UNIVERSITY of ALAKSA, ANCHORAGE

School of Engineering

Computer Science

NASA Student Launch Competition

Website design and deployment

Autonomous Ground Support Equipment Programming

by

Mr. Jacob Dempsey

Supervisor:

Prof. Kenrick Mock, PhD

A Capstone Project Submitted to the Computer Science Department

Anchorage AK, April 2015.

jadempsey@alaska.edu

Version 1.0

Abstract

This capstone was a portion of a larger interdisciplinary project. The parent project was to create a rocket and all necessary support equipment, designed to specifications provided by NASA for entry in the 2015 NASA Student Launch Competition.

The 2015 UAA Rocketry Team for the NASA Student Launch Competition was formed from students in various disciplines. The team consists of Mechanical Engineering Students, a Computer Science Student, and a Business Student. The competition required the unique skills which we each brought to the project.

The competition required the team to design and build a rocket which was to be launched to a target altitude of 3000 feet above the launch elevation. A point was deducted for each foot the rockets apogee was above or below the target elevation. The rocket was required to carry a payload provided by NASA. The payload was to be loaded and secured in the rocket by autonomous ground support equipment (AGSE). The AGSE collected the payload from ground and secured it in the rocket. After apogee, the rocket jettisoned the payload, and the payload and rocket were required to return to the ground with no damage to either. The rocket was required be able to be re-launched without the need for any repairs beyond repacking the rocket shoots and inserting a new engine.

My portion of the project was the design and deployment of the team's website as well as the programming of the AGSE. The website is be hosted by a shared hosting service running Linux. Apache was used for the web server, and the website was built upon WordPress to make updating the site easier for the entire team. The AGSE was controlled by an Arduino Uno, single board computer, which was controlling two stepper motors and two servos in the AGSE and three servos in the rocket.

Acknowledgments

I thank my Great Uncle Earnest Plagge, my Uncle Adrian Orange, and the stories about my Grandpa Harry Orange for inspiring me to learn about electronics.

I thank my parents, Gordon and Karen Dempsey: for allowing me to tinker with electronics, I learned a tremendous amount even from the devices I couldn't put back together; for purchasing a Commodor Vic 20 personal computer for the family in 1980, I found the BASIC code the games were written in to be fascinating (It definitely helped prepare me for today); and most of all for raising me to be the man I am today.

I thank Mr. Bob Cochran for mentoring me in high school; for having me in his principles of technology class for 4 years, and for assisting me with independent study in the class for 2 of those years.

I thank my wife Yvette Dempsey for encouraging me to go to college and earn a degree, and supporting me, especially when I've been spread too thin trying to accomplish too much.

I would like to thank Professor Jim Jacobs for persuading me to declare as a Computer Science and Mathematics major instead of transferring to another school to pursue a degree in Electronic Engineering, and for providing me with a strong base in the fundamentals of programming.

I thank Curvin Metzler for his support in my early years as a college student, and his continued friendship and support almost 2 decades after the first course I took from him.

I thank Kenrick Mock for being an inspiration, for the excellent preparation he always put into his lectures, and for bringing an artificial intelligence course to UAA. I had the privilege of taking the course the first time he taught it here at UAA, and I still remember the course very fondly. I also thank him for being my advisor after Jim Jacobs retired, for continuing to be my advisor when I returned to school after taking a decade off, and for being my advisor for this applied software development project.

While I have listed the three professors who have had the most profound impact on my Computer Science degree. I have to thank all of the professors I have had, particularly those in the Computer Science and Mathematics departments. I have learned valuable lessons from each of them, and could not have completed my degree without them.

I thank the members of the UAA Rocketry Team for coming together, and allowing me to participate in this project. While I was busier than I would have like to have been, and would have like to have put more time into the project, it was a wonderful experience which I am grateful to have been a part of. In particular, I thank Mr. David Erikson, without his National Association of Rocketry (NAR) certification and mentorship, we would not have been able to compete in the competition.

Lastly, I thank the LORD for allowing me to live in such a wonderful time when so much is changing in technology, and for giving me the gifts and talents to pursue the degree and career I have been pursing.

Contents

Table of Figures

Chapter 1 Introduction

License

1.1 Introduction

The NASA Student Launch Mini Mars Ascent Vehicle project[1] is part of an annual competition, the NASA Student Launch. The project is a research based, competitive, exploration project providing relevant research and development experience to the participants. In this year's competition, we built a high powered rocket which we launched to a target altitude of 3,000 feet above ground level. Before launch, our Autonomous Ground Support Equipment (AGSE), loaded and secured a payload, provided by NASA, into the rocket. Specific competition requirements included:

- Carry a payload approximately ¾ inches in diameter and 4 ¾ inches long with a weight of approximately 4 ounces.
- Deliver the payload to, an apogee of 3,000 feet above ground level
- The rocket had to carry at least one commercially available, barometric altimeter for recording the official altitude.
- The rocket had to be recoverable and reusable. (No repairs needed after the launch)
- The vehicle had to contain an electronic tracking device which transmitted its location to a ground receiver.
- The payload had to be loaded autonomously. There had to be one power switch, one pause button, and one indicator light. During this process, the team was not permitted to interact with the rocket or AGSE.
- The AGSE could not use: Sensors which relied on the earth's magnetic fields, ultrasonic or sound based sensors, radio aids, open circuit pneumatics, or air breathing systems.

This year a group of students from varying backgrounds formed a team, UAA Rocketry, to compete in the competition. We worked together, across disciplines, to enter the 2015 NASA Student Launch competition. The team consisted of five mechanical engineering students, one business student, and one computer science student with guidance from a National Association of Rocketry (NAR) representative.

Team Members:

- Stephen Arwine – Mechanical Engineering
- Jordan Shindle – Mechanical Engineering
- Brandon Grimshaw – Mechanical Engineering
- John Berg – Mechanical Engineering
- Nik Butler – Mechanical Engineering
- Jacob Dempsey – Computer Science
- Carolyn Forner – Business Management

Guidance Provided by:

- David Erickson – NAR representative

The mechanical engineering students designed the rocket, launch platform, and the Autonomous Ground Support Equipment (AGSE). The business major is helped to organize efforts, setup appointments such as presentations to the public, and secure donations of equipment and funds. As the computer science student, I designed and built the team website, assisted with the selection of electronic components used in the rocket and AGSE, and programmed the AGSE to load the payload with Stephen Arwine.

We used off the shelf components where appropriate, and designed custom components when they provided the best utility for the project.

My portion of the project was be broken down into three primary sections:

1. **Website**
   The website was our primary method of providing information to NASA throughout the project. It was also be a promotional tool used to promote the UAA Rocketry Team specifically, and STEM generally to the community. The website address we used was http://www.UAARocketry.com. We used WordPress[2] as our content management system, and I was created a custom theme for WordPress using _s (under scores) as the base of the theme, PHP, JavaScript, and CSS as well as creating custom graphics for the site and the WordPress theme.

2. **Rocket, Launch Pad, and AGSE electronics selection, assembly, and testing.**
   The electronics included: the rocket's altimeters; the rocket's GPS location tracking hardware; and the controls, motors, and servos for the AGSE.

3. **Programming the AGSE.**
   The AGSE was ultimately programmed on an Arduino Uno, but started out being programmed on a Beaglebone Black.

1.2 Application

The project has direct applications in aerospace, particularly as it applies to NASA. It also has applications in robotics with the AGSE. Finally, it has applications in mass communication via the internet with the development and publishing of the team's website.

I gained experience providing web development services as I explored the options available to publish the team's content to the web. I gained experience and insight into the robotics design as I helped select the electronic components and did programming for the AGSE.

1.3 Motivation

My motivation for this project is multi-faceted. First, I wanted to do an interdisciplinary team project, as I believe this provides a more realistic real world experience. Second, I wanted to participate in a project which would be out of the ordinary. Third, I have always found robotic systems to be very interesting. This project fulfilled all of these goals. The project as a whole would have been difficult for any one person to provide all of the necessary skills lending itself very well to an interdisciplinary project. I am not aware of anyone from UAA participating in a project like this, which for me, makes it out of the ordinary. While my involvement in the project covers a wide scope, I was most excited about helping to design and program the AGSE.

1.4 Recent Developments

In recent years, the size and power requirements of computers has been decreasing. These developments are allowed us to house sophisticated electronic devices in our rocket, as well as in the creation of our AGSE. We exploited these developments to create a cutting edge rocket and AGSE for the NASA competition.

1.5 Initial Design

1.5.1 Website

The website is being hosted by Jacob's CHAOS on my server space. We chose to use my server space as there is no associated cost. That is, I am not charging the team for the use of the space, and I am already very familiar with the environment.

We are using WordPress as the base of the content management system (CMS). After comparing features and user experiences, we chose to use WordPress as our CMS. While this was be the first project which I developed for WordPress, my previous experience with developing static and database driven websites as well as templates for other CMS systems served me well in developing a theme for it.

I used _s (under scores[3]) as a base for developing the word press theme. It is designed as a starting point for new WordPress themes. It has very little formatting initially, but has a well-defined structure for building the theme. See **figure 1** for a layout of the technology for the website.

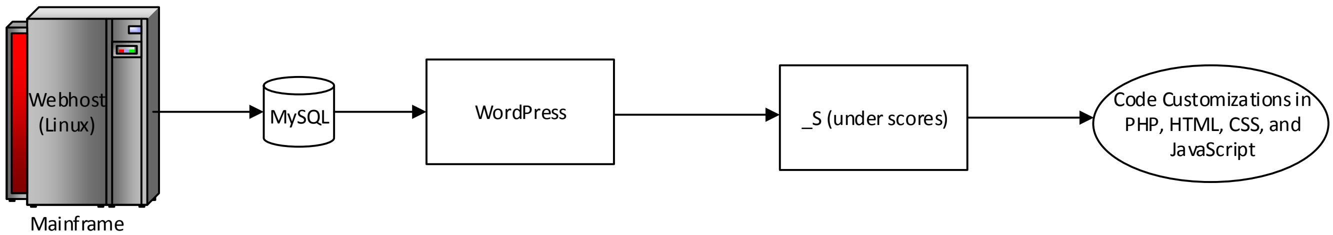

Figure 1: Website technology layout

I coded in html, PHP, JavaScript, and CSS to create the theme.

Team requirement for the site:

- Professional looking site (I used black text on a white background and clean lines to keep the website professional looking)
- Easy to update (Functionality provided by WordPress)
- Unique with a Space/Alaska design which will let people know at a glance they are on the UAA Rocketry website (I created a star field for the background with a parallax scrolling effect when scrolling up or down the page. I also created a transparent aurora borealis (northern lights) image in the site header to build a stronger visual tie to the north and Alaska.

The content of the pages and their relation to one another was decided by the team.

1.5.2 Electronics

We used an off the shelf altimeter per NASA's requirement.

We used an off the shelf GPS location system.

We used an off the shelf ignition system.

We used an Arduino Uno in the final design, but started with a Beaglebone Black[4] for the AGSE controller. I initially chose to use the Beaglebone Black for the added memory and computational power. However, we stitched to using the Arduino Uno to take advantage of the available libraries for the stepper motors and servos.

We used off the shelf stepper motors and motor controllers for manipulating the arm and end effector.

1.5.3 AGSE Design

The AGSE consisted of four main components: the mounting base, the housing, an arm, and an end effector. The mounting base provided the stability for the AGSE, the housing enclosed the electronic controls, while the arm and end effector provided the utility for picking up the payload.

Figure 2 is our preliminary design for our Rocket and AGSE.

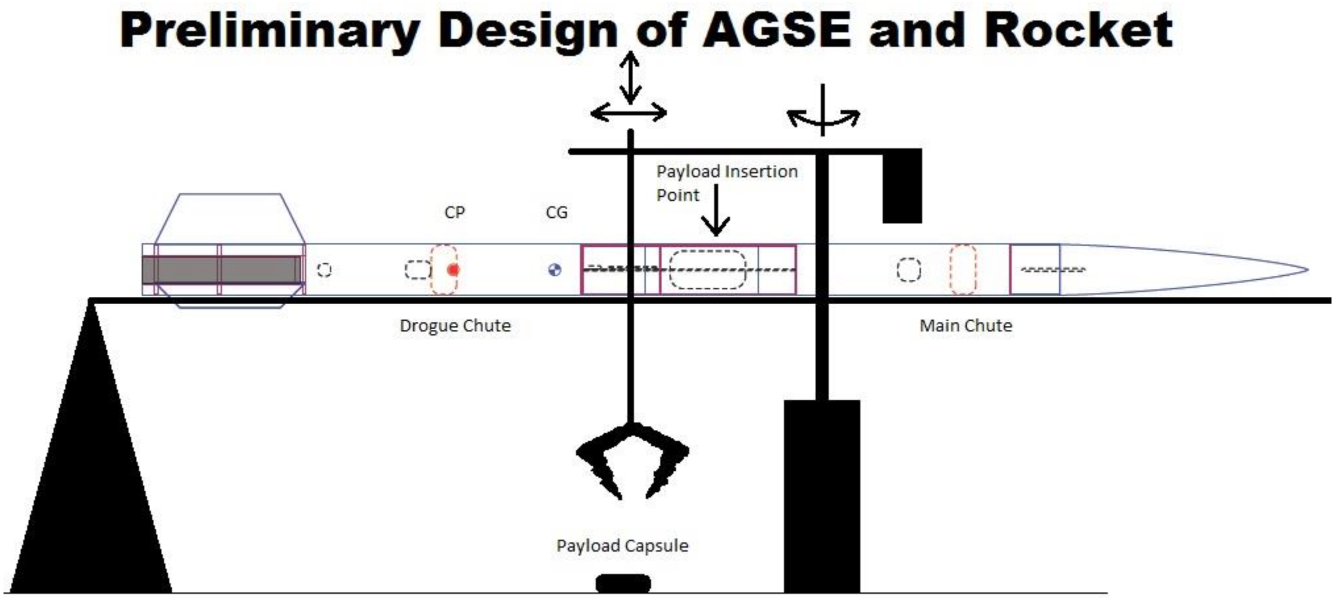

Figure 2: AGSE

We continued to refine our design for both the rocket and the AGSE. The mechanical engineering students worked in SolidWorks to create and render the design. **Figure 3** provides a look at a potential AGSE with the rocket, while **Figure 4** provides an alternative design to the AGSE.

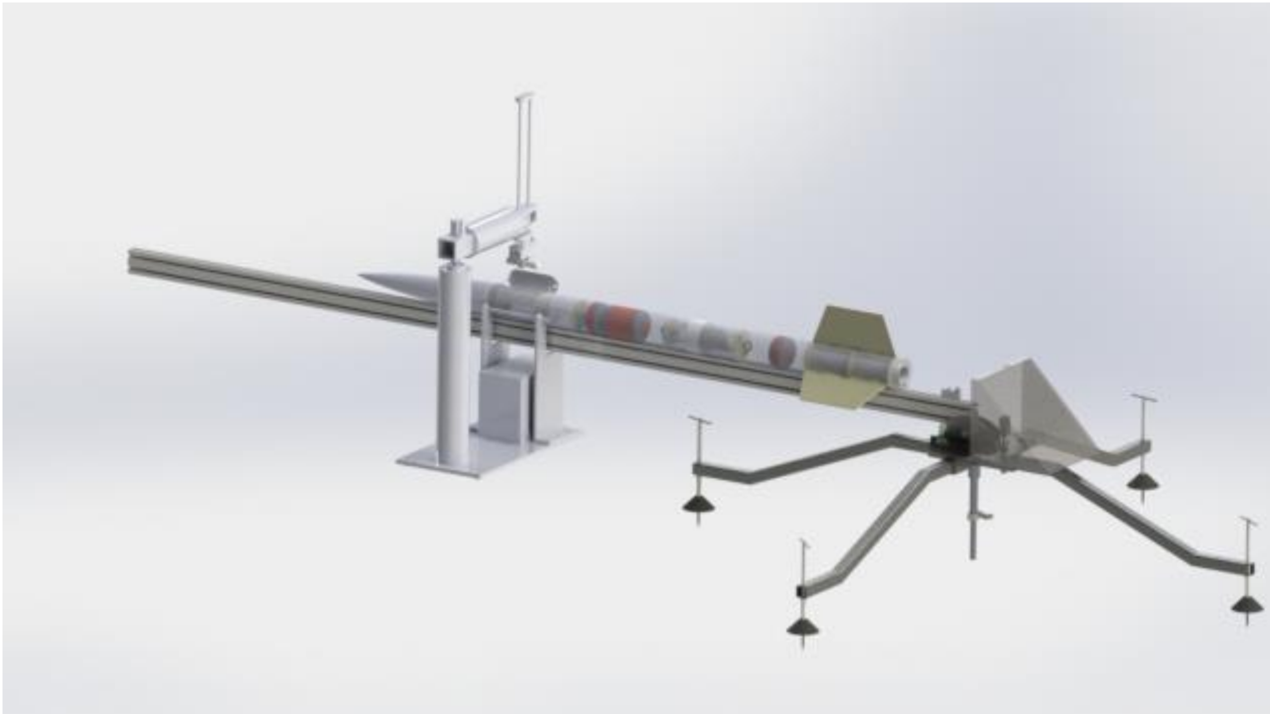

Figure 3: AGSE and Rocket SolidWorks rendering

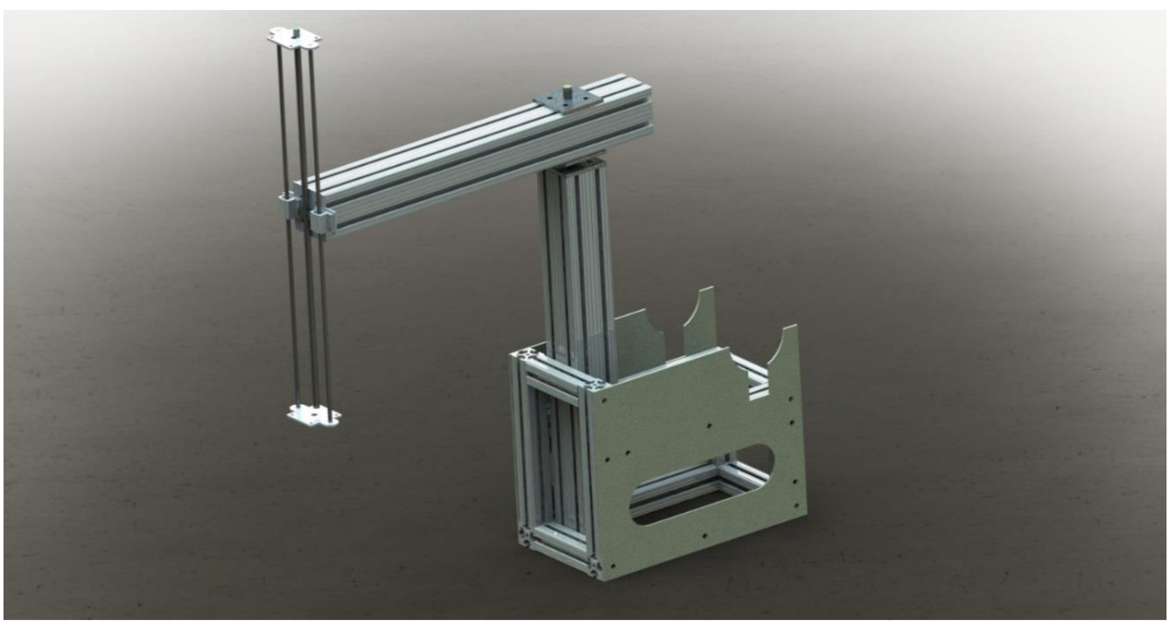

Figure 4: Alternate AGSE SolidWorks rendering

We continued to refine our design until we were ready to build our full scale system.

1.5.4 AGSE Programming

I initially programmed the AGSE in C++ using eclipse5 as my editor and cross compiling using GCC. The final AGSE was programmed by Stephen Arwine and myself in the Arduino language.

Figure 5 Basic UML diagram for the arm controller logic.

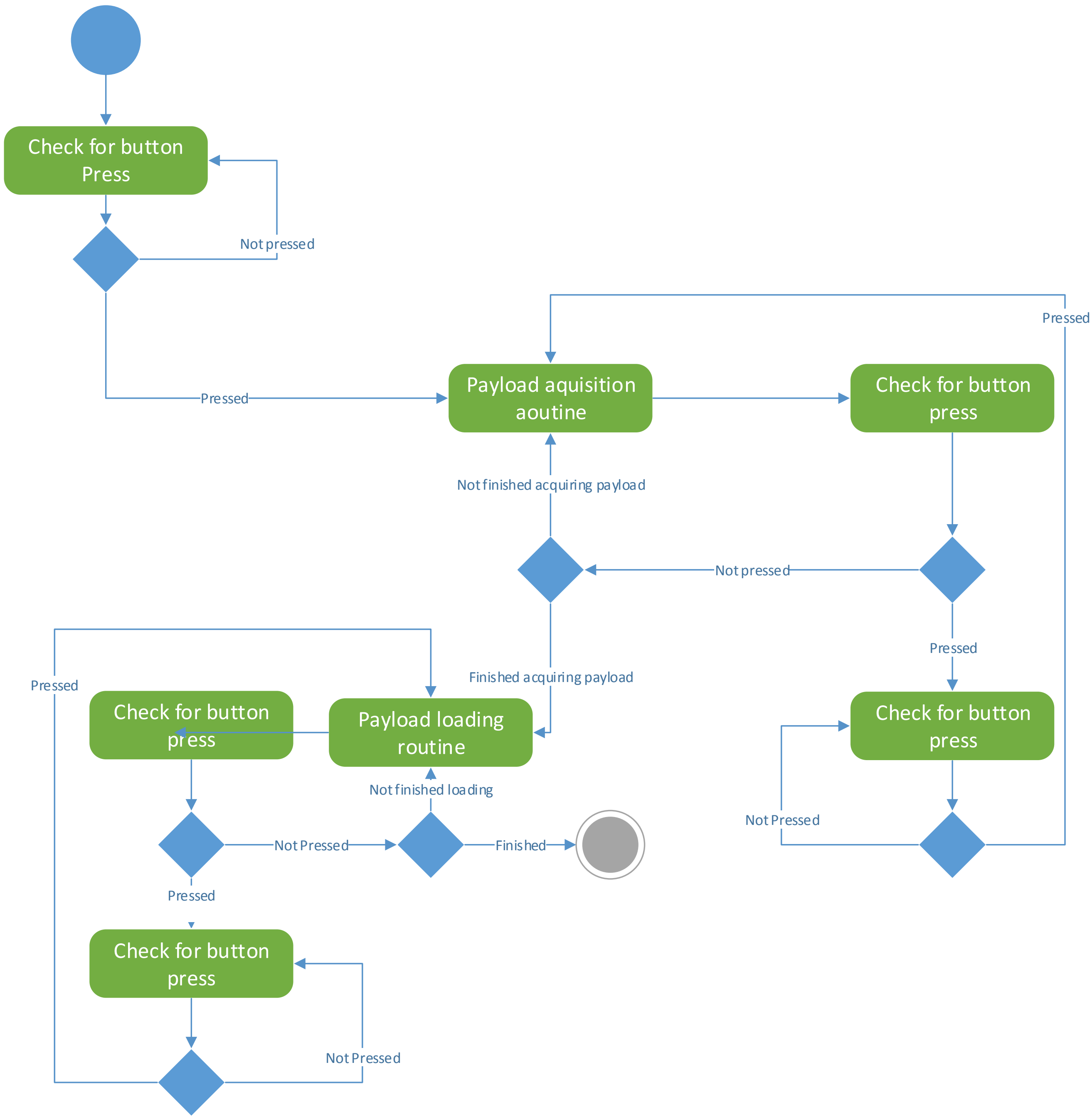

Figure 5: UML Diagram for AGSE

Chapter 2 System Integration and Methodology

2.1 Introduction

The NASA Student Launch competition provides real world experience for teams of students working together to accomplish a common goal. This year's goal was to autonomously load a payload into a rocket of our design and construction, launch the rocket with its payload to a predetermined altitude, and return the rocket and payload safely to the planet's surface.

The portion of the project I was responsible for was the design and construction of the website and the programming of the Automated Ground Support Equipment (AGSE).

2.2 Technology

The technology involved in the project has been around for years, yet it is constantly evolving. This creates many new opportunities for innovation as well as a stable foundation to build upon.

2.2.1 Website Technologies

The technologies the website used are HTTP (hypertext transport protocol), HTML (Hypertext Markup Language) PHP (PHP: Hypertext Preprocessor [1]), CSS (Cascading Style Sheets), JavaScript, and related technologies.

HTTP was the communications protocol which the webserver and web clients (web browsers) used to communicate to display the webpages and content of the team's website. PHP was used to parse and prepare the HTML for display in users web browsers. HTML was used to structure the content within the individual pages. JavaScript was used for client side scripting. Most of the client side scripting was for visual effects. Other technologies relevant to web development and display may be incorporated as needed.

2.2.2 AGSE Technologies

The technologies the AGSE used are device trees, device tree overlays [2][3], C/C++, PCBs (Printed Circuit Boards), ICs (Integrated Circuits), solid state electronics, Electric Motors, and related technologies.

The Beaglebone black uses a generic device tree to define its components. The device tree was used to control the GPIO ports on the Beaglebone Black for controlling components such as the stepper motor controllers, and how the system interacted with them. C/C++ was used to create the logic to control the motors; as well as to instruct the system to send signals to the stepper motor controllers. I used commercially available PCBs, ICs, solid state electronics and electric motors to construct the AGSE's control system.

2.3 Components

2.3.1 Website Components

The website is being run on a shared hosting service. The server is running a variant of the GNU/Linux operating system. I used the Apache webserver with the PHP module to serve the webpages. I used the MySQL database as well as the servers file system to store the websites content, and I used WordPress as the font end to manage and present the content. I used _s (under scores) as the base for my custom template to customize the look and feel of the website.

The code for the website was edited using Notepad++ [4], a free text editor which offers syntax highlighting.

2.3.2 AGSE Components

The AGSE originally used a Beaglebone Black as the controller, but was finished using an Arduino Uno. There was 3 stepper motor controllers driving the units 3 stepper motors. The stepper motor controllers was Big Easy Driver model ROB-11876 [5]. It will have one indicator LED, to indicate the units current state (e.g. paused or running). It will also have a power switch as well as a push button to pause and un-pause the units operation.

Figure 6: Beaglebone Black

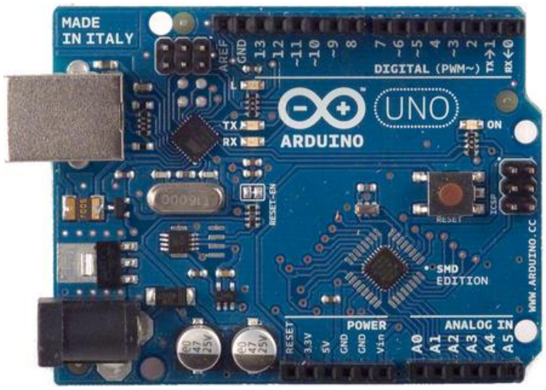

Figure 7: Arduino Uno

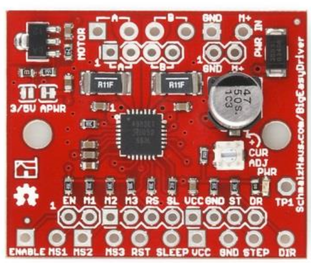

Figure 8: Big Easy Stepper Controller ROB-11876

The code for the AGSE was edited in Eclipse, and cross compiled use GCC before the binaries were placed in the Beaglebone Black via a USB connection. Once we switch to the Arduino Uno, the code was written in the Arduino IDE and transferred directly to the Arduino Uno via USB.

2.4 Initial Design

2.4.1 Website – Initial Design

The website consisted of 5 main section:

1. Background
2. Header
3. Menu
4. Content Area
5. Footer

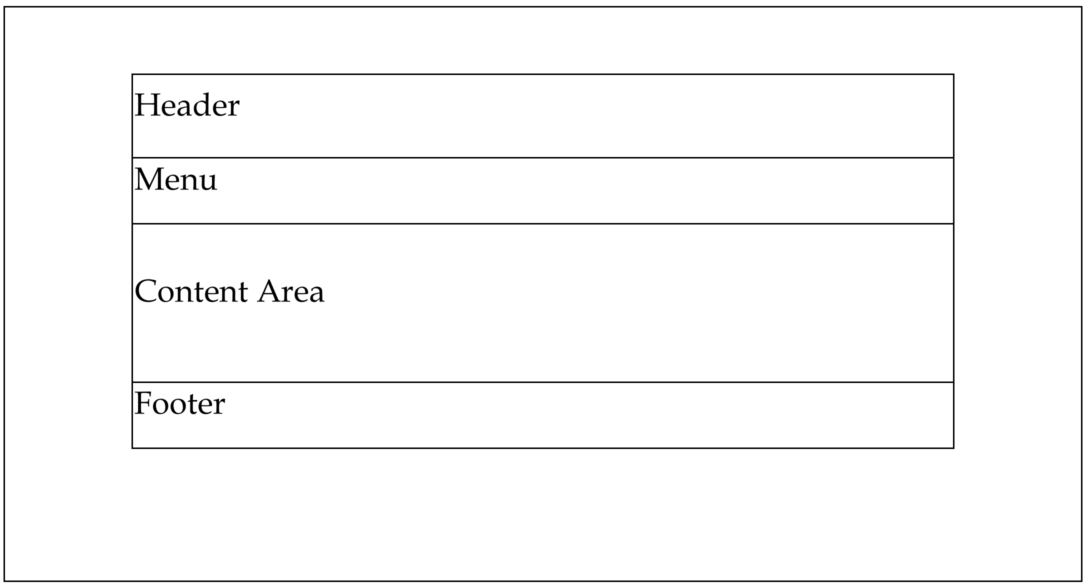

Figure 9: Website page layout

The background was consistent across all pages.

The Header provided basic information about the rocketry team, and was consistent across all pages.

The Menu provided the main navigational links of the site, and was consistent across all pages.

The Content Area will contain the information relevant to a specific page.

The footer will contain information about our sponsors, and was consistent across all pages.

The web design for the website was separated into a number of stages or stories.

- DNS Configuration
- WordPress Installation

- Template – Create the basic structure of the pages
- Template – Customize the look and feel of the pages
- Template – Add special features (parallax scrolling star field)
- User training: WordPress content management.

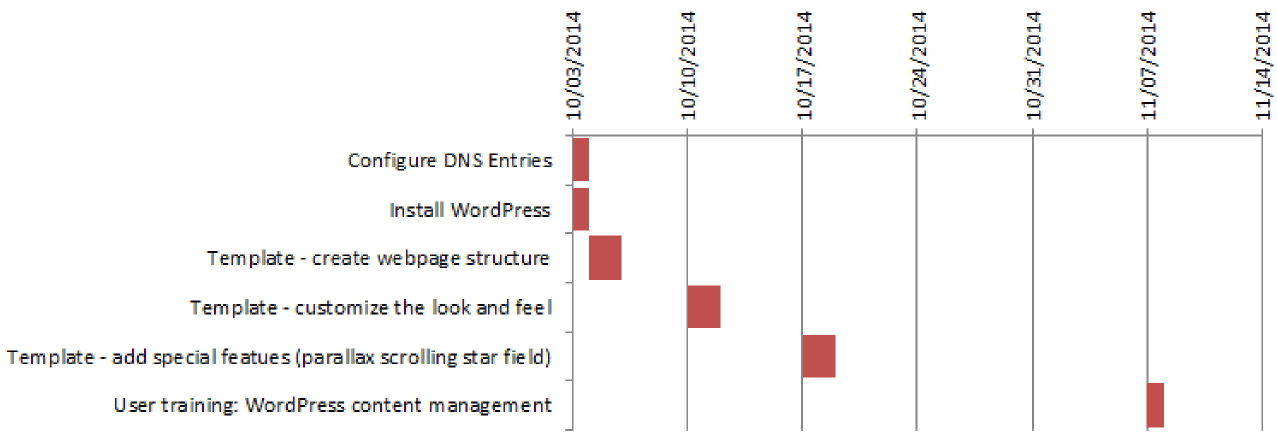

Figure 10: Website Gantt chart

2.4.2 AGSE – Initial Code Design

The code design for the AGSE was separated into a number of stages or stories, and each stage was researched and developed independently.

- Device Tree and Device Tree Overlay
- Reading from and writing to the GPIO
- Controlling an external LED
- Communicating with the stepper controllers
- Creating the Controller Classes for the stepper motors.
- Writing the logic to use the AGSE to acquire and deposit a payload into the rocket.

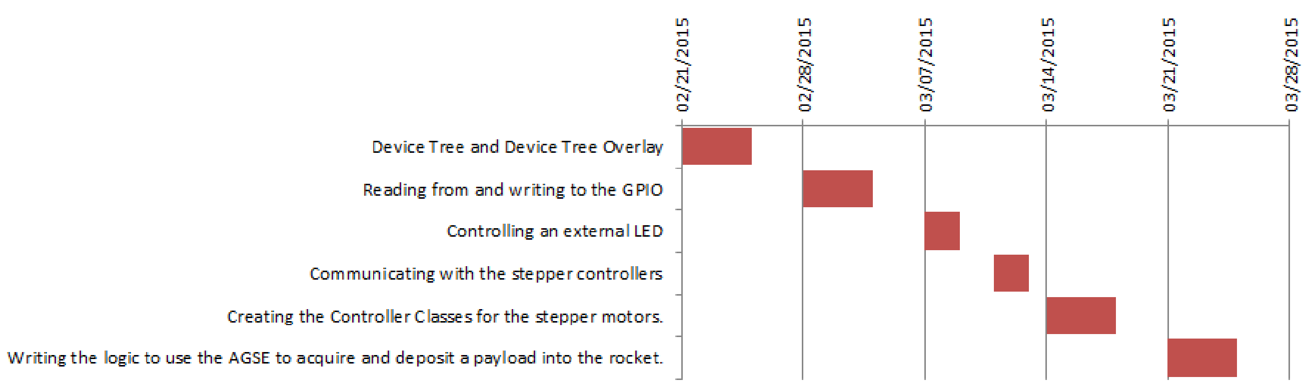

Figure 11: AGSE Software Development Gantt chart

2.5 Agile Code Development

The code for the project is being subdivided into smaller manageable sections of which measurable deliverables can be created. During development, there was validation tests created for each section, and unit testing was performed throughout the process. As portions are completed, they was presented to the UAA Rocketry Team for user acceptance testing. If a section does not pass the user acceptance testing, it was re-worked, re-tested, and then acceptance tested again. I used this process to control the quality of the code being written for the project.

Chapter 3 User Interface Design and Testing

3.1 Introduction

The project has two user interfaces. The user interface of the website and the user interface of the autonomous ground support equipment (AGSE). The user interface for the website is provided largely through the use of WordPress and the _s (under scores) theme template. However, significant customization was required to create the user interface for the website. The user interface for the AGSE on the other had was entirely custom, having some basic requirements set forth by NASA.

3.2 User Interface

3.2.1 Website User Interface

The website user interface uses a page background, a header section, a horizontal menu at the top of the content area of the site, and hyperlinks within text and image elements of the site. The primary content of the site is contained in a central content section. The footer is used to display sponsors as displayed in *figure 4*.

The page background consists of 3 layered images *Figures 1, 2, and 3* of stars created in GIMP (GNU Image Manipulation Program) some tricks with HTML (Hyper Text Markup Language), CSS (Cascading Style Sheets), and JavaScript were used to create a parallax scrolling effect when the user scrolls the page up or down, giving the page a greater sense of depth, and emphasizing the element of space in the NASA Student Launch Competition.

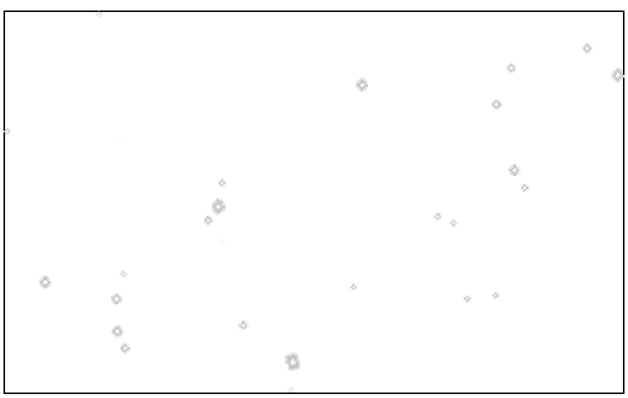

Figure 12: Star field top layer

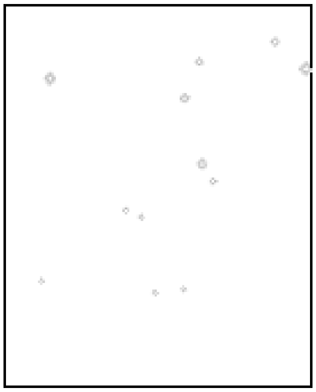

Figure 13: Star field middle layer 1

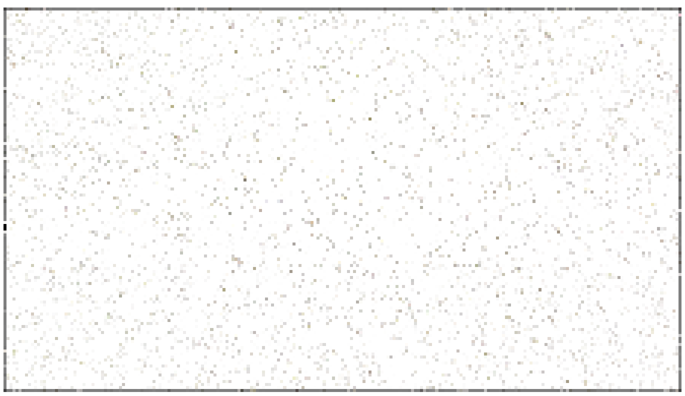

Figure 14: Star field middle layer 2

Figure 15: Star field background

The code for the effect is relatively simple. First, the layers are created as div elements; two wrapper elements which contain the actual layers being animated, and the 3 layers being animated for the parallax effect. The div declarations are followed by a small piece of in-line JavaScript which is tied to the window's scroll event. When the window scrolls, the position of each of the layers adjusted by a percentage which was declared when the div was set up.

The CSS for the layers is stored in a separate file, and ties the star field images to the layers as well as setting up their height and width values.

```
<!-- parallax fields -->

<div class="parallax-container">

	<div class="parallax-wrapper">

		<div class="parallax-field" data-scroll-percent="40"> </div>

		<div class="parallax-field" data-scroll-percent="45"> </div>

		<div class="parallax-field" data-scroll-percent="50"> </div>

	</div>

</div>

<script
src='http://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js'></script>

<script>

	$(function()

	{

		var layers = $('.parallax-field'), $window = $(window);

		$window.scroll(function()

		{

			var scrollTop = $window.scrollTop();

			layers.each(function()

			{

				var $this = $(this);

				var scrollspeed = parseInt($this.data('scroll-percent'));

				var scrolled = - scrollTop * (scrollspeed/100);

				$this.css('transform', 'translateY(' + scrolled + 'px)');

				var hght = $window.height();

				hght = hght - scrolled;
```

```
                    $this.css('height', hght + 'px');
                });
            });
        })
</script>
<!-- end of parallax fields -->
/*------------------------------------------------------------
14.0 Parallax Effect
------------------------------------------------------------*/

.parallax-container
{
        position: absolute;
        top: 0px;
        height: 3000px;
        width: 100%;
        z-index: -1;}

.parallax-wrapper {
        position: fixed;
        width: 100%;
        z-index: -1;}

.parallax-field {
        position: fixed;
        width: 100%;
        height: 3000px;
```

```
	position: absolute;
	z-index: -1;}

.parallax-field:nth-of-type(1)
{
	background: transparent url("/wp-content/themes/rocketry/images/star field 1.png");
}

.parallax-field:nth-of-type(2)
{
	background: transparent url("/wp-content/themes/rocketry/images/star field 2.png");
}

.parallax-field:nth-of-type(3)
{
	background: transparent url("/wp-content/themes/rocketry/images/star field 3.png");
}
```

Figure 16: UAA Rocketry website - user interface

The header section consists of the UAA rocketry logo and a declaration of who we are (UAA Rocketry) and what the site is about (Senior Design and NASA SL competition), placed over the top of a semitransparent image representing the aurora borealis, with a horizontal menu bar at the bottom. The menu bar has a green background with gold lettering to re-enforce the team's connection to UAA. The menu itself is an unordered list. Stylesheet properties are used to cause the list to render as an evenly spaced set of horizontal menu items with a consistent background bar and consistent text colors. The main pages of the website are then added to the menu bar through the WordPress administration panel.

The content area consists of a white background with black text for readability. The title of each page is displayed in the upper left corner of the content area with the content of the page being placed below. Depending on the purpose of the age, it may have text, images, or other embedded media. Some of the text and image in the content may be hyper linked to additional content both in and outside or the team's website.

The footer remains at the bottom of the page at all times, and contains a view of the earth form space without team sponsors over it.

3.2.2 AGSE User Interface

The AGSE user interface consisted of an on/off switch, a status LED, and button to pause and un-pause the system while it is running.

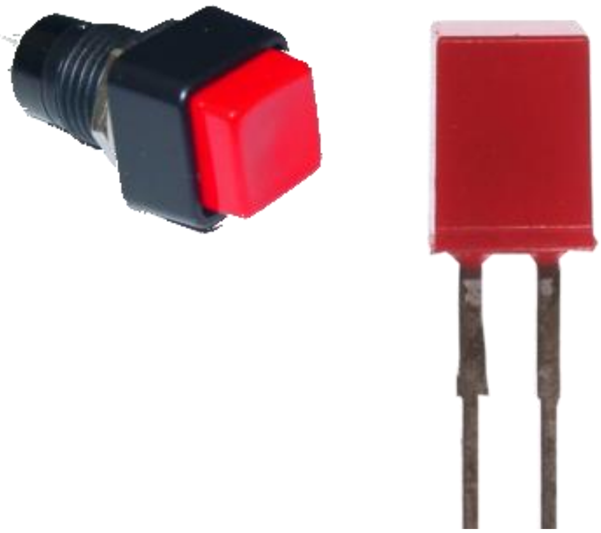

Figure 17: Interface Components

3.3 Testing

3.3.1 Website Testing

The website was tested in stages. First the functionality of WordPress was tested using the default theme. Once it was determined the necessary basic functionality was there, a stripped down template was installed. The basic functionality was tested again. The template was then customized to begin providing the look and feel the Team desired. As the elements were added to the site, the Team would test the changes and provide feedback on the look and feel of the features. Hence, most of the testing on the website was functional user testing as I was developing based on a known platform. In addition to the users testing the site, I tested to make sure it was behaving consistently on chrome, internet explorer, safari, and firefox as well as on android and iOS devices. More programmatic testing was required on the short program to create the parallax effect. It was developed and tested independent of the main site. The function was tested to verify the fields where moving in the correct direction at the correct speeds to give the desired appearance before it was integrated as the background of the site. Once added into the site, the pages were scrolled repeatedly to make sure the function was working consistently over all of the pages, and to see if any memory leaks or other anomalies could be detected. Being unable to detect any abnormal behavior, the code as approved as complete.

The tools used for the development and testing were various web browsers for testing the user interface, and notepad++ for modifying the code.

3.3.2 AGSE Testing

The code for the AGSE was tested in more granular phases. The first part of the code to be tested was used to control the output of the GPIO pins of the Beaglebone Black. The next part of the code to be tested was to read high and low values using the GPIO pins of the Beaglebone Black. The stepper motors, stepper controller and servos were tested using variable voltage DC power supplies to verify their functionality to make sure the correct voltages would be supplied from the Beaglebone Black and external power sources. Less rigorous testing was performed on the code for the Arduino Uno as we were lacking time and using pre written libraries.

In the development and testing on the the Beaglebone Black, the Beaglebone Black System Reference Manual[1] and Beaglebone Black Schematics[2], were heavily used, as well as the Beaglebone Black P8[3] and P9[4] Header Tables, and and article by Derek Molloy "Beaglebone GPIO Programming on Arm Embedded Linux"[5] which were very instrumental in helping me understand how to program the Beaglebone GPIO's.

3.3.3 Agile Methodology

Agile project management looks at the overall length of the project in terms of segments of time. As more of the project is developed the time segments are converted into more well defined segments of time. Stories are created to represent code deliverables which can be produced in short periods of time, typically in one to two weeks. Agile programming is concerned with producing those deliverables and with unit testing as well as user testing and acceptance of the deliverables. Together, the Agile project management and Agile programming create a flexible environment which lends itself well to projects created on time, on budget, and within scope.

Chapter 4 Results and Discussion

4.1 Introduction

The UAA Rocketry Team designed and built a rocket and autonomous ground support equipment (AGSE) which we entered in the 2015 NASA Student Launch Competition. The rocket was designed to carry a payload to 3000 feet, and to jettison the payload at 1000 feet. The AGSE was designed to load and secure the payload in the rocket prior to launch.

4.2 Results

4.2.1 Website

The website received positive feedback from the UAA Rocketry Team, NASA, as well as from the public. The website served two purposes:

1) Presenting documentation to NASA for the competition.
2) Public outreach and the promotion of Science, Technology, Engineering, and Mathematics (STEM)

The use of WordPress as the backend of our content management system proved to be an effective choice. Team members were able to update and modify content without the need to create additional documentation, and with minimal training.

After a user has logged into the site by going to http://www.UAARocketry.com, they are presented with a menu of options.

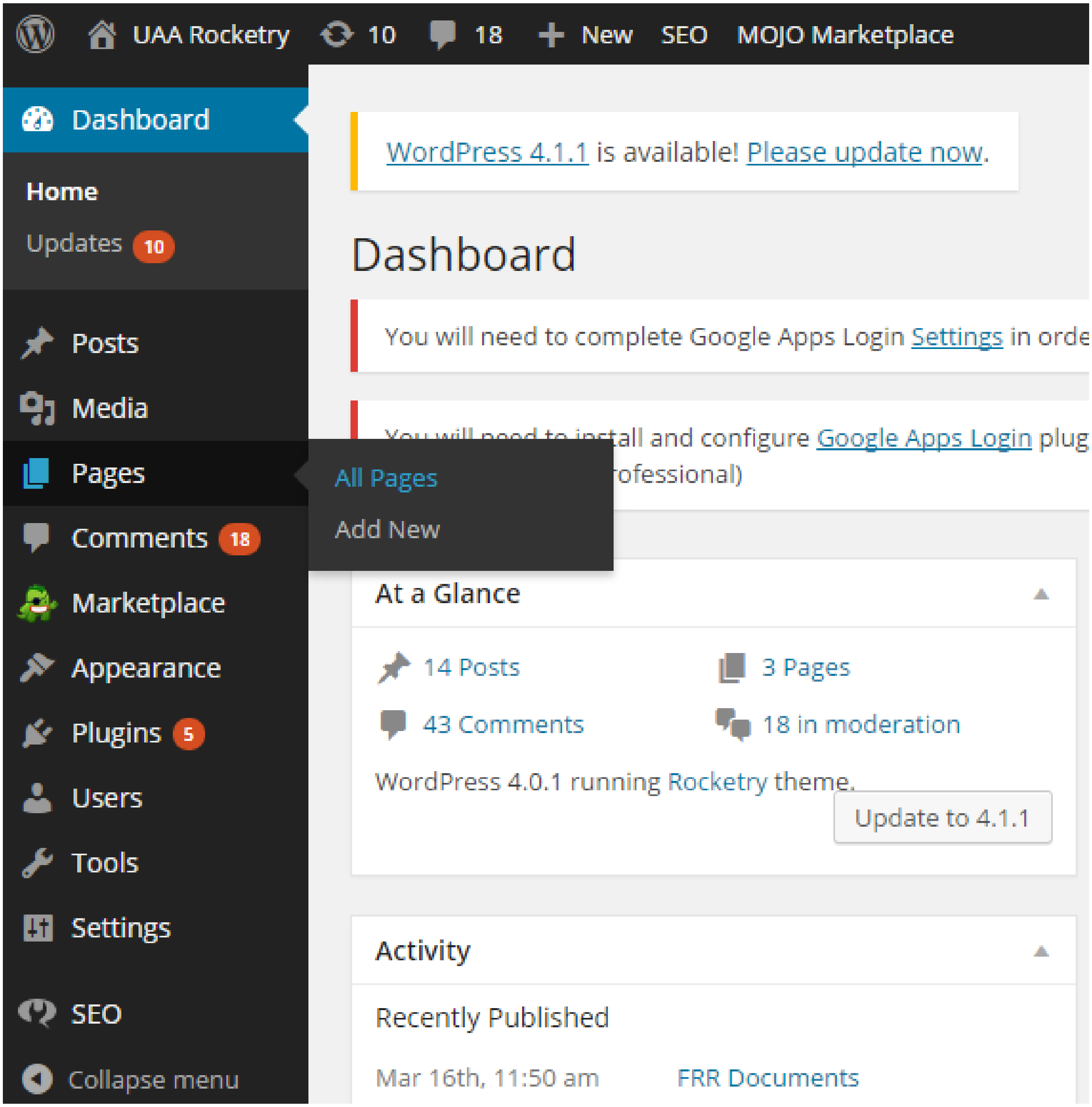

Figure 18: WordPress admin interface

Most of the users are concerned with updating pages, and will navigate to All Pages to select the page they wish to update.

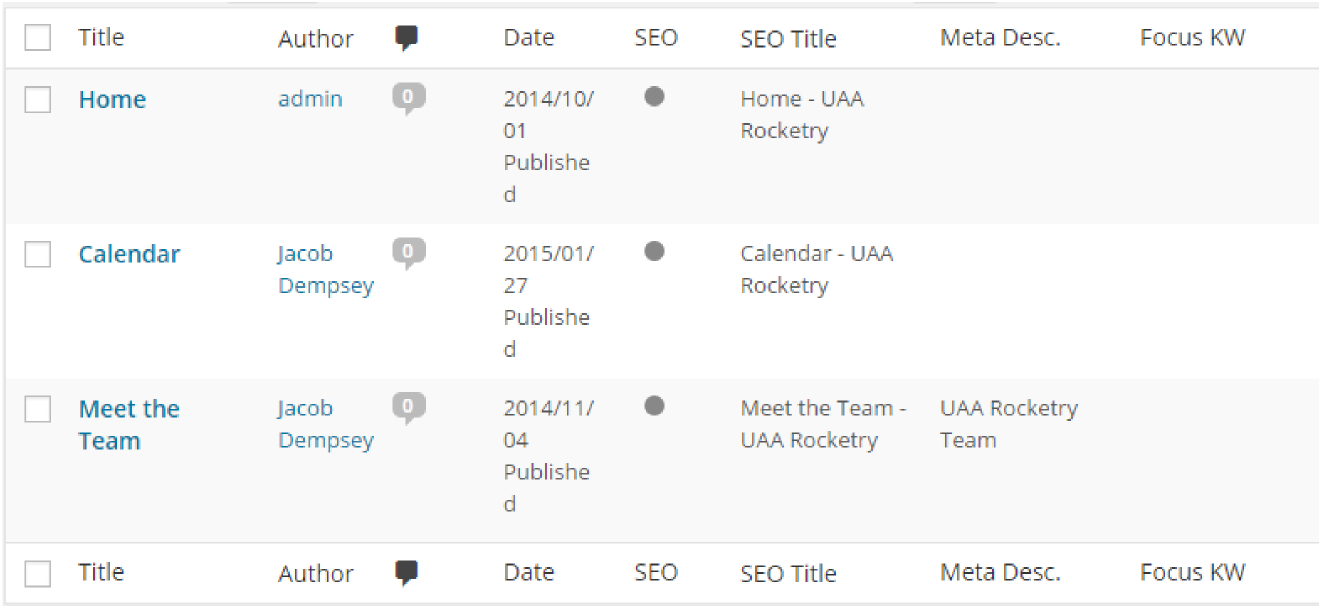

Figure 19: WordPress pages in the admin interface

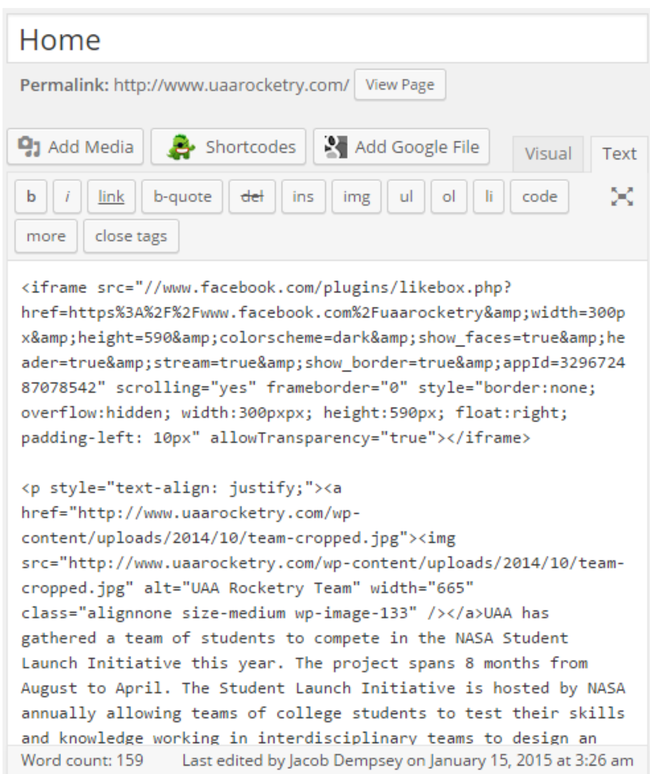

Figure 20: WordPress page content editing, plain text

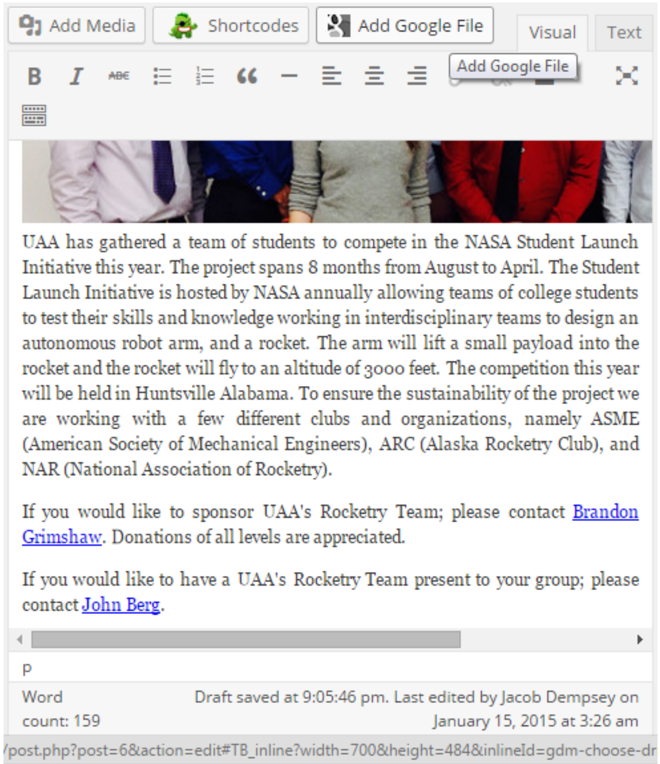

Figure 21: WordPress page content editing WYSIWYG

The will select the page they wish to update, and they can either update the page using WYSIWYG (what you see is what you get) interface, or they are able to update the HTML code directly.

4.2.2 AGSE

The original design of the AGSE called for the use of a BeagleBone Black as the controlling computer, but finished using an Arduino Uno. The ultimate result was a fully functioning autonomous robotic arm which consistently captured, placed, and secured the payload in the rockets payload compartment.

4.2.2 Competition Results

Both the rocket and the AGSE performed flawlessly in the competition. The rocket reached 2840 feet, in the simulations, we predicted the rocket would reach 3000 feet which was the goal. In previous simulation flights the rocket reached about 50 higher than the simulation. At this time, we have not determined what caused the rocket to be so far below the simulations predicted height. However, it was close enough to be the third closest rocket to the 3000 foot mark.

As a team, we won an award for the best overall presentation of our rocket and AGSE as voted by the college teams present in the competition.

4.3 Discussion

4.3.1 Website

The ease of use of the website was a success, and the site worked very well for the submission and presentation of our documents to NASA. However, we found Facebook to be an essential element in engaging the public. Between the two, the website and Facebook, our public outreach on the web was very successful.

4.3.2 AGSE

As noted previously, we started designing the AGSE to run on a BeagleBone Black. I was still writing drivers for the stepper controller and servo controller when the AGSE was nearing completion. In an effort to speed up development, we switch to an Arduino Uno to take advantage of the public availability of written and tested drivers. This proved to be an excellent development decision as the final testing of the AGSE was performed the morning of the competition.

After the initial testing of the AGSE, we found the stepper controllers were overheating, which was causing them to loose steps and therefore, not properly load the payload into the rocket. Initially, we added heat sinks and fans, but found this was still not enough. After slowing the motion down we were able to consistently load the payload even after several hours of operation.

Chapter 5 Conclusion

5.1 Introduction

This was the First time UAA entered a team in the NASA Student Launch competition. The team members individually had no experience with large scale rockets, but all of the team members had a desire to engage in a multi-discipline project.

5.2 Conclusion

Working in a multi-discipline team proved to be great experience. Throughout the project, there was cross over between the Mechanical Engineering students and myself as a Computer Science student. The project highlighted the need for excellent interpersonal communication skills as well as the need to keep an open mind to new ideas.

In the beginning of the project, there was a great deal of excitement, but know clear plan as how to get to the competition. As time progressed our visions wove together into what was ultimately a successful run in our first competition. Team member broke off into sub groups with some focusing on the rocket, and others focusing on the AGSE, with myself and one of the Mechanical Engineers focusing on the electronic systems of both the rocket and the AGSE. Even as we began to specialize in our various rolls, we continued to meet together to discuss our individual portions and the project as a whole.

While the website and programming were primarily my projects, they were both a topics of active discussion at various phases of the NASA Student Launch project. Design ideas were offered, discussed, and voted on by the group. After the ideas were implemented, they were either approved or rejected by the team. This process was the basic model for each of the design phases of the project. For the mechanical parts the Mechanical Engineering students working on them build the prototypes in SolidWorks for the group's approval. The result was a functional website, a stable reliable AGSE, and an accurate airworthy rocket.

We could have improved the project by selecting our motors, servos, and controllers earlier in the project. As it was I received a stepper motor and controller for it about 4 weeks before we were scheduled to leave for Alabama. I spent the first 3 of those weeks trying to figure out how the stepper motor and controller worked. The stepper motor documentation indicated it operated on 3 to 3.6 volts and 2 amps. With the power supply I was testing with, it wasn't able to supply enough amps to move the stepper motor until I was feeding it 5 volts. After two weeks of trying to get the motor and controller to work together, I acquired another controller. When I hooked the second controller up, it worked as I expected it to, and proved the first controller I had been attempting to figure out, was defective. These delays prevented me from writing compete drivers for the stepper motors and servos in a timely fashion. Being short on time, the competition being less than a week away, we opted to switch to using an Arduino Uno to take advantage of the available, stable libraries for stepper motors and servos. This proved to be a great choice as we were able to implement a simple system to control the AGSE in a few days.

Once we began programming the AGSE, we found a few minor mechanical issues which prevented the system from moving through its required range of motion. After correcting the mechanical design, we found the stepper motors were over heating when the motors were operating under load. Our first attempt at correcting this issue was the addition of heat sinks and cooling fans. Initially, we thought we had solved our problem. However, as it ran longer, we found the stepper controllers were still overheating and missing steps. This final bit of heating was overcome through programming. The speed of the motors was decreased, and strategic pauses were added to allow for additional cooling.

In the end UAA was represented very well by its new Rocketry team, and the Rocketry team members learned many valuable lessons in working on multi-discipline projects. We hope we have laid the ground work for future teams from UAA.

5.3 Future work

There were features which we wanted to accomplish, but didn't have time to implement. These features were to add intelligence to the AGSE. We wanted to implement an IR sensor to give positive confirmation the end effector of the AGSE was over the payload. We also wanted to implement a camera system for identifying the location of the payload. If I had more time, these would have been great additions to project. While they were not required by NASA they would likely have been enough to secure the award for best overall AGSE.

Appendix A: References

Chapter 1

[1] NASA, http://www.nasa.gov/audience/forstudents/studentlaunch/home/, retrieved 2/3/2015.
[2] WordPress, https://wordpress.org/, retrieved 2/3/2015.
[3] _S, http://underscores.me/, retrieved 2/3/2015
[4] Beagleboard, http://beagleboard.org/black, retrieved 2/3/2015
[5] Eclipse, https://eclipse.org/downloads/packages/eclipse-ide-cc-developers/lunasr1a, retrieved 2/3/2015

Chapter 2

[1] PHP, http://php.net/manual/en/faq.general.php#faq.general.acronym, retrieved 2/23/2015.
[2] Adafruit, https://learn.adafruit.com/introduction-to-the-beaglebone-black-device-tree/device-tree-overlays, retrieved 2/23/2015
[3] FDTWiki, http://devicetree.org/Device_Tree_Usage, retrieved 2/23/2015
[4] Notepade++, http://notepad-plus-plus.org/, retrieved 2/23/2015.
[5] Sparkfun, https://www.sparkfun.com/products/retired/11876, retrieved 2/23/2015

Chapter 3

[1] Beaglebone Black System Reference Manual, https://github.com/CircuitCo/BeagleBone-Black/blob/master/BBB_SRM.pdf?raw=true, retrieved 3/12/2015.
[2] Beaglebone Black Schematics, https://github.com/CircuitCo/BeagleBone-Black/blob/master/BBB_SCH.pdf?raw=true, retrieved 3/12/2015
[3] Derek Molloy, Beaglebone Black P8 Header Table, https://github.com/derekmolloy/boneDeviceTree/blob/master/docs/BeagleboneBlackP8HeaderTable.pdf, retrieved 3/12/2015
[4] Derek Molloy, Beaglebone Black P9 Header Table, https://github.com/derekmolloy/boneDeviceTree/blob/master/docs/BeagleboneBlackP9HeaderTable.pdf, retrieved 3/12/2015.
[5] Derek Molloy, Beaglebone GPIO Programming on Arm Embedded Linux, http://derekmolloy.ie/beaglebone/beaglebone-gpio-programming-on-arm-embedded-linux/, retrieved 3/12/2015

Appendix B: Code and Documentation

WordPress theme code:

https://github.com/jadempsey/UAARocketry/tree/master/WP%20rocketry%20theme

Doxygen Documentation:

https://github.com/jadempsey/UAARocketry/tree/master/doxygen

AGSE code:

https://github.com/jadempsey/UAARocketry/tree/master/AGSE