**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 |
| **Szabó Bence Sándor** | **NQB6UO** | **szabo.bence.sandor@gmail.com** |

2021. február 28.

# Analízis modell kidolgozása

## Előző dokumentum hibáinak javítása

### 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.

### Felhasználók (változtatás):

* A játékot akármennyi játékos játszhatja.

### 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)**.”

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

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

## Objektum katalógus

### Asteroid

Rajtuk tartózkodnak a telepesek 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.

### Material

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

### Settler

A játékos által irányított karakterek. Az aszteroidákon tudnak lépkedni mindig a szomszédosokra. Több műveletet tud végezni: bányászni, fúrni, 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.

### Robot

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

### 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 telepesek, vagy robotok akkor a másik kapuhoz fogja őket teleportálni.

### 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.

### Game manager

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

### Main asteroid

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

### Radioactive

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

### Vaporizable

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

## Statikus struktúra diagramok

## Osztályok leírása

### 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

### Asteroid

#### Felelősség

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

#### Ősosztályok

Thing → Asteroid

#### Interfészek

-

#### Attribútumok

* **#core: Material[0..1]:** Az aszteroidában található nyersanyag
* **layer: int:** A kőzetréteg aktuális vastagsága
* **nearBySun: boolean:** Tárolja, hogy napközelben 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.

### Controller

#### Felelősség

Az irányítható dolgok irányításáért felelős.

#### Ősosztályok

-

#### Interfészek

-

#### Attribútumok

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

#### Metódusok

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

### EndType <<enumeration>>

#### Felelősség

A játék végének típusát lehet vele jelezni.

#### Ősosztályok

-

#### Interfészek

-

#### Attribútumok

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

### 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 karekter 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.

### 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.

### 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 eléri őket.

#### Ősosztályok

-

#### Interfészek

-

#### Attribútumok

* **doneControllers:** A körüket befejező Conroller osztályok száma**.**
* **controllers:** A GameManager által felügyelt Controller osztályok**.**
* **recipes:** A különböző építésekhez szükséges receptek**.**
* **myMenu:** A Menu, ami a GameManagert 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.

### Inventory

#### Felelősség

Materiálok tárolásáért felelős osztály.

#### Ősosztályok

-

#### 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 hashmappel oldunk meg.

#### Metódusok

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

### MainAsteroid

#### Felelősség

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

#### Ősosztályok

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.

### Material

#### Felelősség

A nyersanyagok tárolásáért felelős osztály. Tudja, hogy kibányászáskor milyen messze van a naptól. Ő valósítja meg az aszteroida magját.

#### Ősosztályok

-

#### Interfészek

-

#### Attribútumok

* **#name: string:** A nyersanyag megnevezése

#### Metódusok

* **void nearSun():** Ha aszteroida napközelbe kerül akkor hívódik meg. A különböző típusú aszteroidákon más hatást fejt ki.

### Menu

#### Felelősség

A játék paramétereinek bekérése/beállítása és a GameManager osztály létrehozása, ezzel együtt a játék el indítása.

#### Ősosztályok

-

#### Interfészek

-

#### Attribútumok

-

#### Metódusok

* **void startGame()**: Elindítja a játékot és beállítja a paramétereit.

### RadioactiveMaterial

#### Felelősség

A radioaktív anyagot valósítja meg. Ha napközelben 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 napközelben 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.

### 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.

### RobotController

#### Felelősség

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

#### Ősosztályok

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.

### Settler

#### Felelősség

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

#### Ősosztályok

Entity → Settler

#### Interfészek

-

#### Attribútumok

* **myController: SettlerController:** Ezzel az attribútummal jelzi a controller osztályának, ha meghal a telepes, 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 telepes által szükséges nyersanyag.
* **gates: TeleportGate[0..2]:** A telepes által létrehozandó kapukat jelöli.

#### Metódusok

* **void buildBase()**: A kezdő aszteroidán építi a bázist (egy körben értelemszerűen csak egy új elemmel).
* **void buildGate()**: A függvény meghívásával épít egy teleport kapupárt a szükséges nyersanyagokból a telepes.
* **void buildRobot()**: A függvény meghívásával épít egy robotot a szükséges nyersanyagokból a telepes.
* **void die()**: Ez a függvény azt jelzi, hogy a telepes meghal.
* **void drill():** A telepes 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 telepes meghal.
* **void mine()**: A telepes bányászik az adott aszteroidán.
* **void move(Thing destination)**: A telepesek 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 telepes teleportál a kiválasztott kapuk között és ezt jelzi a SettlerController osztálynak.
* **void wait()**: A telepes az adott körben „nem csinál semmit”, vagyis várakozik az aktuális helyén és ezt jelzi a SettlerController osztálynak.

### SettlerController

#### Felelősség

A telepesek irányításáért felelős, feldolgozza a felhasználótól kapott utasításokat.

#### Ősosztályok

Controller → SettlerController

#### Interfészek

-

#### Attribútumok

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

#### Metódusok

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

### 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 telepes a pályára, tehát használható-e már.

#### Metódusok

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

### Thing

#### Felelősség

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

#### Ősosztályok

-

#### Interfészek

-

#### Attribútumok

* **neighbour: Thing[1..\*]:** A vele szomszédos objektumok, amelyekre átléphet a telepes 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.

### VaporizableMaterial

#### Felelősség

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

#### Ősosztályok

Material → VaporizableMaterial

#### Interfészek

-

#### Attribútumok

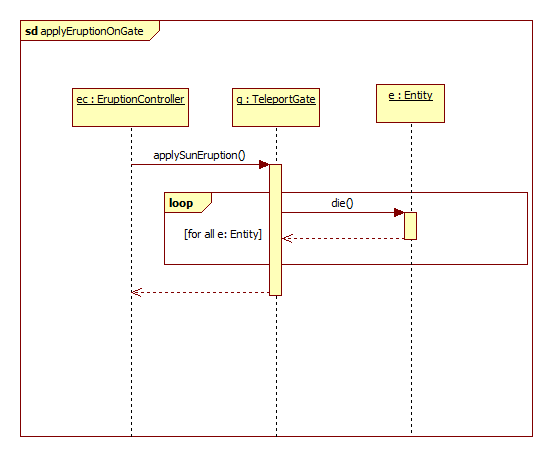
* **#name: string:** Az anyag neve**.**

#### Metódusok

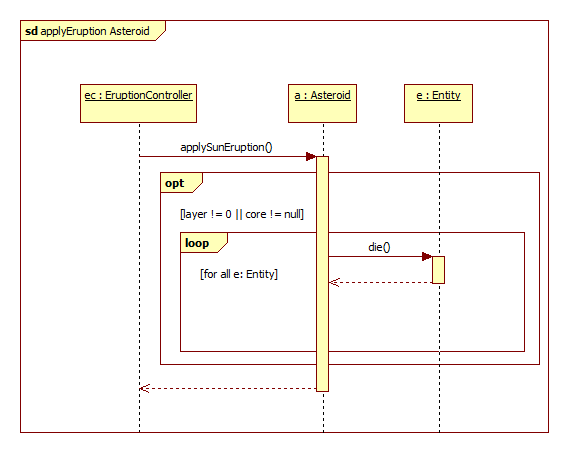
* **void nearSun():** Ha az őt tartalmazó aszteroida napközelben van, amikor kibányásszák, akkor elszublimálódik, vagyis eltűnik a nyersanyag és a játékos nem kap semmit, majd kicseréli magát üres Materialra.

## Szekvencia diagramok

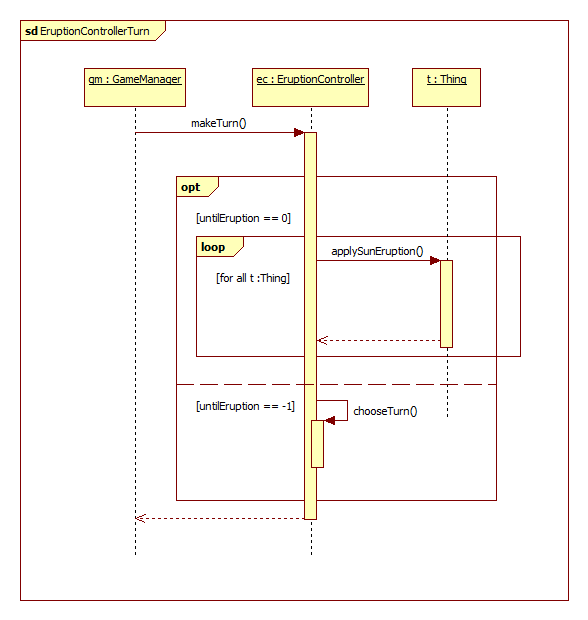
### applyEruptionOnGate



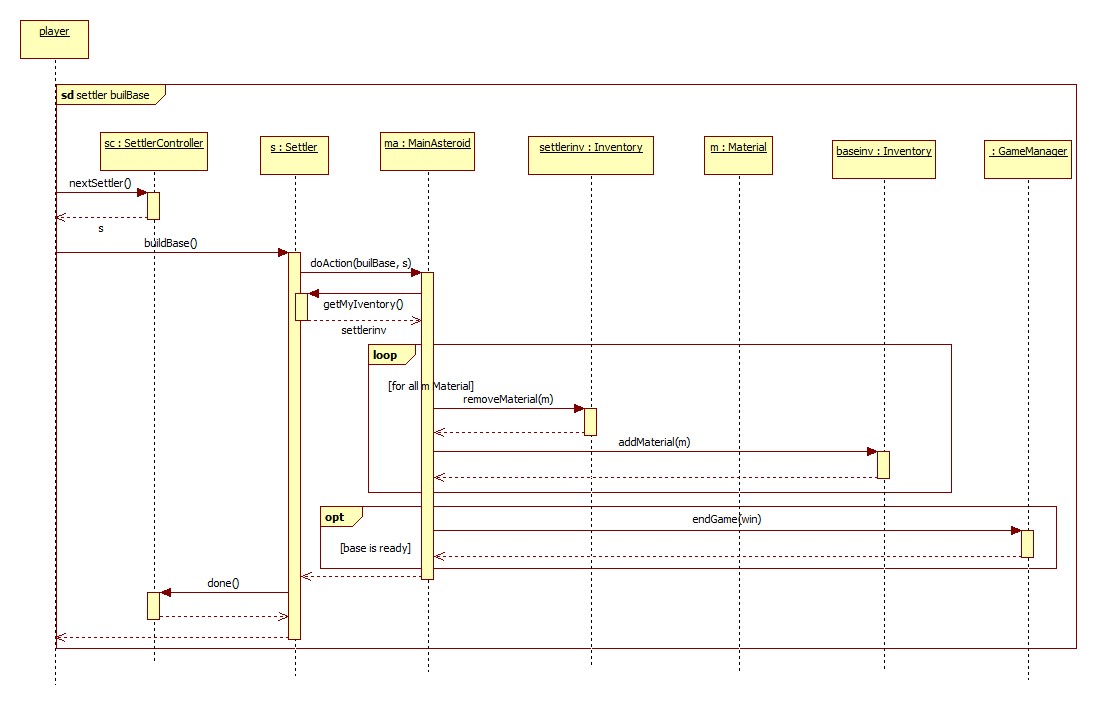
### applyEruption Asteroid



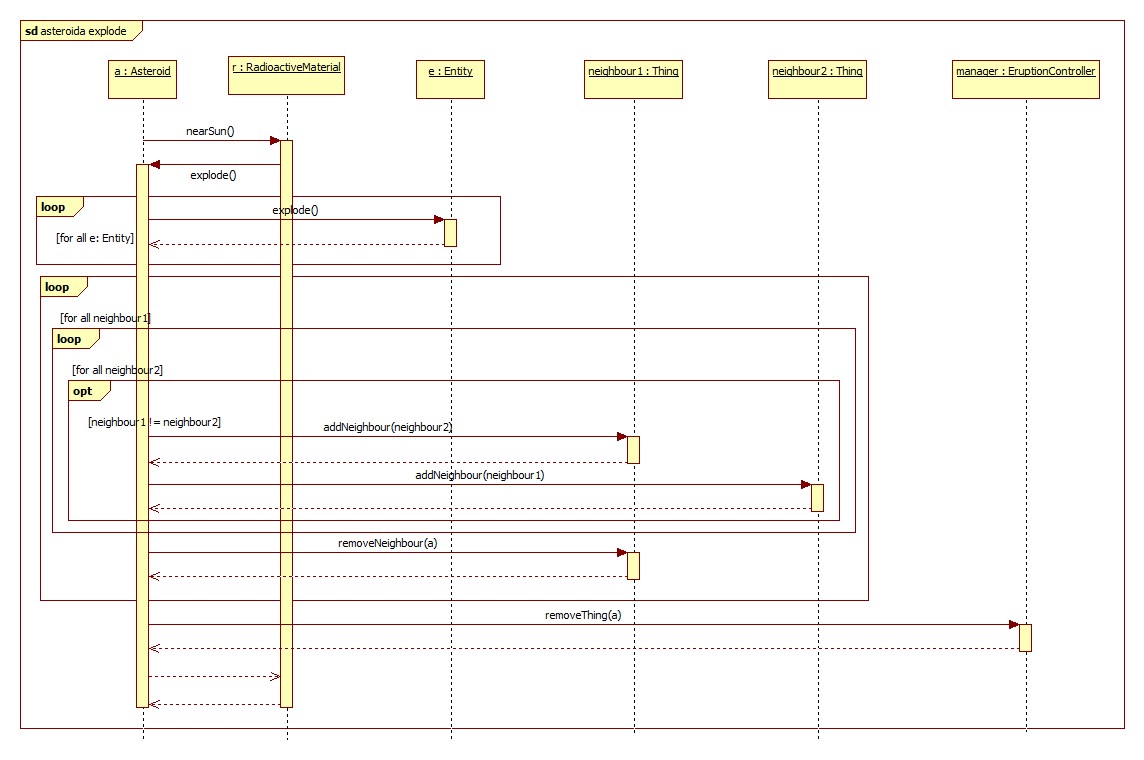
### EruptionControllerTurn



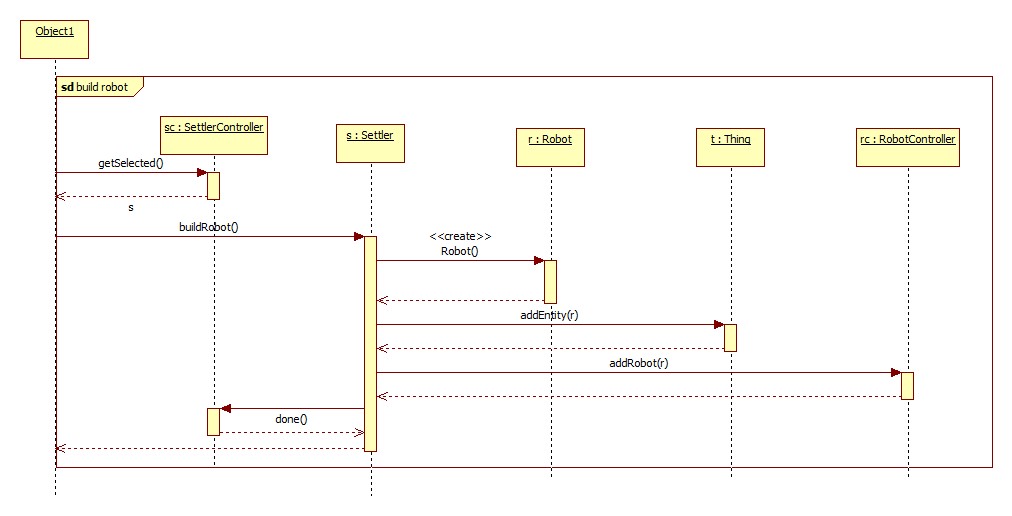
### settler buildbase



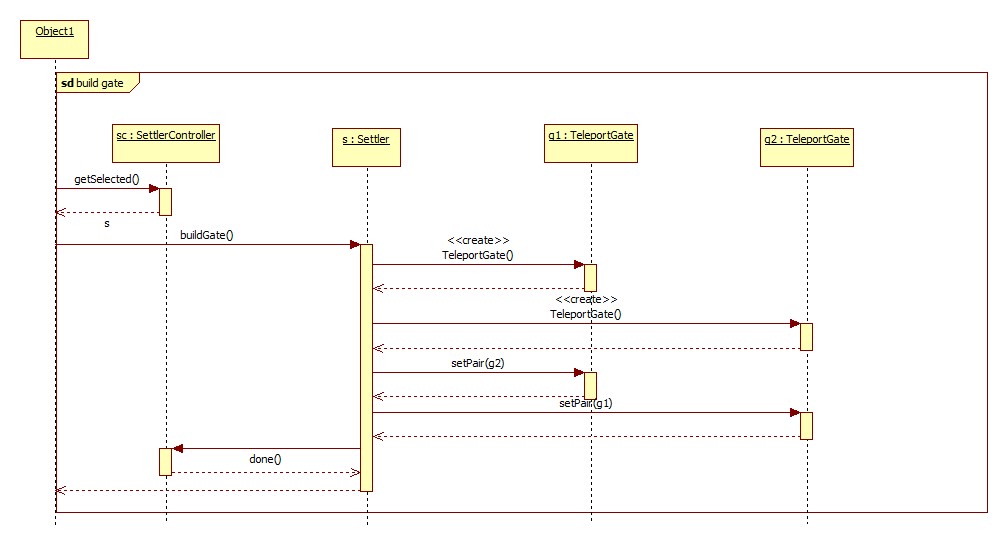
### asteroida explode



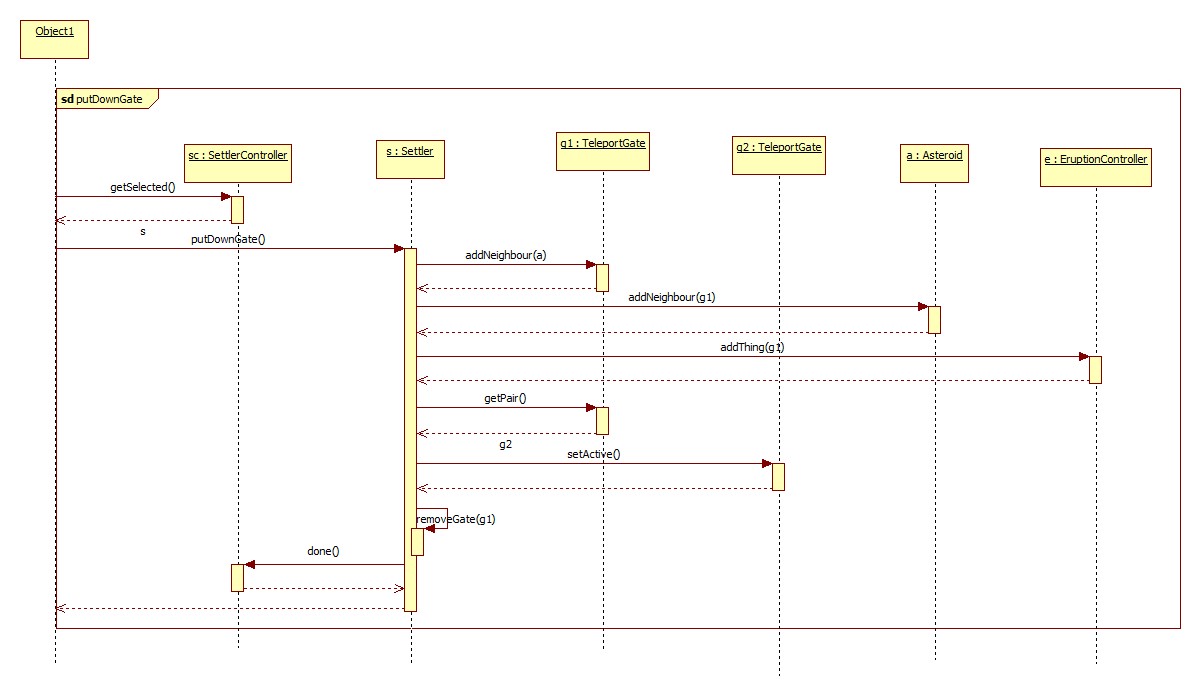
### build robot



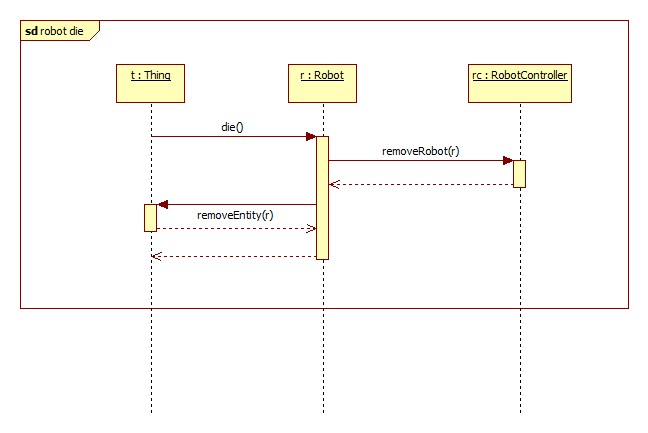
### build gate



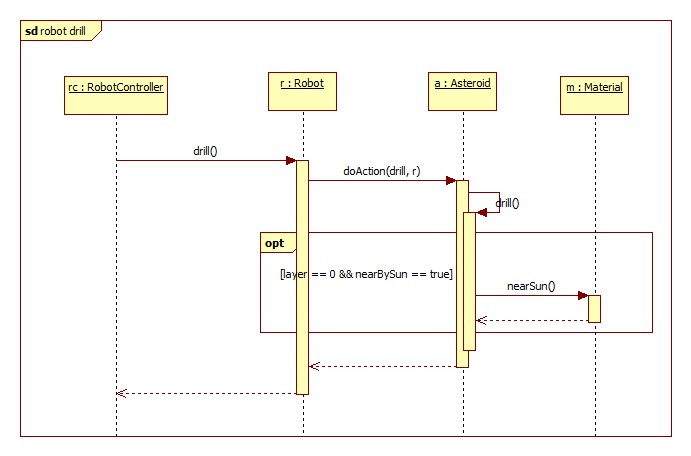
### putDownGate



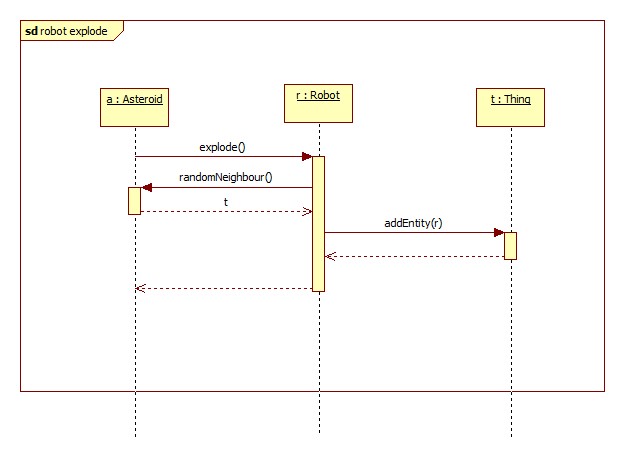
### robot die



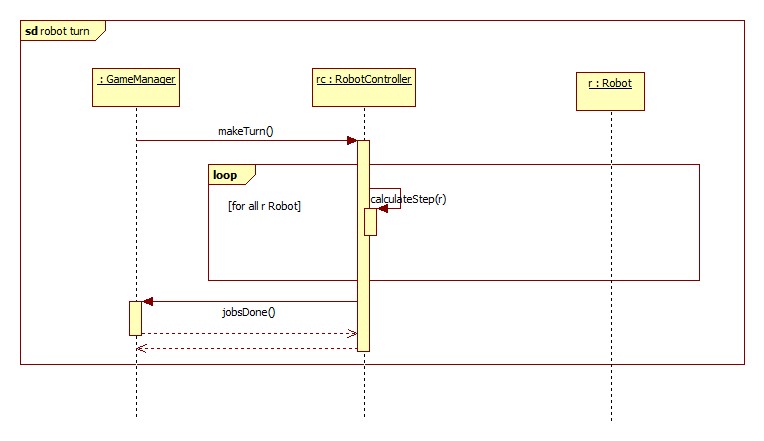
### robot drill



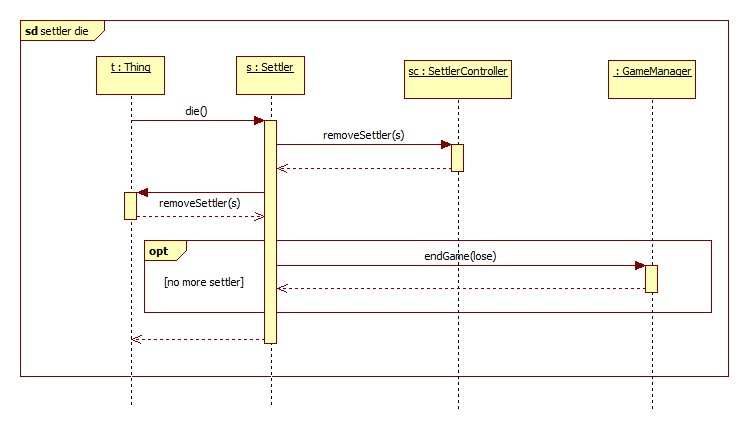
### robot explode



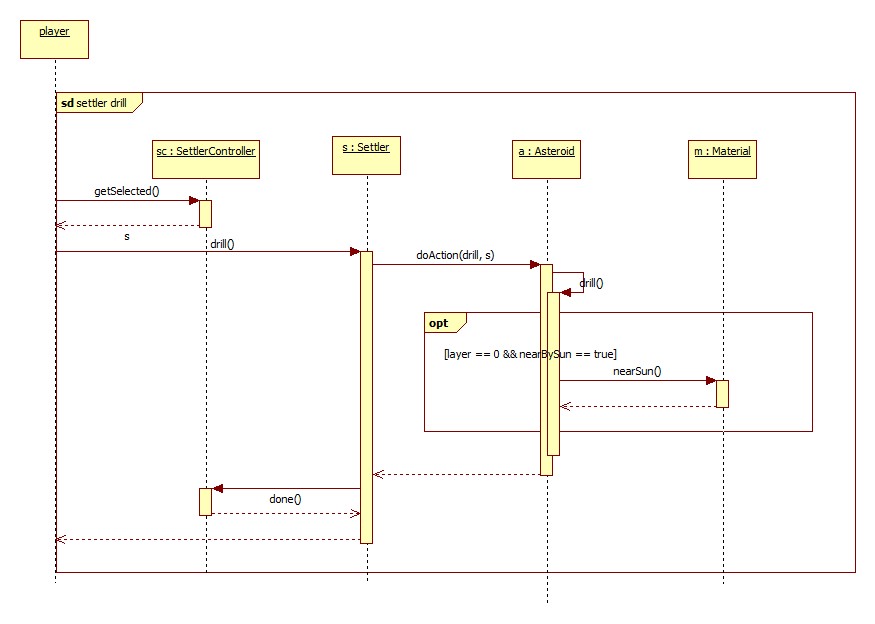
### robot turn



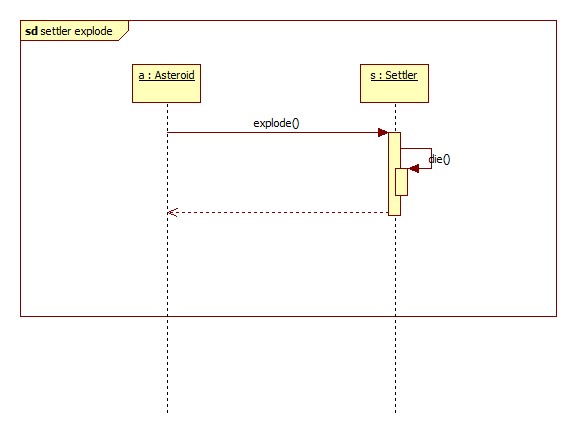
### settler die



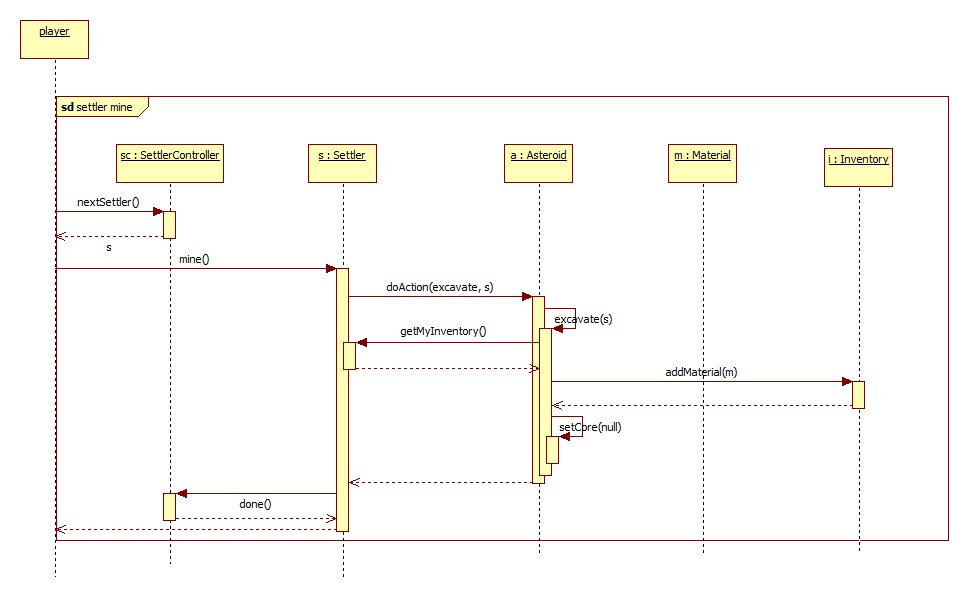
### settler drill



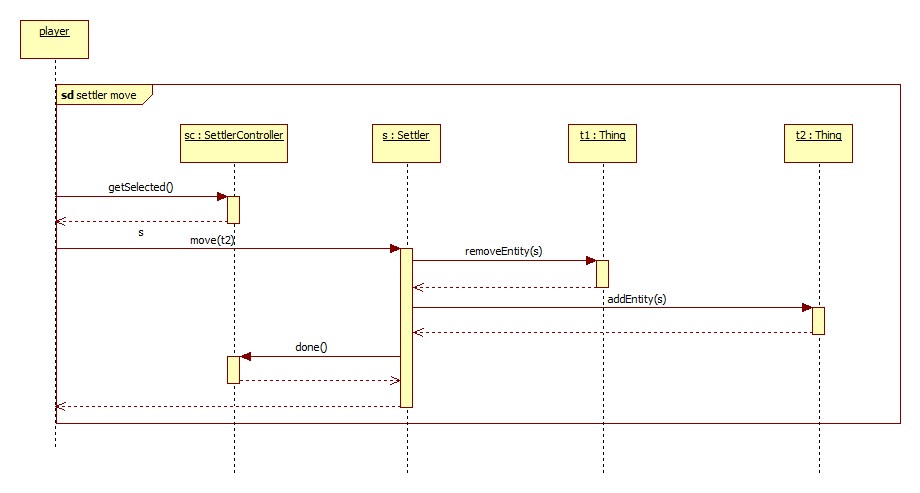
### settler explode



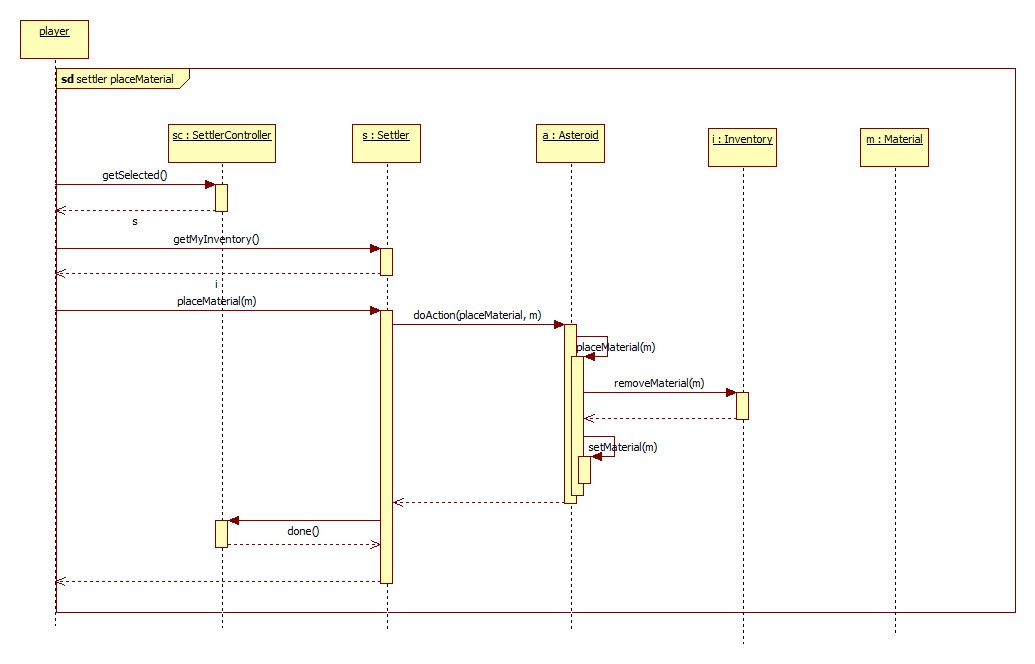
### settler mine



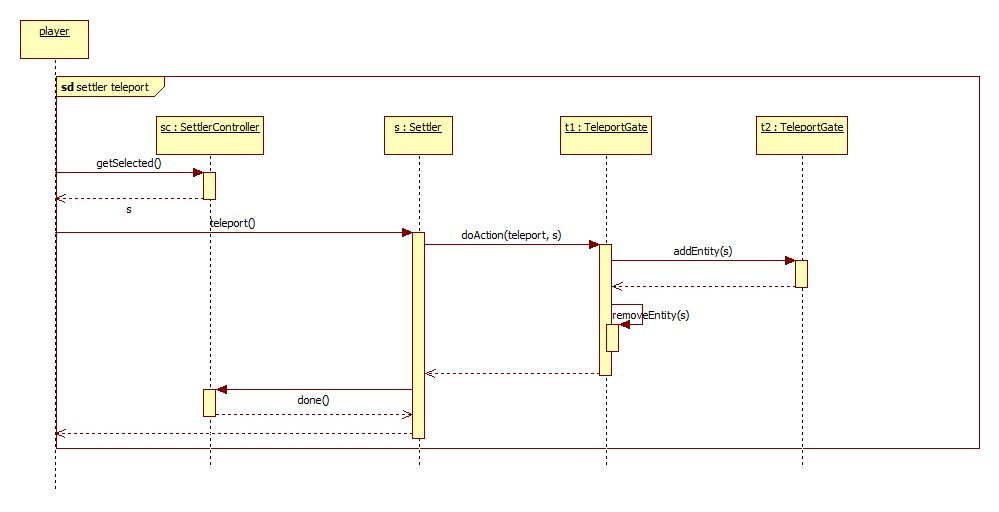
### settler move



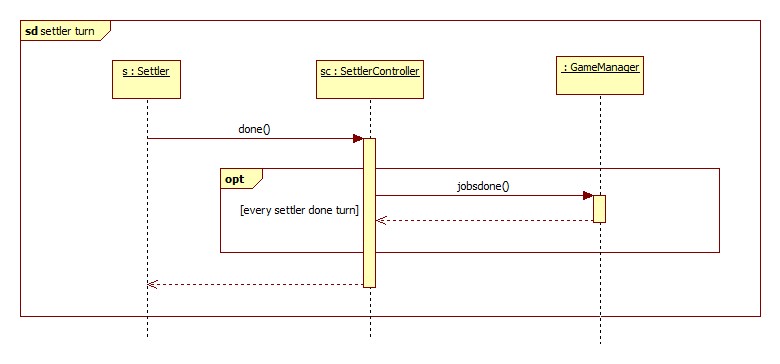
### settler placeMaterial



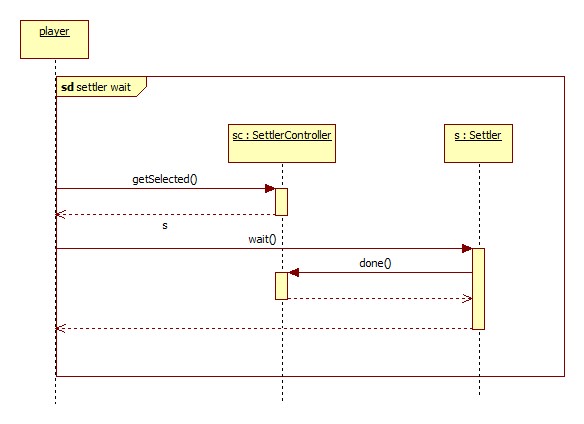
### settler teleport



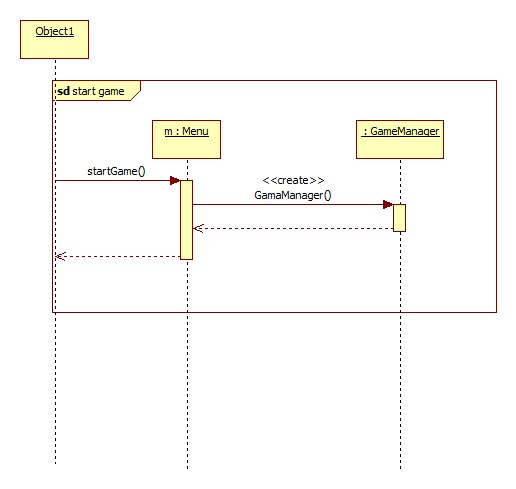
### settler turn



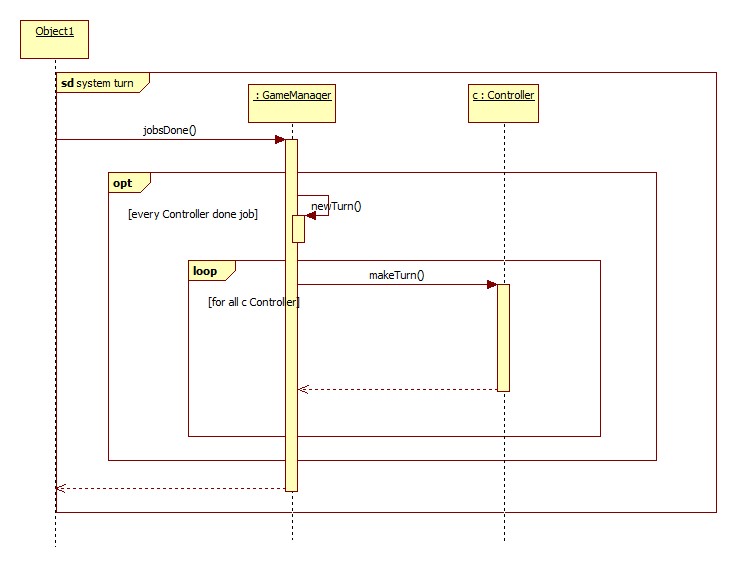
### settler wait



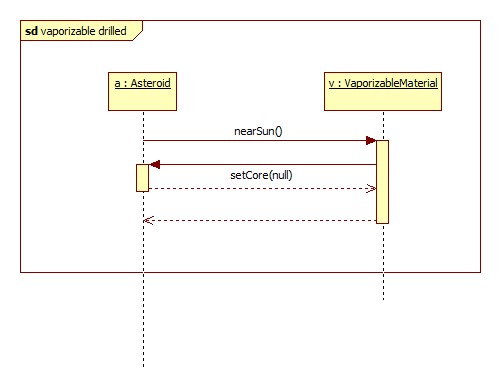
### start game



### system turn



### vaporizable drilled



## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **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 többit (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. |