*Professional Practice*

*Project – Arduino*

*Simon Says Game*

*By Noel Melia*

*&*

*Eoghan Muldoon*

Contents

[Introduction 3](#_Toc38476126)

[System Requirements 3](#_Toc38476127)

[Technology Used and Why 3](#_Toc38476128)

[GitHub Repository 4](#_Toc38476129)

[History 5](#_Toc38476130)

[Design Methodology 5](#_Toc38476131)

[Simply Breakdown of Project 6](#_Toc38476132)

[Features of the Implementation 6](#_Toc38476133)

[Limitations and Known Bugs 6](#_Toc38476134)

[Testing Plans 7](#_Toc38476135)

[Recommendations for Future Development 7](#_Toc38476136)

[Screencast 7](#_Toc38476137)

[Conclusions 8](#_Toc38476138)

[References 9](#_Toc38476139)

# Introduction

This module is intended to bring together many of the best practices that the student has learned in the previous semesters. It gives the student an opportunity to design, develop and deploy a project, either individually or in a group environment, delivering a piece of software in a timely and standards driven manner.

We choose to make a game on Arduino because we wanted to use hardware along with software and thought it would be interesting to use it. We both agreed on the idea and wanted to see if we could complete a project together and try as we knew could do something when we put our minds to it. We also knew it wouldn’t be easy. This was the beginning and knew we would have to do a lot of research as it was a new type of system.

# System Requirements

Requirements for the Arduino Kit and IDE would be as follows: -

Windows 10 or less / Mac / Linux etc

Java 8 requires a Pentium 2 266 MHz processor and 128 MB of Ram – Part of the IDE Arduino has

600MB of free disk space for the install of the Arduino IDE.

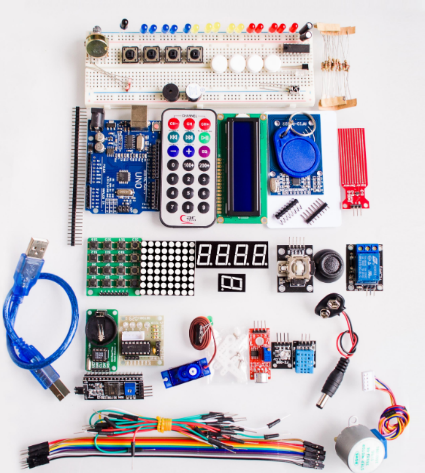
But this is only a guideline for the device to work smoothly and probably.

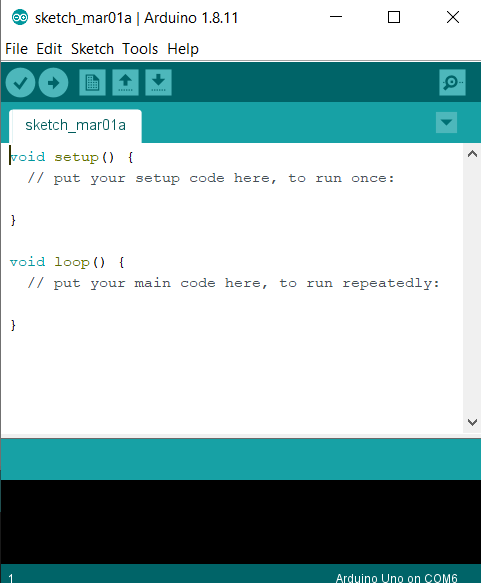
The project would have to be along the lines of what we are doing in college and technology of our era.

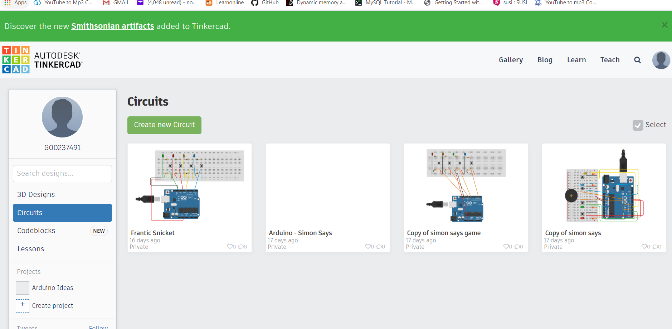
For this reason, we have chosen to design something with Arduino, and it would be interesting in the weeks to come. We both bought an Arduino kit and by the looks of the set and components that came with the kit we would be ready to make something straight away. But over this report we will tell you about how there is two differences in should and will.

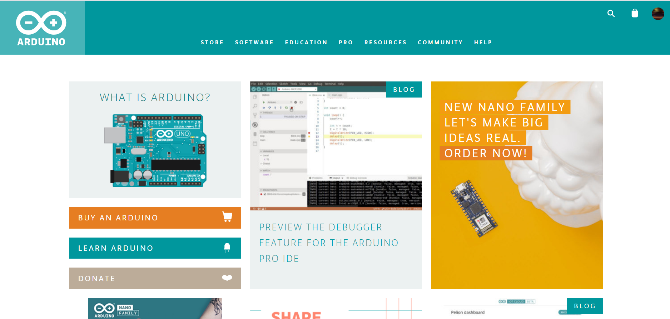
# Technology Used and Why

The technology that we choose was based on the product and experiment that we as a group what to design and build together. It would be tricky, but we were prepared to work with whatever we could get our hands on. The design is based on a product that we seen on YouTube and liked what it done. Throughout the process of the design we found a great website stated below which meant that we didn’t have to carry around the Arduino kit in college and that we could just design the final product online and transfer it on to the kit at the end. So, Noel had bought an Arduino kit first but once Eoghan bought one, it would be easier for use to make up and test the product.

Arduino Uno Kit and Components – Bought online and is a hardware component to make the device. On the right diagram

Arduino IDE – To Program what the device will do and make a program for its s it would run probably. On the left Diagram. C Language – Arduino Ide uses C Language to run program

<https://www.tinkercad.com/> - Website we used to make the device online and run the program there to. As seen on the left. We soon realised that the actual product and the website had faults. Some of the components in the Arduino kit weren’t in the website



<https://www.arduino.cc/> - Home website of Arduino, as shown on the right

# GitHub Repository

GitHub Repo: <https://github.com/NoelMelia/3rdYearProPracticeProject>

# History

From looking at the website of Arduino, I know it was around for a while and there was a lot of designs that would help in the design of the project. Arduino is the world’s leading open source hardware and software ecosystem. The company offers a range of software tools, hardware platforms and documentation enabling almost anybody to be creative with technology.

Arduino is a popular tool for IoT product development and STEM/ STEAM Education. There are many students that can develop or create games, music, smart homes and so many more using Arduino on. Originally started in early 2000’s as a research project by Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino and David Mellis in the Institute of Ivrea in Italy.

The first Arduino board was introduced in 2005 to help design students who had no previous experience in electronics or microcontroller programming and to create working prototypes connecting the physical world to the digital world. Since then it has become the most popular electronic prototyping tool used by engineers and even large corporations.

Arduino is the first widespread Open Source Hardware project and was set up to build a community that could help the use of the tool and benefit from contribution from hundreds of people who helped debug the code, write examples and create tutorials and so much more. Arduino continues to grow and develop to provide open source hardware and source to bring new ideas to everyone’s life.

# Design Methodology

We decide to do Simon say memory game with a few for tweaks the project. In our game we have a screen which tell you what level you on also can tell you what the high score is. The game has 3 buttons, 3 lights, a screen and a buzzer. We add additionally added a joystick but quickly found out that there was limits to the device and how many wires would be able to input into the Arduino kit. So, we had to decide quickly and move on without regrets but there was some. Noel wanted to add a joystick, but it was either the screen for display or the joystick for movement. We choose the screen as it looked nice with the game.

The hardest part of the game was keeping the wires in place and testing out the code with the device. With the certain situation that is going on, we had to do our own work at home. This was a challenge as we had to be part of a team, even though it was difficult because noel had bad internet connect but we always tried our best to stay in contact through messages.

We also found out that with our Arduino kits, they were similar in a lot of ways but Eoghan had certain parts missing. But Noel ploughed ahead and completed the device with what he had.

The games were made of 3 levels which were easy, middle and expert. The screen would start up with a message and display of the Screen Loading to 100%. Just to give the device time to load up. Once loaded, then the game will start once the user/ player presses a button to begin. The game begins and the player sees the light flash of the colour button to press. Each button when pressed has a different tone to recognise.

If correct then the game will recognise and continue to the next light and the screen will give a score and display the level also but if wrong the game will end and a message will display to the user “Hard Luck”. After this the screen displays a high score to the player. If the player wants to start again then they can.

# Simply Breakdown of Project

Simply Simon says game that allows a one player to play at one time. The player will get a chance to play the game to test their memory and how good remember is. In the game there are 3 lights that flash and 3 buttons to press with the lights. The screen is for displaying information to the player of how they are doing but also it can be a distraction to the player when the speed of the game builds up.

The game is simply, and anyone can follow the guidelines of the screen to play the memory-based game. All you need to do is press a button when a light flash that is in front of the led light. The aim of the game is simply but can be tricky if you lose concentration in the game and lights flashing.

The game is only one player but can be multiplayer if the players have patience and play for high score. The game stores a small bit of information of how the game works and can be useful in operation in how it works. The high Score is displayed to the player after each turn of the game.

To run the game the Arduino needs an IDE on the computer to be connected into and the IDE needs to run a simply but tricky C program. Then the program from the computer need to upload onto the Arduino uno the program which will be stored in memory until a new one is put onto it. Once uploaded the game can start by hooking up a battery pack or some sort of battery to power the uno.

Every time the game is powered up or turned on the game will start fresh from the program loaded onto the uno. There isn’t much more to it, apart from the wires and I keep saying it throughout the document, but they were the hardest part as they kept coming apart.

# Features of the Implementation

The buzzer let you know one of the lights are on for the sequence. The tone of the buzzer also change depends on which light is on. It also goes off when you press the button for the sequence, so you know it receive the input.

The screen let the user know what level they are on and the high score they have. Once the game begins then the user sees a Loading message.

The light let the user be able to remember the sequence easier. The lights get quicker as the game goes on

# Limitations and Known Bugs

We ran into a few numbers of bugs. The first bug we discover with using random function. The first time it ran it was random but when reset it would repeat the same sequence until we re-compile the code again. The way we went around that problem was using one of the analog port on the Arduino board to use as a seed. Which made the sequence different on reset.

We ran into a limitation with the amount of port that is on the board itself. The Screen use a lot of port to get working but lucky there a backboard that cut down on the amount of wires that is used. To give the project a more of a clean look to it.

Due to the port we couldn’t add in the joystick and the screen. We both decide that we prefer the screen as it can be used to debug but output the high score and level you are on.

# Testing Plans

Our original testing plan was getting our classmate to test the game out but due to the circumstance we didn’t have the change to. We also wanted to do a survey with our classmates which would condense of 3 questions for the students and testers to answers.

Noel got his family to test out the model and there were some good points made by his family members.

1. The Wires. There were to many wires.
2. The Buttons. Buttons were hard to push.
3. The Sounds from buzzer. The tones of the buzzer were annoying at times.
4. The Lights. They were to bright.
5. Screen was a huge addiction to the game as it kept the player informed in what level they were at.

So, we took all their points into consideration and made improvements with the game and how the user or player would make it easy to use. We changed the sounds which we reduced the impact and duration of the sound. Also, the lights didn’t stay on to long so the player wouldn’t get distracted. Buttons and Wires, we couldn’t make any improvements to as they were part and parcel with the device. We have added the survey that we would have released to the testers if gotten the chance.

<https://docs.google.com/forms/d/e/1FAIpQLSe7Je7rYgrPQ-WWteAY7FtL-p4_AtQre8-iqPE4TXa2Ofn89A/viewform>

# Recommendations for Future Development

Once we got comfortable and familiar with the design and code of the project. We started to investigate more advanced projects, such as: -

* Joystick to make the level harder
* Sensor as part of the game so when the user moves there finger over the sensor

These are only a couple of ideas that we researched, and it seemed that with the research that we accomplished would help greatly in the development of any project that we wanted to develop.

There was a problem that we meet, and we eventually got over, but it took time and patience. Some of the problems was thinking that it would be easy to design a game quick and smooth but that was far from the problem. We had to do a lot of research and divide up the project and meet regularly which we did, and it helped.

The hardest part of the project was coming up with a project that we could design and agree to make. Eventually we decided to make a Simon Says game on an Arduino and decided what was going to be on the game.

# Screencast

Noel did a Screencast about the actually project as Éoghan didn’t have one of the piece that is need for the to make the project. Éoghan is doing a screencast about the documation.

The reason we decide to do the screencast is because our project use both hardware and solfware and we thought it be easier to show now the project work and and why we design the way it is.

# Conclusions

**Noel Melia**

Working on a team with Eoghan was interesting and challenging at the start. But once we set up a plan of action and had regular meetings which we both agreed and concluded that it would help with the progress of the project. From working in an hotel as duty manager, I have worked as part of team before and understand of what had to done.

Some of the Project was challenging, mainly being the kit of the Arduino which I found to be challenging at the end of the process. Eoghan and I decided to get a kit each and we thought that would be easier to make up a project without having to wait for the kit from the other person. It was at the beginning and slowly we found out that the kit that we ordered wasn’t the same and hadn’t gotten all the products and components for both the kits. We then found out about a website that would have a lot of the components of the Arduino. It was slow to react to the movements of the computer but that wasn’t the main fault of the website. It was the lack of components that the website had compared to the Arduino kit. But we sorted something out in the end.

In my opinion if I was to do this project again, I would have a better knowledge of how to manage wires and technology on the device. But knowing now the situation of how to set up a project and the steps in how to manage a team project have made it easier. The weekly meetings with Eoghan helped a lot. Every Wednesday we would meet up and discuss what we had done and where we needed to try.

I must give credit to the weekly lectures in teaching us the basic setups of the course and GitHub was a huge help in that. From using GitHub since 1st year, I had a basic knowledge of the platform and how it worked but there was so much more I had not even known about the weekly meetings and using GitHub helped.

Towards the end of the module, unfortunately the covid– 19 had an impact on the timing and outlook of the project. It was tough with us not living close to keep good contact. We tired to keep up the contact through WhatsApp and it worked and helped a lot.

But overall, we got the project finished but we did meet a few problems but had to minimise the project a small bit but overall, it’s working. In the screencast I will describe the project and code that was inserted into the project from a lot of website and sheer research of what we needed to accomplish.

**Eoghan Muldoon**

Noel and I decide to the project with Arduino as we both like to do with something more with hardware and see how software works with it. The way I heard about Arduino is from YouTube and was looking to get one just to mess with.

It was challenger than expect especially since I start working at the weekend and so does Noel. That mean we had to do most of the work during the weekday. We took it one week at a time so see how we work together. We would meet every Wednesday in the library study room and talk in between lectures to see what the other person was doing and if they need any help with it or if we found something new which can be helpful to the other person we would tell each other for example the tinkercad website.

# References

<https://www.arduino.cc/> - Home website of Arduino

<https://www.tinkercad.com/>

<https://www.youtube.com/watch?v=dZZynJLmTn8>

<https://www.youtube.com/watch?v=kEuvtH1zlCo>