Vision Document

Austin Cheng, James Davis, Jeremy Ethridge, Andrew Fung, Huy Le, Juan David Lopez, Helen Pabst, Alex Tol

Section 1:

- **Purpose**: The goal of this document is to provide the scope and purpose of our Seize Command game.
- **Scope**:This document will describe the creation of our Seize Command game.
- **Definitions, Acronyms, and Abbreviations**:
  - AI - Artificial Intelligence
  - NPC - Non-Player Character
  - UI - User Interface
  - PC - Personal Computer
  - CPU - Central Processing Unit
  - RAM - Random Access Memory
- **References**:
  - Game AI Pro
  - Introduction to Algorithms
  - An Introduction to Machine Learning
  - SteamSpy.com
- **Overview**: This document is organized into several subsections, each encompassing a specific aspect of the overall project. The structure of the article is as follows: business definition, an overview of the product and its features, constraints and other requirements of the product, and the documentation requirements/appendix.

Section 2: Positioning

- **Business opportunity:** Seize Command project will tap into both space-sim and sandbox game markets. Big audiences exist for both of these markets as can be seen by the success of games like *Everspace* and *Space Engineers*.
- **Problem Statement:** Despite being one of the most asked for genres by gamers, sandbox space-sim games either promise you the world and deliver only its surface or are locked in perpetual development. The lack of many good products in this specific niche is slowly killing off interest for it, which may extinguish the genre all together. A good solution for this problem would be for the introduction of a simple but deep 2D sandbox space-sim. The introduction of such a game will reignite enthusiasm for the genre and extend its lifespan to allow for the introduction of other games.
- **Product Position Statement:** For the gamer looking to buy a game that fits the sandbox space-sim niche but doesn't want to buy a poorly made game or wait for Star Citizen to come out. Seize Command will provide gamers with a high intensity, sandbox space game that will include space combat and boarding other ships along with a beautiful map and seamless transitions between the character and ship views. In addition there will be

a multiplayer included for the gamers who wish to work with a team and compete against NPCs or other players.

Section 3: Stakeholder and User Description

- **Market Demographics:** The demographic that this game would be specifically targeted towards would be the Space Simulation players. Action, adventure, and strategy video games set in space are on the rise, with games such as Star Conflict and Fractured Space having been purchased by over two million unique accounts through Steam alone as well as a game called Kerbal Space Program holding a 95% approval rating, ranking among the highest in the entire Steam library (https://steamspy.com/tag/Space+Sim). With the increase in popularity of space sims, this is the perfect time to introduce a unique game to the market that encompasses multiple genres while still set in space.With this in mind, we will be able to attract the attention of a variety of video game players as well as the bulk of the space sim players.

- **Stakeholder Summary:**
  - Name: User of the product.
  - Description: Those who play the game or are interested in the game.
  - Role: The only possible role they would play is if they were alpha or beta testers.
- **User Summary:**
  - Name: PC Video Game Player
    - Description: Someone who plays video games, from simple games such as Bejeweled to billion dollar franchises such as Assassin's Creed. A unique and intense game such as this will appeal not only to space sim fans, but extend to action and strategy fans as well. Additionally, the PC gaming community are very supportive of “indie” games such as this one.
    - Stakeholder: User of Product
- **Stakeholder Profiles:**
  - Representative: Assuming we get to this point in the scope of the project, the majority of users would be represented by alpha and beta testers.
  - Description: User of Product. The users constitute the primary stakeholders of the project.
  - Responsibilities: Reporting bugs and game crashes back to the developers, which is an optional responsibility.
  - Success Criteria: Success would be determined by user happiness with the product, which would be accomplished by eliminating bugs and presenting an entertaining game.
  - Involvement: Testing the game pre-release would be the extent of stakeholder involvement with the development of the product.
  - Comments or Issues: Due to this being a school project, it is unlikely that we will be releasing the game for alpha and beta testing.

- **User Profiles:**
  - PC Video Game Player:
    - Representative and Description mentioned previously.
    - Type: Will range from “casual” to “hardcore” gamers and may have limited knowledge of game design. May over criticize certain aspects of the game when comparing to other games on the market.
    - Responsibilities: Optional pre-release testing and bug reporting.
    - Success Criteria: The game is entertaining and has few, if any, bugs.
    - Involvement: No further involvement beyond optional pre-release testing and bug reporting.
    - Comments or Issues: Due to this being a school project, it is unlikely that we will be releasing the game for alpha and beta testing.
- **Key Stakeholder or User Needs:**
  - The users and stakeholders need an entertaining game that could keep their attention for dozens of hours while expecting the game to be stable and consistently playable.
- **Alternatives and Competition:**
  - There currently is no alternative to our game on the market, as it is a unique blend of multiple genres and contains exclusive gameplay elements. There is, however, a wide variety of space simulation video games currently on the market as well as a large following for upcoming games, such as Star Citizen. These games are a major source of competition for our product.

Section 4: Product Overview

- **Product Perspective**
  - This game will be powered by the Unity engine which will be the only dependency. The sandbox space-sim genre is one of the most in demand genres on the market today, the game will fare well in a niche that’s hungry for a new member.
- **Summary of Capabilities:**

Customer Benefit	Supporting features
The game will include an easy to understand tutorial which will help the player learn the basics of the game.	The tutorial level will be coded using the same assets that will be in the co-op and team deathmatch modes. These assets will represent a small slice of the game which will be a beginning practice tool for the player.

The game will have an easy to use and effective UI for the player to control a ship or their player character.	Unity contains an in-engine library to generate UIs for the player. Our team will use these libraries to make simple and robust UIs for the player to use.
The game will feature seamless transitions from the ship to character views and vice versa. This will maintain the player’s immersion in the game which can be easily broken by jarring transitions.	Unity has an in-library object for the player-camera. Manipulating this object with the right animations will achieve a smooth and instantaneous camera transition.
The player and his/her teammates will be able to board enemy ships and, if successful, scrap the enemy ship for parts or completely assume command of it.	This will be a matter of using the C# scripts in Unity. Once an enemy ship has been disabled (depleted shield banks and disabled engines), the player will be able to activate a function which permits their ship to dock with the enemy and allow the players to board it. Ship capturing will also be achieved by a C# script which triggers after the player has been at the enemy ship’s helm for at least 30 seconds.
The game will include sound effects to enhance player immersion.	Unity supports the vast majority of sound files which our team can interweave within the scripts for believable and satisfying sound effects.
The game will also feature NPCs which will present the players with a satisfying challenge and keep them coming back for more.	The NPC models will be create by using a pixel art editor and then importing the created assets into Unity. NPC behavior will be handled by C# scripts.
Players will be able to post their scores onto an online leaderboard.	We will be using a combination of HTML and CSS to create a website for the game that is home to an online database which stores player usernames and scores.

Assumptions and Dependencies

- Assumptions
    - The project requirements will remain constant.
    - Our team will be able to effectively use the Unity engine.
- Dependencies
    - Unity engine for the game.
    - Internet connection for the multiplayer

Cost and Pricing

- Possible purchase of internet database usage.

Licensing and Installation

- Unity has a feature which exports an existing project into an .exe file. Installation is achieved by running the .exe file. Licensing for sale can be achieved by buying the professional version of Unity. However since this is a school project, that will not be necessary.

Section 5: Product Features

- **Multi-Player Support**
    - Our product will target those looking for a cooperative and competitive experience and will therefore require Multi-Player networking support. We anticipate that a large amount of troubleshooting with regard to server and network configurations will need to take place considering our teams lack of experience in the field. Unity provides easy setup for Multi-Player features which should speed up research related to Networking.
- **Interactive Desktop Application Provided by Unity Engine**
    - Unity 2D provides easy to use build settings allowing for a Desktop application to be created with the click of a button. There could exist optimization issues with regard to different versions of the windows operating system. Ideally we will have to iron out these issues for specific versions and decide which others will be deemed incompatible or “too old.”
- **Enemy Ship Boarding**
    - One of the main selling points of this product is the ability for a team to board an enemy ship. The point of this action is to provide a “high risk high reward” situation that a player's team can undertake. This function offers players to make a vital choice while playing: Do they destroy the enemy ship and move on or risk salvaging for better equipment.
- **Clear and Straight-Forward Tutorial**
    - New players will have the option to experience a tutorial mode which will bring them through the main functions of the game.

- **Non Player Characters**
    - Non Player Characters(NPC) will be included as enemies for the players to overcome. Our team will be developing a baseline Artificial Intelligence(AI) for the NPC to function off of. The complexity of the AI will likely vary from enemy to enemy ranging from very basic fighters to full sized enemy frigates.
- **Sound Effects**
    - Spaceships have a variety of functions and necessities that will require accurate sound effects. For instance, functions such as a ship firing its weapons, activation of thrusters and player interactions would be required to have sound effects. It is also important to consider that no sound effects should sound too similar to any other. This would likely vary from ship to ship.
- **Animations**
    - Much like sound effects, animations will be needed to accompany most player driven functions within the game. Uniqueness of each animation is key so that one function is not mistaken for another. The visual accuracy of each animation will be a challenge and research into Unity's Particle Effects will need to take place.
- **Game Modes**
    - **Team Death Match**
        - Team Death Match is a team vs team game mode where strategy and team player mechanics will determine victory over the other team.
    - **Cooperative**
        - In Cooperative, one team takes on the enemy fleet of AI. The player team starts off with a very low end ship. The goal is to continue to get stronger via better parts and ships. Should a team acquire enough strength. They can take on the AI centralized space station in order to win the game.
- **User Interface**
    - The player's User Interface(UI) will be intuitive and allow for seamless interaction with the player's ship and controls. The UI should also be uncrowded so as to enhance the user's visual experience as much as possible
- **Leaderboards**
    - Players will be able to compare their scores with other players via the player leaderboards which would display personal statistics. Some of the statistics would be but not limited to: username, score, date.
- **Database**
    - A database will be required in order to store leaderboard statistics as well as in game items values. Some of these values might include but are not limited to: weapon damage, ship health, engine speed.
- **Webpage**
    - Our team will make a website to allow for user registration as well advertising for the product. In addition, player leaderboards will be displayed on the website for

players to view. Finally, the game will be available to download on our company webpage.

Section 6: Constraints

- **Design**
    - The game should have the design and appearance of a 2D video game set in outer space. The design defines the game genre and shapes the type of gameplay utilized. We would need to immerse the player into the game and have them feel as though they were in space by utilizing quality scenery and graphics, rather than having the player feel as though they were simply playing a video game.
- **External constraints**
    - The software should be able to run on a Windows system.

Section 7: Quality Ranges

- **Performance**
    - The software should be able to run consistently across a variety of computers with a trivial amount of resources used, including CPU, RAM, and disk space. The latency between user input and response ingame should be minimal, and should have an adequate frame rate.
- **Robustness**
    - The software should run smoothly while it is operating. In the case of this game, we need to ensure that each feature of gameplay has an aspect of robustness to it. We will need to avoid bugs in the programming that will allow for invalid actions to be performed or performance issues.
- **Fault tolerance**
    - The software should be designed to minimize faults and issues during runtime. Since it is a standard video game, we will need to search for issues and resolve them as soon as possible.
- **Usability**
    - The software should be developed with usability in mind.

Section 8: Precedence and Priority.

- *Authenticity*
    - Careful attention will be paid to subtle details of the game, for it is the sum of such details that adds to the realism of a game. These details will include the movement of enemy ships and other NPCs. Movement should mimic that of an entity that is aware of its surroundings and reacts accordingly to events; the role of artificial intelligence algorithms will be crucial. Considerable attention will be lent to background graphics, sound effects, and character entity art to add to the surrealism and the ambience of the game.

- *“Game feel”*
    - Part of what makes any game great is a lack of bugs, glitches, or errors that impact the game. User satisfaction via entertainment is the goal of any game. Games without interruption lead to a seamless playing experience, helping a player to fully submerge himself or herself into the game. Bugs and other related errors will stop gameplay and lead to disgruntled users.
    - The game’s user interface gives a full view of all important features that are present in futuristic spacecraft, like radars to detect enemy ships and the ship’s status. This mimicry of a futuristic spacecraft will add to the realism of the game.
    - It will be important to also include valid, accessible game controls. The keyboard in this case would be a user’s ship controls. Giving full control in situations that seem life-like is what lends to the entertainment aspect of a game; a player is able to perform actions they otherwise cannot do in reality, but given the illusion that they are fully contained in the real realm.

Section 9: Other Product Requirements

- Windows system is required for the software to run.
- The Co-op mode needs at least more than 1 player up to 4 players.
- The software needs an internet connection and a browser to play online and to access the online leaderboard on our website.

Section 10: Documentation Requirements.

- A simple page of instructions/manual will be available for each mode of the game (basic control and objective).
- The game should have a rules page for players (hacking, cheating, verbal violation, …).
- The team also needs to identify and list out the right and owner of all the resources for making this software.