**CSCI 4600 Vision Document**

Erik Anderson

Benjamin Seifried

**Adventure Game Program**

2023-02-01

1. **Introduction**
   1. **Purpose**

This document will outline the vision for the proposed “adventure game”, hereafter referred to as “game” in lieu of a working title. The purpose of the following document will be to:

* Outline the Purpose and Scope of the project.
* Identify Business Opportunities that justify the creation of a game.
* Identify real and potential problems in production of the game.
* Present Stakeholder and User Descriptions
* Present a Product Overview
* Outline Product Features that will be included in the game program.
* Identify Constraints
  1. **Scope**

This document applies to the entire development of the “game”, which will be developed by Erik Anderson and Benjamin Seifried. The “game” will be developed to run on Microsoft Windows by means of running an .exe file.

1. **Positioning**
   1. **Business Opportunity**

Modern gaming utilizes various genres and aesthetic styles to create an entertaining experience for users. The specific genres and aesthetic styles that have been most prominent in the modern games of any given time period have changed dramatically as the video games industry has matured. A direct result of this is that certain genres and aesthetic styles have waned significantly in popularity. Furthermore, individuals that grew up when certain genres or aesthetic styles were popular often maintain nostalgia for these genres and aesthetic styles. Thus, there exists an opportunity to create an adventure game formatted to appeal to the nostalgia held by one or more age groups of gamers, while at the same time containing gameplay that appeals to others who lack nostalgia. This in turn means there exists an opportunity to generate interest and profit.

* 1. **Problem Statement**

|  |  |
| --- | --- |
| **The problem of** | A lack of simple retro-styled RPG games. |
| **Affects** | Nostalgic gamers, and younger gamers that want to see something new. |
| **The impact of the problem is** | A gap in the video game market that can be filled. |
| **A successful solution would provide** | An engaging video game with a retro aesthetic and basic RPG mechanics. |

1. **Stakeholder and User Descriptions**
   1. **Stakeholder Summary**

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Skills** |
| **Erik Anderson** | Designer, Programmer, Manager, Engineer | C#, Visual Studio IDE, WPF |
| **Benjamin Seifried** | Designer, Programmer, Manager, Engineer | C#, Visual Studio IDE, WPF, Traditional tabletop game development experience, graphics experience. |
| **Mir Hasan** | Sponsor |  |

* 1. **User Summary**

|  |  |
| --- | --- |
| **User Type** | **Description** |
| **Younger Users (Ages 9-25)** | Users of the game that may be tired of traditional 1st person shooters and will enjoy exploring a choose-your-own adventure type, turn based roleplaying adventure. |
| **Older Users**  **(Ages 25+)** | Older users who will enjoy the welcoming nostalgic or retro gameplay that is different from many modern game formats which can seem prohibitive. |

1. **Product Overview**
   1. **Product Perspective**

Given that the “game” seeks to target a genre and aesthetic that is no longer popular in the mainstream video gaming marketplace, there are few products that demonstrate close relation to the “game”. While this “game” seeks to be a role-playing-game, a vast majority of modern role-playing-games (RPGs) place the player in an open-world 3D environment (such as in the case of Bethesda RPGs), whereas the “game” will have the player operate within a format more similar to a choose-your-own-adventure book. Likewise, while the “game” will have a retro look, most modern games using a retro aesthetic place it over a more modern formulation of whichever genre the game in question belongs to (such as in the case of Vampire Survivors or Tiny Rogues). Ultimately, the closest related product to the “game” would be WayForward’s Goosebumps: The Game, which adopts choose-your-own-adventure mechanics and an early-2000s aesthetic. However, its core mechanics are those of a point-and-click adventure game, whereas the “game” developed here will focus more on RPG mechanics.

* 1. **Summary of Capabilities**

The “game” will be a role-playing adventure game modeled on the tenets of choose-your-own-adventure books with a retro aesthetic, whose mechanics and aesthetic are meant to appeal to both nostalgic video game players and less-experienced players that are looking for a novel experience. The “game” will feature a client-side account system, save system, and leaderboard. These three features will mesh with each other such that multiple players could player the game on the same computer without interfering with each other’s progress. The core of the “game” will operate around the node-based map, character, item, inventory, and combat systems. These will allow players to interface with the in-game world in multiple ways and customize their gameplay styles. An in-game wiki will provide players with easy-to-access information about gameplay mechanics. Finally, a meta-progress shop system will give players the ability to earn points as they play the “game”, which may then be redeemed outside of a playthrough to customize future playthroughs or provide themselves with additional options in-game.

* 1. **Assumptions and Dependencies**

The primary assumption that would critically impact the “game’s” development would be that the “game” will only operate on modern versions of Microsoft Windows with updated .NET frameworks. Should this not be the case, then the “game’s” developers might lack the skills and technical knowledge required to make it playable on other operating systems or with outdated .NET frameworks.

The developers assume that the “game” will be played on a computer with a mouse, and that the player will interact with the “game” with their mouse.

1. **Product Features**
   1. **Account System**

Allow users to create, access, modify, and delete accounts in the “game”. Accounts may be password-protected. Accounts contain a player’s saves, players-specific leaderboards and statistics, and any meta-progress currency or unlocks a player has obtained. This allows multiple players to play on the same computer without interfering with each other’s progress.

* 1. **Save System**

Allow users to save and load game progress, and to maintain multiple save states. This allows players to leave and return to the “game” without losing progress. Saves should indicate when they were created and allow a player to input a small description describing the save. To prevent user-error, the player should be warned before they are allowed to overwrite or delete a save.

* 1. **Leaderboard**

Each account should have a personal leaderboard containing details of past playthroughs, and all accounts should be able to write playthrough information to a “global” leaderboard shared between all accounts. Ranking on the leaderboard should be determined by some system of points. This allows players to easily access their game history, as well as to compare it to other’s game histories.

* 1. **Node-Based Map System**

The “game” world should be composed of many interconnected nodes that players can traverse, with each node containing some story element, combat instance, or other event that engages the player. Events at map nodes ought to be varied enough that multiple playthroughs of the “game” are differentiated, as to avoid boring the player via repetition.

* 1. **Character System**

A player’s character in the “game” should have a variety of statistics representing the character’s skills or abilities, which can then interface with events at various map nodes. The skills that a player’s character begins with should be determined by an archetype or class selected by the player at the start of a playthrough, and the skills should be able to be modified via interaction with events in the game. This permits the player to customize their playstyle, and to vary their interaction with map node events.

* 1. **Item System**

A player should be able to find items at map node events, with these items either permitting them greater options in traversing and interfacing with map nodes or allowing them more ways to customize their playstyle. This provides variety between different playthroughs.

* 1. **Player Inventory**

A player’s character should be able to carry and/or equip a limited number of the items found at various map node events, such that a player may plan ahead by carrying certain key items with them. The inventory should not be so large as to trivialize gameplay, so the player should have to choose what items they bring with them.

* 1. **Combat System**

As the player explores map nodes, they may find entities that oppose them, which they may interact with via combat. Combat ought to interact with the player’s character’s statistics and items, such that an inadequately prepared player is forced to make trade-offs to succeed and continue their playthrough. Combat should not be so difficult as to require a specific combination of character statistics and items to succeed, with possible exception of an end-game battle. Combat should occur in multiple rounds, and a player should be able to take a single action per round. Combat should end when either combatant is defeated, or when a player voluntarily disengages from combat.

* 1. **Meta-Progress Shop System**

As the player progresses through a playthrough, they should earn currency for their account that can be spent in a shop between playthroughs. Currency should also be earned for completing a playthrough. The shop should allow the player to unlock additional content and modify future playthroughs in a variety of ways. Each player’s account will maintain its own currency reserve and set of purchases made at the shop. This provides a player with agency over their future playthroughs, as well as with a sense of progression between playthroughs.

* 1. **In-Game Wiki**

An in-game wiki should be present that contains information about any major gameplay mechanics, items, enemies, and other features that they’ve seen. This would allow players to learn the game and develop strategies more easily.

1. **Constraints**

The constraints and Risks presented in the creation of this game program include those related to time scope and resources available. More specifically, the development of the game and its associated documentation must be completed before the end of the semester.

The game must be created in the Visual Studio IDE using C#, without relying on powerful external libraries, such as Unity.

One risk that is present includes the need for the program to be created in such a way that a minimally viable, fully playable product is presented in the time frame allotted, even though the stakeholders involved have no professional experience with respect to computer game development.

Another risk is that while the scope of the project has been defined with restraint, scope creep can occur and cause the “game’s” development to balloon in both complexity and the time required to complete it.

Two final risks would be that the team members have not worked together before, and neither team member has experience with developing a program of this size or complexity.

1. **Other Product Requirements**

Given this project’s nature as a small, independently developed game, neither developer identified applicable standards for the “game”.

With respect to hardware, the “game” should run on Intel and AMD desktop and laptop CPUs, whereas with respect to software, the “game” should run on Microsoft Windows 10 and 11 at a minimum.

The “game” should be fully loaded within 30 seconds of launching it at a minimum, and preferably within 10 seconds.

No applicable environmental requirements were identified.