Software Requirements Specification

Crazy Cannonz

Avinash Nooka, Grant Harrison, Joseph Field III, Jared Nightingale

|  |  |  |
| --- | --- | --- |
| Date | Changes | Version |
| 9/15/17 | Initial Document | 1.0 |
| 9/20/17 | Revisions to 1.2, 1.3 | 1.1 |
| 10/27/17 | Revisions to 1.2, 2.2.2, 3.2.1.3.1, 3.2.1.5.7 | 1.2 |

**Table of Contents**

[**1. Introduction 2**](#_30j0zll)

[**1.1 Purpose 2**](#_1fob9te)

[**1.2 Scope 2**](#_3znysh7)

[**1.3 Definitions 2**](#_2et92p0)

[**1.4 Overview 3**](#_tyjcwt)

[**2. Overall Description 3**](#_3dy6vkm)

[**2.1 Product Perspective**](#_1t3h5sf) 3

[**2.2 Product Functions 4**](#_26in1rg)

[**2.3 User Characteristics 4**](#_lnxbz9)

[**2.4 Constraints 5**](#_35nkun2)

[**2.5 Assumptions and Dependencies 5**](#_1ksv4uv)

[**3. Specific Requirements 5**](#_44sinio)

[**3.1 External Interface Requirements 5**](#_2jxsxqh)

[**3.2 Functions 5**](#_z337ya)

[**3.3 Performance Requirements 8**](#_3j2qqm3)

[**3.4 Logical Database Requirements 8**](#_2xcytpi)

[**3.5 Design Constraints 8**](#_1ci93xb)

[**3.6 Software System Attributes 8**](#_qsh70q)

[**4. Appendix 9**](#_3o7alnk)

# 

# 

# 1. Introduction

## 1.1 Purpose

This document’s purpose is to describe the basic design, function, and software requirements for the game, Crazy Cannonz. By enumerating specific design requirements and functionalities, this document will provide a clear guideline as to what will be needed to achieve the stated goal. This document is intended for use by the customer, and by the developers, for the purpose of clearly explaining and understanding what expectations are to be had for the product.

## 1.2 Scope

The application, Crazy Cannonz, is a 2D arena-style multiplayer platformer. Each player will be spawned into the game depicted as a cannon on legs. The objective of the game is to collect ammunition, which will be used to defeat other players and earn points in the process. The user will be able to partake in battles between other players. Each player will begin at a specific spawn location in an arena in which various pickup items will be scattered. Each scattered item will provide the character ammunition that can be fired at other players. Players will use these items to combat one another until the time limit is reached. The player with the highest point total at the end of the match wins. This application will be made using Unity. This application can be obtained via Dropbox.

## 1.3 Definitions

**Matchmaking Process (Matchmaking) -** The matchmaking process (matchmaking) is how a player selects the map and the number of players.

**Graphical User Interface (GUI) -** A graphical user interface (GUI) is the visual way in which the user will navigate the menu system, using various buttons, sliders, and other input fields to select and change various parameters.

**Unity –** Unity is a cross-platform engine for the purpose of designing the complex animations and physics required for games and simulations. Unity can build for multiple platforms, including Windows, Mac, Linux, Android, iOS, and others.

**Video Graphics Array (VGA)** - Video Graphics Array (VGA) is a particular type of format and hardware for displaying video output. A VGA signal is an analog signal.

**Internet Protocol (IP) Address –** An Internet Protocol (IP) Address is a unique numerical value, punctuated by some delimiter, usually interjected periods or colons, and which is assigned to a networked computer for the purpose of identifying it and correctly routing information to it from other computers in the network.

## 1.4 Overview

The remaining portion of this document will contain an overall description of the game and in its entirety, the various specific functional requirements, and an appendix. Topics covered in the overall description include a product perspective, uses of the game, and other information regarding the production of the game. The specific requirements section will describe the requirements of the game in more detail, covering major features, overall design, and other specifications. The appendix will contain any referenced information in the earlier sections of the document.

# 2. Overall Description

## 2.1 Product Perspective

### 2.1.1 System Interfaces

Crazy Cannonz will not require any additional systems to run on any computer operating system.

### 2.1.2 User Interfaces

With a computer, the user interface will consist of the user’s preference of

peripherals ranging from keyboard to USB controller.

### 2.1.3 Hardware Interfaces

To run a Unity application, any system needs at least the following hardware capabilities1:

1. A DX9 graphics card, shader model 3.0, or DX11 with 9.3 abilities.
2. Support for the SSE2 instruction set.

To support the required operating systems, as outlined below, the application will require that the computer it is running on to have the following hardware capabilities:

1. For Windows XP SP2+, the minimum requirements are a Pentium 233 MHz processor, 64 MB of RAM, 1.5 GB of hard drive space, and a VGA monitor of at least 800x600 resolution2.
2. For OS X, 2GB of memory and 8GB of available space are required on an approved list of models of Mac computers3.
3. For Ubuntu 12.04+, a minimum of a 700 MHz processor, 5GB of hard drive space, 512 MiB of RAM, and VGA of at least 1024x768 resolution4.

To interact with the application on a desktop computer, a keyboard will be required along with an additional USB port for a controller. Any keyboard which will meet the requirements of the desktop computer will suffice. Xbox controllers for either the Xbox 360 or Xbox One will meet the requirements. For a laptop computer, no additional hardware is needed.

### 2.1.4 Software Interfaces

All computer devices running either Windows, Mac or Ubuntu will be able to install and run this application5. To run an application developed using Unity, the user will need to meet the following operating system requirements:

1. Windows XP SP2 or higher is needed.
2. Mac OS X 10.9 or higher is required.
3. Ubuntu 12.04 or higher is required.

All computers will need the appropriate driver software to operate their mouse or trackpads and keyboards. Such software, however, will either be automatically downloaded upon interfacing the peripherals with the desktop machine or need to be manually downloaded and installed.

## 2.2 Product Functions

The application shall have the following functions:

2.2.1 The application shall have an icon and an executable file so that opening the application is convenient for the user.

2.2.2 The application shall have a menu system which shall, at a high level, provide the user the ability to manage various settings related to the game, allow them to start the matchmaking process, and exit the application. The user shall navigate this as a typical GUI environment. Keyboard and controller mappings which show how to control the in-match character will be provided in the options menu.

2.2.3 From the matchmaking portion of the menu system, the application shall flow into match functionality. Here, the player is able to compete with one another within the map and time constraints set when beginning the match.

## 2.3 User Characteristics

The application is intended for a user base which is familiar with the general concept of third-person, 2D platform-style shooters. Users may be unfamiliar with such concepts, but familiar with gaming mechanics in general, and be able to interact with the game successfully. Users are expected to have an understanding of the concepts of matches, matchmaking, and movement mechanics. No tutorial will be provided, aside from the keyboard and controller mappings in the options menu, which show how to control the character.

## 2.4 Constraints

Because applications developed with Unity require certain operating system and hardware requirements, the finished product will only be able to run on a subset of all available personal computers, laptops, and desktops. Most computers, however, should be capable of supporting the game.

## 2.5 Assumptions and Dependencies

The game will rely on the functionality of the underlying operating system on which it runs. As such, it requires that software be stable and dependable. The game will also rely on Unity itself to be reliable in how it generates its files and in its ability to run.

# 3. Specific Requirements

## 3.1 External Interface Requirements

All desktop machines running the application will require a keyboard to play the game as well as at least one external controller.

## 3.2 Functions

3.2.1 Phase One

3.2.1.1 Game launch

3.2.1.1.1 The user will be able to click on an icon to launch the game. The user can find this on their desktop, their task bar, or search for it on their system.

3.2.1.2 Menu System

3.2.1.2.1 Match Menu

3.2.1.2.1.1 The user will be able to select the desired map from a list of maps. Each map will have a name and a number of recommended players. An option for choosing a random map will be displayed.

3.2.1.2.1.2 The application requires at least two players but can have a maximum of four for local matches.

3.2.1.2.1.3 Player one will be able to set the match timer. The match will go for as long as the timer lasts.

3.2.1.2.2 Options Menu

3.2.1.2.2.1 The user will be able to select the volume of the background music as well as the sound effects volume on a scale of 0 to 100 on sliders.

3.2.1.2.2.2 The player key/controller input mappings will be listed along with the actions they perform, with no option to switch key bindings.

3.2.1.3 Maps and Aesthetics

3.2.1.3.1 Each map will have a certain number of inbuilt spawn points that will be used to spawn players and ammunition. The ammunition will appear as crates that will give players fireable projectiles; and when these crates are picked up, there will be a delay until the next random crate is spawned.

3.2.1.4 Single machine multiplayer

The application will require single machine multiplayer. A minimum of two players is required, but the game will support up to four players. The user will be able to choose between keyboard and controller input for player 1, but players 2-4 are limited to the controller only.

3.2.1.5 In-Match Gameplay

3.2.1.5.1 Gameplay will begin with all players spawned into a

random location on the map.

3.2.1.5.2 A 5-second countdown will commence before players are given control of their Cannoneer.

3.2.1.5.3 At this time, players can move around the multi-tiered arena and proceed to pick up whatever ammunition they can find for their Cannoneer.

3.2.1.5.4 There will be a variety of types of ammunition that can be picked up. These ammo types carry different movement speeds and point values.

3.2.1.5.5 As soon as the ammunition is fired, it is gone, and the user must find more ammunition to fire.

3.2.1.5.6 After picking up ammunition, the aim of the game is for players to fire at one another, gaining points for successful hits/kills.

3.2.1.5.7 The player will have a delayed respawn at one of the spawn locations if hit by a projectile.

3.2.1.5.8 The player will be able to do the following basic movements: Jump, run, stand and crouch. The player will be able to fire their cannon during each of these movements unless their ammunition prevents them from doing so.

3.2.1.5.9 The match is time-based. Once time is up, the player with the most points wins the match.

3.2.1.5.10 At the end of the match, a score screen shall be displayed, showing each Cannoneer and their point totals. It is at this point that an option to replay the match in the same map, or to go back to the match menu, shall be displayed, to be chosen by player one.

3.2.2 Phase Two

3.2.2.1 Maps and Aesthetics

3.2.2.1.1 More maps will be added for the user to select and play in.

3.2.2.2 Multiple Players

3.2.2.2.1 The user will be able to select and play with up to three additional players on the same machine, totaling four players.

3.2.2.3 Networking

3.2.2.3.1 Match Menu

3.2.2.3.1.1 To incorporate multi-machine matchmaking, the match menu shall have the following new options: Host Game and Join Game.

3.2.2.3.2 Hosting Games

3.2.2.3.2.1 The user can select the Host Game button to start the game hosting process.

3.2.2.3.2.2 The host will set up a password for their hosted game.

3.2.2.3.2.3 When the password is made, the IP address of the host’s machine will be displayed on the match menu.

3.2.2.3.2.4 As players join the game, they will appear on the match menu screen. The host has sole control over when to start the game, being able to select the Start Game option.

3.2.2.3.3 Joining Games

3.2.2.3.3.1 The user will be able to join another player’s server, granted that the user has the host machine’s IP address as well as the correct password.

3.2.2.3.3.2 If a player selects the Join Game option, two text input fields will appear, prompting for the IP address and the password of the server of the hosted match. The player can then hit a third button to submit that data.

3.2.2.3.3.3 Upon submitting this information, a loading symbol shall be displayed. If the password and IP address are correct, and that machine is, in fact, hosting a game, then a dialogue will indicate to the user that they have joined a game, and the other players in the game will be displayed in their match menu.

3.2.2.3.3.4 If an error occurs when trying to join the game, a message will be displayed saying such and will indicate the nature of the error.

3.2.2.3.3.5 The joined player will have to wait for the host to start the game; as such, the Start Game option will be grayed out for all joined players.

## 3.3 Performance Requirements

### 3.3.1 Single Machine Performance Requirements

As long as the user meets the hardware requirements to play the game,

there shouldn’t be any noticeable performance issues. It should run

between 30-60 frames per second, with no discernable lag.

### 3.3.2 Networking Performance Requirements

As long as all hardware requirements are up to par and users have a reliable internet connection, there shouldn’t be many if any, performance issues when playing through LAN. Slight lag and/or hiccups are possible.

## 3.4 Logical Database Requirements

There will be no persistent storage involved for this application, thus no logical database is necessary.

## 3.5 Design Constraints

### 3.5.1 Single Machine Restraints

This application will be available strictly for Windows, Mac OS X,

and Ubuntu.

### 3.5.2 Networking Restraints

Users must be connected to the Internet via WiFi or ethernet in

order to play online.

## 3.6 Software System Attributes

### 3.6.1 Reliability

The application should be reliable enough to run smoothly, with

minimal bugs. No errors or crashes should occur during gameplay

or menu browsing.

### 3.6.2 Availability

The application will be available to access and play at all times.

### 3.6.3 Security

The game will have minimal security features on single machine games. During networking, standard networking protocols will be used, which will provide a measure of security to the connection. Otherwise, no enhanced security precautions will be taken.

### 3.6.4 Maintainability

Developing additional features such as maps and in-match items should be simple to integrate into the existing system. The core game mechanics should be stable; and should any additional game modes or other features need to be added, the menu system framework will provide a convenient way to tack on new functionality.

### 3.6.5 Portability

Because Unity can build games for more than just the three target operating systems, and because Unity makes the build process for specific platforms a simple task, the game can quickly be built for the listed operating systems, and for any other platforms that Unity supports6.

# 4. Appendix

1. <https://unity3d.com/unity/system-requirements>
2. <https://support.microsoft.com/en-us/help/314865/system-requirements-for-windows-xp-operating-systems>
3. <https://support.apple.com/kb/SP702?locale=en_US>
4. <https://help.ubuntu.com/community/Installation/SystemRequirements>
5. <https://unity3d.com/unity/system-requirements>
6. <https://docs.unity3d.com/Manual/BuildSettings.html>