|  |
| --- |
| HALT gAMES PRESENTS |
| Blizzard the Wizzard |
| Technical Specification |
|  |
| **Timothy Veletta** |
| **Version 0.01** |

|  |
| --- |
| [Type the abstract of the document here. The abstract is typically a short summary of the contents of the document. Type the abstract of the document here. The abstract is typically a short summary of the contents of the document.] |

Table of Contents

[Document History 2](#_Toc336158675)

[Gameplay 3](#_Toc336158676)

[Characters 3](#_Toc336158677)

[Game Objects 3](#_Toc336158678)

[Physics 3](#_Toc336158679)

[Artificial Intelligence 3](#_Toc336158680)

[Interface 3](#_Toc336158681)

[Mockups 3](#_Toc336158682)

[Art and Cutscenes 3](#_Toc336158683)

[Packaging 3](#_Toc336158684)

[Terrain 3](#_Toc336158685)

[Gameplay Elements 3](#_Toc336158686)

[Special Effects 4](#_Toc336158687)

[Artistic Goals 4](#_Toc336158688)

[Music and Sound Effects 4](#_Toc336158689)

[Story 4](#_Toc336158690)

[Level Design 4](#_Toc336158691)

# Document History

|  |  |  |
| --- | --- | --- |
| Version | **Date** | **Changes** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Engines

## Rendering Engines

These engines will cover anything which must be displayed on the screen. There might be separate engines for each type of thing being rendered

(e.g. 3d, sprites, backgrounds, interface).

## Level Engine

This concerns all code that encapsulates manipulation of the level data. How levels are loaded, saved, and passed to the rendering engine

## Object Engine

How are objects handled in the game. This might encapsulate processing of currencies in the game and other statistical data.

## Character Engine

Engines to do with the manipulation of properties of the characters in the game.

## Artificial Intelligence Engine

Artificial intelligence engines - how do the game entities think? What algorithms and techniques will be used, and how will this plug into the other game engines?

## Control Loop

This describes how all the parts of the game will be brought together.

## Game Shell

How is the game launched? How is the game configured and how does it run within the operating system framework? What OS libraries does it use?

## Multimedia Engines

What engines are required to play back audio visual data and how are they structured to work?

## Audio Engines

How is sound handled in the game? Sound effects?

# Tools

## Data Conversion Tools

What tools will be needed to convert between formats and place data into the game binary?

## Content Creation Tools

What tools are available or will be required to be built to create content for the game?

# Algorithms

Algorithms used for compression or path finding can be outlined in this section.