

Computer Games Laboratory - SS 2019

Team Rocket



beyond reach

Alexander Müller

Maximilian Mayer

Jan-Philipp Fahlbusch

Lukas Goll

Contents

Contents	1
1. Formal Game Proposal	3
1.1. Game Description	3
1.1.1. Storyline	3
1.1.2. Game Idea	3
1.1.3. Gameplay	4
1.1.4. Relation to Course Theme	4
1.1.5. Concept Art and Sketches	5
1.2. Technical Achievement	8
1.2.1. Generated Solar System	8
1.2.2. AI Factions	9
1.2.3. Hex Based Building	9
1.3. "Big Idea" Bullseye	9
1.4. Development Schedule	10
1.4.1. Plan in Layers	10
1.4.2. Task List	11
1.4.3. Task Timeline	16
1.5. Assessment	20
2. Game Prototype	21
2.1. Overview	21
2.2. Rules and Gameplay	21
2.2.1. Structure	21
2.2.1.1. Resources	21
2.2.1.2. Buildings	22
2.2.1.3. Ships	26
2.2.2. Rules	28
2.2.3. Gameplay	29
2.3. Experience	31
2.4. Learnings from Prototype	31
2.5. Game Revisions	32
3. Interim Report	33
3.1. Gameplay	33
3.1.1. Core Systems	33
3.1.2. Interaction Systems	33
3.1.3. General Gameplay	34
3.2. Planet Generation	34
3.3. AI	35
3.4. Rendering	36

3.5. Models	37
3.5.1. Prefabs	37
3.5.2. LODs	38
3.5.3. Platform Model	38
3.6. UI	39
3.6.1. Functional UI	39
3.6.2. Fancy UI	39
4. Alpha Release	40
5. Playtesting	40
6. Public Presentation and Conclusion	40

Project Structure Document

1. Formal Game Proposal

1.1. Game Description

The game follows the narrative of mankind who leave their dying solar system through the help of artificial intelligence by building a galactic ark.

1.1.1. Storyline

Pressured by vanishing resources and a dying sun, mankind put their fate into the hands of their creation: Masterminds of artificial intelligence, capable of perfect rational decision making. The discovery of a planet with the requirements for life, as well as similar properties to earth sparks the torch of hope. With different ethnological ideas and beliefs concurring between the factions of the homeplanet, a space race begins with the goal to build the first giant vessel for safe passage. Its manufacture will take a lot of resources from multiple planets of the system. Only with the help of their artificial leader, people will be able to complete this journey before it is too late.

1.1.2. Game Idea

The player takes control of one of these leader AIs and tries to lead its group of people to victory, by completing multiple construction steps of the galactic ark. To do so, it will be required to collect enough resources that are scattered on the planets of the system. To harvest these, the player needs to expand its planetary base to provide means of workforce and production. Not only buildings for expansion but also factories, energy sources and housing will require strategic decision making. The base will go through stages of improvement, until it is required to embark on other planets with tougher environmental hazards to gather the required resources.

The player in the role of the AI must plan construction of his bases and maximization of building material, available energy and workforce satisfaction.

Since there are rivaling factions on the home planet, the player will compete against other AI's to be the first one finishing the ark project. Conflict between the fractions is certain, especially towards the endgame, in which later technologies require rarer materials from the edge of the solar system. This conflict might be resolved by fighting forces.

The game features a simulated solar system, with a central sun and planets with different properties. Planets will be connected by a galactic map, which enables the actors to move interplanetary.

At some stages of the game, the AI will be confronted with moral decisions, for instance using part of the population as energy resource to accelerate the expansion. The player is forced to decide as the AI, which is expected to choose the one 'right' decision, since it is rational. These moral dilemmas between technical effectiveness and human ethics will affect the game from there in terms of economic bonuses in contrast to human loyalty.

1.1.3. Gameplay

The Game takes place on spheres that represent planets of the solar system. Their surface consist of hexagonal fields, similar to strategic board games. The player is able to move the gameview around these planets in a top-down fashion. The game actors begin with a base on the homeplanet and aim to expand theirs on the hexagonal fields. They place multiple types of buildings with different effects such as mining a resource, or provide housing. Since construction requires building materials and workers, the actors have to wait until they produced enough material to continue. Therefore, optimal base planning is required to expand faster than their rivals. The effectiveness of tiles will be influenced by the adjacent environment they are placed in. Ultimately, it is the goal to succeed building the galactic ark in multiple steps faster then the others by optimizing his resource output.

At some point an actor gathers enough resources required to move ahead. He/She constructs a space port and ship units that travel on the galaxy map in between planets. These resources are spread throughout the solar system and are required to produce better technology, new tiles, improvements and parts for the ark. The ships feature different types for different purposes, for instance constructing a new base on a different planet, establishing trade routes or even combat and protection. These units are independently manageable and require some time to move between the planets.

1.1.4. Relation to Course Theme

With artificial intelligence as theme of this project, we decided to develop a game in which the player itself takes the role of an artificial intelligence. As AI is supposed to act rational, a strategy game does fit very well into the theme, since core gameplay requires to optimize the chain of actions to be more efficient than one's opponents.

Because humans steadily approach their end in this solar system, the choice of selecting an AI as their leader is justified. The top down view on a strategy game creates a illusion of almost god-like control over the human population. This fits well to an AI whose decision making is exceeding human comprehension. The usage of hexagonal shaped fields will help us to give the planets an organized and analytic feel to it, even with larger bases.

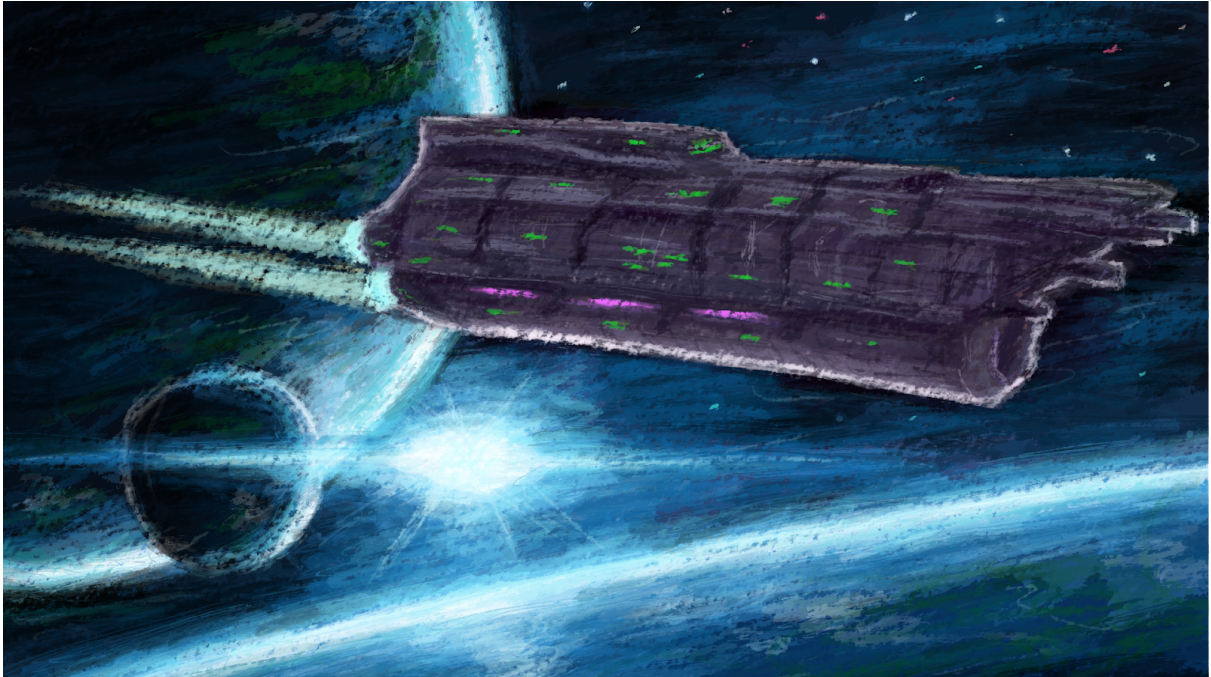
1.1.5. Concept Art and Sketches



Concept art for the look and feel of the player's base on the planet's surface. Here the player will construct buildings, such as the base, iron ore mine, steelworks, observatory, hangar, fuel extraction plant, ship building yard, ship hull factory, space station, population modul, food module, unobtainium ore refinery, advanced component factory, weapons factory, weapons alloy production, laser cell production, and the monumental shipyard for the escape vessel. Each planet will grant the player unique challenges and terrains, generated by our planet generator algorithm.



A concept art viewing the planets from the space station, which circles the planet and functions as a docking hub for larger ships. With this, the many resources featured in the game, such as iron ore, steel beam, fuel cell, ship hull plates, population, food, unobtainium ore, advanced component, weapons, weapons alloy, and laser cell, can be shipped between different planets.



This is a concept art of the large vessel for transporting the civilization to the edge of the galaxy. Building this vessel is the ultimate goal of the player and the first one reaching the edge of the galaxy wins the game.



This concept arts depicts a space battle, which is featured in our high targets. If we have enough time, we also want epic space battles between the factions as a feature in our game.



This image was taken from: <https://i.pinimg.com/originals/d1/e1/07/d1e107cdaaad05ecc217a4137ba4a533.png> . This is an idea on how the surface on our planets might look. The only difference is that our planets are round, as seen in the technical achievements.

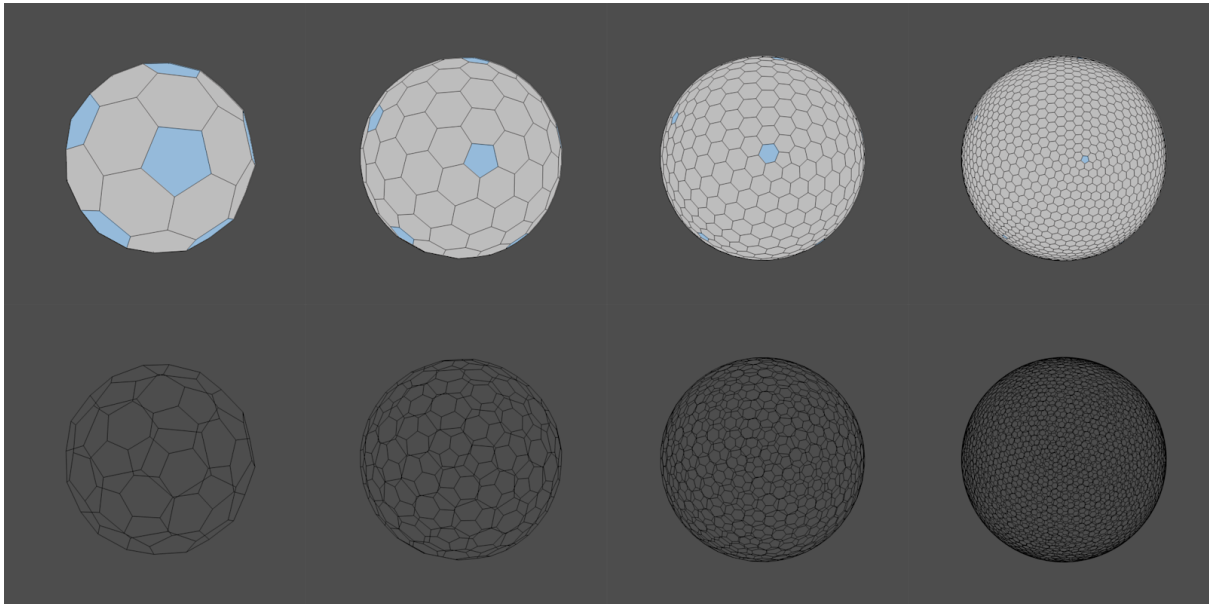


This image was taken from: <https://assetstore.unity.com/packages/3d/environments/sci-fi/polygon-sci-fi-space-pack-138857> . This is a low polygon asset pack in the unity asset store, which we will use for our game. As we have no skilled 3D artists, we choose to use this asset pack, so we can focus on programming tasks.

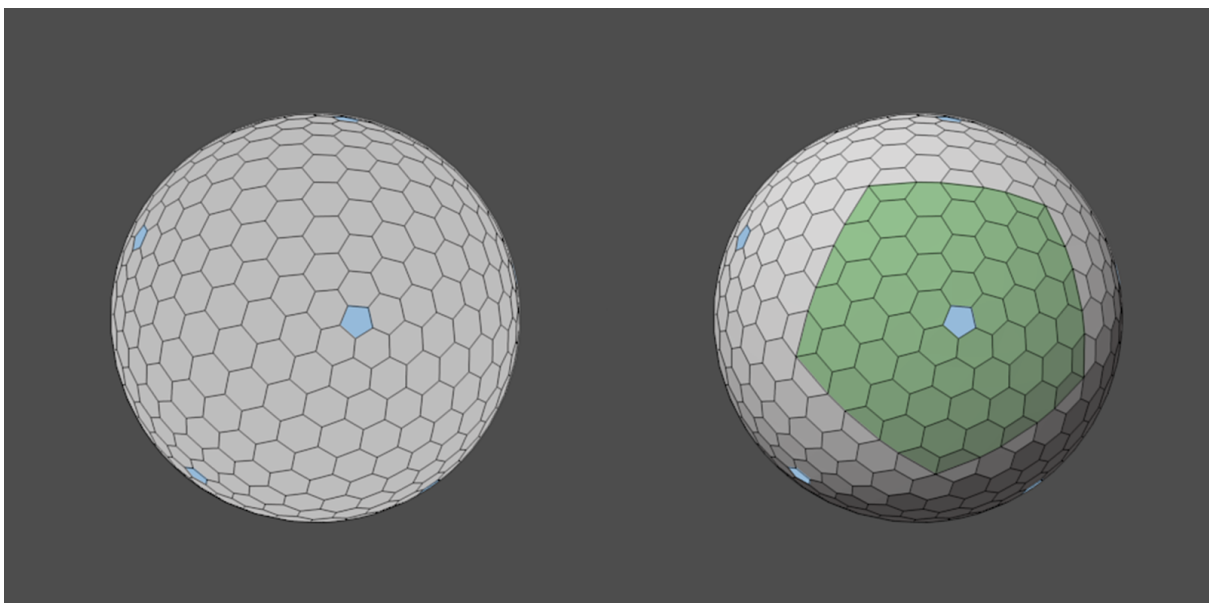
1.2. Technical Achievement

1.2.1. Generated Solar System

The solar system with its various planets is generated for each game. The planets are based on ico spheres, with their size being adjustable by adding hexagons in between the pentagons.



The planets are divided into sectors with the pentagons in their center. Since there are always 12 pentagons, each planet has 12 sectors with varying size. The sector base will always be based on the pentagon, and each sector can only be controlled by a single player.



Combining this with different atmospheres and biomes on the planets, the planets feature plenty of differences each playthrough.

1.2.2. AI Factions

Since the game is a singleplayer game, the two enemy factions are controlled by an advanced AI, capable of controlling all the functions available to the player. The AI has to adjust to the generated planets each playthrough, while still being balanced and its difficulty being comparable each playthrough.

1.2.3. Hex Based Building

Around the base building (on the pentagon), the player can build large bases out of single hex tile buildings. Different resources and biomes influence the players decisions and enforce varying playstyles each playthrough. The building serve a wide variety of functions, from resource mining and shipyards to food and science.

1.3. “Big Idea” Bullseye

In its core, the game is a multi-planetary strategy game. This allows for a wide variety of gameplay systems based on e.g. resources, habitability and biomes.

This concept is supported by the concept of generated planets. Combining different ground biomes with different atmospheres and planet sizes changes the look of those planets and refreshes the visuals for each playthrough. Random generator parameters adjust the gameplay as well and increase replayability a lot. Adding fog of war style exploration to each round forces the player to adjust his/her strategy each round and come up with new ideas on how to build the spaceship that takes the player’s people out of the solar system.



1.4. Development Schedule

1.4.1. Plan in Layers

- Functional minimum
 - Basic planet generation -> different sizes but only water
 - Sun in centre and planet rotates around sun
 - Basic camera movement around the planet
 - Resource system -> building resources for buildings (iron ore, steel beam)
 - First buildings -> base, iron ore mine, steelworks, monument shipyard -> build prefabs for models
 - Platform base model for buildings
 - Placing buildings actions
 - AI can place buildings based on blueprint(ish) system
 - Victory achievement process: Ore Mine -> Steelworks -> Shipyard(Monument)
- Low target
 - Generate more than one planet -> add land and atmospheres to planet generation
 - Build small solar system for the planets to move in and to be colonized
 - New buildings -> observatory, hangar, fuel extraction plant, ship building yard, ship hull factory (Power Plant ?) -> build prefabs for models
 - New resources -> fuel cell and ship hull plates (Electricity ?)
 - Ships -> cargo ship and small explorer -> build prefabs for models
 - More complex camera system for different modes between planets and space
 - Ship movement between planets and transfer of resources between planets
 - Discovery mechanic of new planets with the observatory
 - AI can build new buildings and move ships
 - First UIs for ingame tasks
 - First soundtrack songs
 - Victory achievement process: Ore Mine -> Steelworks -> Observatory -> Other planet base(hangar, cargo, explorer ship) -> ship hull factory -> fuel extraction -> Shipyard(Monument)
- Desirable target
 - Workforce system
 - New resources -> population, food, unobtainium ore, advanced component
 - New buildings -> space station, population modul, food module, unobtainium ore Mine unobtainium ore refinery, advanced component factory -> build prefabs for models
 - New Ships -> people carrier, large cargo ship -> build prefabs for models
 - Menus and fancy in-game UI
 - Update AI to new workforce system and new components
 - Effects and and other visual enhancements
 - Soundtrack and effects, voice over

- Victory achievement process: iron ore mine -> Steelworks -> Observatory -> Other planet base(hangar, cargo, explorer ship) -> ship hull factory -> fuel extraction -> another planet base(space station, large cargo ship, people carrier) -> unobtainium ore mine -> advanced component factory -> Shipyard(Monument)
- High target
 - Tutorial for game, explaining all concepts step by step
 - Ability to load and save game
 - Fancy visual effects (clouds around planets)
 - Add different terrain to planet generation (different production speeds for different buildings on certain terrains)
 - Add different tiers of resources, that are better, on other planets (better statistics for buildings etc.)
 - Add different tiers of buildings, if they are constructed with different tier material
 - Combat system
 - New resources -> weapons, weapons alloy, laser cell
 - New buildings -> weapons factory, weapons alloy production, laser cell production -> build prefabs for models
 - New ships -> small cruiser, battleship, large destroyer -> build prefabs for models
 - Add strategic variation to AI
- Extras
 - Add different difficulty settings
 - Add people and robots walking on different planet field
 - Ship upgrades
 - Tech tree
 - 3rd parties (neutral factions)
 - World events such as quests, hazards
 - Monument travel to edge of solar system
 - Different factions
 - Multiplayer
 - Statistics screen
 - Achievements

1.4.2. Task List

For the high resolution task list PDF, please see our project Wiki page: <https://wiki.tum.de/display/gameslab2019/Team+Rocket?preview=/234292590/234292893/Project%20Task%20List.pdf>

PROJECT DETAILS								HOURS	
STATUS	PRIORITY	START DATE	END DATE	DURATION	TASK NAME	ASSIGNEE	DESCRIPTION	ESTIMATED HOURS	ACTUAL HOURS
Game Idea Milestone									
Complete	Functional Minimum	15/04/2019	21/04/2019	6	Project Setup	Jan	Set everything up to start the project	5	4
In Progress	Functional Minimum	22/04/2019	28/04/2019	6	Report	Everyone	Write the Report for the milestone and upload to the wiki	10	0
In Progress	Functional Minimum	22/04/2019	28/04/2019	6	Presentation	Everyone	Make the Presentation for the milestone and upload to the wiki	1	0
In Progress	Functional Minimum	15/04/2019	28/04/2019	13	Game Concept	Everyone	Create the basic game concept	10	0
In Progress	Functional Minimum	15/04/2019	28/04/2019	13	Draw Concept Arts	Everyone	Draw concepts arts and sketches of the basic mechanics in the game	15	0
In Progress	Functional Minimum	15/04/2019	28/04/2019	13	Game Idea	Everyone	Define the basic game idea	10	0
In Progress	Functional Minimum	15/04/2019	28/04/2019	13	Game Design	Everyone	Define the game in more detail	20	0
In Progress	Functional Minimum	22/04/2019	28/04/2019	6	Technical Achievements	Everyone	Define the technical achievements of the game	5	0
In Progress	Functional Minimum	22/04/2019	28/04/2019	6	Development Schedule	Everyone	Define the tasks and development schedule for the game	10	0
Prototype Milestone									
Not Yet Started	Functional Minimum	29/04/2019	05/05/2019	6	Critiques	Everyone	Written critiques of all other projects as an email to the supervisors	1	0
Not Yet Started	Functional Minimum	06/05/2019	12/05/2019	6	Report	Everyone	Write the Report for the milestone and upload to the wiki	10	0
Not Yet Started	Functional Minimum	06/05/2019	12/05/2019	6	Presentation	Everyone	Make the Presentation for the milestone and upload to the wiki	1	0
Not Yet Started	Functional Minimum	06/05/2019	12/05/2019	6	Mutual Critiques	Everyone	Submit mutual critiques on the wiki (every team member separately)	1	0
Not Yet Started	Functional Minimum	29/04/2019	05/05/2019	6	Paper Prototype Building	Everyone	Build the paper prototype of the game	20	0
Not Yet Started	Functional Minimum	29/04/2019	05/05/2019	6	Paper Prototype Design	Everyone	Design the paper prototype of the game	20	0
Not Yet Started	Functional Minimum	29/04/2019	05/05/2019	6	FM Building Stats	Jan, Maxi	Define the statistics (cost, production etc.) of the bulings in the functional minimum stage	1	0
Not Yet Started	Functional Minimum	29/04/2019	12/05/2019	13	Loading Data System	Jan, Maxi	Implement the loading of building and other important data from file	2	0
Not Yet Started	Functional Minimum	06/05/2019	12/05/2019	6	FM Building Models	Jan, Maxi	Build the prebabs for the building models in the functional minimum stage	8	0
Not Yet Started	Functional Minimum	29/04/2019	12/05/2019	13	FM Building Implementation	Jan, Maxi	Implement the functional minimum buildings into the gameplay	10	0
Not Yet Started	Functional Minimum	29/04/2019	05/05/2019	6	Unity Setup	Alex	Setup the Unity project with assets and settings needed for our game	2	0
Not Yet Started	Functional Minimum	29/04/2019	05/05/2019	6	Controls Design	Jan, Maxi	Design the control scheme of the game	1	0
Not Yet Started	Functional Minimum	06/05/2019	12/05/2019	6	Playtesting Paper Prototype	Everyone	Playtest the paper prototype	10	0
Not Yet Started	Functional Minimum	06/05/2019	12/05/2019	6	Refining Paper Prototype	Everyone	Refine the paper prototype based on the playtesting sessions	10	0
Not Yet Started	Functional Minimum	06/05/2019	12/05/2019	6	Basic Planet Generation	Alex	Build the basic planet generation for our game	10	0

162

0

Not Yet Started	Functional Minimum	06/05/2019	12/05/2019	6	Lighting and Sun	Alex	Build the sun in the middle of the solar system and implement the light emission	1	0
Not Yet Started	Functional Minimum	06/05/2019	12/05/2019	6	Basic Planet Movement	Jan, Maxi	Implement the planets moving around the sun	2	0
Not Yet Started	Functional Minimum	29/04/2019	05/05/2019	6	Basic Controls	Jan, Maxi	Implement the basic controls needed on the first planet	4	0
Not Yet Started	Functional Minimum	29/04/2019	05/05/2019	6	Tools Setup	Alex	Setup and build all tools needed for the project	5	0
Not Yet Started	Functional Minimum	06/05/2019	12/05/2019	6	Basic Resource System	Jan, Maxi	Implement the first basic resource system	2	0
Not Yet Started	Functional Minimum	29/04/2019	05/05/2019	6	Modelling Placeholder Planet	Alex	Model the placeholder for the planet, so that gameplay programming can begin	2	0
Not Yet Started	Functional Minimum	29/04/2019	05/05/2019	6	Modelling of Platform	Alex	Model the base platform, which will house all buildings in our game	4	0
Not Yet Started	Functional Minimum	29/04/2019	05/05/2019	6	Building Placement System	Jan, Maxi	Implement the placement system for the buildings on the planet	10	0
Not Yet Started	Functional Minimum	29/04/2019	12/05/2019	13	Basic AI	Lukas	Implement the basic AI of the enemies for the functional minimum stage	20	0
Not Yet Started	Functional Minimum	06/05/2019	12/05/2019	6	Basic Victory Condition	Jan, Maxi	Implement the first victory condition in the functional minimum stage	5	0

Interim Demo Milestone

Not Yet Started	Functional Minimum	27/05/2019	02/06/2019	5	Report	Everyone	Write the Report for the milestone and upload to the wiki	10	0
Not Yet Started	Functional Minimum	27/05/2019	02/06/2019	5	Presentation	Everyone	Make the Presentation for the milestone and upload to the wiki	1	0
Not Yet Started	Functional Minimum	13/05/2019	19/05/2019	6	Playtesting FM	Everyone	Playtest the functional minimum game from the previous milestone	5	0
Not Yet Started	Functional Minimum	13/05/2019	19/05/2019	6	Adjust Gameplay	Jan, Maxi	Adjust the functional minimum gameplay based on the findings in the playtest	6	0
Not Yet Started	Functional Minimum	13/05/2019	19/05/2019	6	Fine-Tune FM AI	Lukas	Fine-tune the AI based on the findings in the playtest	6	0
Not Yet Started	Functional Minimum	13/05/2019	19/05/2019	6	FM Bug Fixing	Everyone	Fix any bugs found during the playtest and polish the gameplay	10	0
Not Yet Started	Functional Minimum	13/05/2019	19/05/2019	6	Multiple Planet Generation	Alex	Implement the generation of multiple planets and polish the current generation process	10	0
Not Yet Started	Low Target	20/05/2019	02/06/2019	12	Add Components to Generation	Alex	Add new components to the planet generation, such as land masses, atmosphere and animated water	35	0
Not Yet Started	Low Target	13/05/2019	19/05/2019	6	Build Solarsystem	Jan, Maxi	Build a moving solarsystem out of the planets	4	0
Not Yet Started	Low Target	13/05/2019	19/05/2019	6	LT Building Stats	Jan, Maxi	Define the statistics of the new buildings added in the low target	1	0
Not Yet Started	Low Target	13/05/2019	26/05/2019	13	LT Building Models	Jan, Maxi	Build the prefab models for all the buildings	10	0
Not Yet Started	Low Target	13/05/2019	26/05/2019	13	LT Building Implementation	Jan, Maxi	Implement the low target buildings into the game	6	0
Not Yet Started	Low Target	20/05/2019	26/05/2019	6	Add LT Resources	Jan, Maxi	Add the low target resources into the game	3	0
Not Yet Started	Low Target	20/05/2019	26/05/2019	6	LT Ship Stats	Jan, Maxi	Define the statistics of the ships added in the low target	1	0
Not Yet Started	Low Target	20/05/2019	02/06/2019	12	LT Ship Models	Jan, Maxi	Build the prefabs for the low target ship models	4	0
Not Yet Started	Low Target	20/05/2019	02/06/2019	12	LT SHIP Implementation	Jan, Maxi	Implement the low target ships into the game	4	0
Not Yet Started	Low Target	20/05/2019	26/05/2019	6	Advanced Controls	Jan, Maxi	Implement the advanced controls for the camera, which differentiates between the planet and the general solarsystem	8	0

219 0

Not Yet Started	Low Target	20/05/2019	26/05/2019	6	Ship Selection System	Jan, Maxi	Implement the system to select and command ships to new positions	10	0
Not Yet Started	Low Target	20/05/2019	26/05/2019	6	Ship Pathfinding System	Jan, Maxi	Implement the pathfinding system for the ships	6	0
Not Yet Started	Low Target	27/05/2019	02/06/2019	5	Resource Transfer System	Jan, Maxi	Implement the resource transfer system between two planets	12	0
Not Yet Started	Low Target	27/05/2019	02/06/2019	5	Discovery Mechanic	Jan, Maxi	Implement the discovery mechanic, to discover new planets with the observatory	12	0
Not Yet Started	Low Target	20/05/2019	02/06/2019	12	AI Additions for LT	Lukas	Adjust and implement the newly added features from the low target to the AI implementation	35	0
Not Yet Started	Low Target	20/05/2019	02/06/2019	12	Simple In-Game UI	Alex	Design and implement first simple UIs for in-game actions, such as selecting and placing buildings and transferring goods	10	0
Not Yet Started	Low Target	20/05/2019	02/06/2019	12	First Soundtracks	Maxi	Create and add first soundtrack samples for the game (background music)	5	0
Not Yet Started	Low Target	27/05/2019	02/06/2019	5	LT Victory Condition	Jan, Maxi	Implement the victory conditions for the low target game	5	0

Alpha Release Milestone

Not Yet Started	Functional Minimum	17/06/2019	23/06/2019	6	Report	Everyone	Write the Report for the milestone and upload to the wiki	10	0
Not Yet Started	Functional Minimum	17/06/2019	23/06/2019	6	Presentation	Everyone	Make the Presentation for the milestone and upload to the wiki	1	0
Not Yet Started	Low Target	03/06/2019	09/06/2019	6	Playtesting LT	Everyone	Playtest the low target game from the previous milestone	5	0
Not Yet Started	Low Target	03/06/2019	09/06/2019	6	Adjust Gameplay	Jan, Maxi	Adjust the gameplay based on the findings in the playtest	6	0
Not Yet Started	Low Target	03/06/2019	09/06/2019	6	Fine-Tune LT AI	Lukas	Fine-tune the AI based on the experience in the playtests	10	0
Not Yet Started	Low Target	03/06/2019	09/06/2019	6	LT Bug Fixing	Everyone	Fix bugs encountered during the playtesting	10	0
Not Yet Started	Low Target	03/06/2019	09/06/2019	6	Planet Generation Fine-Tuning	Alex	Fine-tune the generation of the planets based on the playtest	6	0
Not Yet Started	Desirable Target	03/06/2019	09/06/2019	6	Workforce System	Jan, Maxi	Define the workforce system that will be added during the desirable target stage	2	0
Not Yet Started	Desirable Target	03/06/2019	09/06/2019	6	Add DT Resources	Jan, Maxi	Implement the resources of the desirable target stage, including to fully implement the workforce system	10	0
Not Yet Started	Desirable Target	03/06/2019	09/06/2019	6	DT Building Stats	Jan, Maxi	Define the statistics for the desirable target buildings	1	0
Not Yet Started	Desirable Target	03/06/2019	16/06/2019	13	DT Building Models	Jan, Maxi	Build the prefab models for the desirable target buildings	10	0
Not Yet Started	Desirable Target	03/06/2019	16/06/2019	13	DT Building Implementation	Jan, Maxi	Implement the new buildings for the desirable target stage	15	0
Not Yet Started	Desirable Target	10/06/2019	16/06/2019	6	DT Ship Stats	Jan, Maxi	Define the statistics of the desirable target ships	1	0
Not Yet Started	Desirable Target	10/06/2019	23/06/2019	13	DT Ship Models	Jan, Maxi	Build the prefab models for the desirable target ships	4	0
Not Yet Started	Desirable Target	10/06/2019	23/06/2019	13	DT Ship Implementation	Jan, Maxi	Implement the new ships for the desirable target stage	4	0
Not Yet Started	Desirable Target	03/06/2019	23/06/2019	20	Game Menu UI	Alex	Implement a game menu to start the game, as well as a pause menu during game play	8	0
Not Yet Started	Desirable Target	03/06/2019	23/06/2019	20	Fancy In-Game UI	Alex	Implement fancy in-game UI, for better usability of our game	10	0
Not Yet Started	Desirable Target	10/06/2019	23/06/2019	13	AI Additions for DT	Lukas	Add all new features from the desirable target to the AI system	35	0
Not Yet Started	Desirable Target	03/06/2019	23/06/2019	20	Visual Effects	Alex	Add different visual effects to the game	10	0

208 0

Not Yet Started	Desirable Target	03/06/2019	23/06/2019	20	Visual Enhancements	Alex	Add different visual enhancements to the game	10	0	0
Not Yet Started	Desirable Target	03/06/2019	23/06/2019	20	Soundtrack	Maxi	Add more soundtracks to the game (menu music, planet music, etc.)	10	0	0
Not Yet Started	Desirable Target	03/06/2019	23/06/2019	20	Sound Effects	Maxi	Add different soundeffects to the game	10	0	0
Not Yet Started	Desirable Target	17/06/2019	23/06/2019	6	Playtesting DT	Everyone	Playtest the almost final version of the desirable target game	5	0	0
Not Yet Started	Desirable Target	17/06/2019	23/06/2019	6	Add Simple Tutorial	Alex	Add a simple tutorial for the upcoming playtests	6	0	0
Not Yet Started	Desirable Target	17/06/2019	23/06/2019	6	Adjust Gameplay	Jan, Maxi	Adjust the gameplay based on the findings in the playtest	6	0	0
Not Yet Started	Desirable Target	17/06/2019	23/06/2019	6	DT Victory Condition	Jan, Maxi	Add the victory conditions for the desirable target	3	0	0

Playtesting Milestone

Not Yet Started	Functional Minimum	01/07/2019	07/07/2019	6	Report	Everyone	Write the Report for the milestone and upload to the wiki	10	0	0
Not Yet Started	Functional Minimum	01/07/2019	07/07/2019	6	Presentation	Everyone	Make the Presentation for the milestone and upload to the wiki	1	0	0
Not Yet Started	Desirable Target	24/06/2019	30/06/2019	6	Playtesting DT	Everyone	Playtest the desirable target game	6	0	0
Not Yet Started	Desirable Target	24/06/2019	30/06/2019	6	Adjust Gameplay	Jan, Maxi	Adjust the gameplay based on the playtest	10	0	0
Not Yet Started	Desirable Target	24/06/2019	30/06/2019	6	Fine-Tune DT AI	Lukas	Fine-tune the AI based on the playtest	10	0	0
Not Yet Started	Desirable Target	24/06/2019	30/06/2019	6	DT Bug Fixing	Everyone	Fix any bugs encountered during playtesting	10	0	0
Not Yet Started	Desirable Target	24/06/2019	30/06/2019	6	Gameplay Polishing	Jan, Maxi	Polish the gameplay for the playtest session	5	0	0
Not Yet Started	Desirable Target	24/06/2019	30/06/2019	6	AI Polishing	Lukas	Polish the AI behaviour for the playtest session	5	0	0
Not Yet Started	Functional Minimum	24/06/2019	30/06/2019	6	Create Questionary	Everyone	Create the questionary for the the playtest session	2	0	0
Not Yet Started	Functional Minimum	24/06/2019	30/06/2019	6	Playtest Scenario	Everyone	Define the playtest session and what the participants are supposed to do	2	0	0
Not Yet Started	Functional Minimum	01/07/2019	07/07/2019	6	Playtest Session	Everyone	Playtest the game with external players	30	0	0
Not Yet Started	Desirable Target	24/06/2019	30/06/2019	6	Visual Polishing	Alex	Polish the visuals of the game	10	0	0
Not Yet Started	Desirable Target	24/06/2019	30/06/2019	6	Sound Polishing	Maxi	Polish the sound of the game	5	0	0
Not Yet Started	Desirable Target	24/06/2019	30/06/2019	6	UI Polishing	Alex	Polish the UI of the game	5	0	0
Not Yet Started	High Target	24/06/2019	07/07/2019	13	Advanced Tutorial	Alex	Create an advanced and more detailed tutorial	6	0	0
Not Yet Started	High Target	01/07/2019	07/07/2019	6	Strategic Variation for AI	Lukas	Start implementing strategic variations in the AI behaviour	10	0	0
Not Yet Started	Desirable Target	01/07/2019	07/07/2019	6	Apply Feedback to Game	Everyone	Apply the feedback from the playtest to the game and fix any complaints	20	0	0
Not Yet Started	Desirable Target	01/07/2019	07/07/2019	6	Bug Fixing	Everyone	Fix any bugs encountered during playtesting	10	0	0
Not Yet Started	Desirable Target	01/07/2019	07/07/2019	6	Polishing	Everyone	Polish any unpolished aspects encountered during playtesting	10	0	0
Not Yet Started	Functional Minimum	01/07/2019	07/07/2019	6	Feedback Analysis	Everyone	Analyse the feedback from the playtest session	2	0	0

Final Release Milestone

Not Yet Started	Functional Minimum	15/07/2019	21/07/2019	6	Report	Everyone	Write the Report for the milestone and upload to the wiki	10	0	0
-----------------	--------------------	------------	------------	---	--------	----------	---	----	---	---

138 0

169 0

Not Yet Started	Functional Minimum	15/07/2019	21/07/2019	6	Presentation	Everyone	Make the Presentation for the milestone and upload to the wiki!	1	0
Not Yet Started	Functional Minimum	15/07/2019	21/07/2019	6	Video	Everyone	Make a video that highlights exciting aspects of the game	2	0
Not Yet Started	Functional Minimum	15/07/2019	21/07/2019	6	Completed Game	Everyone	Completed final version of game with sources	1	0
Not Yet Started	Desirable Target	08/07/2019	21/07/2019	13	Bug Fixing	Everyone	Fix all known bugs for final version	20	0
Not Yet Started	Desirable Target	08/07/2019	21/07/2019	13	Polishing	Everyone	Polish everything for final version	20	0
Not Yet Started	Desirable Target	08/07/2019	21/07/2019	13	Playtesting	Everyone	Playtest the whole game to find any rough edges	20	0
Not Yet Started	High Target	08/07/2019	21/07/2019	13	Fancy Visuals	Alex	If time, add fancy visuals (clouds around planet)	10	0
Not Yet Started	High Target	08/07/2019	21/07/2019	13	Different Terrain	Alex	If time, add different terrain to planet generation (different production speeds)	10	0
Not Yet Started	High Target	08/07/2019	21/07/2019	13	Save and Load System	Jan, Maxi	If time, implement a save and loading system	10	0
Not Yet Started	High Target	08/07/2019	21/07/2019	13	Resource Tier System	Jan, Maxi	If time, implement a tier system for the resources	5	0
Not Yet Started	High Target	08/07/2019	21/07/2019	13	Strategic Variation for AI	Lukas	If time, implement more variations for the playstyles of the AI	10	0
Not Yet Started	Extras	08/07/2019	21/07/2019	13	Difficulty Settings	Jan, Maxi, Lukas	If time, implement different difficult settings	10	0
Not Yet Started	Extras	08/07/2019	21/07/2019	13	Animated Buildings	Alex	If time, implement animated people walking around on the buildings on a planet	5	0
Not Yet Started	Extras	08/07/2019	21/07/2019	13	Monument Travel	Jan, Maxi	If time, make the goal to travel with the ship to the edge of the solar system	4	0

1.4.3. Task Timeline

For the high resolution task list PDF, please see our project Wiki page:

<https://wiki.tum.de/display/gameslab2019/Team+Rocket?preview=/234292590/234292894/Project%20Timeline.pdf>

PHASE	TARGET CODE	DEV	DETAILS	PROJECT WEEK:				APRIL			
				15	22	29	6	13	M		
	Functional Minimum Target	Low Target	Desirable Target	High Target	Extra						
1	Game Design	Everyone	- Deliverables	Project Setup	Report	Critiques	Report Presentation	Playtesting PM			
		Everyone	- Playlist	Game Concept Arts	Draw Concept Arts	Paper Prototype Design	Refining Paper Prototype	LT Building State			
2	Modelling	Jan, Maxi	- Base Models			Director's Plan!	FM Building Models	LT Building Models			
		Alex	- Tile Models			Game Building Platform					
3	Scripting	Jan, Maxi	- Special Effects								
		Alex	- Moving Models								
4	Rendering	Jan, Maxi	- Gameplay			FM Building Implementation	Build Solver system				
		Lukas	- Artificial Intelligence			Basic AI	LT Building Implementation	Multiple Player Generation			
5	UI	Alex	- Player Generation			Building Placement System	Loading Data System!				
		Jan, Maxi	- Game Systems			Basic Control System	Basic Resource System				
6	Sound	Alex	- Tools			Unity Setup					
		Everyone	- Bugfixing & Polishing					Adjust Gameplay			
4	Rendering	Alex	- Render Backend								
		Alex	- Tech Art				Lighting and Sun				
5	UI	Alex	- Shaders								
		Alex	- Menus				Water Shader	Water Shader			
5	UI	Alex	- InGame								
		Alex	- HUD								
6	Sound	Maxi	- Effects								
		Maxi	- Soundtrack								

Game Idea Milestone

Prototype Milestone

PHASE	TARGET CODE	DEV			PROJECT WEEK:	Q2			JUN		
		Functional Low Medium Target	Devide High Target	Extra		IV	20	27	3	10	17
1	Game Design	Everyone	- Deliverables				Report Presentation			Report Presentation	
		Everyone	- Playtest				Playtesting LT			Playtesting DT	
2	Modelling	Jan, Maxi	- Base Models		LT Building Models			DT Building Models			
		Alex	- Tie Models				Visual Effects		Visual Effects	Visual Effects	
3	Scripting	Jan, Maxi	- Special Effects		LT Ship Models			DT Ship Models		DT Ship Models	
		Alex	- Moving Models				Visual Effects		Visual Effects	Visual Effects	
4	Rendering	Jan, Maxi	- Gameplay		Add LT Resources			Add DT Resources			
		Lukas	- Artificial Intelligence		LT Building Implementation			DT Building Implementation		DT Ship Implementation	
5	UI	Alex	- Planet Generation		LT Ship Implementation			DT Ship Implementation		DT Ship Implementation	
		Jan, Maxi	- Game Systems		AI Additions for LT			AI Additions for DT		AI Additions for DT	
6	Sound	Alex	- Tools		Add Components to Generation			Planet Generation Fine-Tune			
		Everyone	- Bugfixing & Polishing		Advanced Controls for Ship Purification System			Resource Transfer System			
4	Rendering	Alex	- Render Backend		Adjust Gameplay			Adjust Gameplay			
		Alex	- Tech Art		LT Bug Fixing			Visual Enhancements		Visual Enhancements	
5	UI	Alex	- Shaders		Visual Enhancements			Visual Enhancements		Visual Enhancements	
		Alex	- Menus		Planet Shader			Planet Shader			
6	Sound	Alex	- In-Game		Game Menu UI			Game Menu UI		Game Menu UI	
		Maxi	- HUD		Simple In-Game UI			Fancy In-Game UI		Fancy In-Game UI	
6	Sound	Maxi	- Effects		Sound Effects			Sound Effects		Sound Effects	
		Maxi	- Soundtrack		First Soundtracks			Soundtrack		Soundtrack	

Interim Demo Milestone

Alpha Release Milestone

PHASE	TARGET CODE				DEV	DETAILS				08			
	Functional Minimum	Low Target	Detail Target	High Target		Exns	PROJECT WEEK:						
1	Game Design	Everyone	- Deliverables - Playtest - Comments	Jan, Maxi	- Base Modelle - Tile Models - Special Effects - Moving Models	24	1	8	15	22	29	JUL	
						Report Presentation	Playtesting	Report Presentation Video Complete Game	Playtesting Milestone				
2	Modelling	Jan, Maxi	- Terrain Models - Tile Models - Special Effects - Moving Models	Jan, Maxi	- Gameplay - Terrain Generation - Game Systems - Tools	24	1	8	15	22	29	JUL	
						Adjust Gameplay	Apply Feedback to Game	Animated Buildings	Animated Buildings	Final Release Milestone			
3	Scripting	Lukas	- Artificial Intelligence - Planet Generation - Game Systems - Tools	Jan, Maxi	- AI Building - Bug Fixing - General Polishing	24	1	8	15	22	29	JUL	
						Strategic Variation for AI	Strategic Variation for AI	Strategic Variation for AI	Strategic Variation for AI	Final Release Milestone			
4	Rendering	Alex	- Render Backend - Tech Art - Shaders	Alex	- DT Bug Fixing - General Polishing	24	1	8	15	22	29	JUL	
						Visual Polishing	Polishing	Bug Fixing	Bug Fixing	Final Release Milestone			
5	UI	Alex	- Menus - In-Game - HUD	Alex	- Advanced Tutorial - UI Polishing - Sound Polishing	24	1	8	15	22	29	JUL	
						Advanced Tutorial	Polishing	Advanced Tutorial	Polishing	Final Release Milestone			
6	Sound	Maxi	- Effects - Soundtrack	Maxi	- Sound Polishing	24	1	8	15	22	29	JUL	
						Sound Polishing	Polishing	Sound Polishing	Polishing	Final Release Milestone			

1.5. Assessment

In our game players embark on a journey through a solar system in which they race for survival in an epic battle of expansion. Resource planning, strategy making or population organisation are only few of the many tasks awaiting players. Starting off in small scope with construction of resource facilities on the home planet, continuing with expansion to other sectors and growing more population, up to exploring the seemingly endless solar system and claiming one's leadership in the race to universe - there are endless possibilities.

Every game cycle will feel and play differently as the random generation of the solar system mixes the world together. Each playthrough can have a different outcome depending on the starting configurations and the decisions the players make throughout the game. As the computer controlled AI is the focus of our project, it contributes a lot to the diversity and is the best incentive for our players to enjoy another round in the manifold universe.

We aim for an audience that is driven by an explorative and construction-loving mind. While already highschool teenagers with sense for adventures might like the game, it is also compelling for every strategy lover who wants slightly more challenge. In general, the audience's age is not restricted by an upper bound. As no to few violence will appear in the game and due to the adjustable difficulty, a wide range of players can be addressed. As a comparable audience fans and followers of games like *Sid Meier's Civilization* or *Stellaris* could be named.

For judging the design success in the end there are multiple factors that have to be taken into account:

- The game itself should be attractive in terms of looks and sounds to the player and feel new every round. The random generation of worlds has to make players want to try the game again in new and different settings over and over again.
- It is necessary that the gameplay itself is immersive and pulls the player into the world by well balanced features, progression and decisions that a player can make.
- The AI has to pose a serious opponent for the player which makes him/her go with the flow in a balance between ability and challenge.

If all those aspects work together, the final product will be a very good looking and charming interstellar strategy game, that offers enough complexity for players to enjoy as well as a challenging, but not unfair, AI that can keep up to expectations of human players.

2. Game Prototype

2.1. Overview

As we got very diverse feedback for our game on the wiki page, we really wanted to evaluate every aspect of the game. As the only aspect almost every person favored and nobody disliked was the planets being made from hexagon fields, we left this featured unchanged. Everything else was reevaluated, because for every other feature we got both positive and negative feedback. This is why we tried to recreate the core gameplay loop with our paper prototype and evaluate if each feature contributes to our targeted gameplay feel, which is to have a fun and challenging real time strategic building game. Everything, that does not contribute to this, will be moved towards high target, extra, or removed entirely from the concept. As combat was already a high target feature, we decided to move it further back, as it does not add as much to our concept, compared to the time and energy we would have to spend for implementing it. This is also the reason, why it is not included in our paper prototype.

2.2. Rules and Gameplay

2.2.1. Structure

The structure chapter of our game prototype will include all resources, buildings and ships used during the prototype process. All these things will be used with similar balancing later in our game (unless serious issues arise). By including the full range of our content in the paper prototype, we wanted to test, if our desirable target is fun and challenging to play and does feel like a strategic building game, similar to the Anno series. The abbreviations behind the names stand for the desired target for which we plan to implement this feature into our game (FM = Functional Minimum; LT = Low Target; DT = Desirable Target; HT = High Target). The time steps are for defining how long something takes. During the prototype, at each time step actions can be performed. In our final game, one time step will be measured in seconds and is currently thought to equal 15 seconds per time step.

2.2.1.1. Resources

Iron Ore (FM):

This is the base resource in the game and one that is needed a lot, as many secondary products require iron as a base element.

Steel Beam (FM):

Steel beams consist of iron ore and are used to construct many buildings.

Fuel (LT):

This resource can only be mined on planets with this resource available. They are not found on the starting planet but usually on those close to the starting planet.

Fuel Cell (LT):

Fuel cells need iron ore and fuel as a resource. With this, the cell can be build and filled with fuel. Fuel cells are needed for electricity and ships. Fuel can only be produced on planets that have the required resource reservoirs.

Ship Hull Plates (LT):

Ship hull plates are made from iron and steel beams as the frame. These plates are mainly used for the ship construction.

Electricity (HT):

Electricity is used by every building in the game. This is a supply value which is not consumed, but rather needs to be high enough to allow all buildings to be supplied with enough electricity. New buildings can only be built if the electricity value is sufficient after the building is placed.

Workforce (Population) (DT):

Workforce is similar to the electricity resource. It is also not consumed and every building needs a certain amount of workforce to function. New buildings can only be built if the workforce value is sufficient after the building is placed.

Food (DT):

Food is a basic need for the population and one of the most important resource, because without food the population in a sector will decrease.

Unobtainium Ore (DT):

This ore is very rare and only found on planets on the edge of the solar system. It is mainly needed for completing the monument ship.

Refined Unobtainium Ore (DT):

The next stage of the unobtainium ore, before it can be processed into a building material.

Advanced Circuits (DT):

The last stage of the unobtainium ore, combined with iron ore, will result in this advanced building material.

2.2.1.2. Buildings

Base (FM):

- **Description:** The first base is always already built for the player and AI. The base is the center of all activity. Other buildings have to be connected to the base building in order to function. The base also supplies the sector with certain default values, so that the player can start to produce the basic needs, before focusing on the more complex mechanics, such as population and electricity. The first 3 bases on each planet have the same costs. The base as an unlimited depo for resources.

- **Construction Cost:** First: Free; 1-3: 70 Steel Beam, 50 Food; 4-8: 140 Steel Beams, 100 Food, 50 Hull Plates, 50 Fuel Cells; 9-12: 280 Steel Beams, 200 Food, 100 Hull Plates, 100 Fuel Cells, 30 Advanced Circuits
- **Default Supply:** 40 Steel Beams, 200 Electricity, 20 Workforce
- **Produces:** None
- **Consumes:** None

Population Habitat (DT):

- **Description:** This is the housing for the workforce/population on the planet and is a passive provider. When the food supply is not sufficient, the population will decrease. This can be placed on any planet.
- **Construction Cost:** 10 Steel Beams, 2 Electricity
- **Produces:** 30 Workforce
- **Consumes:** 1 Food per 2 time steps

Iron Ore Mine (FM):

- **Description:** Building is used to mine the iron ore. This can be placed on any planet.
- **Construction Cost:** 6 Steel Beam, 1 Electricity, 3 Workforce
- **Produces:** 1 Iron Ore per 1 time step
- **Consumes:** None

Unobtainium Ore Mine (DT):

- **Description:** This mine can only be placed on planets with the unobtainium resource available. This is most likely the case on planets at the edge of the solar system.
- **Construction Cost:** 60 Steel Beams, 40 Hull Plates, 100 Electricity, 10 Workforce
- **Produces:** 1 Unobtainium Ore per 1 time step
- **Consumes:** None

Food Production Farm (DT):

- **Description:** Produces food for the population
- **Construction Cost:** 5 Steel Beams, 1 Electricity, 1 Workforce
- **Produces:** 1 Food per 1 time step
- **Consumes:** None

Unobtainium Ore Refinery (DT):

- **Description:** This building refines the unobtainium ore for the advanced circuits
- **Construction Cost:** 75 Steel Beam, 40 Hull Plates, 100 Electricity, 10 Workforce
- **Produces:** 1 Refined Unobtainium Ore per 2 time steps
- **Consumes:** 1 Unobtainium Ore per 2 time steps

Steelworks (FM):

- **Description:** Produces the basic building material early in the game. This is a very important building to build at the beginning
- **Construction Cost:** 3 Steel Beam, 3 Electricity, 1 Workforce
- **Produces:** 1 Steel Beam per 2 time steps
- **Consumes:** 1 Iron Ore per 2 time steps

Fuel Extraction Plant (LT):

- **Description:** This extracts fuel from planets which have this resource available
- **Construction Cost:** 20 Steel Beams, 20 Electricity, 10 Workforce
- **Produces:** 1 Fuel per 2 time steps
- **Consumes:** None

Fuel Cell Factory (LT):

- **Description:** This produces the fuel cells, which can then be used on ships and other buildings
- **Construction Cost:** 30 Steel Beams, 15 Ship Hull Plates, 30 Electricity, 20 Workforce
- **Produces:** 1 Fuel cell per 1 time step
- **Consumes:** 1 Fuel and 1 iron ore per 1 time step

Ship Hull Factory (LT):

- **Description:** Produces Ship hulls, which are mainly used for building ships.
- **Construction Cost:** 10 Steel Beams, 5 Electricity, 5 Workforce
- **Produces:** 1 Ship Hull Plate per 2 time steps
- **Consumes:** 1 Steel Beam and 1 Iron Ore per 2 time steps

Advanced Circuit Factory (DT):

- **Description:** This factory produces the advanced circuits, which is the endgame resource. It is rare and expensive to produce, but used in all late game structures
- **Construction Cost:** 100 Steel Beams, 50 Hull Plates, 20 Fuel Cells, 100 Electricity, 20 Workforce
- **Produces:** 1 Advanced Component per 4 time steps
- **Consumes:** 1 Refined Unobtainium Ore, 1 Steel Beam, 1 Hull Plates per 4 time steps

Ship Building Yard (LT):

- **Description:** This shipyard is for building new ships of any size (except the monument ship) The production cycles of ships take longer and will consume resources at every step (see ships for more information)
- **Construction Cost:** 40 Steel Beams, 10 Ship Hulls, 50 Electricity, 20 Workforce
- **Produces:** Ships
- **Consumes:** Specified by the selected ship

Powerplant (HT):

- **Description:** This plant produces power for the sectors on that planet to use.
- **Construction Cost:** 100 Steel Beams, 5 Workforce
- **Produces:** 500 Electricity
- **Consumes:** 1 Fuel Cell per 1 time step

Observatory (LT):

- **Description:** This Building needs to be Build to discover the planets that are neighbours to the planet this was constructed on.
- **Construction Cost:** 20 Steel Beams, 5 Ship Hull Plates, 20 Electricity, 5 Workforce
- **Produces:** None
- **Consumes:** None

Hangar (LT):

- **Description:** This module is for transferring goods towards another planet. Goods can always be dropped on a planet, but can only be ferried to another planet with the hangar or the space station (for larger ships). Cargo Ship and small explorer can land here.
- **Construction Cost:** 50 Steel Beams, 20 Ship Hull Plates, 50 Electricity, 20 Workforce
- **Default Supply:** 1 Small Explorer, 1 Cargo Ship for the first Hangar built
- **Produces:** None
- **Consumes:** None

Space Station (DT):

- **Description:** This is a late game building and used for the larger ships to transfer goods between planets. While the base is constructed on the surface, the space station will use small ships for transferring goods visually between the planet and the station.
- **Construction Cost:** 100 Steel Beam, 100 Hull Plates, 100 Fuel Cells, 100 Food, 30 Advanced Circuits, 300 Electricity, 30 Workforce
- **Produces:** None
- **Consumes:** None

Monument Shipyard (FM):

- **Description:** The monument shipyard as such is not so expensive, but will need mostly workforce and electricity. The Monument ship will be built in space in a few stages and small ships will traverse between the planet and the ship construction visually.
- **Construction Cost:** 100 Steel Beam, 100 Hull Plates, 100 Fuel Cells, 100 Advanced Circuits, 1000 Electricity, 100 Workforce
- **Produces:** Monument Vessel
- **Consumes:** See Monument Vessel



Top row: Cargo Ship; Middle row: Small Explorer; Bottom row: Space Liner

2.2.1.3. Ships

Cargo Ship (LT):

- **Description:** This ship is build for moving goods between planets. If the planet has no hangar, it will drop the goods and travel back to the starting point if in automated route mode. This ship can only travel to the next planet that is a neighbour of the start of the route.
- **Capacity:** 30 of any Resources times 3 Slots (total of 90)
- **Construction Cost:** 12 time steps total construction time. Consumes per 1 time step: 3 Steel Beams, 2 Hull Plates, 1 Fuel Cell
- **Speed:** 3

Small Explorer (LT):

- **Description:** The small explorer is used to create a new base on another planet or just explore the solar system. If another base should be created, this ship needs to be filled with the resources for building a new base and the ship needs to be moved to the new planet.
- **Capacity:** The Resources for a new base. For Base 1-3: 70 Steel Beams and 50 Food. Can not be used as a cargo ship.
- **Construction Cost:** 8 time steps total construction time. Consumes per 1 time step: 2 Steel Beams, 1 Hull Plate, 1 Fuel Cell
- **Speed:** 3

Large Cargo Ship (DT):

- **Description:** This large cargo ship can travel between all planets, no matter the distance they are apart. Also its capacities are greatly increased compared to the normal cargo ship. It also needs the space station to be able to load resources from the planet.
- **Capacity:** 200 of any Resources times 6 Slots (total of 1200)
- **Construction Cost:** 40 time steps total construction time. Consumes per 1 time step: 6 Steel Beams, 3 Hull Plates, 3 Fuel Cells, 1 Advanced Circuit

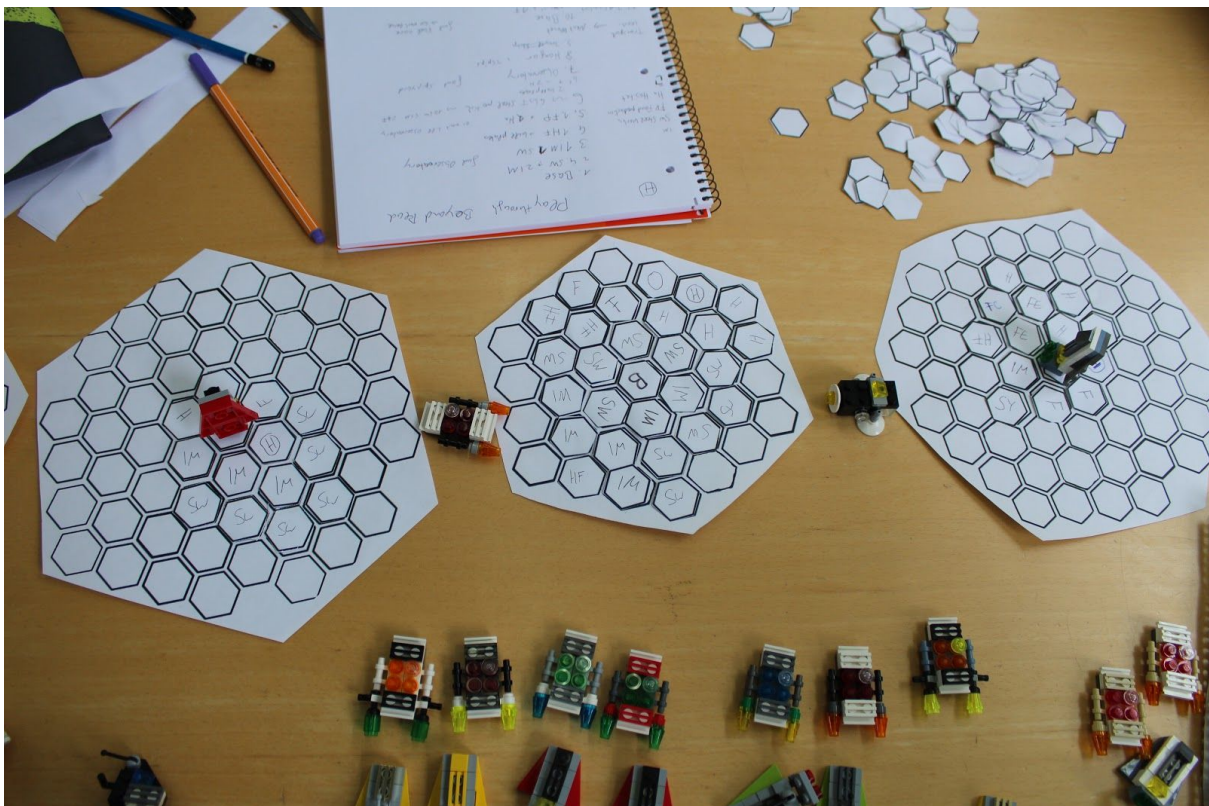
- **Speed:** 2

Space Liner (DT):

- **Description:** The space liner can link the populations from two planets. If this ship is active between two planets their population count is shared.
- **Capacity:** Unlimited
- **Construction Cost:** 40 time steps total construction time. Consumes per 1 time step: 5 Steel Beams, 4 Hull Plates, 3 Fuel Cells, 3 Food, 1 Advanced Circuit
- **Speed:** 2

Monument Vessel (FM):

- **Description:** This is the last thing you will build in the game and will be the most expensive construction. It will feature 3 building stages that have to be completed after the previous one. When this is finished the game is won.
- **Capacity:** Enough to save your race.
- **Construction Cost:** Phase 1: 2000 Steel Beams (every 1 time step. Consumes 50 -> 40 time steps / longer if insufficient supply); Phase 2: 1000 Fuel Cells and 1000 Ship Hull Plates (every 1 time step. Consumes 25 -> 40 time steps / longer if insufficient supply); Phase 3: 1000 Food and 1000 Advanced Component (every 1 time step. Consumes 25 -> 40 time steps / longer if insufficient supply)
- **Speed:** 1



Setup during a round: pen & paper, planet grids, tiles and space ships

2.2.2. Rules

The paper prototype game can be played with two players in a round based manner. For setting up the game, the following parts are needed:

- One similar planet sector grid sheet for each player
- Further planet sector grid sheets for extended play time
- A ton of labelled sector tile cutouts with labels for each of the available buildings
- 5 equal ship models made from Lego per ship type per player, excluding the endgame vessel. This type is only needed once per player.
- Pen and paper for drawing a table of resources for both players

Start of game:

Every player places his/her base tile in the middle of the planet sector grid sheet. Decide which player starts first. A player goes through different phases each turn:

Phases:

- Resource phase: Calculate and collect the amount of resources gained through production on each planet and subtract the ongoing production costs for this turn
- Action phase: Every round a player can choose to do any number of actions listed below.
- Endphase: The player hands over control to the opponent

Actions:

- Build a building on an empty tile in any of the player-owned sectors. The necessary resources according to the type of building have to be paid right away from the player's inventory.
- Destroy a building on a non-empty tile that the current player owns. Only half of its building costs will be refunded and added to the player's inventory.
- Build a new ship in a player-owned shipyard. The ship will be spawned on the shipyard tile after X rounds where X is the amount defined by the ship type. The player has to pay for the ship's construction costs per time step.
- Only one of these per unit:
 - Move a ship to another planet. Ships can travel one planet per turn, unless specified differently, and will be available again for operation the next turn after moving.
 - Load a ship with any amount of resources you have available on one planet or unload the resources carried by a ship.
 - Spawn a new base on a new sector or planet, if an explorer ship is hovering above its center tile while carrying the necessary resources for a new base.

Goal of the game:

The player who is able to build the monument vessel space ship first, wins the game.

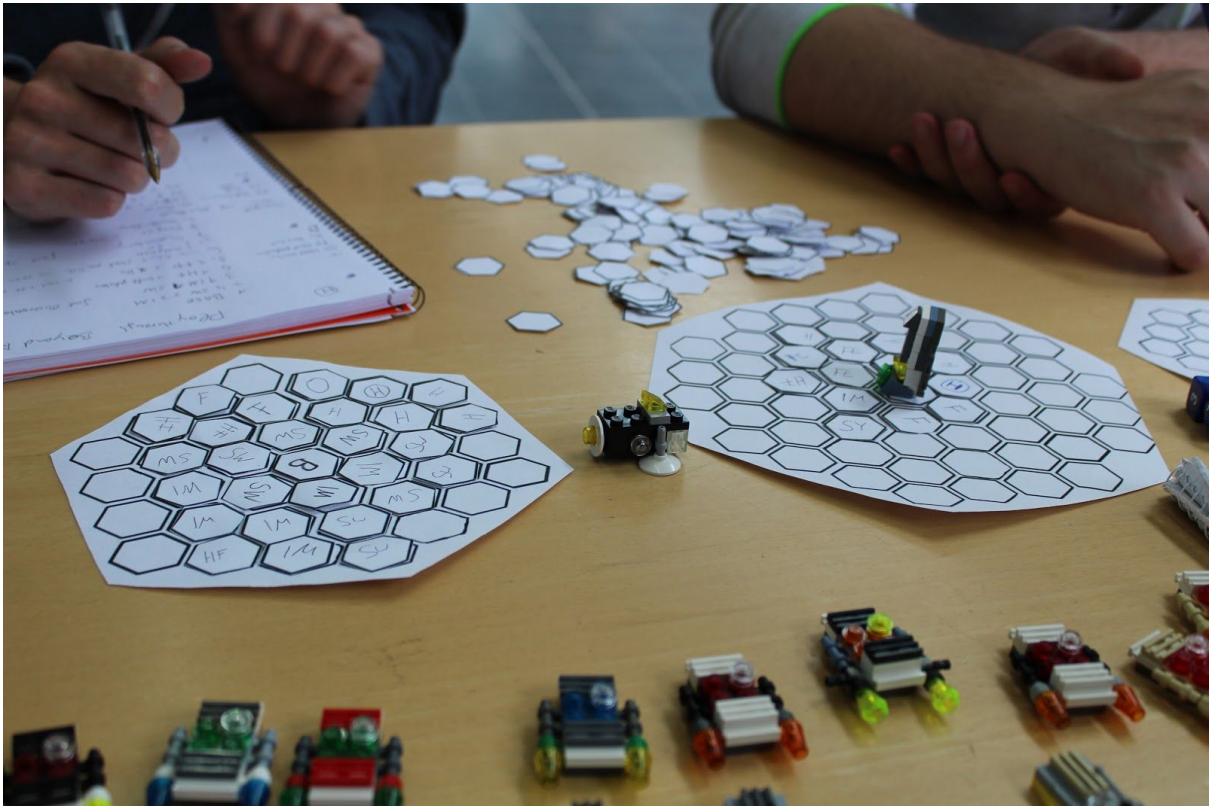


Figure 2.2.3.1: Every round the resources have to be calculated

2.2.3. Gameplay

The gameplay can be defined as a classic round-based strategy board game with the planet sectors being the boards and every time-tick in the game being referred to as a round in the paper prototype. One of the two players is in charge of the computer controlled AI player that

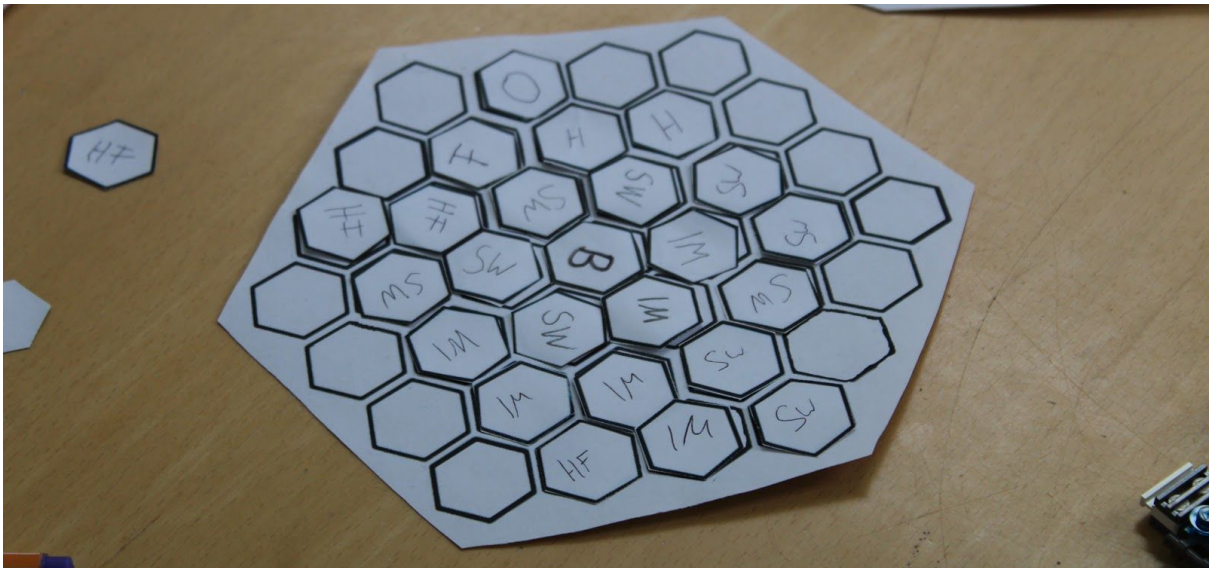


Figure 2.2.3.1: A planet tile after some rounds

will be the opponent in the final game later on. Both players take on the challenge to complete the construction of the monument vessel before the opponent does.

Every round players perform their available actions and try to grow their population and production as fast as possible. Each player first starts with a medium sized planet grid but can later expand to other planets, preferably before the opponent does, in order to choose between the best planets to settle on. As planets and settling space are limited, players have a limited income of resources. Every round they can place buildings by putting paper tiles with the correct label onto any of the free tiles in the sector they are paying for the building (see Figure 2.2.3.1). The paper tiles cannot be moved, but can be removed, which will grant a refund of half the cost. Logic and strategic experience as well as planning resources ahead can help a player to gain a slight advantage over the other.

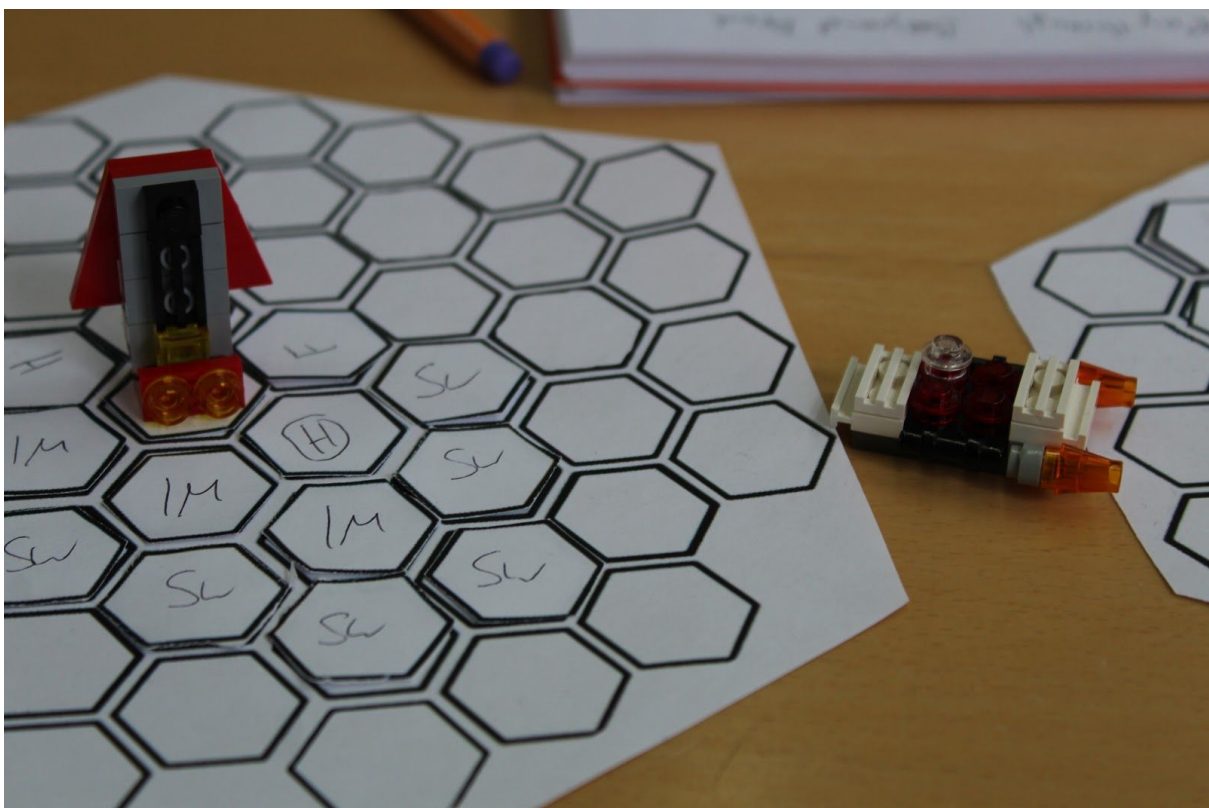


Figure 2.2.3.2: A cargo ship is delivering resources to a newly built base

As players move their spaceships during late game from planet to planet, a steady resource flow is necessary to provide for the construction of the monument vessel. Players want to settle on planets whose resources are necessary for the big ship and move those to, most likely the home planet, where they built the monument shipyard. However, first it is necessary to provide the newly founded sector base with resources itself. Cargo ships can take on the job and pick up resources on the home planet, transfer them to the other planet in the next round, and drop it finally in the third round (see Figure 2.2.3.1). At this point, intelligent planning from the beginning will pay off. For the paper prototype a resource and production table was necessary to handle the high amounts of calculation and as an orientation for players of what to build next during the late game (see Figure 2.2.3.3).

	Amount	Steel Beam	Food	Hull Plate	Fuel Cell	Advanced	Steel Beam	Iron Ore	Food	Population	Electricity	Hull Plate	Fuel	Fuel Cell	Unobtainium	Refined	Advanced
Base (1st)	1	-40	0	0	0	0	0	0	0	20	200	0	0	0	0	0	0
Iron mine	46	276	0	0	0	0	0	46	0	-138	-46	0	0	0	0	0	0
steelworks	48	144	0	0	0	0	24	-24	0	-48	-144	0	0	0	0	0	0
Food	22	110	0	0	0	0	0	0	22	-22	-22	0	0	0	0	0	0
Population	34	340	0	0	0	0	0	0	-17	1020	-68	0	0	0	0	0	0
Ship Hull	20	200	0	0	0	0	-10	-10	0	-100	-100	10	0	0	0	0	0
Observatory	4	80	0	20	0	0	0	0	0	-20	-80	0	0	0	0	0	0
Hanger	4	200	0	80	0	0	0	0	0	-80	-200	0	0	0	0	0	0
Fuel Extract	24	480	0	0	0	0	0	0	0	-120	-480	0	12	0	0	0	0
Fuel Cell	12	360	0	180	0	0	0	-12	0	-120	-360	0	-12	12	0	0	0
Base (1-3)	12	360	600	0	0	0	0	0	0	240	2400	0	0	0	0	0	0
Shipyard	1	40	0	10	0	0	0	0	0	-20	-50	0	0	0	0	0	0
Powerplant	7	700	0	0	0	0	0	0	0	0	3500	0	0	-7	0	0	0
Space Station	4	400	400	400	400	120	0	0	0	-120	-1200	0	0	0	0	0	0
Unobtainium	4	240	0	160	0	0	0	0	0	-40	-400	0	0	0	-4	0	0
Refined	8	600	0	320	0	0	0	0	0	-80	-800	0	0	0	-4	4	0
Advanced	16	1600	0	800	320	0	-4	0	0	-320	-1600	-4	0	0	-4	4	4
Monument Yard	1	100	0	100	100	100	0	0	0	-100	-1000	0	0	0	0	0	0
Base (4-6)	3	330	300	150	150	0	0	0	0	60	600	0	0	0	0	0	0
Base (9-12)	1	250	200	100	100	30	0	0	0	20	200	0	0	0	0	0	0
Cargo Ship	8	256	0	144	72	0	0	0	0	0	0	0	0	0	0	0	0
Small Explorer	2	32	0	16	16	0	0	0	0	0	0	0	0	0	0	0	0
Large Cargo	4	960	0	480	480	160	0	0	0	0	0	0	0	0	0	0	0
Space Liner	4	800	480	640	480	160	0	0	0	0	0	0	0	0	0	0	0
Monument Ship	1	2000	1000	1000	1000	1000	0	0	0	0	0	0	0	0	0	0	0
	289	10778	2980	4600	3118	1570	10	0	5	32	350	6	0	5	0	0	4

Figure 2.2.3.3: Production and Resource table for easier calculation

2.3. Experience

Right from the beginning we knew, that the experience turned out to be less engaging than the real time strategy game that we have in mind. That might be resulting from the missing satisfaction evoked by futuristic buildings and flying ships. Instead, the paper prototype offers increasing numbers of the players resource tables on sheets and plain tiles to build on the planet sector. Additionally we didn't consider how to distribute information, for instance, construction costs, so that the game stretched long by looking into our documents. In the video game, the User Interface will assist in that regard, so that the game is appealing and a good game flow emerges. With our prototype we wanted to be as close as possible to our desired target. That is why engaging combat with some action between the contrahents was discarded for the paper prototype.

The core goal of our paper prototype resided in the exploration of the fun in playing a strategic building game, where the excitement and motivation comes from optimizing the production and planning the next expansion of the base. While playtesting, we definitely had interesting and heated discussion about what the next move of our player should be. So the core gameplay loop and hook of our game was fun and motivating.

By playtesting playthroughs we discovered that variety in possible production sequences allows the player to choose more, and therefore enables more strategic decisions. It helped us to identify our strength, in this case planing your planet bases and managing your resources efficiently. That's why our focus is shifting now towards the base building. During the game, we discussed how to increase the quality of building choices rather than adding new game features. For instance we evaluated, while placing the buildings on our field, that adjacency bonuses would make the placement of tiles more meaningful. Our other findings are listed in 2.4.

2.4. Learnings from Prototype

We already discussed the focus towards building that has emerged from playtesting in our previous sections, now we reflect on some of our learnings.

First of all, writing down numbers on paper about how many resources you have isn't fun. It would appear that our game should do any tedious task like that on its own.

The Building types we designed for our desired target are to few and create an almost streamlined experience, which we will tackle by increasing the amount of different building types in the High Target. With this, we will add as many new buildings, in the polishing process, as allowed by the time constraint, increasing the complexity of our game. Another effective way to increase the complexity of building the base we introduced in 2.3. Adjacency bonuses that is. For instance, constructing food producing tiles next to each other could increase the overall means of production and would require the player to not just choose the location of building randomly.

Even though the player races against other players, our game does not offer much interaction between each other, therefore we want to explore how to increase that aspect. This could be done with trading between the players and possible third party NPCs.

We also discussed that the game should punish deconstructing buildings, so that there is a stronger emphasis on planning your sectors correctly.

2.5. Game Revisions

To close the results of playtesting our prototype, we would like to summarize the revisions we're playing on our game.

First and most heavy weighted decision for us to remove the combat from high target to extras, since we figured that making it work would require additional time which we don't have. Our core gameplay should be as well polished as possible, combat simply drops in our priority list.

Second, with combat removed, we need to experiment with other ideas how to increase interactions between the players. For instance we had trading in mind, yet it is something that won't be pursued before HT.

We limit resources to specific planets as an incentive to conquer these planets, which are rather difficult to reach. In addition we plan the necessary to transport the goods between planets to increase the complexity of resource management and to give build ships a purpose besides combat.

3. Interim Report

Looking back at the past three weeks of development, we can say that we are close to being on track, based on our project timeline. We finished a very large portion of our low target task, all of our functional minimum tasks, and a few desirable target tasks. Also, we can already play our full game loop, complete with main menu and victory screen. However, one large aspect that we expected to happen a lot later in development was optimization. Here we already had to invest a lot of time during our interims milestone development process, because otherwise we would have only had 1 FPS when fully settling a large planet. Moreover, our game is a lot more CPU bound, than GPU bound, which also already forces us to optimize our code. Part of these problems are based on the fact that we built all our models from small model prefabs, which caused us to have a huge vertex count and a lot of batches that the CPU had to process.

3.1. Gameplay

3.1.1. Core Systems

For the core gameplay systems, almost all relevant systems are implemented in the current status of the game. The only aspects that are missing, are the discovery mechanic, resources only available on specific planets, fog of war and predefined star sectors on one planet for all players. These were moved to the next milestone, due to the fact that more time had to be put into code optimization already.

The core systems that are implemented, are data loading from files (for easy balancing later on), planet resource system, placing buildings on planets, building ships, victory condition, ship pathfinding and a resource transfer system. The first five task were fairly straight forward and had no big problems during the implementation process.

For the ship pathfinding system, we had to implement the start of a ship from the planet surface to the orbit, from there to space and back to the orbit of a planet. especially for transforms around the planet, we had many difficulties, as different rotations have to be made around the planet and ships. These operations are currently all functional, but still display the occasional bug, where a ship turns into the wrong direction. These and path smoothing have to be tackled in the next milestone.

Another task that proved to take longer than anticipated, was the resource transfer system between planets. This system is entwined deeply into all the other systems and has many special cases for trade routes, explorer ships and cargo ships. This resulted in a lot more code that needed to be written, just for those systems, including the UI for the player.

3.1.2. Interaction Systems

In Beyond Reach there will be two different views that a player can navigate in between. The first view is an overview of the planets and shows the whole solar system from an orthogonal perspective above the sun. This view can be used for focussing on the greater image and travelling from one planet to the other. The second view is a close up perspective of the

selected planet and puts the camera right up front. In this view players can look at the planet in detail, perform building actions and manage the life on this planet.



(Left) In the space view players navigate forwards, backwards, left and right either by w-a-s-d keys or by pressing down the mouse wheel and moving the mouse in preferred directions. The player can also zoom in or out with the mouse wheel or -/+ buttons in order to get a closer or broader view. A double click on any planet will make the camera transition between space view and planet view and slowly approach the selected planet.

(Right) In the planet view, the camera can rotate around the planet either upwards/downwards or left and right as if it was moving on another sphere further away from the planet. Again, w-a-s-d keys or the mouse pinching with the scrolling wheel is used as input. Zooming in and out is equally implemented to the space view. However, in the planet view players can select hex fields by clicking on it and found new bases, construct buildings or send spaceships on missions. The interaction is very straight forward with few buttons and a clear design pattern.

The camera system and the two views are already working very smoothly. There is few fine-tuning to do, but overall I am very pleased with the result. The only minor hurdle is the iteration of playtesting and adjusting of values in order to make the camera smooth and feeling comfortable for the player in every situation. With the new addition of fog of war in the next milestone, there will be a medium size change to the current system and especially the space view will have to be adjusted to give the player only the possibility of seeing the parts the he/she is supposed to see.

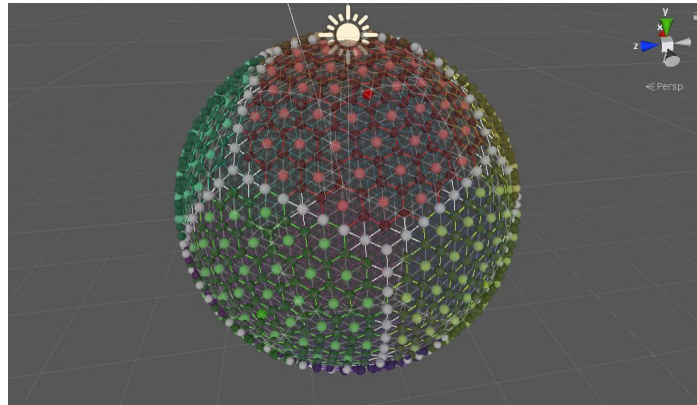
3.1.3. General Gameplay

Besides the core and interaction systems, we also implemented all relevant gameplay, that we listed for our functional minimum and low target. To these belong the following resources, buildings and ships: Iron Ore, Steel Beams, Fuel, Fuel Cell, and Ship Hull Plates; Base, Iron Ore Mine, Steelworks, Monument Shipyard, Fuel Extraction Plant, Fuel Cell Factory, Ship Hull Factory, Ship Building Yard, Observatory and Hangar; Monument Vessel, Cargo Ship and Small Explorer.

3.2. Planet Generation

The planet generator starts with an ico sphere, a sphere like object with 12 vertices and 20 triangles. By subdividing the object we can create more hex fields. To create the actual hex

field later on, we have to subdivide the surfaces as a last step. A couple of additional steps are performed during the generation stage, like the hierarchy setup of the planet, mesh merging by material for performance optimisations and the creation of random biome values for the field (which allows us to create different kinds of planets later on).



Debug view of a planet, it's sectors and fields during generation

3.3. AI

With growing complexity of your project and therefore increasing game logic, the AI module, that simulates enemy players, adapts consistently. For the functional minimum the AI is required to build a sequence of buildings that produce the resources to build the shipyard, that is required to build the arc vessel to win the game. Now, with increasing building tiles, ships, multiple planets and trade routes, the AI integrated these task into a set of actions. To generalize these actions into sets or patterns, the AI evolved from completing a queue of buildings to arrangeable actions called Wildcards.

Wildcards tailor, for instance, to inhabit a new planet, to build a combination of iron ore mines and steel factories to boost steel resources. The idea is that it is possible to design the AI's behaviour priority, but still give the AI the ability to insert new Wildcards based on reactive decisions. For instance, if the amount of free tiles is zero, the AI can insert the action to inhabit a new sector on the same planet.

During the development of the AI the most challenging thing was to work in advance, without knowing how the project internal Game logic and structure will look like. During the development, access of planets' tiles did change. In addition, the AI development was stalled to some degree,



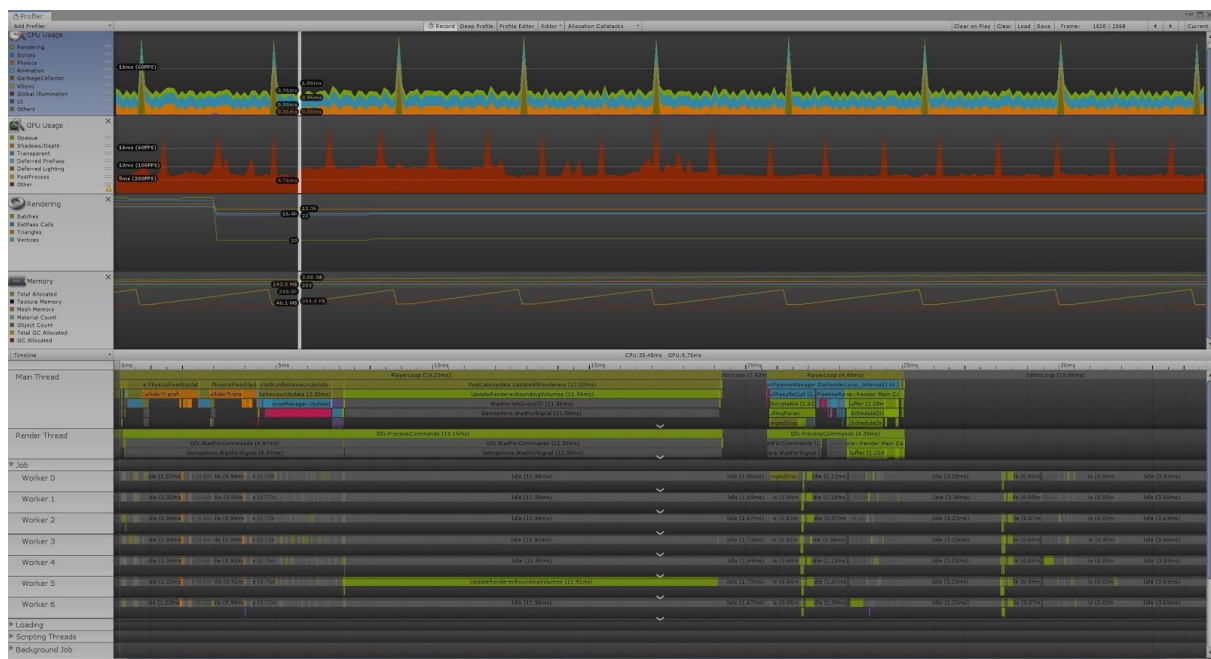
when certain features were not part of the game yet. For instance, settling to a new planet was something we implemented late, therefore, it was necessary to cheat the AI with resources on the new planet in order to test the management of multiple planets. The biggest revision in the AI module has been the Wildcard system, that replaced the State-based AI I had in mind. That was simply because RTS AI tend to have state machines

that separate tasks as manage combat and manage building. It is possible that states come back in later development, for now a linear sequence of states are sufficient for the AI. The image shows the AI sending its first cargo ship to a different planet.

3.4. Rendering

When it comes to rendering, the bulk of the work went into the setup of the pipeline, as well as optimisation. Unity's default render pipeline is built to support as many rendering features as possible and therefore comes with a massive overhead. Adjusting it to the needs of the project yields massive performance improvements.

Optimisation is a key area on this project. We have an extremely large object and vertex count across our multiple planets. We can reduce a lot of the load with material based merging of the meshes in the editor. In combination with our LOD system and the tools we've written for it we can massively reduce the vertex count in our scene. The lowest LOD is a billboarding system, which allows us to keep the details of our buildings, even when watching the whole planet from far away. A custom batcher, in combination with GPU instancing further decreases the frame time. However, as of now we are heavily CPU bottlenecked, with the GPU running at 30-40% depending on the device.



At the moment, our game has to wait for the render thread to submit all draw calls to the GPU.

Some shaders and effects are already implemented, including a PBR building shader, one simple Shader for the atmosphere, a placeholder shader for the planet to support the merged meshes and a building effect for newly built objects.

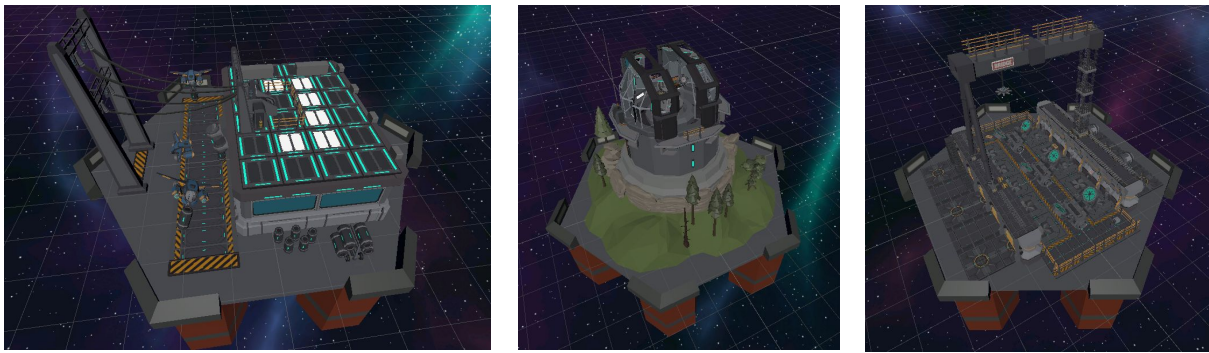


3.5. Models

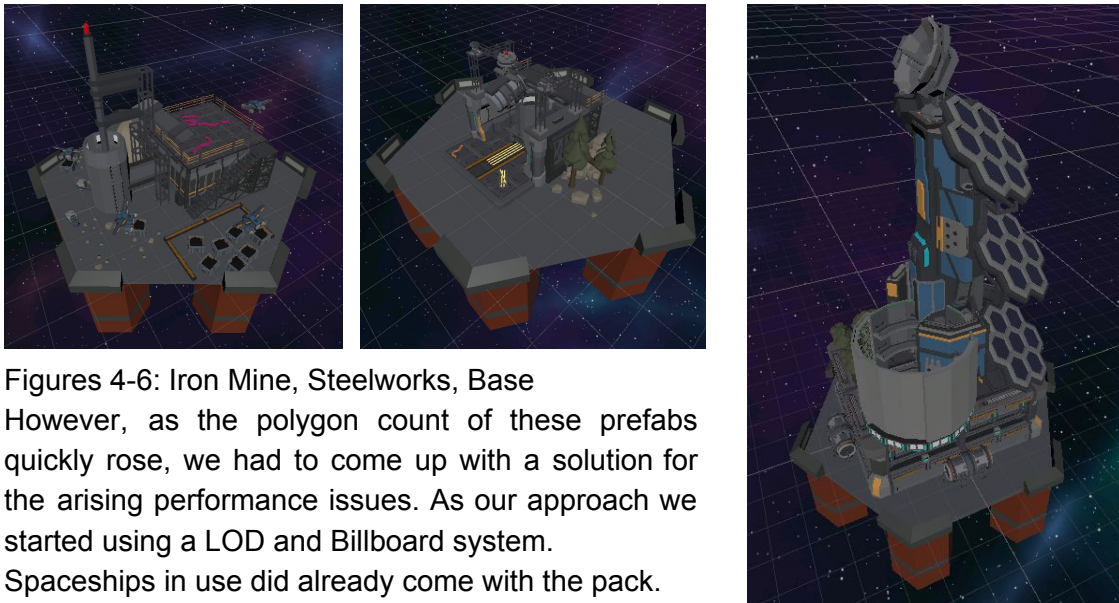
The models we use for our buildings and spaceships are part of two big model sets called “POLYGON - Nature Pack” and “POLYGON - Sci-Fi Pack”. Both sets contain many polygon-style 3D-models and textures that go well along with the hexfield design of planets. With these model prefabs we create buildings and spaceships by fitting the parts together and designing models to our individual needs directly in unity. Finished models will be saved as a new prefab that can be instantiated in the game, e.g. a building prefab can be placed on a hexfield of a planet.

3.5.1. Prefabs

As prefabs we have currently built all necessary low target buildings with more to come soon. In detail, those are the following:



Figures 1-3: Fuel Cell Factory, Observatory, Shipyard (from left to right)



Figures 4-6: Iron Mine, Steelworks, Base

However, as the polygon count of these prefabs quickly rose, we had to come up with a solution for the arising performance issues. As our approach we started using a LOD and Billboard system.

Spaceships in use did already come with the pack.

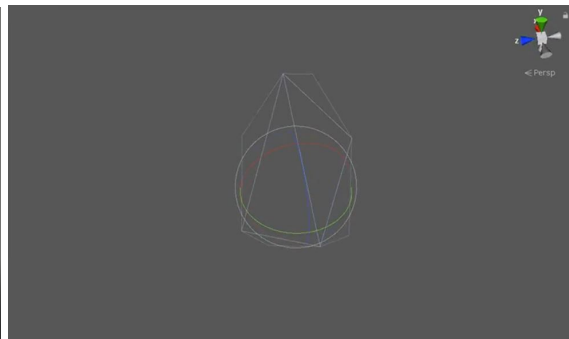
3.5.2. LODs

The LOD system currently consists of 4 different 3D LOD levels, as well as the billboard system. There is a crossfading system set up for them, however the 3D LODs are fairly traditional with no special implementations.

A complex billboard system allows us to render a large amount of objects from far away. We create billboards in the editor, i.e. we place cameras on a hemisphere around the object and bake albedo, alpha, metallic, roughness, depth and normals into texture atlases. During runtime, a simple planar object with 4-6 triangles is set up to always face the camera. A custom shader blends between the textures taken from different angles and uses the baked texture values to render the object as if it is a complex 3D geometry. This allows us to add proper shadows and other rendering features to the billboards with almost no difference in the visual quality.



Billboard from far away



Wireframe view of the billboard



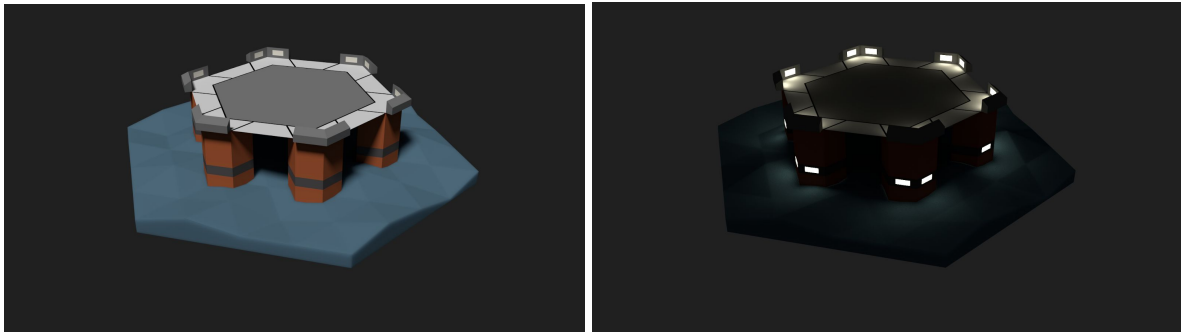
Close-up of a low res billboard without shadows...



... and with realtime shadows

3.5.3. Platform Model

The platform is one of our custom models for the game, which is designed to adjust the building to the terrain it's built on. We have early tests showing the combination of emission, real-time GI and our platform models.



Platform rendered during day and night

3.6. UI

3.6.1. Functional UI

The functional UI was implemented for testing purposes, until the fancy UI is ready to replace it. This UI just incorporates the needed functionality, so a human player can actually play our game. This UI was done for all ten functional minimum and low target buildings, as well as the three functional minimum and low target ships. The UI for the ships took longer than expected to implement, as they are mostly more complex, compared to the building UIs, as they require a lot more functionality for trading and traveling to new planets. Another complex UI, is the trade route menu. Here the player can create, edit and delete his trade routes. We will share no screenshots of this UI, as it does not look very appealing and the functionality will be shown as soon as the fancy UI for each screen has been finished.

3.6.2. Fancy UI

When it comes to UI thus far we mostly implemented placeholder UI. However, we already designed Icons and other UI elements for the game. A masked blur shader for our custom render pipeline was also implemented to help us improve the visual quality of the UI in the future.



Building Icons

Resource Icons

4. Alpha Release

5. Playtesting

6. Public Presentation and Conclusion