Progress Report

LumenX^3

Version 2.3.2

March 30, 2015

Team Members:

Carmen Tang
Davin Mok
Gary (Guo) Yu
Herman Mak
Him Wai (Michael) Ng

OBELXTECH

1.0 Introduction

The LumenX^3 is a mobile computer that utilizes projection as its method of display as opposed to traditional screens. This method of display allows for a screen that is not limited by the size of the device and promotes collaboration amongst its users by employing hand tracking techniques to give users a whole new interactive experience. The development of the LumenX^3 took place in two major phases, modular development and system integration, the progress of our development will be elaborated further in the document.

2.0 Schedule

In terms of our scheduling, we have currently met all project milestones up to the submission of this document. We are currently in between the final stages of Integration and System Testing and Functioning, where remediation steps undertaken in the Core and Touch Gesture Recognition subsystem have added to the scheduled time, slowing down our Integration by six days. Additional system testing and functioning has revealed further improvements that can be made to the overall product and we are in the process of investigating better driver resolution, detection accuracy and calibration and case bring-up. For the final stages of our project, we are confident in delivering the final proof-of-concept device. Preparations for demo and final course requirements are also being started, including presentation and post-mortem.

The Gantt chart outlining our original schedule is shown below, which we have followed closely.

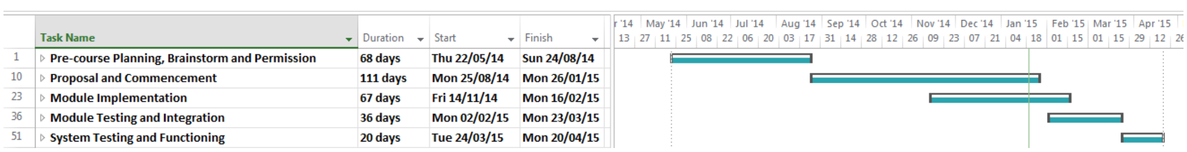

Figure 1 - Simplified Gantt Chart

3.0 Finances

Our first prototype is estimated to cost $748.12 to build. We have gotten $705 from ESSEF and an additional $622.85 from IEEE. We have spent $665.54 on parts thus far, which is approximately 89% of our estimated budget. A comparison of actual costs is shown in Table 1 below:

Table 1 - Project Estimated and Actual Costs

Item	Estimated Cost	Actual Cost	Difference
Microsoft Windows 8.1	$119.99	$0.00	+$119.99
AAXA P3 Pico Projector	$184.91	$184.91	$0.00
MeegoPad T01	$138.03	$138.03	$0.00
Leap Motion Controller	$89.59	$79.68	+$9.91
Arduino Uno	$30.00	$35.00	-$5.00
Minor Electronics and Other Accessories	$55.60	$55.06	+$0.54
Plastic Enclosure	$30.00	$41.43	-$11.43
Contingency (15%)	$100.00	$131.43	-$31.43
Total	**$748.12**	**$665.54**	**+$82.58**

OBELXTECH

We are well within budget and will be using the excess money as emergency funds to replace broken parts as well as to start on our second prototype if time permits. If unforeseen circumstances happen and we go over budget, extra costs not covered by funding will be split evenly amongst the team members.

4.0 Progress

4.1 Core Subsystem Progress

To realize the LumenX^{3}'s two shell design, we initially set out to 3D print both shells. During the fabrication process we observed several production issues that caused us to look into alternatives. After careful examination of our alternative options, we decided to construct the case with laser cut acrylic panels because it offers several important benefits for our application. Firstly, the laser cutter is able to produce our parts extremely quickly, a fraction of the time it takes to 3D print a similar part. Secondly, acrylic offers a similar if not better strength to weight ratio as most 3D printed parts. Finally, the cost of building the case using acrylic is small compared to other methods and that means that we will have more budget for unforeseen problems during fabrication like cracked or scratched panels; we can essentially buy new pieces. The status indication system is complete and going through implementation improvements and testing. The Arduino sketch code is complete and the necessary wiring with LED's is also working.

4.1.1 Case Remediation

If the acrylic panels repeatedly fail to meet our requirements or consistently develop problems during fabrication, we plan to switch to hardboard or thin plywood. Hardboard or thin plywood offers similar structural rigidity to weight characteristics as acrylic but can be mated using conventional glues. Like our current approach, this alternative is also relatively inexpensive allowing us a relatively larger budget to redo parts as necessary.

4.2 Projection Subsystem Progress

The AAXA Pico P3 Projector we purchased is able to project at high definition resolutions very brightly at 50 Lumens. It has been tested numerous times to ensure it is functioning as expected and for long hours, is compatible with the MeegoPad T01, and can clearly project 720p in office-lighted conditions and for long hours. We experimented placing the projector at many different heights and angles in order to find a balance between case size, picture focus and screen size. The best determined configuration has the projector at more than 30 cm height at an angle of 60 degrees below the horizontal. Initially, keystone correction software was researched but none were background programs that could correct at larger than 25 degrees, thus we decided to write our own Windows Display Mini-Port Driver. Setting up the environment and configurations to debug the display driver took about one week longer than expected, as we were very new to writing low-level driver software. We began implementing the perspective correction using functions in the OpenCV open source library, however they could not be compiled into the driver. We then looked into creating our own algorithm, similar to OpenCV by using Homography estimation to calculate a matrix to easily map original screen coordinates to their perspective corrected pixel location. As of today, the driver has been written and installed on the MeegoPad T01 and tested numerous times with the Pico Projector. After just a few days of development the driver worked reliably at a 720p resolution so all words part of the native Operating System were clearly legible. It has since been optimized to only update 'dirty rectangle' subsets of the screen in which pixels have changed, improving speed. It is able to smoothly display pointer movements and window or screen changes, and is currently complete and ready for the demo.

4.3 Touch Gesture Recognition Subsystem Progress

Since the device does not have a physical or capacitive touchscreen, we designed a Touch Gesture Recognition Subsystem to remotely analyze the user's touch gestures performed on the projected screen. Initially, we attempted to create our own recognition system using two computer webcams, a technique known as stereovision. This design however had a notable flaw, in that the camera input was affected by lighting conditions and colour, and thus could not determine accurate finger positions reliably. We then turned to research existing motion sensing hardware, the two best options being the Nimble Sense and the Leap Motion Controller. The Nimble Sense, which was configured to face horizontally which is perfect for our box design and also had better accuracy than the Leap Motion Controller, was unfortunately bought by Facebook and discontinued public sales just before we began this endeavour. Due to this limitation and to time constraints we settled for the Leap Motion Controller, despite it being optimized for use below or behind the user's hands. It came with a mature SDK that can track hands at high precision, which would greatly accelerate our work.

Initially we researched how to write firmware to work with the Windows 8.1 touch drivers, as suggested by Microsoft MSDN documentation. However after thorough investigations it was clear that writing firmware was extremely difficult to learn and would not be implemented well at all for the demo. We decided then to write background software that would inject touch gestures into the Windows OS. The software was divided into two parallel parts: determining whether or not the user has 'touched' the screen, and determining the screen coordinates of the touch. The first part was tested separately and performed with great accuracy. When it was integrated with the second part, problems arose with determining accurate touch locations for the farther parts of the screen since the Leap Motion Controller was not operating in a supported orientation. Experimentations showed inconsistent coordinate positions as the touch distance increased from the Leap Motion Controller. After compensating for this problem, the touch program was able to accurately and consistently determine a touch's coordinate across ¾ of the screen, a major improvement from having one-quarter of the screen accurately recognized. There was also a problem with latency that was solved by configuring the Leap Motion Controller to Low Resource Mode, to alleviate large resource consumption in the Intel Atom processor.

At this moment in time, the touch software is accurate and reliable enough for the demo. It will be recalibrated once it is integrated into the case, which will be a smooth process as we have simplified the recalibration process to easily execute it every time the stand was changed.

5.0 Conclusion

In summary, we have made considerable progress during the term but still have some way to go before the demo. While we may have adjusted our scope, the overall functionality of the LumenX^3 still remains. We are well within budget and although our progress is slightly behind our predicted schedule, we remain optimistic that with our detailed remediation plans, we will have a functional product to show on the day of the demo.