

Budapest University of Technology and Economics



Asteroid Mining

Subject: Software Project Laboratory

Course code: BMEVIIIAB06

Team name: Pied Pipers

Supervisor:

Dr. Balla Katalin

Members:

Abdelrahman Desoki	DOHDZF	adesoki@edu.bme.hu
Neda Radonjic	NQC17Y	neda.radonjic@edu.bme.hu
Tushig Bat-Erdene	QBI3JH	bat-erdene.tushig@edu.bme.hu
Chaitanya Arora	BWTFX8	chaitanya.arora@edu.bme.hu
Janibyek Bolatkhan	BOI6FK	janibyekbolatkhan@edu.bme.hu
Kasay Ito	LX5RFB	kasay.ito@edu.bme.hu

February 27, 2022

2. Requirements, project, functionality

2.1 Introduction

2.1.1 Goal

The goal of the document is to define a clear structure of the requirements, functionality and project plan. This will help us approach the tasks in a structured and planned manner, as we progress through the development.

2.1.2 Application domain

The aim of the game is to mine resources from asteroids, in order to build a space station. The settlers, controlled by players of the game, roam around in individual spaceships, looking for resources. In order to extract resources, settlers have to perform various operations such as travelling and drilling. The game also involves risks like solar flare which can affect the settlers' health, leading to losing the game. The game is won if the settler extracts at least 3 necessary units of each resource used to build the space station.

2.1.3 References

For the execution of this documentation we made used:

1. StarUML (Use case)
2. Google Docs (Drafting and Compilation)

Problem Definition: [Here](#)

2.1.4 Document content

Overview: describes the problem statements and the use cases of the game players

Requirements: describes functional requirements, resource requirements, and non-functional requirements of the project in detail.

Essential use cases: describes the form of dialog between the actor and system, what events the actor does and system does in response.

Glossary: describes the terminology in this documentation and the actual game

Project plan: describes the schedule of deliveries, as well as the organisational structure of the team

Protocol: takes note of the conducted meetings, and their purpose and outcome

2.2 Overview

2.2.1 General overview

The game is based on settlers, robots, and asteroids. Players control settlers, which can build robots, collect resources, and are aiming to build a space station. Resource extraction depends on drilling, which both robots and settlers can do, as well as building teleportation gates for easier travel between asteroids. Extreme weather conditions can kill settlers and damage robots. If they avoid these dangers successfully, and collect enough resources on an asteroid, the player wins the game.

2.2.2 Functions

Asteroid Mining

Humanity has decided to exploit the resources contained in asteroids. In order to do this, the settlers have to build a space station in the asteroid belt. The resources to build a space station would be too expensive to transport there, so the settlers have to mine them from the asteroids.

Players of the game control the settlers. Settlers wander in the asteroid belt with single person spaceships looking for resources.

Asteroids are covered in rocks. The depth of this rock mantle can vary from asteroid to asteroid. The important resources (water ice, iron, carbon, uranium, etc.) can be found in the core of the asteroids. Some resources (like uranium) are highly radioactive. There can be hollow asteroids, whose core is empty. The core of an asteroid is always homogeneous; it is always made of a single kind of a material.

A settler can do one operation in a single move. There are several operations, like travelling, drilling, mining, building robots, building teleportation-gates, etc. When a settler travels, they go to a neighbouring asteroid (an asteroid may have multiple hundreds of neighbours). When a settler drills, they deepen the hole in the mantle with one unit. When a settler mines, they extract the resource from the core of the asteroid. This is only possible if the mantle has been completely drilled through. A single settler can only carry 10 units of resources, that's the spaceships' capacity. A hollow asteroid, however, can be filled with a unit of resource, counting as an operation.

There are some dangers lurking in the asteroid belt. When a fully drilled asteroid with a radioactive core is at perihelion, the asteroid explodes and kills any settler on it. Radioactive material, therefore, can only be mined when the asteroid is at aphelion. Sometimes a sun storm reaches the asteroid belt, and it is also dangerous to the settlers. 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.

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

Using up a unit of iron, a unit of carbon and a unit of uranium, the settlers can build autonomous robots controlled by artificial intelligence. These 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 neighbouring asteroid. Sun storms do damage robots unless they hide in a hollow asteroid.

Using up two units of iron, a single unit of water ice and a single unit of uranium, the settlers can build a pair of 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 traveller (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.

The game can end in two ways. If all the settlers die, the players lose. If, however, 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.

2.2.3 Users

Users of the game are mainly gamers and normal people who are interested in asteroid mining and problem solving.

2.3 Requirements

2.3.1 Functional requirements

ID	Description	Check	Priority	Source	Use-case	Comment
R01	Settler must build a space station in the asteroid belt.	Demonstration	Must have	Problem description	Build	
R02	Players control the settlers	Demonstration	Must have	Problem Description	Move-Build-Mine-Drills-MAntle	

R03	The depth of the rock mantle can vary from asteroid to asteroid	Testing	Must Have	Problem Description	Drill mantle	
R04	A settler can do one operation in a single move	Demonstration	Must have	Problem Description	Move-Build-Mine-Drills Mantle	
R05	When a settler travels, they go to a neighbouring asteroid	Demonstration	Should have	Problem Description	Move	
R06	When a settler drills, they deepen the hole	Demonstration	Should have	Problem Description	Drills Mantle	
R07	When a settler mines, they extract the resource from the core of the asteroid.	Demonstration	Should have	Problem Description	Mine	
R08	A settler can build a robot.	Demonstration	Should have	Problem Description	Build	

R09	A settler can build a teleportation gate.	Demonstration	Should have	Problem Description	Build	
R10	If a radioactive core is at perihelion, the asteroid explodes and kills any settler on it.	Testing	Must have	Problem Description	Determine Danger	
R11	Radioactive material,, can only be mined when the asteroid is at aphelion	Testing	Must have	Problem Description	Determine Danger	
R12	If sun storm reaches the asteroid belt, it might is dangerous to the settlers	Demonstration	Should have	Problem Description	Determine Danger	
R13	A settler can only survive a sun storm if they hide in the core of a hollow asteroid.	Demonstration	Must have	Problem Description	Settler	
R14	When a fully drilled asteroid with water ice in	Demonstration	Must have	Problem Description	Drills Mantle	

	its core is at perihelion, the water ice sublimates (disappears).					
R15	These robots can only travel between asteroids and drill holes	Testing	Could have	Problem Description	Drills Mantle-Robot	
R16	Robots cannot mine because they are unable to transport things	Demonstration	Must have	Problem Description	Robot	
R17	Robots can survive radioactive explosions.	Demonstration	Could have	Problem description	Robot	
R18	The settlers can build a pair of teleportation-gates.	Demonstration	Could have	Problem description	Build	
R19	Gates can be deployed in the vicinity of an asteroid.	Demonstration	Should have	Problem description	Deploy	
R20	A single settler can	Demonstration	Should have	Problem description	Settler	

	only bring two gates.					
R21	If all the settlers die, the players lose.	Testing	Must have	Problem description	Operating Game	
R22	If they collect those materials on a single asteroid, they can build the space station (Win game)	Demonstration	Must have	Problem description	Operating Game	

2.3.2 Resource requirements

ID	Description	Check	Priority	Source	Comment
R01	Installed version of Eclipse (4.10 or higher)	Eclipse IDE must be installed in the machine in order to compile the source code.	Should have	Google:	
R02	Java Development Kit	JRE/JDK 8+ must be installed on the device..	Must have	Problem statements	

R03	Operating System: Windows 7, 8, 10, 64-bit versions only; Mac OS X 10.12.4+; Linux;	In order to compile and run the program. User must have one of the operating system for action to be done.	Must have one of them	What are the minimum requirements of a PC to make a 2D Android game using Unity? (2019). Quora. https://www.quora.com/What-are-the-minimum-requirements-of-a-PC-to-make-a-2D-Android-game-using-Unity	
R04	Standard Mouse, Keyboard	System testing	Must have	The explanation during the second consultation	
R05	Graphical calculation power/ Graphics Card/GPU	System testing	Should have	M. (2019). PC requirements to run a simple 2D game built with UE 4.23. Unreal Engine Forums. https://forums.unrealengine.com/t/pc-requirements-to-run-a-simple-2d-game-built-with-ue-4-23/133142	
R06	Memory requirement	2GB for convenient game play	Should have	Progress KB - How to increase the amount. (2020).	

				<p>Progress Software Knowledgebase. https://knowledgebase.progress.com/articles/Article/P122464</p> <p>Eclipse bug - Maximum heap is limited to 256MB if started without Xmx-Parameter and Java >= 11.0.(2019). https://bugs.eclipse.org/bugs/show_bug.cgi?id=553538</p>	
R07	CPU processor requirements	Any processor that can run Eclipse IDE	Should have	https://wiki.eclipse.org/Eclipse/Installation#Install_a_JVM	

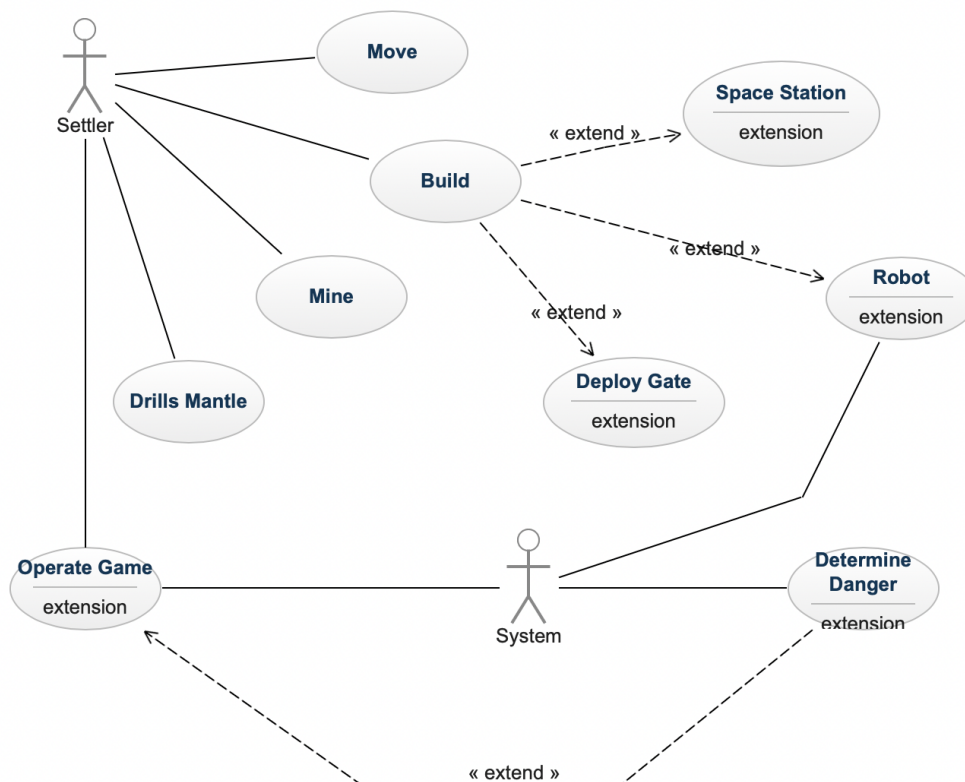
2.3.3 Non- functional requirements, Restrictions

ID	Description	Check	Priority	Source	Comment
R01	Enough storage to download and run the software	Users must check the free storage before downloading/installing the game software.	Must have	Brainstorming	Testability
R02	Understanding Rules of the game	Make sure that the user has enough knowledge to know the rules of the game and play it. (+6 years old)	Must have	Brainstorming	Learnability
R03	Fault Tolerance of game	To make sure all edge cases involving different permutation and combination are included	Must have	Lecture note of Software Engineering subject	Availability
R04	Reliability of the game	Users shall play the game with high performance without	Must have	Lecture note of Software Engineering subject	Reliability

		errors or fall down.			
R05	Coexistence Interoperability	Other developers can continue the developmental process if the environment requirements are fulfilled.	Should have	Lecture note of Software Engineering subject	Compatibility

2.4 Essential use-cases

2.4.1 Use-case diagram



2.4.2 Use-case descriptions

Use-case Name	Operate Game
Short textual Description	The Operating Game involves certain operations like Starting game, ending game etc.The System controls this Use case and then the control goes to the settler
Actors	System, Settler
Main Case Scenario	The Project is run and the System calls the Operating game and the game is started
Alternate Scenario	The game is running and the user conditions to fail become correct and the operate game is called which ends the game
Use-case Name	Moves
Short textual Description	Move function is used to change the position of the actors which can be settler or robot
Actors	Settler
Main Case Scenario	The player moves with the spaceship in directions like left, right, up or down
Use-case Name	Mine

Short textual Description	Mining is a by the settler to mine the metals that they get in the asteroid
Actors	Settler
Main Case Scenario	The settler mines the asteroid metals and extracts the necessary materials
Use-case Name	Build
Short textual Description	Build use case give settlers ability to build Robots, Gate, and
Actors	Settlers
Main Case Scenario	Players can build Robots, Gate, and Space Station by using necessary units of resources.
Alternate Scenario	
Use-case Name	Drills Mantle
Short textual Description	Drills mantle is one of the function executed by the settler to drill the asteroid in the core in order to reach the metals
Actors	Settlers
Main Case Scenario	Settler drills the asteroids

Use-case Name	Space Station
Short textual Description	Building the Space Station will be the aim of the settler. Since it's one of the things that the settler builds, extend is used
Actors	Settler
Main Case Scenario	The settler extracts enough resources and builds the Space Station
Use-case Name	Deploy Gate
Short textual Description	Settler deploys teleportation-gates in the vicinity of the asteroid where settlers ison.
Actors	Settlers
Main Case Scenario	Teleportation-gates give settlers and robots the ability to transport between asteroids.
Alternate Scenario	Settlers can only bring two-gates.
Use-case Name	Robot
Short textual Description	The settler builds the Robot which itself will perform different functions like drilling and will be controlled by the system and not the settler

Actors	System
Main Case Scenario	The Settler is controlled by System to perform different tasks
Alternate Scenario	The Robot is built by the settler
Use-case Name	Determine Danger
Short textual Description	The major task of determining danger is to make sure of the health rate of the Settler and robot from various mishappenings like Blast, solar storm etc.
Actors	System
Main Case Scenario	The system checks the health of the robot and settler

2.5 Glossary

Term	Description	Type
Asteroid	Asteroids are objects that contain resources inside them.	Object
Space station	The objective of this game, namely, the construction of a space station is the goal of the user (the player).	Object
Settler	The main player of the game and miner with single spaceship	Object
Robot	Autonomous robots can be	Object

	built by settlers and controlled artificially. They can travel between asteroids and drill holes. They are unable to mine and transport things. They must land on a neighbouring asteroid to survive radioactive explosions. But, sun storms can damage them unless they hide in a hollow asteroid.	
Mantle	Mantle covers the corresponding asteroids	Object
Drill	Operation to deepen the mantle of the asteroid.	Action
Mine	Operation to collect the resources when drilling is done.	Action
Build	Operation of settlers to launch space stations, robots, and the teleportation gates.	Action
Perihelion	The event that handles some of the dangers in the game. If perihelion is occurred when there are revealed radioactive resources (e.g. uranium).	State
Teleportation-gates	Gate built by robots to transport between asteroids.	Object
Sun Storm	Dangerous scenes can cause damage.	Event
Resources	Materials (Uranium, Water ice, and carbon) are essential to build a station.	Object
Radioactive Explosion	An event that can harm the settlers and robots.	Event

2.6 Project plan

The purpose of this project is to put our coursework knowledge into hands-on practice and improve our team-work skills and experience with working software engineering methodology. In terms of the objective of this project, we carefully follow the problem description and its requirements to build a software which provides the customer's exact need.

We will continuously be working on the project, with expected deliveries almost every week, according to a schedule that can be seen below. Apart from the documentation which will continuously be created and delivered, the code will be delivered in 3 main stages- Skeleton, Prototype, and Complete program.

Mar 07.	Analysis Model - first version -- documentation
Mar 16.	Analysis Model - final version -- documentation
Mar 21.	Planning the Skeleton -- documentation
Mar 28.	Skeleton program -- documentation and source code.
Apr 04.	Concept of Prototype -- documentation
Apr 11.	Detailed plans -- documentation
Apr 25.	Prototype program -- documentation and source code
May 02.	User interface specification -- documentation
May 09.	Creating the Complete part
May 16.	Complete program -- documentation and source code.
May 20.	Summary

2.6.1 Organisational Structure

Name	Role	Demonstration
Abdelrahman Desoki	Testing, Back-end Developer	Responsible for the quality of the code and testing its main functionality. Checking the logic behind the code and developing the database and operating system side Contributing in the

		integration of the backend and frontend.
Neda Radonjic	Project Manager, Testing Engineer	Planning and overseeing the project to ensure the schedule is followed and deadlines are met. Designing and running tests and analysing their results to ensure the quality and functionality of the code.
Tushig Bat-Erdene	Front-End Developer, Software Architect	Responsible for developing and designing the front-end of the software in terms of graphical user interface development. Providing architectural blueprint of the development cycle for the team.
Chaitanya Arora	Project Lead, Backend Developer	Create and communicate a clear list of expectations and goals for team members to follow. Handling the backend implementation of the project and dealing with roadblocks in the implementation.
Kasay Ito	Software Architect, Supporter	Mainly designs how the system should be implemented and maintains the design with appropriate level. Due to the experience of development of the game previously, he can support other team members when they encounter issues including technical problems, understanding the requirements, managing the project, and the like.
Janibyek Bolatkhan	Quality Assurance, Front-End Developer	Responsible for developing and designing the front-end of the software in terms of GUI development. Testing and Improving the quality of the code by integrating backend and frontend interfaces.

2.7 Protocol

Start (date & time)	Duration (hours)	Performer(s) name	Activity description
15.02.2022	40 min	All team members	Our first meeting had the purpose of getting to know the people that we will be working with, and making initial, informal plans for the project and getting to know each other's strengths and weaknesses in different parts like frontend, backend etc. No specific decisions were made, since the topic was not announced at the time.
20.02.2022	1 hour	All team members	The second meeting involved us to understand and comprehend the game and its main functionalities to make sure we all were on the same page. It involved us to structure the problem and decide the problem statement and decide our future steps.
2/21 Monday use case diagram, requirements refinement	3 hours	All team members	<p>This third meeting was an in person meeting conducted to brainstorm and work together. During this time we worked together and discussed the major parts of the submission like the requirements and use case diagram etc.</p> <p>After having to brainstorm the ideas and everyone's views we</p>

			collaborated together to draft the Introduction, overview, Functional Requirements and use case descriptions
2/22 Tuesday consultation	1.5 hours	All team Members	Discussed the previously created use-case diagram with the supervisor and got clarification of some doubts.
16:00-17:30 Thursday	1.5 hours	All team Members	Wrapping up the work on our first piece of documentation. The chapters that we covered were resource requirements, non-functional requirements, project plan, protocol, and glossary.
10:00-11:30 Sunday	1.5 hours	All team members	Deciding on some last-minute changes regarding the organisational structure of the team and modifying the documentation accordingly.