

**Team 5 Guys**

**Kill Mo’ Chicken**

Kill Mo’ Chickens

Request for Proposal  
Version 1.0

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | When | Who | What |
| 1.0 | 09/12/24 | Nathan Hampton,  Austin Walker,  Ankit Paudel,  Clayton Slack,  Sohan Lama | Initial Drafting |
|  |  |  |  |

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

A genre of video games that has seen popularity is tower defense. Its popularity has led to the creation of games ranging from the beloved Balloons Tower Defense (BTD) to the not as well-known Fort Meow. While there has been a lot of development in this area, a game like the original BTD could be what fans of this genre, who are also nostalgic for the original BTD, are looking for. This style includes enemies that follow a predefined path, and towers that can be bought and upgraded with points earned from destroying enemies.

There are many games that fit this category, yet there seems to be a lack of these games that are themed around chickens. This is a missed opportunity, because a tower defense game themed around chickens has the potential to be a really silly game. If done right a goofy game can have gamers playing it because of its silliness. Just look at the mobile game The Battle Cats. The Battle Cats has seen over 10M downloads on the Google Play store alone and has an overall rating of 4.5 stars as of writing. Part of the reason for its popularity is it has a good balance of goofiness to get gamers to download, strategy to get them invested in improving their skills, and content to keep their interest.

1. Project Objectives

The purpose of the project is to make an engaging and strategic tower defense game where players can build and upgrade defenses to protect their base from waves of enemies. The game should feature a variety of towers and enemies to ensure a diverse and challenging experience for players over extended gameplay sessions.

**The game should include:**

* Round Manager
  + Keeps a record of current round, difficultly, enemy spawn patterns, and game progress.
  + Coordinate round start and stop.
  + Can view and manage current game state, and win/loss conditions.
  + Pushes notifications for incoming waves and other important events.
* Towers
  + Different types of towers with unique abilities and upgrades.
  + Towers can be placed on specified locations on the map.
  + Towers automatically attack enemies within their range.
  + Towers can be upgraded or sold for resources.
* Enemies
  + Various enemy types with different attributes and abilities.
  + Enemies follow predetermined paths or patterns toward the base.
  + Enemies drop resources or power-ups when defeated.
  + Enemies become progressively harder as waves advance.
* Map/Levels
  + Multiple maps with varied layouts and paths for enemies.
  + Maps can have environmental elements that impact gameplay (e.g., obstacles, bonuses).
  + Difficulty scales with player progression and wave advancement.
  + Each map provides strategic opportunities for tower placement and upgrades.
* Menu/UI
  + Main menu includes options for starting a new game, loading a saved game, and accessing settings.
  + In-game interface displays current resources, wave information, and tower management options.
  + A settings menu allows players to adjust game options, including audio, difficulty, and gameplay preferences.
  + HUD includes health of the base, remaining waves, and available resources.
* Audio
  + Background music corresponding to the theme of each map.
  + Sound effects for tower attacks, enemy movements, and other in-game events.
  + Audio settings to toggle sound effects and music on/off.
  + Audio files are optimized for performance to reduce load times.
* Save/Load
  + Ability to save the player’s progress, including tower setups, resources, and completed waves.
  + Load saved progress from the main menu to continue where the player left off.

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

Current system(s) used by your group (if any). If none, are there any systems that are similar to the one you would like to build?

1. Intended users and their basic interaction with the system

Users could be in the client (your) organization or outside.

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

**Digital Distribution Platforms:** Distributes the game on Google Play, Apple App Store, and Steam to maximize reach and utilize platform-specific features.

**High-Score Server:** Integrates with a server to manage global high scores, promoting competition and engagement among players.

**Game Host’s Operating System:** Ensures the game performs optimally across various server operating systems to maintain smooth and reliable gameplay.

1. Known constraints to development

**Optimized Gameplay and Accessibility**: Our game is being crafted using the Godot Engine to deliver a seamless experience across different operating systems including macOS and various mobile platforms. We're committed to ensuring that the game runs smoothly on a variety of devices by optimizing graphics and managing resources efficiently. Our aim is to strike the perfect balance between providing a challenging experience for seasoned gamers and being approachable for newcomers. This will be achieved through a user-friendly interface and clear, straightforward in-game guidance, making sure our game is enjoyable for players at all levels.

**Efficient Development and Quality Assurance**: With the constraints of our academic schedule, our team is focused on efficiently managing our time and resources to prioritize essential gameplay elements, particularly those involving dynamic enemy interactions. The development of intricate features such as diverse enemy behaviors and complex tower mechanics adds a layer of complexity that could potentially increase the likelihood of bugs. To mitigate this, we are committed to a stringent cycle of testing and debugging to guarantee a stable and polished final product.

**Resource Management and Streamlining Complexity**: As we develop our game, we are implementing complex features that significantly enhance gameplay but also introduce challenges related to stability and the potential for bugs. Our approach to ensuring a reliable gaming experience includes extensive testing and debugging. We also encounter limitations with the development tools available and need to implement coding practices that optimize resource use. This means we must plan carefully and focus on essential game features to guarantee optimal performance across a range of device capabilities, ensuring our game not only meets our ambitious design standards but also delivers an immersive and engaging player experience.

1. Project Schedule

|  |  |
| --- | --- |
| Date | Description |
| Sep 17, 2024 | Git Repository Set up, Objectives/Features divided and identified |
| Sep 19, 2024 | SA Demos |
| Oct 8, 2034 | Skeleton code running |
| Nov 2, 2024 | First Round Feedback due |
| Nov 12, 2024 | Complete looking project, missing some features |
| Dec 5, 2024 | Final Demo |

1. How To Submit Proposals

Proposals should be submitted as a pdf to Austin Walker at [walk5700@vandals.uidaho.edu](mailto:walk5700@vandals.uidaho.edu).

Any questions can be directed to tkContactInfoHere. Please note that all questions will be reproduced and distributed to all applicants.

1. Dates

Deadline for submission and when respondents will be notified that a winner is chosen.

1. Glossary of terms

y 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.