Galleon Games

A pixel art of a ship

Description automatically generated

**Corsair Clash for Windows Request for Proposal Version 1.0**

**Cole Vanderlaan, Shaun Christie, Spencer Reed, Shreeya Pradhan, Palak Pandey, Hongxi Zhu**

RFP Software Engineering

Table of Contents

1. Problem description
2. Project Objectives
3. Current System(s) – if any or similar systems
4. Intended users and their interaction with the system
5. Known interactions with other systems inside/outside the client organization
6. Known constraints to development
7. Project Schedule
8. How to Submit Proposals
9. Dates
10. Glossary of terms
11. Problem description / opportunity / expression of need

There should be more joy in the world. We can help meet that demand by creating an engaging video game. The rapid advancement and advancement of technology in game production has driven traditional game-making processes to evolve. A paradigm shift is required. Smaller teams of developers are creating an increasing number of games. People who can bring their ideas to reality without the creative constraints that working for huge game companies frequently comes with perks. This liberty enables developers to connect on a much more personal level with their users.

The gaming industry is witnessing a surge in demand for innovative and competitive titles that challenge players' strategic thinking and decision-making skills. Auto Battler Games have gained popularity due to their unique combination of strategy and automation, offering an experience that appeals to both casual and hardcore gamers. However, the market lacks a standout auto battler game that offers a captivating narrative, diverse characters, and an immersive gaming environment. We seek to address this gap by developing an Auto Battler Game that redefines the genre with innovative gameplay and compelling storytelling.

1. Project Objectives

The Objective Is to create an auto-battler strategy game that gives users an extensive variety of play styles. It is to Create a compelling and immersive storyline that draws players into the game world, enhancing their emotional investment and long-term engagement. Sometimes referred to as “Auto-Chess”, for its grid-like level design and a near infinite amount of move combinations allowing for nearly every game played to be different. With an addition of upgradable attacks, ships, and items, add another layer of variables to further guarantee the uniqueness of each encounter.

* + Game Dashboard Design
    - Overworld Map
    - Battle Map/Maps
    - Obstacle Assets to Implement in Battle Maps
  + Levels
    - Difficulty scale increases as the player progresses
    - Populates maps with enemies and objects randomly
  + Player
    - Keep the record of current stats
    - Able to move around the map
    - Able to choose enemy to attack
    - Can interact with objects
  + Overworld enemies
    - AI Enemy Ship
    - AI Enemy Ship Variants
    - Able to attack player inside the battle radius
    - Overworld enemy readout
  + Ship
    - Ship Upgrades
    - Restart On Player Death/Sink
    - Health
  + Battle Physics
    - Damage triggers for Enemy and Player Ships
    - Animation For Enemy and Player Attacks
    - Sprite Movement
  + Weapons
    - Different Guns
    - Explosives
    - Blades
  + Sound
    - Attack SFX
    - Environment SFX
    - Background Soundtrack
    - Battle Theme
    - Gun sound
    - Explosive sound
  + Main menu/UI
    - A settings menu can be accessed in game to include a help screen
    - Player can access inventory in game
    - Sound volume Mixer
    - New Game Option
    - Resume Game Option
    - Toggle for audio on/off

1. Current system(s) – if any / similar systems

**What are Auto Battlers (aka Auto Chess)?**

They are a video game type established in the strategy genre. Where you are generally matched with one opponent be it a player or an AI. The objective, as well as main player agency, being the collection and bolstering of your playable pieces as to be stronger or/and better combined as to counter your opponent. Thus, having your pieces beat your opponents in battle. The main distinction from other chess-like games is that once the battle begins it is hands off in terms of the player. As the armies autonomously engage each other.

# Mechabellum - Auto Battler

Developer: Game River

Platforms: Windows, Mac OS

“Mechabellum is an epic auto-battler taking place on the newly colonized planet of

Far-Away. You’re the Commander of an army of mechs fighting it out in massive PvP battles. Customize and level up your units, strategically place them on the battlefield and watch them destroy your enemies.”

- Steam game description of Mechabellum

# Teamfight Tactics - Auto Battler

Developer: Riot Games

Platforms: iOS, Android, Mac OS, Windows

“Based on *Dota Auto Chess*, a mod for *Dota 2*, the game centers around eight players who construct teams to fight one another and be the last player standing. The battlefield consists of hexagons, where players can strategically place units on the hexagons on their side of the game board between rounds. During each round, a short battle automatically commences, with two players matched randomly for that round, or else paired against computer-controlled enemies. In the rounds against computer-controlled enemies, each enemy has a chance to drop gold, units, or items that the player can use. [player] Health lost from losing a round is calculated with a combination of set damage per round and how many units an opponent had still alive.”

- Wikipedia’s section about gameplay in regard to Teamfight Tactics

4.0 Intended users and their basic interaction with the system

Our target audience is the gamers who enjoy having fun. Their engagement consists of installing and playing the game, as well as progressing through increasingly harder levels till they are beaten.

Users:

* Gamers who have an interest in strategy video games.
* Target demographic being collage age undergraduates

Uses:

* A hobby to spend time on.
* An engaging means of sharpening one's critical thinking.

5.0 Known interactions with other systems within or outside of the client organization.

* Steam
* GoG.com (good old Games)
* Xbox Game Store

6.0 Known constraints to development

There are several constraints in the development process.

1. **Game Deliverable Deadline:** This is the fixed timeframe within which the game must be completed and released to the public or a specific target audience. Meeting this deadline is crucial to capitalize on market trends and to maintain player interest. However, rushing development to meet a deadline can lead to subpar quality, numerous bugs, and an overall unsatisfying gaming experience. Therefore, we must carefully plan and manage their time, setting milestones and deadlines for different stages of development to ensure that the final product is polished and bug-free.
2. **Funding for Art Direction and Assets:** High-quality visuals are a significant factor in attracting players and making a game visually appealing. This includes character designs, environments, animations, and special effects. Insufficient funding for art direction and assets can result in lower-quality graphics, which can negatively impact player reviews and initial impressions of the game. Game developers must secure adequate funding for the art team to ensure that the game's visual elements are not compromised during development.
3. **Game Scalability during development:** Game development is an iterative process, and as ideas evolve, developers may need to make changes to the game's mechanics, features, and content. Ensuring that the game remains scalable means that it can accommodate these changes without causing delays, performance issues, or disruptions in the development pipeline. Scalability is particularly important in modern game development, where updates and post-launch content are often expected, and the game must adapt to changes in technology and player preferences without compromising its core experience.

Managing these constraints effectively requires a balance between meeting deadlines, maintaining quality, securing sufficient resources, and ensuring adaptability. Clear communication and project management are essential to navigate these challenges and deliver a successful game that meets both player expectations and business goals.

1. Project Schedule

Broad overview. Must agree with the course schedule. Can include some of the potential meeting dates with milestones included.

A screen shot of a grid

Description automatically generated

1. How To Submit Proposals

The proposal will be electronically submitted to the GitHub repository under the doc folder section. Proposals would be in .doc format and include the subject line "Corsair Clash RFP Proposal". The deadline for proposal submissions is September 17th at 6:00 am PST. Any inquiries or requests for clarification will be emailed to our professor, Dr.BC.

1. Dates

All Proposals are to be submitted by September 17th at 6:00 am PST.

1. Glossary of terms

**Auto Battler** - A genre of video game where players place units on a battlefield during a preparation phase. These units then battle the opposing units with no direct input from the player during a combat phase.

**Strategy Game** - A genre of video game where the player’s decisions have a strong correlation to how well or how poor they do within the game.

Terms that are meaningful to your client group but might not be understood by someone outside your group.

\*Note: Remember that “system” means product, service, and/or system your group would like to see created, built, upgraded, and/or changed. It is a broad term.