Software Requirements Specification

for

KR0Z Text Based Adventure

Version 1.0 approved

Prepared by Christian Bravo

Exotic Bird, Inc.

November 24, 2013

Table of Contents

[1.Introduction 3](#__RefHeading__632_1844403179)

[1.1Purpose 3](#__RefHeading__634_1844403179)

[1.2Document Conventions 3](#__RefHeading__636_1844403179)

[1.3Intended Audience and Reading Suggestions 3](#__RefHeading__638_1844403179)

[1.4Product Scope 3](#__RefHeading__640_1844403179)

[1.5Product Perspective 3](#__RefHeading__642_1844403179)

[1.6Product Functions 3](#__RefHeading__644_1844403179)

[1.7User Classes and Characteristics 4](#__RefHeading__646_1844403179)

[1.8Operating Environment 4](#__RefHeading__648_1844403179)

[1.9Design and Implementation Constraints 5](#__RefHeading__650_1844403179)

[1.10User Documentation 5](#__RefHeading__652_1844403179)

[1.11Assumptions and Dependencies 5](#__RefHeading__654_1844403179)

[2.External Interface Requirements 5](#__RefHeading__656_1844403179)

[2.1User Interfaces 5](#__RefHeading__658_1844403179)

[2.2Hardware Interfaces 5](#__RefHeading__660_1844403179)

[2.3Software Interfaces 5](#__RefHeading__662_1844403179)

[2.4Communications Interfaces 5](#__RefHeading__664_1844403179)

[3.System Features 5](#__RefHeading__666_1844403179)

[3.1Multiple Outcomes 5](#__RefHeading__668_1844403179)

[3.2Options 6](#__RefHeading__670_1844403179)

[3.3Saving and Loading 8](#__RefHeading__672_1844403179)

[4.Other Nonfunctional Requirements 8](#__RefHeading__674_1844403179)

[4.1Performance Requirements 8](#__RefHeading__676_1844403179)

[4.2Safety Requirements 8](#__RefHeading__678_1844403179)

[4.3Security Requirements 8](#__RefHeading__680_1844403179)

[4.4Software Quality Attributes 8](#__RefHeading__682_1844403179)

[4.5Business Rules 9](#__RefHeading__684_1844403179)

[5.Other Requirements 9](#__RefHeading__686_1844403179)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Christian Bravo | 11/24/13 | Initial Creation | 1.0 |
|  |  |  |  |

# Introduction

## Purpose

This is a SRS for KR0Z - Text Based Adventure.

## Document Conventions

Every requirement is to have its own property.

## Intended Audience and Reading Suggestions

The intended audience is my team of programmers and QA specialists that are working on the project, as well as anyone interested in downloading our app to test and/or play full time upon completion.

## Product Scope

The product scope is to bring a text based adventure game similar to ZORK to the Windows 8 app store. The game will be a “choose your own adventure” game based almost entirely on text being the way to portray the world to the user. Each decision a player makes may change the path they end up taking and will even influence the finale that is created when the game is finished.

## Product Perspective

KR0Z is a new product that will stand on its own among other similar games. The game will be acquired through the Windows 8 app store and will run as such. Thus, Windows 8 or above is a requirement to allow this software to run, and will be the only way that it will be able to be acquired.

## Product Functions

1. User Input
   1. Type in commands
   2. Mouse/Touch navigation of pages
2. User Interface
   1. Display previous moves in a list
   2. Display user options
      1. Save
      2. Exit
      3. Main Menu
3. Options
   1. Save on Exit
      1. Prompts user on exit of game if they would like to save character/map data
   2. Sound Management
      1. Sound on/off
      2. Music on/off
      3. Sound Volume Control
      4. Music Volume Control
   3. Difficulty
      1. Lollygag
      2. Normal
      3. Hardcore
4. Saving and Loading
   1. Character Selection Menu
   2. Save game data to file
5. Story
   1. Multiple outcomes
   2. Unlockable starting items on new characters

## User Classes and Characteristics

The user classes for this product will be distinguishable as those who have played text based adventures and those who have not. The distinguishing factors between the two will be the question of how long the user will end up using our product. Those who have played text based adventures will probably end up using our product more often with a longer period of usage than those who are not familiar with text based adventures. The most important user classes for this product are those who have the will to be invested in our product, while those who are less important to satisfy would be people interested in more interactive gameplay and those of a younger age who rely more on visual stimulus to remain attentive.

## Operating Environment

The software will operate on any hardware that is capable of running Windows 8 applications. This is not limited to computers, with touch tablets being included in our software goals with touch input integration.

## Design and Implementation Constraints

The main constraints that will limit the options available to the developers will be knowledge of the Windows 8 SDK and the ability to implement touch capabilities on hardware that is not touch compatible. The only other constraint would be if the product would get accepted into the Windows 8 app store.

## User Documentation

The way the user will know how to use our product will be with in-game documentation and a help command within the game that will give the user any important information as well as a list of commands for the game.

## Assumptions and Dependencies

The constraint of time available per person during the development cycle will be the main constraint that will affect the requirements stated in this SRS. The ability of each person in the development team to use features they have never used before will also play a major role in how fast each feature will be implemented.

# External Interface Requirements

## User Interfaces

The User Interface will consist of buttons used alongside a standard text box for input. The menu screen will contain buttons for character creation, game start, game data loading, and sound options. The game itself will contain only a text input box and confirm button for user input, with story text being displayed in a scrolling fashion so as to allow the retracing of steps.

## Hardware Interfaces

The app will be compatible with any hardware that is able to run the Windows 8/8.1 operating system. This will include not only personal computers (desktop or laptop), but also tablets. This means that the app will be compatible with mouse/keyboard input, touch input, and stylus input.

## Software Interfaces

The game will be compatible with Windows 8/8.1 as an app only. If one of the optional features is implemented, Facebook will also be interfaced with to allow the sharing of stories between users on Facebook.

## Communications Interfaces

The game is not currently scheduled to feature any sort of network communications between users in the app. For version 1.0, the app will be completely standalone per user, with the optional Facebook sharing being the only way for users to show off their stories through a network.

# System Features

## Multiple Outcomes

3.1.1 Description and Priority

The game’s story will be the main focus of development, requiring the most thought. The story is to be done in a manner which allows for multiple playthroughs based on the user’s decision making. Each decision taken by the user will change the path that will be taken, and may also change the final outcome of the story. This feature is of the highest priority, and will be the first feature to be completed.

3.1.1.1 Decision Making Example

“You stand in an open field.”

Input: ‘Look around’

“You notice a field to your west, a house to your north, a lake south of you, and a desert to your east.”

Input: ‘Slit throat’

“You slit your own throat and die. Game Over.”

3.1.2 Stimulus/Response Sequences

Any action taken by the user will stimulate the story in some way, either changing the outcome of events or changing the environment around the user entirely. The story will dictate what each action by the player does for the outcome of the character’s tale.

3.1.3 Functional Requirements

3.1.3.1 Change end outcome based on encounters and decisions made.

* Take input from user for specific encounters and change story path if certain conditions are met (TBD)

3.1.3.2 Take invalid commands or deadly choices and revert them in some fashion.

* If the user gives an invalid command, display error text to the screen and revert to previous state.
* If a choice would have deadly consequences, end the game after displayed text letting the user know they had died from their choice.
  + If difficulty is lollygag, (See below) revert to the previous state instead of ending the game.

3.1.3.3 Unlock items for new characters to start with upon completion.

* After the user completes the story, unlock new starting items based upon what was done in the story.

## Options

3.2.1 Description and Priority

The game’s options will be the user’s way to create characters, change save options, manage sound and music, and change the difficulty of a particular save data. This feature will be of high priority, and will be one of the app’s core functions.

3.2.1.1 Difficulties

The game will feature 3 difficulties. The difficulties will be as follows:

* + - Lollygag – No penalty for wrong choices. If death occurs, immediate retry of previous choice will occur
    - Normal – Upon death, the game will restart from the beginning of the story on the same map
    - Hardcore – Death is permanent for both character and map. Upon death, immediate termination of current game map will occur, and the character killed will be deleted from the save data slot. No in game saving will be allowed. Saving will only occur on exit, and will always occur even without user input.

3.2.1.2 Sound Options

The sound options will consist of sliders and buttons for sound control. This will include:

* Sliders for how loud/quiet the user wants music/sound effects
* Buttons for sound/music on/off

3.2.1.3 Character Creation

Character creation will simply be giving a name to your character. Characters will have a choice of starting items that will end up being unlocked upon completion. Each character will have multiple save files, allowing you to embark on multiple journeys with the same character.

3.2.1.4 Save Options

Save options in the main menu will be buttons to load external data and a toggle for asking if the user wants to save data on exit or not.

3.2.2 Stimulus/Response Sequences

Each option in the menu screen will have its own button for selection. Character creation will also feature a text input box for naming your character without the need for an on-screen keyboard for personal computers.

3.2.3 Functional Requirements

REQ-1: Allow for character naming and creation.

REQ-2: Have buttons which serve as command selection for options.

REQ-3: Allow for music/sound volume manipulation

REQ-4: Have difficulty selection change what happens upon player failure.

REQ-5: Save on exit toggle

## Saving and Loading

3.1.1 Description and Priority

Saving and loading will be the user’s only means of keeping their progress on a particular character’s story stored for long term use. Saves will be stored individually and will hold various bits of information in a manner yet to be determined. Saves, for testing purposes, will initially be stored as mere text files which can be easily created to allow for specific scenario testing without needing to play through the entire story to reach the element to be tested.

3.1.2 Stimulus/Response Sequences

Any action taken by the user will stimulate the story in some way, either changing the outcome of events or changing the environment around the user entirely. The story will dictate what each action by the player does for the outcome of the character’s tale.

3.1.3 Functional Requirements

REQ-1: Allow for saving and loading of character data

REQ-2: Allow for deletion of characters or specific save data

REQ-3: Allow ability to copy a particular save data

REQ-4: Have each character have separate map/story files within.

# Other Nonfunctional Requirements

## Performance Requirements

The app should be able to run perfectly fine on any system that is able to run Windows 8 apps without issues. This does not include personal computers with parts that are over a certain time and are incapable of running most standard software without system slowdown. Loading and saving will be a bit slow at first, since all data within the game will be saved as is without any sort of algorithm for packing together data

## Safety Requirements

If there is a compatibility problem with a user’s operating hardware and Windows 8 apps, that would be the only possible way that loss could happen. If a user is able to run most apps without crashing, then our product should be able to be ran on the user’s machine.

## Security Requirements

Since the app may end up allowing for Facebook integration and sharing of story scenarios, a possible privacy issue may be found with our product having access to one’s information within the Facebook website. The user will not be forced to have Facebook sharing enabled, however, and the app will only do so at the user’s discretion. Our goal is to allow interactivity between friends, allowing the user’s to show off their story to others. It is not to force a separate requirement of social interaction in order to continue the story.

## Software Quality Attributes

KR0Z will be an app that will be easy to use, but not necessarily easy to learn. Most actions will happen at the click of a button, which will ensure that most users know how to begin. The difficulty of learning will end up being one’s ability to understand each command, how it’s used, and any logistics used with a command. The learning curve will also feature remembering where you have gone and building a mental map of the story world. (unless visuals for the map are also included within version 1.0)

## Business Rules

The player should be able to perform all actions within the game on their own. There will not be any restrictions placed for only the developers within version 1.0. If networking and multiplayer story traversal is done for future revisions, then some restrictions will be placed between host and client.

# Other Requirements

Development will take place using GitHub as a repository for the project, and various communications methods for each team member will be used to ensure that issues and comments are expressed timely and correctly.

Appendix: Glossary

SDK – Software Development Kit

ZORK – Original Text Adventure game. More information at <http://en.wikipedia.org/wiki/Zork>

QA – Quality Assurance