Implementation Plan

The Oregon Trail with Python(s)

Red Team

SDEV265 Summer 2023

Contents

[Introduction: 3](#_heading=h.30j0zll)

[Implementation Methods Pros and Cons: 3](#_heading=h.1fob9te)

[Details: 5](#_heading=h.3znysh7)

[Installation Instructions: 6](#_heading=h.2et92p0)

[Summary: 9](#_heading=h.tyjcwt)

# Introduction:

The purpose of our project is to recreate the popular text-based adventure decision game from the ‘70s “The Oregon Trail” using python. This will not be a 1:1 recreation but simply inspired by it. This version will have python puns included in gameplay to reference the language the game is built with.

This is a linear path game where you try to make it to the next checkpoint before running out of food. Each decision you make only affects how much time it takes to get to the next checkpoint or your food rations. If you take too long, you’ll run out of food and die. There are also special scenarios where you’ll get more food, lose food, or randomly die. It all is related to the food countdown or randomized death.

The purpose of this document is to discuss the plan of implementation. We will cover the implementation method used and the pros and cons vs other methods. We’ll then describe how our implementation satisfied the requirements we set at the start of the project. Finally, we’ll cover the instructions for installation of this game. This document can be used as a guide moving forward for implementation of this program.

# Implementation Methods Pros and Cons:

There are four common types of implementation methods that are used in real-world IT projects. Those types include parallel, phased, pilot, and direct. We will describe the differences between the types and the reasoning behind the method we chose.

First, parallel implementation involves running both the existing and new software systems at the same time. This method allows you to compare the old and new systems. It also provides a safety net in case any issues arise during the implementation. We could not choose this option, as we have a brand new system to implement with no prior system to compare it to. Therefore, this would not be a method appropriate for our project.

Second, there is the phased implementation method. The phased method involves implementing the new software system in phases. Each phase is deployed and tested before moving on to the next one. This method allows for a gradual rollout of the entire system which helps you catch disruptions early. This method also does not work for our project as we are working on a limited timeline to try to get a working software by a fixed date. We do not have the time to slowly roll out the program piece by piece.

Third, there is the pilot implementation method. This is when the software would be rolled out to a small group of users to test and provide feedback before rolling it out to the larger audience. Again, we are low on time for this implementation and need a method that allows us to quickly release the project in its entirety. This would take too much time to roll out for our timeline.

Finally, there is the direct implementation method. This is a quick and full implementation of a software. It requires adequate planning, testing, and training. Since our project needs to be a functioning application by a specific date, this is the method we chose. We are performing all testing during development and providing thorough documentation so that it is easy for users to understand and apply. That is why we chose this implementation over the others.

# Details:

The following are user requirements that we set for our project along with a description of how those requirements were met. We were unable to get to the stretch goals in the time allotted, so we focused only on our mandatory requirements.

**Mandatory Requirements:**

1. Base game mechanics consisting of a player trying to traverse from checkpoint to checkpoint that is presented with random scenarios along their journey. They must manage their food and money in order to survive and arrive at the destination.
   1. We created some methods that enable a player to select their character, travel between checkpoints that include towns and rivers, and we created many different scenarios that challenge the player along the way. Built into the mechanics are trackers for distance between checkpoints, food, and money. We also introduced randomization throughout the entire game to enhance the ability to replay the game over and over.
2. A user interface that has a retro feel similar to the original game with sprites for different scenes and scenarios.
   1. We created a basic user interface using TKinter. This interface includes a section for an image, a section for a description of the game play, and two buttons that are linked to game actions. Then, we used AI graphics generating software to create sprites that describe the scenes in the game.
3. This game will be compatible with most personal computers with an installation of Python 3.0 or newer.
   1. We developed the entirety of this software only using base Python libraries as well as one external library called "Pillow”. Please install this dependency with the following command “pip install -r .\requirements.txt.” This allows for users to play this game on any personal computer with an installation of Python 3.0 or newer simply by downloading the game folder from the GitHub repository.

# Installation Instructions:

To play the Oregon Trail game, you'll first need to install Python on your system. Here are the steps to install Python and download the source code:

**Install Python:**

1. Visit the official Python website at https://www.python.org/.
2. Click on the "Downloads" tab and select the appropriate installer for your operating system (e.g., Windows, macOS, or Linux).
3. Download the latest version of Python 3.x.x (where x represents the specific version).
4. Run the installer and follow the instructions provided. Make sure to check the box that says "Add Python to PATH" during the installation process.

**Clone the repository:**

1. Open your preferred command-line interface (e.g., Command Prompt on Windows or Terminal on macOS/Linux).
2. Navigate to the directory where you want to store the game's source code.
3. Execute the following command to clone the repository:  
   git clone <https://github.com/Crstt/SDEV265_GroupRed.git>

**Access the game directory:**

1. Change your current directory to the "oregon-trail-python" folder within the cloned repository:

cd SDEV265\_GroupRed/oregon-trail-python

1. Install the dependency with the following command “pip install -r .\requirements.txt.”

**Run the game script:**

1. Launch the game by executing the following command:

python oregon\_trail.py

If you don't have Git installed or prefer not to use it, you can still download the source code for the Oregon Trail game directly from the GitHub repository. Here's an alternative method to download the source code:

**Visit the Oregon Trail GitHub repository:**

1. Go to the following URL in your web browser: https://github.com/Crstt/SDEV265\_GroupRed.

**Download the source code as a ZIP file:**

1. On the repository page, click on the green "Code" button.
2. In the dropdown menu, click on "Download ZIP".
3. This will download the entire repository as a ZIP file to your computer.

**Extract the ZIP file:**

1. Once the ZIP file is downloaded, locate it on your computer and extract its contents to a desired location.
2. You can use a built-in ZIP extraction tool or a third-party software like 7-Zip or WinRAR.

**Access the game directory:**

1. Open the extracted folder that corresponds to the downloaded ZIP file.
2. Within that folder, navigate to the "oregon-trail-python" directory.
3. Install the dependency with the following command “pip install -r .\requirements.txt.”

**Run the game script:**

1. Open your preferred command-line interface (e.g., Command Prompt on Windows or Terminal on macOS/Linux).
2. Change your current directory to the "oregon-trail-python" folder you located in the previous step.
3. Execute the following command to start the game:

python oregon\_trail.py

**Requirements:**

Python 3.0 or above: Ensure that you have Python 3.0 or a newer version installed on your system. You can verify the installation by running the command python --version in your command-line interface.

# Summary:

In conclusion, this document describes the plan of implementation for the Oregon Trail with Python(s) game. We will use a direct implementation method to deploy a working game engine with GUI that is accessible to anyone with an installation of Python 3.0 or greater. We also discussed the installation instructions in this document. Please use this document for implementation of the Oregon Trail with Python(s) game.