Pangeo Current State

April 24, 2020

This document is provided as a brief summary of the current condition of the Pangeo project.

Note 1: For any future contributors: The basic project is completed, but has not been alpha/beta tested. As such, when downloading the Github repository from https://github.com/VRtualize/VRtualize-Pangeo.git, please note that the code in the Master branch is only version 0.1. Referencing the Test Plan document, the code has not been run through various stages of higher complexity testing- only unit tests and visual tests have been completed, integration testing and beyond still needs to be done- and as such has not been approved to be published to the Master branch. As such, the Integration branch contains the most current iteration of the project prior to full project testing.
Note 2: When using the application, please note that functionality is implemented for both keyboard and VR controls, but due to time constraints we don't have functionality to switch control schemes dynamically, and the two methods conflict (see Future Work below). Please select the appropriate scene when loading the application: the keyboard tagged scenes when you don't have VR equipment, and the VR tagged scenes when using VR functionality.

Code Structure

This section describes the file structure of the final Pangeo project. The Unity Engine has certain layout expectations concerning some special folders and files in a Unity project, and this document is designed to help enlighten future developers as to where files should go.

- **Documentation Folder** - The root folder for any technical documentation that is passed along with the Github repository.
- **Pangeo Folder** - This folder contains the Unity project that can be opened and compiled from the Unity Editor. The Project is split into 3 main subfolders, the Assets folder, the Packages folder, and the ProjectSettings folder.
  - **Assets Folder** - This folder contains the bulk of the project, as well as certain folders and files necessary for Unity to run correctly. The following folders include files that have been written by the developers, dependencies, and resources.
    - **Animation** - contains folders for fully animating the Main Menu scene.
    - **MySQL Packages** - files that are required to run and properly connect to a local MySQL database.
    - **Resources** - this folder contains Unity assets that help the program. The Prefabs folder contains all of the Unity Prefabs that were created to help develop the project. The Textures folder contains all images used in UI elements of the application.

        - **Scenes** - this folder contains each scene important to the application, the Main Menu scene, which handles the main menu functionality and the WhenWorldsCollideWithMovement scene is the main scene where most of the application takes place and the user can explore the environment.
        - **Scripts** - this folder contains all of the scripts that define the behaviors of the GameObjects in the application. Files in the folder contain logic for running the majority of the program, like cache logic and resource manag, while subfolders contain supporting scripts to support user interaction. The MenuScripts folder contains scripts that relate to UI elements in the Main Menu. MovementScripts contains scripts that handle any input from the keyboard or VR controllers.
        - **SteamVR** - this folder contains generic GameObject assets that represent VR equipment, like controllers, in application scenes.
        - **SteamVR_Input** - this folder contains scripts that help read input from VR equipment, the controller button presses, and camera orientation.
        - **Tests** - this folder contains all test files used for testing the application in Unity's built-in test runner. It also contains two subfolders to account for tests that aren't supported by the test runner:
            - **VisualTests** - these are tests that must be tested by a person to visually confirm the success or failure of each test.
            - **DatabaseTests** - these are tests that test functionality and correctness of the database
    - **Packages Folder** - This folder contains subfolders for each package used to run and support the program.
    - **ProjectSettings Folder** - This folder contains all files that help define settings for various components of the application.

Future Work

Certain features became too large to complete within the allotted time frame and had to become stretch goals for future work.

- **Dynamic Resolutions** - The system uses Bing Maps as its primary Cooperating System to download elevation data and image data from. Bing Maps supports multiple zoom levels of both data and image resolutions, which changes how precise the data in a given area can be displayed. The first stretch goal would be to implement varying resolutions as the user moves up and down in the virtual environment.
    - Going higher would entail lowering the resolution of the zone and being able to display more terrain chunks in a larger area around the user.
    - Going lower would entail increasing the resolution of chunks around the user at the cost of constructing less tiles.
- **Memory Performance Upgrades** - Two main methods to help improve performance of the project are listed below

    - **On-Disk Cache** - Potentially convert Local MySQL Database to an On-Disk Cache system to solve locality issues. Alternatively, a smaller database coupled to the application might be an apt replacement, like SQLite.
    - **Predictive Model** - Prediction of user movement before it happens can help performance by allowing the platform to pre-load expected tiles before the user encounters them, reducing fetch times for tile data and load times into the application.Otherwise, if the user is loading a brand-new tile into the application for the first time, the user may have to wait a long time for fetching the data from the external data location.
    - **Enhanced Tile Replacement** - Currently, the platform keeps a queue of all tiles added in the order they were added to the application. When the platform needs to load new tiles and doesn't have space, it simply removes the oldest tiles from the queue, rather than the tiles furthest away from the user. This can leave holes in the terrain if tiles are loaded in a spiral fashion.
- **Alpha/Beta Testing** - Unfortunately, this goal in the plan became unachievable both due to timeframes and quarantine procedures following the COVID-19 outbreak. As such, alpha and beta testing was not achieved.
- **More Precise Location Tracking** - The current longitude/latitude coordinate of the user is being estimated by taking a fraction of the next tile's coordinates and adding that to the origin. This is not an accurate procedure to find the user's location. Extension to provide the current altitude of the user will also be a good addition.
- **Interchangeable Controls** - VR and Keyboard controls have both been implemented into the project, but VR detection has not been implemented. As such, the final project only uses Keyboard movement, and creating a check for VR equipment and switching the controls dynamically is an important feature to implement.

Known Bugs

- **Reloading Old Tiles** - Once a tile has been unloaded from the application to make room for more tiles to load, if the tile is revisited it does not reload into the application.
- **Toggle Tooltips** - The model used for the controller is out of date and a more current model was not found, the lines that point to each part of the controller are not accurate.

FAQ

- **VR Headset Blacks Out** - Occasionally, while using the VR equipment, the VR goggles will cut out video feed entirely but leave audio active. We are unsure what causes this issue but we have discovered that the issue can be resolved by powering off the VR headset and waiting for 15 seconds before powering it back on and reconnecting it to the computer.
- **Keyboard Scene Doesn't Work with VR** - When using the application, it is important to ensure that all forms of connecting VR equipment are turned off. If Unity detects any VR

equipment connected to the computer, keyboard controls are disabled and the camera is attached to the VR headset instead of being tied to the mouse for looking around. If you experience this, please turn off all of the VR detection equipment, like the Vive Wireless application and the Steam VR application.

- **Exiting Play Mode May Stall -** When exiting play mode, the Unity application may freeze and stop working altogether. Unfortunately, this is due to our Unity version. The 2019 Unity version is not a long-term support distribution and can cause the aforementioned issue. All you can do is force quit the application via the task manager and restarting the Unity application.

Following on the Test Plan

Testing is an important part of the development process and was taken into account when working on Pangeo. However, sometimes things happen and, living up to its name, the unforeseen delays got the better of the project. Certain aspects of the Test Plan were completed successfully, but some parts were not able to be completed due to the COVID-19 breakout. Please note that certain aspects mentioned next may differ from the Test Plan provided; the provided document is provided equally to show what has been accomplished, and in the case of changed goals, to archive our original approach to completing the project. Following is a short listing of the current Test Plan Status are listed as follows:

- **Github** - Github is a great tool for development, even more than just being a repository for code. Using issue tracking was a practice that we tried to fully commit to but always evaded us a little bit.
- **Unit/Visual Tests** - Due to the nature of the project, unit testing proved difficult for many parts of Pangeo, so visual testing was adopted to help bridge the gap. Effectively, visual testing is a form of unit testing things that are better visually confirmed than in code, like testing user movement with controller input and confirming proper terrain generation.
- **Timeline** - The timeline outline in the Test Plan outlines only the first semester of Pangeo's development. The timeline was followed very closely, however we did not complete Certification 3 by the outlined deadline. For the second semester, the timeline became a fluid timeline that changed on a sprint-to-sprint basis, especially amid the COVID-19 outbreak.
- **Certifications -** Certifications were an early idea of the project that were a broad overview of the goals needed to achieve to complete Pangeo in its entirety. All Certifications were completed in their entirety except for Certification 4. It is also noted that Certifications were not completed in numerical order. As well, Certifications 3 and 5 were rolled into a single goal.
- **Effort Estimation** - At the very beginning, effort estimation was an important aspect of gauging the difficulty of certain tasks to complete for each sprint. However, as our experience with Unity and the project accumulated, effort estimations became both easier to determine, and harder to maintain. Certain aspects of the project became almost impossible to estimate ahead of time while others could be accomplished in the

same day. However, sometimes things escaped us and managed to slip through the cracks, and quickly effort estimation became less important than the actual work.

- **Full Application Testing** - Full application testing was a different beast altogether. Unit testing and visual tests became an integral part of ensuring code never regressed when developing, but full testing was harder to accomplish due to the in-practice timeline of the project and the late term arrival of the COVID-19 outbreak. This meant that exploratory, alpha, and beta testing was never accomplished. There are certain areas where this form of testing would greatly benefit, but as of now, any full application testing has yet to be completed.