Weekly Report 3

Activity Log:

5/08/2019:

- It was decided over the weekend that we will merge sprint 1 and the week 4 report; the prototype we present will be the same for both. We think this smartly reduces workload without sacrificing important documentation.
- Jacob will construct a UML-like diagram (though not strictly following syntax due to the sequential nature of the final product) to demonstrate dependencies between variables in the old Wise script. This is based off the Dependencies spreadsheet (which was used to create an order of priorities) and will go in the week 4 report.

6/08/2019:

- Jacob has completed week 4 report, with Giovanni's use case diagram. Just needs Product section finished when we have an early product finished by Friday.
- Giovanni and Siqi working on modules, Jamie working on User Interface.

7/08/2019:

- Jacob constructed some of the user interface. The functionality for the first prototype is implemented, but not all the Wizard screens.

8/08/2019:

- Group meeting. Jamie can finish off Week 4/Sprint 1 prototype using work that has been done up to this point.

9/08/2019:

- Jamie has finished functionality for Sprint 1/Week 4 prototype.
- Jacob has assigned himself a high priority module and will (finally!) get into scripting proper.

Individual reports

Jacob

Made flow diagram (week 4 report). Completed week 4 report. Worked on user interface - this work used Innosetup Directives in the Files and Run sections instead of Pascal code in the Code section; this was thrown out in favour of what Jamie built in Pascal due to greater flexibility. Will try to ensure overlapping work like this doesn't happen again. Starting work on a script module (Set Command Line Variables).

Jamie

Implemented a logging capability in the installer so that the steps the installer completes can be recorded for troubleshooting purposes. Created a wizard with a welcome page and an installation type page so a user can select a client install or a server install. Added functionality for the installer to load a folder of files or run an executable file depending on the user's choice of installation.

Siqi

The content of 50-206 lines in the old script is studied in detail, and all variables involved in WiseScript are redefined in Inno Setup using Pascal scripting language. In order to facilitate future development work. Continue to familiarize myself with the use of Pascal scripts to prepare for future development.

Sprint Report 1

Achievements

This sprint has produced an installer which is partially complete - the user interface is complete from start to finish, the installer can do a basic install of any files we require, and it also activates an external install .exe file during the install process. We believe this is good progress given that we only met with the client for the first time less than three weeks ago, and we believe we are well on track to having the project complete by our deadline. As this is the first sprint and we have been learning two new scripting languages, we did not have set goals for this sprint as we had absolutely no idea how easy or difficult scripting in InnoSetup would be. However, we believe this sprint has had great value in learning the workings of Inno script, plus some basics of Pascal and Wise.

Problems

The major problem our planning phase has had to overcome has been the uncertainty around how to separate tasks out among group members. To begin with, we had absolutely no idea how this was likely to play out, as the final product in this project is intended to be one large sequential installer script. At this stage, though, we have established issues to be worked on in GitHub by separating out the old Wise script into tasks. While Wise scripting and InnoSetup scripting will inevitably work differently from each other, we believe this is a sensible way to modularise the development. We are not deep enough into Inno scripting yet, however, to say if there will be any issues in regards to differences in *functionality* between Inno and Wise - we are currently working on the assumption that they can perform all the same tasks given the right instructions.

Changes

As this is our first sprint, our project plan has only just been constructed, and as such there are no changes to it. One major change between the project description and our project workflow that has already been made is the use of Git instead of Mercurial. The client requested though did not require that we use Mercurial - after discussion with the client, we discovered that ITL did not have strong feelings on this, so we have committed to using Git due to Giovanni's preference for that platform.

Prototype

This prototype has a basic user interface implemented, with the ability to choose Client or Server install and affect the rest of the installation based on this choice. Selecting a client install will do an install of the input files into C:\. Selecting a server install will execute an external .exe file. These functions are all key functionalities of the final product, and demonstrate progress with our knowledge and utilisation of Inno Scripting.

Appendix

GitHub issue tracking and Kanban board (only Shawn and team members can view):
https://github.com/16430978/installer-capstone/projects/1

Project plan spreadsheets:

- Dependencies:
  https://docs.google.com/spreadsheets/d/1SdETWs8JnK-qYgpxqGSBiMN7VYvaaBcpQ_94nxT9jxw/edit?usp=sharing

- Tasks/issues:
  https://docs.google.com/spreadsheets/d/19XezZA3l-hzyD3eZUbFKjSlOmy8_6nIBnk8FFIgPWNo/edit?usp=sharing