**Lessons Learned**

**Turn-Based Combat Game**

**The Flying Mongooses**

**4601 Mid Rivers Mall Dr**

**Cottleville, MO 63376**

**Date**

**7/31/21**

**Table of Contents**

[Introduction 3](#_Toc332377397)

[Lessons Learned Approach 3](#_Toc332377398)

[Lessons Learned from this Project 4](#_Toc332377399)

[Lessons Learned Knowledge Base / Database 5](#_Toc332377400)

[Process Improvement Recommendations 5](#_Toc332377402)

# Introduction

This document will make note of lessons learned by The Flying Mongooses team. For many of us, this was our first step into developing a game and managing the development itself. While the team as a whole did a great job in creating this game, at times the development process wasn’t perfect. With that being said, what worked well and didn’t work well will be categorized in the table below, helping all of us become better developers and managers in all of our future endeavors.

# Lessons Learned Approach

The approach to writing up this lessons learned doc is to show, from the projects start to end, how the team succeeded and how the team failed. Each team member will contribute both individual lessons they themselves learned, as well as lessons learned from others or lessons learned by the team as a whole.

# Lessons Learned from this Project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Category | Issue Name | Problem/Success | Impact | Recommendation |
| Scope | Good, big ideas | During initial development stages, good, but exceeding scope, ideas were presented, causing confusion on what would be developed and what wouldn’t. | Many good ideas for the game were shared, but then held onto for too long and not properly communicated on their status, causing confusion among the developers on what was in or out of development scope | Project manager should better communicate large design changes and make clear to the rest of the team which ideas are being taken forwards, and which ideas need to be cut in order to develop the project in a timely manner |
| Scope | Complicated Systems, Lack of Experience with Online Systems | The planned system of online Player versus Player was not completed in time for the project deadline. | One of the two planned battle modes did not make completion, giving players a much more limited experience | PvP developer(s) should better communicate updates and problems with developing their online matchmaking system |
| Audio | Audio System Implementation | Both music and sound effects were successfully added to the final version of the project utilizing multiple methods | In-game audio provides a more user-orientated experience | Future Unity projects could utilize these methods for ease-of-implementation and user experience |
| Remote Work | Merging different versions of a single project | Combining different versions of a project introduced conflicting files, scripts, and other assets of the same types. This was pushed further with the aforementioned inconsistency in communications. | Much time would/could be spent on just merging and fixing bugs caused from the conflicting files, and different devices and specs. As well as long periods of time downloading and extracting different types of the files. | Improved communication and closer interactions with one another if remote work is to continue and be practiced further. |
|  |  |  |  |  |

# Lessons Learned Knowledge Base / Database

Large scope, little communication. When planning the project, we set aspirations and ideas that increased the scope of the project much more than we thought they would. Our lack of experience with Unity, coupled with a lack of strong communication, caused a lot of confusion, development time being spent on features that would not end up in the final version, and an inefficient amount of time spent on integrating different versions of the project at all the various stages of development across our various personal machines. For future projects, team members are coming away with a heightened respect for the need to actively engage and communicate with their team, and with a new perspective on the true scope of game features and their impact on development.

An audio system in Unity can be built utilizing multiple methods, such as with scripts for UI and non-looped sound effects like a melee hit, and with game objects for music and looped sound effects like wind, which allow a more seamless user experience, and decreases the difficulty of implementing by reducing the need for extra code.

# 

# Process Improvement Recommendations

Going forwards to bigger and better projects, the importance of the project manager properly assigning tasks, maintaining viable project scope, as well and improved communication with the project developers will become more vital. The project manager needs to take more responsibility of ensuring good inter-team communication as well as keeping the project on track and maintaining any scope creep that may arise.

This free Lessons Learned Template is brought to you by [www.ProjectManagementDocs.com](http://www.ProjectManagementDocs.com)