Requirements Specification Document

Impossible3

Table of Contents

5.0 Requirements Specification

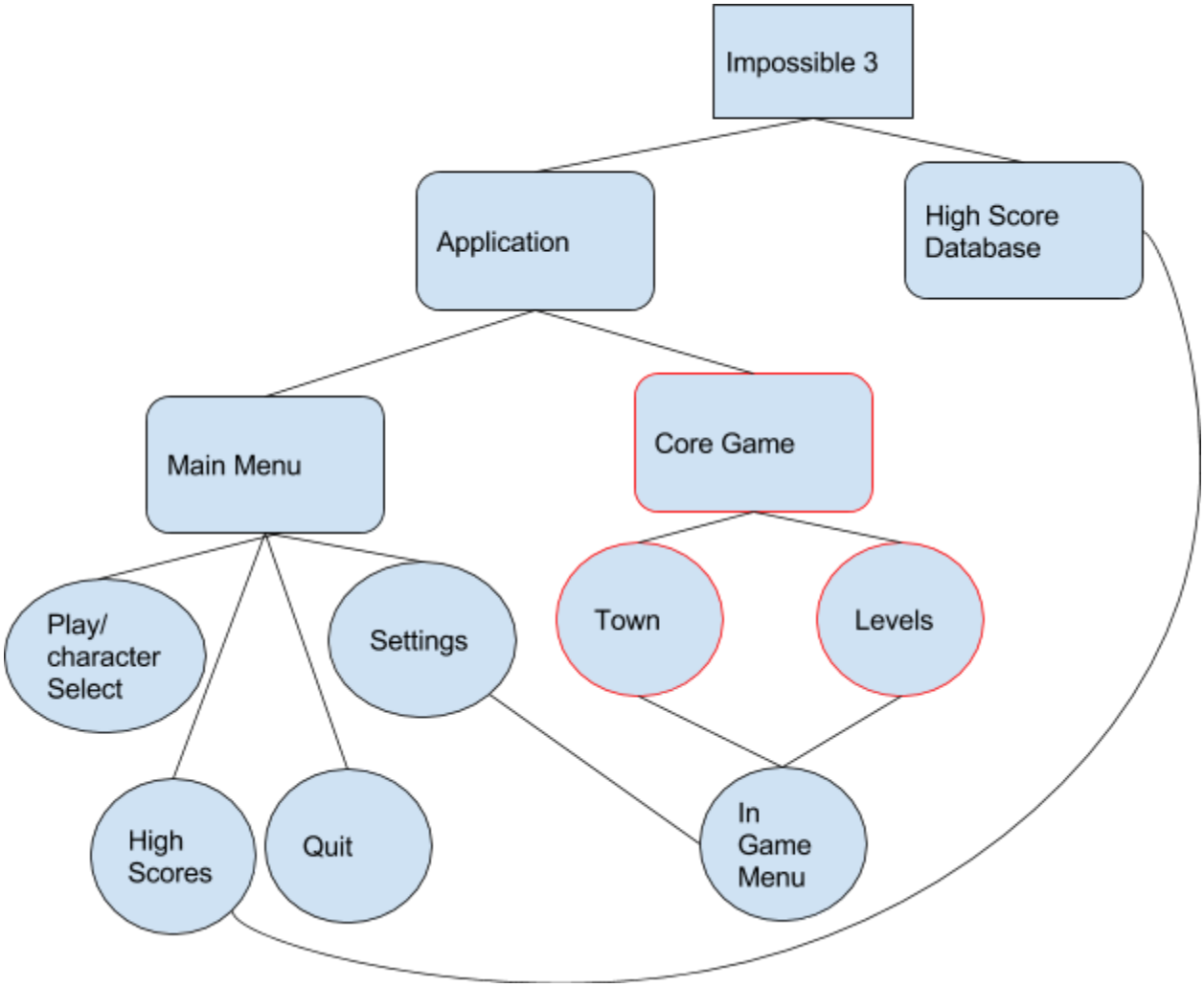
5.1 Introduction

This section introduces the requirements specification document for Impossible3 game. It provides the purpose and outline of the project.

5.1.1 Purpose

Impossible 3 is a roguelike dungeon crawler with an isometric camera view. Roguelike is a term used to describe a game that has procedurally generated levels, leveling characters, and permanent death. The dungeon crawler genre of games have levels of increasing difficulty, dungeon exploration, and loot systems. Impossible 3 is a combination of the two genres. There has been a rise in the popularity of games that created in the 1970s to 1990s. This project's purpose is to familiarize the project members with the game mechanics and complications of old school games.

5.1.2 Diagram



5.1.3 Outline

This document is organized as follows Section 5.2 shows the Computer Software Configuration Item (CSCI) Component Breakdown containing the Computer Software Components (CSC). Section 5.3 shows Functional Requirements by CSC. Section 5.4 details Performance Requirements. Section 5.5 shows the Environmental Requirements section detailing hardware and software requirements.

5.1.4 Table of Definitions

Term	Definition
Player	Person who plays the game
Roguelike game	A game that takes some inspiration from Rogue(1980). Game includes key features such as permanent death, tile based levels, monster encounters, treasure, turn-based combat system, complex interacting systems, procedural generation, resource management
Dungeon Crawler	A role playing game that has the player exploring dungeons and killing monsters. The player controls a character that gets stronger as the game goes on. Key features are leveling system, loot/gear system, increasing enemy difficulty.
Game Client	Piece of software that connects to game server. Server and client shar information
Core Game	The task that is most often experienced by the player
AI	Artificial Intelligence usually controls enemies

5.2 CSCI Component Breakdown

CSCI *Impossible3* is composed of the following CSCs:

5.2.1 Game Client CSC -- <Main application player will launch>
5.2.1.1 Main Menu Screen CSU
5.2.1.1.1 Character Select / Play Button
5.2.1.1.2 High Scores Table (Optional)
5.2.1.1.3 Settings
5.2.1.2 Core Game
5.2.1.2.1 Town / Player Hub -- <Area to upgrade characters and gear>
5.2.1.2.2 Actual Gameplay Levels -- <This is the main game area>
5.2.1.2.3 In Game Menu -- <Access to settings, Exit, Save/load, from in game>

5.2.2 Assets and Resources -- <Anything that is needed for the full game to run>

5.2.3 High Scores Database CSC (Optional) -- <Server to store and get scores>

5.3 Functional Requirements by CSC

5.3.1 The Main Menu Screen will allow players to start the game, change options,

and exit back to the desktop.

5.3.2 The Dungeon Building System will be able to randomly generate obstacles for the player and enemies to move around on.

5.3.3 The Enemy System will be able to put a fair and increasingly difficult array of enemies for the player to combat against.

5.3.4 The Enemy System will mix up the different types of enemies that are given to the player to combat against.

5.3.5 The AI System will be able to accurately track where the player characters are and react intelligently to stay alive while searching out and trying to defeat the player characters.

5.3.6 The GUI System will react to player mouse movement by highlighting the square the mouse is on, and highlighting any selected units.

5.3.7 The GUI System will react to player mouse clicks by selecting a unit and bringing up more options for the player to use with that character, such as move, attack, or a special ability (TBD).

5.3.8 The GUI System will show where the player character can move after clicking that option by highlighting possible squares in green.

5.3.9 The GUI System will show who the player can attack or use abilities on by showing red squares for attack range.

5.3.10 The GUI System will show the health and cooldowns of each character.

5.3.11 The Combat System will be able to add and subtract health to each character after an attack or healing action is made on them.

5.3.12 The Loot System will give characters upgrades based on random chance of spawning different classes of loot from enemies and chests.

5.3.13 The Loot Database will have a list of different armors, weapons, and special items that each player character will be able to randomly find.

5.4 Performance Requirements by

5.4.1 Play Game at Reasonable Speed. Users should be able to play the game without lag or other frame drops on most modern systems.

5.4.2 Immerse Players Within Game. Users should be able to engross themselves in the game and use it as an outlet for entertainment.

5.4.3 Satisfactory Combat and Loot System. The game should provide a strategic combat system and loot system that both beginners and advanced players can learn.

5.4.4 Intuitive Character Menu and GUI System. Users should be able to easily understand the user interface without much difficulty.

5.4.5 Creative Play-style. Players should be able to come up with their own play style. Classes and skills should be balanced to facilitate the above.

5.4.6 Intelligent AI. The game should have a capable AI. The AI should challenge the player. The AI should be able to deal with obstacles as well as have different play

styles depending on the enemy type.

5.4.7 Various Enemy Types. The game should have a variety of challenges / enemies. Enemies should be diverse in their behaviour as well as skills.

5.4.8 Creative Story Line. The game should have a narrative built to suit the style of. the game

5.5 Project Environment

Following are the hardware requirements for *Impossible3*:

Category	Requirement
Processor	SSE2 instruction set support.
HDD Space	TBD
RAM	4 GB+ (Recommended)
Graphics Card	DX9 (shader model 3.0) or DX11 with feature level 9.3 capabilities.
Display	Required
Mouse	Required
Keyboard	Required
Controller	Not Currently Supported

Because *Impossible3* is made in Unity, current hardware specs have been copied from Unity's requirements for running games.

Following are the software requirements for *Impossible3*:

Category	Requirement
Operating System	Windows XP SP2+, Mac OS X 10.8+, Ubuntu 12.04+, SteamOS+.
Additional Software	Will be installed with game

Unity runs games on these modern operating systems.

5.5.1 Development Environment

Each dev will require the following environment:

— An installation of Unity Personal (from: https://store.unity.com/download?ref=personal)
— A working Git installation
— An IDE of their choice (Unity provides an IDE for C#). Later on in the semester we may need to use Microsoft Visual C# if we ever create a C# project that works in

tandem with the Unity project.
— Windows 7 Home Premium (or whichever version)

5.5.2 Execution Environment Requirements

Execution of the final product is simple, all that is required is a working Windows 7 installation. In theory we could also publish the game to Linux, Steam OS, or iOS and so on and so forth. It's preferable that the end user have a dedicated GPU but it's not necessary, several PC builds nowadays have integrated graphics and that works fine. In all likelihood our game will not be that CPU-intensive, or RAM-intensive, at least in comparison to other free and popular games such as *League of Legends* or *Team Fortress 2*, but the the execution machine should not be a netbook (or something similar), as those machines are too weak to run even small games.