2022

Commented [BCP1]: To keep format coherent Titles, and Body text should be Times New Roman. Text body size is 12

Commented [BCP2]: When adding new sections or titles be sure to format them correctly so it is added to the table of contents:

Using the “Styles” in Home:
1. Title = Style: Heading 1
1.2 Title = Style: Heading 2
1.2.3 Title = Style: Heading 3
1.2.3.4 Title = Style: Heading 4
Etc.

Change to Times New Roman, bold, underline, and black text color.

Once done go to table of contents and update all. Double check the newly added section looks correct.

Scavenger Hunt Requirements

SENIOR DESIGN I AND II

PEDRO BUENO, CHRISTIAN CEREZO, DAVID VICTERY

Version: (2)

Date: October 3, 2022

Table of Contents

1. INTRODUCTION 3

1.1 Purpose 3
1.2 Scope 3
1.3 Definitions, Acronyms, and Abbreviations 3
1.4 References 3
1.5 Overview 3

2. THE OVERALL DESCRIPTION 4

2.1 Product Perspective 4
2.1.1 System Interfaces *4*
2.1.2 Interfaces *4*
2.1.3 Hardware Interfaces *4*
2.1.4 Software Interfaces *4*
2.1.5 Memory Constraints *5*
2.1.6 Operations *5*
2.2 Product Functions 5
2.2.1 Finding Items *5*
2.2.2 Navigating the Map *5*
2.2.3 Dynamic Music *5*
2.2.4 Victory Screen *5*
2.3 User Characteristics 5
2.4 Constraints 6
2.4.1 Regulatory Policies *6*
2.4.2 Interface to Other Applications *6*
2.4.3 Reliability Requirements *6*
2.4.4 Security Considerations *6*
2.4.5 Hardware limitations *6*
2.5 Assumptions and Dependencies 7
2.6 Apportioning of Requirements 7
3. Specific Requirements 7
3.1 External Interfaces 7
3.1.1 Finding Items *7*
3.1.2 Navigating The Map *7*
3.1.3 Dynamic Music *7*
3.1.4 Victory Screen *8*
3.2 Functional Requirements 9
3.2.1 Tapping/Clicking Item Objects Function *9*
3.2.2 Finding Items Function *9*
3.2.3 Navigating the Map Function *9*
3.2.4 Dynamic Music Function *9*
3.3 Performance Requirements 9
3.3.1 Number of Devices Supported *9*
3.3.2 Number of Users Supported *9*
3.3.3 Amount of Information Handled *9*
3.3.4 Touch Response *10*
3.4 Logical Database Requirements 10
3.5 Design Constraints 10
3.5.1 Standards Compliance *10*
3.5.2 Hardware Limitations *10*
3.6 Software System Attributes 10
3.6.1 Reliability *10*
3.6.2 Availability *10*
3.6.3 Security *11*

3.6.4 Maintainability *11*
3.6.5 Portability *11*
3.6.6 Usability *11*
3.6.7 Testability *11*
4. CHANGE MANAGEMENT PROCESS **11**
5. DOCUMENT APPROVALS **12**
6. SUPPORTING INFORMATION **12**

1. Introduction

1.1 Purpose

The purpose of our software is to create a fun interactive game that can be played anywhere and by anyone. Whether it be waiting in line at the grocery store or relaxing at home. Our game is made to provide hours of mindless fun through endless amounts of playable levels.

1.2 Scope

The software being made is a 3D Scavenger Hunt Video Game. The game is designed for quick mindless fun; in other words, serve as a means for entertainment. For example, say the user waiting in line at the grocery store, or perhaps they are bored at home, how about while waiting for food at a restaurant. Scavenger Hunt is perfect for these situations, in fact it can be played almost anywhere since it will be playable offline, perfect for those long flights. The premade levels are crafted to where they are easy enough to complete without having to spend hours to solve but challenging enough to where it proves to be engaging and fun for the user. Some levels might even prove to be a bit challenging to complete while waiting in line. Which is why Scavenger Hunt lets users easily pause and save the level they are on, and resume play at their leisure. Scavenger Hunt generates a level procedurally meaning no two levels will be the same.

1.3 Definitions, Acronyms, and Abbreviations

Graphical User Interface (**GUI**): A form of user interface that allows the user(s) to interact with a device through icons, audio, etc.

User Interface (**UI**): The point of human-computer interaction and communication in device e.g., the display screen.

Application Programming Interface (**API):** A set of definitions and protocols for building and integrating application software.

Operating System (**OS**): Software that supports a device's basic function(s) e.g., Windows, MacOS, iOS, Android.

1.4 References

- **Apple App Store Developer Guidelines, June 06, 2022: This reference provides guidelines that must be followed to ensure the software application being developed will be approved on to the Apple App Store**

1.5 Overview

The following portion of the document provides in depth documentation about on how our software will work. Section 2 gives an overview of the product such as the current constraints. It's also where you will find information on the game's features. As well as what will be needed to be able to download and play the game once developed. Video games are always changing and there's many things that can be done over time to improve gameplay and keep it looking modern. Section 3 provides detailed information on how the software will be maintained over time. It will also provide more information on the relationships between different stimulus and how the software handles them with the features implemented.

2. The overall Description

2.1 Product Perspective

The software being developed is independent and self-contained. It will not connect to an external source. All data such as assets, save files, animations, and algorithms will be a part of the software.

Our game is comparable to games such as Zelda Breath of the Wild, Minecraft, and Slender Man. Our game will have procedurally generated levels like Minecraft. Every time the player plays a new instance of the game the level will be completely different from any previous levels they have played. Zelda Breath of the Wild is an open world role playing game that includes puzzle rooms like the software being developed. There will be interactive puzzles that the user will need to solve. The reward for solving these rooms can be a key that can be used to unlock chest placed around the map that can potentially have a findable item in them or a bonus item that gives the player some benefit. The player can choose to use it in their current level, or it can carry on to the next level.

2.1.1 System Interfaces

- Apple iOS API
- Android API

2.1.2 Interfaces

- Touchscreen GUI Interface

2.1.3 Hardware Interfaces

- The system has no hardware interface requirements

2.1.4 Software Interfaces

- No customer desired APIs have been specified

2.1.5 Memory Constraints

- Our app should not take up much more storage space than a typical app that is comparable.

2.1.6 Operations

- The app itself will contain the normal mode of operation.
- The app will process interactive operations when the app is being interacted with by a user.
- The app will process unattended operations in the form of updates when the app is not being used.
-

2.2 Product Functions

2.2.1 Finding Items

- Once a specified item has been found by being tapped the player gains a point and the item is removed from the map (background).

2.2.2 Navigating the Map

- A user can move the screen across the 3D map (background) using two on screen joy sticks if playing on mobile. Keyboard and Game Controllers can also be used on Console and PC.

2.2.3 Dynamic Music

- As the user moves around the map the music playing changes according to the section of the map the screen is located in.

2.2.4 Victory Screen

- Once the user finds all the hidden objects the system displays a victory screen showing the user’s time and stars achieved in correlation to time elapsed.

2.3 User Characteristics

- Users will come from a variety of educational backgrounds.
- Users will have a variety of levels of technical expertise.
- Users will typically be looking for ways to kill small amounts of time throughout their day.

- Users will come from a variety of age groups

2.4 Constraints

2.4.1 Regulatory Policies

Regulations will need to be followed to submit the app to an online marketplace such as the App Store or Google Play. Guidelines for the former: App Store Review Guidelines. Not following established policies may prevent the game from being published.

2.4.2 Interface to Other Applications

Typically, users do not expect to need to download any additional software to run a game. Thus, after game installation, the user should not require fulfilling any other additional steps to start playing the game. Furthermore, the App Store Review Guidelines specifies under section 4.2.3 that an app must “work on its own without requiring installation of another app to function.”

2.4.3 Reliability Requirements

Both the mobile and desktop web apps will need to be stable well-above average user usage levels ensuring that progress while playing levels or creating levels is not lost or interrupted. In the case that unexpected shutdowns or maintenance periods, steps are taken to save level progress. This may take place in the form of automatically saving a user’s progress after a certain time interval and after every instance of a user collecting an item or solving a puzzle.

2.4.4 Security Considerations

For the game to function properly, users should not be allowed to modify elements of the game, including data pertaining to a player’s level score, bonus items found, found items, or game assets. Furthermore, in the interest of security, to access and download the game, a user should have a user account of some sort. Account creation and administration may be handled separately by digital distribution services such as Google Play, the App Store, and Steam meaning that development of a login system and a database for user accounts will not be necessary in order to verify that any user that downloads the app is not a malicious actor.

2.4.5 Hardware limitations

A user’s device resolution will impact the usability of the mobile app. Means of adjusting/scaling GUI components will therefore need to be addressed for the mobile app to be comfortably used by the majority of phone and tablet users.

2.5 Assumptions and Dependencies

- The chosen game engine allows for cross-platform development
- Tools for 3D game development are included in the game engine of choice
- Means of testing the game on all chosen platforms are available

2.6 Apportioning of Requirements

3. Specific Requirements

3.1 External Interfaces

3.1.1 Finding Items

- Once a user picks up an item the system decrements the total number of items. If the user has found all the required items, then the system registers this and displays a victory screen. This allows the scavenger hunt game to progress and end when needed.
- Input will come in the form of the user providing movement input and taps/clicks to pick items up. The point values will be integers from zero to the number of objects needing to be found.
- The object and its counter should respond immediately.
- The total number of objects left to find will be output to the screen.
- Command formats: events and event handling.
- If all found, victory screen.

3.1.2 Navigating The Map

- On PC the user navigates the map by using the mouse and keyboard. The system responds to this input by moving the screen in the desired direction.
- On mobile input is provided from the user dragging the on-screen joysticks.
- The output shown on the screen is the movement of the map.
- The user can only move to the edge of the map and no further.
- The movement along the map will be measured by standard units in the software used to create the map.

3.1.3 Dynamic Music

- As the user moves around the map the music playing changes according to the section of the map the screen is located in. This change might simply be changing static music tracks or be more dynamic in that the same song will be playing but change individual instrument tracks.

- Input to the music engine will come in the form of the user screen being in a certain quadrant of the map. The music will change tracks according to this location and output audio to the user.
- Tracks will change immediately when the user enters a certain quadrant of the map.
- Music data will be organized as separate tracks of a song which are dynamically switched on and off to form the greater whole of the song playing.

3.1.4 Victory Screen

- Once the user finds all the hidden objects the system displays a victory screen showing the user's time and score determined by the time elapsed.
- Output will be in the form of a screen shown to the user.
- The victory screen will take up most of the screen and display user stats such as the time it took to complete the level and the score based on the time elapsed.

3.1.5 Level Generation -

- Levels in game are procedurally generated by an algorithm meaning no two levels will be the same.
- Environmental assets as well as placement of findable objects will be spread out by an algorithm.

3.1.6 Found Items User Interface

- Items that need to be found can be located at the bottom of the screen which makes easy accessibility for the player.
- They will also have the option to view all items at once by clicking a dedicated button to view.

3.1.7 Multiplatform

- The game will be playable on mobile, PC, and console.
- There will be no additional features that are exclusive to any platform.

3.1.8 Level Difficulty

- There will be three set difficulty levels: Easy, Medium, and Hard. These will be determined by the number of items that need to be found and the complexity of the puzzles

3.2 Functional Requirements

3.2.1 Tapping/Clicking Item Objects Function

- The system will handle an event for an object provided that object has been interacted by the user.

3.2.2 Finding Items Function

- The system will only decrement an object's quantity number provided the appropriate object was interacted with.
- The user will tap the hidden object which will then remove the object from the game and update the save system. In the event the found all items meaning there are zero left on the map, a victory screen will be displayed.

3.2.3 Navigating the Map Function

- The system will ensure that the user can't navigate off the map.
- The user will either use on screen joysticks (mobile) or keyboard and mouse for navigation. The screen will then move across the map in a manner corresponding to how the swipe was done.

3.2.4 Dynamic Music Function

- When the user crosses into a certain quadrant of the map, the music will change accordingly.
- The system will ensure that the music can't be too rapidly changed in this manner.

3.3 Performance Requirements

3.3.1 Number of Devices Supported

- The system will support as many devices as there are users who can obtain the app as there will be no online functionality at this stage.

3.3.2 Number of Users Supported

- Each user will use the app independently.

3.3.3 Amount of Information Handled

- Each device running the app will need to be able to handle all the information required to run the game (vague).

3.3.4 <u>Touch Response</u>

- 95% of the inputs provided by the user in the form of taps, swipes, and pinches should be processed within 100 milliseconds.

3.4 <u>Logical Database Requirements</u>

Logical Database will not be used as files/assets are stored locally

3.5 <u>Design Constraints</u>

3.5.1 <u>Standards Compliance</u>

- The app follows the guidelines set by Apple's App Store Review Guidelines

3.5.2 <u>Hardware Limitations</u>

- The app requires at least 200 MB of storage on user's device. (Subject to change)

- The app runs on the following mobile device OS versions:
  - Android 5.0 or later
  - iOS 10.0 or later

3.6 <u>Software System Attributes</u>

3.6.1 <u>Reliability</u>

- Software updates should not alter a user's progress in a level. Progress includes items found, player location, and time elapsed.

- The mobile app runs for multiple hours (at least 10) without crashing.

- The PC app runs for multiple hours (at least 10) without crashing.

3.6.2 <u>Availability</u>

3.6.2.1 <u>System Recovery</u>

- The application performs an auto-save at least every 30 seconds so that progress in a game may be recovered after any unexpected shutdown. User progress consists of scavenger hunt items found, and time elapsed during the game.

3.6.2.2 <u>System Update</u>

- App downtime will not exceed 5 minutes per month. Downtime consists of pushing bug and security fixes, ~~updating the pre-loaded level library~~, adding any new features, and performance updates.

3.6.3 Security

- Users cannot make modifications to their time elapsed (affects score)
- Users cannot make modifications to game assets
- The game is made available on digital distribution services that require users to have an associated account before the game can be downloaded.

3.6.4 Maintainability

Specify attributes of software that relate to the ease of maintenance of the software itself. There may be some requirement for certain modularity, interfaces, complexity, etc. Requirements should not be placed here just because they are thought to be good design practices. If someone else will maintain the system

3.6.4.1 Pre-loaded Level Maintenance

- The app supports updates that provide new hunt items, and/or changes to the UI.
- Maintenance for the app such as bug fixes will also be done through app updates.

3.6.5 Portability

3.6.5.1 Supported Operating Systems

- The application is compatible with Android, iOS, Windows, and macOS operating systems.

3.6.6 Usability

- UI elements are scaled appropriately based on a user's device resolution.
- The game does not require the installation of additional software for a player to run the game.
- The game is colorblind friendly; puzzles do not require a user to distinguish between various colors in order to solve the puzzle

3.6.7 Testability

- The mobile app is testable on supported Android and iOS devices, and Windows and macOS operating systems

4. Change Management Process

5. <u>Document Approvals</u>

6. <u>Supporting Information</u>