FUNCTIONAL SPECIFICATION

HALFWAY HOME

2017

Version History

Version 1.0 – draft

Table of Contents

1. INTRODUCTION, Overview, and Purpose

Halfway Home is a visual novel game with stat-management and puzzle elements. The player will participate in interactive text-based dialog scenes with a diverse cast of characters, manage their time and resources between a small set of room locations, and solve simple logic puzzles.

The final project will be a stand-alone executable that can be run on any PC using Windows 7 or better.

1.1 Outcomes and Scope

The planned outcome of the project is a stand-alone game executable that will demonstrate the developer's abilities to create a compelling character-focused narrative as well as elegant systems design.

Game Goals:

- 7 "days" of content
- 7 characters
    - Player, 3 main, 3 side
- 1-3 hours of gameplay
- Choice-driven dialog trees
- Interactive map navigation
- Full game (main menu, credits, etc.)

Technical Goals:

- Re-useable C# scripts

Design Goals:

- Well-executed, character-driven narrative
- Clean and refined user interface
- Intuitive systems that harmonize well with the narrative experience

Artistic Goals:

- 6 emotional sprite poses per character
- Background art for every room
- A compelling soundtrack containing tracks for each character

1.2 Project Requirements

- Installable on any Windows 7, 8, or 10 PC
- Playable with mouse and keyboard.
- Meets TCRs as set out for GAM400

1.3 Development Requirements

- Unity Personal 2017.1.1 used for project development.
- Visual Studio 2017 (for C# scripts)

- Inkle, a compiler for the narrative scripting language Ink
- Sublime Text 3 (for Ink scripts)
- Art assets will be drawn, inked, scanned, and edited in Photoshop.
- GIT will be used for source control.

1.4 General Constraints

- The project must run well on a standard Windows 7, 8, or 10 PC.
- The project should install all necessary software such as drivers, dlls, etc.
- The project should not require separate installs from the user for any middleware.
- The game should not require developer assistance to complete.
- The project must meet all the required TCRs.

2. FUNCTIONAL DESCRIPTION

2.1 Install and Uninstall

The project should be able to be installed on any Windows 7, 8, or 10 machine with the appropriate hard drive space and other system requirements. An error message should be generated if the project is unable to operate on the machine.

The installer will be simple to use, creating its own directory within the MyDocuments Directory as per Windows 7 requirements. The user data generated will be stored in a separate directory.

The installer will be a full installer, not an unzip or other compression/decompression software.

Uninstalling will remove all installed elements of the project, including any user created content within the user directory.

2.2 Game Mechanics

The game is written in first-person visual novel style. The game is divided into two states: navigation and time allocation via the map interface, and dialog scenes with specific characters and choice interactions. The player manages a set of stats, divided into "wellness" and "social" categories, by allotting time to a location, receiving a stat benefit from the room as well as sometimes engaging in a special scene. At the end of seven in-game days, the week repeats in a "time loop", preserving the character's stats and knowledge but resetting scenes and the like.

2.2.1 Gameplay Managers

The game's principle map and dialog systems are linked via events and properties, allowing seamless transitions between scenes via a node-based structure.

- NODE EDITOR: Links Ink files containing game scenes, informs the map system of unique location events.
- TIMELINE SYSTEM: Schedule of events, scenes, and character actions that drives the map and dialog systems.

- MAP DISPLAY: Interactive interface that displays information for the given time (including character locations via the Icon Display system) and allows players to choose their next actions and scenes.
- DIALOG DISPLAY: Text interface that, via the Description Display, prints text sent from the Story Reader, which parses the Ink JSON for the scene.
- STAGE DISPLAY: Manages which characters are present in the scene, as well as where on the screen those characters are and what pose they are in.

2.2.3 Character Sprites

As mentioned above, there are 7 characters. As currently planned, the player character will not require sprite poses. The other 6 characters have poses listed below:

- Calm
- Happy
- Sad
- Angry
- Afraid
- Surprised

This makes for a total of 36 different sprites, most of which will include a pose change, but at minimum require facial expression changes.

2.2.4 Backgrounds

There are nine rooms in the house which can be visited by the player. Each of these rooms requires a background image.

- Your Room
- Common Room
- Library
- Art Room
- Charlotte's Room
- Café
- Front Desk
- Garden
- Shop

2.2.5 Development of Graphics

The graphics pipeline is handled by Unity. In general, art assets are developed by hand, inked, scanned, and improved upon in photoshop, before being dropped into the game.

2.2.6 Development of Physics

There are no physics in this game

2.2.7 Runtime Menus

See 3.1.1 for all menus that will be added to the game. The game is required to have a main menu and a pause menu. The pause menu must pause the state of the game and either mute or drop in-game audio.

2.3 User Functionality

2.3.1 Use by a Player

A player using the game should be able to figure out the basic controls even without any tutorial instructions. The underlying mechanics of the game, will be explained in narrative, by the player character explaining their experiences to Timothy, the new character in the Halfway Home.

2.3.2 Use by a Potential Employer

An employer using the game will be looking for different things, but by and large the experience is the same. As designers, we need to demonstrate hiring value through actual game flow, not just snippets or video examples.

2.4 Metrics

There are no planned metrics for the project.

3. SPECIFIC REQUIREMENTS

3.1 External Interface Requirements

3.1.1 User Interfaces

a) Hardware interfaces
    a. The project must support keyboard and mouse within the stand-alone operation.
b) Software interfaces
    a. DigiPen Splash Page (can't be bypassed)
    b. Team and/or Game Logo (can't be bypassed)
    c. Main Menu
        i. Start Game – Launches the game from the start
        ii. Credits – Single page that displays all persons credited toward making the game, along with any required copyrights.
        iii. Options (stretch)
            1. Ability to adjust levels of SFX and Music/Ambient audio
            2. Full Screen / Windowed Mode
        iv. v. Quit Game (with confirmation)
    d. Pause Menu (Must pause game and in-game audio)
        i. Return to Game
        ii. Options
        iii. Main Menu (with confirmation)
        iv. Quit Game (with confirmation)
    e. Map Display
    f. Dialog Display

3.1.2 Networking or Communications Protocols
This project will not use networking.

3.1.3 Memory Constraints
There are no memory constraints on this project. However, the developer will be working to limit the processing footprint of the software. The Unity profiler will be utilized to ensure no memory leaks exist and to track processing required during gameplay.

3.1.4 Frame Rate
A target of the project will be to maintain a frame rate of 30fps or higher at all times.

3.1.5 Product Function
The installer should properly handle all required runtimes, libraries, and anything that is needed to run the software. The executable will be retained by the team and shared on a personal basis until such a time as it meets the requirements to be hosted on DigiPen's Game Gallery or published on Steam.

3.1.6 Assumption and Dependency
It is assumed that there will be two developers/designers on the project. Further, it is assumed that the game will focus on the narrative design accomplishment more so than technical or other artistic aspects.

3.2 Software Product Features
No features are planned for this project.

3.3 Software System Attributes
3.3.1 Reliability
The target for reliability is to have no warnings or errors on compile, along with no crashes for progress stopping bugs during gameplay.

3.3.2 Availability
Target availability is to have the software function on idle for 24 hours without any crashes. There will be no hosting database and the software will function without the need for external systems, therefore external downtime will not be an issue.

3.3.3 Security
The target for security is to have the installer pass windows defender installs without any warnings or threats. All assets and content will be protected by Unity compile software.

3.3.4 Maintainability
Any updates to the software will require downloading and installing a full new .exe. There are no plans to support updates, allow for DLC, or provide patch fixes.

3.3.5 Portability
The software will support Windows 7, 8, and 10. As a stretch goal, we may pursue release on Android and iOS tablet devices. Any other platforms or Operating Systems will not be tested and will not be guaranteed to work.

3.3.6 Performance

Target framerate is around 30fps or more. At no point should the software drop below that rate when installed on recommended hardware.

3.4 Database Requirements

There are no planned Data Access components for this project beyond any that are provided from a packaged file provided by Unity

3.5 Other Requirements

There are currently no other requirements for this game.

4. ADDITIONAL MATERIALS

See our Design Document:

https://docs.google.com/document/d/1WdjycKzjwtLdXxBzHb7azbQ6qrKOwlAEdptHmAiqlNA/edit?usp=sharing