

# 3. Analízis modell kidolgozása 1.

27 – NASchA

Konzulens:  
Goldschmidt Balázs

## Csapattagok

Czanik Bálint	H7EEPG	czanik.balint@gmail.com
Nagy Örs	P28RW9	nagyors456@gmail.com
Novák Bálint Huba	OHKFY9	novak.balint.huba@gmail.com
Tokovics Dávid Tamás	H9LGJI	tokovicsdavid00@gmail.com
<b><u>Szabó Bence Sándor</u></b>	<b><u>NQB6UO</u></b>	<b><u>szabo.bence.sandor@gmail.com</u></b>

2021. március 1.

### 3. Analízis modell kidolgozása

#### 3.0 Előző dokumentum hibáinak javítása

##### 3.0.1 Szótár kiegészítése:

- **belép:** A telepes/robot használja az egyik kaput és így teleportál a kapu másik tagjára.
- **égitest:** Az aszteroida övben elhelyezkedő aszteroidák.
- **hírt kap:** A napviharokról értesítést kap a játékos (pár körrel a napvihar kitörése előtt)
- **kerüli:** A program alapvetően úgy mozgatja a robotokat, hogy ne kerüljenek egy aszteroidára többen egyszerre, azonban, ha ez mégis megtörténik, akkor minél kevesebb ilyen esetet próbál generálni a program.
- **stagnál:** Az aszteroidák egyhelyben állnak/maradnak.

##### 3.0.2 Felhasználók (változtatás):

- A játékot akármennyi játékos játszhatja.

##### 3.0.3 Funkciók kiegészítése (a változtatások félkövérrel ki vannak emelve):

- „...A játékos(ok) a játék során telepes(eke)t irányít(anak), körökre osztottan, tehát egy körben egy telepes, csak egy akciót tud végezni. Ezek az akciók lehetnek: mozgás, fúrás, bányászat, robotépítés, teleport kapupár építés, **teleport kapupár elhelyezése, teleportálás**, bázis építése, várakozás és nyersanyag visszahelyezése az aszteroidába...”
- „...A teleport kapuval való mozgás egy külön akcióként tekintendő. Egy telepes vagy egy robot rálép egy teleport kapura, majd a következő körben dönthet arról, hogy várakozik az adott kapunál (vagyis nem csinál semmit az adott körben), vagy belép a kapun és annak a kapunak a párjánál megjelenik, illetve még dönthet úgyis, hogy nem lép be az adott teleport kapun, hanem rálép egy másik szomszédos aszteroidára vagy kapura, ami nem az éppen tartózkodó kapu párja (ez alól kivétel, ha véletlen pont az a kapu lenne, ami éppen a párja, de erre nem azért lépne, mert teleportál).”

##### 3.0.4 Funkcionális követelmények változásai (a változtatások félkövérrel ki vannak emelve):

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Use-case	Komment
fk27	Egy kaput csak akkor lehet használni, <b>ha a telepes egy akcióként rálépett a kapura, és csak az utána lévő körben tud rajta belépni és ezzel megjelenik a kapupár másik tagján.</b>	Telepessel használjuk az egyik kapupárt.	kötelező	feladat	Build, Control  Settler / Robot	

## **3.1 Objektum katalógus**

### **3.1.1 Asteroid**

Rajtuk tartózkodnak a telepések és a robotok. Ezen aszteroidák az aszteroida övben, vagyis a pályán helyezkednek el. Felelősségük az aszteroidának szomszédos pálya elemeinek ismerete, illetve saját aszteroidájának köpeny vastagságát.

### **3.1.2 Material**

Az aszteroidákban rejlő nyersanyagok. Többféle van belőlük: vízjég, vas, szén, urán (radioaktív) és szilícium. Felelősségük, hogy tárolja az aszteroidában található nyersanyagokat.

### **3.1.3 Settler**

A játékos által irányított karakterek. Az aszteroidákon tudnak lépkedni mindig a szomszédosokra. Több műveletet tud végezni: bányászni, fűrni, mozogni, várni, teleport kaput építeni vagy lerakni, bázist építeni vagy robotokat építeni. A Settler magánál tárolja a kibányászott nyersanyagokat, amit vissza is tud helyezni kibányászott aszteroidába. Meg tud halni, ha felrobban az aszteroida, vagy napviharba kerül.

### **3.1.4 Robot**

A robot a rendszer által irányított karakter. A szomszédos aszteroidákon lépked véletlenszerűen. A robot kettő tevékenységet tud: fűrni és lépni. Ha az aszteroida felrobban, akkor egy véletlenszerű szomszédos aszteroidára kerül.

### **3.1.5 Gate**

A Gate-k a pályákon elhelyezkedő teleport kapuk. Az aszteroidák szomszédságában állnak. Minden Gate-nek csak egy párja van. Ha a kapupár egyikébe belépnek a telepések, vagy robotok akkor a másik kapuhoz fogja őket teleportálni.

### **3.1.6 Eruption**

Bizonyos időszakonként előforduló napviharok. A napviharokat a rendszer generálja és értesíti a játékosokat, hogy X kör múlva napvihar lesz. A napvihar megöli a telepest/robotot, ha az nincs egy olyan aszteroidán, amin van magig nyúló lyuk és a magja üreges.

### **3.1.7 Game manager**

Ez az osztály irányítja a játék egészét. Tudja, hogy hol helyezkednek el és mit csinálnak a pályán az aszteroidák, a kapuk, a telepések, a robotok és az esetleges napviharok, és ezeket kézben tartja, illetve irányítja.

### **3.1.8 Main asteroid**

Ez a kezdő aszteroida, amelyen a játék kezdésekor elhelyezkednek a telepések. Ezen az aszteroidán kell felépíteni a bázist.

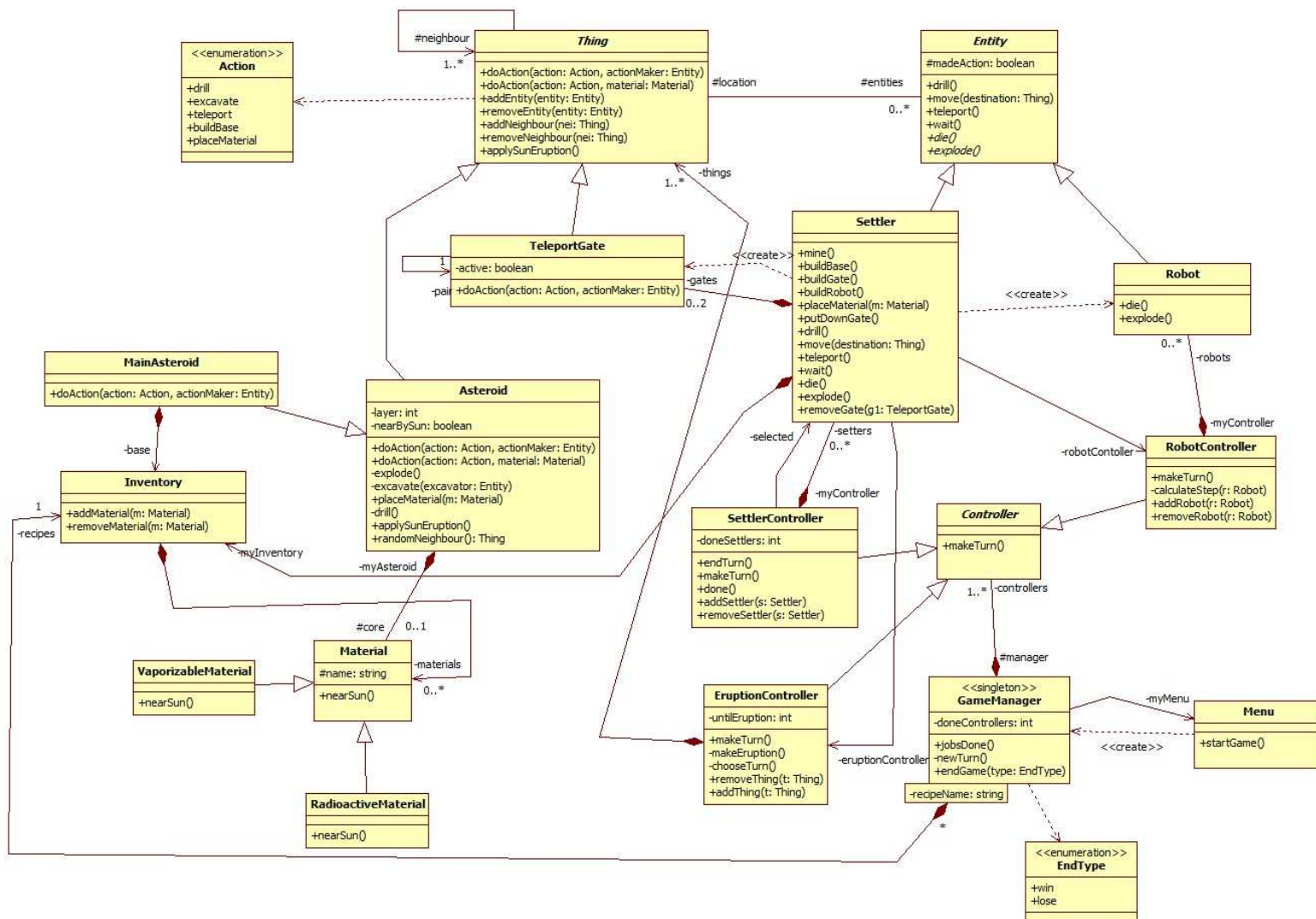
### **3.1.9 Radioactive**

Radioaktív aszteroida. Ha kibányászáskor napközben van, akkor felrobbantja az aszteroidát.

### 3.1.10 Vaporizable

Vízjég aszteroida. Ha kibányászáskor napközelen van, akkor az aszteroida magja elszublimál és eltűnik.

### 3.2 Statikus struktúra diagramok



### 3.3 Osztályok leírása

### 3.3.1 Action <<enumeration>>

- **Felelősség**

Az action fajtákat nevezi meg, a kód átláthatóságának érdekében.

- **Ősosztályok**

- **Interfészek**

- 

- **Attribútumok**

- **drill:** Fúrás akciót jelez.
- **excavate:** Kibányászás akciót jelez.
- **teleport:** Teleportálás akciót jelez.
- **buildBase:** Bázis építést jelez.
- **placeMaterial:** Anyag visszatevése akciót jelez

### 3.3.2 Asteroid

- **Felelősség**

Egy pályán lévő aszteroida, amelyet tudnak fúrni és kibányászni belőle nyersanyagot, valamint a napközelséget is tárolja.

- **Össosztályok**

Thing → Asteroid

- **Interfészek**

- 

- **Attribútumok**

- **#core: Material[0..1]:** Az aszteroidában található nyersanyag
- **layer: int:** A közetréteg aktuális vastagsága
- **nearBySun: boolean:** Tárolja, hogy napközelen van-e az aszteroida

- **Metódusok**

- **void applySunEruption():** Megöl minden telepest és robotot, akik nincsenek elbújva egy üreges aszteroidában.
- **void doAction(Action action, Entity actionMaker):** Egy entitás olyan tevékenysége az aszteroidán, amelynél fontos az, hogy ki csinálja azt a tevékenységet.
- **void doAction(Action action, Material materia):** Egy entitás olyan tevékenysége az aszteroidán, amelynél egy nála lévő nyersanyagot használ fel.
- **void placeMaterial(Material m):** Lerak egy materialt.
- **Thing randomNeighbour():** Az aszteroida egy random szomszédját adja vissza.

### 3.3.3 Controller

- **Felelősség**

Az irányítható dolgok irányításáért felelős.

- **Össosztályok**

-

- **Interfészek**

- 

- **Attribútumok**

- **#manager: GameManager:** a játék irányításáért felelős

- **Metódusok**

- **void makeTurn():** Elvégzi a műveleteket egy adott körben.

### 3.3.4 EndType <<enumeration>>

- **Felelősség**

A játék végének típusát lehet vele jelezni.

- **Ősosztályok**

- 

- **Interfészek**

- 

- **Attribútumok**

- **lose:** Vereséget jelez
  - **win:** Nyerést jelez

### 3.3.5 Entity

- **Felelősség**

Leszármazottjai a Robot és a Settler osztály. Ezen osztály tárolja el a karaktereket és innen adja tovább a Thing osztálynak a karaktereket és a róluk szükséges információkat.

- **Ősosztályok**

- 

- **Interfészek**

- 

- **Attribútumok**

- **#location: Thing:** Az objektum ismeri azt a helyet (aszteroida, kapu), amin tartózkodik.
  - **#madeAction: boolean:** Meghatározza, hogy csinált-e valamilyen akciót a karakter. Minden lenti akció végrehajtása esetén ezt az értéket true-ra állítja.

- **Metódusok**

- **void drill():** A karakter e függvény meghívásával megfúrja az aszteroidát.
  - **void move(Thing destination):** A karakterek mozognak a különböző „thingek”/dolgok (aszteroidák, kapuk) között.

- **void teleport():** A függvény meghívásával a karakter teleportál a kiválasztott kapuk között.
- **void wait():** A karakter az adott körben „nem csinál semmit”, vagyis várakozik az aktuális helyén.
- **abstract void die():** A karakter halálakor meghívódó függvény, vagyis akkor hívódik meg, amikor a karakter valamilyen okból (pl. napvihar) meghal.
- **abstract void explode():** A radioaktív robbanásnál bekövetkező függvényhívás.

### 3.3.6 EruptionController

- **Felelősség**

A Napkitörések irányításáért felelős. Eldönti és jelzi, hogy mikor lesz napkitörés.

- **Ősosztályok**

Controller → EruptionController

- **Interfészek**

-

- **Attribútumok**

- **untilEruption: int:** A körök számát tárolja a következő napkitörésig.
- **things: Thing[1..\*]:** A játékban lévő Thingeket tárolja.

- **Metódusok**

- **void addThing(Thing t):** Hozzáad egy thing-et.
- **void makeTurn():** Eldönti, hogy legyen-e napkitörés a közeljövőben.
- **void removeThing(Thing t):** Kitöröl egy egy thing-et.

### 3.3.7 GameManager

- **Felelősség**

A játék fő rugója, ő kontrolálja a köröket. Minden körben a Controller osztályoknak ad utasítást, hogy végezze el a körét. A kört csak akkor kezd, ha már mindenki befejezte az aktuális körét. Emellett statikus változójaként tárolja a játék során felmerülő recepteket, így egységes, bárki elérheti őket.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **doneControllers:** A köröket befejező Controller osztályok száma.
- **controllers:** A GameManager által felügyelt Controller osztályok.
- **recipes:** A különböző építésekhez szükséges receptek.

- **myMenu:** A Menu, ami a GameManagerert létrehozta.
- **Metódusok**
  - **void endGame(EndType type):** A kapott EndType típusú argumentumtól függően, befejezi a játékot.
  - **void jobsDone():** Növeli eggyel a doneControllers értékét, ezt a metódust hívják meg a Controllerek, amint végeztek a körükkel.

### 3.3.8 Inventory

- **Felelősség**

Materiálok tárolásáért felelős osztály.

- **Össztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **#name: string:** Az anyag neve.
- **selected: Settler:** Az adott settler, aki módosítja az inventory tartalmát. Ez lehet a sajátja vagy a bázisé.
- **materials: Material:** Az anyagokat tároló konténer, amit hashmappal oldunk meg.

- **Metódusok**

- **void addMaterial(Material m):** Egy materialt hozzáad a listához.
- **void removeMaterial(Material m):** Elvesz egy materialt a listából.

### 3.3.9 MainAsteroid

- **Felelősség**

Ez a kezdő aszteroida, ahova a bázist kell megépíteni. Ha a telepések megépítik rajta a bázist, akkor megnyerik a játékot.

- **Össztályok**

Thing → Asteroid → MainAsteroid

- **Interfészek**

-

- **Attribútumok**

- **base: Inventory:** Az aszteroidán a bázis megépítéséhez eddig összegyűjtött nyersanyagok



- **Metódusok**
  - **void doAction(Action action, Entity actionMaker):** A telepes által a kezdő aszteroidán elvégezhető tevékenységek.

### 3.3.10 Material

- **Felelősség**

A nyersanyagok tárolásáért felelős osztály. Tudja, hogy kibányászáskor milyen messze van a naptól. Ő valósítja meg az aszteroida magját.
- **Össztályok**
  -
- **Interfészek**
  -
- **Attribútumok**
  - **#name: string:** A nyersanyag megnevezése
- **Metódusok**
  - **void nearSun():** Ha aszteroida napközelsébe kerül akkor hívódik meg. A különböző típusú aszteroidákon más hatást fejt ki.

### 3.3.11 Menu

- **Felelősség**

A játék paramétereinek bekérése/beállítása és a GameManager osztály létrehozása, ezzel együtt a játék elindítása.
- **Össztályok**
  -
- **Interfészek**
  -
- **Attribútumok**
  -
- **Metódusok**
  - **void startGame():** Elindítja a játékot és beállítja a paramétereit.

### 3.3.12 RadioactiveMaterial

- **Felelősség**

A radioaktív anyagot valósítja meg. Ha napközelen van, amikor kibányásszák, akkor felrobban és megöli a telepeseket, és a robotokat átlöki egy véletlenszerű szomszédos aszteroidára.

- **Ősosztályok**

Material → RadioactiveMaterial

- **Interfészek**

-

- **Attribútumok**

- **#name: string:** Az anyag neve.

- **Metódusok**

- **void nearSun():** Ha az őt tartalmazó aszteroida napközelen van, amikor kibányásszák, akkor felrobban és megöli a rajta lévő telepeseket, valamint a robotokat ellöki egy véletlenszerű szomszédos aszteroidára. A robbanáskor megsemmisül az őt tartalmazó aszteroida is.

### 3.3.13 Robot

- **Felelősség**

A telepes (illetve ezzel együtt a játékos) munkáját segíti.

- **Ősosztályok**

Entity → Robot

- **Interfészek**

-

- **Attribútumok**

- **myController: RobotController:** Az adott robot controller osztálya, ami vezérli őt.

- **Metódusok**

- **void explode():** A radioaktív robbanásnál hívódik meg, azonban itt nem hal meg a robot, hanem csak arrébb kerül egy véletlenszerű szomszédos aszteroidára, vagyis meghívja az ősosztály move függvényét, így át tud menni egy véletlenszerű szomszédos aszteroidára.
- **void die():** Ez a függvény jelzi, hogy a robot meghal/tönkremegy.

### 3.3.14 RobotController

- **Felelősség**

Az autonóm robotokat irányítja. Minden kör elején egy műveletet kiválaszt minden robotnak a játékban.

- **Össztályok**

Controller → RobotController

- **Interfészek**

-

- **Attribútumok**

- **robots: Robot[0..\*]:** Az általa irányítandó robotok

- **Metódusok**

- **void addRobot(Robot r):** Hozzáad egy robotot.
- **void makeTurn():** Elvégezteti a robottal a kisorsolt műveletet.
- **void removeRobot(Robot r):** Kitöröl egy robotot.

### 3.3.15 Settler

- **Felelősség**

A telepés dolgozik a játékosnak, próbálja összegyűjteni a nyersanyagokat és abból egy űrbázist épít, hogy megnyerje a játékot. A telepés hozza létre (nyersanyagokból) a teleport kapukat és a robotokat.

- **Össztályok**

Entity → Settler

- **Interfészek**

-

- **Attribútumok**

- **myController: SettlerController:** Ezzel az attribútummal jelzi a controller osztályának, ha meghal a telepés, illetve ezzel jelzi, ha véget ért számára a kör.
- **robotController: RobotController:** Ismeri a robot controller osztályát, mivel így tudja jelezni neki, hogy ha felépít egy új robotot, akkor azt adja hozzá a robot controller osztályához.
- **eruptionController: EruptionController:** Itt tárolódnak a Thing osztály példányai és ide kerülnek a Settler által létrehozott objektumok is.
- **myInventory: Inventory:** A telepés által szükséges nyersanyag.
- **gates: TeleportGate[0..2]:** A telepés által létrehozandó kapukat jelöli.

- **Metódusok**

- **void buildBase():** A kezdő aszteroidán építi a bázist (egy körben értelemsszerűen csak egy új elemmel).

- **void buildGate():** A függvény meghívásával épít egy teleport kapupárt a szükséges nyersanyagokból a telepés.
- **void buildRobot():** A függvény meghívásával épít egy robotot a szükséges nyersanyagokból a telepés.
- **void die():** Ez a függvény azt jelzi, hogy a telepés meghal.
- **void drill():** A telepés e függvény meghívásával megfúrja az aszteroidát és ezt jelzi a SettlerController osztálynak is.
- **void explode():** Ez a függvény azt jelzi, hogy a radioaktív robbanásnál meghívja a Settler osztály die függvényét, hiszen radioaktív robbanás esetén a telepés meghal.
- **void mine():** A telepés bányászik az adott aszteroidán.
- **void move(Thing destination):** A telepések mozognak a különböző „thingek”/dolgok (aszteroidák, kapuk) között és ezt jelzi a SettlerController osztálynak.
- **void placeMaterial(Material m):** Az adott nyersanyagot visszateszi/elhelyezi az adott magig nyúló lyukas és üreges aszteroidán.
- **void putDownGate():** Lerakja a kaput.
- **void removeGate(TeleportGate g1):** Eltávolítja a kaput.
- **void teleport():** A függvény meghívásával a telepés teleportál a kiválasztott kapuk között és ezt jelzi a SettlerController osztálynak.
- **void wait():** A telepés az adott körben „nem csinál semmit”, vagyis várakozik az aktuális helyén és ezt jelzi a SettlerController osztálynak.

### 3.3.16 SettlerController

- **Felelősség**

A telepések irányításáért felelős, feldolgozza a felhasználótól kapott utasításokat.

- **Össztályok**

Controller → SettlerController

- **Interfészek**

-

- **Attribútumok**

- **doneSettlers: int:** Azon telepések számát méri, akik már kaptak feladatot az jelenlegi körben.
- **settlers: Settler[0..\*]:** A játékban található telepések.
- **selected: Settler:** Az aktuálisan irányított telepés.

- **Metódusok**

- **void addSettler(Settler s):** Hozzáad egy telepest.
- **void done():** Jelzi, hogy minden telepés befejezte a kört.
- **void endTurn():** Jelzi, hogyha az adott telepés befejezte a kört.
- **void makeTurn():** Elvégzi a felhasználó által kiválasztott műveletet.
- **void removeSettler(Settler s):** Kitöröl egy telepest.

### 3.3.17 TeleportGate

- **Felelősség**

A telepesek tudják építeni a teleport kapukat. Mindegyiknek van egy párja a pálya másik pontján, amire át tudnak mozogni 1 lépésben akkor is, ha nem szomszédosak.

- **Ősosztályok**

Thing → TeleportGate

- **Interfészek**

-

- **Attribútumok**

- **pair: TeleportGate:** A teleportkapu párja, ahova át tud mozogni az entitás.
- **active: boolean:** Azt jelzi, hogy lerakta-e már a párját is a telepese a pályára, tehát használható-e már.

- **Metódusok**

- **void doAction(Action action, Entity actionMaker):** Az entitás által a kapun végrehajtott tevékenység(teleportálás).

### 3.3.18 Thing

- **Felelősség**

Azokat az objektumokat reprezentálja, amelyekre az entitások (telepese, robot) mozogni tudnak, és tevékenységeket végrehajtani rajta.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **neighbour: Thing[1..\*]:** A vele szomszédos objektumok, amelyekre átléphet a telepese vagy a robot.
- **entities: Entity[0..\*]:** A rajta lévő entitások.

- **Metódusok**

- **void addEntity (Entity entity):** Egy entitást hozzáad az objektumon tartózkodókhoz.
- **void addNeighbour (Thing nei):** Hozzáad egy objektumot ezen objektum szomszédaihoz.
- **void applySunEruption():** Megöl minden rajta álló telepest és robotot.
- **void doAction(Action action, Entity actionMaker):** Egy entitás olyan tevékenysége az objektumon, amelynél fontos az, hogy ki csinálja azt a tevékenységet.
- **void doAction(Action action, Material material):** Egy entitás olyan tevékenysége az objektumon, amelynél egy nála lévő nyersanyagot használ fel.

- **void removeEntity (Entity entity):** Egy entitást eltávolít az objektumról
- **void removeNeighbour (Thing nei):** Eltávolít egy objektumot a szomszédok közül.

### 3.3.19 VaporizableMaterial

- **Felelősség**

Az elszublimálódó anyagot valósítja meg. Mikor napközelen van az őt tartalmazó aszteroida, akkor eltűnik és a játékos nem kap semmit. Utána üres lesz az aszteroida magja.

- **Ősosztályok**

Material → VaporizableMaterial

- **Interfészek**

-

- **Attribútumok**

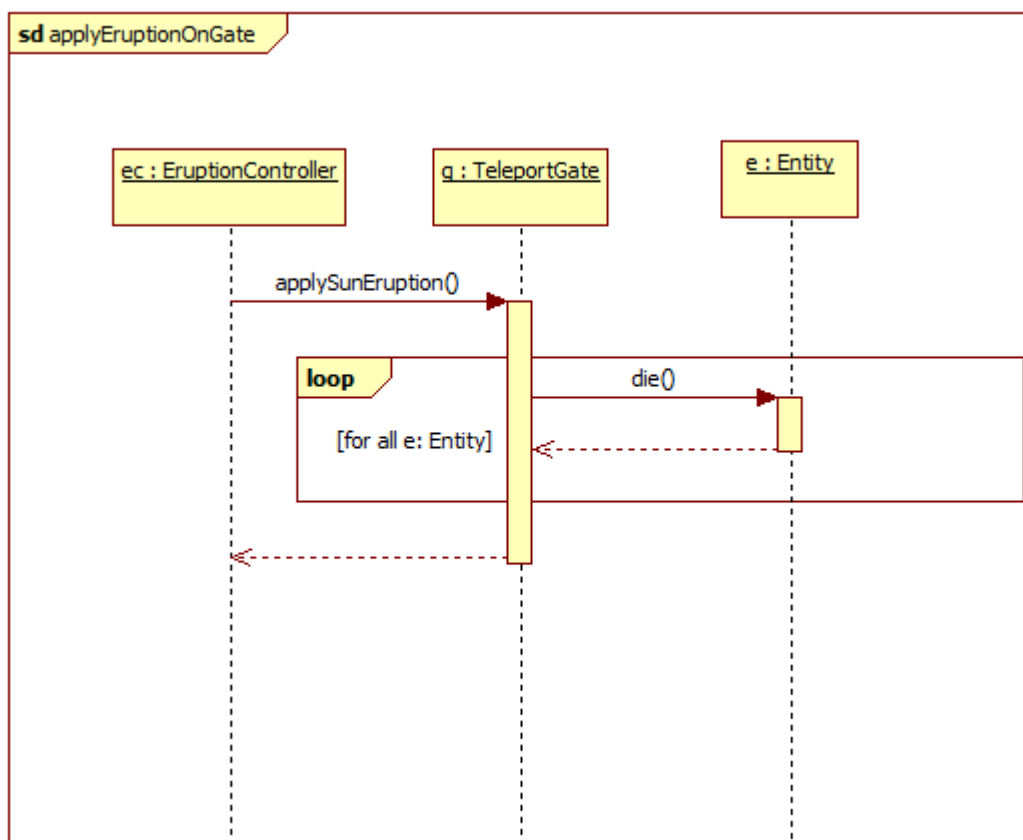
- **#name: string:** Az anyag neve.

- **Metódusok**

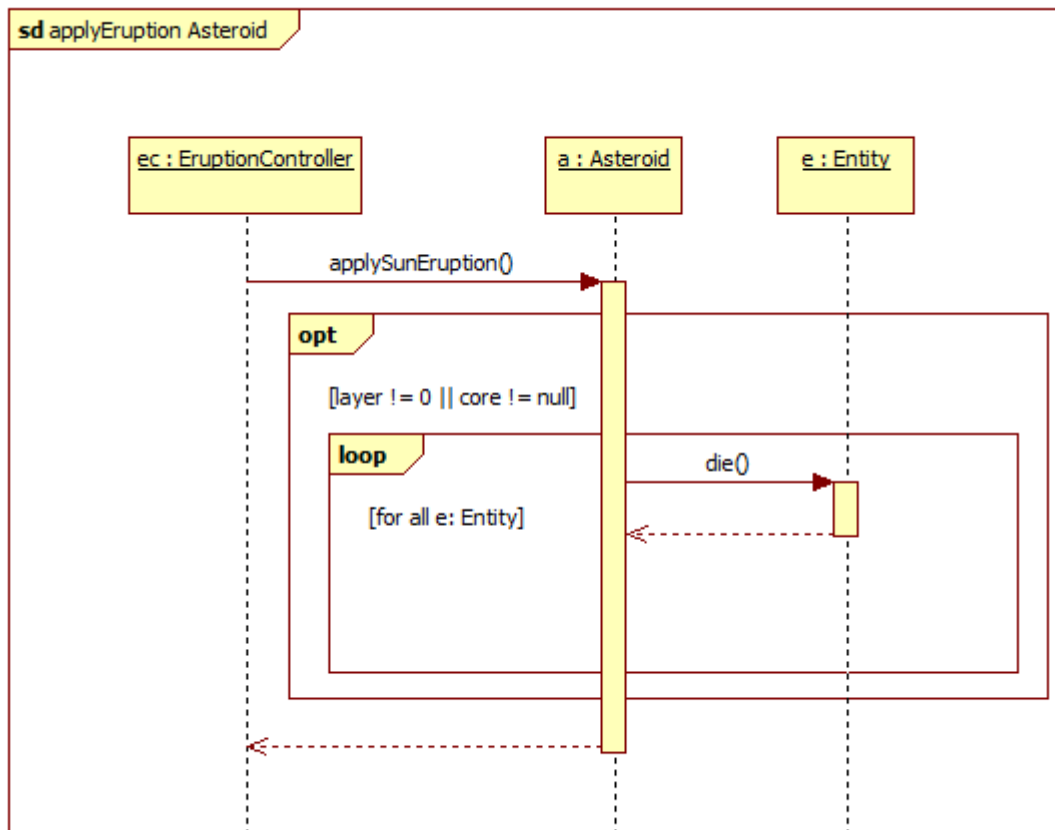
- **void nearSun():** Ha az őt tartalmazó aszteroida napközelen van, amikor kibányásszák, akkor elszublimálódik, vagyis eltűnik a nyersanyag és a játékos nem kap semmit, majd kicseréli magát üres Materialra.

### 3.4 Szekvencia diagramok

#### 3.4.1 applyEruptionOnGate

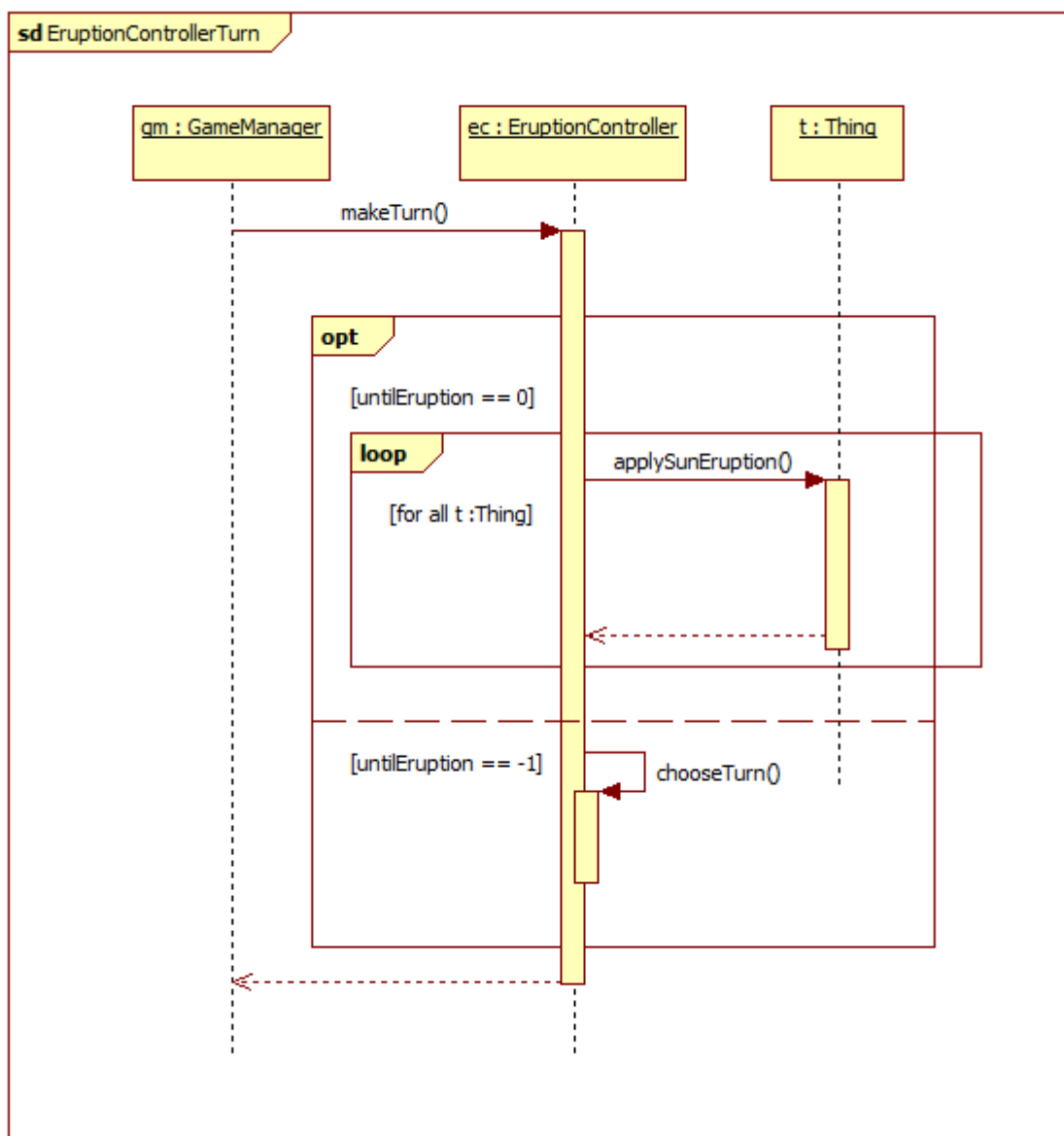


### 3.4.2 applyEruption Asteroid

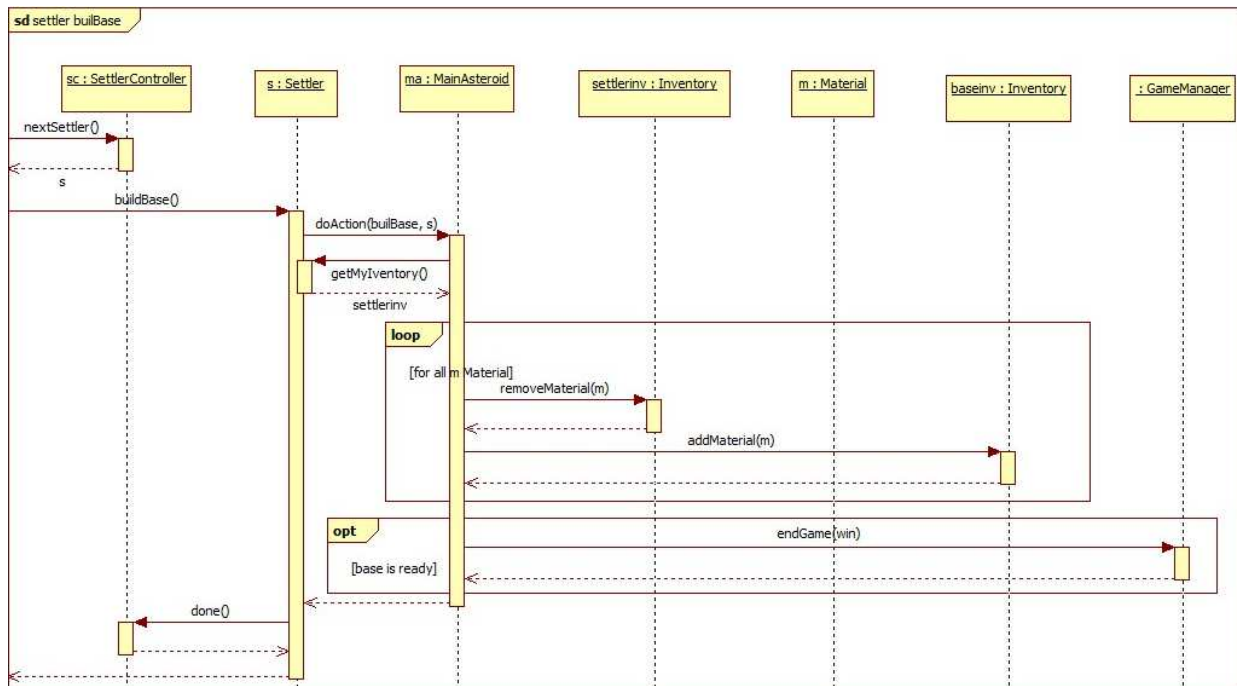




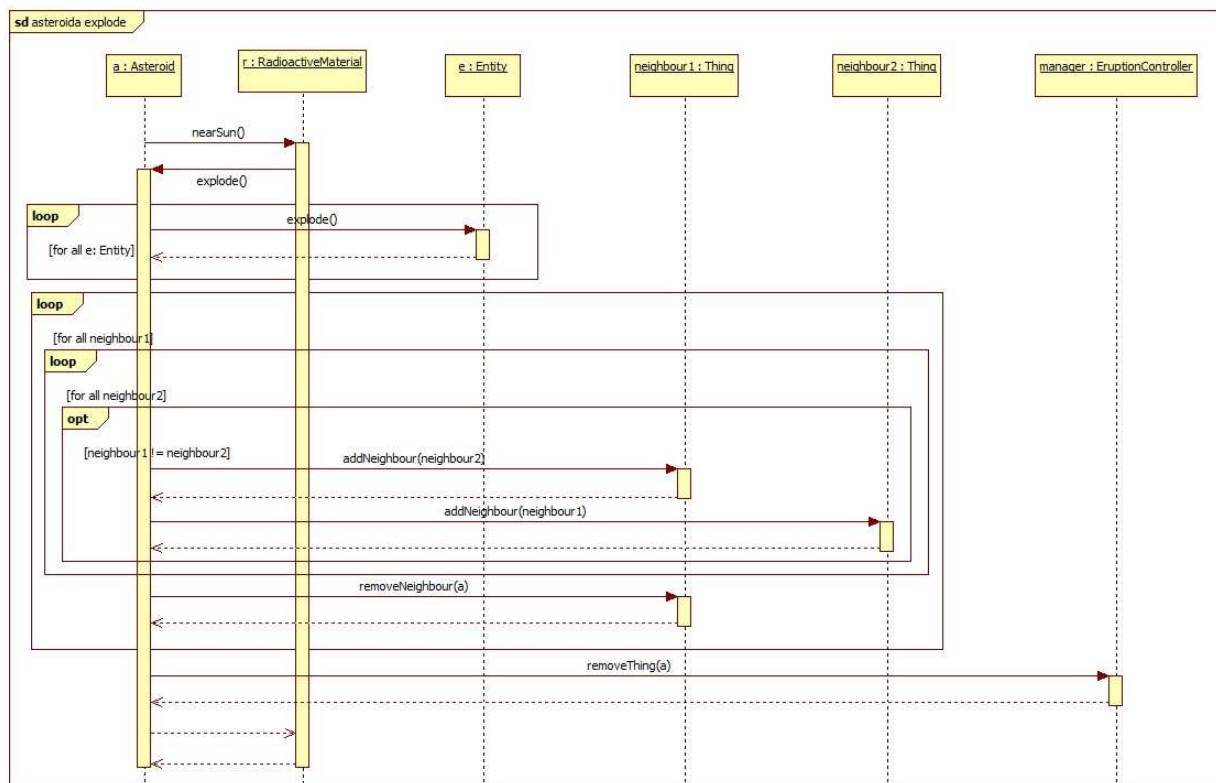
### 3.4.3 EruptionControllerTurn



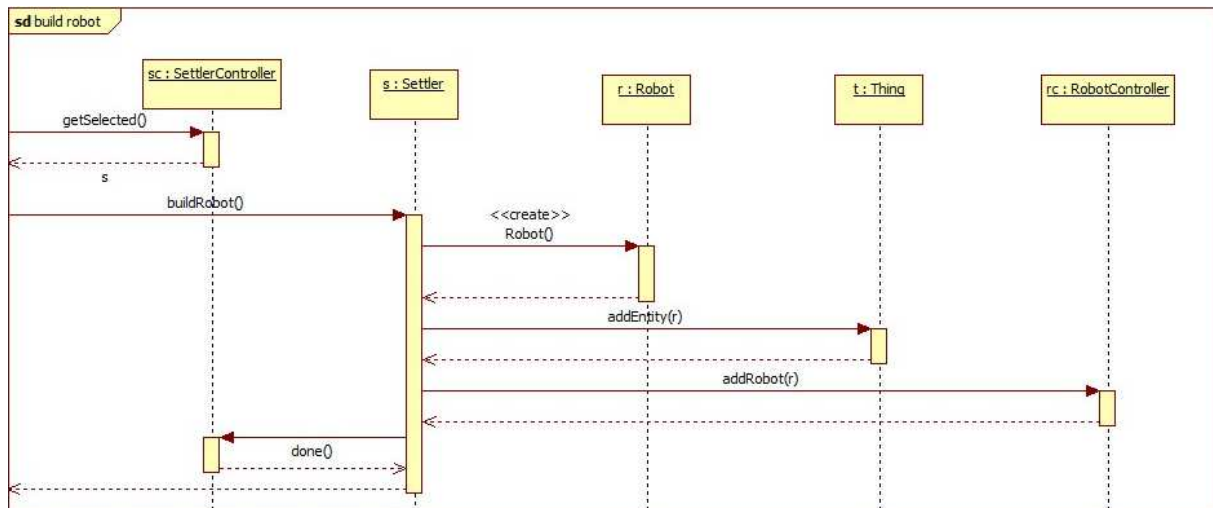
### 3.4.4 settler buildbase



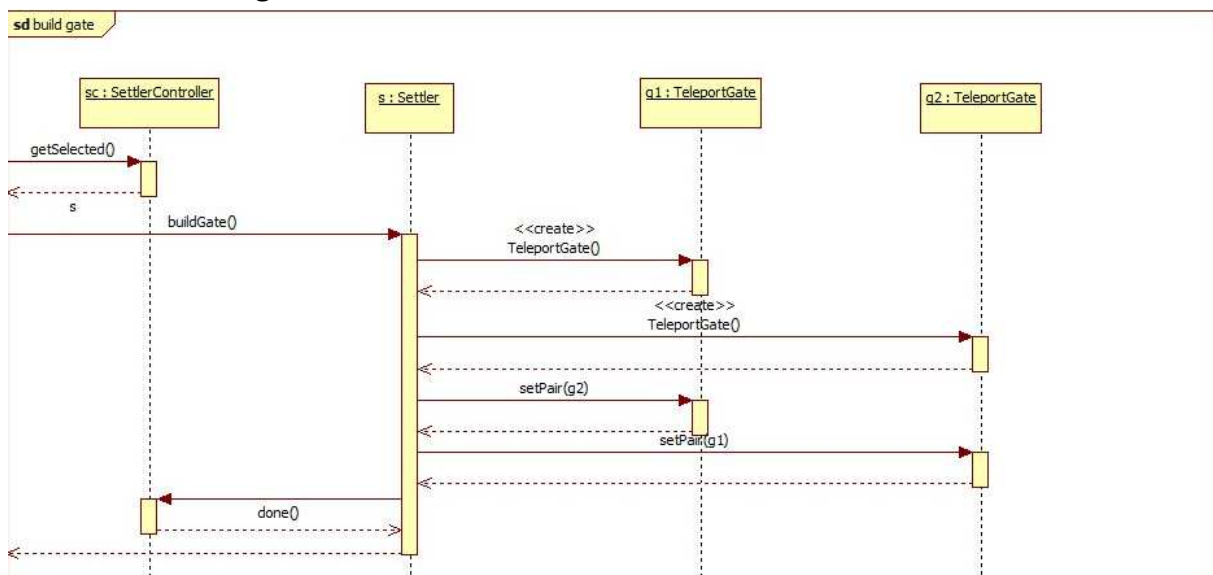
### 3.4.5 asteroida explode



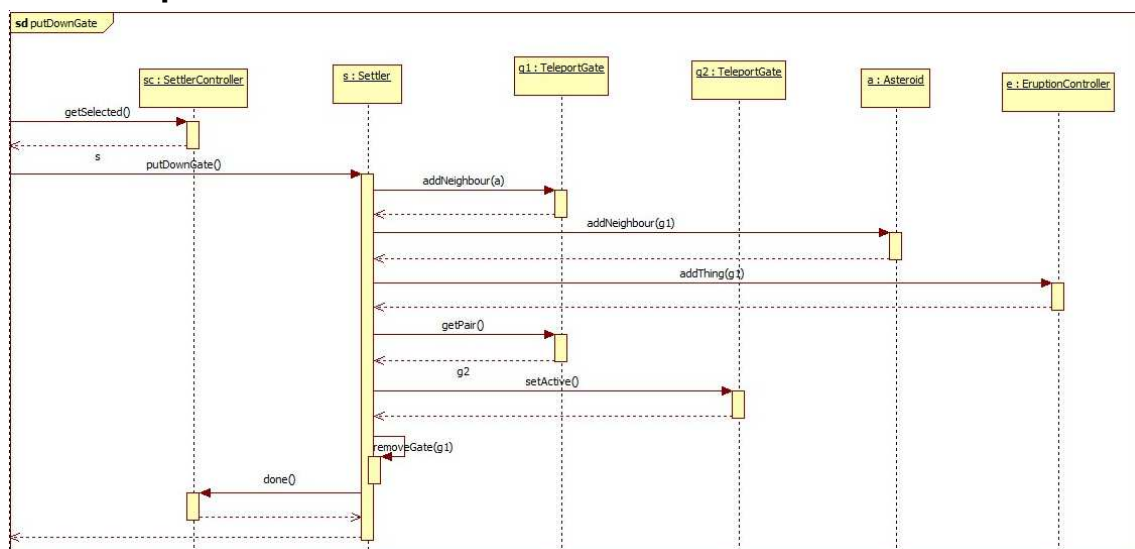
### 3.4.6 build robot



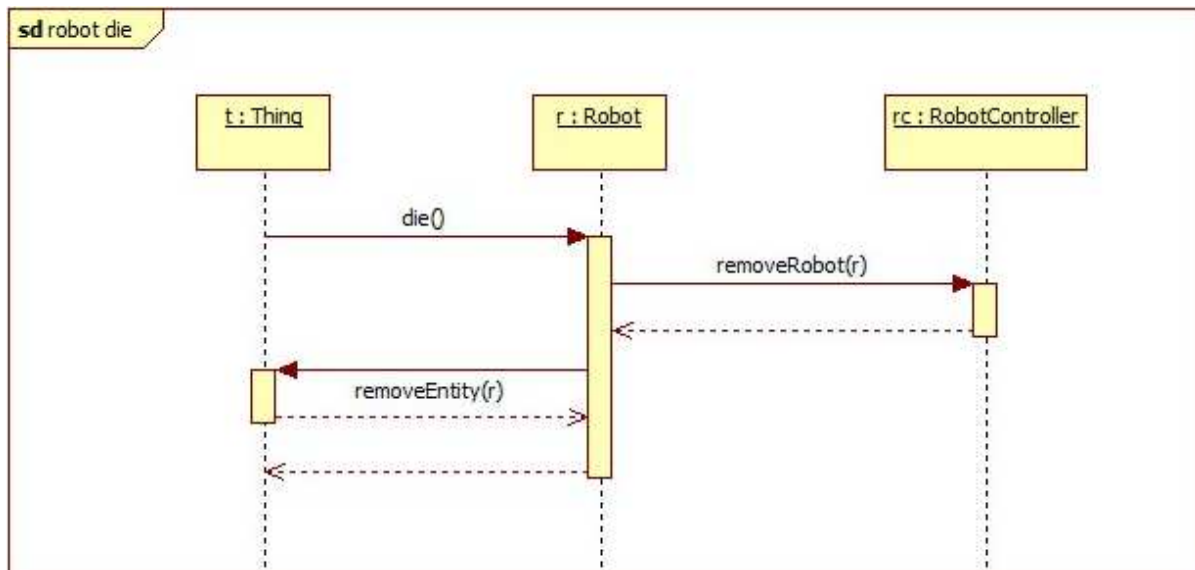
### 3.4.7 build gate



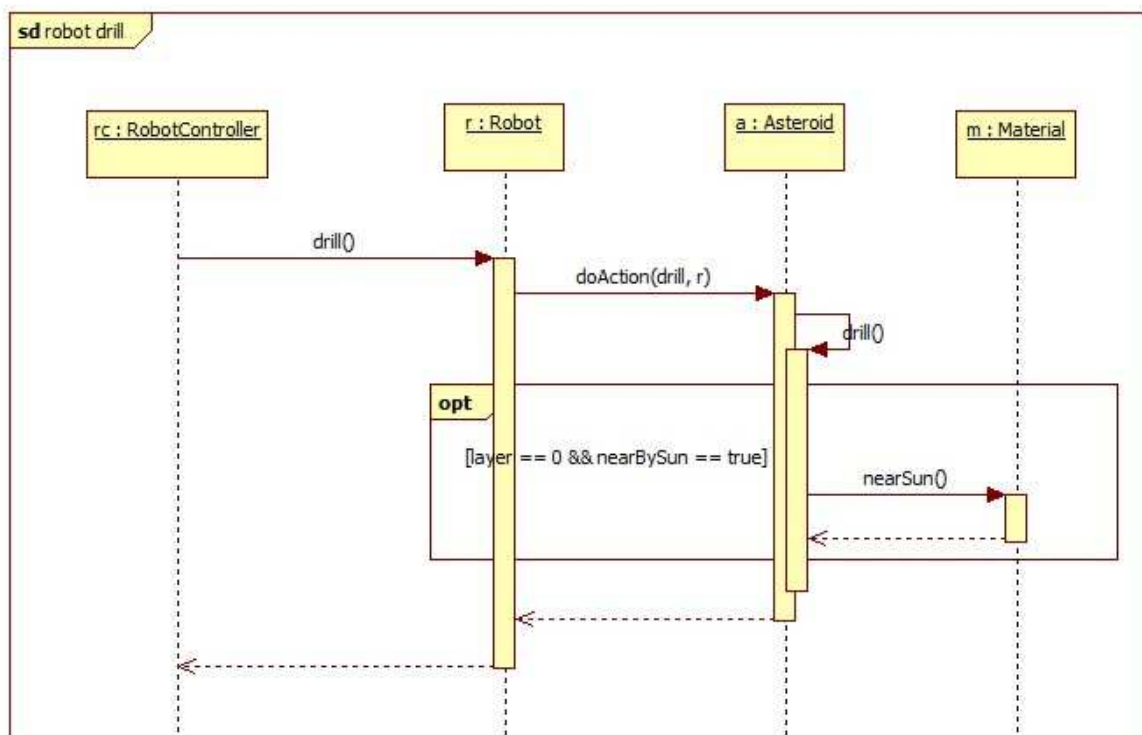
### 3.4.8 putDownGate



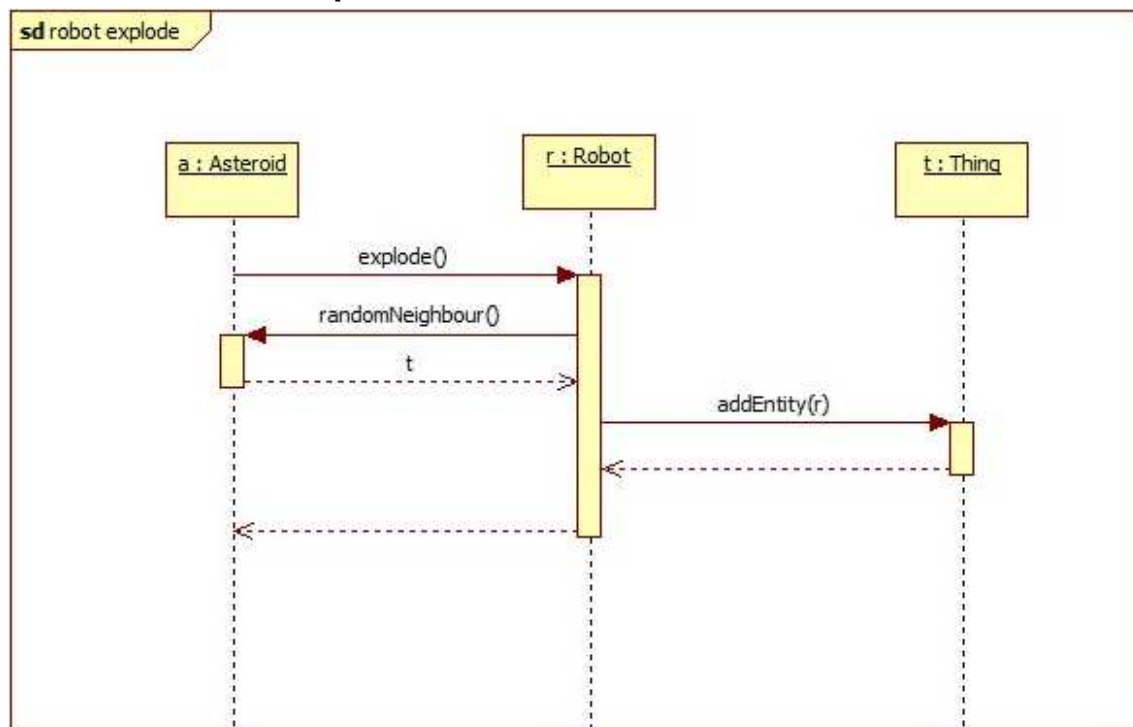
## 3.4.9 robot die



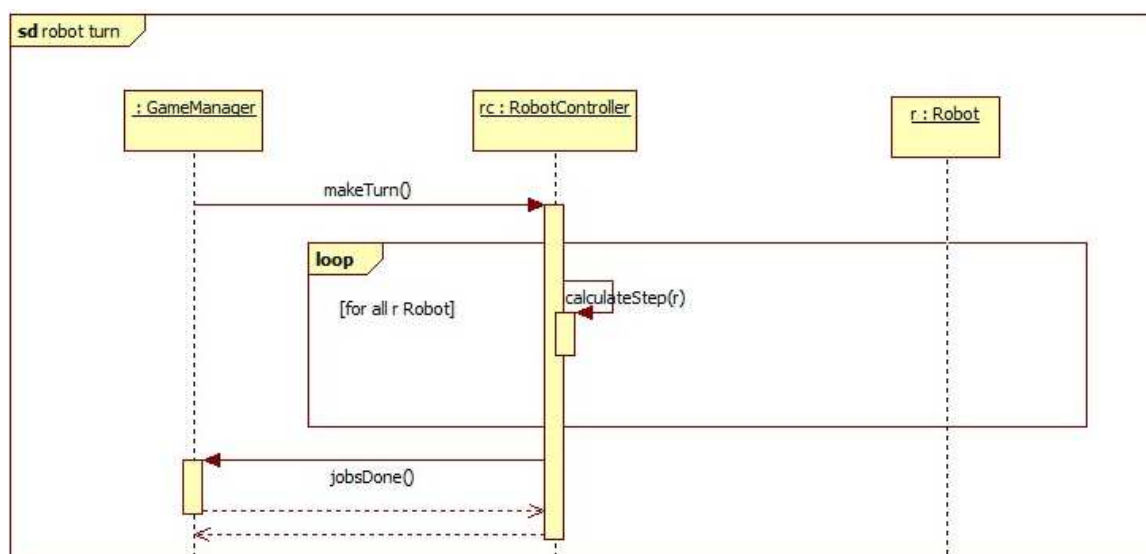
## 3.4.10 robot drill



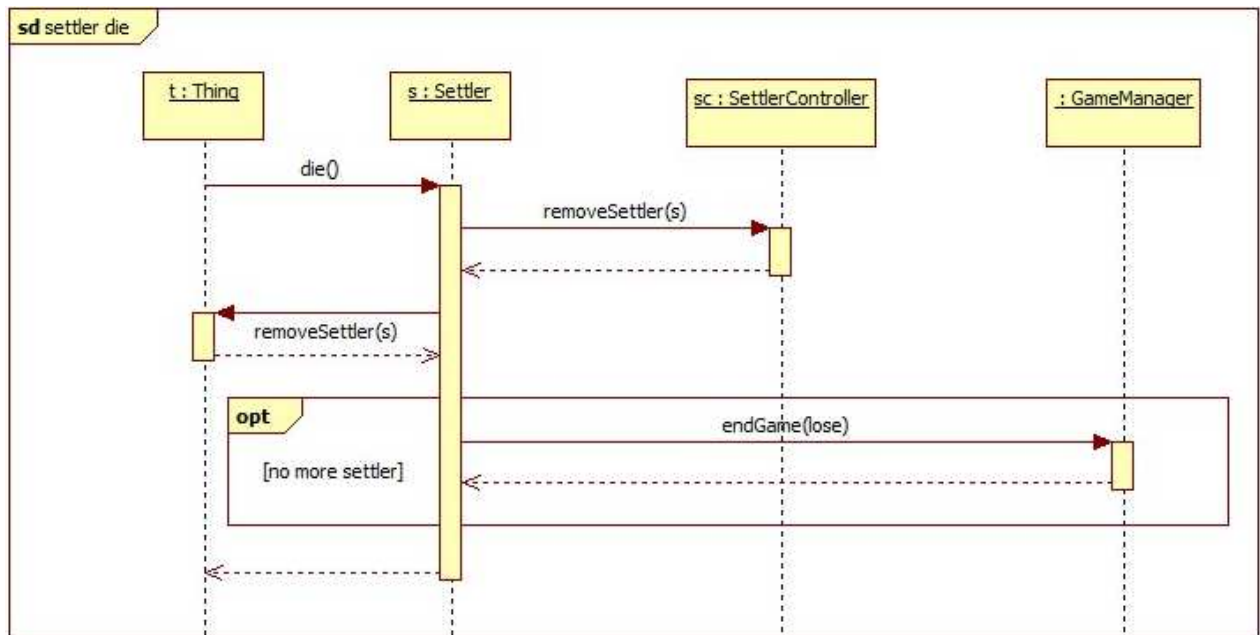
### 3.4.11 robot explode



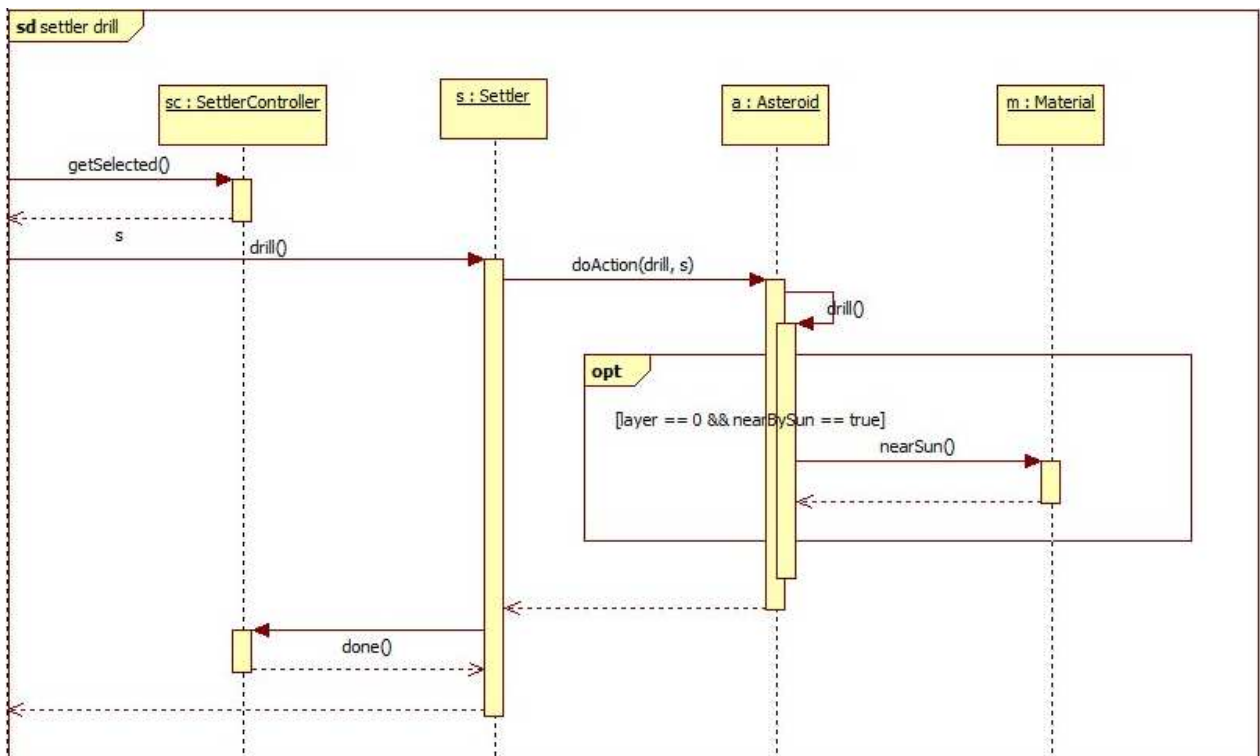
### 3.4.12 robot turn



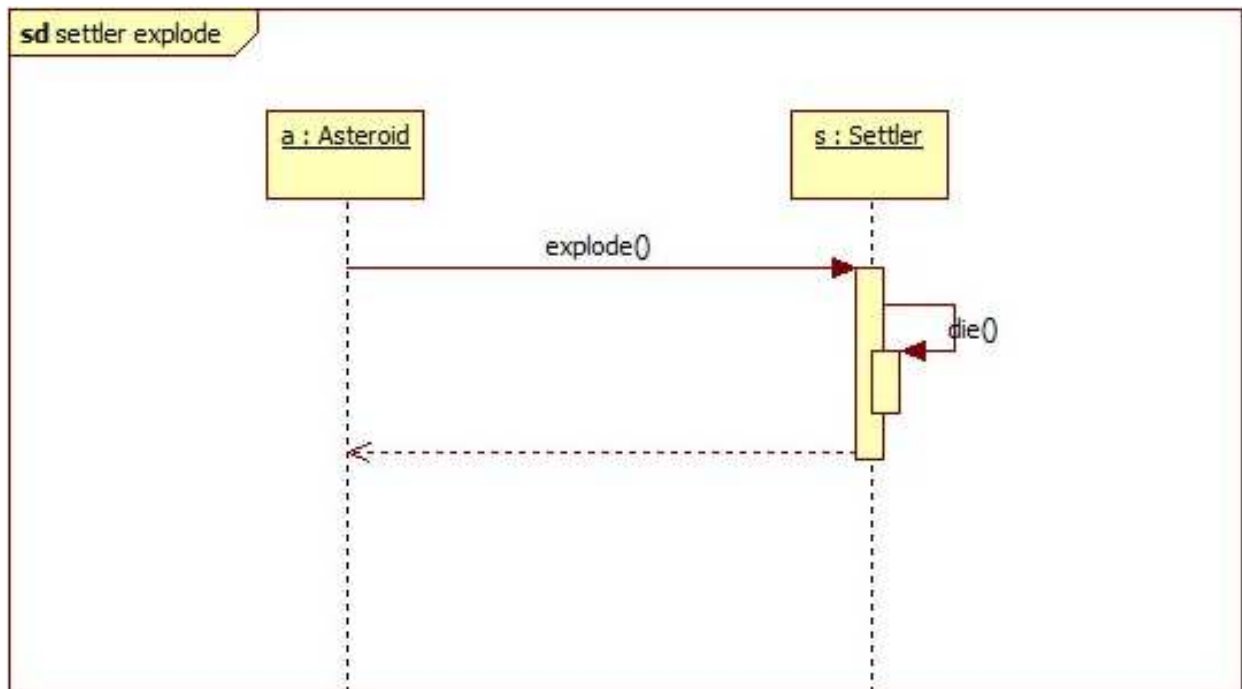
## 3.4.13 settler die



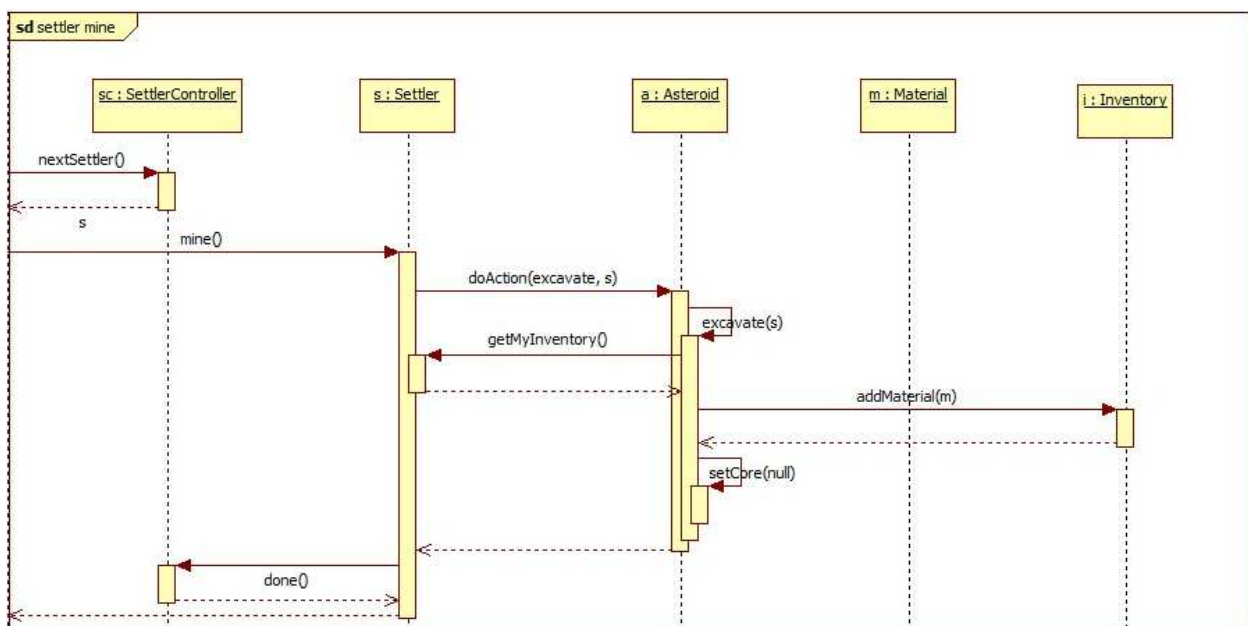
## 3.4.14 settler drill



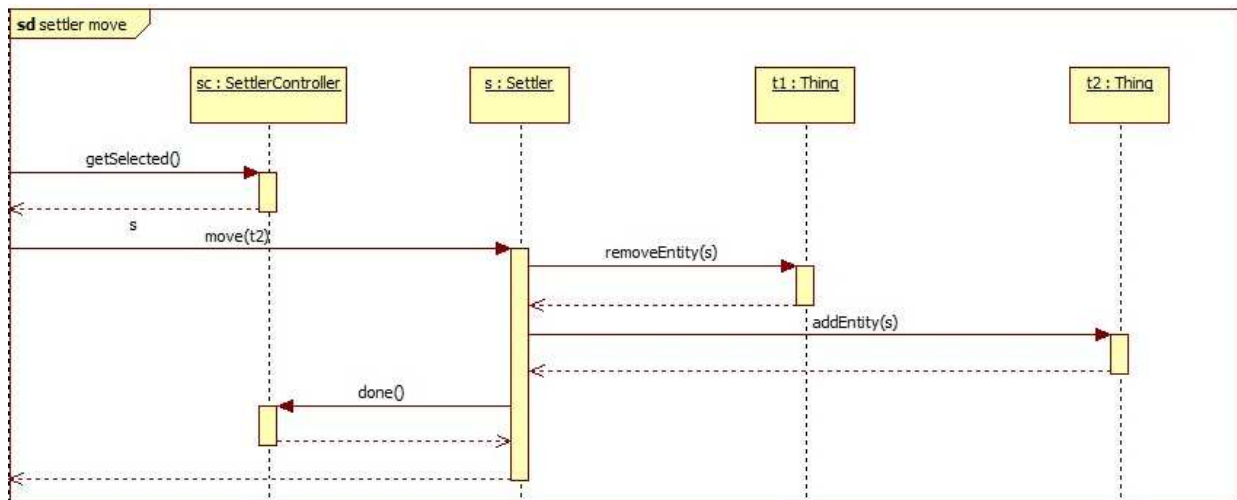
## 3.4.15 settler explode



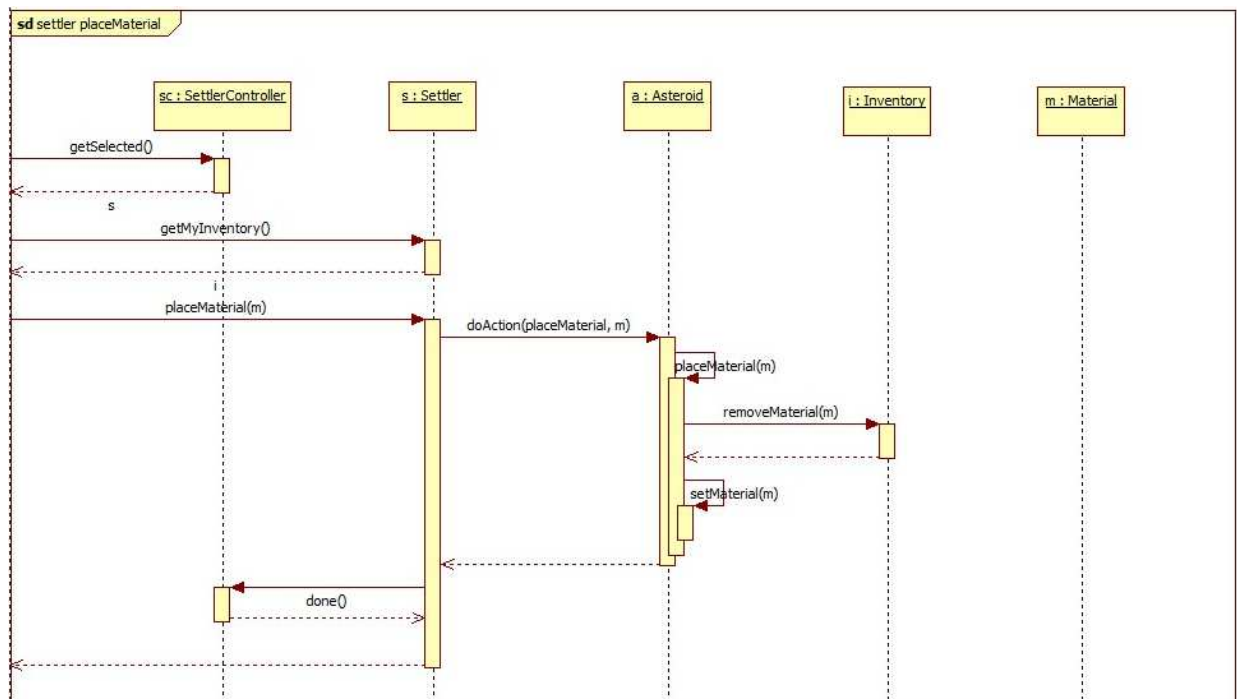
## 3.4.16 settler mine



## 3.4.17 settler move

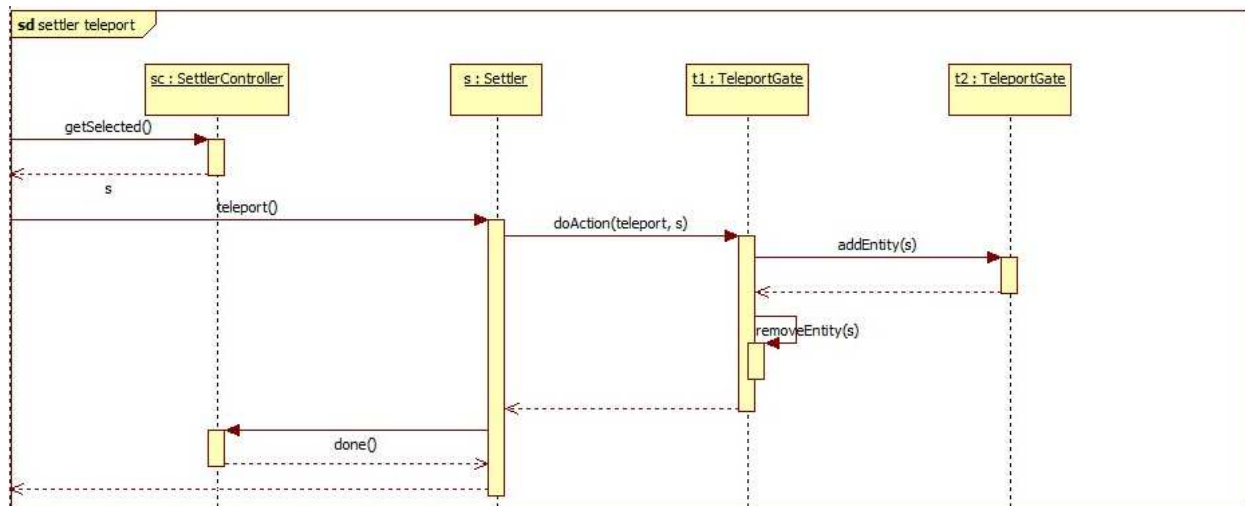


## 3.4.18 settler placeMaterial

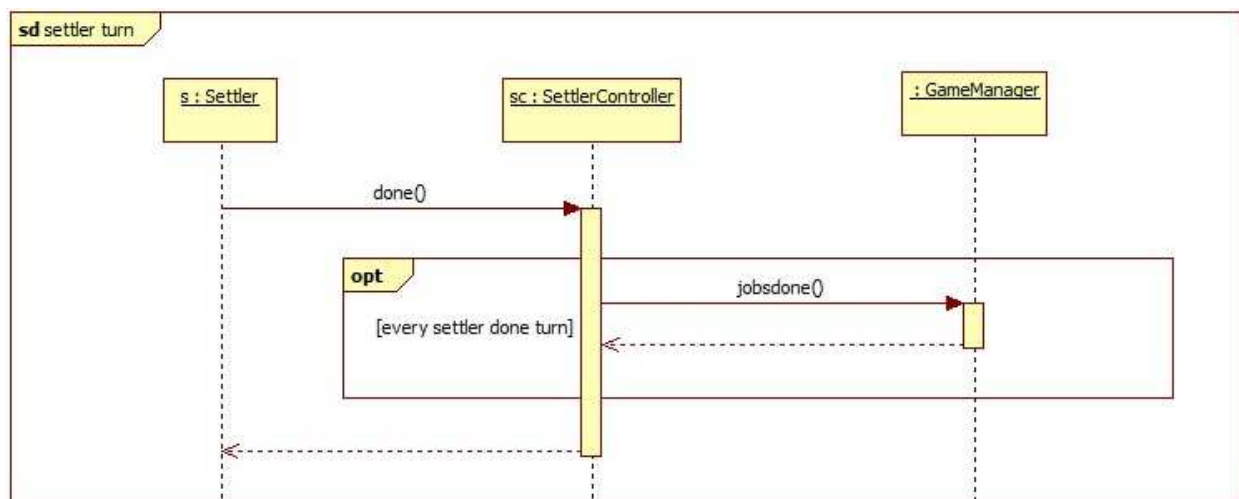


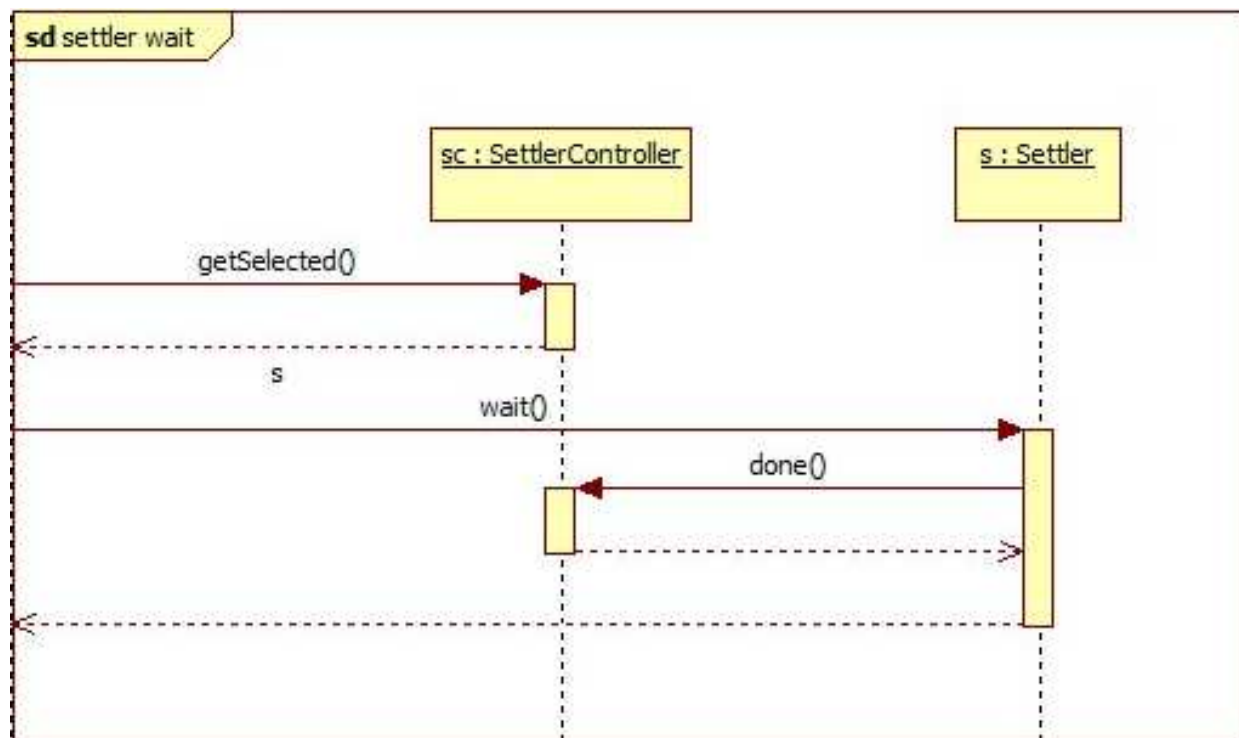
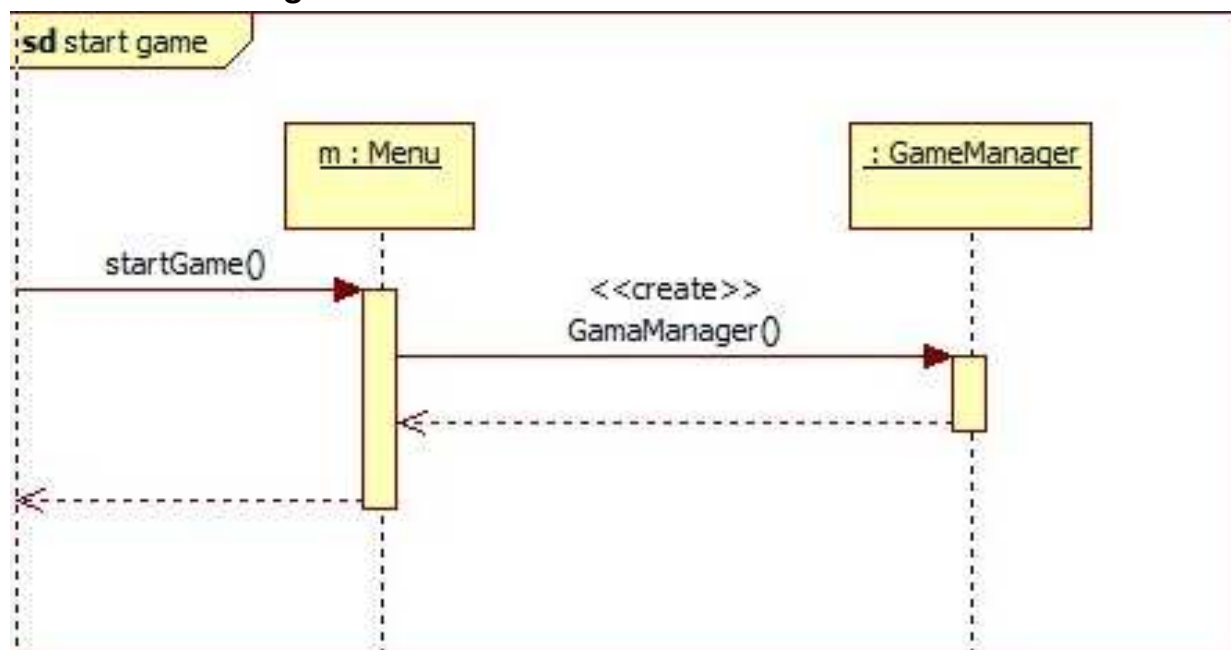


### 3.4.19 settler teleport

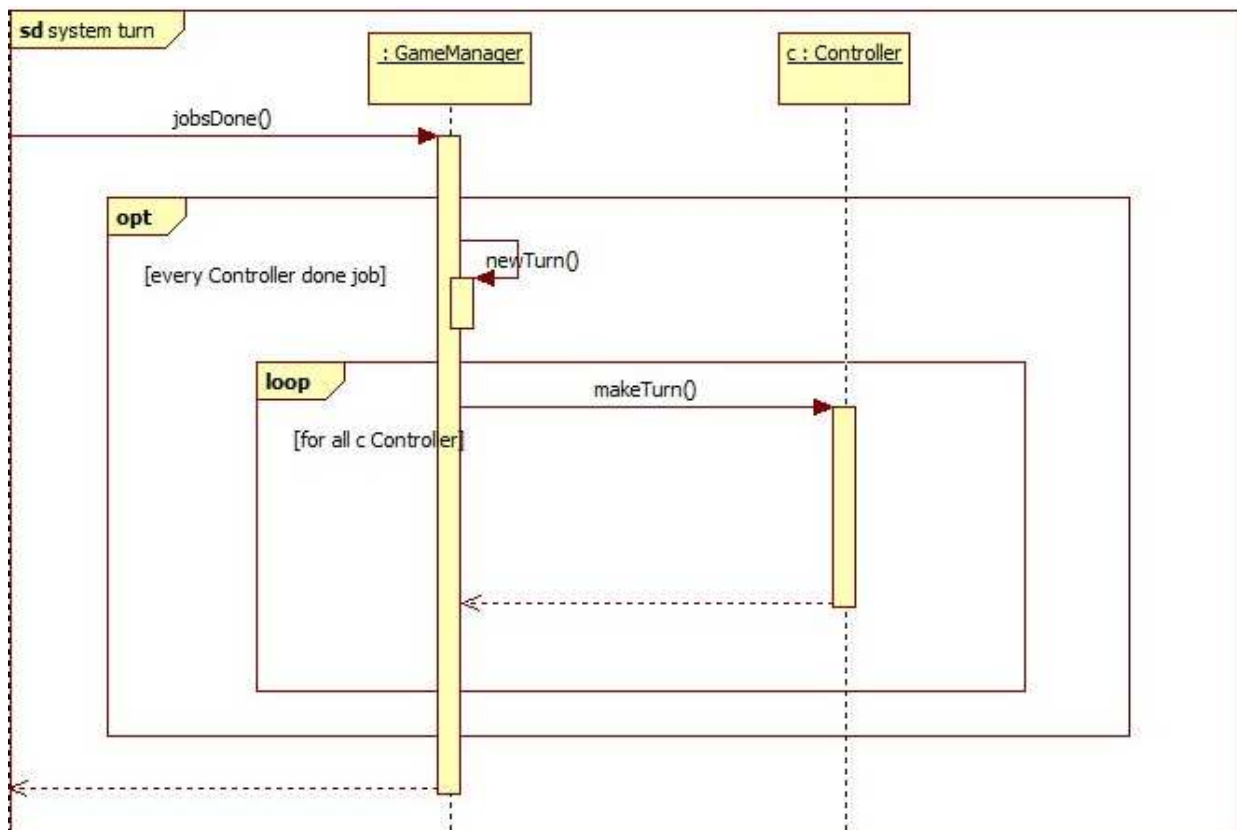


### 3.4.20 settler turn

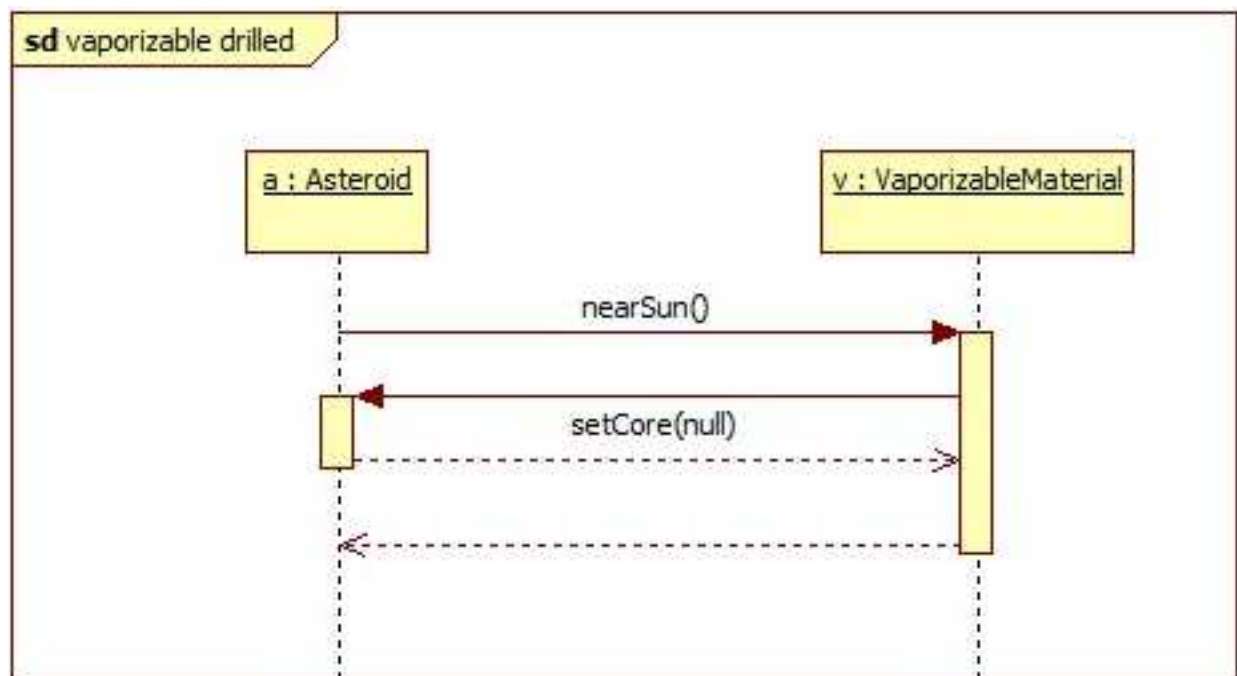


**3.4.21 settler wait****3.4.22 start game**

## 3.4.23 system turn



## 3.4.24 vaporizable drilled



### 3.5 Napló

Kezdet	Időtartam	Résztevők	Leírás
2021.02.25. 13:00	0,75 óra	Czanik	Tevékenység: Objektum-katalógus 1-5. pontjának megírása.
2021.02.25. 14:00	0,75 óra	Novák	Tevékenység: Objektum-katalógus 6-10. pontjának megírása
2021.02.25 15:00	5 óra	Czanik Nagy Novák Szabó Tokovics	Értekezlet. (Osztály diagram alapját közösen megcsináltuk, majd kiosztottuk az osztályokat egyéni kidolgozásra.) Döntés: Szabó csinálja az Action, EndType enumokat, Menu és GameManager osztályokat. Döntés: Novák csinálja az Entity osztályt és annak leszármazottjait. Döntés: Czanik csinálja a Material osztályt és annak leszármazottjait és az Inventory osztályt. Döntés: Tokovics csinálja a Thing osztályt és annak leszármazottjait, emellett a MainAsteroid osztályt is. Döntés: Nagy csinálja a Controller osztályt és annak leszármazottjait.

2021.02.25. 21:00	1 óra	Szabó	Tevékenység: Osztálydiagram vázának létrehozása
2021.02.26. 14:00	2 óra	Novák	Tevékenység: Az előző dokumentum hibáinak kijavítása
2021.02.26. 15:00	1,5 óra	Czanik	Tevékenység: Material és leszármazottjai + inventory osztályleírása
2021.02.26. 17:00	1,5 óra	Szabó	Tevékenység: Active, EndType enum és GameManager és Menu osztályok leírása és diagramra vitele
2021.02.26. 20:00	2 óra	Czanik Nagy Novák Szabó Tokovics	Értekezlet: (Szekvencia diagrammok átbeszélése és kiosztása) Döntés: Szabó csinálja a Win, Lose, applyEruption Asteroid/Gate, eruptionController. Döntés: Tokovics csinálja a többi (21 db).
2021.02.27. 9:00	2 óra	Nagy	Tevékenység: A Controller osztály és annak leszármazottjainak leírása
2021.02.27. 10:00	2 óra	Novák	Tevékenység: Az Entity és leszármazottjainak osztályleírása
2021.02.27. 10:00	1,5 óra	Tokovics	Tevékenység: A Thing osztály és annak leszármazottjainak, emellett a MainAsteroid osztály

			osztályleírásának megalkotása.
2021.02.27. 12:00	8 óra	Tokovics	Tevékenység: A szekvencia-diagrammok nagyrészének megalkotása
2021.02.27. 17:00	2 óra	Szabó	Tevékenység: Szekvencia diagrammok megcsinálása: Win, Lose, applyEruption Asteroid/Gate, eruptionController
2021.02.28. 12:00	3 óra	Nagy	Tevékenység: osztályok leírásának javítása, előző dokumentum hibáit kijavító leírás átnézése javítása, objektum katalógus ellenőrzése, javítása
2021.02.28. 17:00	2 óra	Czanik	Tevékenység: osztályleírások és osztálydiagramok ellenőrzése és javítása
2021.02.28. 18:00	1 óra	Szabó	Tevékenység: Ellenőrzések
2021.02.28. 19:30	3,5 óra	Novák	Tevékenység: A dokumentum összeollózása, helyesírási hibák javítása, egyéb funkcionális hibák javítása, utolsó simítások, illetve ellenőrzések.
2021.02.28. 23:00	0,5 óra	Novák	Tevékenység: Napló megszerkesztése és összeállítása.