Software Requirements Specification

for

Interactive Projection Mapping

Version 1.0 approved

Prepared by Devin Sanders

UAB IEEE Student Branch

22 June 2012

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Project Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. System Features 3

3.1 System Feature 1 3

3.2 System Feature 2 (and so on) 4

4. External Interface Requirements 4

4.1 User Interfaces 4

4.2 Hardware Interfaces 4

4.3 Software Interfaces 4

4.4 Communications Interfaces 4

5. Other Nonfunctional Requirements 5

5.1 Performance Requirements 5

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 6

Appendix C: Issues List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The Interactive Projection Mapping (IPM) is a game or interactive program that will use the façade of the Alys Stephens Performing Arts Center at UAB as a user interface. The visual effects will take precedence from 3D architectural projection mappings performed around the world. The development cycle will begin with version 0.1 and the initial release of the project will be v.1.0.

This project will be for a single presentation and will not be distributed commercially.

## Document Conventions

Code functions will have authors name, date, and initials. A description of the function will also precede the function. Documentation of inputs of function and outputs will accompany description.

## Intended Audience and Reading Suggestions

This document is intended to describe the design of the project; therefore, it is intended only for developers and testers. The project will not be sold commercially and documentation will not be provided for end users.

## Project Scope

Please refer to the Vision and Scope document written by Devin Sanders for the Interactive Projection Mapping project.

## References

References will include documentation pertaining to Xbox Kinect and Kinect API.

Also see “IPM\_vision\_and\_scope.docx”.

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

# Overall Description

## Product Perspective

The Interactive Projection Mapping is a supplement to the Architectural Projection Mapping created by Jean Jacques Gaudel to be presented on April 13 2013 at the Alys Stephens Center. The IPM will have a 20 minute presentation time, and allow individuals or groups of guests to have approximately 5 minutes each to interact with the program.

## Product Features

The IPM will use a Kinect® Camera as a controller for the project. The user interface will be projected onto the façade of the Alys Stephens Center using two projectors designed for projection mapping. Users will be able to use gestures and body movements to control the game.

## User Classes and Characteristics

The audience for the presentation is expected to contain multiple demographics. A wide variety of age groups as well as different cultural, educational and professional backgrounds are expected to be in attendance. For this reason, retro-style games have been chosen as the basis for the project. These games are anticipated to bring nostalgia to many of the older generations while still being engaging to the younger audience. The overall presentation should be impressive regardless of demographics.

## Operating Environment

The program will run on a Mac OSX machine. The Mac OSX machine will be expected to have an Intel processor, greater than 1 GHz CPU clock, and greater than 1 GiB of RAM.

## Design and Implementation Constraints

C/C++ will be used in conjunction with OpenKinect and a C++ library for use with the Kinect® controller. The interface for each game should utilize characteristics of the façade of the Alys Stephens Center. Each program should be executable in less than 5 minutes.

The event could possibly take place in the evening, so the programming will have to take into account lighting conditions.

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

## User Documentation

User documentation will not be necessary for the IPM.

## Assumptions and Dependencies

The projectors for the projection mapping require a special interface that must be used. This interface will be determined by the artist Jean Jaques Gaudel.

# System Features

## Motion Tracking Controls (via Kinect®)

### Description and Priority

This will be the main control interface for the IPM. A Microsoft© Kinect® Motion Controller will be used to track the movements of the user. Predetermined movements will cause specific interactions with the program. This feature is High Priority.

### Stimulus/Response Sequences

TBD per game/program.

Example 1: Pinball Game – Hand gestures will move the paddles. Body/hip movements will tilt the table left/right. Leg movments will bump the table.

Example 2: Breakout Game – Body movements used to slide the paddle left or right.

Similar controls can be used for any game/program for use with the IPM.

### Functional Requirements

REQ-1: Should be able to calibrate Kinect® for any user such as the user’s height

REQ-2: Should not react to subtle movements such as the player shifting their weight

REQ-3: Should be able to adjust the sensitivity of the controller.

## Building Façade

### Description and Priority

Programs should utilize characteristics of the building façade, i.e. windows, columns, etc. This is a Medium priority feature.

### Stimulus/Response Sequences

Program objects should interact with building façade. Such as the game Pinball, windows should break when struck, bricks should fall when hit, etc.

### Functional Requirements

REQ-1: Characteristics should be part of the game

REQ-2: Should be scalable for testing on a scale model and use on the building

# External Interface Requirements

## User Interfaces

TBD, control and calibration interface. This will be an interface for the controllers to set up the system and select games, not for the end users. There are no specific requirements for this interface. It must be functional and allow calibration of the controller and selection of programs. The control screen should display on the computer used for the IPM but not on the screen being projected on the building. The screen layout of each program should match the façade of the building. See section 3.2 for layout requirements.

## Hardware Interfaces

The Kinect® controller will connect to the computer using a USB interface. The projectors will be connected to the computer using the standard video output (for Mac this is usually DVI).

Kinect (USB) projector (DVI)

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

Mac OSX, OpenKinect

The Kinect® will connect to a MacOSX machine though USB. The interface for the Kinect® will be created using the OpenKinect C/C++ libraries. The UI for the IPM will be displayed on projectors connected to the standard video output of the machine.

Flow charts and UML diagrams TBD and put in the Appendix

## Communications Interfaces

No communication interfaces will be used with the IPM.

# Other Nonfunctional Requirements

## Performance Requirements

Due to time constraints, each program should run on low system resources to increase speed and reliability.

## Safety Requirements

Cables must be placed out of the way or covered in a manner to prevent users from being tripped.

## Security Requirements

This will be an open source project and therefore no security is required for the project. Git will be used for revisions.

Equipment must be secured and watch before, during, and after event.

## Software Quality Attributes

The movements used to control the programs should be simple to minimize training for using the IPM. The programs should start quickly and run reliably throughout the presentation. The layout of the program should be scalable for testing on a scale model.

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

GiB – GibiByte (230, or 10243) see IEC 60027-2 A.2 and ISO/IEC 80000 specifications

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: Issues List

< This is a dynamic list of the open requirements issues that remain to be resolved, including

TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>