

# *The Starlight Rebellion*

Luna Y.

SFWRENG 3GB3 – Assignment 2

February 14, 2022

## Contents

1	Basic Information	1
2	Gameplay	2
3	Aesthetics	3
4	Player Experience	3
5	Changes from Original	3
6	Works Used	4
6.1	Concepts . . . . .	4
6.2	Code . . . . .	4
6.3	Art . . . . .	4
6.4	Fonts . . . . .	5
6.5	Music . . . . .	5

## 1 Basic Information

In this game, the player will play as the leader of a rebellion against an interstellar empire. The game itself will be a top-down style turn-based strategy game with the gameplay dynamic of “board control” at its core. The tiles of the board will be the star systems making up the aforementioned empire, and the game pieces are represented by fleets of starships. The player starts with a single fleet in a single star system, and must use it to capture more systems and eventually win the game by controlling the entire board.

The game takes aesthetic inspiration from similar games such as *Stellaris*, *Starcraft*, and *Star Wars: Empire at War*. The layout of the map will also be similar to that of the galaxy map of *Empire at War*, which is essentially just an undirected graph with nodes being star systems, and edges being hyperlanes. I believe this style of movement encourages slower, more strategic gameplay, as units on the game board are considerably more limited in their movement than they would be in a standard RTS or turn-based strategy game with conventional tiles. However, this game differs in that, where *Empire at War* has static maps that do not change, this game will have a single map whose layout will be randomly generated between different playthroughs, possibly having some basic settings that affect difficulty as well. This will allow for more variety between playthroughs, as well as more replayability.

The combat of the game will be somewhat similar to *Chess*. Each fleet can move to any adjacent system. For simplicity, if a player moves a fleet onto a system containing a fleet from the opposing player, the attacking fleet defeats the defending one and conquers the system. Any attacking fleet will conquer a system if they arrive unopposed as well. However, the combat differs from *Chess* in that there is only one type of unit, with basic movement mechanics (so nothing similar to knights, bishops, etc.). Using *Chess*-style combat as opposed to a more conventional system where units have attack and defence values also encourages more strategic gameplay. For example, if an imperial fleet is defending a system, and the adjacent systems were empty, the player will not be able to approach the defended planet without taking losses, since moving in will leave the player's fleets vulnerable to an attack. As such, the player will have to either lure the imperial fleet out somehow, or send in more fleets to overwhelm it, which is a trade-off they must consider. The player might also choose to be more aggressive, which likely leads to more losses, or they can choose to be more defensive, which allows their opponent to grow stronger over time.

## 2 Gameplay

As mentioned above, the game board will consist of various tiles, represented by star systems, and pieces, represented by fleets of starships. The star systems will be connected to one another via "hyperlanes". The tiles of the board should form a connected graph, but two systems might not necessarily be directly connected to one another. The player begins the game with one fleet and one system, and must use their initial fleet to capture more tiles in order to win the game. The AI opponent will start with considerably more resources at its disposal than the player, which the player will have to overcome. The player may obtain more fleets by capturing/liberating systems from the AI, while the AI will receive new fleets every couple of turns. This also fits the game's narrative rather well; the player is leading a rebellion that has popular support, so liberating systems allows more people to join them, while the AI-controlled empire must assemble new fleets the regular way.

The board will also be procedurally generated, in order to save time on level design and add variety between playthroughs. Parameters such as the number of systems or number of hyperlanes should also be modifiable, which will also have an effect on how difficult the game is.

The game will feature turn-based movement and combat, with each fleet being able to move independently from one another (i.e. the player can move all their fleets in one turn, rather than only one fleet per turn). The combat will also be very simple, with attacking fleets always winning over any defenders, as described above. Because of this, each tile of the board will also be limited to holding one fleet at a time. This might cause issues with how new fleets are spawned, so in the event that a new fleet is trying to spawn on an occupied system, it will instead be placed at the nearest unoccupied friendly system.

### 3 Aesthetics

The visuals of the game should look futuristic, and themed around science-fiction and space. The user interface of the game should somewhat resemble a kind of spaceship/computer console (similar to the Terran interface in *Starcraft*).

The graphics of the game will be predominantly created using 2D sprites. That being said, graphics are not extremely important for this game, and the visuals should be kept relatively simple to save time.

The music should also have background music that is not too obtrusive, and this music should have a futuristic, possibly militaristic feel to it.

The player will receive a brief text-based introduction to the game universe when they start a new game, as well as some flavour pop-ups over the course of gameplay. Story is not *extremely* important, but the player does need an introduction to the game's setting, however brief it may be.

### 4 Player Experience

Overall, the game should feel slower-paced and methodical. This is reinforced by the turn-based nature of the game, which allows the player to take their time and really think strategically. As a side effect of this, it should theoretically be less likely for the player to get frustrated at the game for reasons like being caught off-guard by the AI, or losing a system or fleet, since they would presumably be able to predict these situations to a reasonable degree, and course correct where necessary.

The player should also feel like they are overcoming some very difficult (but not necessarily insurmountable) odds in order to achieve victory against the Empire. This is supported by the game's narrative, as well as the player's starting position. Realistically, any uprising against a large interstellar state would be hard-pressed to succeed, and this is reflected by the vast difference between the player, who starts with one system and a single fleet, and the AI, which starts with the rest of the star systems and several more fleets. It is up to the player to overcome these odds and win, and if they do, they should feel a sense of triumph as well. That being said, however, this difficulty will likely be indirectly adjustable through various map generation settings before the game starts, for players who prefer a slightly more relaxed experience.

### 5 Changes from Original

There were not very many drastic changes from the original design; they were mainly small adjustments made in order to improve gameplay. For example, one feature that was added is that when the player first loses all of their fleets, they are given a single backup fleet so that they have a chance to recover. This was added, as there is a lot of randomness in the game's setup, which led to some situations that I felt were a bit too unfair; this should hopefully help alleviate that somewhat.

One aspect of the game that received a lot of changes over the course of the assignment was the map generation. Different parameters were experimented with before settling on the current default settings of 15 stars, between 1 and 3 hyperlanes, and a minimum distance between stars of 5 units. These settings seem to result in a decently challenging game while not being extremely overwhelming. It's also interesting to see the effects that the map generation parameters have on gameplay; the number of star systems is closely related to the "early-game" difficulty, as it affects the starting strength of the AI opponent, in addition to also affecting the amount of time it takes to win the game (as an aside, the minimum star count of 5 seems to work quite well as a tutorial of sorts, as it allows the player to familiarize themselves with the controls while being fairly easy).

Changing the number of hyperlanes each star can have also affects the difficulty, albeit not linearly; lowering this number results in the player having less manoeuvrability while also reducing the possible directions an attack might come from, while raising it gives the player more room to manoeuvre fleets and retreat, while also increasing the amount of directions they need to defend. Minimum distance between stars, on the other hand, doesn't affect difficulty much; it mainly just allows the player to reduce clutter by increasing it.

Another minor change that was added is that certain statistics, such as the game's length, and how many casualties the player took, are now tracked and shown to the player when they successfully beat the game. This encourages the player to create meta-goals to try and minimize how long it takes them to win, as well as how many losses they take. Another fun meta-goal that might arise comes from the fact that the backup fleet which spawns as the player's last chance has a special name; this might encourage the player to try and make sure this fleet survives to the end of the game, if it appears.

This design document was also updated to reflect the given feedback, as well as to document any third-party assets and other such things that were used.

## 6 Works Used

### 6.1 Concepts

The map generation algorithm uses a technique involving grids in order to efficiently test for overlap, which was learned from [this video](#).

### 6.2 Code

The game features many script assets originally made for a personal project of mine called *Nightscape*, which is currently not released publicly. This includes camera functionality, object selection, as well as random name generation. Many of the names themselves include segments generated from [Fantasy Name Generators](#).

This game also uses a lot of miscellaneous code originally created for *Lightrunner*, including various shaders, as well as the audio player.

The game also uses the [A\\* Pathfinding Project](#) as its pathfinding algorithm, which uses the [Standard Unity Asset Store EULA](#).

### 6.3 Art

#### UI Minimalism SciFi

- The game uses UI assets from the “[UI Minimalism SciFi](#)” pack, created by [Wenrexa](#).
- These assets are in the public domain, under the [CC0 1.0](#) license.

## 6.4 Fonts

### Nova Square

- Licensed under OFL
- Sourced from [Google Fonts](#)

### Spartan

- Licensed under OFL
- Sourced from [Google Fonts](#)

## 6.5 Music

Auroras of Saturn by MusicLFiles

Link: <https://filmmusic.io/song/6782-auroras-of-saturn>

License: <https://filmmusic.io/standard-license>

Dawn by Alexander Nakarada

Link: <https://filmmusic.io/song/4867-dawn>

License: <https://filmmusic.io/standard-license>

Horizon Flare by Alexander Nakarada

Link: <https://filmmusic.io/song/4837-horizon-flare>

License: <https://filmmusic.io/standard-license>