University of Plymouth

School of engineering, Computing and Mathematics

COMP300 – Final Report

Hacker Adversary

Word Count: 9131

Key Words: Game, Project, HTML, API, Virus, User, Tello, GitHub, Waterfall, Agile

Code Repository: https://github.com/DigSwine/Final-Year-Proj

Acknowledgments

Firstly, I would like to thank Dr Kimberly Tam for overseeing the creation of the project and supporting my freedom to take approaches that might not be the best but work for the project.

Secondly, I would also like to thank the rest of the lecturers at Plymouth University for helping with difficulties throughout my university life, some lesson will remain embedded for years to come.

Thirdly, thank you to my friends and family for the support that they have provided throughout my time at university, even though COVID-19 and social distancing, and especially to those who kept me on track with work.

Finally, to my student support who helped kick me in the right direction when they had to and keeping motivated to keep a steady workflow.

Contents

Abstract

The primary reason behind the game is to provide an alternative method of education to aid the learning of students at the University of Plymouth. This is accomplished by providing a less textbook version to teach key words and allow students to gain an insight to how fast a virus can spread throughout the world by giving an up-to-date count of all computer systems that are currently in use across the globe.

The report will follow a basic structure, to allow the reader to find their way from one section to another without the reader becoming lost by lack of flow. What will be covered will be an introduction to introduce the concept and followed by an in-depth analysis to delve deeper into the reasoning behind the project. The introduction will be followed by a discussion on the method that has been taken to produce a plan of creation and the reasoning behind the tools that had been selected to carry out the proposed plan. This will be proceeded by an investigation into the Legal, Social, Ethical, and Professional Issues that have been faced when creating the project and the steps that have been taken to avoid breaking any regulations that could be found. After this the report will then talk about the establishment of the project, along with examining and analysing the difficulties and challenges the difficulties that have been found during the creation of the game and when using the supporting software. This will then be followed by the pre/post-game information and uses that the game provides. After the game has been talked about the report will then be provided on the achievement of objectives that had been set out in the planning phase of the project. With the creation kept in mind, the report shall then investigate future developments and improvements that the system will undergo when developed further. Finally, the

report will conclude with a summary of the development and reflection on the project and provide concise details of the report.

The creation of the project will follow the method selected later in project planning, this will help to provide the best results for the project’s outcome and useability that can be achieved. The method will have to incorporate dynamic developments that will be used to avoid issues that might arise when creating the game, this will have to be investigated further during the report.

Introduction

The project is a game that will be designed to aid computer security students at the University of Plymouth, this game was originally going to be called “Virus Inc.” but this was later changed to “Technical Adversary”. The reason for the name change is that Virus Inc was too like another game: “Plague Inc”. Therefore, a new name had to be created, that represented the game for what it is, a real time strategy game based on computer viruses.

The game was inspired by “Plague Inc”, so some similarities were bound to be expected, but this was to be offset by the games supporting website to ensure that the project was unique from similar games that can be found on the marketplace. The supporting website will have a sleek retro feel using being simple colours and easy to use controls.

The game is based on a simple process: build a virus then hack the world. There is no point system or in game upgrades as in real virus making you cannot upgrade the virus that has been created while it is infecting computers across the globe. The virus builder uses a simple true or false to determine the difficulty of the game, each continent has a difficulty for the virus to hack that is based on a percentage determined by the difficulty. While the virus spreads there is a chance that a virus will be removed from a few computers. The rate of infection is set depending on the difficulty, starting at where the difficulty is counted as hard, the computer spreads through large corporations' emails at 40 per day, medium spreads through companies and small businesses at 70 emails per day and easy spreads through every person at each continent at 100 emails per day.

The game was originally only the interactive map and click a continent to start at and that was it, until all computers had been hacked and a notification said that the user had won. The difficulty selection includes a lot of features which makes the game harder, but these are not interactable by the user and unfortunately this is the biggest downfall of the game, fundamentally the game works but there is very little interaction.

The report follows the same flow as listed in the Abstract: Introduction, Discussion on planning, creation of the project, pre/post-game and finally an evaluation on the game. The report will have an appendix for all images and where applicable the appendix will be referenced intext throughout the project, demonstrating what is being discussed. Tables will be included in the main body of text to provide a clear example of what had been tried to overcome issues.

Main Aims of the project

The project dose not have any laid-out goals, as a result some personal goals had to be created, this will allow the project to have direction for design and creation. The goals that had been set for the project are:

- Understand what it means for a website to support a game.
- Attempt new techniques to overcome issues that arise in the creation of the project and see what works best.
- Develop a program that works smoothly with as little issues as possible.
- Set out a clear work order and follow it until the project's completion.

Method of Approach

When I was starting to write out a plan on pen and paper, the overall thought of how I was going to build this project started to arise and this led to me thinking about getting a method of approach written, to then determine the best way that the project can be built. When examining the most appropriate methodology, there are two project management options that could be selected from: Waterfall and Agile (Santos J, 2020a).

Waterfall

The waterfall method takes the project into consideration before the work can begin, this means that the workload is strictly organised and progressing onto a new segment can only begin once the current segment is complete, this is due to the waterfalls linear approach, and segmentation of the project. Performing each task this way allows a single project to be completed with swift resolve as its layout means that the focus is purely on the product delivery. The waterfall method stipulates that all requirements for the project have been set out before the project can be conducted, this means that once the project has started the scope cannot be altered and therefore there can be no ‘gold plating’ by adding unnecessary code (Santos J, 2020b). Once the project has been completed entirely then testing can be conducted and according to the waterfall approach this is conducted by test teams who do not get involved in the requirements of the build. Finally, typically the waterfall method requires a project manager to oversee each member is performing their correct task, this however If I were to have chosen waterfall this aspect would be overlooked in my project as this is a solo project (Santos J, 2020c).

Agile

The agile approach differs from the waterfall greatly as this method is flexible with the tasks that can be completed, and instead of the project being split into segments like the waterfall method, agile allows the user to report sprints to prove that work is being completed. This means that the overall project can be described as being made up of lots of small sprints which are all stitched together to create the overall project (Santos J, 2020d). This approach is not focused on the project but more focused on customer satisfaction, this means that testing can be done often during the project to ensure that the user is fully happy with the product. As a result of doing this the customers can provide feedback and then the requirements can be changed each day to ensure that the useability is to the customers satisfaction. This method denotes that each the entire design team is responsible for managing the project without a project manager (Santos J, 2020e).

Method Decision

The method that has been selected has been considered carefully, the agile method will be the method that will be employed for the project will allow me to manage the project and deal with changes to the requirements as the project progresses, also this will allow me to produce sprint reports and provide them to my supervisor so they can see that work is being conducted on the project. Also, the way that I plan to test the system is once each page has been completed to ensure that someone who does not have any knowledge tests each page and provides feedback allowing requirements to be changed as the feedback has been provided. To ensure that progress is not lost, a virtual noticeboard will be used to keep organisation of the project management, this will also have the ability to use labels so that a record can be kept of what tasks issues had been found and they can be recorded later on in the project.

Legal, Social, Ethical, and Professional Issues

During the planning process the original plan had to be that I create all assets for the project myself this means that there would be no issue of any copyright arising as there would not have been any external assets to be used. However, for the best results during the process the plan slightly changed to allow a few external assets such as the navigation bar, which was used in a w3school template and then changed to fit the website’s design, this will be discussed further later in this section.

The game uses a small concept of Social Engineering when selecting the difficulty of the game this is a result of asking members of staff at the University of Plymouth if providing exact code for the selected virus would be a good idea or not. The answer that I had received would be ‘it might be best to “gamify” the code and provide a nonworking code that falls into social expectations of what a virus would look like’. This was taken literally, therefore an image of what a computer virus code looked like was used as a reference and the code that is provided changes depending on the code options that have been selected.

As the game takes all computer units that have been sold for each continent, and the technical advancement of the continent to produce a level of each continent’s general security to provide the game with a level of difficulty. This game does not have any ethical issues, as a result of this and that the game does not involve any virtual contact with humans therefore there is no risk of any psychological harm as it was only tested by other students at the Plymouth University.

As the game does not furnish any information that is provided to any third party, therefore there are low professional issues that can arise within the project this is due to the lack of need for confidentiality, anonymity or consent. However, there is one section of code which does have to be properly cited, the navigation bar and a rough template for the project, these had been used and as originally from w3schools. Additionally, any other code that has been used has been cited, this can be found in the “References” documentation found in the Git repository. There is also another game on the marketplace this was mentioned in the abstract and the game is called

"Plague Inc", mindful of the need to embrace all ethical and date protection requirements and to ensure to ensure that there could be no legal ramifications from the company, scrutiny and examination of both games for similarities were undertaken followed by appropriated for both games similarities and then appropriated steps were taken to ensure that my project was not in breach of any copyright laws. This was achieved by the one-time virus creation where as "Plague Inc" has in game upgrades and bonuses which my project does not provide.

Planning and design

Planning

Carefully planning the process that the game will be create is key to a successful game and website design. To ensure that the methodology that has been selected is also kept in mind will also help the project to be created smoothly and additionally, that the development does not 'run out of steam' and become abandoned. With Covid-19 having a large impact on how teaching and learning occurs affecting all student's ability to go onto university grounds meant that the location of the project might have changed over the course of the year. Therefore, it was important to have a trusted method of keeping the project well organised and version controlled to ensure that the code did not become redundant. The best method of transferring data without risk of corruption or data loss would be to use GitHub to manage the project files. Trello was used to ensure that the project is managed in an orderly fashion, Trello also helps to maintain the agile methodology as the tasks can be selected from one list and moved as it is being completed, this does not rely on any order as each card can be moved in any order.

Trello

Trello was designed to be a team management tool but quickly expanded to be used vastly by project managers to keep track of their workflow by using the agile methodology as described above, this means that Trello is a perfect fit for this project to use as the main project management software.

When talking about Trello, the board consists of ‘cards’ and ‘lists’, what is meant by these terms is that; a ‘list’ is exactly what is expected as it is a list that ‘cards’ get put on to ensure that the ‘cards’ are managed into different lists to track workflow. As for ‘cards’ they are essentially tasks that must be completed, so if tasks are split into three different lists, then it can be clearly seen what tasks are to be done, tasks that are currently being worked on and then tasks that have been finished. The use of lists can be further expanded to then state what needs to be added to each page of the website, this can then be attached to the page that needs to be done so that all tasks that each page performs can be clearly seen and worked on as the project progresses.

The use of a Trello board was something that I was quite familiar with as this had been used in other group projects previously, this meant that setting up different lists and cards was easy. The use of Trello as a project management tool helped ensure that each page of the project was designed and then created with only coding errors and not useability errors, this means that when the testers tested the project the reports only showed improvements with the overall game play and small grammatical issues that the students had pointed out. Trello also allowed me to take a step back from the project and look at the overall scope to properly understand the quantity of work that will have to be put into the project for it to work smoothly.

GitHub

GitHub was designed for large teams to work on projects and collaborate without having a separate manager imputing all the code that had been edited into the working files, this website works as a hosting platform for version control, allowing teams that are spread across the globe to work without damaging a project. As a result of this GitHub worked perfectly for version control for my project as it is uncertain where my work will mostly be preformed as Covid-19 has had a large affect on the productivity available inside of university grounds.

GitHub does have its own terminology, when talking about 'Repositories' this is where the data for the project is saved, this can be code sheets, documentation and other information that the project requires or is useful. 'Cloned' is when the repository is copied from the website and put into a folder on the local system for use, this is like copy and paste that should only be done once to ensure that the link has been created with the folder and the website, meaning that there is no need to delete the work once it has been edited. 'Pushed' is where the data that has been edited is sent to the website this is essentially saving the work that has been done allowing anyone else who has access to the repository to gain access to the new data. Finally, the last term of 'Pulled' this is where the data is copied from the website and saved into the folder that the clone had originally created, this will automatically overwrite any code that had been changed since the last time the data had ben pulled.

The use of GitHub helped with transferring data to and from University when I eventually was able to enter the computing rooms at the end of the academic year. This was a huge success as the files that had been created at home did not have to take up unnecessary space on a USB device and all code that had been done was readily available at home once I had returned from working at University. This website was helpful as I had expected it to be, also the use of this website means that the system is backed up in case there is a corruption with a file for the website as this causes massive issues.

Tools to be used

When designing the project, it was clear that this game's supporting website would have to take a few things into consideration, and this led to me thinking about the projects creation and how specific things will function. Due to the nature of the project the game will be hosted on a website, and this means that without any form of login system that anyone could come onto the website and start to understand how hacking works, but this was in direct opposition to the aim of the game that states that it is to teach users from Plymouth University, therefore a login system will have to worked into the project.

As a result of needing a login page the best approach that could be used was a database, this was planned to be hosted on MySQL as this is a straightforward database system which works nicely with websites and can easily send data to and from the database and the website by using API's.

As for the actual website, this was created by using Visual Studio 2019 as this system is easy to use and is a system that I am familiar with, also the system had been used previously for other projects so the knowledge of setting up the program was already within the computer. As for the game, the main game works on an interactive map, so this would need to be created by using photo editing software, the one that was chosen was paint, and this took an exceptional amount of time to do. The world map then had to be worked into the website and a collision box was required to be drawn around each continent.

Design

Before the game and supporting website could be designed it was important to make sure that the game followed the five most critical components that make learning games a success, this is to ensure that the game achieves the main aim of being a learning aid for teachers at the University of Plymouth. The five main components are: Goals and objectives, Rules/Instructions, Interaction, Conflict, Outcomes and feedback. By ensuring that the game has all five of these the game will then be a successful learning game.

Goals and objectives

The goal for the game itself had to be something to do with seeing how the created virus interacted with the world, and after a bit of discussion and evaluation with peers it had been decided that the main goal should be to hack either a continent or the world, this was later decided that hacking a continent could be added as a mini game rather than the main game as it would only take a portion of the time to complete than the main game, and as a result of this it was noted that if the main game is finished then this will be added on to develop the game further.

Rules/Instructions

For rules, the game is strict to ensure that the virus once developed cannot be changed, so the rules of the game had been designed to be simple; hack the world to successfully infect each computer unit that had been sold in each continent. Although these rules have been made, they are not provided to the user, as the game follows its own strict rules to ensure that the game runs smoothly and the only way to bypass these rules is to hack the game and change the code to ensure that the virus created can beat the game, this in turn would make the virus successfully hack the world. This rule has purely been made to be broken by users so that the game can be won, once this has happened the game can be considered a success.

In considering the instructions for the game it was decided for the game was decided that they would be on the opening "about" page of the game, and the instructions are: "Create a virus on the virus and select Confirm once the virus had been created, then go to the game page and select what continents you want to start the virus in, then watch as the world's networks start to slowly become infected". This instruction had been designed be talking with game design students at the University of Plymouth to ensure that they are clear and user friendly but most importantly they give clear instructions of how to play the game.

Interaction

The game itself does not have much interaction, but the interaction that the game does have is mainly interaction when designing the virus by providing the user with keys words that could come up in examinations. The game can have multiple run throughs to then find out the best way to create the virus and find the best starting location, this means that the game has a lot of playability meaning that the user will learn from repeatedly running different simulations.

This area had originally been planned to have upgrades in game to give the user more features to explore terminology and effects of different viruses, but as previously mentioned it would have been too like other games on the market and doing the game this way meant that the game stood apart from these similar games.

Conflict

In terms of conflict there is not any visible conflict planned due to the game's nature however there will be a percentage that each day that passes the virus will spread, this will then be split into three categories; spread the infection, spread to a new continent or cure some computers. This is a passive conflict and if this project could be developed further, I do plan to add this feature to give the user more interaction with the game. However, getting the game to run is my main goal and this will be added at the very end of the project with time permitting.

Outcomes and Feedback

Where outcomes are concerned the outcome for the game is that it is simply a win or loss game, at the end of the game the user is told if they successfully hacked the world or if after 2 years the virus did not successfully infect each computer and was therefore beaten by the world's network security.

The game does not also provide any feedback to the user, as this is a game purely to teach the user key terminology that could appear in examinations later in their university life. However, if time permits this will be added into the game, as a tips page and possible have a little bit of detail about the type of virus that the user had created and why it was beaten or why it won, it could offer some advice to the user to try other methods of infection instead (Gutierrez, 2012).

Creation, Difficulties and Overcoming

The procedure for deciding what the best first step should be was simple as it was a simply decided by the inability to test the game if it had started to be created, as there was no supporting website to put the game on, therefore it was decided that creating the supporting website would be the first step.

The creation of the webpage was straight forward at first, by creating a website folder and then adding three webpages and then using these pages to ensure that each element that was listed in Trello had the page that had been designed, this was then followed by naming each page.

Template

At this point the real work on the project could begin, so I had a look on w3schools for a template that suited the project, this took a considerable amount of time as this had to be carefully considered and evaluated. Once a decision had been made and the code was transferred and a proper citation had been made, the code was edited to remove anything that was unneeded or redundant and the navigation bar that was provided was changed to suit my project's needs, at this point the buttons on the navigation bar did not work as I was going to add each page as it was about to start to be worked on, so I could get around the page and see it working.

About Page

Once the template had been sorted out, I started to work on the about page as this page should only be a few instructions for the actual game but at this point I had not built the game so I was unaware of the rules so place holder information had been typed so that I could see how the page will look. Eventually once the login system had been made and along with the other pages of the game had been made this was edited to be the actual information that had to be on this page.

Login system

Next, I referred to my planning stage where I talked about needing a login page, so another page was added, and the template code added, this is where things started to slow down, as this page required an API to send data from the two boxes and compare the boxes against the database. Once this page had been sorted, I then worked on a personal information page, to then ensure that the user could change any information they wish about their account. This then led to the discovery of an issue, when the user logged in, the API would send data to and from the database to allow the user to log in but once this had been done and the user logged in the data appeared as if it had been dropped. To overcome this issue, I tried to send the data across to the new page, but this did not work and reviled the user’s data in the URL, this meant that it was not secure and therefore unsafe to use. After a lot of trying different techniques, I ended up putting the data into a session variable and using this to store the data and then transfer the data into the relevant textbox, this however this also did not work, so I had a look into the session variable from both the website and the API, as it turns out the session variable was different for both of these, so the code was changed to allow both the API and website to be on the same session variable and this worked, allowing the data to be displayed onto the users information page.

Once the initial login page had been made, I then needed a way for new users to be added to the database, this was simple as it was just inserting what is provided into the database. Code injection is often an issue at this point of logging into a system so a twenty-letter word limit was added to minimise the risk of code injection, this was then supported by only allowing "@students.plymouth.ac.uk" at the end of the email address, as for names only letters can be added to ensure that code is not added into the name place with a proper email address. Once the protection from code injection had been dealt with the code was tested and the new user was logged in successfully and viewed on the database.

There is one issue that I did find with the login system, that has yet to be done, the system does not have a delete user option, this was originally a design concept before the virus was gamified so that the users who used the game a lot could be identified in case they had used the website to build a virus and then use it in the real world. This is planned to be changed when the game has been finished but this is on a time restriction as the end of the project deadline is coming up soon. The main reason why the user's logins can now be deleted is due to the virus code being gamified and poses no risk if the code was taken and set upon the world.

Navigation bar

Once this issue had been found I decided to work on something less stressful, making the website's navigation bar have two states; logged out, and logged in. In the state of logged out only two pages would appear, "About" and "Login". Whereas the logged in state would have four pages to navigate to, "Logout", "Information", "Virus" and "Game". This was surprisingly easy to do as it only took editing the navigation bar itself, so the code was not needed on every page, or the addition of another navigation bar. Once this had been done, I decided to reduce the length of each code by removing the navigation bar from each page and adding it to its own page that could then be included into each webpage, which was a huge success, and when I was testing this feature, it was so easy to use and made sense, when testers used this system too, they also like the fact the navigation bar changed and how easy it was to use.

Game

This next step was difficult to decide, but I ended up coding the main game page, this started with defining a game area, this was done with HTML instead of a canvas, this was an issue that had not been noticed until the game had already been made. Once the area had been made the next step was to work on a world map, to do this I used paint to create a general map outlining each country that could be remembered, this was done using a trace tool allowing me to gain an accurate map of countries, this meant that if I were to start the game aimed at countries then this could be done. The next step was to then sort out the on-mouse hover code to then make the mouse interact with the game, this was done with multiple tools, firstly a decision had been made to keep it unique from other games by making the targets continents instead of countries.

The map for this was done in paint as this only required recolouring the map that had been created. The final step of creating the interactive map required the map to then react to the location of the mouse was to create a map that the mouse would interact with, this is kind of like a collision box. This was done by firstly mapping the location of the mouse, and was accomplished with code that is now redundant but had been saved on GitHub inside the Redundant Code this was called “mouse_finder”. What this does is track where the mouse was on click, the output of this was then used in the code for the collision box, and this is where an issue was found, the game changed the image URL for where the mouse was, allowing the mouse to freely move without clicking and the map would then change to what was being hovered over.

The next step of creating the game was to then make the mouse hovering display information for each continent, the data that is displayed is the name of the continent, the number of units of computers sold, the infected computers and the healthy computers that are left to be infected. This was done quickly after a little bit of research into the number of units sold in each country of each continent.

Finally, the game then had to be working so a method of infection was then created causing a mouse click to then infect 1000 computers at once, and this was successfully displayed on the continent’s information page. The next step was to then make the game run, to do this I used a day counter which starts on the current day and counts up each day an action that is performed every five seconds, this then is used as the system running clock, so that the infection can spread. This way that spreading is done depends on the type of infection mechanism that had been created when determining the virus as a result see the virus section below for more details. Once the virus had been made to be able to infect one country it was then decided that the virus needed to spread to other countries, so the method of spread had been changed to be determined by a ‘roll of the dice’ making a random number between 1 and 100 to create a percentage, the lowest rolls remove a few viruses, high rolls create new viruses and medium rolls infect a new country with 1000 computer infected. The country that is infected is not determined by any rules as the internet is everywhere and anyone can become infected from emails or embedded code. It is clear that the first infection that is made creates 1000 infections this had been a decision made to firstly create enough infections that the game would be able to keep running without destroying all the infections before the game has started, but also the numeric number in some countries is so large that starting with 100 computers infected will make the game last too long to keep a person’s attention and this would then cause the game to be boring and what is learnt will be forgotten.

Virus

The virus page was made by adding three drop boxes to determine the build of the virus, this is done by presenting the drop boxes to the user and when the infection mechanism has been selected, a gamified version of this will then appear inside the list of where the “virus” is displayed. The next option for the virus builds is the trigger when this is also selected the code for this also appears in the gamified version of the code.

Infection mechanisms

Infection mechanism are how the virus spreads from one computer to another, for the game there had been three infection mechanisms that had been selected, these are: emails, embedded code, or software bugs. These three types of infections have been chosen as they are often the most common types of virus infection.

Infected emails are often sent to unsuspecting addresses with the hope of catching a few people to open a link to an external website or download an infected file as soon as this happens then the virus is free to infect and spread throughout the system and eventually evolve, create new instances and sending its own emails to new targets. An email can be all it takes as the virus can be embedded into the email or be hidden within a phishing email (Bowen C, 2020). This is what the infection mechanism of 'emails' means. This method of infection has been deemed as medium due to its wide net and fast acting method of infection.

Embedded viruses are often used and are considered as a form of malware, that infects a webpage this is also referred to as an iframe virus and works by searching websites for their index.php page and then implanting its virus into the code of the HTML that is found within the page. Normally the virus can be found at the front of the webpage to ensure that it is loaded first making it the priority for the computer to infect the system. This method does have a high success rate but is also slow as it takes time for people to enter the webpage for themselves, this method for the virus has been deemed as medium due to the time to infect the computer.

The final type of infection mechanism that is available to the user is, self-upload, the method of upload would be exhausting for the user and would require a team for each continent, this method would require hacking into every computer in the world and slowly putting the file into the computer, this method is definitely the one that has a low security penetration rate purely due to the amount of time it would take for the infection to be placed in every computer.

Triggers

A trigger is also referred to as a logic bomb, and this is a bit of code that is used to cause harm to a network when specific conditions have been met (Fitzgibbons L, 2019). In terms of the game there is one embedded into the virus as a default, and this is where a date is selected in two years and the virus destroys itself, as a result of the game being lost. For the actual playability of the game this section does nothing as there is no gain given from this, but it is that students learn that a virus has three key parts, so this is only to determine the difficulty of the game when selected with the other two options.

Time and date are often used as a trigger to ensure that the virus does not cause any issues allowing it to spread unnoticed, this is until the trigger time and date has been reached and every infected computer all feel the effect of the virus at the same time. The game is simply to infect every computer, the date could be set for 1 year and 364 days this means that the outcome of each area becoming infected will need to be changed saying that each country has been infected and when the date is reached then there will be an outburst of information saying each country has fallen into darkness, this presented a little bit of difficulty as often it made the game crash, so I had to implement a 3 second clock that stored the information and then presented the data as it was filtered through, to give the game enough time to process the information. As a result, this method has a high security penetration rate.

On click, as soon as the virus is ready to go, and the user clicks the mouse to wake the computer, this in turn means the virus then wakes up and attacks the computer while the computer is still booting up, this can cause slow booting and serious issues for the end user as all sorts of information can be damaged or stolen, as the virus is working while the computer is still booting this means that the virus has more time to do damage. However, this could also be an immediate response and the virus quickly noticed. As a result, this method has a medium security penetration rate.

Immediate, this trigger is less of a trigger and is simply as soon as it is uploaded it then starts to preform any task that it has been given this will almost definitely be noticed by the computer and the end user for the computer. This method is not very reliable, but it does mean that It could spread faster if that is the only task for it to keep the infection spreading. This method has been giving the low security penetration rate as this method can be quickly detected and can be cleaned out of a computer quickly.

Payload

This final component of the computer virus is where the user decides the malicious activity of the virus they are creating, this will be used to determine how noticeable the virus is and how much damage it causes, the more powerful the payload the more damage it causes. This however is not always the case as some viruses do not carry a payload, where as some carry data destruction, offensive messages or theft/eavesdropping (Techopedia, 2021).

Nothing, this option has been given to the user purely so that the virus can infect the worlds networks allowing the user future access through backdoors into every system. This method has been given the high penetration rate this method will go undetected by any end user unless they actively perform a full computer security scan and ensure there is no viruses for themselves.

Offensive messages, this has been added so that the user knows not all payloads have to cause actual harm some can just be to repeatedly insult someone, this method is essentially a spam messenger designed to be annoying to the end user, this of course is noticed and can be detected by any system like ad blockers. This payload has been given the low penetration rate as this is here to just annoy the end user and whoever made the virus gains no benefit from this method at all.

Data destruction, this payload is often used as it can create backdoors into a system and can often be used to seriously disrupt large corporations from carrying out work, as a result of the impact the large companies feel from this method this have been given the medium penetration rate as this is often used to cause harm but not benefit the creator directly.

The final type of payload that is available for the user is theft/eavesdropping, this has been given both titles as both of these are very similar. Theft is where the virus accesses all data on the computer and stat is saved online, and then sends them back to the creator, for them to use this could be usernames and passwords, or bank details, or the end user’s personal details – name, address, etc. As mentioned, this is similar to eavesdropping as this is where a virus hijacks the computers microphone, keyboard, and mouse and then records what has been said, typed or clicked on, this method can be very useful and valuable to a lot of companies. As a result of this this method has been to have a high penetration rate as it does benefit the user massively.

Pregame and Postgame

The pregame to this game is that, when the user opens the website, they are immediately met with the instruction and the rule. This has been done this way to make the website as concise as possible, the next part of the website the user is presented with that is part of the pregame is the creation of the virus, this part as mentioned above is broken down into three parts and is used to determine the difficulty of the game, this has been done with a lot of IF code's this method may not be the easiest method of doing this but it was a technique that worked for the purpose of the project and does not produce much lag so the user feels no effect of this. As the options of for the virus is being made the gamified code for the selected options appear inside of a text box below them and this then provides a gamified virus. This is then the end of the pregame as the game then uses the virus to be used on the game.

The post-game is simply a notification saying the game had been beaten with a congratulations to the user for creating a working virus. However, if the user is beaten then the notification is different and offers tips that the method of spreading might be better if the computer does it, and that more dangerous payloads will be spotted faster. When the 'ok' button on these has been pressed then the user is sent to the user information page, and the virus and game information is then cleared allowing the user to then create a new virus and run the game again.

End of Project Report

Taking a small break from this project to focus on other projects allowed me to come back into this with fresh eyes and look at the work that had been done. This allowed me to see the work for what it is and find any flaws that had been overlooked when the game was in development, and some of these issues had been fixed as a result of this.

There is clearly a large amount of work gone into user friendliness, when I was looking at the website the amount of work that could be seen was apparent and the login system worked surprisingly well with next to no time delay for the data to pass from the website to the database and back. This meant that the website felt like it was made with a high-quality standard in mind. When testing the game, it was clear that it did take a bit of time and the lack of interaction did not help pass the time, this would be something to be added if I have time before the final submission but only if all other aspect of the game could be added first. Overall, however the game felt sleek and performed really well, there a few minor changes that could be edited but as they were minor this was not a main concern for completing the project.

When looking at the useability an expectation that the code behind all the pages was vast but surprisingly all the pages had a lack of code as most of it was reusable and had been used in such a way that there was very little code that was not used by each system, this meant that there was only one or two lines that did not get used once they had been performed. There were only two pages that had more than 300 lines of code and these were the virus page and the game page, the main reason for the virus page being large was due to the IF statement that determines the gamified code that is produced. As for the game this was vastly taken up by the function to make the virus spread as this had to do multiple checks to ensure that there were no issues or too many computers been infected for each country to prevent a minus number of computers left to infect.

The objects of the game were vastly met, as the first object was to understand what a supporting website means, this was met by the website that the game was hosted on being well designed and user friendly, this allowed the website to perform quickly and easy to be set up. The next object was to try new techniques, this had been done with the login system, but this was also met with the use of familiar techniques as I knew they worked, I would say that this was a success as the main parts of the game was completely new territory that I had not explored before the start of this project. The third object was to create a program that worked smoothly, personally I would say this was a successful as this was done both in the login page, the user details page, virus creation and the game. The game occasionally experiences some lag when the user moves their mouse over all of the continents in quick succession and the game slows down while it tries to catch up, but this issue cannot be avoided. In terms of the final objective to use a clear work order and follow it, was unsuccessful as this project was all based on my project management skills, if I found any issues I fixed them meaning that sometimes the work order had been messed up and I didn’t use Trello after the first two months of the project’s start, meaning that when the time came to finish the project there were small parts that had been overlooked that were quickly added into the system.

Currently if I were to put the game on the market, I would not be happy with this. This is due to the issues that can be fixed as they can sometimes be annoying, for example one issue of the day being incorrect and this causes the counter to not work and the game does not run, the fix to this is to reload the page and this often fixes the issue. And personally, I think she game should be more interactive, and this is why I do not think that the game should be put onto the open market at the moment.

Conclusion

To conclude the project report, the process of developing the game and supporting website, was an enjoyable experience, even if at some points the issues that arise were difficult to overcome. This experience allowed me to explore aspects of HTML coding that I had not tried to tackle before, this meant that I could freely try out new ways of coding and an attempt to ensure that all code could be reused for multiple aspects of the game. Overall, the game was not a step forward in terms of the gaming industry, but this had been a huge step for personal experience with coding games and some techniques that have been learnt may show in future works.

The planning of the game was enjoyable and a learning experience that I will not forget when it comes into thinking about what needs to be added into a simple game let alone what the code behind complex games looks like. The planning stage of this project talks about how I used Trello and although this is true, I think that I could have used it more than what I did, occasionally I opened it to have a look at the key aspects of what the project required and that was it, and I would have found somethings easier if I had used Trello more. However, as also stated GitHub was an impeccable tool that was selected to be used, as swapping between home and university writing and building the project meant that there was no risk of corruption from unplugging USB devices.

The building of this project allowed me to explore website layouts, even though I used a templated from w3school to help with the layout of the website, exploring this code gave me an in-depth look into what was necessary for the code to work without affecting the user interface. The was a few issues that had been discovered within the code and often these were sorted in a few hours of the problem being discovered as often they required a bit of complex organising for the issue to be bypassed. There are still a few minor issues that the game will have to undertake before it could be published for further use, but overall, the game works as intended, and so does the virus builder and the login system. This means that the system that has been built was a success, and all but one objective had been met, and the one that was not met was personal for me to use more effective planning techniques, personally I feel that this would have been met if this project were undertaken as a group but as it was individual there was a lack of urgency for things to be done in order.

I would have liked to build upon the project by adding in more interactions that depend on the virus that the user has created, for instance if the user selects manual upload have them move a file very so often, and other small interactions that the user could do these would have been similar to the minigames that can be found in various other games. This would have helped to make the game apparent that the game changes with what the virus looks like. Also, I would of like to add a few different stats for the counteracting the virus to ensure the user knew about the enemy of computer systems this could have appeared as news reports to give feedback to the user, this would have made the game both have more interaction with the user and also helped with the conflict part of an effective learning game.

Personally I think the game and the supporting website was successful as a few people that had tested the game when I asked them "what sort of thing was an infection mechanism" they answered with the options from the game with very little hesitation, they did however question how they knew it but the important thing was that they did know it, and this was one of the expectations I had hoped to achieve from playing the game. Also, the supporting website does allow a user to navigate through quickly so that it can be worked into a normal lesson without causing massive disruption to the class.

References

Bowen C, (2020) "*What is an email virus and how can healthcare businesses can protect themselves*" [Online] Available at: https://www.paubox.com/blog/email-virus/#:~:text=Email%20viruses%20almost%20always%20are,and%20can%20continue%20to%20evolve (accessed on 24th may 2021)

Gutierrez, (2012), "*the 5 decisive components of outstanding learning game*" [Online] Available At: https://www.shiftelearning.com/blog/bid/234495/The-5-Decisive-Components-of-Outstanding-Learning-Games (Accesses on 15th January 2021)

Fitzgibbons L, 2019) "*logic bomb*" [Online] Available at: https://searchsecurity.techtarget.com/definition/logic-bomb (assessed on 17th January 2021)

Techopedia, (2021), "Payload" [Online] Available at: https://www.techopedia.com/definition/5381/payload#:~:text=A%20payload%20refers%20to%20the,tend%20to%20be%20more%20harmful (accessed on 25th may 2021)

Santos J, (2020a, b, c, d, e) "*Agile vs. Waterfall: Differences in Software Development Methodologies*" [Online] Available At: https://project-management.com/agile-vs-waterfall/#:~:text=Some%20of%20the%20distinct%20differences,helps%20complete%20one%20single%20project (Accessed on 13th December 2020)