NULL\REFERENCE

Proxima Centauri

Final Design Report

Team Members:	**Project Roles**:
Aidan Harries	Team Leader
Ryan Zimmerman	Game Designer, Audio Designer, Programmer, Quality Assurance
Riley Gronewoller	Game Designer, Programmer
Drew Valenzuela	Art Director, Audio Designer, Quality Assurance
Jadyn Maloney	Art Director, Programmer, Quality Assurance
Jared Gastner	Game Designer, Programmer, Quality Assurance

December, 13 2023

Executive Summary

Team NULL\REFERENCE is pleased to present the final design report for our game, Proxima Centauri. This arcade-style game, developed for the Windows PC platform, builds on our initial vision of blending the feel of classic arcade games with modern gameplay elements, making it an enjoyable experience for both seasoned and new gamers alike.

In this report, we detail the entire development process of Proxima Centauri. Our driving force was to fill a noticeable gap in the gaming market for retro-style games that attract a wide range of players. We identified early on that there was a desire for this type of game, and we aimed to fulfill that demand. Our success criteria remained consistent - a game that not only looks good but also plays well. From the initial conceptual sketches to alpha testing and beta releases, every stage was meticulously planned and executed. With clear requirements and a robust testing plan in place, we've transformed our early designs into a fully-fledged game ready for release.

Our technical design section breaks down the game's core components, explaining our decisions every step of the way. We relied heavily on feedback and a weighted criteria matrix to finalize aspects like control schemes, game modes, and enemy behaviors. As we navigated through the development process, we frequently returned to our initial design principles, ensuring that we stayed true to our foundational ideas. This section provides a bridge between our early designs and the final product, outlining how we've refined our ideas over time.

The work plan and roadmap have been updated to reflect our progress since the preliminary report. This process was essential, acting as our guiding light throughout development. Here, we detail the tasks completed, the challenges faced, and the steps taken to overcome them. Unforeseen challenges did arise, as they do in any project, but our team's adaptability ensured that we remained on course. Our financial plan remains largely unchanged, with a few adjustments to account for any unforeseen expenses.

Risks are inevitable in any project, and Proxima Centauri was no exception. In our feasibility assessment, we revisit the challenges we anticipated and discuss how we managed or circumvented them. This not only provides a record of our problem-solving efforts but can also serve as a learning tool for similar projects in the future.

Lastly, in our lessons learned section, we take a moment to reflect on our journey. We've grown as a team and as developers, realizing that collaboration is the key to overcoming obstacles and reaching our goals. This project has taught us invaluable skills that we will carry forward, both in professional and personal capacities.

In conclusion, as we wrap up the development of Proxima Centauri, our focus has been and will always be on creating a memorable gaming experience for our audience. Our journey was filled with ups and downs, but our commitment never wavered. We believe that with our dedication, teamwork, and clear vision, we have successfully brought our project to life.

Table of Contents

Project Description

Background and Motivation

To understand the context and motivation for our project, we conducted a thorough literature search, reviewing sources such as academic journals, books, industry reports, and technical standards. These sources provided us with valuable insights into the gaming market, player preferences, and the evolution of arcade-style space games.

Our research revealed that there is a growing interest in retro-style games, particularly among the millennial and Gen Z demographics, who appreciate the combination of nostalgic aesthetics and modern gameplay mechanics (Garda, 2017). Additionally, we discovered that, despite the rise of more complex and realistic games, there remains a strong demand for accessible and enjoyable arcade-style games that can be played by both casual gamers and experienced players (Wulf, 2018).

Identifying this opportunity in the gaming market, our project aims to design and develop an arcade-style game that combines the nostalgic appeal of classic titles with modern gameplay mechanics and graphics. This will allow us to create a unique and engaging experience that caters to both nostalgic and new players.

Additionally, this project serves as a learning experience for our team, allowing us to sharpen our skills in various aspects of game development, such as programming, art, sound design, and project management.

Problem Statement

There is a distinct lack of new arcade games that cater to the rising demand for pixel-art genres and successfully appeal to both new and nostalgic players in the current gaming market. The success of our project will be evaluated based on its ability to design a game that garners favorable player feedback for its captivating gameplay and nostalgic aesthetics.

Project Requirements

These project requirements will serve as a basis for evaluating the success of our game. As the project progresses, we may refine and update these requirements to ensure the final design meets our goals and objectives.

Table 1 - Project Requirements

ID	Project Requirement	Description
1.0	Engaging Gameplay	**Primary Functional Requirement:** Gameplay that can retain user interest for an average of 15 minutes or longer per session and suitable for all skill levels.
2.0	Nostalgic Appeal	**Primary Functional Requirement:** Incorporate 16-bit pixel-art graphics reminiscent of classic arcade games.
3.0	Accessibility	**Subfunction**: Intuitive controls that allow users to start gameplay within 1-2 minutes and levels with increasing difficulty to challenge both new and returning players.
4.0	Platform Compatibility	**Subfunction**: Native support for Windows 10.
5.0	Performance Optimization	**Subfunction**: Game should run at a minimum of 30fps on systems with standard specifications (i.e., Intel i5 7th gen, 8GB RAM) without any lag.
6.0	User Interface (UI)	**Subfunction**: Clear, organized menu structures with visual feedback, allowing players to navigate in 3 clicks or fewer.
7.0	Game Controls	**Subfunction**: Immediate response (<0.1s) to player input with a concise control tutorial available.
8.0	Audio Design	**Subfunction**: Minimum of 5 distinct arcade sound effects and 3 unique retro-style background tracks, ensuring variety and engagement.
9.0	Scoreboard	**Subfunction**: Real-time high score tracking, with the ability to store at least the top 10 scores for competitive gameplay.
10.0	Low Development Cost	**Objective/Design Goal:** Ensure that the game's development cost remains under a predefined budget (~$50).
11.0	Development Timeline	**Objective/Design Goal:** Game launch by December 5, 2023, with at least one beta release by November 9, 2023.
12.0	Replayability	**Objective/Design Goal:** Pseudo-random enemy spawns for varied challenges in each playthrough.

Project Scope

Our project, Proxima Centauri, centers around creating an immersive, arcade-style game that captivates players with engaging gameplay, nostalgic aesthetics, and broad accessibility. To accomplish this, our team will combine our unique skills and creativity, with established tools and resources throughout the development of our game.

Original Contributions:

1. **Game Concept and Design**: We will develop the overall idea, game mechanics, and progression system.
2. **Game Programming**: Our team will write the code for the game's core mechanics, controls, and interactions.
3. **Custom Art & Audio Assets**: We will create original pixel-art graphics, animations, and retro-style audio.

Borrowed Resources:

1. **Game Engine**: We will use Unity, a powerful commercial game engine, to develop the game.
2. **Programming Language**: Our team will write code using C#.
3. **Version Control**: Our team will utilize GitHub for version control.
4. **Asset Creation**: We will use Aseprite for creating custom pixel-art assets and animations.
5. **Audio Design**: To design retro-style audio, we will employ tools such as BFXR, LMMS, NES VST, and DefleMask.
6. **External Libraries**: We will leverage existing libraries for specific functionalities, such as physics or collision detection.
7. **Tutorials and Online Resources**: We will use online tutorials and forums to refine our skills and learn best practices.

Validation and Acceptance Tests

To ensure the game's success and alignment with our goals and requirements, we have devised a set of validation and acceptance tests. These tests will assess various game aspects, from its performance to its user experience. Through this, we aim to identify and rectify any potential shortcomings, ensuring a high-quality player experience.

Below are the tests designed for a comprehensive evaluation of critical game components:

1. **Internal Testing**
    a. **Objective:** Evaluate the game's overall experience, difficulty, and enjoyment.
    b. **Method**: The project team will play the game, focusing on its mechanics, controls, and fundamental functionality.

2. **Platform Compatibility Testing**
    a. **Objective:** Ensure game stability and smooth operation on Windows PCs.
    b. **Method:** The game will be tested exclusively on the Windows 10 operating system to confirm compatibility.

3. **Performance Optimization Testing**
    a. **Objective:** Ensure smooth gameplay across diverse hardware setups.
    b. **Method**: The game will be tested on at least 10 varied hardware configurations. This includes setups with different processors (from Intel i5 to i9, AMD Ryzen 5 to Ryzen 9) and varying RAM sizes (8GB to 32GB) to ensure optimization.

4. **Accessibility Testing (Alpha Testing)**
    a. **Objective:** Validate the game's appeal and accessibility across different player proficiencies.
    b. **Method:** A group of at least 15 testers, comprising friends and family members of various ages and gaming proficiencies, will evaluate the game. They will gauge the game's learning curve, controls, and difficulty progression.

5. **Visual and Audio Testing (Beta Testing)**
    a. **Objective**: Assess the integration of nostalgic appeal through pixel-art graphics and retro-style gameplay, and evaluate the game's sound effects and music for immersive and engaging qualities.
    b. **Method**: A group of at least 30 testers, including classmates and instructors from the ECE 591 and CIS 580 classes, will play the game. They will provide feedback on the visual and audio aspects of the game.

6. **Playtesting and Feedback Gathering**
    a. **Objective:** Gather comprehensive feedback on all aspects of the game to understand user experience and identify areas of improvement.
    b. **Method:** After testers have played the game, they will be prompted to fill out a feedback survey. This survey will encompass a range of questions related to various facets of the game.
    c. Sample Survey Questions:
        i. Did the game run successfully on your PC setup? (Yes/No)
        - If no, please specify the issues encountered.
        ii. Rate your overall engagement with the gameplay.
        - 1 (Very Poor) - 5 (Very Good)
        - Elaborate on your rating (optional)

iii. **Rate the nostalgic appeal of the game's visuals and themes.**
- 1 (Very Poor) - 5 (Very Good)
- Elaborate on your rating (optional)

iv. **Evaluate the game's accessibility in terms of control intuitiveness and difficulty scaling.**
- 1 (Very Poor) - 5 (Very Good)
- Elaborate on your rating (optional)

v. **Additional Feedback**
- Please provide any additional feedback or suggestions that you believe would improve the game.

7. **Statistical Measures**
   a. **Objective:** Stay within budget and adhere to the timeline.
   b. **Method:** The project team will continuously monitor expenses and progress, adjusting strategies as required to meet these constraints.

Table 2 - Verification Matrix

ID	Requirement	Requirement Verification Method			
		Similarity	**Inspection**	**Analysis**	**Test**
1.0	Engaging Gameplay				X
2.0	Nostalgic Appeal	X			X
3.0	Accessibility				X
4.0	Platform Compatibility				X
5.0	Performance Optimization			X	X
6.0	User Interface (UI)		X		X
7.0	Game Controls				X
8.0	Audio Design		X		X
9.0	Scoreboard				X
10.0	Low Development Cost			X	
11.0	Development Timeline		X		
12.0	Replayability				X

Technical Design

Pre-Conceptual and Conceptual Design Descriptions

Possible Solutions and Design Alternatives

Given the project requirements and goals, we explored various design alternatives for our game. The primary focus was on achieving an engaging gameplay experience while maintaining the nostalgic appeal of classic arcade games. Keeping this in mind, the following design alternatives were considered:

1. **Control Schemes**: We explored different control schemes, such as keyboard, mouse, and game controller. Each having its advantages and drawbacks, the chosen control method influences the overall game experience.

2. **Movement Mechanics**: We considered various movement mechanics, including traditional 2D side-scrolling, top-down view, and 360-degree movement. Each one affecting the game's complexity and how the player interacts with the environment.

3. **Game Modes**: To cater to different player preferences, we looked into incorporating various game modes, such as story mode, endless mode, and single game mode. These modes each provide a unique experience and replay value for the player.

4. **Power-ups and Upgrade System**: We analyzed different power-up and upgrade systems, including permanent upgrades, temporary power-ups, and a combination of both. They each have an effect on the player's progression and overall game balance.

5. **Enemy Types and Difficulty Scaling**: We explored different enemy types, attack patterns, and difficulty scaling methods. These factors would determine the challenge and variety of gameplay experiences for the player.

Selection Process for Preliminary Design Solution

To select the best design solution, we used a weighted criteria matrix to evaluate and compare each alternative based on the project requirements and goals. The criteria included factors such as gameplay engagement, ease of use, nostalgia factor, replayability, and development complexity.

Table 3 - Weighted Criteria Matrix

Criteria	Gameplay Engagement	Ease of Use	Nostalgia Factor	Replayability	Development Complexity	Total Score
Weight	**0.3**	**0.2**	**0.2**	**0.15**	**0.15**	
Control: Keyboard	4	5	3	5	5	**4.30**
Control: Mouse	3	4	3	3	3	**3.20**
Control: Game Controller	4	3	4	3	3	**3.50**
Movement: 2D Side scrolling	3	4	2	3	5	**3.30**
Movement: Top-down	4	5	4	3	4	**4.05**
Movement: Forward propulsion only	5	3	3	5	4	**4.05**
Movement: Inertia based movement	5	5	3	4	3	**4.15**
Movement: Screen wraparound	5	3	5	4	3	**4.15**
Movement: 360°	5	4	4	5	4	**4.45**
Game Mode: Story	4	4	3	2	3	**3.35**
Game Mode: Endless	4	4	3	5	2	**3.65**
Game Mode: Single	5	5	5	3	5	**4.70**

Upon evaluating each design alternative, we found that a combination of the following elements best satisfied our project requirements and goals:

1. **Control Scheme**: In addition to keyboard controls, which were chosen for their simplicity and familiarity among players, mouse and controller inputs were also integrated. This combination aims to cater to a wider range of player preferences and offers more diverse ways to interact with the game.

2. **Movement Mechanics**: A combination of top-down and 360-degree movement was selected to provide a more engaging and dynamic gameplay experience, incorporating elements like forward propulsion only, inertia based movement, and screen wraparound for interesting movement mechanics while still preserving the classic arcade feel.

3. **Game Modes**: A single game mode with progressive difficulty was chosen to keep development complexity low and focus on delivering a polished gameplay experience.

4. **Power-up and Upgrade System**: Contrary to our initial plan of including both permanent upgrades and temporary power-ups, we decided to only implement a system of permanent upgrades. This approach emphasizes strategic long-term planning and player progression, as opposed to the short-term benefits provided by temporary power-ups.

5. **Enemy Types and Difficulty Scaling**: A variety of enemy types with increasing difficulty and unique attack patterns were chosen to maintain player interest and provide slow-building challenge for new players.

The chosen design solution strikes a balance between engaging gameplay, nostalgic appeal, and development feasibility. This design was chosen over other alternatives due to its potential to satisfy both the project requirements and player expectations, while still providing a unique and enjoyable gaming experience

Preliminary Design Description

Global Issues

In the design and development of our game, we must consider not only the technical aspects of gameplay but also the broader, global issues that may impact the overall success of our project. These global issues can range from technical considerations, like device compatibility, to more socially-oriented aspects, such as cultural sensitivity and accessibility. Addressing these issues in our project plan ensures that our game is not only fun to play, but is also considerate of a diverse player base and varying device configurations. In this section, we will identify and discuss these issues and outline our strategies for addressing each.

1. **Device Compatibility**
    a. **Summary**: Our game, being a digital product, is intended for the Windows PC platform. The game should be compatible with all Windows PC systems regardless of their underlying hardware configurations.
    b. **Resolution**: We will ensure that our game is optimized to run on a wide range of hardware configurations, including older and lower-end devices. This will increase the game's accessibility to a wider range of users.

2. **Language and Cultural Sensitivity**
    a. **Summary**: Although our game is primarily for a school project, it will still be showcased to a broader audience, including people from various cultural backgrounds. It's essential to be mindful of different cultural sensitivities.
    b. **Resolution**: We will ensure our game's content does not include any material that could potentially be offensive or insensitive to different cultures.

3. **Accessibility**
    a. **Summary**: Not all potential users of the game may have the same level of physical ability. It's important that the game is accessible to as wide a range of users as possible.
    b. **Resolution**: We will implement intuitive controls and a difficulty curve that is accommodating to players of all skill levels. We will also consider features like adjustable text sizes and color-blind-friendly design elements.

4. **Energy Efficiency**
    a. **Summary**: The energy consumption of the game on users' devices can vary based on the game's optimization.
    b. **Resolution**: We will optimize our game's code to minimize energy consumption. This includes efficient use of processing power, which can also contribute to smoother gameplay.

System-Level Overview

Our game, Proxima Centauri, is a fast-paced, arcade-style game inspired by classic space shooters like *Galaga* (1981) and *Space Invaders* (1978). The goal of the game is for players to defend their home planet from increasingly challenging waves of enemies. The game integrates elements of strategy, skill, and luck with a variety of ship and planet upgrades to enhance gameplay. The game is grounded in the principles of classic arcade game design, with a focus on responsive controls, engaging gameplay loops, and replayability.

Our design follows a modular approach, breaking down the entire game into distinct stages or modules. Each module has a clear, single purpose and can be designed, developed, and tested independently before being integrated into the game. This approach promotes a clean and efficient design process, allowing us to tackle each aspect of the game separately while ensuring they all work together to create an engaging player experience.

Applicable Standards

In developing Proxima Centauri, our team followed a set of relevant standards and guidelines that are crucial for both game development and a quality user experience. This section outlines the key standards that have been applied:

1. **Game Development Standards:** We adhere to industry-recognized game development practices, particularly focusing on modular development, which is critical for maintaining a structured and efficient workflow. This approach ensures each component of the game, from the Main Menu to the High Score Menu, is developed, tested, and integrated systematically.

2. **Gameplay Design Standards:** Inspired by classic arcade games, our design strictly follows the principles of arcade-style gameplay. This includes prioritizing simple yet challenging mechanics, intuitive control schemes, and progressive difficulty levels, all essential for recreating the classic arcade experience in a modern context.

3. **User Interface (UI) Design:** Our UI design conforms to standard practices that emphasize clarity, ease of navigation, and minimalistic design, ensuring that players have a seamless and intuitive interaction with the game's various modules.

By integrating these standards into our development process, Proxima Centauri not only honors the legacy of classic arcade games but also meets the current expectations for quality, performance, and accessibility in the gaming industry.

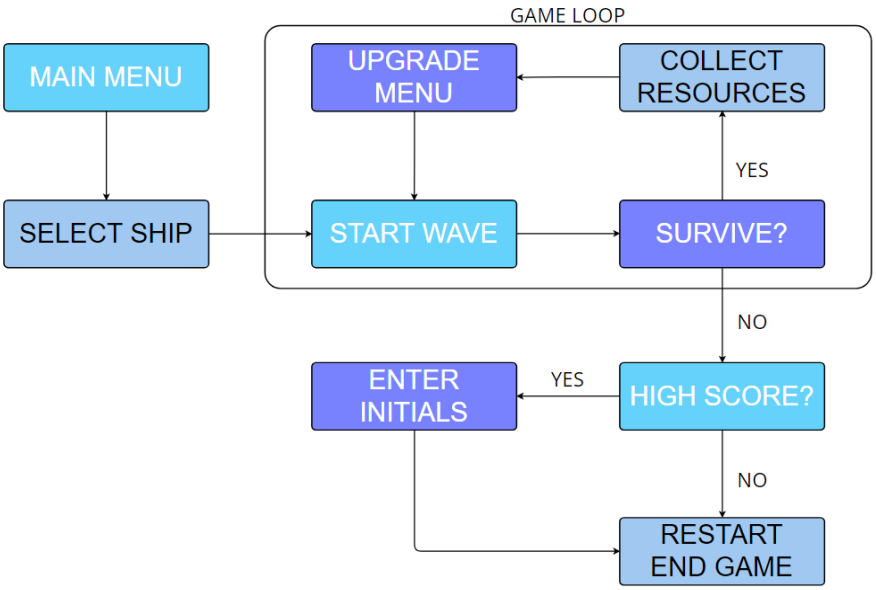

Figure 1 - Game Flow

Module-Level Descriptions

1. **Main Menu Module:**
    a. **Inputs**: Player's choice (start, high scores, settings, quit)
    b. **Outputs**: Activation of corresponding module
    c. **Function:** Acts as the game's entry point, allowing players to start a game, view high scores, change settings, or exit.

2. **Ship Selection Module:**
    a. **Inputs**: Player's choice of ship
    b. **Outputs**: Selected ship information
    c. **Function:** Presents a variety of ships for the player to choose from, each with a unique style. The selected ship information is then passed to the Gameplay Loop Module.

3. **Gameplay Loop Module:**
    a. **Inputs**: Selected ship information
    b. **Outputs**: Detailed Game State (ship health, enemy status, score, upgrades, resources)
    c. **Function**: The core gameplay area where players battle enemy waves. It tracks the game state, including health, enemy progress, score, and resources, directing to either the Game Over Module or Resource and Upgrade Selection Module based on the player's performance.

4. **Resource and Upgrade Selection Module:**
    a. **Inputs**: Game state, collected resources
    b. **Outputs**: Selected upgrades, remaining resources
    c. **Function**: Allows players to use their gathered resources to purchase upgrades for their ship and home planet. The module then updates the Gameplay Loop Module with these enhancements for the next wave.

5. **Game Over Module:**
    a. **Inputs**: Game state, collected resources
    b. **Outputs**: Final score, High Score check
    c. **Function**: Activated when the player's ship or planet is destroyed. It calculates the final score from the game state and resources. The module then checks if this score qualifies for the High Score list. If so, it prompts the player to enter their initials and adds the score to the list, followed by displaying the High Score screen. If not, it offers options to restart the game or return to the Main Menu.

6. **High Scores Module:**
    a. **Inputs**: Final score (from Game Over Module)
    b. **Outputs**: Updated high score list
    c. **Function**: Updates the high score list with new entries from the Game Over Module. After updating, it allows players to return to the Main Menu, providing a seamless transition between the end of one session and the start of another.

Detailed Design Descriptions

1. **Main Menu Module**

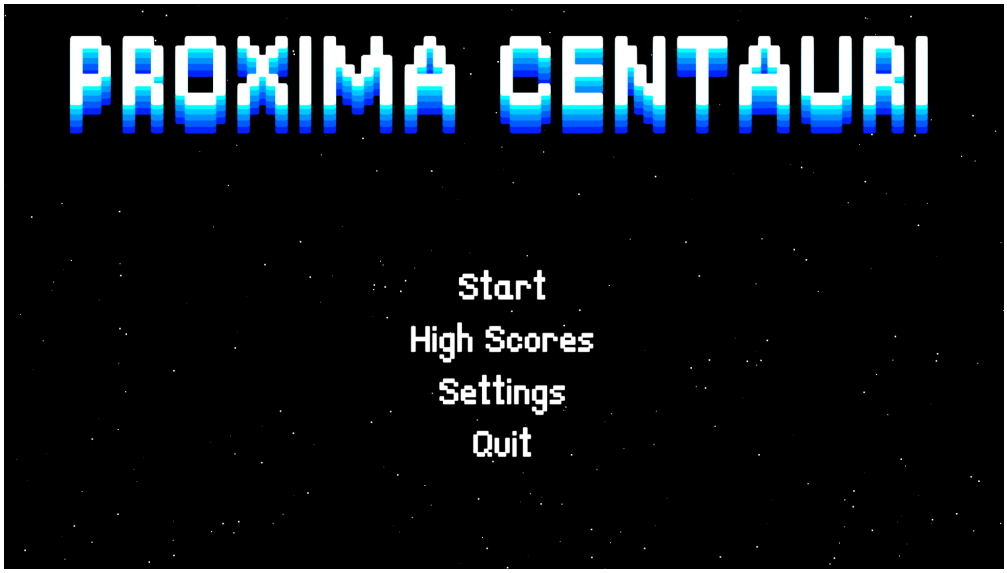

Figure 2 - Main Menu

Overview:
The Main Menu Module serves as the starting point of the game. Here, players can initiate a new game, view high scores, adjust settings, or exit the game.

Interface Layout:
- **Title:** "Proxima Centauri" displayed prominently at the top.
- **Options:**
  - **Start:** Begins the game.
  - **High Scores:** Opens the leaderboard displaying the top ten high scores.
  - **Settings:** Opens the game's settings interface where players will eventually be able to customize various game settings.
  - **Quit:** Closes the game application.

2. Ship Selection Module

Figure 3 - Ship Selection

Overview:
Players are presented with three aesthetically distinct ships to choose from. The chosen ship is then used for gameplay. The functionality of each ship is consistent, with only visual differences setting them apart.

Interface Layout:
- **Background:** A space-themed backdrop showcasing the three ships.
- **Ships**: Displayed side by side.
- **Press ENTER to Start**: Press enter to select currently highlighted ship.

3. Gameplay Loop Module

Figure 4 - Gameplay Loop

Overview:
In this module, players control their selected ship to battle through successive waves of enemies. After each wave, players are taken to the Resource and Upgrade Selection Module, where they can use collected resources to upgrade their ship and planet. This loop of combat and upgrades continues, with the difficulty escalating as the player progresses.

Interface Layout:
- **Background:** Parallax starry space backdrop with a central planet.
- **Player's Ship:** Placed at the game's start, free for navigation.
- **Enemy Waves:** Spawn off-screen, moving towards the player or the planet.
- **Ship & Planet Health Bars**: Bottom left.
- **Resource Counter**: Top left, showcasing accumulated scrap.
- **Total Score**: Displayed in the top right.
- **Wave Counter**: Displayed at the top-center of the screen at the beginning of each wave.

4. **Resource and Upgrade Selection Module**

Figure 5 - Upgrade Menu

Overview:
After successfully navigating through a wave, players can utilize collected scrap to upgrade either their ship or the planet. This helps to improve defenses and increase survivability.

Interface Layout:
- **Ship Upgrades**: Menu listing all potential ship enhancements, showing available levels.
- **Planet Upgrades**: A separate section detailing enhancements for the home planet.
- **Resource Availability**: Displayed prominently, showing the amount of scrap a player has for upgrades.

Ship Upgrades
- **Movement Speed**: Level 1-4 | Boost ship speed incrementally.
- **Shooting Speed**: Level 1-4 | Enhance shooting speed.
- **Armor Plating:** Level 1-4 | Reduce incoming damage progressively.
- **Max Ship Health**: Level 1-4 | Expand max health capacity.

Planet Upgrades
- **Defense Turrets:** Level 1-4 | Turrets that statically shoot at enemies.
- **Resource Multiplier:** Level 1-4 | Amplify resource collection rate.
- **Moon:** Level 1-4 | Moon that orbit the planet and help block enemy damage.
- **Max Planet Health:** Level 1-4 | Expand max health capacity.

Resources
Scrap is the primary currency in this universe. Collect scrap to pay for upgrades and defend the planet. These can be obtained by defeating enemies and completing waves.

5. Game Over Module

Figure 6 - Game Over

Overview:
Triggered when the player's ship or the planet is destroyed. It calculates the session score based on gameplay metrics and introduces players to the high score input if they've achieved a high score.

Interface Layout:
- **Game Over Screen:** Details about the player's performance, including total score and waves completed.
- **High Score Prompt:** If applicable, allows players to input initials for the leaderboard.

6. High Scores Module

Figure 7 - High Scores

Overview:
The High Scores Module serves as the game's leaderboard system. This module showcases the top-ten highest scores that have been achieved on the local machine.

Interface Layout:
- **Title:** "High Scores" Prominently displayed.
- **Top Ten High Scores**: A list presenting the top-ten scores achieved on the local machine, along with the associated player initials.

7. **Setting Module**

Figure 8 - Settings

Overview:
This module is still in development, but eventually it should allow players to customize gameplay mechanics and controls, catering to their preferences.

Interface Layout:
- **Control Selection:** Options for keyboard, mouse, or game controller inputs.
- **Movement & Shooting**: Dropdowns or toggle switches to adjust controls.
- **Audio & Graphics**: Additional settings to tweak game appearance and sound.

Validation and Acceptance Test Results

Objective:
To ensure that the game "Proxima Centauri" meets all the design and functional specifications, delivering an optimal user experience and meeting performance expectations.

Verification:

All verification testing was conducted internally by individual team members on their personal devices.

Engaging Gameplay
- Avg. wave reached by player in the first 20 minutes: **8.3 Waves**
- Avg. session duration: **27.4 minutes**

Nostalgic Appeal
- Alignment with classic arcade styles: **Rated 3/5**

Accessibility
- Avg. time taken for players to grasp mechanics: **2 minutes**
- Difficulty progression based on player progression: **Rated 2/5** (Needs more balance)

Platform Compatibility
- Compatibility checks on various Windows PC setups: **100% compatible**

Performance Optimization
- Avg. frame rate on standard setups: **58.7 fps**
- (Tested across 10 different hardware configurations)

UI Survey
- Navigation ease assessment: **Rated 4/5**

Control Latency
- Avg. measured control response times: **108.34 milliseconds**
- (tested across 10 different hardware configurations)

Audio Quality
- Sound clarity and theme assessment: **Rated 0/5**
- (Audio had not been added yet)

Scoreboard Test
- Avg. time taken for player understanding of scoreboard: **1 minute**

Validation:

All validation testing was conducted externally by two sets of playtesters:
- 15 Alpha Testers: Comprised of close friends and family.
- 30 Beta Testers: Comprised of the students and instructors in ECE 591 and CIS 580.

Engaging Gameplay
- Player feedback on gameplay engagement: **Rated 4.5/5**

Nostalgic Appeal
- Alignment with arcade nostalgia as per player feedback: **Rated 4.2/5**

Accessibility
- Feedback on intuitive controls: **Rated 4.8/5**
- Game challenge and balance assessment: **Rated 4.6/5**

Platform Compatibility
- Reported Windows PC compatibility issues: **None**

Performance Optimization
- Player feedback on performance: **Rated 4.8/5**

UI Survey
- Player feedback on interface appeal: **Rated 4.5/5**

Control Latency
- Player feedback on responsiveness: **Rated 4.6/5**

Audio Quality
- Player rating of music and sound effects: **Rated 4.7/5**
- (Audio had since been added)

Scoreboard Test
- Player understanding of scoring mechanics: **Rated 4.7/5**

Assessment of Test Results

"Proxima Centauri" has shown commendable performance in our validation and acceptance tests, closely aligning with our design and functional goals. The gameplay is engaging, and the nostalgic elements resonate well with our target audience, as shown by positive feedback from both internal and external testers. Accessibility features and the difficulty progression have been well-received, indicating a balanced experience for players of varying skill levels.

In terms of technical performance, the game has demonstrated excellent compatibility across various Windows PC setups, a critical aspect of our development goals. The performance optimization, particularly in terms of frame rate and loading times, has met our expectations, contributing to a smooth gaming experience for most players. And the intuitive user interface, combined with responsive controls, has been a highlight, enhancing the overall playability and immersion of the game.

The audio quality, especially the background music, has been praised for adding depth to the game's atmosphere. And the results from the Scoreboard Test indicate that players can easily understand and engage with the game's scoring mechanics.

Overall, we are very pleased with the outcomes of our testing phases for "Proxima Centauri." The results affirm that the game has successfully met our envisioned standards and objectives.

Work Plan

Work Breakdown Structure (WBS) & RACI Chart

Throughout the development of our project, we will utilize the Work Breakdown Structure (WBS) & RACI chart to provide clear task assignment and efficient workflow among team

members. The WBS is a hierarchical breakdown of the project into detailed tasks, which helps in clarifying what needs to be done and allows us to assign responsibilities more efficiently.

The RACI chart is a simple matrix that matches tasks with team members, defining their roles in relation to each task. RACI is an acronym that stands for Responsible, Accountable, Consulted, and Informed. However, for the purposes of this project, we will only be using 'Responsible' (R) and 'Assisting' (A) to represent the roles of team members. 'Responsible' indicates the person who carries out the task, while 'Assisting' refers to those who provide support or aid in the task's completion.

Here's the breakdown of the high level tasks and responsibilities for our game:

Table 4 - WBS & RACI Chart

Task#	Task	Aidan	Ryan	Riley	Drew	Jadyn	Jared
1	**Concept & Design**	R	A	A	A	A	A
1a	Develop Game Concept	R	A	A	A	A	A
1b	Design Game Mechanics	A	A	R			
1c	Design User Interface	A	R				
2	**Art & Audio**	A	R		A	R	
2a	Create Game Artwork	A			A	R	
2b	Develop Game Audio	A	R		A		
3	**Programming**	R	A	R		A	A
3a	Program Game Mechanics	A	A	R		A	A
3b	Program User Interface	A	A			A	R
3c	Program Enemy AI	A		R		A	A
4	**Testing & Quality Assurance**	R	A	A	A	A	A
4a	Test Game Mechanics	A		R			
4b	Test User Interface	A		A	R		A
4c	Test Enemy AI	A	A	R			
5	**Finalize & Submit Project**	R	A	A	A	A	A

(R = Responsible, A = Assisting)

Roles:

- **Aidan Harries**: Team Leader
  - Will oversee the entire project, ensuring a smooth operation and timely task completion. Will also be assisting with every aspect of the project.

- **Ryan Zimmerman**: Game Designer, Audio Designer, Programmer, Quality Assurance
  - Responsible for designing the user interface and developing the game audio, as well as, assisting with programming and quality assurance testing.

- **Riley Gronewoller**: Game Designer, Programmer
  - Responsible for designing and programming the game mechanics and enemy AI, and assisting with the overall game design and quality assurance testing.

- **Drew Valenzuela**: Art Director, Audio Designer, Quality Assurance
  - Responsible for testing the User Interface, and assisting with both the art and audio design of the game.

- **Jadyn Maloney**: Art Director, Programmer, Quality Assurance
  - Responsible for directing art design, and assisting with the overall development and programming.

- **Jared Gastner**: Game Designer, Programmer, Quality Assurance
  - Responsible for programming the user interface and assisting with both quality assurance and game design.

Gantt Chart

The following is a Gantt chart representation of the project timeline for the development of our game. The Gantt chart serves as a visual representation of our project timeline, showing the start and end dates of the individual tasks and their duration. It provides a clear snapshot of the project schedule, helping us to manage and allocate our resources effectively.

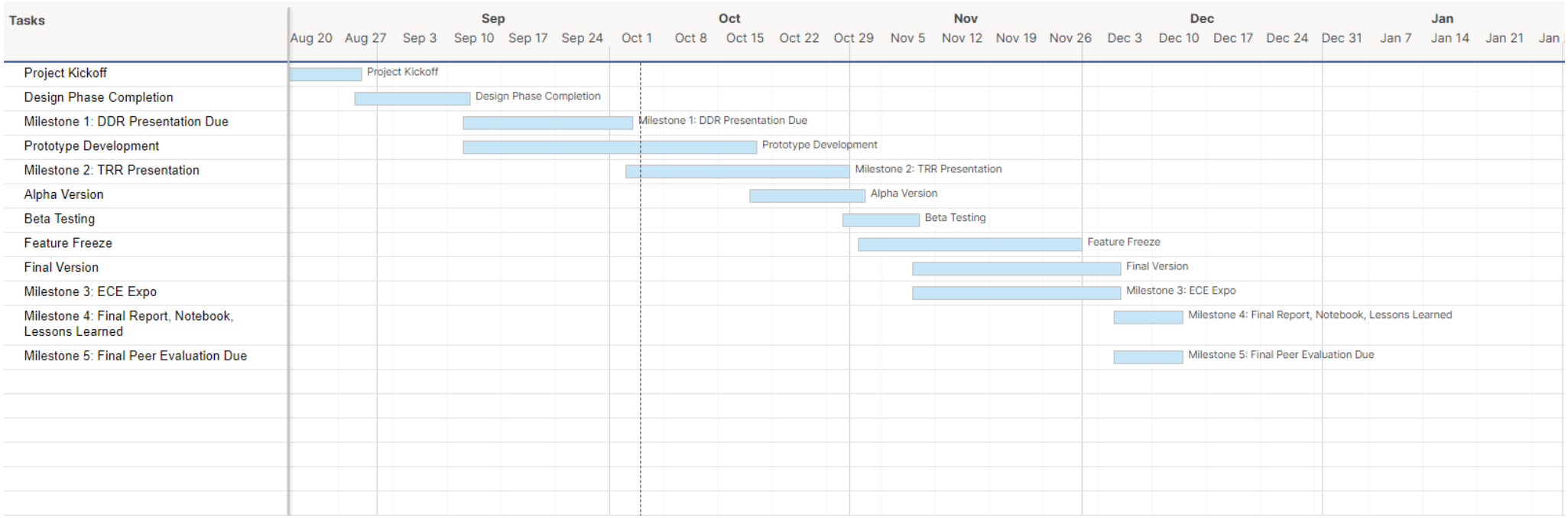

Figure 9 - Gantt Chart

Prototyping and Testing Protocol

Proxima Centauri will undergo a thorough prototyping and testing process to ensure that it provides the best possible gameplay experience. This process will take place in phases, each phase focusing on different aspects of the game.

Prototyping

The prototyping phase will be iterative, where each module of the game (UI, game mechanics, enemy AI, upgrades, and scoreboard) will be developed and integrated one after the other. This approach allows us to test each component separately before integrating it into the larger game system.

Testing

The testing phase will occur concurrently with the prototyping phase, with each module undergoing testing after it is developed. The testing phase consists of two main stages:

1. **Module Testing**: Each module will undergo unit testing after it is developed. This stage will involve testing each function of the module independently to ensure that it works as expected.

2. **Integration Testing**: After all modules have been developed and unit-tested, they will be integrated and tested as a complete system. This stage involves testing the game as a whole to ensure that all modules work together properly and that the game runs smoothly.

Throughout the testing phase, we will create a bug report documenting any issues encountered during testing, along with their corresponding solutions. This will be important for troubleshooting and ensuring the improvement of our game.

Test Environment

The testing will be conducted on Windows-based computers, which is our targeted platform for the game. Our objective is to confirm that the game runs smoothly across a wide variety of Windows PC setups. To achieve this, we are aiming to test on at least 10 different PC configurations. This will include variations in hardware, screen resolution, and system performance.

Financial Plan

Our financial strategy for this project is centered on utilizing resources that are freely available, while keeping some provision for potential costs that might arise during the game development process. Given that the Unity game engine and other tools we are using are free, we have a significant amount of flexibility and contingency in our budget. Should the need for additional third-party assets arise, we are prepared to adjust our plan and allocate funds accordingly. However, the main cost for this project is time and labor that our team will invest.

Proposed Budget

The budget below provides a line-item detail of the materials needed and their associated costs:

Table 5 - Proposed Budget

Item	**Tool/Software**	**Cost**
Game Engine	Unity Game Engine	$0
Version Control	GitHub	$0
Asset Creation	Aseprite & Clip Studio Paint	$0
Audio Design	Audacity, BFXR, LMMS, NES VST	$0
Third-Party Assets	TBD	$50
Total		**$50**

Note: Third-party assets represent an estimated cost for potential purchases of art and sound assets. The actual cost may vary depending on specific needs that arise during the development process.

Proposed Budget vs Final Expenditures

As the development of "Proxima Centauri" continues, we've been diligently monitoring the expenditures in relation to our initial budget. Here's a current breakdown of the spending:

Table 6 - Final Expenditures

Item	**Tool/Software**	**Cost**
Game Engine	Unity Game Engine	$0
Version Control	GitHub	$0

Asset Creation	Aseprite	$0
Audio Design	BFXR	$0
	DefleMask	$9.99
Third-Party Assets	All assets made in-house	N/A

Initial Budget: ~$50

Total Expenditures: $9.99

Remaining Budget: ~$40

Given the current progression, the project has utilized a minimal portion of the set budget. We anticipate future expenditures primarily for potential third-party assets or additional tools if needed. The current in-house approach to asset creation has proved cost-effective.

Feasibility Assessment

After evaluating our project, we created a list of risks associated with the creation of our game. These risks are inherent in any game development project. However, we've outlined mitigation strategies for each of them to ensure our project's success.

1. **Technical Risks**: Integrating graphics, sound, and gameplay mechanics can lead to unforeseen technical challenges. This includes compatibility issues between different components, and difficulties in coding and programming.

   Mitigation Strategy: Regular testing will be performed throughout the development process to ensure ideal interaction between all game components. Additionally, we will conduct regular code reviews to ensure the code follows best practices and maintains implicity and readability.

2. **Resource Risks**: Project success relies on the skills and availability of our team members. The risk here lies in potential changes in team composition or a lack of necessary skills or experience to complete certain tasks.

   Mitigation Strategy: Collaborate with our team members and support each other when difficulties arise. Regular communication and updates regarding individual progress and challenges will help the team adapt and overcome any issues.

3. **Scope Risks**: Unclear or changing project scope can affect the schedule. These risks could originate from modified requirements or unforeseen development hurdles.

Mitigation Strategy: Strictly following the established project plan is essential. Any changes should only be considered if they are critical to the project's success. We will have regular team meetings to discuss and reassess the project scope to help manage these risks effectively.

By acknowledging these risks and setting up proactive measures to address them, we're confident that our project is feasible and positioned for success. But risk management is an ongoing process, and we must stay adaptable throughout the project's duration.

Lessons Learned

Dealing with the challenges of game development, we've learned numerous lessons that reshaped our approach and perspectives. Sharing these insights not only helps us reflect on our journey but also to offer guidance to future teams embarking on similar projects.

1. **Scope Management**: In the initial stages of our project, the excitement of creating a video game led us to brainstorm an abundance of ideas to enhance the gameplay. However, we soon realized the importance of managing the project's scope. We learned to focus on creating a minimum viable product (MVP) before delving into additional features. This approach allowed us to establish a solid foundation for our game, and will remain an important lesson as we continue throughout the development process.

2. **Effective Communication**: Another valuable lesson was understanding the significance of effectively presenting our work to audiences. Early on, our presentations were filled with text and disproportionate emphasis on certain topics. We learned that clarity and balance were key to effective communication. Consequently, we simplified our slides to concise bullet points, finding a balanced discussion across all aspects of our project. This change not only made our presentations more engaging but also ensured a comprehensive understanding of our project.

Reflecting on our project, if we were to start over, we would concentrate on defining the game's essential mechanics from the beginning. This would ensure a solid base to build upon as the project progresses. Additionally, we would dedicate more time to preparing our presentations to ensure they're concise, clear, and comprehensive.

To anyone embarking on a similar project, we would advise managing the project scope diligently and focusing on developing an MVP first. Don't let the excitement of potential features cloud your focus on the core functionality. Also, invest time in developing your communication skills. Being able to present your project effectively is equally as important as the development work itself.

Conclusion

Our game development process has been both exciting and challenging, pushing our creativity, technical skills, and collaboration. Our team's experience of developing this game has exposed us to many different aspects of game design and the dynamics of working in a team.

Over the course of our project, we have had to navigate various challenges, from technical difficulties to managing the project's scope and effectively communicating our progress. Each challenge has served as a good step towards our growth, both as individuals and as a team. We learned to focus on the essential, streamline our work, and communicate effectively, all while maintaining a creative and shared passion for our project.

Our proposed game has taken shape into a project we are proud of. Our chosen game engine, Unity, has provided a flexible and powerful platform for realizing our vision, and our careful planning has ensured that we will move forward in a structured, efficient manner.

As we finish the development of our game, we will carry the lessons learned and experiences gathered so far. We are excited about the road ahead, aware of the challenges it may bring, but confident in our ability to navigate them. In conclusion, we are proud to deliver a game that resonates with our vision, provides an engaging experience for players, and stands testament to our learning and growth in game development.

References

Garda, Maria B. “Nostalgia in Retro Game Design.” University of Lodz, 2017.

Hulaj, Rame, et al. “A Motivational Model Explaining Performance in Video Games.” *Frontiers*, 5 June 2020, www.frontiersin.org/articles/10.3389/fpsyg.2020.01510/full

Team, GameDev.tv, et al. “C# Unity Developer 2D Coding: Learn to Code Video Games.” Udemy, www.udemy.com/course/unitycourse/. Accessed 9 May 2023

Wulf, Tim, et al. “Video Games as Time Machines: Vide Game Nostalgia and the Success of Retro Gaming.” Cogitatio, 2018.