ASSIGNMENT 3 REPORT

Aaron Storey (s3501900)

Alina Nguyen (s3588914)

Joshua Lawerson (s3333205)

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

**Group’s Website:**

<input website link when done>

**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.

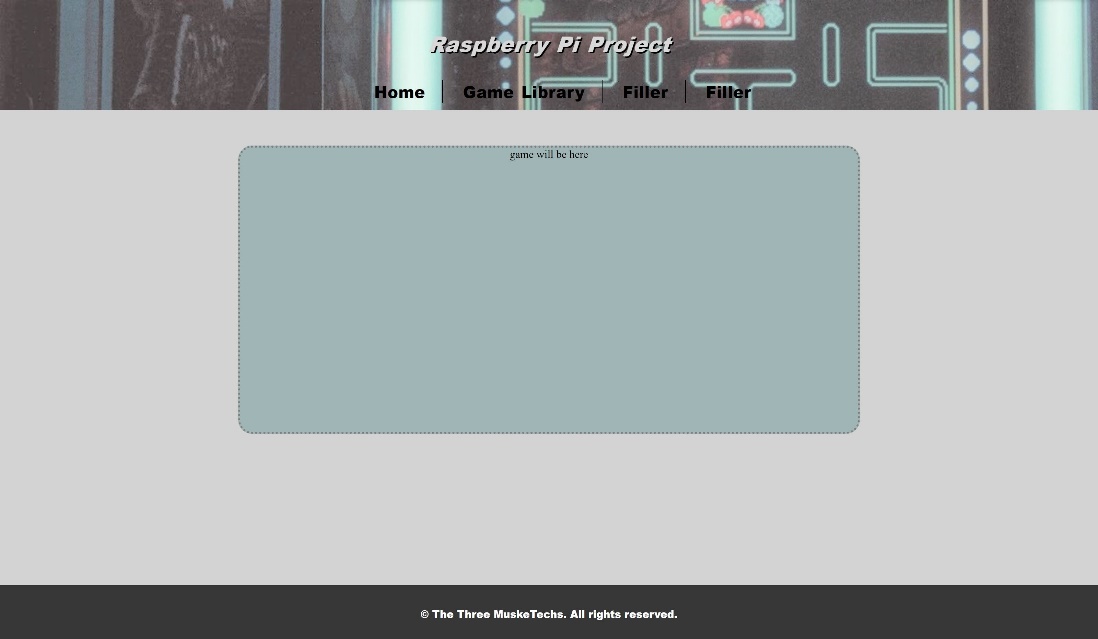
We have also created a website container for future games which can be expanded as required. The development of the container to house the games has not been completed, what we currently have is more a proof-of-concept, however the layout will remain similar if not the same should we have made further 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?***

***Roles***

This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler.

***Scope and Limits***

This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler.

***Tools and Technologies***

This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler.

***Testing***

This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler.

***Timeframe***

This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler.

***Risks***

This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler.

***Group Processes and Communication***

This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler.

**SKILLS AND JOBS**

***Filler***

***Filler***

This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler.

***Filler***

This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler.

**GROUP REFLECTION**

***Filler***

***Filler***

This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler.

***Filler***

This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler. This is a filler.