## Introduction

### Introduction

Write this at the end

#### Purpose

The purpose of the document is to explain in detail the intricacies of the design choices.

Define the different parameters of the project, different rules and constraints.

The document is addresses to programmers and will be using a language that is very technical. Will often reference code and assumes that the reader is familiar with software development methods.

goal is to make sure any team of programmers can produce the wanted result with this document

#### Term definition

Write this at the end

### Policies and tactics

#### Goals

Maximize usability (user-friendliness)

Maximize simplicity (less key commands, less input combinations)

Maximize accessibility (multiplatform, also impact some tech choices and design choices explained later in this doc) (implies minimize system reqs)

#### Guidelines

Constrained to a particular coding style and a set of repo rules

Time constraint

Financial constraint (therefore we need free stuff)

Communication (everyone should be communicating and everyone should be aware of the exact progression of the project)

#### Development methods

Slack standup meeting every day to monitor progress and make sure no one is left behind.

Sprint meeting every week

Backlog of tasks

## System Overview

### Architecture considerations

#### Technology selection

Unity because of accessibility (multi platform), time constraint (no complex graphical libraries to learn) and financial constraint (free for educational purposes)

C# instead of Javascript because it’s a cleaner language, better OO language, easier to design for, also similar to java which we are familiar with

MySQL because it’s free

#### Idea choices

Public repository for open source, help from community. we believe community builds better software

We believe in sharing

We learned from other people’s project, we want our project to be a learning resource for other people

### Design considerations

#### Assumptions

We don’t assume that the user is familiar with our type of games or has experience with any software. Therefore it needs to be simple (simplicity constraint).

We can’t make the assumption that the user has powerful computers (accessibility constraint)

#### Dependencies

Not sure about this yet

#### Constraints

Explained above

### High-level design

#### System overview

#### Design pattern

#### Final top level domain model

#### Final top level deployment diagram

#### 