

Interim Report
3D Video Game: Hope and Control

267533

October 25, 2024

0.1 Introduction

While the written word can gain a form of dynamism through ambiguity, applying that to an interactive and visual medium becomes challenging to do concretely. Instead, gameplay elements, such as simulation and customisation, can be used to engender a narrative in the player, giving them the space to fill in the gaps in their own minds. When done well even non impactful decisions can be imbued with a sense of weight, this feeling that everything contributes. On the other end when done poorly, all decisions can become meaningless whether they are impactful or not. What matters is the player believing they're having an impact.

This form of narrative interaction is seen commonly in the large-scale real-time strategy genre (e.g. Stellaris, Crusader Kings, etc.) where interlocking systems tell the stories of large factions. The objective of this project is to first create a beta version of a game in this genre, then perform a round of user testing. Insights gained from that process will then be used to complete a vertical slice. Given the constraints inherently imposed by solo development across such a limited time scale, it is unlikely a fully finalized version of the game will be completed.

To allow the player to make immediate, impactful decisions at the beginning of any given run, unlike most games in the genre, the player's power (i.e. their ability to affect the simulated environment) will be organized differently to the other factions. Instead of playing as a larger force the player will be a single powerful entity (i.e. a single ship, a single character, etc.) with a large amount of upfront ability to effect change. Alongside this, the player will be immediately presented with customization options for their faction typical to the genre (e.g. faction symbol, faction origin, etc.), leading them into a roleplaying headspace.

To ensure that the player's actions result in dynamic outcomes, the faction simulation will also need to be suitably complex. As such, this project takes a code-agnostic, data driven approach. This involves keeping faction data separate from any given simulation element, as opposed to a typical object-oriented approach, which would intertwine them. This allows simulation modules to be designed and implemented in a flexible manner, as well as allowing extra simulation elements to be added easily as an extension goal.

This report consists of several sections, the first of which is "Professional Considerations", where user testing and acquiring consent for personal data collection is discussed in detail. Next is "Requirements Analysis", which outlines the objectives of the project in distinct steps alongside expanding on the explanation of the problem space and my solutions. This includes background research on games in the genre and proposed solutions to common game design issues. After that is "Art Design", which details some techniques used to render the large scale solar system, followed by "Simulation Design". That section of the report discusses in detail all the parts that make up the simulation's structure, with some expansion on specific finer details. Next is the "Project Plan", consisting of a breakdown of the work required to complete the project into distinct phases,

illustrated using a Gantt chart. Then there is an "Interim Log" detailing the steps of the Project Plan completed as of the submission of this Interim Report. Finally, there are the references and appendices.

0.2 Professional Considerations

0.3 Requirements Analysis

This project aims to produce a player-story driven RTS (Real Time Strategy), in the form of a vertical slice (a version of a game that lacks some features but still showcases what the final product will look like). Unlike a typical RTS, there will be a larger focus on the world reacting directly to the player as a singular powerful entity and less on long term historical strategy. As such that world will need to be deeply simulated, comprising a large amount of the technical challenge of the project. While the game will be in part targeted at long time RTS players, this change in power balance (alongside a shift into more roleplay focused elements of the genre) is designed to allow a wider audience to engage with a typically complex genre.

This project uses Unity instead of a custom engine, the primary reason for this choice is my familiarity with its workflow. I've previously created a full game using Unity (including shipping it to Steam) and I greatly enjoy its flexibility. Unity makes it easy to extend the editor through creating additional context menu items and entirely new editor tools. The C# (which is Unity's primary coding language) ecosystem is also very broad, with a colossal number of resources to pull from.

To assist with the development process, multiple rounds of in-person user testing should occur. The sample of players should be made up of both people familiar with the RTS genre and those that are not, ideally spread evenly across that range. Two primary testing rounds should occur, one before an initial version of the game is produced and one afterwards, utilizing a beta version. This beta version can differ from the vertical slice dramatically in terms of completeness but should still be functional. Any additional user testing (including online testing) is helpful but should be considered an extension objective given the time constraints.

0.3.1 Background Research, Design Problems and Solutions

Game 1: Stellaris

Stellaris (2016) is a large scale space-based strategy game, taking place across a substantially sized galaxy. It involves managing an interstellar faction, taking them from their first ships to combating a galaxy wide threat.

The game puts a heavy emphasis on roleplay right from the beginning of a run. Immediately upon selecting "New Game" you are presented with the

screen pictured in Figure 1.



Figure 1: Stellaris Faction Creation Screen

The left side of the screen showcases a list of premade factions (highlighted in yellow) that the player can pick from then customize, these can act as a jumping off point for player ideas. There is additionally an option to create an entirely unique faction at the very top. Positioned directly in the center of the screen is the information on the currently selected faction, including; their governmental practices (highlighted in red), their species traits (highlighted in green), their species origin (highlighted in blue) and their faction's flag (highlighted in orange). All these elements (alongside others) create a distinct feeling for this particular faction and, more importantly, they have gameplay ramifications. Within a minute of starting the game proper the faction shown in Figure 1. (an Imperial Divine Empire) was notified it had received a new heir, a special type of unit that acted (in this case) as a scientific advisor.

This focus on roleplay is reflected in the reviews for the game, many are simply retellings of their own faction's story. An example of this is pictured in Figure 2.

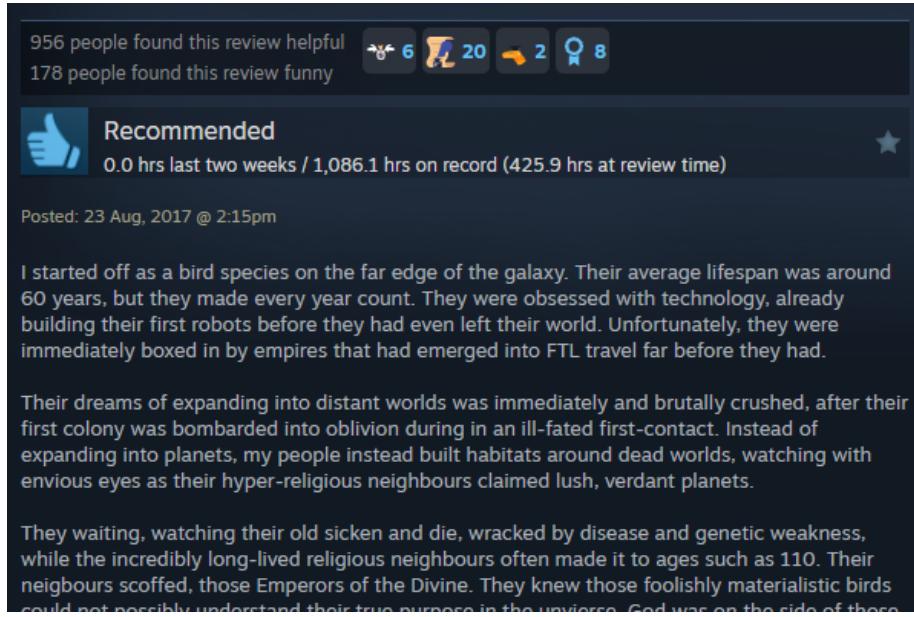


Figure 2: A retelling of a Stellaris game, (full review is 650+ words long)

Other systems also assist this storytelling, in particular the Galactic Crises. These are large, late game events that typically have ramifications for the whole galaxy and act as a definitive final boss for a playthrough. Interestingly, there are also specific cases where a player or AI can become the crisis, another avenue for potential roleplay. The player's relationship to other factions can be vitally important during these scenarios, meaning earlier (possible roleplay based) decisions can have a major impact.

Space combat is another important part of Stellaris but is more used as a tool to express roleplay decisions than a system that encourages roleplay itself. The functional act of combat instead enhances the grand strategy side of the game, engagements are often won by decisions made before the battle starts. For example, the ships used by an endgame crisis, "Extradimensional Invaders", largely rely on shields as a defensive measure for their ships. This means that kinetic weapons are particularly effective [2], outfitting ships with these weapons is a decision made when they are created. When actually fighting, fleets of ships will "auto-battle" each other, with no input required by the player. While this project does focus more on a singular powerful entity, it will take a similar pre-planning approach to combat. This is in part due to the time constraints, as time spent making combat rewarding to execute takes time away from the simulation side (and main goal) of the project.

While Stellaris succeeds in many areas and is arguably (with its balance of strategy and roleplay) the closest real example of this project's goal, it has a

few outstanding flaws. The game has a large amount of upfront complexity, the main Stellaris wiki’s Beginner Guide [1] is roughly 17000 words long, split into 20+ sections and subsections. Figure 3. showcases the first screen players will see outside of the faction creation menu. It hosts a suite of information, shown to the player all at once. This includes (but is not limited too); eight types of basic resources, several more advanced resources (including fleet and starbase capacity), a list of all controlled entities (planets, fleets, construction and science ships and shipyards), the current date, many links to other UI windows (including Government Control, a larger galaxy map, Species Control, Technology and Research Management) and buttons for switching map mode (it should be made clear those buttons do nothing on this screen but are still present).



Figure 3: Stellaris Solar System View Screen

However, this complexity is an important, and arguably defining, part of the genre [3]. The principal issue here is how it is introduced to the player. Given this project’s intention to, in part, target a wider audience, this issue is quite important. The intended solution is to introduce player elements gradually, in the context of the larger narrative brace of the game this can be expressed as the player’s controlled entity turning back on after many years inactive. To enhance replayability players could be allowed to start already in a fully active state rather than repeating the tutorial each time.

Stellaris also suffers from performance issues, particularly in the late game [4] [5]. Given the relative scales of this project and Stellaris this is a less relevant issue, performance goals are discussed further in a later subsection ”Performance and Stability”.

Game 2: Disco Elysium

This project's roleplay centric aims require it to take additional inspiration from outside the RTS genre. Disco Elysium (2019) is a detective RPG, famous for its engaging writing. It has many systems that incentivise roleplay such as its "Thought Cabinet" system, pictured below.

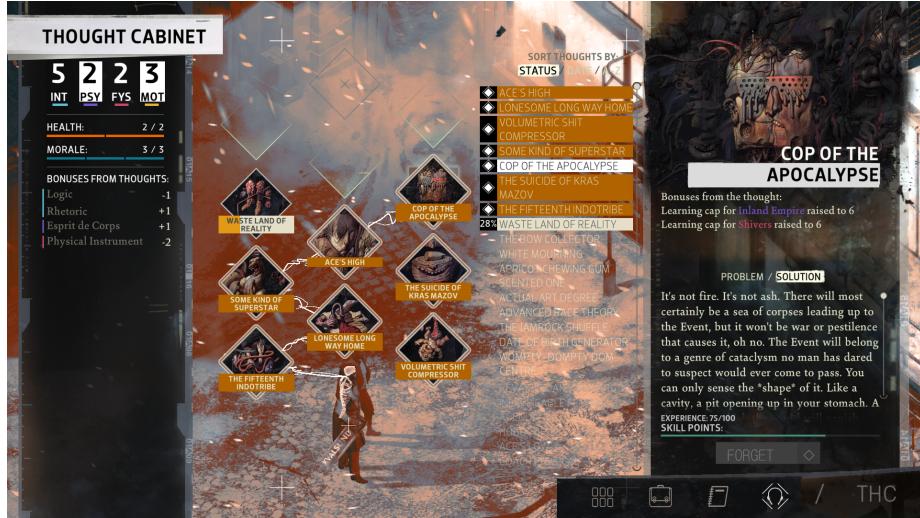


Figure 4: Disco Elysium Thoughts Screen

"Thoughts" are discovered naturally on your adventure throughout Elysium's world, after being found they can then be slotted into an open position in the cabinet before taking some time to be "internalized". This system mirrors Stellaris's governmental polices, providing an avenue for roleplay but grounding it in a more understandable mechanical benefit/cost. In the case of Disco Elysium, you use a skill point (that could be used on the game's more traditional statistics based leveling system) to unlock more slots in your cabinet. This allows players to make the active choice to engage more with roleplay, potentially trading off a traditional power increase.

Disco Elysium's approach to roleplay has some pain points, in particular the various bottle-neck checks. The primary example is an "Authority" check during a pivotal conversation with Titus Hardie. This check (involving summing the result of two dice and your authority value to try and meet a threshold) is necessary to move the story forward, with a low enough investment in Authority (a seemingly optional stat up until this point), the chance to pass it can be as low as 3%. In the context of a game trying to tell a specific story, with a character that values authority as much as Titus, this (despite the potential player frustration) works quite well, however, in a game more centered on player driven narrative it would not. Given the more dynamic form of storytelling this project will seek to employ, it will be important to ensure these bottle necks

only appear in a foreseeable manner. For example, it makes sense that a faction's military will be tested if it goes to war, hence, potential war declarations should be clearly telegraphed.

0.3.2 Performance and Stability

Crashes and bugs should be kept to an ideal minimum. While the time constraints imposed don't allow the thoroughness typical of a standard project, ample time can still be allocated to unit testing and the like. The aforementioned rounds of user testing can also assist with this issue.

Alongside functioning as intended, the final version of the project should also run at a consistent and reasonable frame rate (60 frames per second) on low to medium range hardware. Given the relatively low intended visual complexity of the game, this objective is still important but easier to reach than others. Various different pieces of hardware are available for testing, with varying hardware capabilities in each.

0.3.3 Formalized Requirements

Below are the requirements discussed in the above sections formalized into a list;

R1: *Existing games in the genre should be researched, their gameplay focus points, style and design problems should be identified.*

R2: *Interviews should be conducted with 5+ potential players, questions should be shaped based on insights gained from completion of R1.*

R3: *At least two versions of the game should be created. An initial prototype for additional user testing and a final vertical slice.*

R4: *The game should run at atleast 60 FPS on low to medium range hardware.*

R5: *The game world should be suitably simulated, recognizable as realistic relationships between factions.*

R6: *The player should be able to have an effect on the game world, represented through simulation statistics changes.*

R7: *The game should have sound effects that fit the themeing.*

R8: *The player should be able to move about the game's world.*

R9: *The game should be functionally stable, crashes should occur very infrequently.*

R10: *The game should have UI that accurately describes statistics about the*

player's faction.

R11: *The game should have UI that accurately describes known statistics about NPC factions.*

0.3.4 Formalized Extension Requirements

If time allows, there are some additional requirements that would ideally be met;

E1: *Additonal simulation elements. For example, if the current version of the simulation models a planet's resources as a static value, a new simulation routine could decrease that value over time based on population and industrial activity.*

E2: *Online functionality. This would likely take the form of online leaderboards, showcasing statistics such as "Time Survived", "Enemies Killed", "Highest Bounty", etc.*

E3: *More advanced combat functionality, taking the form of additional actions ontop of the exisiting auto battle system.*

E4: *Music. While many royalty free options exist, unique music would help set the mood properly for players.*

E5: *Releasing the game on Steam. Generally speaking this is a capstone objective and would signal the end of the code side of the project.*

0.4 Art Design

Visuals are an important part of game development but they are not the focus of this report. This section of the report will focus on the system used for large scale rendering (used for celestial bodies) as an example of some of the work done on that side of the game.

The effect is made of two key parts, an "impostor sphere" shader and a system that converts real-space (As in the simulation position of the object being drawn) coordinates to a visual scale.

Note: could talk about our system for displaying far away
large things (shelled size based rendering)

0.5 Simulation Design

Note: Use that draw.io diagram you made

Note: Simulation is made up of Simulation Manager, Faction routines, Faction data, the Simulation Manager Execution Clamp and the faction drawer

0.6 Project Plan

0.7 Interim Log

References

- [1] Stellaris Community, "Beginner's guide", "Stellaris WIki", 21 October 2024, Beginner's Guide
- [2] Stellaris Community, "Crisis/Endgame crisis factions/Extradimensional Invaders/Successfully repelling the Extradimensional Invaders", "Stellaris Wiki", 18 October 2024, Extradimensional Invaders Stratery
- [3] Jónsson, B. Pages 6 to 7, "REPRESENTING UNCERTAINTY IN RTS GAMES"
- [4] "Skullcrusher", "Stellaris Steam Review", "Steam", 11 September 2022, Link
- [5] "A Random Fish", "Stellaris Steam Review", "Steam", 18 October 2019, Link

Appendices

Appendix A

Project Proposal

Space Based Simulation and Tactics Game

Candidate Number: 267533

Supervisor Name: Dr Paul Newbury

Aims

Many games try to create a branching narrative. When done well this can imbue even non impactful decisions with a sense of weight, this feeling that everything contributes. On the other end, when done poorly, all decisions can become meaningless, whether they are impactful or not. What matters is the player thinking they're having an impact.

In this project I intend to (with the constraints inherently imposed by solo indie development in mind) create a video game that creates an emergent narrative through gameplay rather than text, maximising the feeling of impact for the player. This form of narrative is seen commonly in the large-scale real-time strategy genre (e.g. Stellaris, Crusader Kings, etc.) where interlocking systems tell the stories of large factions. There will be some simplification to allow users that would normally be turned off by the upfront complexity of these games to also engage with the final product.

Objectives

Primary Objectives

1. Research existing games in the genre, identifying their gameplay focus points & style.
2. Examine what games in the genre do well and what they do poorly.
3. Conduct interviews with 5+ potential players, utilizing insights gained in Primary Objectives 1 and 2 to shape the questions asked
4. Examine what games in the genre do well and what they do poorly.

5. Use information from all Previous completed Objectives to design a game that tells player driven stories.
6. Create a beta version of the completed game that can be tested and critically evaluated, both by me and potential players.
7. Use insights gained from Primary Objective 5 to iterate and improve on the beta version.
8. Ultimately finish a vertical slice that would lack some features but showcase accurately what the final product could look like.

Extension Objectives

1. Perform Primary Objectives 5 and 6 continuously to further improve the final product.
2. Conduct critical evaluation using the vertical slice, in the same manner as Primary Objective 5.
3. Utilize the insights gained in Extension Objective 2 to further improve upon the vertical slice, expanding it into a fully completed game.
4. Implement additional simulated features into the game, refining the model.
5. Implement online leaderboard functionality for various statistics.
6. Release the fully completed game on platforms such as Steam and Itch.io.

Relevance

This project reflects my intended career path, as someone that has already been working with Unity and C# for the last five years. During my gap year (2021 - 2022) I went through the full development process, releasing a 2D game on Steam in early July. Since then, I've been working on both my development and 3D skills and hope to release another game I've been working on a couple months post university.

By doing this project I hope to further both my coding and artistic skills, combining them to create a hopefully great video game. This project tests both of those disciplines alongside my ability to gather effective feedback and implement proper HCI principles.

Resources Required

I will require use of study rooms/seminars to conduct in person user testing. Testing will also be conducted online but that should require minimal resources. No funding will be required as I expect to purchase games used for market research myself. I also expect to pay for the Steam direct fee if Extension Objective 6 is completed.

Some use of lab computers is required (primarily on Mondays) though I intend to use my home PC on a day-to-day basis. This is due to several factors; my extensive backlog of previous projects I can pull reusable systems/assets

from, the comfortability of the workspace and the ease of access, alongside various others.

Time Management

| | Monday | Tuesday | Wednesday | Thrusday | Friday | Saturday | Sunday |
|-------|----------------|---------|----------------|----------------|----------------|----------------|----------------|
| 9:00 | | | | | | | |
| 10:00 | Lecture | | Lecture | | | | |
| 11:00 | Project | | | | | Project | Project |
| 12:00 | Project | | Project | <i>Seminar</i> | Project | Project | Project |
| 13:00 | Lab | Lecture | Project | | Project | Project | Project |
| 14:00 | | | | | | | |
| 15:00 | Lecture | | | Lab | Lecture | | |
| 16:00 | Lecture | | | Lab | Lecture | | |

Project time per week is equal to 12 hours. This acts as a minimum goal and (due to my general passion for the subject matter) I will likely do more depending on the week. Any blank slots will be used for other modules, whether coursework or study. The minimum time spent each week will increase depending on unseen factors such as unintended scope creep or hidden complexity.

I intend to have the initial early beta version of the finished shortly after the interim report (Week 8 to 9 of Term 1), utilizing feedback gained over the Christmas break. I then expect to finish the final vertical slice a couple weeks before the final report is due. Any additional time gained, by being ahead of schedule, will be used to complete extension objectives, primarily Extension Objectives 1 and 2.

This proposal has been adapted from Word to L^AT_EX, so exact formatting differs from the original proposal document. Structure and content remains the same.