***SOFTWARE REQUIREMENTS SPECIFICATION***

**1.0 Introduction**

This section provides an overview of the entire requirement document. This document describes all data, functional and behavioral requirements for software.

**1.1 Goals and objectives**

Overall goals and software objectives are described.

**1.2 Statement of scope**

A description of the software is presented. Major inputs, processing functionality and outputs are described without regard to implementation detail.

**1.3 Software context**

The software is placed in a business or product line context. Strategic issues relevant to context are discussed. The intent is for the reader to understand the 'big picture'.

**1.4 Major constraints**

Any business or product line constraints that will impact the manner in which the software is to be specified, designed, implemented or tested are noted here.

**2.0 Usage scenario**

This section provides a usage scenario for the software. It organized information collected during requirements elicitation into use-cases.

**2.1 User profiles**

The profiles of all user categories are described here.

**2.2 Use-cases**

All use-cases for the software are presented.

**2.3 Special usage considerations**

Special requirements associated with the use of the software are presented.

**3.0 Data Model and Description**

This section describes information domain for the software

**3.1 Data Description**

Data objects that will be managed/manipulated by the software are described in this section.

**3.1.1 Data objects**

Data objects and their major attributes are described.

**3.1.2 Relationships**

Relationships among data objects are described using an ERD- like form. No attempt is made to provide detail at this stage.

**3.1.3 Complete data model**

An ERD for the software is developed

**3.1.4 Data dictionary**

A reference to the data dictionary is provided. The dictionary is maintained in electronic form.

**4.0 Functional Model and Description**

A description of each major software function, along with data flow or class hierarchy (OO) is presented.

Major Software Components used:

C++ programming language – main component of development

OpenGL – basic graphical rendering libraries

glut – advanced graphical rendering libraries

SDL – audio input and output libraries that work well with OpenGL/glut

**4.1 Description for Function n**

A detailed description of each software function is presented. Section 4.1 is repeated for each of n functions.

SMB.cpp

initSounds – initialize SDL sound files

drawScene – draw the scene

drawPlayer – render the player

AI – artificial intelligence enemy code

boundaryTests – testing boundaries for movement

printToScreen – print bitmap text to the screen

calls drawText to be able to render

smoothMoves – smooth keyboard-based movement

jump – make the player jump

would modify variables in drawPlayer

display – render everything in the game

calls all rendering functions as well as AI and boundaryTests

idle – various items to process while game is idle (time variables, etc.)

interacts with smoothMoves to generate movement WRT time

drawText – allow text to be rendered and drawn to the screen

special\_down – keep track of when special buttons are held down

special buttons are arrow keys, function keys, etc. (no ASCII value)

special\_up – keep track of when special buttons are let go

special buttons are arrow keys, function keys, etc. (no ASCII value)

keyboardUp – keep track of when keyboard buttons are let go

buttons with ASCII values (letters, numbers, etc.)

keyboard – keep track of when keyboard buttons are held down

buttons with ASCII values (letters, numbers, etc.)

changes variables in drawPlayer

CreateGlutWindow – create a full screen game window

CreateGlutCallbacks – call glut functions constantly for checks

functions called: keyboard, keyboardUp, special\_down, special\_up, idle, display

also allocate memory to store which key is held down

InitOpenGL – clear the background color to white

main – main loop where functions are constantly called for rendering

functions called: CreateGlutWindow, CreateGlutCallbacks, InitOpenGL, initSounds

**4.1.1 Processing narrative (PSPEC) for function n**

A processing narrative for function n is presented.

**4.1.2 Function n flow diagram**

A diagram showing the flow of information through the function and the transformation it undergoes is presented.

**4.1.3 Function n interface description**

A detailed description of the input and output interfaces for the function is presented.

**4.1.4 Function n transforms**

A detailed description for each transform (subfunction) for function n is presented. Section 4.1.4 is repeated for each of k transforms.

**4.1.4.1 Transform k description (processing narrative, PSPEC)**

**4.1.4.2 Transform k interface description**

**4.1.4.3 Transform k lower level flow diagrams**

**4.1.4.4 Transform k interface description**

**4.1.5 Performance Issues**

Special performance required for the subsystem is specified.

**4.1.6 Design Constraints**

Any design constraints that will impact the subsystem are noted.

**4.2 Software Interface Description**

The software interface(s)to the outside world is(are) described.

**4.2.1 External machine interfaces**

Interfaces to other machines (computers or devices) are described.

**4.2.2 External system interfaces**

Interfaces to other systems, products or networks are described.

**4.2.3 Human interface**

An overview of any human interfaces to be designed for the software is presented.

**4.3 Control flow description**

The control flow for the system is presented with reference to Section 5.0 of this document.

**5.0 Behavioral Model and Description**

A description of the behavior of the software is presented.

**5.1 Description for software behavior**

A detailed description of major events and states is presented in this section.

**5.1.1 Events**

A listing of events (control, items) that will cause behavioral change within the system is presented.

Menu – This will display a graphical menu to the user, and will display all of the options contained in the system.

New Game – This option will be selected from the menu, and will create a new game session for the user with all values set at default. The default value for extra lives will be 2.

Pause Game – This event will be initiated by pushing the p key on the computer’s keyboard, and will allow the user to pause their current game session. As there will be no data persistence, the game will only be paused for as long as the user’s computer is running.

Resume Game – This event will be initiated by pushing the ENTER key on the computer’s keyboard, and will allow the user to resume a previously paused game with game settings as they were at the time the pause occurred.

Quit Game – This option will be selected from the menu, and will allow the user to exit from the game. This will cause all game-related processes running on the computer to terminate.

Sound On/Off – This option will be selected from the menu, and will allow the user to turn the game sounds on or off. The game will not be sound dependant, so this will not affect the playability of the game.

Select Character – This option will allow the user to select either the male Mountaineer character or the female Mountaineer character to use in gameplay.

Jump – This event will be initiated by pushing the up arrow on the computer’s keyboard. Jump will cause the playable game character to jump in the 2d space of the game. Jumping is the main way of attacking enemy characters. To kill them, the user must skillfully jump on the heads of the enemy characters.

Move Right – This event will be initiated by pushing and holding the right arrow on the computer’s keyboard. Move Right will cause the playable game character to move to the right in the 2d space of the game for the duration of the key being held down by the user.

Move Left – This event will be initiated by pushing and holding the left arrow on the computer’s keyboard. Move Left will cause the playable game character to move to the left in the 2d space of the game for the duration of the key being held down by the user.

Duck – This event will be initiated by pushing and holding the down arrow on the computer’s keyboard. Duck will cause the playable game character to crouch, or duck, under other objects in game play. The character will remained crouched for as long as the button is pressed by the user.

Go to Menu – This event will be initiated by pushing the Esc key on the computer’s keyboard. Go to Menu will cause the user to be directed to the games main menu, where they may choose what they wish to do from the available choices on the menu.

Throw Item – This event will be initiated by pushing the spacebar on the computer’s keyboard. If the playable game character is in possession of throw-able items, pressing the space bar will throw the items when the space bar is pressed for the duration of the possession of said items.

Touch Enemy Character – Coming into contact with an enemy character other than when jumping on top of them will result in a loss of health and items from the playable character. Depending on the health status of the character, the contact may cause the playable character to die.

Fall Off World – Falling off of the world happens when the playable character falls down a hole in the 2d world. This results in the death of the playable character.

Collect Item – Items will be collected throughout game play. There will be a variety of different items, which will cause different events to happen. Those items are:

* Football Helmet - These items will cause the playable game character to grow in size, increase health, and will allow special items to be collected and utilized.
* Beer Cans – These items will be collected for points as well as extra lives. Collecting 100 of them will grant the playable character an extra life.
* Musket Balls – Collecting this item will enable the playable character to shoot musket balls at enemy characters.
* First Aid Kit – Collecting this item will grant the playable character an extra life.

**5.1.2 States**

A listing of states (modes of behavior) that will result as a consequence of events is presented.

This section describes the main states of behavior that will result from an action being presented.

Main Menu – Allows the user to choose what they would like to do. The options in the main menu are: New Game, Quit Game, and Sound On/Off.

New Game – The action of the user selecting the new game button from the main menu will result in a new game session being created for the user. All game values will be set to default in this state, and the playable character will be positioned at the start point in the game.

Pause Game – The action of selecting the pause game button from the main menu will cause the application to temporarily stop running, and all aspects of the game will be retained until the user resumes the game. A constraint of this state is that the computer must be kept on for the game data to be retained at the paused position. There is no way to save a game session in memory.

Resume Game – The action of selecting the resume game button from the main menu will allow the user to resume playing a previously paused game session. Game play will resume precisely where it left off when the game was paused. Again, the computer must remain on for all time between pausing and resuming the game session.

Game Over – This state occurs when the playable character runs out of lives in the game. The user may choose to start a new game, or exit the game from this state.

Game in Session – This state occurs when the user is currently engaged in a game session. The user may perform all in-game actions in this state, and can press the Esc key to switch to the main menu of the system.

Quit Game - Selecting the Quit Game option from the Menu allows the user to exit from the game system. The user will be prompted with a question to make sure they wish to exit, if they confirm that they wish to exit, then the system will be exited.

Sound On/Off - Selecting the Sound On/Off option from the Menu allows the user to turn the sound on or off in their current game session. The default value for sound will be on.

**5.2 State Transition Diagrams**

State Transitions of the Super Mountaineer Brothers system.

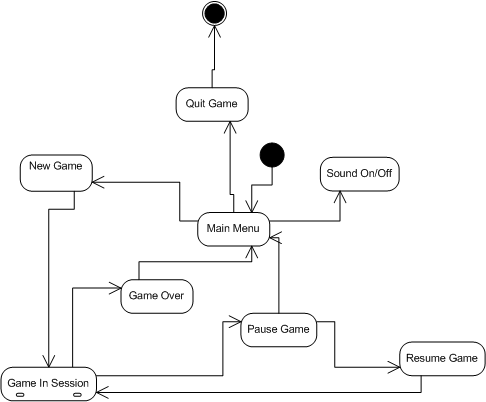


Figure 5-1 State Transition Diagram

Game in Session State Transition Diagram

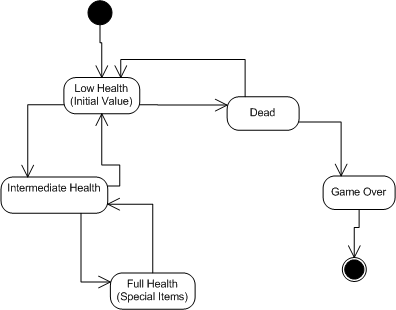


Figure 5-2 Game in Session State Transition Diagram

**5.3 Control specification (CSPEC)**

The first method where the software manages control is through opening the game application. Once the user is in the game environment, they will be directed to the main menu of the game. They will be presented with the options of “New Game”, “Quit Game”, and “Sound On/Off”. If the user selects “new Game”, the system will generate a new game session for the user to play. While in game play, the user may press “P” to pause the game. This will cause the system to pause game play. When in the paused state, the user may press “ENTER” to resume game play. At any point during game play or game paused, the user may press “Esc” to be directed to the main menu.

If the user selects “Quit Game” from the main menu, a prompt will be displayed to verify if they wish to quit. If they answer yes, the game application will be exited. If the user selects “Sound On/Off” from the main menu, they will be given the option to turn the game sounds on or off. By default, sound is turned on.

**6.0 Restrictions, Limitations, and Constraints**

The game application will run on a personal computer, and as such it will be restricted to the limits of that machine. The game will require that the personal computer be running a Windows operating system, XP or later. The user must have sound drivers installed on the computer in order for the game sound to be played. Also, the user must have a keyboard to play the game.

**7.0 Validation Criteria**

The validation process is intended to be a way of testing the game application to ensure that all possible errors are caught and handled properly and the system does not crash. The possible error scenarios are listed with their appropriate response in the following section.

**7.1 Classes of tests and Responses**

**7.1.1 Incorrect keyboard input**

If the user presses incorrect keys in an attempt to control the game, the system will ignore these keystrokes and wait for a correct key to be pushed.

**7.2 Performance bounds**

Due to the fact that this application is a 2d graphical game, graphical rendering will be limited by the graphics card contained in the computer the game is being played on. The game sounds depend on sound drivers and some sort of speaker being installed on the local machine as well.

**8.0 Appendices**

Presents information that supplements the Requirements Specification

**8.1 System traceability matrix**

A matrix that traces stated software requirements back to the system specification.

**8.2 Product Strategies**

If the specification is developed for a product, a description of relevant product strategy is presented here.

**8.3 Analysis metrics to be used**

A description of all analysis metrics to be used during the analysis activity is noted here.

**8.4 Supplementary information (as required)**