**Software Requirements: Specification for the Shadowbrume Game**

**Document # 1**

Version B

**19th February 2015**

**Table of Contents**

[1 Introduction 3](#_Toc412318957)

[1.1 Purpose 3](#_Toc412318958)

[1.2 Definitions, Acronyms and Abbreviations 3](#_Toc412318959)

[1.3 Reference Documents 3](#_Toc412318960)

[1.4 Document Overview 3](#_Toc412318961)

[2 Overall Description 4](#_Toc412318962)

[2.1 System Perspective 4](#_Toc412318963)

[2.2 System Functions 4](#_Toc412318964)

[2.3 User Characteristics 5](#_Toc412318965)

[2.4 Design Constraints 5](#_Toc412318966)

[2.5 Assumptions and Dependencies 5](#_Toc412318967)

[3 Overview 5](#_Toc412318968)

[3.1 External Interface Requirements 5](#_Toc412318969)

[3.1.1 User Interfaces 5](#_Toc412318970)

[3.1.2 Hardware Interfaces 5](#_Toc412318971)

[3.1.3 Software Interfaces 6](#_Toc412318972)

[3.1.4 Communication Interfaces 6](#_Toc412318973)

[3.2 Functional Requirements 6](#_Toc412318974)

[3.2.1 Player Controlled Character CSCI (PCC) 6](#_Toc412318975)

[3.2.2 Mapping CSCI (MP) 7](#_Toc412318976)

[3.2.3 Environment Interactive Objects CSCI (EIO) 8](#_Toc412318977)

[3.2.4 Enemy Artificial Intelligence CSCI (EAI) 10](#_Toc412318978)

[3.3 Nonfunctional Requirements 11](#_Toc412318979)

[3.3.1 Game Startup 11](#_Toc412318980)

[3.3.2 PCC Load Time 11](#_Toc412318981)

[3.3.3 EAI Load Time 11](#_Toc412318982)

[3.3.4 PCC Controls 11](#_Toc412318983)

[3.3.5 EAI to PCC Interaction 11](#_Toc412318984)

[3.4 Testing Constraints 12](#_Toc412318985)

[3.4.1 Standards Compliance 12](#_Toc412318986)

[3.5 Software System Attributes 12](#_Toc412318987)

[3.5.1 Reliability 12](#_Toc412318988)

[3.5.2 Availability 12](#_Toc412318989)

[3.5.3 Security and Privacy 12](#_Toc412318990)

[3.5.4 Maintainability 12](#_Toc412318991)

[3.5.5 Portability 12](#_Toc412318992)

[3.5.6 Safety 12](#_Toc412318993)

[3.5.7 Training-related Requirements 12](#_Toc412318994)

[3.5.8 Packaging Requirements 12](#_Toc412318995)

[3.6 Other Requirements 12](#_Toc412318996)

[4 Analysis Models 13](#_Toc412318997)

[4.1 Data Flow Diagram 13](#_Toc412318998)

# Introduction

This Requirements Specification document establishes the functional, performance, and development requirements for the initial release of the Shadowbrume videogame software.

## Purpose

Shadowbrume is an interactive computer game with a graphical interface in which the user explores an environment with interconnected rooms and an over world map, while helping the protagonist find her family and escape the monsters that are chasing her.

## Definitions, Acronyms and Abbreviations

CSCI Computer Software Configuration Item

NPC Non Player Characters

SRS Software Requirements Specification

PCC Player Controlled Character CSCI

MP Mapping CSCI

EIO Environment Interactive Objects CSCI

EAI Enemy Artificial Intelligence CSCI

Collectable - Items that disappear after the player interacts with them. Simulating item gathering

## Reference Documents

The following standards apply

DOD-STD-498A US Department of Defence Software Documentation Standard

J-STD-016-1995 IEEE/EIA Standard for Information Technology, Software Lifecycle Processes, Software Development, Acquirer-Supplier Agreement IEEE-STD-P1063 IEEE Standard for Software User Documentation

## Document Overview

Section 1 identifies the scope of this document, the purpose of the software, and lists the definitions, acronyms and reference documents. Section 2 contains the overall description of the software. Section 3 identifies the four main Computer Software Configuration Items (CSCIs) that comprise the system, and gives the functional requirements and constraints for each CSCI. Section 3 also describes the quality Srequirements for the software.

# Overall Description

## System Perspective

Throughout the short time that videogames have been around many kinds of genres have introduced themselves to the ever growing audience of people who enjoy the medium that is gaming. The one defining genre in videogames has arguably been the adventure game. Adventure games are one of the earliest forms of videogames and while the lines between genres have blurred a bit; most gamers can still distinguish what a true adventure game is. Adventure games often times can have little to do with combating enemies or growing stronger as the player progresses; they can also merely be about avoiding danger, making huge choices and ultimately; survival

This particular piece of software is focused on those particular aspects of the genre. It will be a single player experience, driven mostly by its narrative and the user’s ability to manage the supplies that are given to them.

## System Functions

There are essentially four main functional areas, which correspond to the four CSCIs specified in section 3:

* Control Management: This software uses a traditional mouse and keyboard control scheme. There is no combat in the game so the players controls over the character are limited a few contextually needed moves. The first level of the game will be a tutorial area where users can play with the controls and get used to them. Once they are comfortable with the control scheme, they then have the option to move on with the game.
* Mapping: When players begin the game they are loaded into the tutorial area that is attached to a much larger world called the overworld. Traditionally in games; overworlds are used to tie multiple smaller levels together. The overworld for this software will serve the same purpose. It will link each of the areas in the game together and will allow for players to travel to them in any particular order that they want. Each of these areas will have NPC’s assigned to them that will chase the PCC until they exit that particular area. These NPCs will have EAI that will activate all of the actions required to chase the PCC.
* Item Management: The player will have the ability to interact with several objects in the game world. Most of these interactions will be limited to touching the object and triggering a dialogue box that will explain who the object belongs to and why it may be there. The remaining objects will be items the player will use to keep the NPC at bay while they are adventuring in the games many locations.
* Choice Management: A players encounters with the creatures and environments will be dynamic in that the player will have the choice in which area they chose to go through first, what paths they wish to take, and whether or not shortcuts are the best possible route from one point to another.

.

## User Characteristics

This game will require a bit of skill in order for a user to have some success in playing it. The player is required to use the limited amount of moves that are given to them to help the PCC survive their journey. Those commands are:

Walk: A default movement animation that will be triggered by the user via the W, S, A or D keys on the keyboard.

Jump: A simple jump the player can use to give themselves a small boost, it will be helpful but limited. Triggered by the Space key.

Run: Perhaps the most useful of all the moves in the game, also the most important, this will give the PCC a boost in movement speed. Triggered by the Shift key.

Touch: This is the primary action of the PCC as this will allow the user to interact with items in the environment. Triggered by left clicking the mouse.

## Design Constraints

The one constraint with this particular software is the game engine. The game runs on Unreal Engine 4 and; as such is the case, will require a stronger than usual computer to finish development. Along with only running on Unreal Engine 4 is the fact that C++ is the only code it can run off of since Unreal only uses that language as its source code. Another note is that only Visual Studio 2013 is compatible with the Unreal Software. Every constraint is related directly to the actual usage of Unreal Engine 4 to complete the project.

## Assumptions and Dependencies

No special assumptions or dependencies have been identified.

# Overview

## External Interface Requirements

### User Interfaces

All interactions by the user are done via the mouse and keyboard. Upon starting the game the user will be place in a demo level and will have the ability to experiment with the controls and get used to them. If the user hits a button that is not assigned an action nothing will happen.

### Hardware Interfaces

This game uses the Unreal Engine 4. As that is case we will need to have a machine that meets the recommended specifications to run Unreal Engine 4.Those specifications are:

Processor: Quad Core Intel or AMD Processor

Graphics Card: Nvidia GTX470 or ATI Radeon HD 6870

Memory: 8GB RAM

### Software Interfaces

The system shall be capable of running the newest version of the Unreal game engine. Unreal Engine 4.

### Communication Interfaces

None

## Functional Requirements

The functional requirements are described for each of the four CSCIs: Player Controlled Character (PCC), Mapping (MP), Environment Interactive Objects (EIO), and Enemy Artificial Intelligence (EAI).

The structure of the game can be broken down into these four major categories. The PCC is the avatar by which the user will interact with the virtual world. The MP will dictate where the player may venture throughout the world with the overworld map in particular serving as a central hub. The EIO’s will carry story information and some will also serve as tools for the player to use to defend themselves against EAI. EAI will be similar across all four enemies, the only difference will be cosmetic appearances.

### Player Controlled Character CSCI (PCC)

The player controlled character CSCI will be the core interaction between the user and the software. It does not hold data about the world nor the enemy creatures or interactive items. It will merely be the way the user interacts with these other elements of the software.

The PCC is the main controlling part of the software. It will give the user access to other parts of the software in any particular order they choose, save the first area (the demo level) which is where the game must begin.

*3.2.1.1 PCC Internal Data*

PCC will be given a short set of instructions based what buttons are pressed by the user, and in what location the user press these buttons in accordance to what is on the screen.

* Movement (e.g. move forward, backwards, strafe) – this is essential to the games core interaction. The player will traverse the virtual landscape with this set of instructions primarily
* Jump (e.g. start jump, ascending jump, falling jump) - this is a state that the PCC can enter upon the press of a particular button. Jumping will be limited, so that if the character can’t make a particular jump the user must find a different way around the obstacle blocking their way.
* Run – this state will modify the movement state by accelerating spee in whichever direction the user is pushing the PCC.
* Touch – this state allows the user to interact with certain objects in the environment, it will also be a catalyst for a good portion of the storytelling in the game. The player’s freedom of choice is partially at the mercy of this function. The more the user interacts the more they learn about the world.

*3.2.1.2 PCC Control Structure*

After initialization of the areas and objects, the game loads the PCC into the game world. After a brief text pertaining to the story the player is given control of the PCC.

1. The user will spawned into the demo (tutorial) area.
2. The user will be given the freedom to roam around said area using touch, jump or run to interact with the environment and get used to the controls.
3. Once the user has gotten comfortable with the controls they may then venture outside of the tutorial area where they will again be presented with a new set of choices, though the choices will not imposed upon them as far as an order task is concerned they will have the freedom to explore the game hub and it’s interlocking worlds at their own discretion.
4. Eventually, upon entering one of the interlocking worlds, the player will be chased by one of the games NPC’s who are controlled by the EAI. The goal will then be to find a particular object in this zone before they are killed by them. There is no direct combat in the game so they will only be able to use run and touch to keep they NPC’s away.
5. Once they gain an item from a particular zone they can then leave that zone and go the next, where they will repeat the previous process.
6. If the player is dead the game ends. Otherwise this loop continues until the player collects all of the items.

### Mapping CSCI (MP)

The mapping CSCI holds all of the areas together based on their predetermined geological locations in the games directory. These are ultimately brought together by a central map call the overworld which will serve as central pathway from one place to another for the user

*3.2.2.1 MP Internal Data*

Each area is represented within this CSCI using the following information:

* A unique id – Each area has a name unique to it that will let the player know where it is they are going or leaving from. The game will always start; for instance, at the “indoors” area.
* A unique object – Each area hold a particular object of value to the protagonist
* A unique enemy – Each area will have a unique enemy attached to it that will chase the PCC once they enter their assigned area.

*3.2.2.2 MP External Interface*

MP will provide the following functions to be used by other CSCIs:

* Essentially there are 7 gameplay areas and each area will have a distinct id attached to it. Of these 7 areas one (overworld) will be keep track of the location of the player both when he/she occupies that area or they are in another area
* An area is preassigned a unique object. This object will be objective for the user once they enter this particular area and; once the user obtains this object, they are done with that particular area and can leave whenever they choose.
* An enemy will be assigned to each room as well. They will be made aware of the user’s presence once they have entered these assigned zones. The player will then have a limited amount of time to find an object in the level or they will be killed by the enemy and sent back to the beginning.

### Environment Interactive Objects CSCI (EIO)

EIO are the items in the game that the user will interact with throughout the course of their time playing the game. They are assigned to unique locations and all but a few a minimal uses. The matchbox, in particular will reset the light source that serves as the shield for the protagonist throughout the course of the game.

*3.2.3.1 EIO Internal Data*

Objects are represented with the following information:

* A short name – objects in the game will be given a short name. This name will be revealed to the player once they focus the camera on the object.
* A brief description – Each object will have a brief dialogue that reveals who it belonged to and why it might be there.
* Additions to inventory (In some cases) – As mentioned before there will be cases such as the matchbox where them item will reset the light source the PCC uses to defend themselves. As it stands the only item that does this is the matchbox. This will be set number not revealed to the player.

*3.2.3.2 EIO External Interface*

The object manager will provide the following functions to be used by other CSCIs:

* An object namer: This will only appear once the object is the focus of the users camera view
* An object describer: once the user initiates the touch action they will be given a brief description of the item as well a small portion of the stories plot. When the player turns the camera away from the object this description will fade off of the screen
* Inventory + - if the object is more than a mere plot device, it will be added to the user’s inventory. The only object with such behavior is the matchbox.

*3.2.3.3 List of Interactive Objects and their purposes*

Brick of Moss: This is an introductory level item. It will necessary to grab this item and the Flint & Steel to light the Fireplace in the house. It is collected by the player.

Flint & Steel: This is an introductory level item. It will be necessary to grab this item along with the Brick of Moss to light the Fireplace. It is collected by the player.

Fireplace: The Fireplace will be lit once the player has collected the Brick of Moss and the Flint & Steel. Once it was turned on it stays on.

Driftwood Shield: The Driftwood Shield belongs to the Eilin’s older brother Conall. Upon Interacting with it the player will see a text box that reveals part of the story.

Boggart Mask: The Boggart Mask belongs to Fitz, the Eilin’s younger brother. When the player interacts with it they will get a text box that reveals a part of the story.

Horn Flask: The Horn Flask is an interactive item that belongs to the Eilin’s father (Pa). When the player interacts with it they will be shown a text box that reveals part of the story.

Mortar & Pestle: The Mortar and Pestle item belong to the Eilin’s mother (Ma). When the player interacts with it. It will give them a text box that reveals part of the story.

Misc. Mask: The Misc. Mask is an interactive item in the introductory level. While it holds no particular meaning and when the player interacts with it they will get a text box that reveals part of the story

Book: The Book is an interactive item that will activate a text box upon interaction by the player. Holds no real story context. Not a collectable.

Shell: The Shell is an interactive item that will activate a text box upon interaction by the player. Holds no real story context. Not a collectable.

Cloak: The Cloak is an interactive object that was made by the main characters mother (Ma). It is a collectable. The player will need this item to leave the introduction level.

Lantern: The Lantern is an interactive object that the player needs to collect in order to leave the introduction level.

Lorebook A: The Lorebook A is an interactive object that explains the legend of the Shadowbrume. It is not a collectable.

Lorebook B: The Lorebook B is an interactive object that explains the Feyfolk. It is not a collectable

Lorebook C: The Lorebook C is an interactive object that explains the Creatures of the Deep. It is not a collectable.

Elder Tree: This item is an end game interact able object. Once the player reaches the end of the game the player will have a choice to make. Interacting with this item should trigger this choice: “Join my family” or “Live by myself”

Candle Box:: The Candle Box (matchbox) is an interact able object that will give the player more candles as the player reaches them. The more candles (matches) the player has the more often they can brighten the light that keeps the Shadowbrumes away from them. They are distributed; one per map.

Door: The Door is an interactive object from the introductory level. Once Eilin (PCC) has collected the proper items; she can use this door to enter the overworld area.

Favorite Mask: The Favorite Mask is an item that belongs to Eilin’s brother Fitz: It is revealed to the player that Fitz has passed away once the player finds this. It is a collectable item.

Foot Traps: Foot Traps are interactive objects that are triggered when a player walks over them. They will cause instant death if triggered.

Rune Necklace: The Rune Necklace is an interactive item that belongs to Eilin’s father (Pa). When the player interacts with it they are told how her father has passed away. It is a collectable item.

Herb Satchel: The Herb Satchel is an interactive item that belongs to Eilin’s mother (Ma). When the player interacts with it they told how she passed away. This item is a collectable.

### Enemy Artificial Intelligence CSCI (EAI)

The EAI will manage the activities of the enemy characters in the game. When a character enters an area that a particular enemy is assigned to the EAI will be activated. All the enemies in the game will behave the same the will merely have cosmetic differences.

*3.2.4.1 EAI Internal Data*

Creatures are represented using the following information:

* A unique id – This information is never divulged to the user, each of these characters will be assigned to an area which will also contain an item that belongs to them.
* An attack pattern – each enemy will be assigned the same attack. These attacks will differ in animations but will have the same logic
* A standing animation – Each enemy will have an idol animation for when they are incapable of getting within a certain distance of the PCC

* Mobility – Like the attack patterns, these will be similar across all versions of the enemies. They will be animated differently but the will have the same level of mobility regardless of this.

*3.2.4.2 EAI External Interface*

* Attack patterns – the enemies will attack the player by constantly poking at the light that emanates from them the will do this until they are retracted by the user refilling the lights power, or until the light runs out. If this is the case the PCC is defenseless and will faint upon direct contact with the creature.
* An idol animation – when the creature is unable to touch the player and the PCC is at an idol state itself; the enemy will be in an idol state and will perform an animation based on which enemy id they carry.
* Enemy Mobility – NPCs will be assigned to certain areas of the game based on their id, they will follow the PCC through their assigned areas.

*3.2.4.3 Enemy Names*

Brother1: This is Fitz as a Sadowbrume form. Eilin’s younger brother

Brother2: This is Conall as a Shadowbrume form. Eilin’s older brother.

Ma: This is Ma as a Shadowbrume. Eilin’s mother.

Pa: This is Pa as a Shadowbrume. Eilin’s father

## Nonfunctional Requirements

### Game Startup

The Shadowbrume project will load from the uproject in Unreal Editor. The game should load within 5 seconds. Usually with the aforementioned recommended hardware settings that should be within this scope for the project.

### MP (Mapping) Load Time

Each map being both, the Indoors and the Overworld, should load within 5 seconds just as the project loads. If it is loaded on a new PC, the shaders will take longer to allocate but once loaded onto the HDD, each map should load within the aforementioned time of 5 seconds or less.

### EIO Load Time

The Environmental Interactive Objects should load along with the entire project and MP (Mapping). Both indoors and the overworld objects should lead within 5 seconds as the game starts up and each individual map loads.

### PCC Load Time

Eilin (PCC) should load simultaneously with the uproject and indoors/overworld as all the assets should load together as one Unreal Project. Once the assets and animations of the PCC is properly constructed and configured, it should load within 5 seconds similar to game startup.

### EAI Load Time

The EAI (Four Ghosts) should lead as the PCC reaches certain areas of the Overworld that would trigger each EAI. These EAI should load no more than 2 seconds from the time the PCC triggers it. These assets would then be prompt by the PCC to follow and damage the EIO (Lantern).

### PCC Control Input

The user should be free to control Eilin with latency free inputs. This means that there should be almost 10 milliseconds or less of loading from the input of any of the action buttons or directional keys from the keyboard to control the PCC. In other words, inputs should be almost instantaneous.

### EAI to PCC Interaction

The PCC (Eilin) should take damage to her illuminated light source being the Lantern + Candlebox (EIO) as the EAI (Four Ghosts) strike this illumination. This means that the PCC must decrease in illumination/light radius as the EAI are able to strike the EIO (Lantern). This again should be instantaneous damage to the light source on par with the input delay of the action and movement controls from the keyboard.

### PCC Death

The PCC should faint when the illumination for the EIO (Lantern) reaches 0. At 0 illumination, the PCC will revert into a panic animation which will inform the user that they were hit too many times by the EAI. The animation will prompt and take no more than 2 seconds to complete. This will then trigger a “Game Over” screen. Checkpoints will possibly be implemented as the project is completed, to allow the User to continue in the area where the PCC last fainted. This should be within the loop of the project running the entire time.

### Game Ending

Once every EIO is collected, this will prompt a choice by user. This will allow the choice for the PCC (Eilin) to either live alone or be reunited with her family and turn into a Shadowbrume ala ghost which is the EAI that was chasing the PCC through the entire project. Either ending should only take around 5 seconds to load with each lasting around 1 minute each.

## Testing Constraints

### Standards Compliance

Usage of the game (other than planned testing episodes) on Wayne State University equipment shall be governed by the current guidelines for game playing.

## Software System Attributes

### Reliability

The system should never crash or hang, other than as the result of an operating system error.

### Availability

There are no specific availability requirements

### Security and Privacy

There are no specific security and privacy requirements.

### Maintainability

All code shall be fully documented. All program files shall include comments concerning authorship and date of last change.

The code should be modular, to permit future modifications. Anticipated updates include changes to the environment enemy behavior and character actions.

### Portability

The system should be portable to any system capable of running Unreal Engine 4. No other specific portability items have been identified.

### Safety

No safety requirements have been identified.

### Training-related Requirements

No specific training should be necessary for a user to begin playing the game. The game should load a player into a demo level. The player can experiment with the controls here, but are by no means required to do so.

### Packaging Requirements

There are no specific packaging requirements.

## Other Requirements

There are no other requirements.

# Analysis Models

## Data Flow Diagram

