Basic Information

Project name	Expansion of a space survival game “Lost in Space”
ID	*LostInSpace*
Supervisor	*RNDr. Tomáš Holan, Ph.D. <Tomas.Holan@mff.cuni.cz>*
Consultants	*Mgr. Jakub Gemrot, Ph.D. <jakub.gemrot@mff.cuni.cz>*
Annotation	*Expansion of a game which was created as a project for Game development course. We will add some nontrivial mechanics and features.*

Motivation

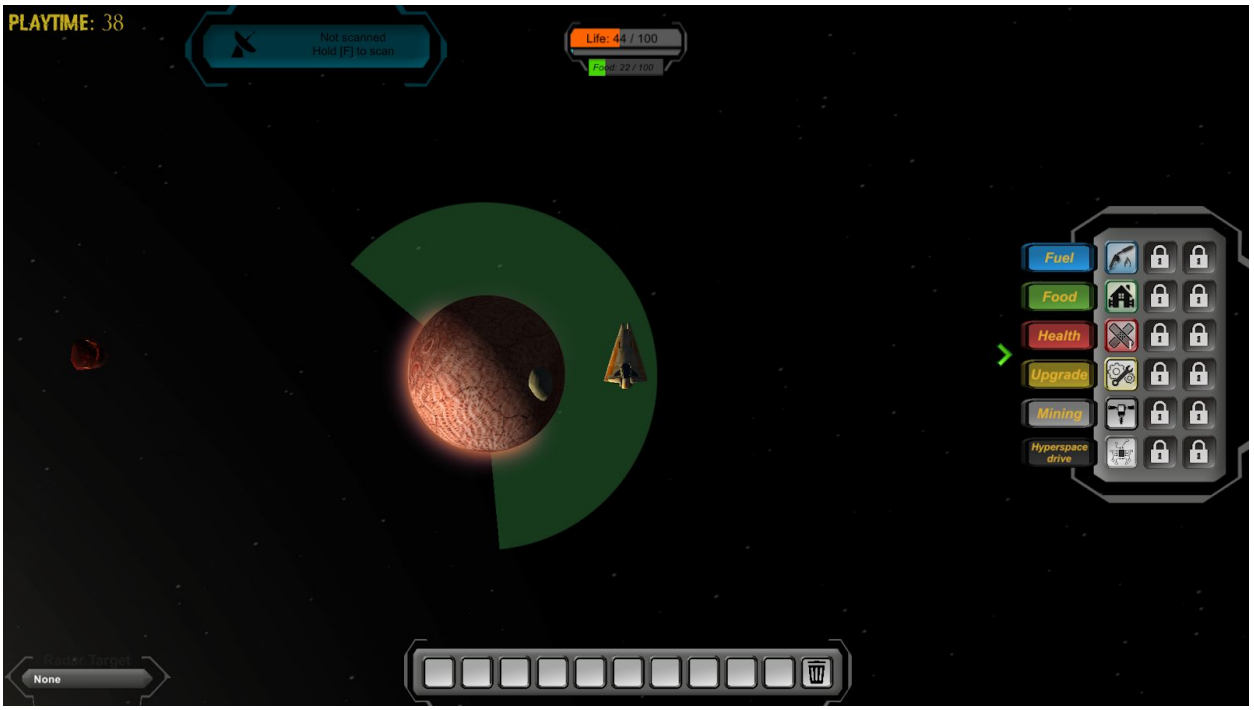
The main motivation for this software project was the desire to continue working on the game “Lost in Space”. The development of this game started during the course “Computer Games Development” in winter semester 2018/19. Informations about “Lost in Space” can be found here: https://dreamers-lostinspace.webnode.cz. For this project, we would like to expand the current state of the game with features like realistic simulations of physics, a story-campaign, a polished user interface, etc. For the outcome of the project, we are aiming to deliver a game with the “look-and-feel” of a professional game and without emerging the impression that our game is “just a school project”.

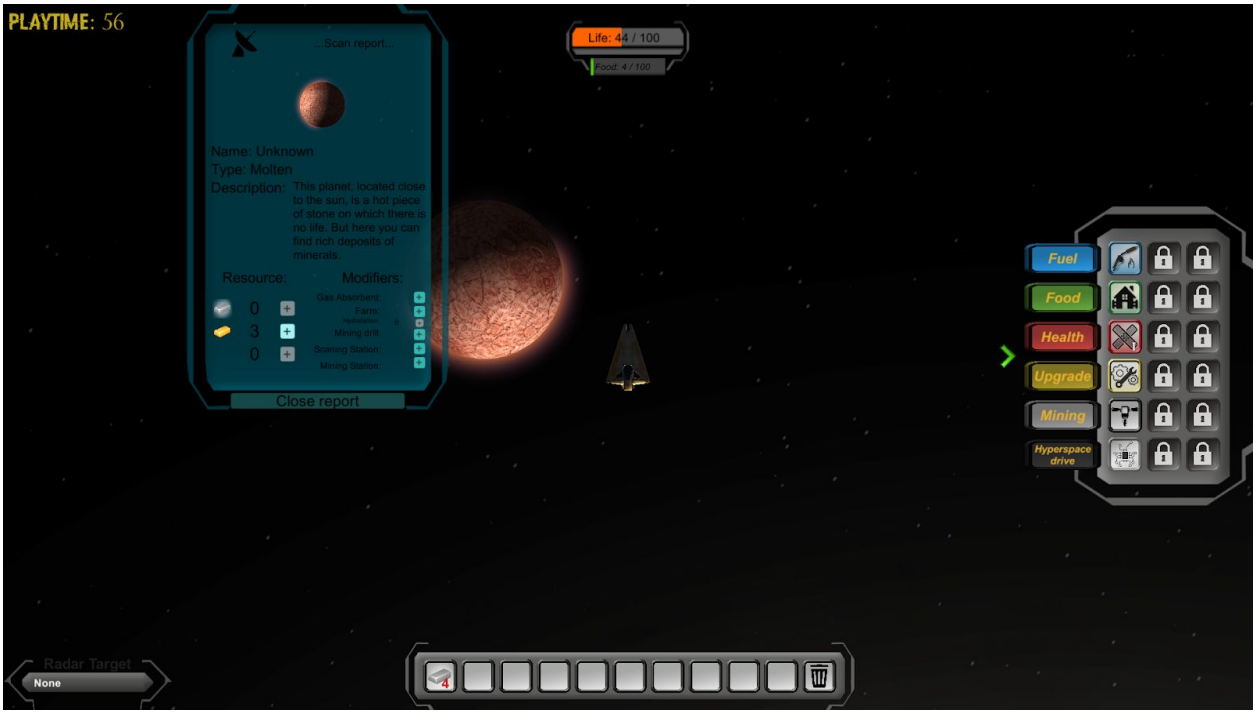
Description

State zero:

“Lost in Space” is 2.5D survival game, in the tradition of games like “Don’t Starve” and “This War of Mine”, situated in space. The player controls a spaceship flying from one planet to another with the goal of repairing his hyperspace drive to get to his/her home planet. In order to survive during the player’s journey, different items can be crafted from materials that can be found and harvested on planets or/and asteroids.

- **Basics**: player controls ship, any movement uses fuel, food is consumed by time.
- **Planets**: have “safe zone” which keep players ship in orbit if near and protects player from asteroids, can be mined for resources, can be upgraded by mining equipment.
- **Asteroids**: two types, killer asteroid that damages ship if struck, mining asteroids that yield random resources and can be picked up, special mining asteroid can be recognised by visually.
- **Inventory**: drag and drop inventory, item stacking, item activation.
- **Crafting**: three stages, player can craft items or ship upgrades.
- **Radar/minimap**: navigation arrow that point to planet player want to visit, radar can be later upgraded to minimap.





Expansion:

More influential parts:

- **Map Generation:** solar system will be generated anew every time the game starts or player gets to another level. Generation of a map will be constrained by basic rules such as presence of everything that is needed for repairing hyperspace engine (winning the game).

- **Ship modifications:** ship will be built out of parts. Every ship part will have its purpose like storage or engines. Parts can be crafted. Each ship part can be destroyed individually.
- **Story and missions:** generating missions to give player some purpose of what to do next and story to narrate players progress. There will be one main storyline which will follow the player through whole gameplay and two or three side missions which will be generated with map. Missions will have only local impact (in current solar system).
- **Combat system:** creating enemies and weapons will give players another challenge which will stand in their way to victory. System will consist of few types of weapons as well as multiple types of enemies and their AI. Also, for combat system the ship will need to be more responsive to player input.
- **Save and load options**
- **Collecting statistics**

Less influential parts:

- **Upgrade asteroids:** tweak asteroids spawning and movement to make them more predictable and possible to collect. On the other hand "killer" asteroids should be more dangerous but with better player awareness (and much less spawn rate). Make collecting asteroids more "fun" by giving player more complex interaction with it than click of one button.
- **Grid-like inventory:** items will be stored directly on the ship. Ship will have special storage part where items can be stored. Items can occupy multiple spaces of the grid inventory.
- **Realistic physics:** create more physically plausible model of gravity, orbit rotation, collision etc. One idea is using only physics to move objects in space around (player ship, asteroids and planets).

Platform and Technology

Target platform: Windows

Technology: Unity

Others: Git, Github, Audacity, Photoshop

Challenge Estimation

We are working in team with four members. Two members are from original team. Every member will be responsible for certain parts of expansion of project.

Ondrej Čakloš - new inventory system, ship modifications

Denis Judin - map generation, story and missions

Vojta Řehák - realistic physics, combat system, asteroid upgrade

Sebastian Schimper - GUI, animations and graphics, collecting statistics

Work plan:

Month 1:

- Getting new members familiar with the game
- Research what changes will be delivered to the game
- Prepare project specification

Month 2:

- Start working implementing new parts
    - GUI and visuals of original game
    - Start reworking item, crafting and inventory system
    - Start developing combat system
    - Basic map generation

Month 3:

- GUI and visuals for new components
- Basics of inventory system with working items
- Physics improvement
- Implementing story and mission concept

Month 4:

- Basic crafting system
- Continue on combat system
- Create mission generator

Month 5:

- Basic AI for enemies
- Concept of ship modularity
- Vertical slice of game (at end of month)

Month 6:

- Improving inventory and crafting system
- Updating GUI and visuals
- Mission generator
- Improving map generation

Month 7:

- Finishing on major expansion parts
- Polishing

Month 8:

- General improvements
- Create documentation
- Polishing

Month 9:

- Polishing

Definition of the Project

The project focuses on the following areas:

Discrete Models and Algorithms	
	Discrete Mathematics and Algorithms
	Geometry and Mathematical Structures in Computer Science
	Optimization
Theoretical Computer Science	
	Theoretical Computer Science
Software and Data Engineering	
	Software Engineering
X	Software Development
	Web Engineering
	Database Systems
	Analysis and Processing of large Data
Software Systems	
	System Programming
	Reliable Systems
	Powerful Systems
Mathematical Linguistics	
	Computational and Formal Linguistics
	Statistical Methods and Machine Learning in Computational Linguistics
Artificial Intelligence	
X	Intelligent Agents

	Machine Learning
	Robotics
Computer Graphics and Computer Game Development	
	Computer Graphics
X	Computer Game Development