[[1]](#footnote-1)

Creating a Competitive Open-Arena 2D

Twin-Stick Shooter

Anthony Cloudy and Squirrel Eiserloh

*Executive Summary*— This thesis intends to demonstrate an all-around mastery of the lessons and skills developed through the Guildhall’s software development track by building a competitive, open-arena, two dimensional twin-stick shooter from the ground up. The artifact will take *Kirby Air Ride*’s “City Trial” gameplay to the next level by creating a more competitive and progression focused game. *Realm of The Mad God* and *Sinistar* create tense shoot-em-up experiences that mirror the intent and feel of the artifact. The project will be built using concentric development, separating work into multiple tiers to organize dependencies and create specific deliverables. The artifact’s progress will be logged and tracked using a development diary and GitHub commit messages.

*Index Terms*— Game Development, Real-time Rendering, Software performance, Software quality

# INTRODUCTION

Mastery of programming for game development requires a vast knowledge base covering a multitude of difficult and varied skills. While the majority of the software development industry requires more niche roles with less cross-pollination of disciplines, game engine programmers must understand everything from input and real-time rendering to advanced data structures and networking. This thesis intends to demonstrate a mastery of the lessons and skills developed through the Guildhall’s software development track by building a competitive, open-arena, two dimensional twin-stick shooter from the ground up. The project aims to demonstrate an all-around mastery by creating a well-polished game as opposed to demonstrating a specific mastery by delving deep into a single area, such as AI or rendering.

The artifact will apply optimization techniques to create a 60fps gameplay experience for up to four players, even while they roam the map in different areas. The game will push its engine’s codebase to its limits, and act as a demonstration of what’s possible in the engine built from Guildhall experience. The game will feature split-screen multiplayer that pits players against one another in an arms race to build the most powerful ship in a set time. Players will explore an open arena, destroying cargo crates and enemy ships in order to earn upgrades to their ships. Upgrades affect the base stats of the player’s ships, enabling them to go faster, tank more damage, or shoot more powerfully. Players are able to hunt each other in this game mode, but no winners are decided after the timer is up. Players will then have to use their powered-up machines in three randomly-chosen contests: including but not limited to a battle royale, a race, or a survival challenge. Because the competition is picked randomly, players have no idea what they’re going to compete in during the arms race, which adds to frantic and fast-paced nature of the game.

The artifact will be built using a public GitHub repository to log all code commit messages, which will help discern what was done when and keep progress transparent. A development diary will be kept not only to document what challenges arise during development, but to also document decisions and problems resolved during the artifact’s creation. Concentric development will separate the game’s feature development into tiers, defining clear stages for the project and creating milestones. The project will be in a finished and defensible state when all of the tier 3 tasks (described in the methodology section) have been completed.

# Research Review

Because of the competitive nature of the thesis, the literature review focuses primarily on finding resources on competitive game design and creating multiplayer experiences. This also includes research into specific problems the artifact faces, including split-screen game design and shoot-em-up (shmup) design. Games that demonstrate either a strong competitive/multiplayer design are of equal importance to the research, and provide proven examples of what works and what does not. Other games are in the research review for their specific shmup qualities or control styles that provide reference for the artifact’s design. The researcher consulted Professor Squirrel Eiserloh and Professor Christopher Forseth from The Guildhall at Southern Methodist University to find games with mechanics, playstyle, and gameplay similar to the proposed mastery project. The researcher then studied on their suggested games, searching Steam with keywords including *Galak-Z* and *Space Pirates and Zombies*. The researcher expanded off of those initial results, using the “More Like This” section to find even more related games. The researcher also utilized his personal games library to find more similar games, such as *Kirby Air Ride*. Finally, the researcher utilized Bing and the SMU Central Libraries search with keywords such as “Competitive Games”, “Competitive Game Design”, and “Splitscreen Game Design” to find articles and research various aspects of gameplay the artifact will utilize. The researcher limited search results and games only to those localized in English.

## Literature Review

In "Rock Paper Scissors - A Method for Competitive Game Play Design", author Victor Chelaru discusses the nature of Rock Paper Scissors, or RPS design in games, where certain attacks have an absolute advantage or tie with others (just like the game the design’s namesake shares). The article goes in-depth on the metagame of “Pure RPS”, where the attacks have no lead up or predictability (grounded units vs flying units in an RTS), “RPS and Signals”, where attacks do have readability (such as the wind-up animation of a punch), and “RPS with separate Attacks and Signals”, where attacks have signals, but experienced players can cancel or feint signals. The article reveals the dominant, winning strategies for the above games that emerge from gameplay, and discusses ways to keep the game from incentivizing the wrong kind of gameplay. Because some of the dominant strategies that evolve include “be random and fast” and “don’t initiate any attacks” for the more basic RPS designs, ignoring the insight this article has could destroy the metagame for the thesis artifact, leaving it dead on arrival. The dominant strategy that evolves from the most advanced RPS design is to adapt to your opponent’s patterns, which will develop a healthy, competitive game that prioritizes player skill and reading your opponent without promoting stale tactics [1].

The article on “Shared-Multi-Split Screen Design” by Richard Terrell assesses and compares the different design considerations and limitations provided by various types of multiplayer screen designs. The article exposes some of the tradeoffs and design challenges that split-screen games face. Split-screen gameplay requires a reduction in graphical quality, as the game has to render two to four separate views every frame. The reduced screen space also can cause problems for players, and because the thesis artifact’s design requires this space to convey location, this could create problems. Other design hurdles the article mentions include the introduction of screen-peeking, a need for increased monitor size to prevent feeling constrained, and increased team communication if players want to cooperate. The article fails to mention any positive aspects of split-screen as opposed to multiple screen, which include cheaper setups, greater flexibility when playing with other people, no network latency, and the potential for more positive experiences that come from playing with others in person [2].

The postmortem for *Good Robot* provides valuable insight into some of the unique design challenges shmups face. The developer, Shamus Young, started the project as a solo project, but eventually transitioned to work with another studio once he realized that his game’s design had issues. The postmortem outlines how he managed to resolve the game’s flaws by working with the other team’s ideas, which included establishing a dynamic gameplay rhythm, with valleys and peaks of activity, and adding consequence to player death. Many of Young’s concerns are pitfalls this thesis could encounter while in development, especially in regards to game design and mechanics not panning out or a lack of proper pacing in the game. If the artifact fails to give players the sense of enjoyable tension, or fails to create meaningful and interesting player interactions, the project will be at risk [3].

A paper titled “Group Report: Progression Systems” from Project Horseshoe 2014 deconstructs the nature of progression systems. The report broke progression systems down into a series of building blocks that make up system fundamentals, as well as tactics to strengthen player motivation towards interacting with the systems. Of the system building blocks described, the most relevant to the thesis include the Progression loops, which spiral upward as the players gain power in order to accomplish new feats that grant them new powers. The thesis’ planned tiered weapons system matches the definition of a complexity loop, wherein acquiring new gameplay mechanics or tools grant access to new options, tactics, and areas. The power up system, in which players continually make incremental progress on their ships, matches a power loop, where playing the game improves the player’s avatar’s power, which improves their “virtual skill” for the round. The paper also links player motivations, such as superiority and control, to rewards like competition and power, via “progression atoms”. Progression atoms are in-game components that serve as the conversion from the player’s motivations into rewards. By giving a player who wants better control of their character a set of character stats, they can give the player the reward of power through those stats [4].

## Field Review



1: Kirby Air Ride's City Trial mode features power-ups scattered all over the level that alter the characteristics of player’s machines [5]

*Kirby Air Ride* is a multiplayer, 3D racing game created for the Nintendo GameCube, which put players head to head while piloting a variety of quirky “Air Ride Machines”. The game features an alternate game mode called “City Trial”, in which players were put in an open map and given free roam for 5 minutes. Players began on a basic, sub-par Air Ride, and are tasked with finding a better machine and collecting power-ups to customize their machine within in the time limit. At the end of the game, all players compete in a random minigame that tests the player’s skill and powered-up ride, with the winner of the competition winning the whole game. Although Terrel’s paper describes many of the design limitations of split-screen, *Kirby Air Ride* manages to utilize split-screen successfully to create an enjoyable experience despite the limitations, and many of the performance tradeoffs are either hidden by the game’s design or minor incidents (such as a few occasional framerate hiccups). The artifact for the thesis draws heavily upon City Trial’s gameplay for inspiration, and aims to push the boundaries of this original idea and take it to a new level. This thesis aims to utilize *Kirby Air Ride*’s unique gameplay style that provides randomness without arbitrary outcomes, while improving upon the game’s minimal player interactions and unwieldy combat [5].



2: Players battling monsters in Realm of the Mad God have to pay extremely close attention to their surroundings, as bullets come in various speeds and patterns that can end players lives instantly [6]

A fantasy bullet-hell with fast leveling and permadeath (the game deletes a player’s character when they die), *Realm of the Mad God* is an unconventional massively multiplayer online game (MMO). Players are thrust onto an open world where they travel to defeat enemies, gain experience, and loot corpses until all major bosses on the map have been vanquished. Once all the players have defeated the bosses, the whole server is thrust into a battle with the game’s final boss. The game is a twin-stick shooter, where players avoid bullets while desperately trying to land shots on the hordes of enemies on the screen. *Realm of the Mad God’s* map and player versus enemy (PvE) combat line up a significant amount with the design of the thesis artifact. Whereas players are incentivized to defeat enemies through the chance of rare equipment upgrades in the MMO, the artifact aims to use the power-up system to incrementally boost the player’s stats. The artifact ends up giving out a multitude of small power ups with a few and far-between large slot items, instead of a constant stream of items you may or may not be able to use. The artifact also aims to explore a similar control scheme to *Realm of the Mad God*, in order to give the players more control quicker without having to worry about being unable to “pilot” a physics-based ship. *Realm of the Mad God*’s pickup and equipment system is also important to the thesis, as players are inundated with a steady supply of weapons, armor and potions at a rate that matches the quick-paced nature of the game [6].



3: A player avoiding Sinistar while trying to create sinibombs. The game’s open arena and obstacles match the thesis’ design [7]

*Sinistar* is a top-down, multi-directional shooter where the player is locked in an arms race against “Sinistar”, the game’s villain. While workers attempt to reconstruct Sinistar, the player attempts to survive gunfire and mine planetoids to create “Sinibombs”, the only weapon that can defeat Sinistar. Once Sinistar is created, the player either needs to destroy him or run away, as getting caught by Sinistar results in instant death for the player. *Sinistar* provides an example of a PvE RPS balance that shifts over time, as it requires players to balance mining, direct attacks, and evasive maneuvers in order to win against Sinistar [1]. The player has to mine for sinibombs in order to attack Sinistar, but the act of mining leaves the player open to attacks from warriors. Without sinibombs, players can only evade Sinistar, as his attack trumps the player’s standard laser. The player’s options change in value before and after Sinistar is activated, creating gameplay dynamics that change over the course of the play session. *Sinistar* has very similar theming, handling, enemies and obstacles that the proposed artifact will contain. The idea of shooting level obstacles in order to acquire resources, the way the player navigates through the level, and the feeling of PvE combat in Sinistar match the thesis’ ideas. However, while *Sinistar* generates tension via PvE, the artifact will generate this tension mostly via PvP, as the arms race is between players, not against an almighty boss [7].



4: Piloting a ship in a dogfight against AI enemies [8]

*Space Pirates and Zombies* is a top-down multidirectional space shooter intertwined with real-time strategy gameplay. Players are able to construct ships and pilot a squadron on missions within various galaxies. As far as this thesis is concerned, this game provides a few examples of what not to do, as the dogfighting controls are clunky and the UI and progression are confusing. Because the game’s primary focus is real-time strategy, the shooting gameplay feels tacked-on and controls awkwardly, an experience this artifact will not emulate. [8]



5: Galak-Z's unique handling and polish set it apart from other titles in the genre, creating the feel of actually driving a spaceship [9]

*Galak-Z* is a Shoot ‘em up, 80’s Sci-fi Anime styled Roguelike that casts players as a lone pilot fighting against enemies in cavernous planetary dungeons. The gameplay combines Roguelike gameplay with Shoot ‘em up controls to create a unique experience, as players pilot a physics-based rocket through various “dungeon rooms”. The game also values stealth, as the player’s rockets make noise that alert enemies to the player’s presence. The game’s unique aesthetic and levels of polish are high quality, and while mostly out of scope for the constraints of the thesis, serve as a great reference to aspire and work towards. *Galak-Z*’s ship controls are also intuitive, and the artifact aims to explore the physics-based nature of the ship to see if that would improve gameplay over *Realm of the Mad God*’s control scheme [9].

The majority of the games listed have some sort of RPS gameplay, as outlined by Chelaru’s paper [1]. *Kirby Air Ride* features jousting-based combat that utilizes RPS with separate attacks and signals, as players must approach one another to attack, and can easily feint an approach to sway their opponent’s behavior [5]. *Kirby Air Ride*’s constant stream of power ups grants the players more control, but the game fails to provide progression systems for other common competitive player motivations [4]. The flow and rhythm concerns that arose during the development of *Good Robot* are an obstacle some of these games overcome as well [3]. Although extremely hectic, *Realm of the Mad God* manages to establish this rhythm through the spacing of enemies in dungeons, and by giving the players the ability to break out of tight situations via instant teleport to a hub world [6]. Because players are able to lure and stack multiple enemies to create hordes that would obliterate the game’s flow via incredibly intense moments, giving the player the option to take a break at any point prevents the game from becoming overwhelming [6]. *Galak-Z* comes from the other end of the spectrum, where the majority of the gameplay isn’t hectic, but tension and flow is generated through stealth and using level obstacles to alleviate pressure.

## Summary

This artifact aims to create interesting competitive multiplayer gameplay while attempting to avoid the various pitfalls and issues discovered through research. By utilizing RPS with separate Attacks and Signals as a foundation for designing player’s options and interactions, the artifact can avoid stale or boring dominant strategies [1]. Without the separation of attack and signal, the best course of action becomes never initiating attacks, which detriments the game [1]. Although the thesis intends to be competitive, the players should not always be at each other’s throats, as mentioned in the postmortem for *Good Robot* [3]. This thesis attempts to establish a good gameplay rhythm by balancing player interaction with the map’s scale, allowing players the choice to fight and the space to run off and recover, without making the map too large for players to find one another. While *Kirby Air Ride* creates an interesting play space and encourages moments of interaction through gameplay events, the game fails to incentivize combat enough. Players must be extremely close to one another to consistently battle, and with the scale of the map and handling of the machines, the game fails to deliver an incredible PvP experience [5]. This project intends to play with and combine *Realm of the Mad God* and *Galak-Z*’s differing control styles to create the best combat experience for the artifact. As mentioned in Terrell’s article, split-screen has a host of downsides and technical limitations that the artifact has to overcome [2]. Considering that optimization is a part of the mastery the thesis wants to demonstrate, the project attempts to ensure that the game runs well even with 4 players on screen.

# Methodology

The artifact will be created within the primary developer’s personal engine. The majority of the engine’s subsystems and design is ready to handle the bulk of the game’s design work, with a few remaining bugs to fix before beginning development work. The project employs concentric development to organize the game’s components and features into discrete tiers. Each of the tiers consists of a set of features that build off of the previous tier’s work, and provide a clear path for the project’s dependencies. The tiers also provide priorities for the sets of features, and naturally divide up the features into milestones. The last tier is considered optional, and serves as the stretch goals of the project.

Tier 0 consists of any mandatory engine work that needs to be done before beginning the project. Although the majority of the engine’s subsystems are up to the quality needed for the project, a few bugs with rendering and particle systems need to be addressed before continuing on to the remainder of the project.

Tier 1 consists of all the core gameplay elements that make up the game. These features focus on getting the game functional first, proving out the core loop and the gameplay elements before moving on to polish tasks. This tier includes implementing multiplayer, programming player ships and rudimentary enemies, the game’s basic power ups, and a level to fly around in. The game will also have two distinct modes, assembly (the main game) and challenge (the minigames segment), along with start and end UI, including the victory screen. Most of the content will not be polished to final quality, but instead be the foundation for the rest of the game’s features and polish to build off of. This tier is much akin to a Proof of Concept Gameplay milestone, but will be first playable as well.

Tier 2 contains tasks with a focus on getting the core gameplay smooth, polished, and feeling good. This tier is to mitigate the risk of overscoping up front and running out of polish time, so that before any new non-core features and functionality are added, the game already feels good. This will establish the minimum-viable product for the game, and will ensure that the project meets the goal of creating a complete and polished game.

Tier 3’s tasks focus on augmenting the current gameplay to improve gameplay quality and replayability. This tier introduces socket items/equipment, and adds the remaining power-up items and stats. Completion of the tier’s tasks will also add 2-6 more challenge variations, as well as procedurally generated map zones during assembly phase. These tasks will be polished to final quality, in order to match the quality of the game after tier 2 has been finished. Once tier 3 is completed, the developer should be able to halt development at any point and the game should still feel like a complete and polished product, ready for defense.

All of the remaining tasks and stretch goals reside in tier 4, which is optional for completion. This tier includes tasks such as implementing bosses during assembly phase and adding more challenges. The tier will also add new slot items to the game, such as new weapons and chassis types. This tier will continue to push the bar of quality and polish for the game, and will be worked on during any of the remaining time in the project.

Table : Calendar of deliverables and respective submission dates

|  |  |
| --- | --- |
| **DELIVERABLES** | **SUBMISSION DATE** |
| Present Topic to Class/Potential Advisors | September 2, 2016 |
| Updated Annotated Bibliography Due | September 16, 2016 |
| Research Selection Due | September 23, 2016 |
| Methodology Selection Due | September 30, 2016 |
| Tier 0 tasks completed | October 4, 2016 |
| C25 Proposal Sent to Advisor | October 4, 2016 |
| Advisor Assigns Grade to Proposal | October 7, 2016 |
| Tier 1 tasks completed | November 10, 2016 |
| Mastery Thesis Proposal Signature Due | November 10, 2016 |
| IRB Submission date | November 11, 2016 |
| Tier 2 tasks completed | December 9, 2016 |
| Artifact Progress Due per Proposal Advisor assigns CR/NC | December 9, 2016 |
| Tier 3 tasks completed | February 20, 2017 |
| Mastery Thesis Artifact Due to Advisor | February 20, 2017 |
| First Draft Paper Due to Advisor | February 27, 2017 |
| Advisor Feedback on Thesis Paper Sent to Student and Reader | March 16, 2017 |
| Advisor Approves Thesis Paper | March 30, 2017 |
| Reader Comments Thesis Paper Due | March 30, 2017 |
| Defense Date Confirmed by Masters Committee | March 30, 2017 |
| Defense Required Revisions Complete to Advisor | April 21, 2017 |
| Thesis Defense | TBD |

By using concentric development, the project is not only organized into discrete milestones with clear objectives and deliverables, but is separated into a chain of dependencies that prioritize the core components of the project. By December 9th, 2016, the artifact projects to be at minimum viable product. By February 20th, 2017, the artifact projects to be ready to submit for thesis defense, with all tier 3 tasks and below completed.

# Conclusion

This thesis aims to demonstrate an all-around mastery of the lessons and skills developed through the Guildhall’s software development track by creating a well-polished game. By building a competitive, open-arena, two dimensional twin-stick shooter from the ground up, polishing and optimizing the game, the artifact will support the thesis’ claim of mastery.

The artifact will be built using a public GitHub repository to log all code commit messages. Because GitHub commits bind developer comments to specific code updates, any progress comes with in the moment commentary that keeps development transparent. Higher-level commentary on the project and design decisions will be noted in a development log, which will provide a bigger picture view of the project.

Concentric development will define clear stages for the project and create milestones by splitting the game’s features into tiers. All of the tier 1 tasks will be completed by November 10th, 2016, which compose the skeleton of the game. At this date, the core gameplay will be present and the game will be playable, yet unpolished. By December 9th, 2016, all of the tier 2 tasks will be completed, signifying a playable and polished project, but still lacking in some extra gameplay features. By February 20th, 2017, the artifact projects to be ready to submit for thesis defense, with all tier 3 tasks completed. Completion of all tiers through tier 3 will result in a complete and polished game that demonstrates a mastery of software development for games, the goal of this thesis.

# References

|  |  |
| --- | --- |
| [1] | V. Chelaru, "Rock Paper Scissors - A Method for Competitive Game Play Design," 23 January 2007. [Online]. Available: http://www.gamasutra.com/view/feature/130150/rock\_paper\_scissors\_\_a\_method\_for\_.php. [Accessed 8 September 2016]. |
| [2] | R. Terrell, "Shared-Multi-Split Screen Design," 17 June 2011. [Online]. Available: http://www.gamasutra.com/blogs/RichardTerrell/20110617/88846/SharedMultiSplit\_Screen\_Design.php. [Accessed 8 September 2016]. |
| [3] | S. Young, "Good Robot Postmortem #2: Gameplay," 19 July 2016. [Online]. Available: http://www.shamusyoung.com/twentysidedtale/?p=33343. [Accessed 8 September 2016]. |
| [4] | J. Hoffstein et. al., "Group Report: Progression Systems," in *Project Horseshoe*, Comfort, 2014. |
| [5] | *Kirby Air Ride*. (GameCube). JP: HAL Laboratory, Nintendo, 2003. |
| [6] | *Realm of the Mad God*. (Adobe Flash). USA: Wild Shadow Studios, Deca Games, 2011. |
| [7] | *Sinistar*. (Arcade). USA: Williams Electronics Inc., Williams Electronics Inc., 1982. |
| [8] | *Space Pirates and Zombies*. (Microsoft Windows). USA: MinMax Games, MinMax Games, 2011. |
| [9] | *Galak-Z: The Dimensional*. (Microsoft Windows). JP: 17-BIT, 17-BIT, 2015. |

1. This paper will be submitted for review on November 10, 2016.

   **Anthony Cloudy** is with Southern Methodist University Guildhall, 5232 Tennyson Parkway, Building 2, Plano, Texas 75024 USA (e-mail: acloudy@smu.edu). He graduated from Southern Methodist University in 2015 (B.A. Computer Science), and has previously done contract work with Fractal Fox as a professional game developer.

   **Squirrel Eiserloh** is a game programming faculty Lecturer at SMU Guildhall, Southern Methodist University’s game development graduate program. Since he graduated from Taylor University in 1996 (B.A. Physics) he has been working as a professional game developer in the Dallas area, contributing to over a dozen commercial game titles. He co-chairs the Dallas chapter of the IGDA, and coordinates the Math for Game Programmers sessions at the annual Game Developers Conference in San Francisco. (e-mail: beiserloh@mail.smu.edu) [↑](#footnote-ref-1)