2407-BSE

Investigative Studio II – Proposal

CS301.1

Alexander Legner
William Tetzlaff

Document Outline

Table of Contents

Table of Figures

Table of Tables

Intentionally Blank

Proposal

Executive Summary

Project Title: Architects in Void

Project Overview: Architects in Void will be a space exploration game that aims to immerse players in the deep void of space. Players will embark on a journey to create their colony in space while battling the dangers of being in space and other entities in the empty void around them.

Objective: The primary objective of Architects in Void is to deliver an immersive yet fun space exploration experience that balances realism with accessible gameplay mechanics. By utilizing realistic space physics simulations and intuitive building and flying mechanics, the game aims to immerse the players in realism of space while still engaging in an entertaining gameplay experience.

Key Features:

- **Realistic Physics Simulation:**
  Players will experience realistic space physics including zero gravity physics and realistic thrust/flight dynamics. The realism enhances gameplay by challenging players to master the complexities of being in space.
- **Ship Customization and Construction:**
  Players will be able to design and build their spaceships. This will allow players to create unique ships suited to their mission's needs. Each design choice will impact performance, manoeuvrability, and functionality in the game environment.
- **Exploration and Discovery:**
  The game offers a procedurally generated void with unique asteroid belts and resources. The universe will be primarily based on our solar system the Milky Way to keep with the realism but will allow players to explore our other planets like never before.
- **Resource Management and Survival:**
  The player will be responsible for maintaining their spaceship and equipment. This will require the player to bring or find certain resources to help them stay alive. The unique resources will allow the player to upgrade/edit the ship's components.
- **Missions and Bases:**
  The gameplay loop will be around completing certain missions (single-player campaign) which will involve making trips to existing bases or making new bases, upgrading ships, fighting enemies, which will allow the player to feel a purpose in the void. These bases will also be key for the player to resupply and start/end these missions.

Conclusion: the proposed space simulation game "Architects in Void" promises to deliver an engaging mix of realism, creativity, and exploration. By combining innovative gameplay mechanics with realistic physics, the game aims to inspire curiosity about space while offering an engaging and immersive gaming experience.

Research Question

How can realistic space physics and mechanics be balanced with intuitive gameplay to create an immersive and enjoyable experience for players in a space exploration and spacecraft piloting game?

Explanation:

- **Realistic Space Physics:**
  Having actual physics of space which includes orbital mechanics, gravitational forces, momentum, and many more is crucial in creating a believable and immersive experience in space. Research would look at how realistic physics can be incorporated into the game without overwhelming the player with complexity.

- **Intuitive Gameplay:**
  While realism is important, it is also equally important to make the game accessible and enjoyable for players. The research would explore how to simplify the complexity of space physics into simple and intuitive gameplay and mechanics that players can learn easily which would allow them to focus on space exploration rather than being overwhelmed by technical details.

- **Immersion and Enjoyment:**
  The end goal is to create a unique experience that is both realistic, engaging, and entertaining for players. Research would include how to create a good gameplay loop within a realistic space environment including the challenges and interactions within the game.

- **Realistic Space Distances**
  Space has large distances between each planet. These distances if used directly 1:1 scale would be too large for the typical computer to run. Research would explore how we can make the game seem realistic to a player while still being able to run modern computers.

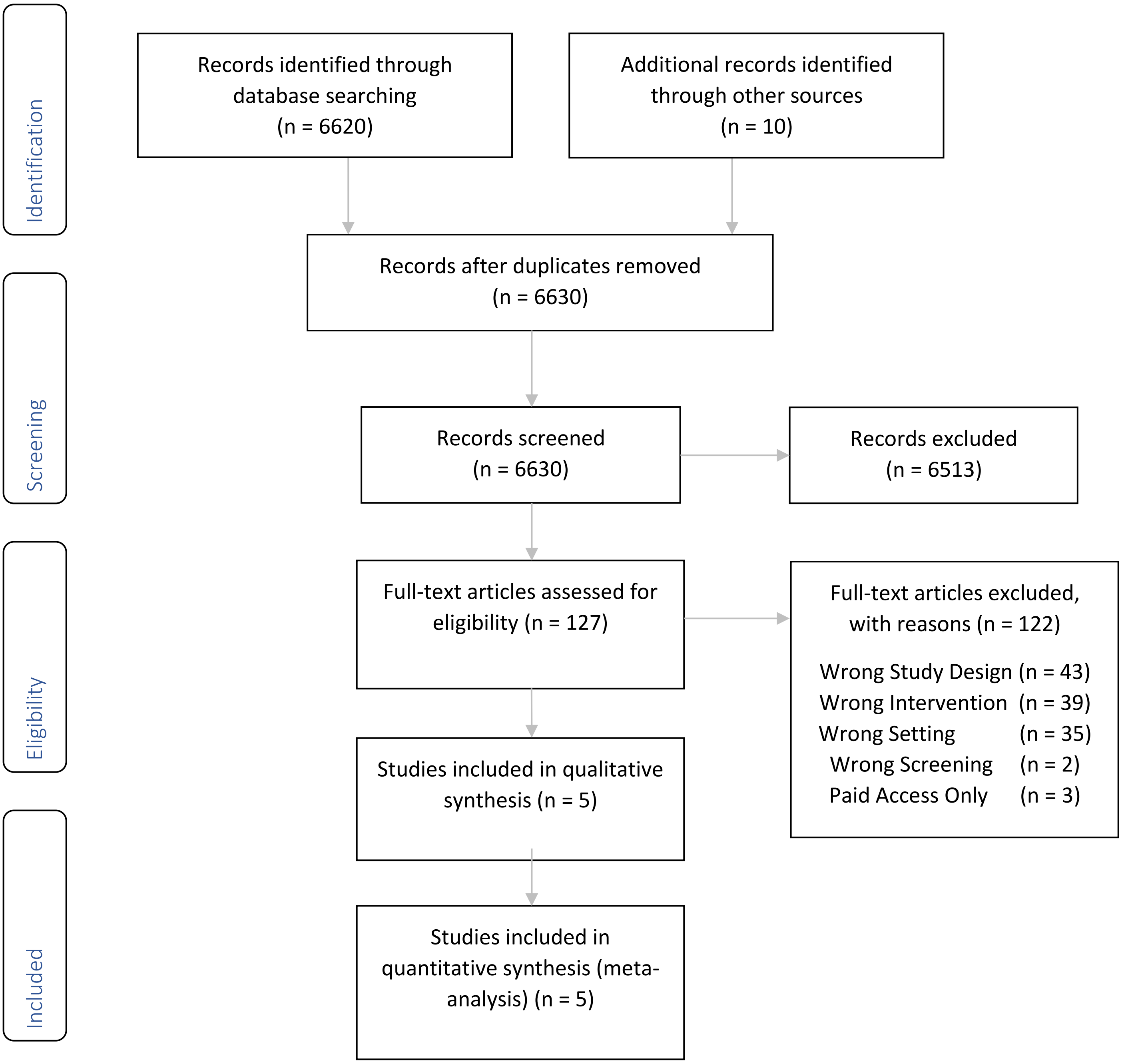
Discussion

A clear solution to this research question/problem would be to create a game that supports all the parts of the research question. Some tools we will need to investigate to help develop a game would be a game engine. This would allow us to develop a game/prototype to help us answer the research question. A knowledge gap we have identified would be the use of doubles for world positions and matrices of transforms. These key ideas are used in manipulating the position, rotation, scale of a transform easily and are standard practice. We will need to research matrix manipulation and double floating-point precision.

After doing further research and creating a prototype we should be able to answer the question and have a product we can further develop in future projects if we choose to continue with it. This research has the potential to also open new ideas around space and video games which we will need to explore during the development phase.

Literature Review

Prisma Diagram


Identification
Screening
Eligibility
Included
Records identified through database searching (n = 6620)
Additional records identified through other sources (n = 10)
Records after duplicates removed (n = 6630)
Records screened (n = 6630)
Records excluded (n = 6513)
Full-text articles assessed for eligibility (n = 127)
Full-text articles excluded, with reasons (n = 122)
Wrong Study Design (n = 43)
Wrong Intervention (n = 39)
Wrong Setting (n = 35)
Wrong Screening (n = 2)
Paid Access Only (n = 3)
Studies included in qualitative synthesis (n = 5)
Studies included in quantitative synthesis (meta-analysis) (n = 5)

Papers | Articles | Resources

“Implementation of a Double Precision Floating Point Arithmetic” by Yasaswini Sudarsanam

This paper talks about how they implemented double-floating precision cores for addition and multiplication for use around sparse linear systems that use matrix multiplication. Although this is from 2007, it is still relevant as most computers today have CPUs that are 64bit and can utilize double precision cores and matrix manipulation. They describe floating points as: Every language supports a floating-point data type, demonstrating how important it is to manipulate floating points efficiently. Every computer has a floating-point processor or specialized accelerator that satisfies the requirements of precision using detailed floating-point arithmetic (Sudarsanam, 2007). There are two types of floating points mentioned – single and double floating point. Single floating point as described in this paper is as using 32 bits of memory to store a number whereas double floating point used 64 bits of memory. This extra memory allows for both higher precision and higher range of values. Compared to single precision, which has an approximate range of 10** (-38) to 10**38 with roughly 7 decimal digits of precision, double precision offers a larger range of approximately 10** (-308) to 10** 308 and approximately 15 decimal digits of precision (Sudarsanam, 2007).

“Game Development with Construct 2 From Design to Realization” by Lee Stemkoski

This book talks about a game development journey with all the requirements for a simple game and how they went about implementing it using a software called “Construct 2”. They talk about everything from the design process to the implementation of specific features. This book teaches you how to make video games using the Construct 2 game engine, which is a great tool for both seasoned game developers and aspiring developers with no prior experience to quickly create game prototypes (Lee Stemkoski, 2017).

“Core Techniques and Algorithms in Game Programming” by Daniel Sánchez-Crespo Dalmau

This book talks about many topics of programming and video games including history of video games, game architecture, data structures and algorithms, design patterns, user input, AI technology, scripting, networking, rendering, animation, cinematography, shading, textures, particle systems, procedural techniques, geometrical algorithms, performance tuning, matrices and more. This book also requires a solid understanding of C/C++ code which we have already and there are code examples for every concept that is talked about. This book also describes what a ‘transform’ is: A transform can be defined more broadly as a 4x4 matrix that left-multiplies incoming vertices to obtain transformed vertices back. This stage also manages the projection of transformed vertices to screen-coordinates, utilizing projection matrices to convert 3D coordinates into 2D coordinates that are subsequently rendered to the screen (Daniel Sánchez-Crespo Dalmau, 2003).

“The floating point: Tales of the unexpected” by David A. Faux

This paper talks about the issues with the limited precision of floating-point values have on simulations around space including planetary orbit and the distance errors associated with them. They compare both single precision 32bit floats and double precision 64bit floats, explained how having more memory or “mantra bits” allowed for increased accuracy and distance for the simulation, but also talked about how no matter what, there will always be some precision error. They also talked about how adding identical sums resulted in different outcomes when executed with the order of addition differently: In other words, $(a + b) + c \neq (a + c) + b$. Two "identical" sums are performed in the order of addition given by the summation for the other cell. This results in a difference in A that progressively propagates and causes the observed symmetry-breaking (Faux & Godolphin, 2021).

“32/64-Bit 80x86 Assembly Language Architecture” By James Leiterman
This book talks about the how a CPU computes certain actions including both single and double floating-point addition, subtraction, multiplication and division. Windows has supported 64bit instructions since Windows XP in 2005. Since the advent of the SIMD (single instruction multiple data) processors, such as the Pentium III from Intel, which powers the Xbox, and all other x86s, such as the Pentium IV and the 3DNow! extension instructions from AMD, which power PCs, the 80×86 processor has entered the realm of the supercomputer. These processors are now available in 64-bit versions, and both fixed-point (including integer) and floating-point math are used in assembly and vector-based operations in computer games and embedded environments (James Leiterman, 2005).

Existing Systems

Space Engineers

Figure 1: Space Engineers Gameplay

What is ‘Space Engineers’ about?
Space Engineers is a space sandbox game where you can build ships, and go mining, fighting, and exploring.

Strengths
Space Engineers excels at its accurate and robust physics simulation. The combat feels visceral and meaty, and ships can be split apart and deformed by collision damage. The entire world is a single instance, so no loading screens or anything like that. There is an endless abyss of depth to ship design, with many improvements that can be made to designs to make them function better at their role. The game also features an in-game scripting system, where players can write actual C# code to be compiled on and automate their ships. The game also features seamless cross-play between various PC platforms, Xbox, and PlayStation. It also features thorough mod support.

Weaknesses
Space Engineers run poorly on modern systems, and more than 16GB of RAM is required to play the game without severe stuttering. The game is ridden with bugs too. While there are no major ones now to list off, new bugs will crop up as they fix the old, and they’re too minor for most players just building stuff to notice. Additionally, the game has a global, hard-capped, explicit speed limit. While not unusual, it does put an artificial and arbitrary cap on something that should be implicitly defined at worst.

Opportunities
Time could be spent squashing bugs and optimizing the game to make it run better on modern hardware. More NPC interactions should be added (this is being worked on right now).

Threats
Space Engineers right now seems to have a focus on DLC and adding new blocks rather than polishing up the existing content. What is added feels rushed, and the update cycle for the game means that huge game-breaking bugs can exist in the game for as long as years between updates. The networking is extremely rough around the edges, and the game doesn't "support" playing on servers above 100ms of ping by default. The game also lacks NPC interaction of any degree, and space feels completely lifeless. Certain DLCs offer advantages over other players without the DLCs.

Further Research
Note: reviews were filtered to 100+ hours of playtime only.

From a review by (Dwarf-Lord Pangolin, 2019) their review mostly praises the strengths of the game as mentioned here but touches on some of the negative aspects too. Another review from (TJ_Taz, 2023) mostly focuses on the negative aspects and the focus on DLC over content. Note the lack of specifics when it comes to bugs – there are too many to focus on. Although they praise the ability of modders to fix the game. Finally, a review by Ichigo Usagi, their review is neutral and does a good job of stepping back and looking at things objectively. They praise the game for its modding support, and its ability to scratch an itch, while criticizing the game for its glaring bugs that seem to go unnoticed by many. (Ichigo Usagi, 2020).

An article posted on IGN (Andrew Hayward, 2014) talks about Space Engineers when it was released as 'early access'. Many features have been added since then, but they talk about the important parts that are still valid today: There's plenty of antagonistic fun to be had here as well; I had a great time building ships and structures and then playing around with them. I especially enjoyed teaming up with an online friend to arm our ships with warheads and then bash them together to see the explosive results (Hayward, 2014).

Conclusion
Space Engineers is an engaging, yet flawed game. There is a lot that it does right but a lot that it does wrong and doesn't sit with the player base well.

StarMade

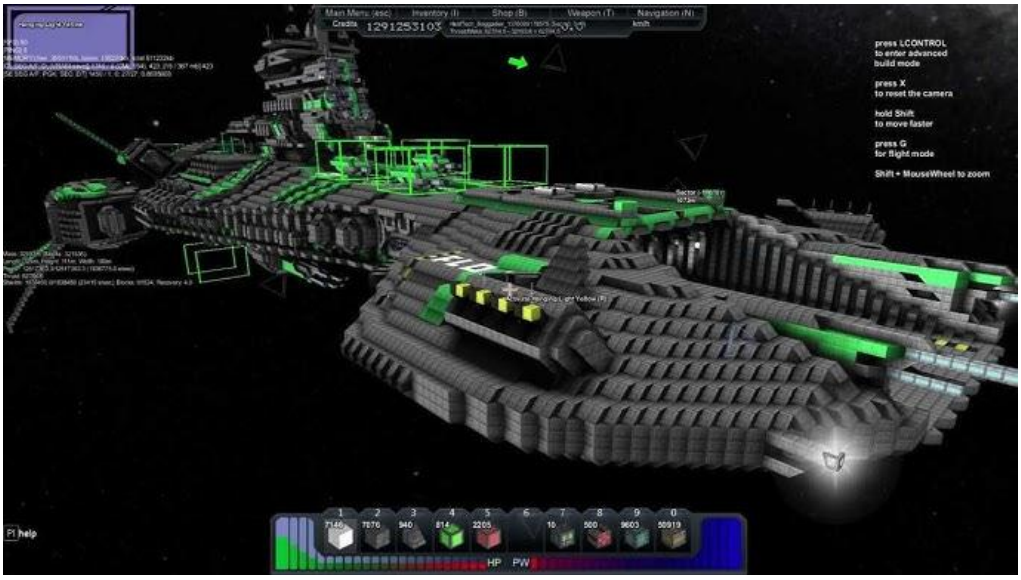


Figure 2: StarMade Gameplay

What is 'StarMade' about?
StarMade is a voxel-based spaceship-building game where you can explore an infinite, procedurally generated universe.

Strengths
StarMade, as advertised, is virtually infinite, with trillions of galaxies, containing thousands of star systems each, filled with asteroid belts and planets, distributed across sectors contained within those systems. By far, it supports the largest ship sizes out of its competitors, owing to the per-block simplicity. Ships can be kilometres long with literally millions of blocks, and the game will run – mostly – fine on modern hardware. Ships scale the best out of its competitors, due to the block systems system and adjacent systems contributing to each other. Block production is done via a resource tree, and dedicated factories can be made. In terms of modding support, there is a community-made mod loader that adds a lot of support to the game, and the StarMade Dock is full of ships for you to download. There are also placeable logic blocks, with which you can quite literally achieve anything, as this system is turing complete. Additionally, if you get your settings right, lighting can look stunning in the game. Some self-sustaining NPC factions can be interacted with, but otherwise behave as a closed system constantly running in the background, collecting resources, and building ships.

Weaknesses
The game is – as of now – dead. The developers aren't actively maintaining it (but have shown an inclination to return at some stage). There are several game-breaking bugs, and the net code can be a nightmare. There is no break apart for ships, so detached ship components will remain attached. There is also no collision damage, and the collision physics is somewhat primitive. Ship turning feels odd when scaled up, and combat feels weak and pathetic. There are some major balance concerns, in particular, missiles out scale everything else in terms of damage. NPC factions, while interesting, don't do much and send peace requests on a set timer after being attacked, taking away any kind of weight of actions against them. There is currently a clash of three different power systems in the game, two of which are legacy, which makes NPC's effectively unkillable. Exploration gets old quickly, as while everything is procedural, it feels very copy-paste after not too much time at all. Power system changes were extremely divisive in the community, with many hating them entirely.

Opportunities

It would be beneficial to the game for development to resume. More focus on niche interactions, an update of NPCs, and bug fixing would go a long way to bringing the game up to speed. A rewrite of the ship turning would be valuable too. A good look at balance would help deal with the balance issues.

Threats

StarMade features no ship break-apart, which may be appealing to some, but to most, it looks weird. Combat isn’t particularly compelling, and scaling to late game is trivial due to the existence of shops around the world. There are several gameplay loops in the game, but the primary one is mining, and that gets old fast. The lack of active developers in the game means that issues will not be fixed.

Further Research

A video made by GMODISM on YouTube (GMODISM, 2022) describes the game has a dedicated fanbase, but it’s not being maintained, but there are plenty of players waiting around for something to happen. A Reddit post was made by a player (AsXApproaches, 2024) asking about active servers, with the general response being that the game is dead, with blame pinned on the new power system. One user in a forum (Dr. Whammy, 2024) commented about how StarMade has the greatest capabilities in its niche.

An article posted on PC Gamer (Christopher Livingston, 2015) describes StarMade as: "Minecraft in space" which is a bit of a lazy term, but since I am a lazy person, I will use it anyhow. Besides, it's fairly accurate—not in a condescending, eye-rolling kind of way, but in a slightly excited one. Hey, it's Minecraft! In space! In this early access space sandbox, you mine, collect, craft, build, fight, and explore—all within a voxel-based 3D universe that is procedurally generated and appealing (Livingston, 2015).

Conclusion

The consensus with StarMade is that it has lost favour with the player base due to questionable game design choices, and a lack of maintenance from the developers.

StarBase

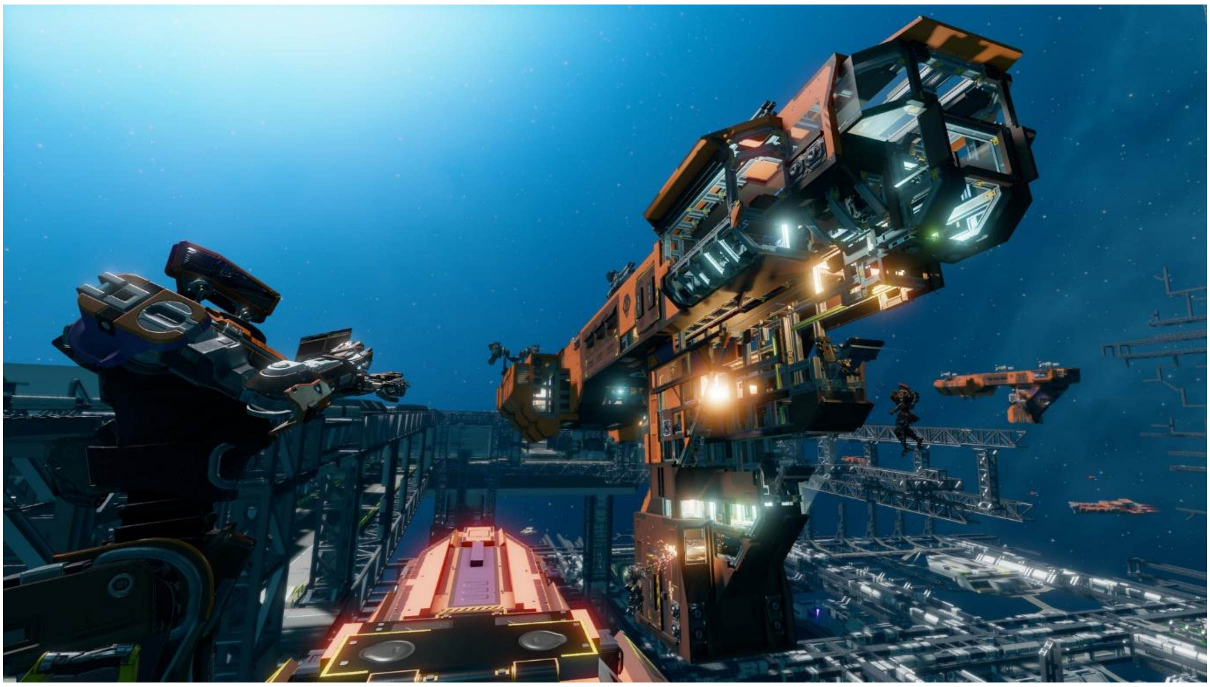


Figure 3: StarBase Gameplay

What is 'StarBase' about?
StarBase is a space-based MMO where you can build ships, pilot ships, go mining, and do various other jobs relating to ships.

Strengths
StarBase features an extremely large single-session galaxy set around a gas giant with a huge, daunting asteroid belt to enter. The world feels larger than any of the other games; while it is not technically the largest, it isn't populated by copy-paste procedural content and has player-built ships and stations scattered throughout the world to keep gameplay fresh.

It relies almost exclusively on implicit systems for navigation and IFF and faction affairs, with explicit UI's reserved for only things that would be hard to manage without.

In terms of visuals, the game is visually stunning and renders huge scenes with – relative – ease.

The game features an extremely simple scripting language called YOLOL, which is relatively simple compared to its competitors.

Ships and shipbuilding are extremely high fidelity, breaking away from the standard "block placement" and leaning more into a varying size placement system with variability down to float (possibly double?) precision. Ships must be constructed out of beams first, and then plates and components attached, wired, and piped together, and finally, the variables must be set up to get everything to interact.

Destruction of ships is high fidelity, too. Depending on the type of damage taken, blocks can either be detached from each other, "shatter" into multiple pieces, or have chunks blown out of them, rendering as voxel holes.

Weaknesses
Being an MMO, StarBase relies on player interactions for most of its content and suffers as a result if it doesn't have a player base to build on.

The graphical fidelity demands a rather powerful system as a baseline, although it scales well.

Implicit systems hurt in a lot of ways, such as being completely unable to navigate without a navigation computer running a script that links into an existing player-built navigation system.

There are no dedicated AI turret systems, and while AI systems can be shoehorned with various detection equipment, it's generally inferior to player aim, which presents an issue where solo players are completely unable to thrive, and any kind of large-scale combat requires multiple players working together, per ship.

Additionally, the building is extremely difficult to get into, with so many details to remember, and certain systems not making much sense.

To try and mitigate this, there is an easy build mode, that essentially works by snapping together prefabs of many components like Lego, however, this has the drawback of being unable to work on ships that have been modified outside of the easy ship builder and is generally much less efficient than just building the ships yourself.

Debugging and fixing issues with your ship is a pain, and very difficult to understand for new players.

As of now, the game has only recently re-entered development after being on hiatus due to a lack of funding, meaning there are a few long-standing bugs that need to be worked out.

Everything in the game is painfully slow and grindy.

Opportunities
The game could benefit from some more explicit systems. As of now, there is a map view being worked on in the public test branch, which is a good step in the right direction.

Adding in NPC interactions to build up a player base before throwing everyone together may help get the game on its feet.

Threats
Being an MMO, there is no way to play StarBase in a single-player instance or on a private server. In the TOS, they go as far as to say that you may be banned for tampering with your client to attempt to do such things. As such, even when you are alone in the void, you must be always connected to the server. Even when you're designing a ship, and you are placed in a pocket universe hosted on your computer, you are forced to connect to the server. If someone misbehaves enough to warrant a ban, rather than being handled on a server-by-server basis like in other games, that individual is completely barred from enjoying the game in the slightest.

The building system is very intimidating for most players, and players who buy the game for the building may refund it when finding out just how complex it is.

The game lacks any kind of NPC interaction, which may be a turn-off for some players, especially given that there aren't nearly enough players to make up the numbers right now.

There is nothing to stop players from "seal clubbing" new players right out of the gate.

Further Research
Steam user Gus (Gus, 2024) talks about how the builder has huge strengths, but he was unable to stay engaged due to the lack of a player base and developer support. A Reddit post (ataraxic89, 2021) has

a comment on a thread that also talks about unfinished features, issues with the easy build mode, and poor player retention.

An article posted on Rock Paper Shotgun (Craig Peterson, 2021) describes StarBase as: You start on a station in the orbit of a planet with the other players (it is multiplayer, and claims to be a single, persistent universe), and StarBase looks like one of those games you could lazily suggest throws everything but the kitchen sink at the player. However, after seeing the asset browser in the most recent trailer, I think it might even have one of those in there, too. You can code. CODE! It's very much a game where you need your big space brain engaged to make things happen (Pearson, 2021).

Conclusion

At the moment, StarBase doesn’t have an active player base, which for an MMO means 90% of its engagement. It’s also not in the best technical state, and development is going very slowly.

From the Depths

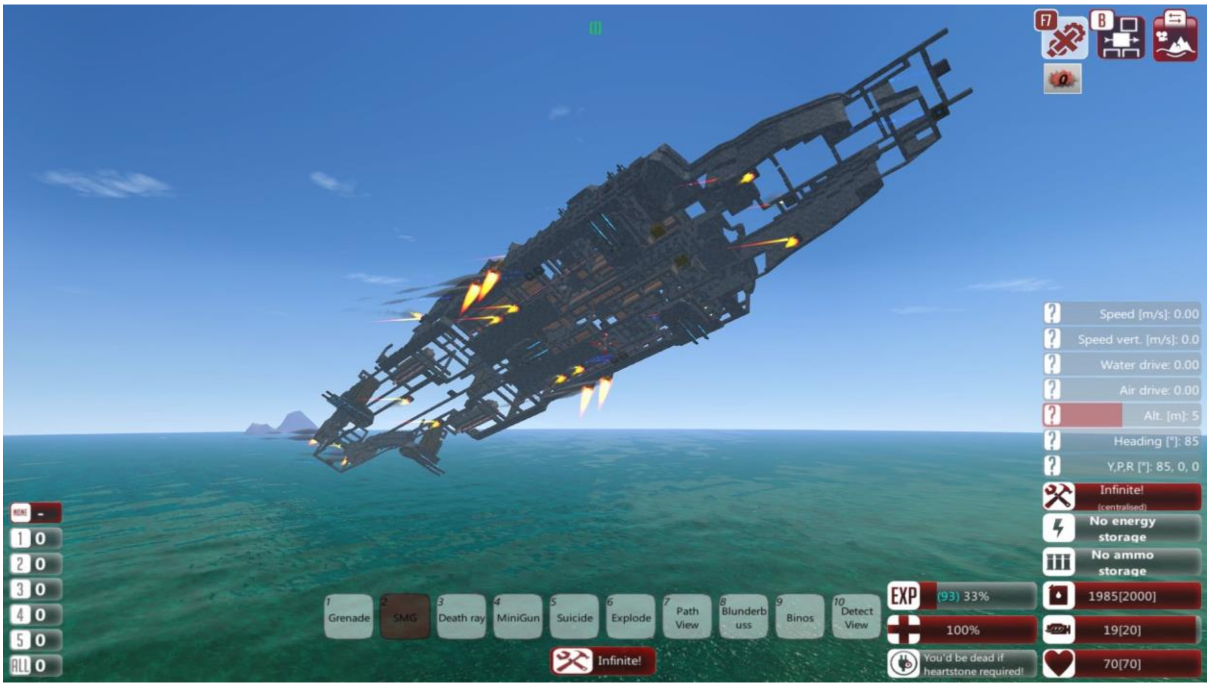


Figure 4: From the Depths Gameplay

What is ‘From the Depths’ about?

From the Depths is a craft-building game where you can build battleships, aircraft, subs, and all sorts of other combat vehicles. It has a heavy emphasis on combat, and unlike the other games, takes place primarily on the surface of a planet with a small amount of space content. It features a similar build system to other games but with radically different controls, and by far the most complex block-to-block interaction.

Strengths

There is endless tuning you can do to your craft to bring the maximum performance out of it.

The skill ceiling is very high.

Build freedom is very good – with the use of the standard block palette and mimic blocks, you can aesthetically make just about anything.

Unlike most of the other games, From the Depths has a very honed campaign and a very clear goal to the game.

There is a huge weapon variety, and it is compounded by the ability to change the weapons characteristics based on how you build them.

The campaign is an RTS in disguise – once you’ve designed and built your ships, they become units for you to command against the enemy’s units.

Weaknesses

From the Depths has a high-skill floor, with unhelpful built-in tutorials, complex controls, and very complex block interactions that can easily turn away most players picking up the game.

Damage visuals vs damage can sometimes feel underwhelming – there could be a huge explosion, but nothing gets destroyed.

Opportunities
From the Depths would benefit greatly from “survival multiplayer PVP” game mode. This would open the game to a new niche of players.

It would benefit largely from enhanced, more hands-on tutorials, rather than “place block here” and “place block there”.

Threats
From the Depths entirely lacks a “survival multiplayer PVP” mode. There is a multiplayer creative, or multiplayer co-op, but it lacks any real PvP stakes, and there is no option for dedicated survival servers.

Further Research
YouTuber Hazzor talks about how the game is a time sink and felt misled by other YouTubers making the game appear much less complex than it is (Hazzor, 2022). One Steam user praises the game for rewarding good building and creativity (oneshot2killls, 2022). Another player found the building system to be too complex (sam, 2022).

An article posted on Rock Paper Shotgun (Nathan Grayson, 2014) talks about From the Depths: I will grant that From the Depths looks ambitious. It's a first-person naval mermaidurer where you construct all your own vehicles and then fly, sail, submarine, or hot air balloon them into battle. It's also voxel-based, of course, so expect massive capital ships that appear as though they were put together by a child who hasn't learned how to capitalize letters. That's not inherently a bad thing, but it has kind of become The Style of The Time (Grayson, 2014).

Conclusion
From the Depths is a complex but rewarding game that is fairly issue-free.

Empyrion

Figure 5: Empyrion Gameplay

What is 'Empyrion' about?
Empyrion is a space sandbox survival with an emphasis on simulation, construction, survival, and first-person shooting.

Strengths
Out of the games we picked, Empyrion by and large has the best survival exploration aspect, with points of interest scattered across procedural planets.

The crafting system, while it has a complex tech tree, is very simple to craft stuff, as it crafts all the subcomponents for you.

The weapon complement is large, giving the player plenty of options.

Weaknesses
The game feels extremely unfinished. Despite having been released out of early access, the game feels like an early access game, with inconsistent art and questionable sound design.

The developers have been bashed for releasing an unfinished game and then releasing DLC.

The combat doesn't feel particularly powerful.

The game runs very poorly even on modern systems.

Opportunities

Weapons could do with better visual effects.

The whole game could use an art pass, both for visuals and audio.

Threats
Other games have more intriguing and visceral combat.

Other games are better at running larger ships while maintaining acceptable performance.

Further Research

Steam user “Stain” compares the game’s performance to several other games and makes the point that the current performance is atrocious (Stain, 2024). Another player praises the exploration aspects of the game (Rimon, 2024).

An article posted on Cyber Power PC (Ian Kane, 2017) summarizes Empyrion: All things considered, Empyrion – Galactic Survival is the ideal game for fans of science fiction without being unduly complicated. It boasts an amazing RPG-style progression system, constantly improving graphics, attentive developers, a strong, welcoming (generally) community, and open-ended, enjoyable gameplay. If you're growing weary of the same old post-apocalyptic settings, zombies, or both, you might want to give Empyrion a try; it's a lot of fun to play (Kane, 2024).

Conclusion

Empyrion, in its niche, has unparalleled exploration aspects, however, combat and performance leave a lot to be desired.

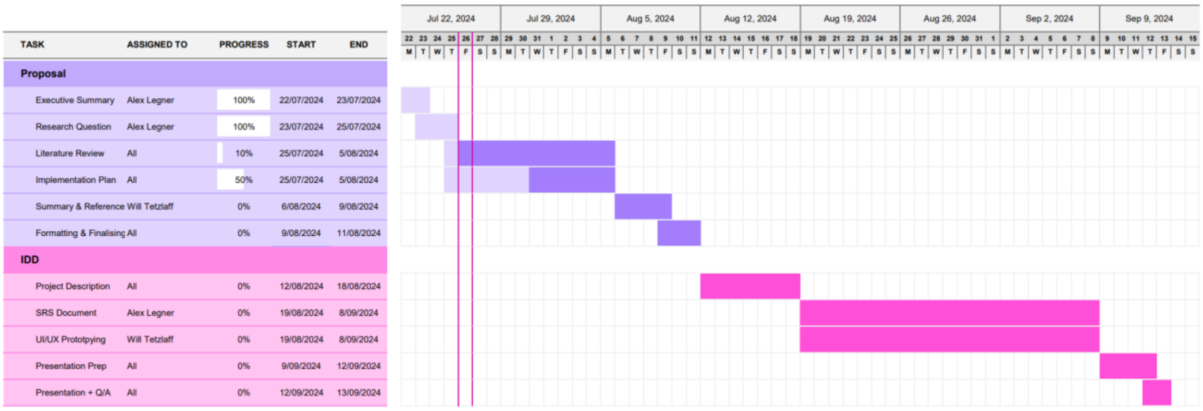
Implementation

Timeframe

The project is going to be split into 4 phases:

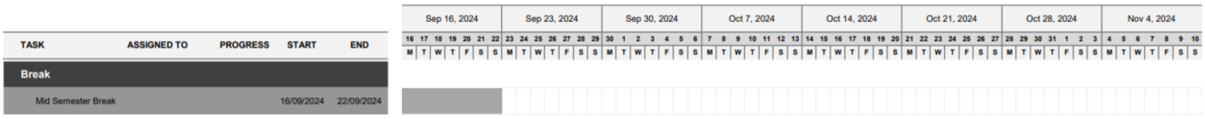
- Phase 1: Project Proposal (Week 1 -> Week 3)
- Phase 2: IDD + SRS Research (Week 4 -> Week 8)
- Phase 3: MVP Development (Week 9 -> Week 16)
- Phase 4: R&D Report (Week 9 -> Week 16)

Gantt Chart with Phases



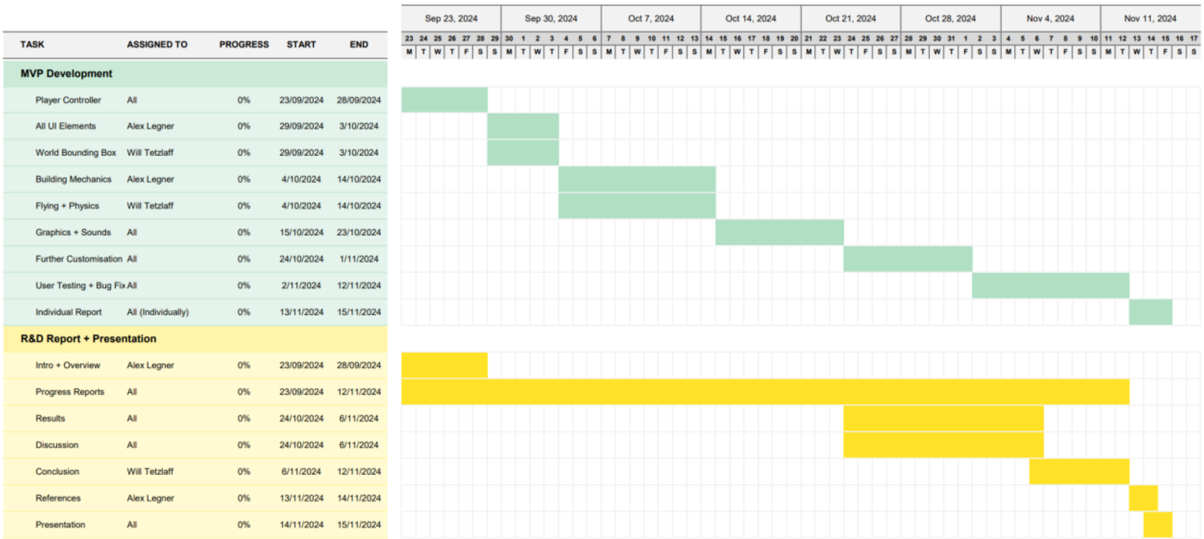
TASK	ASSIGNED TO	PROGRESS	START	END
Proposal				
Executive Summary	Alex Legner	100%	22/07/2024	23/07/2024
Research Question	Alex Legner	100%	23/07/2024	25/07/2024
Literature Review	All	10%	25/07/2024	5/08/2024
Implementation Plan	All	50%	25/07/2024	5/08/2024
Summary & Reference	Will Tetzlaff	0%	6/08/2024	9/08/2024
Formatting & Finalising	All	0%	9/08/2024	11/08/2024
IDD				
Project Description	All	0%	12/08/2024	18/08/2024
SRS Document	Alex Legner	0%	19/08/2024	8/09/2024
UI/UX Prototpying	Will Tetzlaff	0%	19/08/2024	8/09/2024
Presentation Prep	All	0%	9/09/2024	12/09/2024
Presentation + Q/A	All	0%	12/09/2024	13/09/2024

Figure 6: Gantt Chart Phase 1&2



TASK	ASSIGNED TO	PROGRESS	START	END
Break				
Mid Semester Break			16/09/2024	22/09/2024

Figure 7: Gantt Chart Break



TASK	ASSIGNED TO	PROGRESS	START	END
MVP Development				
Player Controller	All	0%	23/09/2024	28/09/2024
All UI Elements	Alex Legner	0%	29/09/2024	3/10/2024
World Bounding Box	Will Tetzlaff	0%	29/09/2024	3/10/2024
Building Mechanics	Alex Legner	0%	4/10/2024	14/10/2024
Flying + Physics	Will Tetzlaff	0%	4/10/2024	14/10/2024
Graphics + Sounds	All	0%	15/10/2024	23/10/2024
Further Customisation	All	0%	24/10/2024	1/11/2024
User Testing + Bug Fix	All	0%	2/11/2024	12/11/2024
Individual Report	All (Individually)	0%	13/11/2024	15/11/2024
R&D Report + Presentation				
Intro + Overview	Alex Legner	0%	23/09/2024	28/09/2024
Progress Reports	All	0%	23/09/2024	12/11/2024
Results	All	0%	24/10/2024	6/11/2024
Discussion	All	0%	24/10/2024	6/11/2024
Conclusion	Will Tetzlaff	0%	6/11/2024	12/11/2024
References	Alex Legner	0%	13/11/2024	14/11/2024
Presentation	All	0%	14/11/2024	15/11/2024

Figure 8: Gantt Chart Phase 3&4

Resources

We will use both our computers at home and 2 computers at our educational provider's classroom during class hours. In terms of people, both Alex Legner and Will Tetzlaff will both be committing 100% of their time to studying for this project.

There are no servers required for this project because it will be single player. If we wish to expand on this project in the future, there will be no need for servers as well as we plan to use Steam networking which is free and allows players to self-host their own games which is considered standard practice for this type of game. This is described in one article where in the gaming industry, one of the most popular methods for adding multiplayer to games is the local player-hosted game server (Henry, 2021).

Costs

There will be no costs related to the development of this project. However, If the game were to be published onto a platform – there may be a one-time fee. We aim to release this game on Steam, where they have a fee of USD 100 (NZD 167.60 as of 23/07/24) to publish the game.

All costs will be covered by the students if we so wish to publish the game.

Deliverables

The following deliverables will be created for the project at certain milestones:

Table 1: Project Deliverables

Deliverable	Description	Expected By
Proposal Document	The identification of a problem you can solve through a proper research process in a formal document.	Week 3
IDD Document	This document contains all the processes the team will follow throughout the research and development process including Project Goals/Objectives, Project Timelines, Project Constraints, Team Members and responsibilities, Software Process Model, Development Tools, Programming Languages, Technology Frameworks, an SRS Document, UI/UX Sketches, UI/UX LoFi Mock-up, UI/UX HiFi Prototype and a presentation regarding the project.	Week 8
SRS Document	Located within the IDD Document, the SRS Document contains all the specifications of the described product including all the function and nonfunctional requirements, Class Diagrams, and an activity diagram.	Week 8
MVP Prototype	A functional prototype with all features outlined in the IDD/SRS implemented that has been tested for bugs and follows good programming practices.	Week 16
R&D Report	The report covers the whole development process of the project and will reflect our understanding of the software development and research process on a week-by-week basis and a conclusion around all findings we made throughout the project.	Week 16

Ownership

As per the Yoobee Student Handbook 2024 – “Display and ownership of student work at Yoobee Colleges: The school can archive, display, or utilise any student work generated during the programme of study. Where students are working on commercial products Yoobee Colleges will not share in any monetary profits, instead will seek 'bragging rights' through media outlets.”

Both Alex Legner and Will Tetzlaff will own the project in its entirety, however as per the student handbook, Yoobee Colleges will be entitled to bragging rights where they can display the project to anyone. Yoobee Colleges may archive the project for students’ academic records. Yoobee Colleges will not be entitled to distribute this project without authorization from the owners/students.

Technical Supervisor: Ranju Raveendran

Potential Risks

Through our previous experience in game development, both of us knew of a problem that can occur in game development called floating point error. This is when objects at large distances away lose precision in accuracy which is known to cause jittering and errors. It also means that it can severely limit the world size if we are limited to floating points instead of double floating points in a game engine. This could cause serious issues further into the project if we don’t consider this early on. We will discuss this in further detail when we choose a game engine for the development of this project.

Another potential risk is that we only have 16 weeks total for this entire project. The MVP stage of creating the product is only 8 weeks out of the 16 weeks total. This is a small amount of time to have a complete product. We will schedule out what needs to be done, however, there are no extensions for this project so there that potential risk we fall behind and run out of time.

The product we want to create is going to be complex. It deals with realistic physics and multiple gameplay mechanics. If something is too complex, we are going to be expected to study it and overcome the complex issue. We also have a technical advisor who can help us. However, if something is too complicated for both of us to learn and our technical advisor, then we risk not being able to implement a specific feature in the product.

Another risk that we may end up facing could be performance issues within the game itself. Due to the scale of the project, the game could start to “lag” and run slower than required. There are ways to optimize code to increase the performance of the game, so if we do encounter performance issues, we will have to make time to optimize the code to make the game playable.

Success Criteria

To determine if this project will be a success or failure, we will refer to the deliverables of the project. If we can complete the deliverables on time and answer the question defined in this proposal in the R&D Report, then this project will be a success as we will have completed what we set out to do. If we do not have a complete MVP by the end of the project, that also doesn’t mean the project is a failure, we can still learn what we need to know from the development of the MVP and answer the question set out for this project.

With all that, we created a simple list of the success of this project:

- All deliverables completed on time.
- Answer the research question within the R&D report.
- MVP does everything outlined in the IDD and SRS.

Summary

To summarize this project, we aim to create a video game in space where you can build ships/bases, mine/refine resources, explore the void, and complete missions. We will create this game to answer the research question:

“How can realistic space physics and mechanics be balanced with intuitive gameplay to create an immersive and enjoyable experience for players in a space exploration and spacecraft piloting game?”.

This project will consist of 4 key phases:

- Proposal Document	Week 1 -> Week 3
- IDD Document/SRS Document	Week 4 -> Week 8
- MVP Prototype	Week 9 -> Week 16
- R&D Report	Week 9 -> Week 16

We have looked at existing systems of similar games and analysed them to see what they do well and what could be done better. With this knowledge and some further research in the IDD/SRS, we will create a plan to create an MVP prototype by the end of the project and answer the research question.

References

32/64-Bit 80x86 Assembly Language architecture. (n.d.). Google Books. https://books.google.co.nz/books?hl=en&lr=&id=uxkz5ZBwOWEC&oi=fnd&pg=PP1&ots=Gp31Cd9B7X&sig=u13MQ1GJTQVbknVl38Y5OUwKrwQ&redir_esc=y#v=onepage&q&f=false

A reflection on Quickfire and Power 2.0. (2024, February 16). StarMade Dock. https://starmadedock.net/threads/a-reflection-on-quickfire-and-power-2-0.31856/#post-382500

Core techniques and algorithms in game programming. (n.d.). Google Books. https://books.google.co.nz/books?hl=en&lr=&id=IfDouSUqOUIC&oi=fnd&pg=PR25&ots=h2cj0J8Gl-&sig=L9vHCKD64HO-1VZvicvv11n5tAQ&redir_esc=y#v=onepage&q&f=false

Empyrion - Galactic Survival on Steam. (n.d.). https://store.steampowered.com/app/383120/Empyrion__Galactic_Survival/

Empyrion Example Image. (n.d.). https://shared.akamai.steamstatic.com/store_item_assets/steam/apps/383120/ss_d54b8fe0ef5b0109e2001a6ed84fdf2c11aee00a.1920x1080.jpg?t=1709747913

Faux, D. A., & Godolphin, J. (2021). The floating point: Tales of the unexpected. *American Journal of Physics*, *89*(8), 806–814. https://doi.org/10.1119/10.0003915

From the Depths Example Image. (n.d.). https://www.reddit.com/media?url=https%3A%2F%2Fpreview.redd.it%2Fylshhfcl04811.jpg%3Fauto%3Dwebp%26s%3Ddc4cc900c4048e039ce8b38736cc601e15b1cad2

From the depths on Steam. (n.d.). https://store.steampowered.com/app/268650/From_the_Depths/

Game Development with Construct 2. (n.d.). Google Books. https://books.google.co.nz/books?hl=en&lr=&id=CXckDwAAQBAJ&oi=fnd&pg=PR3&ots=jmOU52GI0d&sig=im7AmXheA15lydKUDl_UqjKU5R4&redir_esc=y#v=onepage&q&f=false

GMODISM. (2022, May 12). *The current state of StarMade 2022 - Is StarMade dead?* [Video]. YouTube. https://www.youtube.com/watch?v=noC4YCVaMxQ

Grayson, N. (2014, February 19). It came: from the depths. *Rock Paper Shotgun*. https://www.rockpapershotgun.com/it-came-from-the-depths

Gunaratne, T. K. (2023). Evaluation of the use of low precision Floating-Point arithmetic for applications in radio astronomy. In *Lecture notes in computer science* (pp. 155–170). https://doi.org/10.1007/978-3-031-32180-1_10

Hayward, A. (2014, May 8). *Space Engineers: Minecraft amidst the stars? - IGN*. IGN. https://www.ign.com/articles/2014/05/08/space-engineers-minecraft-amidst-the-stars

Hazzor. (2022, July 29). *From the depths is one of the video games of all time.* [Video]. YouTube. https://www.youtube.com/watch?v=HdXOd-4bNk8

Henry, B. (2021, November 24). *Multiplayer Server Basics | Creating a multiplayer game Server - Part 1*. DEV Community. https://dev.to/ibmdeveloper/multiplayer-server-basics-ep-1-creating-a-multiplayer-game-server-5aed

I don’t think there’s any active servers. Last I heard this game was all but completely abandoned and save the [Comment on “Active Servers? Hey all, wondering if anyone’s still around playing this beauty of a game. Any active servers around?”]. (2024, July 17). https://www.reddit.com/r/Starmade/comments/1e56jfz/active_servers/

Kane, I. (2024, January 25). Empyrion – Galactic Survival (Updated Early Access) review – Galaxy-Sized Fun - CyberPowerPC. *CyberPowerPC*.

https://www.cyberpowerpc.com/blog/empyrion-galactic-survival-updated-early-access-review-galaxy-sized-fun-2/

Livingston, C. (2015, January 15). *Exploring StarMade, a voxel-based space sandbox*. Pcgamer. https://www.pcgamer.com/exploring-starmade-an-voxel-based-space-sandbox/

Page, M. J., Moher, D., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., . . . McKenzie, J. E. (2021). PRISMA 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews. *BMJ*, n160. https://doi.org/10.1136/bmj.n160

Pearson, C. (2021, July 29). Starbase's early access space race has begun. *Rock Paper Shotgun*. https://www.rockpapershotgun.com/starbase-early-access-release-1

PRISMA statement. (n.d.). PRISMA Statement. https://www.prisma-statement.org/

Space Engineers Example Image. (n.d.). https://assetsio.gnwcdn.com/space-engineers-is-finally-heading-to-xbox-one-in-april-1582240603802.jpg?width=1200&height=630&fit=crop&enable=upscale&auto=webp

Space engineers on steam. (n.d.). https://store.steampowered.com/app/244850/Space_Engineers/

StarBase Example Image. (n.d.). https://nichegamer.com/wp-content/uploads/2020/02/starbase2-10-20.jpg

Starbase on Steam. (n.d.). https://store.steampowered.com/app/454120/Starbase/

StarMade Example Image. (n.d.). https://i.ytimg.com/vi/PTmtUsYu2A4/hq720.jpg?sqp=-oaymwEhCK4FEIIDSFryq4qpAxMIARUAAAAAGAElAADIQj0AgKJD&rs=AOn4CLAbh_oaCrk8vYIYNuv8k0TgcDuGEQ

StarMade on Steam. (n.d.). https://store.steampowered.com/app/244770/StarMade/

Steam Community :: Dwarf-Lord Pangolin :: Review for space Engineers. (n.d.). https://steamcommunity.com/profiles/76561198018141136/recommended/244850/

Steam Community :: Gus :: Review for Starbase. (n.d.). https://steamcommunity.com/id/kurangasu/recommended/454120/

Steam Community :: Ichigo Usagi :: Review for space Engineers. (n.d.). https://steamcommunity.com/id/ichigo_usagi/recommended/244850/

Steam Community :: oneshot2killls :: Review for From The Depths. (n.d.). https://steamcommunity.com/profiles/76561198853960799/recommended/268650/

Steam Community :: Rimon :: Review for Empyrion - Galactic Survival. (n.d.). https://steamcommunity.com/profiles/76561197989217232/recommended/383120/

Steam Community :: sam :: Review for From The Depths. (n.d.). https://steamcommunity.com/id/sam4413/recommended/268650/

Steam Community :: Stain :: Review for Empyrion - Galactic Survival. (n.d.). https://steamcommunity.com/profiles/76561198163879882/recommended/383120/

Steam Community :: TJ_Taz :: Review for space engineers. (n.d.). https://steamcommunity.com/id/TJ_Taz/recommended/244850/

Sudarsanam, Y. (2007, March 8). *Implementation of double precision floating point arithmetic*. https://repository.lib.ncsu.edu/items/74cd3b91-e8fc-4c8b-a1af-205f32914be7

What the hell happened? [Online forum post]. (2021). Reddit. Retrieved August 2, 2024, from https://www.reddit.com/r/starbase/comments/s0nklx/comment/hs2uvti/

Yoobee Colleges. (2024). *Yoobee Student Handbook 2024*. Retrieved August 2, 2024, from https://myacg.sharepoint.com/sites/YoobeeCollegesStudentHub/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2FYoobeeCollegesStudentHub%2FShared%20Documents%2FStudent%20Handbook%202024%20v2%2Epdf&parent=%2Fsites%2FYoobeeCollegesStudentHub%2FShared%20Documents