

Terrafarm

1. Formal Game Proposal

1.1 Game Description

1.2 Story

1.3 World Layout

1.4 Resource Types

1.5 Building Ideas

1.6 Climate Effects Ideas

1.7 Concept Art

1.8 Technical Achievement

1.9 Big Idea Bullseye

1.10 Development Schedule

1.10.1 Layered Development Schedule

1.10.2 Timeline

1.10.3 Task Outline

1.11 Assessment

2. Prototype

2.1 Prototype Description

2.2 Prototype Insight

3. Interim Demo

3.1 World Generation

3.2 Climate Effects

3.3 Resources

3.4 Buildings

3.5 UI

[3.6 Audio](#)

[3.7 Issues](#)

[3.7.1 Models](#)

[3.7.2 Turn Processing Order](#)

[4. Alpha Release](#)

[4.1 World Generation](#)

[4.1 High Scores](#)

[4.2 Fancier Graphics / Animated Assets](#)

[4.3 Quality of Life Improvements](#)

[4.4 More Climate Effects](#)

[4.5 More Building Types](#)

[4.6 Background Music / Sound Effects](#)

[4.6 Issues](#)

[5. Playtesting](#)

[5.1 Playtesting Sessions](#)

[5.2 First Gameplay Experiences](#)

[5.3 General Gameplay and Usability](#)

[5.4 Balancing](#)

[5.5 Changes](#)

[6. Conclusion](#)

[6.1 Final Results](#)

[6.2 Experience](#)

[6.3 Personal Impressions](#)

1. Formal Game Proposal

1.1 Game Description

Terrafarm is a turn-based strategy game set on a previously uninhabited planet, in which the player establishes a foothold to quickly exploit the planet before the collapse of its ecosystem. The player is able to erect several types of buildings to acquire energy and mass, which can be used to expand. While the player's empire grows and might eventually span the entire planet, they also need to deal with climate catastrophes and other hazards that are caused by careless exploitation of the planet's resources.

The player needs to fight against the threats caused by self-induced climate change and acquire as much money as possible before the planet's ecosystem collapses – leading to the inevitable demise of the player's empire. As the world is generated randomly, each game offers a slightly different experience.

1.2 Story

The player is part of an alien species which is rapidly expanding their reach to ultimately colonize the entire galaxy. To reach this goal, they have adapted their biology to quickly transform themselves into industrial buildings. The player settles on an unmarred planet, rich with resources, with the ultimate goal of generating as many resources as possible while interacting with the planet's climate and ecosystem.

1.3 World Layout

The shape of the planet is a truncated icosahedron. Each hexagon (or pentagon) serves as a spot where the player can construct a building. The tile itself has a terrain type, which can constrain what type of building can be constructed, or influence its performance in positive or negative ways.

1.4 Resource Types

The game features two basic types of resources: Energy and Mass.

Some buildings generate energy, which is required to sustain the operation of buildings. Other buildings generate mass, which is used to create new buildings. Both can be exchanged for the other at the galactical market (at a reduced rate), and at the end of the game are sold for credits to calculate your score.

1.5 Building Ideas

The player can erect various power plants to generate energy and mass. However, each building has advantages and drawbacks. Some may generate more power in exchange for more pollution, while others may pose different dangers.

- **Combustion Power Plant:** This power plant burns biomass to generate a fair amount of energy, but also quickly pollutes the environment, possibly reducing the player's turn limit due to climate events.
- **Fission Reactor:** The fission reactor splits atoms to generate vast amounts of energy and pollution. Additionally, it carries the risk of exploding, damaging the player's structures and leaving behind a large financial gap. It is a high-risk, high-reward building!
- **Wind Turbines:** Wind turbines generate a low amount of pollution, but also a limited amount of energy.
- **Mass Extractor:** Pulls biomass from the planet's terrain. Uses large amounts of energy.

1.6 Climate Effects Ideas

Pollution from buildings will begin to heat up the atmosphere, ultimately making the planet uninhabitable due to high temperatures.

Additionally, during gameplay, other random events may happen depending on the player's pollution:

- Lightning Strike: May hit player's building causing it to take large damage
- Earthquake: Deals low damage over a wide area
- Ion Storm: Lowers the production of a building for the next X turns

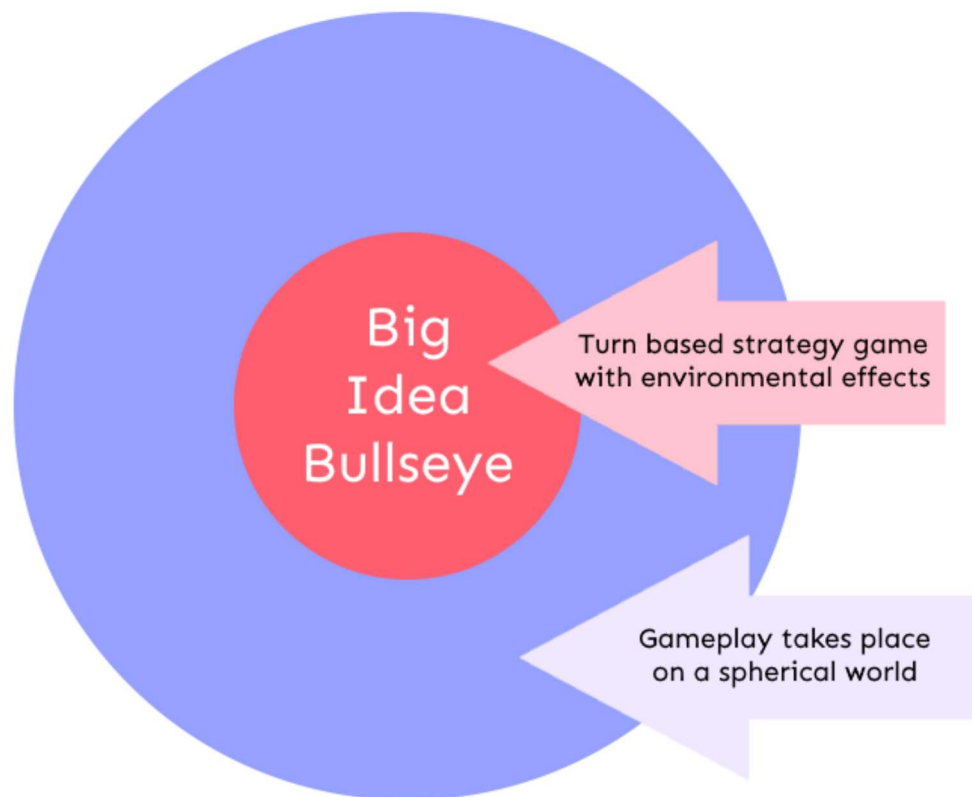
1.7 Concept Art



1.8 Technical Achievement

Our technical achievement will come from the fact that the game takes place on a truncated icosahedron, so extra precautions need to be taken in areas such as graphics, camera control and coordinate systems. Additionally, we would like to incorporate procedural elements into the world layout such as mountains and water to increase replayability.

1.9 Big Idea Bullseye



1.10 Development Schedule

1.10.1 Layered Development Schedule

Functional Minimum

- World Generation
- Input System
- Camera Controls
- Selectable Starting Location
- Buildable Buildings
- Collectable Resources

Low Target

- World features (Water, Mountains)
- Climate effect depending on player-generated pollution
- Simple UI

Desirable Target

- Buildings only placeable on certain terrain
- More building types
- More climate effects
- High scores
- Background Music
- Sound effects

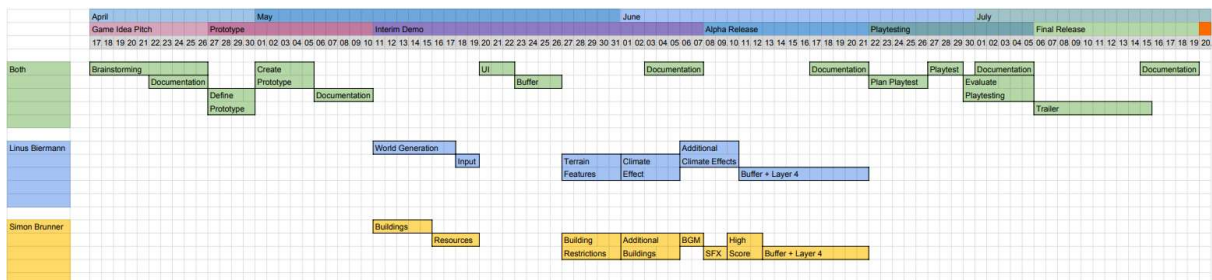
High Target

- Fancier graphics
- Save / Load
- Animated Assets
- More resource types
- Upgradable buildings
- Terrain modification
- More world features
- Implement Story
- World Events

Extras

- Asymmetric Multiplayer
- Online leaderboard
- Mobile platform

1.10.2 Timeline



1.10.3 Task Outline

Task	Description	Member	Planned	Actual
World Generation	<ul style="list-style-type: none"> - generate truncated icosahedron with variable size - layout data to easily access buildings, tiles, etc - foundation for world features - selectable starting location - flat shading 	Linus	15	
Input System	<ul style="list-style-type: none"> - selectable tiles - camera controls 	Linus	4	
Buildings	<ul style="list-style-type: none"> - integrate 4+ buildings - resource generation - pollution / effect on the environment 	Simon	15	
Resources	<ul style="list-style-type: none"> - energy - mass - credits 	Simon	5	
UI	- user interface for buildings, construction, resources and various other information panes	Both	10	
Core	- integrate all systems to create a playable prototype	Both	10	
Terrain Features	<ul style="list-style-type: none"> - mountains, water, ... - make data easily accessible for building perks / detriments / restrictions 	Linus	5	
Building	-make buildings placeable on certain	Simon	10	

Restrictions	tiles -impact of the environment (mountains, water) on resource generation			
Climate Effect	- layout foundation different types of effects with variable sizes, effects, ... - create visuals for effects	Linus	10	
Additional Content	- more buildings - more climate effects - extra resource types? - upgradable buildings - world events - terrain modification - add deeper story integration - asymmetric multiplayer - online leaderboards - mobile platform?	Both	∞	

1.11 Assessment

We hope to appeal to a wide audience through the combination of simple graphics, strategic elements and slower gameplay. Players will have time to think about where they want to place buildings, and how those buildings will interact with the resources they have, the terrain around them and their remaining time. However, since we have only half the members at a team of two, we must be cautious with our planning and make sure the scope of the game does not get too large.

Our plan is as follows: Work on the core features (buildings, resources, world generation, climate effects) and finish these as quickly as possible to establish a good baseline, then work on additional content to hopefully create an engaging experience.

2. Prototype

A turn-based strategy game like this one is perfectly suited for a physical paper prototype, as there are no reaction-based action elements involved. In theory, the entire game could be played on paper, without the need for a computer.

2.1 Prototype Description

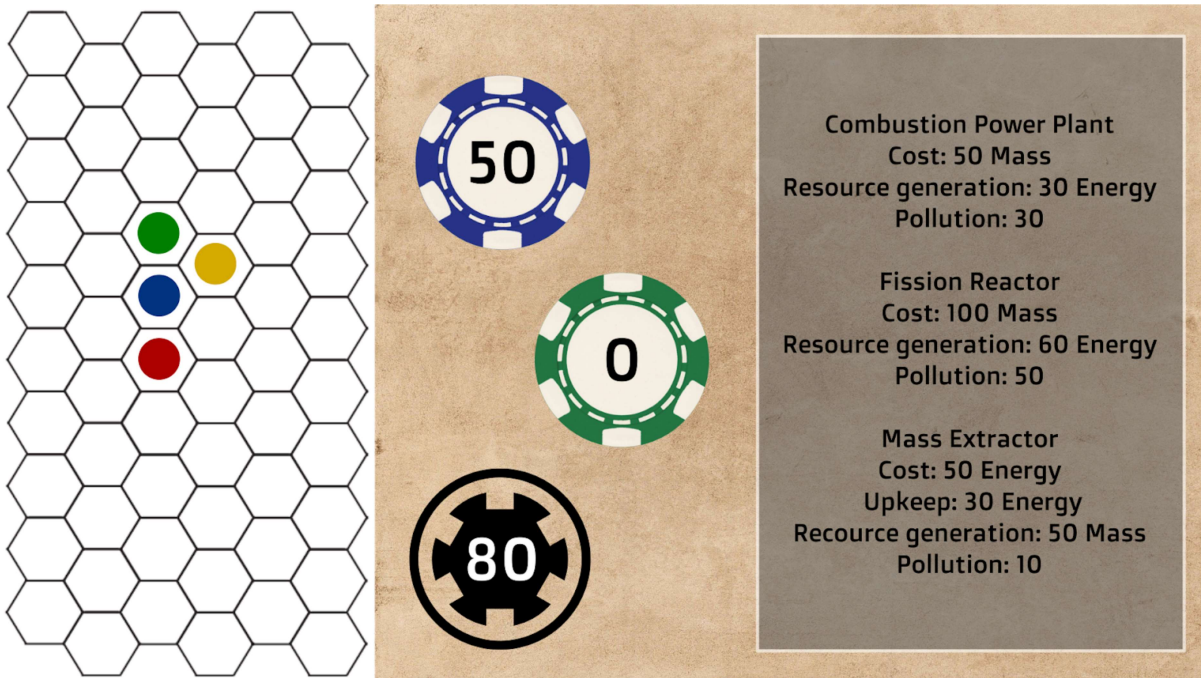
First, we printed out the hexagonal grid from our game. We used a flat piece of paper instead of a spherical object, as the underlying shape of the grid structure has only very little impact on the gameplay. For our buildings, we used pieces from other board games such as small figurines and dice. To keep track of resources, money, and pollution we used poker chips.



Then we set up some basic rules that the player needs to follow in order to play the game. Generally the game runs in two phases. The first phase resembles the gameplay of the actual video game: The player uses his resources to construct new buildings or trade for money.

In the second phase, the player needs to calculate by hand what will be calculated automatically in the finished game:

- Pay the energy upkeep for each building
- All buildings generate resources and pollution
- The Player rolls dice to see if any random events happen
- If the pollution exceeds the pollution limit, end the game. Otherwise, return to phase 1



2.2 Prototype Insight

The goal of the prototype is to answer a few core questions and hopefully discover early flaws in the core game design to save time later on.

Is the game fun? Playing the game turn for turn and watching your empire and resources grow is already a rewarding experience, and we think with a few tweaks it can definitely grow into something that is engaging and fun.

There were however a few minor issues we ran into when prototyping our early concept.

One observation regards the exchange of energy for money. In the game, the player can trade energy for money – and gaining as much money as possible is the ultimate goal of the game, as it is based on a highscore system. While playing the prototype we realized that there is no real reason to make this exchange earlier than the last round of the game. This leads to a 1:1 relation between energy and money. To address this issue, we plan to add interest to the players current credits on every round. This

incentivises the player to exchange a part of his energy for credits early for a larger return on investment as the rounds go by, introducing an additional strategic element.

Another important observation that we made regards pollution and the pollution limit. In our prototype, every building generates pollution. As soon as the pollution exceeds the pollution limit, the game ends. The method to decide how many turns this takes is a very important part of the design, as it will decide in part how long a gameplay session is. By introducing interest on credits, we create the possibility of the ideal strategy being a minimal base while selling energy each turn to maximize profit via interest. We want to incentivise the player to construct more buildings, instead of maximizing the profit by playing as many turns as possible while earning money purely by exploiting the interest mechanic and doing nothing each turn. We plan to address this by looking into other pollution to turn ratios, iterating on building values, as well as playing around with additional potential "win" condition that give the player a credit bonus. For now it is something to keep in mind while iterating.

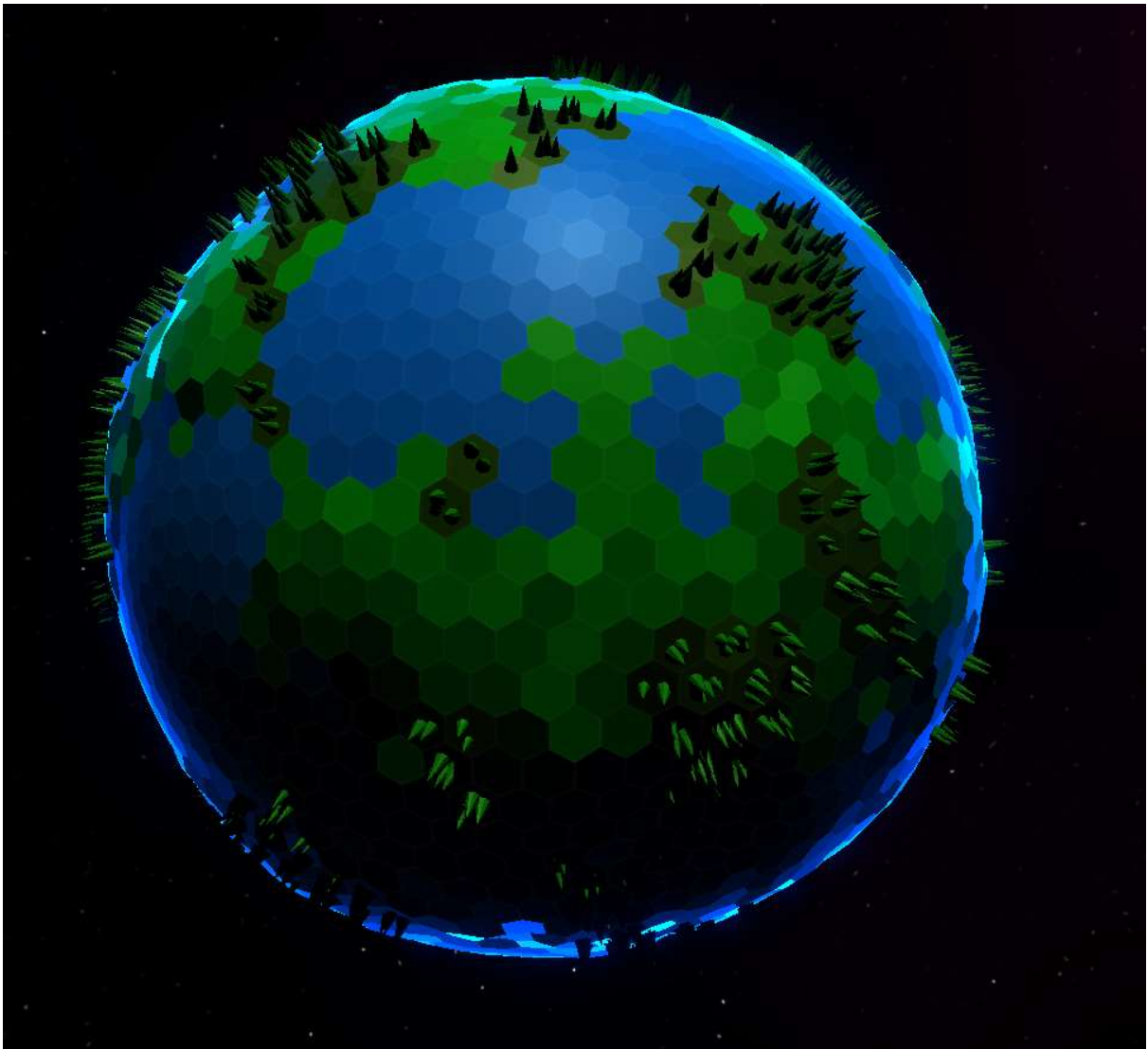
One general observation that also was mentioned in the reviews on our formal game proposal is that we should focus more on strategy and less on RNG-heavy elements. To achieve this, we plan to tune climate events to be more predictable, such as rising water levels or spreading fires. This should make them feel less unforgiving and also add more dynamic gameplay as players can react to the events and plan accordingly. Additionally, we plan to expand on the world generation aspects to create more strategic opportunities. The player must more carefully consider his starting location, how much room he has to build, as well as which tiles are nearby. We also plan to add more building-building interaction, such as buildings that modify the surrounding buildings in some way such as increasing resource generation or reducing pollution production. This way, the player will more carefully have to plan the layout of his buildings, rather than just focusing on the raw statistics provided by each building. This will hopefully lead to more interesting and strategic decision making during gameplay.

During the prototype we did not overly concern ourselves with balancing as we think this is generally easier to do in an alpha development stage anyways. There will be numerous features, like the terrain as well as additional buildings that have a large impact on the balance and can't easily be simulated as a prototype. We plan to adjust the separate values for buildings and terrain iteratively once we are in a position to test the game, as this is rather difficult and tedious with a pure paper prototype.

3. Interim Demo

Good progress has been made, and we are currently roughly halfway through our layer 3 features.

3.1 World Generation



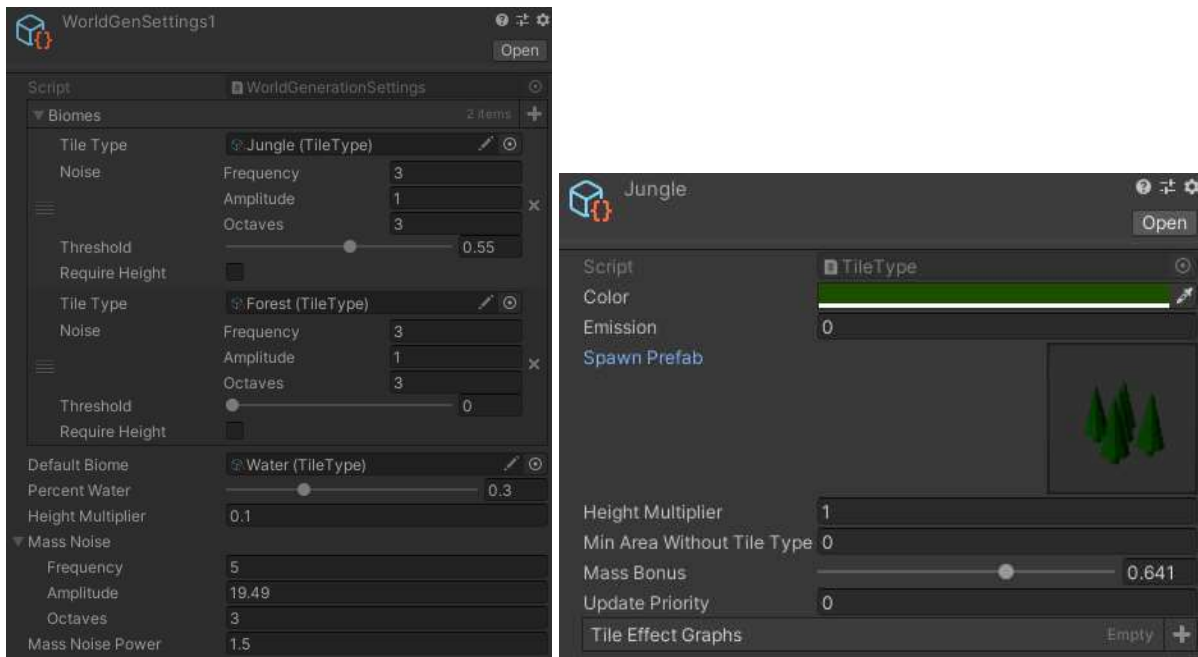
The world's shape is generated by iteratively truncating an icosahedron. Due to the algorithm, we are somewhat restricted in choosing our world size and have settled on 2432 hexagons as a fitting size.

Depending on the seed of the world, a template of world layouts is chosen. These define the biome generation parameters and thereby make it possible to have varied worlds by creating many of these layouts.

Once a layout is chosen, biomes are created using simplex noise and a few simple rules. Each tile type is defined as a scriptable object making it easy to create

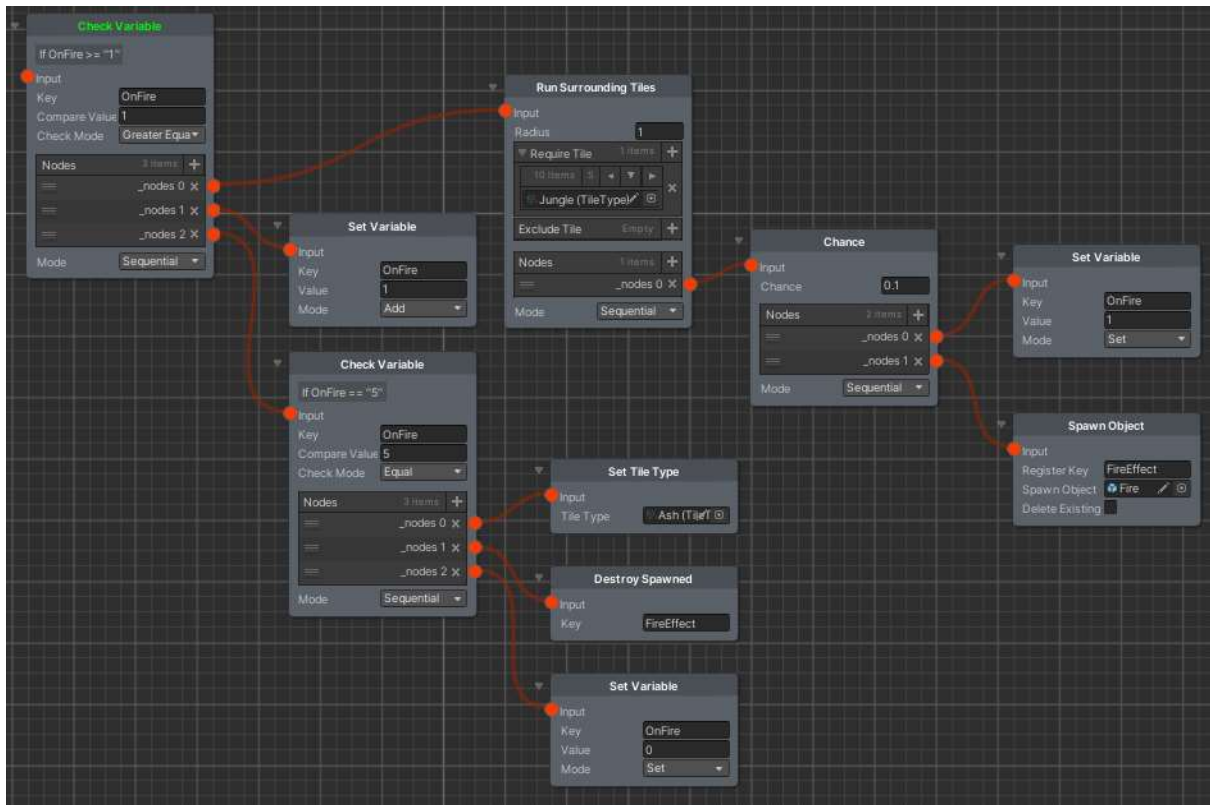
additional tiles. These scriptable objects also carry additional information, such as which object to spawn on it to make it look nicer. Since we have many tiles and potentially even more objects, the tile objects are handled using Unity ECS.

Additionally, each tile carries a "mass value", which gives a bonus to mass extractors. This intends to make base planning more interesting, and that climate events can change the amount of mass a tile carries. To make it easier to plan their base, the player can visualize these mass values at any time.



3.2 Climate Effects

Since climate effects can vary greatly, they are implemented using a graph system. Each graph node is executed sequentially (although concurrent execution is also possible), giving time to pan the camera or show effects. Once a graph for a tile has executed, the next tile will execute its graphs. This gives flexibility and a high-level overview to designing new climate effects. New nodes can quickly be added to enable even more complex effects.



Example graph for the spreading fire climate effect

3.3 Resources

Energy, Mass and Pollution are internally all treated as resources (even though pollution probably isn't a resource the player wants to obtain), and it is very easy to add more types of resources to the game. For example, money will likely be added as resource type at a later stage of development, once the implementation of the high score feature takes off.

3.4 Buildings

The buildings are processed one by one after each turn. An update priority ensures that certain buildings are processed first. For example, mass extractors are processed rather early, as the player needs to pay upkeep for them. If the player can't afford the upkeep, the mass extractors do not generate mass in this turn, even though the player's other power plants might generate additional energy afterwards.

Each building can contain one or several building processors, that generate resources. For example, the combustion power plant has two building processors that generate a flat amount of resources each turn, one that generates energy and one that generates pollution. The mass extractor, on the other hand, uses a slightly more complex

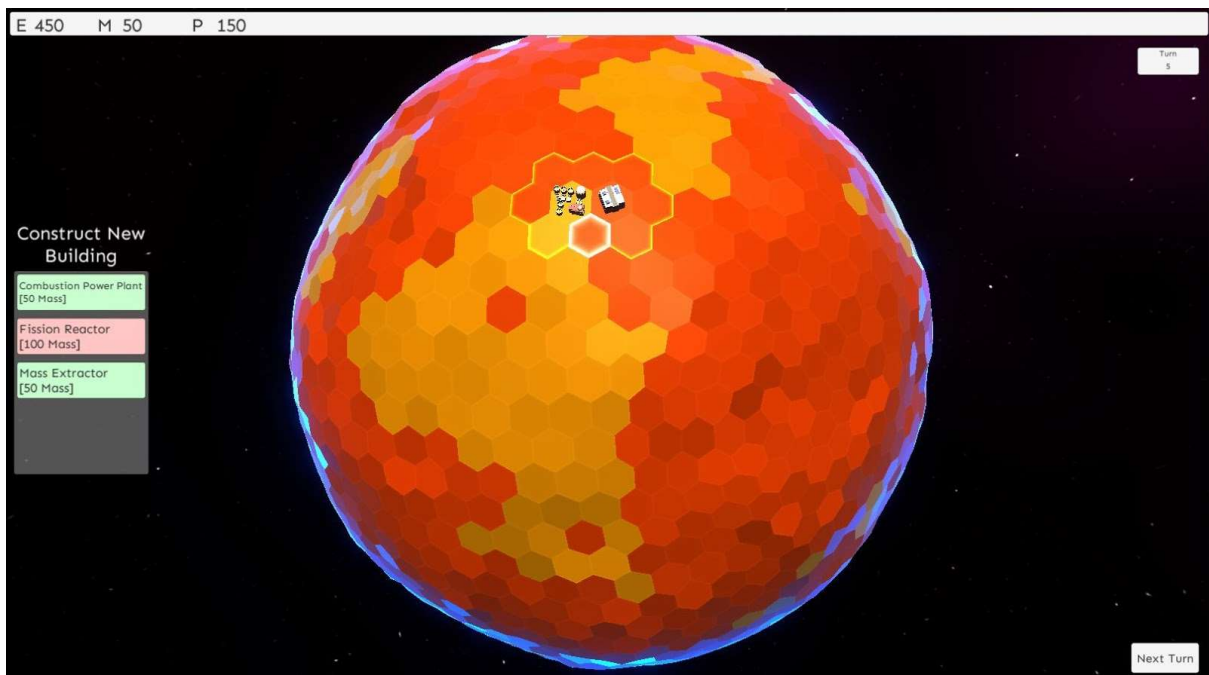
resource generator that takes the mass value of the tile the mass extractor resides on into account.

We created the buildings using low poly models from the asset store, as neither of us is proficient in modelling.

3.5 UI

Our UI currently comprises several panels that show information about the currently selected tile and the player's resources. There are buttons to select a starting position and to advance one turn. A sidebar allows the player to construct new buildings or modify existing ones. Starting a new game is currently done using a debug UI, so we need to add some sort of main menu later on.

The art style of the UI is rather simple, as we used Unity's default sprites for buttons and panels.



3.6 Audio

We implemented a basic Audio Manager that allows game objects to play any sound whenever it is necessary. However, we didn't yet decide, which sound effects and background music we want to use.

3.7 Issues

3.7.1 Models

The use of models from the asset store is up to change, as this approach involves several concessions. First of all, it is hard to maintain a coherent art style, as not all types of building are covered within one asset pack, and different asset packs can vary a lot in style.

Additionally, as most buildings are viewed from the top and take up only very little space on the screen, it is rather hard to spot lots of the models' details. Furthermore, the most important aspect of the buildings' visual design is for the player to be able to identify each building within a split second, which becomes harder for more detailed models. Therefore, it is very likely that we will switch back to models that consist of only a few primitives, but that are coloured very diverse.

3.7.2 Turn Processing Order

We have a lot of things on each turn that need to happen sequentially, and in the correct order. Every turn update, buildings need to be processed, which can include tile and resource modifications. Additionally, tiles need to be updated, meaning their climate effects need to be updated. These, in turn, have different effects that need to be played, and may or may not modify underlying tiles or even destroy buildings. Depending on the order these things are executed in, the results may vary. Whether or not this is a problem, we will see in the actual playtesting of a game. If it is, we have systems in place (like the priority ordering of buildings and tiles) to combat this to some extent.

4. Alpha Release

We made a lot of progress, finished layer 3 of our layered development schedule and implemented most of our high target features.

4.1 World Generation

The hexagons of the world mesh are now connected via quads, to properly enable elevated hexagons. Additionally, the water is now animated to subtly move, making it look less static. Each tile now also has a wind direction value that is used in various parts of gameplay.

4.1 High Scores

We added Credits as the fourth resource to our three existing resources Energy, Mass and Pollution. These can be gained by exchanging energy for credits at a 10:1 ratio (for now). Additionally, the player gets 5% interest each turn. This opens up more strategies, such as buying credits early on to profit off of the interest in later turns. A shortage of energy can, however, cause the player to lack the upkeep required for buildings.

Credits will be the ultimate resource that will determine the player's score. No matter how many buildings were placed or how many turns the player was able to sustain the planet, all that matters is the amount of credits earned, which determines their highscore.

Highscores can be viewed in the menu, where a player can also change their highscore name if they want to.

4.2 Fancier Graphics / Animated Assets

We switched the models of our buildings back to simple primitives. This makes them more easily distinguishable and provides a cleaner look in general. To make the game look less static, we added tweening and animations to building construction. Additionally, some buildings have moving parts, such as the mass extractor, wind turbines or power amplifier.

Our UI now also looks a lot more visually appealing, as we added custom graphics to all UI elements. We implemented a title menu, where the player can immediately start a new game, or decide to set a custom seed first, and a small settings menu, where the player can set the volume of music and sound effects, as well as forfeit

early. Many of the UI elements are tweened to make transitions look better.



Gameplay scene showing new UI and some example buildings

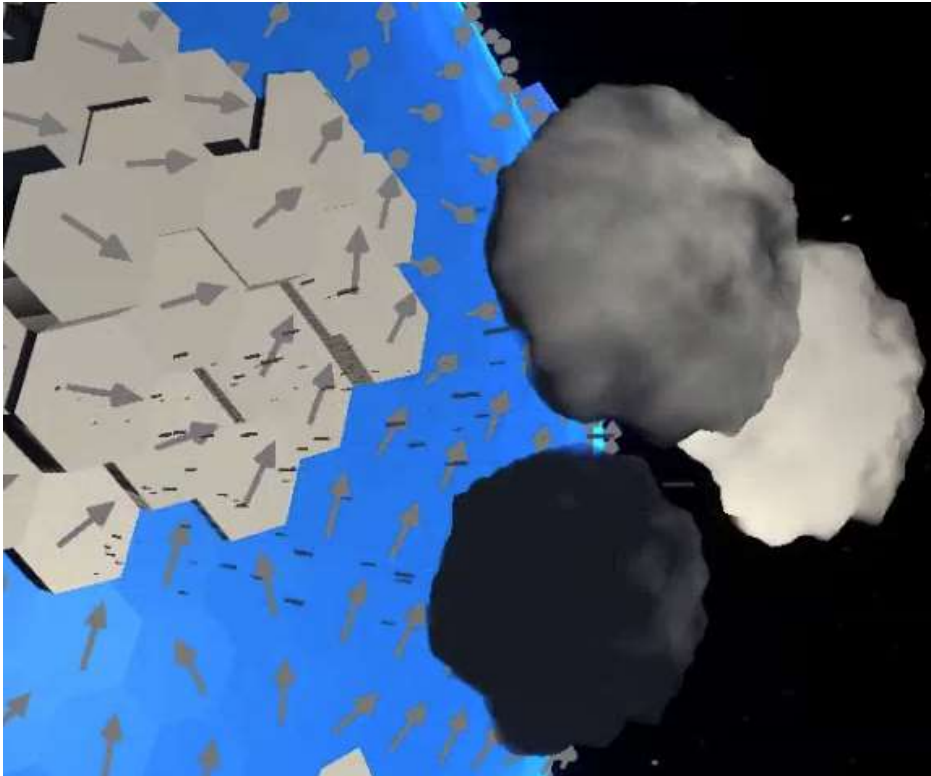
4.3 Quality of Life Improvements

Many quality of life improvements have been added.

- Right-clicking a building in the construction menu gives additional information
- Hovering over a building in the construction menu shows how much mass it costs
- Pressing F3 toggles per-building output UI elements to have a better overview of what is being produced where
- A help menu has been added to the main menu which shows a few tips to the player as well as how some mechanics work

4.4 More Climate Effects

As announced previously, we implemented more climate effects, namely lightning strikes and rain. Rain extinguishes fire and sometimes causes lightning to strike the tile underneath, damaging the building and potentially destroying it. Rain clouds always spread downwind, which can be visualized with F2. Additionally, fire also now only spreads downwind, giving the player a better way to plan ahead. Meteor impacts are now rarer, but deal impact damage and spread more fire. The flood effect was deprecated in favour of raising the water level over time. Tiles that fall below the waterline are converted to water and destroy potential buildings on them if they cannot reside in water.



Rain effect with wind direction visualization enabled

4.5 More Building Types

We implemented the feedback that we got during the presentation of the interim demo, and added streets and bridges that can be used to expand the player's territory. Streets can be placed on most tiles, with liquids being the only exceptions, while bridges can only be placed on water tiles. Furthermore, streets and bridges need to be built on tiles that are adjacent to other streets/bridges, or the player's base. These can be used to quickly expand outward from the initial location to reach another part of the world.

We also added:

- The wind turbine, an energy-generating building with a great energy to pollution ratio, but expensive in mass when built
- The blockade, which blocks the advance of fire and is very durable
- The power amplifier, which boosts the energy generation of the surrounding buildings

4.6 Background Music / Sound Effects

We started the development of a basic audio manager during the interim demo phase. During the last two weeks, we further fleshed out our audio manager and started to actually implement some sound effects and background music. The game starts by playing calm menu music and then smoothly transitions to a randomly selected track once the game has been started.

4.6 Issues

There still are some minor bugs in the game that need to be fixed. The next big step is playtesting the game and balancing all the various factors.

5. Playtesting

5.1 Playtesting Sessions

Most of our playtesting sessions were conducted remotely. During the playtesting sessions, we focused primarily on user experience and playability, while we tested the game ourselves for balancing purposes. While the participants were playing the game, we tried to keep helpful discussion to a minimum, but prompt them to talk about their current experience with the game. Afterwards, we discussed their gameplay experience and gathered further feedback by asking questions.

- Did you have trouble with the controls?
- How was the turn resolution? Did it confuse you? Was it clear what was happening?
- What was your first impression? How did it change as you played?
- Did the game drag at any point?
- What did you find frustrating?
- What did you like the most? What did you like the least?
- Were the game sessions too long or too short?
- Did you have a particular strategy?
- Did you understand what the individual buildings did?
- What would you like to add or remove?
- Did you feel the impending destruction of the planet was clear?
- Did you like the visual? Were they clear enough? What would you change?
- Is the audio too loud? Did you like the BGM and SFX?
- Was it fun?
- What would you say the target group would be?

All the playtesters were told to play on the same seed (237981401) so that we would get a better idea of what playtesters thought of the core gameplay ideas. The following sections detail what we changed about the game by taking into account the feedback we got.

5.2 First Gameplay Experiences

A surprising amount of playtesters completely ignored the help menu before jumping into the game. It was interesting to see how they would fare in comparison to those that read the help menu beforehand.

Even without reading the help menu, camera controls such as panning and selecting tiles came naturally as many games share these basic control schemes. However, some of the things such as right-clicking buildings for more information was not utilized. Gameplay still progressed reasonably even without the tutorial, but slower than we would like, as some core concepts would have to be discovered during gameplay.

To combat this, we initially displayed an arrow next to the “Help” button on first launch. However, players still did not click the help dialogue. Now the player has to click on help before any other buttons become visible. We also added more information to the help panel explaining some more game concepts.

The majority of players had no clear strategy when going into the game, but this is to be expected when playing an unknown game for the first time.

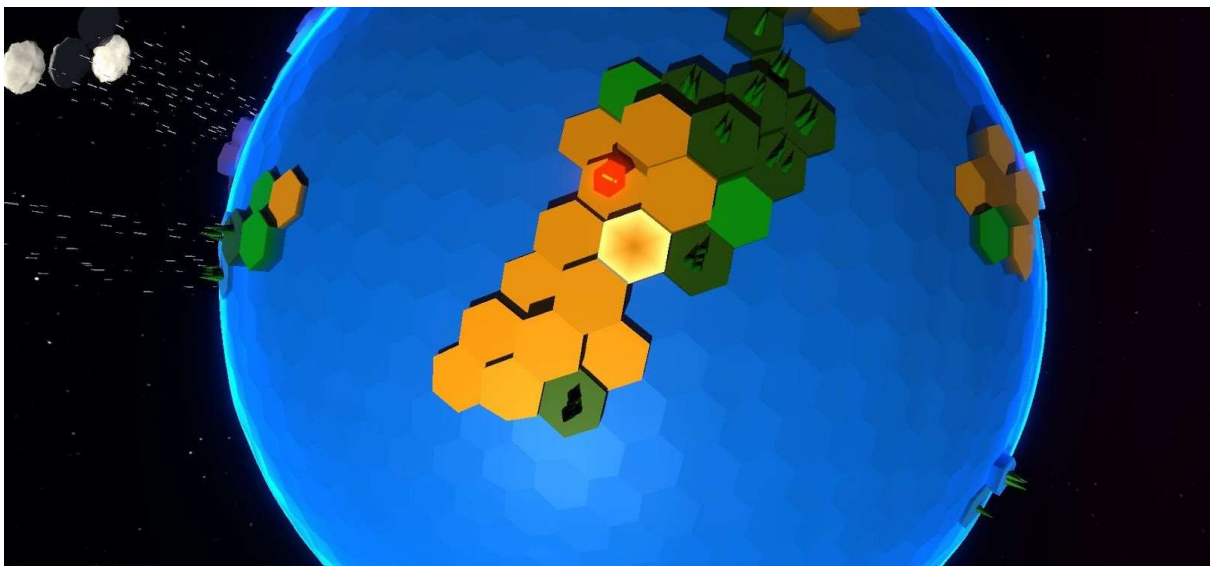
Something interesting a player mentioned, that we had never thought of, is that he disliked the fact that the buildings are randomly rotated, and chalked it up to minor OCD related issues. To address this, we added a toggle in the settings to disable building rotation.

5.3 General Gameplay and Usability

Some players (us included) felt that the game was still missing a few informational elements to support the player during gameplay, so we expanded on this. We added UI elements that show the water level, the height of the current tile, and the required upkeep per turn.

We also added a loading animation during turn updates because some players remarked that they thought the game was “stuck” during longer turn updates.

Meteors have also been reworked entirely as they still felt a bit unsatisfactory and too random. Now, danger markers are placed depending on the player’s pollution level instead of immediately dropping a meteor. These markers have a chance per turn to drop a meteor, giving the player more risk / reward gameplay options to work with.



Danger indicator for meteors

For players that have played multiple rounds, a complaint was that it took too long to place buildings as you constantly have to open and scroll in the build menu. We addressed this by implementing hotkeys, so buildings can be constructed with the

press of a button. The hotkey for a building is highlighted in a different colour in the build menu. This vastly speeds up the pace of gameplay for more experienced players.

The player can activate several overlays that contain per-tile information like the mass value, the wind direction or the resource generation of each building. Previously, these were only activated while a button is held. To account for different playstyles (some players like to really immerse in the game, while others prefer to min-max their strategy as good as possible), we decided to change this to toggle: The first press of a button activates the overlay, the second press deactivates it again.



Several overlays are activated: The mass value and wind direction for each tile, and the resource generation for each building

Another quality of life feature we added after getting feedback is the ability to reset the camera position and orientation. Moving the camera around can change the camera rotation in relation to the player base, which can become confusing. Pressing escape once resets the position, pressing again resets the rotation.

Another issue multiple players (ourselves included) had is that turn updates, especially later on, would take too long. We have since vastly increased the update speed for climate effects and buildings, making the game feel more responsive.

We have also improved on some visuals after getting feedback about them. The base building and fission reactor have been reworked visually, and the sun's light direction now changes every time the turn is updated.

5.4 Balancing

By observing playtesting and playing the game ourselves, we have made additional changes to the balance of the game.

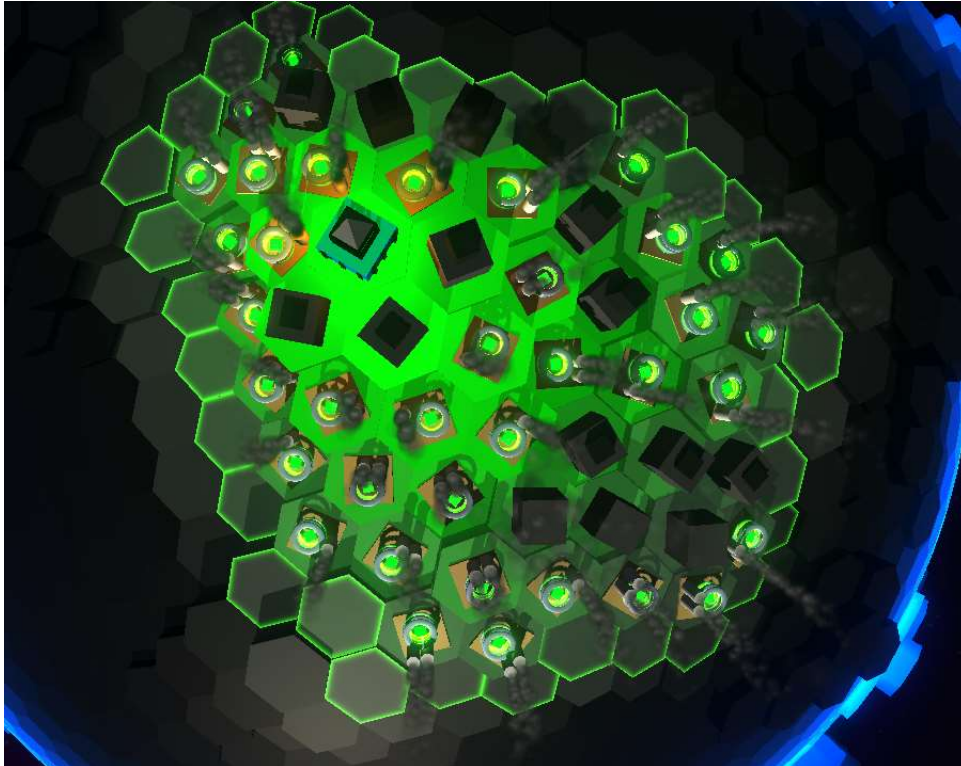
In general, the amount of mass a player gained was too high, forcing them to only build a few mass extractors. To combat this, mass extractors now require more upkeep and slowly drain the mass of the tile they are on, forcing the player to build more mass extractors.

Wind turbines were a too dominant strategy, so we have increased their cost. We also increased the cost of streets and bridges as we felt they were too cheap. They also do not return any money when sold as a player would simply sell all but the last street or bridge upon arriving at their intended destination.

Fission reactors are now a very powerful energy generation option, but they now explode upon destruction, destroying all surrounding buildings. This can even lead to a chain reaction, forcing careful base planning to mitigate potential damage. This also adds more risk / reward to the game as players have to decide if they want to keep or sell a damaged fission reactor.

Meteors have also been reworked. They now show a danger icon before dropping. To offset this, the chance of meteors spawning has been increased.

We also noticed that often climate effects were not localized enough in relation to where the player was building his base (e.g. a meteor dropping on the other side of the world). To address this, we added per-tile pollution values in addition to the "global" pollution. This per-tile pollution can also be visualized using the F4 key. This opens up more design options when working on climate effects, and also gives the players more strategic options. Large clustered bases may be more optimal due to things such as the power amplifier, but these will also create zones of high pollution which may "attract" more meteors. We also feel it makes the pollution of the planet and the destruction of its ecosystem more "real" and visually accessible.



Per-tile pollution; Green tiles are more polluted

5.5 Changes

Here is a rough overview of the changes we made:

- Core
 - Reduced session length -> maximum of 50 rounds down from 100
 - Water level is adjusted every turn instead of every 5
 - Makes progression more clear
 - Water level interpolation time based off of amount
- Gameplay
 - Made turn updates complete much more quickly
 - Meteors now work differently
 - Tiles now spawn a danger sign before dropping the meteor
 - Each turn the tile has a chance to drop the meteor, letting players gauge risk / reward
 - Increased meteor spawn rate to compensate for danger warning, meteors have a higher chance of spawning on tiles with high pollution values
 - Fission reactors explode on death destroying all surrounding buildings, potentially causing a chain reaction
 - Rain extinguishes fire more quickly
 - Rain can now happen on lava planets
 - Bridges can be built on lava
 - Added per-tile pollution
- Balance
 - Wind Turbine cost up from 5 to 7
 - Street cost up from 2 to 3
 - Bridge cost up from 4 to 6
 - Mass extractor upkeep up from 6 to 8
 - Mass extractors now drain mass from the tile they are on
 - Mass extractor productions scales higher with more mass to offset tile drain
 - Streets and bridges now have 0% sell value down from 50%
- Quality of Life
 - Visualizations are now toggles instead of holding the key
 - Mass visualization is paused during turn upkeep
 - Settings toggle to turn building rotation on or off
 - Added "HP" to building HP to make it clear what was being shown
 - Added tile height to tile info panel to make it easier to compare tiles
 - Added energy upkeep UI
 - Added more help text to main menu
 - Added arrow pointing to help menu on first launch to make players more likely to click it
 - Players are now forced to click the help menu on first launch
 - Added loading animation while turn is updating
 - Escape now centers camera to base, pressing again resets camera rotation
 - Increased build menu size
 - Added hotkeys to build menu
- Visuals
 - Changed the way the base looks
 - Changed the way fission reactors look
 - Sun direction changes on turn update
 - Base building explodes on destruction
- Bugs
 - Flooded tiles now have 0% mass
 - Fixed a few ultrawide UI bugs

- Fixed Soft-lock bug
- Power amplifier animation only hits energy producing buildings now
- Fixed multiple smaller bugs

6. Conclusion

6.1 Final Results

We developed a turn-based strategy game, in which the player exploits a planet to gather resources, raise an empire and gain money. The player can construct various buildings to generate the two major resources of the game: Mass and Energy. But while draining the planet from its natural reserves, the pollution level is rising. The player needs to fight against climate catastrophes and other hazards caused by self-induced climate change.

The Game allows for a wide variety of strategies, ranging from rather conservative approaches that aim to keep the amount of pollution low, to highly aggressive strategies that try to squeeze out as much money as possible, before facing the inevitable demise of the planet.

Furthermore, as the game features procedurally generated planets with different biomes, every game will be different than the one before. Also, each biome demands a completely different approach from the player.

6.2 Experience

We are very satisfied with our final game. We managed to carry out the spirit of the game (being an intricate strategy game aimed towards short sessions) as we intended it to be from the very beginning. The regular meetings turned out to be very beneficial for us, as they allowed us to gather feedback from people who are not directly involved in the development process. We revised our game design choices several times, implementing numerous small suggestions, and also very significant recommendations, like making the game less RNG heavy. Aside from that, we didn't really face large time issues and implemented most of the features that we initially proposed during the formal game proposal. As a matter of fact, we actually implemented additional features that initially were not contained within our development schedule, like a playable tutorial.

Although Terrafarm, a turn-based strategy game that is played on tiles, is perfectly suited for a physical prototype, we have mixed feelings regarding the physical prototype. On one hand, we gained valuable insights during the prototyping phase, especially regarding balancing and how exactly our turns should take place. On the other hand, very little of these insights really affected the early development of our game. While we understand the purpose of a physical prototype, we didn't really see the benefits of it in our particular case.

One concern that arose in the beginning of the practical regards the group size: As two of our initial group members dropped out of the course shortly before the beginning of the practical, we had to divide the development of the entire game and

the generation of the project notebook and the presentations between only two people. However, this did not turn out to be a big hindrance, as we were able to complete all tasks in time.

6.3 Personal Impressions

1. What was the biggest technical difficulty during the project?

Linus: The turn resolution is a big part of the game, and managing what updates when and how that influences the game can be tricky, especially since requirements can always change when feedback comes in or something doesn't work out during development. Also, while not truly a technical difficulty, there were many features we would like to have tried during playtesting, such as more unique building effects such as "raising" tiles to combat the rising water level and open more strategies. However, with only two weeks of playtesting implementing game changing mechanics can quickly get out of control, especially as it devalues previous feedback to a certain extent. This is no fault of the organizers however, just a side effect of the tight development schedule.

Simon: While it is not solely a technical difficulty, balancing the game was one of the more challenging tasks for me. This ranges from early decisions that influence the technical implementation, like when and in which order buildings update and climate effects take place, and how buildings are affected by climate effects, to pure numbers like energy generation or cost. Each additional feature opens up a wide range of new features, meaning that the game needs to be rebalanced all the time.

2. What was your impression of working with the theme?

Linus: I initially felt a bit boxed in by the theme, but that probably has more to do with the games I am familiar with. Climate change made me immediately think of resources and a Civilization-type of game. In retrospect there are many interpretations of "Climate Change", and while I am happy with the final result, I wish the lecturers had made it a bit more clear at the beginning how far we are "allowed" to stray from the literal meaning of the theme.

Simon: At first, I was a bit concerned if I liked the theme, especially as I found it difficult to generate ideas that go well with climate change, which is a rather heavy subject. However, once I talked to Linus, we quickly settled on an idea that I really liked. I think our game took full advantage of the theme.

3. Do you think the theme enhanced your game, or would you have been happier with total freedom?

Linus: I think a theme is important, as it challenges students to come up with game ideas and concepts that are out of their normal comfort zone (i.e. games which they

play in their free time). In my opinion working with a concept or theme you are less familiar with leads to more experience gained when the project is finished.

Simon: I think it is really important to be provided with a theme, as, in my opinion, limiting the options enhances creativity. Although our game fits the theme very well, I think it is really important that students should not be penalized if they interpret the theme rather freely, as the sole purpose of the theme is to enhance creativity. Once someone settled on an idea, the theme doesn't matter anymore (in my opinion).

4. What would you do differently in your next game project?

Linus: I would make a bit more use of the prototype phase in our next project. It was a bit difficult for us, in part due to the pandemic as well as the game relying heavily on the world update steps, which are difficult to model with a physical prototype. While I think we covered the important parts during the prototype phase (gameplay feel and initial balance), I would be interested in exploring more dynamic prototyping techniques in a future project.

Simon: There is not much I would change, as the project management and the cooperation between the two of us was fairly straightforward. I think we had a very similar vision for the game, so we're able to carry out our plans without any difficulties. One thing I would really enjoy is to work in a larger team, as this would open up much more possibilities and contribute a lot to expand on my soft skills.

5. What was your greatest success during the project?

Linus: I am very happy with how the core gameplay loop turned out (build->updateworld->build->...) and there are plenty of ways I could think of expanding it in the future. I also like the gameplay which, in my opinion, feels very snappy after our multiple rounds of feedback, which I would attribute to faster turn updates and building hotkeys.

Simon: The greatest success for me is that the game actually came into being more or less exactly as I had envisioned it initially. It looks and plays very well. Although it is not polished enough to actually publish it, it really feels like a solid, complete game.

6. Are you happy with the final result of your project?

Linus: I am pleased with the final result, although there are a few more concepts I would have liked to play around with (certain climate effects and buildings).

Simon: For the reasons described in the previous section [6.2 Experience](#), I am very pleased with the final results of our game.

7. Do you consider the project a success?

Linus: All in all I would consider it a success, especially with the addition of a proper interactive tutorial it feels like a much more "complete" game.

Simon: As I am happy with the final result of our project, I also consider it a success.

8. To what extent did you meet your project plan and milestones (not at all, partly, mostly, always)?

Linus: As we initially kept our targets very conservative as we were only a group of two, we quickly met a lot of our milestones. We did have to shuffle a few things around towards the end however, but I would assume this is normal as priorities change during development, and the "feel" of the game is very important.

Simon: As described previously, we met most of our milestones. We needed to ditch some of the high target and extra milestones, like asymmetric multiplayer and (unfortunately) the story. However, we implemented all crucial milestones, and even more (like the playable tutorial).

9. What improvements would you suggest for the course organization?

Linus: I am happy with the course organization overall, and liked the "hands-off" approach. As mentioned earlier however, I would appreciate it if the course organizers could define a little more clearly what the actual requirements are that define the final grade, and how literally the theme needs to be taken. These are very minor issues however and if I could I would retake the course as I had a lot of fun during development and learned quite a bit.

Simon: I am very happy with the course organization.