

Software Technologie

Hausaufgabe

Hrotkó Renátó
OIT6HD

2020

1 Problem Beschreibung

Siehe Aufgabenbeschreibung.

Ich habe solche Bedingungen gestellt dass,

- *gleichzeitig können die Spieler spielen, und sie kommen nicht nacheinander*
- *es eine Controller (Spielleiter) ist der ganze Spiel leitet*
- *der Spiel so beginnt dass alle Spieler und Roboter schon auf einem Asteroide stehen*
- *die Asteroiden haben einen Storage(Lager) wohin die geförderte Rohstoffe geliefert werden und wird es kontrolliert ob die Spieler schon alle Arten gesammelt haben.*
- *In einem Sonnensystem (einfach in dem Spiel) kann nur eine Sonne existieren*

2 Funktionale Anforderungen

2.1. Primäre Anforderungen

Id	Beschreibung	Use-case-Name
F01	Die Spieler steuern die Siedler in dem Weltraum	Move Settler, Drill Asteroide, Mine Asteroide, View Solarsystem
F02	Die Spieler steuern Weltraumschiffe, die für einen Astronaut geeignet sind	Move Settler
F03	Die Asteroiden haben verschiedene Dicke	Drill Asteroide
F04	Die gesuchte Rohstoffe befinden sich in den Kern der Asteroiden	Mine Asteroide
F05	Die gefährliche Asteroiden haben einen radioaktiven Kern	Mine Asteroide
F06	Es gibt leere Asteroiden ohne Rohstoffe	Mine Asteroide
F07	Die Asteroiden haben einen homogenen Kern	Mine Asteroide
F08	In einem Schritt kann nur eine Tätigkeit durchgeführt werden	Move Settler, Drill Asteroide, Mine Asteroide
F09	Die Tätigkeiten sind: bewegen, bohren, fördern	Move Settler, Drill Asteroide, Mine Asteroide
F10	Der Spieler fährt zu einem benachbarten Asteroide	Move Settler
F11	Jeder Asteroide hat mehrere aber mindestens 1 Nachbarn	Move Settler
F12	Beim Bohren wird der Mantel mit einer Einheit tiefer	Drill Asteroide
F13	Beim Fördern wird der Rohstoff aus der Kern ausgenommen	Mine Asteroide
F14	Fördern ist nur dann erlaubt falls der Mantel gebohrt wurde	Drill Asteroide, Mine Asteroide
F15	Der radioaktive Rohstoff kann nur in Sonnenferne gefördert werden.	Mine Asteroide, View Solarsystem
F16	Solarflares können den Asteroidezone erreichen	Control Solarstorm
F17	Solarflares töten die Siedler	Control Solarstorm
F18	Solarflares kann der Spieler überleben falls in einem leeren Asteride ist	Control Solarstorm
F19	Den Hohlraum zu erreichen muss der Mantel voll gebohrt werden	Drill Asteroide
F20	Roboter helfen die Spieler	Control Robots
F21	Ein Robot kann nur eine benachbarte Asteroide wählen um weiterzugehen	Move Settler
F22	Roboter können bohren	Drill Asteroide
F23	Roboter können nicht fördern	Mine Asteroide
F24	Roboter überleben den Solarsturm falls sie in einem leeren	Control Solarstorm,

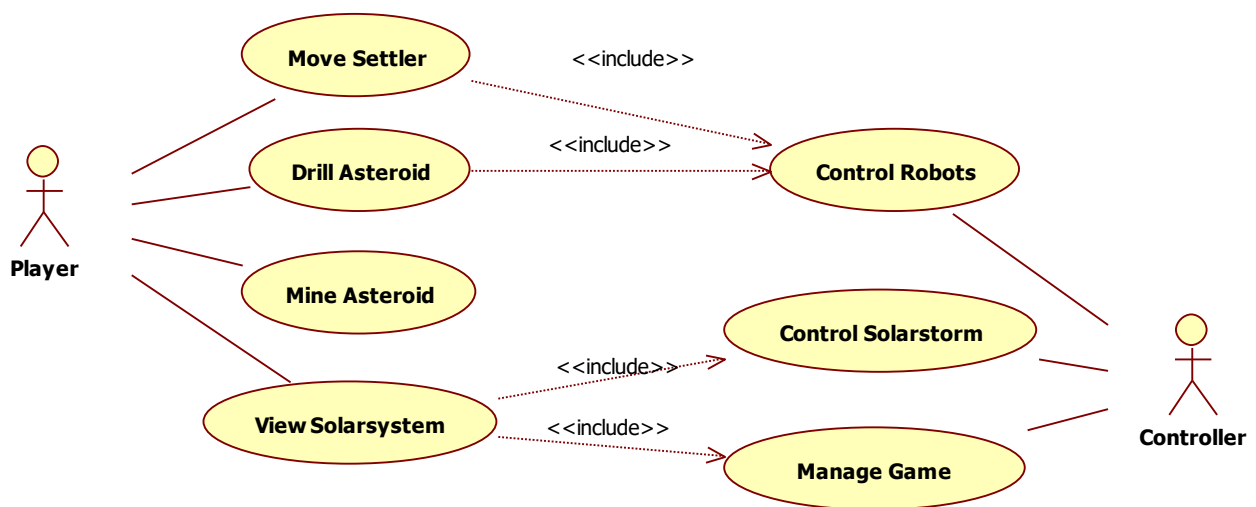
	<i>Asteroide sind</i>	<i>Constrol Robots</i>
<i>F25</i>	<i>Falls alle Siedlern tot sind dann ist der Spiel zu Ende gekommen</i>	<i>Manage Game</i>
<i>F26</i>	<i>Falls von aller Art von Rohstoffen mindestens eine Einheit gefördert wurde, und auf einen Asteroid gesammelt bereitsteht, dann haben die Spieler gewonnen</i>	<i>Manage Game</i>

2.2. Zusätzliche Anforderungen

Id	Beschreibung	Use-case-Name
<i>F27</i>	<i>Asteroiden haben einen Lager in denen die geförderte Rohstoffe gelegt werden</i>	<i>Mine Asteroide</i>
<i>F28</i>	<i>Nur eine Sonne existiert in dem Spiel</i>	<i>View Solarsystem</i>

3 Anwendungsfall (Use-case) Modell

3.1 Use-case Diagramm



3.2 Beschreibung der Use-case

Use-case-Name (Titel)	Move Settler
Kurze Beschreibung	Der Spieler leitet einen Siedler in dem Sonnensystem
Akteur	Player
Haupttätigkeit	Der Siedler bewegt auf dem nächsten benachbarten Asteroide
Alternative Tätigkeit	Der Siedler bleibt auf dem Asteroide
Alternative Tätigkeit	Der Siedler bringt die Ressourcen

Use-case-Name (Titel)	Drill Asteroid
Kurze Beschreibung	Der Spieler bohrt den Mantel der Asteroide
Akteur	Player
Haupttätigkeit	Der Spieler während des Bohrens macht den Mantel der Asteroide 1 Einheit kleiner
Alternative Tätigkeit	Der Spieler hat den Kern erreicht und muss nicht mehr bohren

Use-case-Name (Titel)	Mine Asteroid
Kurze Beschreibung	Der Siedler fördert den Rohstoff
Akteur	Player
Haupttätigkeit	Der Siedler fördert der Ressource aus dem Kern der Asteroide
Alternative Tätigkeit	1.Falls der Kern leer ist dann kann nichts gefördert werden
Alternative Tätigkeit	2.Falls die Asteroide radioaktiv ist und ist in Sonnennahe dann wird es explodieren
Alternative Tätigkeit	3.Falls die Asteroide radioaktiv ist und ist in Sonneferne dann wird es gefördert

Use-case-Name (Titel)	View Solarsystem
Kurze Beschreibung	Der Spieler kann den Solarsystem besichtigen/anschauen
Akteur	Player
Haupttätigkeit	1. Der Solarsystem wird auf dem Bildschirm gezeichnet 2. Der Spieler kann den aktuellen Zustand der Solarsystem anschauen

Use-case-Name (Titel)	Control Robots
Kurze Beschreibung	Die Roboter arbeiten in dem Solarsystem um dem Spieler zu helfen
Akteur	Controller
Haupttätigkeit	Sie arbeiten so dass bohren die Asteroiden falls es noch nicht gebohrt wurde
Alternative Tätigkeit	Falls es schon gebohrt wurde dann bewegt er uf den nächsten benachbarten Asteroide

Use-case-Name (Titel)	Control Solarstorm
Kurze Beschreibung	Manchmal passiert ein Solarflair
Akteur	Constroller
Haupttätigkeit	Wenn ein Solarflair passiert, werden die Spiler und Roboter sterben
Alternative Tätigkeit	Falls während der Solarflair die Spieler/roboter in einem Asteroide versteckt sind dann werden sie es überleben
Alternative Tätigkeit	Kein Solarflair passiert

Use-case-Name (Titel)	Manage Game
Kurze Beschreibung	Behandlung der Speilbefehle
Akteur	Controller
Haupttätigkeit	Ein neues Spiel zu erstellen
Alternative Tätigkeit	Der Spiel beenden falls die Spieler gewonnen oder verloren habem
Alternative Tätigkeit	Der Spiel pausieren/unterbrechen
Alternative Tätigkeit	Der Spiel fortsetzen nach einem Unterbrechung

4 Strukturbeschreibung

WICHTIG! Ich habe angenommen dass die Setter und Getter Funktionen zu den Attributen schon existieren. Sie erscheinen deswegen in dem Diagramm und Klassenbeschreibung nicht.

4.1 Klassenbeschreibungen

4.1.1. Player

- **Zuständigkeit**

Ein Spieler kann bohren, fördern und zu einem anderen Asteroide bewegen. Er kann explodiert werden und danach sterben oder verstecken von einem Sturm.

- **Attributen**

-asteroid: Asteroid	Ein Asteroide auf denen der Spieler zurzeit steht
-isAlive: Boolean	Es zeigt ob der Spieler noch lebt oder nicht
-isHiddenFromStorm : Boolean	Es zeigt ob er in einem Asteroide von der Sturm verstecken konnte oder nicht
-asteroid: Asteroid	Es speichert den aktuellen Asteroide auf denen der Spieler steht

- **Methoden**

+move(asteroid: Asteroid)	Der Spieler fliegt auf den banachbarten Asteroide
+drill(asteroid: Asteroid)	Der Spieler bohrt den Mantel der Asteroide und verringert den Wert der Dicke der Asteroidemantel mit 1 Einheit.
+isThereSolarstorm(sun: Sun): Boolean	Es gibt zurück ob in der Solarsystem einen Sturm ist. Er fragt es aus der Parameterliste angegebene Sun Objekt.
+die()	Es stellt der Wert der isAlive auf false und wird danach diese Spieler von dem Controller aus der Sonnensystem gelöscht.
+explodeHits()	Das ist eine abstrakte Methode die die Robot und Settler definieren, falls ein Explosion passiert.
+hide(asteroid: Asteroid)	Der Spieler wird in einem Asteroide verstecken falls es durchgebohrt und leer ist.

4.1.2. Settler

- **Zuständigkeit**

Ein Siedler kann, außer der vererbten Methoden, eine Asteroide fördern und der Rohstoff in den Lager platzieren.

- **Attributen**

-ressorces: Ressource[0..*]	Der Siedler speichert die Rohstoffe die er gefördert hat
-----------------------------	--

- **Methoden**

+mine(asteroid: Asteroid)	Der Siedler fördert der Rohstoff aus der Asteroide
+placeRessourceInStorage(ressource: Ressource, asteroid: Asteroid)	Platziert die Rohstoffe in dem Lager an der Asteroide.
+explodeHits()	Der Siedler wird vernichtet falls ein Explosion passiert.
+move()	Der Siedler fliegt auf den banachbarten Asteroide

4.1.3. Robot

- **Zuständigkeit**

Ein Robot kann in einem Schritt nur bohren oder bewegen. Außerdem kann er den radioaktiven Explosion überleben.

- **Attributen**

- **Methoden**

+step(asteroid: Asteroid)	In einem Schritt wird ausgewählt was er machen will. Möglichkeiten: Bohren, Bewegen oder auch einen Runde auslassen.
+explodeHits()	Der Roboter wird nicht vernichtet falls ein Explosion passiert sondern auf einem Asteroide platziert.
+move()	Der Roboter fliegt auf den benachbarten Asteroide was durch den Controller bestimmt wurde.

4.1.4. Asteroid

- **Zuständigkeit**

Eine Asteroide hat einen Mantel mit einem bestimmten Dicke, und ob es schon explodiert hat. Aus ihnen kann Rohstoffe gefördert werden. Spieler können auf sie landen, und in ihnen verstecken.

- **Attributen**

-core: Ressource	Es beinhaltet den Material der inneren Kern der Asteroide. Falls es null ist dann ist dort ein Hohlraum
-ressourceStorage: Ressource	Hier werden die gesammelte Materialien gespeichert
-player: Player	Der Spieler der zurzeit an der Asteroide ist. Es ist null falls keine Spieler da sind
-layerThickness: Integer	Die Dicke des Mantels der Asteroide
-isExploded: Boolean	Es zeigt ob es schon explodiert hat oder nicht

- **Methoden**

+getNeighbour(): Asteroid	Es gibt zurück eine Asteroide auf denen der Spieler weitergehen kann
+drilling(player: Player)	Diese Methode verringert den Wert des Mantels der Asteroide mit 1 Einheit
+checkStorage(): Boolean	Falls auf diesem Asteroide schon alle unterschiedliche Arten von Rohstoffe gesammelt wurden, dann gibt es True zurück und zeigt dass der Spiel zu Ende gekommen ist
+scenarioSolarstorm()	Falls ein Sturm passiert dann wird es untersucht ob der Kern leer ist wenn ja dann wird der Spieler versteckt
+mining(player: Player)	Der Kern wird von einem Spieler gefördert. Abstrakte Methode weil radioaktive und nicht radioaktive Asteroiden werden unterschiedlich gefördert.
+acceptPlayer(player: Player)	Akzeptiert dass der Spieler auf den Asteroide gelandet hat.
+removePlayer(player: Player)	Entfernt der Spieler von der Asteroide

4.1.5. NonRadioactiveAsteroid

- **Zuständigkeit**

Eine Asteroide mit einem Kern, in denen Rohstoffe sind, die nicht radioaktiv sind. Es kann auch leer sein. Der Rohstoff, falls ein drin ist, kann gefördert werden.

- **Attributen**

- **Methoden**

+mining(player: Player)	Der Siedler fördert der Rohstoff aus der Asteroide. Der (Core) Kern wird auf null gesetzt also später können hier Siedler verstecken. Falls es originell null war (also leer) dann kann nicht gefördert werden aus der Asteroide
-------------------------	--

4.1.6. RadioactiveAsteroid

- **Zuständigkeit**

Eine Asteroide mit einem Kern, in denen Rohstoffe sind, die radioaktiv sind. Der Rohstoff kann gefördert werden. Es kann auch explodieren falls es zu nah zu der Sonne gefördert wird.

- **Attributen**

-isCloseToTheSun(): Boolean	Es zeigt ob es zu nah zu der Sonne ist oder nicht weil falls der Kern noch radioaktive Material enthält dann kann es explodieren.
-----------------------------	---

- **Methoden**

+explode(player: Player)	Die Asteroide wird explodieren und der Spieler töten.
+mining(player: Player)	Der Siedler fördert der Rohstoff aus der Asteroide. Der (Core) Kern wird auf null gesetzt also später können hier Siedler verstecken. Aber da es radioaktiv ist und wenn nicht in Sonnenferne gefördert wird dann kommt die Explosion.

4.1.7. Ressource

- **Zuständigkeit**

Ein Rohstoff die während der Spiel gebraucht ist. Sie werden gesammelt um einen Basis zu bauen.

- **Attributen**

-name: String	Der Name der Material
-isRadioActive: Boolean	Logische Wert ob der Material radioaktiv ist

- **Methoden**

4.1.8. Sun

- **Zuständigkeit**

Es kann einen Solarflair machen, der die Spieler töten kann

- **Attributen**

-isThereSolarstorm: Boolean	Es zeigt ob die Sonne eine Sonnenflair verursachte oder nicht
-----------------------------	---

- **Methoden**

+solarStorm()	So passiert einen Sonnenflair die die Spieler töten kann falls sie nicht in einem leeren Asteroide stecken. Daneben wird auch der obige Variable auf Wahr gesetzt.
---------------	--

4.1.9. SolarSystem

- **Zuständigkeit**

In einem Sonnensystem sind die Asteroiden und Spieler und eine Sonne gespeichert.

- **Attributen**

-players: Player[0..*]	Liste der Spieler die zurzeit spielen
-asteroids: Asteroid[0..*]	Liste der Asteroide in dem Sonnensystem
-sun: Sun	Die Sonne Objekt in dem Sonnensystem (es kann nur eine Sonne in einem Sonnensystem sein)

- **Methoden**

+removeAsteroid(asteroid: Asteroid)	Es entfernt einen Asteroide aus dem Sonnensystem falls es explodierte
+removePlayer(player: Player)	Es entfernt einen Spieler aus dem Sonnensystem falls es gestorben ist

4.1.10. Controller

- **Zuständigkeit**

Es leitet der Spiel. Hier wird ein neues Spiel erstellt, falls es zu Ende gekommen ist dann beendet und falls die Spieler eine Pause möchten dann wird es unterbrochen und dann fortgesetzt.

- **Attributen**

-solarSystem: SolarSystem	Den ganzen Sonnensystem ist gespeichert
---------------------------	---

- **Methoden**

+createNewGame()	Ein neues Spiel wird hergestellt mit einem neuen Sonnensystem
+finishGame()	Der Spiel wird beendet und kontrolliert ob alle schon gestorben sind oder sie könnten die Rohstoffe sammeln
+pauseGame()	Der Spiel wird pausiert wenn die Spieler Zeit brauchen
+continueGame()	Der Spiel wird nach einem Pause fortgesetzt

4.1.11. Steppable

- **Zuständigkeit**

Es repräsentiert die solche Klassen die in Zeit einen Schritt machen können

- **Attributen**

- **Methoden**

+step()	Es führt die Befehle zu dem aktuellen Operation durch.
---------	--

4.1.12. Timer

- **Zuständigkeit**

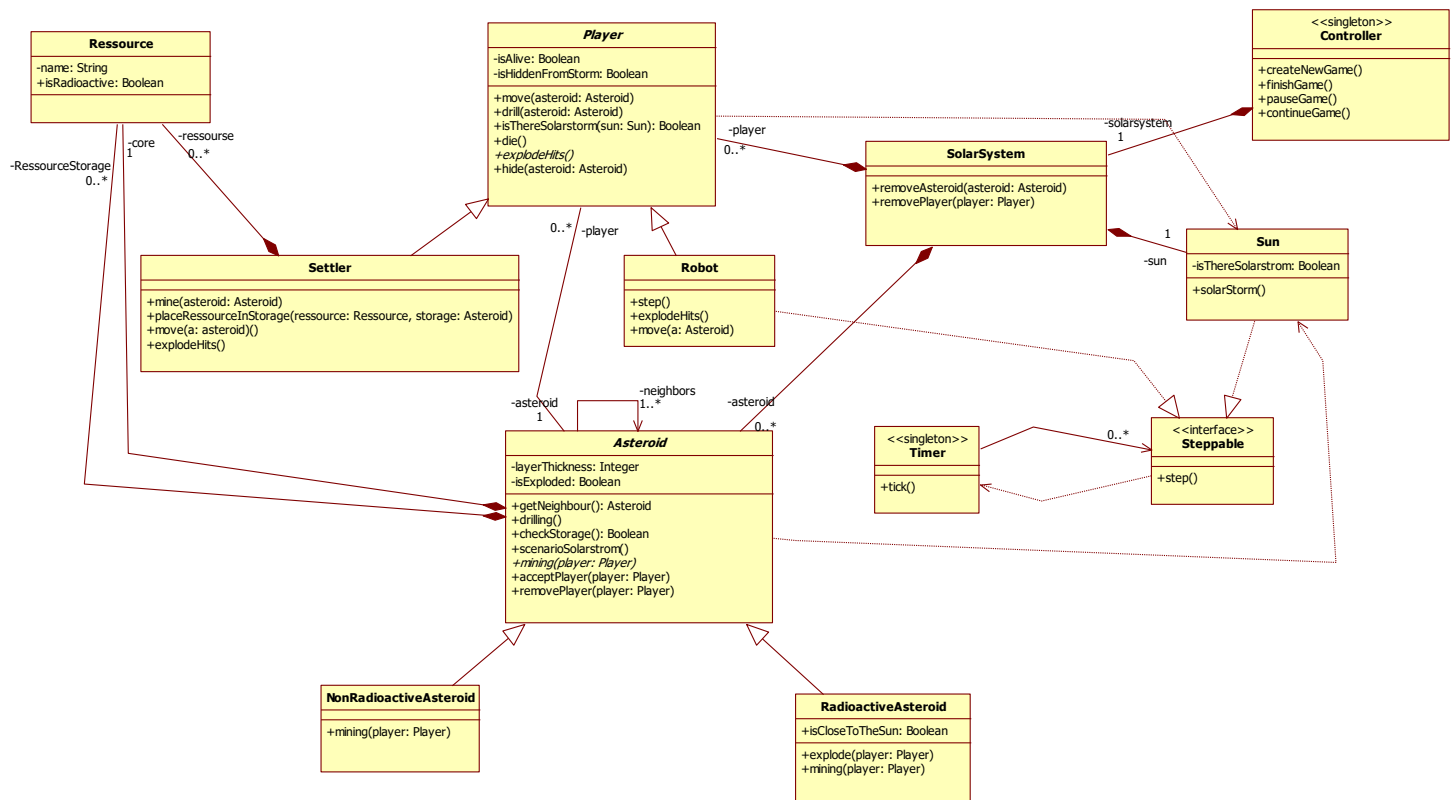
Es ist ein Taktsignal das in jedem Sekunden einen Signal ausgibt zu dem Klassen die Die Steppable Klasse implementieren.

- **Attributen**

- **Methoden**

+tick()	Ein Signal wird gesendet zu den Klassen, die die Steppable Klasse streut.
---------	---

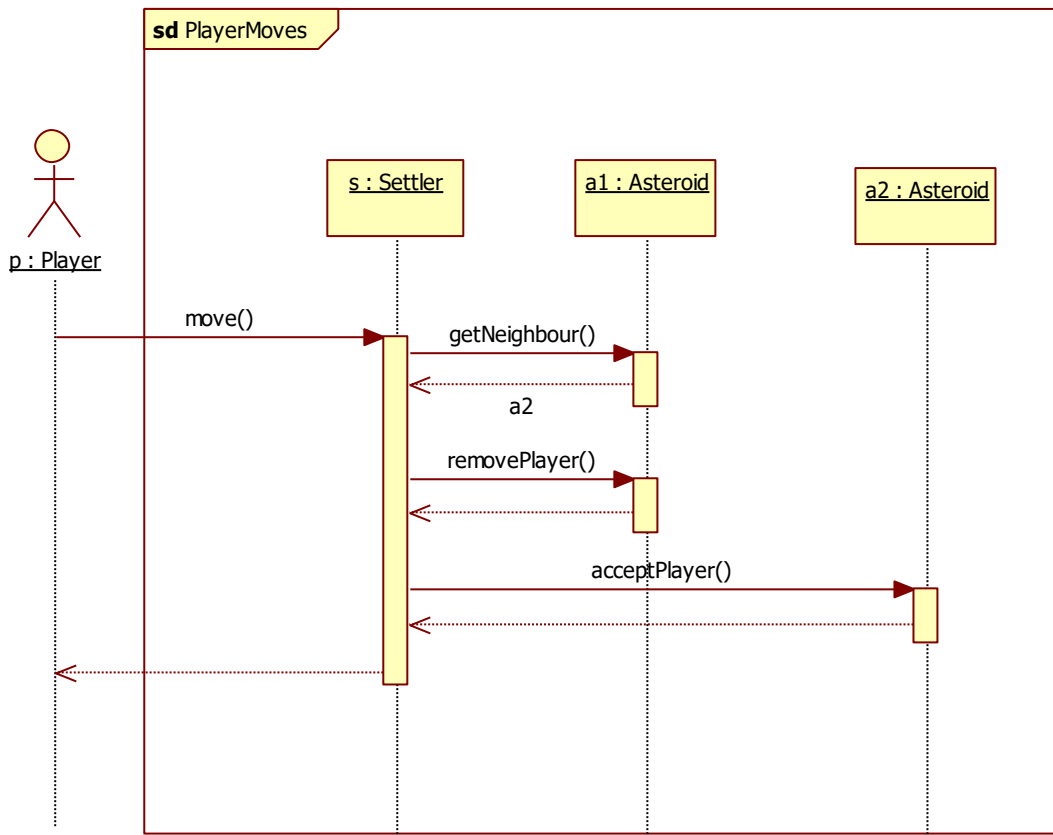
4.2 Klassendiagramm



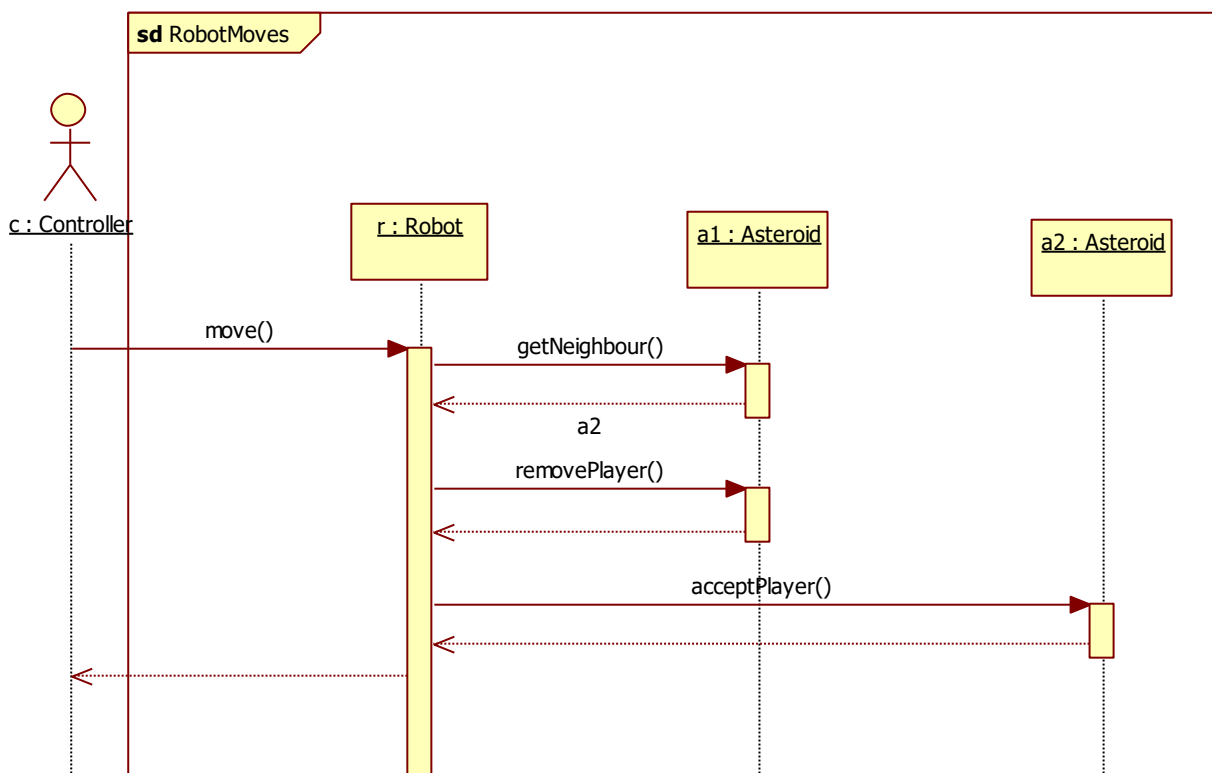
5 Funktionale Beschreibung

5.1 Ablaufdiagramme

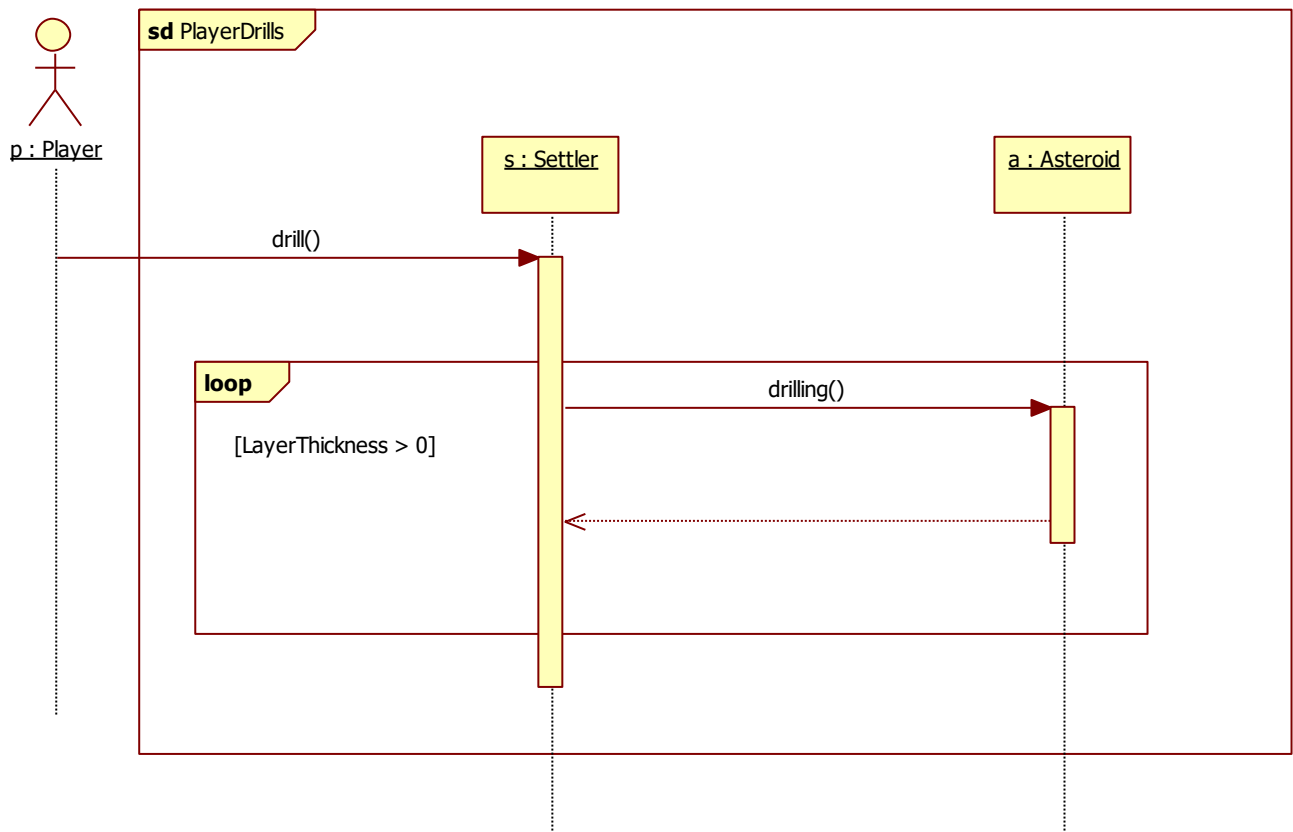
5.1.1. Player Moves / Spieler bewegt sich



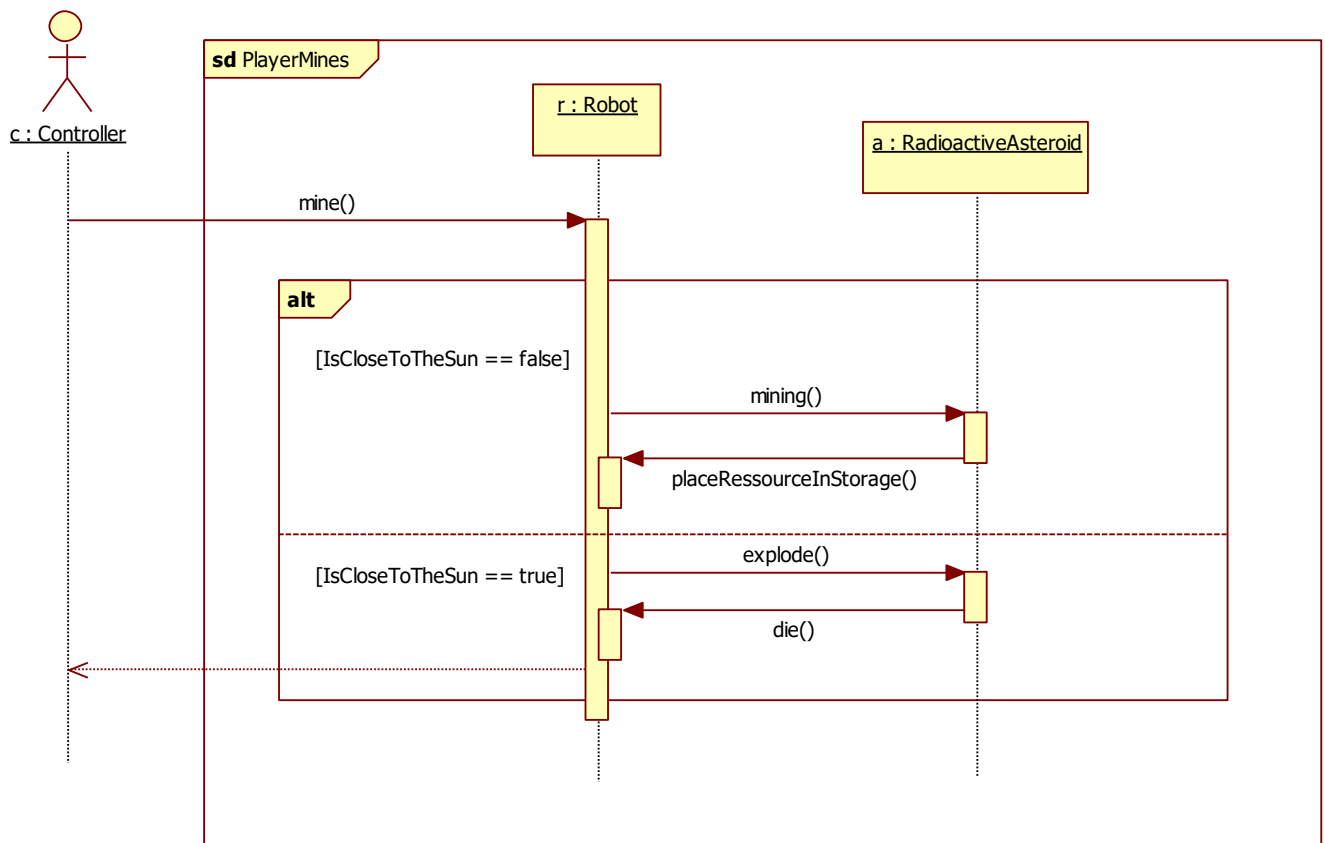
5.1.2. Robot Moves / Robot bewegt sich



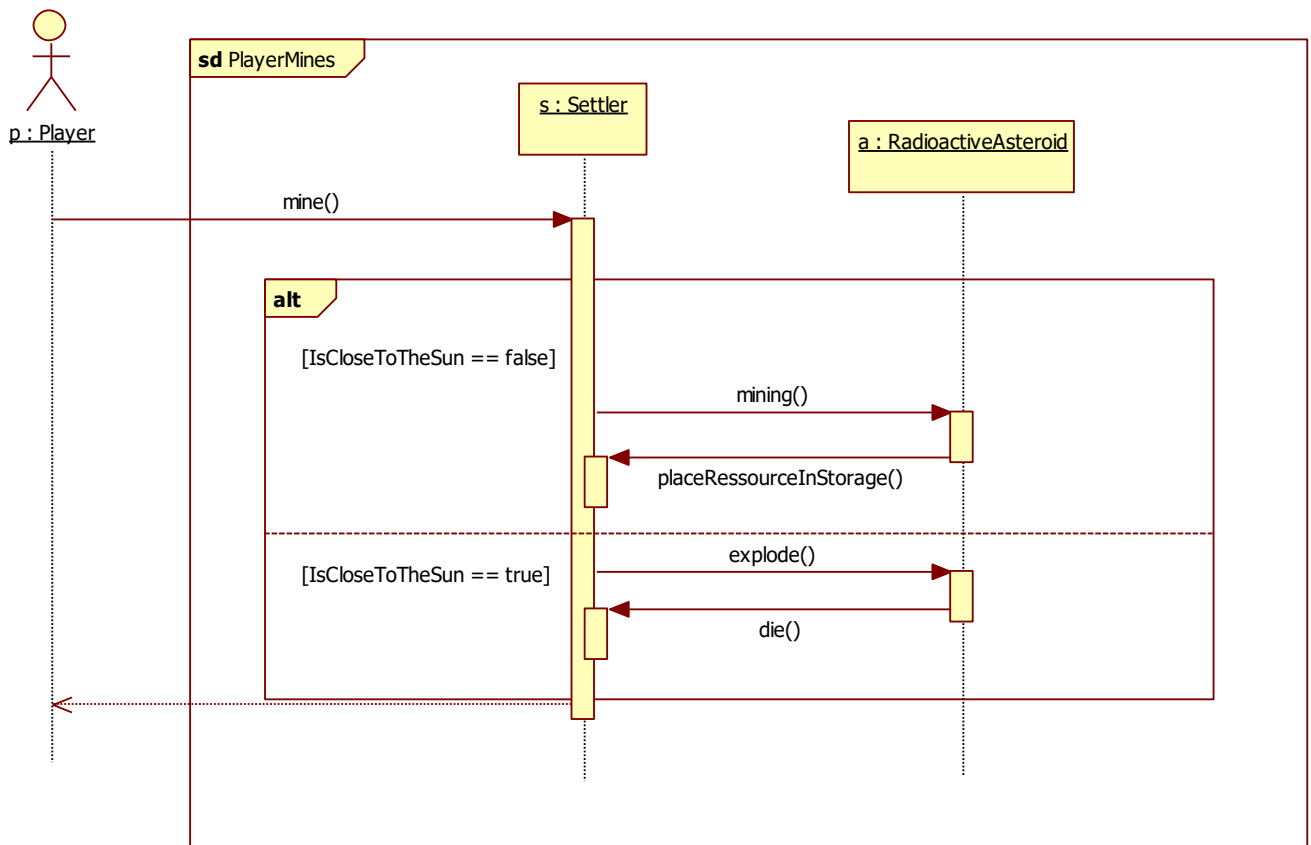
5.1.3. Settler Drills / Sideler bohrt



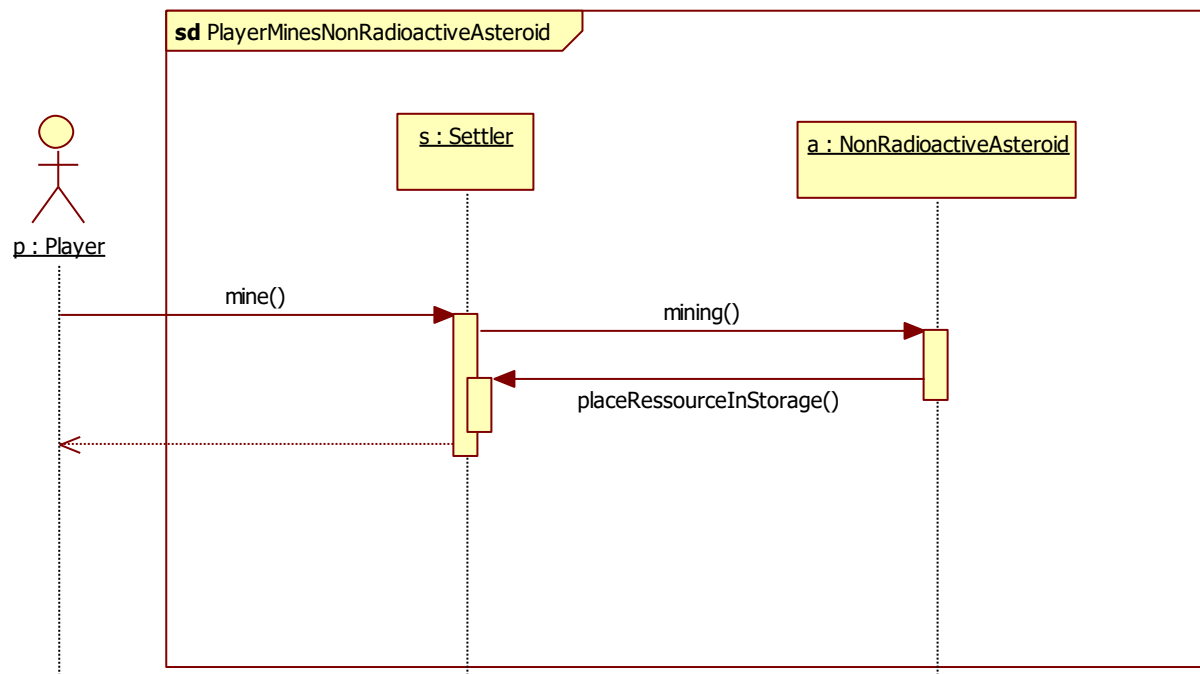
5.1.4. Robot Drills / Robot bohrt



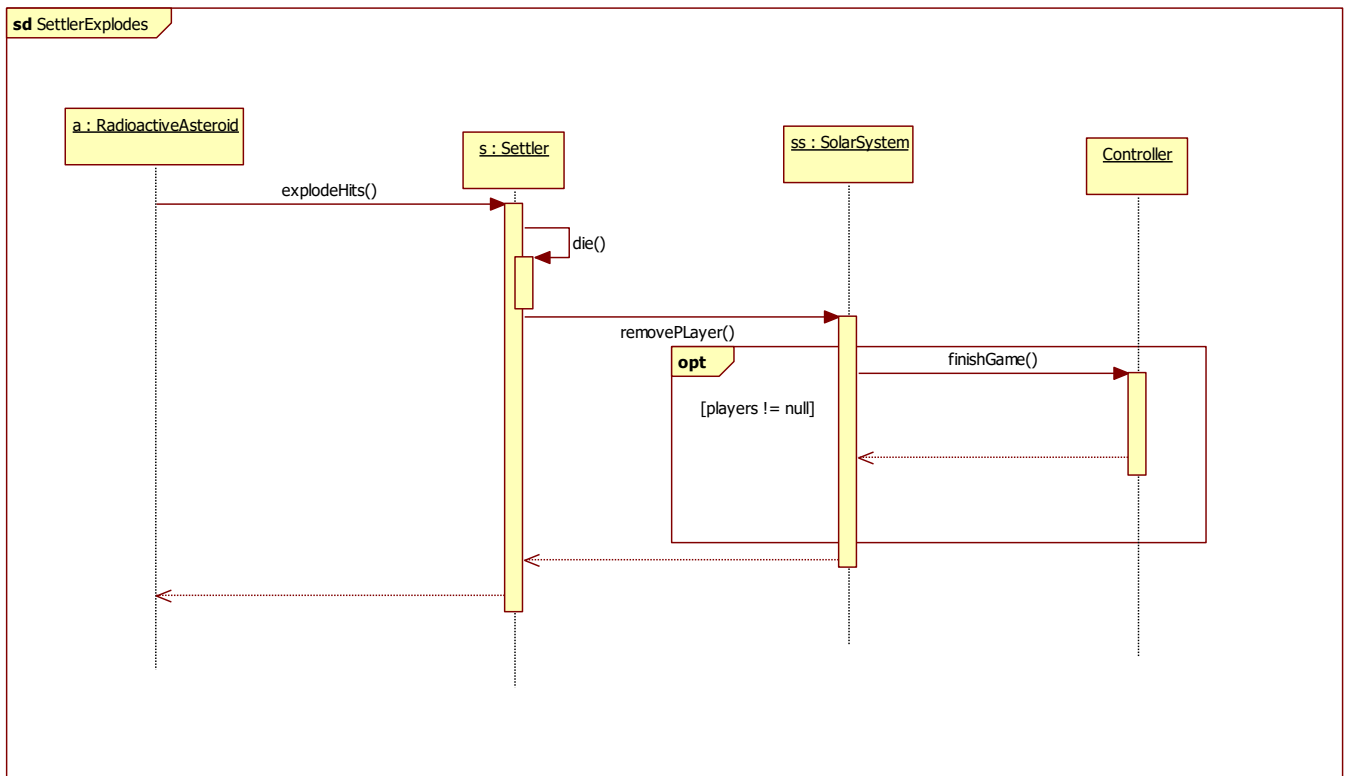
5.1.5. Settler Mines Radioactive Asteroid / Siderler fördert radioaktive Asteroide



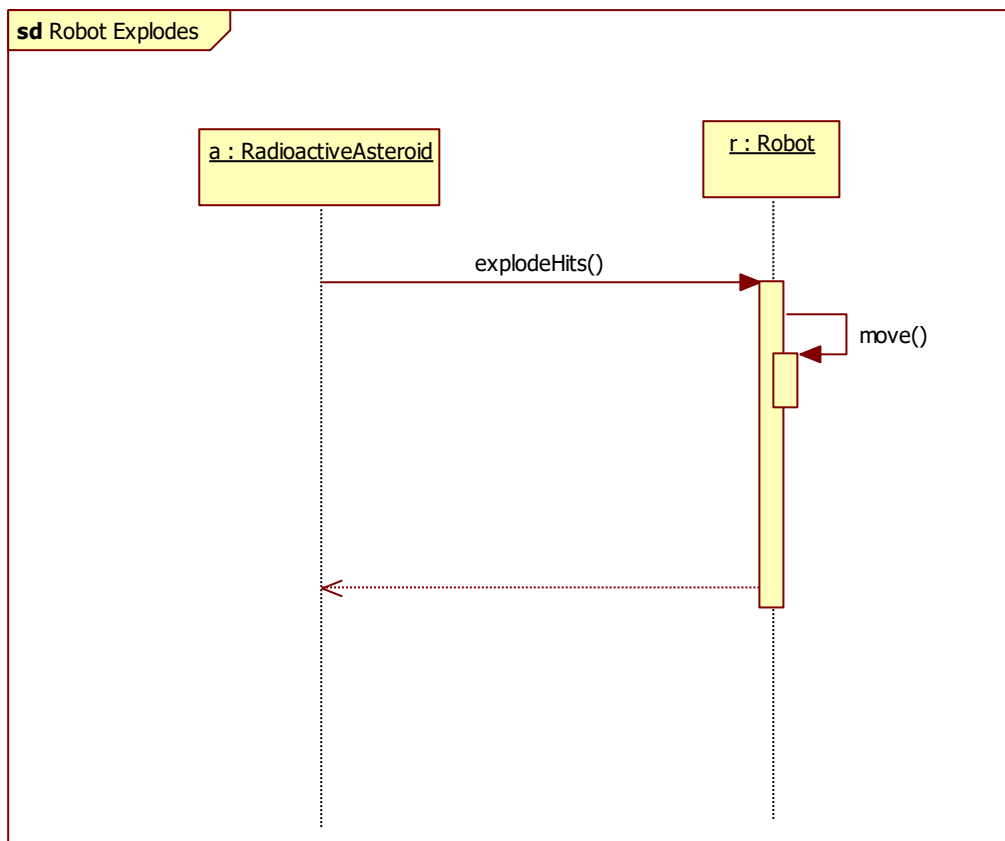
5.1.6. Settler Mines Non-Radioactive Asteroid / Siderler fördert nicht radioaktive Asteroide



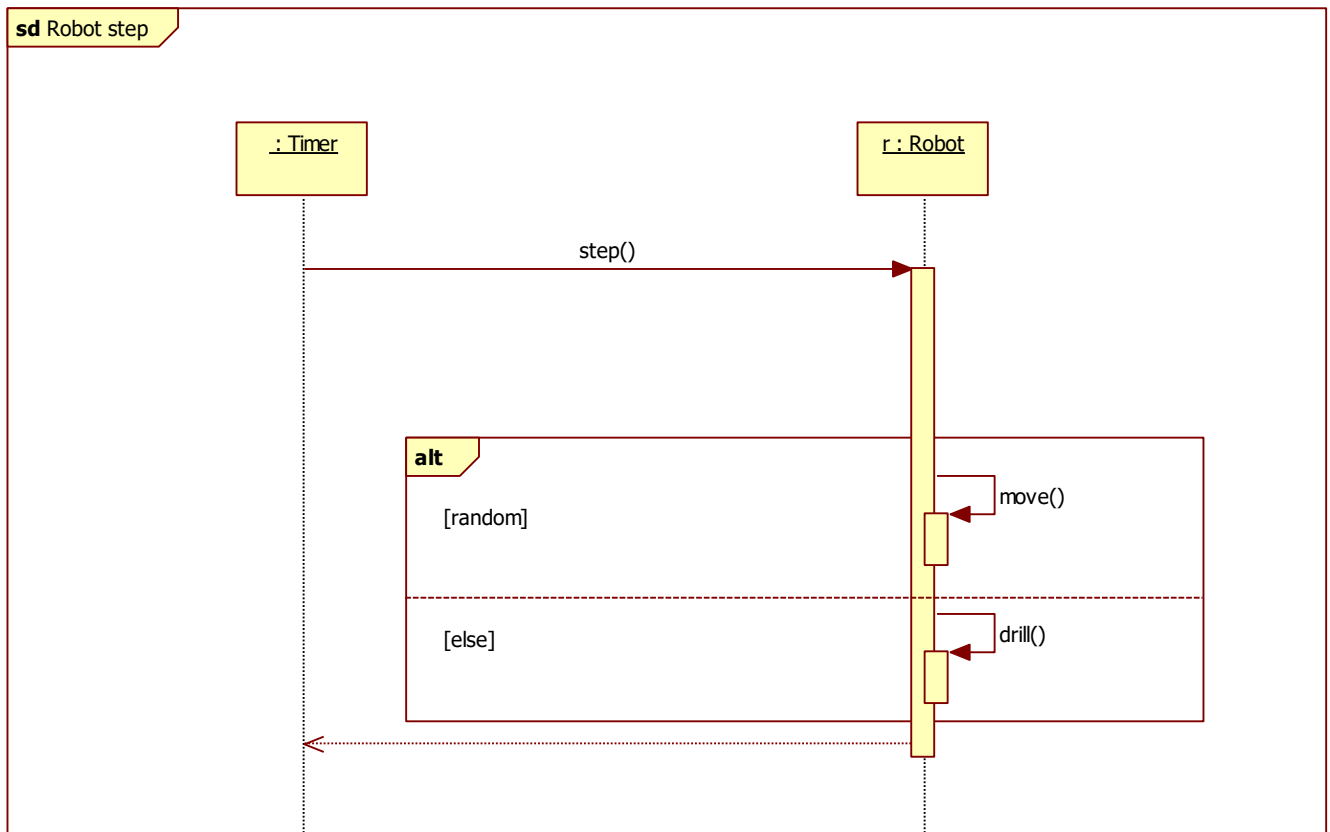
5.1.7. Settler Explodes / Siedler explodiert



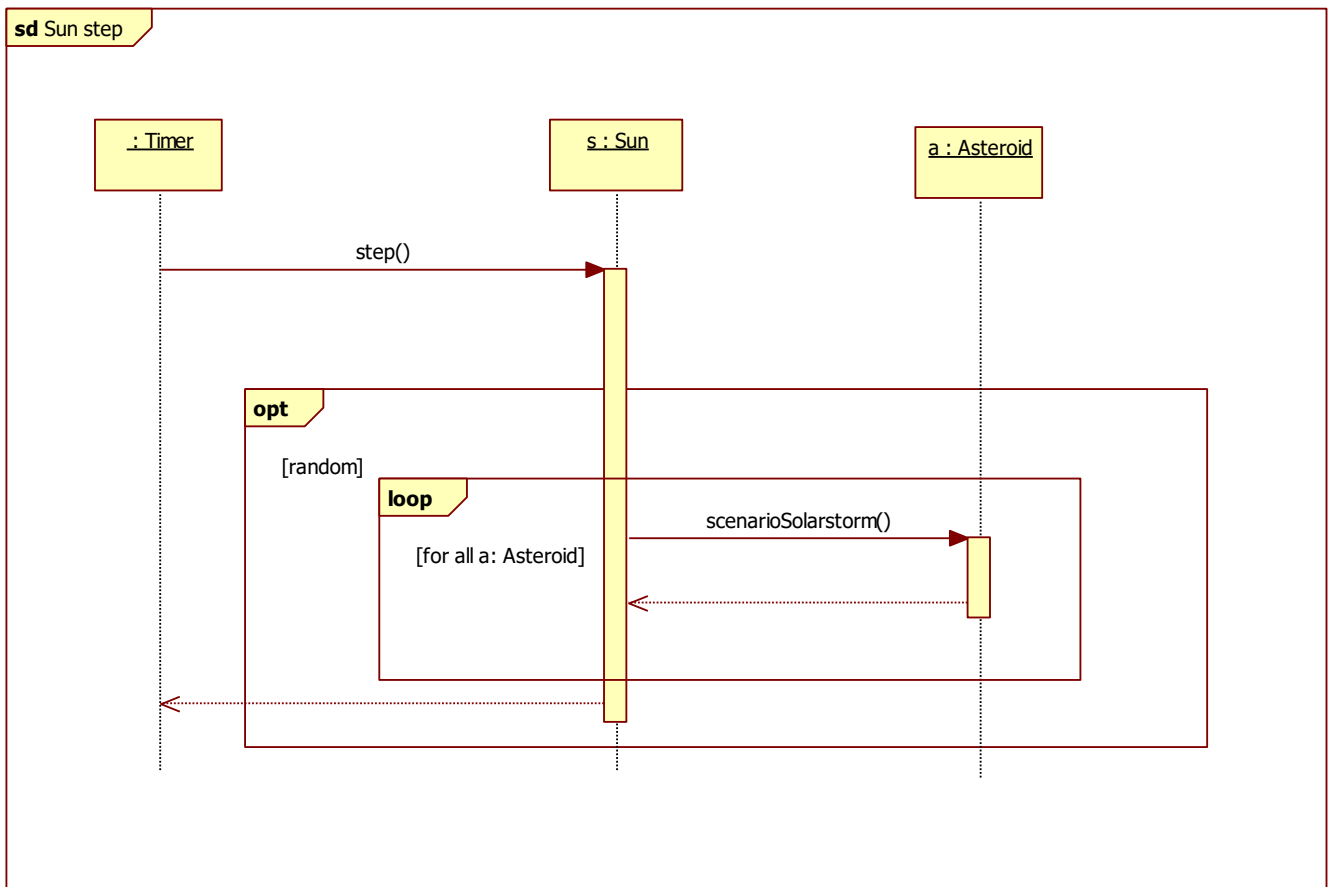
5.1.8. Robot Explodes / Robot explodiert



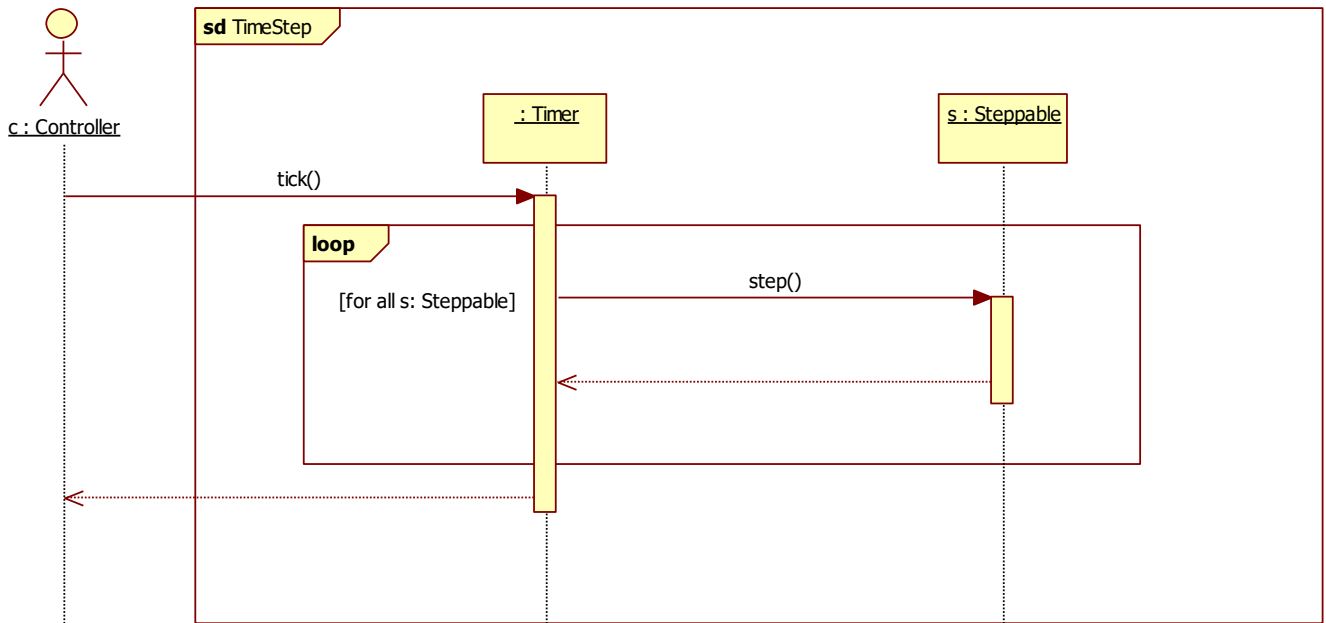
5.1.9. Robot Automatic Step / Robot automatischer Schritt



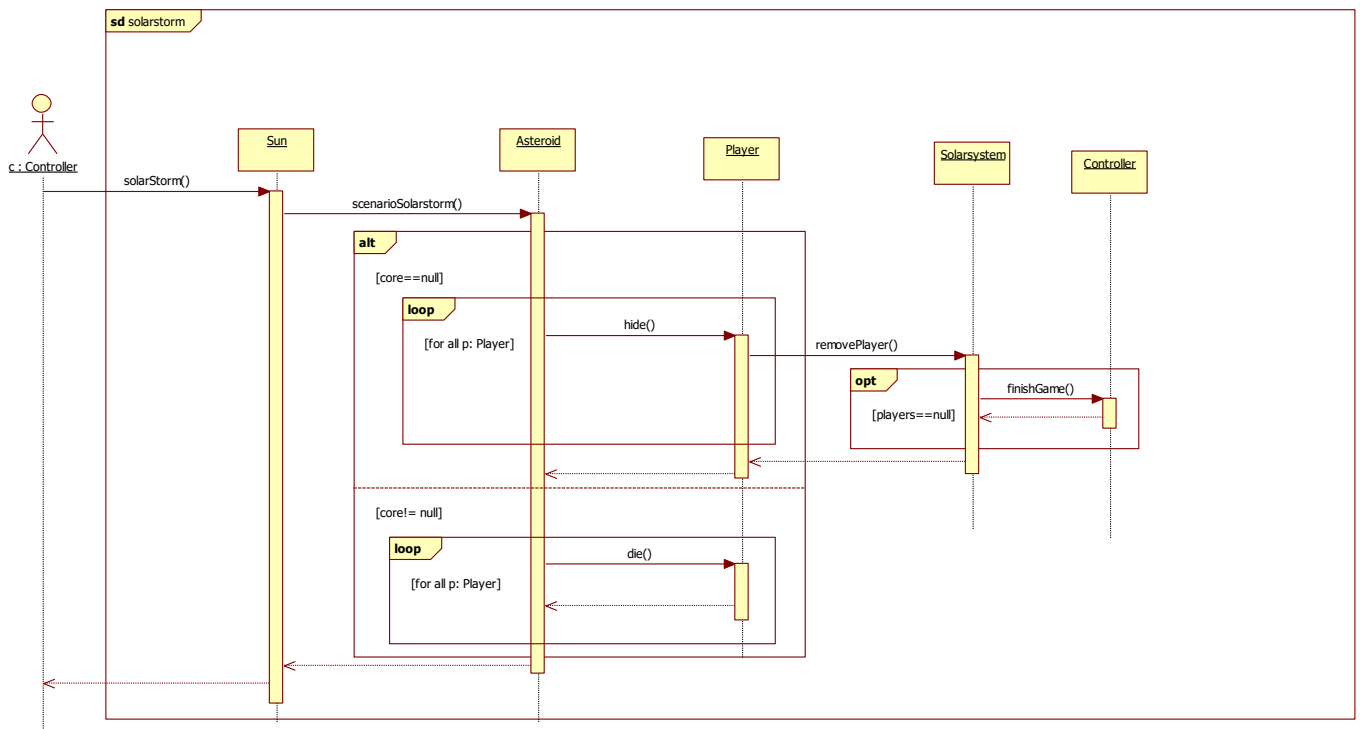
5.1.10. Sun Step / Sonne Schritt



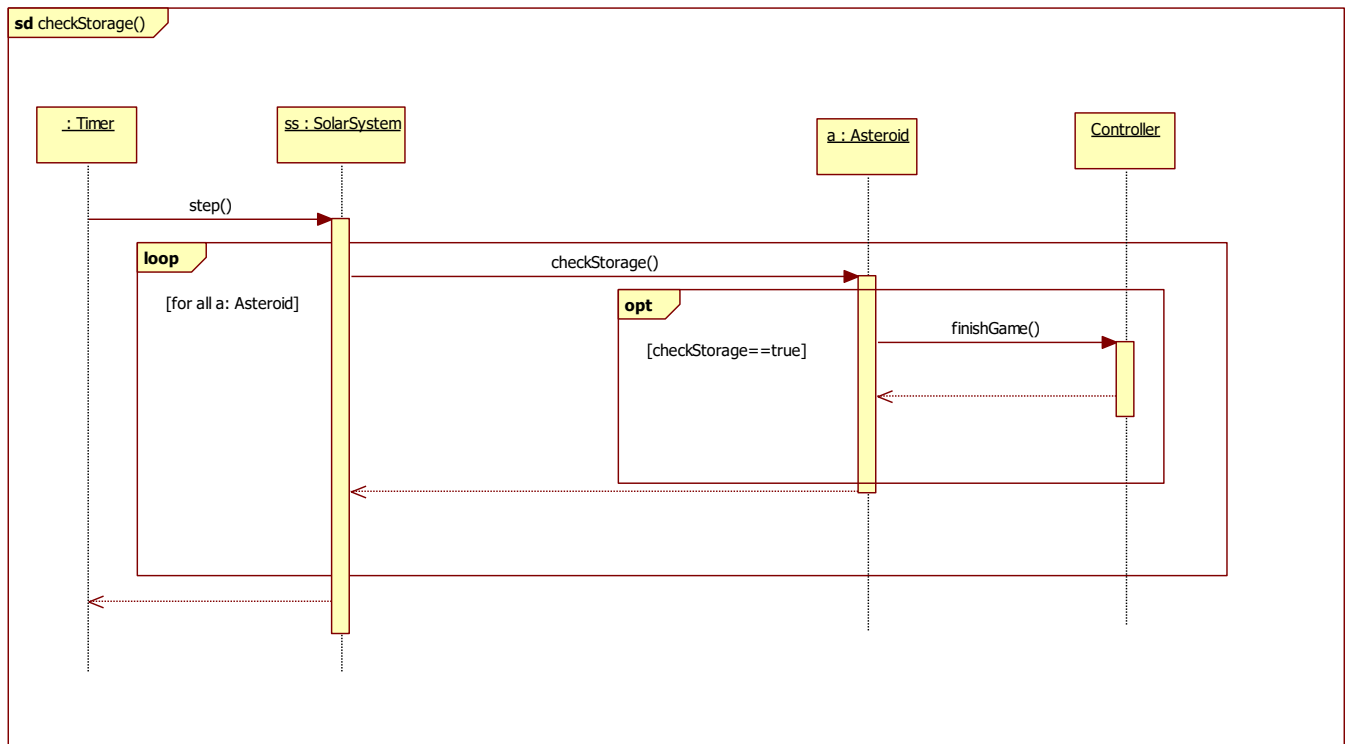
5.1.11. Time Step / Zeit Schritt



5.1.12. Solar Storm / Sonnenfaiir

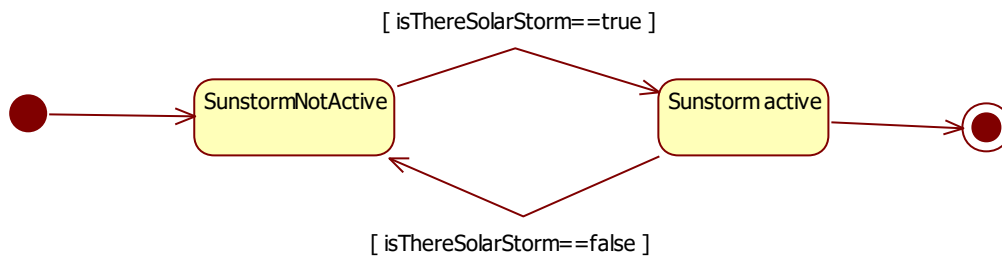


5.1.13. Check Storage (Kontrollieren ob der Spieler gewonnen haben oder nicht)

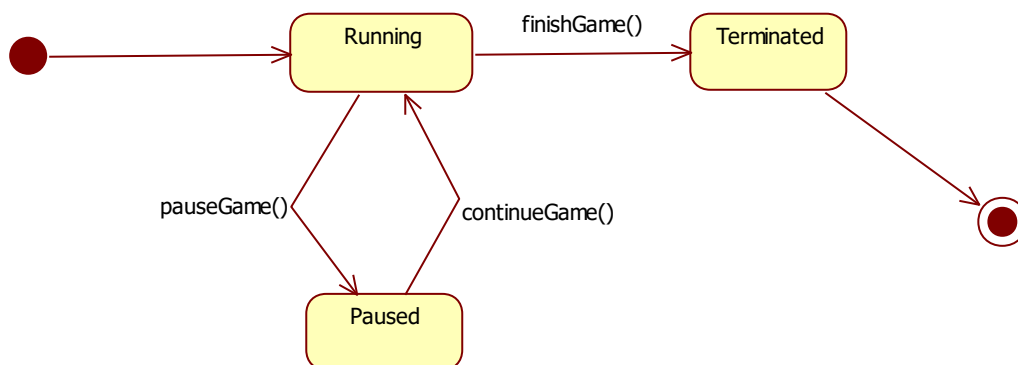


5.2 Zustandsdiagramme

5.2.1. Sunstorm active, Sonnenflair aktiv



5.2.2. Game States, Spiel Zustände



6 Aktivitäten Protokoll

Start Zeitpunkt	Aufwand in Stunden	Aktivität	Verweis
2020.10.28 16:00	6 Stunden	Durchlesen der Aufgabe, Nachdenken und Planung Anforderungen und Use-Case Diagramme erstellen	1, 2.1, 3.1, 3.2
2020.10.29 10:00	3 Stunden	Klannsendiagramm Planung	4.1
2020.11.02 17:00	4 Stunden	Klannsendiagramm Planung	4.1
2020.11.06 18:00	5 Stunden	Sequenzdiagramm Planung	5.1
2020.11.07 09:00	3 Stunden	Sequenzdiagramm Korrektur	5.1
2020.11.08. 14:00	2 Stunden	Dokumentation für Klassendiagramme	4.2
2020.11.09 10:00	4 Stunden	Dokumentation für Sequenzdiagramma, Zustandsdiagramme erstellung, andere Fehlerkorrekturen	4.1, 5.2

Arbeitsaufwand insgesamt in Stunden: 27 Stunden

SW Werkzeug verwendet für die Modellierung: WhiteStarUml
Andere Werkzeuge, und Ziel ihrer Anwendung: MicrosoftWord