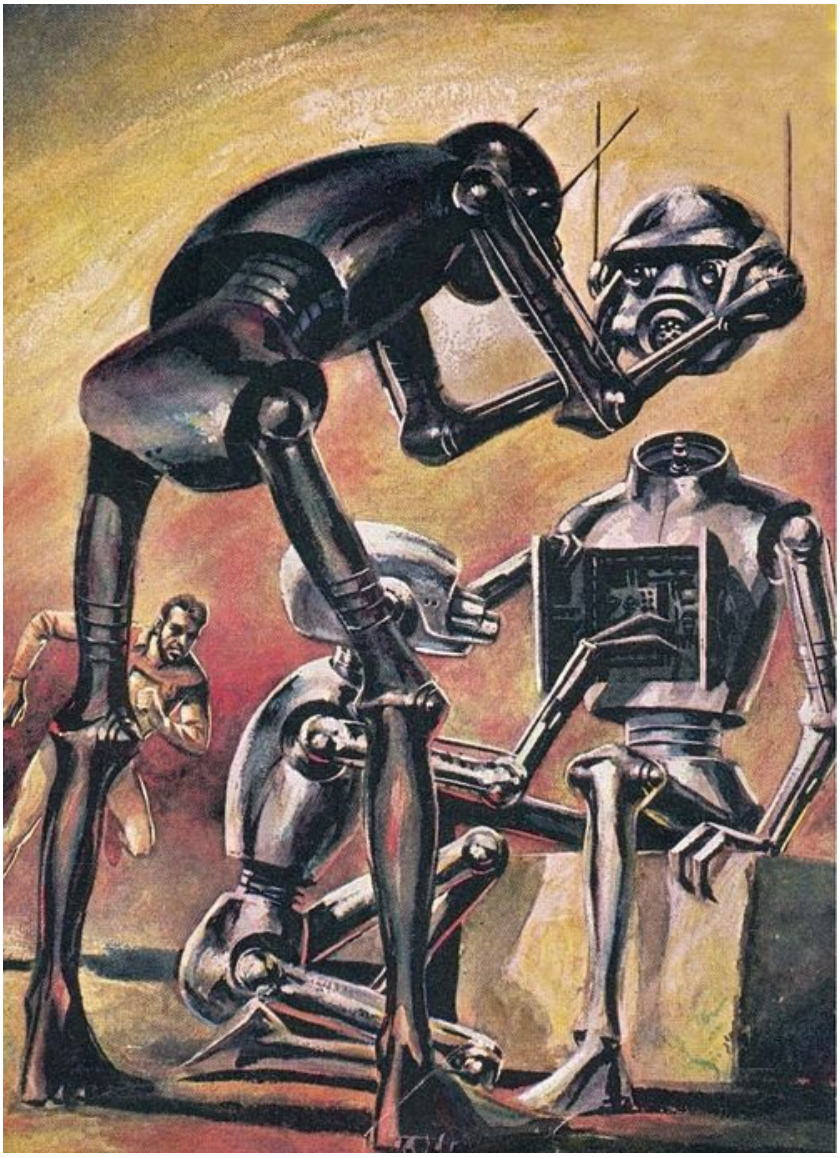
Glyn Ellis

Desktop Companion Robot



STATEMENT OF ORIGINALITY

CS3D661 Individual Project

This is to certify that, except where specific reference is made, the work described within this project is the result of the investigation carried out by myself, and that neither this project, nor any part of it, has been submitted in candidature for any other award other than this being presently studied.

Any material taken from published texts or computerized sources have been fully referenced, and I fully realize the consequences of plagiarizing any of these sources.

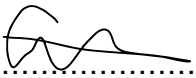
Student Name (Printed)	GLYN ELLIS
Student Signature	 ……..
Registered Course of Study	Computer Science
Date of Signing	29/03/2019

Table of contents:

Desktop Companion Robot 1
STATEMENT OF ORIGINALITY 2
Table of contents: 3
Chapter 1: Introduction 4
Chapter 2: Literature survey 5
Chapter 3: Methodology 11
Chapter 4: Implementation/Experiments performed 14
Chapter 5: Results and analysis 18
Chapter 6: Conclusion and future work 20
Appendix: 22
Sources: 22
Weekly diary: 22
Compiling and running the code: 24
Ethics form: 24

Chapter 1: Introduction

The project design was steered by the need for a practical solution to the ever increasing forms of communication, it has now become the norm to be active across a variety of platforms and services. The average person must be able to keep track of all inbound/outbound information from these platforms. That means logging on to each one whenever it is required, navigating menus taking unnecessary time that would be better used elsewhere. Workflow and productivity also steered the design into a simple solution. Displaying everything in their simplest most easy to understand form was a must.

When receiving the title project, many ideas were bounced around, such as a digital only version connected between two win form apps or a single application on screen. Ultimately it was decided that to achieve the best outcome for the project it would be best to have an actual physical robot that sits on your desk. A meeting was conducted with project manager Christopher Tubb and this was all confirmed.

By the end of the project the goals are to achieve the following:
-A working desktop companion that can assist the user by displaying key information (emails received, weather, twitter, deadlines…).
-Integration with robotic elements (arm, ultrasonic sensor, microphone, speaker...).
-A way of interacting with the user autonomously or on triggered actions.

The end goal is to have a small screen with a disembodied head/face on it that is basically a personal assistant. It will handle notifications such as received emails, tweets and messages. It would also be great to introduce some robotic elements to it such as a small arm or have it sit on a servo and be able to “look” around the room.
Its main purpose would be to assist the user by improving their productivity and providing them with information that would otherwise break their workflow. It is also important to keep in mind that the solution must be its most understandable and simplistic form while also keeping a sense of professionalism and style. This may mean that elements may be sacrificed for the sake of a simple solution.

This project is very applicable to a lot of people, particularly computer scientists and other professions where desk work makes up the bulk of the workday. It is easy to get distracted and stray away from work when you are sitting at a desk all day. The project can help as it gives users an excuse not to look at their phone. It stops them closing their work to “check emails” or “check the weather” as they are attending an event later, which means go on a tangent on the internet for an hour or so allowing for work rate to grind to a halt. Hopefully this project will continue to be updated and used long after it has been officially submitted.

Chapter 2: Literature survey

When looking for research literature it is important to find sources that directly relate to the project and not try to fit sources to the project. Studies that involve any aspects of the user interface such as overall design, size, icons, text. It is also important to look for user ergonomics information as the project has a physical presence as well as digital. These sources would provide the most relevant information to the project and help give insight that will help create a better, more informed final solution.

Zhao, Chen, Zhang, Tao, & Zhang, Kan. (2001). User Interface Design for the Small Screen Display. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 45*(22), 1548-1550.
An academic research paper in which an experiment was performed to see how ui design characteristics on small displays affects a persons performance as opposed to a regular sized pc interface. The experiment was a memory style game using varying sized matrices containing icons. Results showed that when testers had to remember high and low amounts of icons they performed significantly better when viewing it with a colourful display as opposed to a black and white display. When there was a medium amount of data it was only slightly better performance in favour of the colourful display. From this it was gathered that when icons were displayed in colour they became a lot easier to remember and that user performance is better when there is less information to deal with. The testers were also asked to rank their preferences, they gave the colour interface the highest ranking and also preferred the lower amounts of information. This information can be used for better understanding of the way that the user thinks and what display they would feel most comfortable with. From reading this study it seems that the best way to go about the design is to include a low amount of information and display it in a distinguished, colourful manner.

Yan, R. (2011). Icon Design Study in Computer Interface. *Procedia Engineering, 15*, 3134-3138.
A paper trying to understand and make icons that users can quickly and accurately comprehend and obtain information. The paper also discusses the principles of icons when used in digital interface and the design methods for icons when used within the same interface. Icons are graphical indication of a specific command, data, option, program and so on. In most cases the icon has a direct connection with the function that it is referring to or it is performing. Icons are used for understanding software features and helping the user in a more visual way. Icon design should be related to the overall look of an interface but also have its own unique feel and set of rules with regards to its design.
The ability to identify the icons is the core of the icon language, because if a user cannot understand the purpose of icons or their implications then they have completely failed their objective and are essentially meaningless.
Icon interaction provides for a more meaningful experience with the user, it may be necessary to include an icon in certain situations but it helps the user feel more of a participant within the interface.
An icon should meet the needs of functionality first and then should be able to convey itself as stylistic and elegant. The icon design needs to highlight its quality while also blending in with the overall design of the interface. Icons also need to make use of people's ability to identify and think of what the icon is conveying, if the icon is unclear then its functionality will also be unclear. There are many categories of icon types, the ones most related to use within the project are:
-Icon type with basic image feature
(all image objects)
-Text icon type
(anything that involves a form of text)

In comparison to text, the simplicity and intuitiveness of icons seem a lot more timesaving and easier to understand to people, also icons can better stay in peoples memory than text. Studies show that the average person can remember 5-6 word forms per 600 words read, but on the other hand can identify around 540 words. There has also been studies that show that people's identification ability is much weaker with regard to texts as apposed to images, a person can identify on average around 77% of 10000 images viewed previously. (Yan R, 2011)

Icons also have cross language recognition, as there is no text there is nothing to translate and therefore the language barrier will not be a problem.
Due to life becoming more fast paced and time efficient this has also caused people's habits to change, people have now gone from reading to viewing, the area in which icons excel. People in this modern age are dependant on quick visual stimulation that captures all the information they need at a glance.
The design process of an icon has multiple steps but the ability to have the icon accurately identified is the end goal of the designer. The beginning step of designing an icon is to think about the user, what will they be able to identify with and be able to discern when it is a slightly more abstract icon version of a concept they are familiar with. The design method that should be followed when creating an icon is:

-Select common/appropriate metaphor objects.
-Evaluate target audience and their ability when perceiving the icons.
(Culture and language backgrounds).
-Understanding of metaphors
(If the user has to think too much then its impractical)
-Uniqueness of metaphors.
(Overlap may cause confusion).
-Avoid unmatched metaphors
(if there is no correlation between an icon and its function it will confuse users and cause identification issues).

Icons should follow a constant design style as it reduces the understanding time of the users viewing them. It is worth investigating the users/target audience software usage to understand what icons they are familiar with as to not design icons that are too unfamiliar.
For the first time users are viewing the icons designed it is worth including:
-A text based version or text guide to accompany the icons so that the user is not left to figure it out themselves.
-Show the icon in combination with its corresponding action so that the user can make the mental connection and can identify the function of each icon.
-Provide a tutorial/talk explaining icons and methodology in detail to help users.
-Redesign or simplify any icons that are poorly designed or are hard to understand.
-Take feedback from users on icon design.

This paper is key as the system relies on the use of icons to provide information to the user. The most important thing that should be taken from this paper is the methodology as it encapsulates the perfect icon design and should be taken into account whenever an icon is designed.

Ko, P. (2012). Effects of Computer Display Design on Health and Productivity. 84.
A study into the effects of computer displays on user health and productivity, such information should be taken into account when designing the project. If the user has to look down constantly, or move the position of their head it may lead to pain and posture issues. If the display is too small it can lead to visual disorders, this is particularly important to the outcome of the project as it effects the final design and placement due to the size of the icons. The most common health problems that come from prolonged computer usage are eye, neck and shoulder issues. The act of leaning forward with the head and neck causes sensitisation of the pain receptors and then shoulder and neck pain.
The risk factors that come with prolonged screen use are:
-The characteristics of the visual functions
(Reduced ability to focus on text and icons, focusing on multiple items).
-Visual demands
(Viewing distance not optimal, text size, lighting situation).

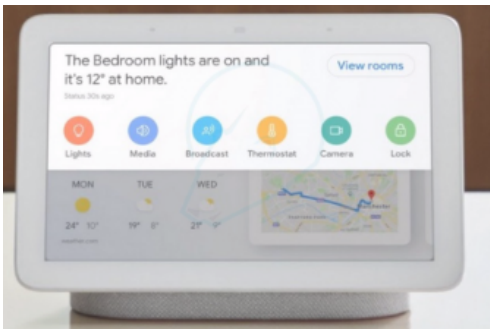
When users are in a natural environment where they are not constrained by viewing distance they will optimise their own viewing distance that best suits them. VAF or visual angle of font is expressed as the ratio of the font height to viewing distance. For normal sighted users it is recommended a VAF arcmin of 16-22. Smaller font sizes cause users to move head, neck and shoulders into uncomfortable positions which eventually causes pain. This is one of the most common reported problems in office ergonomics. These problems can be counteracted by

moving the chair, monitor or increasing font size. The average computer user will not do this and instead reduce the distance to the monitor by leaning the shoulders or neck closer to the text, or by moving the chair closer. The moving of the chair when done with a correct posture will not cause any pain while the other two options will. When it comes to the design of the system these issues should be thought about and also how the average user under natural conditions will adjust posture and viewing distance to the project as it is smaller than an average computer monitor and thus will cause them more stress and strain naturally.

https://store.google.com/product/google_home_hub :
The google home hub is a product sold by google that is a multi-purpose home assistant. It features a 7' display and built in speakers. It is one of few products that inspiration can be drawn from when designing the project. It is made to handle a user's daily life through the use of google apps. It can add events and locations to a calendar when asked, it can also perform google searches or connect to music streaming apps to play music. It has a variety of features, but not all are relevant to the project. It would be beneficial to break down the overall design and more specifically the icons used to gain insight on how a top tech company approaches a system in the same vein as the project.

Google home hub design:
The whole product should be viewed when analysing the design, one of the most noticeable aspects is the round edges. The border of the device features rounded corners, the font is very curved whenever possible with heavy use of white. The softer edges seem to convey a more informal and relaxed design, making the user feel more welcome and help the product fit better within a home environment.

(Picture 1 forecast)(picture 2 blue background sunny)
When there is less on the screen the text is larger, you can clearly tell that what is being displayed is the current temperature. It then morphs into a weather forecast window, everything is spaced evenly, the background colour does not conflict with the information that is being displayed on screen. The icons themselves are small but as the screen is 7' they should be viewable. The weather icons do follow a consistent theme with the sun being the focal point of the design, the only issue is that when it is extremely cloudy the sun is a lot smaller, this is confusing until understood. (Yan, R. 2011) Discuss how the icon should be distinguishable at first glance, but the radical size change of the sun makes you have to take a look and really think about, this is less than ideal when it comes to designing icons. The use of a simple up and down arrow to show the temperature high and low for the day is very clever, it is a recognisable symbol and is instantly understood. Some of the text does seem quite small for the user to have to read, but it is not meant to be viewed for extended periods of time so it will not become an issue.



(Weather and notifications)(Dropdown menu)
When more information is being shown the screen splits into different columns, and a more condensed view is shown. For example; when the notifications are being shown the weather forecast gets formatted into compact design, the notification tabs are well designed imitating a post-it note which makes it feel more of a 'home' item. Some text seems to be small and may take getting close or squinting to see it. The drop down menu screenshot showcases the feature set of the device, it is extremely multipurpose and each purpose is clearly labelled. They are distinguishable by colour but also are clearly a set as the icons are designed in the same manner; they are a white outline of a single object that is not too busy and can easily be identified at a quick glance.

(Goodnight screen)
The home hub also seems to offer stylised good night/morning screens, this gives the user something different to look at instead of the information rich background that is usually shown. As you can see in the night screen they are extremely clear in showing the elements of night, the moon shining through the window, the clock set to a late time, the lamp is on and other industrial buildings outside have all lights turned off. All of these things help build this comforting scene and it is not noticed unless the scene is analysed. The brain subconsciously processes all this information for you and you just understand what is being conveyed from a quick glance.
From looking at the different home hub scenarios it is clear that a lot of thought has gone into the design of each element and how it all fits together to create the smoothest user experience possible. One thing that is present in almost all frames is an element of the Google logo branding colours, it is unmissable that the product is a google product, the yellow, red, green and blue are seen throughout even though no direct Google branding is shown.
Tying all this information in with the project, the closest thing to branding that could be included would be a recognisable colour scheme. With regards to insight into the design of the project, the icons used should follow a shape, either very straight and rigid or quite loose and with flow. Also if text needs to be shown it should be larger than some of the font sizes used on the home hub, it is quite small at times. It is important to distinguish the home hubs purpose as compared to the project, the home hub is built upon a voice recognition system where you talk to it to have it perform actions, the project is being built from the ground up to be digital. The home hub is a home companion not a desktop companion, so its design is built around handling a lot of things that are considered 'daily life' and won't be included in the final solution of the project.

https://www.amazon.co.uk/Amazon-Smart-Speaker-Screen-Alexa/dp/B01J2BK6CO:
The amazon echo spot is a smaller condensed version of the amazon echo, it has the ability to video call, play music, command with voice and use amazon services. At first glance it looks to be very close to the design direction of the project, and is worth investigating thoroughly.

The first thing you notice about the spot is its unique shape, it is a sphere with a flattened bottom and an angular flat side for a round screen with a thick border. The information is displayed within the screen, it feels a lot more formal than the home hubs 'cartoonish' design.

(Good morning screen)
It uses a harder more straight line white font, the icons also follow the same white outline only design.
It seems as though there is a lot of text for such a small display, this may hinder user experience as it would be hard to see.

(picture 2 video call)
The system has a video calling feature, which would not be beneficial for long term usage as the screen is so small, it seems a lot more suitable for their larger line of products.
Whenever an application is being used the screen seems busy, as the screen is so small it has to make use of the space in creative ways.

(BBC news in a kitchen setting)
Another feature of the echo spot is that it can display video applications, you can view the news, video clips and film trailers. At first this sounds like a good feature but can get old quick as there

are many other better viewing platforms such as your phone, pc or tv which are larger and may be more favourable. The settings used in the display photos show different rooms each time and also the company brands it as being “designed for every room”, for its size this may be a stretch.

(Digital clock, bedside setting)
This photo seems to sum up its best usage, as a bedside clock. When someone is lying in bed and does not have their phone they can speak to the device or press a few buttons to access its range of features. This is where it would stand out over the other, larger models, as most bedside spaces are extremely limited. It is also hands free so that you wouldn’t have to get up to use your phone, but most phones have a digital assistant that can perform similar tasks.
Overall I think the design is not as aesthetically pleasing as the google home hub, it also does not feature a prominent style throughout the system, it feels as though it is multiple separate applications stitched together with no overall connection. The round screen could have made for interesting displays but this does not seem to have been taken advantage of.
If there is anything to gain from delving into the product it’s that it is best to not overcomplicate the system with too many features, doing this makes its overall purpose hard to see. All features added to a system need to be as important as each other, with this system it seems certain aspects may go unused. It can also be used as a thinking point when it comes to factoring in the hardware shape to the software, a portrait screen would look better with more elongated long icons and a landscape with wider icons. All of this is important to building the most satisfactory experience for the user.

After reviewing both of the products it is clear that they have their own agendas, Google is trying to get you to use as many of their services as they can and Amazon is trying to get you to purchase things from their website. The purpose of this project however is not to advertise or sell to the user but to assist them.

Chapter 3: Methodology

The project requires the use of the agile methodology and will implemented by:
-Keeping a weekly diary and conducting scrums at the start of the week to determine tasks that need to be completed. This will be done every week.
-Using the evolutionary prototype method in which versions of the robot/program are created and re-used/build upon each time ending up with an increasingly complex solution until a final solution is reached.
-Treat each prototype as an agile development loop, getting feedback and reassessing design aspect when needed.
-A 'prototype diary' will also be kept.
-Have a Trello board to increase productivity.

Meetings are also held throughout the development cycle with the main assessor of the dissertation Chris Tubb. Meetings are confirmed via email, they are informal and advice can be given on issues or current direction. This keeps the project from stalling and keeps it on track, it is important to have a constant second opinion as it can be the difference maker in the outcome of the project.

It's a good idea to self reflect and develop prototypes going in different directions, exploring ideas and concepts early. For example: working with an Arduino and robotic elements to see if it has validity and if it should be pursued at the same rate as programming the system.

A cloud based method of storage is a must for the project as it ensures an offsite backup is kept at all times. The platform that will be used for the project is Github, this is used as it is tailored for use with coding projects. It also has features that will be beneficial to the project such as a visual graph of changes made to a project, a development log detailing all previous changes and the comments that came with them.

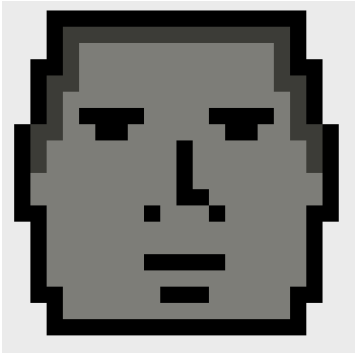
Visual studios code is used whenever programming in python for the project as it is a light programming IDE with a more than adequate python compiler and debugger allowing for easy error checking. It also has GitHub version control built in, allowing for side by side viewing of commit changes and the ability to make commits and push/pull said commits from within the IDE. As the project will be run on more than one machine GitHub integration is a requirement.

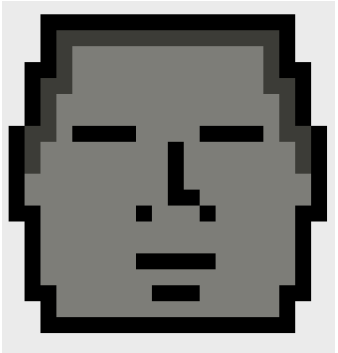
A raspberry pi is hopefully being used to display the project in its final form. The pi is small enough to fit nicely on the corner of a desk and the I/O pins can support the use of a display. The term desktop companion is stated within the project title therefore it is necessary to have some form of working and ready desktop setup towards the end of the project timeline. If this does not work the backup is displaying the final code solution on a MacBook, this is what development is being done on and is an adequate fall back as it can display the program with no issues.

Api's and api handling make up and will continue to be a large portion of the code. Api's are the core programming aspect of the system, without them it would be extremely difficult to reach the desired state. They are used as almost all online based services have api integration and python also caters to the usage of api's easily and quickly. The api's that are used are specific to services that are used on a day to day basis, they are also services most people use in an IT environment and will capture a regular person's workday best. Wrappers for Api's will be used when available, this cuts development time and provides a benefit by simplifying things such as syntax and authentication at no cost to program speed.

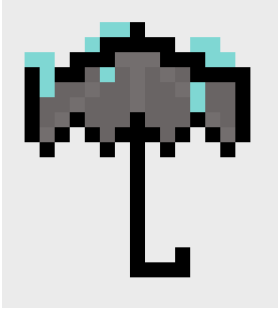
When creating icons the (Yan, R. 2011) design methodology was followed, following this ensured that icons are original in their own right but also fit in with an overall design. Icons are clear and understandable, the user does not have to think extensively when viewing an icon. As advised by (Ko, p. 2012) the icons are large and offer a varied colour palette, this is done to combat eyestrain after long periods of viewing/usage. All icons used in the final solution go through multiple design drafts. Text is also avoided as it would be small and cause eyestrain.

As icons are the basis of the project it is of utmost importance that they are done correctly, therefore each icon has gone through a review process and been re-designed or tweaked multiple times. Re-reviewing ensures that they are the best quality, they are reviewed against (Yan, R. 2011) study of icon design, as this source is very informative and if icons match this methodology they are to a high standard. Re-reviewing also means that the older icons are compared against newer icons, so if there are any inconsistencies in style and design they can be updated, keeping a consistent theme throughout.
Using images would work but it was important to give a sense of dynamism and character to the system for user interaction, so it was decided icons should be animated. Animating icons came with its own set of both problems and advantages as it does add a layer of complexity to the design, but allows for more information to be conveyed and gives that sense of dynamism that was needed to keep a users attention.
When beginning to design icons for the project, the first concept that took charge was an overarching pixel design, this was due to the fact that they are readable and there are many sources available to create them online. Working with pixels forces simple designs ensuring that too much time could not be spent designing one icon.
After reviewing the Amazon echo spot and the Google home hub, the more realistic the overall system design is the more it becomes unclear. This was another reason for using a pixel icon design as it is the style that ties icons together throughout the system.
The first icon designed was a pixelated head, this would be the focal point of the systems design. This was done as when it comes to making a system that directly interacts with a user there is usually a human element to it to make it seem more friendly. Such as with amazon's Alexa and apple Siri, it is an easier experience for users when the product can be humanised. Humans can also identify facial expressions and with the use of gifs, those expressions could be lot more lively showing a wider range of emotions that you can't get from single frame images.
A large amount of expressions and movements were made and then refined, this was done to see what kind of face detail you could get from a pixel animation. The most appropriate animations are then refined a step further as they are going to be used.



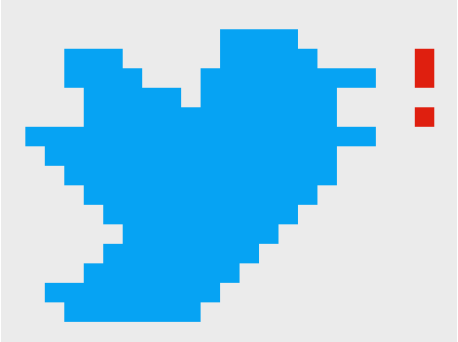


(Face animation) (Blink animation)
As these are screenshots of animations you don't get the full effect of them, but as you can see this is the face that is used. it has enough detail to convey emotions clearly. They all follow the same face design, if they did not it would look like a different character. This is a problem as this face represents the system as if it is a person.



(Frame of cloudy animation) (Frame of rain animation)

As there is a lot of different types of weather, designing icons is difficult. This is due to the fact that different weather conditions can be very close such as mist and haze, but also they can differ widely such as sun and snow. This means that when designing icons they must be clearly defined to not confuse the user as to the weather. And for weather icons that look nothing like each other they must follow a theme so that they fit in with the overall design. Both animations shown above are not similar but you can clearly see they are part of the same set. They have very smooth animation that work well when repeated and are distinguishable on their own.



(Frame of Twitter animation)
When designing the twitter animation it was important to keep the proposed design style but as gathered from the research it was important to include some kind of branding as that is how people will identify the icon fastest. The most memorable part of twitter is the bird logo so it seemed natural to use that, it also has the distinct pixel style that can be seen throughout all icons.

(Frame of Trello 'completed' list animation)
Trello also features a very distinct logo like twitter, so that seemed like the obvious choice and adding colour coding can say a lot with very little information. Trello also has coloured labels which can help the user create a better mental connection with what each icon is trying to convey.

When creating all icons there were a few rules that were put in place not pertaining to the icon design study to ensure that they would fit together within the overarching design. All animations must be limited in frames, if one is longer/shorter than the rest that could throw users off and make them think too much. Another rule was that all animations had to have a flow and not end abruptly, the aim of the design is to have seamless movement of icons. If an icon is being played multiple times it must look like its not reloading itself over and over as it will take away from the overall feel. If the icon feels boring or does not have an interesting/memorable design or animation it will be re-designed and if that still doesn't fix it then it must be removed. Icons must also have more than one colour, this stops them being dull and uninteresting.

Chapter 4: Implementation/Experiments performed

https://zapier.com/blog/what-are-webhooks/ :
A fellow student recommend researching webhooks as they may be of use, and after finding the website it seems a good idea to try integrating them into the project. Webhooks allow for apps to talk to each other, this is valuable when connecting the program to an email or twitter and have it receive a notification when an update occurs automatically.

http://doc.qt.io :
The official qt website has a wide variety of guides, they provide information on libraries, getting started and tools. This has been invaluable as its an unfamiliar IDE, having all the information in one location from a reliable source means more time saved searching the web.

https://www.youtube.com/watch?v=EkjaiDsiM-Q&list=PLS1QulWo1RIZiBcTr5urECberTITj7gjA :
This youtube users playlist was used to learn more about the IDE. The tutorials were also used when creating some starter programs as stated below.

https://www.elegoo.com :
A starter kit was purchased for the Arduino from this website, with it they supply a PDF document with a list of tutorials and circuit diagrams which has been used when experimenting. This will be used as a reference guide whenever any questions or problems arise.

https://wiki.wxpython.org
For all python gui programming the wxpython library was used, they are known for having an extremely extensive amount of information available on their wiki. The wiki was used when creating beginner solutions and whenever advice, or specific information regarding a problem with the library was needed.

https://www.piskelapp.com
Used to create all animated gif icons that are seen in the project, they will be used to display different emotions, actions and notifications to the user. Animated icons are created by colouring selective squares on a grid, they are created frame by frame and then played. Piskel has a very intuitive system with many tools to assist users in creating their own animated gifs.

The first complete prototype (c++) contained:
-Different face gifs displayed when buttons are pressed.
-Buttons for all the different emotions available.
No unexpected issues were encountered, animations are triggered on button click events.

Work was also done on some basic Arduino elements, building a simple red LED circuit that can blink and also an RGB LED circuit that fades through colours on a timer. This can be built upon by using the LEDs for things like specific notifications or as a light. Investigating the use of an Arduino further seems advised as this would be the key to handling all external robotic elements of the project. Experiments were performed with building a user following robot, a servo was programmed to move within a specified scan range. It would move from 0-180 degrees in ticks of one degree at a time, as this would be looking for a human sized objects the tick size can be increased. An ultrasonic sensor was used in conjunction with an led light that would be constantly measuring the distance of any object in front of the sensor, if it came within a certain distance it would turn the light on.

Webhooks:
Web hooks were discussed in an assessor meeting as a way to connect all api together within the program, it was deemed that it would hinder the code and final outcome too much and that the handling of all api's should be done within the program itself. It is an interesting concept and could be implemented in future versions.

Python:
As the project began stalling in C++ with regards to the http code, python was investigated as a possible avenue. Python was chosen as it is a proven solution for the projects current problems therefore it seemed like the logical way to make progress. A simple program was created that accessed data from an open source weather API. The program makes a request to a weather API to get data on a specified location and returns JSON data, that JSON data is then converted to a string. The string is searched for keywords such as sun, cloud and rain, if any of those appear it outputs a small message based upon which was contained in the string. This took no time at all to complete and worked perfectly, the only problem is that it is not C++. It showed pythons usefulness and piqued interest in possibly programming in a python GUI environment instead of C++.

Wxpython:
Wxpython is a library that you can install to do gui programming in python, it worked with the basic weather api prototype that built upon a previous cli text only version. The prototype polls a weather api and displays the correct gif for 'n' period of time, before returning to an idle animation state in which it polls the api again. Wxpython has extensive documentation and was recommended by many online users. It has proved its usefulness and has become the method of choice for displaying the program in a gui format after the ability to display animated gifs in wxPython was implemented. As previously mentioned the weather program was first implemented, in text form and then through the use of the animated icons. From there experiments were performed on displaying faces and calling functions, this kept being updated and is still used due to the rapid prototype methodology.

Weather api (OpenWeatherMap):
The weather api uses a simple get request that is authenticated by including the api key and city id in the request, it returns a json string with all information necessary to tell the correct weather. The way icons were implemented with the weather api is that it takes the string that contains all the information regarding the current forecast and checks the main weather forecast prediction. Depending on what the weather is an animation tied to a keyword is played. The (Yan, R. 2011) icon design methodology was used when implementing the weather icons. It helped provide a structure when designing icons as personal identification abilities of the designer may differentiate from other peoples and following a known set methodology will counter this.

Tweepy:
Tweepy is a downloadable python library that allows for the communication between python and Twitter through the use of the Twitter API, and has been used in the project as such. As it is open source it comes with lots of supporting documentation both official and user submitted. It was used to see wether the user has been mentioned in any new tweets. It also has access to most of twitters features making it a tool that can be used in future development. As it is icon based the full functionality of tweepy was not used, it would require text to display full tweets and such. What was ultimately shown was if the user has been mentioned in a tweet, an animation is played of the twitter logo tweeting a little red exclamation mark. From the Google home hub system breakdown that was done earlier it was evident that effective branding is what makes companies so identifiable. Therefore the icon was modelled after twitters most identifiable icon, its logo.

Py-Trello:
Py-Trello was used to handle the Trello API, it allows for easy navigation by accessing its different class methods/objects. It was implemented to allow for the monitoring of a specific board and lists. It contains extensive supporting documentation from developers. The Trello implementation is specific to a board with 3 lists (not completed, in progress, completed). If a task is moved into a list a colour coded animation is shown to the user.

Gmail:

A lot of time was put into trying to get the Gmail api working, it was a key feature that is important to the project. There is extensive official documentation when it comes to using the api, but it is very complex and hard to find the correct information when it came to particular problems. A program was made that checks the users inbox, outputs snippets of each message and lists the number of emails in said inbox, the only problem with this is that it could only be run from a terminal and would not run within the IDE compiler.

Idle state:
The program is built around displaying api's, but this is not done all the time as they wouldn't be updated that regularly so an idle state was added into the program. The idle state shows the head looking at the user for a period of time before the api's are called again, this time can be adjusted in the code but for demo purposes is very short. Ideally the idle time would be in the realm of 10-15 minutes as a user will be sitting at a desk for a few hours at a time. Showing the same animation for this will make it a bit boring and not very memorable so a random expression is chosen from a group of preset choices. It is chosen at random but is slightly more favoured towards the default blink expression. This was done as it is the least distracting for users, but when more dynamic expressions are shown it gives the system more personality and hopefully causes more user engagement.

Wifi check:
The api's require an internet connection to work therefore a failsafe was implemented after the idle function has run that will check wifi connection. It will try this three times before displaying a unique sleep icon waiting a specified time before trying again. Doing this stops the program trying to poll websites and getting stuck due to lack of internet connection.

Using (Yan, R. 2011) icon methodology:
An example of when the methodology was used was when designing an animation to display to the user when there were no notifications for a particular service. What was in place already was a simple head shake 'no' icon, this was deemed too vague and a redesign was necessary.
This is the methodology and how it was followed:

-Select common/appropriate metaphor objects:

The left/right head shake is understood by everyone as meaning no, this was used instead of a red X or something similar as the face is the core character aspect of the design. This seemed the most commonly understood metaphor while still fitting in with the design and therefore the most appropriate.

-Evaluate target audience and their ability when perceiving the icons:

The icon in question is dealing with the Trello api service, it expected that user will identify the Trello logo and the traffic light colour system for the lists as they are most likely using it themselves. The only issue would come from users not familiar with Trello.

-Understanding and uniqueness of metaphors:

The head shake no metaphor is very understandable and cannot be confused with other facial metaphors, it is also unique in the sense that it does not overlap with another function or meaning within the program therefore cannot confuse the user.

-Avoid unmatched metaphors:

There is a strong correlation between the metaphor and what it's trying to convey to the user, getting opinions from other people will help see the strength of the design and whether anything need to be changed.

By breaking down the design process and referencing the methodology the newly designed icon hits all the points that make an effective and stylistic icon. The screenshots shown are the icons that came from the process described above. Also by using icons shown previous it strengthens the users mental connection between the icon and its function and helps keep the overarching

design theme. This icon can be seen in the final outcome when there is no new notifications for Trello's 'completed' board.



(Face looking at the Trello completed icon)(Shakes head right)(Shakes head left)

Chapter 5: Results and analysis

C++:
The results from qtcreator and using C++ for the project were sub-par, the desired outcome ended up being difficult and too tedious for what was necessary. Trying to achieve a http get request became too big of a problem so it was decided to look at other methods. A lot was learned about signals and slots but not much was learnt that would be put to use with regards to the final outcome of the project. C++ was going to be the foundation of the code but this quickly changed when python became more and more credible. It is important to understand when an idea is not going to work and that sinking time into it is not what is currently needed. At a certain point it is important to understand ones knowledge and ability and to channel that into a more certain solution than the one that is currently failing.

Arduino:
Researching the use of Arduino has resulted in some interesting possibilities, but would take too much time and effort from the current direction of the project and can be investigated more extensively if deemed necessary further down the line. The user following prototype talked about in implementation was not finished as time was restricted. It was deemed that the robot would move too much and cause too much of a distraction to the user. Also the user is not moving very often as they are sitting at a desk making it less practical. It may make the system feel more ‘alive’ as the face icon would be following you or it could move the screen left or right as a face icon on the screen is looking in the same direction.

Python:
Python has proven itself to be the most suitable language for the task at hand, and wxpython has been the best GUI library for python. This conclusion has been drawn after testing multiple forms of basic python and C++ programs and only making steady progress using python and the wxpython library. Python is the right choice of language for the project as it excels at api calling and quick condensed programming that gets the job done. You can argue that it does not perform as quickly and is not as robust as a C++ gui program, this is all depending on the circumstance it is used.
Wxpython is used as it requires no specific IDE to be downloaded only a library download from the CLI. This allows for easier use across multiple devices as it will need to be run on a raspberry pi for demos and the final poster presentation. Wxpython also has a class named wxCallLater(), this is essential for running the program on a continuous loop within the gui loop and not having the program lock up. It was very difficult to display gifs in the wxpython environment. After much research animated gifs were implemented and the progress made showed the value of the library and solidified it as the primary method of development.

Gmail:
As mentioned previously there was a lot of difficulty with the Gmail api, the main issue being that it could be run from terminal but not the IDE compiler. This is such an issue as the main solution is not being run from the terminal it is being run from the IDE, this causes a large split in development. It is always difficult knowing when to stop trying to solve a problem and knowing that it is only going to eat into other much needed time that could be better used polishing code and writing the report.

System analysis:
The system that was created does what it’s supposed to do and it does it very well, there seems to be no major issues with the constant running of the program. It is a functional based program that is run within a main loop that occurs within all gui based programs to keep the frame on screen at all times. It transitions from function to function performing checks and polling various api’s without faults. wxpython’s call later class saved a lot of tedious work, without it the system would have had to be run on two threads in parallel. There are some issues with wifi connection loss when polling api’s, if this happens it can cause the program to get stuck in an infinite loop. There are checks in place to stop this but it still occurs on some api calls. Each api has its own function where it will perform its necessary checks before calling the next, in some cases api’s have more than one function such as the Trello api. Trello has three functions, one for each list

that is checked. This is because if we needed to display more than one gif it would not work very well in conjunction with the wxcallLater method, so it is a lot simpler solution to do it over the three functions. Some may argue that the solution is simple, the goal was not to create a complex system it was to create an efficient, identifiable program that runs well and provides a great and memorable user experience.

Analysis of icons:
Overall the icons are a success, they present to the user well, they are clean and come across as a full set of icons that also stand out individually. It was mentioned in the goals that achieving simplicity was key, displaying everything without relying on solid information, it can be said that this goal has been achieved. (Yan R, 2011) States that icons that are intractable are beneficial as the user feels more of a participant. Although users cannot interact directly with the icons in this system, the animation of the icons help provide that feeling of communication that is needed to have the user engaging and having a favourable experience when using the system. If it was a cycle of still images the effect would have no where near the impact it does currently. The head animations are especially engaging as it provides that human element that would be lost if it was just a wait screen or only api cycle animations. They do a good job of filling the idle time, in which the user would be working and not viewing the companion, but if glanced at, it can provide them with some fun. All weather icons do a great job of showing the weather in a lively way. Animations and weather go hand in hand, using an animation gives so much more life to the weather icons and they are able to come across a lot more interesting than previously thought. With regards to the Twitter and Trello api, it was a lot harder to find a solution that would do the other icons justice, but branding and logo recognition became the solution to that issue. People who use the services will instantly recognise what the logo is and what is being conveyed. With Trello, colour coordination was used, this idea came about as when Trello is used boards always contain three lists at its core; incomplete, working on and completed and they use the common traffic light colour labelling which almost everyone can identify.

Students and friends experience:
When implementation of the design is done by a single person, the end goal can get lost and be confused with that persons vision leading to a stray in methodology and effect the final outcome in negative ways. That is why it is important to have feedback from fresh eyes or from people uninformed on the goals and project, they will hopefully give an honest opinion. Both fellow students and family members were asked to give comments on the system. The two groups of people were specifically chosen as students are more likely to give an opinion based upon prior knowledge of programming and system design. Friends and family are likely to be less informed on the technical aspects and give an opinion based upon usability and enjoyment. Feedback is focused on the overall system design and if it would used when it came to their workday. Below are some notable comments, names were not included for anonymity purposes:

“Animations are understandable and fun to look at, sounds would be good to direct peoples focus.”

“I think its good, there is the potential there to make it multifaceted. The good thing about it is that you can tailor it to the individuals needs and usages. I would enjoy having it on my desk and think it would benefit my workflow.”

“I use the services that are included in the program in my daily work, so it would be a great fit for me.”

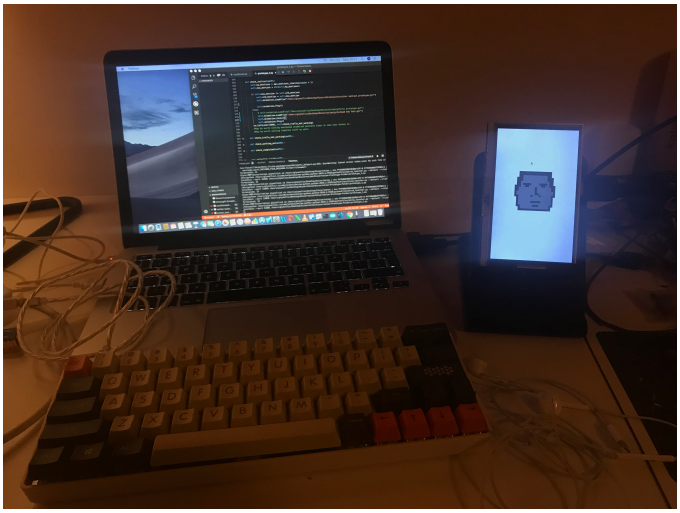
“I always find myself procrastinating when trying to work on assignments and such so this seems like a really good idea, I would maybe need a few more features but the icons look great.”

“I like the face, it adds a layer of character and makes it more personable.”

Chapter 6: Conclusion and future work

The end goal of the project was to create a desktop companion robot. Main sub-goals were listed at the start of the project, they included:
-A working desktop companion that can assist the user by displaying key information.
-Integration with robotic elements.
-A way of interacting with the user autonomously or on triggered actions.

A working desktop companion was created that assists the user, it displays key information in the form of identifiable icons. It interacts with the user in a way that requires no human input, it does this through animated icons, the face in particular provides for a memorable and meaningful interaction. Robotic elements were researched and investigated. It reached a point where prototypes were made, unfortunately they were not implemented in the final design due to time constraints, but if given more time they would have been used. By reflecting upon these goals and the final solution it is fair to say that the project can be deemed a success, the main goal has been achieved with two of the three sub-goals also being accomplished. The overall design has been implemented in a way that fits the introductions perceived outcome, it is simplistic and professional. When used correctly it will increase a users workflow by handling tedious tasks and presenting them in a fun way that will excite users.



(The companion robot sitting on a desk)
Near future work that can be implemented:
If given a few more weeks with the project it would have been great to fully incorporate the more simple robotic elements into the design, it would give it another layer of depth that would work well with the overall feel of the project. Things such as an led light system that is triggered depending on certain actions. Also having the robot physically look around room by use of servo would have been a great addition to the project and is not too complex that it would not take more than a week to implement. It would be implemented by using an ultrasonic sensor on a servo that is moved over a specified movement angle (180 degrees), similar to sonar. If it picks up an object close to it, it will lock on and face that direction for a period of time until looking around again. Time can be adjusted so that it is constantly looking for an object to follow. When an object is found the ‘search’ angle range will use the last know location as a centre point and work +/- degrees from that point.

Code features that could be implemented if given a few more weeks would be api’s that were researched and began development but had to be left due to time restrictions. Github connectivity was researched but not implemented, it would notify the user if a commit has been made or a comment and so on. A program was also made that checks a users inbox through the gmail api, the only problem is that it cannot be compiled from within the IDE only the terminal. If given more time it could have hopefully be implemented to work within the program. Messaging services would also be implemented, such as slack messenger as it is a professional service that

companies use within their internal organisation. Slack has an api and it would fit within the overall design of the program.

Future work that could be implemented:

The solution is very flexible to implementing new features due to its simplicity, that means it is open to a wide range of updates and design tweaks.

It is open to deal with as many api's as possible, if a user would like a service monitored for any updates and that service has an api it can be added to the solution. All that would be needed to be done is to update the code and create new icons. This also makes it a very open ended project as new services and api's are always being created.

If given more time with the solution more avenues could have been explored, such as an all text based approach. This would require a larger design as text has seemed very unfavourable when it is small (Zhao, Chen, Zhang, Tao, & Zhang, Kan. 2001), a new system would have to be designed to perform this task.

Including sounds would also be a feature that would have been implemented if more time was given, specific sound files would help further user understanding and identification of the icons. Sounds could have been used in conjunction with specific weather forecasts, emotions and known service noises (facebook message, email received, etc…). If sounds can be accurately identified then the user wouldn't even have to look at a screen to get their information.

This could be taken one step further and a system could be designed that is only audio based similar to the Amazon Alexa and Google home.

Complex robotic elements would also become a more viable option if given a lot more time, using an Arduino connected with various robotic elements such as a camera or microphone could assist the user in more ways and hopefully improve workflow.

A possibility is that it could become a system that is tailor based to the user, they input all services they use on startup and any codes/passwords and the program sets everything up for them ready to use. This is obviously very complex and would require putting in a lot of development time.

Appendix:

Sources:

Amazon.co.uk. (2019). *Amazon echo spot*. [online] Available at: https://www.amazon.co.uk/Amazon-Smart-Speaker-Screen-Alexa/dp/B01J2BK6CO [Accessed 26 Feb. 2019].

Descottes, J. (2019). *Piskel - Free online sprite editor*. [online] Piskelapp.com. Available at: https://www.piskelapp.com [Accessed 17 Jan. 2019].

Doc.qt.io. (2019). *Qt Documentation*. [online] Available at: http://doc.qt.io [Accessed 17 Feb. 2019].

Elegoo.com. (2019). *Elegoo Industries,Ingenious & fun DIY electronics and kits*. [online] Available at: https://www.elegoo.com [Accessed 7 Feb. 2019].

Google Store. (2019). *Google Home Hub*. [online] Available at: https://store.google.com/product/google_home_hub [Accessed 25 Feb. 2019].

Ko, P. (2012). Effects of Computer Display Design on Health and Productivity. 84.

Wiki.wxpython.org. (2019). *FrontPage - wxPyWiki*. [online] Available at: https://wiki.wxpython.org [Accessed 17 Jan. 2019].

Wood, W. (1960). *Image for Galaxy magazine*. [image] Available at: https://i.pinimg.com/originals/5f/48/f7/5f48f7871a1c48348ce3bde88cc8a026.jpg [Accessed 3 Jan. 2019].

Yan, R. (2011). Icon Design Study in Computer Interface. *Procedia Engineering, 15*, 3134-3138.

YouTube. (2019). *Qt Tutorials For Beginners 1 - Introduction*. [online] Available at: https://www.youtube.com/watch?v=EkjaiDsiM-Q&list=PLS1QulWo1RIZiBcTr5urECberTITj7gjA [Accessed 7 Feb. 2019].

Zapier. (2019). *What Are Webhooks?*. [online] Available at: https://zapier.com/blog/what-are-webhooks/ [Accessed 17 Feb. 2019].

Zhao, Chen, Zhang, Tao, & Zhang, Kan. (2001). User Interface Design for the Small Screen Display. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 45*(22), 1548-1550.

Weekly diary:

My initial thoughts:
When receiving the project I wasn't sure on what to do with it, I thought about an all digital version connected between 2 win form apps. But ultimately I was drawn to the idea of an actual physical robot that sits on your desk, I met with my project manager Christopher Tubb and he agreed with me.

My vision:
My end goal is to have a small screen with a disembodied head/face on it that is basically a personal assistant. I would like it to handle notifications such as received emails, tweets and messages, also have a clock/calendar system which can update you of important dates and the time and let you know if you should take a break. I would also like to introduce some robotic elements to it such as a small arm or have it sit on a servo and be able to "look" around the room.

Beginning:
I think the use of a raspberry pi3 and and a 3.5 lcd screen is a good starting point. I have made a wish-list of key features I think the robot needs to be functional and I'm going to start small and hopefully build up to something that I can be proud of.

First group meeting with Chris:
On the 10/10/2018 I attended the first group meeting with Chris, I saw other students projects and we all spoke a little about what our vision was and where we were and Chris gave his thoughts and input as well. During the meeting I had come up with a set of goals to complete in the coming week, they are:
-Download and create a basic program in QT creator.
-Look into starting the agile development method on my design process.
-Possible version control using Github.
-Start a Trello board.

Week 3:
I have found the website https://www.piskelapp.com and used it to create sprites of a face as I think I want to use it as the center of my robot. I have downloaded Qt creator and am completing basic tutorials to get a feel for the program.

Week 4:
I have now started a Trello board to organise the workflow and give myself a clearer idea of the direction i'm heading. I plan on using Trello and conducting weekly scrums/standups with myself to decide tasks that need to be completed per week, this will keep me on a straight path and hopefully not lose sight of the end goal. I'm also continuing with the qt tutorials and am making steady progress.

Week 5:
I'm slowly building a knowledge of the Qt environment from watching the tutorials and plan on hopefully starting a prototype next week. Work has also come to a dead stop due to a Qt update failing and deleting the exe file from the pc meaning I have to uninstall and reinstall the program again, this would not be an issue except my internet at home is extremely unreliable due to connection cuts and low speeds the download would take hours. An alternative is to buy a pi and program in Qt directly from the pi as it is a smaller download. My final project is based around displaying it on a pi and I feel as though it may be worth buying one myself sooner than waiting for one to become available through the university as it seems unsure that I may even get one at this time. I plan on giving it a few days focusing on other modules and then starting fresh.

Week 6:
No work was done due to focusing on other projects.

Week 7:
Had a meeting/lecture with Mike Ready where he discussed research methods and how to maximise successful academic searches using methods such as the CRAAP test and the universities resources (findit, bob…). The meeting lecture helped me understand my project better and how to get the best quality sources.

Weeks 8-12 + Christmas break:
Struggled with Qt creator get request, decided to investigate python possibilities in the cli. Downloaded a library named wxpython that allows for python gui programming. Currently on a prototype that gets information from a weather api and displays the correct gifs. Next step is running a multithreaded program that can handle multiple events as the program is currently locking itself.

Week 17(first week back):
The weekly report was not continued after Christmas term as it was used to ensure the project gained steam and work was not being neglected. After Christmas it was decided that the project was in such a flow that the weekly diary would not be kept but Trello usage and such will still be

ongoing. Below are some key occurrences that happened between week 17 and the milestone 3 hand in:

-Investigated threads as it is a way to solve the blocking issues within the program. By doing this found a wx python class named wx.CallLater() that accepts a time in milliseconds and a function call within the parameters. Using this allows me to constantly call functions at runtime, but there is an issue where it still locks up/freezes occasionally which needs to be looked into.

-Getting other api's working individually so that they can be added to the main program. Created a git repository for the project, will be using this for version/prototype control. Added a function that checks the wifi signal before requesting api information, was done for security reasons.

-Week 20 milestone 2 hand in.

-Added twitter api for functionality, checking mentions and displaying a twitter gif when you have been mentioned. Tweepy allows for the polling of 'n' items, therefore it was a case of polling the first item, saving it as oldest mention and then checking them against each other next poll. If they don't match then we have a new mention.

-Trello api integration, program interacts with three Trello lists notifying a user if an element is added to the list.

-Lots of meetings with Chris, discussing work and such.

-Lots of dissertation writing, research and general tinkering to the program.

-Icons were created following a specified design methodology found while researching.

-Other module assessments interfering with workflow and causing some issues.

-Raspberry pi would crash and run extremely slow and the idea had to be abandoned due to time constraints. Exporting the program as an exe was investigated but there were too many dependencies to make putting development time in worthwhile.

Compiling and running the code:

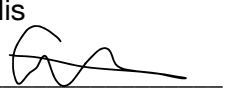
It was advised to not include the code in the appendix but to include it in the submission, this is being done but there are few steps that need to be taken to compile and run the project.
All the gifs included are taken from a file and require the full path when calling them, this will be different when run on different machines therefore all gif paths will need to be changed within the program.
Also a few libraries must be installed from the terminal as they are not included in basic python.
-Tweepy
-Wxpython
-Pytrello
Expect some issues when running this on another machine, if there are any problems please contact 13197061@students.southwales.ac.uk.

Ethics form:

This form is only applicable for assessed exercises that use other people ('participants') for the collection of information, typically in getting comments about a system or a system design, or getting information about how a system could be used, or evaluating a working system.

If your proposed activity does not comply with any one or more of the points below then please contact your project supervisor and/or project coordinator for advice. If your evaluation does comply with all the points below, please sign this form and submit it with your assessed work.

1. Participants were not exposed to any risks greater than those encountered in their normal working life. Investigators have a responsibility to protect participants from physical and mental harm during the investigation. The risk of harm must be no greater than in ordinary life. Areas of potential risk that require ethical approval include, but are not limited to, investigations that occur outside usual laboratory areas, or that require participant mobility (e.g. walking, running, use of public transport), unusual or repetitive activity or movement, that use sensory deprivation (e.g. ear plugs or blindfolds), bright or flashing lights, loud or disorienting noises, smell, taste, vibration, or force feedback.
2. The experimental materials were paper-based, or comprised software running on standard hardware. Participants should not be exposed to any risks associated with the use of non-standard equipment: anything other than pen-and-paper, standard PCs, mobile phones and PDAs.
3. All participants explicitly stated that they agreed to take part, and that their data could be used in the project. If the results of the evaluation are likely to be used beyond the term of the project (for example, the software is to be deployed, or the data is to be published), then signed consent is necessary. A separate consent form should be signed by each participant. Otherwise, verbal consent is sufficient, and should be explicitly requested in the introductory script.
4. No incentives were offered to the participants. The payment of participants must not be used to induce them to risk harm beyond that which they risk without payment in their normal lifestyle.
5. No information about the evaluation or materials was intentionally withheld from the participants. Withholding information or misleading participants is unacceptable if participants are likely to object or show unease when debriefed.
6. No participant was under the age of 16. Parental consent is required for participants under the age of 16.
7. No participant has an impairment that may limit their understanding or communication. Additional consent is required for participants with impairments.
8. Neither I nor my supervisor is in a position of authority or influence over any of the participants. A position of authority or influence over any participant must not be allowed to pressurise participants to take part in, or remain in, any experiment.
9. All participants were informed that they could withdraw at any time. All participants have the right to withdraw at any time during the investigation. They should be told this in the introductory script.
10. All participants have been informed of my contact details. All participants must be able to contact the investigator after the investigation. They should be given the details of both student and module co-ordinator or supervisor as part of the debriefing.
11. The evaluation was discussed with all the participants at the end of the session, and all participants had the opportunity to ask questions. The student must provide the participants with sufficient information in the debriefing to enable them to understand the nature of the investigation.
12. All the data collected from the participants is stored in an anonymous form. All participant data (hard-copy and soft-copy) should be stored securely, and in anonymous form.

Student Name: Glyn Ellis
Student ID: 13197061
Student's Signature: 
Date: 29/03/2019