Semester Break Report

Week 1 of break

Jacob worked on and integrated modules which perform the following tasks:

- Check disk space before install
- Startup processing (various tasks before install)
- Run the License creator .exe
- Install files for the Enabler hardware
- Install SDK documentation after install
- Check if a reboot is necessary
- Remove variables stored in the registry as install finishes
- Check if the installer is returning from a restart
- Set Command Line vars if the installer is run from cmd
- Setup environment variables and registry keys before install
- Add information to the Install Log for uninstall purposes
- Update the system configuration after an install (e.g. turn off sleep/hibernate).

Additionally, the following modules have 'completed' boilerplate/skeleton code but could not be integrated at this time due to bugs.

- Make sure a previous version of the EnablerAPI.msi is uninstalled before the new version is installed
- Save configuration settings to registry - regedits are working but an external .exe which checks if the drive is compressed is currently returning false positives.

A SPRINT_2/v0.2 release was made available on 31/08 and master branch was made even with all of our work to date.

I would like to note that this work (+reports) is the result of a 40+ hour week dedicated to the project - I need time to dedicate to Intelligent Machines and Professionalism assignments, so I did what I consider to be a huge amount of work between 24/8 and 31/8 to free up time for these other projects. As such, I do not expect to have much personal progress to report for the next two weeks.

Jamie indicated this week he was struggling greatly with workload from Programming for Computer Graphics. He fixed a bug that prevented correct initialisation of the main installation directory, and fixed another bug that caused an error to be thrown when a user cancelled the installation wizard.

Ricky was travelling this week; the team was given notice that this would be happening a few weeks ago.

Giovanni provided a report which can be found here:
https://documents.onecloudapps.net/files/weeklyreport-august26-30.pdf

Week 2 of break

Jacob fixed a couple of small bugs but, as planned, did not make significant progress. A mistake I made was discovered in that, because of the nature of the project (integrating code into one large main file), when I made a SPRINT 2 version in a personal branch and then *copied* it into master instead of *merging* with master (so that I did not have to resolve merge conflicts in the **heavily** altered main .iss file), I then deleted the dev branch (our buffer branch between master and personal branches) and made a new branch off master (called 'dev' again). Of course, this lost all the prior commit history to dev (although there was never any risk of losing code). Ricky was able to get

this commit history back by merging his local version of dev, but needless to say I have learned a lesson and this won't happen again. I also deleted personal branches (for example I had one called 'mergeModulesJacob'), with the intent to keep the repository tidy, but of course this lost the extensive commit history to that branch, and now our main file leaps from about 500 lines to about 1890 in a single commit to master. I put this down to inexperience with using git (I had thought commit history for deleted branches would be visible somewhere in analytics), and I will not delete branches from now on.

Jamie

Spent all week implementing the .Net installation module on a local branch. Unfortunately, when pushing the completed task to the repo on Friday, I discovered that one of the files involved in the .Net installation exceeded GitHub's 100Mb file limit and that it was introduced to the branch four commits previously. Have managed to remove the file from the commit history a few different ways, but every one so far has required making a compromise somewhere else which isn't ideal. I'm still experimenting with removing the file in a way that has the least impact to the project. Note to self: pay closer attention to file size when committing files to a VCS and push my local commits to the remote repo more often.

Ricky

Completed the development of the Setup Access Permission module and integrated it into the dev branch.

Giovanni

Week 7

Jacob scripted up the Install C++ Redistributables module (https://github.com/16430978/installer-capstone/issues/19), but this has been left unintegrated due to a bug with running a utility .exe.

Unintegrated modules I worked on with known bugs are currently as follows:

- Install API Assemblies - most code is present, but the code that has been left out involves trying to uninstall the API installed by the EnablerAPI.msi file in case it already exists, before installing the current version of the API. Despite being given the correct parameters, neither the uninstall nor the install succeed.
- C++ Redists and Check Disk Compression both have bugs where executing a utility .exe gives a result code which leads to an Abort statement. In the case of the Disk Compression checker, I can actually run the utility individually on my drive and get result code 0 (success), but when the installer runs it (on the same drive), it gets result code 1 (failure, abort install).
- Install SDK docs is **integrated** and does not cause any issues, but an Exec statement has been found to be coded wrong and currently does nothing, this needs fixing.

Ricky

Write script which is in the Install Server Files module(https://github.com/16430978/installer-capstone/issues/27), but not all of that part of module, because there are some of scripts is about server part, I only write some necessary part for our client installation, in the next week, we will start to test the client installation, at first, we need some test case or some test plan, in order to we can have an efficient test work in the ITL 's office.

Jamie

Removed excessively large file from commit history. Committed implementation of .Net 3.5 installation (https://github.com/16430978/installer-capstone/issues/13). Implemented .Net 4 installation (https://github.com/16430978/installer-capstone/issues/14). Started compiling a test checklist in preparation of onsite testing of the installer.

Giovanni

Sprint 2 Report

Achievements

During this sprint we achieved an almost-complete user-interface (only specialized wizard pages for uncommon scenarios are missing), plus integration of ~13 of our modules. Functionalities include:

- Two different startup processing/initialization processes
- Disk Space Checks
- User License creation
- Hardware drivers install
- User Docs install
- Deciding whether a reboot is needed and recognising when returning from a reboot
- Saving and removing variables to/from the registry at different stages of the install
- Accepting and processing command line options (though a silent install is not operational at this stage)
- Logging capabilities and log entries for almost every action the installer performs
- Update system configuration settings like power and sleep/hibernation

Problems

We discovered some statements like 'If System has Windows 95 Shell Interface' or 'Edit 5 registry keys' in the old Wise script were truncated by the compiler, and did not have enough information for us to figure out what task was actually being performed. We contacted the client and received a less-truncated version, plus the original, barely-human-readable .wse file, which we have had to dip into from time to time to figure out these commands.

The old Wise script doesn't indicate a variable's data type. To initialise variables with the correct data type in the new Pascal script, a team member has to search for all the values that variable holds throughout the old script and decide an appropriate data type for the new script. This problem is not difficult to solve, just time consuming.

As a port job where functionalities feed into one another and only a handful of modules can be separated cleanly into Client/Server, it is difficult to set sprint goals. While it is not a perfect measure of progress, I would say that our LOC count (which has gone from ~150 to ~1800 since sprint 1) is a good indication that many, many hours of work have gone into this project in the last three weeks. Our goal for sprint 3 (which coincides with the week 8 report) is to have the client install operational, however...

Changes

...we have discovered that a few more modules are necessary for the client install to be fully operational. Install .NET 3.5, for example, is the module where the RESTART registry key is deleted after it is set to 1. If this registry key is never deleted, then the installer will always start in 'returning from a restart' mode. 'Install C++ Redistributables' may also be changed to high-priority as it does

not specify whether it is only called when a server install is taking place. This is a failing of our initial analysis of the Wise script and has become clear as we've learned more about the old installer's processes.

Prototype

Due to the nature of an installer, the new/added processes the installer performs are not transparent, but running the Enabler4Setup.exe in the SPRINT_2\Output folder and then navigating to Program Files(x86)\Enabler4 should demonstrate that files are now being installed throughout different directory trees. The tester will note the cmd dialog boxes that briefly pop up before and after the install, indicating background processes.

If the tester has Innosetup installed and compiles the Enabler4Setup.iss file and immediately runs the install, the grey/green boxes on the left-hand side of the IDE indicate which statements are being hit and which evaluate to false or are otherwise ignored.