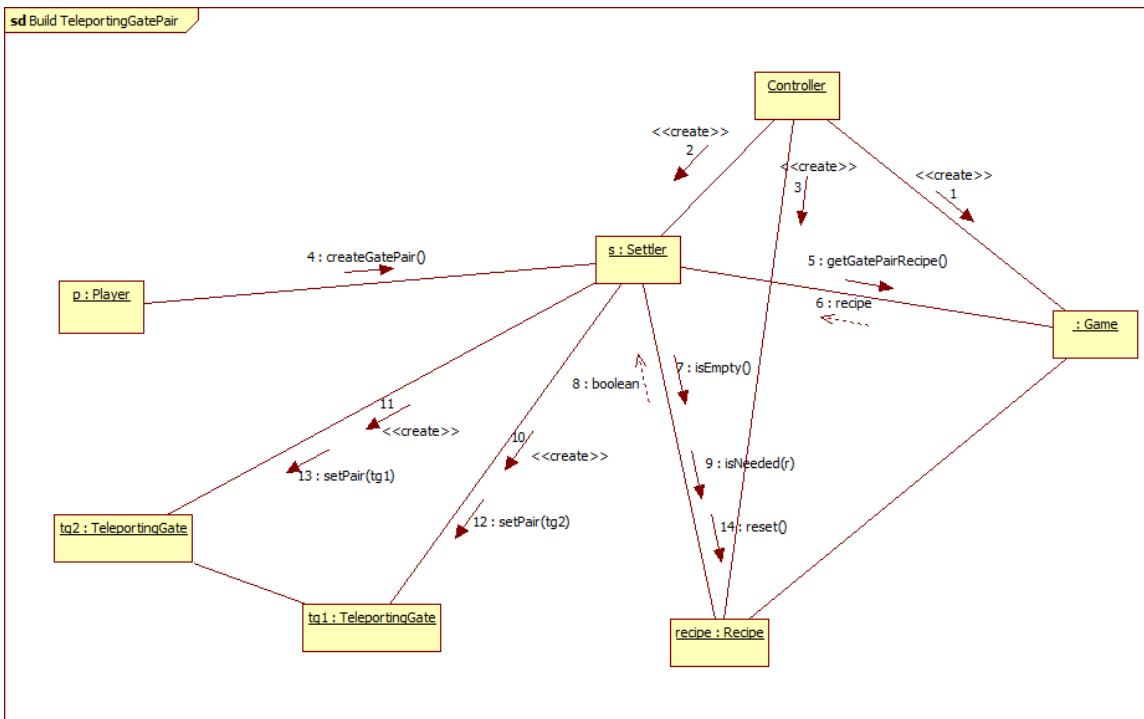
5.4.5 Mine Resource



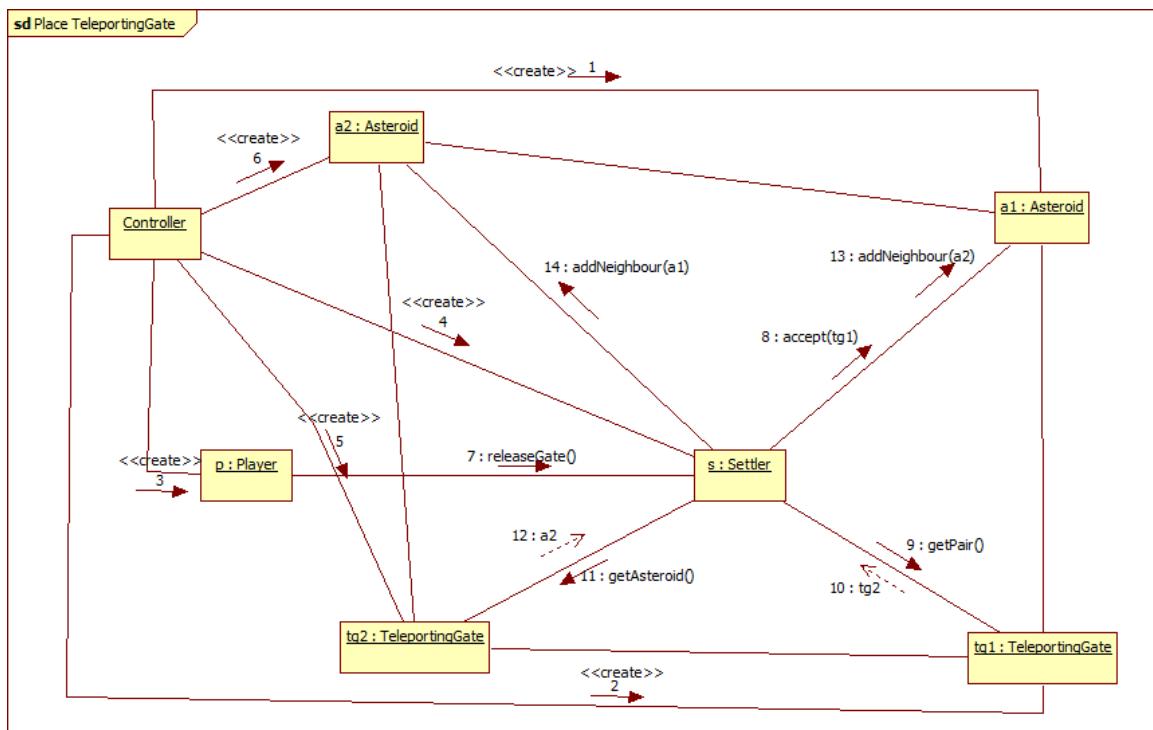
5.4.6 Build AI Robot



5.4.7 Build TeleportingGatePair

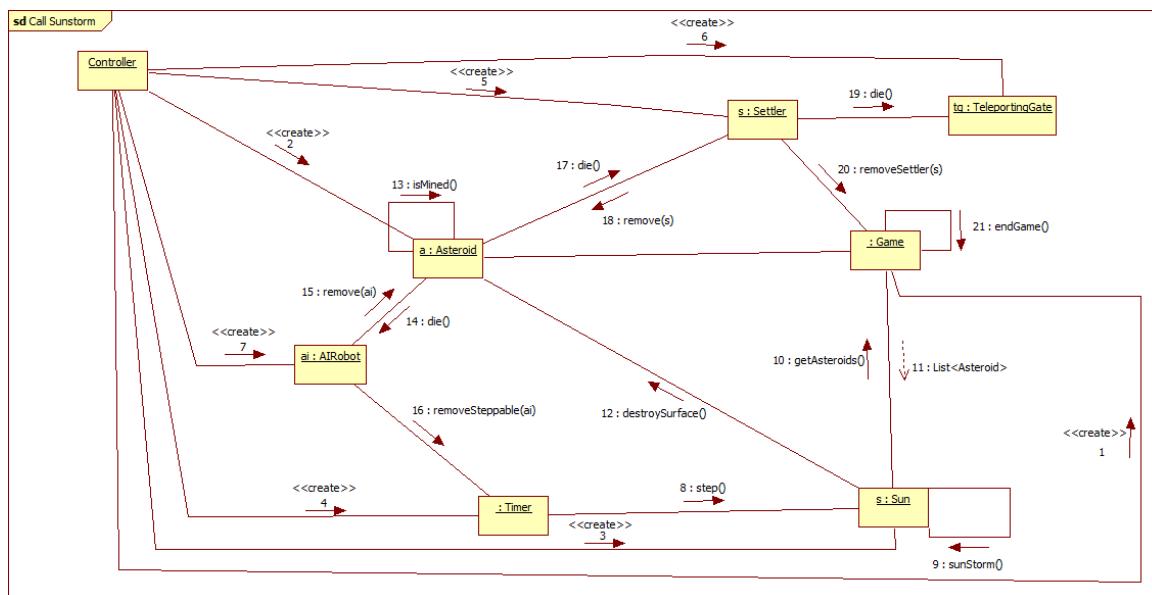


5.4.8 Place TeleportingGate



5.4.9 Call Sunstorm

A Sun steps és a Call Sunstorm szekvencia diagramokhoz közös kommunikációs diagramot készítettünk.



5.5 Napló

Kezdet	Időtartam	Részttvevők	Leírás
2021.03.11. 18:00	1 óra	Havasi Kreták Mesterházi Radnai Szögi	Értekezlet. Szekvenciák, hibák átnézése, feladat megbeszélése, kiosztása.
2021.03.13.14:00	2 óra	Havasi Kreták Mesterházi Radnai Szögi	Értekezlet: Feladat értelmezése, tesztek, usecase-ek meghatározása.
2021.03.13.15:00	1,5 óra	Kreták	Tevékenység. Kommunikációs diagramok rajzolása.
2021.03.13.15:00	2 óra	Havasi Kreták Mesterházi Radnai Szögi	Értekezlet: Elkészített feladatok, diagramok átnézése, ötletelések, javítgatások
2021.03.15.18:00	1.5 óra	Mesterházi	Tevékenység: Dokumentum összeszerkesztése

Az adott heti munka csapattagonkénti aránya:

Csapattag	Munka aránya százalékban
Havasi	20
Kreták	20
Mesterházi	20
Radnai	20
Szögi	20

6. Szkeleton beadás

7 – berenyi_kft

Konzulens:

Csapattagok

Kreták Barnabás András

Mesterházi Marcell

Szögi Richárd

Havasi Balázs

Radnai Bálint

2021.03.22

6. Szkeleton beadás

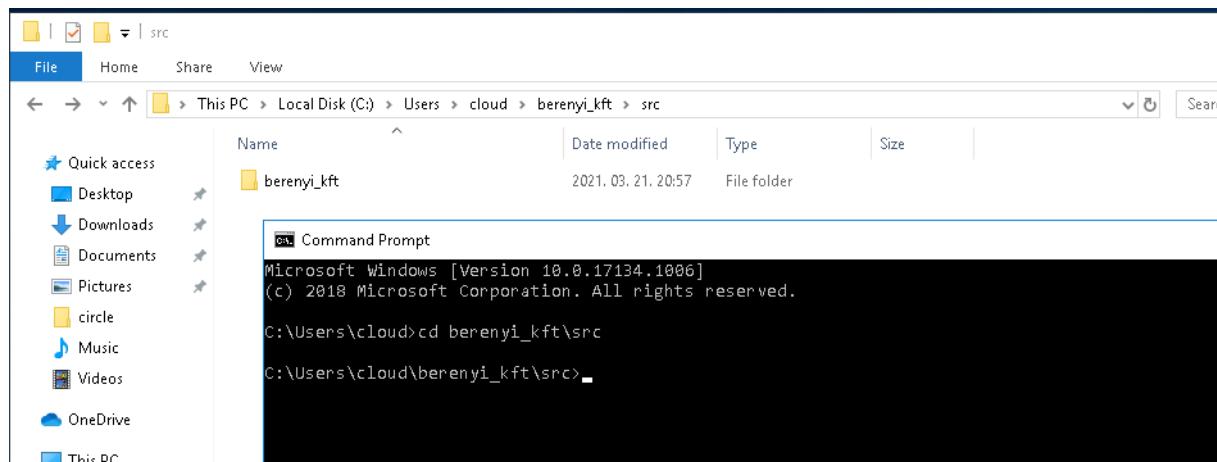
6.1 Fordítási és futtatási útmutató

6.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
AIRobot.java	1717 bytes	2020.03.16	AIRobot osztály megvalósítása
Asteroid.java	12 102 bytes	2020.03.16	Asteroid osztály megvalósítása
Character.java	2700 bytes	2020.03.16	Character osztály megvalósítása
Coal.java	628 bytes	2020.03.16	Coal osztály megvalósítása
Game.java	5721 bytes	2020.03.16	Game osztály megvalósítása
ISteppable.java	293 bytes	2020.03.16	ISteppable osztály megvalósítása
Ice.java	892 bytes	2020.03.16	Ice osztály megvalósítása
Iron.java	625 bytes	2020.03.16	Iron osztály megvalósítása
RadioactiveResource.java	293 bytes	2020.03.16	RadioactiveResource osztály megvalósítása
Recipe.java	480 bytes	2020.03.16	Recipe osztály megvalósítása
Resource.java	1910 bytes	2020.03.16	Resource osztály megvalósítása
Settler.java	7724 bytes	2020.03.16	Settler osztály megvalósítása
Skeleton.java	33 435 bytes	2020.03.17	Tesztelések megvalósítása, hívása
Sun.java	2 406 bytes	2020.03.16	Sun osztály megvalósítása
TeleportingGate.java	2 679 bytes	2020.03.16	TeleportingGate osztály megvalósítása
Timer.java	2 241 bytes	2020.03.16	Timer osztály megvalósítása
Uran.java	648 bytes	2020.03.16	Uran osztály megvalósítása

6.1.2 Fordítás

A program természetesen fordítható Eclipse-en keresztül, de cmd-ből is fordíthatjuk, úgy, hogy a gyökérkermappába megyünk, az *src* mappán belül, ahol a **javac berenyi_kft*.java** parancssal fordíthatunk.



6.1.3 Futtatás

Futtatás pedig a **java berenyi_kft.Skeleton** paracsal hajtható végre, amely hatására programunk el is indul.

```
C:\Users\cloud\berenyi_kft_hazi\berenyi_kft\src>java berenyi_kft.Skeleton
Welcome in berenyi_kft's Skeleton tester!

Please choose from our test cases:

1: Win game
2: Move Spaceship
3: Drill Asteroid
4: Restore Resource
5: Mine Resource
6: Build AI Robot
7: Build TeleportingGatePair
8: Place TeleportingGate
9: Call Sunstorm
10: Quit
My choice:■
```

Amennyiben **Eclipse**-ben nyitják meg a programot, a projekt a **JDK 11**-et használja, azért, hogy a virtuális gépen futtathatóvá váljon.

6.2 Értékelés

Tag neve	Tag neptun	Munka százalékban
Havasi		20
Kreták		20
Mesterházi		20
Radnai		20
Szögi		20

6.3 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2021.03.16. 18:00	2,5 óra	Havasi	Tevékenység: Git kezelése, osztályok alapvető megvalósítása.
2021.03.17. 14:00	3 óra	Havasi Kreták Mesterházi Radnai Szögi	Tevékenység: A szétosztott osztályainak függvényeinek megvalósítása
2021.03.17. 14:00	3 óra	Havasi Kreták Mesterházi Radnai Szögi	Értekezlet: Teszesetek megbeszélése, elkészítési módjának meghatározása, megbeszélése, tesztek kiosztása
2021.03.19. 20:00	3 óra	Havasi Kreták Mesterházi Radnai Szögi	Tevékenység: kiosztott tesztek megírása a Skeleton függvényben.
2021.03.21. 14:00	2 óra	Havasi Kreták Mesterházi Radnai Szögi	Értekezlet: Tesztek, merge conflictok átbeszélése, végső tesztek kipróbálása.
2021.03.21. 20:00	3 óra	Radnai	Tevékenység: Utolsó simítások végzése, kommentek, tesztek ellenőrzése

7. Prototípus koncepciója

7 – berenyi_kft

Konzulens:

Csapattagok

Kreták Barnabás András

Mesterházi Marcell

Szögi Richárd

Havasi Balázs

Radnai Bálint

2021.03.31

7. Prototípus koncepciója

7.0 Változás hatása a modellre

Az Uranium osztály egy privát egész típusú *life* attribútummal bővült, amelynek értéke kezdetben 3, és az urán minden napközeli felszínre kerülésekor (kifúrás vagy nyersanyag-visszatöltés esetén) 1-gel csökken (a decLife() metódussal). Ez az attribútum az adott urán egység aktuális stabilitását fejezi ki. Az aszteroidáját akkor robbantja, ha a life attribútum 0-ra csökkent, vagyis ha harmadjára kerül felszínre napközelben.

A TeleportingGate osztály ezentúl megvalósítja az ISteppable interfészét, és bővült a goMad(), step() és move(int d) függvényekkel. Ha eléri a napvíhar azt az aszteroidát, amely körül kering, akkor megkerül (goMad()), léptethető lesz (Timer::addSteppable()), és minden lépésben (step()) egy véletlen szomszédos aszteroida körüli pályára áll át (move(int d)).

A Settler 3 elkészült teleportkaput is tárolhat (gatesCreated), így mostantól akkor építhet kapupárt (createGatePair()), ha legfeljebb 2 elkészült teleportkapu van nála.

A napvíhar ezentúl nem az összes aszteroidát éri el, hanem csak a Nap közvetlen szomszédait (neighboringAsteroids), és azok szomszédait, vagyis a másodszomszédokat (Asteroid::getNeighbors()). A napközelséget pedig eddig is a Naphoz képesti legfeljebb másodszomszédsággal modellezük.

Módosítottunk az Asteroid osztály getNeighbors() és getNeighbor(int d) metódusain. A teleportkapuk általi ideiglenes szomszédságokat innentől minden lépésben újraszámoljuk; ehhez felhasználjuk a neighbors kollekciót és a gates teleportkapuk általi szomszédokat. A getNeighbor(int d) függvény a getNeighbors() függvény által visszaadott, aktuális szomszédokat tartalmazó listából adja vissza a modulo d-edik elemet.

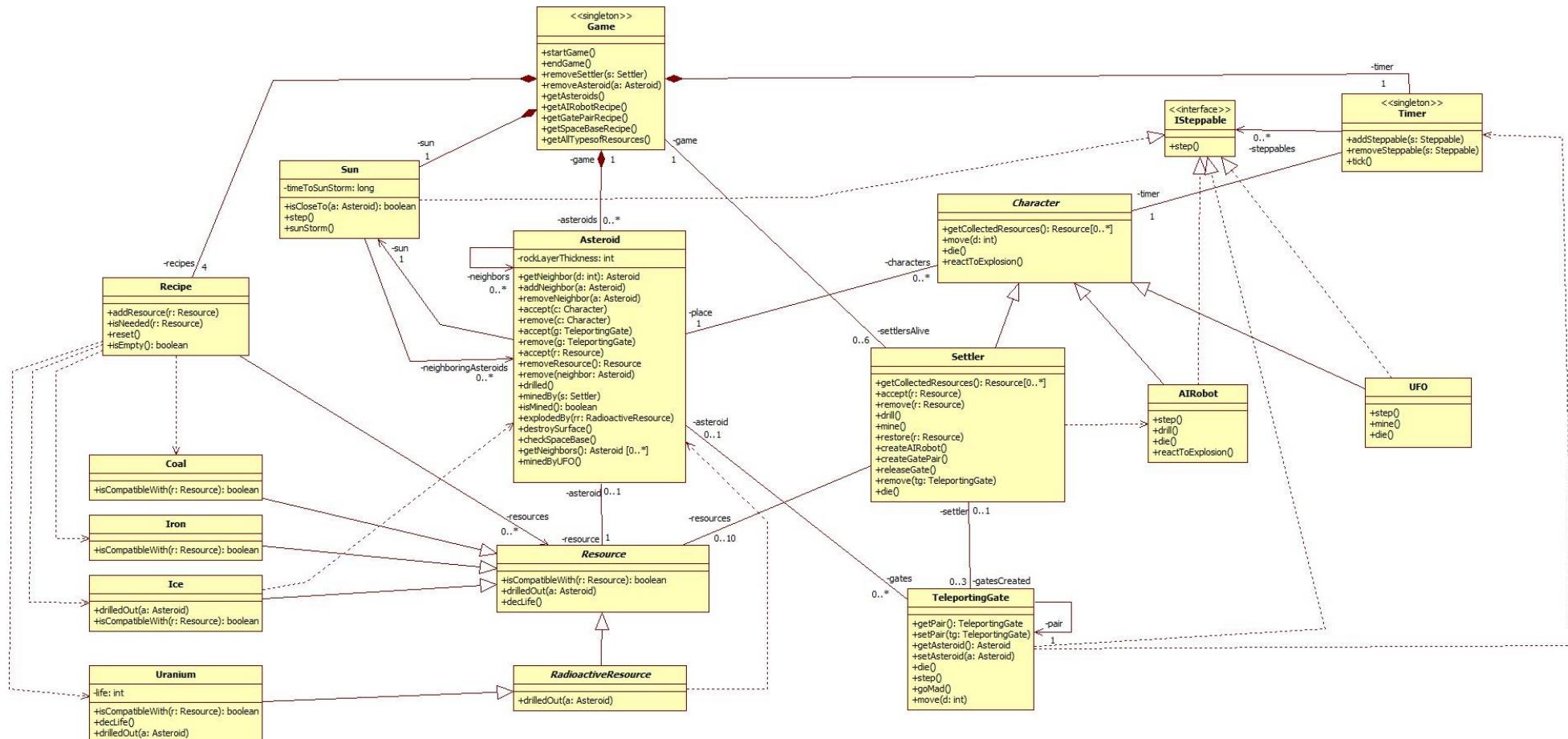
A Timer osztályt a beépített java.util.Timer osztályból származtattuk. A konstruktora Timer(long delay, long period) típusú lett, amelyben meg kell adnunk a kezdeti késleltetést, majd az időzített periódusok hosszát ezredmásodpercekben. Az osztály egy Timertask típusú belső osztályt is tartalmaz, amely az időzített eseményekért felel, és amelynek run() függvénye saját maga tick() függvényét hívja. A tick() metódus továbbra is léptet minden léptethető objektumot (steppables). A privát long típusú ticks adattagban számlálja is a játékban eltelt időt.

A modellt bővítettük az UFO osztállyal. Az UFO-k is, képesek aszteroidák közötti mozgásra, tudnak bányászni, de nem tudnak fújni. Az UFO-t is a Character ősosztályból származtatjuk, és annak move(int d) függvényét örökli.

Megvalósítja az ISteppable interfészét. step() függvényében megpróbál bányászni, majd akár sikerül, akár nem, egy véletlenszerű szomszéd aszteroidára lép. A kibányászott nyersanyagra később sehol nem lesz szükség, ezért az UFO a modellünkben nem tárol nyersanyagot, csak eltávolítani tudja az aszteroida magjából.

A die() függvényében nemcsak az aszteroidáról távolítja el magát, hanem a Timer steppables listájából is. A drill() és a mine() függvényeket a háromfajta karakter közül már csak kettő-kettő valósítja meg, ezért kikerütek a Character osztályból. Úgy döntöttünk, hogy az UFO-k a radioaktív robbanásra és a napvíharra is reagáljanak ugyanúgy, mint a Settler-ek, vagyis a die() függvényük hívódik ezekben a helyzetekben.

7.0.1 Módosult osztálydiagram



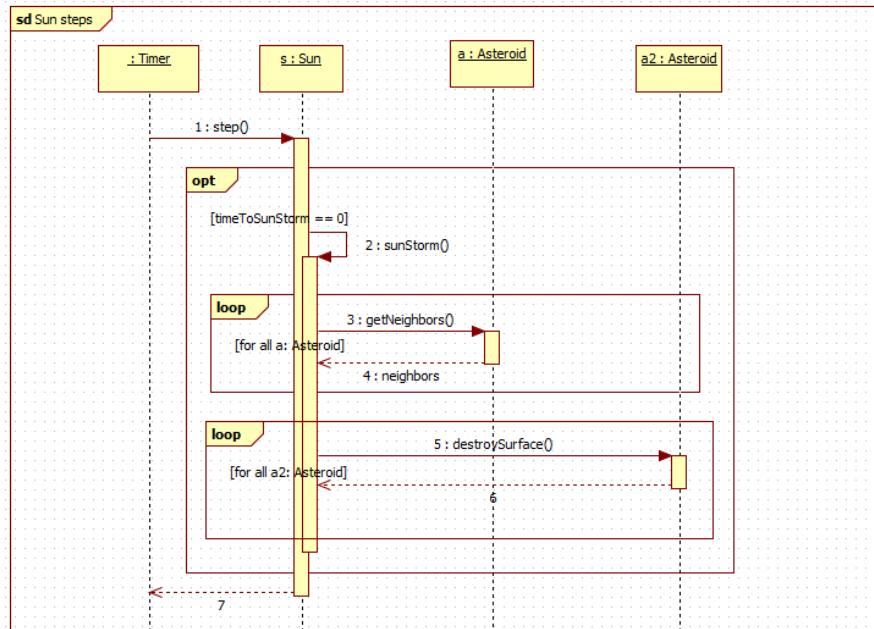
7.0.2 Új vagy megváltozó metódusok

- **void Uranium::decLife()**: az Uranium life adattagjának értékét egyelőre csökkenti.
- **void UFO::step()**: Meghívja az UFO mine() és move() metódusát.
- **void UFO::die()**: Az UFO megsemmisül: eltávolítja magát a léptethető objektumok listájából.
- **void UFO::mine()**: Az UFO megpróbálja kibányászni az adott aszteroida magjában található nyersanyagot. Ehhez meghívja az aszteroida minedByUFO() metódusát.
- **void TeleportingGate::step()**: Meghívja a teleportkapu move() metódusát. Ez csak már megkerült teleportkapuknál következik be.
- **TeleportingGate::goMad()**: Hozzáadja a timer steppables kollekciójához a teleportkaput, ami ezáltal megkerül.
- **void TeleportingGate::move(int d)**: Az aktuális aszteroida egy szomszédjára vándorol át, ehhez elvégzi a régi kapcsolat bontását és az újak felépítését is.
- **Settler::createGatePair()**: A telep leskérdezi a Game-tól a teleportkapu-pár megépítéséhez szükséges receptet a Recipe getGatePairRecipe() metódusának meghívásával, majd pedig összehasonlíta a saját nyersanyagait a receptben lévőkkel. **Amennyiben rendelkezik a szükséges nyersanyagokkal, és maximum 2 darab elkészített teleportkaput tárol**, akkor létrehozza a TeleportingGate két példányát. Párba állítja őket a setPair(tg: TeleportingGate) függvények meghívásával, majd az űrhajóján a gatesCreated kollekcióban eltárolja őket. Végül meghívja a Recipe reset() függvényét, amelyben ezáltal visszaállítja a recept listájának a tartalmát.
- **void Asteroid::destroySurface()**: Az aszteroidán végigsöpör a napvihar. Ha az aszteroida nincs megfúrva (rockLayerThickness > 0), vagy a mag nem üreges (getResource() ≠ null), akkor a függvény meghívja az aszteroidán tartózkodó karakterek die() függvényét, **továbbá minden esetben meghívja az aszteroidához kapcsolt teleportkapuk goMad()** metódusát.
- **Asteroid Asteroid::getNeighbor(d: int)**: Visszaadja az aszteroida d-edik szomszédját: A neighbors kollekcióból és a teleportkapuk által elért aszteroidákból kollekciót képez, majd ebből a kollekcióból adja vissza a d-edik elemet, modulo a kollekció elemszáma szerint.
- **void Asteroid::minedByUFO()**: Ha az aszteroida rétegvastagsága nulla és tartalmaz nyeranyagot, akkor meghívja a removeResource() metódusát.
- **void Sun::sunStorm()**: Lekérdezi a Nappal szomszédos aszteroidákat és azoknak is a szomszédait, majd egyesével meghívja mindegyik aszteroida destroySurface() metódusát.

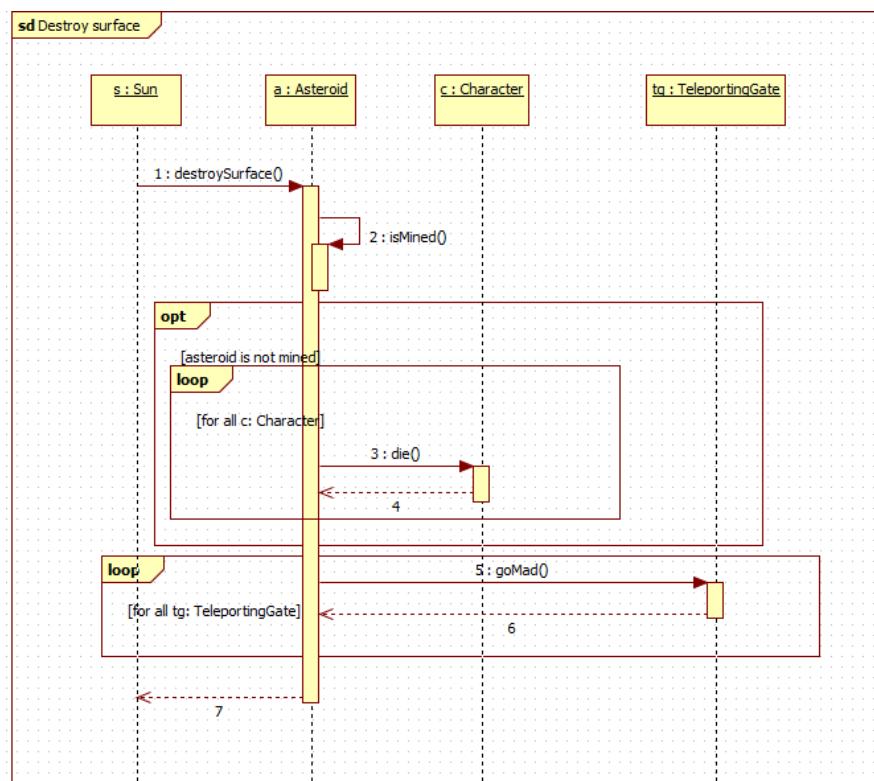
7.0.3 Szekvencia-diagramok

7.0.3.1 Sun steps

A step() metódusban létrejön egy aszteroidákat tároló kollekció, amelyhez hozzáadódnak a Sun szomszédos aszteroidái és az azokkal szomszédos aszteroidák. Ezen kollekció elemein hívódik meg a destroySurface() metódus. A szekvenciadiagramon a gyűjtemény létrehozását és az aszteroidák hozzáadását nem lehet megjeleníteni, ezért szükséges a magyarázat.

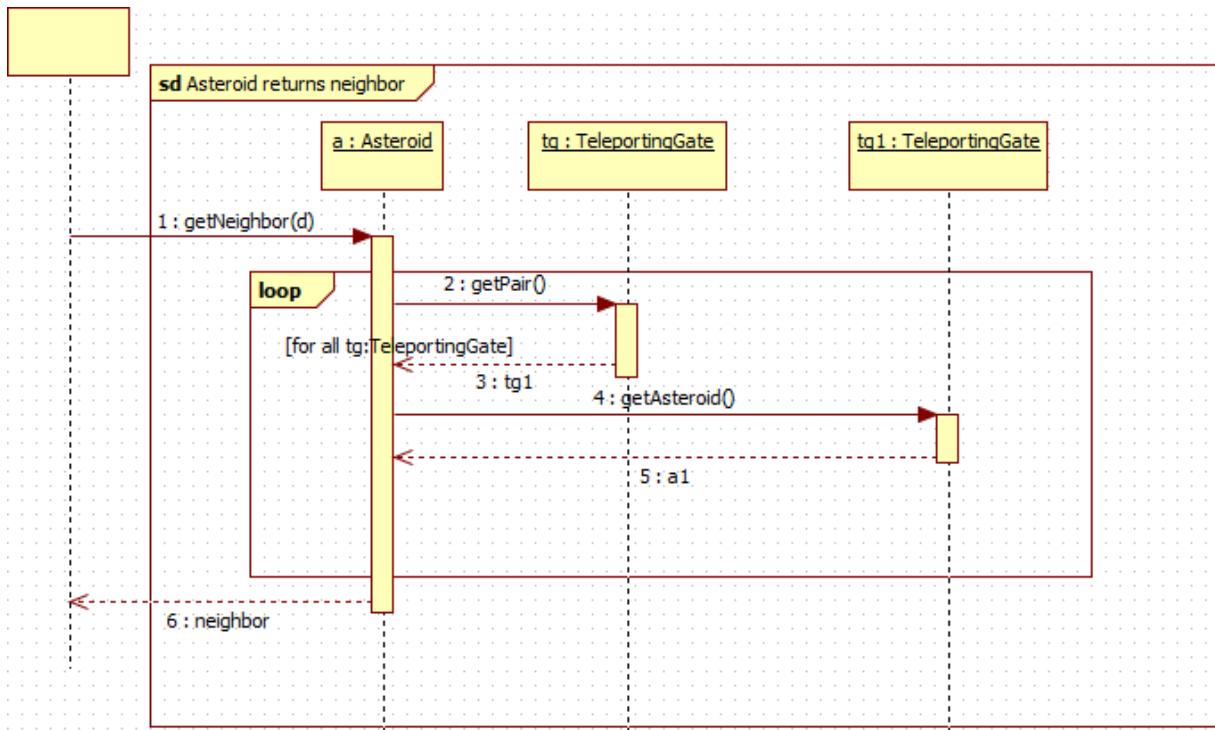


7.0.3.2 Destroy surface

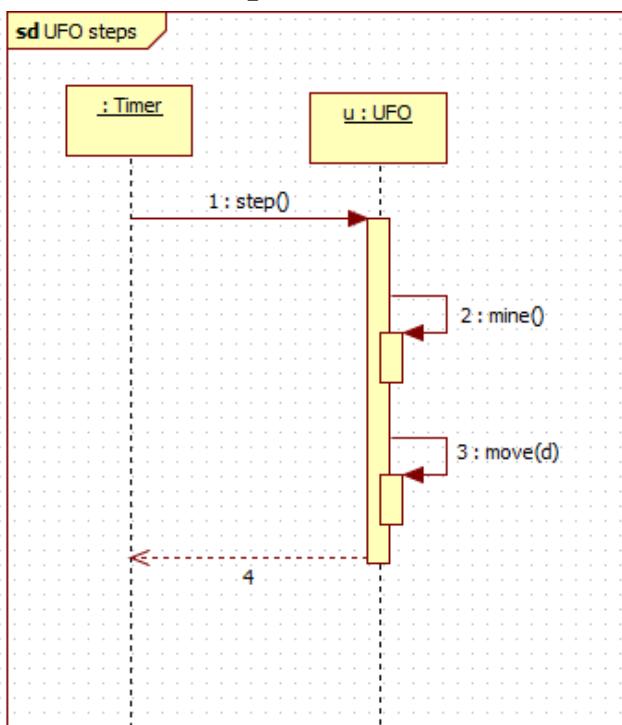


7.0.3.3 Asteroid returns neighbor

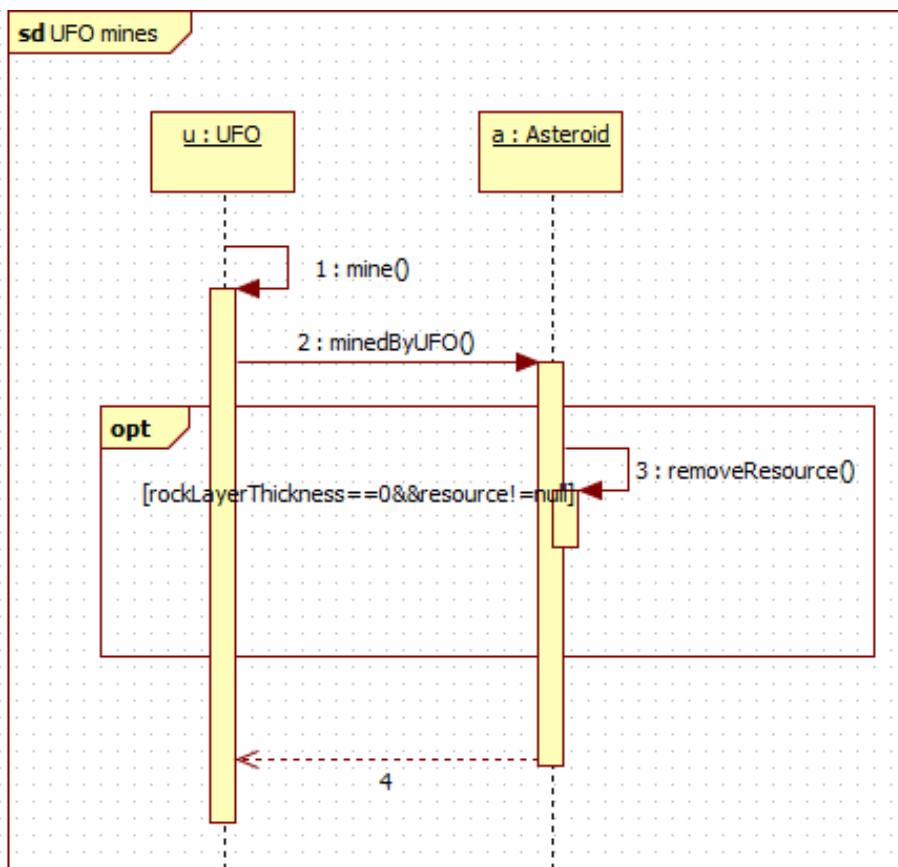
A getNeighbor(d: int) metódusban létrejön egy aszteroidákat tároló kollekció, amelyhez hozzáadónak az aszteroidán lévő teleportkapuk párainak az aszteroidái és az aszteroida szomszédos aszteroidái. Ennek a gyűjteménynek a modulo d-edik elemével tér vissza a függvény. Ezt a szekvenciadiagramon nem lehet kifejezően ábrázolni, ezért szükséges a magyarázat.



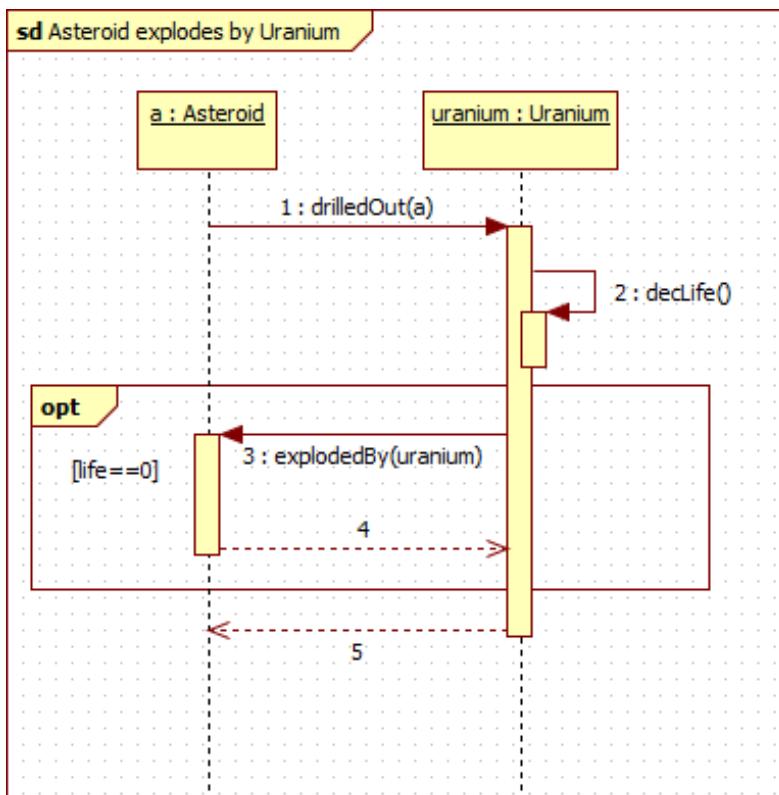
7.0.3.4 UFO steps



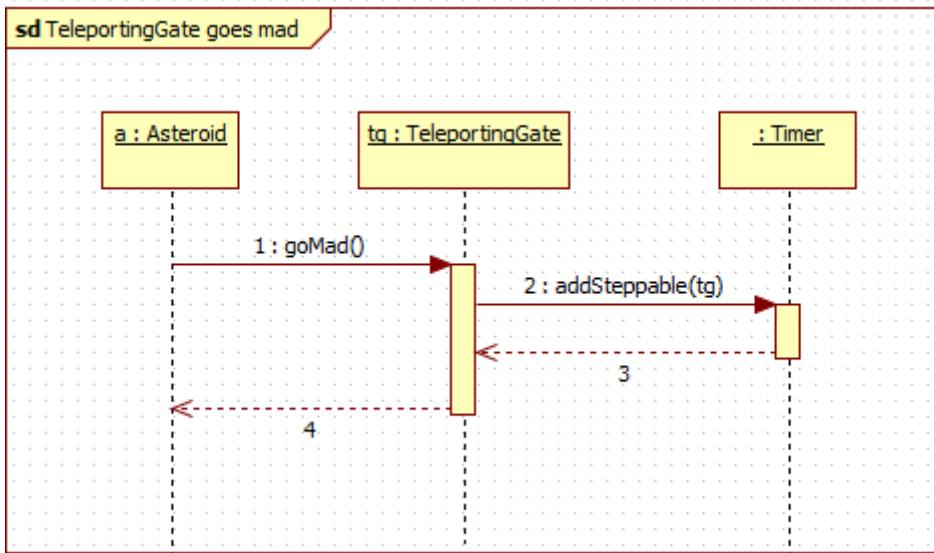
7.0.3.5 UFO mines



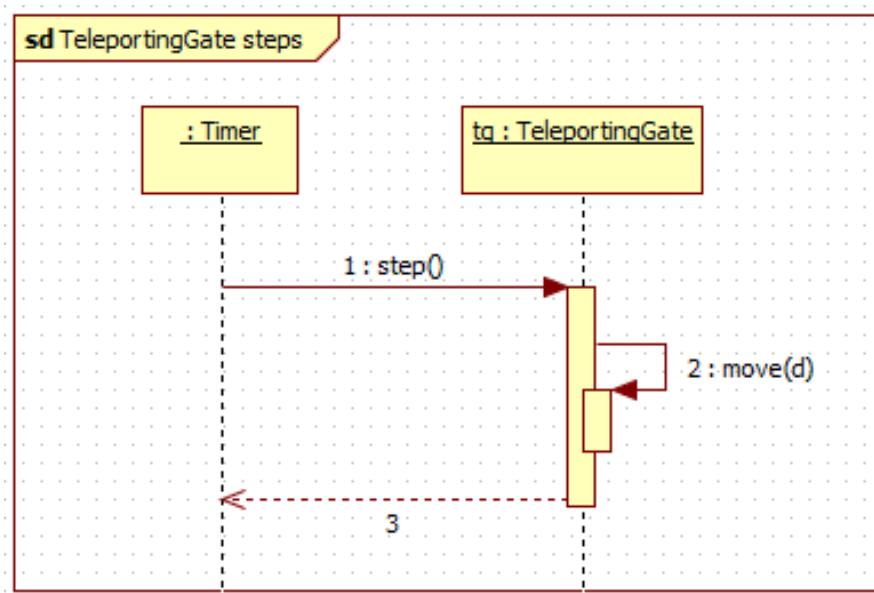
7.0.3.6 Asteroid explodes by Uranium



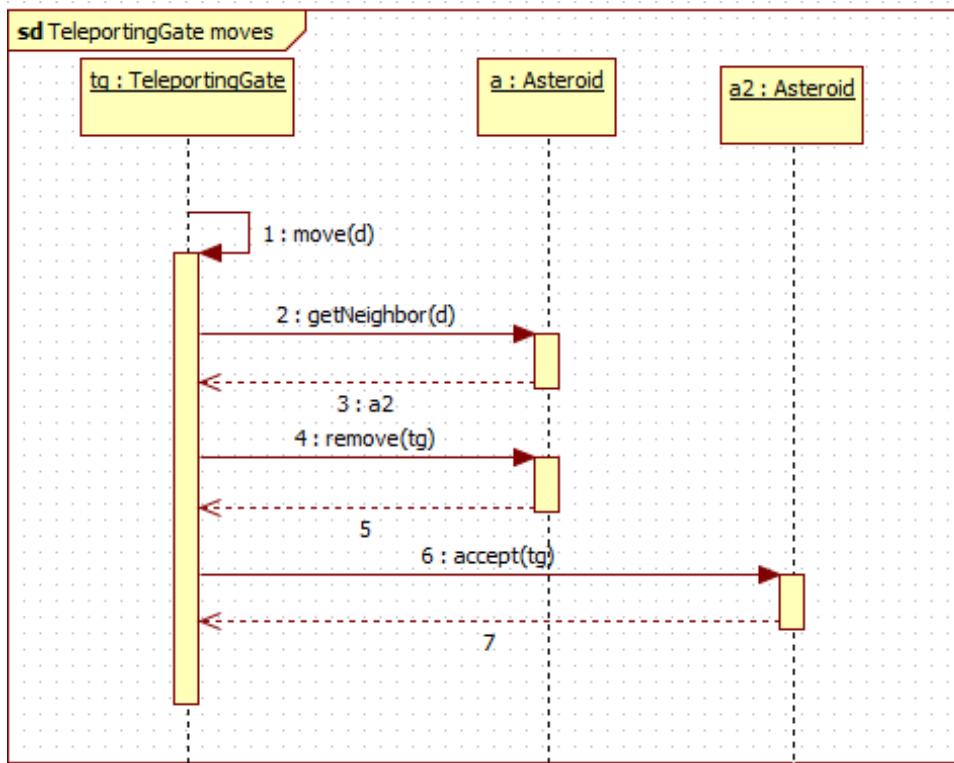
7.0.3.7 TeleportingGate goes mad



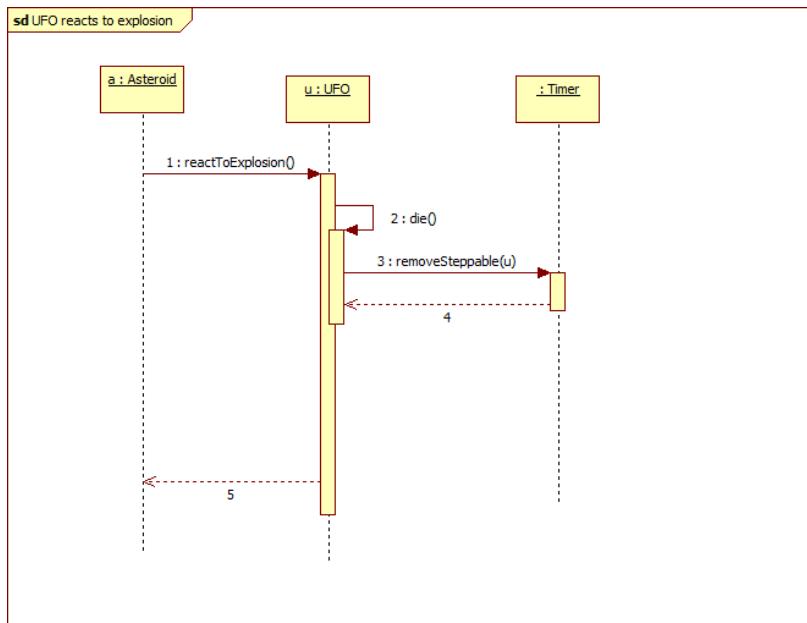
7.0.3.8 TeleportingGate steps



7.0.3.9 TeleportingGate moves



7.0.3.10 UFO reacts to explosion



7.1 Prototípus interface-definíciója

7.1.1 Az interfész általános leírása

A prototípus program a konzolos standard bemeneten, karakteres parancsokkal irányítható. Használható arra, hogy élő játékban próbáljuk ki a működését, de támogatja a programozott tesztelést is. A felhasználó a parancsokat soronként írhatja a konzolra, és a parancsértelmezőnk is soronként dolgozza fel a beírt utasításokat. A használható parancsokat, a bemenet-kimenet formátumát és a pálya konfigurációs nyelvtanát a 7.2.2 és a 7.2.3 pontban részletezzük.

Több parancs argumentumokat is vár, ezekkel paraméterezhetőek. A parancsok nevei és argumentumai csak nyomtatható, nem whitespace karaktereket tartalmazhatnak. A parancs nevét és az argumentumokat egy-egy szóköznek kell elválasztani a bemeneten. A parancs hatása Enter (új sor) beütése esetén érvényesül.

A begépelt parancsok általános felépítése:

<parancs> <argumentum1> <argumentum2> ... <argumentumN>

Egyes parancsoknál az utolsó paraméter elhagyható, viszont a paraméterek sorrendje mindenhol rögzített.

Lehetőség van teszteseteket fájlban megadni, ilyenkor a program a batch jellegű fájlban megadott utasításokat sorban végrehajtja. A játék konfigurációját a program bemeneti fájlban is megkaphatja, nem csak véletlenszerű pályát inicializáló parancssal.

A legtöbb parancs valamilyen változtatást végez a modellen vagy a tesztelés beállításaiban, és a képernyőre nem ír semmit. A játék konfigurációján vagy állapotán értjük az adott pillanatban a játék objektumai összességét, azok attribútumainak és referenciáinak értékeivel. Rendelkezésre áll olyan parancs, amellyel a standard kimeneten megjeleníthető a játék teljes állapota, vagy a játék egy-egy objektumának az állapota. A játékállapot fájlba is menthető, a tesztelést segítőd.

A prototípus véletlenszerű működést is tartalmaz. Egy kapcsoló parancssal válthatunk véletlenszerű és determinisztikus működés között. Determinisztikus működés esetén a modell szabályainak nem ellentmondó konkrét viselkedést implementálunk.

7.1.2 Bemeneti nyelv

7.1.2.1 Rendszerutasítások

Ezekkel a parancsokkal a játék elindítható, illetve vezérelhető a tesztkörnyezet és a tesztelés. A prototípus programba kezdetben nincs játék betöltve, ezt az *init* vagy a *load* parancssal lehet megtenni. Innentől az így betöltött játék konfigurációt módosít minden egyéb parancs. A betöltő parancsok újbóli kiadásával új játék konfiguráció írja felül az addigit.

A *random* parancssal beállítható, hogy a játék inicializálása és futása kötelezően determinisztikus legyen-e. Hatással van az *init* parancs eredményére.

A játék a *start/stop* parancsokkal indítható, illetve szüneteltethető.

A *show/save* parancsok a standard kimenetre, illetve fájlba mentik az aktuális játékállapotot.

A *run_script* parancssal előre összekészített parancsokat tartalmazó tesztfájl futtatható.

Megváltoztathatja a betöltött modellt, és kiadhatja ugyanezeket a parancsokat is, kivéve magát a *run_script* parancsot (hogy elkerüljük a rekurzív hívásokból származó esetleges memóriarobbanást).

exit

Leírás: Terminálja és bezárja az alkalmazást. Nem végez mentést.

Opciók: exit (nincs paramétere)

random

Leírás: Beállítja, hogy a pálya inicializálásakor és a játék során történhetnek-e pszeudovéletlen események (*is_random*=”true”), vagy pedig determinisztikus lefutású legyen a program (*is_random*=”false”).

Opciók: random <*is_random*>

log

Leírás: Beállítja, hogy a játék futtatása során a meghívott függvények naplózódjanak-e a konzolon (*is_logging*=”true”, különben *is_logging*=”false”). A naplózás során indentálva megjelennek a meghívott függvények (a beágyazott meghívott függvények eggyel beljebb tabulálva), ezáltal nyomonkövethető a program futása; a teszteléshez használható.

Opciók: log <*is_logging*>

init

Leírás: Inicializál egy új játékot, és beállítja az addig tesztelt játék helyett. Az inicializált pálya a *random* beállítástól függően tartalmaz véletlenszerűen sorsolt komponenseket.

Opciók: init

load

Leírás: Betölti a paraméterként megadott fájlban található pályakonfigurációt (beleértve az időzítő állapotát, a játékosokat, és azt is, hogy ki a soron következő játékos). Ha volt már betöltött tesztelt játék, ez a parancs felülírja.

Opciók: load <file>

start

Leírás: Elindítja a betöltött játék futását. Szünetelő tesztjátékon hívva, a futás a szüneteltetéskori állapot alapján folytatódik, és az időzítő is tovább számol.

Opciók: start (nincs paramétere)

stop

Leírás: Szünetelteti a játék futását, nem változtatva a pálya aktuális állapotán. Megállítja az időzítőt (így a léptethető objektumokat is), és a játékosok sem léphetnek. Használható adott pillanatbeli pillanatkép készítéséhez (*show*, *save*).

Opciók: stop (nincs paramétere)

show

Leírás: Megjeleníti az *object_name* azonosító nevű objektum adatait a standard kimeneten. A kimenet formátuma megfelel a konfigurációs fájlok nyelvtanában szereplő mintáknak. Paraméter nélkül is hívható, ekkor a teljes játékállapotot kiírja a standard kimenetre.

Opciók: show <object_name>

save

Leírás: A paraméterben megadott fájlba menti a tesztelt pálya aktuális állapotát, vagyis pillanatképet készít (és nem módosítja a betöltött pályát).

Opciók: save <file>

run_script

Leírás: A paraméterben megadott batch jellegű fájlban szereplő utasításokat sorban végrehajtja. A megadott *file* csak az ebben a fejezetben felsorolt parancsokat tartalmazhatja, újabb *run_script* utasítást nem. A *file* tartalmazhat “//”-rel kezdődő sorokat, ezeket a program nem dolgozza fel a futtatás során.

Opciók: run_script <file>

7.1.2.2 Játékutasítások

Ezeket a mindenkorai soron következő játékos írhatja be, és a játékos által vezérelt telepes által hajtódnak végre a modellben. A játékos use-caseit valósítják meg. A játék konkrét működését vizsgálhatjuk általuk (akár tesztelési szándék nélkül).

pass

Leírás: A játékos passzol, ennek hatására a soron következő játékos körébe lép a végrehajtás.

Opciók: pass (nincs paramétere)

move

Leírás: A játékos által irányított telepes a *direction* irányban levő aszteroidára mozog az űrhajójával. A szomszédos aszteroidák között ott szerepelnek a teleportkapuk általi szomszédok is.

Opciók: move <direction>

A *direction* paraméter egy természetes szám, amely modulo adja meg, hogy az aszteroidája hányadik szomszédra lépjen tovább a telepes.

drill

Leírás: A játékos telepese fúrja az aszteroidát, amelyen tartózkodik. Hatására az aszteroida köpenyvastagsága 1-gyel csökken, feltéve, hogy pozitív volt.

Opciók: drill (nincs paramétere)

mine

Leírás: A játékos telepese kibányássza az aszteroidájában található nyersanyagot, feltéve, hogy van hely az ūrhajóján, és az aszteroida köpenyvastagsága 0.

Opciók: *mine* (nincs paramétere)

restore

Leírás: A játékos telepese visszatölti a *resource* nyersanyagot az aszteroidája magjába, feltéve, hogy az aszteroida meg van fúrva, és a magja üreges.

Opciók: *restore <resource>*

A *resource* egy, a telepesnél található nyersanyagegység azonosító neve.

create_robot

Leírás: A játékos telepese robotépítésbe fog. Ha rendelkezik az építéshez szükséges nyersanyagokkal, akkor létrehoz, és az aszteroidáján útjára enged egy új (léptethető) AIRobotot.

Opciók: *create_robot* (nincs paramétere)

create_gate_pair

Leírás: A játékos telepese teleportkapupár építésébe kezd. Ha rendelkezik a szükséges nyersanyagokkal, és van legalább két kapunak szabad hely az ūrhajóján, akkor létrehoz két kaput. Beállítja őket egymás pároként, és magánál eltárolja őket.

Opciók: *create_gate_pair* (nincs paramétere)

release_gate

Leírás: A játékos telepese az aktuális aszteroidája körüli pályára állítja a *teleporting_gate* kaput.

Opciók: *release_gate <teleporting_gate>*

A *teleporting_gate* egy, a telepesnél található teleportkapu azonosító neve.

7.1.2.3 A játékállapotot leíró konfigurációs fájl formátuma

A proto program *save*, *load* és *show* parancsai hívásakor ennek a mintának megfelelő fájlformátumot használunk. A fájlban egy játék objektumainak adott pillanatbeli állapota tárolódik az összes attribútummal és referenciával. A fájl használható bementként, mint konfigurációs fájl, illetve kimenetként, mint lenyomat; tulajdonképpen egy manuális sorosítás formátuma.

A fájl csak nyomtatható ASCII karaktereket tartalmazhat. A "://" sztringgel kezdődő sorok kommentek. A fájl ilyen módon metainformációval kiegészíthető, ezeket a kommenteket a modell feldolgozása során nem vesszük figyelembe.

A fájl első része a deklarációs blokk, amely felsorolja az adott játék összes konkrét objektumát típusazonosító névvel és egyedi azonosító névvel. A típus a példányosítandó osztály neve kell, hogy legyen; absztrakt osztály és interfész nem szerepelhet típusként. Az azonos típusú objektumokat egy sorban, az osztály neve után soroljuk fel, egy-egy szóközzel elválasztva (pl. „Settler s1 s2”). Az első részt üres sor követi; illetve az eddig felsorolt sorokat nem tagolhatják üres sorok.

A fájl második részében felsoroljuk az egyes objektumokat az összes attribútumukkal és referenciájukkal (láthatóságtól függetlenül). Ha egy objektumnak nincs attribútuma, illetve referenciája, akkor nem jelenítjük meg a felsorolásban. Az egyes objektumok konfigurációit üres sorokkal tagoljuk. Egy objektum attribútumnevei előtt tabulátor karakter szerepel, az attribútum értéke(i) egy szóközt követően következnek a sorában. A kollekció adattagok elemeit is egy-egy szóköz választja el egymástól.

Az attribútumok értékei:

- A primitív típusú attribútumok esetén azok konkrét értékei jelennek meg a fájlban (pl. „rockLayerThickness 5”).
- Referencia adattagok értékét a fájl első részében is szereplő azonosító nevekkel adjuk meg. A referenciai értéke lehet „null”, ha nem mutatnak semelyik létező objektumra. A referencia adattagok heterogén kollekcióra is mutathatnak. (Példák: „timer null”, „characters s1 s2 air1 ufo1”.)

A fájl első részén belül, illetve a második részén belül is a fentiek szerinti kell, hogy legyen a sorok sorrendje. A sorokon belül az egyes kollekciók elemeinek sorrendje is számít. Ez a generált kimeneti fájlok egyszerű összehasonlíthatósága miatt lényeges.

A következőkben látható egy minta fájl, amely bemutat egy teljes játékot leíró konfigurációt, megjelenítve az objektumok összes lehetséges kapcsolatát. A fájl első része a deklarációs blokk. A második rész – az objektumok listája leírásokkal – a következő oldalon látható, a tömörseg kedvéért három hasábos elrendezésben, csökkentett betűmérettel.

```

Controller controller
Player p1 p2
Game game
Recipe robotRecipe, gatePairRecipe, spaceBaserecipe, allResourcesRecipe
Timer timer
Sun sun
Asteroid a1 a2 a3 a4
Coal co1 co2 co3 co4 co5 co6 co7 co8
Iron ir1 ir2 ir3 ir4 ir5 ir6 ir7 ir8 ir9 ir10
Ice ic1 ic2 ic3 ic4 ic5 ic6 ic7
Uranium ur1 ur2 ur3 ur4 ur5 ur6 ur7 ur8 ur9
Settler s1 s2
AIRobot air1
UFO ufo1
TeleportingGate tg1 tg2 tg3 tg4

```

(A minta fájl második része (következő oldalon található): az objektumdiagram szöveges formában. A Game recipes attribútuma és a spaceBaseRecipe resources attribútuma csak több sorba fér ki, de valójában csak egy-egy sort foglalnak el a fájlban.)

Controller	Asteroid a2 game game sun sun neighbors a1 a4 resource co6 characters s2 gates null	Settler s1 timer timer game game place a1 resources co7 co8 ir9 ic5 ur8 gatesCreated null
Player p1 name Bela settler s1 isAlive true	Asteroid a3 game game sun sun neighbors a1 a4 resource ur7 characters air1 gates null	Settler s2 game game timer timer place a2 resources ir10 ic6 ic7 ur9 gatesCreated tg4
Player p2 name Eugen settler s2 isAlive true	Asteroid a4 game game sun sun neighbors a2 a3 resource ir8 characters null gates tg2	AIRobot air1 timer timer place a4
Game game timer timer recipes robotRecipe gatePairRecipe spaceBaserecipe allResourcesRecipe sun sun asteroids a1 a2 a3 a4 settlersAlive s1 s2	Uranium ur1 lives 3	UFO ufo1 timer timer
Recipe robotRecipe resources co1 ir1 ur1	Uranium ur2 lives 3	TeleportingGate tg1 timer timer pair tg2 settler null asteroid a1
Recipe gatePairRecipe resources ir2 ir3 ic1 ur2	Uranium ur3 lives 3	TeleportingGate tg2 timer null pair tg1 settler null asteroid a4
Recipe spaceBaserecipe resources co2 co3 co4 ic2 ic3 ic4 ir4 ir5 ir6 ur3 ur4 ur5	Uranium ur4 lives 3	TeleportingGate tg3 timer null pair tg4 settler null asteroid a1
Recipe allResourcesRecipe resources co5 ic5 ir7 ur6	Uranium ur5 lives 3	TeleportingGate tg4 timer null pair tg3 settler s2 asteroid null
Timer timer ticks 36 delay 1000 period 500 steppables sun air1 ufo1 tg1	Uranium ur6 lives 3	
Sun sun timeToSunStorm 3000 neighboringAsteroids a1 a2	Uranium ur7 lives 1	
Asteroid a1 game game sun sun neighbors a2 a3 resource null characters s1 gates tg1 tg3	Uranium ur8 lives 3	
	Uranium ur9 lives 3	

7.1.3 Kimeneti nyelv

A *show <object>* parancs kiírja a standard kimenetre az *object* objektum (példányosítható) típusának nevét (*class*), az objektum azonosító nevét (*object*), majd soronként az attribútumai neveit és értékeit. A formátum megfelel a konfigurációs fájl második részében szereplő, egy-egy objektumot leíró rész formátumának. Az attribútumokat leíró sorok tabulátorral kezdődnek, majd a sor többi azonosítóját egy-egy szóköz választja el:

```
<class> <object>
<tab><attr1> <attr1_val>
<tab><attr1> <attr1_val1> <attr1_val2> ...
...
```

A paraméter nélküli *show* parancs kimenete a képernyőre íródik, de megegyezik a *save <file>* parancs kimeneti fájljába írandó adatokkal. A kimenet formátuma az előző, bemeneti nyelvet taglaló bekezdésben szerepel.

A többi parancsnak helyes működés esetén nincs kimenete sem a konzolon, sem fájlban. Kimenetet csak hiba esetén adnak, ilyenkor egy hibaüzenet jelenik meg a konzolon, amely röviden leírja a hibát.

A logolás formája, hogy a legfontosabb hívott függvényhívásokat kiírjuk a kimenetre. Ha visszatérnek, akkor azt is, hogy sikerült-e, illetve mi az oka, ha nem (röviden).

7.2 Összes részletes use-case

7.2.1 Pass

Use-case neve	Pass
Rövid leírás	A játékos dönthet úgy, hogy nem végez érdemi lépést, csak passzol.
Aktorok	Player
Forgatókönyv	A játékos a telepessel ilyenkor nem végez érdemi lépést.

7.2.2 Move Spaceship

Use-case neve	Move Spaceship
Rövid leírás	A játékos képes a telepes űrhajójával egy másik szomszédos aszteroidára lépni.
Aktorok	Player
Forgatókönyv	1. A játékos az aktuális aszteroidáról egy szomszédos aszteroidára mozog.
Alternatív forgatókönyv	1.A.1. A játékos az aktuális aszteroidáról teleportkapuk segítségével aszteroidára mozog.
Alternatív forgatókönyv	1.A.2. A játékos nem tudja használni a teleportkaput, mert a kapu párja nincs lerakva.
Alternatív forgatókönyv	1.B.1. A játékos nem tud lépni, mert nincs már szomszédja.
Alternatív forgatókönyv	2. A mozgás után az aszteroidán tartózkodó játékosoknál elegendő nyersanyag lett az ürbázis építéséhez.

7.2.3 Drill Asteroid

Use-case neve	Drill Asteroid
Rövid leírás	A játékos egy egységet lefűr az aszteroidán.
Aktorok	Player
Forgatókönyv	1. A játékos egy egységnit fűr az aszteroidán, amin épp tartózkodik.
Alternatív forgatókönyv	1.A.1. A játékos napközelben fűr.
Alternatív forgatókönyv	1.B.1. A játékos nem tud fűrni, mert az aszteroida ki van már fűrva.
Alternatív forgatókönyv	1.A.1.A.1. A játékos napközelben kifűr egy olyan aszteroidát, aminek a közepén Vízjég van, ami ilyenkor elszublimál.
Alternatív forgatókönyv	1.A.1.B.1. A játékos napközelben kifűr egy olyan aszteroidát, aminek a közepén Urán van, ami ilyenkor felszínre kerül, és veszít a stabilitásából.

7.2.4 Restore Resource

Use-case neve	Restore Resource
Rövid leírás	A játékos nyersanyagot helyez egy üres aszteroida belsejébe.
Aktorok	Player
Forgatókönyv	1. A játékos az űrhajón tárolt egy egységni nyersanyagot az aszteroida üres magjába tesz.
Alternatív forgatókönyv	1.A.1. A játékos nem tudja letenni a nyersanyagot, mert az aszteroida nem üres.
Alternatív forgatókönyv	1.A.2. A játékos nem tudja letenni a nyersanyagot, mert üres a tárolója.
Alternatív forgatókönyv	1.B.1. A játékos napközelben tölt vissza nyersanyagot.
Alternatív forgatókönyv	1.B.1.A.1. A játékos napközelben egy Vízjeget tesz vissza, ami ilyenkor elszublimál.
Alternatív forgatókönyv	1.B.1.B.1. A játékos napközelben egy Uránt tesz vissza, ami ilyenkor felszínre kerül és veszít a stabilitásából.
Alternatív forgatókönyv	1.B.1.B.2. A játékos napközelben egy Uránt tesz vissza, ami ilyenkor felszínre kerül és már annyira instabil, hogy felrobban.

7.2.5 Mine Resource

Use-case neve	Mine Resource
Rövid leírás	A játékos kinyeri az erőforrást az aszteroida magjából.
Aktorok	Player
Forgatókönyv	1. A játékos kinyeri a fűrt lyukon keresztül az aszteroida magjában található erőforrást, és az űrhajóján tárolja.
Alternatív forgatókönyv	1.A.1. A telepes nem tudja kinyerni a magból a nyersanyagot, mert az űrhajó tárolója tele van, nem tud többet beletenni.
Alternatív forgatókönyv	1.A.2. A telepes nem tudja kinyerni a magból a nyersanyagot, mert a köpeny még nem lett átfűrva.
Alternatív forgatókönyv	1.A.3. A telepes nem tudja kinyerni a magból a nyersanyagot, mert a mag üres.
Alternatív forgatókönyv	2. A bányászat után a játékosoknál elegendő nyersanyag lett az ūrbázis építéséhez, ezáltal nyernek.

7.2.6 Build AI Robot

Use-case neve	Build AI Robot
Rövid leírás	A játékos létrehoz egy AI robotot.
Aktorok	Player
Forgatókönyv	1. A játékos létrehoz egy AI robotot, amit a mesterséges intelligencia irányít, és segít neki a fűrészekben.
Alternatív forgatókönyv	1.A.1. A játékosnak nincs elég anyaga létrehozni robotot.

7.2.7 Build TeleportingGate Pair

Use-case neve	Build TeleportingGate Pair
Rövid leírás	A telepes teleportkapu-párt épít.
Aktorok	Player
Forgatókönyv	1. A játékos teleportkapu-párt épít.
Alternatív forgatókönyv	1.A.1. A játékosnak nincs elég anyaga kapupárt létrehozni.
Alternatív forgatókönyv	1.A.2. A játékos nem tud építeni, mert van nála legalább 2 darab teleportkapu.

7.2.8 Place TeleportingGate

Use-case neve	Place TeleportingGate
Rövid leírás	A játékos képes egy aszteroida körüli pályára állít egy teleportkaput.
Aktorok	Player
Forgatókönyv	1. A játékos pályára állít egy teleportkaput.
Alternatív forgatókönyv	1.A.1. A játékos olyan teleportkaput tesz le, amelynek a pálya már pályára állt, így a teleportkapu-pár használhatóvá válik.

7.2.9 UFO steps

Use-case neve	Ufo Steps
Rövid leírás	Az UFO timer hatására lép egyet.
Aktorok	Controller
Forgatókönyv	1. Az UFO mozog egy aszteroidára.
Alternatív forgatókönyv	2. Az UFO kibányássza a nyersanyagot az aszteroidáról, amin épp tartózkodik.

7.2.10 UFO dies

Use-case neve	Ufo dies
Rövid leírás	Az UFO meghal vagy napszél vagy radioaktív robbanás hatására.
Aktorok	Controller
Forgatókönyv	1. Az UFO meghal, eltűnik a pályáról egy radioaktív robbanás hatására.
Alternatív forgatókönyv	2. Az UFO meghal, eltűnik a pályáról napszél hatására.

7.2.11 AIRobot steps

Use-case neve	AIRobot steps
Rövid leírás	Az AIRobot timer hatására lép egyet.
Aktorok	Controller
Forgatókönyv	1. Az AIRobot mozog egy aszteroidára.
Alternatív forgatókönyv	2. Az AIRobot lefűr egy réteget az aszteroida köpenyéről, amin épp tartózkodik.

7.2.12 AIRobot explodes

Use-case neve	AIRobot explodes
Rövid leírás	Az AIRobot meghal napszél hatására vagy ellőkődik szomszéd aszteroidára radioaktív robbanás hatására.
Aktorok	Controller
Forgatókönyv	1. Az AIRobot ellőkődik szomszéd aszteroidára radioaktív robbanás hatására.
Alternatív forgatókönyv	2. Az AIRobot meghal, eltűnik a pályáról napszél hatására.

7.2.13 TeleportGate goes mad

Use-case neve	TeleportGate goes mad
Rövid leírás	Az teleportkapu meghal napszél hatására lépegetni kezd szomszédos aszteroidák között.
Aktorok	Controller
Forgatókönyv	1. A teleportkapu napszélütés hatására mozogni kezd.

7.3 Tesztelési terv

7.3.1 Pass Game

Teszt-eset neve	Pass Game
Rövid leírás	Játékos passzolásának modellezése
Teszt célja	A játékos passzol, telepese nem hajt végre érdemi műveletet.

7.3.2 Win Game with moving

Teszt-eset neve	Win Game with moving
Rövid leírás	Lemodellezük azt a folyamatot, hogy a telepes egy aszteroidára érkezik, és a játék véget ér a telepesek győzelmével, feltételezve, hogy az aszteroidán lévő telepeseknek rendelkezésére áll az űrbázis építéséhez szükséges nyersanyagmennyiség.
Teszt célja	A telepesek megnyerik a játékot.

7.3.3 Win Game with mining

Teszt-eset neve	Win Game with mining
Rövid leírás	Lemodellezük azt a folyamatot, hogy a telepes egy aszteroida nyersanyagát kitermeli, aminek hatására a játék véget ér a telepesek győzelmével, feltételezve, hogy az aszteroidán lévő telepeseknek rendelkezésére áll az űrbázis építéséhez szükséges nyersanyagmennyiség.
Teszt célja	A telepesek megnyerik a játékot.

7.3.4 Move Spaceship neighboring Asteroid

Teszt-eset neve	Move Spaceship neighboring Asteroid
Rövid leírás	Annak a folyamatnak a lemodellezése, hogy a telepes egy szomszédos aszteroidára lép.
Teszt célja	A telepes szomszédos aszteroidára lép.

7.3.5 Move Spaceship not neighboring Asteroid

Teszt-eset neve	Move Spaceship not neighboring Asteroid
Rövid leírás	Annak a folyamatnak a lemodellezése, hogy a telepes egy nem szomszédos aszteroidára lép.
Teszt célja	A telepes lépése meghiúsul, nem tud nem szomszédos aszteroidára lépni.

7.3.6 Move Spaceship through TeleportingGate

Teszt-eset neve	Move Spaceship through TeleportingGate
Rövid leírás	Annak a folyamatnak a lemodellezése, hogy a telepes használja az aszteroidához tartozó teleportkaput.
Teszt célja	A telepes kapu általi szomszédos aszteroidára lép.

7.3.7 Move Spaceship through a single TeleportingGate

Teszt-eset neve	Move Spaceship through a single TeleportingGate
Rövid leírás	Annak a folyamatnak a lemodellezése, hogy a telepes használja az aszteroidához tartozó teleportkaput, de a kapu párja nincs letéve.
Teszt célja	A telepes képtelen lépni.

7.3.8 DrillAsteroid one layer

Teszt-eset neve	DrillAsteroid one layer
Rövid leírás	Lemodellezük azt a folyamatot, amikor egy adott aszteroidán lévő telepes megfúrja az aszteroidát, csökkentve annak rétegvastagságát, feltételezve azt, hogy a rétegvastagsága legalább 1.
Teszt célja	Az aszteroida rétegvastagsága 1-gyel csökken.

7.3.9 DrillAsteroid fail

Teszt-eset neve	DrillAsteroid fail
Rövid leírás	Annak a folyamatnak a lemodellezése, amikor a telepes megpróbálja megfúrni az aszteroida köpenyét, azonban annak vastagsága már 0.
Teszt célja	Az aszteroida fúrása meghiúsul.

7.3.10 Drill radioactive Asteroid's last layer near Sun

Teszt-eset neve	Drill radioactive Asteroid last layer near Sun
Rövid leírás	Annak a folyamatnak a lemodellezése, amikor a telepes megpróbálja megfúrni az aszteroida köpenyét napközelben, de annak magjában radioaktív nyersanyag található, ezért felrobban.
Teszt célja	Az aszteroida fúrása után az Urán stabilitása sérül.

7.3.11 Drill icy Asteroid's last layer near Sun

Teszt-eset neve	Drill icy Asteroid last layer near Sun
Rövid leírás	Annak a folyamatnak a lemodellezése, amikor a telepes megpróbálja megfúrni az aszteroida köpenyét napközelben, de annak magjában vízjég nyersanyag található, ezért az elpárolog.
Teszt célja	Az aszteroida fúrása után elszublimál a Vízjég.

7.3.12 Restore Resource

Teszt-eset neve	Restore Resource
Rövid leírás	Annak a folyamatnak a lemodellezése, amikor a telepes nyersanyagot helyez el az aszteroida belsejébe.
Teszt célja	Az aszteroida nyersanyaga a telepes által behelyezett nyersanyag lesz.

7.3.13 Restore Resource with empty container

Teszt-eset neve	Restore Resource with empty container
Rövid leírás	Annak a folyamatnak a lemodellezése, amikor a telepes nyersanyagot helyezne el az aszteroida belsejébe, de a tárolója üres.
Teszt célja	A telepes nem képes letenni semmit.

7.3.14 Restore Resource Fail not empty

Teszt-eset neve	Restore Resource Fail not empty
Rövid leírás	Annak a folyamatnak a lemodellezése, amikor a telepes egy nyersanyagot próbál meg elhelyezni az aszteroida belsejébe, azonban ott már van nyersanyag.
Teszt célja	A nyersanyag elhelyezése meghiúsul.

7.3.15 Restore RadiactiveResource near Sun

Teszt-eset neve	Restore RadiactiveResource near Sun
Rövid leírás	Annak a folyamatnak a lemodellezése, amikor a telepes egy nyersanyagot próbál meg elhelyezni az aszteroida belsejébe, azonban mindezt napközelben teszi, így az aktiválódik és felrobban.
Teszt célja	A nyersanyag elhelyezése után felrobban az Aszteroida.

7.3.16 Restore IceResource near Sun

Teszt-eset neve	Restore IceResource near Sun
Rövid leírás	Annak a folyamatnak a lemodellezése, amikor a telepes egy nyersanyagot próbál meg elhelyezni az aszteroida belsejébe, azonban mindezt napközelben teszi, így az elpárolog.
Teszt célja	A nyersanyag elhelyezése után a Vízjég elszublimál.

7.3.17 Mine Resource

Teszt-eset neve	Mine Resource
Rövid leírás	Lemodellezzük azt a folyamatot, amikor a telepes kinyeri nyersanyagot az aszteroida belsejéből.
Teszt célja	A telepes birtokolja a nyersanyagot.

7.3.18 Mine Resource fail when Asteroid is empty

Teszt-eset neve	Mine Resource fail when Asteroid is empty
Rövid leírás	Lemodellezzük azt a folyamatot, amikor a telepes megpróbálja kinyeri a nyersanyagot az aszteroida belsejéből, azonban az üres.
Teszt célja	A nyersanyag kinyerése meghiúsul.

7.3.19 Mine Resource fail when Spaceship is full

Teszt-eset neve	Mine Resource fail when Spaceship is full
Rövid leírás	Lemodellezük azt a folyamatot, amikor a telepes megpróbálja kinyeri a nyersanyagot az aszteroida belsejéből, azonban az nincs hely az űrhajóján.
Teszt célja	A nyersanyag kinyerése meghiúsul.

7.3.20 Mine Resource fail when Asteroid is not drilled

Teszt-eset neve	Mine Resource fail when Asteroid is not drilled
Rövid leírás	Lemodellezük azt a folyamatot, amikor a telepes megpróbálja kinyeri a nyersanyagot az aszteroida belsejéből, azonban az aszteroida nincs kifúrva.
Teszt célja	A nyersanyag kinyerése meghiúsul.

7.3.21 Build AIRobot

Teszt-eset neve	Build AIRobot
Rövid leírás	Annak a folyamatnak a lemodellezése, amikor a telepes egy új AIRobotot hoz létre, feltételezzük, hogy rendelkezésre áll az összes szükséges nyersanyag.
Teszt célja	Új AIRobot keletkezik.

7.3.22 Build AIRobot fail

Teszt-eset neve	Build AIRobot fail
Rövid leírás	Annak a folyamatnak a lemodellezése, amikor a telepes egy új AIRobotot hoz létre, azonban nem áll rendelkezésre az összes szükséges nyersanyag.
Teszt célja	Nem keletkezik új AIRobot

7.3.23 Build TeleportingGate pair

Teszt-eset neve	Build TeleportingGate pair
Rövid leírás	Annak a folyamatnak a lemodellezése, amikor a telepes egy új Teleportkapu párt hoz létre, feltételezzük, hogy rendelkezésre áll az összes szükséges nyersanyag.
Teszt célja	Új teleportkapu pár keletkezik

7.3.24 Build TeleportingGate pair fails when Player has not enough Resources

Teszt-eset neve	Build TeleportingGate pair fails when Player has not enough Resources
Rövid leírás	Annak a folyamatnak a lemodellezése, amikor a telepes egy új Teleportkaput hozna létre, azonban nem áll rendelkezésre az összes szükséges nyersanyag.
Teszt célja	Nem keletkezik új Teleportkapu-pár

7.3.25 Build TeleportingGate pair when Player has not enough space

Teszt-eset neve	Build TeleportingGate pair when Player has not enough space
Rövid leírás	Annak a folyamatnak a lemodellezése, amikor a telepes egy új Teleportkaput-párt hozna létre, azonban nem áll rendelkezésre elegendő tárhely az űrhajóban.
Teszt célja	Nem keletkezik új Teleportkapu-pár

7.3.26 Place TeleportingGate

Teszt-eset neve	Place TeleportingGate
Rövid leírás	A telepes lerakja a teleportkapu-pár egyik tagját az aszteroidára, feltételezzük, hogy rendelkezik a szükséges kapuval.
Teszt célja	Az aszteroida megkapja a kaput

7.3.27 Place second TeleportingGate

Teszt-eset neve	Place second TeleportingGate
Rövid leírás	A telepes lerakja a teleportkapu pár második tagját az aszteroidára, így a teleport használhatóvá válik. Feltételezzük, hogy rendelkezik a szükséges kapuval.
Teszt célja	Az aszteroida megkapja a kaput, a teleport használható lesz.

7.3.28 Call Sunstorm not mined asteroid with AIRobot

Teszt-eset neve	Call Sunstorm not mined asteroid with AIRobot
Rövid leírás	A napvíhar folyamatának a modellezése, feltételezve, hogy az elért aszteroida nincs kibányászva.
Teszt célja	Az aszteroidán lévő AIRobot szomszédos aszteroidára lép

7.3.29 Call Sunstorm not mined asteroid with Settler

Teszt-eset neve	Call Sunstorm not mined asteroid with Settler
Rövid leírás	A napvíhar folyamatának a modellezése, feltételezve, hogy az elért aszteroida nincs kibányászva.
Teszt célja	Az aszteroidán lévő telepes meghal.

7.3.30 Call Sunstorm mined asteroid

Teszt-eset neve	Call Sunstorm mined asteroid
Rövid leírás	A napvíhar folyamatának a modellezése, feltételezve, hogy az elért aszteroida ki van bányászva.
Teszt célja	Az aszteroidán lévő karakterek túlélik a napvihart.

7.3.31 UFO moves

Teszt-eset neve	UFO moves
Rövid leírás	Az UFO mozgásának modellezése.
Teszt célja	Az UFO szomszédos aszteroidára lép

7.3.32 UFO mines

Teszt-eset neve	UFO mines
Rövid leírás	Az UFO bányászatának modellezése. Az UFO a bányászás után eltünteti a nyersanyagot.
Teszt célja	Az UFO elveszi az aszteroida nyersanyagát.

7.3.33 UFO dies

Teszt-eset neve	UFO dies
Rövid leírás	Az UFO halálának modellezése, ami a telepes életére veszélyes dolgokkal egyezik meg (napvihar, napszél).
Teszt célja	Az UFO meghal robbanás vagy napszél esetén.

7.3.34 AIRobot moves

Teszt-eset neve	AIRobot moves
Rövid leírás	Az AIRobot mozgásának modellezése. Akkor lép, ha nem tud fűrni.
Teszt célja	Az AIRobot szomszédos aszteroidára lép

7.3.35 AIRobot drills

Teszt-eset neve	AIRobot drills
Rövid leírás	Az AIRobot furásának modellezése. A robot fűrni fog, ha egy olyan aszteroidán tartózkodik, ami nincs kifúrva.
Teszt célja	Az AIRobot egy egységet lefűr az aszteroida rétegében.

7.3.36 AIRobot got pushed

Teszt-eset neve	AIRobot got pushed
Rövid leírás	Az AIRobot ellkötésének modellezése. A robot a szomszédos aszteroidára repül egy radioaktív robbanás esetén.
Teszt célja	Az AIRobot a robbanás után szomszédos aszteroidára repül.

7.3.37 AIRobot dies

Teszt-eset neve	AIRobot dies
Rövid leírás	Az AIRobot halálának modellezése. A robot meghal napvihar esetén, ha nem tud elrejtőzni.
Teszt célja	Az AIRobot meghal egy napvihar következtében.

7.3.38 TeleportingGate goes mad

Teszt-eset neve	TeleportingGate goes mad
Rövid leírás	Az teleportkapu megörülésének modellezése. A kapu mozogni kezd aszteroidák között, ha napszél éri.
Teszt célja	Az teleportkapu napvihar hatására mozgni kezd.

7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

A tesztelést egy önálló Java alkalmazás segíti.

A tesztelést végző program soronként egy-egy *test* parancsot vár. Ez a parancs hívható paraméter nélkül is (*test*) vagy egy teszteset sorszámát paraméterként megadva (*test <number>*).

- Paraméter nélküli futtatás esetén a vele egy mappában található összes összetartozó input-output fájlhoz tartozó tesztesetet futtatja.
- Paraméteres futtatás esetén egyetlen pozitív egész számot (*number*) vár paraméterként (szóközzel elválasztva), amely a tesztelendő teszteset sorszáma. Ekkor csak a megadott sorszámú input-output fájlokhoz tartozó tesztesetet futtatja.

Minden teszteset két fájlból áll: *test_<number>.in* és *test_<number>.out* (például *test_7.in* és *test_7.out*).

A *test_<number>.in* fájl tartalmazza a program bemeneti parancsait, amelyek futtatása eredményeképp előálló játék konfigurációinak meg kell egyeznie a *test_<number>.out* fájl tartalmával.

A tesztprogram minden tesztesethez kiírja egy üzenetet a konzolra arról, hogy a teszt sikeres volt-e, vagy sem. A sikerhez szükséges, hogy a megfelelő input és output fájlok létezzenek a futtatott tesztprogrammal egy mappában, és hogy a formátumuk megfeleljen a 7.2.2.3 pontban leírt játékkonfigurációs fájlformátumnak. Ha a futtatás befejeződik, és a kimenet (tartalma) megegyezik az elvárt kimenettel, akkor a teszt sikeres, különben meghiúsult.

A tesztelés a létrejött objektumok kötelező sorrendű leírása miatt lehetséges. Emiatt a különböző játékállapotok különböző konfigurációs fájlokba képződnek le, és megfordítva, nincs két különböző konfigurációs fájl, amely ugyanazt a játékállapotot írná le.

7.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2021.03.25. 18:00	2 óra	Havasi Kreták Mesterházi Radnai Szögi	Értekezlet. Változások átbeszélése, átgondolása, új osztályok, függvények, attribútumok megbeszélése, megvalósítása. Részfeladatok kiosztása
2021.03.26. 18:00	4 óra	Havasi Kreták Mesterházi Radnai Szögi	Tevékenység: A kiosztott feladatok elkészítése
2021.03.27. 16:00	2,5 óra	Havasi Kreták Mesterházi Radnai Szögi	Értekezlet. A kiosztott feladatokban felmerült újabb kérdések, problémák feloldása, egymás feladatainak átnézése, átbeszélése
2021.03.28. 20:00	2 óra	Radnai	Tevékenység: A prototípus interfészéről, illetve a tesztprogramról szóló fejezetek kiegészítése, kisebb javítások a dokumentumban
2021.03.30. 21:00	1 óra	Mesterházi	Tevékenység: További tesztesetek hozzáadása
2021.03.31. 10:00	2 óra	Havasi Kreták Mesterházi Radnai Szögi	Értekezlet + Tevékenység: Kommentek átnézése, megegyezés apróbb részletekről, dokumentum véglegesítése

Az adott heti munka csapattagoknál aránya:

Csapattag	Munka aránya százalékban
Havasi	20
Kreták	20
Mesterházi	20
Radnai	20
Szögi	20

8.

Részletes tervezek

7 – berenyi_kft

Konzulens:

Csapattagok

Kreták Barnabás András

Mesterházi Marcell

Szögi Richárd

Havasi Balázs

Radnai Bálint

2021.04.11

8. Részletes tervezés

A pszeudokódokban intendálással jelöljük a parancsok elágazásba és ciklusba való tartozását.

8.1.1 AIRobot

- **Felelősség**

A karakterek egyik fajtája a robot, amely minden időegységben fújni vagy szomszédos aszteroidára mozogni tud.

- **Ősosztályok**

Character

- **Interfészek**

ISteppable

- **Attribútumok**

- **Metódusok**

- **public void step()**: A robot megkérdezi az aszteroidától, amelyen áll, hogy mekkora a köpenyvastagsága. Ha az aszteroida nincs teljesen átfúrva, akkor a robot drill() műveletet végez, különben valamelyik szomszédos aszteroidára lép (move(int d)).

Pszeudokód:

```
Ha place.getRockLayerThickness () >= 1  
    akkor drill()  
Különben  
    move (random szám)
```

- **public void drill()**: meghívja az aszteroida drilled() metódusát.
- **public void die()**: A robot megsemmisül: meghívja az ősosztály die() metódusát, majd a Timer removeSteppable(Steppable s) metódusát.
- **public void reactToExplosion()**: A robot egy véletlenszerű szomszédos aszteroidán landol: sorsol egy d számot, majd meghívja a move (int d) metódust.

8.1.2 Asteroid

- **Felelősség**

Az aszteroidákon karakterek tartózkodhatnak, valamint teleportkapuk keringhetnek körülöttük. Az aszteroidák felelőssége a rajtuk lévő karakterek és teleportkapuk, illetve a szomszédos aszteroidák nyilvántartása. Tárolja a köpenye aktuális vastagságát és a magjában található nyersanyagot. Robbanás, illetve napvihar esetén a követelményekben leírtak alapján kezeli az ott található nyersanyagot, karaktereket és teleportkapukat.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **private int rockLayerThickness:** az aszteroida köpenyvastagsága, vagyis a magot borító sziklarétegek száma
- **private Game game:** a játékot reprezentáló osztály
- **private Sun sun:** az aszteroidaövben levő Nap
- **private Asteroid[0..*] neighbors:** az aszteroidával szomszédos aszteroidák listája, ebben a kollekcióban nincsenek benne a teleportkapuk általi szomszédok.
- **private Resource resource:** az aszteroida magjában található egységnyi nyersanyag; ha a mag üres, akkor értéke null
- **private Character[0..*] characters:** az aszteroidán tartózkodó karakterek (telepesek, robotok, stb.) kollekciója
- **private TeleportingGate[0..*] gates:** a közvetlenül az aszteroida körül keringő teleportkapuk halmaza

- **Metódusok**

- **public void accept(Character c):** A c karakter megérkezik az aszteroidára, az aszteroida hozzáadja a characters kollekciójához. Ezután ellenőrzi az ūrbázis megépíthetőségét: meghívja az checkSpaceBase() függvényt.
- **public void remove(Character c):** A c karakter elhagyja az aszteroidát, az aszteroida eltávolítja a characters kollekciójából.
- **public void accept(TeleportingGate tg):** A tg teleportkapu pályára áll az aszteroida körül, az aszteroida hozzáadja a gates kollekciójához.
- **public void remove(TeleportingGate tg):** A tg teleportkaput eltávolítja az aszteroida körüli pályáról, az aszteroida törli a gates kollekciójából.
- **public void accept(Settler s, Resource r):** Egy az aszteroidán tartózkodó telepes megpróbálja behelyezni az r nyersanyagot az aszteroida üreges magjába: ha az aszteroida üres, akkor az aszteroida beállítja azt a resource attribútumának és meghívja a Settler remove(r: Resource) metódusát. Ha a paraméterül kapott r=null, akkor a függvénynek nincs hatása, a resource attribútumot nem írja felül. Végül, ha az aszteroida napközelben van, meghívja a nyersanyag drilledOut(a: Asteroid) metódusát.

Pszeudokód:

```
Be: Settler s, Resource r
Ha isMined()
    resource=r
    s.remove(r)
Ha sun.IsCloseToSun(aktuális aszteroida)
    resource.drilledOut(aktuális aszteroida)
```

- **public void removeResource()**: Az aszteroida a resource attribútumát null-ra állítja.
- **public void accept(Asteroid neighbor)**: Ha a neighbors gyűjtemény nem tartalmazza a neighbor aszteroidát, akkor hozzáadja a neighbor aszteroidát a kollekcióhoz.
- **public void remove(Asteroid a)**: Ha a neighbors gyűjtemény tartalmazza az a aszteroidát, akkor eltávolítja az a aszteroidát a kollekcióból, majd pedig meghívja az a remove(neighbor: Asteroid) metódusát.

Pszeudokód:

```
Be: Asteroid a
Ha neighbors tartalmazza a-t
    neighbors.remove(a)
    a.remove(aktuális aszteroida)
```

- **public void drilled()**: Az aszteroida rockLayerThickness attribútumát egyelőre csökkenti, amennyiben az 1, vagy annál nagyobb. Ha ezután a köpenyvastagság 0, és a resource attribútuma nem null, akkor a sun objektum isCloseTo(Asteroid a) függvényével lekérdezi, hogy napközelben található-e. Ha igen, akkor meghívja a resource nyersanyag drilledOut(a: Asteroid) függvényét, jelezve, hogy napközeli aszteroidán felszínre került.

Pszeudokód:

```
Ha rockLayerThickness>=1
    rockLayerThickness=rockLayerThickness-1
Ha rockLayerThickness=0 és resource!=null és
sun.IsCloseToSun(aktuális aszteroida)
    resource.drilledOut(aktuális aszteroida)
```

- **public void minedBy(Settler s)**: Az s telepes bányászik az aszteroidán. Ha az aszteroida köpenyvastagsága nem 0, akkor a függvénynek nincs hatása. Ha a köpenyvastagság 0, és az aszteroida resource attribútuma nem null, akkor eltávolítja azt a magjából (removeResource()), és eltároltatja azt az s telepessel (s.accept(resource)), majd pedig meghívja a checkSpaceBase() függvényt.

Pszeudokód:

```
Be: Settler s
Ha rockLayerThickness=0 és resource!=null
    s.accept(resource)
    removeResource()
    checkSpaceBase()
```

- **public boolean isMined()**: Visszatér annak logikai értékével, hogy az aszteroida megfűrt és üreges, vagyis rockLayerThickness=0 és resource=null.

- **public void explodedBy(RadioactiveResource rr):** A megfúrt, napközelben lévő, radioaktív nyersanyagot tartalmazó aszteroida felrobban. minden rajta tartózkodó karakterek meghívja a reactToExplosion() függvényét. Az aszteroida szomszédain a remove(Asteroid neighbor) függvényt hívja, amivel törli magát a szomszédai szomszédsági listáiból, majd megsemmisíti a körülötte keringő teleportkapukat a párjaikkal együtt. Végül eltávolítja magát a játékból a Game osztály removeAsteroid() függvényét hívva.

Pszeudokód:

```
Be: RadioActiveResource rr
Ciklus a characters minden elemére visszafelé
    reactToExplosion()
Ciklus a gates minden elemére visszafelé
    die()
Ciklus a neighbors minden elemére visszafelé
    remove(aktuális aszteroida)
game.removeAsteroid(aktuális aszteroida)
```

- **public void destroySurface():** Az aszteroidán végigsöpör a napvihar. Ha az aszteroida ki van bányászva, tehát az isMined() visszatérési értéke true, akkor a függvény meghívja az aszteroidán tartózkodó karakterek die() függvényét, továbbá minden esetben meghívja az aszteroidához kapcsolt teleportkapuk goMad() metódusát.

Pszeudokód:

```
Ha nem isMined()
    Ciklus a characters minden elemére visszafelé
        die()
    Ciklus a gates minden elemére visszafelé
        goMad()
```

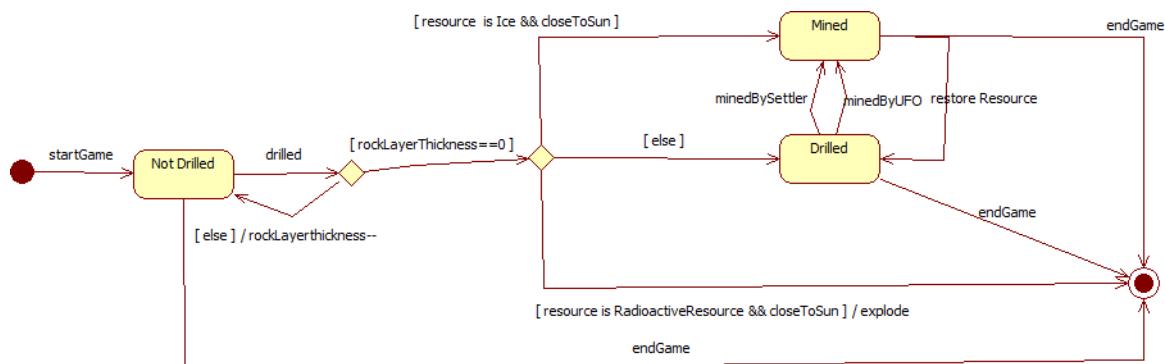
- **public void checkSpaceBase():** Ellenőrzi, hogy az adott aszteroidán lévő telepeseknél rendelkezésre áll-e az űrbázis felépítéséhez szükséges nyersanyagmennyiség. Létrehoz egy Resource típust tároló kollekciót és hozzáadja az aszteroidán lévő összes karakter eltárolt nyersanyagait. Elkéri a Game-től az űrbázis receptjét és a kollekció minden elemével meghívja a Recipe osztály isNeeded(r: Resource) metódusát. Ha a recept listája üres, akkor meghívódik a Game endGame() metódusa, végül pedig meghívja a Recipe reset() metódusát.

Pszeudokód:

```
Létrehozunk egy Resource objektumokat tároló temp nevű
listát
Ciklus a characters minden elemére
    Hozzáadjuk a temp-hez az aktuális elem
    collectedResources kollekciójának tartalmát
Létrehozunk egy Recipe típusú recipe nevű változót
recipe=game.getSpaceBaseRecipe()
Ciklus a temp minden elemére
    Ha recipe.isEmpty()
        ciklus vége
        recipe.isNeeded(aktuális elem)
    Ha recipe.isEmpty()
        game.endGame()
    recipe.reset()
```

- **public Resource getResource():** Visszaadja az adott aszteroida magjában található nyersanyagot.
 - **public Sun getSun():** Visszaadja a játékban levő Napot.
 - **public Character[0..*] getCharacters():** Visszaadja az adott aszteroidán tartózkodó karakterek kollekcióját.
 - **public Asteroid[0..*] getNeighbors():** Visszaadja az adott aszteroidával szomszédos aszteroidák kollekcióját (neighbors kollekciót).
 - **public Asteroid getNeighbor(int d):** Visszaadja az aszteroida d-edik szomszédját: A neighbors kollekcióból és a teleportkapuk által elérhető aszteroidákból kollekciót képez, majd ebből a kollekcióból adja vissza a modulo d-edik elemet, a kollekció elemszáma szerint.
- Pszeudokód:
- Be: egész d szám
- ```
Létrehozunk egy Asteroid objektumokat tároló list nevű listát
Hozzáadjuk a listához a neighbours kollekció tartalmát
Ciklus a gates minden elemére
 Hozzáadjuk a list-hez az aktuális elem aszteroidáját
 d = d % list mérete
 Ki: list d-edik eleme
```
- **public TeleportingGate getGates():** Visszaadja az adott aszteroidához tartozó teleportkapukat.
  - **public void minedByUFO():** Ha az aszteroida rétegvastagsága nulla és tartalmaz nyersanyagot, akkor meghívja a removeResource() metódusát.
  - **public int getRockLayerThickness():** Visszaadja a RockLayerThickness értékét.
  - **public void setRockLayerThickness(int value):** Beállítja a rockLayerThickness értékét a paraméterként kapott értékre.
  - **public void addResource(Resource r):** Beállítja a resource attribútum értékét a paraméterként kapottéra.
  - **public void setGame(Game g):** Beállítja a game attribútumot a paraméterként kapottéra.
  - **public void setSun(Sun s):** Beállítja a sun attribútumot a paraméterként kapottéra.

Asteroid állapotgépe:



### 8.1.3 Character

- **Felelősség**

Ez az osztály absztrakt ōsosztályként szolgál a játék karaktereihez. A karakterek aszteroidákon tartózkodhatnak, és köztük mozoghatnak. Különbözöképpen reagálhatnak aszteroidák robbanására.

- **Ósosztályok**

- **Interfészek**

- **Attribútumok**

- **protected Asteroid place:** az aszteroida, amelyen az adott karakter aktuálisan tartózkodik.
- **protected Timer timer:** A játékban található időzítő.

- **Metódusok**

- **public Resource[0..\*] getCollectedResources():** Visszatér a karakter által tárolt nyersanyagok listájával, alapértelmezetten egy üres listával.
- **public void move(int d):** Átmozog az aktuális aszteroidáról annak egy szomszédjára: Először lekérdezi az aktuális aszteroida modulo d-edik szomszédját (a getNeighbor(int d) függvényel). Ezután eltávolítja magát a jelenlegi aszteroidáról (remove(Character c)), majd a visszakapott szomszéd aszteroidára lép (accept(Character c)), végül pedig beállítja a place attribútumot az új aszteroidára.

Pszeudokód:

```
Be: egész d szám
Létrehozunk egy Asteroid típusú neighbor nevű objektumot
neighbor=place.getNeighbor(d)
place.remove(aktuális karakter)
neighbor.accept(aktuális karakter)
place=neighbor
```

- **public void die():** A karakter halálát, jelenti: eltávolítja a karaktert az aszteroidájáról (place.remove(Character c)). A leszármazottakban felüldefiniálható.
- **public void reactToExplosion():** A metódus, amit a karakternek akkor kell végrehajtania, ha az aszteroida, amelyen tartózkodik, felrobban. Alapértelmezetten a karakter die() függvényét hívja. A leszármazottak felüldefiniálhatják.
- **public Asteroid getPlace():** Visszatérési értéke az az aszteroida, amelyen a karakter épp tartózkodik.
- **public void setPlace(Asteroid a):** Beállítja a place attribútumot a paraméterként kapott aszteroidára.

### 8.1.4 Coal

- **Felelősség**

A szén nyersanyagot reprezentálja.

- **Ósosztályok**

Resource

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **public boolean isCompatibleWith(r: Resource):** Összehasonlítja magát a paraméterként kapott nyersanyaggal. Egyezés esetén igaz értékkel tér vissza, ellenkező esetben hamissal.

### 8.1.5 Controller

- **Felelősség**

A játékban található játékosok kezeléséért, a játékosok lépéseinek

- **Ósosztály**

- **Interfészek**

- **Attribútumok**

- **private Game game:** Az elindított játék.
- **private Player[0..\*] playersAlive:** Az életben lévő játékosok kollekciója.
- **private Player actPlayer:** Az éppen lépő játékos.

- **Metódusok**

- **public void startGame():** A játék elindítását valósítja meg, megkérdezi a játékosok számát és neveit, majd pedig példányosítja őket. Példányosítja a Game-t, illetve a Timer-t, majd pedig meghívja a Game startGame() metódusát.

- **public Player getActPlayer():** Visszatér az actPlayer attribútummal.

- **public void nextPlayer():** Az actPlayer-t a playersAlive-ban lévő következő Player-re állítja, ha a kollekció végére ért, akkor a gyűjtemény 0. elemére állítja be és meghívja a game-ben található Timer tick() metódusát.

- **public void endGame(enum s):** Az s értékének megfelelően kiírja azt, hogy a telepesek nyertek vagy pedig veszítettek.

- **public void removePlayer(Player p):** Törli a paraméterként kapott p-t a playersAlive kollekcióból.

### 8.1.6 Game

- **Felelősség**

A játék indításáért és befejezéséért felel, a játékban lévő telepesek, aszteroidák, a Nap, valamint a felépíthető dolgok receptjeinek nyilvántartását végzi.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **private Asteroid[0..\*] asteroids:** a játékban jelenlévő aszteroidák kollekciója
- **private Sun sun:** az aszteroidaövbeli Nap
- **private Settler[0..6] settlersAlive:** az adott pillanatban életben levő telepesek listája
- **private Recipe[3] recipes:** A játékban felépíthető dolgok receptjeinek a kollekciója.
- **private Timer timer:** A játékban található időzítő.
- **private Controller controller:** A játékot vezérlő controller.
- **Player[0..\*] allPlayers:** A még élő telepesek játékosainak a kollekciója.

- **Metódusok**

- **public void startGame():** Inicializálja az aszteroidákat, a Napot, és elhelyezi a telepeseket valamelyik aszteroidán. Egyes aszteroidák magját üresen hagyja, másokban pedig véletlenszerűen valamilyen nyersanyagot hoz létre. Beállítja az aszteroidák szomszédsági viszonyait is.
- **public void endGame():** Meghívja a controller endGame(enum s) metódusát, a paraméter jelzi, hogy a telepesek nyertek vagy pedig veszítettek.
- **public void removeSettler(Settler s):** Törli a meghaló telepest és a telepeshez tartozó játékosat a játékból. Ha nincs több életben levő telepes, akkor meghívja az endGame() függvényt.
- **public void removeAsteroid(Asteroid a):** Törli a megsemmisülő a aszteroidát az asteroids kollekcióból.
- **public Asteroid[0..\*] getAsteroids():** Visszaadja a játék összes aszteroidájából álló kollekciót.
- **public Recipe getGatePairRecipe():** Visszaadja a teleportkapu-pár felépítéséhez szükséges receptet tároló Recipe objektumot.
- **public Recipe getAIRobotRecipe():** Visszaadja az AIRobot felépítéséhez szükséges receptet tároló Recipe objektumot.
- **public Recipe getSpaceBaseRecipe():** Visszaadja az űrbázis felépítéséhez szükséges receptet tároló Recipe objektumot.
- **public Timer getTimer():** Visszaadja a játékban található időzítőt.
- **public void setTimer(Timer timer):** Beállítja a timer attribútumot a paraméterként kapottra.
- **public void addSettler(Settler s):** Hozzáadja a paraméterként kapott Settler-t a settlersAlive kollekcióhoz.
- **public void addAsteroid(Asteroid a):** Hozzáadja a paraméterként kapott aszteroidát az asteroids kollekcióhoz.
- **public void setSun(Sun s):** Beállítja a sun attribútumot a paraméterként kapottra.
- **public int getSizeOfSettlersAlive():** Visszatér a játékban lévő Settlerek számával.

- **public Settler[ 0..\*] getSettlers():** Visszatér a settlersAlive kollekcióval.
- **public void addRecipe(Recipe recipe):** Hozzáadja a paraméterként kapott receptet a recipes kollekcióhoz.

### 8.1.7 Ice

- **Felelősség**

A vízjég nyersanyagot reprezentálja. Ha napközeli, teljesen megfűrt aszteroida magjában található, akkor elszublimál.

- **Ősosztályok**

Resource

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **public boolean isCompatibleWith(r: Resource):** Összehasonlítja magát a paraméterként kapott nyersanyaggal. Egyezés esetén igaz értékkel tér vissza, ellenkező esetben hamissal.
- **public void drilledOut(a: Asteroid):** Hatására a vízjég elszublimál: meghívja az asteroid.removeResource() metódust a paraméterként kapott aszteroidán.

### 8.1.8 Iron

- **Felelősség**

A vas nyersanyagot reprezentálja.

- **Ősosztályok**

Resource

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **public boolean isCompatibleWith(r: Resource):** Összehasonlítja magát a paraméterként kapott nyersanyaggal. Egyezés esetén igaz értékkel tér vissza, ellenkező esetben hamissal.

### 8.1.9 ISteppable

- **Felelősség**

Olyan objektumot jelölő interfész, amely a játékban külső irányítás (játékos aktor beavatkozása) nélkül léptethető.

- **Ősosztályok**

- **Metódusok**

- **public void step()**: A művelet, amelyet az objektum az adott lépésben végez.

### 8.1.10 Player

- **Felelősség**

A játékban található játékosokról tárol adatokat, a játékos kéréseit közvetíti.

- **Ősosztály**

- **Interfészek**

- **Attribútumok**

- **private string name**: A játékos nevét tárolja el.
- **private Settler settler**: A játékoshoz tartozó telepest tárolja el.
- **private boolean isAlive**: A játékos állapotát tárolja el.
- **private enum Command**: A játékos által kiadható parancsok halmaza.

- **Metódusok**

- **public void ActonSettler(Command cmd)**: A paraméterül kapott parancsnak megfelelően meghívja a Settler adott metódusát.
- **public Settler getSettler()**: Visszaadja a játékoshoz kapcsolt telepest.
- **public String getName()**: Vissza adja a játékos nevét.
- **public void setName(String name)**: Beállítja a játékos nevét a paraméterül kapotttra.
- **public boolean isAlive()**: Visszatér az isAlive logikai értékkel.
- **public void setAlive(boolean isAlive)**: Paraméterként kapott logikai értékre állítja az isAlive attribútumot.

### 8.1.11 Proto

- **Felelősség**  
A felhasználó által kiadott vagy beolvasott parancsokat megfelelteti a játék belső működésében használatos metódushívásoknak, valamint a kiíratás intendálásáért felel.
- **Ősosztály**
- **Interfészek**
- **Attribútumok**
  - **private static Map<Object, String> ids:** A programban található objektumok és azonosítóiuk
  - **enum State:** A program lehetséges állapotainak halmaza
  - **private static int tabs:** az intendálás mértéke
  - **private static State state:** A program aktuális állapota
- **Metódusok**
  - **public static void println(String line):** Logolás intendálását végzi.
  - **public static void incrTabs():** Növeli a tabs értékét eggyel.
  - **public static void decrTabs():** Csökkenti a tabs értékét eggyel.
  - **public static void main(String[] args):** main
  - **public static void Test():** Teszteseteket futtat.
  - **public static void Commands():** Parancsokat vár a felhasználótól.

### 8.1.12 RadioactiveResource

- **Felelősség**

A radioaktív nyersanyagtípuszt reprezentálja, radioaktív nyersanyagok ősosztályaként szolgál. Ha napközelben lévő, megfűrt aszteroida tartalmazza, speciálisan viselkedik.

- **Ősosztályok**

Resource

- **Interfészek**
- **Attribútumok**
- **Metódusok**
  - **public void drilledOut(a: Asteroid):** Meghívja a paraméterként kapott aszteroida explodedBy(RadioactiveResource rr) metódusát (robban). Az Uranium osztályban felül van definiálva.

8.1.13

### 8.1.14 Recipe

- **Felelősség**

A játékban felépíthető dolgok létrehozásához szükséges nyersanyagok listáját tartalmazza. A telepeseknél lévő nyersanyagokat összehasonlíta a listájában lévőkkel. Különböző példányai különböző dolgok megépítéséhez szükséges nyersanyagokat tartalmazzák.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **private Resource[0..\*] resources:** A szükséges nyersanyagok kollekciója.
- **private Resource[0..\*] resources\_backup:** Az eredeti recept nyersanyagait tárolja el, a resources kollekció visszaállításához szükséges.

- **Metódusok**

- **public void addResource(r: Resource):** A paraméterként kapott nyersanyagot hozzáadja a resources kollekcióhoz, majd pedig klónozza a nyersanyagot és a klónt hozzáadja a resources\_backup kollekcióhoz. A recept módosításához szükséges, ha változnának a feltételek.
- **public void isNeeded (r: Resource):** A paraméterként kapott r nyersanyagot összehasonlíta a resources kollekcióban lévő nyersanyagokkal, az isCompatibleWith(r: Resource) metódussal. Amennyiben az isCompatibleWith(r: Resource) függvény igaz értékkel tér vissza valamelyik kollekcióból nyersanyagra, törli az adott nyersanyagot a resources kollekcióból.

Pszeudokód:

```

Be: Resource r
Ciklus a resources kollekció minden elemére visszafelé
 Ha r.isCompatibleWith(aktuális elem)
 Töröljük a resources kollekcióból az
 aktuális elemet
 Ki: true
Ki: false

```

- **public void reset():** Visszaállítja az eredeti receptet a resources\_backup segítségével.
- **public boolean isEmpty():** Ha a resources kollekció üres, igaz értékkel tér vissza, ellenkező esetben pedig hamissal.
- **public Resource[0..\*] getResources():** Visszatér a resources kollekcióval.

### 8.1.15 Resource

- **Felelősség**

A játékban levő nyersanyagok ōsosztálya. Többféle nyersanyag létezik, amelyek lehetnek aszteroida magjában vagy telepesnél tárolva. Ha egy nyersanyag napközelben levő megfűrt aszteroida magjában található, akkor az aszteroidán - a konkrét fajtájától függően - speciális hatást válthat ki.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **public void drilledOut(a: Asteroid):** Azt a helyzetet kezeli, amikor a nyersanyag éppen egy napközelben levő, megfűrt aszteroida magjában található. A metódus általános esetben nem csinál semmit, de a speciális működésű leszármazott osztályokban felüldefiniálható.
- **public abstract boolean isCompatibleWith(r: Resource):** Összehasonlítja magát a paraméterként kapott nyersanyaggal és egyezés esetén igaz értékkel tér vissza, ellenkező esetben hamissal.
- **public Resource clone():** Visszatér a nyersanyag egy klónjával.

### 8.1.16 Settler

- **Felelősség**

A karakterek egyik fajtája a telepes, amely a Character-ben megfogalmazottakon túl képes fűrni, egységes nyersanyagot kibányászni vagy visszahelyezni az aszteroidamagba. Emellett építhet robotot és teleportkapu-párt, illetve az elkészült teleportkapukat pályára állíthatja. Nyilvántartja a nála található nyersanyagokat és teleportkapukat.

- **Ősosztályok**

Character

- **Interfészek**

- **Attribútumok**

- **private Game game:** a játékot reprezentáló osztály
- **private Resource[0..10] resources:** a telepes által tárolt nyersanyagok kollekciója; a kollekció kapacitása 10
- **private TeleportingGate[0..3] gatesCreated:** a telepes által tárolt teleportkapuk kollekciója

## Metódusok

- **public Resource[0..\*] getCollectedResources():** Visszatér a telepes által összegyűjtött nyersanyagok kollekciójával. Felüldefiniálja az ősosztály azonos nevű metódusát.
- **public void accept(Resource r):** A telepes eltárolja a kibányászott r nyersanyagot a resources kollekciójában.
- **public void remove(Resource r):** A telepes eltávolít egy darab paraméterként kapott típusú nyersanyagot a resources kollekciójából.
- **public void remove(TeleportingGate: tg):** A telepes eltávolítja a tg teleportkaput a gatesCreated kollekciójából.
- **public void mine():** A telepes kibányássza az adott megfűrt aszteroida magjában található nyersanyagot. Ehhez meghívja az aszteroida minedBy() metódusát. Ha a bányászat kezdetén az űrhajóban már nincs hely újabb nyersanyagnak, akkor a telepes nem tud bányászni, a függvénynek nincs hatása.
- **public void restore(Resource r):** Meghívja az aktuális aszteroida accept(Settler s, Resource r) metódusát.
- **public void createAIRobot():** A telepes lekérdezi a Game-től az AI robot megépítéséhez szükséges receptet a Recipe getAIRobotRecipe() metódusának meghívásával, majd pedig összehasonlíta a saját nyersanyagait a receptben lévőkkel. Amennyiben rendelkezik a szükséges nyersanyagokkal, létrehoz egy AIRobot példányt, ezután pedig hozzáadja az aktuális aszteroidához az accept(c: Character) metódus meghívásával. Végül meghívja a Recipe reset() függvényét, amelyben ezáltal visszaállítja a recept listájának a tartalmát.

Pszeudokód:

```
Létrehozunk egy Recipe típusú aiRobotRecipe nevű
objektumot
aiRobotRecipe=game.GetAIRobotRecipe()
Ciklus a collectedResources minden elemére visszafelé
 Ha aiRobotRecipe.isEmpty()
 ciklus vége
 AiRobotRecipe.isNeeded(aktuális elem)
 Ha aiRobotRecipe.isEmpty()
 aiRobotRecipe.reset()
 Ciklus az aiRobotRecipe.getResources() minden elemére
 remove(aktuális elem)
 Létrehozunk egy AIRobot típusú air nevű objektumot
 place.accept(air)
 aiRobotRecipe.reset()
```

- **public void createGatePair():** A telepes lekérdezi a Game-től a teleportkapu-pár megépítéséhez szükséges receptet a Recipe getGatePairRecipe() metódusának meghívásával, majd pedig összehasonlíta a saját nyersanyagait a receptben lévőkkel. Amennyiben rendelkezik a szükséges nyersanyagokkal, és maximum 1 darab elkészített teleportkaput tárol, akkor létrehozza a TeleportingGate két példányát. Párba állítja őket a setPair(tg: TeleportingGate) függvények meghívásával, majd az űrhajóján a gatesCreated kollekcióban eltárolja őket. Végül meghívja a Recipe reset() függvényét, amelyben ezáltal visszaállítja a recept listájának a tartalmát.

Pszeudokód:

```

Ha gatesCreated mérete>1
 Metódus vége
 Létrehozunk egy Recipe típusú gatePairRecipe nevű
 objektumot
 gatePairRecipe=game.getGatePairRecipe()
 Ciklus a collectedResources minden elemére visszafelé
 Ha gatePairRecipe.isEmpty()
 ciklus vége
 GatePairRecipe.isNeeded(aktuális elem)
 Ha gatePairRecipe.isEmpty()
 GatePairRecipe.reset()
 Ciklus a gatePairRecipe.getResources() minden elemére
 remove(aktuális elem)
 Létrehozunk két TeleportingGate típusú tg1, tg2 nevű
 objektumot
 tg1.setPair(tg2)
 Hozzáadjuk a gatesCreated kollekcióhoz tg1-et és
 tg2-t
 GatePairRecipe.reset()
```

- **public void releaseGate():** A telepes a gatesCreated kollekcióból kiveszi az első teleportkaput, majd az aktuális aszteroidája körüli pályára állítja az Asteroid accept(TeleportingGate tg) függvényével. Ha a telepésnél nincs elkészült teleportkapu (gatesCreated üres), akkor nem történik semmi.

Pszeudokód:

```

Ha gatesCreated mérete>=1
 Place.accept(gatesCreated nulladik eleme)
 Kitöröljük a gatesCreated nulladik elemét
```

- **public void die():** A telepes meghal: megsemmisíti a nála levő teleportkapukat azok die() függvényeivel, eltávolítja magát az aszteroidájáról, végül meghívja a game removeSettler(Settler s) metódusát.

Pszeudokód:

```

Meghívja az ósosztály die() metódusát
Ciklus a gatesCreated minden elemére visszafelé
 die()
 game.removeSettler(aktuális Settler)
```

- **public void drill():** Csökkenti az aktuális aszteroida köpenyvastagságát: meghívja az aszteroida drilled() metódusát.
- **public void accept(TeleportingGate tg):** Hozzáadja a paraméterként kapott teleportkaput a gatesCreated kollekcióhoz.
- **public TeleportingGate[0..\*] getGatesCreated():** Visszaadja a gatesCreated kollekciót.
- **public void setGame(Game game):** Beállítja a game attribútumot a paraméterként kapotttra.

### 8.1.17 Sun

- **Felelősség**

A játékban levő Napot reprezentálja. A napviharok kiindulópontja, valamint a napközelség számításához szükséges.

- **Ősosztályok**

- **Interfészek**

ISteppable

- **Attribútumok**

- **private long timeToSunStorm:** a következő napvihar bekövetkezéséig hátralévő idő
- **private Asteroid[0..\*] neighboringAsteroids:** a Nappal közvetlenül szomszédos aszteroidák listája
- **private Game game:** a játékot reprezentáló osztály

- **Metódusok**

- **public void step():** Ha a timeToSunStorm attribútum pozitív, akkor eggyel csökkenti (várakozik). Ha pedig 0, akkor napvihart indít a sunStorm() függvényt hívva, majd a timeToSunStorm-nak beállít egy véletlen egész értéket

Pszeudokód:

```
Ha timeToSunStorm>0
 timeToSunStorm=timeToSunStorm-1
Különben
 sunStorm()
 timeToSunStorm=random szám
```

- **public void sunStorm():** Létrehoz egy aszteroidákat tároló kollekciót, amelyhez hozzáadja a neighboringAsteroids kollekció elemeit és az elemekkel szomszédos aszteroidákat is, majd pedig meghívja a létrehozott kollekció összes elemén a destroySurface() metódust.

Pszeudokód:

```
Létrehozunk egy Asteroid típus tárolására alkalmas list
nevű kollekciót
Ciklus a neighboringAsteroids minden a1 elemére
 Ciklus az a1 minden a2 szomszédjára
 Ha a list nem tartalmazza a2-t
 Hozzáadjuk a2-t a listhez
 Ha a list nem tartalmazza a1-et
 Hozzáadjuk a1-et a listhez
 Ciklus a list minden elemén
 destroySurface()
```

- **public boolean isCloseToSun(Asteroid a):** Megállapítja, hogy az a aszteroida napközeli-e, vagyis legfeljebb másodszomszédos-e a Nappal. Ehhez felhasználja a neighboringAsteroids listáját: megvizsgálja a benne szereplő aszteroidákat, valamint azok összes szomszédait is.

Pszeudokód:

```

Be: Asteroid a
Ha neighboringAsteroids tartalmazza a-t
 Ki: true
Ciklus a.getNeighbors() minden elemére
 Ha neighboringAsteroids tartalmazza az aktuális
 elemet
 Ki: true
 Ki: false

```

- **public void setGame(Game game):** Beállítja a game attribútumot a paraméterként kapotttra.
- **public void setTimeToSunStorm(long time):** Beállítja a timeToSunStorm attribútumot a paraméterként kapotttra.
- **public void addNeighbor(Asteroid a):** Hozzáadja a paraméterként kapott aszteroidát a neighboringAsteroids kollekcióhoz.

### 8.1.18 TeleportingGate

#### • Felelősség

Két teleportkapu egy teleportkapu-párt alkot, amely aszteroidák közötti összeköttetést biztosít. A frissen épített kapuk az építő telepes űrhajóján tárolódnak. Aszteroida körüli pályára állítás után az adott aszteroidához tartoznak. Ha az egyik kapu megsemmisül, a párja vele együtt automatikusan eltűnik, bárhol is van éppen.

#### • Ósosztályok

#### • Interfészek

ISteppable

#### • Attribútumok

- **private TeleportingGate pair:** az adott teleportkapu párja, amellyel összeköttetésben áll
- **private Asteroid asteroid:** az aszteroida, amely körül az adott teleportkapu kering
- **private Settler settler:** a telepes, aki tárolja a létrehozott, de még pályára nem állított teleportkaput
- **private Timer timer:** A játékban lévő időzítő.

#### • Metódusok

- **public TeleportingGate getPair():** Visszaadja a teleportkapu pair párját.
- **public void setPair(TeleportingGate tg):** Beállítja tg-t a teleportkapu párjaként.
- **public Asteroid getAsteroid():** Visszaadja az aszteroidát, amely körül a teleportkapu kering. Ha a kaput még nem állították pályára, akkor null-lal tér vissza.
- **public void setAsteroid(Asteroid a):** Beállítja a teleportkapu aszteroidáját, amely körül keringeni fog.

- **public void die()**: Eltávolítja a párját (pair) annak aszteroidájáról/űrhajójáról attól függően, hogy azt már pályára állították-e (remove(TeleportingGate tg)). Beállítja a párjának a párját null-ra (megszünteti az összeköttetést), végül saját magát is eltávolítja az aszteroidájáról/űrhajójáról, valamint ha a kapu meg volt már kergülve, akkor törli a Timer steppables kollekciójából saját magát. Ha a függvényhívás elején a teleportkapu párja már null, akkor csak magát távolítja el: ekkor a másik teleportkapu semmisült meg előbb, és az már megszüntette ennek a kapunak a kapcsolatait is.

Pszeudokód:

- ```

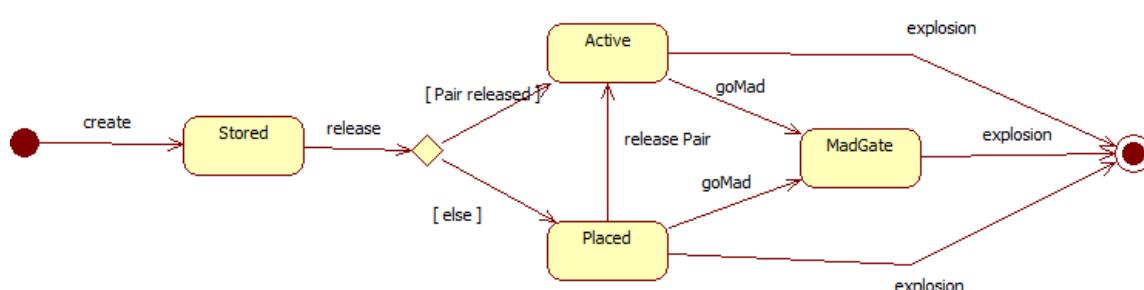
Ha pair!=null
    pair.setPair(null)
    pair.die()
    setPair(null)
Ha settler!=null
    settler.remove(aktuális TeleportingGate)
Különben
    asteroid.remove(aktuális TeleportingGate)
Ha      timer.getSteppables()      tartalmazza      aktuális
TeleportingGate-t
    timer.removeSteppable(aktuális TeleportingGate)
  
```
- **public void step()**: Meghívja a teleportkapu move(d: int) metódusát, paraméterként egy véletlenszerűen generált számot ad át. Ez csak a már megkergült teleportkapuknál következik be.
 - **public void goMad()**: Hozzáadja a timer steppables kollekciójához a teleportkaput, ami ezáltal megkergül.
 - **public void move(int d)**: Az aktuális aszteroida egy szomszédjára vándorol át, ehhez elvégzi a régi kapcsolat bontását és az újak felépítését is.

Pszeudokód:

```

Be: egész d szám
Létrehozunk egy Asteroid típusú a nevű objektumot
a=asteroid.getNeighbor(d)
asteroid.remove(aktuális TeleportingGate)
a.accept(aktuális TeleportingGate)
  
```

TeleportGate állapotgépe



8.1.19 Timer

- **Felelősség**

A Timer osztályt a beépített java.util.Timer osztályból származtattuk. A konstruktora Timer(long delay, long period) típusú lett, amelyben meg kell adnunk a kezdeti késleltetést, majd az időzített periódusok hosszát ezredmásodpercekben. Az osztály egy Timertask típusú belső osztályt is tartalmaz, amely az időzített eseményekért felel, és amelynek run() függvénye saját maga tick() függvényét hívja. A tick() metódus továbbra is léptet minden léptethető objektumot (steppables). A privát long típusú ticks adattagban számlálja is a játékban eltelt időt.

- **Ósosztályok**

Java.util.Timer

- **Interfészek**

- **Attribútumok**

- **private ISteppable[0..*] steppables:** a játék léptethető objektumainak listája
- **private long delay:** Az időzítő kezdeti késleltetése
- **private long period:** Az időzítő periodikus késleltetése
- **private TimerTask timertask:** Időzített taszk, amelyben a léptetések megvalósulnak. minden lépésben lépteti a játék összes léptethető objektumát és egyúttal számlálja a játékban eltelt időt.

- **Metódusok**

- **public void addStepable(Steppable s):** Hozzáadja s-t a steppables listához.
- **public void removeStepable(Steppable s):** Eltávolítja s-t a steppables listából.
- **public void tick():** minden steppable listabeli objektumot léptet úgy, hogy meghívja a step() függvényét.
- **public ISteppable[0..*] getStepables():** Visszatér a steppables kollekcióval.

8.1.20 UFO

- **Felelősség**

A karakterek egyik fajtája, amely minden időegységen bányászni vagy szomszédos aszteroidára mozogni tud.

- **Ősosztályok**

Character

- **Interfészek**

ISteppable

- **Attribútumok**

- **Metódusok**

- **public void step()**: Ha az aszteroida köpenyvastagsága nulla és tartalmaz nyersanyagot, akkor meghívja az UFO mine() metódusát, különben a move(int d) metódusát.

Pszeudokód:

```
Ha place.getRockLayerThickness ()=0 és
place.getResource!=null
    mine ()
Különben
    move (véletlen szám)
```

- **public void die()**: Az UFO megsemmisül: eltávolítja magát az aszteroidáról és a léptethető objektumok listájából.
- **public void mine()**: Az UFO megpróbálja kibányászni az adott aszteroida magjában található nyersanyagot. Ehhez meghívja az aszteroida minedByUFO() metódusát.

8.1.21 Uranium

- **Felelősség**

Az urán nyersanyagot reprezentálja, amely stabilitással rendelkezik. Az adott urán egység a harmadik napközeli felszínre kerülésekor (kifürás vagy nyersanyag-visszatöltés esetén) robban fel.

- **Ósosztályok**

RadioactiveResource → Resource

- **Interfészek**

- **Attribútumok**

- **private int life:** Ez az attribútum az adott urán egység aktuális stabilitását fejezi ki. Az urán minden napközeli felszínre kerülésekor (kifürás vagy nyersanyag-visszatöltés esetén) 1-gyel csökkentjük.

- **Metódusok**

- **public boolean isCompatibleWith(r: Resource):** Összehasonlítja magát a paraméterként kapott nyersanyaggal. Egyezés esetén igaz értékkel tér vissza, ellenkező esetben hamissal.
- **public void decLife():** A life attribútum értékét egygyel csökkenti.
- **public void drilledOut (Asteroid a):** Csökkenti a life attribútum értékét egygyel a decLife() függvény meghívásával, ha nullára csökken, azaz ha harmadjára kerül felszínre napközelben, akkor meghívja az aszteroida explodedBy(resource: RadioactiveResource) metódusát.

Pszeudokód:

```
Be: Asteroid a
decLife()
Ha life=0
    a.explodedBy(aktuális Uranium)
```

8.2 A tesztek részletes tervezet, leírásuk a teszt nyelvén

Minden tesztesetnél a Bemenet pontban a teszt kezdeti játékkonfigurációja szerepel. Csak a teszt futtatásához elengedhetetlen objektumokat tüntettük fel. Az objektumok attribútumai között is csak azokat soroltuk fel, amelyek a teszteset szempontjából meghatározóak; a tetszőleges érvényes értéket felvehető adattagok és a nem releváns, *null* értékű referenciák nem szerepelnek a bemenetek leírásában.

A Kimenet vázlatpontban az adott teszteset elvárt kimeneti játékkonfigurációjának objektumai és azok releváns attribútumai szerepelnek. A Bemenet és a Kimenet pontok közé beiktattunk egy *Kiadott parancs* vázlatpontot, amely a tesztre jellemző kiadandó parancs: ezzel indítható a tényleges tesztelt funkció. A tesztelő segédosztályban részletezzük a szerepét.

8.2.1 Pass

- **Leírás**

Leteszteljük azt a folyamatot, amikor a telepes passzol.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes nem végez érdemi lépést.

Várható hibahelyek:

- Megváltozik a modell.

- **Bemenet**

Player p1

Asteroid a1

Settler s1

Player p1

name Marci

settler s1

isAlive true

Asteroid a1

rockLayerThickness 1

characters s1

Settler s1

place a1

- **Kiadott parancs**

pass

- **Elvárt kimenet**

Player p1

Asteroid a1

Settler s1

Player p1

 name Marci

 settler s1

 isAlive true

Asteroid a1

 rockLayerThickness 1

 characters s1

Settler s1

 place a1

8.2.2 Win Game with moving

- **Leírás**

Annak a folyamatnak a modellezése, amikor egy telepes mozgásának eredményeképp összegyűlik egy aszteroidán az űrbázis építéséhez szükséges nyersanyagmennyiséget.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes szomszédos aszteroidára mozog, és az aszteroida ellenőrzi a rajta lévő telepeseknél található összes nyersanyagot, hogy az elegendő-e az űrbázis építéséhez.
Várható hibahelyek:

- A telepes nem tud szomszédos aszteroidára mozogni
- Nincs meg a telepeseknél a szükséges nyersanyagmennyiségek
- A játék nem ér véget

- **Bemenet**

Controller controller

Player p1 p2

Game game

Recipe nullRecipe1 nullRecipe2 spaceBaseRecipe

Asteroid a1 a2

Coal co1 co2 co3 co4 co5 co6

Iron ir1 ir2 ir3 ir4 ir5 ir6

Ice ic1 ic2 ic3 ic4 ic5 ic6

Uranium ur1 ur2 ur3 ur4 ur5 ur6

Settler s1 s2

Controller controller

game game

allPlayers p1 p2

actPlayer p1

state paused

Player p1

name Marci

settler s1

isAlive true

Player p2

name Balint

settler s2

isAlive true

Game game

recipes nullRecipe1 nullRecipe2 spaceBaseRecipe

asteroids a1 a2

settlersAlive s1 s2

Recipe spaceBaserecipe

resources co1 co2 co3 ir1 ir2 ir3 ic1 ic2 ic3 ur1 ur2 ur3

Asteroid a1
 rockLayerThickness 5
 game game
 neighbors a2
 characters s1

Asteroid a2
 rockLayerThickness 2
 game game
 neighbors a1
 characters s2

Uranium ur1
 life 3

Uranium ur2
 life 3

Uranium ur3
 life 3

Uranium ur4
 life 3

Uranium ur5
 life 3

Uranium ur6
 life 3

Settler s1
 game game
 place a1
 resources co4 ir4 ir5 ic4 ir6 ur4 co5 ic5

Settler s2
 game game
 place a2
 resources ur5 co6 ur6 ic6

- **Kiadott parancs**
 move 0

- **Elvárt kimenet**

Controller controller

Player p1 p2

Game game

Recipe nullRecipe1 nullRecipe2 spaceBaseRecipe

Asteroid a1 a2

Coal co1 co2 co3 co4 co5 co6

Iron ir1 ir2 ir3 ir4 ir5 ir6

Ice ic1 ic2 ic3 ic4 ic5 ic6

Uranium ur1 ur2 ur3 ur4 ur5 ur6

Settler s1 s2

Controller controller

game game

allPlayers p1 p2

actPlayer p2

state won

Player p1

name Marci

settler s1

isAlive true

Player p2

name Balint

settler s2

isAlive true

Game game

recipes nullRecipe1 nullRecipe2 spaceBaseRecipe

asteroids a1 a2

settlersAlive s1 s2

Recipe spaceBaserecipe

resources co1 co2 co3 ir1 ir2 ir3 ic1 ic2 ic3 ur1 ur2 ur3

Asteroid a1

rockLayerThickness 5

game game

neighbors a2

characters null

Asteroid a2

rockLayerThickness 2

game game

neighbors a1

characters s1 s2

Uranium ur1

life 3

Uranium ur2
life 3

Uranium ur3
life 3

Settler s1
game game
place a2
resources co4 ir4 ir5 ic4 ir6 ur4 co5 ic5

Settler s2
game game
place a2
resources ur5 co6 ur6 ic6

8.2.3 Win Game with mining

- **Leírás**

Annak a folyamatnak a modellezése, amikor egy telepes bányászásának eredményeképp összegyűlik az aszteroidáján az űrbázis építéséhez szükséges nyersanyagmennyiségek.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes bányászik az aszteroidáján, majd az aszteroida ellenőrzi a rajta lévő telepeseknél található összes nyersanyagot, hogy az elegendő-e az űrbázis építéséhez; a játék véget ér, ha az összegyűjtött nyersanyagok elegendőek ehhez.

Várható hibahelyek:

- A telepes nem tud bányászni, mert az aszteroida nincs megfűrva
- Az aszteroida üreges
- Nem lesz meg így sem a telepeseknél a szükséges nyersanyagmennyiségek
- A játék nem ér véget

- **Bemenet**

Controller controller

Player p1 p2

Game game

Recipe nullRecipe1 nullRecipe2 spaceBaseRecipe

Asteroid a1

Coal co1 co2 co3

Iron ir1 ir2 ir3

Ice ic1 ic2 ic3

Uranium ur1 ur2 ur3

Settler s1 s2

Controller controller

game game

allPlayers p1 p2

actPlayer p1

state paused

Player p1
 name Balazs
 settler s1
 isAlive true

Player p2
 name Balint
 settler s2
 isAlive true

Game game
 recipes nullRecipe1 nullRecipe2 spaceBaseRecipe
 asteroids a1
 settlersAlive s1 s2

Recipe spaceBaserecipe
 resources co1 co2 co3 ir1 ir2 ir3 ic1 ic2 ic3 ur1 ur2 ur3

Asteroid a1
 rockLayerThickness 0
 game game
 resource ir6
 characters s1 s2

Uranium ur1
 life 3

Uranium ur2
 life 3

Uranium ur3
 life 3

Uranium ur4
 life 3

Uranium ur5
 life 3

Uranium ur6
 life 3

Settler s1
 game game
 place a1

Settler s2
 game game
 place a1

- **Kiadott parancs**

mine

- **Elvárt kimenet**

Controller controller

Player p1 p2

Game game

Recipe nullRecipe1 nullRecipe2 spaceBaseRecipe

Asteroid a1

Coal co1 co2 co3

Iron ir1 ir2 ir3

Ice ic1 ic2 ic3

Uranium ur1 ur2 ur3

Settler s1 s2

Controller controller

game game

allPlayers p1 p2

actPlayer p2

state won

Player p1

name Balazs

settler s1

isAlive true

Player p2

name Balint

settler s2

isAlive true

Game game

recipes nullRecipe1 nullRecipe2 spaceBaserecipe

asteroids a1

settlersAlive s1 s2

Recipe spaceBaserecipe

resources co1 co2 co3 ir1 ir2 ir3 ic1 ic2 ic3 ur1 ur2 ur3

Asteroid a1

rockLayerThickness 0

game game

characters s1 s2

Uranium ur1

life 3

Uranium ur2

life 3

Uranium ur3
life 3

Settler s1
game game
place a1

Settler s2
game game
place a1

8.2.4 Move Spaceship neighboring Asteroid

- **Leírás**

Modellezük, ahogyan egy telepes egy szomszédos aszteroidára lép (nem teleportkapukon keresztül).

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes szomszédos aszteroidára mozog.

Várható hibahelyek:

- Az aszteroidának nincs aszteroida szomszédja

- **Bemenet**

Player p1
Game game
Asteroid a1 a2
Settler s1

Player p1
name Ricsi
settler s1
isAlive true

Game game
asteroids a1 a2
settlersAlive s1

Asteroid a1
rockLayerThickness 0
game game
neighbors a2
characters s1

Asteroid a2
rockLayerThickness 3
game game
neighbors a1
characters null

Settler s1
 game game
 place a1

- **Kiadott parancs**
`move 0`
- **Elvárt kimenet**
 Player p1
 Game game
 Asteroid a1 a2
 Settler s1
 Player p1
 name Ricsi
 settler s1
 isAlive true
 Game game
 asteroids a1 a2
 settlersAlive s1
 Asteroid a1
 rockLayerThickness 0
 game game
 neighbors a2
 characters null
 Asteroid a2
 rockLayerThickness 3
 game game
 neighbors a1
 characters s1
 Settler s1
 game game
 place a2

8.2.5 Move Spaceship not neighboring Asteroid

- **Leírás**
 Ellenőrizzük, ahogyan egy telepes nem tud szomszédos aszteroidára lépni, mert az aszteroidájának nincs szomszéda.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
 A telepes szomszédos aszteroidára mozogna, de nem sikerül.
 Várható hibahelyek:
 - A telepes mégis másik aszteroidára mozog.
 - A szomszédos aszteroida kiválasztásakor kivételt kapunk.

- **Bemenet**

Player p1

Game game

Asteroid a1 a2

Settler s1

Player p1

name Barni

settler s1

isAlive true

Game game

asteroids a1

settlersAlive s1

Asteroid a1

rockLayerThickness 2

game game

neighbors null

characters s1

Settler s1

game game

place a1

- **Kiadott parancs**

move 0

- **Elvárt kimenet**

Player p1

Game game

Asteroid a1 a2

Settler s1

Player p1

name Barni

settler s1

isAlive true

Game game

asteroids a1

settlersAlive s1

Asteroid a1

rockLayerThickness 2

game game

neighbors null

characters s1

```
Settler s1
  game game
  place a1
```

8.2.6 Move Spaceship through TeleportingGate

- **Leírás**

Modellezük a folyamatot, ahogyan egy telepes szomszédos aszteroidára lép, teleportkapu-páron keresztül.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes szomszédos aszteroidára mozog teleportkapu-páron keresztül.

Várható hibahelyek:

- A telepes aszteroidája körül nem kering kapu.
- A kapu párja nincs lerakva aszteroida mellé.

- **Bemenet**

Player p1

Game game

Asteroid a1 a2

Settler s1

TeleportingGate tg1 tg2

Player p1

name Balint

settler s1

isAlive true

Game game

asteroids a1 a2

settlersAlive s1

Asteroid a1

rockLayerThickness 3

game game

neighbors null

characters s1

gates tg1

Asteroid a2

rockLayerThickness 1

game game

sun null

neighbors null

characters null

gates tg2

Settler s1

game game

place a1

TeleportingGate tg1

pair tg2

asteroid a1

TeleportingGate tg2

pair tg1

asteroid a2

- **Kiadott parancs**
move 0

- **Elvárt kimenet**

Player p1

Game game

Asteroid a1 a2

Settler s1

TeleportingGate tg1 tg2

Player p1

name Balint

settler s1

isAlive true

Game game

asteroids a1 a2

settlersAlive s1

Asteroid a1

rockLayerThickness 3

game game

neighbors null

characters s1

gates tg1

Asteroid a2

rockLayerThickness 1

game game

neighbors null

characters null

gates tg2

Settler s1

game game

place a1

TeleportingGate tg1

pair tg2

asteroid a1

```
TeleportingGate tg2
  pair tg1
  asteroid a2
```

8.2.7 Move Spaceship through a single TeleportingGate

- **Leírás**

Ellenőrzük, ahogyan egy telepes szomszédos aszteroidára lépne teleportkapun keresztül, de ez nem sikerül, mert a kapu párja nincs lerakva.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes teleportkapu általi szomszédos aszteroidára lépne.

Várható hibahelyek:

- A telepes aszteroidája körül nem kering kapu
- Nincs párja a kapunak
- A kapu párja nincs lerakva, mégis sikerül a mozgás

- **Bemenet**

Player p1

Game game

Asteroid a1

Settler s1

TeleportingGate tg1 tg2

Player p1

name Balint

settler s1

isAlive true

Game game

asteroids a1 a2

settlersAlive s1

Asteroid a1

rockLayerThickness 0

game game

neighbors null

characters s1

gates tg1

Asteroid a2

rockLayerThickness 1

game game

neighbors null

characters null

gates null

Settler s1

game game

place a1

gatesCreated tg2

TeleportingGate tg1

pair tg2
asteroid a1

TeleportingGate tg2

pair tg1
settler s1
asteroid null

- **Kiadott parancs**
move 0

- **Elvárt kimenet**

Player p1
Game game
Asteroid a1
Settler s1
TeleportingGate tg1 tg2

Player p1
name Balint
settler s1
isAlive true

Game game
asteroids a1 a2
settlersAlive s1

Asteroid a1
rockLayerThickness 0
game game
neighbors null
characters s1
gates tg1

Asteroid a2
rockLayerThickness 1
game game
neighbors null
characters null
gates null

Settler s1
game game
place a1
gatesCreated tg2

```
TeleportingGate tg1
  pair tg2
    asteroid a1
```

```
TeleportingGate tg2
  pair tg1
    settler s1
    asteroid null
```

8.2.8 Drill Asteroid one layer

- **Leírás**

Modellezük, hogy egy telepes egy réteget lefűr az aszteroidája köpenyén, amelyen tartózkodik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes fúrja az aszteroidát.

Várható hibahelyek:

- Az aszteroida köpenyvastagsága 0 (már meg van fúrva).
- A köpenyvastagság nem pontosan 1-gyel csökken.

- **Bemenet**

Player p1

Game game

Asteroid a1

Settler s1

Player p1

name Balazs

settler s1

isAlive true

Game game

asteroids a1

settlersAlive s1

Asteroid a1

rockLayerThickness 2

game game

characters s1

Settler s1

game game

place a1

- **Kiadott parancs**
drill

- **Elvárt kimenet**

Player p1

Game game

Asteroid a1

Settler s1

Player p1

 name Balazs

 settler s1

 isAlive true

Game game

 asteroids a1

 settlersAlive s1

Asteroid a1

 rockLayerThickness 1

 game game

 characters s1

Settler s1

 game game

 place a1

8.2.9 Drill Asteroid fail

- **Leírás**

Modellezük, hogy egy telepes már megfúrt aszteroida köpenyét fúrná.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes fúrja az aszteroidát, de a fúrásnak nincs hatása.

Várható hibahelyek:

- A köpenyvastagság nem marad 0.

- **Bemenet**

Player p1

Game game

Asteroid a1

Settler s1

Player p1

 name Balazs

 settler s1

 isAlive true

Game game

 asteroids a1

 settlersAlive s1

Asteroid a1

 rockLayerThickness 0

 game game

 characters s1

Settler s1

 game game

 place a1

- **Kiadott parancs**

drill

- **Elvárt kimenet**

Player p1

Game game

Asteroid a1

Settler s1

Player p1

name Balazs

settler s1

isAlive true

Game game

asteroids a1

settlersAlive s1

Asteroid a1

rockLayerThickness 0

game game

characters s1

Settler s1

game game

place a1

8.2.10 Drill radioactive Asteroid's last layer near Sun

- **Leírás**

Modellezük, hogy egy telepes az utolsó réteget fúrja le a napközeli aszteroidája köpenyén, amely ráadásul 1 stabilitású uránt tartalmaz a magjában. A fúrás eredményeképp az urán és az aszteroida felrobban, a rajta levő telepes meghal, (a nyersanyagai is elvesznek), a robot szomszédos aszteroidára repül, és a teleportkapuja is megsemmisül. Mivel a telepes az egyetlen élő telepes volt a játékban, a játék a telepesek vereségével véget ér.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes fúrja az aszteroidát, a radiaktív nyersanyag felrobbantja az aszteroidát, a rajta levő karakterek és teleportkapuk reagálnak a robbanásra. A játék véget ér a telepesek vereségével.

Várható hibahelyek:

- A köpenyvastagság nem 0-ra csökken
- Az urán nem robban fel
- A telepes nem hal meg, és a játék nem ér véget
- A robot nem landol a szomszédos aszteroidán
- A teleportkapuk nem semmisülnek meg

- **Bemenet**

Controller controller
 Player p1
 Game game
 Timer timer
 Sun sun
 Asteroid a1 a2
 Iron ir1
 Uranium ur1
 Settler s1
 AIRobot air1
 UFO ufo1
 TeleportingGate tg1 tg2

Controller controller
 game game
 allPlayers p1
 actPlayer p1
 state paused

Player p1
 name Balint
 settler s1
 isAlive true

Game game
 timer timer
 sun sun
 asteroids a1 a2
 settlersAlive s1

Timer timer
 steppables sun air1 ufo1

Sun sun
 neighboringAsteroids a1

Asteroid a1
 rockLayerThickness 1
 game game
 sun sun
 neighbors a2
 resource ur1
 characters s1 air1 ufo1
 gates tg1

Asteroid a2
rockLayerThickness 3
game game
sun sun
neighbors a1
characters null

Uranium ur1
life 1

Settler s1
game game
timer timer
place a1
resources ir1
gatesCreated tg2

AIRobot air1
timer timer
place a1

UFO ufo1
timer timer
place a1

TeleportingGate tg1
timer null
pair tg2
settler null
asteroid a1

TeleportingGate tg2
timer null
pair tg1
settler s1
asteroid null

- **Kiadott parancs**
drill

- **Elvárt kimenet**

Controller controller

Player p1

Game game

Timer timer

Sun sun

Asteroid a2

AIRobot air1

Controller controller

game game

allPlayers p1

actPlayer null

state lost

Player p1

name Balint

settler s1

isAlive false

Game game

timer timer

sun sun

asteroids a2

settlersAlive null

Timer timer

steppables sun air1

Sun sun

neighboringAsteroids null

Asteroid a2

rockLayerThickness 3

game game

sun sun

neighbors a1

characters null

AIRobot air1

timer timer

place a2

8.2.11 Drill icy Asteroid's last layer near Sun

- **Leírás**

Modellezük, ahogyan egy telepes a vízjég nyersanyagot tartalmazó, napközeli aszteroidája utolsó rétegét fúrja. Ennek hatására a vízjég elszublimál.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes fúrja az aszteroidát, amelyből a vízjég elszublimál.

Várható hibahelyek:

- Az aszteroida köpenyvastagsága nem 0 lesz.
- A vízjég nem szublimál el.

- **Bemenet**

Player p1

Game game

Sun sun

Asteroid a1

Ice ic1

Settler s1

Player p1

 name Marci

 settler s1

 isAlive true

Game game

 sun sun

 asteroids a1

 settlersAlive s1

Sun sun

 neighboringAsteroids a1

Asteroid a1

 rockLayerThickness 1

 game game

 sun sun

 resource ic1

 characters s1

Settler s1

 game game

 place a1

- **Kiadott parancs**

drill

- **Elvárt kimenet**

Player p1

Game game

Timer timer

Sun sun

Asteroid a1

Settler s1

Player p1

 name Marci

 settler s1

 isAlive true

Game game

 sun sun

 asteroids a1

 settlersAlive s1

Sun sun

 neighboringAsteroids a1

Asteroid a1

 rockLayerThickness 0

 game game

 sun sun

 resource null

 characters s1

Settler s1

 game game

 timer timer

 place a1

8.2.12 Restore Resource

- **Leírás**

Annak a folyamatnak a tesztelése, amikor a telepes nyersanyagot helyez el sikeresen az aszteroida belsejébe.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Lellenőrzük azt a folyamatot, amikor a telepes sikeresen elhelyez egy nyersanyagot egy aszteroida belsejében.

Várható hibahelyek:

- Nincs a telepesnél nyersanyag
- Az aszteroida magja nem üres
- Az aszteroida Vízjeget próbál visszatenni

- **Bemenet**

Player p1

Asteroid a1

Coal co1

Settler s1

Player p1

```
name Marci
settler s1
isAlive true
```

Asteroid a1

```
rockLayerThickness 0
characters s1
```

Settler s1

```
place a1
resources co1
```

- **Kiadott parancs:**

```
restore co1
```

- **Elvárt kimenet**

Player p1

Asteroid a1

Coal co1

Settler s1

Player p1

```
name Marci
settler s1
isAlive true
```

Asteroid a1

```
rockLayerThickness 0
resource c1
characters s1
```

Settler s1

```
place a1
```

8.2.13 Restore Resource with empty container

- **Leírás**

Annak a folyamatnak a tesztelése, amikor a telepes nyersanyagot helyezne el az aszteroida belsejébe, de a tárolója üres.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Lellenőrizzük azt a folyamatot, amikor a telepes nyersanyagot helyezne el az aszteroida belsejébe, de a tárolója üres, ilyenkor a telepes nem képes letenni semmit.

Várható hibahelyek:

- Van a telepesnél nyersanyag
- Az aszteroida magja nem üres
- Az aszteroida Vízjeget próbál visszatenni

- **Bemenet**

Player p1

Asteroid a1

Settler s1

Player p1

```
name Marci
settler s1
isAlive true
```

Asteroid a1

```
rockLayerThickness 0
characters s1
```

Settler s1

```
place a1
```

- **Kiadott parancs:**
restore ur1

- **Elvárt kimenet**

Player p1

Asteroid a1

Settler s1

Player p1

```
name Marci
settler s1
isAlive true
```

Asteroid a1

```
rockLayerThickness 0
characters s1
```

Settler s1

```
place a1
```

8.2.14 Restore Resource Fail not empty

- **Leírás**

Annak a folyamatnak a lemodellezése, amikor a telepes egy nyersanyagot próbál meg elhelyezni az aszteroida belsejébe, azonban ott már van nyersanyag.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A nyersanyag elhelyezése meghiúsul, mert az aszteroida magjában található nyersanyag.

Várható hibahelyek:

- A telepesnél nincs nyersanyag
- Az aszteroida magja üres

- **Bemenet**

Player p1

Asteroid a1

Coal co1, co2

Settler s1

Player p1

 name Marci

 settler s1

 isAlive true

Asteroid a1

 rockLayerThickness 0

 resource co2

 characters s1

Settler s1

 place a1

 resources co1

- **Kiadott parancs:**

 restore co1

- **Kimenet**

Player p1

Asteroid a1

Coal co1, co2

Settler s1

Player p1

 name Marci

 settler s1

 isAlive true

Asteroid a1

 rockLayerThickness 0

 resource co2

 characters s1

Settler s1

 place a1

 resources co1

8.2.15 Restore RadiactiveResource near Sun

- **Leírás**

Annak a folyamatnak a tesztelése, amikor a telepes egy nyersanyagot próbál meg elhelyezni az aszteroida belsejébe, azonban mindezt napközelben teszi, így az aktiválódik, teljesen instabil lesz és felrobban.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A nyersanyag elhelyezése után felrobban az Aszteroida.

Várható hibahelyek:

- Az telepes nem tud elhelyezni nyersanyagot.
- Az aszteroida nem robban fel.
- A telepes nem robban fel.

- **Bemenet**

Controller controller

Player p1

Game game

Asteroid a1

Uranium ur1

Settler s1

Controller controller

allPlayers p1

actPlayer p1

gameRunning true

Player p1

 name Marci

 settler s1

 isAlive true

Game game

 sun sun

 asteroids a1

 settlersAlive s1

Sun sun t

 TimeToSunStorm 3000

 neighboringAsteroids a1

Asteroid a1

 rockLayerThickness 0

 game game

 sun sun

 characters s1

Uranium ur1

 life 1

Settler s1

 place a1

 resources ur1

- **Kiadott parancs:**
restore ur1

- **Elvárt kimenet**

Controller controller

Player p1

Uranium ur1

Controller controller

gameRunning false

Player p1

name Marci

settler s1

isAlive false

Game game

sun sun

asteroids a1

Sun sun

TimeToSunStorm 2000

8.2.16 Restore IceResource near Sun

- **Leírás**

Annak a folyamatnak a lemodellezése, amikor a telepes egy nyersanyagot próbál meg elhelyezni az aszteroida belsejébe, azonban mindezt napközelben teszi, így az elpárolog.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A nyersanyag elhelyezése után a Vízjég elszublimál.

Várható hibahelyek:

- Az telepes nem tud elhelyezni nyersanyagot.
- A Vízjég nem szublimál el.

- **Bemenet**

Player p1

Asteroid a1

Ice ic1

Settler s1

Player p1

name Marci

settler s1

isAlive true

Sun sun t

TimeToSunStorm 3000

neighboringAsteroids a1

Asteroid a1

rockLayerThickness 0
sun sun
characters s1

Settler s1

place a1
resources ic1

- **Kiadott parancs:**
`restore ic1`

- **Elvárt kimenet**

Player p1

Asteroid a1

Settler s1

Player p1

name Marci
settler s1
isAlive true

Sun sun t

TimeToSunStorm 2000
neighboringAsteroids a1

Asteroid a1

rockLayerThickness 0
sun sun
characters s1

Settler s1

place a1

8.2.17 Mine Resource

- **Leírás**

Leteszteljük azt a folyamatot, amikor a telepes kinyeri nyersanyagot az aszteroida belsejéből.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes birtokolja az aszteroidából kinyert nyersanyagot.

Várható hibahelyek:

- Az telepes nem tud felvenni nyersanyagot, mert tele van a tárolója.
- Az aszteroida köpenye nincs átfúrva.
- Az aszteroida magja üres.

- **Bemenet**

Player p1

Asteroid a1

Coal co1

Settler s1

Player p1

 name Marci

 settler s1

 isAlive true

Asteroid a1

 rockLayerThickness 0

 resource co1

 characters s1

Settler s1

 place a1

- **Kiadott parancs:**

mine

- **Elvárt kimenet**

Player p1

Asteroid a1

Coal co1

Settler s1

Player p1

 name Marci

 settler s1

 isAlive true

Asteroid a1

 rockLayerThickness 0

 characters s1

Settler s1

 place a1

 resources co1

8.2.18 Mine Resource fail when Asteroid is empty

- **Leírás**

Leteszteljük azt a folyamatot, amikor a telepes megpróbálja kinyerni a nyersanyagot az aszteroida belsejéből, azonban az üres.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A nyersanyag kinyerése meghiúsul.

Várható hibahelyek:

- Az aszteroida köpenye nincs átfúrva.
- Az aszteroida magja nem üres.

- **Bemenet**

Player p1

Asteroid a1

Settler s1

Player p1

```
name Marci
settler s1
isAlive true
```

Asteroid a1

```
rockLayerThickness 0
characters s1
```

Settler s1

```
place a1
```

- **Kiadott parancs:**
mine

- **Elvárt kimenet**

Player p1

Asteroid a1

Settler s1

Player p1

```
name Marci
settler s1
isAlive true
```

Asteroid a1

```
rockLayerThickness 0
characters s1
```

Settler s1

```
place a1
```

8.2.19 Mine Resource fail when Spaceship is full

- **Leírás**

Leteszteljük azt a folyamatot, amikor a telepes megpróbálja kinyeri a nyersanyagot az aszteroida belsejéből, azonban az nincs hely az űrhajóján.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A nyersanyag kinyerése meghiúsul.

Várható hibahelyek:

- A telepes tárolója nincs tele
- A telepes nem képes nyersanyag kinyerésére

- **Bemenet**

Player p1

Asteroid a1

Coal co1

Iron ir1 ir2 ir3 ir4 ir5 ir6 ir7 ir8 ir9 ir10

Settler s1

Player p1

 name Marci

 settler s1

 isAlive true

Asteroid a1

 rockLayerThickness 0

 resource co1

 characters s1

Settler s1

 place a1

 resources ir1 ir2 ir3 ir4 ir5 ir6 ir7 ir8 ir9 ir10

- **Kiadott parancs:**
mine

- **Elvárt kimenet**

Player p1

Asteroid a1

Coal co1

Iron ir1 ir2 ir3 ir4 ir5 ir6 ir7 ir8 ir9 ir10

Settler s1

Player p1

 name Marci

 settler s1

 isAlive true

Asteroid a1

 rockLayerThickness 0

 resource co1

 characters s1

Settler s1

```
place a1
resources ir1 ir2 ir3 ir4 ir5 ir6 ir7 ir8 ir9 ir10
```

8.2.20 Mine Resource fail when Asteroid is not drilled

- **Leírás**

Leteszteljük azt a folyamatot, amikor a telepes megpróbálja kinyeri a nyersanyagot az aszteroida belsejéből, azonban az aszteroida nincs kifúrva.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A nyersanyag kinyerése meghiúsul.

Várható hibahelyek:

- Az aszteroida köpenye ki van fúrva
- A telepes nem képes nyersanyag kinyerésére

- **Bemenet**

Player p1

Asteroid a1

Coal co1

Settler s1

Player p1

```
name Marci
settler s1
isAlive true
```

Asteroid a1

```
rockLayerThickness 1
resource co1
characters s1
```

Settler s1

```
place a1
```

- **Kiadott parancs:**
mine

- **Elvárt kimenet**

Player p1

Asteroid a1

Coal co1

Settler s1

Player p1

 name Marci

 settler s1

 isAlive true

Asteroid a1

 rockLayerThickness 1

 resource co1

 characters s1

Settler s1

 place a1

8.2.21 Build AIRobot

- **Leírás**

Annak a folyamatnak a lemodellezése, amikor a telepes egy új AIRobotot hoz létre, feltételezzük, hogy rendelkezésére áll az összes szükséges nyersanyag.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Új AIRobot keletkezik.

Várható hibahelyek:

- Nincs elég nyersanyag a robot megépítéséhez
- Nem keletkezik új AIRobot

- **Bemenet**

Controller controller

Player p1

Game game

Recipe robotRecipe

Timer timer

Sun sun

Asteroid a1

Coal co1

Iron ir1

Uranium ur1

Settler s1

Controller

game game

allPlayers p1

actPlayer p1

gameRunning true

Player p1

name Bela

settler s1

isAlive true

Game game

timer timer

recipes robotRecipe

sun sun

asteroids a1

settlersAlive s1

Recipe robotRecipe

resources co1 ir1 ur1

Timer timer

ticks 0

delay 5000

period 500

Sun sun
 timeToSunStorm 3000
 neighboringAsteroids a1

Asteroid a1
 game game
 sun sun
 characters s1

Uranium ur1
 life 3

Settler s1
 game game
 timer timer
 place a1
 resources co1 ir1 ur1

- **Kiadott parancs:**
create_robot

- **Elvárt kimenet**

Controller controller
 Player p1
 Game game
 Recipe robotRecipe
 Timer timer
 Sun sun
 Asteroid a1
 Coal co1
 Iron ir1
 Uranium ur1
 Settler s1

Controller
 game game
 allPlayers p1
 actPlayer p1
 gameRunning true

Player p1
 name Bela
 settler s1
 isAlive true

Game game
timer timer
recipes robotRecipe
sun sun
asteroids a1
settlersAlive s1

Recipe robotRecipe
resources co1 ir1 ur1

Timer timer
ticks 0
delay 5000
period 500
steppables air1

Sun sun
timeToSunStorm 3000
neighboringAsteroids a1

Asteroid a1
game game
sun sun
characters s1 air1

Settler s1
game game
timer timer
place a1

AIRobot air1
timer timer
place a1

8.2.22 Build AIRobot fail

- **Leírás**

Annak a folyamatnak a lemodellezése, amikor a telepes egy új AIRobotot hoz létre, azonban nem áll rendelkezésre az összes szükséges nyersanyag.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Nem keletkezik új AIRobot.

Várható hibahelyek:

- Rendelkezésre áll a szükséges nyersanyag
- Új AIRobot keletkezik

- **Bemenet**

Controller controller

Player p1

Game game

Recipe robotRecipe

Timer timer

Sun sun

Asteroid a1

Coal co1

Iron ir1

Uranium ur1

Settler s1

Controller

game game

allPlayers p1

actPlayer p1

gameRunning true

Player p1

name Bela

settler s1

isAlive true

Game game

timer timer

recipes robotRecipe

sun sun

asteroids a1

settlersAlive s1

Recipe robotRecipe

resources co1 ir1 ur1

Timer timer

ticks 0

delay 5000

period 500

steppables null

Sun sun
 timeToSunStorm 3000
 neighboringAsteroids a1

Asteroid a1
 game game
 sun sun
 neighbors null
 resource null
 characters s1

Uranium ur1
 life 3

Settler s1
 game game
 timer timer
 place a1
 resources co1 ir1
 gatesCreated null

- **Kiadott parancs:**
create_robot

- **Elvárt kimenet**

Controller controller
 Player p1
 Game game
 Recipe robotRecipe
 Timer timer
 Sun sun
 Asteroid a1
 Coal co1
 Iron ir1
 Uranium ur1
 Settler s1

Controller
 game game
 allPlayers p1
 actPlayer p1
 gameRunning true

Player p1
 name Bela
 settler s1
 isAlive true

Game game
timer timer
recipes robotRecipe
sun sun
asteroids a1
settlersAlive s1

Recipe robotRecipe
resources co1 ir1 ur1

Timer timer
ticks 0
delay 5000
period 500
steppables null

Sun sun
timeToSunStorm 3000
neighboringAsteroids a1

Asteroid a1
game game
sun sun
neighbors null
resource null
characters s1

Uranium ur1
life 3

Settler s1
game game
timer timer
place a1
resources co1 ir1
gatesCreated null

8.2.23 Build TeleportingGate pair

- **Leírás**

Annak a folyamatnak a lemodellezése, amikor a telepes egy új Teleportkapu pár hoz létre, feltételezzük, hogy rendelkezésére áll az összes szükséges nyersanyag.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Új teleportkapu pár keletkezik

Várható hibahelyek:

- Nem keletkezik új teleportkapu pár
- Nem áll rendelkezésére a szükséges nyersanyag

- **Bemenet**

Controller controller

Player p1

Game game

Recipe gatePairRecipe

Timer timer

Sun sun

Asteroid a1

Iron ir2

Iron ir3

Ice ic1

Uranium ur2

Settler s1

Controller

game game

allPlayers p1

actPlayer p1

gameRunning true

Player p1

name Bela

settler s1

isAlive true

Game game

timer timer

recipes gatePairRecipe

sun sun

asteroids a1

settlersAlive s1

Recipe gatePairRecipe

resources ir2 ir3 ic1 ur2

Timer timer
 ticks 0
 delay 5000
 period 500
 steppables

Sun sun
 timeToSunStorm 3000
 neighboringAsteroids a1

Asteroid a1
 game game
 sun sun
 neighbors
 resource null
 characters s1

Uranium ur2
 life 3

Settler s1
 game game
 timer timer
 place a1
 resources co2 ir2 ic2 ur2
 gatesCreated null

- **Kiadott parancs:**
create_gate_pair

- **Elvárt kimenet**

Controller controller

Player p1

Game game

Recipe gatePairRecipe

Timer timer

Sun sun

Asteroid a1

Iron ir2

Iron ir3

Ice ic1

Uranium ur2

Settler s1

Controller
 game game
 allPlayers p1
 actPlayer p1
 gameRunning true

Player p1
 name Bela
 settler s1
 isAlive true

Game game
 timer timer
 recipes gatePairRecipe
 sun sun
 asteroids a1
 settlersAlive s1

Recipe gatePairRecipe
 resources ir2 ir3 ic1 ur2

Timer timer
 ticks 0
 delay 5000
 period 500
 steppables null

Sun sun
 timeToSunStorm 3000
 neighboringAsteroids a1

Asteroid a1
 game game
 sun sun
 neighbors
 resource null
 characters s1

Settler s1
 game game
 timer timer
 place a1
 resources null
 gatesCreated tg1 tg2

TeleportingGate tg1
 timer timer
 pair tg2
 settler s1
 asteroid null

TeleportingGate tg2
 timer null
 pair tg1
 settler s1
 asteroid null

8.2.24 Build TeleportingGate pair fails when Player has not enough Resources

- **Leírás**

Annak a folyamatnak a lemodellezése, amikor a telepes egy új Teleportkapu-párt hozna létre, azonban nem áll rendelkezésre az összes szükséges nyersanyag.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Nem keletkezik új Teleportkapu-pár

Várható hibahelyek:

- Rendelkezésre áll a szükséges nyersanyag
- Új teleportkapu pár keletkezik

- **Bemenet**

Controller controller

Player p1

Game game

Recipe gatePairRecipe

Timer timer

Sun sun

Asteroid a1

Iron ir2

Iron ir3

Ice ic1

Uranium ur2

Settler s1

Controller

game game

allPlayers p1

actPlayer p1

gameRunning true

Player p1

name Bela

settler s1

isAlive true

Game game

timer timer

recipes gatePairRecipe

sun sun

asteroids a1

settlersAlive s1

Recipe gatePairRecipe

resources ir2 ir3 ic1 ur2

Timer timer
 ticks 0
 delay 5000
 period 500
 steppables

Sun sun
 timeToSunStorm 3000
 neighboringAsteroids a1

Asteroid a1
 game game
 sun sun
 neighbors
 resource null
 characters s1

Uranium ur2
 life 3

Settler s1
 game game
 timer timer
 place a1
 resources co2 ir2 ur2
 gatesCreated null

- **Kiadott parancs:**
create_gate_pair

- **Elvárt kimenet**

Controller controller

Player p1

Game game

Recipe gatePairRecipe

Timer timer

Sun sun

Asteroid a1

Iron ir2

Iron ir3

Ice ic1

Uranium ur2

Settler s1

Controller
 game game
 allPlayers p1
 actPlayer p1
 gameRunning true

Player p1
name Bela
settler s1
isAlive true

Game game
timer timer
recipes gatePairRecipe
sun sun
asteroids a1
settlersAlive s1

Recipe gatePairRecipe
resources ir2 ir3 ic1 ur2

Timer timer
ticks 0
delay 5000
period 500
steppables

Sun sun
timeToSunStorm 3000
neighboringAsteroids a1

Asteroid a1
game game
sun sun
neighbors
resource null
characters s1

Uranium ur2
life 3

Settler s1
game game
timer timer
place a1
resources co2 ir2 ur2
gatesCreated null

8.2.25 Build TeleportingGate pair fails when Player has not enough space

- **Leírás**

Annak a folyamatnak a lemodellezése, amikor a telepes egy új Teleportkapu-párt hozna létre, azonban nem áll rendelkezésre elegendő tárhely az ūrhajóban.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Nem keletkezik új Teleportkapu-pár

Várható hibahelyek:

- Új teleportkapu pár keletkezik
- Nem veszi figyelembe a helykorlátot

- **Bemenet**

Controller controller

Player p1

Game game

Recipe gatePairRecipe

Timer timer

Sun sun

Asteroid a1

Iron ir2

Iron ir3

Ice ic1

Uranium ur2

Settler s1

TeleportingGate tg1 tg2 tg3 tg4 tg5 tg6

Controller

game game

allPlayers p1

actPlayer p1

gameRunning true

Player p1

name Bela

settler s1

isAlive true

Game game

timer timer

recipes gatePairRecipe

sun sun

asteroids a1

settlersAlive s1

Recipe gatePairRecipe

resources ir2 ir3 ic1 ur2

Timer timer

ticks 0

delay 5000

period 500
steppables

Sun sun
timeToSunStorm 3000
neighboringAsteroids a1

Asteroid a1
game game
sun sun
neighbors
resource null
characters s1

Uranium ur2
life 3

Settler s1
game game
timer timer
place a1
resources ir2 ir3 ic1 ur2
gatesCreated tg1 tg2 tg3 tg4 tg5 tg6

TeleportingGate tg1
timer timer
pair tg2
settler s1
asteroid null

TeleportingGate tg2
timer null
pair tg1
settler s1
asteroid null

TeleportingGate tg3
timer null
pair tg4
settler s1
asteroid null

TeleportingGate tg4
timer null
pair tg3
settler s1
asteroid null

TeleportingGate tg5
timer null

pair tg6
settler s1
asteroid null

TeleportingGate tg6
timer null
pair tg5
settler s1
asteroid null

- **Kiadott parancs:**
create_gate_pair

- **Elvárt kimenet**

Controller controller
Player p1
Game game
Recipe gatePairRecipe
Timer timer
Sun sun
Asteroid a1
Iron ir2
Iron ir3
Ice ic1
Uranium ur2
Settler s1
TeleportingGate tg1 tg2 tg3 tg4 tg5 tg6

Controller
game game
allPlayers p1
actPlayer p1
gameRunning true

Player p1
name Bela
settler s1
isAlive true

Game game
timer timer
recipes gatePairRecipe
sun sun
asteroids a1
settlersAlive s1

Recipe gatePairRecipe
resources ir2 ir3 ic1 ur2

Timer timer
 ticks 0
 delay 5000
 period 500
 steppables

Sun sun
 timeToSunStorm 3000
 neighboringAsteroids a1

Asteroid a1
 game game
 sun sun
 neighbors
 resource null
 characters s1

Uranium ur2
 life 3

Settler s1
 game game
 timer timer
 place a1
 resources ir2 ir3 ic1 ur2
 gatesCreated tg1 tg2 tg3 tg4 tg5 tg6

TeleportingGate tg1
 timer timer
 pair tg2
 settler s1
 asteroid null

TeleportingGate tg2
 timer null
 pair tg1
 settler s1
 asteroid null

TeleportingGate tg3
 timer null
 pair tg4
 settler s1
 asteroid null

TeleportingGate tg4
 timer null
 pair tg3
 settler s1
 asteroid null

TeleportingGate tg5

timer null

pair tg6

settler s1

asteroid null

TeleportingGate tg6

timer null

pair tg5

settler s1

asteroid null

8.2.26 Place TeleportingGate

- **Leírás**

A telepes lerakja a teleportkapu-pár egyik tagját az aszteroidára, feltételezzük, hogy rendelkezik a szükséges kapuval.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az aszteroida megkapja a kaput

Várható hibahelyek:

- Mindkét teleportkapu párt lerakja ugyan ahhoz az aszteroidához
- Nem kerül lerakásra csak az egyik kapu

- **Bemenet**

Controller controller

Player p1

Game game

Timer timer

Sun sun

Asteroid a1

Iron ir2

Iron ir3

Ice ic1

Uranium ur2

Settler s1

TeleportingGate tg1 tg2

Controller

game game

allPlayers p1

actPlayer p1

gameRunning true

Player p1

name Bela

settler s1

isAlive true

Game game
 timer timer
 recipes null
 sun sun
 asteroids a1
 settlersAlive s1

Timer timer
 ticks 0
 delay 5000
 period 500
 steppables null

Sun sun
 timeToSunStorm 3000
 neighboringAsteroids a1

Asteroid a1
 game game
 sun sun
 neighbors
 resource null
 characters s1

Uranium ur2
 life 3

Settler s1
 game game
 timer timer
 place a1
 resources ir2 ir3 ic1 ur2
 gatesCreated tg1 tg2

TeleportingGate tg1
 timer timer
 pair tg2
 settler s1
 asteroid null

TeleportingGate tg2
 timer null
 pair tg1
 settler s1
 asteroid null

- **Kiadott parancs:**
release_gate tg1

- **Elvárt kimenet**

Controller controller

Player p1

Game game

Timer timer

Sun sun

Asteroid a1

Iron ir2

Iron ir3

Ice ic1

Uranium ur2

Settler s1

TeleportingGate tg1 tg2

Controller

game game

allPlayers p1

actPlayer p1

gameRunning true

Player p1

name Bela

settler s1

isAlive true

Game game

timer timer

recipes null

sun sun

asteroids a1

settlersAlive s1

Timer timer

ticks 0

delay 5000

period 500

steppables null

Sun sun

timeToSunStorm 3000

neighboringAsteroids a1

Asteroid a1

game game

sun sun

neighbors

resource null

characters s1

gates tg1

Uranium ur2
life 3

Settler s1
game game
timer timer
place a1
resources ir2 ir3 ic1 ur2
gatesCreated tg2

TeleportingGate tg1
timer timer
pair tg2
settler null
asteroid a1

TeleportingGate tg2
timer null
pair tg1
settler s1
asteroid null

8.2.27 Place second TeleportingGate

- **Leírás**

A telepés lerakja a teleportkapu pár második tagját az aszteroidára, így a teleport használhatóvá válik. Feltételezzük, hogy rendelkezik a szükséges kapuval.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az aszteroida megkapja a kaput, a teleport használható lesz.

Várható hibahelyek:

- A szomszédság nem alakul ki, nem átjárható a kapu
- A második teleportkapu nem kerül letételre

- **Bemenet**

Controller controller

Player p1

Game game

Timer timer

Sun sun

Asteroid a1 a2

Iron ir2

Iron ir3

Ice ic1

Uranium ur2

Settler s1

TeleportingGate tg1 tg2

Controller
game game
allPlayers p1
actPlayer p1
gameRunning true

Player p1
name Bela
settler s1
isAlive true

Game game
timer timer
recipes null
sun sun
asteroids a1 a2
settlersAlive s1

Timer timer
ticks 0
delay 5000
period 500
steppables null

Sun sun
timeToSunStorm 3000
neighboringAsteroids a1

Asteroid a1
game game
sun sun
neighbors a2
resource null
characters null
gates tg1

Asteroid a2
game game
sun sun
neighbors a1
resource null
characters s1
gates null

Uranium ur2
life 3

Settler s1
 game game
 timer timer
 place a2
 resources ir2 ir3 ic1 ur2
 gatesCreated tg2

TeleportingGate tg1
 timer timer
 pair tg2
 settler null
 asteroid a1

TeleportingGate tg2
 timer null
 pair tg1
 settler s1
 asteroid null

- **Kiadott parancs:**
`release_gate tg2`

- **Elvárt kimenet**

Controller controller
 Player p1
 Game game
 Timer timer
 Sun sun
 Asteroid a1 a2
 Iron ir2
 Iron ir3
 Ice ic1
 Uranium ur2
 Settler s1
 TeleportingGate tg1 tg2

Controller
 game game
 allPlayers p1
 actPlayer p1
 gameRunning true

Player p1
 name Bela
 settler s1
 isAlive true

Game game
timer timer
recipes null
sun sun
asteroids a1 a2
settlersAlive s1

Timer timer
ticks 0
delay 5000
period 500
steppables null

Sun sun
timeToSunStorm 3000
neighboringAsteroids a1

Asteroid a1
game game
sun sun
neighbors a2
resource null
characters null
gates tg1

Asteroid a2
game game
sun sun
neighbors a1
resource null
characters s1
gates tg2

Uranium ur2
life 3

Settler s1
game game
timer timer
place a2
resources ir2 ir3 ic1 ur2
gatesCreated null

TeleportingGate tg1
timer timer
pair tg2
settler null
asteroid a1

TeleportingGate tg2
 timer null
 pair tg1
 settler null
 asteroid a2

8.2.28 Call Sunstorm not mined asteroid with AIRobot

- **Leírás**

A napvihar folyamatának a modellezése, feltételezve, hogy az elért aszteroida nincs kibányászva.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az aszteroidán lévő AIRobot szomszédos aszteroidára lép

Várható hibahelyek:

- Az aszteroida köpenye ki van fűrva
- A rajta levő AIRobot elbújt benne

- **Bemenet**

Controller controller

Game game

Timer timer

Sun sun

Asteroid a1 a2

AIRobot air1

Controller

game game

allPlayers null

actPlayer null

gameRunning true

Game game

timer timer

recipes null

sun sun

asteroids a1 a2

settlersAlive null

Timer timer

ticks 0

delay 0

period 1000

steppables sun

Sun sun

timeToSunStorm 1000

neighboringAsteroids a1

Asteroid a1
 rockLayerThickness 1
 game game
 sun sun
 neighbors a2
 resource null
 characters air1
 gates null

Asteroid a2
 rockLayerThickness 1
 game game
 sun sun
 neighbors a1
 resource null
 characters null
 gates null

AIRobot air1
 timer timer
 place a1

- **Elvárt kimenet**

Controller controller
 Game game
 Timer timer
 Sun sun
 Asteroid a1 a2
 AIRobot air1

Controller
 game game
 allPlayers null
 actPlayer null
 gameRunning true

Game game
 timer timer
 recipes null
 sun sun
 asteroids a1 a2
 settlersAlive null

Timer timer
 ticks 0
 delay 0
 period 1000
 steppables sun

Sun sun
 timeToSunStorm 0
 neighboringAsteroids a1

Asteroid a1
 rockLayerThickness 1
 game game
 sun sun
 neighbors a2
 resource null
 characters null
 gates null

Asteroid a2
 rockLayerThickness 1
 game game
 sun sun
 neighbors a1
 resource null
 characters air1
 gates null

AIRobot air1
 timer timer
 place a2

8.2.29 Call Sunstorm not mined asteroid with Settler

- **Leírás**

A napvihar folyamatának a modellezése, feltételezve, hogy az elért aszteroida nincs kibányászva.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az aszteroidán lévő telepes meghal.

Várható hibahelyek:

- Az aszteroida köpenye ki van fúrva
- A rajta levő telepes elbújt benne

- **Bemenet**

Controller controller

Game game

Timer timer

Sun sun

Asteroid a1 a2

Settler s1

Controller

game game

allPlayers p1

actPlayer p1

gameRunning true

Player p1

name Bela
 settler s1
 isAlive true

Game game
 timer timer
 recipes null
 sun sun
 asteroids a1
 settlersAlive s1

Timer timer
 ticks 0
 delay 0
 period 1000
 steppables sun

Sun sun
 timeToSunStorm 1000
 neighboringAsteroids a1

Asteroid a1
 rockLayerThickness 1
 game game
 sun sun
 neighbors null
 resource null
 characters s1
 gates null

Settler s1
 game game
 timer timer
 place a1
 resources null
 gatesCreated null

- **Elvárt kimenet**

Controller controller
 Game game
 Timer timer
 Sun sun
 Asteroid a1 a2
 Settler s1

Controller
 game game
 allPlayers p1
 actPlayer p1
 gameRunning true

Player p1
name Bela
settler null
isAlive false

Game game
timer timer
recipes null
sun sun
asteroids a1
settlersAlive null

Timer timer
ticks 0
delay 0
period 1000
steppables sun

Sun sun
timeToSunStorm 1000
neighboringAsteroids a1

Asteroid a1
rockLayerThickness 1
game game
sun sun
neighbors null
resource null
characters null
gates null

8.2.30 Call Sunstorm mined asteroid

- **Leírás**

A napvihar folyamatának a modellezése, feltételezve, hogy az elért aszteroida ki van bányászva.

- **Ellenőrzött funkcionális, várható hibahelyek**

Az aszteroidán lévő karakterek túlélik a napvihart

Várható hibahelyek:

- Az aszteroida nincs kibányászva.
- A rajta levő karakterek nem tudnak elbújni

- **Bemenet**

Controller controller

Game game

Timer timer

Sun sun

Asteroid a1 a2

Settler s1

Controller

```
game game
allPlayers p1
actPlayer p1
gameRunning true
```

Player p1

```
name Bela
settler s1
isAlive true
```

Game game

```
timer timer
recipes null
sun sun
asteroids a1
settlersAlive s1
```

Timer timer

```
ticks 0
delay 0
period 1000
steppables sun
```

Sun sun

```
timeToSunStorm 1000
neighboringAsteroids a1
```

Asteroid a1

```
rockLayerThickness 1
game game
sun sun
neighbors null
resource null
characters s1
gates null
```

Settler s1

```
game game
timer timer
place a1
resources null
gatesCreated null
```

- **Elvárt kimenet**

Controller controller

Game game

Timer timer

Sun sun

Asteroid a1 a2

Settler s1

Controller

```
game game
allPlayers p1
actPlayer p1
gameRunning true
```

Player p1

```
name Bela
settler s1
isAlive true
```

Game game

```
timer timer
recipes null
sun sun
asteroids a1
settlersAlive null
```

Timer timer

```
ticks 0
delay 0
period 1000
steppables sun
```

Sun sun

timeToSunStorm 1000
neighboringAsteroids a1

Asteroid a1

rockLayerThickness 0
game game
sun sun
neighbors null
resource null
characters null
gates null

Settler s1

game game
timer timer
place a1
resources null
gatesCreated null

8.2.31 UFO moves

- **Leírás**

Az UFO mozgásának tesztelése.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az UFO szomszédos aszteroidára lép.

Várható hibahelyek:

- Az UFO nem tud szomszédra lépni.

- **Bemenet**

Timer timer

Asteroid a1, a2

Ufo ufo1

Timer timer

ticks 0

delay 0

period 1000

steppables ufo1

Asteroid a1

rockLayerThickness 0

neighbors a2

characters ufo1

Asteroid a2

rockLayerThickness 0

neighbors a1

UFO ufo1

 timer timer

 place a1

- **Kiadott parancs:**

 nem kell parancs, a játékban az UFO-k maguktól tevékenykednek.

- **Elvárt kimenet**

Timer timer

Asteroid a1, a2

Ufo ufo1

Timer timer

 ticks 0

 delay 0

 period 1000

 steppables ufo1

Asteroid a1

 rockLayerThickness 0

 neighbors a2

Asteroid a2

 rockLayerThickness 0

 neighbors a1

 characters ufo1

UFO ufo1

 timer timer

 place a2

8.2.32 UFO mines

- **Leírás**

Az UFO bányászatának modellezése. Az UFO a bányászás után eltünteti a nyersanyagot.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az UFO elveszi az aszteroida nyersanyagát.

Várható hibahelyek:

- Az UFO nem tud bányászni.

- **Bemenet**

Timer timer

Asteroid a1

Coal co1

Ufo ufo1

Timer timer

 ticks 0

 delay 0

 period 1000

 steppables ufo1

Asteroid a1

 rockLayerThickness 0

 resource co1

 characters ufo1

UFO ufo1

 timer timer

 place a1

- **Kiadott parancs:**

nem kell parancs, a játékban az UFO-k maguktól tevékenykednek.

- **Elvárt kimenet**

Timer timer

Asteroid a1

Ufo ufo1

Timer timer

 ticks 0

 delay 0

 period 1000

 steppables ufo1

Asteroid a1

 rockLayerThickness 0

 characters ufo1

UFO ufo1

 timer timer

 place a1

8.2.33 UFO dies

- **Leírás**

Az UFO halálának tesztelése, ami a telepes életére veszélyes dolgokkal egyezik meg (napvihar, napszél). Tesztelés egy napvihar hatását mutatja be egy naphoz közeli aszteroidán.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az UFO meghal robbanás vagy napszél esetén.

Várható hibahelyek:

- Az UFO nem hal meg a napszél következtében.

- **Bemenet**

Controller

Timer timer

Sun sun

Asteroid a1

Ufo ufo1

Timer timer

ticks 0

delay 0

period 1000

steppables ufo1

Sun sun

timeToSunStorm 1000

neighboringAsteroids a1

Asteroid a1

rockLayerThickness 2

sun sun

characters ufo1

UFO ufo1

timer timer

place a1

- **Kiadott parancs:**

nem kell parancs, a játékban az UFO-k maguktól tevékenykednek.

- **Elvárt kimenet**

Timer timer

Sun sun

Asteroid a1

Timer timer

 ticks 0

 delay 0

 period 1000

Sun sun

 timeToSunStorm 0

 neighboringAsteroids a1

Asteroid a1

 rockLayerThickness 2

 game null

 sun sun

 neighbors null

 resource null

 characters null

 gates null

8.2.34 AIRobot moves

- **Leírás**

Az AIRobot mozgásának modellezése. Akkor lép, ha nem tud fújni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az AIRobot szomszédos aszteroidára lép

Várható hibahelyek:

- Az aszteroida nem lép.

- **Bemenet**

Timer timer

Asteroid a1, a2

AIRobot air1

Timer timer

 ticks 0

 delay 0

 period 1000

 steppables air1

Asteroid a1

 rockLayerThickness 0

 neighbors a2

 characters air1

Asteroid a2

 rockLayerThickness 3

 neighbors a1

AIRobot air1
 timer timer
 place a1

- **Kiadott parancs:**

nem kell parancs, a játékban az AIRobot-ok maguktól tevékenykednek.

- **Elvárt kimenet**

Timer timer
 Asteroid a1, a2
 AIRobot air1

Timer timer
 ticks 0
 delay 0
 period 1000
 steppables air1

Asteroid a1
 rockLayerThickness 0
 neighbors a2

Asteroid a2
 rockLayerThickness 3
 neighbors a1
 characters air1

AIRobot air1
 timer timer
 place a2

8.2.35 AIRobot drills

- **Leírás**

Az AIRobot furásának modellezése. A robot fúrni fog, ha egy olyan aszteroidán tartózkodik, ami nincs kifúrva.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az AIRobot egy egységet lefür az aszteroida rétegében.

Várható hibahelyek:

- Az AIRobot nem fúr.

- **Bemenet**

Timer timer

Asteroid a1

AIRobot air1

Timer timer

ticks 0

delay 0

period 1000

steppables air1

Asteroid a1

rockLayerThickness 3

characters air1

AIRobot air1

timer timer

place a1

- **Kiadott parancs:**

nem kell parancs, a játékban az AIRobot-ok maguktól tevékenykednek.

- **Elvárt kimenet**

Timer timer

Asteroid a1

AIRobot air1

Timer timer

ticks 0

delay 0

period 1000

steppables air1

Asteroid a1

rockLayerThickness 2

characters air1

AIRobot air1

timer timer

place a1

8.2.36 AIRobot got pushed

- **Leírás**

Az AIRobot ellköésének modellezése. A robot a szomszédos aszteroidára repül egy radioaktív robbanás esetén.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az AIRobot a robbanás után szomszédos aszteroidára repül.

Várható hibahelyek:

- Az AIRobot nem lökődik el.

- **Bemenet**

Controller controller

Player p1

Timer timer

Asteroid a1, a2

Uranium ur1

Settler s1

AIRobot air1

Controller controller

allPlayers p1

actPlayer p1

gameRunning true

Player p1

 name Marci

 settler s1

 isAlive true

Timer timer

 ticks 0

 delay 0

 period 1000

 steppables air1

Asteroid a1

 rockLayerThickness 0

 neighbors a2

 resource ur1

 characters air1

Asteroid a2

 rockLayerThickness 3

 neighbors a1

Uranium ur1

 life 1

Settler s1

 timer timer

 place a1

AIRobot air1
 timer timer
 place a1

- **Kiadott parancs:**
 mine

- **Elvárt kimenet**

Player p1
Timer timer
Asteroid a2
AIRobot air1

Controller controller
gameRunning false

Player p1
 name Marci
 settler s1
 isAlive false

Timer timer
 ticks 0
 delay 0
 period 1000
 steppables air1

Asteroid a2
 rockLayerThickness 3
 characters air1

AIRobot air1
 timer timer
 place a2

8.2.37 AIRobot dies

- **Leírás**

Az AIRobot halálának tesztelése. A robot meghal napvihar esetén, ha nem tud elrejtőzni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az AIRobot meghal egy napvihar következtében.

Várható hibahelyek:

- A robot nem hal meg a napvihar következtében.

- **Bemenet**

Timer timer

Sun sun

Asteroid a1

AIRobot air1

Timer timer

ticks 0

delay 0

period 1000

steppables air1

Sun sun

timeToSunStorm 1000

neighboringAsteroids a1

Asteroid a1

rockLayerThickness 2

sun sun

characters air1

AIRobot air1

timer timer

place a1

- **Kiadott parancs:**

nem kell parancs, a játékban az AIRobot és a Sun maguktól tevékenykednek.

- **Elvárt kimenet**

Timer timer

Sun sun

Asteroid a1

Timer timer

ticks 0

delay 0

period 1000

Sun sun

timeToSunStorm 0

neighboringAsteroids a1

Asteroid a1
 rockLayerThickness 2

8.2.38 TeleportingGate goes mad

- **Leírás**

A teleportkapu megőrülésének modellezése. A kapu mozogni kezd aszteroidák között, ha napszél éri.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teleportkapu napvihar hatására mozogni kezd.

Várható hibahelyek:

- A teleportkapu nem mozog.

- **Bemenet**

Timer timer

Sun sun

Asteroid a1, a2, a3

TeleportingGate tg1, tg2

Timer timer

ticks 0

delay 0

period 1000

Sun sun

timeToSunStorm 1000

neighboringAsteroids a1

Asteroid a1

rockLayerThickness 0

neighbors a2, a3

gates tg1

Asteroid a2

rockLayerThickness 3

sun sun

neighbors a1, a3

Asteroid a3

rockLayerThickness 3

sun sun

neighbors a1, a2

gates tg2

TeleportingGate tg1

pair tg2

asteroid a1

TeleportingGate tg2

pair tg1

asteroid a3

- **Kiadott parancs:**

nem kell parancs, a játékban az TeleportingGate és a Sun maguktól tevékenykednek.

- **Elvárt kimenet**

Timer timer

Sun sun

Asteroid a1, a2, a3

TeleportingGate tg1, tg2

Timer timer

ticks 0

delay 0

period 1000

steppables tg1

Sun sun

timeToSunStorm 1000

neighboringAsteroids a1

Asteroid a1

rockLayerThickness 0

sun sun

neighbors a2, a3

Asteroid a2

rockLayerThickness 3

sun sun

neighbors a1, a3

gates tg1

Asteroid a3

rockLayerThickness 3

neighbors a1, a2

gates tg2

TeleportingGate tg1

timer timer

pair tg2

asteroid a2

TeleportingGate tg2

pair tg1

asteroid a3

8.3 A tesztelést támogató programok tervez

A tesztelés során a projekten belül egy másik (**berenyi_kft.test**) névterben létrehozunk egy **Tester** osztályt, amelynek a feladata a tesztesetek futtatása a tesztkörnyezetben. A **test <test_num>** parancs a **test_num** sorszámú tesztesetet futtja, a **test** parancs (paraméter nélkül) pedig az összes tesztesetet az 1. sorszámuktól az utolsóig. A prototípus program **exit** parancsával kiléphetünk az alkalmazásból.

A **<test_num>** sorszámú teszteset a **test_<test_num>.in** bemeneti fájlból olvassa be a kiindulási játékállapotot, majd a **test_<test_num>.result** nevű fájlba írja a teljes aktuális állapotot a teszt lefuttatása után. Ezt a **test_<test_num>.out** elvárt kimeneti fájl tartalmával hasonlítja össze, és az eltéréseket a képernyön megjeleníti, az eltérő sorok kiírásával. Ha valamelyik fájl nem található meg a futtatott prototípus programmal egy mappában, akkor a hiba szintén a képernyön megjelenik, a következőhöz hasonló formában: „File <filename> cannot be found.”.

Egy-egy teszt futtatása több elemi lépést tartalmaz, de ezek váza közös az összes tesztesetre. minden teszteset egy-egy a tesztre jellemző tesztspecifikus parancsot tartalmaz, ami a konkrét tesztelt működést elindítja. Ezeket a tesztesetek tervezénél a *Kiadott parancs* menüpontokban adtuk meg.

A következő parancsokkal ekvivalens metódushívások történnek egy-egy teszteset során:

```
random false
log true
load test_<test_num>.in
start
[<tesztspecifikus parancs paramétereikkel>]
stop
save test <test num>.result
compare test <test num>.result test <test num>.out
```

A működést a tesztesetekben egységesen determinisztikussá tesszük, és engedélyezzük a meghívott metódusok logolását a konzolon. Betöljük a teszt sorszámának megfelelő játékkonfigurációs fájlból a kiindulási állapotot, majd elindítjuk a modell futtatását.

Ezután kiadjuk a tesztesetre jellemző tesztspecifikus parancsot (ha szükséges ilyen parancs). Amikor egy játékos lép a telepesével, akkor ez a specifikus parancs lehet egy **pass**, **move**, ..., **release_gate** parancsnak megfelelő metódus. Ha léptethető objektum **step()** metódusát szeretnénk előidézni, akkor nincs szükség ilyen extra parancsra; a **start** utasítás elindítja az időzítőt, és ő lépteti majd a tesztelendő léptethető objektumot.

Ha a tesztre jellemző működésnek vége, leállítjuk a teszt futását, és ideiglenes fájlba mentjük a teszt utáni játékállapotot. Végül a kimenetet összehasonlítjuk az elvárt kimeneti konfigurációval.

A sok megszűnő objektumot nem soroljuk fel újra a kimeneti fájlokban, emiatt az elvárt kimeneti fájl sokkal rövidebb is lehet a bemenetnél. Az egyes objektumoknak azonban itt megjelenítjük a nem használt attribútumait is (csak a dokumentációban nem soroltuk fel a nem használt tagokat), hogy a bemenet és a kimenet formátuma azonos lehessen, és egyszerűbb legyen az összehasonlítás. Az időzítés jellegű adattagoknál (*például timeToSunStorm*) nem várunk el pontos egyezést, hiszen nem tudjuk garantálni, hogy a programot különböző gépeken, különböző időpontokban futtatva az időzítések és a futási idők megegyeznek.

8.3.1 Tester

- **Felelősség**

A tesztesetek futtatását végzi, és összehasonlítja a futási eredményeket az elvárt kimenettel.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **private static int num_tests:** a tesztesetek száma, alapértelmezett értéke 38

- **Metódusok**

- **public void testOne(int test_num):** Futtatja a kiválasztott sorszámú tesztet a fenti 8.3 pontban leírt parancsok mintájára. A `test_<test_num>.in` bemeneti fájlból beolvassa a kiindulási játékállapotot, kiadja a tesztspesifikus parancsot, majd a `test_<test_num>.result` nevű fájlba írja a teljes aktuális állapotot. Ezt a `test_<test_num>.out` elvárt kimeneti fájl tartalmával hasonlítja össze (`compare(test_<test_num>.result, test_<test_num>.out)`), és az eltéréseket a képernyön megjeleníti.
- **public void testAll():** Az összes teszt futtatását sorban elvégzi az 1. sorszámúktól a `num_test` sorszámúig a `testOne(int test_num)` függvény segítségével.
- **public void compare(File f1, File f2):** Összehasonlítja az f1 és f2 konfigurációs fájlok tartalmát. Ahol eltérést talál, az eltérő sorokat egymás után megjeleníti a képernyőn. Ha valamelyik fájl nem található, a hibát jelzi a kimeneten.

8.4 Napló

Kezdet	Időtartam	Résznevők	Leírás
2021.04.02. 17:00	4 óra	Kreták Szögi	Tevékenység: Részletes tervezés megírása, a kód, a tervezés és a szekvenciadiagramok összehangolása I.
2021.04.03.18.00	4 óra	Kreták Szögi	Tevékenység: Részletes tervezés megírása, a kód, a tervezés és a szekvenciadiagramok összehangolása II.
2021.04.05.14:00	4 óra	Radnai Mesterházi	Tevékenység: Tesztek nyelvének, átbeszélése, tesztelő program megtervezése, beadandó feladat megoldása.
2021.04.07.12:00	1 óra	Radnai	Tevékenység: Tesztek elkészítése 1-10-ig
2021.04.07.20:00	1 óra	Mesterházi	Tevékenység: Tesztek elkészítése 11-20-ig
2021.04.08.16:00	1 óra	Havasi	Tevékenység: Tesztek elkészítése 21-30-ig
2021.04.08.16:00	3 óra	Kreták Szögi	Tevékenység: Pszeudokódok megírása a fontosabb metódusokhoz a részletes tervezében
2021.04.09.00:00	1 óra	Mesterházi	Tevékenység: Tesztek elkészítése 30-38-ig
2021.04.09.14:00	3 óra	Kreták Szögi Havasi Radnai Mesterházi	Értekezlet: Tesztek, feladatok, Controller, Player átbeszélése, javítások, elmélkedés az implementáláson.

Az adott heti munka csapattagokonkénti aránya:

Csapattag	Munka aránya százalékban
Havasi	20
Kreták	20
Mesterházi	20
Radnai	20
Szögi	20

10. Szkeleton beadás

7 – berenyi_kft

Konzulens:

Csapattagok
Kreták Barnabás András
Mesterházi Marcell
Szögi Richárd
Havasi Balázs
Radnai Bálint

2021.04.19

10. Prototípus beadása

10.1 Fordítási és futtatási útmutató

10.1.1 Fájllista

Fájl neve	Méret (bájt)	Keletkezés ideje	Tartalom
AIRobot.java	3030	2021.03.16	AIRobot osztály megvalósítása
Asteroid.java	16968	2021.03.16	Asteroid osztály megvalósítása
Character.java	2908	2021.03.16	Character osztály megvalósítása
Coal.java	1401	2021.03.16	Coal osztály megvalósítása
Game.java	12718	2021.03.16	Game osztály megvalósítása
ISteppable.java	305	2021.03.16	ISteppable osztály megvalósítása
Ice.java	1804	2021.03.16	Ice osztály megvalósítása
Iron.java	1397	2021.03.16	Iron osztály megvalósítása
RadioactiveResource.java	543	2021.03.16	RadioactiveResource osztály megvalósítása
Recipe.java	4365	2021.03.16	Recipe osztály megvalósítása
Resource.java	1612	2021.03.16	Resource osztály megvalósítása
Settler.java	11252	2021.03.16	Settler osztály megvalósítása
Skeleton.java	34612	2021.03.17	Tesztetek megvalósítása, hívása
Sun.java	4394	2021.03.16	Sun osztály megvalósítása
TeleportingGate.java	5960	2021.03.16	TeleportingGate osztály megvalósítása
Timer.java	5270	2021.03.16	Timer osztály megvalósítása
Uranium.java	3174	2021.03.16	Uranium osztály megvalósítása
Proto.java	25881	2021.04.07	Proto osztály megvalósítása
Tester.java	9126	2021.04.07	Tester osztály megvalósítása
Ufo.java	2456	2021.04.07	Ufo osztály megvalósítása
State.java	1100	2021.04.07	State osztály megvalósítása
PlayerCommand.java	856	2021.04.07	PlayerCommand osztály megvalósítása
Controller.java	6203	2021.04.07	Controller osztály megvalósítása
Player.java	3029	2021.04.07	Player osztály megvalósítása

Fájl neve	Méret (bájt)	Keletkezés ideje	Tartalom
test_1.in	169	2021.04.16	Teszt bemenet
test_2.in	1164	2021.04.16	Teszt bemenet
test_3.in	1078	2021.04.16	Teszt bemenet
test_4.in	592	2021.04.16	Teszt bemenet
test_5.in	251	2021.04.16	Teszt bemenet
test_6.in	690	2021.04.16	Teszt bemenet
test_7.in	473	2021.04.16	Teszt bemenet
test_8.in	253	2021.04.16	Teszt bemenet
test_9.in	253	2021.04.16	Teszt bemenet
test_10.in	1221	2021.04.16	Teszt bemenet
test_11.in	342	2021.04.16	Teszt bemenet
test_12.in	254	2021.04.16	Teszt bemenet
test_13.in	229	2021.04.16	Teszt bemenet
test_14.in	214	2021.04.16	Teszt bemenet
test_15.in	486	2021.04.16	Teszt bemenet
test_16.in	274	2021.04.16	Teszt bemenet
test_17.in	514	2021.04.16	Teszt bemenet
test_18.in	169	2021.04.16	Teszt bemenet
test_19.in	294	2021.04.16	Teszt bemenet
test_20.in	194	2021.04.16	Teszt bemenet
test_21.in	682	2021.04.16	Teszt bemenet
test_22.in	746	2021.04.16	Teszt bemenet
test_23.in	895	2021.04.16	Teszt bemenet
test_24.in	897	2021.04.16	Teszt bemenet
test_25.in	1041	2021.04.16	Teszt bemenet
test_26.in	796	2021.04.16	Teszt bemenet
test_27.in	956	2021.04.16	Teszt bemenet
test_28.in	675	2021.04.16	Teszt bemenet
test_29.in	646	2021.04.16	Teszt bemenet
test_30.in	651	2021.04.16	Teszt bemenet
test_31.in	366	2021.04.16	Teszt bemenet
test_32.in	321	2021.04.16	Teszt bemenet
test_33.in	374	2021.04.16	Teszt bemenet
test_34.in	375	2021.04.16	Teszt bemenet
test_35.in	304	2021.04.16	Teszt bemenet
test_36.in	1124	2021.04.16	Teszt bemenet
test_37.in	382	2021.04.16	Teszt bemenet
test_38.in	814	2021.04.16	Teszt bemenet

Fájl neve	Méret (bájt)	Keletkezés ideje	Tartalom
test_1.out	300	2021.04.16	Teszt elvárt kimenete
test_2.out	1335	2021.04.16	Teszt elvárt kimenete
test_3.out	1209	2021.04.16	Teszt elvárt kimenete
test_4.out	745	2021.04.16	Teszt elvárt kimenete
test_5.out	397	2021.04.16	Teszt elvárt kimenete
test_6.out	925	2021.04.16	Teszt elvárt kimenete
test_7.out	704	2021.04.16	Teszt elvárt kimenete
test_8.out	398	2021.04.16	Teszt elvárt kimenete
test_9.out	398	2021.04.16	Teszt elvárt kimenete
test_10.out	774	2021.04.16	Teszt elvárt kimenete
test_11.out	460	2021.04.16	Teszt elvárt kimenete
test_12.out	375	2021.04.16	Teszt elvárt kimenete
test_13.out	366	2021.04.16	Teszt elvárt kimenete
test_14.out	314	2021.04.16	Teszt elvárt kimenete
test_15.out	280	2021.04.16	Teszt elvárt kimenete
test_16.out	369	2021.04.16	Teszt elvárt kimenete
test_17.out	627	2021.04.16	Teszt elvárt kimenete
test_18.out	300	2021.04.16	Teszt elvárt kimenete
test_19.out	392	2021.04.16	Teszt elvárt kimenete
test_20.out	309	2021.04.16	Teszt elvárt kimenete
test_21.out	844	2021.04.16	Teszt elvárt kimenete
test_22.out	793	2021.04.16	Teszt elvárt kimenete
test_23.out	1043	2021.04.16	Teszt elvárt kimenete
test_24.out	910	2021.04.16	Teszt elvárt kimenete
test_25.out	1082	2021.04.16	Teszt elvárt kimenete
test_26.out	829	2021.04.16	Teszt elvárt kimenete
test_27.out	956	2021.04.16	Teszt elvárt kimenete
test_28.out	617	2021.04.16	Teszt elvárt kimenete
test_29.out	489	2021.04.16	Teszt elvárt kimenete
test_30.out	647	2021.04.16	Teszt elvárt kimenete
test_31.out	489	2021.04.16	Teszt elvárt kimenete
test_32.out	365	2021.04.16	Teszt elvárt kimenete
test_33.out	381	2021.04.16	Teszt elvárt kimenete
test_34.out	497	2021.04.16	Teszt elvárt kimenete
test_35.out	373	2021.04.16	Teszt elvárt kimenete
test_36.out	928	2021.04.16	Teszt elvárt kimenete
test_37.out	381	2021.04.16	Teszt elvárt kimenete
test_38.out	819	2021.04.16	Teszt elvárt kimenete

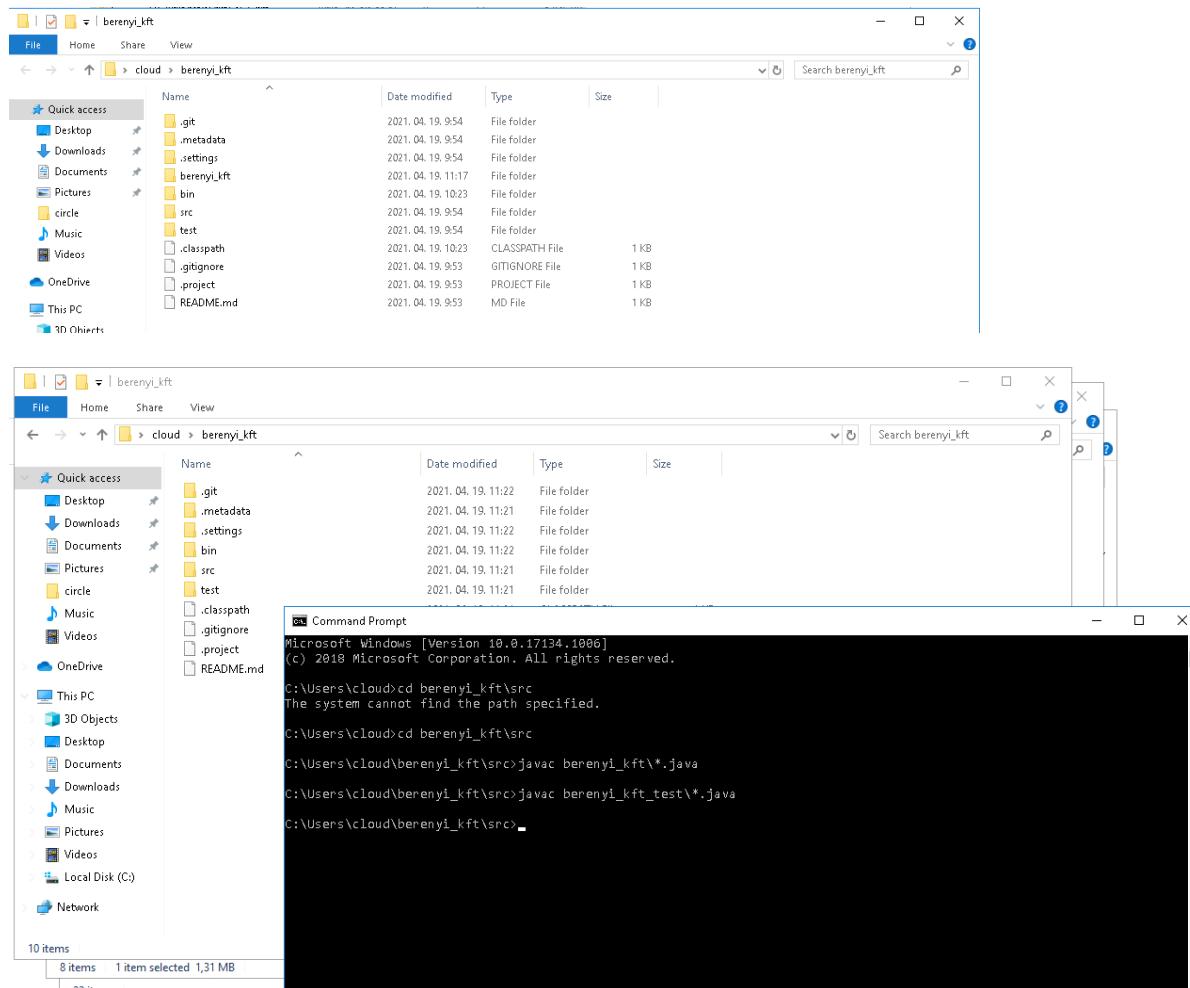
10.1.2 Fordítás

A program természetesen fordítható Eclipse-en keresztül.

Fontos, hogy ez a program virtuális gépen futó változata. A cmd fájlelérése más, mint az Eclipse-nek, ezért virtuális gépen cmd-n csak abszolút útvonallal fut biztosan.

Ezért helyezzük a projekt gyökérmapját a C:\Users\cloud könyvtárba, majd adjuk ki az src mappában a következő két parancsot:

```
javac berenyi_kft\*.java
javac berenyi_kft_test\*.java
```



10.1.3 Futtatás

Futtatás pedig a **java berenyi_kft**.Proto paracsal hajható végre, amely hatására programunk el is indul.

```
C:\Users\cloud\berenyi_kft\src>java berenyi_kft.Proto
Welcome in berenyi_kft's Proto program!
Please select whether you wish to test (Y) or play (n) the prototype game. [Y/n]: -
```

A tesztelések futtatásához a menüben az y parancsot kell megadni, ezután választhatunk:

- Az összes tesztet szeretnénk futtatni: **test**

```
C:\Users\cloud\berenyi_kft\src>java berenyi_kft.Proto
Welcome in berenyi_kft's Proto program!
Please select whether you wish to test (Y) or play (n) the prototype game. [Y/n]: y
Here you can test our prototype program by predefined test cases.

List of the applicable test control commands:
  test <test_num>      Executes test number <test_num> and prints the result.
  test                  Runs all the test from number 1 to 38.
  exit                 Exits from this test program.

test
```

melynek végén ez fogad:

```
Configuration saved to C:\Users\cloud\berenyi_kft\src\test_data\test_results\test_38.result.

Comparing result (C:\Users\cloud\berenyi_kft\src\test_data\test_results\test_38.result)
to expected output (C:\Users\cloud\berenyi_kft\src\test_data\test_outputs\test_38.out):

The compared files' contents are the same.

***The test is successful.***
// ----- berenyi_kft's test 38 ended. ----- //
```

Number of successful tests: 38
Number of failed tests: 0

- Egy tetszőlegesen kiválasztottat: **test <sorszam>**

```
test 10
      // ----- berenyi_kft's test 10 started. ----- //
random disabled
logging enabled
-> AIRobot()
-> UFO()
```

melynek végén ez fogad:

```
Configuration saved to C:\Users\cloud\berenyi_kft\src\test_data\test_results\test_10.result.

Comparing result (C:\Users\cloud\berenyi_kft\src\test_data\test_results\test_10.result)
to expected output (C:\Users\cloud\berenyi_kft\src\test_data\test_outputs\test_10.out):

The compared files' contents are the same.

***The test is successful.***
// ----- berenyi_kft's test 10 ended. ----- //
```

A teszt futtatása után további parancsokat adhatunk meg(**test/test <sorszam>**) vagy ki is léphetünk(**exit**).

Amennyiben Eclipse-ben nyitják meg a programot, a projekt a **JDK 11**-et használja, azért, hogy a virtuális gépen is futtathatóvá váljon.

A **berenyi_kft/src/test_data** mappában találhatók a tesztekhez tartozó fájlok:
A bemeneti (.in) fájlokat a **test_inputs** mappa tartalmazza, az elvárt kimeneti (.out) fájlokat a **test_outputs** mappa, a tényleges kimeneti (**.result**) fájlokat a **test_results** mappa.
Ezek a fájlok szövegfájlként megnyithatók és szemrevételezhetők, de a tesztprogram önállóan elvégzi a .result és .out fájlok tartalmi összehasonlítását a tesztelő felhasználó kényelme érdekében.

10.2 Tesztek jegyzőkönyvei

Az előző beadáshoz képest általános változtatás, hogy bizonyos tesztek lefutásához a szükséges volt a bemeneti fájlban Game, Recipe, illetve Timer objektumok hozzáadására is, ezt külön a teszeseteknél nem részletezzük.

10.2.1 Teszteset1: Pass

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:30

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:30
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.2 Teszteset2: Win Game with moving

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:31

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:31
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.3 Teszteset3: Win Game with mining

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:32

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:32
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.4 Teszteset4: Move Spaceship neighboring Asteroid

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:33

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:33
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.5 Teszteset5: Move Spaceship not neighboring Asteroid

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:34

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:34
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.6 Teszteset6: Move Spaceship through TeleportingGate

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:00

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:00
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.7 Teszteset7: Move Spaceship through a single TeleportingGate

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:01

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:01
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.8 Teszteset8: Drill Asteroid one layer

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:02

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:02
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.9 Teszteset9: Drill Asteroid fail

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:03

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:03
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.10 Teszteset10: Drill radioactive Asteroid's last layer near Sun

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:04

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:04
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.11 Teszteset11: Drill icy Asteroid's last layer near Sun

Tesztelő neve	Mesterházi Marcell
Teszt időpontja	2021.04.18. 19:00

Tesztelő neve	Mesterházi Marcell
Teszt időpontja	2021.04.18. 19:00
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.12 Teszteset12: Restore Resource

Tesztelő neve	Mesterházi Marcell
Teszt időpontja	2021.04.18. 19:01

Tesztelő neve	Mesterházi Marcell
Teszt időpontja	2021.04.18. 19:01
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.13 Teszteset13: Restore Resource with empty container

Tesztelő neve	Mesterházi Marcell
Teszt időpontja	2021.04.18. 19:02

Tesztelő neve	Mesterházi Marcell
Teszt időpontja	2021.04.18. 19:02
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.14 Teszteset14: Restore Resource Fail not empty

Tesztelő neve	Mesterházi Marcell
Teszt időpontja	2021.04.18. 19:03

Tesztelő neve	Mesterházi Marcell
Teszt időpontja	2021.04.18. 19:03
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.15 Teszteset15: Restore RadiactiveResource near Sun

Tesztelő neve	Mesterházi Marcell
Teszt időpontja	2021.04.18. 19:04

Tesztelő neve	Mesterházi Marcell
Teszt időpontja	2021.04.18. 19:04
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.16 Teszteset16: Restore IceResource near Sun

Tesztelő neve	Radnai Bálint
Teszt időpontja	2021.04.18. 20:00

Tesztelő neve	Radnai Bálint
Teszt időpontja	2021.04.18. 20:00
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.17 Teszteset17: Mine Resource

Tesztelő neve	Radnai Bálint
Teszt időpontja	2021.04.18. 20:01

Tesztelő neve	Radnai Bálint
Teszt időpontja	2021.04.18. 20:01
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.18 Teszteset18: Mine Resource fail when Asteroid is empty

Tesztelő neve	Radnai Bálint
Teszt időpontja	2021.04.18. 20:02

Tesztelő neve	Radnai Bálint
Teszt időpontja	2021.04.18. 20:02
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.19 Teszteset19: Mine Resource fail when Spaceship is full

Tesztelő neve	Radnai Bálint
Teszt időpontja	2021.04.18. 20:03

Tesztelő neve	Radnai Bálint
Teszt időpontja	2021.04.18. 20:03
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.20 Teszteset20: Mine Resource fail when Asteroid is not drilled:

Tesztelő neve	Radnai Bálint
Teszt időpontja	2021.04.18. 20:04

Tesztelő neve	Radnai Bálint
Teszt időpontja	2021.04.18. 20:04
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.21 Teszteset21: Build AIRobot

Tesztelő neve	Havasi Balázs
Teszt időpontja	2021.04.18. 17:30

Tesztelő neve	Havasi Balázs
Teszt időpontja	2021.04.18. 17:30
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.22 Teszteset22: Build AIRobot fail

Tesztelő neve	Havasi Balázs
Teszt időpontja	2021.04.18. 17:31

Tesztelő neve	Havasi Balázs
Teszt időpontja	2021.04.18. 17:31
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.23 Teszteset23: Build TeleportingGate pair

Tesztelő neve	Havasi Balázs
Teszt időpontja	2021.04.18. 17:32

Tesztelő neve	Havasi Balázs
Teszt időpontja	2021.04.18. 17:32
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.24 Teszteset24: Build TeleportingGate pair fails when Player has not enough Resources

Tesztelő neve	Havasi Balázs
Teszt időpontja	2021.04.18. 17:33

Tesztelő neve	Havasi Balázs
Teszt időpontja	2021.04.18. 17:33
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.25 Teszteset25: Build TeleportingGate pair fails when Player has not enough space

Tesztelő neve	Havasi Balázs
Teszt időpontja	2021.04.18. 17:34

Tesztelő neve	Havasi Balázs
Teszt időpontja	2021.04.18. 17:36
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	3 pár helyett 1 pár teleportkapuval indul a telepes, mivel ez már elég ahhoz, hogy ne legyen hely új kapupárnak.

10.2.26 Teszteset26: Place TeleportingGate

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:35

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:35
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.27 Teszteset27: Place second TeleportingGate

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:36

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:36
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.28 Teszteset28: Call Sunstorm not mined asteroid with AIRobot

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:37

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:37
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.29 Teszteset29: Call Sunstorm not mined asteroid with Settler

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:38

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:38
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.30 Teszteset30: Call Sunstorm mined asteroid

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:39

Tesztelő neve	Kreták Barnabás András
Teszt időpontja	2021.04.18. 18:39
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.31 Teszteset31: UFO moves

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:05

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:05
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.32 Teszteset32: UFO mines

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:06

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:06
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.33 Teszteset33: UFO dies

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:06

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:06
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.34 Teszteset34: AIRobot moves

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:07

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:07
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.35 Teszteset35: AIRobot drills

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:08

Tesztelő neve	Szögi Richárd
Teszt időpontja	2021.04.19. 01:08
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.36 Teszteset36: AIRobot got pushed

Tesztelő neve	Mesterházi Marcell
Teszt időpontja	2021.04.18. 19:05

Tesztelő neve	Mesterházi Marcell
Teszt időpontja	2021.04.18. 19:05
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.37 Teszteset37: AIRobot dies

Tesztelő neve	Radnai Bálint
Teszt időpontja	2021.04.18. 20:05

Tesztelő neve	Radnai Bálint
Teszt időpontja	2021.04.18. 20:05
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.2.38 Teszteset38: TeleportingGate goes mad

Tesztelő neve	Radnai Bálint
Teszt időpontja	2021.04.18. 20:06

Tesztelő neve	Radnai Bálint
Teszt időpontja	2021.04.18. 20:06
Teszt eredménye	Sikeres
Lehetséges hibaok	
Változtatások	

10.3 Értékelés

Tag neve	Tag neptun	Munka százalékban
Havasi		20
Kreták		20
Mesterházi		20
Radnai		20
Szögi		20

10.4 Napló

Kezdet	Időtartam	Részttvevők	Leírás
2021.04.15. 18:00	2,5 óra	Havasi Mesterházi Kreták Szögi Radnai	Értekezlet. Program terveinek megbeszélése, feladatok kiosztása, kód javítás.
2021.04.16. 14:00	5 óra	Havasi Mesterházi Kreták Szögi Radnai	Tevékenység: beolvasás, menü, játékmód implementálása
2021.04.16. 19:00	5 óra	Radnai	Tevékenység: játékmód implementálása
2021.04.16. 14:00	2 óra	Mesterházi	Tevékenység: tesztek in/out fájljainak megvalósítása
2021.04.17. 14:00	6 óra	Havasi Mesterházi Kreták Szögi Radnai	Tevékenység: tesztek implementálása
2021.04.18. 10:00	10 óra	Mesterházi Kreták Szögi Radnai	Tevékenység: 38 teszt sikeres implementálása a programban
2021.04.18. 22:00	5 óra	Radnai	Tevékenység: játékmód, logolás, menü elkészítése
2021.04.19. 9:00	2 óra	Mesterházi Kreták Szögi Radnai	Tevékenység: dokumentum összeszerkesztése, program kipróbálása a virtuális gépen.

11.

Grafikus felület specifikációja

7 – berenyi_kft

Konzulens:

Csapattagok

Kreták Barnabás András

Mesterházi Marcell

Szögi Richárd

Havasi Balázs

Radnai Bálint

2021.04.25

11. Grafikus felület specifikációja

11.1 A grafikus interfész

11.1.1 A játékban szereplő objektumok

Settler



Asteroid



TeleportingGate



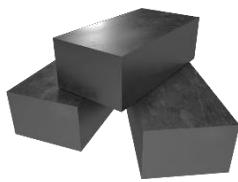
AIRobot



UFO



Iron



Coal



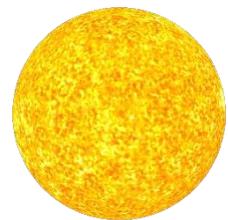
Ice



Uranium

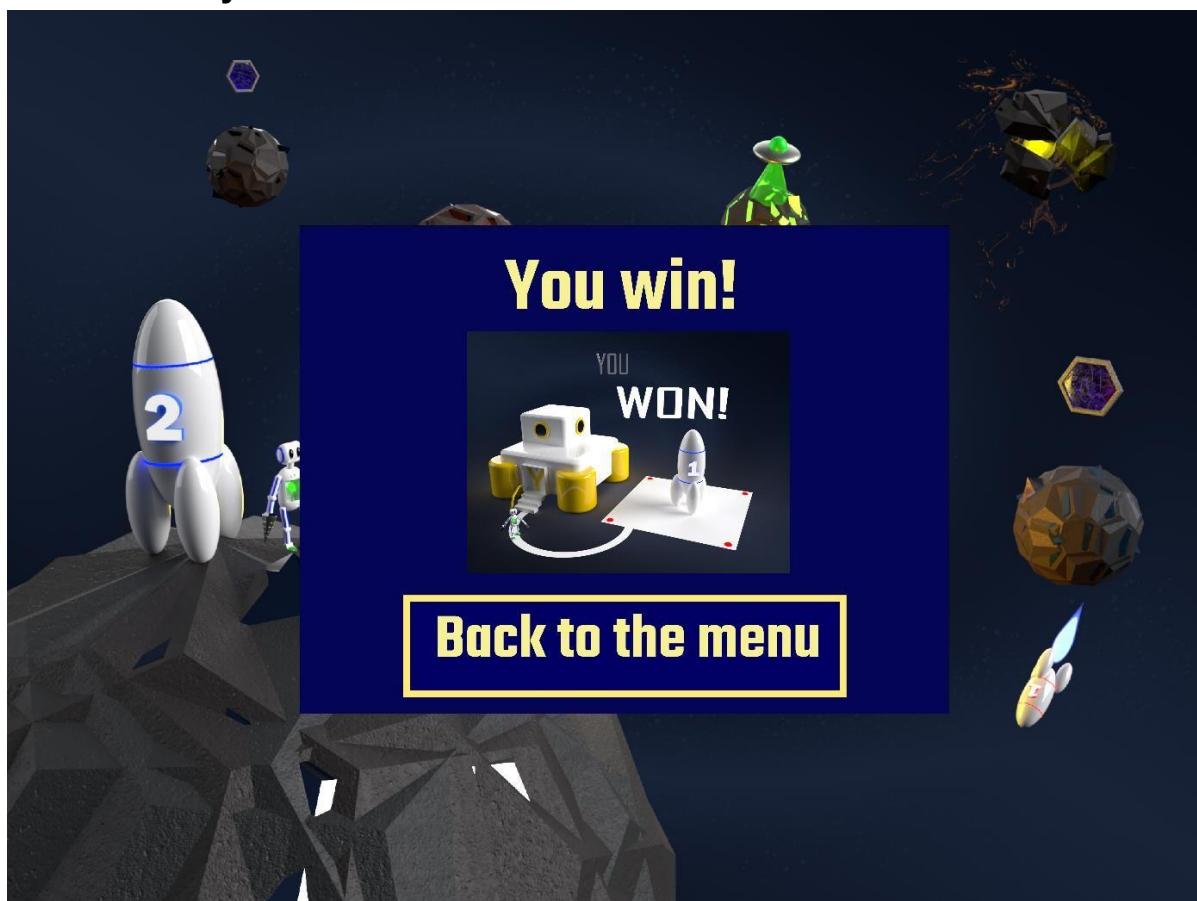


Sun



11.1.2 Főmenü



11.1.3 Új játék ablak**11.1.4 Nyertél! ablak**

11.1.5 Vesztettél! ablak**11.1.6 Aktív játék ablaka**

11.2 A grafikus rendszer architektúrája

11.2.1 A felület működési elve

Az alkalmazásablak egy MainFrame példány, amit a JFrame Swing-osztályból származtatunk. Az ablakban különféle panelek jelennek meg a játék állapotától függően, ezek JPanel-ek: MenuPanel, AddPlayersPanel, GamePanel, EndGamePanel. A játékbablak egy Cards típusú cards példányt tartalmaz, amely vezérli, hogy az ablakban melyik panel jelenjen meg éppen. A Cards osztály JPanelek megjelenítését menedzseli CardLayout segítségével.

Az egyes paneltípusok szerepe röviden:

- A MenuPanel típusú panel a főmenüt tartalmazza, innen indíthatunk új játéket, betölthetjük a legutóbbi elmentett játéket, vagy kiléphetünk az alkalmazásból.
- Az AddPlayerPanel új játék esetén a játékosok neveinek hozzáadását, majd a játék elindítását teszi lehetővé.
- A GamePanel a játéktér (aszteroidamező és az összes Game-beli objektum) megjelenítéséért felelő panel osztály.
- Az EndGamePanel osztály a játék végi eredmény szöveges- és képes megjelenítését végzi.

A játékunk főablaka tehát a cards panelmegjelenítő menedzseren keresztül egy-egy példányt tartalmaz a fenti négyféle panelból.

A Controller osztály a játékosok kattintásaira figyel a GamePanel-en. Ha egy érvényes utasítást adott meg a lépő játékos, végrehajtja a lépést a modellben. A megoldásunk során a push modellt alkalmazzuk. A Controller minden látható cselekvés után a szól a GamePanel-nek, hogy rajzolja újra a pályát a drawAll() függvénye segítségével.

A játékalkalmazás teljes folyamata a következő. Az alkalmazás indításakor a főmenü jelenik meg. A „New Game” feliratú gombra kattintáskor a cards objektum show() metódusa segítségével az addPlayersPanel jelenik meg, és a felhasználói események ide érkeznek. Az Add gombbal a játékosok listájához adjuk a nameField szövegmezőbe írt nevet. A név csak alfanumerikus karaktereket tartalmazhat. A „Start Game!” gombbal ezután elindul a játék, és a gamePanel-en figyelhetjük az elindult játék futását.

Ha a játék véget ér a telepesek győzelmével vagy vereségével, akkor az időzítés leáll, a játékállapotot fájlba mentjük, majd megjelenik az endGamePanel a megfelelő tartalommal. Az endGamePanel „Back to Menu” gombjára kattintva visszatérünk a főmenübe.

A játékállapot akkor is elmentődik, ha az ablakot bezárjuk az „X” gombbal. A főmenü „Load Game” gombjával betöltődik a legutóbbi mentett játékállapot. Attól függően, hogy a játék véget ért-e, a gamePanel, vagy az endGamePanel jelenik meg.

A program kijelzőjén látható változások a bányászat, fűrás, visszatöltés, mozgás, robot/teleportkapu-pár építés, teleportkapu lerakása, valamint a robbanások.

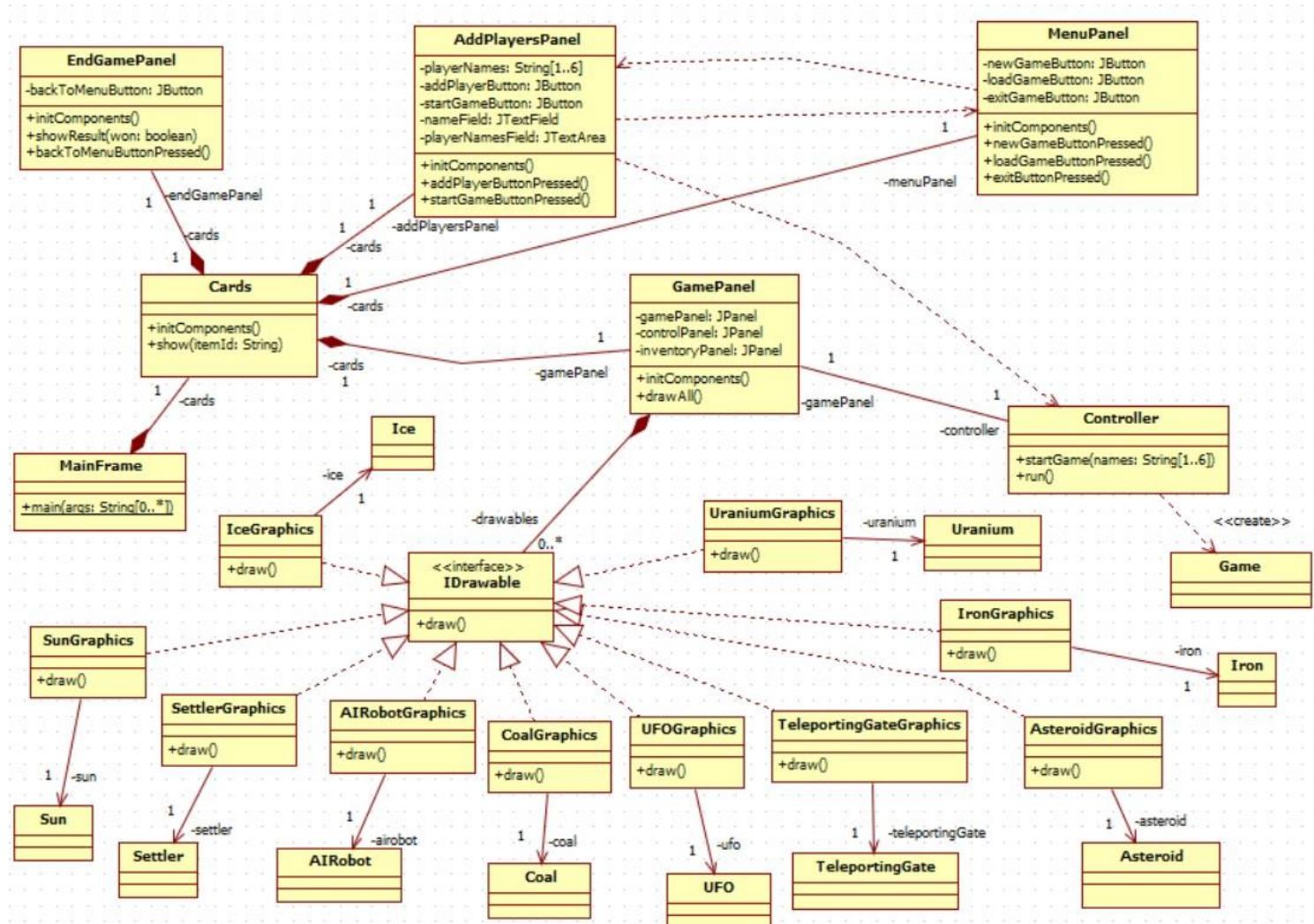
A cselekvések nagy részét a játékosok irányítják, különböző gombnyomásokkal:

- mozgás: az aktuális játékos aszteroidájának szomszédai kiemelve látszanak a lépés során (a szomszédok között találjuk a teleportkapukon keresztül elérhető szomszédokat is). A *move* gombra kattintás után a játékos egy ilyen szomszédos aszteroidára kattintva tud oda mozogni.
- fúrás: a *drill* gombra kattintva adhatjuk ki ezt a parancsot
- bányászat: a *mine* gombra kattintva adhatjuk ki ezt a parancsot
- visszatöltés: először a *restore* gombot kell megnyomnunk, majd a tárolónkból ki kell választanunk a visszatölteni kívánt objektum képét, és rákattintanunk
- építések: a *Create Robot* vagy *Create Teleport* gombra kattintva adhatjuk ki ezeket a parancsokat; az építés automatikusan megtörténik, ha lehetséges
- teleportkapu lerakása: a *Place Teleport* gombra kattintva adhatjuk ki ezt a parancsot
- passzolás: a *pass* gombra kattintással továbbadhatjuk a kört a következő játékosnak.

Az időzítő eközben természetesen lépteti a Napot, a robotokat, az ufókat és a megkergült teleportkapukat is.

A felületen található egy szövegdoboz, ami a játék állapotáról informál. Megjeleníti az éppen aktív játékos állapotát. Az aszteroidák gombok: a felületükre kattintva kiemelődnek a szomszédaik.

11.2.2 A felület osztály-struktúrája



11.3 A grafikus objektumok felsorolása

11.3.1 MainFrame

- **Felelősség**
A játékprogram főablaka. Egy Cards példányt tartalmaz.
- **Ősosztályok**
JFrame
- **Interfészek**
- **Attribútumok**
 - **private Cards cards:** A játék paneljait összefogó osztálypéldány.
- **Metódusok**
 - **public static void main(String[0..* args]):** A program belépési pontja. Létrehoz és megjelenít egy új MainFrame-et, ezzel elindul a grafikus alkalmazás.

11.3.2 Cards

- **Felelősség**
A főablakban megjelenítendő panelokat CardLayout-ban összefogó osztály.
- **Ősosztályok**
- **Interfészek**
- **Attribútumok**
 - **private EndGamePanel endGamePanel:** A játék végét és eredményét a felhasználó számára esztétikusan megjelenítő panel
 - **private AddPlayersPanel addPlayersPanel:** Új játékosok és neveik hozzáadására szolgáló panel
 - **private MenuPanel menuPanel:** A játék főmenüjét megjelenítő panel
 - **private GamePanel gamePanel:** Az aktív játékért felelős panel
- **Metódusok**
 - **public void initComponents():** Meghívja a panelok intiComponents() metódusait.
 - **public void show(String itemId):** A paraméterül kapott panelt (card-ot) jeleníti meg.

•

11.3.3 EndGamePanel

- **Felelősség**

A játék végét és eredményét a felhasználó számára esztétikusan megjelenítő panel.

- **Ősosztályok**

JPanel

- **Interfészek**

- **Attribútumok**

- **private JButton backToMenuButton:** gomb, amely a főmenübe irányít vissza
- **private Cards cards:** a játék paneljait összefogó osztálypéldány

- **Metódusok**

- **public void initComponents():** Inicializálja a panelen található komponenseket.
- **public void showResult(boolean won):** Kiírja a panelra, hogy a játékosok győztek, vagy vesztettek, és ugyanezt egy megfelelő képpel megjeleníti.
- **public void backToMenuButtonPressed():** Visszatér a főmenübe, azaz megjeleníti a MenuPanelt.

11.3.4 AddPlayersPanel

- **Felelősség**

Új játékosok és neveik hozzáadására szolgáló panel.

- **Ősosztályok**

JPanel

- **Interfészek**

- **Attribútumok**

- **private String[1..6] playerNames:** a panelon hozzáadott játékosok neveit ideiglenesen tároló lista
- **private JTextField nameField:** új játékos nevének szövegmezője
- **private JTextArea playerNamesField:** a már felvett játékosok neveinek többsoros szövegmezője
- **private JButton addPlayerButton:** játékos hozzáadását reprezentáló gomb
- **private JButton startGameButton:** játék indítását reprezentáló gomb
- **private Cards cards:** a játék paneljait összefogó osztálypéldány.

- **Metódusok**
 - **public void initComponents()**: Inicializálja a panelen található komponenseket.
 - **public void addPlayerButtonPressed()**: Az addButton gomb megnyomására hozzáadja a nameField szövegmezőben levő sztringet a playerNames listához, feltéve, hogy a játékosok száma egyelőre kisebb 6-nál, és a szöveg csak alfanumerikus, nem whitespace karaktereket tartalmaz. Ha sikeres a felvétel, frissíti a playerNames listából képzett sztringgel a playerNamesField szövegmező tartalmát.
 - **public void startGamesButtonPressed()**: A Controller startGame() függvényét hívva elindítja az új játékot, átadva a játékosok neveinek listáját; közben a gamePanelt jeleníti meg.

11.3.5 MenuPanel

- **Felelősség**

A játék főmenüjét megjelenítő panel, amelyen lehetőség van új játékot indítani, a legutóbbi játékokat betölteni, vagy kilépni a programból.

- **Ősosztályok**

JPanel

- **Interfészek**

- **Attribútumok**

- **private JButton newGameButton**: új játék elindítását reprezentáló gomb
- **private JButton loadGameButton**: a mentett játék elindítását reprezentáló gomb
- **private JButton exitGameButton**: kilépés eseményét reprezentáló gomb
- **private Cards cards**: a játék paneljait összefogó osztálypéldány

- **Metódusok**

- **public void initComponents()**: Inicializálja a panelen található komponenseket.
- **public void newGameButtonPressed()**: Új játék elindítása gombnyomással, hatására az addPlayersPanel jelenik meg.
- **public void loadGameButtonPressed()**: Betölti, majd elindítja a legutóbb elmentett játékot, ha van ilyen; megjeleníti a gamePanelt.
- **public void exitButtonPressed()**: Kilép az alkalmazásból; a grafikus képernyő is bezárul.

11.3.6 GamePanel

- **Felelősség**

A játékot grafikusan megjelenítő panel, ahol egy játék folyik.

- **Ősosztályok**

JPanel

- **Interfészek**

- **Attribútumok**

- **private Cards cards:** a játék paneljait összefogó osztálypéldány
- **private IDrawable[0..*] drawables:** az IDrawable interfést megvalósító grafikus objektumokat tartalmazó kollekció
- **private Controller controller:** a játék vezérlője
- **private JPanel gamePanel:** maga a játéktér, az aszteroidamező grafikus képe
- **private JPanel controlPanel:** a teleppel végezhető tevékenységek vezérlőgombjait tartalmazó panel
- **private JPanel inventoryPanel:** az összegyűjtött nyersanyagokat reprezentáló panel

- **Metódusok**

- **public void initComponents():** Inicializálja a panelen található komponenseket.
- **public void drawAll():** Meghívja a drawables kollekció minden elemének draw() metódusát, ezzel kirajzolja a játéktér változó objektumait a panelre.

11.3.7 CoalGraphics

- **Felelősség**

A Coal osztály grafikus megjelenítésére szolgál.

- **Ősosztályok**

- **Interfészek**

IDrawable

- **Attribútumok**

- **private Coal coal:** Az a Coal példány, amit kirajzol

- **Metódusok**

- **public void draw():** Újrarárajzolja az objektumot a referenciaként tárolt objektum aktuális állapota alapján

11.3.8 IceGraphics

- **Felelősség**

Az Ice osztály grafikus megjelenítésére szolgál.

- **Ősosztályok**

- **Interfészek**

IDrawable

- **Attribútumok**

- **private Ice ice:** Az az Ice példány, amit kirajzol

- **Metódusok**

- **public void draw():** Újrarájzolja az objektumot a referenciaikként tárolt objektum aktuális állapota alapján

11.3.9 IronGraphics

- **Felelősség**

Az Iron osztály grafikus megjelenítésére szolgál.

- **Ősosztályok**

- **Interfész**

IDrawable

- **Attribútumok**

- **private Iron iron:** Az az Iron példány, amit kirajzol

- **Metódusok**

- **public void draw():** Újrarájzolja az objektumot a referenciaikként tárolt objektum aktuális állapota alapján

11.3.10 UraniumGraphics

- **Felelősség**

Az Uranium osztály grafikus megjelenítésére szolgál.

- **Ősosztályok**

- **Interfészek**

IDrawable

- **Attribútumok**

- **private Uranium uranium:** Az az Uranium példány, amit kirajzol

- **Metódusok**

- **public void draw():** Újrarájzolja az objektumot a referenciaként tárolt objektum aktuális állapota alapján

11.3.11 SunGraphics

- **Felelősség**

A Sun osztály grafikus megjelenítésére szolgál.

- **Ősosztályok**

- **Interfész**

IDrawable

- **Attribútumok**

- **private Sun sun:** Az a Sun példány, amit kirajzol

- **Metódusok**

- **public void draw():** Újrarájzolja az objektumot a referenciaként tárolt objektum aktuális állapota alapján

11.3.12 AsteroidGraphics

- **Felelősség**

Az Asteroid osztály grafikus megjelenítésére szolgál.

- **Ősosztályok**

JButton

- **Interfészek**

IDrawable

- **Attribútumok**

- **private Asteroid asteroid:** Az az Asteriod példány, amit kirajzol

- **Metódusok**

- **public void draw():** Újrarájzolja az objektumot a referenciaként tárolt objektum aktuális állapota alapján

11.3.13 SettlerGraphics

- **Felelősség**

A Settler osztály grafikus megjelenítésére szolgál.

- **Ősosztályok**

- **Interfész**

IDrawable

- **Attribútumok**

- **private Settler settler:** Az a Settler példány, amit kirajzol

- **Metódusok**

- **public void draw():** Újrarájzolja az objektumot a referenciaként tárolt objektum aktuális állapota alapján

11.3.14 AIRobotGraphics

- **Felelősség**

Az AIRobot osztály grafikus megjelenítésére szolgál.

- **Ősosztályok**

- **Interfészek**

IDrawable

- **Attribútumok**

- **private AIRobot airobot:** Az az AIRobot példány, amit kirajzol

- **Metódusok**

- **public void draw():** Újrarájzolja az objektumot a referenciaként tárolt objektum aktuális állapota alapján

11.3.15 UFOGraphics

- **Felelősség**

Az UFO osztály grafikus megjelenítésére szolgál.

- **Ősosztályok**

- **Interfész**

IDrawable

- **Attribútumok**

- **private UFO ufo:** Az az UFO példány, amit kirajzol

- **Metódusok**

- **public void draw():** Újrarájzolja az objektumot a referenciaként tárolt objektum aktuális állapota alapján

11.3.16 TeleportingGateGraphics

- **Felelősség**

A TeleportingGate osztály grafikus megjelenítésére szolgál.

- **Ősosztályok**

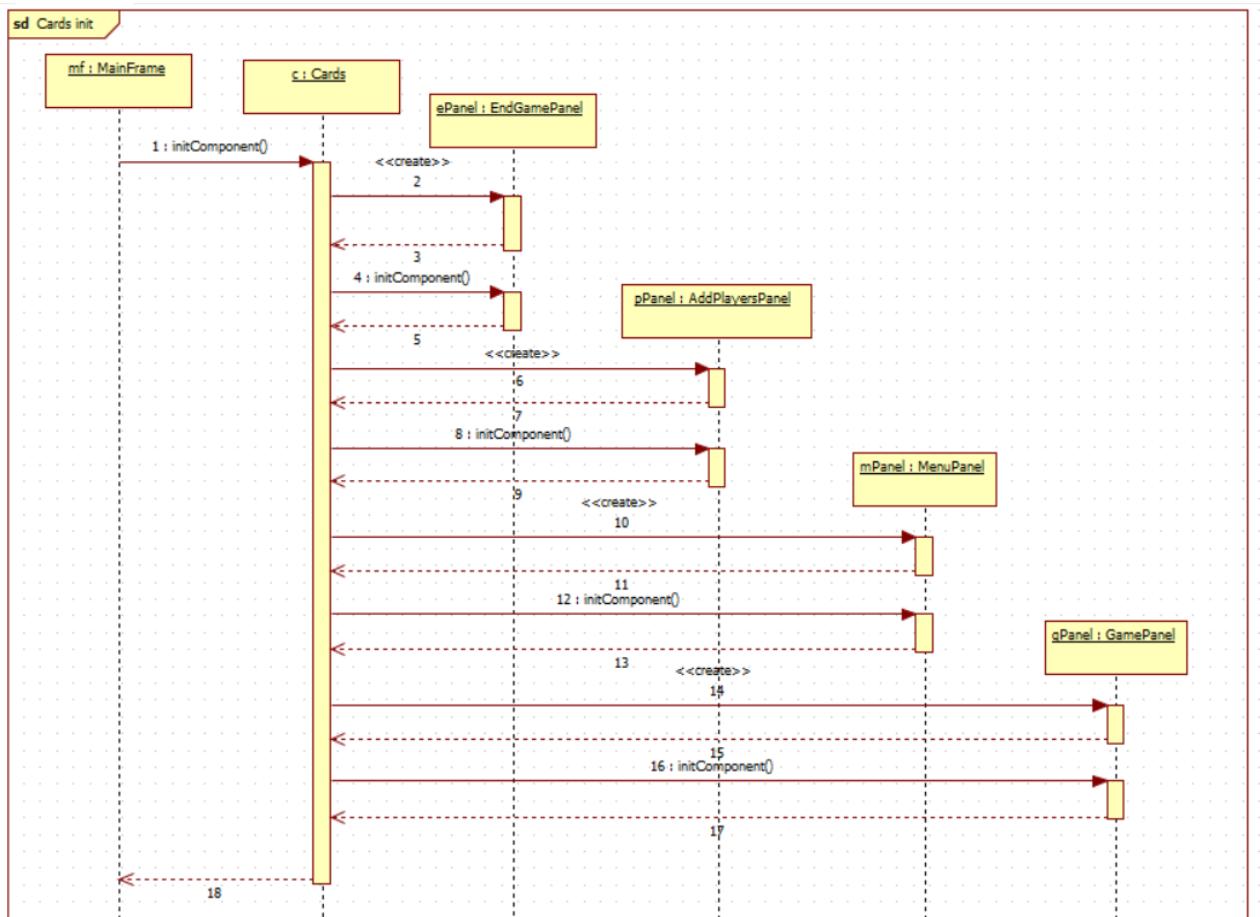
- **Interfész**

IDrawable

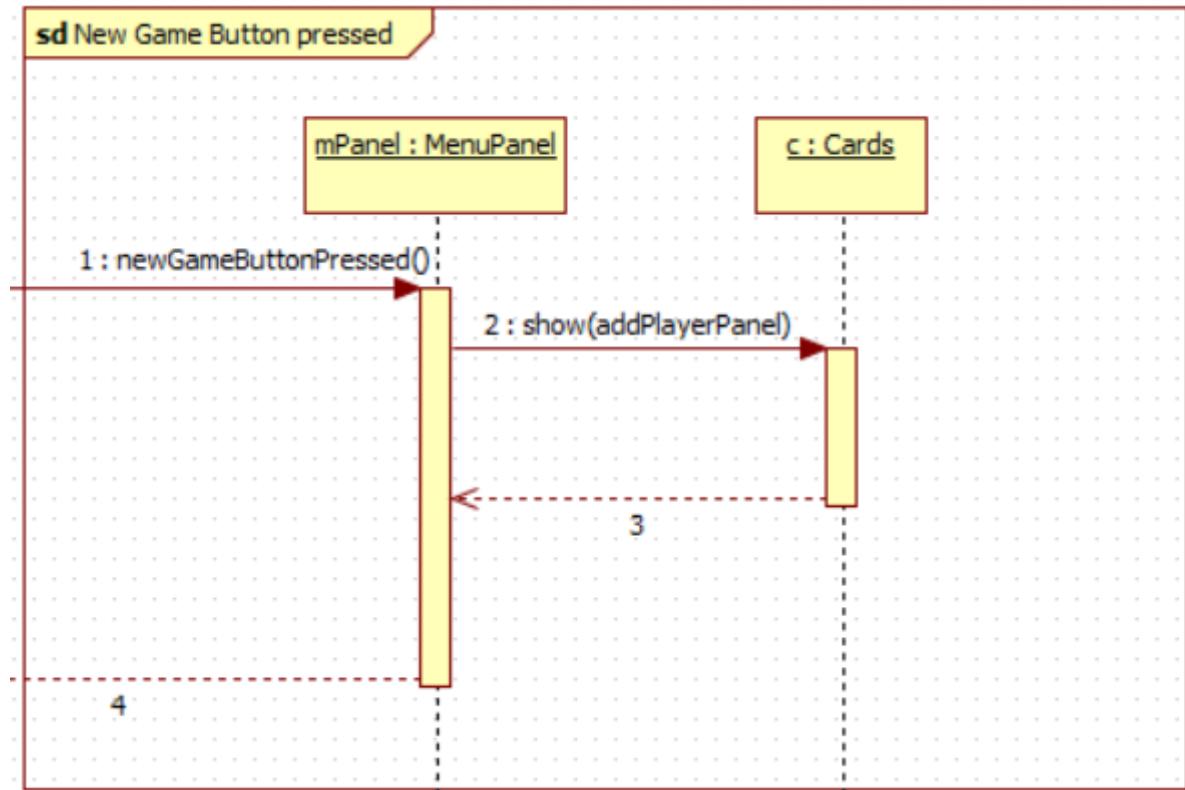
- **Attribútumok**
 - **private TeleportingGate teleportingGate**: Az a TeleportingGate példány, amit kirajzol
- **Metódusok**
 - **public void draw()**: Újrarajzolja az objektumot a referenciaként tárolt objektum aktuális állapota alapján

11.4 Kapcsolat az alkalmazói rendszerrel

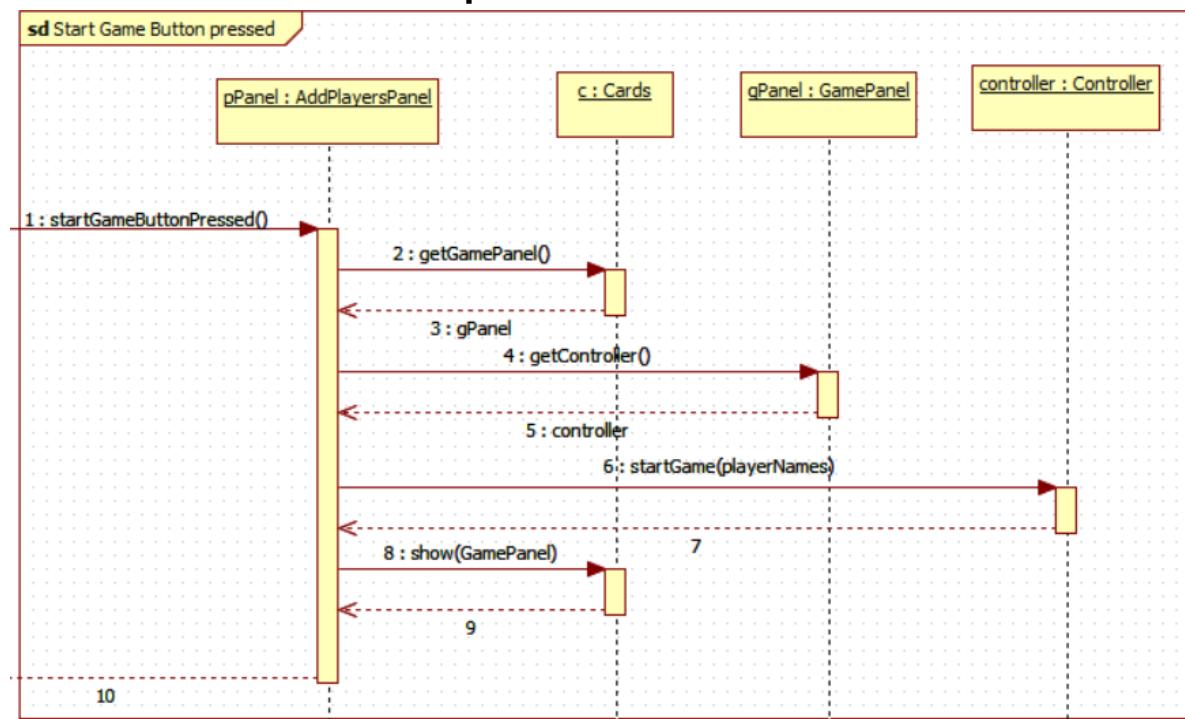
11.4.1 Cards init



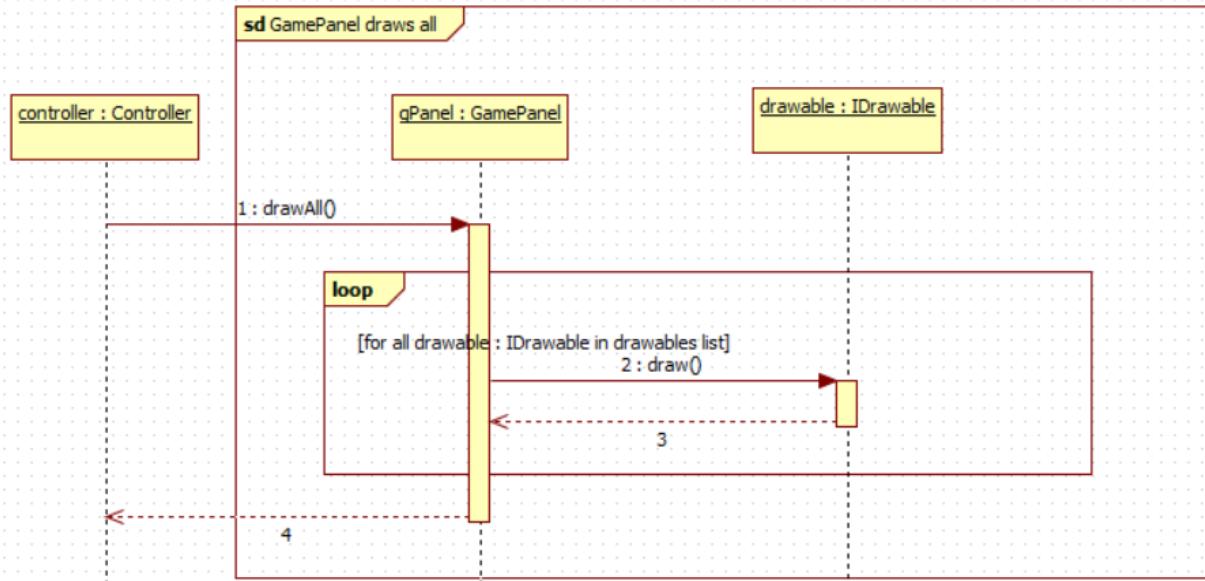
11.4.2 New Game Button pressed



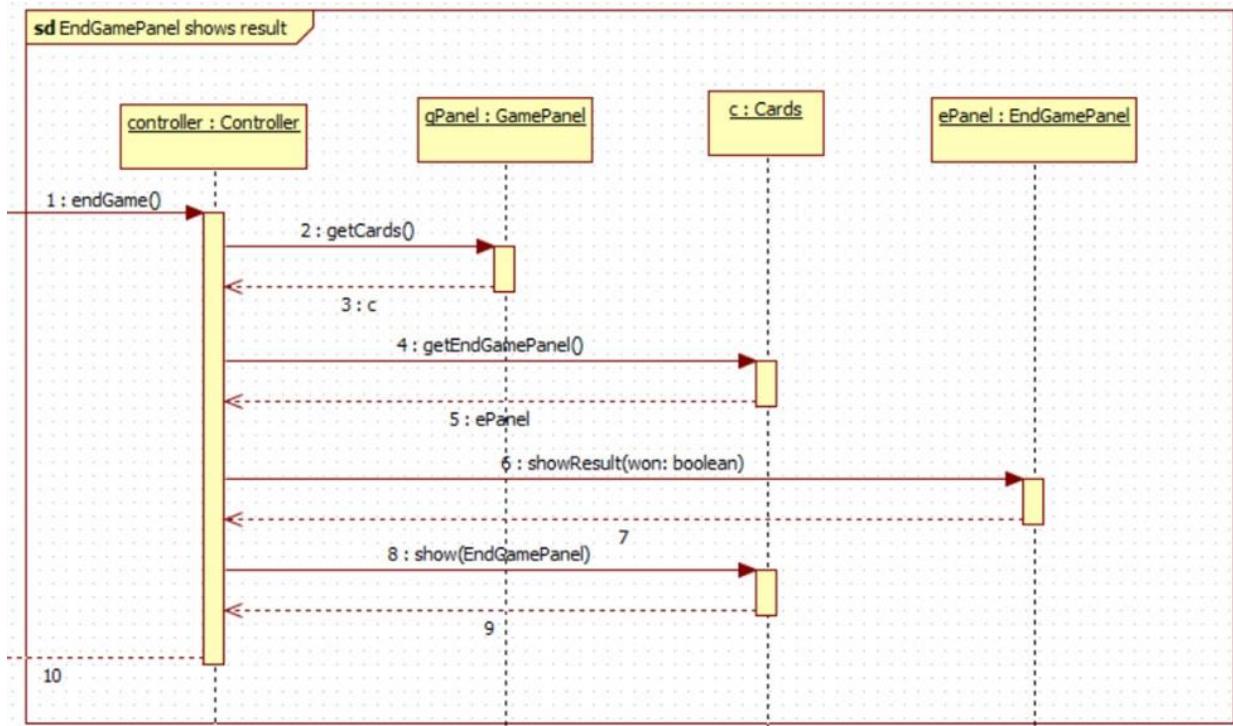
11.4.3 Start Game Button pressed



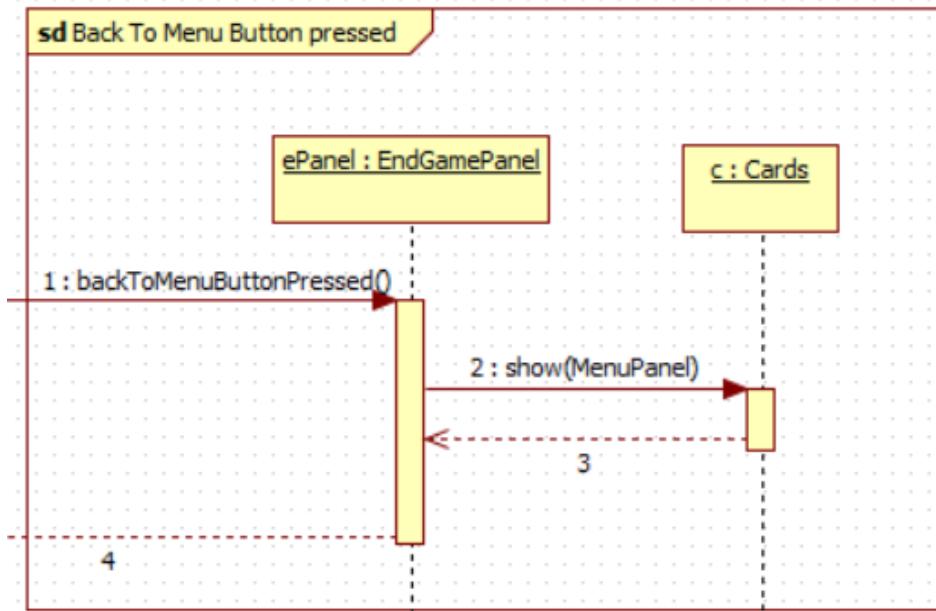
11.4.4 GamePanel draws all



11.4.5 EndGamePanel shows result



11.4.6 Back To Menu pressed



11.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2021.04.24. 16:00	3 óra	Kreták Szögi Mesterházi Havasi Radnai	Értekezlet. Döntés: Kreták és Szögi megrajzolja a diagramokat, Havasi és Radnai a felület működési elveit tervezik meg, Mesterházi a grafikus interfész készít el
2021.04.24. 19:00	3 óra	Radnai Havasi	Tevékenység: Menü leprogramozása, kód tisztázása
2021.04.24. 19:00	2 óra	Kreták Szögi	Tevékenység: Szekvenciadiagramok megrajzolása, az osztálydiagram kiegészítése.
2021.04.24. 22:00	2,5 óra	Mesterházi	Tevékenység: 11.1 és 11.2 fejezet dokumentálása Menü és pályaképek vázlatának elkészítése.
2021.04.24. 21:00	2 óra	Szögi	Tevékenység: Az osztályok dokumentációjának leírása.
2021.04.25. 14:00	2 óra	Kreták Szögi Mesterházi Havasi Radnai	Értekezlet. Elvégzett feladatok ellenőrzése, kisebb javítások a dokumentációban és a diagramokon.

Az adott heti munka csapattagoknál aránya:

Csapattag	Munka aránya százalékban
Havasi	20
Kreták	20
Mesterházi	20
Radnai	20
Szögi	20

13. Grafikus változat beadása

7 – berenyi_kft

Konzulens:

Csapattagok

Kreták Barnabás András

Mesterházi Marcell

Szögi Richárd

Havasi Balázs

Radnai Bálint

2021. 05. 06.

13. Grafikus változat beadása

13.1 Fordítási és futtatási útmutató

13.1.1 Fájllista

A src/berenyi_kft mappában található forrásfájlok:

Fájl neve	Méret (bájt)	Keletkezés ideje	Tartalom
AIRobot.java	3062	2021.03.16	AIRobot osztály megvalósítása
Asteroid.java	20604	2021.03.16	Asteroid osztály megvalósítása
Character.java	3441	2021.03.16	Character osztály megvalósítása
Coal.java	1401	2021.03.16	Coal osztály megvalósítása
Controller.java	7735	2021.04.07	Controller osztály megvalósítása
Game.java	16474	2021.03.16	Game osztály megvalósítása
ISteppable.java	305	2021.03.16	ISteppable osztály megvalósítása
Ice.java	1804	2021.03.16	Ice osztály megvalósítása
Iron.java	1397	2021.03.16	Iron osztály megvalósítása
Player.java	4162	2021.04.07	Player osztály megvalósítása
PlayerCommand.java	1461	2021.04.07	PlayerCommand osztály megvalósítása
Proto.java	28449	2021.04.07	Proto osztály megvalósítása
RadioactiveResource.java	543	2021.03.16	RadioactiveResource osztály megvalósítása
Recipe.java	4365	2021.03.16	Recipe osztály megvalósítása
Resource.java	1612	2021.03.16	Resource osztály megvalósítása
Settler.java	14391	2021.03.16	Settler osztály megvalósítása
Skeleton.java	34578	2021.03.17	Tesztelések megvalósítása, hívása
State.java	1100	2021.04.07	State osztály megvalósítása
Sun.java	5006	2021.03.16	Sun osztály megvalósítása
TeleportingGate.java	7029	2021.03.16	TeleportingGate osztály megvalósítása
Timer.java	6036	2021.03.16	Timer osztály megvalósítása
Ufo.java	2488	2021.04.07	Ufo osztály megvalósítása
Uranium.java	3371	2021.03.16	Uranium osztály megvalósítása
last_level_config.txt	4854	2021.05.05	a legutóbbi játék konfigurációját tároló fájl

A src/berenyi_kft_GUI mappában található forrásfájlok:

Fájl neve	Méret (bájt)	Keletkezés ideje	Tartalom
AddPlayersPanel.java	5678	2021.04.28	A játékosok hozzáadására szolgáló panel megvalósítása
AIRobotGraphics.java	3665	2021.04.28	AIRobot osztály grafikus megjelenítésének megvalósítása
AsteroidGraphics.java	10600	2021.04.28	Asteroid osztály grafikus megjelenítésének megvalósítása
Cards.java	1974	2021.04.28	A panelokat összefogó osztály megvalósítása
CoalGraphics.java	2669	2021.04.28	Coal osztály grafikus megjelenítésének megvalósítása
EndGamePanel.java	4570	2021.04.28	A játék végén megjelenő panel megvalósítása
GamePanel.java	23575	2021.04.28	GamePanel megvalósítása
IceGraphics.java	2646	2021.04.28	Ice osztály grafikus megjelenítésének megvalósítása
IDrawable.java	328	2021.04.28	IDrawable osztály megvalósítása
IronGraphics.java	2694	2021.04.28	Iron osztály grafikus megjelenítésének megvalósítása
MainFrame.java	1449	2021.04.28	MainFrame megvalósítása
MenuPanel.java	5480	2021.04.28	MenuPanel megvalósítása
SettlerGraphics.java	4315	2021.04.28	Settler osztály grafikus megjelenítésének megvalósítása
SunGraphics.java	3534	2021.04.28	Sun osztály grafikus megjelenítésének megvalósítása
TeleportingGateGraphics.java	4824	2021.04.28	TeleportingGate osztály grafikus megjelenítésének megvalósítása
UFOGraphics.java	3583	2021.04.28	UFO osztály grafikus megjelenítésének megvalósítása
UraniumGraphics.java	2698	2021.04.28	Uranium osztály grafikus megjelenítésének megvalósítása

Köszönnettel tartozunk **Radnai Viktornak** a játékban szereplő képek és háttérképek (rajzok) elkészítéséért (Radnai Bálinton keresztül egyeztetve a csapattal).

A src/berenyi_kft_GUI/Icons mappában található képek:

Fájl neve	Méret (bájt)	Keletkezés ideje	Tartalom
aiRobot.png	178122	2021.05.03	AI Robotot ábrázoló kép
asteroid_emphasized.png	351211	2021.05.03	kiemelt aszteroidát ábrázoló kép
asteroid_emphasizer_glow.png	63680	2021.05.03	kiemelt aszteroida gyűrűjét ábrázoló kép
asteroid_old.png	155974	2021.05.04	régebbi, aszteroidát ábrázoló kép
asteroid_solarstorm_glow.png	63550	2021.05.03	aszteroidán napvihar gyűrűt ábrázoló kép
asteroid_solarstorm.png	350259	2021.05.03	aszteroidán napvihart ábrázoló kép
asteriod.png	318311	2021.04.28	aszteroidát ábrázoló kép
background.png	2265771	2021.05.03	a játékpanel hattérképe, galaxist ábrázoló kép
background1.png	430663	2021.05.05	régebbi, galaxist ábrázoló kép
coal.png	425154	2021.05.03	szenet ábrázoló kép
gameFailed.png	1831247	2021.05.04	a játékosok vereségét ábrázoló kép
gameWon_old.png	3014710	2021.05.04	régebbi, a játékosok győzelmét ábrázoló kép
gameWon.png	2955209	2021.05.04	a játékosok győzelmét ábrázoló kép
gate_emphasizer_glow.png	46191	2021.05.04	kiemelt teleport gyűrűjét ábrázoló kép
ice.png	328498	2021.05.03	vízjeget ábrázoló kép
inventorypanel.png	586	2021.05.04	nyersanyagraktár hátterét ábrázoló kép
iron.png	157347	2021.05.03	vasat ábrázoló kép
logo.png	28656	2021.05.04	a játék ikonját ábrázoló kép
settler_emphasized.png	128325	2021.05.04	kiemelt űrhajót ábrázoló kép
settler_emphasized_glow.png	39561	2021.05.04	kiemelt űrhajó gyűrűjét ábrázoló kép
settler.png	101111	2021.04.28	űrhajót (telepest) ábrázoló kép
start_16_9.png	2237344	2021.05.04	a játékablak 16:9 arányú háttérképe
start_old.png	2973613	2021.05.04	régebbi játék kezdőkép
start.png	1023931	2021.05.03	4:3 arányú kezdőkép
sun_old.png	578383	2021.05.04	régebbi, Napot ábrázoló kép

sun_solarstorm_glow.png	477583	2021.05.03	Nap napszél-gyűrűjét ábrázoló kép
sun_solarstorm.png	1025601	2021.05.03	napviharos Napot ábrázoló kép
sun.png	578383	2021.05.03	Napot ábrázoló kép
teleportingGate_emphasized.png	376496	2021.05.04	kiemelt teleportkaput ábrázoló kép
teleportingGate.png	342509	2021.05.03	teleportkaput ábrázoló kép
ufo.png	139439	2021.05.03	UFO-t ábrázoló kép
uranium.png	307336	2021.05.03	uránt ábrázoló kép

A projekt gyökérképában található futtatható fájlok:

Fájl neve	Méret (bájt)	Keletkezés ideje	Tartalom
berenyi_kft_runnable.jar	23841099	2021.05.08	A futtatható játékot tartalmazó JAR fájl
README.md	302	2021.05.09	A projekt README információs fájlja

13.1.2 Fordítás és telepítés

A program futtatása kétféle módon történhet:

1. Eclipse-be importálva, fordítva, majd futtava a projektet
2. Futtatható JAR fájlon dupla kattintással

A második lehetőség egyszerűbb, így azt javasoljuk. Ha viszont szeretnénk részletesebben megvizsgálni a projekt forrásfájljait, akkor válasszuk az Eclipse-es megoldást.

A fordítás és futtatás szekciók után tömören összefoglaljuk a játék nem teljesen egyértelmű használati eseteit.

Fordítani csak az első változatnál kell.

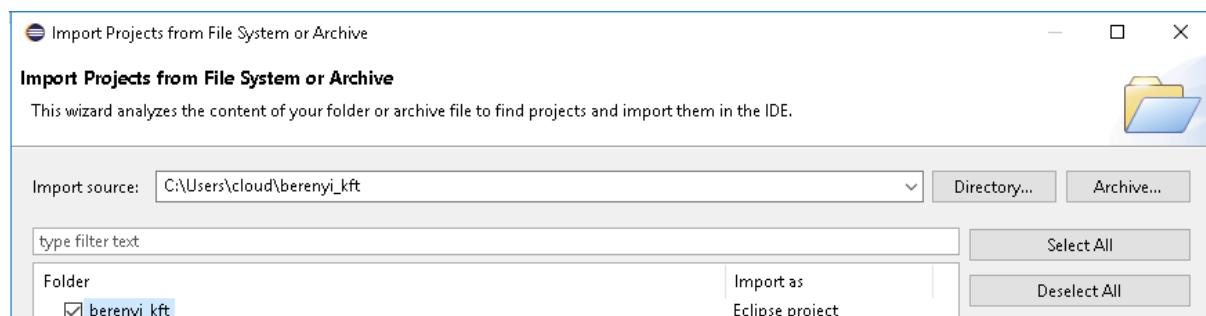
A fordításhoz a virtuális gépeken megtalálható JDK 11 környezet szükséges.

A programunkat Eclipse-en keresztül fordítjuk:

A kicsomagolt mappát importáljuk, majd Eclipse-ben navigálunk így:

File -> Open Projects from File System

majd itt a Directory-ban válasszuk ki a gyökérképát.



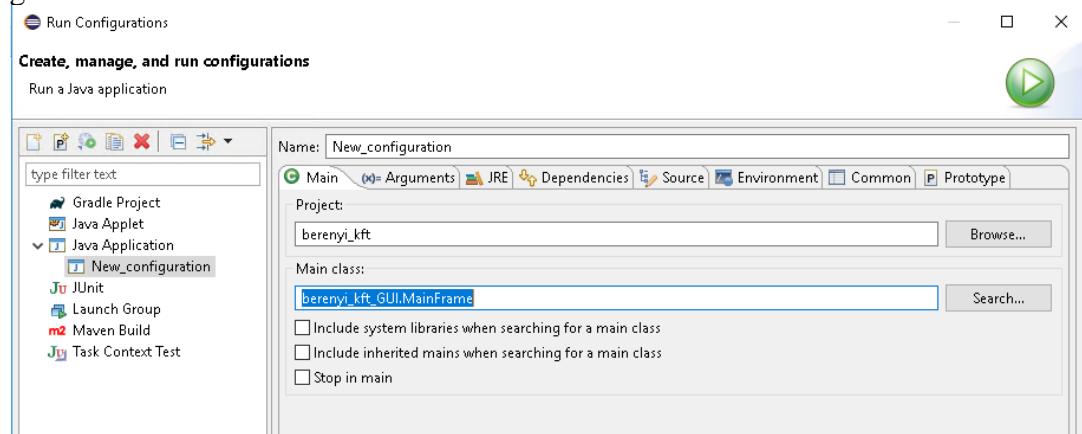
A fordítást ezután a Project menüben található Build Project menüelemre kattintva tehetjük meg.

13.1.3 Futtatás

13.1.3.1 Első lehetőség - Eclipse-ben:

A futtatáshoz a virtuális gépeken megtalálható JRE 11 futtatókörnyezet szükséges.

Vállaszuk ki a **Run/Run Configurations...** fület, a projekt a *berenyi_kft*, a Main class pedig *berenyi_kft_GUI.MainFrame*. Végül kattintsunk az Apply, majd a Run gombra.



A lépések után elindul a program:



13.1.3.2 Második lehetőség - futtatható JAR-ral (javasolt):

A futtatáshoz a virtuális gépeken megtalálható JRE 11 futtatókörnyezet szükséges.

JAR közvetlen indítása:

Keressük meg a projekt src mappájában a berenyi_kft_runnable.jar futtatható fájlt. A programot elindíthatjuk a JAR fájlra duplán kattintva.

Elindíthatjuk akár parancssorból is, ehhez navigálunk a kicsomagolt projekt gyökérmapjába (berenyi_kft), majd adjuk ki a java –jar berenyi_kft_runnable.jar parancsot.

Információ a futó játék használatához:

- A játékablakot javasolt az eredeti méretezésben hagyni.
- Az addPlayersPanel-en írjuk be a szövegmezőbe az új játékos nevét. A név nem lehet üres, nem tartalmazhat whitespace karaktert, és ékezes karaktert sem. Az Add feliratú gombra kattintva adhatjuk hozzá a játékosok listájához, játékost törölni nem lehet (új játék indítása esetén üresedik a lista).
- A játékpanelen egy aszteroidára kattintva kék színnel emeljük ki a szomszédait. Az aktuális játékos telepsét is kiemeljük, öt narancssárga színnel. Kattintsunk a telepes aszteroidáján, majd a kékkel kiemelt szomszédok közül az egyikre. Ezután nyomjuk meg a Move gombot, ekkor a telepes átrepül a legutóbb kattintott aszteroidára.
- Nyersanyag visszatöltéséhez először kattintsunk az inventoryPanel-en a megfelelő nyersanyagra. A Restore gombra nyomva a kiválasztott típusú nyersanyag töltődik vissza az aszteroidába (ha ennek feltételei teljesülnek).
- A játékpanel End Game gombjával megszakíthatjuk a játékot, amely ezután a menuPanel Load gombjára kattintva tölthető be. A többi parancsgomb működése magától értetődő.
- A léptethető objektumok 5 másodpercenként lépnek. Ha napvíhar közeledik, figyelmeztető üzenet jelenik meg az üzenőfalon. 3 tick-kel, azaz 15 másodperccel a napszél kitörése előtt a nap és a napközeli aszteroidák pirossal kiemelődnek. Ha megérkezik a napvíhar, akkor azok a karakterek, amelyek ideiglenesen el tudnak bújni, eltünnek a képről, de a napvíhar múltával újra előbújnak (megjelennek).

13.2 Értékelés

Tag neve	Tag neptun	Munka százalékban
Havasi Balázs		20
Kreták Barnabás András		20
Mesterházi Marcell		20
Radnai Bálint		20
Szögi Richárd		20

13.3 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2021.04.28. 14:00	2 óra	Havasi Kreták Mesterházi Radnai Szögi	Értekezlet: Grafikus felület megvalósításának megbeszélése, feladatok átgondolása Döntés: A játékpanelen abszolút koordinátákat használunk. Az aszteroidák gombokként, a többi játékbeli objektum címkekkel jelentek meg a képernyőn. Döntés: Első körben Radnai és Havasi kialakítja a grafikus felület alapjait, Mesterházi készíti a játékpanel gombjait és az üzenőfalat, Kreták és Szögi a játékpanelen kívüli paneleken dolgozik.
2021.04.28. 16:00	3 óra	Radnai Havasi	Tevékenység: Grafikus felület alapvető kialakítása, panelek, gombok és grafikus osztályok létrehozása
2021.05.01. 18:00	3 óra	Radnai	Tevékenység: Grafikus osztályok implementálása: aszteroidák, karakterek, kapuk, nap
2021.05.02. 17:00	2 óra	Radnai	Tevékenység: Karakterek táblázatos megjelenítése az aszteroidákon, szomszédos aszteroidák kiemelése
2021.05.01. 13:00	1,5 óra	Mesterházi	Tevékenység: Vezérlőpanel GUI elkészítése (gombok messagebox)
2021.05.03. 14:00	1 óra	Kreták Mesterházi Radnai Szögi	Értekezlet: A grafikus felület részletesebb tervezének áttekintése, újabb feladatok kiosztása Döntés: Mesterházi görgethetővé teszi az üzenőfalat, és rendezi az AddPlayerPanel-t. Kreták elrendezi az EndGamePanelt és a képeket. Havasi implementálja az InventoryPanel-t. Radnai ellenőrzi a grafikus objektumok életciklusát. Szögi megjeleníti az aktuális Settler-t és a napvihart.
2021.05.03. 13:00	2 óra	Mesterházi	Tevékenység: InventoryPanel GUI elkészítése
2021.05.03.14:00	2 óra	Szögi	Tevékenység: Aktuális Settler és a napvihart élvező aszteriodák kijelölésének megoldása
2021.05.03.18:00	2 óra	Kreták	Tevékenység:

			A menuPanel és az endGamePanel háttérének beállítása, gombok elrendezése
2021.05.04.16:00	1 óra	Kreták Mesterházi Radnai Szögi	Értekezlet: Helyzetjelentés, további feladatok keresése és kiosztása.
2021.05.04. 17:00	3 óra	Radnai	Tevékenység: Vezérlőpanel gombjai listener-einek megírása, vezérlés megvalósítása, időzítés alapok
2021.05.04. 18:00	3 óra	Kreták Szögi	Tevékenység: A grafikus aszteroidák megfelelő távolságra való megjelenítése. Egyéb bugok javítása.
2021.05.04. 20:00	2 óra	Mesterházi	Tevékenység: GUI további formázása, bugok javítása, bombok igazítása, messagebox formázások
2021.05.05. 14:00	2 óra	Radnai	Tevékenység: Karakterek táblázatos megjelenítése az aszteroidákon, szomszédos aszteroidák kiemelése
2021.05.05. 16:00	1 óra	Radnai	Tevékenység: Save-load megoldása, Timer bekötésének javítása
2021.05.05. 17:00	1 óra	Radnai	Tevékenység: karakterek elbújásának implementálása, kisebb panelformázások
2021.05.05. 18:00	1 óra	Kreták Mesterházi Radnai Szögi	Értekezlet: Helyzetjelentés, további feladatok keresése és kiosztása.
2021.05.05. 19:00	2 óra	Kreták	Tevékenység: A telepes által eltárolt nyersanyagok számának megjelenítése az inventory-n
2021.05.06. 18:00	3 óra	Szögi	Tevékenység: Kiíratás javítása: aktuális játékos, vastagságok, sikeres volt-e a művelet stb.
2021.05.06. 19:00	0,5 óra	Kreták	Tevékenység: A telepes által eltárolt teleportkapuk számának megjelenítése
2021.05.07. 19:00	1 óra	Radnai	Tevékenység: Futtatható JAR fájl és parancsikon hozzáadása
2021.05.08. 15:00	2 óra	Kreták Mesterházi Radnai Szögi	Tevékenység: Dokumentációk írása, tisztázása, projekt véglegesítése

14.

Összefoglalás

7 – berenyi_kft

Konzulens:

Csapattagok

Kreták Barnabás András

Mesterházi Marcell

Szögi Richárd

Havasi Balázs

Radnai Bálint

2021. 05. 07.

14. Összefoglalás

14.1 A projektre fordított összes munkaidő

Tag neve	Munkaidő (óra)
Kreták Barnabás András	118
Mesterházi Marcell	128
Szögi Richárd	119
Havasi Balázs	115
Radnai Bálint	137
Összesen	617

- A feltöltött programok forrássorainak száma**

Fázis	KódSOROK száma
Szkeleton	2623
Prototípus	5705
Grafikus változat	8990
Összesen	8990

14.2 • Projektösszegzés

14.2.1 Mit tanultak a projektből konkrétan és általában?

Megtanultunk elosztott verziókezelő rendszert használni, így jó gyakorlatunk lett benne. Jobban odafigyeltünk az időbeosztásunkra, és a félév előrehaladtával az elvégzendő feladatok mennyiségével egyre arányosabban szerveztük a megbeszéléseinket. Hatékonyabbnak bizonyult több megbeszélést tartani, és azokon kisebb feladatokat elvállalni/kiosztani magunknak.

14.2.2 Mi volt a legnehezebb és a legkönnyebb?

Legnehezebb a teljes modell megtervezése az első hetekben és a Prototípus tesztjeinek kivitelezése. A grafikus felület megtervezése sem bizonyult könnyű feladatnak, mivel mindegyikünk csak az előző félév óta ismerte valamilyen szinten a Swing keretrendszerét. Nehéz volt a nehezebb heteken - közeledve a végéhez - időt és energiát találni.

Könnyű volt a feladatokat elosztani, általában jól elkülöníthető részek voltak. Hatékonyan együtt tudtunk működni, mint csapat, mindenki lelkesen kivette a részét a munkából.

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

Általában igen, de szerintünk a kivitelezések több pontot kellene, hogy érjenek, mint azok megtervezése (természetesen nagyon fontos volt a megfelelő tervezés, de a kivitelezés során sok apró részfeladatot meg kellett még oldanunk, és összességében több munkát jelentett).

A kreditek számával nem találtuk arányosnak a befektetett munkaórák számát; a tárgy 4 vagy akár 5 kreditet is érhetet.

14.2.4 Ha nem, akkor hol okozott ez nehézséget?

14.2.5 Milyen változtatási javaslatuk van?

A heti feladatok elvégzéséhez kaphatnánk alapból több iránymutatást. Esetleg konkrétabb példákkal: egy-két központilag kiadott diagram vagy dokumentáció-szekció biztosabb iránymutatást adna.

14.2.6 Milyen feladatot ajánlanának a projektre?

A következőkben bemutatjuk a *Crypt Raider* nevű flash játékot, ami egy izgalmas új játék ötletét adhatja. "Sajnos" létezik elérhető és részletes másolat-implementáció a játékhoz (<https://github.com/GambuzX/Ascii-Raider>), de valamilyen jelentősen módosított változatban feladható lehet.

A játék az objektumok jellegében, a léptethetőségben, és a reakciók polimorfizmusában egyébként hasonló a Pacmanhez, illetve a Sokobanhoz, ebből a szempontból teljesen alkalmas lenne projekt laboratórium feladatnak (például a kulcs, az ajtó, a múmia vagy a dinamit lehetnek a megrendelő módosítási kérelmei). A többjátékos módnak sincs akadálya, sőt, egy szabotőr játékos bújhatalna egy múmia bőrébe is.

Crypt Raider



A játékban a kalendorunkat irányítjuk egy kriptában. A játék pályákból áll. minden pályán található néhány varázsgömb, amelyeket a pálya speciális portáljába kell juttatnunk a kalendorral. Ha minden varázsgömb a portálba került, a portál aktiválódik, és ha a kalendorral félállunk, a következő pályára jutunk.

A pályák téglalap alakúak, négyzetrácsosak. Bizonyos mezőkön stabil, áthatolhatatlan falak vannak. minden pályán egyetlen portál van. A pálya többi mezője lehet szabad, de lehet rajtuk valamilyen akadály is (körök, földkupac, vagy ajtó). minden ilyen mezőn egyidőben legfeljebb egy dolog lehet.

Az akadályokat tartalmazó mezők falként viselkednek, de amikor az akadály eltűnik róluk, üresek, szabadok lesznek. A földkupacot a kalandor azonnal eltakarítani, ha a mezőre lép, de megteheti azt is, hogy szomszédos mezőről takarítja el a földet. A körakás csak robbanás hatására tűnik el, de a robbanás megsemmisíti a földkupacot is. Az ajtóra nem hat a robbanás, csak a kulcs felvételével nyitható ki. minden pályán legfeljebb egy ajtó van, és ha van ajtó, pontosan egy kulcs tartozik hozzá.

A pálya szabad mezőin karakterek mozognak, akik minden lépésben egy oldalszomszédos szabad mezőre lépnek. A kalandor az a karakter, akit a játékos irányít a joystick-gombokkal. A kalandorra szörnyek leselkednek, amelyek a játékban időzítetten lépnek. A szkarabeusz bogarak minden lépésükben közeledni próbálnak a kalandorhoz – ha nem tudnak, egyhelyben maradnak. Ha a kalandor szörnyel ütközik, a szörny felrobban, és a karakter is meghal. A szörnyek egymásra nem veszélyesek.

A pályán különféle tárgyak vannak, például varázsgömb, szikla, kulcs és dinamit. Közös jellemzőjük, hogy gravitáció hat rájuk (ami a karakterekre nem): ha az alattuk levő mező szabad, akkor leesnek, és csak akkor állnak meg, ha talajt érnek. A leesés is időzítetten történik. Ha a karakter ugyanarra a mezőre lép, mint ahol a kulcs van, akkor felveszi, és az ajtó kinyílik. A többi tárgy nehéz: ha karakterre esik, a karakter meghal; ha dinamitra esik, a dinamit felrobban. Az éppen nem leeső nehéz tárgyakat a kalandor vízszintesen a szomszédos mezőre tudja tolni, de egyszerre legfeljebb csak egyet.

Ha egy varázsgömb a portált tartalmazó mező fölötti mezőre kerül, a portál beszippantja, és ez 10 ponttal növeli a játékos pontszámát. Ha a pálya összes (eredetileg is meglevő) varázsgömbje a portálba került, a portál a föléálló kalandort is fogadja. Más dolog nem kerülhet a portálba. Ha a játékos teljesített egy pályát, a maradék másodpercek bónuszként szintén a pontszámához adódnak.

A dinamit bárhol is ért talajt, felrobban. A karakterek és a dinamit robbanása is olyan, hogy a csúcs- és oldalszomszédos nyolc mező mindegyikére kiterjed a hatása, kivéve, ha fal, portál vagy ajtó van a mezőn. Ha egy robbanás szele elér egy mezőt, a mezőn levő dolog megsemmisül. Ha pedig a megsemmisülő dolog még robbanó dolog is (például szörny vagy dinamit), akkor az is felrobban, továbbterjesztve a robbanást.

Minden pályán van a pályára jellemző időlimit. Ha ez letelik, a játékos meghal, az életeinek száma eggyel csökken, és újra kell kezdenie a pályát. A kalandornak kezdetben három élete van, ha mind elfogyott, a játéknak vége. Ha viszont sikerült az összes pályát teljesítenie, a játékos végigvitte a játékot. minden pályákhoz tartozik jelszó is, amely beírásával a játékos az adott pályától kezdheti a játékot.

14.2.7 Egyéb kritika és javaslat

A fentebb említetteken kívül nincs más.