

## 2. Requirements, Project, and Functionality

### 5 – SoftProLab Team

Supervisor:  
Dr. Katalin Balla

#### Members:

Hagverdiyev Subhan	NHL9KN	<a href="mailto:subhan.hakverdiyev@gmail.com">subhan.hakverdiyev@gmail.com</a>
Madatov Ali	XVEARA	<a href="mailto:ali.madatov@hotmail.com">ali.madatov@hotmail.com</a>
Shoaib Areeba Tabassum	EZXFWK	<a href="mailto:areebashoaib99@gmail.com">areebashoaib99@gmail.com</a>
Salahov Kamal	IG5LSM	<a href="mailto:salahovkamal@hotmail.com">salahovkamal@hotmail.com</a>
Singh Gurdeep	ERXIPV	<a href="mailto:sandhugoldy433@gmail.com">sandhugoldy433@gmail.com</a>

22 Feb. 20201

## 2. Requirements, project, functionality

### 2.1 Introduction

#### 2.1.1 Goal

This document summarizes general information about the implementation of the “Asteroid Mining” game.

#### 2.1.2 Application domain

The purpose of the software to be developed is to implement the “Asteroid Mining” game dreamed up by the instructors of the subject of the Software Project Laboratory, focusing on the greatest possible satisfaction and experience of the instructors and game developers. A secondary goal is to create another copy of thinking games to meet the needs of all our fellow human beings who are hungry for such things.

#### 2.1.3 References

Software technology lesson note and lecture slides  
Prog. 3 notes and lecture slides

#### 2.1.4 Document content

Will be added later

### 2.2 Overview

#### 2.2.1 General overview

The program to be created is a game in which the player has to build an space station in the asteroid belt. The goal of the development team is to make a program that fullfills all the specification and requirements and which can be run on any machine.

The final version of the program has Graphical User Interface and can easily be controlled by keyboard and mouse. The goal is to make the game enjoyable and fun for user. The flexibility is also provided by the development environment as the game can be executed in any machine which has appropriate Java Runtime Environment. The game can only be played in a single player mode. The user is directly connected to the Graphical User Interace through which he/she can access the game menu. Throught the game menu, the user can chose the level of the game and start playing it. If the user want to exit the game then he/she can click on exit button through game menu. The game does not need any network connection on database etc.

#### Development Tools Used:

- Eclipse
- Microsoft Word, Google Docs.
- Dropbox
- Junit (Testing)

### 2.2.2 Functions

First, menu-like interface must be implemented, so User might choose when to start the game. In the game, graphics must be implemented for various objects listed further. Game map is asteroid belt located on space. Players must be able to view the whole map when they want. Game time consists of rounds. There must be settlers, that consist of a single person spaceship. Players must be able to move these settlers through space and perform specific actions during rounds, when conditions are fulfilled. Players must be able to select settlers and operations using mouse or keyboard. Round ends when players choose what to do and finish. After each end of round, players must be notified about the results of the actions chosen in the round. There must be asteroids of two types: At the core of a single asteroid must be single random resource - normal asteroid, or an asteroid can be with empty core - hollow asteroid. Mantle depth of this asteroids must be random for each of them, also each asteroid must have neighbors ranging from 1 to multiple hundred. Resource pool must at least contain: water ice, iron, carbon, uranium. Resources like uranium are radioactive. In single move, settlers can perform only one operation. Settler can travel, destination can be chosen among neighboring asteroids, settler can also drill mantle by one unit. Players must be able to see how many units of depth left in mantle of asteroid. If, mantle has been drilled through, then player must be able see what is inside the core. Settler can mine the resource in the core and extract it. Carried resources are shown, when viewing settler. Settler can carry maximum 10 units of resources, but also, settler can store single unit of resource in the core of hollow asteroid. Storing counts as operation. Also, asteroids can be at aphelion state or perihelion state. If settler tries to mine the core of an asteroid containing radioactive resource, asteroid explodes and kills settler. That's why radioactive resource can only be mined when asteroid is aphelion stance. Player must be able to see which state is asteroid currently in. After some time, game starts, sun storms must reach asteroid belt, periodically. Game must notify the player some time period before this happens, so player may react, because settler dies, if caught in sun storm. To avoid sun storms, settler may choose to hide in the core of hollow asteroid if mantle is drilled through. Hiding counts as operation too. Settlers can build autonomous robots by using single units of iron, carbon, and uranium. Robots must be controlled by computer itself, by simple artificial intelligence. Goal of these robots is to help settlers. Robots must be able to perform some operations too. Robots just like settlers, can move from asteroid to neighboring one, drill mantle by one unit and hide in the core of hollow asteroid to survive from sun storm. Robots cannot mine or carry resources. They are also not affected by radioactive explosion. When explosion happens, they land in the neighboring asteroid. Settlers can also build a pair of transportation gates, by using single units of iron, water ice and uranium. Settler can only carry a pair of transportation gates and can only place it in the vicinity of the asteroid the settler is on. Both settlers and robots can use these transportation gates. Entering from one gate must result in the leaving from pair of that gate. Game must end if all settlers die, resulting in loss for players. If, settlers bring and collect 3 units of each resource on single asteroid, game ends by settlers building space station and players winning. There must be end game screen for each case.

### 2.2.3 Users

There are 2 users in the game:

- Settlers – controlled by players. They can perform actions only chosen by players, so their characteristics are based on player.
- Autonomous robots – controlled by computer, their goal is to assist settlers. So, they operate in the most beneficial way they can find by evading dangers and completing work.

## 2.3 Requirements

Before we dive into the various types of requirement, we must assign priority level to them. Which one is the must do and which is not important? We can categorize these priority levels into the following parts:

\* **High Level Priority:** This is a must have requisite and it is in top of hierarchy.

\* **Indispensible Level Priority:** User should pay attention to it. This falls in second level of hierarchy.

\* **Low Level Priority:** This is at the bottom of hierarchy and user can pay attention to this level of functionality if they're done with all the other functions.

*We can divide requirements for the product into following three parts:*

#### ● **Basic Level Requirement:**

These requirements are highest priority level in the hierarchy. Basic level of requirement is the minimum functionality one's product must have in order to satisfy the customer and to run the product. Unable to do so will result in unfinished product, which will amount to nothing.

#### ● **Predominant Level Requirement:**

While we absolutely need the basic level function to run the program, this is the level where we provide functionality which will either break or make the program. Without having the functions of this level, system may misbehave and not run at all. The Predominant level requirement is corresponding to "should have" specification that was given and it falls under indispensable priority level.

#### ● **Discretionary Level Requirement:**

At this level, one can choose to add some additional feature in the product and it all upto user's description. This is placed at low level priority in the list.

### 2.3.1 Functional requirements

ID	Description	Check	Priority	Source	Use-case	Comment
SPL1	Game Menu	Prototyping	Indispensible	Predominant		
SPL2	Start	Demonstration	Indispensible	Predominant		
SPL3	Exit	Demonstration	Indispensible	Predominant		
SPL4	Game Over	Prototyping	High Level	Predominant		
SPL5	Asteroids	Testing	High Level	Basic	Inside Space	
SPL6	Settler	Testing	High Level	Basic	Inside Space, Mine, Drill	
SPL7	Robot	Testing	High Level	Basic	Travel, Mine	
SPL8	Settings	Prototyping	Low Level	Discretionary		
SPL9	Resolution Change	Prototyping	Low level	Discretionary		

### 2.3.2 Resource requirements

ID	Description	Check	Priority	Source	Comment
SPLL1	Intel Core 2 , AMD or Other CPU Equivalent	Demonstration	High	Indispensible	Essential requirements for product to work properly
SPLL2	CPU 1 GHz	Demonstration	High	Indispensible	Essential requirements for product to work properly
SPLL3	Peripherals Devices	Demonstration	High	Indispensible	Input output devices like keyboard, monitor etc
SPLL4	1 GB RAM	Demonstration	High	Indispensible	Essential requirements for product to work properly
SPLL5	Java	Demonstration	High	Indispensible	Essential

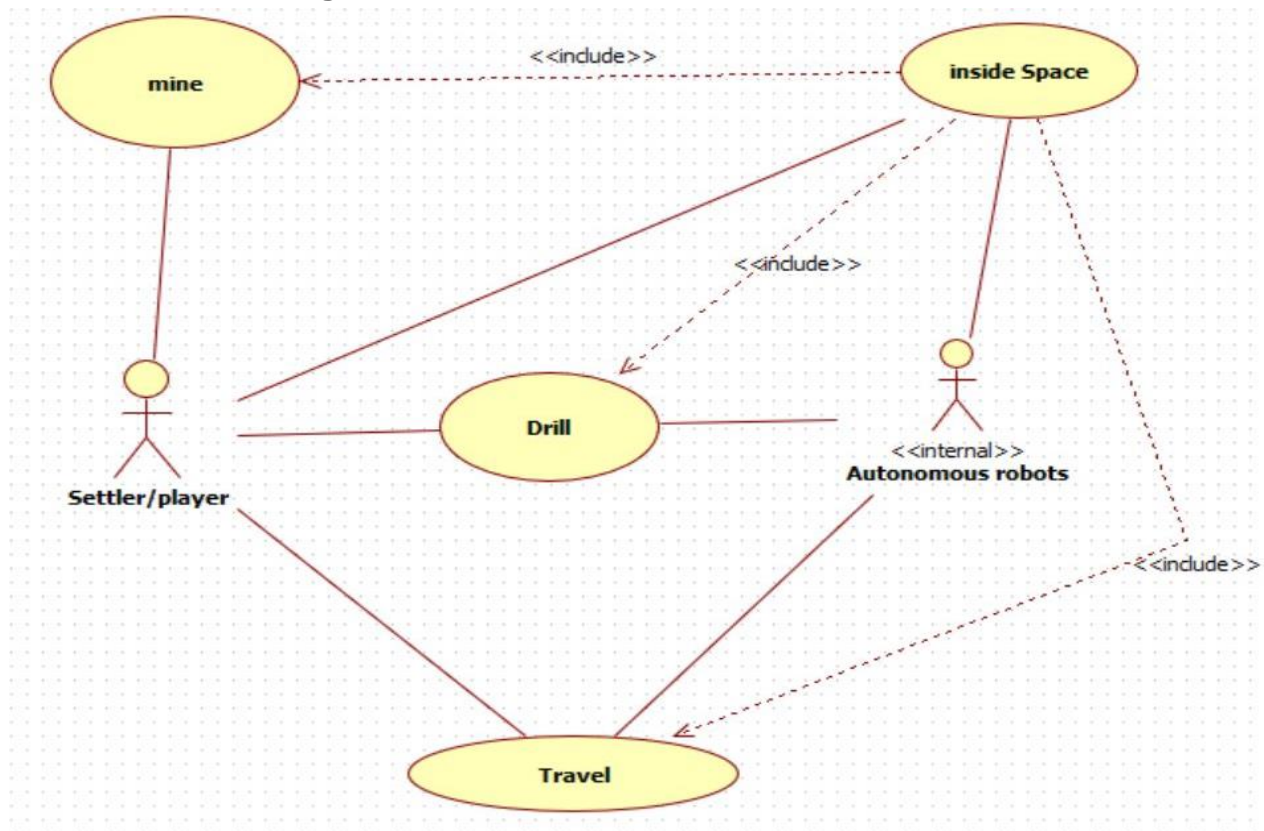
	<b>Runtime Environment (JRE 1.6 or higher)</b>				<b>requirements for product to work properly</b>
<b>SPPL6</b>	<b>800x600 resolution</b>	<b>Demonstration</b>	<b>High</b>	<b>Indispensible</b>	<b>Minimum Screen Resolution</b>

### 2.3.3 Non- functional requirements, Restrictions

<b>ID</b>	<b>Description</b>	<b>Check</b>	<b>Priority</b>	<b>Source</b>	<b>Comment</b>
<b>SPLLL1</b>	Everything which is a basic requirement must be met	Presentation evaluation	High	Basic	
<b>SPLLL2</b>	To run the product requires Java Runtime Existence of Environment 8.	presentation	High	Basic	
<b>SPLLL3</b>	Can be installed according to instructions, without outside help	presentation	High	Basic	

## 2.4 Essential use-cases

### 2.4.1 Use-case diagram



### 2.4.2 Use-case descriptions

<b>Use-case name</b>	<b>Mine</b>
<b>Description</b>	The player can mine inside the space in order to search for the resources to build a space station.
<b>Actors</b>	Settler/player
<b>Dialog, scenario</b>	<ul style="list-style-type: none"> <li>a) The player can mine without Successfully any problem. When the settler mine, he/she cannot travel or drill.</li> <li>b) Settlers can find resources in the asteroid while mining.</li> <li>c) The time for mining depends on the depth/size of the asteroid</li> <li>d) Mining is unsuccessful if the core of the asteroid is empty or it hasn't been drilled properly.</li> <li>e) If the settler mines and find proper resources, it gets point.</li> <li>f) The settler dies if he hits a radioactive asteroid at perihelion.</li> <li>g) The settler dies if sun storm comes while mining.</li> <li>h) Radioactive asteroids can be mined only at aphelion.</li> <li>i) If all the settler loses lives, the game ends.</li> <li>j) the player wins if they mine at least one unit and collect the resources and build the space station successfully.</li> </ul>

<b>Use-case name</b>	<b>Drill</b>
<b>Description</b>	The settler can drill and deepen the hole of the mantle with one unit.
<b>Actors</b>	Settler/player
<b>Dialog, scenario</b>	<ul style="list-style-type: none"> <li>a) The settler can drill without any problem.</li> <li>b) When the settler drills he/she cannot travel or mine</li> <li>c) Resource can be extracted from the asteroid if the mantle has been completely drilled through.</li> <li>d) The settler dies if he drills radioactive asteroid at perihilion</li> <li>e) The settler survives ih he drills a radioactive asteroid at apelihiion</li> <li>f) The settlers/robots survives the sunstorm if they hide in the core of the hollow astroid which is possible only if mantel is completely drilled through</li> <li>g) If the settlers loses the lives, the game is over.</li> <li>h) If the settlers extracts resources, it gets points.</li> </ul>



<b>Use-case name</b>	<b>Travel</b>
<b>Description</b>	The Settlers travel and they can go to the neighbouring asteroids.
<b>Actors</b>	Settlers/player, Autonomous Robots
<b>Dialog, scenario</b>	<ul style="list-style-type: none"> <li>a) Settlers can easily go to the neighboring asteroids</li> <li>b) When the settler is travelling, he/she cannot drill or mine.</li> <li>c) There can be hundreds of neighboring asteroids inside space.</li> <li>d) The settler can travel to neighboring asteroid for the purpose of drilling and mining.</li> <li>e) Robots can travel between the asteroids.</li> </ul>

<b>Use-case name</b>	<b>Inside Space</b>
<b>Description</b>	Everything is present inside space. Inside space, the settler can travel, mine and drill. The robot can also drill and travel. Each event is taking place inside space.
<b>Actors</b>	Settlers/player, Autonomous Robots
<b>Dialog, scenario</b>	<ul style="list-style-type: none"> <li>a) Settler can easily travel, drill and mine inside space. Robot can also easily travel and drill inside space</li> <li>b) Settler can easily extract the resources from the mantle inside space.</li> <li>c) Settler can die due to radioactive asteroid and sandstorm inside space</li> <li>d) Settler can successfully build the space ship inside the space.</li> <li>e) Every type of asteroid is present inside the space.</li> <li>f) Robots can survive radioactive explosion in the asteroid present inside space.</li> <li>g) In short, everything is happening inside space.</li> </ul>

## **2.5 Glossary**

**asteroid belt-** where all space station is located

**asteroid-** minor planets in asteroid belt where settler can settle

**radioactive asteroid-** asteroids which contains uranium as resource and can only be mined when the asteroid is at aphelion

**Perihelion** -is the point of the Earth's orbit that is nearest to the Sun

**Aphelion** - is the point of the Earth's orbit that is farthest away from the Sun

**settler-** spaceship that contains single person

**game map-** shows the asteroid belt on space

**hallow asteroid-** asteroid with empty core

**resources-** can be mined from asteroid belt (water ice, iron, carbon, uranium)

**drop box-** cloud service to save files

**uranium-** is a chemical element

**Robots-** the settlers can build autonomous robots controlled by artificial intelligence, robots can only travel between asteroids and drill holes. 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. Sun storms do damage robots unless they hide in a hollow asteroid.

**mantle-** The Settler may also drill a single unit of the mantle. Players need to be able to see how many units of depth exist in the asteroid mantle. If the mantle has been drilled, then it is important for the player to see what is inside the core.

**drill-** settler drills the asteroid until the mantle.

**travel-** going from one asteroid to another

**game end-** if all the players die

**game win-** if all the space station is successfully constructed

## 2.6 Project plan

DEADLINE	TASK	RESPONSIBLE
1 MARCH	Analysis of model -1st version documentation	Ali
8 MARCH	Analysis Model - final version -- documentation	Subhan
16 MARCH	Planning the Skeleton -- documentation	Kamal
22 MARCH	Skeleton program -- documentation and source code	Singh
29 MARCH	Concept of Prototype -- documentation	Subhan
12 APRIL	Detailed plans -- documentation	Shoaib
19 APRIL	Prototype program -- documentation and source code	Kamal
26 APRIL	User interface specification -- documentation	Singh
10 MAY	Complete program -- documentation and source code	Shoaib
18 MAY	Summary	Ali

## Resources

Tools used for documentation: Microsoft Word, Google Docs

Communication: Microsoft Teams, Whatsapp

Modelling tool: WhiteStarUML

Development tool: Eclipse/IntelliJ

Sharing documents: Google drive

Source code sharing, version control: Git

**2.7 Protocol**

<b>Start (date &amp; time)</b>	<b>Duration (hours)</b>	<b>Performer(s) name</b>	<b>Activity description</b>
20.02.2021	40 minutes	Gurdeep Singh, Salahov Kamal, Subhan Hagverdiyev, Areeba T. Shoaib, Ali Madatov	Assigning Tasks
20.02.2021	2 hour	Gurdeep Singh	2.3, 2.3.1,2.3.2
20.02.2021	2 hours 25 minutes	Areeba T. Shoaib	2.4, 2.4.1, 2.4.2
20.02.2021	30 minutes	Areeba T. Shoaib	2.2.1, Editing
21.02.2021	2 hours	Salahov Kamal	2.2.2, 2.2.3
22.02.2021	2 hours	Subhan Hagverdiyev	2.5, 2.6
22.02.2021	2 hours	Ali Madatov	2.1, 2.3.3