

# **[ASTEROID MINING]**

**[TEAM NNUMBER 7]**

Supervisor:  
**Dr. Balla Katalin**

## **Members:**

[Abouzabady Mohamed]	[GJVKZQ]	[mohamedabozabady1@gmail.com]
[Kirolos Sawiris]	[Y8B5KD]	[Kirolossawiris@gmail.com]
[Oussama dhouioui]	[XFCBM7]	[Oussemadhououi04@gmail.com]
[Mohamed Amine]	[ARPMPB]	[mzaghdoudi01@gmail.com]
[Ahmed Mousa]	[EWLQXU]	[amousa@edu.bme.hu]
[MohammadParsa Kolivand]	[RLU15I]	[parsakolivand@edu.bme.hu]
[Mahdi Yahia]	[WF72QQ]	[mahdiyahia18@gmail.com]

[28.02.2022]

## 2. Requirements, project, functionality

### 2.1 Introduction

#### 2.1.1 Goal

*The goal of this document is to give an overview of the game. It aims to describe the game thoroughly as it explains how the game is made and how it is played. It shows all the ideas that were put together in order to create such a game.*

#### 2.1.2 Application domain

*The application is mainly a game that the user can play on any console for entertainment.*

#### 2.1.3 References

Problem Definition Webpage:

<https://www.iit.bme.hu/targyak/BMEVIIAB02/problem-definition>

#### 2.1.4 Document content

*The document will contain a description of the game. It will present its functionality and requirements. It contains big parts which are divided into minor and more detailed parts:*

2.2 Overview of the project: an overall view on the game

2.3 Requirement: contains a detailed description of the functional and non-functional requirement

2.4 Essential Use cases: contains use case diagram and explanations.

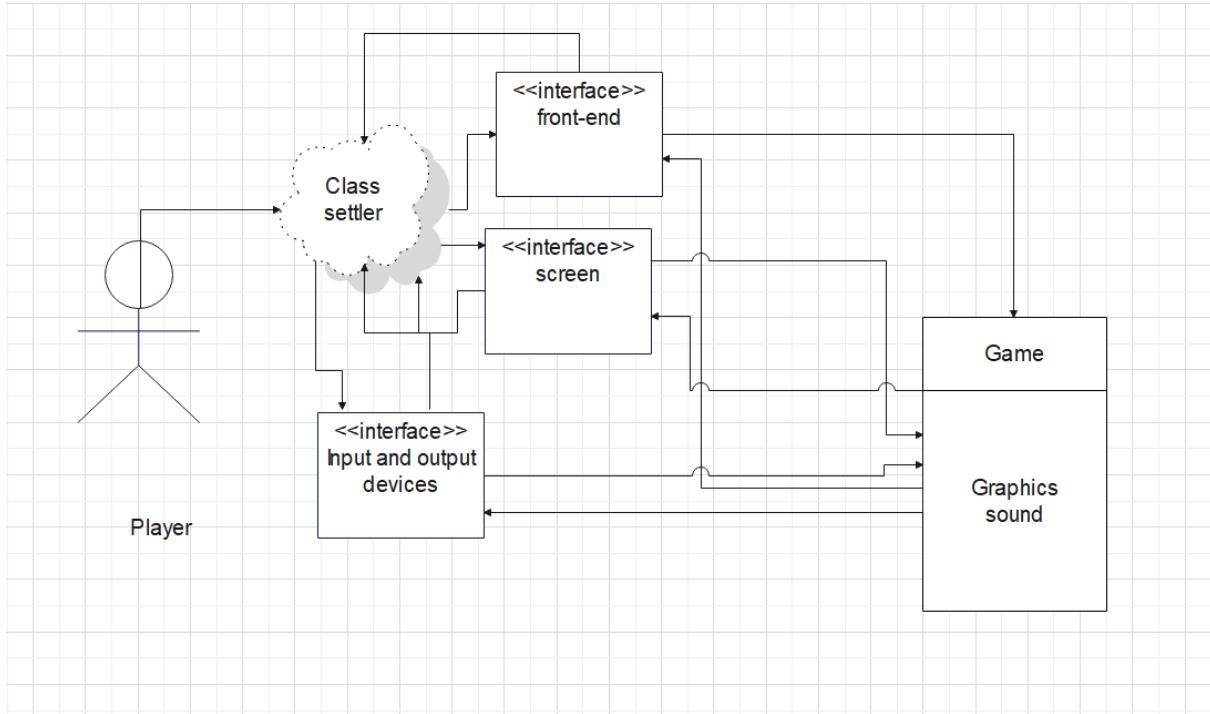
2.5 Glossary: explanation of hard terms related to the problem.

2.6 Project plans: explains the purpose of work and methodology of applying this work.

2.7 Protocol: Activity description and time spent on the project.

## 2.2 Overview

### 2.2.1 General overview



### 2.2.2 Functions

Exploiting the asteroids' resources is the main goal of this application. To do that, settlers have to build a space station in the asteroid belt. They can't transport any resources because it will be very expensive to do that, so the solution for that is settlers have to mine all resources from the asteroids.

The game is played by players who control the settlers. Settlers wander in the asteroid belt with single-person spaceships looking for resources.

Different kinds of asteroids are covered by rocks, whose depth can vary from asteroid to asteroid. There are many different types of resources:

Type 1: Important resources (water ice, iron, carbon, uranium, etc.) which can be found in the core of the asteroids.

Type 2: Some resources (like uranium) are highly radioactive.

Type 3: There can be hollow asteroids, whose core is empty.

The core of an asteroid is always homogeneous, which means that it is always made of a single kind of material.

In the game, there are several operations but one important thing is that a settler can do one operation in a single move. The operations are :

Traveling: When a settler travels, they go to a neighboring asteroid (an asteroid may have multiple hundreds of neighbors)

Drilling: When a settler drills, they deepen the hole in the mantle with one uni

Mining: they extract the resource from the core of the asteroid.

Other operations also like, building robots, building teleportation gates, etc.

Extracting the resource from the core of the asteroid is only possible if the mantle has been completely drilled through.

The next operation will be the resources transportation, The settler can carry 10 units as maximum resources due to the limited spaceships' capacity.

A hollow asteroid, however, can be filled with a unit of resource, counting as an operation.

Let's now talk about the hidden part which is the dangers which we can face:

The asteroid can explode and kill any settler on it When a fully drilled asteroid with a radioactive core is at perihelion. The only solution to mine the Radioactive material is when the asteroid is at aphelion.

The second danger is sun storms, Settlers can die because of this only if they do not hide in the core of a hollow asteroid.(the only way to survive is hiding there)

When a fully drilled asteroid with water ice in its core is at perihelion, the water ice sublimates (disappears).

The settlers can build autonomous robots controlled by artificial intelligence which only travel between asteroids and drill holes. They cannot mine because they are unable to transport things. Robots, however, can survive radioactive explosions, and in this case, they land on a neighboring asteroid. robots react the same way as settlers to sun storms, sun storms do damage robots if they don't hide in a hollow asteroid.

The settlers can build a pair of teleportation-gates using two units of Iron, a single unit of water ice, and a single unit of uranium.

The gates can be later deployed in the vicinity of the asteroid the settler is on. The two gates of a pair remain in contact for good, and entering either the traveler (settler, robot, etc) will be immediately transported to the other. The settlers can bring the freshly built gates with themselves, but at the same time, a single settler can only bring two gates.

To win the game players need to build a space station. The road to the building is to mine at least three units of each resource and they collect those materials on a single asteroid. The game can end, If all the settlers die, the players lose.

### 2.2.3 Users

User	Characteristics
<i>Player</i>	<ul style="list-style-type: none"> <li>-opens the game</li> <li>-Modify game settings</li> <li>-choose the number of asteroids and settlers</li> <li>-Start playing</li> <li>-end game(Win or Lose)</li> <li>-restart from the beginning</li> </ul>

## 2.3 Requirements

### 2.3.1 Functional requirements

ID	Description	Check	Priority	Source	Use-case	Comment
R01	<i>Players control the settlers.</i>		<i>Must have</i>	<i>problem description</i>	<i>Control Settlers</i>	
R02	<i>Settlers wander in the asteroid belt individually looking for resources.</i>		<i>Must have</i>	<i>problem description</i>	<i>Control Settlers</i>	
R03	<i>Asteroids are covered in rocks.</i>		<i>Should have</i>	<i>problem description</i>	<i>Control Asteroids</i>	
R04	<i>The depth of the rock mantle can vary from an asteroid to another.</i>		<i>Could have</i>	<i>problem description</i>	<i>Control Asteroids</i>	

R05	<i>The important resources are found in the core of the asteroids.</i>		Must have	problem description	Control Asteroids	
R06	<i>Some resources are highly radioactive.</i>		Should have	problem description	Control Asteroids	
R07	<i>There are hollow asteroids, whose core is empty.</i>		Could have	problem description	Control Asteroids	
R08	<i>The core of an asteroid is always homogeneous.</i>		Should have	problem description	Control Asteroids	
R09	<i>A settler can do one operation in a single move.</i>		Should have	problem description	Control Settlers	
R10	<i>When a settler travels, they go to a neighboring asteroid</i>		Should have	problem description	Control Settlers	
R11	<i>An asteroid may have multiple hundreds of neighbors</i>		Could have	problem description	Control Asteroids	
R12	<i>When a settler drills, they deepen the hole in the mantle with one unit.</i>		Should have	problem description	Control Settlers	
R13	<i>When a settler mines, they extract the resource from the core of the asteroid.</i>		Must have	problem description	Control Settlers	
R14	<i>Mining is only possible when the</i>		Should have	problem description	Control Settlers	

	<i>mantle has been completely drilled through.</i>					
R15	<i>A single settler can only carry 10 units of resources.</i>		Could have	problem description	Control Settlers	
R16	<i>A hollow asteroid, however, can be filled with a unit of resource.</i>		Could have	problem description	Control asteroids	
R17	<i>Filling an asteroid is counted as an operation.</i>		Should have	problem description	Control Settlers	
R18	<i>A fully drilled asteroid with a radioactive core explodes at perihelion.</i>		Should have	problem description	Control asteroids	
R19	<i>An exploding asteroid kills any settler on it.</i>		Should have	problem description	Control asteroids	
R20	<i>Radioactive material can only be mined when the asteroid is at aphelion.</i>		Should have	problem description	Control asteroids, Control settler	
R21	<i>Sometimes a sun storm reaches the asteroid belt.</i>		Should have	problem description	Control asteroids	
R22	<i>A sun storm kills the settler if it reaches him.</i>		Must have	problem description	Control asteroids, Control sun storm	
R23	<i>A settler survives the sun storm if he hides in a</i>		Must have	problem description	Control asteroids, Hide	

	<i>hollow asteroid.</i>					
R24	<i>The water ice sublimates, when a fully drilled asteroid containing it, is at its perihelion.</i>		<i>Could have</i>	<i>problem description</i>	<i>Control asteroids</i>	
R25	<i>The settlers can build autonomous robots controlled by artificial intelligence using a unit of iron, unit of carbon, and a unit of uranium.</i>		<i>Could have</i>	<i>problem description</i>	<i>Control Settlers</i>	
R26	<i>Robots can travel between asteroids.</i>		<i>Could have</i>	<i>problem description</i>	<i>Control robots, Control asteroids</i>	
R27	<i>Robots can drill holes</i>		<i>Could have</i>	<i>problem description</i>	<i>Control robots</i>	
R28	<i>Robots cannot mine because they are unable to transport things.</i>		<i>Could have</i>	<i>problem description</i>	<i>Control robots</i>	
R29	<i>Robots can survive radioactive explosions.</i>		<i>Could have</i>	<i>problem description</i>	<i>Hide</i>	
R30	<i>In case of a radioactive explosion robots land on a neighboring asteroid.</i>		<i>Could have</i>	<i>problem description</i>	<i>Control robots</i>	
R31	<i>Sun storms damaged robots.</i>		<i>Could have</i>	<i>problem description</i>	<i>Control robots</i>	

<i>R32</i>	<i>Robots hide in hollow asteroids to survive sun storms</i>		<i>Could have</i>	<i>problem description</i>	<i>Control robots</i>	
<i>R33</i>	<i>Settlers can build a pair of teleportation-gates using two units of iron, a single unit of water ice and a single unit of uranium.</i>		<i>Could have</i>	<i>problem description</i>	<i>Control Settlers</i>	
<i>R34</i>	<i>The gates can be later deployed in the vicinity of the asteroid the settler is on.</i>		<i>Could have</i>	<i>problem description</i>	<i>Control Settlers</i>	
<i>R35</i>	<i>The two gates of a pair remain in contact for good, and entering either the traveler will be immediately transported to the other.</i>		<i>Could have</i>	<i>problem description</i>	<i>Control Settlers</i>	

R36	<i>The settlers can bring the freshly built gates with themselves.</i>		<i>Could have</i>	<i>problem description</i>	<i>Control Settlers</i>	
R37	<i>A single settler can only bring two gates.</i>		<i>Could have</i>	<i>problem description</i>	<i>Control Settlers</i>	
R38	<i>If all the settlers die, the players lose.</i>		<i>Must have</i>	<i>problem description</i>	<i>Control settler</i>	
R39	<i>If settlers can mine at least three units of each resource and they collect those materials on a single asteroid, they can build the space station and the player wins the game.</i>		<i>Must have</i>	<i>problem description</i>	<i>Build space station, Control Settlers, Control asteroid</i>	

### 2.3.2 Resource requirements

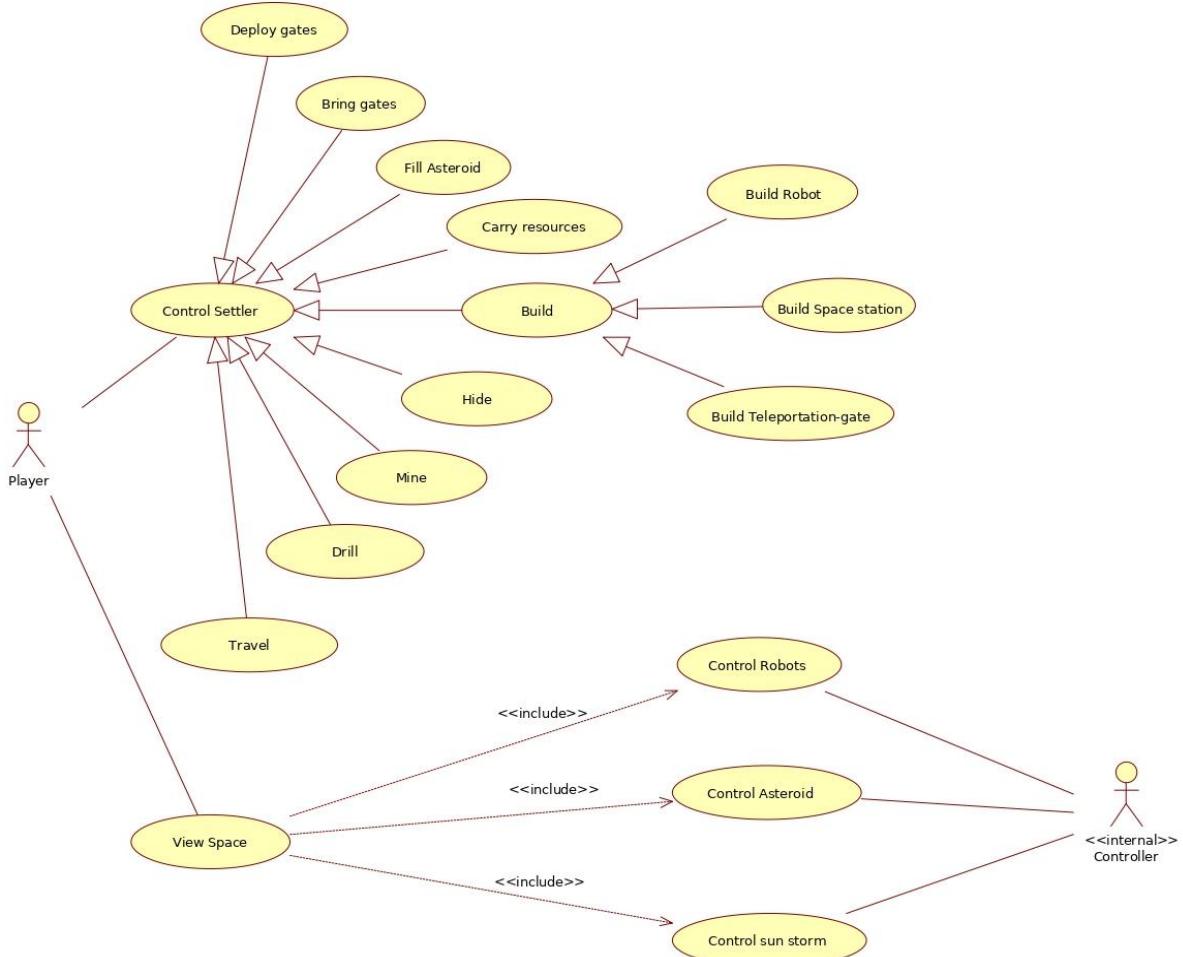
ID	Description	Check	Priority	Source	Comment
RR01	Java Development Kit	Implementation	Must have		
RR02	Modeling programs(WhiteStarUML)	Modeling	Must have		
RR03	MS Word, PDF	Documentation	Must have		
RR04	Minimum memory needed: 512MB Recommended memory: 1GB		Must have		
RR05	Recommended Free disk space: 1GB		Must have		
RR06	Recommended Processor Speed: 1GHz		Must have		

### 2.3.3 Non-functional requirements, Restrictions

ID	Description	Check	Priority	Source	Comment
NFR01	Reliability		Must have	The software must be able to perform all its functions under required conditions.	
NFR02	Efficiency		Must have	The software should use hardware resources that grant good performance without not necessary waste	
NFR03	Maintainability		Must have	The software has to be made in a way that makes it easily maintainable and modified.	

## 2.4 Essential use-cases

### 2.4.1 Use-case diagram



### 2.4.2 Use-case descriptions

<b>Use-case name</b>	<b>Control Settler</b>
<b>Short textual description</b>	<b>The player controls the settler through the Space.</b>
<b>Actors</b>	<b>Player</b>

<b>Use-case name</b>	<b>Travel</b>
<b>Short textual description</b>	<b>The player control which neighboring asteroid the settler will go to.</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>1. The player moves the settler to another asteroid.</b>

<b>Use-case name</b>	<b>Drill</b>
<b>Short textual description</b>	<b>The players control the settler to make or deepen the hole in the mantle with one unit.</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<p><b>1. The settler drills in the mantle with one unit.</b></p> <p><b>1.A.1. The settler can't drill in the mantle as the settler is fully drilled.</b></p> <p><b>1.B.1. The settler dies if he fully drilled asteroid with a radioactive core is at perihelion</b></p>

<b>Use-case name</b>	<b>Mine</b>
<b>Short textual description</b>	<b>The settler extracts the resources from the core of the asteroid.</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>1. The settler extracts the resources from the core of the asteroid.</b>

<b>Use-case name</b>	<b>Hide</b>
<b>Short textual description</b>	<b>The settler hides from the sun storm</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<p><b>1. The settlers go to the core of a hollow asteroid to hide from the sun storm.</b></p> <p><b>1.A.1. The settler can't go to the core of a hollow asteroid as it's not fully drilled</b></p>

<b>Use-case name</b>	<b>Build</b>
<b>Short textual description</b>	<b>The settler build objects</b>
<b>Actors</b>	<b>Player</b>

<b>Use-case name</b>	<b>Build Robot</b>
<b>Short textual description</b>	<b>The settler build a robot</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<p><b>1. The settler builds a robot as he has the required resources</b></p> <p><b>1.A.1. The settler cannot build a robot as he doesn't have the required resources.</b></p>

<b>Use-case name</b>	<b>Build Space station</b>
<b>Short textual description</b>	The settler builds a space station
<b>Actors</b>	Player
<b>Dialog, scenario</b>	<p>1. The settler built a space station as he managed to collect the required resources on a single asteroid.</p> <p>1.A.1. The settler couldn't build a space station as he didn't manage to collect the required resources on a single asteroid.</p>

<b>Use-case name</b>	<b>Build Teleportation-gate</b>
<b>Short textual description</b>	The settler build a teleportation-gate
<b>Actors</b>	Player
<b>Dialog, scenario</b>	<p>1. The settler builds a teleportation-gate as he has the required resources</p> <p>1.A.1. The settler cannot build a teleportation-gate as he doesn't have the required resources.</p>

<b>Use-case name</b>	<b>Deploy gates</b>
<b>Short textual description</b>	The settlers deploy the gates in the vicinity of the asteroid.
<b>Actors</b>	Player
<b>Dialog, scenario</b>	1. The settlers deploy the gates in the vicinity of the asteroid.

<b>Use-case name</b>	<b>Bring gates</b>
<b>Short textual description</b>	The settler brings gates.
<b>Actors</b>	Player
<b>Dialog, scenario</b>	<p>1. The settler brings a gate.</p> <p>1.A.1. The settler couldn't bring a gate as he already carry two gates.</p>

<b>Use-case name</b>	<b>Fill Asteroid</b>
<b>Short textual description</b>	The settlers fill the asteroid with resources.
<b>Actors</b>	Player
<b>Dialog, scenario</b>	<p>1. The settler fills the asteroid with one unit of resources.</p> <p>1.A.1. The settlers cannot fill the asteroid as it's not a hollow asteroid.</p> <p>1.B.1. The settler cannot fill the asteroid as it's not fully drilled.</p>

<b>Use-case name</b>	<b>Carry resources</b>
<b>Short textual description</b>	<b>The settler collects the resource</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<p><b>1. The settler collects the resource to move it from place to place.</b></p> <p><b>1.A.1. The settler can't collect the resource as he already carries 10 units of resources.</b></p>

<b>Use-case name</b>	<b>View Space</b>
<b>Short textual description</b>	<b>The player views the space</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<p><b>1. The system draws the current state of the space.</b></p> <p><b>2. The player observes the current state of the space</b></p>

<b>Use-case name</b>	<b>Control Robots</b>
<b>Short textual description</b>	<b>The Robots move in the space</b>
<b>Actors</b>	<b>Controller</b>
<b>Dialog, scenario</b>	<p><b>1. The Robot moves in space</b></p> <p><b>1.A.1. The Robot drills in the mantle of the asteroid.</b></p> <p><b>1.A.1.A.1. The Robot cannot drill in the mantle as it's fully drilled.</b></p> <p><b>1.B.1. The Robot go to the core of the asteroid to hide from the sun storm.</b></p>

<b>Use-case name</b>	<b>Control Asteroids</b>
<b>Short textual description</b>	<b>The asteroid explodes</b>
<b>Actors</b>	<b>Controller</b>
<b>Dialog, scenario</b>	<p><b>1. The asteroid explodes if it has a radioactive core and it is at perihelion</b></p>

<b>Use-case name</b>	<b>Control sun storm</b>
<b>Short textual description</b>	<b>The sun storm moves to reach the asteroid belt</b>
<b>Actors</b>	<b>Controller</b>
<b>Dialog, scenario</b>	<p><b>1. The sun storm moves to reach the asteroid belt and kills the settler.</b></p> <p><b>1.A.1. The sun storm moves to reach the asteroid belt and damages the robot.</b></p>

## 2.5 Glossary

*Settlers: the person that is currently settling in the space in our case.*

*Drill: making a hole in the asteroid mantle with a drill machine*

*Mine: An operation to collect the resources in the asteroids.*

*Resources: Materials that the settler is looking for to complete the building operations*

*Mantle: it is the layer of rocks that covers the asteroids.*

*Asteroid: a small rocky body orbiting the sun containing different resources*

*Belt: A region that surrounds the asteroid occupied by many asteroids.*

*Perihelion: Position of the asteroid nearest to the sun*

*Aphelion: Position which is the farthest from the sun.*

*Space Station: A station in space that needs resources to build.*

*Sun Storm: A storm that occurs from time to time which puts the life of settlers in danger.*

*Robot: A machine that the settler builds to help him drill the asteroids*

*Teleportation gate: A gate that helps the settler teleport from place to place.*

## 2.6 Project plan

The team consists of 7 members. On our first meeting, we put a plan for the project, once we receive the task it will be divided into sub-tasks, on the other side these tasks will be assigned to subgroups of the team.

Every week there will be at least 2 meetings with all team members and one meeting gathers the subgroup individuals as follows:

First Meeting (All Team Members): Dividing assignments into sub-tasks and assigning them to subgroups.

Second Meeting(Subgroup Individuals): It will be among each sub-group individuals to perform the assigned task. Each meeting is held according to the free time slots of the individuals

Third Meeting(All Team Members): Normally, this meeting will be after finalizing all the tasks to gather all the work and submit.

**Each subgroup can hold other meetings in case they need more time.**

**The subgroup is not fixed, all team members will get a chance to work on a subtask together.**

## 2.7 Protocol

Start (date & time)	Duration (hours)	Performer(s) name	Activity description
22.02.2022. 22:00	1 hour	All Team Members	It was the first meeting between the team members, the purpose was to put a plan for performing the assignments and divide this document into subtasks.
23.02.2022. 22:00	2 hours	All Team Members	In-person meeting to do the first draft to the sub-tasks, thus we worked in subgroups and decided the way of gathering the work together.
25.02.2022 13:00	3 hours	Abouzabady Mohamed Mousa Ahmed Sawiras Kirolos	Our task was mainly about the use case diagram, we read the game script multiple times in detail and also filled the table explaining each use case's possible scenarios.
25.2.2022 17:00	4 hours	Mohammadparsa Kolivand  Mahdi Yahya	The Requirement section has been accomplished by analyzing the Introduction and tasks entry. Use cases have been assigned to each related requirement.
24.2.2022 17:00	2 hours	Mohamed Amine Zaghoudi  Oussama Douihoui	We worked on the first two sections: Introduction and Overview. We had to go through the problem definition attentively as our task was mainly to give an overview of the game itself and its description which will determine our work on the other sections.

# **[ASTEROID MINING]**

**[TEAM NNUMBER 7]**

Supervisor:  
**Dr. Balla Katalin**

## **Members:**

[Abouzabady Mohamed]	[GJVKZQ]	[mohamedabozabady1@gmail.com]
[Kirolos Sawiris]	[Y8B5KD]	[Kirolossawiris@gmail.com]
[Oussama dhouioui]	[XFCBM7]	[Oussemadhououi04@gmail.com]
[Mohamed Amine]	[ARPMPB]	[mzaghdoudi01@gmail.com]
[Ahmed Mousa]	[EWLQXU]	[amousa@edu.bme.hu]
[MohammadParsa Kolivand]	[RLU15I]	[parsakolivand@edu.bme.hu]
[Mahdi Yahia]	[WF72QQ]	[mahdiyahia18@gmail.com]

[07.03.2022]

### 3. Analysis model – version 1

#### 3.1 Object catalog

##### 3.1.1 Settler

The whole game is based on settlers who are controlled by the players. Each move from the player is considered as an operation of the settler.

The operations are

Traveling: When a settler travels, they go to a neighboring asteroid (an asteroid may have multiple hundreds of neighbors)

Drilling: When a settler drills, they deepen the hole in the mantle with one unit

Mining: they extract the resource from the core of the asteroid.

Other operations also like, building robots, building teleportation gates, etc.

These latest operations can only be done after collecting the necessary resources.

In the other hand settlers can face some serious threats that may lead them to die,

For example, if settlers try to mine radioactive asteroids at perihelion, they die due to the explosion of the asteroid.

Add to that there are sun storms which can damage any settlers nearby if they don't hide inside hollow asteroids.

##### 3.1.2 Robots

The settlers can build autonomous robots controlled by artificial intelligence which only travel between asteroids and drill holes. . This operation requires one unit of iron, one unit of carbon, and one unit of uranium. Robots cannot mine because they are unable to transport things. Robots, however, can survive radioactive explosions, and in this case, they land on a neighboring asteroid. robots react the same way as settlers to sun storms, sun storms do damage robots if they don't hide in a hollow asteroid.

##### 3.1.3 Asteroids

Asteroids are an important entity in this game. An asteroid has the following attributes: depth, state, resource, isRadioactive,

The settler can extract resources, hide or die Inside them. Asteroids are distinguished by type, resource, and position,

If the asteroid is hollow the settler cannot collect any resources but it can hide inside it .

If the asteroid is radioactive and it is located at perihelion, then the asteroid will explode and so the settler will be dead (robots are not affected).

If there are none of the previous two conditions then the settler can collect the resource of this asteroid.

### 3.1.4 Teleportation Gates

The settlers have to collect two units of iron, one unit of water ice, and a single unit of uranium in order to build these teleportation gates.

The gates can be later deployed in the vicinity of the asteroid the settler is on. The two gates of a pair remain in contact for good, and entering either the traveler (settler, robot, etc) will be immediately transported to the other. The settlers can bring the freshly built gates with themselves, but at the same time, a single settler can only bring two gates.

### 3.1.5 Resources:

There are many different types of resources:

Important

resources (water ice, iron, carbon, uranium, etc.) which can be found in the core of the asteroids. (The core of an asteroid is always homogeneous, which means that it is always made of a single kind of material)

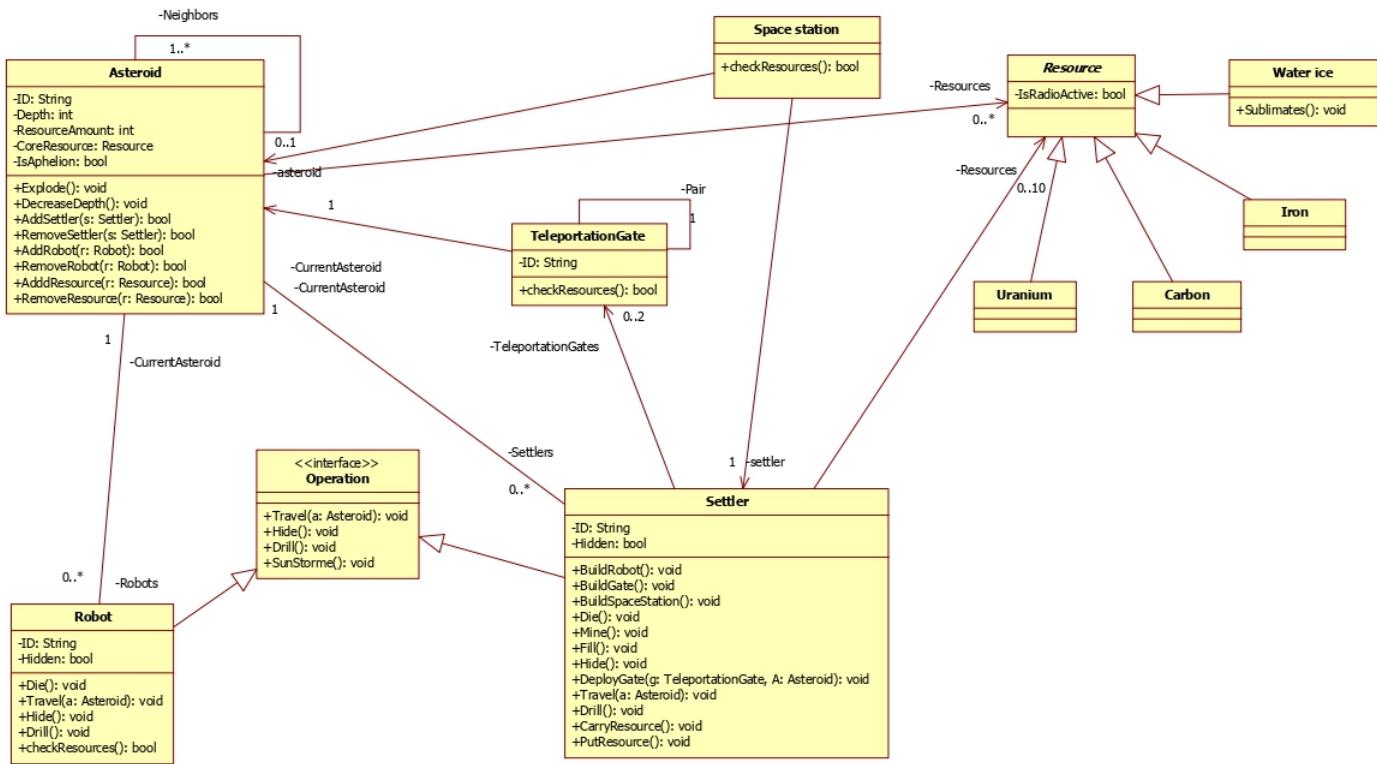
Some resources

(like uranium) are highly radioactive. (it can explode and kills the settler if mined at perihelion because it is radioactive)

They are mined only by settlers and can be transported whether by settlers or robots.

These resources are the key to building teleportation gates, robots, and space stations.

## 3.2 Static structure diagrams



### 3.1 Class description

#### 3.1.1 Class1. Settler

- Responsibility**

**It is one of the most significant classes in this game, since the settler is the main actor that represents the player's move, and actions directly.**

- Superclasses**

**None**

- Interfaces**

**Operations**

- Attributes**

- String ID:** each settler has a unique ID, and this represent the ID of the settler
- Bool Hidden:** it represents if the settler is hidden in case of the sunstorm or not and that affects the settler, if hidden is true then it will

**survive the sun storm. If it's false then the settler may explode and lose the game.**

- **ArrayList<Resource> resources resources:** *ArrayList* it contains a list of the class Resource that the settler manages to get.
- **ArrayList<Asteroid> asteroids:** It record the data of the current asteroid that the settler currently in.
- **ArrayList< TeleportationGate > TeleportationGates:** It has the data of the pair of gates that the settler built and uses.

#### Methods

- **Void buildRobot():** This method is used to build the robots, and it is one operation that the settler can do, after building the robot, it can help in performing other operations like traveling and drilling.
- **Void buildGate ():** It represents the operation of building a pair of gates using up two units of iron, a single unit of water ice and a single unit of Uranium
- **The gates can be deployed later, they are used for entering either the traveler is robot or a settler.**
- **Void buildSpaceStation():** It is the final purpose of the game, that if the settler reached will win the game. they can mine at least three unit of each resource and they collect those materials on a single asteroid, they can build the space station and the players win the game.
- **Void die():** The settler dies if it cannot survive the sun storm, or the radioactive materials.
- **Void mine():** It represents the operation of mining that the settler usually does to extract materials that will help in building the space station. Also settlers can only mine when the asteroid is at aphelion.
- **Void fill():** A hollow asteroid can be filled with a unit of resource, counting as an operation of the operations the settler can do.
- **Void hide():** A settler can only survive a sun storm if they hide in the core of a hollow asteroid. This, of course, is only possible if the mantle has been drilled through.
- **Void deployGate(TeleportationGate g, Asteroid a):** The gates can be deployed in the vicinity of the asteroid the settler is on, the method uses two parameters, the current asteroid and the gates which will be deployed.
- **Void Travel(Asteroid a):** The settler can travel between the asteroids, to mine and extract resources, the method uses the asteroid as a parameter
- **Void drill():** It is one of the operations that the settler can do, the settler can drill in the asteroids, they travel to so they can mine for resources.
- **Void carryResources():** A single settler can only carry 10 units of resources, that's the spaceships' capacity.
- **Void putResources():** The settler can put the carried resources in the desired place, to travel to other asteroids and move the resources out from the spaceship.

### 3.1.2 Class2. Asteroid

- **Responsibility**

*Settlers wander in the asteroid belt with single person spaceships looking for resources.*

- **Superclasses**

**None**

- **Interfaces**

**None**

- **Attributes**

- **String ID:** each asteroid has a unique ID, and this represents the ID of the asteroid.
- **int Depth:** each asteroid has a depth, The depth of this rock mantle can vary from asteroid to asteroid.
- **int ResourceAmount:** it stores the amount of resources of each asteroid that the settler can mine.
- **Resource CoreResource:** it represents the kind of resources that exist inside the asteroid.
- **Bool IsAphelion:** it represents the state of the asteroid if it is aphelion or not.
- **String SpaceStation:** it shows which space station, 0 or 1 max.
- **ArrayList<Resource> resources :**ArrayList it contains a list of the class Resource that the asteroid contains and available to be used 0 or many.
- **ArrayList<Robot> robots:** it shows the robots that are currently in a given asteroid.
- **ArrayList<Neighbors> neighbors :** It shows the data of the neighboring asteroids.
- **ArrayList<Settler> settlers :** It shows the data of the neighboring asteroids

- **Methods**

- **Void Explode():** . When a fully drilled asteroid with a radioactive core is at perihelion, the asteroid explodes and kills any settler on it.

- **Bool addSettler(Settler s): When a settler travel to the asteroid, it will be added through this method**
- **Bool removeSettler(Settler s): when the traveled settler leave the asteroid, traveling to other asteroid or if the settler died, it will be removed from the asteroid**
- **Bool addRobot(Robot r): When a robot travel to the asteroid, it will be added through this method**
- **Bool removeRobot(Robot r): when the traveled robot leaves the asteroid, traveling to another asteroid.**
- **Bool addResources(Resource r): it represents the available resources to be used in the asteroid, mining the resources and extracting them, will be added using this method since they will be available to be used, also carrying and putting the resources from different asteroids will be added through this method.**
- **Bool removeResource(Resource r): The resources will be carried into the spaceship to be moved to other asteroid, will be deducted directly from the current asteroid and added to the new one**

### 3.1.3 Class3. Robot

- **Responsibility**

*The settler built the robot and the robot can perform some common operations that the settler normally does.*

- **Superclasses**

**None**

- **Interfaces**

**Operation**

- **Attributes**

- **String ID: each robot has a unique ID, and this represents the ID of the robot.**
- **CurrentAsteriod: it shows the current asteroid that the robot exists in .**

- **Methods**

- **Void die(): Sun storms do damage robots unless they hide in a hollow asteroid.**

- **Void hide(): Robots hide in a hollow asteroid, so they can survive the sun storm**
- **Void drill(): The robots can drill holes in the asteroids although they can not mine.**
- **Void travel(Asteroid a): The robot travels between asteroids, to do a mission or to survive the radioactive explosions.**

### 3.1.4 Class4. Teleportation Gate

- **Responsibility**
- 
- The settler can build a pair of gates, they can be used for entering to travel or can be also deployed
- **Superclasses**
- None**
- **Interfaces**
- None**
- **Attributes**
  - **String ID:** each gate has a unique ID, and this represents the ID of the gate.
  - **Settler:** It shows which settler built and uses the pair of gates.
  - **Asteroid:** It store the data of the asteroid in which the gates exist.
- **Methods**
  - **Bool checkResources():** It checks that the current available resources are enough to build Teleportation gates, and returns true or false.

### 3.1.5 Class5. Resource

- **Responsibility**
- 
- The class contains all the information of the resources minded and available to be used in building the space station.

- **Superclasses**

**None**

- **Interfaces**

**None**

- **Attributes**

- **Bool IsRadioActive:** That refers if the resource is radioactive or not

- **Methods**

**None**

### **3.1.6 Class 6. WaterIce**

- **Responsibility**

*it is one of the main resources that is used to build the space station,*

- **Superclasses**

**Resource**

- **Interfaces**

**None**

- **Attributes**

**None**

- **Methods**

- **Void Sublimates():** When a fully drilled asteroid with water ice in its core is at perihelion, the water ice sublimates.

### **3.1.7 Class7. Carbon**

- **Responsibility**

*it is one of the main resources that is used to build the space station,*

- **Superclasses**

**Resource**

- **Interfaces**

**None**

- **Attributes**

- **Bool IsRadioActive: That refers if the resource is radio active or not**

- **Methods**

**None.**

### **3.1.8 Class8. Uranium**

- **Responsibility**

*it is one of the main resources that is used to build the space station,*

- **Superclasses**

**Resource**

- **Interfaces**

**None**

- **Attributes**

- **Bool IsRadioActive: That refers if the resource is radio active or not**

- **Methods**

**None.**

### **3.1.9 Class9. Iron**

- **Responsibility**

*it is one of the main resources that is used to build the space station,*

- **Superclasses**

**Resource**

- **Interfaces**

**None**

- **Attributes**

- **Bool IsRadioActive: That refers if the resource is radioactive or not**

- **Methods**

**None**

### 3.1.10            *Class10. Operation*

- **Responsibility**

*It is the base interface that is implemented in two classes (Settler, Robot). It contains the common operations between the settler and the robot.*

- **Superclasses**

**None**

- **Interfaces**

**None**

- **Attributes**

**None**

- **Methods**

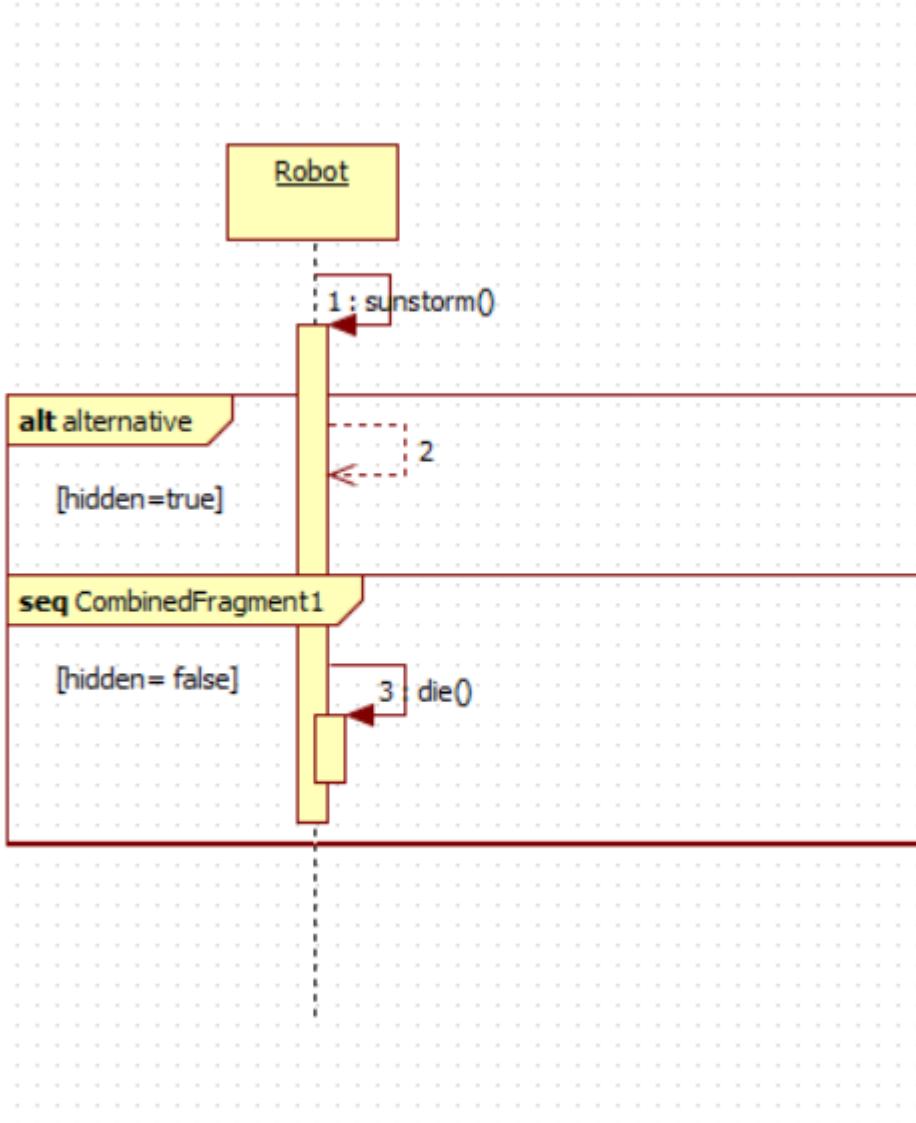
- **Void sunStorm(): Sun storms do damage robots unless they hide in a hollow asteroid and also it harm the settler and that may result to make him die, unless the settler hide too**
- **Void hide(): Robots hide in a hollow asteroid, so they can survive the sun storm, settlers hide as well**
- **Void drill(): The robots can drill holes in the asteroids although they can not mine, but settlers can drill holes and mine as well.**
- **Void travel(Asteroid a): The robot travels between asteroids, to do a mission or to survive the radioactive explosions, settlers can travel to the neighboring asteroids.**

### 3.1.11 Class11. Space Station

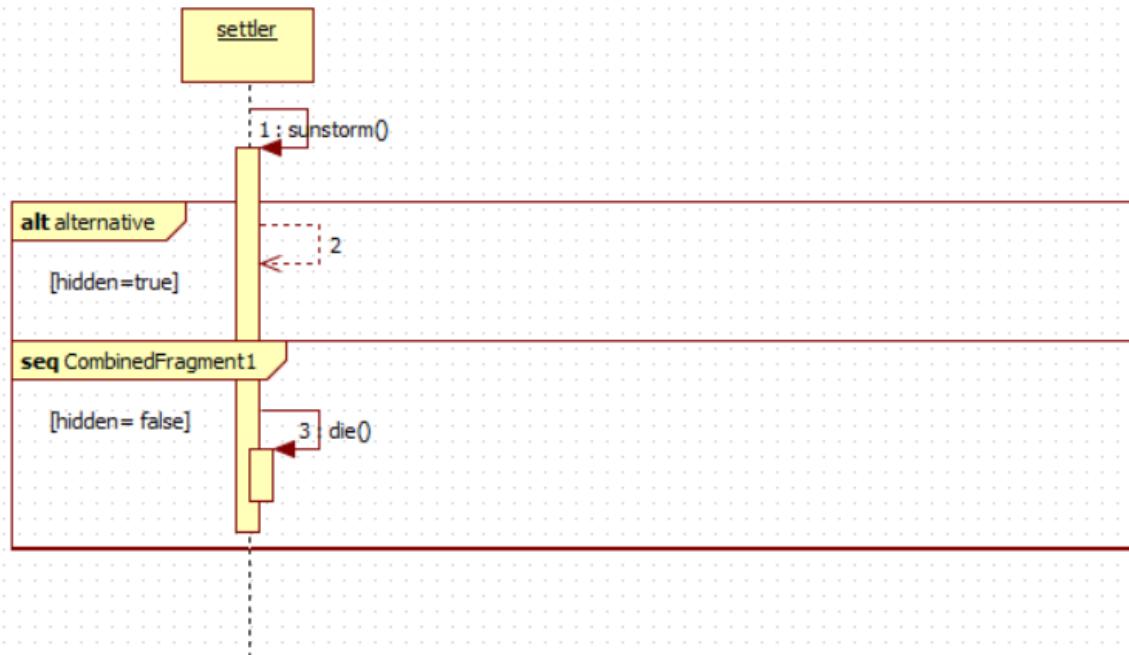
- **Responsibility**
  - It is the space station that the settlers are trying to extract enough resources to build.
- **Superclasses**
  - None**
- **Interfaces**
  - None**
- **Attributes**
  - None**
- **Methods**
  - **Bool checkResources(): It checks that the current available resources are enough to build a space station, and returns true or false.**

### 3.3 Sequence diagrams

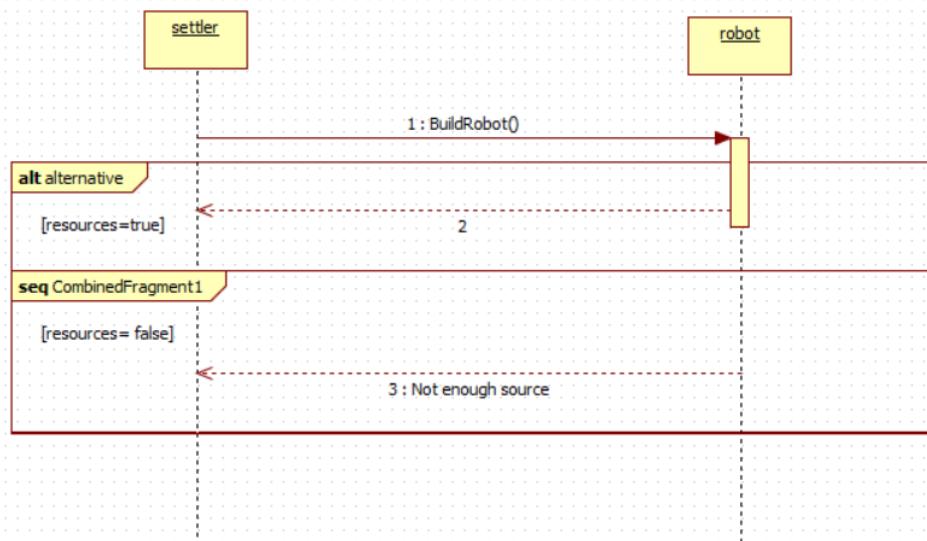
1/ Robots die:

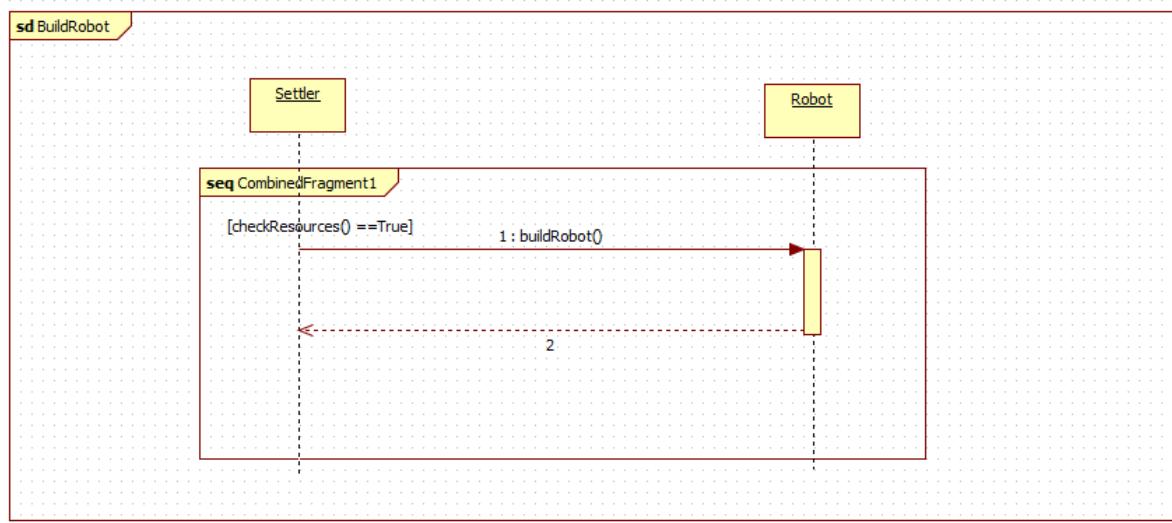


2/ settlers die:

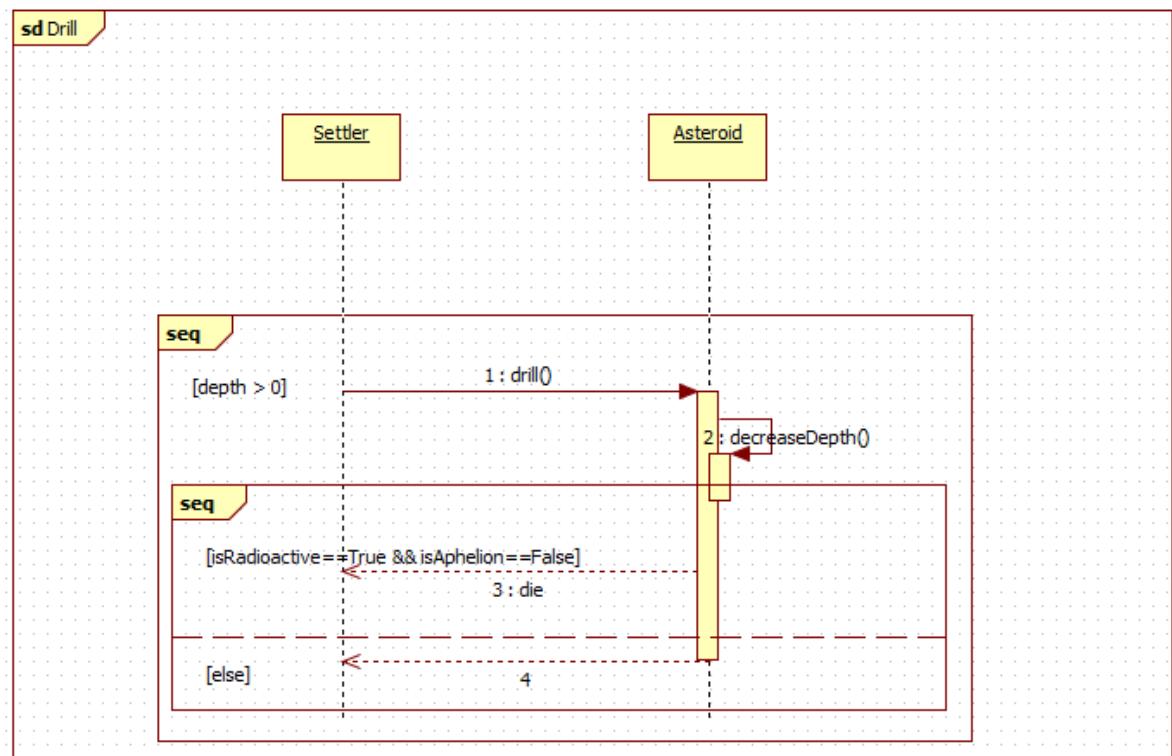


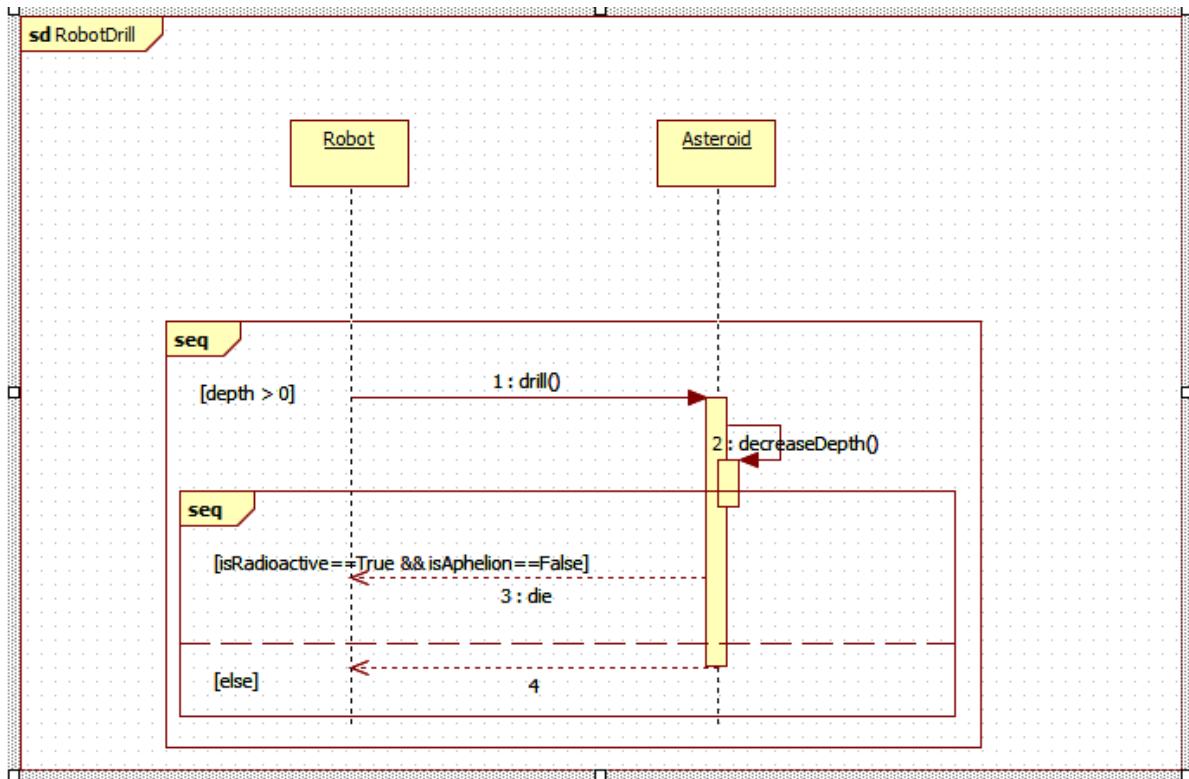
### 3/ Build robot :

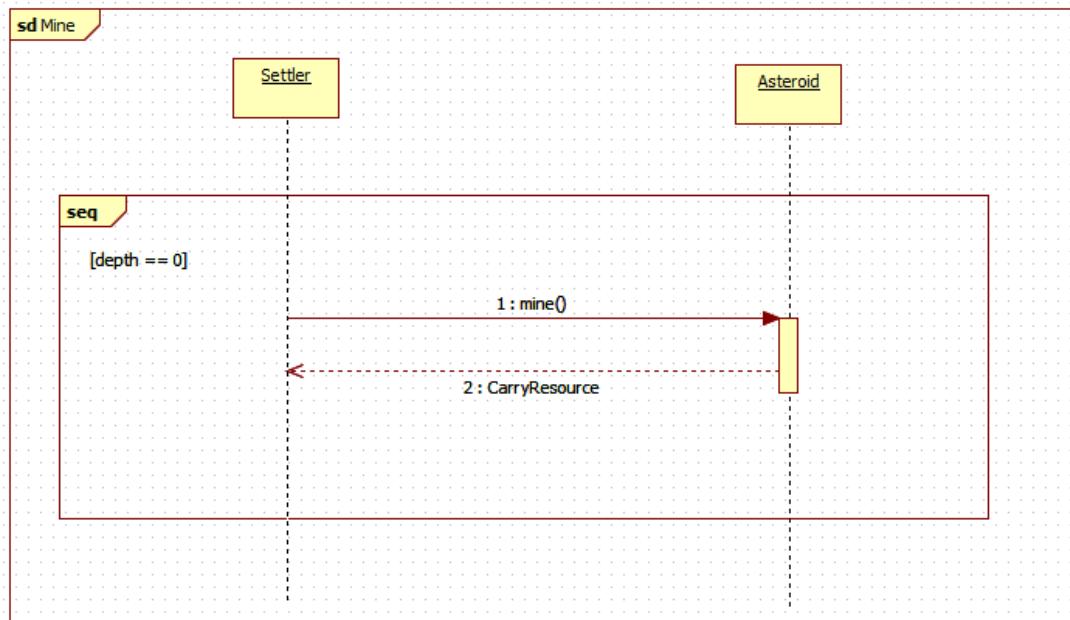
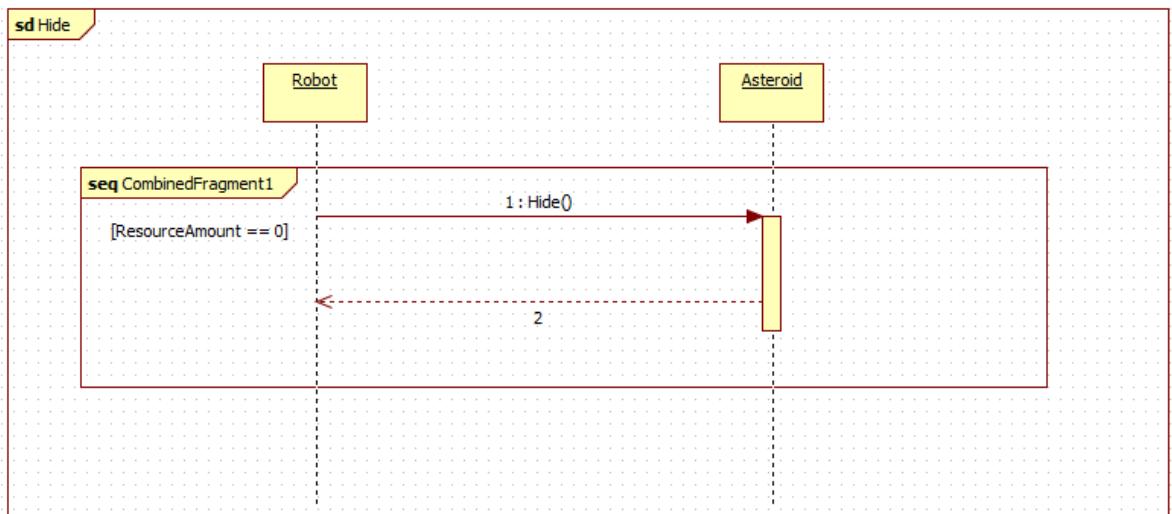


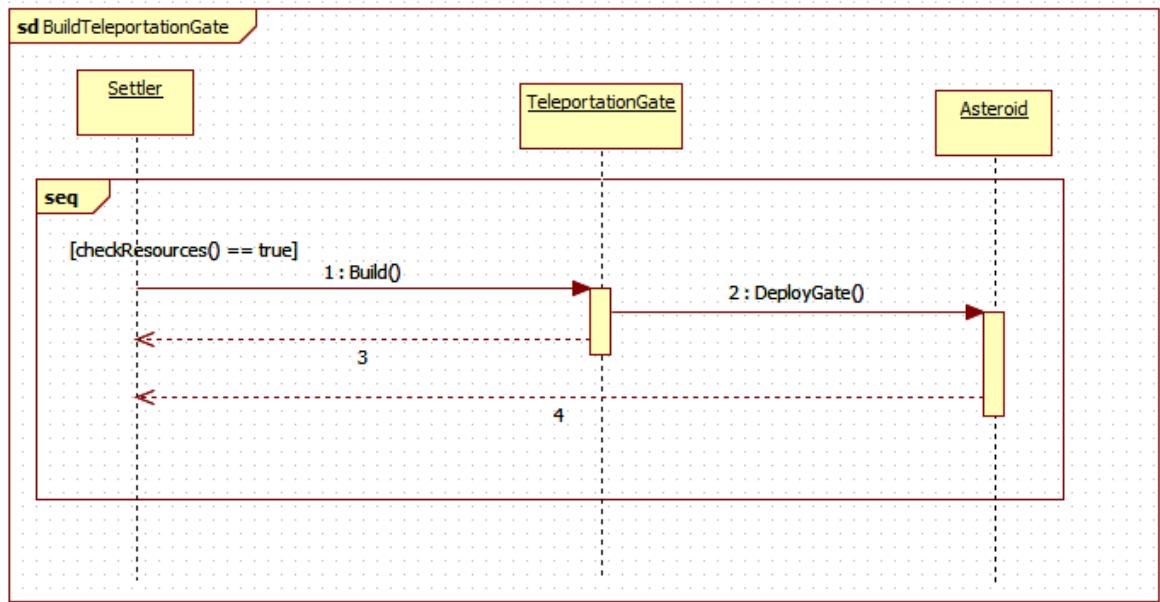
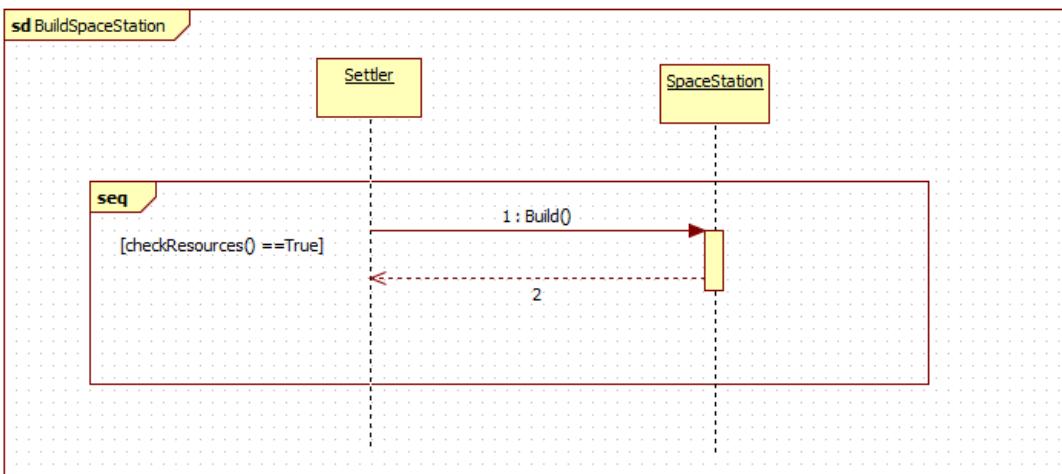


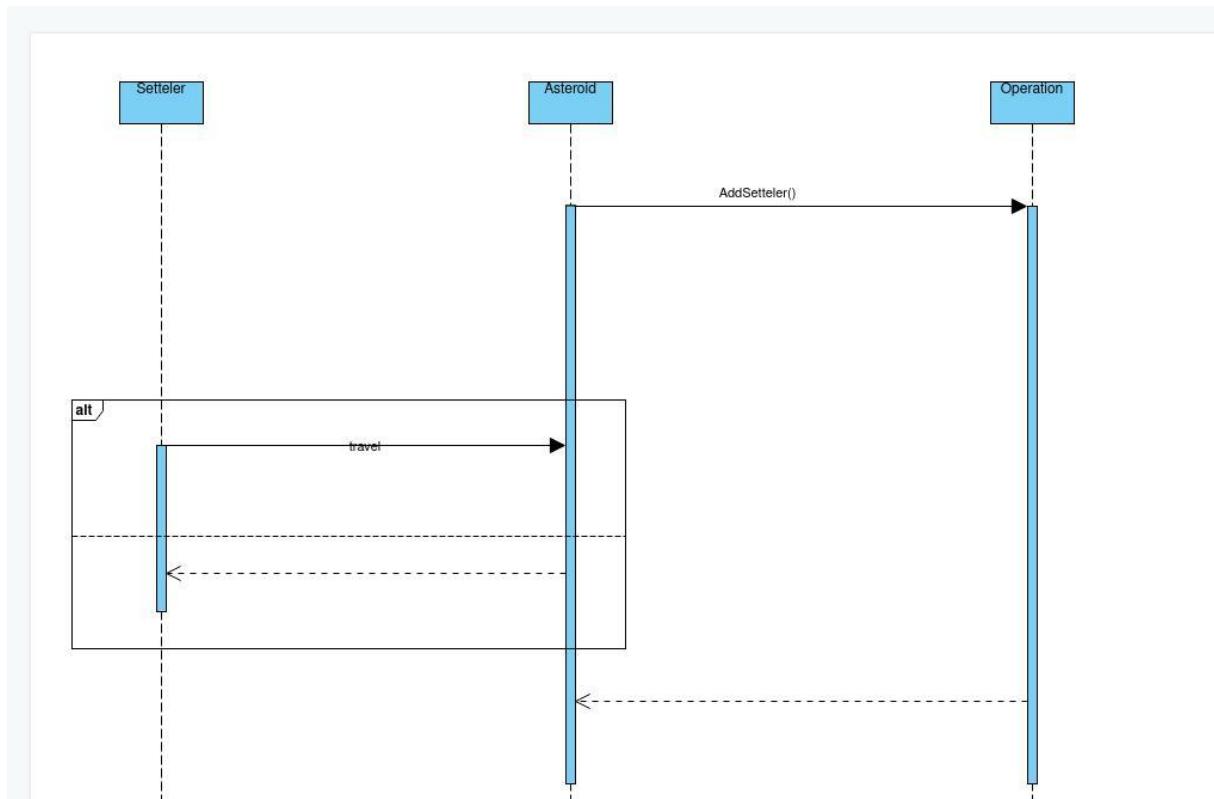
## 4/ Drill :



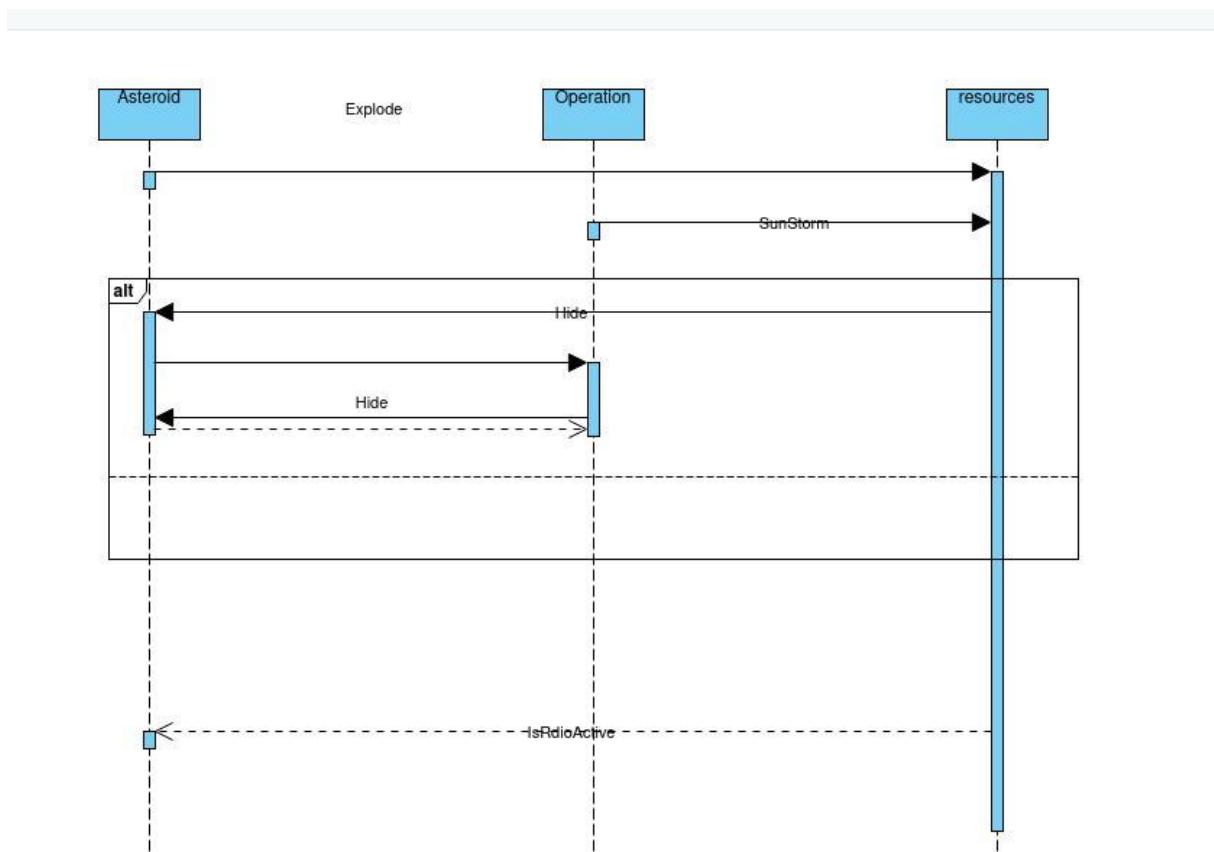
**5/ Robot Drill:**

**6/ Mine:****7/ Hide:**

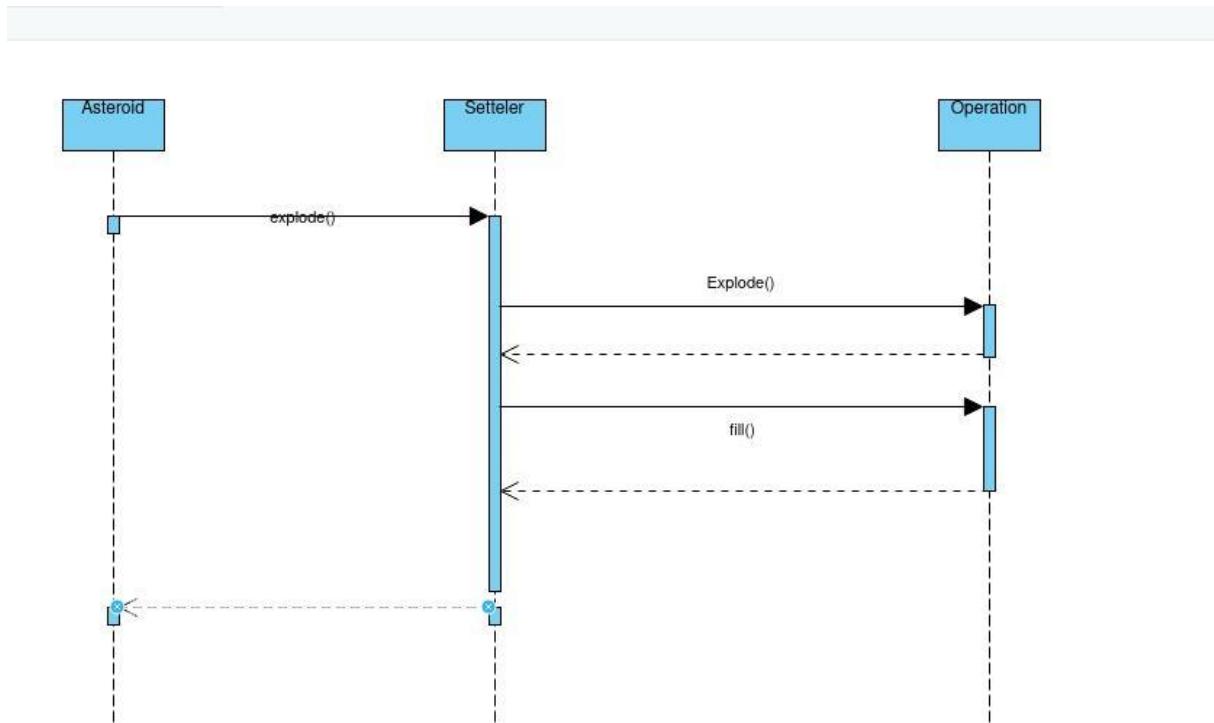
**8/ Build Teleportation Gate:****9/ Build Space Station:****10/ Travel:**



11/Hide:

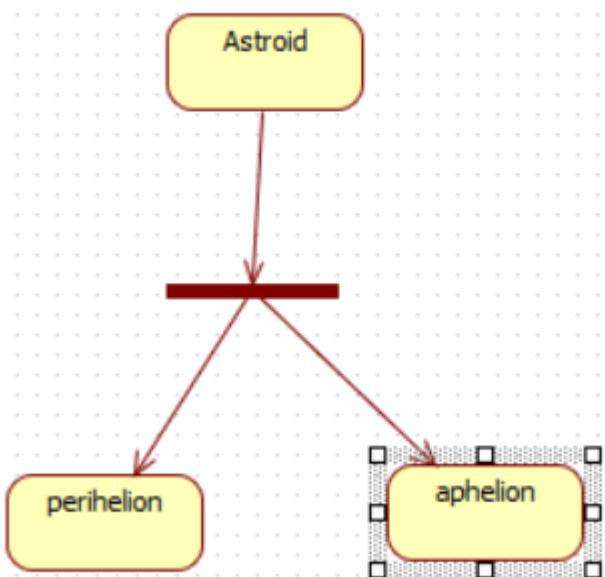


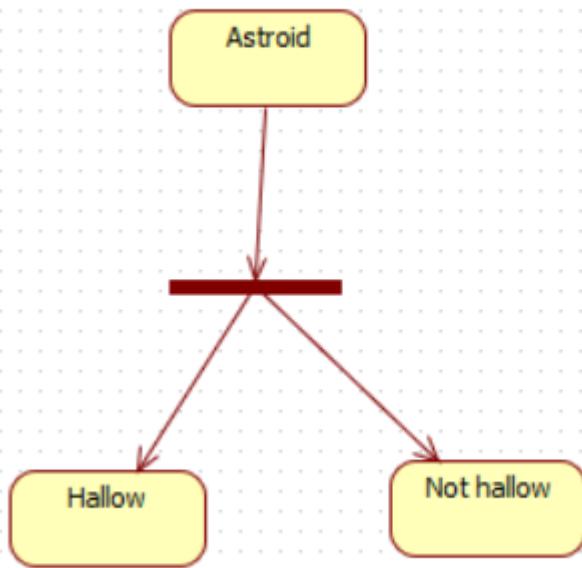
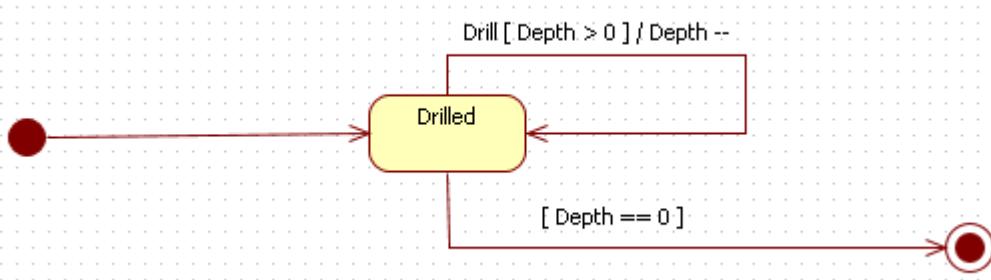
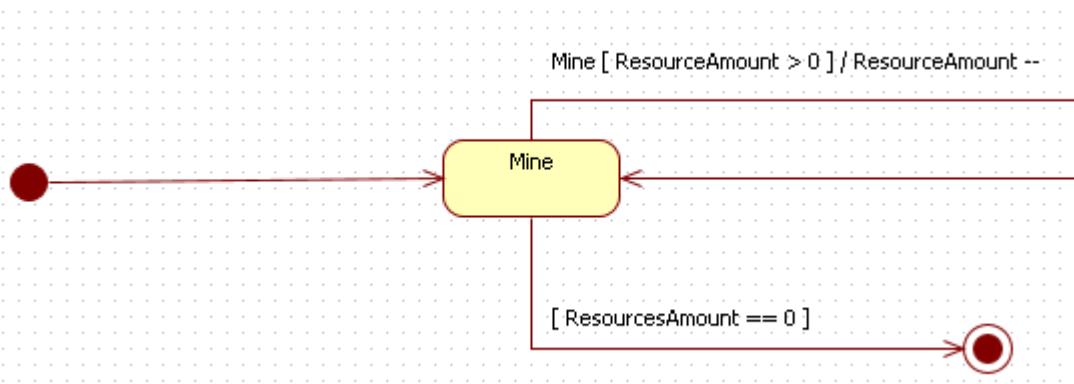
12/ Explode:



### 3.4 State-charts

Astroid state:



**Asteroid hallow or not:****Drill:****Mine:**

### 3.5 Protocol

Start (date & time)	Duration (hours)	Performers name	Activity description
4.03.2022 21:00	2 hours	dhououi oussama	Object catalog
5.03.2022 21:00	4 hours	dhououi oussama Parsa Mahdi Amine	sequence diagrams and state charts
04.03.2022 14:00	10 hours	Kirolos Sawiris Mousa Ahmed	Static structure diagrams(Class diagram)
05.03.2022 11:30	4 hours	Abouzabady Mohamed	Class description
06.03.2022 22:00	2 hours	Kirolos Sawiris	2 state charts, editing the class diagram

# **[ASTEROID MINING]**

**[TEAM NNUMBER 7]**

Supervisor:  
**Dr. Balla Katalin**

## **Members:**

[Abouzabady Mohamed]	[GJVKZQ]	[mohamedabozabady1@gmail.com]
[Kirolos Sawiris]	[Y8B5KD]	[Kirolossawiris@gmail.com]
[Oussama dhouioui]	[XFCBM7]	[Oussemadhououi04@gmail.com]
[Mohamed Amine]	[ARPMPB]	[mzaghdoudi01@gmail.com]
[Ahmed Mousa]	[EWLQXU]	[amousa@edu.bme.hu]
[MohammadParsa Kolivand]	[RLU15I]	[parsakolivand@edu.bme.hu]
[Mahdi Yahia]	[WF72QQ]	[mahdiyahia18@gmail.com]

## 4. Analysis model – version 2

### 4.1 Object catalog

#### 4.1.1 Traveler:

This is a super abstract class that cannot be instantiated. It was created to avoid code duplication in both robot and settler as they have common attributes like ID and Hidden and functions like travel, hide, drill and Sunstorm.

#### 4.1.2 Settler:

The whole game is based on settlers who are controlled by the players. Each move from the player is considered as an operation of the settler. The operations are Traveling: When a settler travels, they go to a neighboring asteroid (an asteroid may have multiple hundreds of neighbors) Drilling: When a settler drills, they deepen the hole in the mantle with one unit Mining: they extract the resource from the core of the asteroid. Other operations also like, building robots, building teleportation gates, etc. These latest operations can only be done after collecting the necessary resources. On the other hand, settlers can face some serious threats that may lead them to die. For example, if settlers try to mine radioactive asteroids at perihelion, they die due to the explosion of the asteroid. Add to that there are sun storms which can damage any settlers nearby if they don't hide inside hollow asteroids.

#### 4.1.3 Robot:

Autonomous robots are controlled by artificial intelligence. Robots can only perform two operations, the first one is traveling between asteroids, and the second one is deepening the hole in the mantle with one unit. In addition to this, we also state that robots can survive radioactive material by landing on a neighboring asteroid. Robots are created by settlers, this operation requires one unit of iron, one unit of carbon, and one unit of uranium. Sun storms do damage robots also, only in case they hide in hollow asteroids. Also, robots can move between two asteroids using teleportation gates.

#### 4.1.4 Resources: (Uranium, Carbon, Iron, Water ice)

Is a super abstract class that cannot be instantiated. It was created to avoid code duplication in all four resources. It contains the IsRadioActive attribute.

There are many different types of resources: Important resources (water ice, iron, carbon, uranium, etc.) which can be found in the core of the asteroids. (The core of an asteroid is always homogeneous, which means that it is always made of a single kind of material) Some resources (like uranium) are highly radioactive. (it can explode and kill the settler if mined at perihelion because it is radioactive) They are mined only by settlers and can be transported whether by settlers or robots.

Settlers can do the following with resources:

- Hide resources inside hollow asteroids
- Transport at most 10 resources at most per single settler
- Build robots using: one unit of iron, one unit of carbon, and one unit of uranium.
- Build teleportation gates using: two units of iron, one unit of water ice, and a single unit of uranium.

-Build space station: if three units are collected for each resource, and all of them are successfully transported to a single asteroid.

#### 4.1.5 Asteroids

Asteroids are an important entity in this game. An asteroid has the following attributes: depth, state, resource, isRadioactive, The settler can extract resources, hide or die Inside them. Asteroids are distinguished by type, resource, and position. If the asteroid is hollow the settler cannot collect any resources but it can hide inside it . If the asteroid is radioactive and it is located at perihelion, then the asteroid will explode and so the settler will be dead (robots are not affected). If there are none of the previous two conditions then the settler can collect the resource of this asteroid.

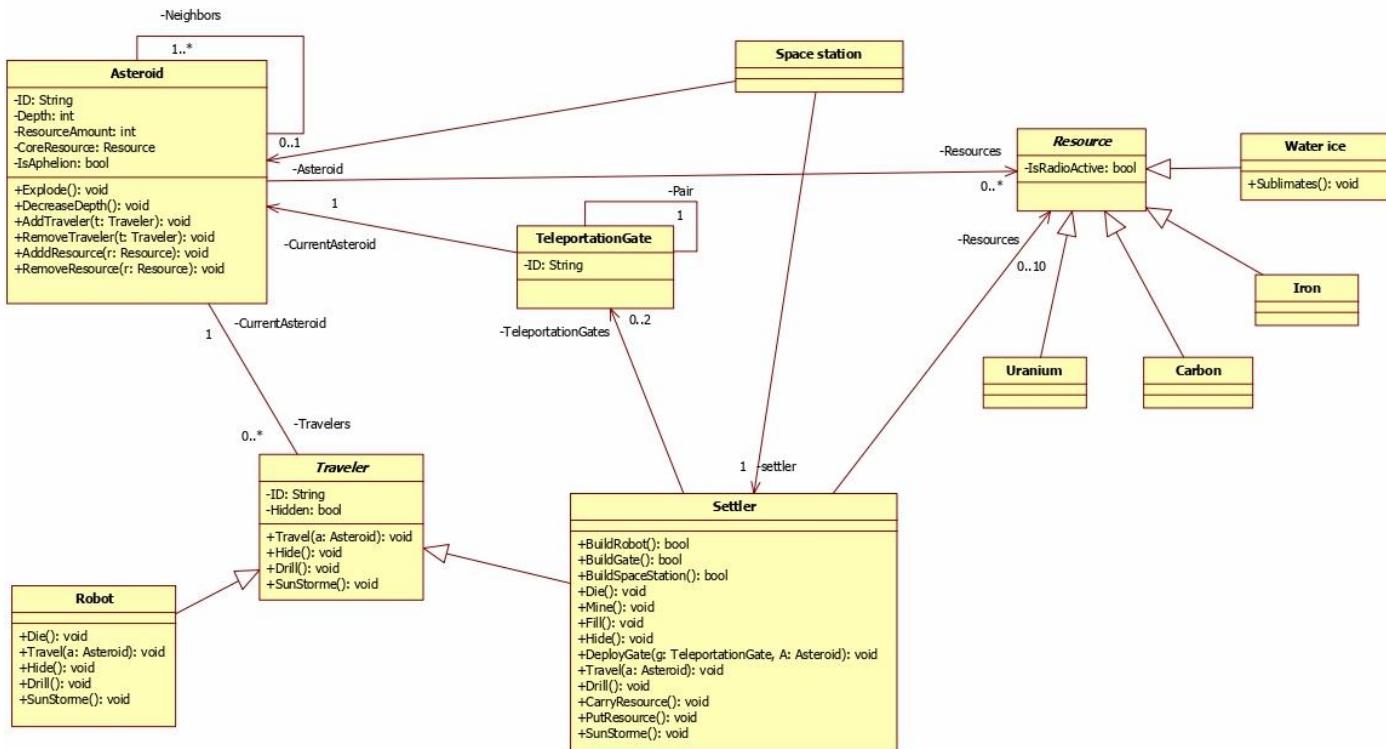
#### 4.1.6 Teleportation Gates

The settlers have to collect two units of iron, one unit of water ice, and a single unit of uranium in order to build these teleportation gates. The gates can be later deployed in the vicinity of the asteroid the settler is on. The two gates of a pair remain in contact for good, and entering either the traveler (settler, robot, etc) will be immediately transported to the other. The settlers can bring the freshly built gates with themselves, but at the same time, a single settler can only bring two gates.

#### 4.1.7 Space station

To win the game the space station needs to be created. If the settler manages to collect three units from each resource and transport all of them successfully to a single asteroid. So we call the function buildspacestation of the settler and then the space station is built.

## 4.2 Static structure diagrams



## 4.3 Class description

### 1. Settler

- **Responsibility**

It is one of the most significant classes in this game, since the settler is the main actor that represents the player's moves, and actions directly. The main goal of the settler is to build a space station then the player wins.

- **Superclasses**

Traveler

- **Interfaces**

None

- **Attributes**

- **ArrayList<Resource> resources**: ArrayList contains a list of the class Resource that the settler manages to get and those are counted as available resources that can be used in building (Gates, Robots, Space Station, etc..)

- **ArrayList< TeleportationGate > TeleportationGates:** It has the data of the pair of gates that the settler brings.
- **Methods**
- **Bool buildRobot():** This method is used to build the robots, and it is one of the operations that the settler can do. After building the robot, it can help in performing other operations like traveling and drilling.  
The function checks that the settler collected enough resources to build a robot then it returns true to add a new robot to the game, otherwise, it returns false.
- **Bool buildGate ():** It represents the operation of building a pair of gates using up two units of iron, a single unit of water ice, and a single unit of Uranium. The gates can be deployed later, they are used for entering whether the traveler is a robot or a settler.  
The method checks if the settler collected enough resources to build gates then it returns true to add a new pair of gates and connect them to the asteroid, otherwise it returns false.
- **Bool buildSpaceStation():** It is the final purpose of the game, that if the settler reaches it will win the game. They can mine at least three units of each resource and they collect those materials on a single asteroid, they can build the space station and the players win the game.  
The method checks that the settler collected enough resources to build a robot then it returns true to add a new robot to the game, otherwise, it returns false.
- **Void die():** The settler dies if it cannot survive the sun storm or the radioactive materials.  
The asteroid removes the dead settler from its list of travelers.
- **Void mine():** It represents the operation of mining that the settler usually does to extract materials that will help in building the space station. Also, settlers can only mine when the asteroid is at aphelion.  
The mined resources are deducted from the amount of the resources (it represents the resource amount in the core of the asteroid and ready to be mined) of its asteroid as well as a new object will be added to the ArrayList of the type Resource (It represents the resources that are ready to be used to building objects.), the settler can mine a unit per each time.
- **Void fill():** A hollow asteroid can be filled with a unit of resource, counting as an operation of the operations the settler can do.  
Once the settler fill a hollow asteroid, the resource will be deducted from the ArrayList and added to the asteroid core resources (As It will not be ready to be used without mining again.)
- **Void hide():** A settler can only survive a sun storm if they hide in the core of a hollow asteroid. This, of course, is only possible if the mantle has been drilled through.  
Successful hide operation has 2 conditions: the value of resource amount of the

asteroid is equal to 0 (That means it has no resources), the second condition is the asteroid depth is equal to 0 (That means the asteroid is fully drilled)

- **Void deployGate(TeleportationGate g, Asteroid a):** The gates can be deployed in the vicinity of the asteroid the settler is on, the method uses two parameters, the current asteroid and the gates which will be deployed, and also add the gates to the TeleportationGates list of each asteroid.
  
- **Void Travel(Asteroid a):** The settler can travel between the asteroids, to mine and extract resources, the method uses the asteroid as a parameter and once the settler reached the asteroid, it will be added to the list of travelers of the current asteroid and removed from the previous asteroid.
  
- **Void drill():** It is one of the operations that the settler can do, the settler can drill in the asteroids, they travel to so they can mine for resources.  
The depth variable of the asteroid is always set to the max. The more the settler drill, the less depth will be, until the asteroid is fully drilled then the depth will become Zero. The settler drill one unit each time the function is called
  
- **Void carryResources():** A single settler can only carry 10 units of resources, that's the spaceships' capacity. The carried resources will be deducted from the resources list of the asteroid and added to the core resources, it is a condition that the settler can not use the carried resources until they are put back. The settler carries one unit each time the function is called
  
- **Void putResources():** The settler can put the carried resources in the desired place, to travel to other asteroids, and move the resources out from the spaceship. Once the settler puts the resources, they are counted again in the available resources of the current asteroid to be used in building objects. The settler put one unit each time the function is called.

## 2. Asteroid

- **Responsibility**

Settlers wander in the asteroid belt with single-person spaceships looking for resources.

- **Superclasses**

None

- **Interfaces**

None

- **Attributes**

- **Sring ID:** each asteroid has a unique ID, and this represents the ID of the asteroid.
- **int Depth:** each asteroid has a depth, The depth of this rock mantle can vary from asteroid to asteroid, the depth is always set to the max and decreased gradually if the settler drilled more units.
- **int ResourceAmount:** it represents the kind of resources that exist inside the asteroid and is counted as available resources that can be used in building objects.
- **Resource CoreResource:** it stores the number of resources of each asteroid that exist in the core and the settler can mine (They are not counted as the available resources and cannot be used directly without mining).
- **Bool IsAphelion:** it represents the state of the asteroid if it is aphelion or not. Accordingly, that decides the behavior of the asteroid whether if it returns false, the asteroid will explode (If it's fully drilled and contains radioactive resources). Otherwise, it doesn't explode.
- **ArrayList<Resource> resources :**ArrayList it contains a list of the class Resource that the asteroid contains and is available to be used 0 or many also it stores all the data of the resources such as the type.
- **ArrayList<Traveler>travelers:** it shows the travelers(Robots and Settlers) that are currently in a given asteroid.
- **ArrayList<Neighbors> neighbors :** It shows the data of the neighboring asteroids.

- **Methods**

- **Void Explode():** . When a fully drilled asteroid with a radioactive core is at perihelion, the asteroid explodes and kills any settler on it. The settlers and robots are removed from the travelers' list of the given asteroid.
- **Void addTraveller(Traveler t):** When a settler or robot travel to the asteroid, it will be added through this method to the traveler list
- **Void removeTraveller(Traveler t):** When a settler or robot travel from the asteroid or die, it will be removed through this method from the traveler list.
- **Void addResources(Resource r):** it represents the available resources to be used in the asteroid, mining the resources and extracting them, will be added using this method since they will be available to be used, also carrying and putting the resources from different asteroids will be added through this method.
- **Void removeResource(Resource r):** The resources will be carried into the spaceship to be moved to other asteroids, will be deducted directly from the current asteroid and added to the new one, and also will be removed if the settler decided to hide the resource in the core of an asteroid

### 3. Robot

- **Responsibility**

The settler built the robot and the robot can perform some common operations that the settler normally does.

- **Superclasses**

Traveller

- **Interfaces**

No

- **Attributes**

- **Sring ID:** each robot has a unique ID, and this represents the ID of the robot.
- **Bool Hidden:** It returns if the traveler is hidden or not upon that the behavior of the traveler will be decided when the sunstorm comes.

- **Methods**

- **Void die():** Sun storms do damage robots unless they hide in a hollow asteroid. The asteroid removes the dead settler from its list of travelers
- **Void hide():** Robots hide in a hollow asteroid, so they can survive the sun storm
- **Void drill():** The robots can drill holes in the asteroids although they can not mine. They can drill one unit per method call.
- **Void travel(Asteroid a):** The robot travels between asteroids, to do a mission or to survive the radioactive explosions. The robot that leaves the asteroid will be deleted through the remove traveler method and added to the current asteroid.

### 4. TeleportationGate

- **Responsibility**

The settler can build a pair of gates, they can be used for entering to travel or can be also deployed

- **Superclasses**

None

- **Interfaces**

None

- **Attributes**

- **String ID:** each gate has a unique ID, and this represents the ID of the gate.

- **Methods**

None

## 5. Resource

- **Responsibility**

The class contains all the information of the resources minded and available to be used in building the space station and shows the type of each as well.

- **Superclasses**

None

- **Interfaces**

None

- **Attributes**

- **Bool IsRadioActive:** That refers if the resource is radioactive or not, accordingly the resource can do radioactive explosions.

- **Methods**

None

## 6. WaterIce

- **Responsibility**

It is one of the main resources that is used to build the space station,

- **Superclasses**

Resource

- **Interfaces**

None

- **Attributes**

None

- **Methods**

- **Void Sublimates():** When a fully drilled asteroid with water ice in its core is at perihelion, the water ice sublimates.

- **Bool IsRadioActive:** That refers if the resource is radioactive or not, accordingly the resource can do the radioactive explosion.

## 7. Carbon

- **Responsibility**

It is one of the main resources that is used to build the space station.

- **Superclasses**

Resource

- **Interfaces**

None

- **Attributes**

- **Bool IsRadioActive:** That refers if the resource is radioactive or not, accordingly the resource can do the radioactive explosion.

- **Methods**

None.

## 8. Uranium

- **Responsibility**

It is one of the main resources that is used to build the space station,

- **Superclasses**

Resource

- **Interfaces**

None

- **Attributes**

- **Bool IsRadioActive:** That refers if the resource is radioactive or not, accordingly the resource can do the radioactive explosion.

- **Methods**

None.

## 9. Iron

- **Responsibility**

It is one of the main resources that is used to build the space station,

- **Superclasses**

Resource

- **Interfaces**

None

- **Attributes**

- **Bool IsRadioActive:** That refers if the resource is radioactive or not, accordingly the resource can do the radioactive explosion.

- **Methods**

None

## 10. Traveler

- **Responsibility**

Is the base class that is extended into two classes (Settler, Robot). It contains the common operations between the settler and the robot, it is also an abstract class so no instances can be initialized.

- **Superclasses**

None

- **Interfaces**

None

- **Attributes**

None

- **Methods**

- **Void sunStorm():** Sun storms do damage Travelers unless they hide in a hollow asteroid and also it harms the settler and that may result to make it die, unless the settler hides too

- **Void hide():** Travelers hide in a hollow asteroid, so they can survive the sun storm.
- **Void drill():** The Travellers can drill holes in the asteroids although they can not mine, settlers can drill holes and mine as well. The Traveler can drill one unit per a function call.
- **Void travel(Asteroid a):** The Travellers travel between asteroids, to do a mission or to survive the radioactive explosions, settlers can travel to the neighboring asteroids. The travelers were removed from the previous asteroid list and added to the current asteroid

## 11. SpaceStation

- **Responsibility**

It is the space station that the settlers are trying to extract enough resources to build it.

- **Superclasses**

None

- **Interfaces**

None

- **Attributes**

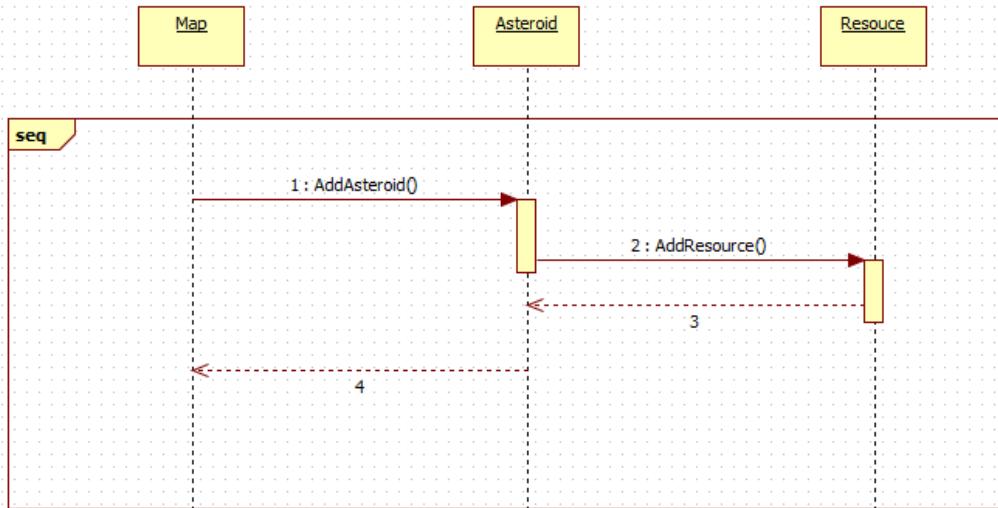
None

- **Methods**

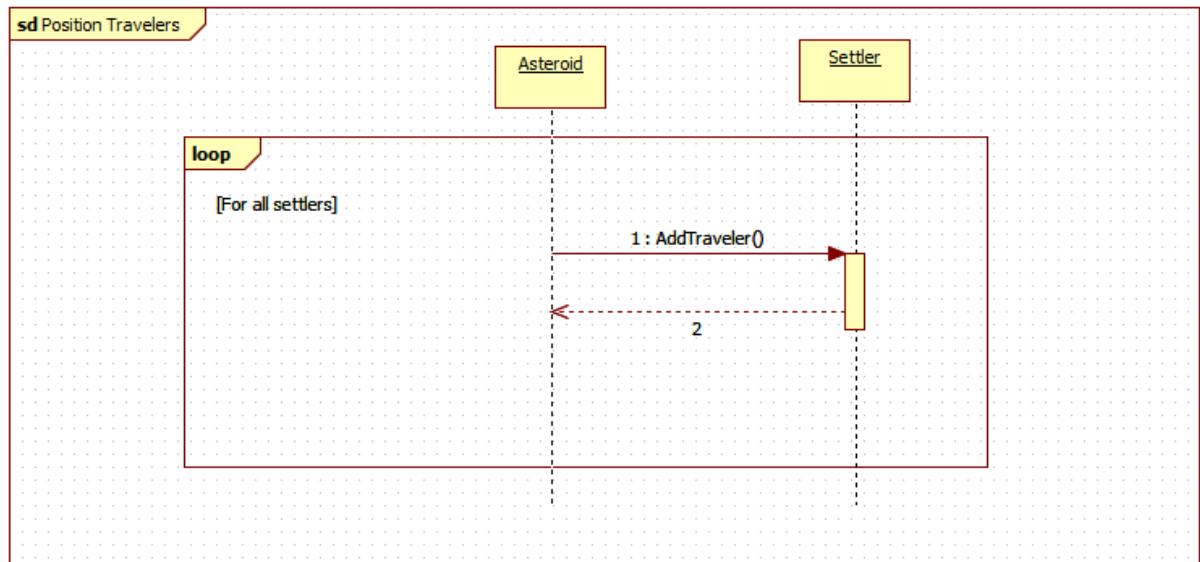
None

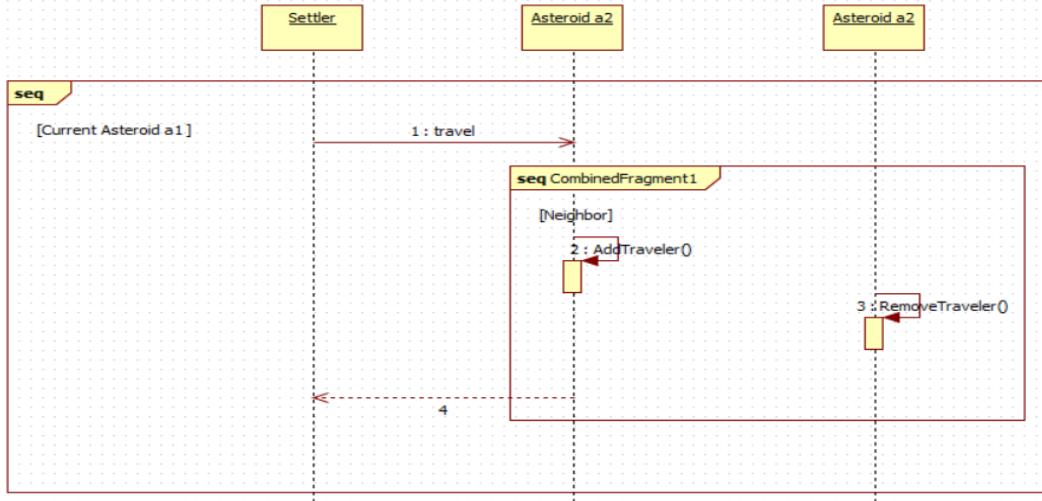
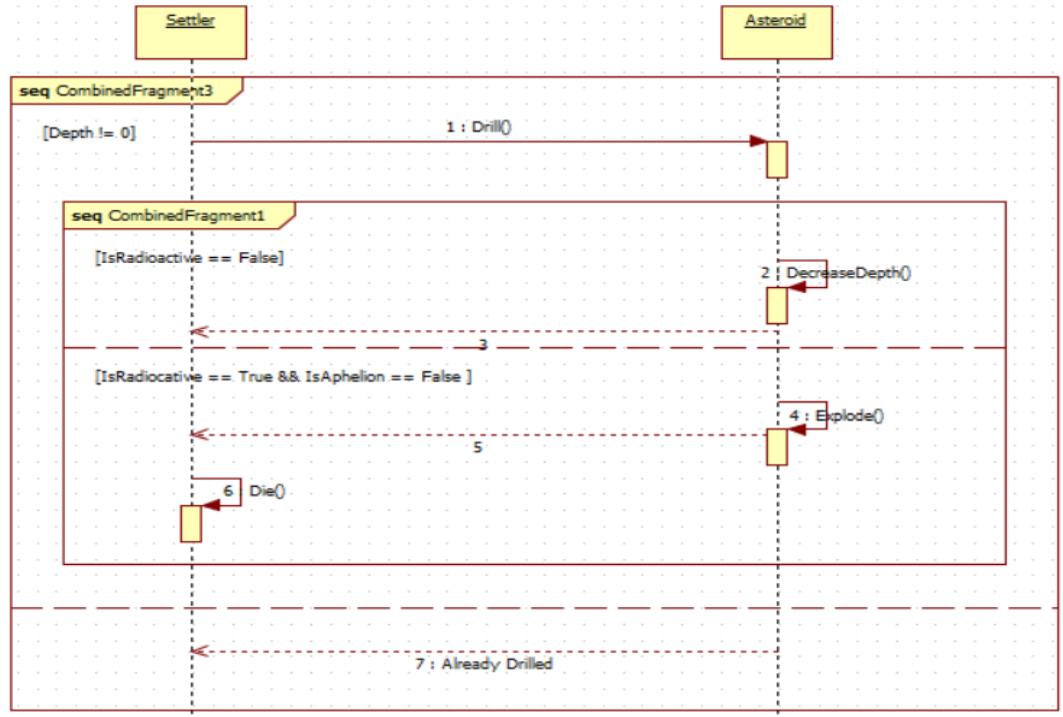
## 4.4 Sequence diagrams

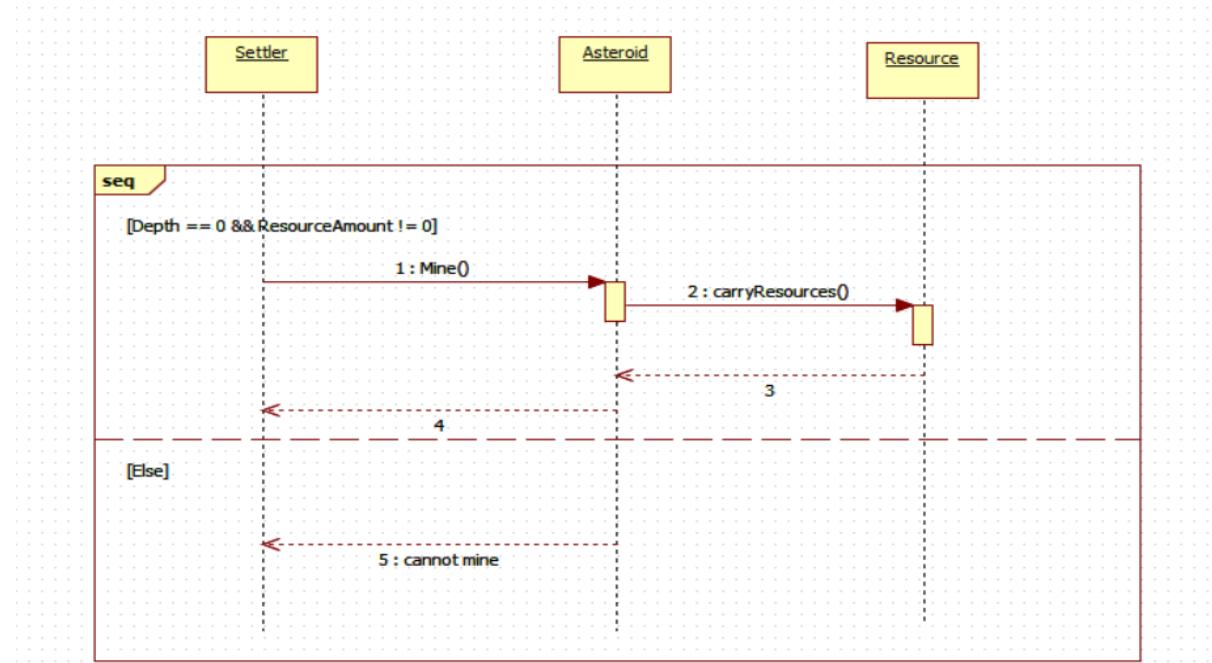
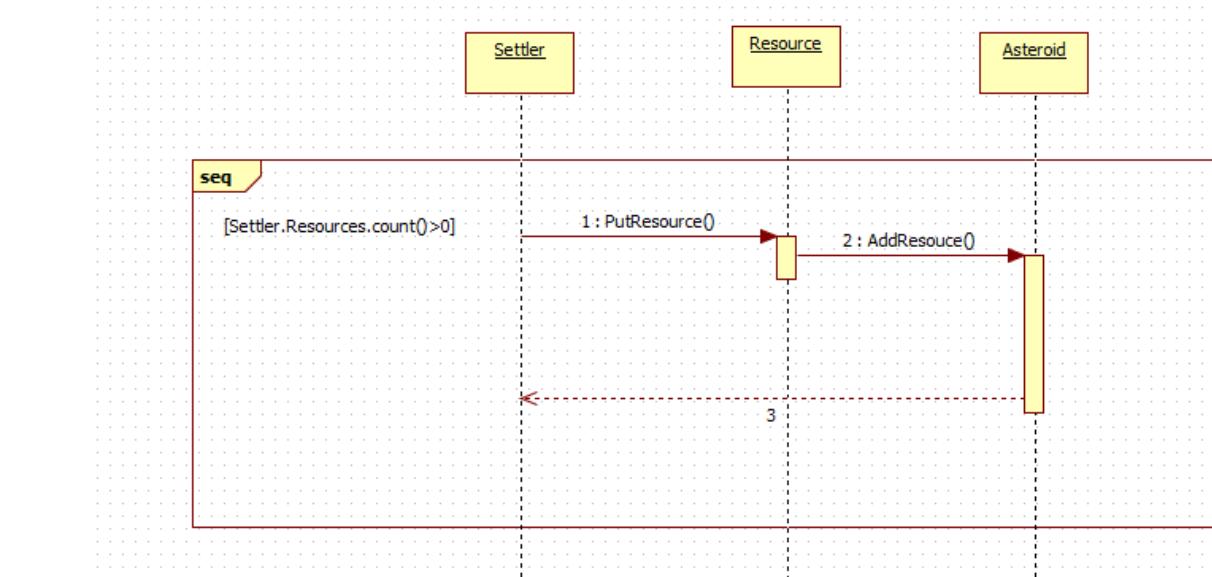
**Initialization:**

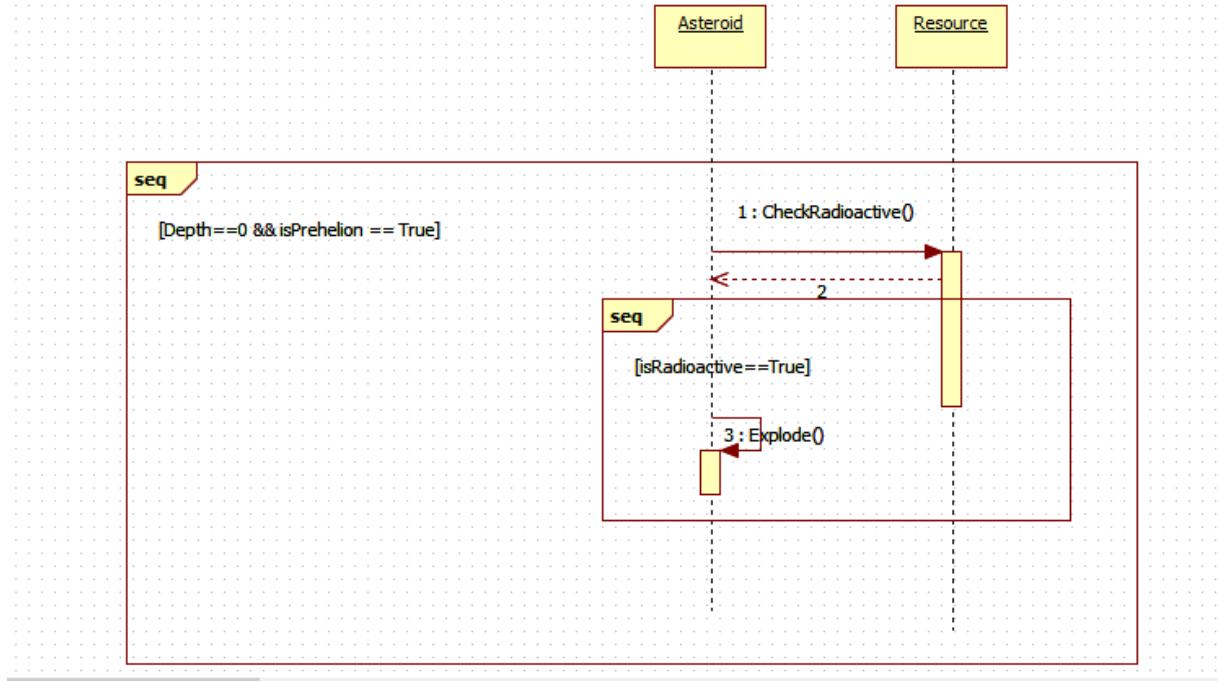


**Position settlers:**

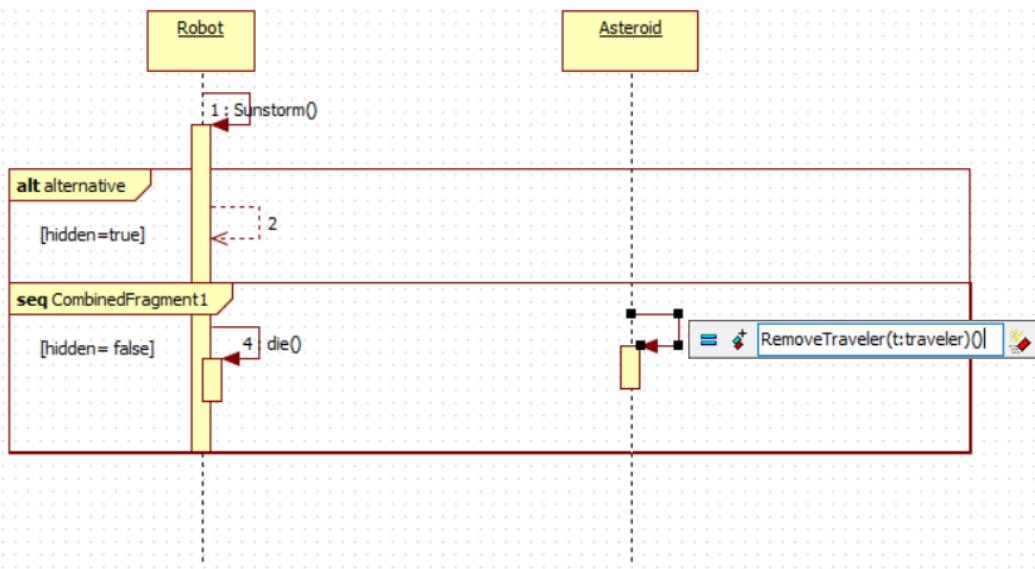
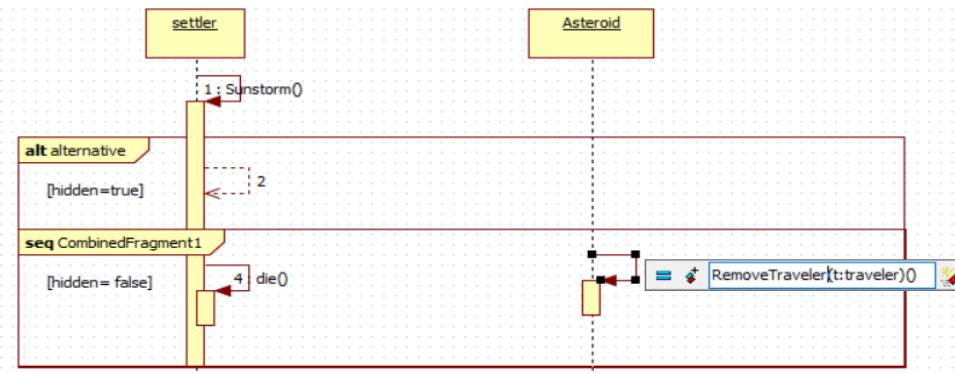


**Settler moves:****Settler drills:**

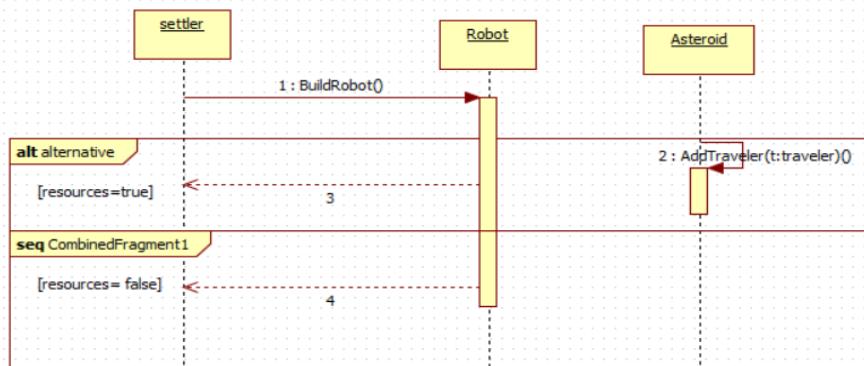
**Settler mines:****Put Down Resources:**

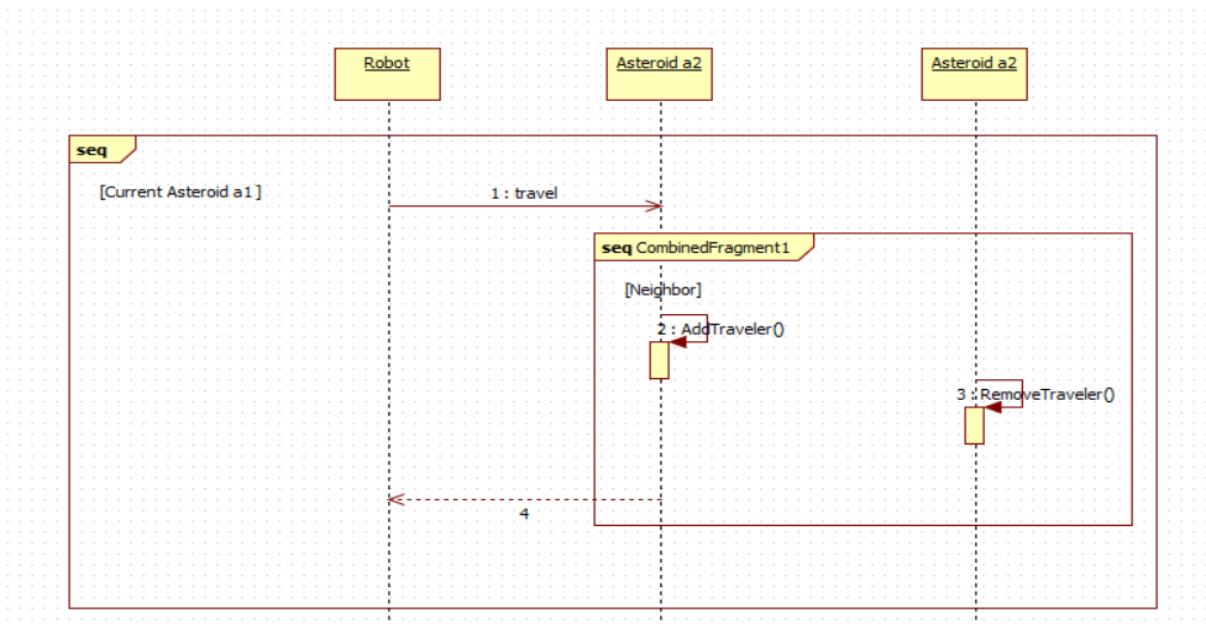
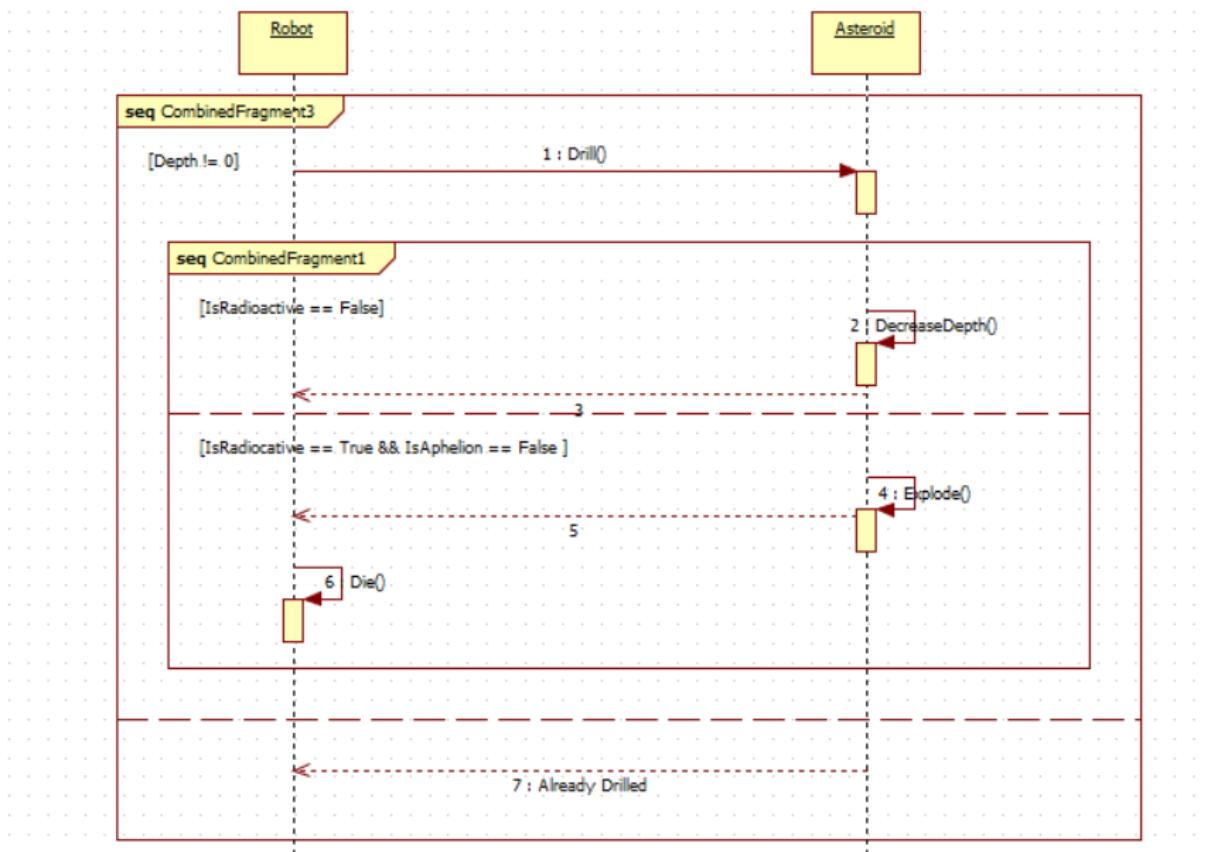
**Explosion At Perihelion:**

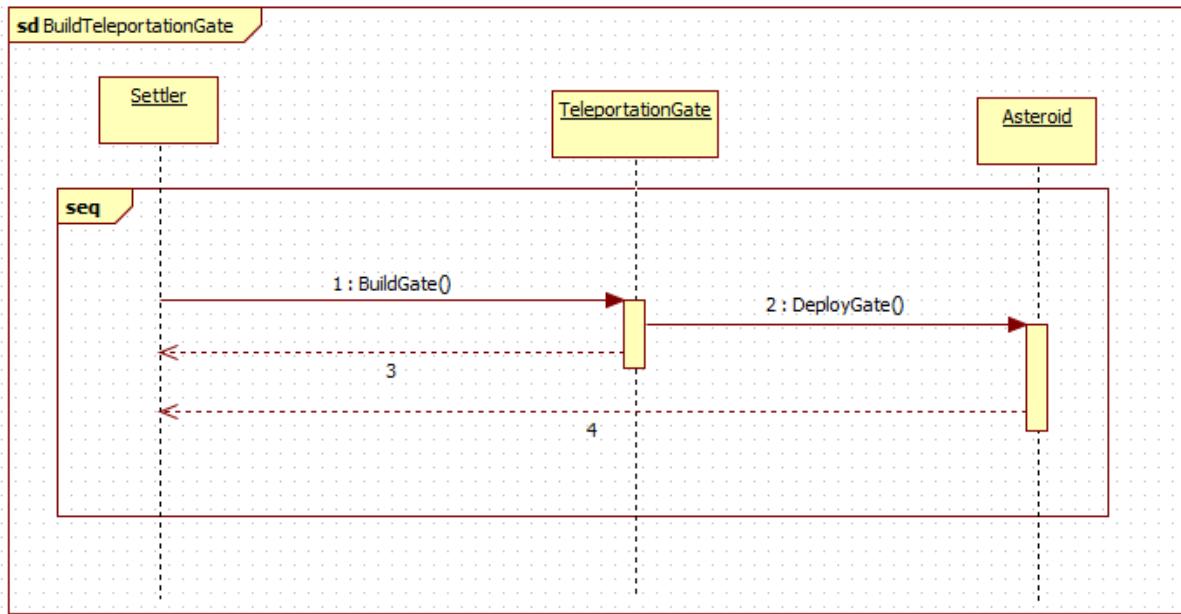
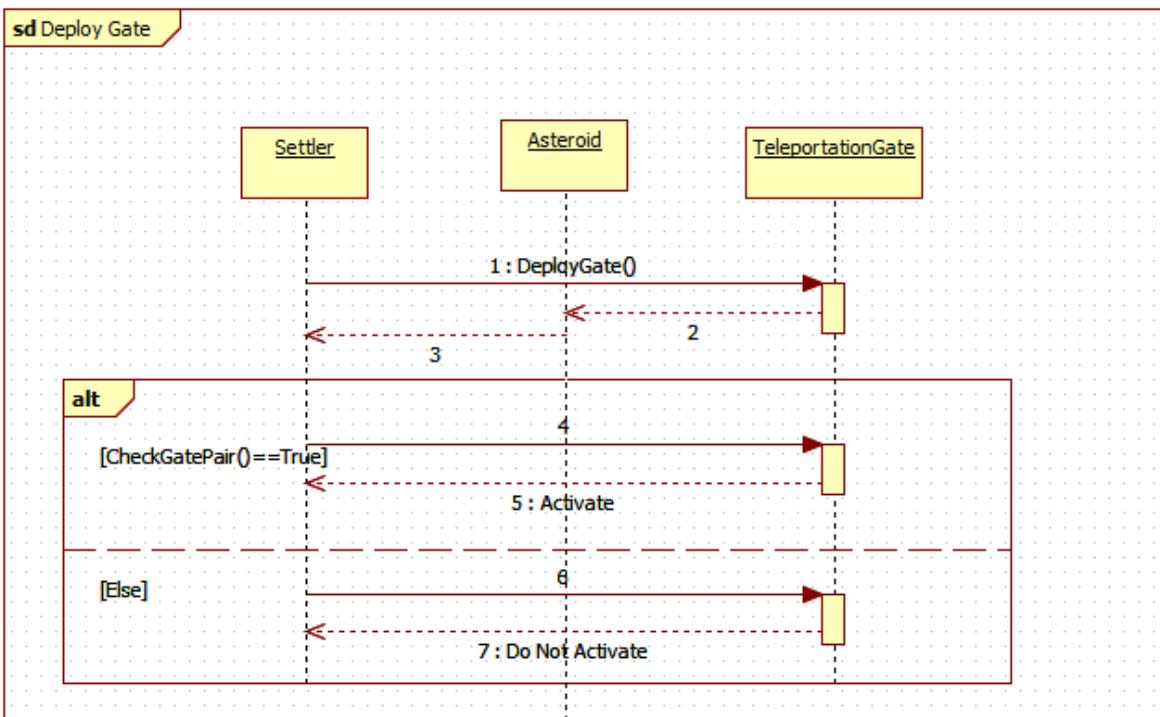
**Sun-wind occurs( 2 cases one for robot and one for settler):**

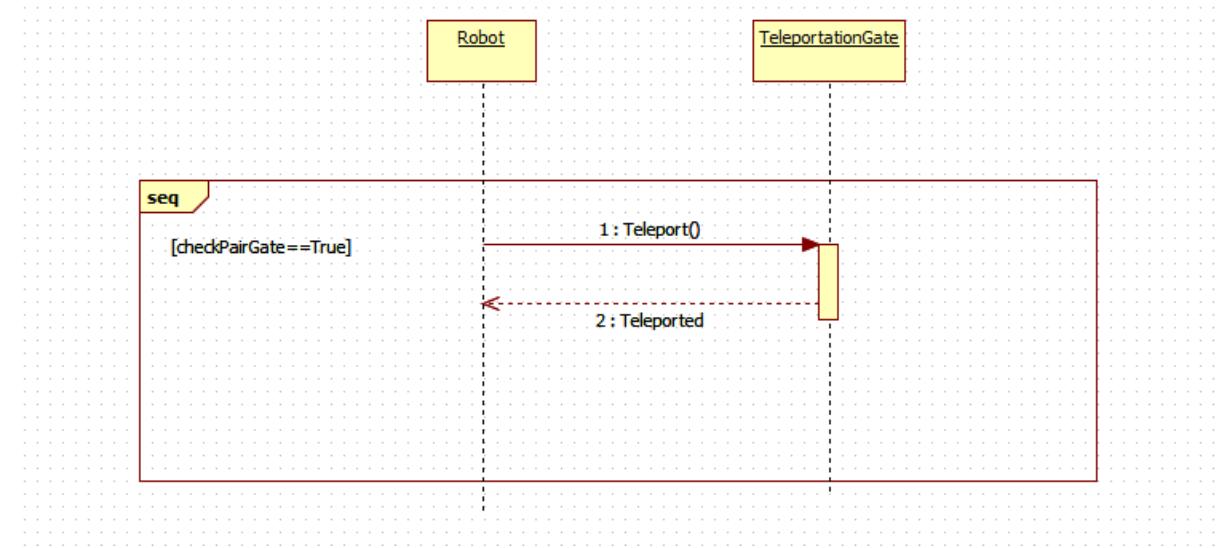
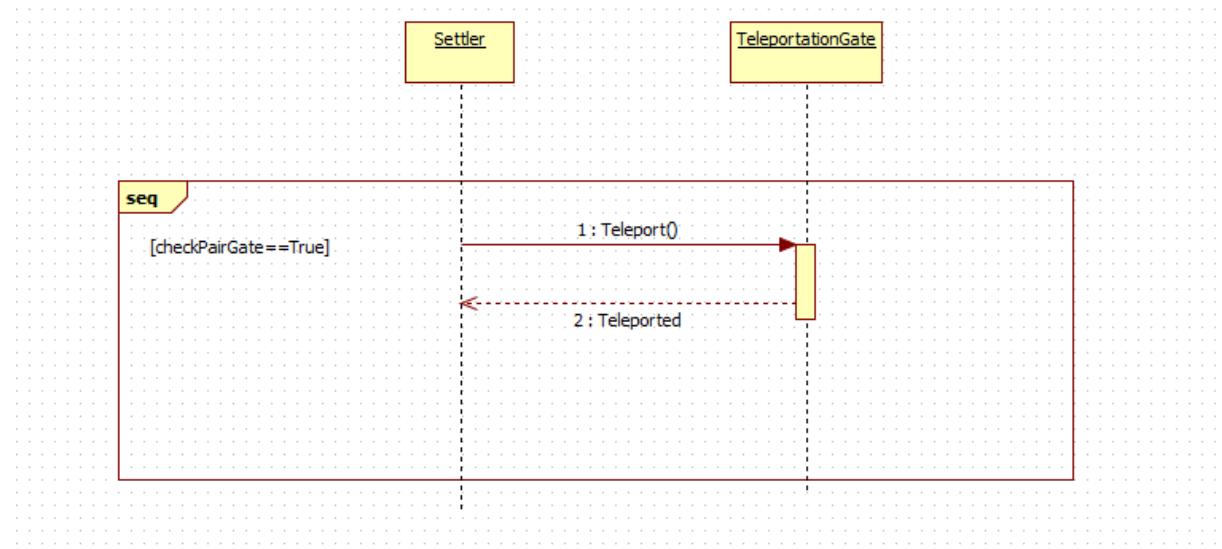


**Robot is created (created, put on the asteroid):**



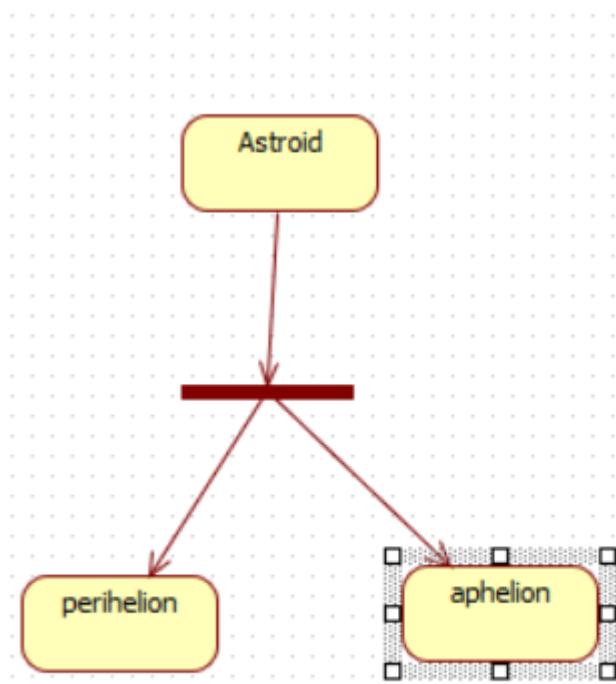
**Robot moves:****Robot drills:**

**Build Teleportation Gate:****Deploy Teleportation Gate:**

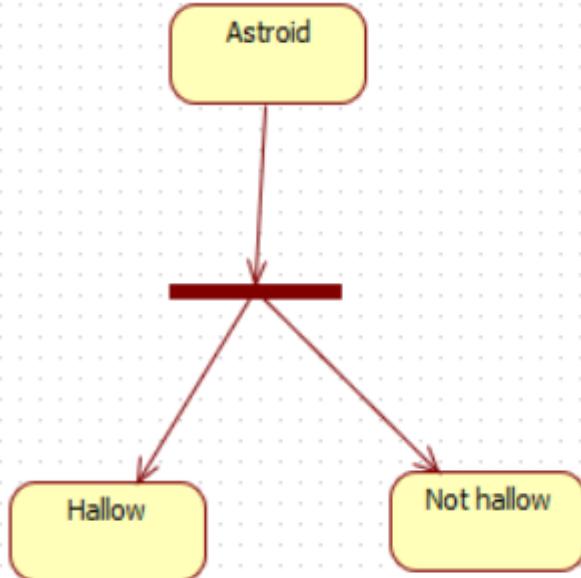
**Robot Teleportation:****Settler Teleportation:**

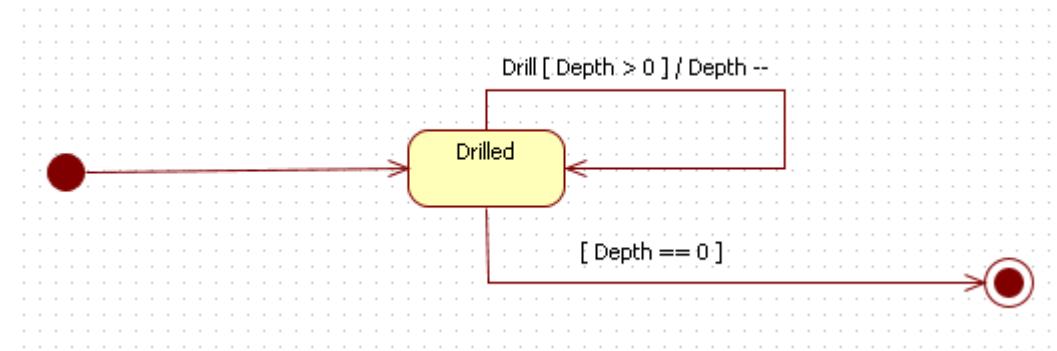
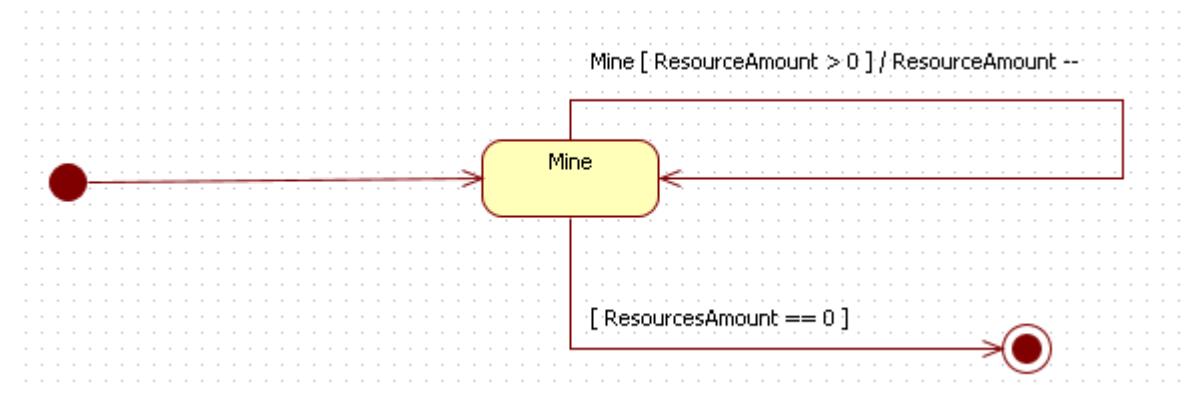
## 4.5 State-charts

Astroid state:



Asteroid hallow or not:



**Drill:****Mine:****4.6 Protocol**

Start (date & time)	Duration (hours)	Performer name	Activity description
11/03/2022 and 12/03/2022	5 hours	Dhouiou Oussama	Object catalog 3 sequence diagrams 2 state charts
10/03/2022	5 hours	Mohamed Amine Zaghoudi	7 Sequence Diagrams
08/03/2022	6 hours	Kirolos Sawiris Ahmed Fathy	Static structure diagram(Class diagram)
08/03/2022	3 hours	Mohamed Abouzabady	Class Description
12/03/2022	4 hours	Mohammadparsa Kolivand	Working and checking the sequence diagrams
13/03/2022	5 hours	Mahdi Yahia	Working on and reviewing the sequence diagrams.

# **[ASTEROID MINING]**

**[TEAM NNUMBER 7]**

Supervisor:  
**Dr. Balla Katalin**

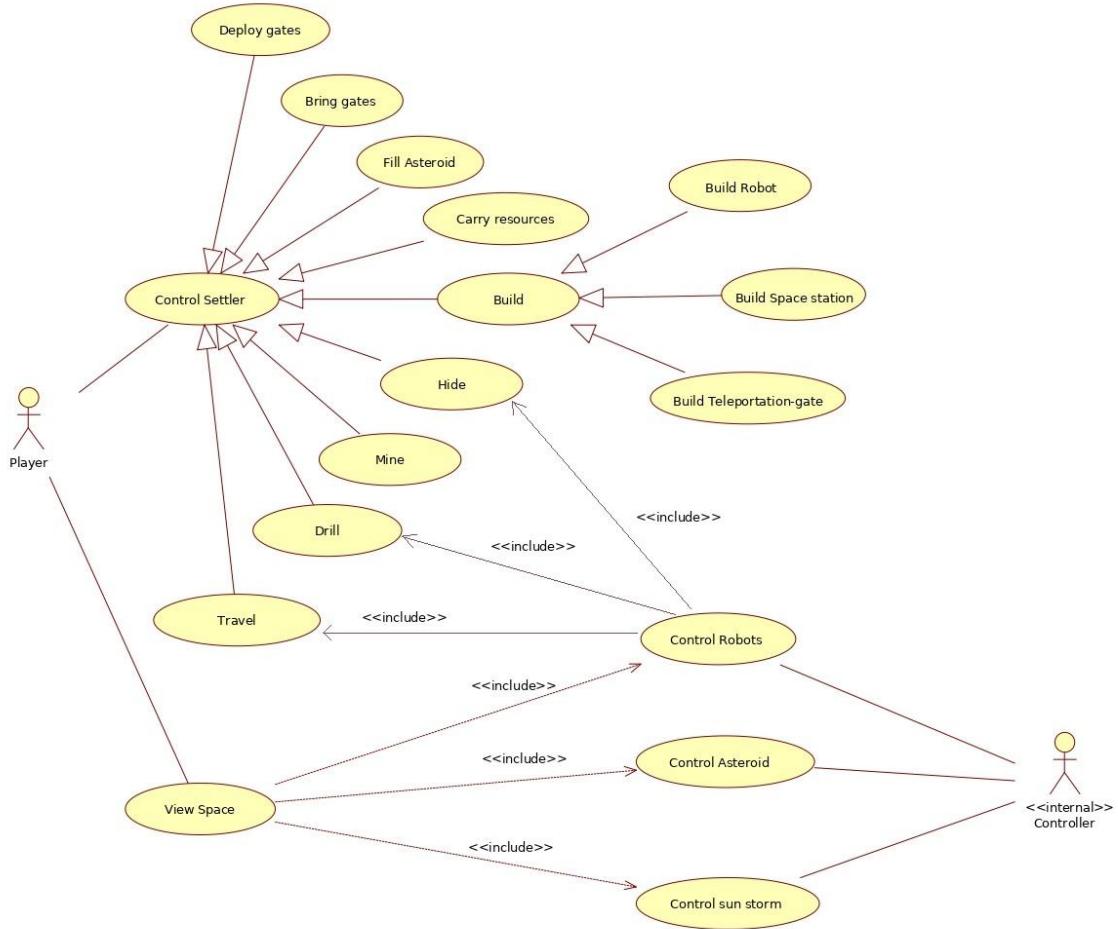
## **Members:**

[Abouzabady Mohamed]	[GJVKZQ]	[mohamedabozabady1@gmail.com]
[Kirolos Sawiris]	[Y8B5KD]	[Kirolossawiris@gmail.com]
[Oussama dhouioui]	[XFCBM7]	[Oussemadhououi04@gmail.com]
[Mohamed Amine]	[ARPMPB]	[mzaghdoudi01@gmail.com]
[Ahmed Mousa]	[EWLQXU]	[amousa@edu.bme.hu]
[MohammadParsa Kolivand]	[RLU15I]	[parsakolivand@edu.bme.hu]
[Mahdi Yahia]	[WF72QQ]	[mahdiyahia18@gmail.com]

## 5. Planning the skeleton

### 5.1 Real use-cases of the skeleton model

#### 5.1.1 Use-case diagram



#### 5.1.2 Use-case descriptions

<b>Use-case name</b>	<b>Control Settler</b>
<b>Short textual description</b>	<b>The player controls the settler through the Space.</b>
<b>Actors</b>	<b>Player</b>

<b>Use-case name</b>	<b>Travel</b>
<b>Short textual description</b>	<b>The player control which neighboring asteroid the settler will go to.</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>1. The player moves the settler to another asteroid.</b>

<b>Use-case name</b>	<b>Drill</b>
<b>Short textual description</b>	<b>The players control the settler to make or deepen the hole in the mantle with one unit.</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>1. The settler drills in the mantle with one unit. 1.A.1. The settler can't drill in the mantle as the settler is fully drilled. 1.B.1. The settler dies if he fully drilled asteroid with a radioactive core is at perihelion</b>

<b>Use-case name</b>	<b>Mine</b>
<b>Short textual description</b>	<b>The settler extracts the resources from the core of the asteroid.</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>1. The settler extracts the resources from the core of the asteroid.</b>

<b>Use-case name</b>	<b>Hide</b>
<b>Short textual description</b>	<b>The settler hides from the sun storm</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>1. The settlers go to the core of a hollow asteroid to hide from the sun storm. 1.A.1. The settler can't go to the core of a hollow asteroid as it's not fully drilled</b>

<b>Use-case name</b>	<b>Build</b>
<b>Short textual description</b>	<b>The settler build objects</b>
<b>Actors</b>	<b>Player</b>

<b>Use-case name</b>	<b>Build Robot</b>
<b>Short textual description</b>	<b>The settler build a robot</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<p><b>1. The settler builds a robot as he has the required resources</b></p> <p><b>1.A.1. The settler cannot build a robot as he doesn't have the required resources.</b></p>

<b>Use-case name</b>	<b>Build Space station</b>
<b>Short textual description</b>	<b>The settler builds a space station</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<p><b>1. The settler built a space station as he managed to collect the required resources on a single asteroid.</b></p> <p><b>1.A.1. The settler couldn't build a space station as he didn't manage to collect the required resources on a single asteroid.</b></p>

<b>Use-case name</b>	<b>Build Teleportation-gate</b>
<b>Short textual description</b>	<b>The settler build a teleportation-gate</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<p><b>1. The settler builds a teleportation-gate as he has the required resources</b></p> <p><b>1.A.1. The settler cannot build a teleportation-gate as he doesn't have the required resources.</b></p>

<b>Use-case name</b>	<b>Deploy gates</b>
<b>Short textual description</b>	<b>The settlers deploy the gates in the vicinity of the asteroid.</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>1. The settlers deploy the gates in the vicinity of the asteroid.</b>

<b>Use-case name</b>	<b>Bring gates</b>
<b>Short textual description</b>	<b>The settler brings gates.</b>
<b>Actors</b>	<b>Player</b>

<b>Dialog, scenario</b>	<b>1. The settler brings a gate.</b> <b>1.A.1. The settler couldn't bring a gate as he already carry two gates.</b>
-------------------------	--

<b>Use-case name</b>	<b>Fill Asteroid</b>
<b>Short textual description</b>	<b>The settlers fill the asteroid with resources.</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>1. The settler fills the asteroid with one unit of resources.</b> <b>1.A.1. The settlers cannot fill the asteroid as it's not a hollow asteroid.</b> <b>1.B.1. The settler cannot fill the asteroid as it's not fully drilled.</b>

<b>Use-case name</b>	<b>Carry resources</b>
<b>Short textual description</b>	<b>The settler collects the resource</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>1. The settler collects the resource to move it from place to place.</b> <b>1.A.1. The settler can't collect the resource as he already carries 10 units of resources.</b>

<b>Use-case name</b>	<b>View Space</b>
<b>Short textual description</b>	<b>The player views the space</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>1. The system draws the current state of the space.</b> <b>2. The player observes the current state of the space</b>

<b>Use-case name</b>	<b>Control Robots</b>
<b>Short textual description</b>	<b>The Robots move in the space</b>
<b>Actors</b>	<b>Controller</b>
<b>Dialog, scenario</b>	<b>1. The Robot moves in space</b> <b>1.A.1. The Robot drills in the mantle of the asteroid.</b> <b>1.A.1.A.1. The Robot cannot drill in the mantle as it's fully drilled.</b> <b>1.B.1. The Robot go to the core of the asteroid to hide from the sun storm.</b>

<b>Use-case name</b>	<b>Control Asteroids</b>
<b>Short textual description</b>	<b>The asteroid explodes</b>
<b>Actors</b>	<b>Controller</b>
<b>Dialog, scenario</b>	<b>1. The asteroid explodes if it has a radioactive core and it is at perihelion</b>

<b>Use-case name</b>	<b>Control sun storm</b>
<b>Short textual description</b>	<b>The sun storm moves to reach the asteroid belt</b>
<b>Actors</b>	<b>Controller</b>
<b>Dialog, scenario</b>	<b>1. The sun storm moves to reach the asteroid belt and kills the settler. 1.A.1. The sun storm moves to reach the asteroid belt and damages the robot.</b>

## 5.2 Plans of the skeleton's UI, dialogs

Case 1

**Initialization :**

**System Output:**

A message will be displayed when the first asstroid is created and the another message “resource added” when resources will be added to the asteroid

**User Input:**

`AddAstroid()`

`AddResources()`

Case 2

**Position settlers:**

(At this stage we will see the number of the players )

How many travelers should be added ?

**System Output:**

`AddTraveller()`

User will enter the number

**System output:**

Traveler has been added .

Case 3

**Settler moves:**

(At this stage settlers will travel between the asteroids , the asteroid the settler will leave will be removed )

**System Output:** Firstly the neighbor should be added !

`travel ()`

**User Input:** User will press the key !

`AddTraveler()`

`removeTraveler()`

**System output:**

Traveler has been added

Traveler has been removed

Case 4

**Settler drills:**

(settlers will check the necessary conditions , Depth , Radioactive )

**System Output:** Check the depth

**User Input:** user will enter the key

If the depth is not !=0 then will continue

```

DecreaseDepth()
If RadioAvtive = true and IsAphelion= false
explode()

```

Case 5

**Settler mines:**

(in mine section the resource will be collected if the requirement is met)

**System Output:** Traveler will press the key

If the Deep =0 and resourceamount !=0 then

**User Input:**Mine(), CarryResources

**System output :** resources would be added to the inventory.

**else**

**System output:**

DecreaseResource()

Case 6

**Put Down Resources:**

(Settlers will collect and count the amount of Resources )

**System Output:** PutResource()

**System Output :** AddResource ()

**User Input:**

Resources have been added to the asteroid

Case 7

**Explosion At Perihelion:**

(Radioactive explosion will happen )

**System Output:**CheckRadioActive

Will check if Deep=0 and isPrehelion=True

**User Input:** Explode ()

If RadioActive =0

Case 8

**Sun-wind occurs( 2 cases one for robot and one for settler):**

den We have the attribute hidden for both robot and settler . This attribute is boolean and will inform us whether (the settler or robot ) are hidden or not and based on that we call functions

1st Case For settler :

**System Output:** if Hidden=true so the settler is safe when the sunstorm occurs but if the attribute is false in this case we call the function die() and removetraveler() and we display a messgae “your settler died”

**User Input:** sunstorm() die() removetraveler()

2nd Case For Robot :

**System Output:** if Hidden=true so the robot is safe when the sunstorm occurs but if the attribute is false in this case we call the function die() and removetraveler() and we display a message “your robot died”

**User Input:** sunstorm() die() removetraveler()

Case 9

**Robot is created (created, put on the asteroid):**  
(by having enough resources the settlers will build robots )

**User Input:** Buildrobot() / Addtraveler()

**Parameters :** resources

**System Output:**

If resources are enough so a message will be displayed “Robot Built Successfully”

Else this message will be displayed “Not Enough Resources”

Case 10

**Robot moves:**

(Robots will move in belt )

**System Output:**Travel () to get the neighbor

**User Input:**enter asteroid

AddTraveler ()

RemoveTraveler ()

Case 11

**Robot drills:**

(Robot will start drilling an asteroid )

**System Output:**Drill()

If the Deep!=0

**User Input:**Checking the Depth of the current Asteroid!

**System Output :** Decreasedepth()

If RadioActive=False

else

explode()

if RadioActive =True and IsApheline=false

Decreasing the depth of the Asteroid to expose the core!

Case 12

**Build Teleportation Gate:**

In order to build teleportation gates, there is a certain amount of resources the settler must own.

The build gate method will check the availability of resources and decide whether to build the gates or ignore the user's command.

**User input:**

BuildGate()

**System Output:**

Gates Built Successfully

Not Enough Resources

Case 13

**Deploy Teleportation Gate:**

The DeployGate() method will deploy a gate on the asteroid the player is on, after that the CheckGatePair() will verify if the gate coupled with the gate we just deployed is already deployed or not, in order to decide to activate the couple of gates or not.

**User input:**

DeployGate(), CheckGatePair().

**System Output:**

Gate Deployed

No Gate Owned

Activate the Gates

Do Not Activate the Gates

Case 14

**Settler Teleportation:**

The Teleport() method will verify if the gate is activated, and if it is the case it will position the settler on the asteroid where the other gate is located.

**User input:**

Teleport()

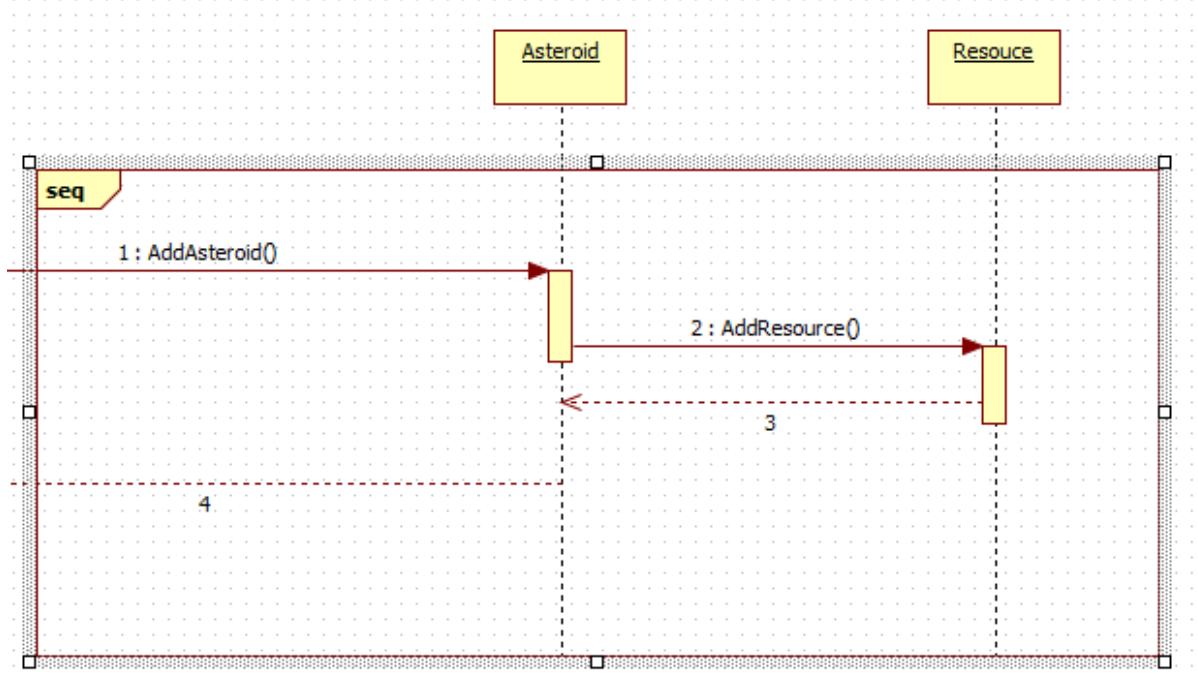
**System output:**

Gates Pair Activated

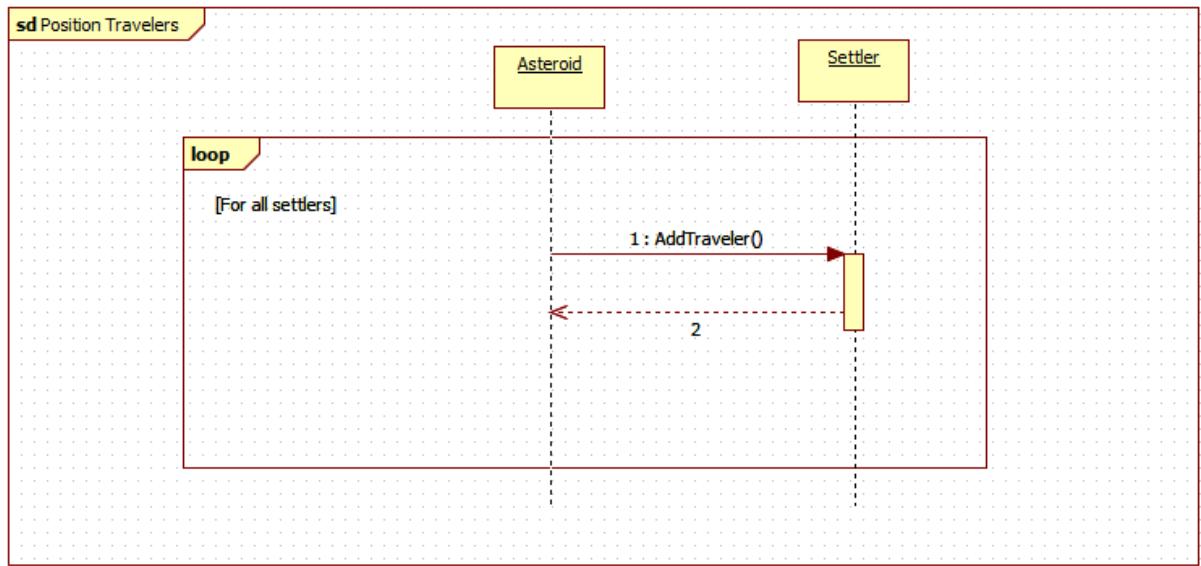
Gates Pair Not Activated

### 5.3 Detailed sequence diagrams for internal activities

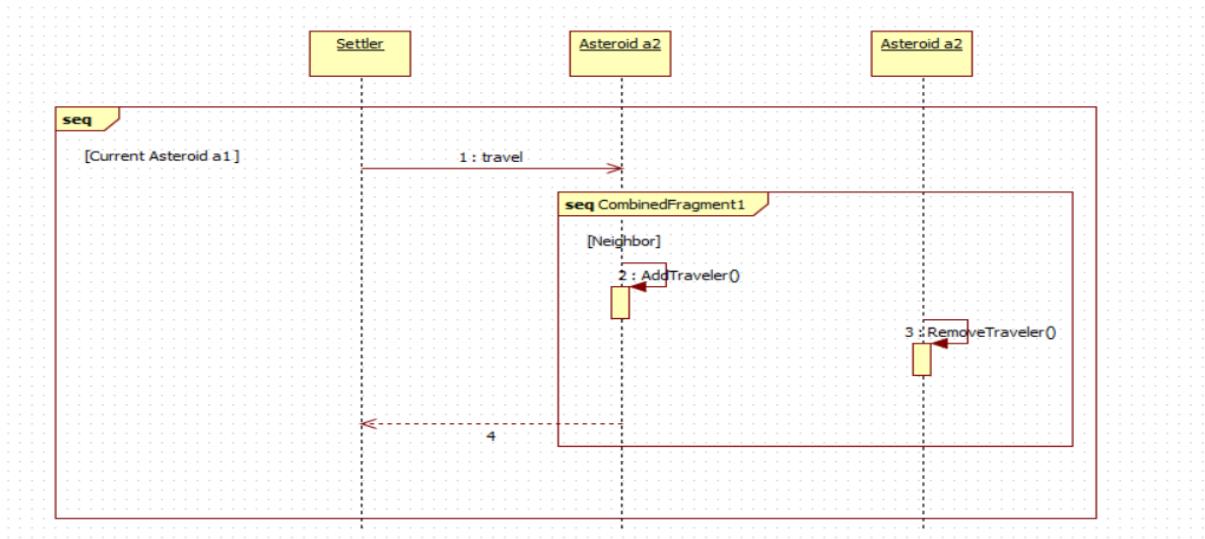
#### 5.3.1 Initialization :



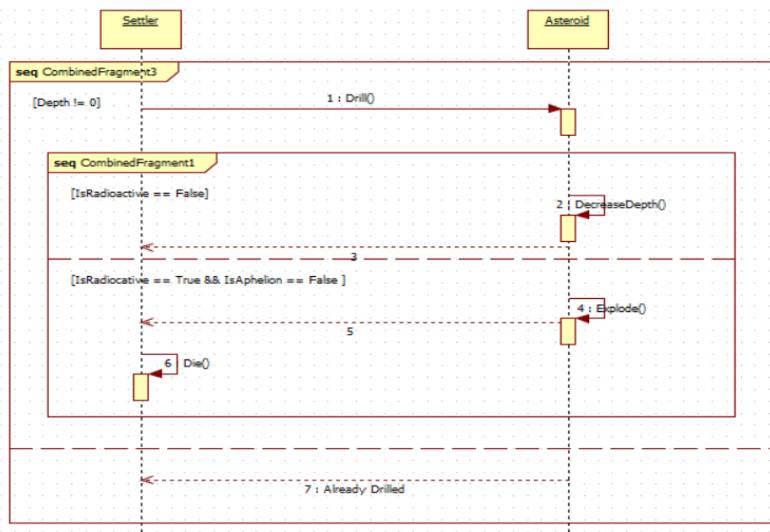
### 5.3.2 Position settlers :



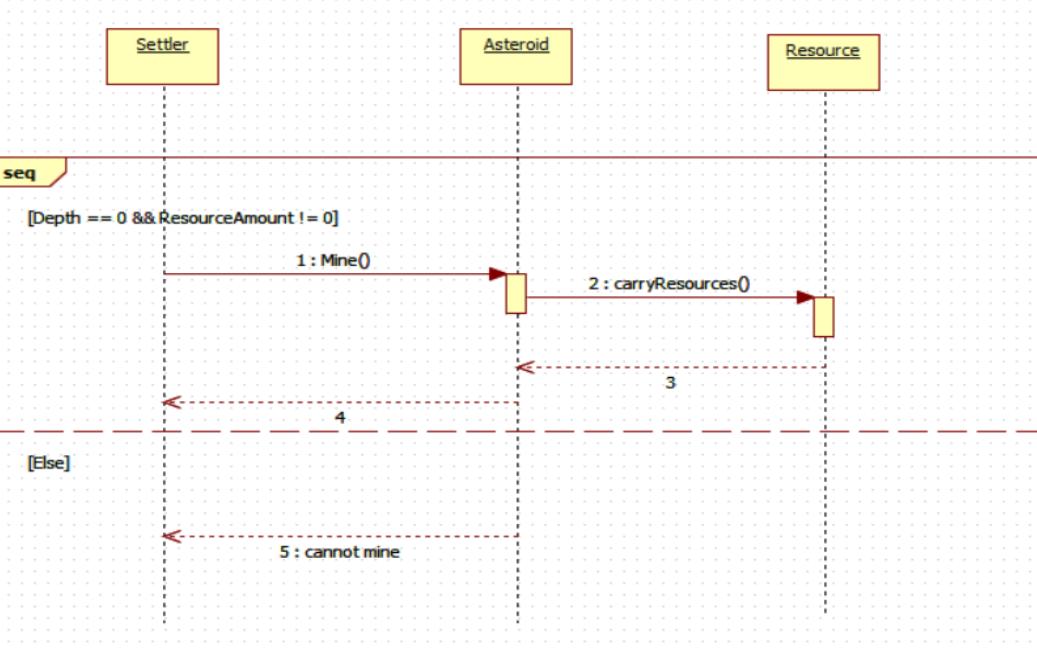
### 5.3.3 Settler moves:



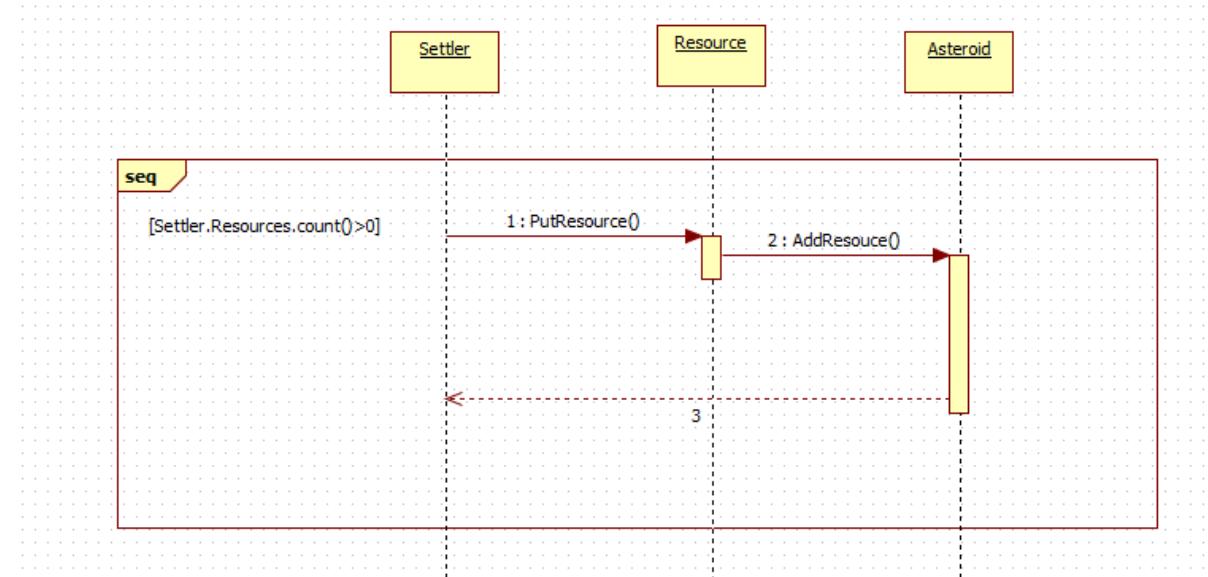
### 5.3.4 Settler drills:



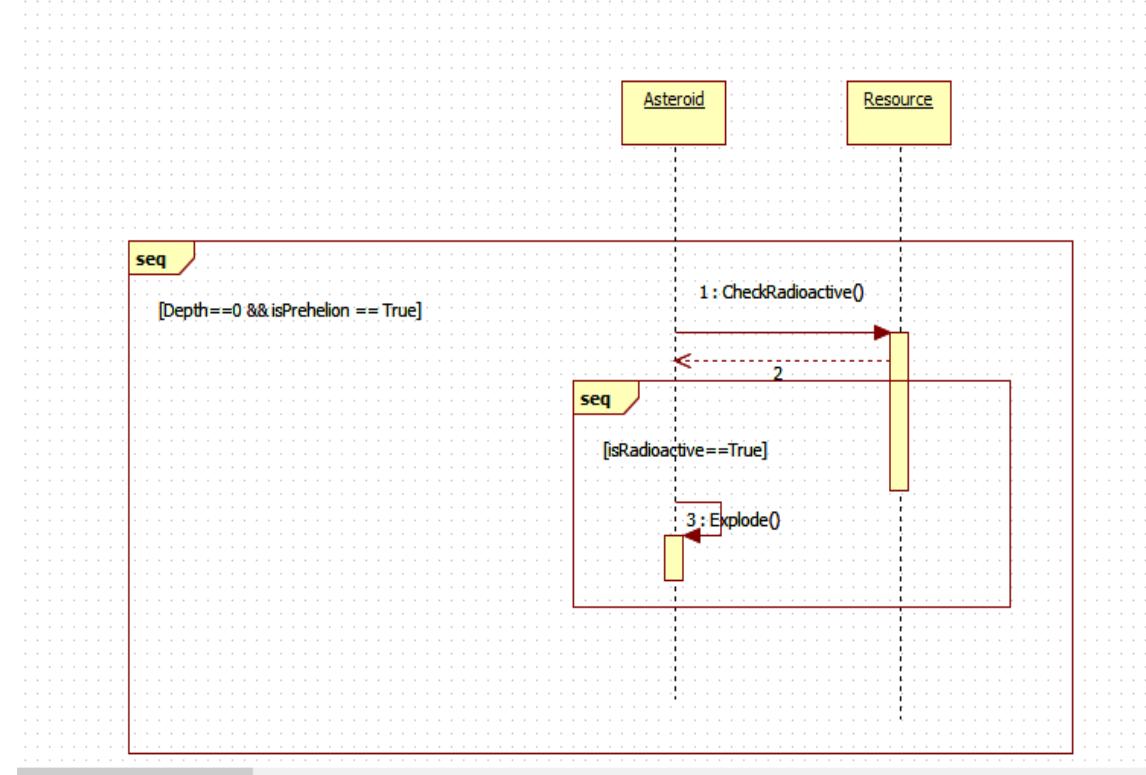
### 5.3.5 Settler mines:



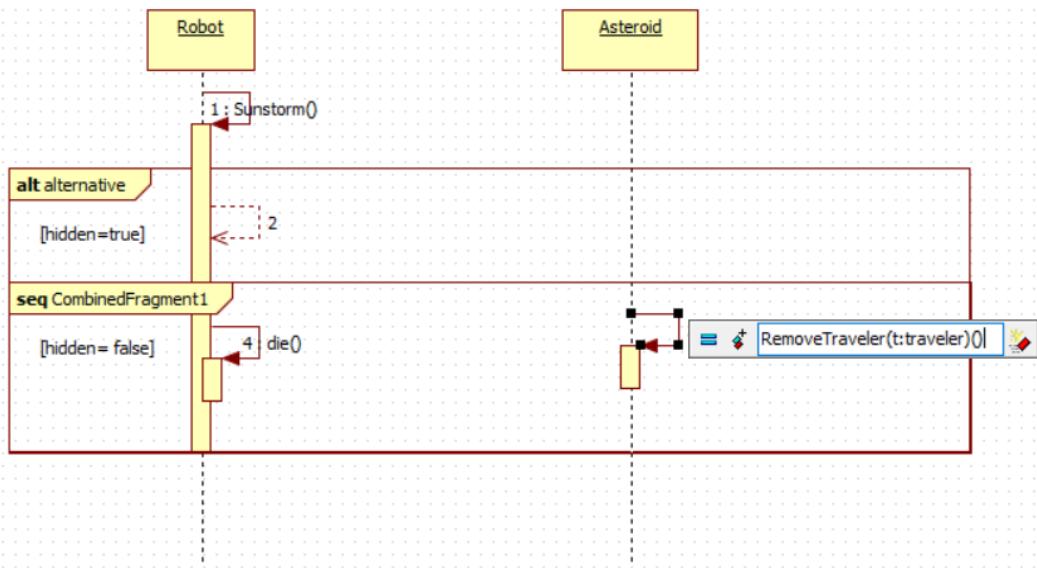
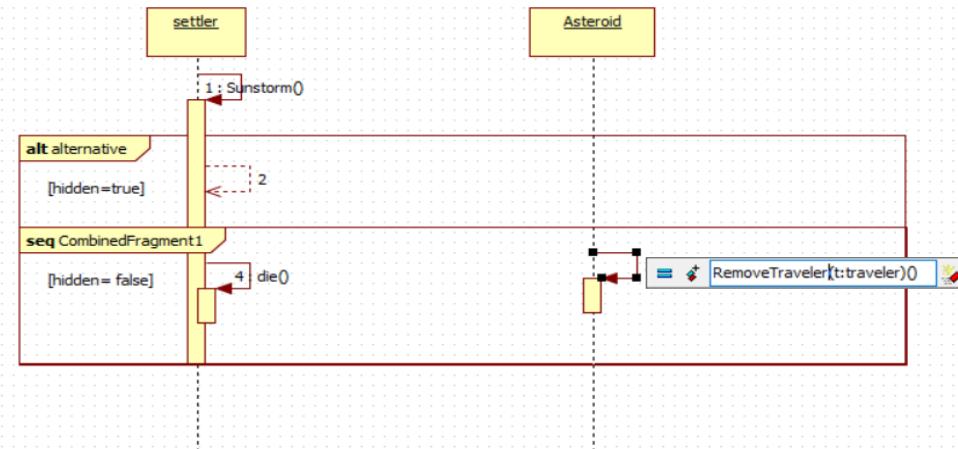
### 5.3.6 Put Down Resources:



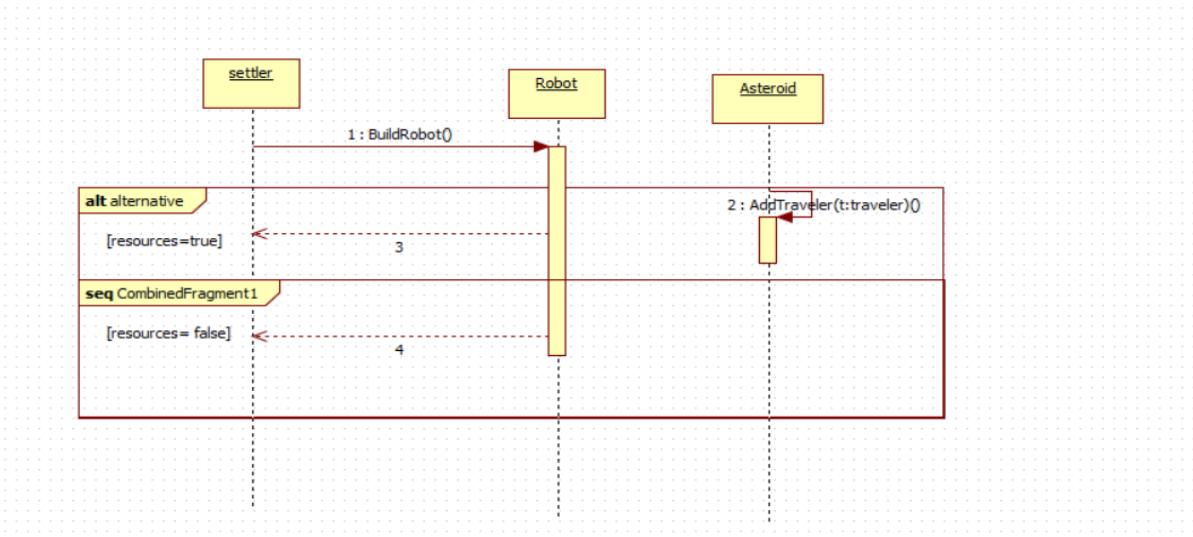
### 5.3.7 Explosion At Perihelion: (Radioactive explosion will happen )



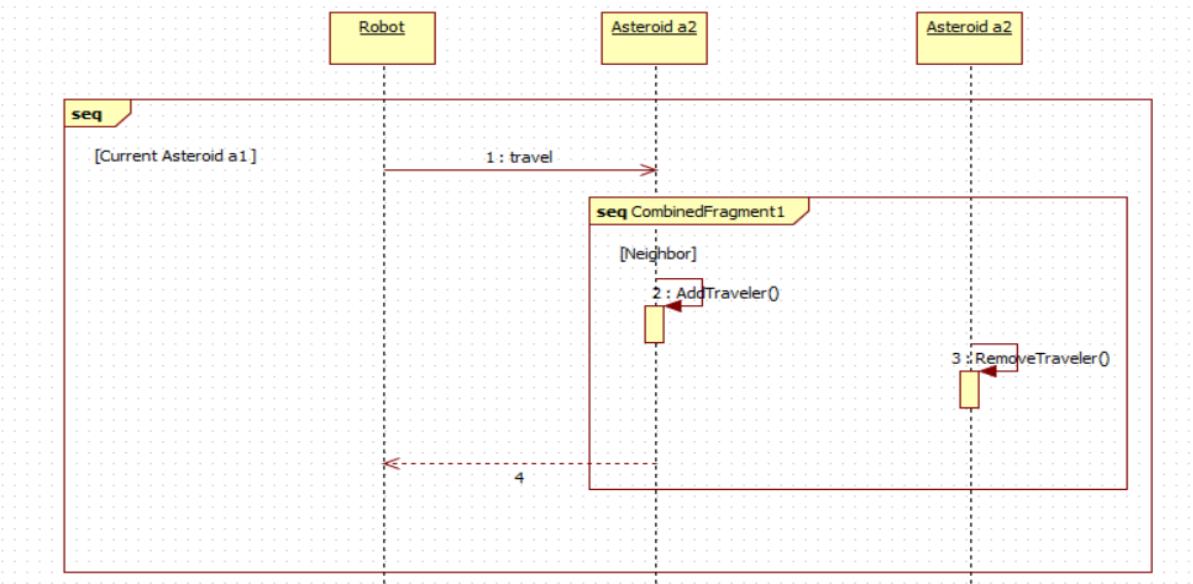
### 5.3.8 Sun-wind occurs( 2 cases one for robot and one for settler):



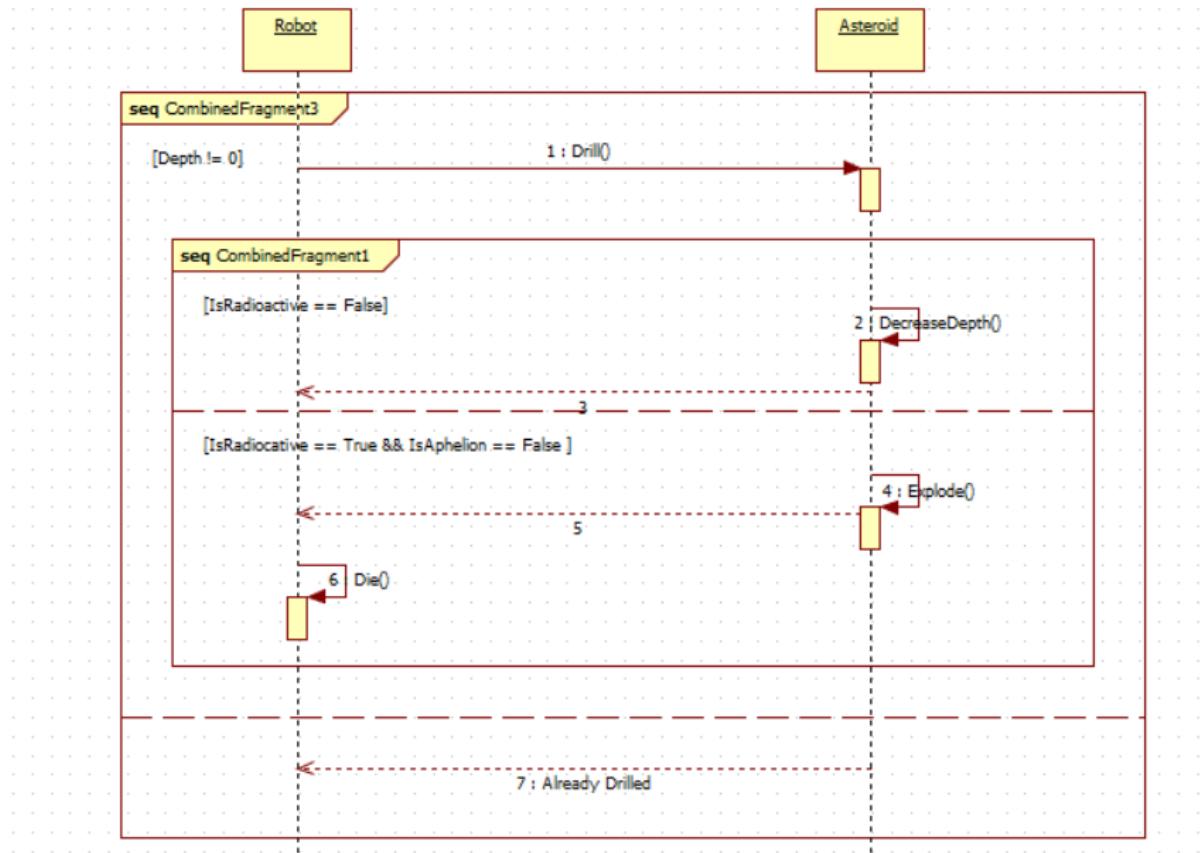
### 5.3.9 Robot is created (created, put on the asteroid): (by having enough source the settlers will build robots )

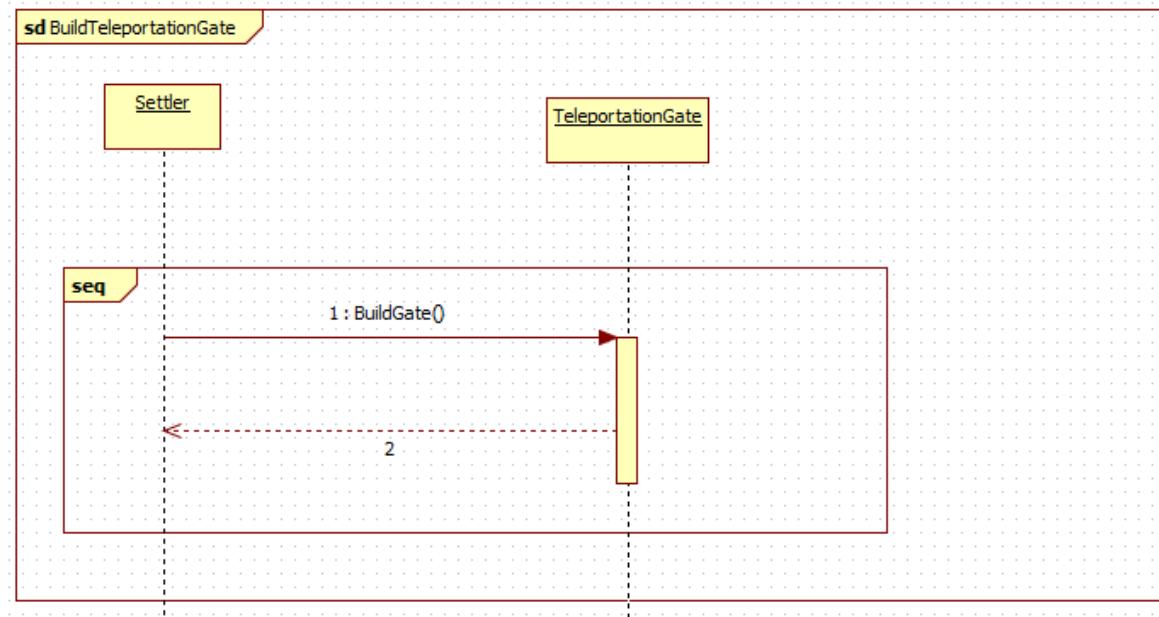


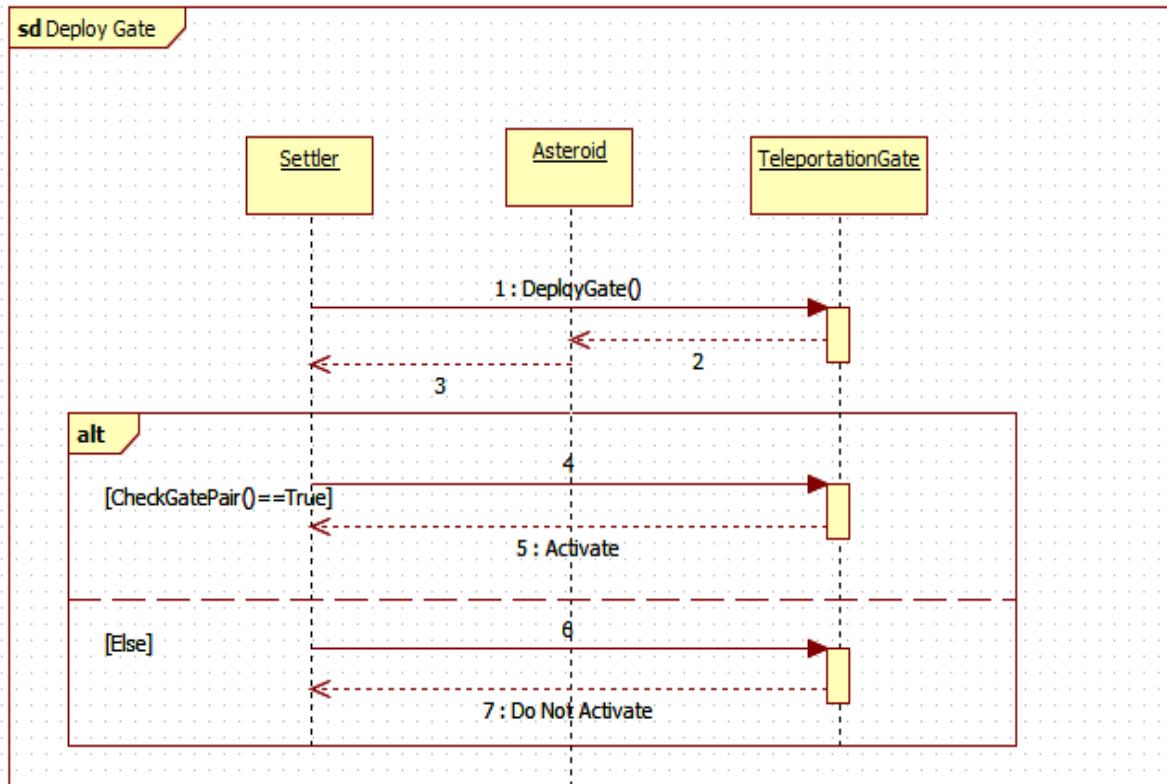
### 5.3.10 Robot moves: (Robots will move in belt )

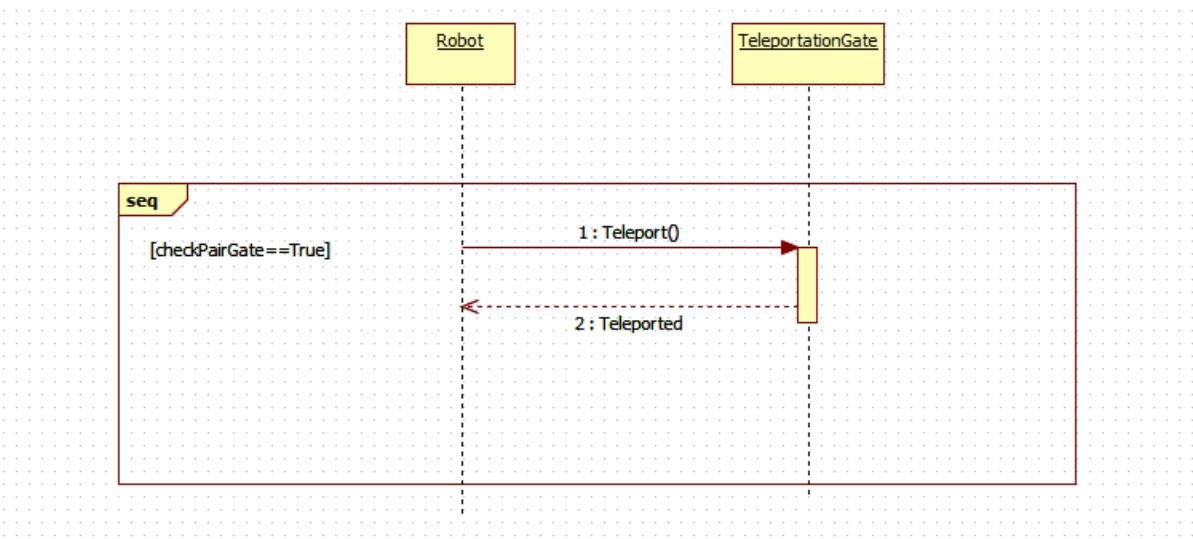


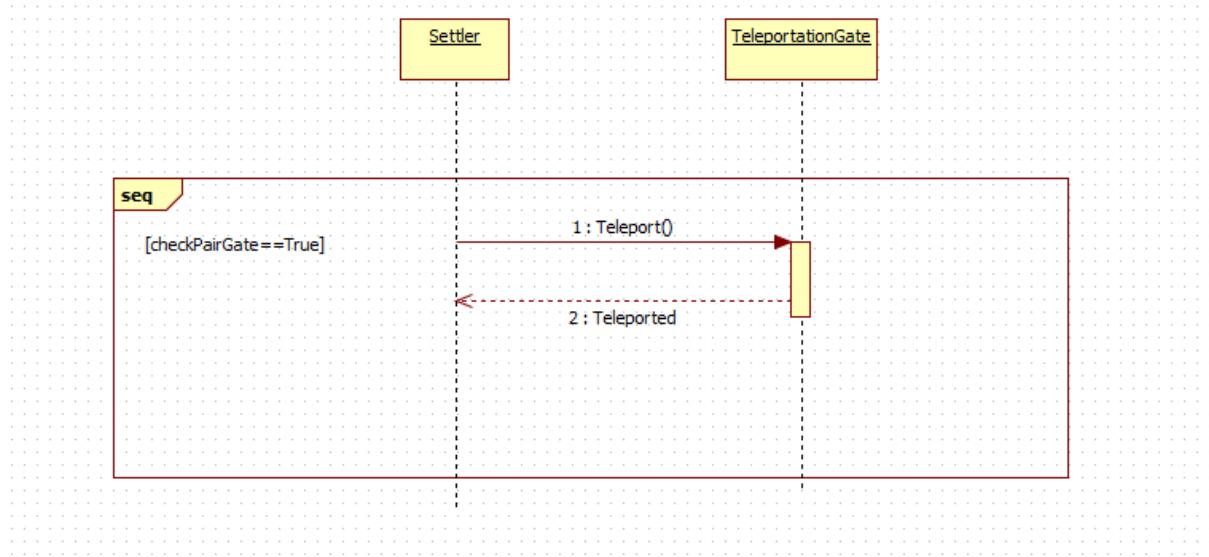
### 5.3.11 Robot drills: (Robot will star drilling an astroid )



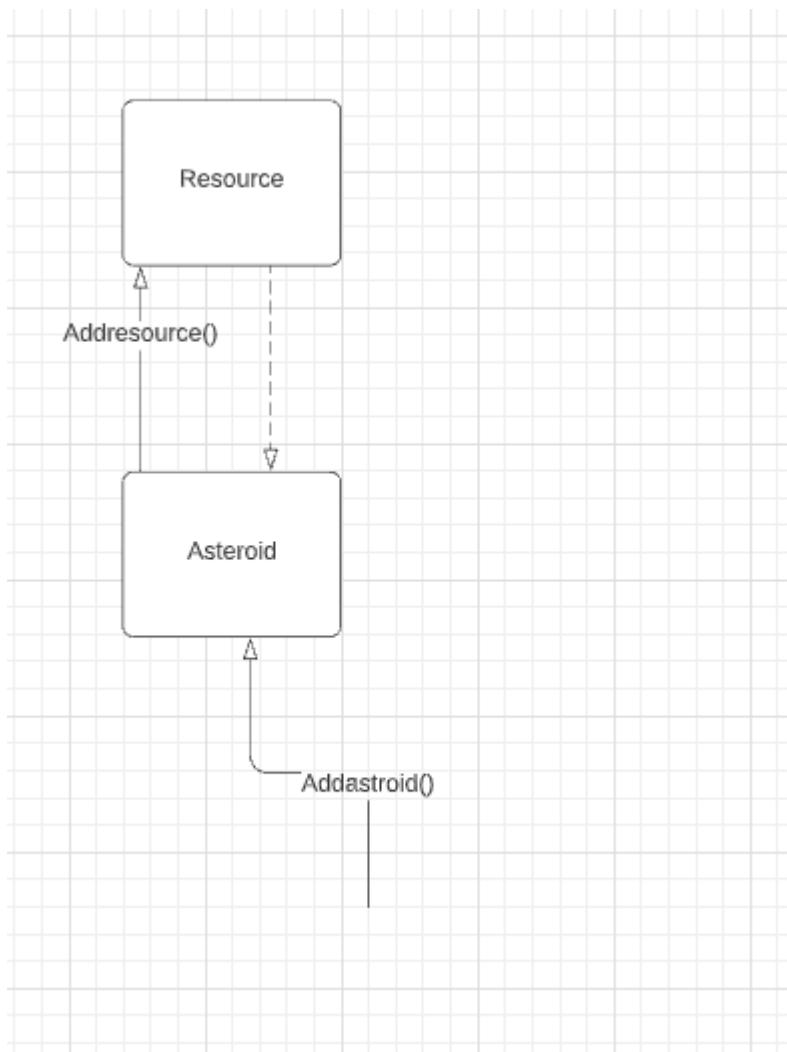
**5.3.12 Build Teleportation Gate:**

**5.3.13 Deploy Teleportation Gate:**

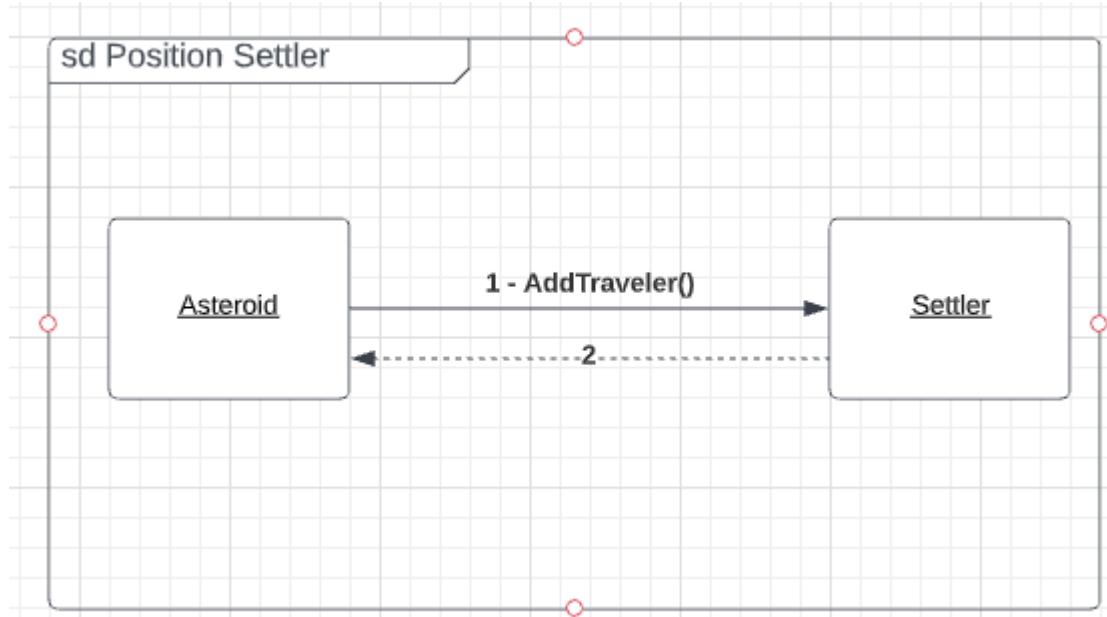
**5.3.14 Robot Teleportation:**

**5.3.14 Settler Teleportation:****5.4 Communication diagrams**

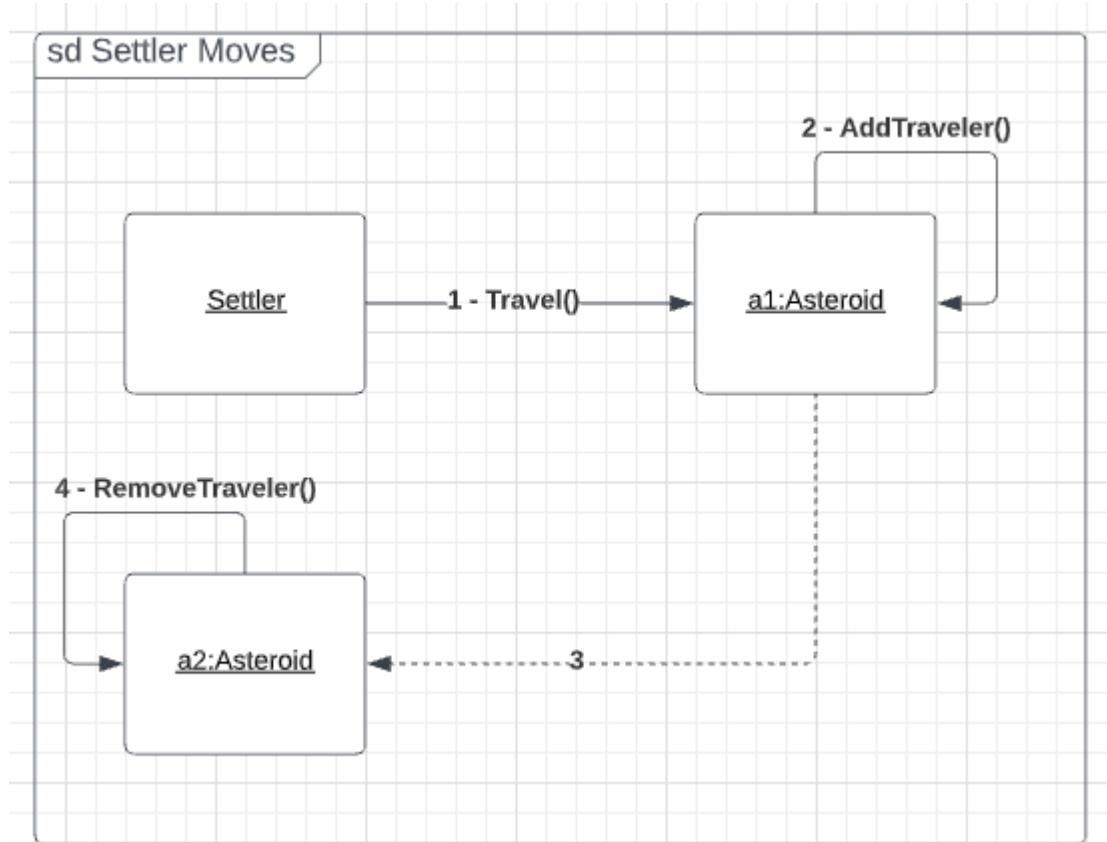
## 5.4.1) Game Initialisation



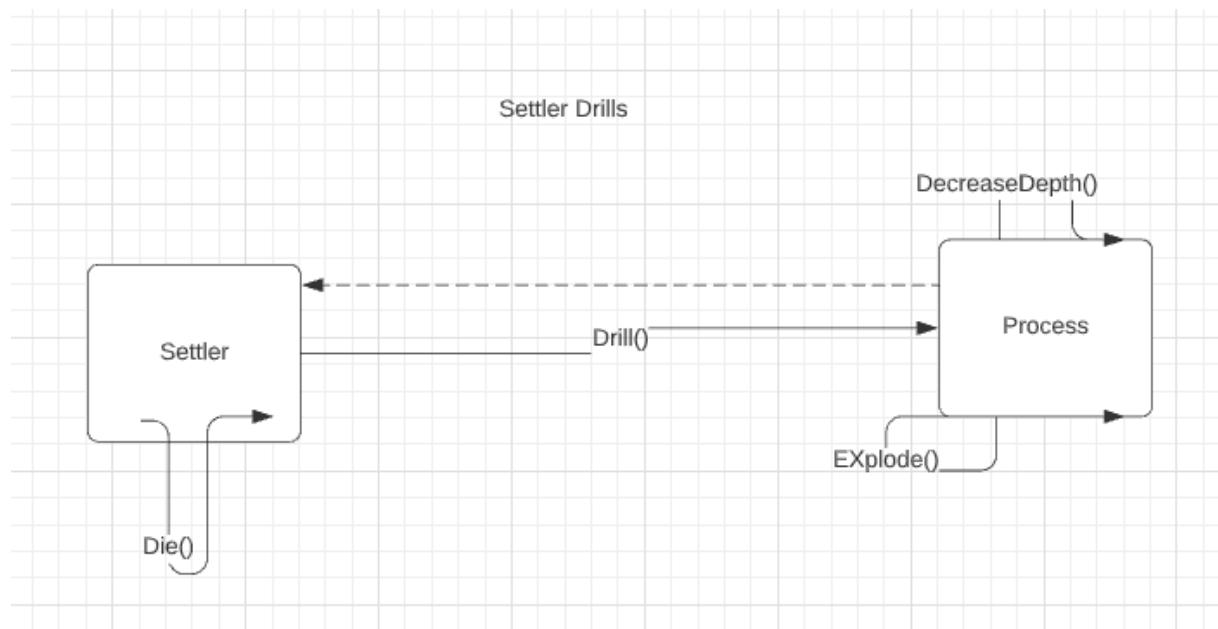
## 5.4.2 ) Position settler



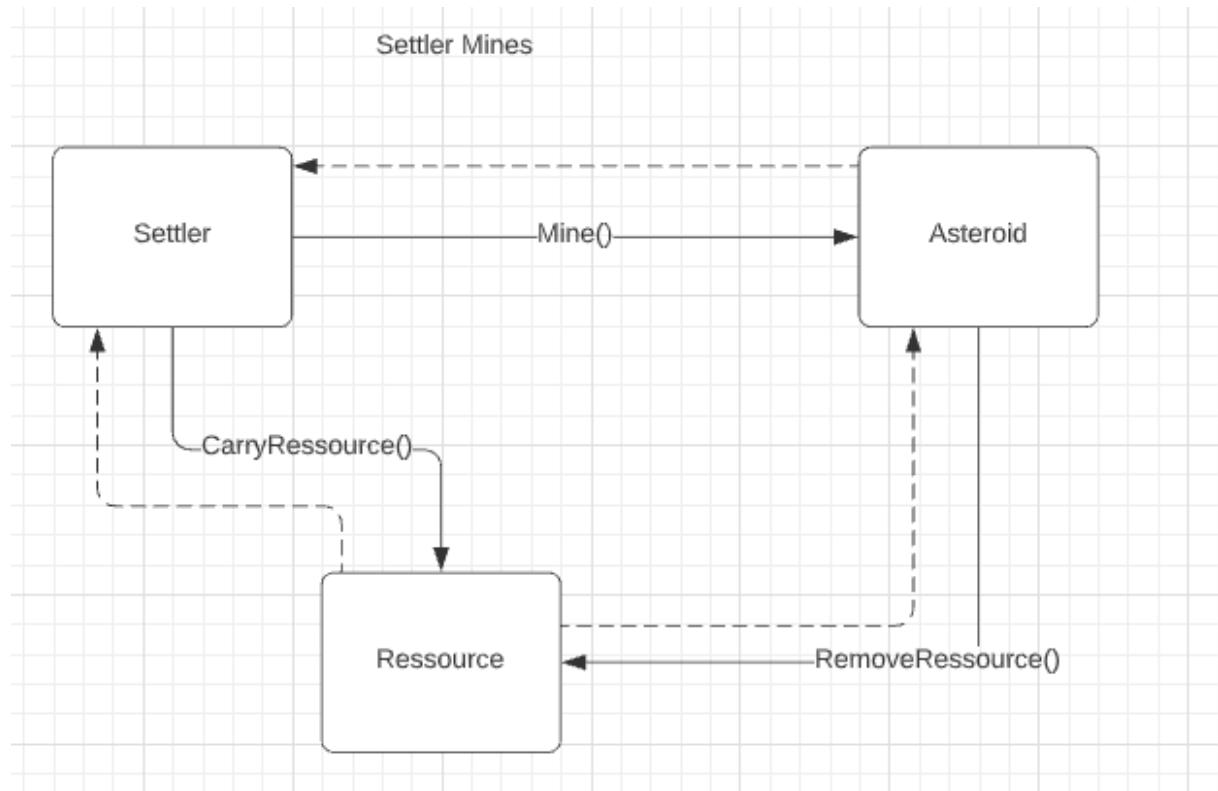
## 5.4.3) settler moves



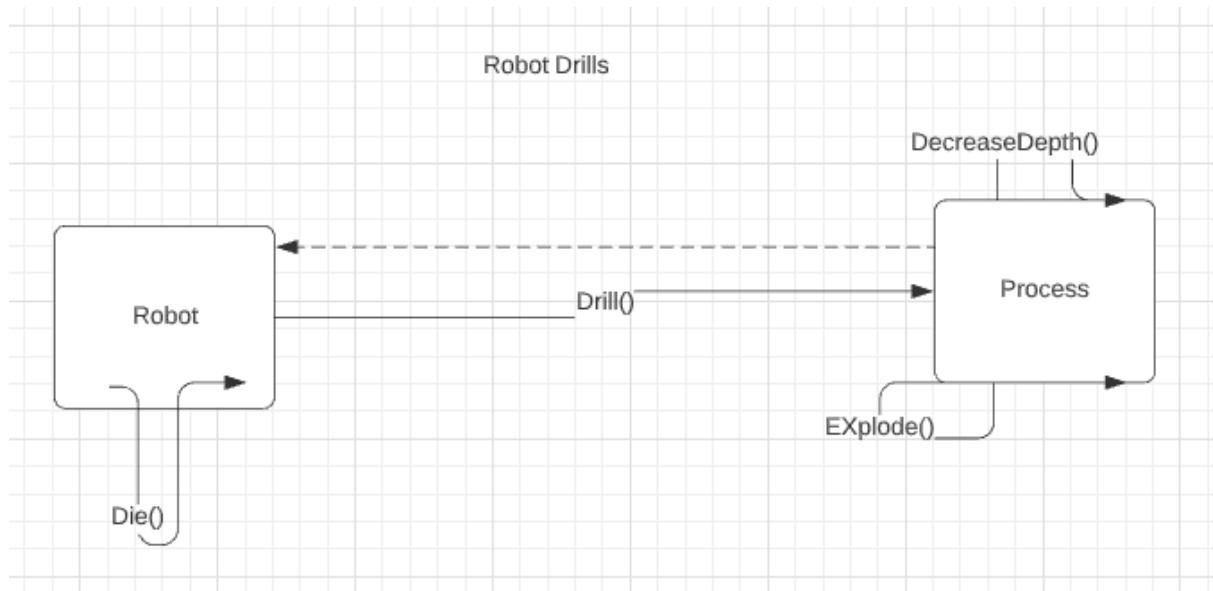
## 5.4.4) settler Drills



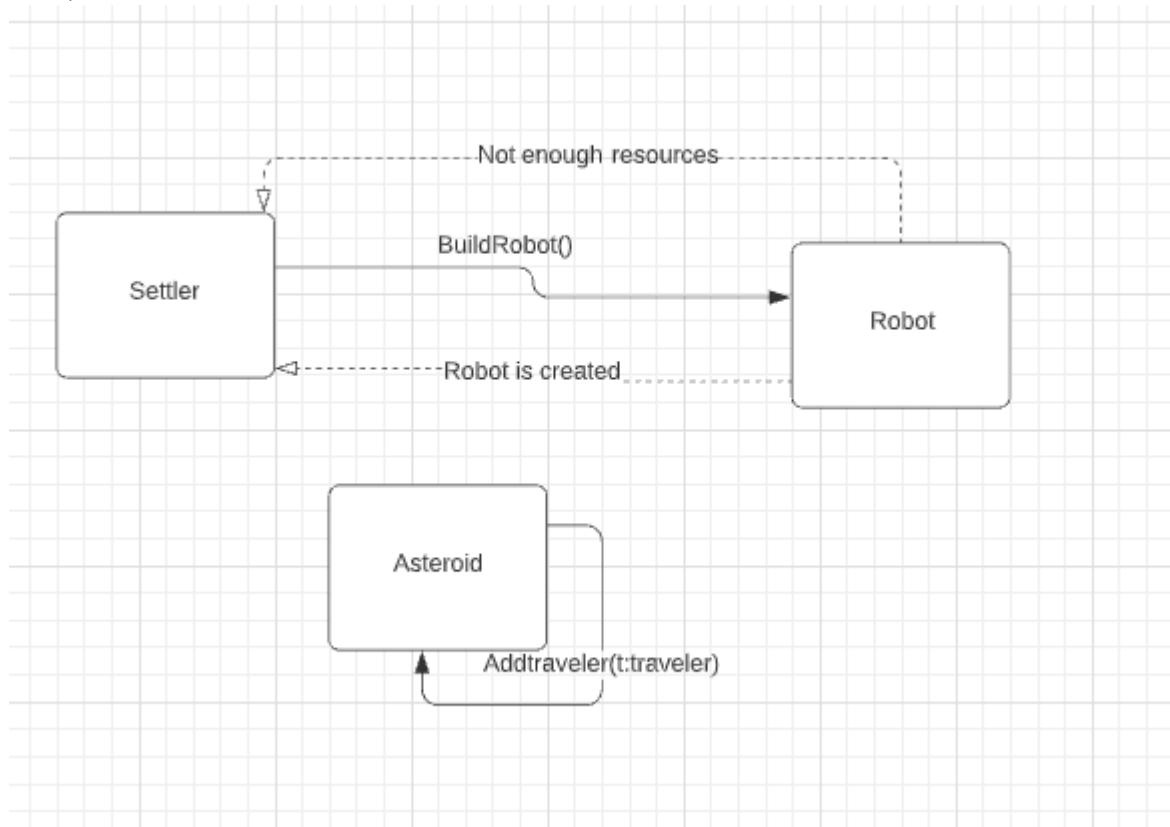
## 5.4.5) Settler Mines



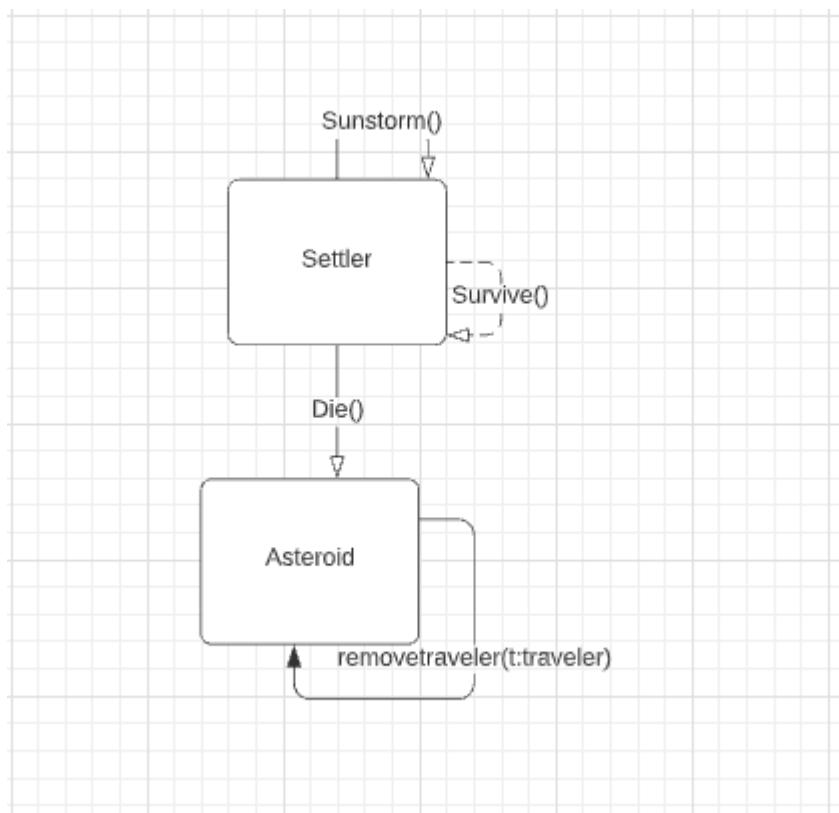
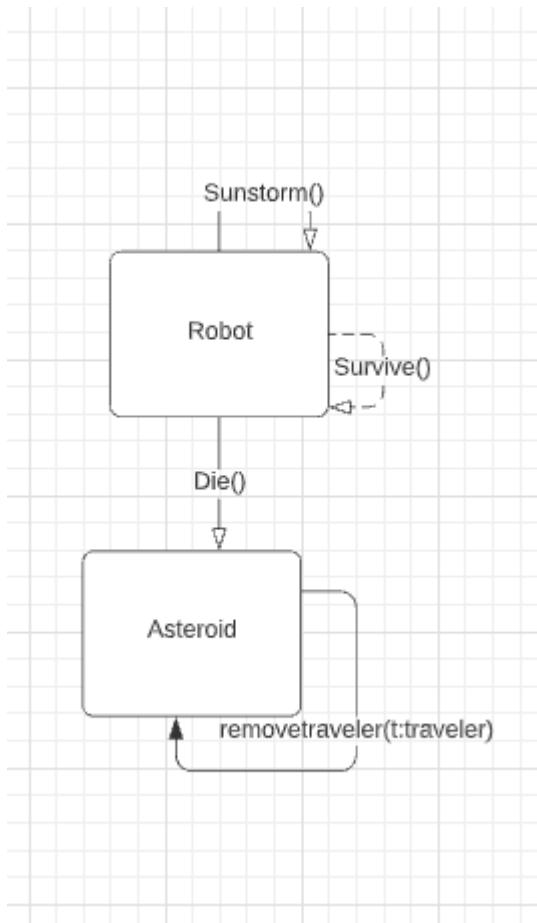
## 5.4.6) Robot drills



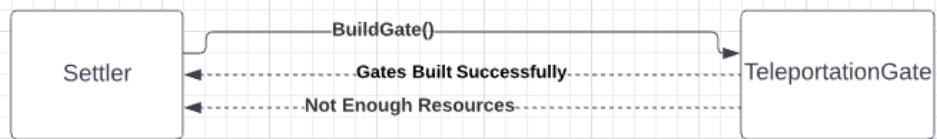
## 5.4.7) Create robot



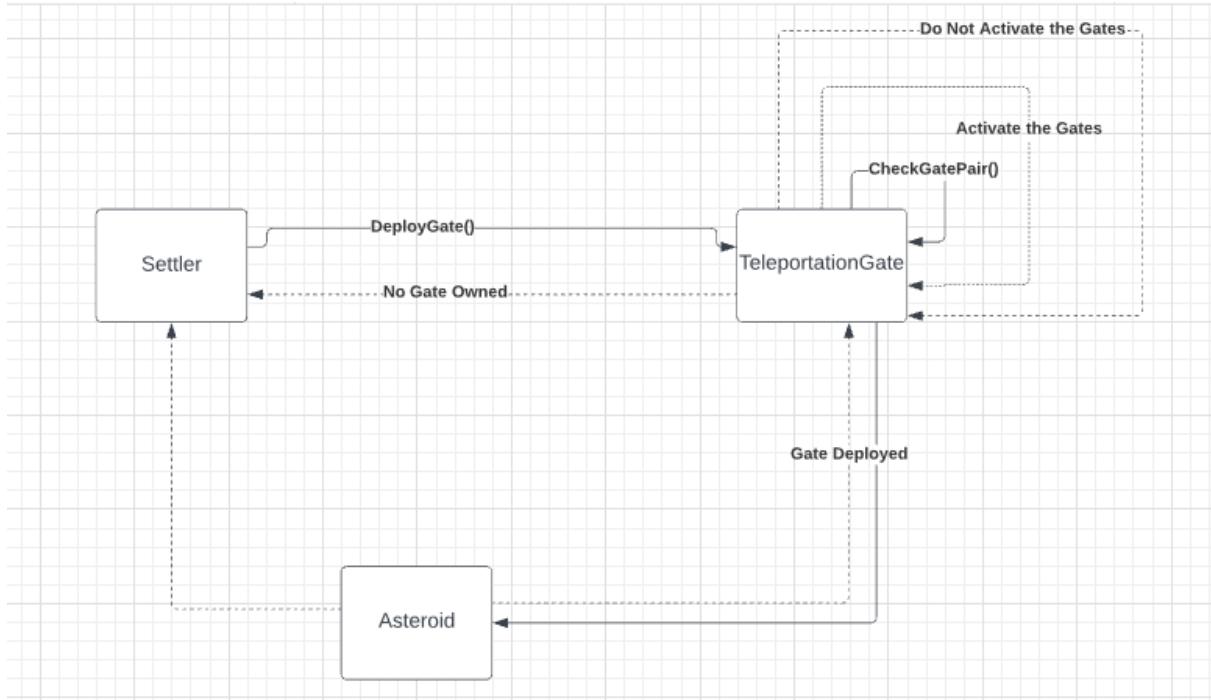
## 5.4.8) Sunstorm occurs ( 2 cases)



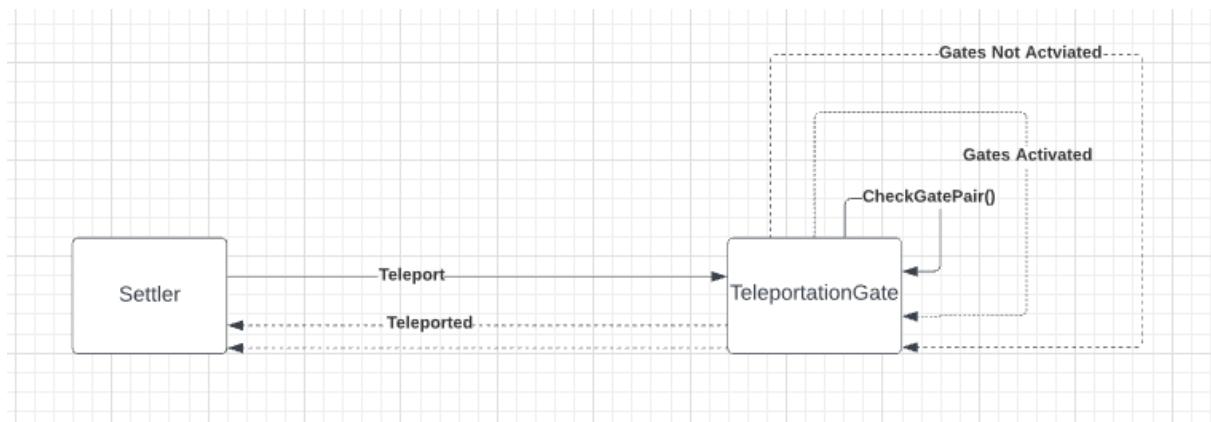
## 5.4.9) Build Teleportation Gates



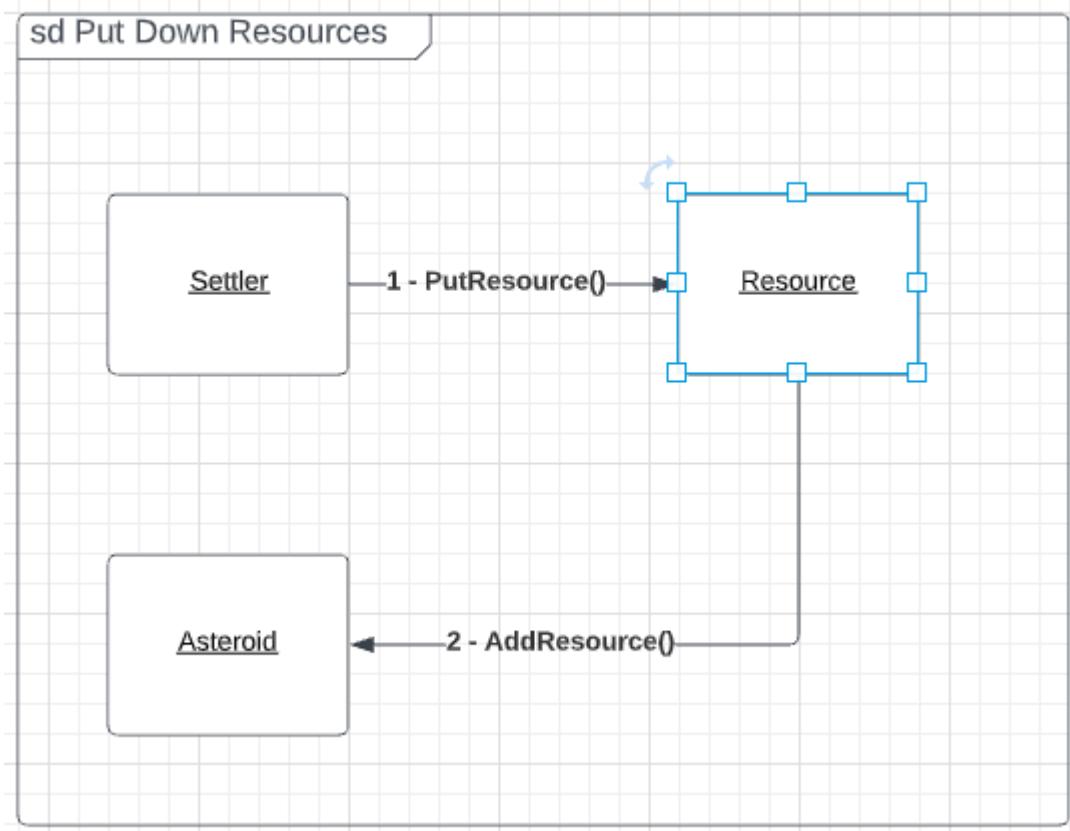
## 5.4.10) Deploy Teleportation Gate:



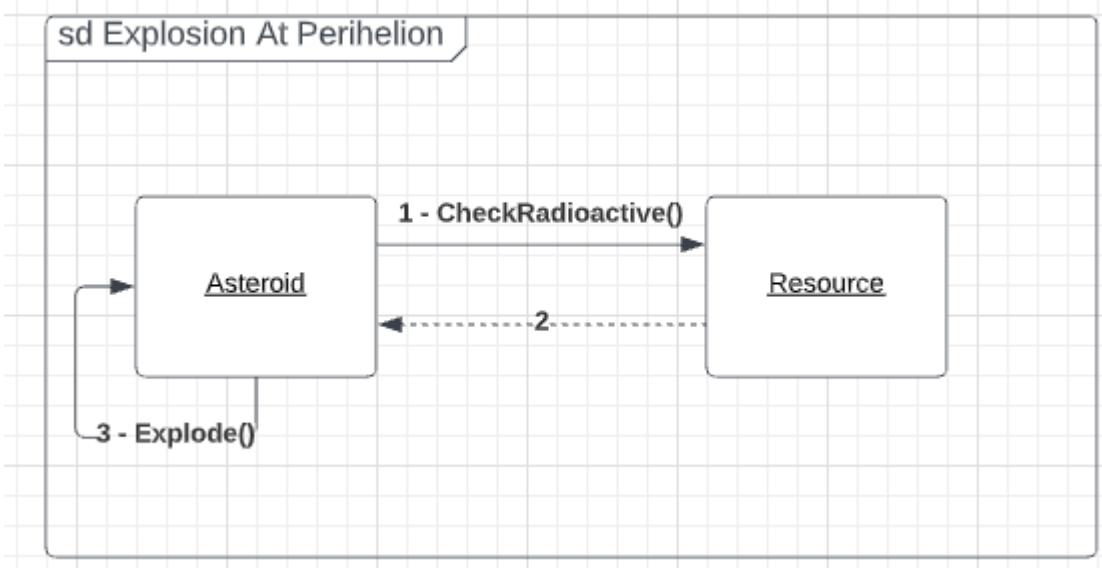
## 5.4.11) Game Initialisation Teleporting:



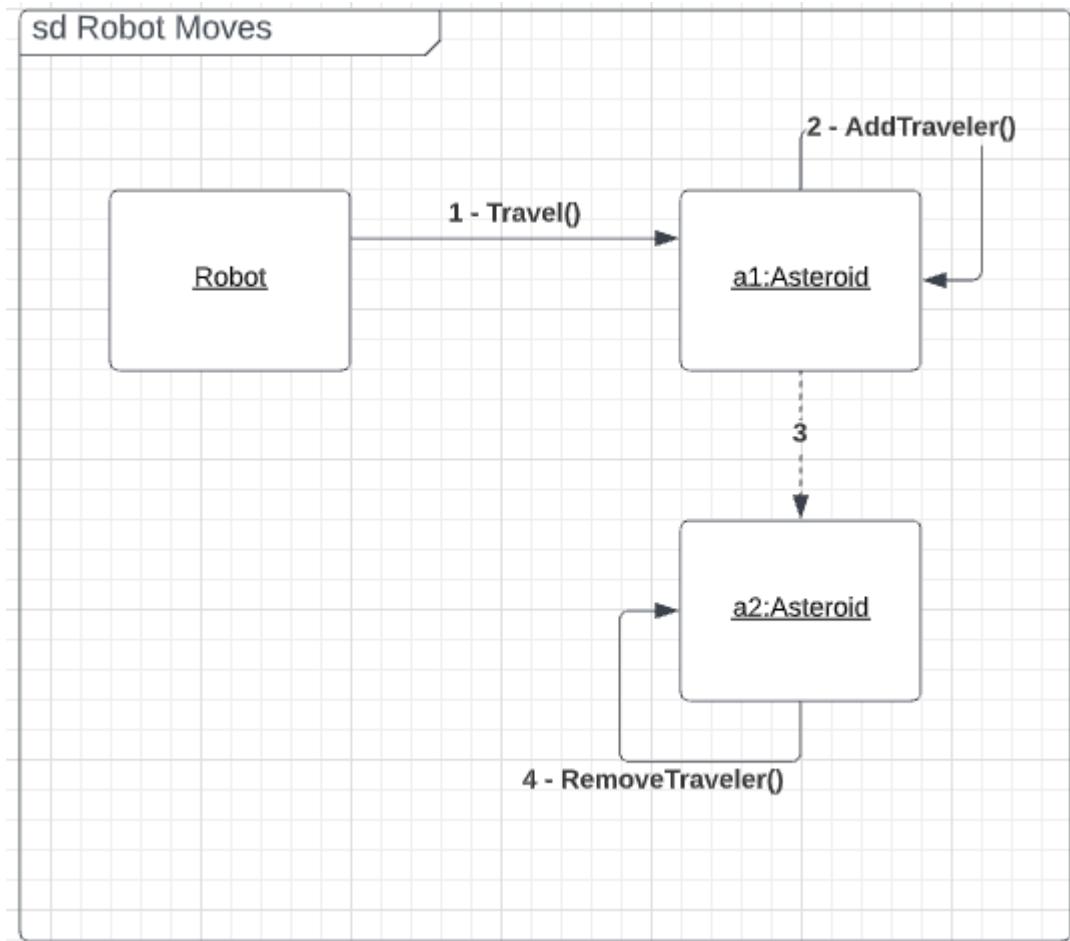
## 5.4.12) Put down resources



## 5.4.13) Explosion at perihelion



## 5.4.14) Robot moves

**5.5 Protocol**

Start (date & time)	Duration (hours)	Performer(s) name	Activity description
20/03/2022	3	Mahdi Yahia	Sequence diagrams description , Communication diagrams
20/03/2022	3.5	Mohammadparsa Kolivand	Description in sequence diagram , communication Digram
20/03/2022	3	Mohamed Amine Zaghdoudi	sequence description and communication diagrams
20/03/2022	3	Oussama dhouioui	sequence description and communication diagrams
20/03/2022	3	Kirolos Sawiris	use case / communication diagrams
20/03/2022	3	Ahmed Fathy	use case / communication diagrams
20/03/2022	3	Mohamed Abouzabady	use case / communication diagrams

# **[ASTEROID MINING]**

**[TEAM NNUMBER 7]**

Supervisor:  
**Dr. Balla Katalin**

## **Members:**

[Abouzabady Mohamed]	[GJVKZQ]	[mohamedabozabady1@gmail.com]
[Kirolos Sawiris]	[Y8B5KD]	[Kirolossawiris@gmail.com]
[Oussama dhouioui]	[XFCBM7]	[Oussemadhououi04@gmail.com]
[Mohamed Amine]	[ARPMPB]	[mzaghdoudi01@gmail.com]
[Ahmed Mousa]	[EWLQXU]	[amousa@edu.bme.hu]
[MohammadParsa Kolivand]	[RLU15I]	[parsakolivand@edu.bme.hu]
[Mahdi Yahia]	[WF72QQ]	[mahdiyahia18@gmail.com]

## 6. Skeleton program

### 6.1 Deployment guide

#### 6.1.1 List of files

File name	Size	Date	Content
Astroid	2KB	26/03/2022	Class Astroid -Attributes and Methods
Carbon	1KB	26/03/2022	Class Carbon -Attributes and Methods
Iron	1KB	26/03/2022	Class Iron -Attributes and Methods
Main	1KB	26/03/2022	Class Main -Attributes and Methods
Resources	1KB	26/03/2022	Class Resources -Attributes and Methods
Robot	1KB	26/03/2022	Class Robot -Attributes and Methods
Settler	1KB	26/03/2022	Class Settler -Attributes and Methods
SpaceStation	1KB	26/03/2022	Class SpaceStation -Attributes and Methods
TeleportationGate	1KB	26/03/2022	Class TeleportationGate -Attributes and Methods
Traveler	1KB	26/03/2022	Class Traveler -Attributes and Methods
Uranium	1KB	26/03/2022	Class Uranium -Attributes and Methods
WaterIce	1KB	26/03/2022	Class WaterIce -Attributes and Methods

### 6.1.2 Compilation

- For game to be successful the following operation is required

1)JDK (8 or later )

-The following IDEs have been used for developing

1 ) Eclips

2)IntelliJ IDEA

- Operation system required

1)Windows

### 6.1.3 Run

The user can follow provided steps to be able to Run the Game .

First download the resource file . Open the Preferred IDE

File ->open-> Open the Game file

On the “ Game Class “ The Main class should be found , Navigate to the “Public static void main() “ and Run the code .

Either IDE has a “Green play “ option to run the program or User can simply click on F5 or Shift-F10 (depending on the IDE)

Expected Run :

The case has been assigned in the Main Class ,

There are Four types of core Resources that can be selected ,

Carbon ,Uranium ,Iron ,IceWater

The Main Method will select 3 types of Resources which the user can carry on the Game !

Once the Resource is checked , there will be a check for RadioActive

Checking on Radioactive will be done by passing parameters the getDepth() method , if (Radioactive==True){ return True} else {false} .

Since multiple asteroids have been created , there will be getId method which will be assigned to the Traveler .

getID will also be initialized so later on Traveler would be able to find its neighbor

Furthermore in the following week , the option of Sunstorm and Hiding will be added so the User will decide about different possibilities .

## 6.2 Evaluation

Name of the team member	Participation (%)
Abouzabady Mohamed	14.28
Mohamed Amine Zaghdoudi	14.28
Oussama Dhououi	14.28
Kirolos Sawiris	14.28
Ahmed Mousa	14.28
Mohammadparsa Kolivand	14.28
Mahdi Yahia	14.28

## 6.3 Protocol

Start (date & time)	Duration (hours)	Performer(s) name	Activity description
26/03/2022	5 hours	Mohammadparsa Kolivand	Documentation Worked on Resource classes
25/03/2022	5 hours	Mohamed Amine Zaghdoudi	Worked on traveler, settler and robot classes
25/03/2022	5 hours	Oussama Dhououi	Worked on traveler, settler and robot classes
25/03/2022	5 hours	Abouzabady Mohamed	Worked on Resource,Astroid, TeleportationGate classes
25/03/2022	5 hours	Kirolos Sawiris	Worked on Resource,Astroid, TeleportationGate classes
25/03/2022	5 hours	Ahmed Mousa	Worked on Resource,Astroid, TeleportationGate classes
25/03/2022	5 hours	Mahdi Yahia	Worked on traveler, settler and robot classes

# **[ASTEROID MINING]**

**[TEAM NNUMBER 7]**

Supervisor:  
**Dr. Balla Katalin**

## **Members:**

[Abouzabady Mohamed]	[GJVKZQ]	[mohamedabozabady1@gmail.com]
[Kirolos Sawiris]	[Y8B5KD]	[Kirolossawiris@gmail.com]
[Oussama dhouioui]	[XFCBM7]	[Oussemadhououi04@gmail.com]
[Mohamed Amine]	[ARPMPB]	[mzaghdoudi01@gmail.com]
[Ahmed Mousa]	[EWLQXU]	[amousa@edu.bme.hu]
[MohammadParsa Kolivand]	[RLU15I]	[parsakolivand@edu.bme.hu]
[Mahdi Yahia]	[WF72QQ]	[mahdiyahia18@gmail.com]

## 7. Concept of prototype

### 7.1 Interface definition of Prototype

#### 7.1.1 General description

The interface will consist of two input/output files; the game functions will be saved in the input file, and the output file will contain the results after the changes. Using the standard input, the user can choose which function to perform from the input file (eg. each function can correspond to a key). It can also be used for testing; running the functions in the input file with different defined parameters for different scenarios (the parameters can be saved in another file) and saving the test case results in the output file; both implemented on a character level for commands and separators.

#### 7.1.2 Input language

##### *Travel*

**Description:** This command is for moving a Traveler object whether the settler or the robot.

**Options:** go to the above asteroid: up arrow, go to the below asteroid: down arrow, go to the left asteroid: left arrow, go to the right asteroid: right arrow

##### *Drill*

**Description:** this command is for drilling the mantle of the asteroid by the settler or the robot.

**Options:** Drill: D

##### *Mine*

**Description:** this command is for mining the resource that exists in the core of the asteroid by the settler.

**Options:** Mine: M

##### *Build*

**Description:** this command is for building a robot, a teleportation gate, or a space station

**Options:** build robot: R, build teleportation gate: T, build space station: S

##### *Hide*

**Description:** this command is for Hiding the settler of the robot.

**Options:** Hide: H

##### *Fill*

**Description:** this command is for filling the settler the core of a hollow asteroid with resources

**Options:** Fill: F

***CarryResource***

**Description:** this command is for the settler to carry resources.

**Options:** Carry: C

***PutResource***

**Description:** this command is for the settler to carry resources.

**Options:** Put: P

***DeployGate***

**Description:** this command is for the settler to deploy the gate.

**Options:** Deploy: G

### 7.1.3 Output language

***Travel:***

You traveled to the next asteroid successfully!

Cannot travel, there are no neighboring asteroids!

***Drill:***

You drilled the mantle successfully the new depth is decreased by 1!

cannot drill the asteroid is fully drilled!

***Mine:***

you mined the core successfully, and a Resource is added to the asteroid.

cannot mine the asteroid is not fully drilled or there are no resources to mine.

***Build:***

Not enough resources to build -object.

Object- successfully built.

***Hide :***

You are safe.

You are still in a danger zone.

***Fill :***

Resources successfully installed.

No resources to install.

This asteroid is not hollow.

***CarryResource :***

**Resources successfully loaded.**

**Not enough capacity.**

**No resources to load.**

***PutResource:***

**Resources successfully installed.**

**No resources to install.**

***DeployGate:***

**Gate deployed successfully.**

**No gate to deploy.**

## 7.2 Real use-cases

<b>Use-case name</b>	<b>Travel</b>
<b>Short textual description</b>	<b>The Settler or robot can travel to other asteroids.</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>The settler can travel to neighboring asteroids(Multiple Hundreds of neighbors)</b>

<b>Use-case name</b>	<b>Drill</b>
<b>Short textual description</b>	<b>Both settlers and robots can drill in the asteroids</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>Settler or robot can deepen the hole in the mantle with one unit</b>

<b>Use-case name</b>	<b>Mine</b>
<b>Short textual description</b>	<b>Settlers can mine for resources</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>They extract the resources from the core of the asteroids and that is only possible if the mantle has been completely drilled</b>

<b>Use-case name</b>	<b>Build Robots</b>
<b>Short textual description</b>	<b>Settler can build robots to perform tasks</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>Using a unit of iron, a unit of carbon, and a unit of uranium, the settlers can build autonomous robots controlled by artificial intelligence.</b>

<b>Use-case name</b>	<b>Build gates</b>
<b>Short textual description</b>	<b>Settler can build gates</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>Using up a unit of iron, a unit of water ice , a unit or carbon, and a single unit of uranium, the settler can build a pair of teleportation gates.</b>

<b>Use-case name</b>	<b>Hide</b>
<b>Short textual description</b>	<b>Both robots and settlers can get damaged from sun storm</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>Both robots and settlers can get damaged until they hide in a hollow asteroid</b>

<b>Use-case name</b>	<b>Carry Resources</b>
<b>Short textual description</b>	<b>Settlers and robots can carry resources and move them</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>A single settler can carry 10 units of resources since that is the capacity of space station</b>

<b>Use-case name</b>	<b>Put resources</b>
<b>Short textual description</b>	<b>Settlers put down the carried resources</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>Settlers can set the carried resources to a given asteroid</b>

<b>Use-case name</b>	<b>Deploy gates</b>
<b>Short textual description</b>	<b>Gates can be deployed</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>The two gates of a pair remain in contact for good and entering the settler and robot.</b>

<b>Use-case name</b>	<b>Fill with resources</b>
<b>Short textual description</b>	<b>Settlers can hide the resources</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>Settlers can fill a hollow asteroid with a unit of resources.</b>

<b>Use-case name</b>	<b>Create Sun Storm</b>
<b>Short textual description</b>	<b>Sun storm cause some damage to settlers and robots</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>Sun storm do damage robots unless they hide in a hollow asteroid</b>

<b>Use-case name</b>	<b>Add asteroids to the game</b>
<b>Short textual description</b>	<b>When the player start the game can find asteroids</b>
<b>Actors</b>	<b>Controller</b>
<b>Dialog, scenario</b>	<b>Creating some asteroids in the game and add the resources to each</b>

<b>Use-case name</b>	<b>Start the game</b>
<b>Short textual description</b>	<b>The player can go to start the game option</b>
<b>Actors</b>	<b>Players</b>
<b>Dialog, scenario</b>	<b>It set the player score back to 0 and put the resources and settlers at their appropriate position</b>

<b>Use-case name</b>	<b>Win</b>
<b>Short textual description</b>	<b>The player wins the game</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>If the settlers can collect three units of each resource and collect those materials on a single asteroid, they can build a space station and win the game.</b>

<b>Use-case name</b>	<b>Lose</b>
<b>Short textual description</b>	<b>The player loses the game</b>
<b>Actors</b>	<b>Player</b>
<b>Dialog, scenario</b>	<b>If the settler died</b>

### 7.3 Test plan

<b>Name of the test-case</b>	<b>Settler moves:</b>
<b>Goal</b>	To verify if a traveler(user) is standing in a specific location.
<b>Short description</b>	The test will verify the location of the player at a specific moment in time.

<b>Name of the test-case</b>	<b>Sun Storm death</b>
<b>Goal</b>	To verify if the player will get the results he is supposed to get in case of a sunstorm.
<b>Short description</b>	We will create a sun storm of a radius that will make a player hit by the sun storm and check if that will kill him in case he is not hidden, and if he will be protected in case he has hided.

<b>Name of the test-case</b>	<b>Settler drills</b>
<b>Goal</b>	To check that the drilling operation works without problems.
<b>Short description</b>	The player will drill an asteroid that is not hollow and we will verify that the depth of the asteroid decreased by one or not.

<b>Name of the test-case</b>	<b>Settler mines</b>
<b>Goal</b>	To check the behavior if we collect resources from an asteroid that is drilled through.
<b>Short description</b>	We will test the owned resources of the player and the resources in the core of the asteroid and verify that the resources in the core of the asteroid will be 0, and the resources owned by the player will be incremented.

<b>Name of the test-case</b>	<b>Put Down Resources</b>
<b>Goal</b>	To check the behavior of the program after we put down carried resource
<b>Short description</b>	We will check that owned resources by the player will be decremented and that the resource of the asteroid will be incremented.

<b>Name of the test-case</b>	Hidden resources in Asteroid
<b>Goal</b>	To check the Availability of the resources .
<b>Short description</b>	Checks if the asteroid is empty, Check the availability of the resources . It also need the Traveler status to hidden .

<b>Name of the test-case</b>	check operation
<b>Goal</b>	Check the operation
<b>Short description</b>	The settler can not have multiple operation at once so this test-case will verify that

<b>Name of the test-case</b>	inventory Availability
<b>Goal</b>	Check the added gate
<b>Short description</b>	By using e DeployGate() we will verify if the gate have been added . Also CheckGatePair(). will be used to check with previous Gate .

<b>Name of the test-case</b>	Gate activation
<b>Goal</b>	verify if the gates are active .
<b>Short description</b>	By using Teleport() , we can check the Gates , If it return true , It will place the settler in astroid

<b>Name of the test-case</b>	Robot placement
<b>Goal</b>	Check the availability of the Robots
<b>Short description</b>	With enough resources and Buildrobot() we will check if the robot has been created , If yes It should return true ;

<b>Name of the test-case</b>	Check explosion
<b>Goal</b>	Will check the Aphelion's return and RadioActive to verify if the asteroid will explode or not.
<b>Short description</b>	By using RadioActive() =true, IsAphelion= false then use the Explode()'s return to check if the explosion happened

## 7.4 Support Programs

The support Program will be created to provide the environment needed for testing, it will instantiate the game, generate the field with all the components that will make the game playable.

The tester will be presented with a set of test cases, with the required output to check how accurate is our implementation of the code.

## 7.5 Protocol

Start (date & time)	Duration (hours)	Performer(s) name	Activity description
03/04/2022	2 hours	Mahdi Yahia	Test -Plan and support programs
03/04/2022	2 hours	Mohammadparsa Kolivand	<i>working on test plan and checking the details for support programs</i>
02/04/2022	3 hours	Mohamed Abouzabady	Real-use Cases
02/04/2022	3 hours	Ahmed Mousa, Kirolos Sawiras	Interface definition of Prototype
04/04/2022	1.5 hours	Mohamed Amine Zaghdoudi	Reviewing work and verifying coherency
04/04/2022	1.5 hours	Dhouioui oussama	Reviewing work and verifying coherency

# **[ASTEROID MINING]**

**[TEAM NNUMBER 7]**

Supervisor:  
**Dr. Balla Katalin**

## **Members:**

[Abouzabady Mohamed]	[GJVKZQ]	[mohamedabozabady1@gmail.com]
[Kirolos Sawiris]	[Y8B5KD]	[Kirolossawiris@gmail.com]
[Oussama dhouioui]	[XFCBM7]	[Oussemadhououi04@gmail.com]
[Mohamed Amine]	[ARPMPB]	[mzaghdoudi01@gmail.com]
[Ahmed Mousa]	[EWLQXU]	[amousa@edu.bme.hu]
[MohammadParsa Kolivand]	[RLU15I]	[parsakolivand@edu.bme.hu]
[Mahdi Yahia]	[WF72QQ]	[mahdiyahia18@gmail.com]

## 8. Detailed plans

### 8.1 Design level plan of classes

#### 8.1.1 Settler

- **Responsibility**

It is one of the most significant classes in this game, since the settler is the main actor that represents the player's moves, and actions directly.

The main goal of the settler is to build a space station then the player wins.

- **Superclasses**

Traveler

- **Interfaces**

None

- **Attributes**

- **private ArrayList<Resource> resources:** ArrayList contains a list of the class Resource that the settler manages to get and those are counted as available resources that can be used to build (Gates, Robots, Space Station, etc..)
- **private ArrayList< TeleportationGate > TeleportationGates:** It has the data of the pair of gates that the settler brings.

- **Methods**

- **public bool buildRobot():** This method is used to build the robots, and it is one of the operations that the settler can do. After building the robot, it can help in performing other operations like traveling and drilling. The function checks that the settler collected enough resources to build a robot then it returns true to add a new robot to the game, otherwise, it returns false.
- **public bool buildGate ():** It represents the operation of building a pair of gates using up two units of iron, a single unit of water ice, and a single unit of Uranium. The gates can be deployed later, they are used for entering whether the traveler is a robot or a settler. The method checks if the settler collected enough resources to build gates then it returns true to add a new pair of gates and connect them to the asteroid, otherwise it returns false.
- **public bool buildSpaceStation():** It is the final purpose of the game, that if the settler reaches it will win the game. They can mine at least three units of each resource and they collect those materials on a single asteroid, they can build the space station and the players win the game. The method checks that the settler collected enough resources to build a robot then it returns true to add a new robot to the game, otherwise, it returns false.
- **public void die():** The settler dies if it cannot survive the sun storm or the radioactive materials. The asteroid removes the dead settler from its list of travelers.

- **public void mine():** It represents the operation of mining that the settler usually does to extract materials that will help in building the space station. Also, settlers can only mine when the asteroid is at aphelion. The mined resources are deducted from the amount of the resources (it represents the resource amount in the core of the asteroid and ready to be mined) of its asteroid as well as a new object will be added to the ArrayList of the type Resource (It represents the resources that are ready to be used to building objects.), the settler can mine a unit per each time.
- **public void fill():** A hollow asteroid can be filled with a unit of resource, counting as an operation of the operations the settler can do. Once the settler fill a hollow asteroid, the resource will be deducted from the ArrayList and added to the asteroid core resources (As It will not be ready to be used without mining again.)
- **public void hide():** A settler can only survive a sun storm if they hide in the core of a hollow asteroid. This, of course, is only possible if the mantle has been drilled through. Successful hide operation has 2 conditions: the value of resource amount of the asteroid is equal to 0 (That means it has no resources), the second condition is the asteroid depth is equal to 0 (That means the asteroid is fully drilled)
- **public void deployGate(TeleportationGate g, Asteroid a):** The gates can be deployed in the vicinity of the asteroid the settler is on, the method uses two parameters, the current asteroid and the gates which will be deployed, and also add the gates to the TeleportationGates list of each asteroid.
- **public void Travel(Asteroid a):** The settler can travel between the asteroids, to mine and extract resources, the method uses the asteroid as a parameter and once the settler reached the asteroid, it will be added to the list of travelers of the current asteroid and removed from the previous asteroid.
- **public void drill():** It is one of the operations that the settler can do, the settler can drill in the asteroids, they travel to so they can mine for resources. The depth variable of the asteroid is always set to the max. The more the settler drill, the less depth will be, until the asteroid is fully drilled then the depth will become Zero. The settler drill one unit each time the function is called
- **public void carryResources():** A single settler can only carry 10 units of resources, that's the spaceships' capacity. The carried resources will be deducted from the resources list of the asteroid and added to the core resources, it is a condition that the settler can not use the carried resources until they are put back. The settler carries one unit each time the function is called
- **public void putResources():** The settler can put the carried resources in the desired place, travel to other asteroids, and move the resources out from the spaceship. Once the settler puts the resources, they are counted again in the available resources of the current asteroid to be used in building objects. The settler put one unit each time the function is called.

### 8.1.2 Asteroid

- **Responsibility**

This class handles the Asteroids in the asteroid belt. an object of this class is a single asteroid, when created, an identifier is assigned to it, along with the asteroid specs and location. This class handles the relation between the asteroids and each other, and what affects the asteroid from different classes objects. Settlers wander in the asteroid belt with single-person spaceships looking for resources

- **Superclasses**

None

- **Interfaces**

None

- **Attributes**

- **private string ID:** each asteroid has a unique ID assigned to it at the creation of the asteroid. It is very critical to differentiate between the asteroids. The different IDs will be used in determining neighbors and the Travel method restrictions for Traveler objects. initialized in an order in the constructor.
- **private int Depth:** each asteroid has a depth, The depth of this rock mantle can vary from asteroid to asteroid, the depth is always set to the max and decreased gradually if the settler drilled more units.
- **private int ResourceAmount:** it stores the number of resource units of each asteroid that exist in the core and the settler can mine (They are not counted as the available resources and cannot be used directly without mining).
- **private Resource CoreResource** it represents the type of resources that exist inside the asteroid and is counted as available resources that can be used in building objects
- **private bool IsAphelion:** it represents the state of the asteroid if it is aphelion or not. Accordingly, that decides the behavior of the asteroid whether if it returns false, the asteroid will explode (If it's fully drilled and contains radioactive resources). Otherwise, it doesn't explode.
- **private ArrayList<Resource> resources:** an ArrayList of resource objects that contains the resources that the asteroid has (an object per unit)and is available to be used 0 or many also it stores all the data of the resources such as the type.
- **private ArrayList<Traveler> travelers:** it shows the travelers(Robots and Settlers) that are currently on a given asteroid.
- **private ArrayList<Asteroid> neighbors:** contains the neighboring asteroids list.

- **Methods**

- **Void Explode():** When a fully drilled asteroid with a radioactive core is at perihelion, the asteroid explodes and kills any settler on it. The settlers and robots are removed from the travelers' list of the given asteroid.
- **Void addTraveller(Traveler t):** When a settler or robot travel to the asteroid, it will be added through this method to the traveler list
- **Void addResources(Resource r):** it represents the available resources to be used in the asteroid, mining the resources and extracting them, will be added using this method since they will be available to be used, also carrying and putting the resources from different asteroids will be added through this method.

- **Void removeTraveller(Traveler t):** When a settler or robot travels from the asteroid or die, it will be removed through this method from the traveler list.
- **Void removeResource(Resource r):** The resources will be carried into the spaceship to be moved to other asteroids, will be deducted directly from the current asteroid and added to the new one, and also will be removed if the settler decided to hide the resource in the core of an asteroid
- **Void DecreaseDepth():** Decreases the depth variable by one when a settler does a mine operation on the asteroid.

### 8.1.3 SpaceStation

- **Responsibility**

A Class for the Space Station that is sought to be created by the settlers when they collect enough resources. creating one object of this class will get the settlers the win.

- **Superclasses**

*None*

- **Interfaces**

*None*

- **Attributes**

*None*

- **Methods**

*None*

### 8.1.4 Robot

- **Responsibility**

*The settler built the robot and the robot can perform some common operations that the settler normally does.*

- **Superclasses**

*Traveler*

- **Interfaces**

No

- **Attributes**

- **String ID:** each robot has a unique ID, and this represents the ID of the robot.

- **Bool Hidden:** It returns if the traveler is hidden or not upon that the behavior of the traveler will be decided when the sun storm comes

- **Methods**

- **Void die()**: Sun storms do damage robots unless they hide in a hollow asteroid. The asteroid removes the dead settler from its list of travelers
- **Void hide()**: Robots hide in a hollow asteroid, so they can survive the sun storm.
- **Void drill()**: The robots can drill holes in the asteroids although they can not mine. They can drill one unit per method call.
- **Void travel(Asteroid a)**: The robot travels between asteroids, to do a mission or to survive the radioactive explosions. The robot who leaves the asteroid will be deleted through remove traveler method and added to the current asteroid.

### 8.1.5 Resource

- **Responsibility**

The class contains all the information of the resources minded and available to be used in building the space station and shows the type of each as well.

- **Superclasses**

None

- **Interfaces**

None

- **Attributes**

- **private bool IsRadioActive**: That refers if the resource is radioactive or not, accordingly the resource can do radioactive explosions.

- **Methods**

None

### 8.1.6 WaterIce

- **Responsibility**

It is one of the main resources that is used to build the space station.

- **Superclasses**

Resource

- **Interfaces**

None

- **Attributes**

None

- **Methods**

- **public void Sublimates()**: When a fully drilled asteroid with water ice in its core is at perihelion, the water ice sublimates.

### 8.1.7 Carbon

- **Responsibility**

It is one of the main resources that is used to build the space station.

- **Superclasses**

Resource

- **Interfaces**

None

- **Attributes**

None

- **Methods**

None

### 8.1.8 Uranium

- **Responsibility**

It is one of the main resources that is used to build the space station.

- **Superclasses**

Resource

- **Interfaces**

None

- **Attributes**

None

- **Methods**

None

### 8.1.9 Iron

- **Responsibility**

It is one of the main resources that is used to build the space station.

- **Superclasses**

Resource

- **Interfaces**

None

- **Attributes**

*None*

- **Methods**

*None*

### **8.1.10 Traveller**

- **Responsibility**

It is the base class that is extended in two classes (Settler, Robot). It contains the common operations between the settler and the robot, it is also an abstract class so no instances can be initialized.

- **Superclasses**

*None*

- **Interfaces**

*None*

- **Attributes**

*None*

- **Methods**

- **Void sunStorm():** Sun storms do damage Travelers unless they hide in a hollow asteroid and also it harm the settler and that may result to make it die, unless the settler hide too
- **Void hide():** Travelers hide in a hollow asteroid, so they can survive the sun storm.
- **Void drill():** The Travelers can drill holes in the asteroids although they can not mine, but settlers can drill holes and mine as well. The Traveler can drill one unit per a function call.
- **Void travel(Asteroid a):** The Travelers travel between asteroids, to do a mission or to survive the radioactive explosions, settlers can travel to the neighboring asteroids. The traveler is removed from the previous asteroid list and added to the current asteroid

### **8.1.11 TeleportationGate**

- **Responsibility**

The settler can build a pair of gates, they can be used for entering to travel or can be also deployed

- **Superclasses**

*None*

- **Interfaces**

*None*

- **Attributes**

- **String ID:** each gate has a unique ID, and this represent the ID of the gate

- **Methods**

*None*

## 8.2 Detailed plan of testing

[The test-case defined in the chapter 7.3 should be described using the languages defined in the chapter 7.1. In this point the input data-streams - with which the functioning of program can be controlled - have to be given. For each data-stream: a definition (which parts and which functions of the program we expect to control by executing this data-stream) and a result (what kind of results we expect) is needed. Use the syntax for both input and output as defined in the previous document.]

### 8.2.1 Test-case 1 for Player Moves

- **Description**

*This Command is responsible for moving the settler in the field of the game, and for enabling him to hide.*

- **Unit of functionality to be tested, possible failures**

**The ability to move to neighboring fields and to hide the operator.**

Moving will fail in case we do not have neighboring fields and hiding will fail in case the asteroid is not completely drilled through or not hollow.

- **Input**

*MoveUp, MoveDown, MoveLeft, MoveRight, Hide*

- **Output**

*Traveler has been added*

*Traveler has been removed*

*Cannot Move*

*Successfully hidden*

*Could not hide*

### 8.2.2 Test-case 2 Teleport

- **Description**

*This command will be done by using the moving command to teleport through a teleportation gate.*

- **Unit of functionality to be tested, possible failures**

The ability to teleport through a teleportation gate.

Not being able to teleport if the gate couple is not activated.

- **Input**

*Keyboard key "T"*

- **Output**

*Gates Pair Activated*

*Gates Pair Not Activated*

### 8.2.3 Test-case 3 Player Drills

**Description**

*This command is responsible for drilling the rock mantle of an asteroid.*

**Unit of functionality to be tested, possible failures**

The ability to drill the rock mantle of an asteroid.

Not being able to drill in case the asteroid is already drilled through.

**Input**

Keyboard key “D”

**Output**

*Rock mantle already drilled through,*

*Rock mantle drilled (Depth decreased by one)*

*Explosion*

### 8.2.4 Test-case 4 Put Down Resource

**Description**

*This command will put a resource owned by a player on an asteroid.*

**Unit of functionality to be tested, possible failures**

The ability to put down a resource, in case the player does not own any the operation will fail.

**Input**

Keyboard key “P”

**Output**

*Resources have been added to the asteroid*

*No resources owned*

### 8.2.5 Test-case 5 Explosion at Perihelion

**Description**

*This test case will verify the behavior at the conditions that must make it explode.*

**Unit of functionality to be tested, possible failures**

A drilled through asteroid must explode if it carries a radioactive resource and if it is at perihelion.

**Input**

Keyboard key “D”

**Output**

*Explosion.*

*Asteroid drilled through.(In case of no explosion)*

### 8.2.6 Test-case 6 Settler mines

**Description**

*This command is responsible of the mining operation done by the settler*

**Unit of functionality to be tested, possible failures**

In case of a fully drilled asteroid, the asteroid can be full of resources or empty. Settler will not be able to min if he has full capacity of resources or the asteroid is empty.

**Input**

Keyboard key “M”

**Output**

*Resource added to the settler's capacity*

*Full capacity reached*

*Current asteroid is out of resource*

### 8.2.7 Test-case 7 Sunstorm

#### Description

*This test case verifies if the settler will survive in case of a sunstorm or not*

#### Unit of functionality to be tested, possible failures

In case of a sunstorm, if the settler is protected inside a hollow asteroid he will survive otherwise he will die.

#### Input

#### Output

*Settler is not protected -> Settler died*

*Settler is protected -> Settler survived*

### 8.2.8 Test-case 8 Robot Creation

#### Description

*This test case verifies if the settler can create a robot or not*

#### Unit of functionality to be tested, possible failures

If the settler has all the necessary resources, the robot will be created otherwise the operation will fail

#### Input

*Keyboard key "R"*

#### Output

*Robot Built Successfully*

*Not Enough Resources*

### 8.2.9 Test-case 9 Teleportation Gate Building

#### Description

*This test case verifies if the settler can build a teleportation gate or not*

#### Unit of functionality to be tested, possible failures

If the settler has all the necessary resources, the gate will be built otherwise the operation will fail

#### Input

*Keyboard key "G"*

#### Output

*Gate Built Successfully*

*Not Enough Resources*

### 8.2.10 Test-case 10 Teleportation Gate Deploy

#### Description

*This test case verifies if the settler can deploy a teleportation gate or not*

#### Unit of functionality to be tested, possible failures

*If the settler already has a gate it will be deployed. If he does not have it, the operation will fail.*

#### Input

*Keyboard key "H"*

#### Output

*Gate Deployed*

*No Gate Owned*

### **8.3 Plans of the supporting programs**

*For this Part we will be using white box testing method . We will examine the internal structure and the code.*

*We will make a functional testing program to test our functions and to see whether they are working correctly or not. ( More details about this part will be mentioned in the execution of the test cases in the second part) .*

*Currently we are not thinking of applying non-functional testing methods like compatibility testing, compliance testing or performance testing .....*

*The execution of the test cases :*

*<1> Start the game ( by running the test program ) Instantiating the game .*

*<2> The user now has to enter the name of the class which he wants to test . Our testing is based on the classe's names.*

*<3> After entering the name of the class the program will load all the needed functions for that class and for each function loaded , the program starts the testing for it based on a specific test file which contains the inputs and outputs . Each function has its own text file which contains the input and the corrected output of each test case .*

Using the inputs of the text file we will test our function and store all the outputs in another file .

*<4> Now the conclusion is to compare the first file which contains the corrected output to our new file which contains our function's output . The program will print :  
“There is a mistake” .*

Otherwise

“Successfully passed all the test cases” is printed to the output

## **8.4 Protocol**

<b>Start (date &amp; time)</b>	<b>Duration (hours)</b>	<b>Performer(s) name</b>	<b>Activity description</b>
10/04/2022	2 hours	Mohamed Amine Zaghdoudi	Worked on the detailed plan of testing
10/04/2022	6 hours	Kirolos Sawiris, Ahmed Mousa, Mohamed Abouzabady	Worked on Design level plan of classes
10/04/2022	2 hours	Mahdi Yahia	Worked on the detailed plan of testing
10/04/2022	1.5 hours	Oussama Dhouioui	Plans of the supporting program
10/04/2022	2 hours	MohammadParsa Kolivand	Worked on the detailed plan of testing and Documentation

# **[ASTEROID MINING]**

**[TEAM NNUMBER 7]**

Supervisor:  
**Dr. Balla Katalin**

## **Members:**

[Abouzabady Mohamed]	[GJVKZQ]	[mohamedabozabady1@gmail.com]
[Kirolos Sawiris]	[Y8B5KD]	[Kirolossawiris@gmail.com]
[Oussama dhouioui]	[XFCBM7]	[Oussemadhououi04@gmail.com]
[Mohamed Amine]	[ARPMPB]	[mzaghdoudi01@gmail.com]
[Ahmed Mousa]	[EWLQXU]	[amousa@edu.bme.hu]
[MohammadParsa Kolivand]	[RLU15I]	[parsakolivand@edu.bme.hu]
[Mahdi Yahia]	[WF72QQ]	[mahdiyahia18@gmail.com]

## 10. Prototype program

### 10.1 Deployment guide

#### 10.1.1 List of files

File name	Size	Date	Content
Asteroid	3.5 kB	24/04/2022	Class Asteroid -Attributes and Methods
Carbon	137 bytes	24/04/2022	Class Carbon -Attributes and Methods
Iron	134 bytes	24/04/2022	Class Iron -Attributes and Methods
Main	2.1 kB	24/04/2022	main class; contains test cases options
Resources	268 bytes	24/04/2022	Class Resources -Attributes and Methods
Robot	225 bytes	24/04/2022	Class Robot -Attributes and Methods
Settler	4.5 kB	24/04/2022	Class Settler -Attributes and Methods
SpaceStation	171 bytes	24/04/2022	Class SpaceStation -Attributes and Methods
TeleportationGate	504 bytes	24/04/2022	Class TeleportationGate -Attributes and Methods
Traveler	912 bytes	24/04/2022	Class Traveler -Attributes and Methods
Uranium	139 bytes	24/04/2022	Class Uranium -Attributes and Methods
WaterIce	128 bytes	24/04/2022	Class WaterIce -Attributes and Methods

#### 10.1.2 Compilation

For game to be successful the following operation is required

1)JDK (8 or later )

-There following IDEs have been used for developing

1 ) Eclips

2)IntelliJ IDEA

If the user wants to run a test case, they should enter the input character related to the test case in a single line, and then the test case will be automatically run.

### 10.1.3 Run

main creates several asteroid objects, and assigns for them depths, resource amounts and position; a settler object, and assigns for it an ID and an Asteroid; and a teleportation gate object to run the tests with. each test can be called with one letter line command until it receives an exit command.

Test "D" : tests the Drill() method of the traveler, it checks if the depth of the current asteroid is above zero, then it drills the asteroid by one unit and returns a positive check message, if not, returns a false message.

Test "M" : tests the Mine() method of the settler, it checks if the current asteroid is fully drilled, the settler has capacity and there are resources to mine, if so, it mines the asteroid by one unit successfully and returns a positive check message, if not, returns a false message.

Test "R" : tests the BuildRobot() method of the settler, it checks if the settler has the required resources to build the robot, if so, builds successfully and adds it to the current asteroid, if not, returns a false message.

Test "G" : tests the BuildGate() method of the settler, checks if the settler has the required resources to build the gate, if so, builds it successfully and assigns the current asteroid for it, if not, the boolean method returns false.

Test "H" : tests the DeployGate() method of the settler, passes the gate and destination asteroid for it, the method sets them and returns a positive check message.

Test "P" : tests the PutResource() method of the settler, transfers the resource objects of the settler to the current asteroid, and returns a positive check message for each resource.

Test "T" : test the Travel() method of the traveler, a teleportation gate is passed to it, if run successfully, a message for the removal from the first asteroid and a message for the adding to the second asteroid should return.

Test "B" : tests the MoveBack() method of the traveler, it will run successfully if there exists a left neighbor to the current asteroid that the traveler can travel to, and will return a positive check message containing the source and the destination, otherwise it will return a failure message.

Test "F" : same as the last test except with the MoveForward() method.

## 10.2 Test protocols

### 10.2.1 Drill:

Name of the tester	Kiroos Sawiris
Date & time of test	24/04/2022 12:09 Am

### 10.2.2 Move Forward:

Name of the tester	Mahdi Yahia
Date & time of test	24/04/2022 14:00

### 10.2.3 Move Backward:

Name of the tester	Mahdi Yahia
Date & time of test	24/04/2022 14:30

**10.2.4 Drill:**

<b>Name of the tester</b>	<b>Mahdi Yahia</b>
<b>Date &amp; time of test</b>	<b>42/04/2022 15:30</b>

**10.2.5 Mine:**

<b>Name of the tester</b>	<b>Mahdi Yahia</b>
<b>Date &amp; time of test</b>	<b>24/04/2022 22:00</b>

**10.2.6 Build Robot:**

<b>Name of the tester</b>	<b>Mahdi Yahia</b>
<b>Date &amp; time of test</b>	<b>24/04/2022 22:45</b>

**10.2.7 DeployGate:**

<b>Name of the tester</b>	<b>Kirolos Sawiris</b>
<b>Date &amp; time of test</b>	<b>24/04/2022 23:00</b>

**10.2.8 PutResource:**

<b>Name of the tester</b>	<b>Kirolos Sawiris</b>
<b>Date &amp; time of test</b>	<b>24/04/2022 23:10</b>

**10.2.9 Travel:**

<b>Name of the tester</b>	<b>Kirolos Sawiris</b>
<b>Date &amp; time of test</b>	<b>24/04/2022 23:15</b>

**10.2.10 ExitCommand:**

<b>Name of the tester</b>	<b>Kirolos Sawiris</b>
<b>Date &amp; time of test</b>	<b>24/04/2022 23:20</b>

**10.3 Evaluation**

Name of the team member	Participation (%)
Abouzabady Mohamed	14.28
Kirolos Sawiris	14.28
Mohamed Amine Zaghdoudi	14.28
Oussama Dhouioui	14.28
Mahdi Yahia	14.28
Ahmed Mousa	14.28
Mohammadparsa Kolivand	14.28

## **10.4 Protocol**

<b>Start (date &amp; time)</b>	<b>Duration (hours)</b>	<b>Performer(s) name</b>	<b>Activity description</b>
23/04/2022	2.5 hours	Mohammadpara Kolivand	Testing code and documentation
23/04/2022	2.5 hours	Ahmed Mousa	Testing and documentation
23/04/2022	2.5 hours	Kirolos Sawiris	implementation and testing
23/04/2022	2.5 hours	Abouzabady Mohamed	Testing and documentation
22/04/2022	3 hours	Mohamed Amine Zaghdoudi	Worked on the implementation of the code
22/04/2022	3 hours	Oussama Dhououi	Worked on the implementation of the code
24/04/2022	3.5 hours	Mahdi Yahia	Testing the code and help make corrections.

# **[ASTEROID MINING]**

**[TEAM NNUMBER 7]**

Supervisor:  
**Dr. Balla Katalin**

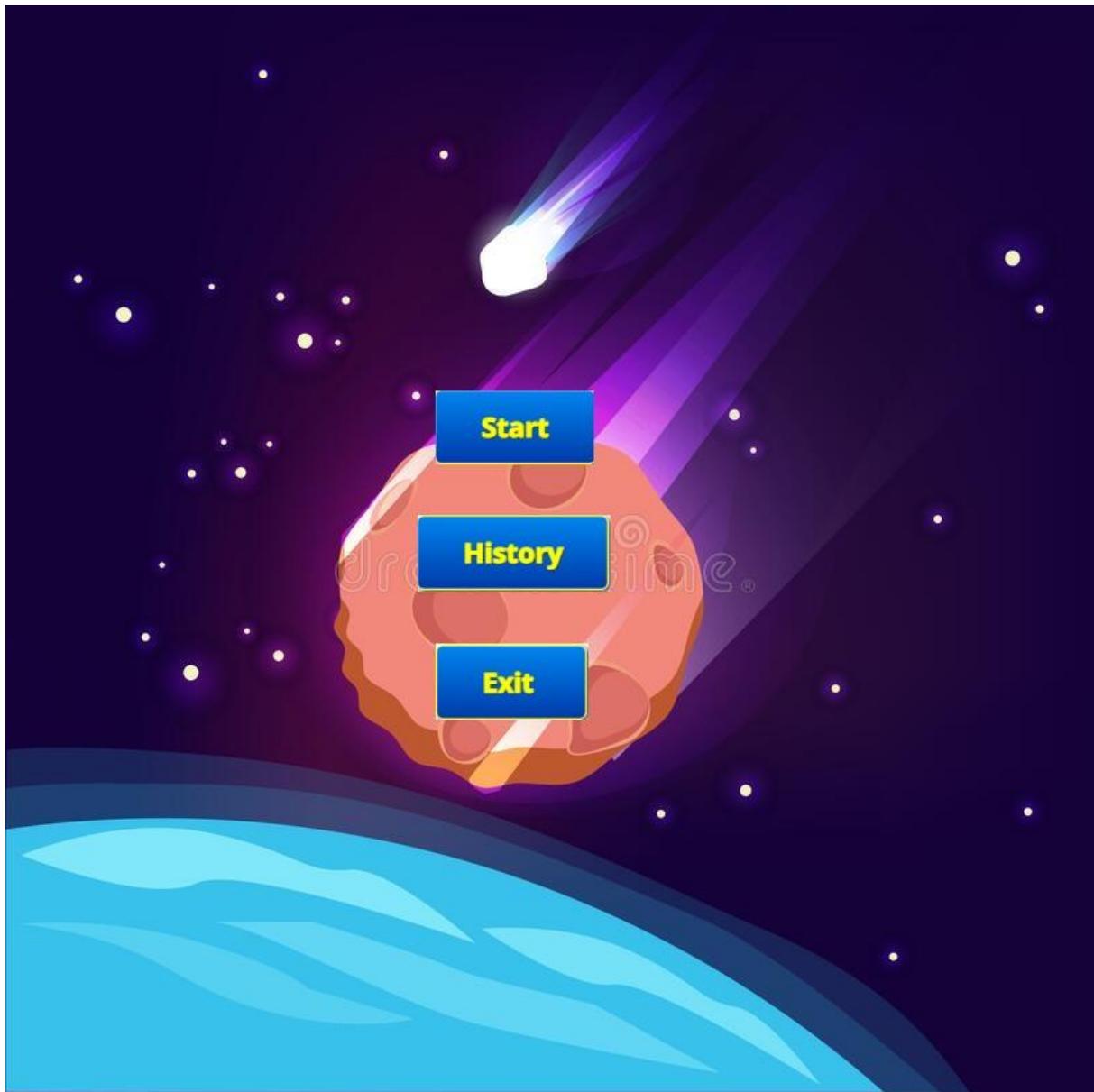
## **Members:**

[Abouzabady Mohamed]	[GJVKZQ]	[mohamedabozabady1@gmail.com]
[Kirolos Sawiris]	[Y8B5KD]	[Kirolossawiris@gmail.com]
[Oussama dhouioui]	[XFCBM7]	[Oussemadhououi04@gmail.com]
[Mohamed Amine]	[ARPMPB]	[mzaghdoudi01@gmail.com]
[Ahmed Mousa]	[EWLQXU]	[amousa@edu.bme.hu]
[MohammadParsa Kolivand]	[RLU15I]	[parsakolivand@edu.bme.hu]
[Mahdi Yahia]	[WF72QQ]	[mahdiyahia18@gmail.com]

## 11. User interface specification

### 11.1 Graphical User Interface

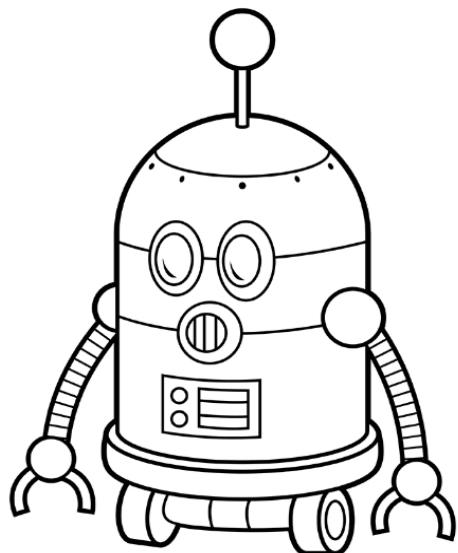
Main Menu:



**Settler :**

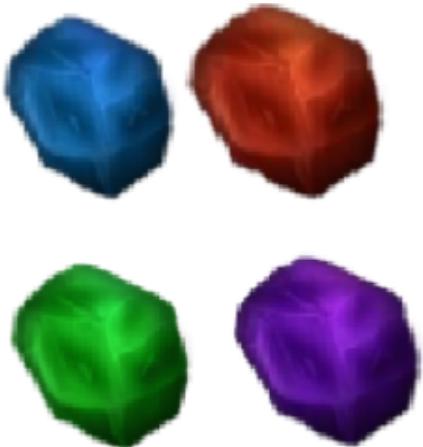


**Robot :**

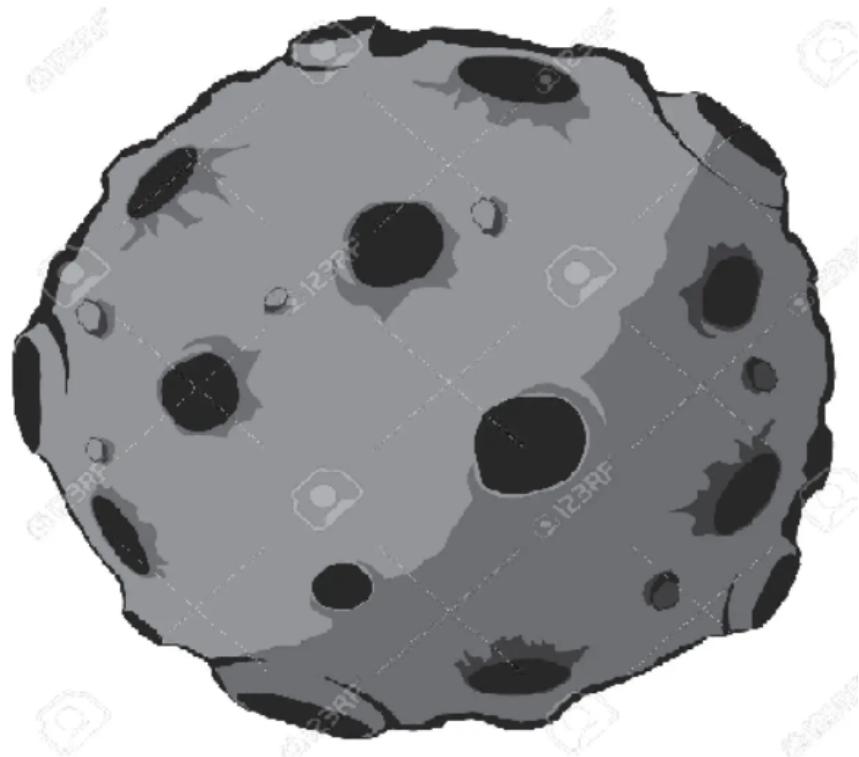


documentation.pdf - |

**Resources :**



**Asteroid :**



**Game won :**



**Game over:**



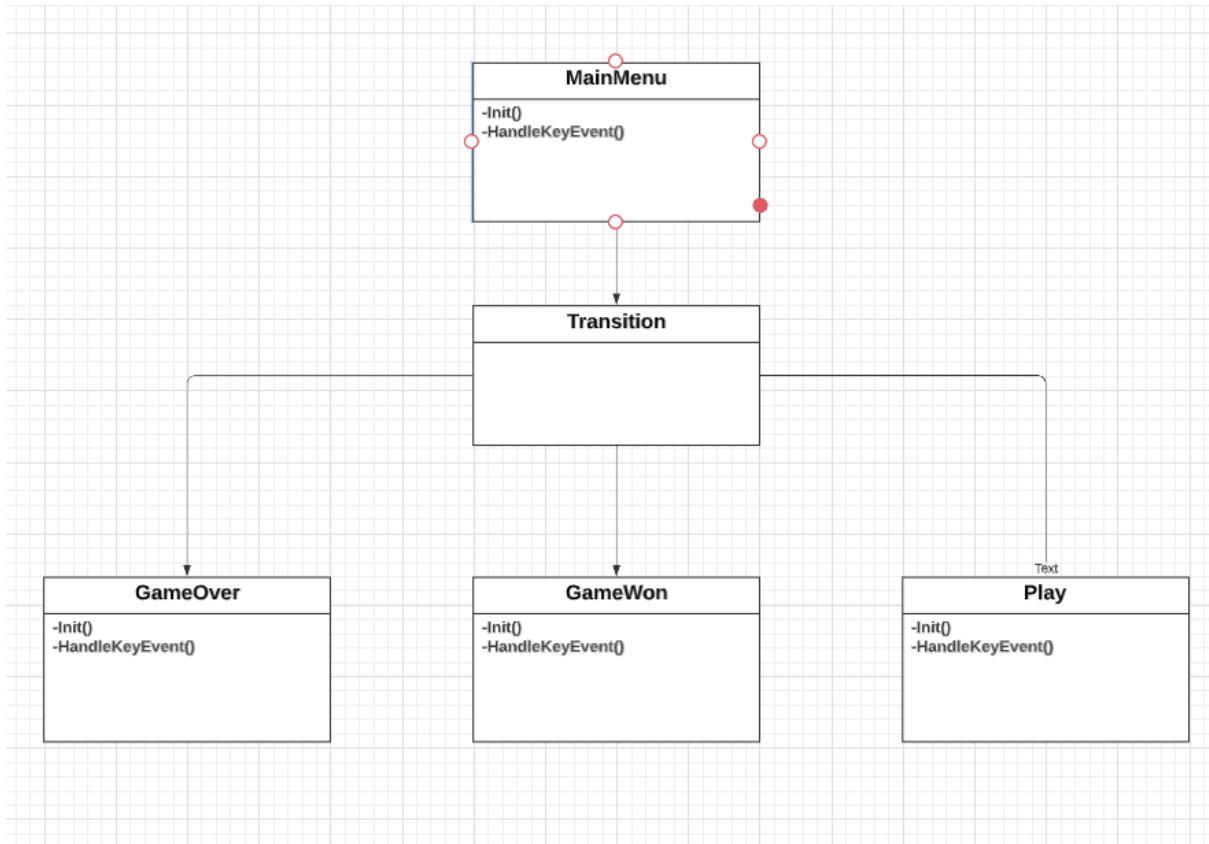
**Teleportation Gate:**



## 11.2 Architecture of the graphical system

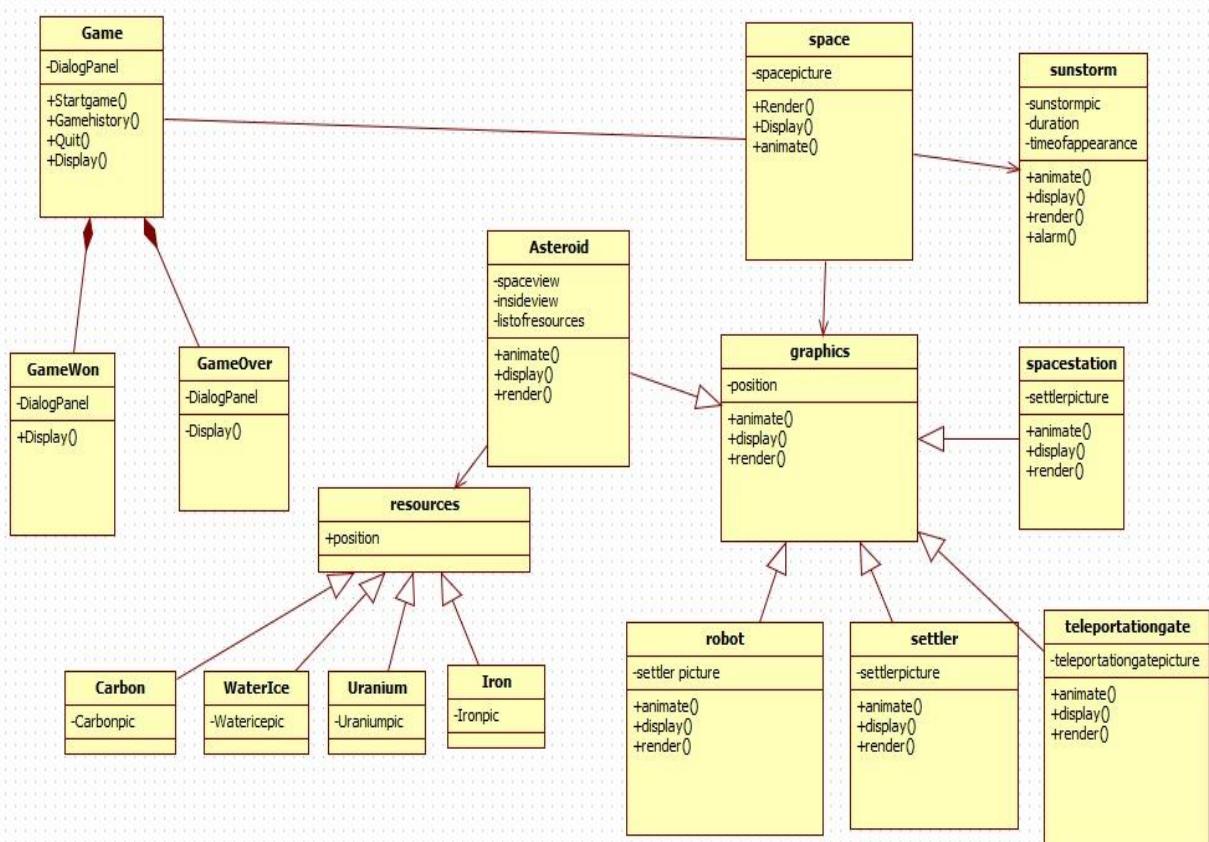
### 11.2.1 Principles of the GUI

Transition will make the communication between the states easier as it enables us to navigate between the states: GameOver, GameWon, Play. Each stat will be containing the functions of the GUI program.



HandleKeyEvent() will give the user the ability to interact by input.

## 11.2.2 GUI Structure diagram



## 11.3 GUI Classes

### 11.3.1 Game

#### Responsibility

*This class is responsible for setting the game menu for the player when he starts the game.*

#### Superclasses

#### Interfaces

*IEventListener*

#### Attributes

- **attribute1:**

private DialogPanel: the panel that will show up at the beginning of the game.

#### Methods

- **Public void Startgame(): The method that will launch a new game.**
- **Public void Gamehistory(): The method that will display the last results.**
- **Public void Quit(): The method that will leave the current game.**
- **Public void Display(): The method that is responsible for drawing the picture on the screen.**

### 11.3.2 GameWon

#### Responsibility

*This panel will be displayed when the player wins the game.*

#### Superclasses

#### Interfaces

*IEventListener*

#### Attributes

- **attribute1:**

private DialogPanel: the panel that will show up when the player wins the game.

#### Methods

- **Public void Display(): The method that is responsible for drawing the related picture on the screen.**

### 11.1.3 GameOver

#### • Responsibility

*This panel will be displayed when the player loses the game.*

#### • Superclasses

#### • Interfaces

*IEventLister*

#### • Attributes

- **attribute1:**

`private DialogPanel`: the panel that will be displayed when the player loses the game.

#### • Methods

- **Public void Display(): The method that is responsible for drawing the picture on the screen.**

### 11.1.4 Space

#### • Responsibility

*The whole game will be built on this class, since all the elements that constitute the game are placed on it.*

#### • Superclasses

#### • Interfaces

*IEventLister*

#### • Attributes

- **attribute1:**

`private SpacePicture(BufferedImage)`

#### • Methods

- **Public void Render(): Passing the frame to the buffer when the picture is drawn**
- **Public void Display(): Displaying the space where the game will be played.**
- **Public void Animate(): switches between different frames.**

### 11.1.5 Sunstorm

#### Responsibility

*This class is responsible for managing the sunstorms by starting an alarm sound and displaying a red zone showing the affected area.*

#### Superclasses

#### Interfaces

*IEventListener*

#### Attributes

- **attribute1:**

**private Sunstormpic: picture to represent the sunstorm**

- **attribute2:**

**private duration: the time duration of the sunstorm**

- **attribute3:**

**private timeofappearance: the time of start of the sunstorm**

#### Methods

- **Public void render(): Passing the frame to the buffer when the picture is drawn**
- **Public void display(): Displaying the picture of the sunstorn.**
- **Public void animate(): switches between different frames.**
- **Public void alarm(): responsible for making the sound of the alarm.**

### 11.1.6 Graphics

#### Responsibility

*Is an abstract class that is used to handle the graphic elements of the game.*

#### Superclasses

- **Interfaces**

- IEventListener**

- **Attributes**

- **attribute1:**

**private position:** *represents the position of the graphic element in the space.*

- **Abstract Methods to be implemented in the child classes :**

- **Public void render():**
- **Public void display():**
- **Public void animate():**

### 11.1.7 Spacestation

- **Responsibility**

*A class for handling the graphic element of the space station.*

- **Superclasses**

*Graphics*

- **Interfaces**

*IEventListener*

- **Methods**

- **Public void render():** Passing the frame to the buffer when the picture is drawn
- **Public void display():** Displaying the graphic element.
- **Public void animate():** switches between different frames.

### 11.1.8 Teleportationgate

- **Responsibility**

*[What is the responsibility of the class ? Circa 1 paragraph long.]*

*A class for handling the graphic element of the teleportationgate.*

- **Superclasses**

*Graphics*

- **Interfaces**

*IEventListerner***Methods**

- **Public void render():** Passing the frame to the buffer when the picture is drawn
- **Public void display():** Displaying the graphic element.
- **Public void animate():** switches between different frames.

**11.1.9 Settler****Responsibility**

*A class for handling the graphic element of the settler.*

**Superclasses**

*Graphics*

**Interfaces***IEventListerner***Methods**

- **Public void render():** Passing the frame to the buffer when the picture is drawn
- **Public void display():** Displaying the graphic element.
- **Public void animate():** switches between different frames.

**11.1.10 Robot****Responsibility**

*A class for handling the graphic element of the robot.*

**Superclasses**

*Graphics*

**Interfaces***IEventListerner***Methods**

- **Public void render():** Passing the frame to the buffer when the picture is drawn
- **Public void display():** Displaying the graphic element.
- **Public void animate():** switches between different frames.

### 11.1.11 Asteroid

#### Responsibility

*A class for handling the graphic element of the asteroid.*

#### Superclasses

*Graphics*

#### Interfaces

*IEventListener*

#### Methods

- **Public void render():** Passing the frame to the buffer when the picture is drawn
- **Public void display():** Displaying the graphic element.
- **Public void animate():** switches between different frames.

### 11.1.12 Resources

#### Responsibility

*An abstract class to handle a graphic element representing the resource inside the asteroid*

#### Superclasses

*Graphics®Asteroid*

#### Interfaces

*IEventListener*

#### Attributes

- **attribute1:**

**private position:** *position of the resource inside the asteroid*

### 11.1.13 Carbon

#### Responsibility

*Graphic element that shows that the resource is carbon*

**Superclasses**

*Graphics ® Asteroid ® Resources...*

**Interfaces**

*IEventListerner*

**Attributes**

- **attribute1:** *[feature: why is it good ? type, visibility, initial value]*

**carbonpic:** *picture that represents the carbon element*

### 11.1.14 Uranium

**Responsibility**

*Graphic element that shows that the resource is uranium.*

**Superclasses**

*Graphics ® Asteroid ® Resources...*

**Interfaces**

*IEventListerner*

**Attributes**

- **attribute1:**

**carbonpic:** *picture that represents the uranium element*

### 11.1.15 WaterIce

**Responsibility**

*Graphic element that shows that the resource is waterice.*

**Superclasses**

*Graphics ® Asteroid ® Resources...*

### Interfaces

*IEventListener*

#### Attributes

- **attribute1:**

**carbonpic:** *picture that represents the waterice element.*

## 11.1.16 Iron

### Responsibility

*Graphic element that shows that the resource is iron.*

### Superclasses

*Graphics* ® *Asteroid* ® *Resources...*

### Interfaces

*IEventListener*

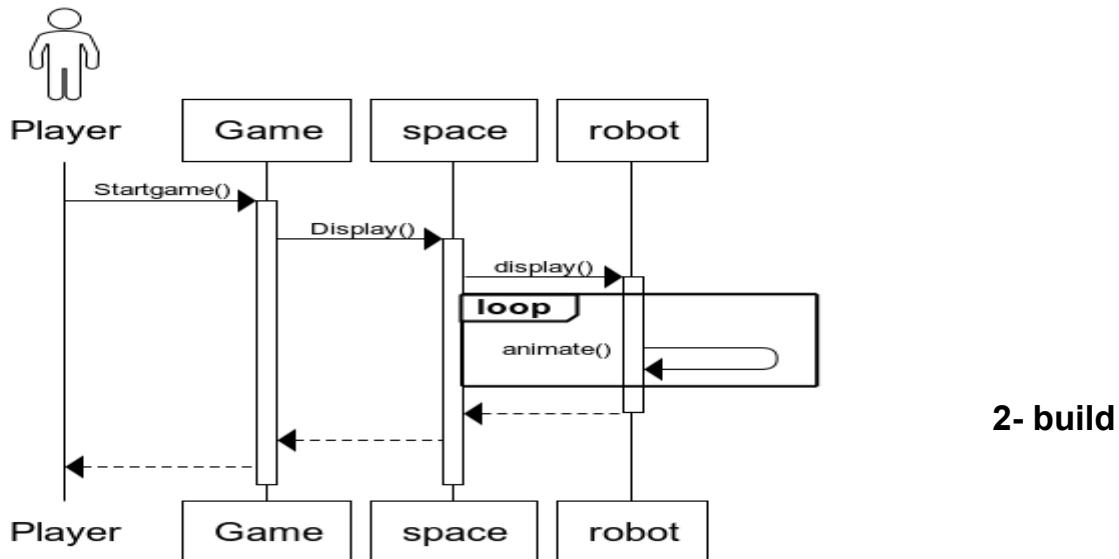
#### Attributes

- **attribute1:**

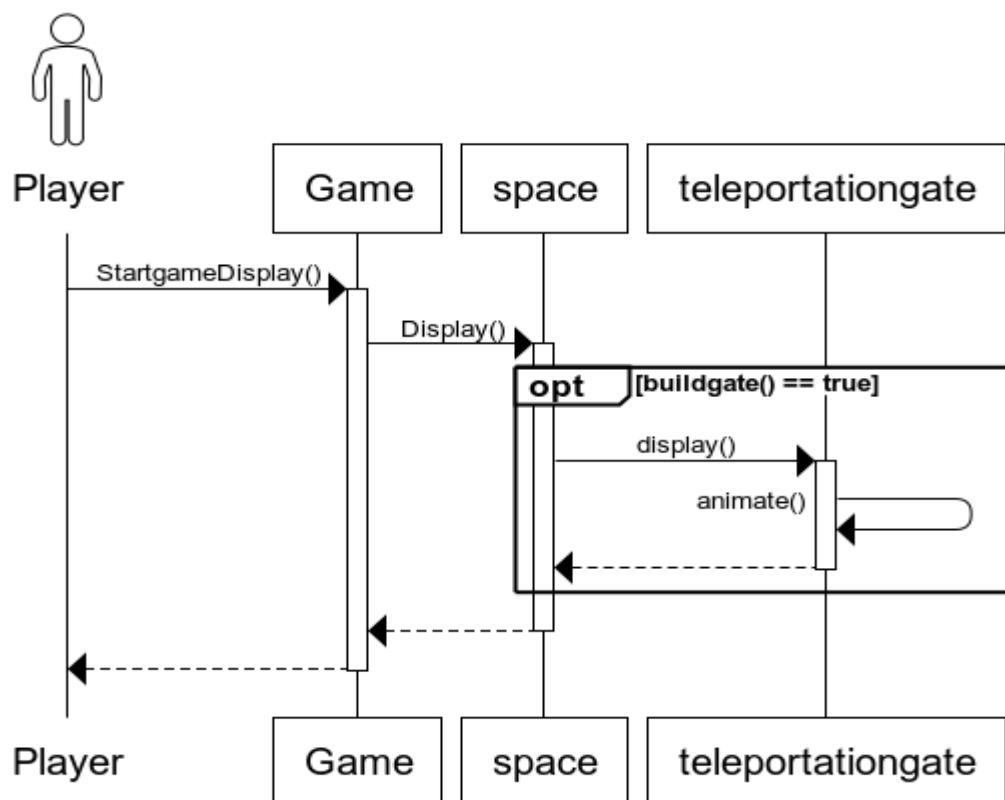
**carbonpic:** *picture that represents the iron element.*

## 11.4 Dynamic connection between the model and the GUI

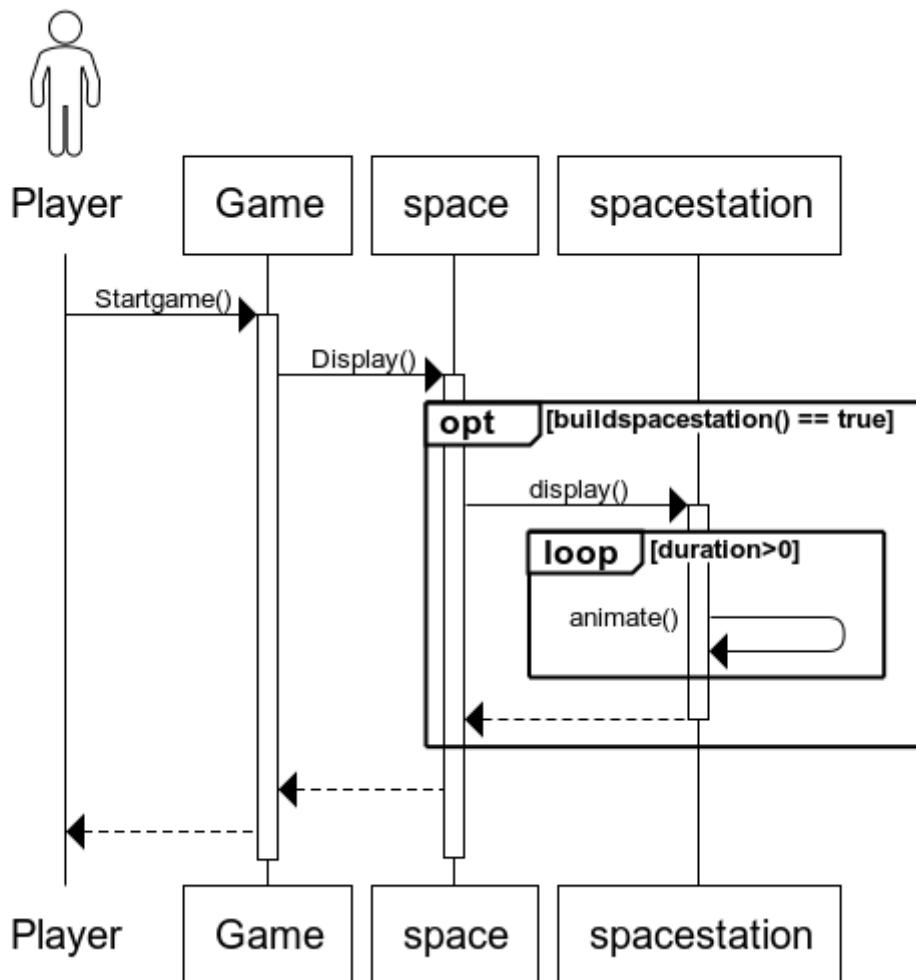
### 1-build robot



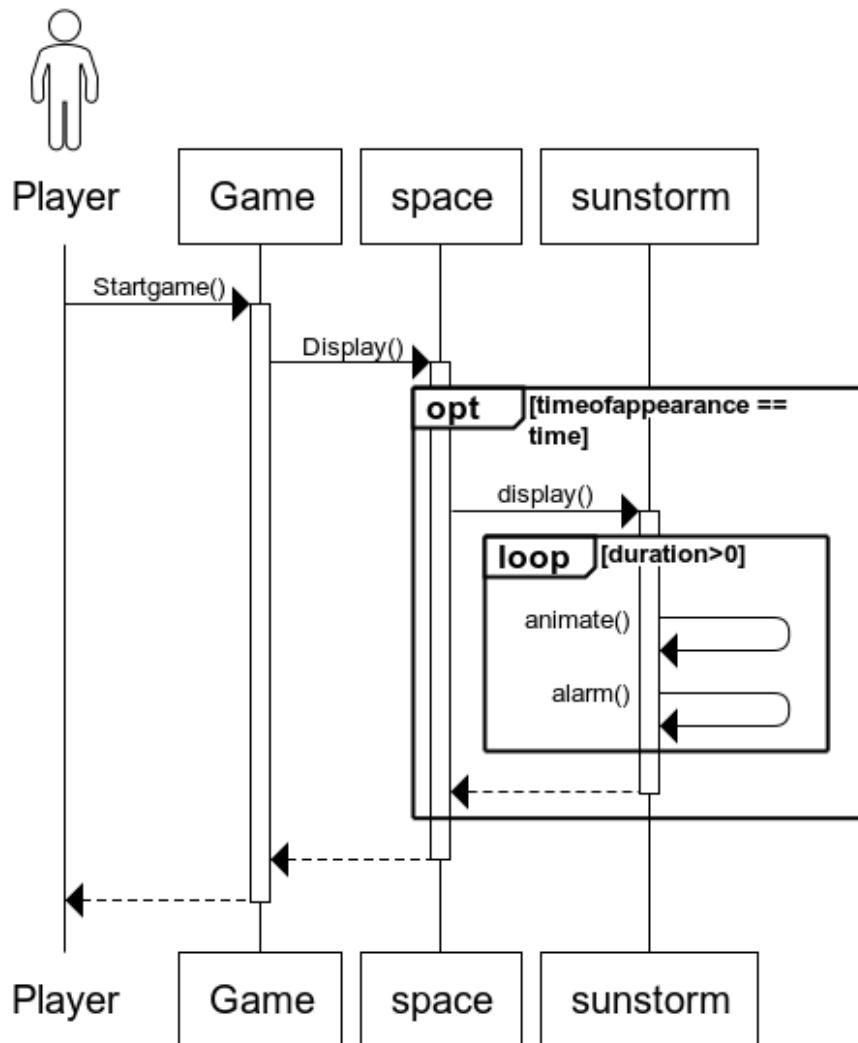
### teleportation gate



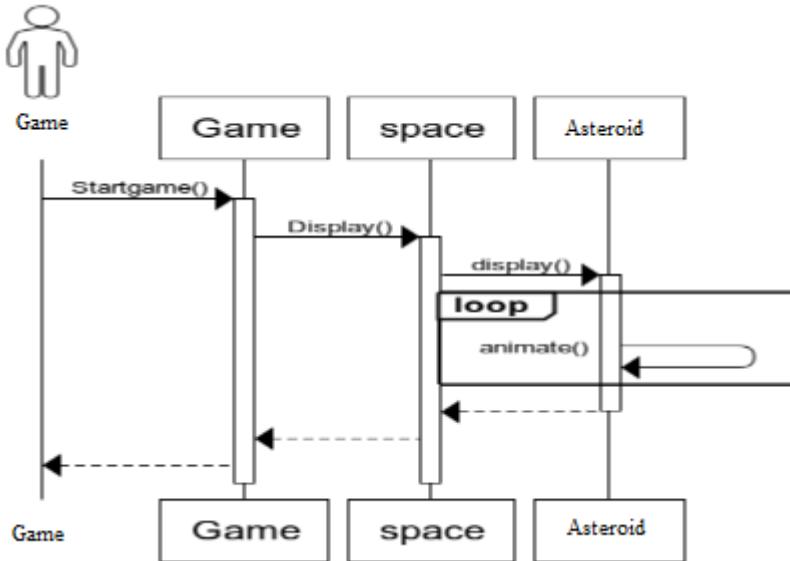
### 3- build space station



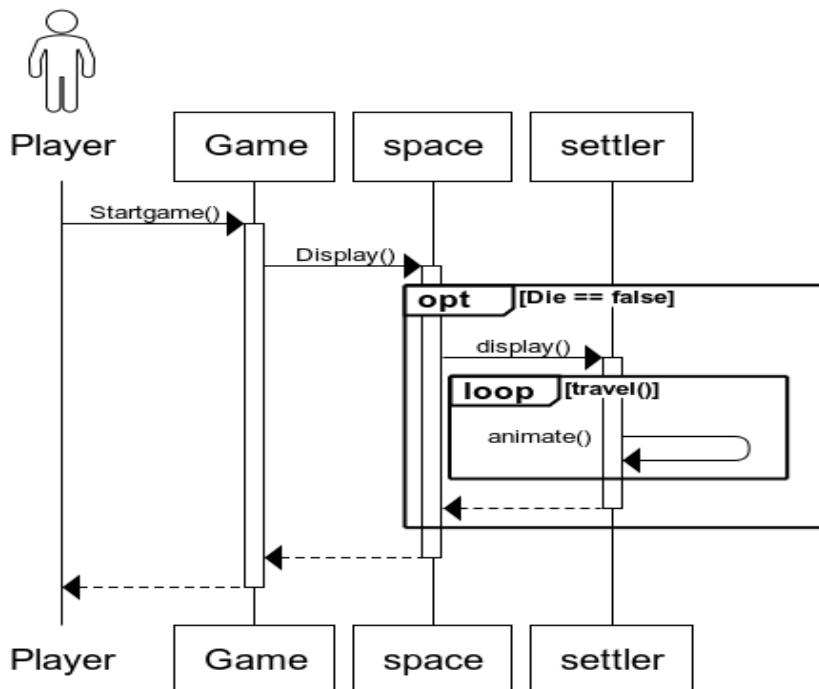
#### 4- Appearance of sunstorm

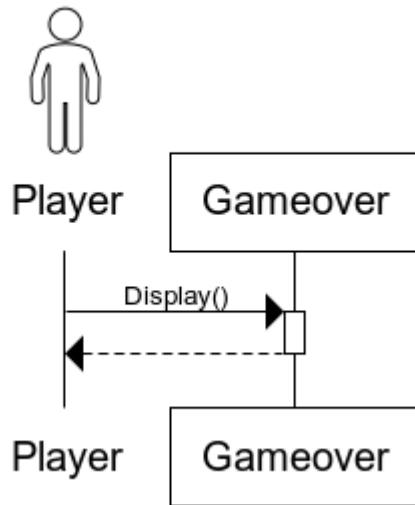
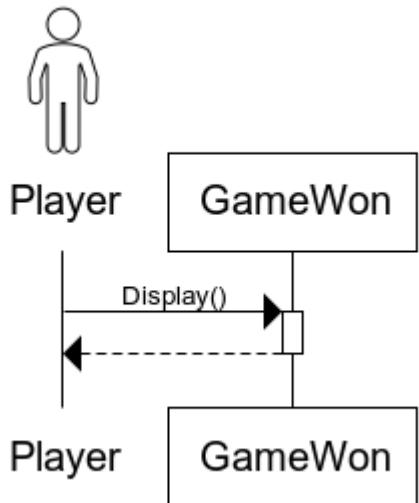


## 5- asteroid GUI



## 6- settler GUI



**7- Game over****8- Game Won**

## **Protocol**

<b>Start (date &amp; time)</b>	<b>Duration (hours)</b>	<b>Performer(s) name</b>	<b>Activity description</b>
01.05.2022	2 hours	Mohammadpar sa Kolivand	<i>Documentation and Classes descriptions</i>
01.05.2022	2,5 hours	Mahdi Yahia	GUI Classes Description
01.05.2022	2 hours	Mohamed Amine Zaghoudi	Worked on 11.1 and 11.2
01.05.2022	2 hours	Kirolos Sawiris	GUI sequences
01.05.2022	2 hours	Dhouioui Oussama	GUI Class diagram and some descriptions.
01.05.2022	2 hours	Ahmed Mousa	GUI sequences
01.05.2022	2 hours	Mohamed Abouzabady	GUI sequences

# **[ASTEROID MINING]**

**[TEAM NNUMBER 7]**

Supervisor:  
**Dr. Balla Katalin**

## **Members:**

[Abouzabady Mohamed]	[GJVKZQ]	[mohamedabozabady1@gmail.com]
[Kirolos Sawiris]	[Y8B5KD]	[Kirolossawiris@gmail.com]
[Oussama dhouioui]	[XFCBM7]	[Oussemadhououi04@gmail.com]
[Mohamed Amine]	[ARPMPB]	[mzaghdoudi01@gmail.com]
[Ahmed Mousa]	[EWLQXU]	[amousa@edu.bme.hu]
[MohammadParsa Kolivand]	[RLU15I]	[parsakolivand@edu.bme.hu]
[Mahdi Yahia]	[WF72QQ]	[mahdiyahia18@gmail.com]

## 13. Complete program

### 13.1 Deployment guide

#### 13.1.1 List of files

File name	Size	Date	Content
GUI	4 KB	15/5/2022	Renders all the elements according to the game state
game	1 KB	15/5/2022	Contains the main function
Panel	4 KB	15/5/2022	Presents the game on the screen
Field	1 KB	15/5/2022	Making the field class
Map	5 KB	15/5/2022	Making the map
Animation	2 KB	14/5/2022	Handle the movement in the map
Asteroid	3 KB	15/5/2022	Contains asteroid methods
Carbon	1 KB	15/5/2022	Contains the carbon resource methods
Hollow	1 KB	15/5/2022	Contains the Hollow resource methods
Iron	1 KB	15/5/2022	Contains the Iron resource methods
Object	5 KB	15/5/2022	Base class for all the objects in the project
Resource	1 KB	15/5/2022	Base class for all the Resources classes in the project
Robot	1 KB	15/5/2022	Contains Robot methods
Settler	7 KB	15/5/2022	Contains the settler methods
SpaceStation	1 KB	15/5/2022	Contains the Space station methods
TeleportationGate	2 KB	15/5/2022	Contains the gate methods
Traveler	1 KB	15/5/2022	Contains the traveler methods
Uranium	1 KB	15/5/2022	Contains the Uranium Resource methods
WaterIce	1 KB	15/5/2022	Contains the WaterIce methods
GameOver	1 KB	15/5/2022	Elements of the losing case and ending the game
GameWon	1 KB	15/5/2022	Elements of the winning case and ending the game
Loader	2 KB	15/5/2022	Loading the game
MainMenu	3 KB	15/5/2022	Displaying the main menu
Manager	2 KB	15/5/2022	Desplaying the settings
Play	6 KB	15/5/2022	Playing the game
State	1 KB	15/5/2022	The state of the game

### 13.1.2 Compilation

The Input using the keyboard buttons.

The output will be displayed on the GUI interface.

### 13.1.3 Run

We will start the game by pressing selecting start and pressing enter button then we will enter the game with the base and the resources we have then we will move using arrows to each asteroid and press (I) to show the info then do what we want to do. we can press (q) to drill the asteroid, (m) to mine the asteroid, and (h) to hide from the sunstorm.

## 13.2 Evaluation

Name of the team member	Participation (%)
Abouzabady Mohamed	14.28
Kirolos Sawiris	14.28
Oussama dhouioui	14.28
Mohamed Amine	14.28
Ahmed Mousa	14.28
MohammadParsa Kolivand	14.28
Mahdi Yahia	14.28

### 13.3 Protocol

Start (date & time)	Duration (hours)	Performer(s) name	Activity description
10/5/2022	8 hours	Abouzabady Mohamed	Coding and testing the game
10/5/2022	8 hours	Kirolos Sawiris	Coding and testing the game
10/5/2022	8 hours	Oussama dhououi	Coding and testing the game
10/5/2022	8 hours	Mohamed Amine	Coding and testing the game
10/5/2022	8 hours	Ahmed Mousa	Coding and testing the game
10/5/2022	8 hours	MohammadPars a Kolivand	Coding and testing the game
10/5/2022	8 hours	Mahdi Yahia	Coding and testing the game