



Malware Inc.

Project Leader: Jay Guillard

Supervisor: Kimberly Tam

Contents

[Statement of Word Count 5](#_Toc36288341)

[Project Links 5](#_Toc36288342)

[Introduction (read to make sure) 5](#_Toc36288343)

[Project outcome (revenue and education) 5](#_Toc36288344)

[Processes: 6](#_Toc36288345)

[GitHub: 6](#_Toc36288346)

[Supervisor Meetings: 6](#_Toc36288347)

[Legal, Social, Ethical and Professional Issues 6](#_Toc36288348)

[Data Protection: 6](#_Toc36288349)

[Ethical Issues: 6](#_Toc36288350)

[Intellectual Property: 6](#_Toc36288351)

[Method of Approach 6](#_Toc36288352)

[Technologies: 7](#_Toc36288353)

[Risk Assessment: 7](#_Toc36288354)

[Risk Matrix: 7](#_Toc36288355)

[Summary: 8](#_Toc36288356)

[Design Diagrams: 8](#_Toc36288357)

[Interface Design (World Tab): 8](#_Toc36288358)

[Interface Design (Upgrade Tab): 9](#_Toc36288359)

[Start Page: 10](#_Toc36288360)

[Instruction/Help: 10](#_Toc36288361)

[Class Diagrams 11](#_Toc36288362)

[Country Class 11](#_Toc36288363)

[Malware Class 11](#_Toc36288364)

[ReadWrite Class: 12](#_Toc36288365)

[Logic Design: 12](#_Toc36288366)

[Infecting Devices and Unrestricting Countries: 12](#_Toc36288367)

[Upgrade Buttons (logic for both income and infection upgrades): 14](#_Toc36288368)

[Task Run Down 15](#_Toc36288369)

[Interface: 15](#_Toc36288370)

[Classes: 15](#_Toc36288371)

[Countries: 15](#_Toc36288372)

[Malware: 15](#_Toc36288373)

[Timers: 15](#_Toc36288374)

[Win and Lose Scenario: 15](#_Toc36288375)

[Income and Infection rate upgrades: 16](#_Toc36288376)

[Difficulty Levels: 16](#_Toc36288377)

[Casual: 16](#_Toc36288378)

[Normal: 16](#_Toc36288379)

[Hard: 16](#_Toc36288380)

[Infecting Restricted Countries: 16](#_Toc36288381)

[Leader board: 16](#_Toc36288382)

[Operating System Restrictions: 17](#_Toc36288383)

[Supervisor Meeting Summaries: 17](#_Toc36288384)

[Week 1: 17](#_Toc36288385)

[Week 2: 17](#_Toc36288386)

[Week 3: 17](#_Toc36288387)

[Week 4: 17](#_Toc36288388)

[Week 5: 17](#_Toc36288389)

[Week 6: 17](#_Toc36288390)

[Week 7: 18](#_Toc36288391)

[Week 8: 18](#_Toc36288392)

[Week 9: 18](#_Toc36288393)

[Brochure and Poster Design 18](#_Toc36288394)

[Logo 18](#_Toc36288395)

[Video Walkthrough 18](#_Toc36288396)

[Trello Boards: 18](#_Toc36288397)

[Trello Board 1: 19](#_Toc36288398)

[Trello Board 2: 19](#_Toc36288399)

[Trello Board 3: 19](#_Toc36288400)

[Trello Board 4: 20](#_Toc36288401)

[Trello Board 5: 20](#_Toc36288402)

[Trello Board 6: 21](#_Toc36288403)

[Trello Board 7: 21](#_Toc36288404)

[Trello Board 8: 22](#_Toc36288405)

[Trello Board 9: 22](#_Toc36288406)

[Testing: 23](#_Toc36288407)

[User Testing 23](#_Toc36288408)

[Summary: 23](#_Toc36288409)

[Alterations: 23](#_Toc36288410)

[Project Post-Mortem: 23](#_Toc36288411)

[Evaluation: 23](#_Toc36288412)

[What went well: 23](#_Toc36288413)

[Feedback: 23](#_Toc36288414)

[Future Improvement: 23](#_Toc36288415)

[Conclusion: 23](#_Toc36288416)

[Reference List: 23](#_Toc36288417)

# Statement of Word Count

# Project Links

All code written will be submitted to the final submission point. If for whatever reason it is unavailable, the software is available via my GitHub repository linked below. Included is also a link of my trello board for the project:

Trello: <https://trello.com/b/dhhDBJbr/malware-inc>

GitHub: <https://github.com/JayKay202/Malware_Inc>

# Introduction (read to make sure)

This project is based on the popular mobile game Plague Inc which allows the player to develop their own virus and to facilitate its spread across the world, but with the idea of spreading malware rather than a virus. Just like in Plague Inc, the user will be able to develop their malware so that it can be spread using different formats (e.g. app stores, emails, etc.) and different devices (operating systems) that it can infect. From the originating country, other countries could potentially be infected by using email, the app store, etc, but countries that have restricted internet access (China, North Korea, etc) will only be accessible by physically transporting (plains/boats) the malware. The more people in a country with infected devices, the more likely that the infection will spread electronically and/or make it onto a plane or boat into a “restricted country”. The user will also be able to evolve the malware so that it becomes a different type of malware, making it difficult to track and eradicate. At the start it could be adware and by the end it could have evolved into ransomware and so generate a cash windfall for the creator. Like the game, you will have the ability to speed up time and which will advance the game quicker and so reduce the time taken to earn the money needed to buy enhancements to your malware. Once a certain number of days has passed, countries will start researching a solution to the malware and which will result in the malware being eradicated once the research reaches one hundred percent complete. If the malware spreads far enough and can infect modern operating systems, then it will start to infect devices being used to eradicate the malware and slow down their progress.

# Project outcome (revenue and education)

The purpose of this project is to demonstrate how much we rely on technology and how one piece of malware can affect all of our lives. This should encourage people to go into cyber security roles to try and prevent these sorts of organizations from creating these pieces of malware. As this is an app there is also the potential for monetary gain in the form of pay once or with ads or with addition content that is hidden with a pay wall (freemium content). The purpose of this game is to therefore develop the users strategizing and resource management skills, while providing an enjoyable gaming experience.

Strategizing Skills: should they focus on making the malware produce more money or make the malware more infectious.

Resource management skills: as they will have to consider what advancements to make with their malware and the impacts/benefits of each change made.

Although there is no given client for this project, the possibility to sell this application to an app developer is a possibility.

# Processes:

Trello Board: This will be where all tasks that make up the project will be displayed and will be organised into three different categories. The first category is the backlog which is where all tasks for the project start. Once a task has been started it will be moved to the in-progress category, this is so you can see how many tasks are currently being worked on. Once a task has been complete it will be categorized as complete so that it is not mistaken as being in another category. All Trello boards can be found later in this report to demonstrate the project development over the weeks.

## GitHub:

This is how we will be organising are change management and are backups. This will allow for me to see how my application has change from previous pushes to GitHub. Therefore, if the application stop working after a change, we can look back at what exactly was changed and take the correct items to resolve the issue. As the application will additionally be stored online, if the machine that the application is being developed on is destroyed or lost. Then the application can be downloaded onto another machine, as long as you remember the username and password for the GitHub account.

## Supervisor Meetings:

The supervisor meeting is done in order to cover scrum meeting, which is needed given that we are doing agile project management. During these meeting we will be covering what we have done in the prior week and what we plan to do in the coming week. This will also allow for other students and supervisor to give advice on what feature can be add to the project and possible solutions to issues that I am experiencing with the project.

# Legal, Social, Ethical and Professional Issues

## Data Protection:

As We will not be storing any of the user’s information and all the information is stored locally and will not therefor be stored on any of our servers. Meaning that we will not have to worry about data protection when it comes to this project and can spend more time on other issues.

## Ethical Issues:

In terms of testing for this project the idea was to get students that are part of the supervisor group I am in to test which would result in no ethical issues. But because of the current situation of coronavirus, this has led to all university facilities being shut down and all teaching being transferred to online learning. Therefore, there is no easy way to get the students to test out the game. This is why the university have mad an exception and are allowing students to get there friends and family to take part in there testing, without having to go through the ethical application process.

## Intellectual Property:

All code is written by me as the only member of the project and therefore all code/intellectual property belongs to me. If the program is sold to an organization then they are the owners of the code in question and will fold all rights. Given that this game is based on a pre-existing game there is the issue with copy right. But Plague Inc was created by Ndemic Creations and there is a similar game known as Infection Bio War created by Fun Games For Free, which has the exact same game play as Plague Inc and no legal action has been take against Fun Games For Free.

# Method of Approach

Given the requirements of this cause the project is being managed using agile methodology with a scrum framework. Using Plague Inc feature a backlog of tasks that need to be complete in sprints. GitHub was used in order to do version control, which is in line with agile sprint to provide backups and evidence of sprints undertaken. In order to keep track of the progress of the project trello was used in order to see which task had or hadn’t been completed.

# Technologies:

This program will be done with the uses of C# windows forms, this is mainly as it is the solution that I am most familiar. There are some restrictions with using this technology like the fact I will not be able to have a graphics of the world with changing colours to show the spread of the virus. But I believe that this will be the technology that will allow for the greatest chance of project success given the limited time and limited resources (manpower). All other functionality will be achievable and the main display for the spread will be the world and countries summaries, that is included in Plague Inc. as a secondary means of checking the spread of the virus. With the readymade assets (buttons, datagrids, etc.) this will reduce the amount of time need to create the interface and therefore allow for more time to be spent working on the logic behind the interface. This will also allow for more features that are not specifically require, but will improve it function, to be more likely to be completed before the end of the project.

I have not used a database as this would require the application to be connect to the internet at all times. As Plague Inc. is what my project is based upon, I have done testing to see what functionality it has. One such feature is that it does not require an internet connection to run. Therefore, I have stored the countries information within the code so that the user can still uses the application when not connected to the internet.

In order to backup and change manage the project all files are stored on GitHub, which will allow for changes made to be pushed to the master copy on the GitHub server. There is the device with all files goes missing or damaged all of the data can be recovered by downloading it from GitHub. There is also a log of all of the changes made to the program over the time of the project. If there was an error that was not present during the last chunk of code, you can check the log to see what has changed since the last instant and uses that to narrow down the cause of the error.

# Risk Assessment:

## Risk Matrix:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Description | Likelihood | Impact | Overall Risk | Level | Action (if needed) |
| 1 | Requirements for the project change. | 1 | 7 | 7 | Ignore | N/A. This is because this project is based on an application that already exist and changing it to be malware rather than a virus. Also, as I do not have a client for this project, there is no possibility of them changing their mind. |
| 2 | Holidays and sickness. | 4 | 4 | 16 | Consider | Mitigation: The impact of this can be mitigated by ensuring that we account for extra time during the development of the game. |
| 3 | Coding delays caused by inaccurate estimates. | 6 | 6 | 36 | Consider | Mitigation: We can account for this by building in a degree of flexibility in the timings. |
| 4 | Errors discovered in module testing. | 3 | 8 | 24 | Consider | Mitigation: We can mitigate this by running tests throughout the project and so ensure that small errors do not become big errors. This will also make the error easier to locate as you will know that it is something you did after the previous test. |
| 5 | Having to change technologies mid project. | 6 | 10 | 60 | Take Action | This can be mitigated by researching beforehand to ensure that the technologies will be suitable for the project and allow for the project to be completed on time. |
| 6 | Loss of data/progress | 1 | 10 | 10 | Ignore | N/A. The reason the probability of this happening is so low is that the project is additionally stored on GitHub. All changes are pushed to GitHub, so if something happens to the device being used the code can be downloaded onto another machine. |

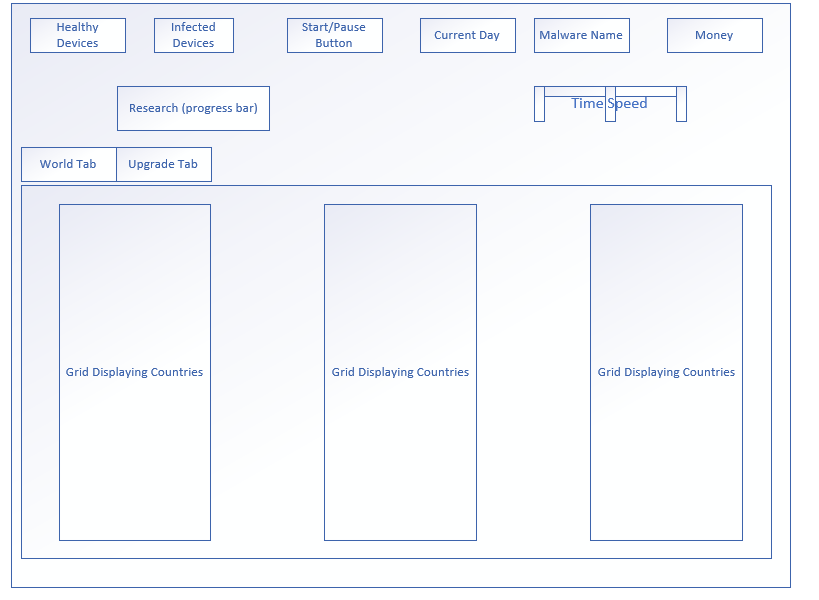
## Summary:

There may be risks that have been missed, but without some outside expect, theses risks may not be discovered until they have taken affect. Which is why it is important to have contingencies in place to make sure that any unknown risks have a reduced impact.

# Design Diagrams:

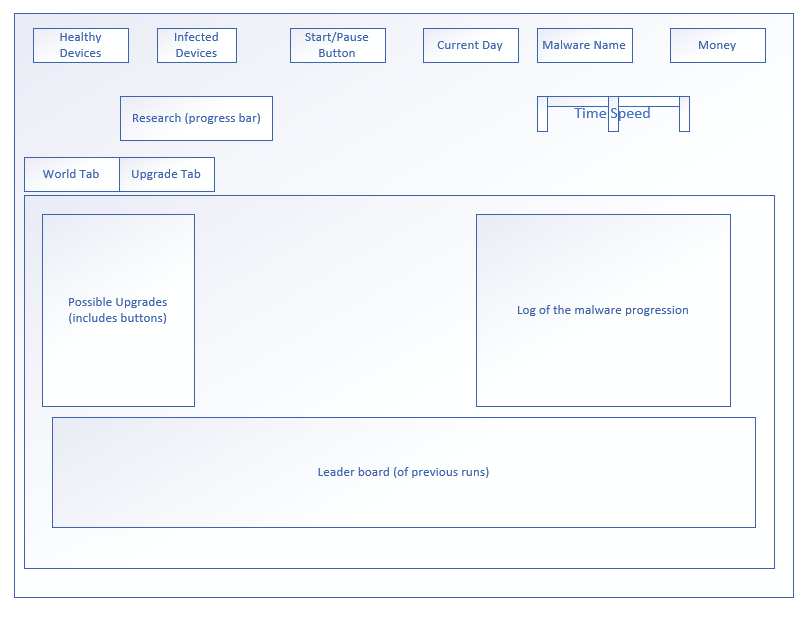
## Interface Design (World Tab):

All information that is stored in the application will be displayed on the interface. At all times the user will be able to see the overall healthy and infected devices, allowing them to make decisions of how to proceed given the information. The user can also pause the game at any point to allow for better strategizing.



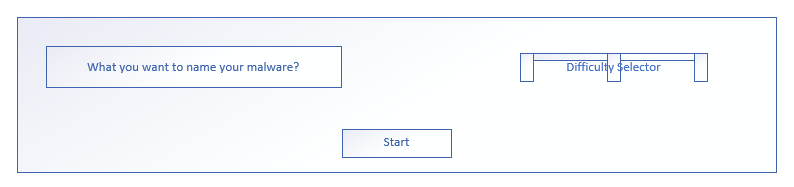
## Interface Design (Upgrade Tab):

Once the user has change to this tab the game will automatically pauses, so that the user does not have to worry about how the malware is progressing while upgrading the malware. This will also include the leader board, so the user can see how they are currently doing compare to previous runs. Once the user has selected an upgrade it will be display in the upgrade log and will include a description of how the malware upgrade, this is to educate the user of the type of malware and how it can get onto your devices.



## Start Page:

This page is where the user will make their decisions for the game they are playing. In order to start the game the user needs to select there difficulty (Casual, Normal, Hard) and the name that they want to call there malware.



## Instruction/Help:

The Instruction page for will contain a short paragraph that instructs the user how to play the game and what the overall objective is, below is a copy of what is included in the form:

The virus will progress without the user’s involvement, the user is responsible for upgrading the virus to make sure that it has a better chance of succeeding. This is done through the upgrades can and a variety of properties: Infection Rate, Income Rate, Possible Infectable Devices and Research Rate Difficulty.

Infection rate will determine the minimum and maximum amount of device that can be infected in a country every day.

Income rate will determine how much money is generated per infected device.

Not every device is infectable at the start, this is to represent operating system limitation on the virus. By upgrading the possible infectable devices the virus can infect more advanced operating systems.

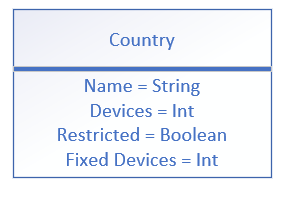
By upgrading the research rate difficulty this will make the research take a long period of time, allowing for the virus to have more time to spend.

Objective:

Infect every device in the world with your virus and aim for the shortest time possible.

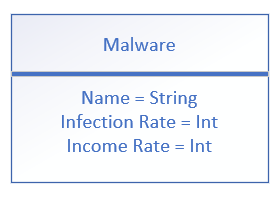
## Class Diagrams

### Country Class



At the start of the game the fixed device and the devices are the exact same values. During the game the devices will reduce as they become infected by the malware. The purpose of the fixed device is for the operating system restricts. As we need to know how many there were original for how many can be infected with the current restrictions of operating systems. The restricted variable is to determine which countries have restricted internet access and will be change to false once a certain condition has been met, at which time the country will start losing devices. The name will contain the countries name so that the name of the country can be seen in the data grid with the number of the devices.

### Malware Class



The malwares name is stored so that if a leader board is implemented, it will be a method for the user to identify which play through is what entry. The infection and income rate will increase as the user upgrades the virus. As the infection rate increases so will the number of devices infected per day and the income rate will increase the amount of money generated per day.

### ReadWrite Class:

This class is used in order to facilitate the leader board. It contains the code that will allowed for the malwares to be added to the text file saving the info and when the application loads that info is read by the application and displayed on the data grid located in the upgrade section.

## Logic Design:

### Infecting Devices and Unrestricting Countries:



This logic demonstrates how the program determines where or not to infect devices in a country and if so, how many countries. If a country cannot be infected and the world percent is above a certain value, it will run code that will randomly generate a number and if it generates a certain value it will un-restrict that country and therefore allow for the devices in the country to be infected.

### Upgrade Buttons (logic for both income and infection upgrades):



This code works by running through each of the possible outcomes of pressing the button until one is either met or it has reached the last possible outcome. As the only reason the button would not do anything is if the user has not got the money necessary for the upgrade. Once the code has decided which upgrade the user is buying the level is increased and the money for that upgrade is deducted. In case the user the user presses the button once they have fully upgrade a pop up will display to tell them they have already fully upgrade.

# Task Run Down

## Interface:

The user will be interacting with two forms when using this app. The first is the Malware Creation form, which will allow the user to come up with a name for their virus and the difficulty of the game. The second interface is the main game menu which will have multiple tabs, so the user does not have to keep opening different forms.

## Classes:

One of the features of C# is that it allows for the coder to create there own class inorder to store data, this is mainly to make the code easier to read and to also reduce

### Countries:

All countries are made up of the name of the country and how many devices are within each country. The value of the device will reduce as the virus spreads. All this information is then saved to a list and displayed in a data grid.

### Malware:

Although there is only one malware that will be created, I still thought that creating its own class for the malware would allow for the code to be more readable and easier to understand. The malware is made of it name, which is displayed in on the interface. It will also be used later in the leader board in order to identify which virus is which. It also includes an income and infection level, which are the type integer. These will affect how many devices are infected per day and how much money the virus also produces during the day. These levels will increase as the user purchases updates.

## Timers:

There are two timers that are responsible for the majority of the applications functionality. The first timer (timer1) is responsible for the processes that take place during each simulated day in the applications. This involves how many devices are infected and how much they have made. The number of infected devices is randomly generated using the malware infection level to determine the minimum and maximum and uses that to generate a number in between. The income is determined by the number of infected devices overall and then multiplying that by a different value depending on the income level of the virus. The data grids are updated to show the changes and so is the money display.

The second timer (timer2) is responsible for the pausing and unpausing of timer1 when the user when the user goes to the upgrade tab. This is accomplished by checking what tab is currently selected and if the upgrade tab is currently selected then timer1 is enabled and if the world tab is selected it reactivates timer1 allowing for the game to continue.

## Win and Lose Scenario:

Theses scenario’s will only happen if either the research or the infection has reached one hundred percent. In both scenario’s the user will either be congratulated or condoned for their lose. If the user wins the states for the game will be save in the leader board so the user can look back on their most successful games.

## Income and Infection rate upgrades:

Upgrading the income and the infection rate are done in the same way. They both are done by the user pressing the upgrade button for the specific thing they want to upgrade. Once the button is pressed the game will check what level the stat is and how much the next upgrade cost. If the user has the correct amount of in game currency, then that stat of the malware will increase, and the appropriate amount of currency will be deducted from the players total. As mentioned prior in classes as the stats increases so will either the currency generated per day or the number of device infected per day.

## Difficulty Levels:

The level that the user selects will change the conditions that cause research to commence and how quickly it is complete by default. There are three different difficulty levels which are Casual, Normal and Hard. The user can select the difficulty when they start the game and enter a name for the malware.

### Casual:

For this difficulty level research will start as soon as either days has reached 100 or the user has upgraded the malware infection level have reached 6 or income level to 5. Once research has started it will increase by 0.1% per day and this may change if the user upgrades the virus to make it harder to research.

### Normal:

For this difficulty level research will start as soon as either days has reached 75 or the user has upgraded the malware income or infection level have reached 4. Once research has started it will increase by 0.25% per day and this may change if the user upgrades the virus to make it harder to research.

### Hard:

For this difficulty level research will start as soon as either days has reached 50 or the user has upgraded the malware income or infection level have reached 3. Once research has started it will increase by 0.40% per day and this may change if the user upgrades the virus to make it harder to research.

## Infecting Restricted Countries:

As there are some countries in the world that have restricted internet access, in order to make the game more challenging these countries cannot be infected until a certain condition has been met. Once that condition has been met each day there is a 1 in 4 chance that a country will be infected by the malware.

<https://eu.usatoday.com/story/news/world/2014/02/05/top-ten-internet-censors/5222385/>

## Leader board:

Once the user has won a game the data from that malware will be save and display on a leader board. This will include the name, infection level, income level, date and the difficult. This is so that the user can review their previous their previous games and see how they can improve. This leader board will be located in the upgrades tab, so that they can reflect on previous games while they are upgrading their malware for the current game.

## Operating System Restrictions:

Not all malware is affective against every type of device, which is why the user will have to upgrade what devices they can infected. At the start of the game the user can only infect fifteen percent of all devices and by upgrading they will be able to infect all operating systems eventually.

# Supervisor Meeting Summaries:

The purpose of this section is to document all of the process and ideas that were generated during theses meeting.

## Week 1:

During this meeting all participants of the meeting introduced themselves and what they are creating for their final year project. This session was more of an introduction with are supervisor and what we had to do for the start of the project (e.g. share trello and add them as a user to GitHub repository).

## Week 2:

This week involved the group sharing what we have accomplished during the following week and what we plan to do during the coming week. When it came to my plan for the coming week there were no suggestion as to do anything differently. We also discussed the layout of the report and what sort of content should be included.

## Week 3:

During this meeting it was suggested that I implement operating systems into my program, so that the virus is limited by what device it can infect until it unlocks other operating systems. This was a good suggestion so I have now added it to my trello board and will be testing to see how exactly I will go ahead with this new feature.

## Week 4:

During this meeting it was brought up about games and the need for persistent data. It was decided that if it is not necessary for you game then it is not required to have a database in place to store any data.

## Week 5:

This meeting we covered what we had done over the last two weeks, as meetings are now every other week. The summary of this was that the plan of work was acceptable and that are supervisor would find more information about the requirements of the poster.

## Week 6:

This week’s meeting has been cancelled due to the closure of the campus, which is a result of the coronavirus. This has led to Face to Face supervisor meeting not taking place and the implementation of virtual meeting taking place. All aspect of the showcase and the viva have been changed to allow for them to be done virtual. The showcase will now be replaced by the poster, which will now be a digital poster and will act as a gateway for business professional to explore your project. They will be able to see the GitHub repository of the project and a video walkthrough, to replace the hands-on experience at the showcase.

## Week 7:

## Week 8:

## Week 9:

# Brochure and Poster Design

## Logo

The logo is a traditional danger of infection logo, but with the yellow background removed and replacing the curves of the sign with lines.

A picture containing clipart

Description automatically generated

## Video Walkthrough

The purpose of this video will be to give uses an experience of how the game operates. This is due to the project showcase being change to a digital format, due to the campus being shut down because of the coronavirus. The video will be replacing the people trying the software at the event and will allow for the business profession and next year student to still have a look at what the final year students have created, without having to through a physical event. A link to the walkthrough will be include in the post, as this will act as the gateway to all the material that we would have shown to the guest at the showcase.

In order to create this walkthrough, I will be using screencast-o-matic screen recorder. As it will allow for uses to capture the game play on the screen and do some of the editing before it is saved and published. It also allows for voice testing and screen adjustments before recording, to make sure that we don’t have to mess around with setting while recording.

# Trello Boards:

The trello board is used to keep track of how the project is progressing, by seeing what tasks have been completed, in progress or have yet to be started. This means that as the end of the project grows nearer, we will be able to see how many tasks we have left and tell if we will be able to complete all of the them or if the project will run over.

## Trello Board 1:

A screenshot of a cell phone

Description automatically generated

## Trello Board 2:

A screenshot of a cell phone

Description automatically generated

## Trello Board 3:

A screenshot of a cell phone

Description automatically generated

## Trello Board 4:

A screenshot of a cell phone

Description automatically generated

## Trello Board 5:

A screenshot of a cell phone

Description automatically generated

## Trello Board 6:

A screenshot of a cell phone

Description automatically generated

## Trello Board 7:

A screenshot of a cell phone

Description automatically generated

## Trello Board 8:

## Trello Board 9:

# Testing:

## User Testing

By getting other users to test the game we will be able to find error and get their feedback on how to improve the application.

## Summary:

## Alterations:

# Project Post-Mortem:

# Evaluation:

## What went well:

## Feedback:

## Future Improvement:

# Conclusion:

# Reference List:

Top tier of infection:

<https://www.cisco.com/c/dam/assets/offers/pdfs/midyear-security-report-2016.pdf>

ransomware most profitable

<https://ieeexplore.ieee.org/abstract/document/7579103>

drive by infection

uses interview on the 7th feb as reference for using the easy option.