ASSIGNMENT 3 REPORT

Aaron Storey (s3501900)

Alina Nguyen (s3588914)

Joshua Lawerson (s3333205)

IT World 23 (a.k.a. The three musketechs)

**Group’s Website:**

<http://assignment3musketechs.s3-website-ap-southeast-2.amazonaws.com>

**Group’s GitHub Repo:**

<https://github.com/s3588914/Assignment3-Group23>

**Group’s Google Docs:**

<https://docs.google.com/document/d/1f9AGp4uO_NSoeJ6-G2ClmDe_Jcoav3Z9WyKuUPksXdQ/edit>

**Group’s Presentation Slides:**

<https://docs.google.com/presentation/d/1uFhPQWXhLHBsUhhBBj0FsZngzCNunnmFuIn6v5yhEno/edit?ts=5b0368a5#slide=id.g3b61b10202_1_7>

**TEAM PROFILE**

***Personal Profile***

***Aaron Storey***

My name is Aaron Storey, s3501900 is my student number, and I am part of The Three MuskeTechs. My background is a heavy interest in IT, specifically programming and development. Some hobbies include building and woodworking, I enjoy the process of starting with nothing but an idea, and turning it into a finished product (this applies to physical and digital products).

My interest in IT like many kids born in or close to the 90s was apparent in my childhood, but really spiked when I first started learning how to program using PHP when I was 15.

My experience with IT started with working as a retail salesperson at a company selling all sorts of technology and worked my way up to the corporate IT department of said retail company, where I help maintain systems and tools that stores use.

***Alina Nguyen***

My name is Alina Nguyen, a member of The Three MuskeTechs that is currently studying Associate Degree of Information Technology - my student number is s3588914. Back when I was in high school, I focused on the business side of things, especially in Accounting.

Unfortunately, once I begun studying Business in University, I realized that it wasn’t something that I wanted to have a career in - so while looking for a different field to study, Information Technology caught my interest.

Therefore, I decided to give Information Technology a try, and so far my interest has grown. At the moment my experience in IT is very limited, what I have at the moment is what I have gained through my studies at RMIT.

***Joshua Lawerson***

My name is Joshua Lawerson, proud member of The Three MuskeTechs. I’ve spent several years working in advocacy and have come to a point now where I’m looking for a career change. I used to fence competitively and still enjoy the sport recreationally.

I’m a deeply competitive person and think that this would transition well into the IT field. My experience in IT so far has been limited to things that I’ve taught myself, largely revolving around working with excel.

My interest in IT has changed as I’ve grown, starting with an obsession with any video game I could get my hands on to now throwing myself into every programming language I have the opportunity to study.

***Group Processes***

***How well did your group work together in Assignment 2?***

Our group overall worked really well together, we were able to delegate tasks to each team member based on their strengths quite easily without any issues - also providing help to one another when needed. As a group we also came to a decision on what to do for our final project idea quite quickly, as it contains elements from each of our project ideas from Assignment 1.

***Will you be introducing any changes in process for Assignment 3?***

We will be utilizing GitHub in more depth as a method to store and share documentation between each group member. While we did this in our last Assignment, we will be doing this with a larger number of resources and files.

We will also ensure to create our report as a single PDF file as intended in the assignment brief, as our last assignment we instead included our report as a part of the group’s website.

***Career Plans***

***Compare and contrast the career plans, including ideal jobs for each person in the group.***

***What common elements are there?***

It’s difficult to point out job-specific skills here, as these can and will change drastically in a short amount of time. Instead, we’ve decided to dot-point interpersonal elements which would be required for all 3 jobs.

* Adaptability/Dynamic
* Communication skills
* Able to be a part of a team
* Able to embrace change
* Keep up to date with new trends and standards
* Organised
* Planning skills /Time Management
* Eye for detail

***What differentiates each position from the others?***

All 3 jobs are similar in many ways, and as above, without going into too much job-specific detail, noted down are some elements which make each job unique in comparison to the others.

*Full stack developer:*

* Broad and quality knowledge on both front-end and back-end design and implementation
* No specific expertise on one particular development technology

*Web developer*

* Focus on design and creativity

*Penetration Tester*

* Focus on testing and finding security issues/vulnerability

***How similar or different are your career plans across the group?***

All 3 career plans are based in core parts of IT, and are fundamental to any business IT solution, however the path and prerequisites slightly differ in terms of experience required.

To get a job as a full-stack developer, you would need a long and experienced history, working in different fields of software and web development. This is not a job you would typically receive as your first role.

To get a job as a web developer, you would need some sort of portfolio to demonstrate and show your previous works, professional or casual, along with displaying an eye for design and understanding of current technologies.

To get a job as a security penetration tester, you would require a vast knowledge of computer operating systems, IT infrastructure, common company practices, how to expose vulnerabilities, and how to then exploit said weaknesses. You would also have to show some level of ethics, to demonstrate to a potential employer that you’re not in the business for malicious intent, but rather to protect and prevent.

All 3 jobs all share the same basic elements which many jobs in and outside of IT require, such as communication skills, punctuality, eye for detail and adaptability. They also all require some form of experience, although web-development is one which out of the 3 would require the least formal experience.

**PROJECT DESCRIPTION**

***Overview***

***Topic***

Our project idea is to preload a Raspberry Pi with simple programmable games. This game device is marketed towards primary school children, as programming and coding in primary schools is becoming increasingly popular and essential in today's educational society.

Our objective is to create a straightforward device to use, so instructors can understand and help kids get started quickly and easily - we want this to be as simple as “Plug and Play”. Built on a Raspberry Pi, it is extremely portable. All the user needs is a monitor with HDMI.

***Motivation***

***Motivations for this project and why is the project important or interesting?***

Our main motivation is to encourage students at a primary school age to check out coding and programming earlier on in their education, as different types of development is quickly becoming a highly sought-after skill in the IT industry. Its new to all of us, therefore learning and getting hands on experience with a small computing device, and how to develop a game through programming with Javascript, HTML5 and CSS will definitely be interesting.

***How does it fit in with current IT trends?***

Small computing devices like Raspberry Pi computers are a part of a new era of portable, powerful and highly versatile devices. Using a Raspberry Pi in our project allows us to gain knowledge on using a new device for both software and hardware integration. Typically, projects created on Raspberry Pi’s are open-source, and some popular projects also contain documentation on what’s going on, how something works, and in-depth configuration. Learning is easy when there’s a massive community behind a new type of technology.

***What would it show to a future employer if you were able to work on this project?***

It would show we are a team who can utilise current technologies, interfacing with both software and hardware to create an interactive gaming experience. It would also highlight our ability to adapt to new and dynamic technologies and concepts, as our passion for IT and encouraging our younger generation to get into coding and exploring IT.

***Landscape***

***What similar systems or products are available?***

While we were unable to find an exact alternative of our planned product, there are various small computing devices in existence which wouldn’t require much alteration to achieve what we are trying to do. Listed below are some alternative small computing devices to the Raspberry Pi:

* [MinnowBoard Turbot Dual Ethernet Quad-Core](https://minnowboard.org/minnowboard-turbot-dual-e/)
* [BBC micro:bit](http://microbit.org/)
* [C.H.I.P.](http://nextthing.co/pages/chip)
* [Parallella](http://www.parallella.org/)
* [NanoPC-T3](http://www.friendlyarm.com/index.php?route=product/product&product_id=210)
* [Huawei HiKey 969](https://www.96boards.org/product/hikey960/)

***What competitors are there?***

Our major competitor would be the Makey Makey, as this is a product which allows the user to make all sorts of controllers for different applications, using quite literally anything that can conduct a small electric current. They use alligator clips to interface directly into their PCB board. Their product however seems to be more for tinkering and entertainment purposes only, and do not require any programming or coding.

***What points of difference are there about your project compared to what exists now?***

As of this moment, the closest product as mentioned above is the Makey Makey device, however their product appears to be more for tinkering and entertainment purposes only, and do not require any programming or coding.

What separates our device from simply purchasing a Raspberry Pi standalone, is ours would come preloaded with Raspbian (the Pi’s OS), and would also include our own open-source game software and API, along with documentation on how to modify and edit the existing games.

Our product is not a conventional arcade game machine such as “RetroPi”, ours is more of an educational device aimed towards children and early learning.

***Detailed Description***

***Aims***

***Specific aim***

*O*ur overall aim for this project is to design and create a fully functional programmable arcade game developer device that runs off a Raspberry Pi. Also to have a working product complete with an external arcade-style controller to interface directly with the game. The Raspberry Pi and game controller together will allow an end-user to plug straight into a TV or Monitor, power on the Pi, and have the game menu screen auto-launch to begin paying a game in the browser

***Smaller Goals***

How do we achieve our aim? Below are some paragraphs which outline and detail what is required to reach our aim.

First and foremost, we need to ascertain a Raspberry Pi, Micro SD card with adapter, and the parts required for constructing a simple joystick and button controller. Josh has taken charge of designing, ordering parts, and constructing the game controller, while Aaron has purchased a Raspberry Pi Model B3+.

Our next goal is to research the software and operating system required to install on to the Micro SD card for the Raspberry Pi to run. There are many different kinds out there, however actually installing the OS has been made very simple thanks to many guides and tools.

Alongside this, we need to investigate more on our preference of HTML5/JS library called Phaser, and begin learning how to construct a basic game using their framework.

After ensuring the OS is installed on the Pi, we can then begin programming a basic starter game to learn how the language works. Getting a basic sample game created will allow us to continue to work on and develop our understanding of how the library works, what is possible, how to best structure elements of the source, how to best run the game in the browser and how to allow for expansion and modification.

After developing a working prototype, the next goal would be to find an investor to fund our project further, and bring on additional professionals to assist in marketing and further developing our product.

After working with a professional developer and technical lead, the source code of the game should be complete, and the question of how it will be run in the browser should be discovered. The next mission will be to develop a solid and concise set of documentation on the games we have created.

Testing our product will be essential, using contracted testers and QA professionals, we will go through our product, identify any issues, ranging anywhere from how easy the unit it to “Plug and play”, or any problems with the games themselves, or any issues with modifying or coding your own game. All issues raised will be prioritized, and fixed accordingly before release.

Marketing and advertising our product will be the last step while working closely with real end-users as testers. We will need to get in touch with various organisations who educate or teach kids, and reach out in hopes to run informative demonstration sessions, in a bid to raise interest.

Full release will only be done when the product is stable for an initial release, some minor bugs or issues are tolerable so long as they do not inhibit or seriously affect the user’s experience. There will never be a final stage to this project, as it will be constantly updated, transformed/upgraded, and expanded upon.

***What are the most important parts of the project?***

There are multiple milestones in this project.

The first major milestone, was to research and learn about a suitable language to begin programming a game in, in this case we selected and began researching the HTML5/JS language called Phaser.

The second major milestone was actually purchasing and receiving the hardware components required - as we knew whatever we ended up deciding for programming would inevitably run on a physical device, so this has become an important task.

Other important tasks such as installing and configuring a Raspberry Pi (assuming at this point we had received the Pi), along with building a game controller, were also required to allow us to build a working prototype for the Raspberry Pi for testing and further development.

***Which parts should have priority over the others?***

* + *Creating at least one game, as we should be able to at least provide a proof-of-concept.*
  + *Installing the Raspberry Pi’s OS, as this will allow us to begin building and testing the game on the Pi itself.*
  + *Construct a game controller, as we can then expand development to support an external game controller.*
  + *Further configuration of Raspberry Pi, to set things such as auto-run game upon boot, setup proper directory structure and source code comments.*

***If we have only enough time or resources for one of our goals, which should it be?***

Developing at least one game which can function to the point of being somewhat playable and modifiable.

***Plans and Progress***

***What will our project will do?***

Our project idea is to preload a Raspberry Pi with simple programmable games. This game device is marketed towards primary school children, as programming and coding in primary schools is becoming increasingly popular and essential in today's educational society.

This type of device is great for those with little-to-no knowledge on programming, because the games available will be entertaining and easy to manipulate as the game codes will be annotated, making it easier for users to modify. This device would serve well in schools, and has many applications. One example would be a fantastic group project or assignment for kids to explore coding.

Our objective is to create a straightforward device to use, so instructors can understand and help kids get started quickly and easily - we want this to be as simple as “Plug and Play”. Built on a Raspberry Pi, it is extremely portable. All the user needs is a monitor with HDMI.

There will be an included controller that will be very customisable in the sense that users could add and map different buttons and controls to perform different actions in the games. This helps teach interfacing software and hardware.

***How we will do it?***

Using a game programming framework, we will design and develop at least one working game to deploy to the Raspberry Pi, which will interface with an external controller.

This is dependent on obtaining a Raspberry Pi model, building a working controller (including finding a way to interface the controller to the Pi successfully).

This also relies on learning enough about the previously mentioned game framework to commence programming a simple game.

***How far we have gotten with developing any features or outcomes from our project?***

At this point in time, we have chosen a game library called Phaser, which is an HTML5/JS framework to assist in creating browser-based games. Using this language, we developed a basic “brick” game which involves using a paddle to deflect a ball to knock out all the “bricks” above.

We have also obtained a Raspberry Pi Model 3B+, and the parts required to build a basic controller that consists of a joystick and 2 buttons. We installed Raspbian on the Pi as the base OS, and also constructed a basic controller using a box as the shell. The controller has a USB 2.0 Type A interface which can plug directly into the Raspberry Pi.



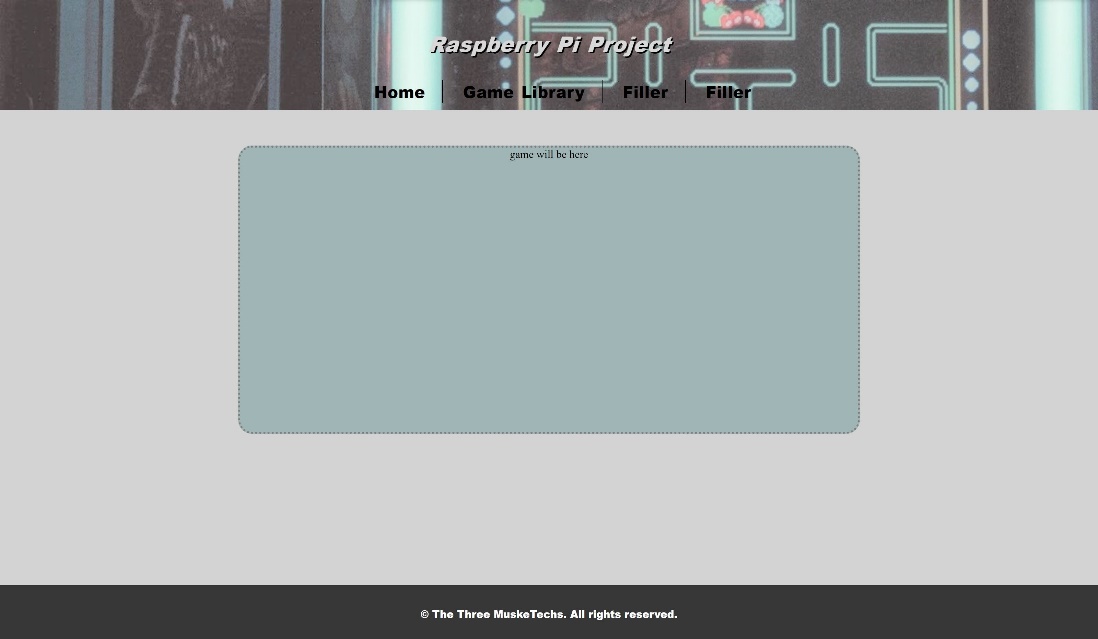
We have also created a website container for future games which can be expanded as required. The development of the container to house the games has not been completed, what we currently have is more a proof-of-concept, however the layout will remain similar if not the same should we have made further progress. This website can be adapted to run off a Node.js server, or it can remain as a simple HTML/CSS website, served directly off the Pi’s filesystem.

***How it began , how it has progressed & what stage of the plan we are up to?***

It began with our main motivation which was to encourage students at a primary school age to check out coding and programming earlier on in their education, as different types of development is quickly becoming a highly sought-after skill in the IT industry. With our product, the end user can directly edit the game source code, this way kids will get a hands-on learning experience to modify and change the way a particular game works. This helps bring out creativity, and encourages them to think outside the box.

Currently, in the market there is no Raspberry Pi devices with our product idea directly marketed towards children. The closest competitor would be Makey Makey is quite similar in the sense that their product is encourage the user to tinker with a device and using their imagination and creativity. Makey Makey is a device which can change basically anything that can conduct a small electric current into a controller by using alligator clips to interface directly into their PCB board. However, they do not require any programming or coding - not allowing any modifications.

As of today, we have created a controller, created and programmed a basic proof-of-concept game, and started work on a simple website container to host the game. While the container website is not complete, the basic wireframe and layout has been chosen and created (which can be seen down below).



In regards to the game controller interfacing with the Raspberry Pi, we are yet to establish a successful connection to directly control the POC game we developed. This is because we have not yet chosen or created an HID adapter to listen and map the input from the controller, to key-down events on the Pi.

Our next steps will be to successfully interface the controller to our game by developing a HID adapter to run in the background on the Raspberry Pi, to handle the input. Also to merge the game with the website container, and finish off programming for the game. This also includes cleaning up the source code, commenting and documenting the various methods and classes, and deciding if the code will run on a local Node server or compile to a includable JavaScript file.

If funded, we would hire IT professionals to help advance the device to the next stage, implementing additional games, and make it easier for kids to play and modify. (Developer with web/game experience, technical lead, business analyst and product marketer).

***Dead-ends that we have followed? Decisions we made? Changes that have been made to the project plan?***

At this point in time, we have not run into any dead-ends on this project, however we have changed our direction regarding the project target market.

We originally had the idea to target our product to kids for the sole intent of entertainment. However through group discussions and research, the market for educational coding platforms was found to be substantial. Therefore we decided to not only create basic fun games to play, but to also build them to be modifiable and easy to tinker with.

We were not clear at first on where the source code for the included games are stored, and this is still something we are discussing. Tentative plans are to preload the source code to include a set of standard games, with the ability to download additional games as they are developed, from a repository of sorts.

Besides the above, we have not made any other changes to our project plan or idea.

***Roles***

For our project we decided to delegate tasks to team members accordingly to their strengths to ensure maximum efficiency - however the team will help one another when another requires assistance.

Alina will be in charge of web development, as this was her ideal job. She has some knowledge in the area, and she sees this as an opportunity to further expand on her experience.

Aaron will be the lead developer, similarly as Alina this is his ideal job. Out of the group, he has the most experience with programming in other languages. This is the first time Aaron is doing any game programming using a JavaScript/HTML5 framework.

Josh will be our Business Analyst, as he has previous experience in business during previous career positions. He suggested marketing ideas and explaining the benefits of each one. The group as a team then decided on a final idea. He also assisted by creating a controller for the device.

***Scope and Limits***

The scope of this project is one we feel is reasonable and deliverable. We don’t expect to create a fully functioning end product, or something revolutionary as a first project. Our scope is defined as follows:

* Creating at least one working game to run in a browser
* Design a website container to hold the game
* Construct a game controller to interface with the Raspberry Pi

Additional deliverables which we may possibly achieve, but do not expect to initially reach at this stage of the project are:

* Compiling a simple functional game built within the website container on the Raspberry Pi
* Interfacing the controller with the Raspberry Pi to directly control the above game
* Developing additional games of similar complexity and interactivity
* Constructing a housing for the Raspberry Pi similar to the style of the controller

We don’t expect to achieve the additional deliverables in our project due to lack of time, however should time somehow permit, we have them to work on.

***Tools and Technologies***

***Raspberry Pi***

Our project runs off a Raspberry Pi Model 3B+. This is a small computer which has both wired and wireless networking built-in, along with USB interfaces to allow for direct connection of many different peripherals and accessories.

The Pi uses a Micro SD card to store it’s operating system, and we have chosen a particular linux distribution (or “distro” for short) called Raspbian. With this free distro installed, we can use the Raspberry Pi in similar ways to using a regular computer, albeit smaller and much more portable.

Raspbian as an OS is very popular to use on a Pi, as it is very versatile, and is the most popular “general use” distro. This particular distro as with many others has a FOSS (Free and Open-Source Software) license which allows us to use this software with minimal limits.

**Development Software**

We are using a mixture of HTML5, CSS3 and JavaScript to create our websites and games. For our game development, after researching different options we decided to use a JavaScript HTML5 framework called Phaser.

Phaser has an MIT License, which allows anybody to edit, modify, commercialise, and profit off their software provided their copyright notice is included in each distribution of anything that uses their code. Phaser is easy to obtain, and cross-platform.

Another thing which makes Phaser a great choice is how it can be run. Phaser is written in an OOP (Object Oriented Programming) style, and can be either run off a Node.js server (locally or remote), or it can be compiled into a series of .js files ready to be included into a simple HTML file.

**Game Controller**

The game controller that Josh built is comprised of a small PCB board which allows for components to be directly plugged in to, and has a single USB 2.0 cable running out of it. The components we have for the controller currently include a joystick, and 2 buttons. These can be added to later if need be should we require additional buttons.

***Testing***

Testing our product will be vital to create a kid-friendly educational device. We need to make our product fun and easy for kids to interact with, otherwise they will lose interest quickly. To test our product, we will contact various primary schools for the purpose of having children play with our device.

On-site at each of the primary schools, we will put the kids (ideally from year 3 to year 6) into smaller groups to initially play the games we have created, before showing them how the game code can be modified to do something different. The goal will be to get the students interested enough to have a shot at modifying particular parts of the games.

Throughout these sessions, we will be noting down what worked well, what seemed to have students react positively, and when they are starting to lose interest. Also asking them questions such as; Was it easy to set up the device? Were the annotations in the game code easy to understand? etc.

This process will then be repeated after a few minor or major changes for different groups of students to ensure that the product is improving.

***Timeframe***



***Risks***

It is risky using any new technology for a project which you haven’t used before, and many of the components we will be using are all brand new to us. There is the risk of learning a new framework such as Phaser being very time consuming and difficult.

One way we have tried to minimise this risk is using a lot of documentation and guides online on the basics of this framework, which has helped substantially. With good documentation, it is not so much of *if* we will be able to learn the language, but more a question of how long it will take to figure out.

Another risk was the type of game we originally had in mind to create was a basic snake game. This was going to take more time than we had to develop, so we instead chose a less complex game - Brick. This not only was easier to develop, but has provided us with a more fundamental understanding of how to use Phaser. This will allow us to expand and create more games in the near future.

***Group Processes and Communication***

Our group has had great experience reporting to each other and discussing ideas through our google docs. We have also all exchanged phone numbers and exchange important or urgent information through here, whenever necessary. As our communications cross multiple platforms there’s not a large risk of group members not seeing or responding to a message. If we’re not online at the same time, working on the google doc, it’s incredibly easy to just give each other a quick call or send a text.

We started this project with a firm, shared belief that discussing our assignment in person and working together physically would be to a massive advantage. To this effect we have had multiple meetings in the University library and even excuse ourselves from the communal class so we can work more comfortably. Throughout the assignment period we’ve been meeting regularly, at least once a week within class and then also meeting outside of class hours.

**SKILLS AND JOBS**

***Programming Lead***

***Position: Gameplay Programmer***

***Company: Party Pi, Inc.***

***Minimum Requirements***

* + - 2 Year industry experience, gameplay design.
    - 3 years industry experience, HTML5, Java coding.
    - Provide 2 references.

***Key Duties***

Party Pi’s new gameplay programmer will be working closely with a children’s education consultant to develop educational but more importantly, fun gameplay ideas and mechanics. The Gameplay programmer will be responsible for the development of a software platform enabling easily altered gameplay mechanics, targeted towards children aged 5-12. HTML5 and Java coding are deemed necessary with experience in other, similar areas looked upon favorably.

***UCD Lead***

***Position: Design Lead***

***Company: Party Pi, Inc.***

***Minimum Requirements***

* + - 5 years, any design experience.
    - 2 years experience designing for software, consumer electronics.
    - 1 Years industry experience, education/children’s targeted material.
    - Experience in fabrication.
    - Portfolio to accompany resume.
    - Provide 2 references.

***Key Duties***

The design lead at Party Pi will have their finger in a lot of pies, pun intended. The lead will be in charge of all design decisions within the developing software, as well as a physical interface, i.e. controller and console. The successful candidate will have experience in a fast paced and innovative environment, being able to quickly fabricate low fidelity mockups as well as high fidelity prototypes for promotion. As the UCD lead will also be responsible for the creation of much of the design of the system, experience in that field will be looked upon favorably.

***Website Developer***

***Position: Webste Designer***

***Company: Party Pi, Inc.***

***Minimum Requirements***

* + - 3 years industry experience with HTML.
    - Experience with development across multiple platforms.
    - Demonstrated UCD Experience.
    - Portfolio may be submitted with resume.

***Key Duties***

The website developer at party pi will be responsible for all development in our on-line space. As there is the intention for some aspects of Party Pi’s platform to be hosted online the successful candidate must be comfortable with actioning this. Experience across multiple platforms will help candidates.

***Early Education Advisor***

***Position: Education Consultant***

***Company: Party Pi, Inc.***

***Minimum Requirements***

* + - 3 Years industry experience with children’s education.
    - Industry experience in the IT space. This will exclude education.

***Key Duties***

The Education Advisor with Party Pi is a crucial role, albeit not a overly technical one. The successful candidate in this space will have gad multiple years experience educating children, preferably between the ages of 5-12. They will also have had a background, to some extent, within the IT world, and know how significant education in this area would be to kids.

**GROUP REFLECTION**

***This entire unit was largely based around group assignment work, and one major project topic. Below are some notes from each group member regarding what went well, what could be improved, surprising elements, and some things we have learned.***

***Aaron***

***What went well?***

I think our group was compatible regarding our interests and knowledge. We each had something to contribute to the team, and got along well. Our group work was done primarily on Google Docs, and we each put effort in to completing the work we assigned each other.

***What could be improved?***

Our group’s size in numbers was low, I think we were one of the only groups who had just 3 members. This was not done on purpose, and I don’t think it impeded our ability significantly, however having an extra team member may have helped in some cases.

***At least one thing that was surprising.***

The number of students present in the last tutorial in comparison to the number of students who came to the presentation (this is not actually surprising).

***At least one thing that you have learned about groups.***

The value of having people with different opinions and strategies, and how that can be extremely positive in various instances.

***Alina***

***What went well?***

Overall, I think we did quite well once again for this final assignment. All members of the team participated and contributed in the project. Even though we all had different schedules, everyone make sure to try and put effort in meeting up to discuss what they were doing and if they needed help with anything. Since we did our work on Google Docs, it was easy for everyone to put in their work and keep track of what needed to be done.

***What could be improved?***

Besides the number of group members, maybe next time it would be better to meet up more often to do the work together rather than doing most of the work remotely on Google Docs – but really depends on everyone's schedules.

***At least one thing that was surprising.***

Even though we only had three team members in total, we managed to get quite a bit of work done, however would be nice to have extra team members to share the workload.

***At least one thing that you have learned about groups.***

While working as a team, you become more open-minded to different ideas and learn how to incorporate it together to form a final idea. Also by doing this, it allows you to think differently in comparison to how you usually think and the way you do things.

***Joshua***

***What went well?***

The construction of our Raspberry pi uit and remote control was exceedingly straight forward. There were so many online tutorials and resources that any question I had could instantly be solved. The community online for small computing devices is so helpful.

***What could be improved/or was surprising?***

My presentation skills were surprisingly rusty. Despite years of industry experience, presenting to large groups, I was intimidated having our lecturer see my ideas. This could be improved by me taking some time to get to know James and allowing myself to become more comfortable around him.

***At least one thing that you have learned about groups.***

I’ve learnt that I really enjoy sounding ideas off of people who share the same passion as I do. Coming from a career where I was being paid to be the negative voice in the room, it was really liberating to be able to get excited about an idea with Aaron and Alina. They’re really good kids and will do amazingly in their careers.

***Group***

Our group worked together exceedingly well. Aaron and Alina started off this unit already knowing each other through past study and Josh was readily welcomed into the little group, which we ended up naming the MuskeTechs, after the three Musketeers. The comment trail from our github might not be the most reliable thing in comparing group input. Our project involved a couple of physical project items as well as a couple of online project items. All up the amount of work into the assignment was probably equal, but the amount of work put into building our website was not. We divided our tasks in a way we felt would express each members talents best, and Alina asked to spearhead the design of the website and organising the content there. She’s done an absolutely brilliant job. Aaron and Josh focused on the construction of the raspberry pi unit, the joystick and the interaction between the two. Aaron also design a simply “brick” game, as a way to show off our computer and controller.