



UNIT 9 IT PROJECT MANAGEMENT

Assignment 2

Learning Aim B & C

Carry out project initiation for an IT project and carry out the planning, execution, monitoring, and controlling an IT project, using an appropriate methodology

Oliver Collins-Cope
2102775@rutc.ac.uk

Contents

Introduction	3
Project idea generation and solution creation	3
Project IT Problem	3
Creativity tools to solve problems	4
Solution	4
First alternative solution	5
Second alternative solution	5
Feasibility study for solution	5
Technical assessment	5
Economic assessment	6
Legal assessment	7
Operational assessment	7
Scheduling assessment	7
Sustainability assessment	7
Security assessment	8
Usability assessment	8
Feasibility study for alternative solution	9
Technical assessment	9
Economic assessment	9
Legal assessment	9
Operational assessment	9
Scheduling assessment	10
Sustainability assessment	10
Security assessment	10
Usability assessment	10
Preferred solution and evaluation	11
Project Requirements	12
Introduction to project including description	12
Requirements	13
Success criteria	14
Project Phasing	14
Functional and non-functional requirements	14

Requirement specification documentation	15
Typical project management process	18
Planning project management processes	18
Resource Plan	18
Time plan	18
The purpose of risk and issue management	19
Management of risks and issues	19
Project contingency	20
Project constraints.....	21
Ongoing monitoring.....	21
Execution and management processes	22
Time	22
Cost	23
Quality.....	23
Test log.....	31
Monitoring project management processes.....	31
Conclusion	33
Bibliography	34

Introduction

This document will aim to document all of the project procedures that take place in order to ensure that any mistakes or issues that arise can be learnt from in the future and allow future projects to be more successful based on this one. It will detail the whole project, from start to finish, ranging from an IT problem related to the project that will be solved and implemented, to planning the project and executing it consistently at the same depth throughout.

Project idea generation and solution creation

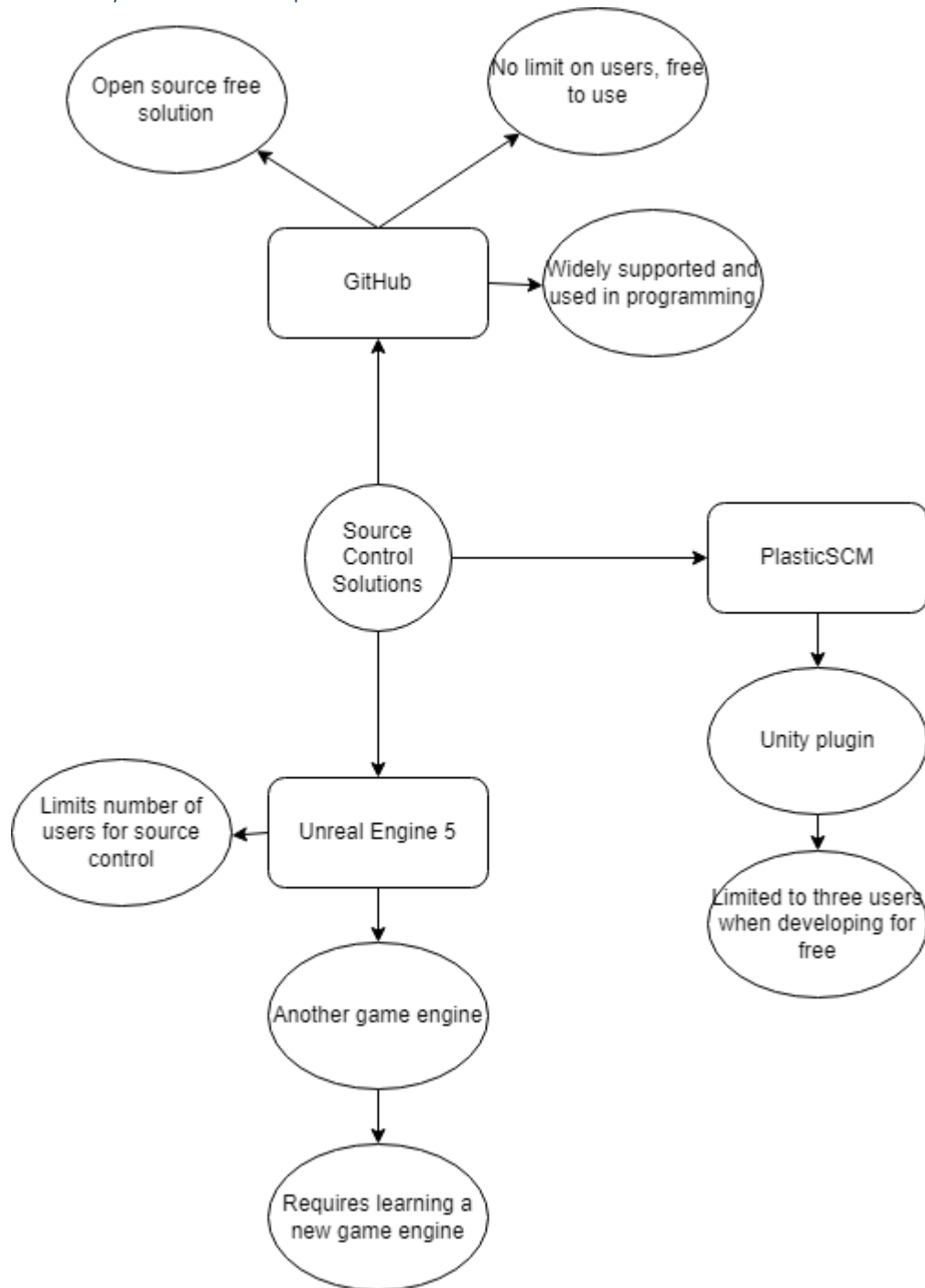
Project IT Problem

In my IT project where I am designing a traffic light mobile application, I have chosen to use the Unity game engine in order to successfully complete this and achieve an adequate outcome with my application. This means that all of the interfacing and coding will be done in C# and Unity respectively, and through this I will eventually be able to make my application successfully meet the criteria that was required.

Unfortunately, Unity is lacking in any kind of version control or cloud back up software. Version control, also known as source control, is the practice of managing and tracking different changes to software. This can either be things like code or interactive elements in some choice scenarios like a game engine. The ability to track changes and keep a log of when they happened is an integral part of developing code, as it allows efficient troubleshooting and crucial rollbacks to working versions of the code if a bug is introduced that has devastating effects. This also allows programmers/developers to track when changes are introduced and potentially identify and bugs like this, further proving that version control/source control software is crucial when developing.

This presents the final issue that has to be resolved. Due to the fact that there is no version control that comes with Unity, I will have to look at other software that enables me to do this. Fortunately, there are many solutions available, ranging from extensions to Unity or switching software completely, and this will be adequately researched and discussed below.

Creativity tools to solve problems



Solution

Due to the nature of creating an application in Unity, I plan to implement source control to allow multiple members of the team to work on the project without having to rely on a

convoluted way of tracking progress and giving each other the most updated version of the project.

My first and most likely solution that I have chosen for this is to use GitHub as my source control software. GitHub is a version control software, that “helps developers track and manage changes to the software projects code.” (Kinsta, 2022) It is widely used in software development as a way to manage source control, and is perfect for this small project that has an even smaller team to develop it.

Furthermore, GitHub is completely free and getting setup and started with it is relatively simple as it offers both GUI application options and CLI terminal based operations which allows the user to fully optimise their experience and use GitHub to the best of their abilities.

First alternative solution

An alternative to GitHub could be to use PlasticSCM. PlasticSCM is “Plastic SCM is a version control system optimized for game development and 3D projects, and as such, it supports separate workflows within the same repo.” (Unity, 2023)

This means that PlasticSCM is used with Unity in order to allow for version control software. On the other hand, while this means that the version control software will be all inside of Unity, it is likely that developers will have to expand and learn new skills in order to be able to use this, which is not favourable when considering deadlines and timescales.

Finally, PlasticSCM limits its free options to just three free members, before charging for each additional member, meaning that it could become very expensive for such a small scale project that will likely not generate much revenue.

Second alternative solution

Finally, the third alternative solution to using GitHub and PlasticSCM, is to use another game engine called Unreal Engine, and develop the mobile application for the traffic light in there.

The source control for Unreal Engine is built into the program, and can be activated by going through the content browser or the level editor’s preference window. Within Unreal Engine (UE), “Source control is used to manage changes over time to code and data and enables teams to coordinate their game development efforts.” (Unreal Engine, 2023)

Unreal engine source control is free to use for users although it does require some setup and therefore might require some time and assistance when establishing, and therefore is an effective alternative solution to the above mentioned two source control options.

Feasibility study for solution

I will be performing a feasibility study for the first solution I chose, GitHub.

Technical assessment

A technical assessment for my feasibility study means that I have to ensure that Git is able to fulfil its role as a source control software. GitHub has an extensive number of resources

that enable it to perform a large number of functions, such as testing a deploying applications.

Furthermore, it offers a comprehensive API that enables the developers and programmers to interact with GitHub programmatically. One of the crucial advantages of GitHub is its scalability. GitHub is designed to inherently work with a large group of developers and therefore it can be used for many different applications such as a traffic light application.

Finally, GitHub is also very secure due to it's need for two factor authentication, HTTPS encryption and other access controls, meaning that it is suitable for this project as there is minimal risk of there being a leak in the work.

This is a big advantage for GitHub as it ensures that developers have many options available to them in terms of features and security, ensuring that GitHub will remain significant and relevant in the future.

Economic assessment

GitHub offers many pricing plans that can be used by my team when development, however it is likely that given the smaller nature of our team and applications.

GitHub offers many plans, but these are some of the most common. These are:

Free plan;

This plan offers unlimited private and public repositories, however it limits the users to only 3 for the repository, among other features which are listed below:

- Unlimited public and private repositories
- Up to 3 collaborators
- 500MB of storage per repository
- Issue tracking and project management tools
- Code review tools
- Integrations with third-party services

Team plan;

This offers all the features of the free plan, but it adds additional features, most notably the option of up to 10 collaborators, and additional permissions, which are all listed below.

- All features of the Free plan
- Up to 10 collaborators
- 2GB of storage per repository
- Access controls and permissions
- Advanced project management tools
- SAML single sign-on (SSO) support
- Priority support

This costs 4\$ per person per month and is very viable as an option.

The cheap cost per person for the number of features available for 4\$ is an advantage that puts GitHub ahead of PlasticSCM which has a more expensive plans.

Legal assessment

GitHub has various legal requirements that it must follow as a web based application that holds sensitive data such as passwords and financial information. This means that it adheres to many laws such as GDPR, and intellectual property laws. GitHub complies with these through data protection features, code of conducts for reporting users, and things like license templates to ensure that users are not take advantage of.

The fact that GitHub complies with all of the relevant laws ensures that it remains significant in the development industry as software developers and engineers will not run into issues with the law when developing. (GitHub, 2023)

Operational assessment

Due to the issue being source control and not being able to do it with Unity inherently, I have looked to outside software to solve this issue. This means that in order for GitHub to be able to pass an operational assessment adequately, it has to excel primarily in source control, and this happens to be the case.

Alongside having many features available to help with the issue of source control, such as Git version control system, code review tools, issue tracking and project management, and integrations, it is clear to see, once going to a remote repository on Git, that Git is highly equipped to deal with source control and has a number of features available that do not need to be explained to be noticed.

Additionally, due to the fact that GitHub is well established within the development industry, its reputation and experience as software control is an advantage when using it in our project.

Scheduling assessment

The scheduling assessment determines the likelihood of the project being successful when using GitHub, provided that it is successful *and* on time.

This factor depends largely on the complexity of the project, the number of developers and the resources available for the project, such as hardware, software, and documentation.

GitHub does include tracking tools that allow progress to be checked and assigning tasks, meaning that using GitHub within the assignment seems like a logical solution and likely to help advance due to this built in feature.

Sustainability assessment

For the sustainability assessment of GitHub, while using GitHub itself does not expend much technology beyond setting it up, using the website and possibly a GUI application or terminal based application for Git, GitHub as a service does do its part for the environment.

This is made apparent through the sustainability of initiatives that GitHub is taking part in, and these include;

- Investing in renewable energy, which means that GitHub uses renewable energy for its servers and data centres.

- Optimising energy efficiency, this means that GitHub optimises its energy usage to be as efficient as possible and reduce any unnecessary consumption, helping them be more sustainable.
- Carbon neutrality since 2019, which means that GitHub has produced no more carbon than they have used since July 2019. (Brescia, 2021)

Security assessment

There are a number of features that GitHub employs to ensure the security of its website and services, and these include:

- Secure servers, GitHub employs a secure server system to prevent any unauthorised access and potential leaks.
- Two factor authentication, GitHub offers 2FA to users that wish to enable it, providing their account with more information.
- Software source control, GitHub offers source control that has access permissions to prevent unauthorised users from making changes to the software and pushing those changes.
- Penetration testing, GitHub performs regular testing on its own services and website to ensure that nothing can be breached, and the website remain secure, alongside the services it offers.
- Bug Bounties, GitHub also offers bug bounties, which are a way of incentivising users to alert them of bugs as they will provide them with compensation if they do so.

These security features of GitHub are significant and relevant as they ensure that GitHub will be well secured against any potential weaknesses, making it a big strength of the software.

Usability assessment

GitHub offers both a GUI application and a CLI terminal based option that will enable my developers to choose what they are comfortable with, ensuring that there is maximum productivity and work done. Furthermore, there are also a number of other features that allow the development to be enhanced. These include:

- Documentation, GitHub has an extensive documentation of its features that allow anyone to learn it and troubleshoot any issues.
- Integration with other systems, GitHub has a range of other systems that it integrates with, such as CI/CD tools.
- User testing, GitHub tests any usability issues and addresses those issues in case they might affect the users, which in my case would be the developers.

Finally, the usability assessment of GitHub determines that the multiple relevant factors, such as documentation and multiple options like GUI and CLI put GitHub ahead of its competitors and at an advantage.

Feasibility study for alternative solution

For my most likely alternative solution, I have chosen to use PlasticSCM as this way I am able to continue using Unity which my developers are experienced in and PlasticSCM expands on this.

Technical assessment

For the technical assessment, PlasticSCM also provides source control features that mirror that of GitHub, such as branches, merging, and general version control. Additionally, PlasticCM integrates with Unity as it is supported by Unity, (Unity, 2023) and was picked up by Unity to become an official part of the software. Finally, it also provides the option to manage binary assets which allows developers to modify scenes from inside of PlasticSCM.

The fact that PlasticSCM is apart of Unity is definitely an advantage as we know for sure that it works well when used with Unity.

Economic assessment

Similarly to GitHub, PlasticSCM has several pricing plan options the enable developers and our team to choose between the options that are available and those that work well for us.

The prices available range from \$7.50 per person per month for the small team edition, this includes up to 10 users and includes all of the standard features like merge tracking, code review, and branch visualisation.

There is also the professional edition that includes up to 50 users, however this costs up to \$45 per user per month, and therefore this is a very pricy option for our small project that likely will not need this many users or features that come with it. The extra features include, audit logs, automated testing, and integration with external tools.

Unfortunately, the higher price for PlasticSCM compared to GitHub puts it at a disadvantage when considering the economic assessment of both of the software options available and this is a weakness.

Legal assessment

Just like GitHub, PlasticSCM has to follow a number of laws and regulations like GDPR. The list of requirements that PlasticSCM has to follow include:

- Data Protection.
- Health and Safety for the employees of PlasticSCM.
- Intellectual Property, meaning that the project is properly managed and that PlasticSCM will help in managing this.

Operational assessment

PlasticSCM is easy to integrate into Unity as it is now apart of the software and simply has to be enabled and installed. Furthermore, there are a number of features of PlasticSCM that enable its operations to a higher level, and these include:

- Different features that allow the streamlining of development, such as branching, merging, and source control.

- Integration where PlasticSCM is able to integrate with a wide variety of other tools, like Visual Studio, Unity, Jenkins, and many more. This will enable my developers to work with the tools they wish to and that make them more comfortable.
- Ease of Use because PlasticSCM has a user friendly interface, enabling my developers to quickly pick up on it and expand on their skillset.
- Scalability where PlasticSCM is able to scale up and down depending on the size of the team and needs, and the complexity of the project.
- Finally support, where PlasticSCM offers a range of support options that enable developers to receive assistance if they need it.

As mentioned above, due to the fact that PlasticSCM is a part of Unity is a strength of this software and therefore puts it above GitHub when considering compatibility and overall usability of the software.

Scheduling assessment

The time taken to establish and maintain PlasticSCM, specifically the setup time, will not be long at all, and this means that getting it established and involved into the project will likely not take away from the project being able to complete successfully.

Given that PlasticSCM is a part of Unity, it will not be difficult to enable it and make it work for the project and the developers will be able to use it for the traffic light software.

Sustainability assessment

Compared to a large scale software like GitHub that is used all over the world, PlasticSCM is a lot smaller and more niche to access and use, and therefore it has a much smaller environmental impact.

Furthermore, due to the wide variety of source control software options, this means that the chance of PlasticSCM being used is lower and this lowers the environmental impact of it. Finally, despite this, PlasticSCM is committed to sustainability practices, which includes renewable energy.

Security assessment

PlasticSCM uses a number of features in order to ensure that the security of the software is high. These features includes:

- PlasticSCM uses HTTPS, similarly to GitHub, and also supports SSL (secure sockets layer)
- PlasticSCM also uses multiple authentication methods and permissions settings to restrict access from unauthorised users.
- There is also regular penetration testing on PlasticSCM to ensure that the software is secured against external threats.

Usability assessment

PlasticSCM has many features available that ensure it is a highly usable software that complies with current technological trends and ideas that people want in their software.

This includes an intuitive and modern user interface, the version control features it has that enable it to perform its role.

Additionally, its usability is supported by the many integration options it offers, and this combined with the customisation options help to make it a very usable software. Finally, the documentation of the software helps to enable my developers to learn to use PlasticSCM effectively with Unity.

Preferred solution and evaluation

My preferred solution for the project of making a traffic light application is to use GitHub. This is for a number of reasons, each of which shall be listed below and discussed in depth, but a summary would be the usability/ease of use, and the pricing.

My initial reason for choosing GitHub over PlasticSCM is due to the fact that GitHub is much more widely used and accepted by developers. This acceptance leads to more usage of this software and in the future if there is to be another software project, whether it be making something inside of Unity or programming it from scratch, GitHub can and likely will be used in this scenario, meaning that it would be much better to use something that can be reused at a higher level in the future. Furthermore, the higher use rate of GitHub is a strong advantage as it means it is approved by a large number of users and therefore likely continues to work well while focusing on community feedback, making this a strong strength of GitHub. Comparatively, PlasticSCM is much newer compared to GitHub and therefore has not had as much time to grow and become as stable, leading to this being a disadvantage and making it weaker than GitHub as an option due to the difference in quality and relevance of the software.

Furthermore, the developers that I will hire to create this application will likely appreciate using GitHub over PlasticSCM, whether or not they know how to use it. This is due to the fact that it will allow them to expand on their skills with GitHub, an industry standard for programming, and ensure that they grow as a prospective software developer and employee. Additionally, the wider range of options and features of GitHub, ranging from GUI applications and CLI options, to security features, this propels GitHub well above PlasticSCM in terms of quality, making GitHub much stronger and more significant than PlasticSCM, as well as giving it more relevance due to the chance of future uses by developers.

Continuing on, due to the optimisation and flexibility of GitHub, I believe that it will be a better tool to work on the development of the traffic light application, even if PlasticSCM is made for Unity. This is due to the fact that there is a chance the project will have to branch out of just using Unity in the future and incorporate some other languages to allow for more functionality, such as wireless connection to the traffic lights which might not be possible straight from Unity. The fact that GitHub has much more flexibility compared to PlasticSCM is, again, another huge advantage compared to PlasticSCM which is restricted to Unity, and pushes me more in the direction of choosing GitHub.

Next, due to the high pricing of PlasticSCM and the limited budget, I believe that GitHub would be the superior source control to choose. Coming in at \$3.5 cheaper per person per month for the smaller team of ten could eventually save us \$35 monthly, and due to the small nature of this project and the startup costs, it is important to save as much as possible for the project to ensure all of the staff are paid and the project can be completed efficiently. This is the largest point of contention due to the price difference it puts GitHub as much more advantageous compared to PlasticSCM, especially when considering the budget of the project and therefore making the pricing a weakness/disadvantage of PlasticSCM and an advantage of GitHub.

When considering the feasibility studies of both the solutions, most of the assessment sections remain relatively similar despite the fact it is a different product, possibly due to the fact that they are both source control software, and therefore this means that it does not do more than inform myself (the project manager) as to what would work better based on the context of our project and the software that we have to create.

To conclude, while I believe that PlasticSCM has many bonuses, such as being made specifically for Unity and having options like binary modification for the scenes, I believe that the widespread usage of GitHub/Git, alongside the many previously mentioned advantages of GitHub and weaknesses/disadvantages of PlasticSCM, along with the ability to use GitHub again in the future, that GitHub is the superior choice for my solution and is what I will introduce and implement into my project. Additionally, given the pricing and the budget that the project will be proceeding under, GitHub being the cheaper option only helps to further promote itself as the better option for this project and for the future as well. Furthermore, the pricing puts GitHub well above PlasticSCM as it is a strength of GitHub that I highly value compared to the weakness of the high price of PlasticSCM. Finally, its position as industry standard for software projects and within IT, I believe that using GitHub will help make my employees that I hire more valuable and increase their skillset with a relevant and valuable skill, likely meaning that they will stay with me as a part of the project team in the future.

Project Requirements

Introduction to project including description

As mentioned at the start of this report, this project will be developing a mobile application that holds a traffic light system. This means that the program we develop will have to function on multiple phones OS, such as iPhone and Android, and will have the continued functionality and features.

Also as mentioned above, we will be developing this within Unity as this provides port options to both iPhone and Android, as well as options to build on desktop if this is required of the project.

Additionally, the project will be using GitHub as its source control software, and this will enable the developers to work together and create the software efficiently and effectively create the traffic light application.

The base idea of a traffic light is simple enough to conceptualise as traffic lights are involved in day to day life and therefore this does need references to help create this traffic light application.

Finally, the purpose of this document is to help display the project going through its different steps, such as planning, design, executing, and monitoring and controlling. For this project we will be using an AGILE methodology as this allows for constant work flow and swift changes in case something needs to be adjusted and fixed if there are any issues. Furthermore, due to the nature of working with software, there is a high likelihood that we will not be able to introduce all the features that we would like to, and therefore it is important to be able to adjust on the fly.

Requirements

The requirements for the project have already been detailed by the client, and they are detailed below inside the quotations.

“Highway-RUTC Road Services needs to close one side of the road to traffic so that the repair work can be done safely. A temporary traffic light system is required to control the movement of vehicles safely through the roadworks. The traffic must be allowed to pass in one direction for the set time period before the traffic lights change, so that the traffic from the other side can pass through for the set time period. The client (Highway-RUTC) has provided a plan drawing of the roadworks, as shown in Figure 1, with the traffic currently moving from right to left. In Figure 1, traffic light Set A is on red and traffic light Set B is on green, which is Step 3 in the operational sequence in Figure 2.”

The figure can be seen below.

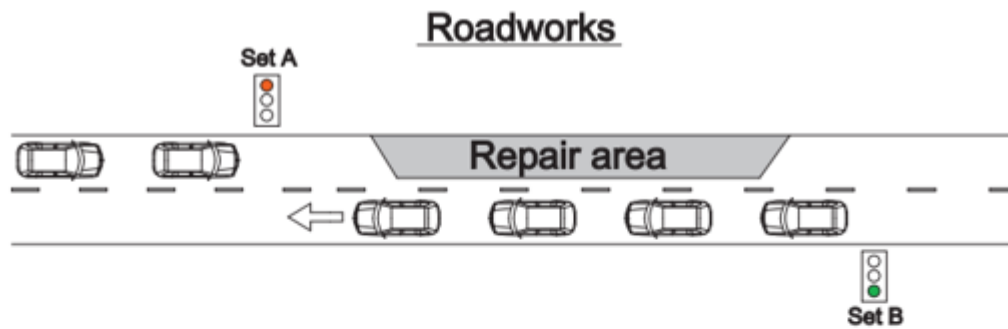


Figure 1

Traffic lights operational sequence									
	Step	1	2	3	4	5	6	7	8
Set A	Red								
	Amber								
	Green								
	Time (seconds)	5	3	10	3	5	3	10	3
Set B	Red								
	Amber								
	Green								

Figure 2

As stated, the requirements are to create a functioning traffic light application that has the above timings for the traffic light and loops through them, while also having an emergency stop and start ability in case something occurs and there is a need for that.

Success criteria

The success criteria for my project can also be found detailed within this, and it becomes clear what needs to be done for the project. The success criteria include:

- Correct operational sequence for the Set A traffic lights
- Correct operational sequence for the Set B traffic lights
- An option to override the operational sequence and change both sets of traffic lights to red, e.g. to stop all traffic on both sides of the roadworks when delivery vehicles need to enter or leave the repair area
- The ability to safely restart the traffic light operational sequence, e.g. after the override has been triggered or at any other time.
- A mobile application that will control the traffic lights remotely.

Project Phasing

Functional and non-functional requirements

Our functional requirements can be seen documented below and elaborated on. These include:

- Inputs – For the inputs of the project we will be taking button presses on the screen as the requirements for the application make it seem like this is the best option for creating a positive and high quality application. Other than this, there will be no functional input requirements for our application.
- Processes – For the processes of the of the project we will be focusing on the internal clock kept within the program which therefore tracks which sequence the sequence should be on. Furthermore, keeping track of whether or not the emergency stop button has been pressed will be another internal process to keep track of as this is what ensures that a correct restart can be applied to the traffic light system.
- Output – Finally, for the output functional requirements, the application will display on the screen what the lights are currently, so the program has to have the traffic lights within it to display this. Additionally, the emergency stop button will output and audio and therefore this will be another output that we have to include into the project.

The non-functional requirements for our project are also detailed below:

- The most important part of the project is to ensure that the program is highly documented and of high quality, meaning that we can maintain and resolve any issues or bugs that occur with the program in the future. For example, this includes comments throughout the code to ensure that everything is well documented and clearly understood, even to someone just starting on the project.
- An additional non-functional requirement of our project is to ensure that the emergency stop maintains functional no matter what might happen. Naturally, keeping the whole application functional is the goal, however it is crucial to recognise the value and importance of the emergency stop function which could hold life saving consequences if it is not maintained, and therefore has its own non-functional requirement to maintain.

Requirement specification documentation

Here are some of the base plans for the project before it is even introduced to Unity.

Pseudocode of the program:

Input Driver Age

If Driver Age is greater than sixteen

Then they are allowed to drive

Else

They are not allowed to drive

Is there an accident?

Input Yes/No

If Yes

Then Emergency Stop

Else

Continue running traffic lights

Input Touch screen

If button program is pressed

 Traffic lights Set A Off/Red (5)

 Traffic lights Set B Off/Red (5)

 Traffic lights Set A Off/Red (3)

 Traffic lights Set B Half/Red&Yellow (3)

 Traffic lights Set A Off/Red (10)

 Traffic lights Set B On/Green (10)

ETC

Else

 Program does not run and traffic lights stay off

Input Age

If Age is greater than sixteen

 Return Yes

Else

 Return No

Module Module1

 Sub Main()

 Dim Age as Integer = Nothing

 Console.WriteLine("What is your age?")

 Age = Console.ReadLine()

 If Age >= "16" Then

 Console.WriteLine("You are old enough to drive")

 Else

 Console.WriteLine("You are not old enough to drive")

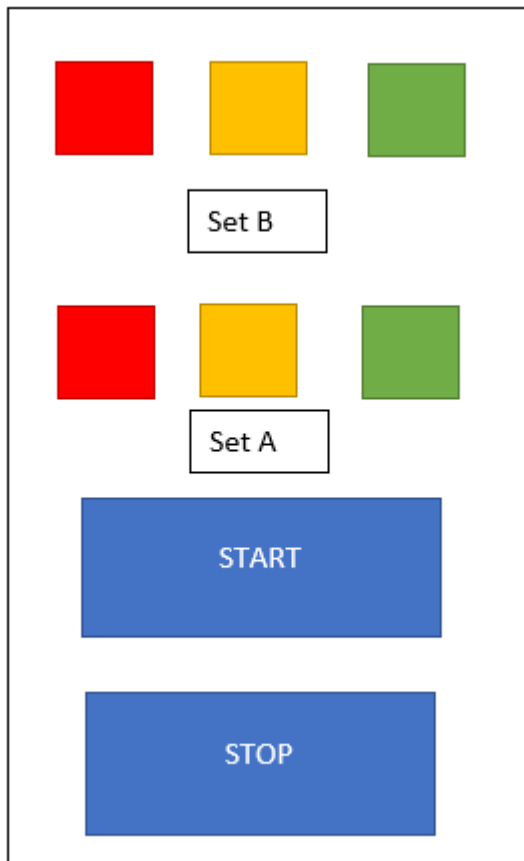
 End If

 Console.ReadLine()

 End Sub

End Module

Diagram of program.



IF BUTTON START PRESSED Then

Start coroutine sequence1

“Or I will establish a timer and when that timer reaches past a certain value, it will run a specific function which sets the light to the right colours.”

IF BUTTON START PRESSED Then

timerLights += Time.deltaTime

IF timerLights > int step1Time Then

Sequence1();

IF timerLights > int step2Time Then

Sequence2();

Ect...

This is some of the base planning for the program and will be reintroduced in the future to allow for more detail.

Typical project management process

There are a number of parts in the management process that have to be dealt with in order to begin properly planning and executing the project. These are detailed and documented below.

Planning project management processes

Resource Plan

The resources that will be involved in my project will be vast in order to ensure that project is high quality and covers all of the necessary goals and therefore is successful. Currently, the primary source of resources will come from human knowledge of Unity, which can be expanded and improved through the use of different sources, such as academic books written about Unity, the Unity documentation, and finally the general internet.

These will all be used efficiently and effectively in order to ensure that the traffic light mobile application is high quality and therefore meets the standards of the client. Furthermore, other resources that could be of use include high quality tech to develop the software as this will ensure that any issues that arise are not technically based, purely human/knowledge based, and therefore can be resolved by taking steps like researching to solve the issue.

Time plan

This is our Gantt chart of the events that we predict will take place in order to create the traffic light software application.

TASKS	06/02/2023	13/02/2023	20/02/2023	27/02/2023	06/03/2023	13/03/2023	20/03/2023	27/03/2023
Requirements planning for app								
Meetings								
Pseudocode of application								
Priority planning								
Risk management								
Application GUI design								
Development of application								
Adjustments for application based on development								
Application Testing								
Review of application								
Evaluation of application								

Through this, we can clearly outline where the priorities, for example the constant meetings weekly about the project and the risk management ongoing to ensure that nothing goes wrong throughout the project. Additionally, we can see that later on in the project the plan is to switch between development and adjusting for the development in one week sprints, just like how the Agile project management methodology functions. This allows the project to run smoothly with clear priorities established and critical paths for a successful outcome to be outlined and maintained, such as constant communication and meetings.

The purpose of risk and issue management

The purpose of the management of risks and issues is to ensure that any and all issues that arise with the project can be dealt with adequately and swiftly as they have been previously prepared to be dealt with and therefore negligible when considering their impact on the project. This means that the project has to correctly identify potential threats to the development of the application and figure out different contingency plans in order to ensure that the project finishes on time.

Management of risks and issues

There are a number of risks and potential issues that must be considered when running a project, and these will be listed below alongside their severity.

Risk	Probability of occurrence	Issue severity	Expected impact on project
Technical Problems	Likely	Negligible to Severe	Technical issues such as bugs or crashes could present themselves during development or after release of the mobile application, leading to a range of consequences from minor inconvenience to significant financial losses depending on the severity of the technical problem.
Poor Project Management	Unlikely	Mild to Severe	Poor project management such as an inability to communicate with the team, which is crucial for the AGILE project management methodology, could result in delays, missed deadlines, and the failure of the project.
Inadequate Testing	Unlikely	Mild to Severe	Testing is one of the ongoing stages during development and if done poorly could result in a broken and bugged final product that was not correctly fixed due to lack of oversight in finding them.
Inadequate Resources	Low	Mild to Severe	Not enough resources such as time, budget, or scope/people, could lead to rushed development, making the product not as high quality as intended.
Security Breaches	Low to Likely	Dangerous to Severe	Security breaches or vulnerabilities are one of the most dangerous parts of development, and if these were to be exploited it could result in a tirade of issues, meaning that the project could possibly be shutdown.
Changing Requirements	Low	Mild to Severe	Changing requirements, such as shifting user needs or evolving technologies, could instigate changes to the project and therefore make it run off schedule
Legal and Regulatory Issues	None	Mild to Severe	Legal and regulatory issues, such as patent disputes or compliance violations, could lead to internal issues in the project and shut it down.

External Dependencies	Medium	Mild to Severe	External dependencies, such as for Unity and GitHub, could leave the project down if something happens to those software and therefore put the project at risk.
Personnel Issues	Low	Mild to Severe	Personnel issues, such as team members leaving or conflicts between team members, could disrupt the development process and lead to delays or subpar work.
Lack of User Adoption	Unlikely	Mild to Severe	Even if the application is well-designed and technically sound, a lack of user adoption could result in low engagement and ultimately, the failure of the project.

Project contingency

As seen previously, the project has extended periods of time to allow for development and testing to make sure that the traffic light application has a successful outcome. Additionally, when we discuss the budget for our project later in the documentation, it will clearly note down how much is set aside for any emergency situations.

Additionally, the final week that has no development and simply review will be a great time for any final bug fixes or issues to be resolved in the scenario that something comes up at the end.

Some additional contingency plans are detailed below.

Risk	Probability	Preparation	Response
Technical problems	Low	In preparation for any technical problems, I have ensured that there will be many backups throughout the project that can be reverted to and relied on in case something occurred.	If there are any technical problems that occur we will ensure that we track back to before they begin and then taking a step by step approach to isolate the problem and resolve it.
Poor project management	Low	In the scenario that the project management is not up to standard, there will be an external third party that comes in and ensures that management is continuing at a high level to ensure the project will be successful.	If the project management becomes a vocalised issue then I will step back and let someone else take over briefly to ensure that the project will run well while I take notes on what I can improve.
Inadequate testing	Unlikely	In preparation for any testing, I will ensure that there is outlined detail on how to perform testing, including documentation and what steps to take when doing the testing to maximise the efficiency.	If the tests that occur are not good enough, using the source control software I will go back to the version that the test was done during, and redo the testing to make sure that test will be done better.
Inadequate resources	Medium	To ensure that the adequate resources for the project, I will ensure that the resources are managed well such as budgeting what is available and rationing	If there is a lack of resources then I will have to ask for more from the stakeholders, as otherwise the

		the resources to make sure that nothing is stretched thin.	project will have to shutdown and this will end the project.
Changing requirements	Likely	To prevent much issues to the project if there are many changes to the requirements, I will ensure that there is space within the time plan to adjust for any changes.	If there are changes in the requirements I will ensure that it is integrated as seamlessly as possible to make sure that the project continues running on schedule.
Personnel Issues	Low to medium	To prevent personnel issues during the project I will ensure that all the members of the team are spoken to and make sure that they are satisfied. Furthermore, I will promote the option of communication with a mediator if something occurs.	If any issues occur I will ensure that the project team members will have an opportunity to talk about it and raise any issues, while also looking for satisfactory solutions together.

Project constraints

The project is to run just shy of two months and supposed to be completed at the end of March. This means that there is a time constraint regarding the schedule of the project that has to be followed in order to create a successful product, the traffic light application, and to complete the project successfully.

Additionally, there will be budget constraints as the funding is limited due to the small nature of the project. This means that the team used to create this application will be smaller than might ordinarily be seen for creating software, however they will be supported adequately to ensure that everything runs smoothly. Furthermore, the start up costs will likely be the most expensive, such as purchasing the technology for the developers to create the traffic light applications, such as hardware *and* software i.e., purchasing GitHub and laptops to work on.

There are little to no sustainability issues for this project as it is small scale and once the project has been closed, the application should run fine with minimal changes needed. If there are any issues, those can be resolved with external contractors, i.e., the developers that made it initially, and of course they will be paid in the future.

Finally, there are no ethical and legal constraints as this project does not involve anything that could be considered ethically ambiguous, and has no possible legal issues, such as the use of private data or information that is collected about customers, and therefore remains free of this issue.

Ongoing monitoring

This will be further documented later, however as the project will be ongoing, there will be many logs and documentation of the project, such as meetings, test plans, errors identified, and improvements to be made. This ensures that the project can stay on track and heading into the right direction, while keeping the stakeholders involved and providing them with clear evidence that the project is progressing forward at a steady pace, ensuring that the project remains funded and operational due to the stakeholder's satisfaction.

Execution and management processes

Time

Throughout the project this section will detail the time spent on sections, such as planning, followed by development and testing.

Date	Work completed	Problems and issues
03/02/23	Hired team and assigned job roles to everyone	Some developers were unable to work when they were required to start so some agencies were contacted to ensure that the project could continue on time.
10/02/23	Finished design of applications and held meetings with shareholders about design	None
15/02/23	Pseudocode of application finished, framework of traffic light application clearly shown	The developers struggled a bit initially to keep to the schedule with the pseudocode but they managed to finish it on time.
22/02/23	Priority planning completed and risk management meetings took place. Shareholders informed of biggest risks and they confirmed we could continue.	None.
27/02/23	Application has begun development alongside finishing the GUI design. Testing starts soon alongside adjustments for any mistakes	There were some personnel issues however they were resolved in mediated meetings between both parties
01/03/23 – 20/03/23	Application has continued development. No large issues arisen and any issues that have arisen were dealt with due to previous planning. Contingency plans of allowing for more time in case of errors helped.	Work continued fine, no reported issues arisen and all resolved.
27/03/23	Application is finished by this point, now only the review of the work	None, stakeholders happy with final product and

	completed and any improvements that have to be made to the project.	development team is finished.
--	---	-------------------------------

Cost

Below are the costs detailed with my project and what the money is used on.

Item	Cost
Electricity	£752
Developer salaries	£75/hour (includes all developers)
Team manager salary	£32.50/hour
Unity	Free
GitHub	£32/month
Other software	£460
Hardware for each developer	£600 x 4
Temp office space	£3200/month

Quality

The quality of this project will be defined by its ability to function and work without issues, i.e., any bugs preventing the software from running. This means that the definition of quality for this application is a high quality intuitive user interface of a traffic light application that operates smoothly without any issues. This can be seen documented in the development of the application, and some snapshots below.

First the code:

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using TMPro;
5  using UnityEngine.UI;
6
7  public class TrafficLights : MonoBehaviour
8  {
9      #region Public Variables
10     public bool emergencyStop = false; //sets emergency stop to public so other scripts can access it and sets to false by default
11     #endregion
12     #region Inspector Variables
13     [SerializeField] private TextMeshProUGUI SetARed;
14     [SerializeField] private TextMeshProUGUI SetBRed;
15     [SerializeField] private TextMeshProUGUI SetAYellow;
16     [SerializeField] private TextMeshProUGUI SetBYellow;
17     [SerializeField] private TextMeshProUGUI SetAGreen;
18     [SerializeField] private TextMeshProUGUI SetBGreen; //the text elements for accessibility can be changed by being assigned in inspector
19     [SerializeField] private Button stopButton; //
20     [SerializeField] private Button startButton; //assigns stop and start button in inspector
21     #endregion
22     #region Private Variables
23     private SpriteRenderer SARed;
24     private SpriteRenderer SBRed;
25     private SpriteRenderer SAYellow;
26     private SpriteRenderer SBYellow;
27     private SpriteRenderer SAGreen;
28     private SpriteRenderer SBGreen; //calls the "sprite" part of the traffic lights so I can change the colour
29     #endregion
30     #region Components
31     private StartButton timerCheck; //
32     private SequenceTimer timerChanges; //calls from other scripts so they can be used here
33     private AudioSource alarm; //component to play audio
34     #endregion

```

This is where the variables were assigned for the project, such as the text on the screen and the lights that change colour.


```
private void Start()
{
    SARed = GameObject.FindGameObjectWithTag("SARed").GetComponent<SpriteRenderer>(); //
    SBRed = GameObject.FindGameObjectWithTag("SBRed").GetComponent<SpriteRenderer>(); //
    SAYellow = GameObject.FindGameObjectWithTag("SAYellow").GetComponent<SpriteRenderer>(); //
    SBYellow = GameObject.FindGameObjectWithTag("SBYellow").GetComponent<SpriteRenderer>(); //
    SAGreen = GameObject.FindGameObjectWithTag("SAGreen").GetComponent<SpriteRenderer>(); //
    SBGreen = GameObject.FindGameObjectWithTag("SBGreen").GetComponent<SpriteRenderer>(); //I moved t
    timerCheck = GameObject.FindGameObjectWithTag("SceneManager").GetComponent<StartButton>(); //Gets
    timerChanges = GameObject.FindGameObjectWithTag("SceneManager").GetComponent<SequenceTimer>(); //
    alarm = GetComponent<AudioSource>(); //lets me play audio as it gets audiosource from current gam
}
```

Here is where all of these variables were initialised and assigned values which could be modified in the script.

```
private void Update()
{
    if (!emergencyStop)
    {
        if (timerCheck.timer > timerChanges.step0TL)
        {
            SequenceStep1();
        }
        if (timerCheck.timer > timerChanges.step1TL)
        {
            SequenceStep2();
        }
        if (timerCheck.timer > timerChanges.step2TL)
        {
            SequenceStep3();
        }
        if (timerCheck.timer > timerChanges.step3TL)
        {
            SequenceStep4();
        }
        if (timerCheck.timer > timerChanges.step4TL) //changes traffic lights dependi
        {
            SequenceStep5();
        }
        if (timerCheck.timer > timerChanges.step5TL)
        {
            SequenceStep6();
        }
        if (timerCheck.timer > timerChanges.step6TL)
        {
            SequenceStep7();
        }
        if (timerCheck.timer > timerChanges.step7TL)
        {
            SequenceStep8();
        }
        if (timerCheck.timer > timerChanges.step8TL)
        {
            timerCheck.timer = 0; //resets to 0 to start loop all over again
        }
    }
}
```

This is where the traffic lights sequences are set to update which change based on the timers current value.

```

public void SetTrue() //function to set emergency stop true. assigned to "emergency stop" button
{
    emergencyStop = true; //changed to camel naming convention
    alarm.Play(); //plays alarm audio clip component, aka alarm noise

    timerCheck.timer = 0;

    SARed.color = Color.red;
    SBRed.color = Color.red;
    SetARed.text = "ON";
    SetBRed.text = "ON";
    SAYellow.color = Color.grey;
    SBYellow.color = Color.grey;
    SetAYellow.text = "OFF";
    SetBYellow.text = "OFF";
    SAGreen.color = Color.grey;
    SBGreen.color = Color.grey;
    SetAGreen.text = "OFF";
    SetBGreen.text = "OFF";

    stopButton.interactable = false; //once this function is called, the stop button stops being interactable
    startButton.interactable = true; //the start button is now interactable again
}

private void SequenceStep1() //a lot of repetition. runs through sequence and changes to what is needed for that sequence.
{
    SARed.color = Color.red;
    SBRed.color = Color.red;
    SetARed.text = "ON";
    SetBRed.text = "ON";

    SAYellow.color = Color.grey;
    SBYellow.color = Color.grey;
    SetAYellow.text = "OFF";
    SetBYellow.text = "OFF";

    SAGreen.color = Color.grey;
    SBGreen.color = Color.grey;
    SetAGreen.text = "OFF";
    SetBGreen.text = "OFF";
}

```

This is where the sequences are defined that change when called in the update function. This refers to the traffic light colours and changes based on the predetermined colours that they should change to in the sequences, such as red, yellow, or green.

```

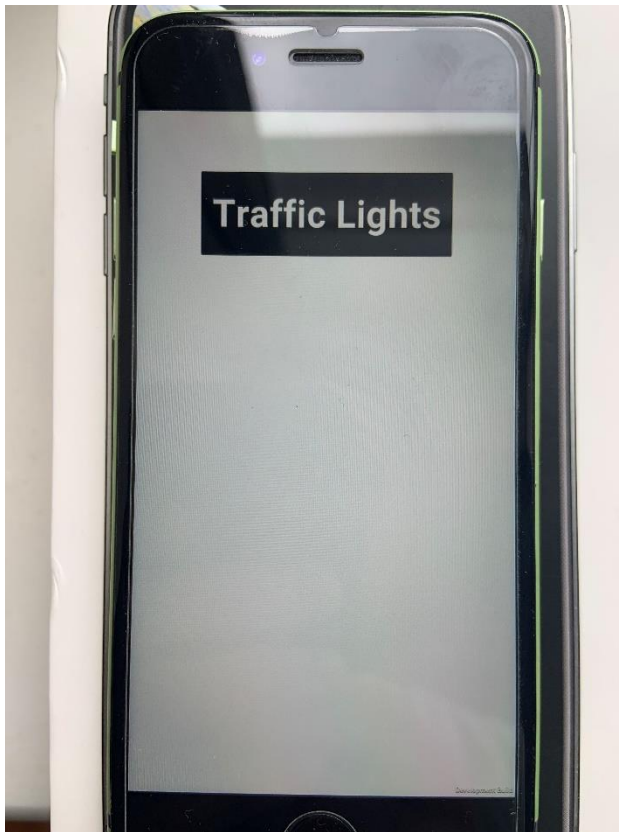
private void SequenceStep2()
{
    SARed.color = Color.red;
    SBRed.color = Color.red;
    SetARed.text = "ON";
    SetBRed.text = "ON";

    SAYellow.color = Color.grey;
    SBYellow.color = Color.yellow;
    SetAYellow.text = "OFF";
    SetBYellow.text = "ON";

    SAGreen.color = Color.grey;
    SBGreen.color = Color.grey;
    SetAGreen.text = "OFF";
    SetBGreen.text = "OFF";
}

```

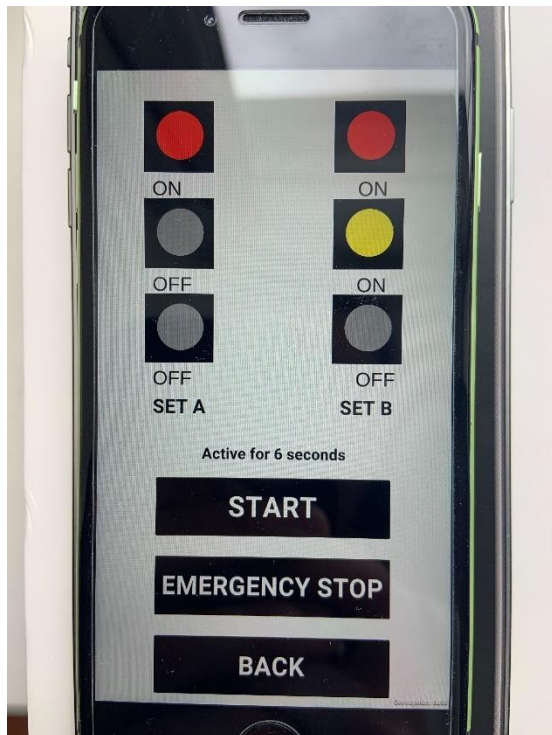
This is simply another example of this to showcase how it changes, and this repeats for all the sequences and colours.



This is the main menu of the application that opens when the application is opened.



Once pressing “traffic light” users are presented with this screen which shows them how to start the program and how to proceed with it.



```
private void SequenceStep2()
{
    SARed.color = Color.red;
    SBRed.color = Color.red;
    SetARed.text = "ON";
    SetBRed.text = "ON";

    SAYellow.color = Color.grey;
    SBYellow.color = Color.yellow;
    SetAYellow.text = "OFF";
    SetBYellow.text = "ON";

    SAGreen.color = Color.grey;
    SBGreen.color = Color.grey;
    SetAGreen.text = "OFF";
    SetBGreen.text = "OFF";
}
```

This is an example of sequence step 2 in which set B yellow traffic light is turned on and can be reviewed how it works in the code above.



```
private void SequenceStep3()
{
    SARed.color = Color.red;
    SBRed.color = Color.grey;
    SetARed.text = "ON";
    SetBRed.text = "OFF";

    SAYellow.color = Color.grey;
    SBYellow.color = Color.grey;
    SetAYellow.text = "OFF";
    SetBYellow.text = "OFF";

    SAGreen.color = Color.grey;
    SBGreen.color = Color.green;
    SetAGreen.text = "OFF";
    SetBGreen.text = "ON";
}
```

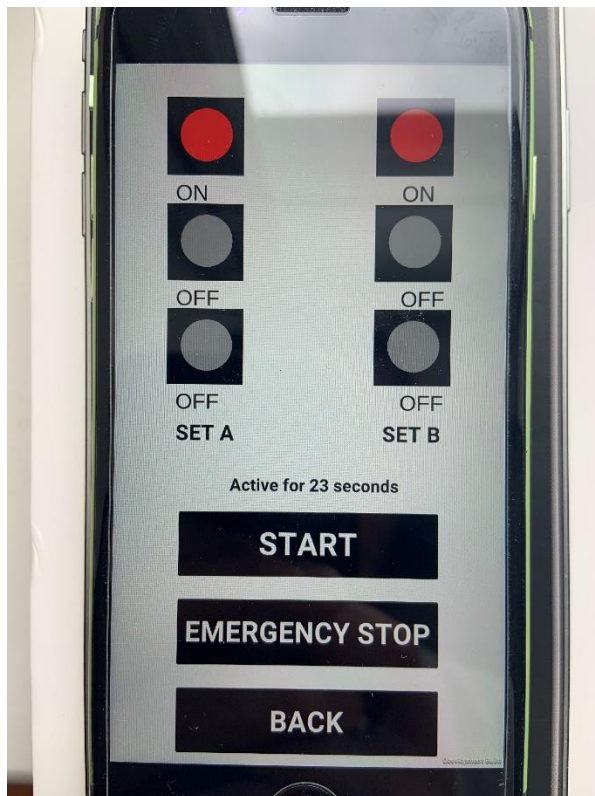
This is a side by side comparison for reference of the third step, detailing what traffic light and text should be enabled/changed and how it changes the program. Here, the traffic light red and yellow turns off for set B, and green is enabled, which can be seen where it says "SBGreen.color = Color.green;". This process, as mentioned previously, repeats for all of the sequences, in total with there being 8 of them.



```
private void SequenceStep4()
{
    SARed.color = Color.red;
    SBRed.color = Color.grey;
    SetARed.text = "ON";
    SetBRed.text = "OFF";

    SAYellow.color = Color.grey;
    SBYellow.color = Color.yellow;
    SetAYellow.text = "OFF";
    SetBYellow.text = "ON";

    SAGreen.color = Color.grey;
    SBGreen.color = Color.grey;
    SetAGreen.text = "OFF";
    SetBGreen.text = "OFF";
}
```

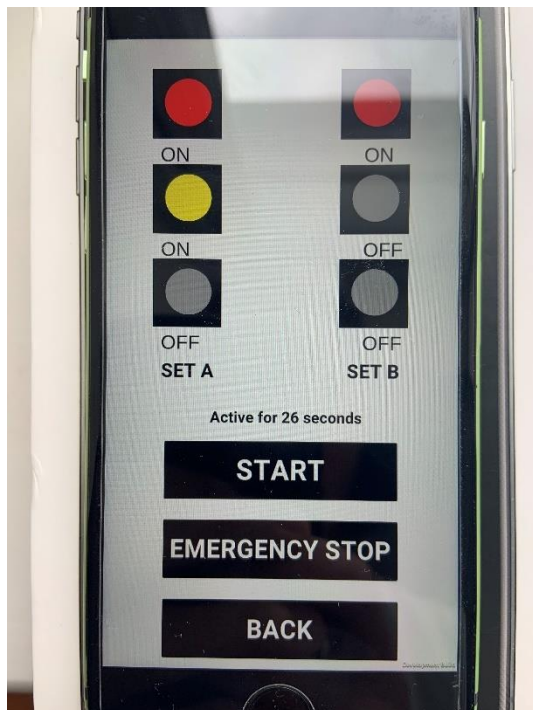



```
private void SequenceStep5()
{
    SARed.color = Color.red;
    SBRed.color = Color.red;
    SetARed.text = "ON";
    SetBRed.text = "ON";

    SAYellow.color = Color.grey;
    SBYellow.color = Color.grey;
    SetAYellow.text = "OFF";
    SetBYellow.text = "OFF";

    SAGreen.color = Color.grey;
    SBGreen.color = Color.grey;
    SetAGreen.text = "OFF";
    SetBGreen.text = "OFF";
}
```

This is continuing the side by side to show the code and the outcome on the application.

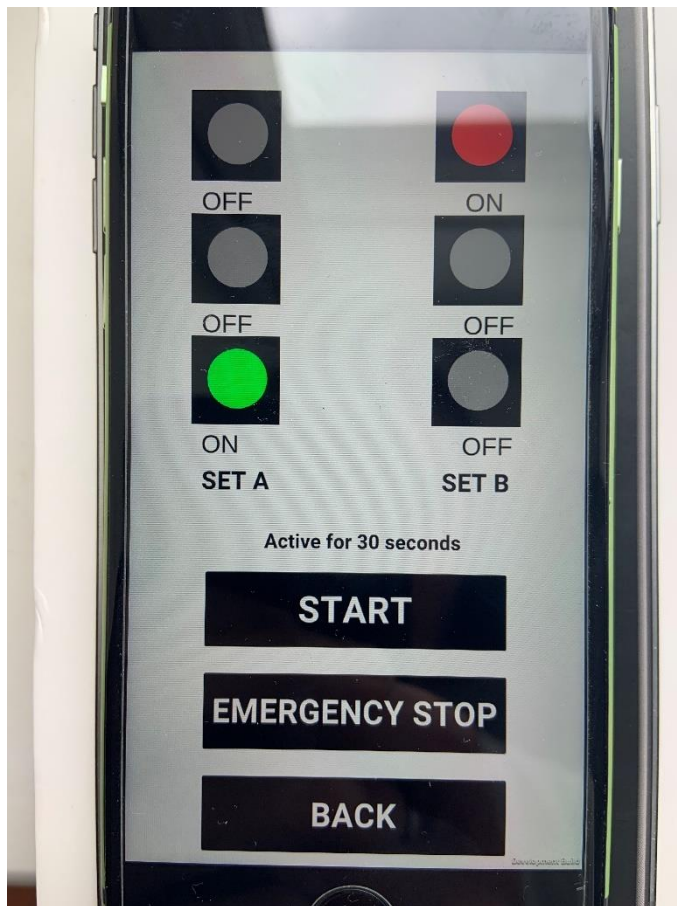


```
private void SequenceStep6()
{
    SARed.color = Color.red;
    SBRed.color = Color.red;
    SetARed.text = "ON";
    SetBRed.text = "ON";

    SAYellow.color = Color.yellow;
    SBYellow.color = Color.grey;
    SetAYellow.text = "ON";
    SetBYellow.text = "OFF";

    SAGreen.color = Color.grey;
    SBGreen.color = Color.grey;
    SetAGreen.text = "OFF";
    SetBGreen.text = "OFF";
}
```

Continuing the side by side.

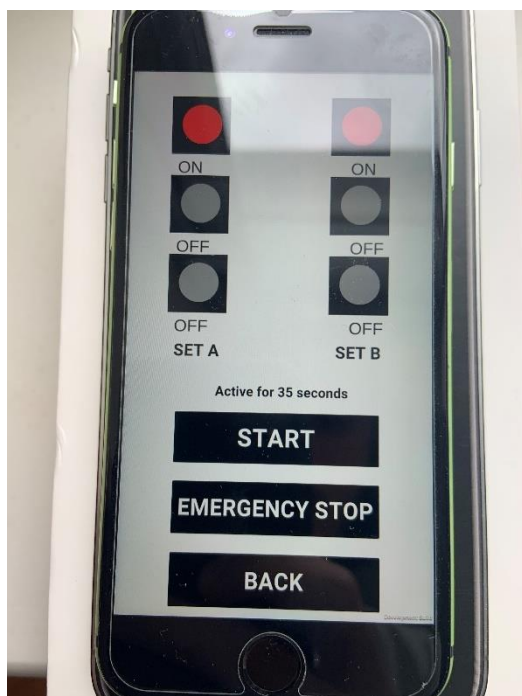


```
private void SequenceStep7()
{
    SARed.color = Color.grey;
    SBRed.color = Color.red;
    SetARed.text = "OFF";
    SetBRed.text = "ON";

    SAYellow.color = Color.grey;
    SBYellow.color = Color.grey;
    SetAYellow.text = "OFF";
    SetBYellow.text = "OFF";

    SAGreen.color = Color.green;
    SBGreen.color = Color.grey;
    SetAGreen.text = "ON";
    SetBGreen.text = "OFF";
}
```

This is the penultimate step in the sequence.



```
private void SequenceStep8()
{
    SARed.color = Color.grey;
    SBRed.color = Color.red;
    SetARed.text = "OFF";
    SetBRed.text = "ON";

    SAYellow.color = Color.yellow;
    SBYellow.color = Color.grey;
    SetAYellow.text = "ON";
    SetBYellow.text = "OFF";

    SAGreen.color = Color.grey;
    SBGreen.color = Color.grey;
    SetAGreen.text = "OFF";
    SetBGreen.text = "OFF";
}
```



















This is the proof of quality from the final product operating on a mobile device. This is the final sequence and from here it loops back to the first sequence.

Test log

Test	Description	Expected Result	Actual Result	Comments
1	The traffic light button will go to the main application with the traffic light buttons there.	Pass	Pass	It loaded quickly and adequately with no issues.
2	The back button will return to the main menu.	Pass	Pass	The back button worked as intended and lets the user switch between the two menus.
3	The emergency stop button will not work prior to the traffic lights running.	Pass	Fail	The emergency stop would make noise before the traffic lights run
4	Fix: The emergency stop button will not work prior to the traffic lights running	Pass	Pass	The noise issue was resolved, and the button is disabled prior to the start button being pressed
5	The start button will start the traffic lights	Pass	Pass	The traffic lights begin to change after pressing the start button
6	The UI timer will change once the start button is pressed	Pass	Pass	The timer changes only after the start button is pressed, working as intended
7	The traffic lights change in sequence when they are supposed	Pass	Pass	The traffic lights change as intended and cycle through the predetermined sequence
8	The emergency stop button works after the start button is pressed, setting both traffic lights to red	Pass	Pass	Successfully sets both traffic lights to red and playing audio.

Monitoring project management processes

Here is a list of the commits made on GitHub by the team over the course of the development of the traffic light application.

	<div>added timer to keep track and working on input options for user to ch... ...ange times copenluu committed</div>	<div>8a59038</div>	<div></div>
	<div>saved changes copenluu committed</div>	<div>f5fca46</div>	<div></div>
	<div>added sequences and fixed ui stuff copenluu committed</div>	<div>be7d437</div>	<div></div>
	<div>updated script to have comments for work copenluu committed</div>	<div>657665e</div>	<div></div>
	<div>first update with very basic outlines copenluu committed</div>	<div>1890d46</div>	<div></div>
	<div>Initial commit copenluu committed</div>	<div>Verified8e30992</div>	<div></div>

And here are the meetings that took place for the project.

Date	Attendance	Agenda	Task Allocation	Due Date	Comments
06/02/23	Stakeholders and I. Afterwards, developers and team manager	Stakeholder meeting to discuss project and the direction they want to go, after we had a meeting about requirements for the app	Developers and team manager set about creating requirements for app	13/02/23	N/A
13/02/23	Developers and team manager	Reviewed created requirements and then assigned pseudocode task, alongside priority planning of tasks	Developers and team leader	20/02/23, 27/03/23	Team completed good work and is on schedule
20/02/23	Developers and team manager	Reviewed pseudocode and checked in on priority planning. Assigned GUI design alongside active development	Developers to do design and development, Team manager to continue priority planning	27/02/23, 06/03/23	Working on time and on schedule, AGILE project management methodology is working well.
27/02/23	Stakeholders, after that the whole team	Informed stake holders of current position in project, received approval to proceed. Developers have finished first stage of development and design of app. Testing, developments and adjustments now being executed in parallel	Developers to test, develop, and adjust the program, team manager to keep them on track	20/03/23	The team is working good and very slightly ahead of schedule, allows for more breathing room later on. Now the team will start alternating between testing, development adjustments, so weekly meetings will simply be updates
06/03/23	Developers and managers	Updated on current position, working on schedule and implementing new features	Continue as previously	20/03/23	All going according to plan, no bugs or issues
13/02/23	Developers and managers	Had some bugs with the software and some developers had called in sick, however the extra time from being ahead of schedule and alongside more agency	Continue as previously	20/03/23	All going well, bugs and issues all resolved.

		developers has led to the project continuing at normal pace			
20/03/23	Team meeting, then stakeholder	Team showed us final product to give to stakeholders, working as intended and no bugs. Provided them with the task to review the application and evaluate it for the future. Had meeting with stakeholders who were happy with final product and accepted it, project soon proceeding to closing phase	Review and evaluate application	27/03/23	The project is almost completed, all parties involved are happy and all is going well.
27/03/23	Team and stakeholders	Final meeting including all parties, the team manager gave a final rundown on the project, including the review and evaluation, stakeholders complimented the work and ended there. Project closing now.	N/A	N/A	Project is closing now. Successful final project and product.

Conclusion

To conclude, while throughout the project there were a number of bugs and issues that arose, they were resolved through the use of the contingency plans and preparing for any issues that might arise. Overall, the project was huge success, the final product was delivered on time, everyone was paid and the work was concluded. I would be happy to use this project as an example to base future projects on as it allowed for a completed final product with minimal issues during the execution and monitoring of the project. This led to a relatively stress free project and an enriching experience for hopefully all involved.

Bibliography

Brescia, E. (2021, 04 22). *Github*. Retrieved from Github blog: <https://github.blog/2021-04-22-environmental-sustainability-github/>

GitHub. (2023, April 17). *GitHub*. Retrieved from GitHub Terms of service: <https://docs.github.com/en/site-policy/github-terms/github-terms-of-service>

Kinsta. (2022, 12 13). *What is github*. Retrieved from Kinta web site: <https://kinsta.com/knowledgebase/what-is-github/>

Unity. (2023, 03 22). *Unity PlasticSCM*. Retrieved from Unity: <https://unity.com/support-services/redeem/plastic-scm>

Unity. (2023, 03 23). *Unity Solutions*. Retrieved from Unity: <https://unity.com/solutions/version-control>

Unreal Engine. (2023, 03 22). *Unreal Engine Documentation*. Retrieved from Unreal Engine Documentation: <https://docs.unrealengine.com/4.27/en-US/Basics/SourceControl>