Project Outline

Using software development principles and practices to develop software that can be utilized to effectively manage addictions to software by employing behavioural modification techniques.

By Muhammed Khidr

Supervisor: Stilianos Vidalis

**Project:**

The project that will be undertaken is Using software development principles and practices to develop software that can be utilized to effectively manage addictions to software by employing behavioural modification techniques.

**Project Hypothesis:** The problem at hand is addictions to software. Specifically for this project I will be targeting Desktop software addictions because Mobile software addiction has been tackled both by manufacturers and independent developers; both native and open-source implementations (Example being Android 11 which natively introduced setting application usage limits as well as apps on the Play Store like Quitzilla). The hypothesis for this project is: By utilizing software development principles and methodologies to devise an automated Software solution which implements behavioural reinforcement techniques, it will effectively reduce the addictive component of utilizing said software in the long term safely.

**Project Motivation:**

The inspiration behind this project is an epiphany I had whilst on my commute to university one morning, as I stood aboard the train I glanced around and noticed that without fail every person in this carriage(myself included) was utilising a piece of technology; be it a Phone or a laptop, and most likely have done so unconsciously out of habitual instinct for wanting to pass time until they reach their destination. The reason why I chose this as my project is because a vast number of people unconsciously are addicted to certain software programs and they lack the cognition to admit it, and the reason for this is due to society normalizing excessive technology usage, therefore my project helps to aid those who are suffering in an effective and accessible way. The reason why I have chosen to target desktop specifically is because there have already been implementations for software addiction control on Mobile platform.

**Project Aims and Objectives:**

The aim for this project is to have a functional application that will implement a behaviour modification technique correctly and effectively to treat addictions to desktop software applications.

The development objectives that I would like to achieve are the application GUI being visually pleasing and designed to promote usability and accessibility, the application should be cross platform to run on multiple desktop operating systems; Linux, Windows, Mac OS, I would like to create a functional menu system which offers additional functionality such as a tutorial page as well as serve navigation purposes.

The research objectives that I would like to complete are furthering my understanding and application of the principles and practices of GUI based software development, to discover and familiarise myself with tools for development and project management, additionally I want to become more familiar with the components of addiction as well as the causes and treatments available.

The evaluation objective of the project with regards to determining if the project is successful or not is to test over a given period whether the usage of said application is decreased in the test subject over the duration of the evaluation period, if it has then that would deem the project to be a success and conversely.

Additional personal objectives that I would like to achieve are improving my time and project management skills regarding delivering work, I aim to improve my test design and testing ability as these are highly sought after in the tech industry.

**Resource Analysis:**

Python and Tkinter: The first available resource to complete the project’s GUI is one that I am familiar which is Tkinter. Tkinter Is a GUI Library that is included within the default installation of python, meaning that no external installation of modules or packages is required.

The advantages of using Python and Tkinter are due to having prior experience with them there is a less time to make a start on the development which is beneficial is it allows for more time to spend on biger tasks such as design and prototyping and less on learning the library. Tkinter is included in the python installation and is lightweight therefore there are less dependencies on the project and therefore the final application will have low storage requirements and low performance requirements allowing for a range of system specifications to be compatible. Tkinter is also known to be extremely stable and flexible and has a simple learning curve to master. Additionally, Tkinter is used in many open-source projects because it is also open source and therefore there is no cost or licensing fee. (Moore.A.D, 2018, Python GUI Programming with Tkinter: Develop responsive and powerful GUI)

However, the disadvantages are; Simplicity, it is not possible to create advanced widgets and vector graphics, nor is it possible to create high end GUI’s which have data driven views and modern multimedia integration, so there are aesthetic limitations as well as complexity of design limitations.

Java FX GUI Library:

The next option to consider is using Java FX, I do have some prior experience with Java FX albeit much less in comparison to Tkinter. Java FX is also included within the standard API library of a Java installation hence there is no requirement for external modules or software.

The biggest advantage in favour of using JavaFX is that unlike many GUI libraries, JavaFX is cross compatible and cross operating system compliant. Due to the nature of Java being a cross system language, JavaFX inherits this property and therefore if the artefact is created for desktop, it can also be used on Mobile with no alterations needed which allows the project to be available to another userbase, but now the project is solely for desktop environments. JavaFX is also easy to use and works in many ways like Tkinter and therefore has a shallow learning curve whilst maintaining enough functionality to where it is still popular in industry applications. (Clarke. J, Connors. J, Bruno. E.J, 2009, JavaFX: Developing rich internet applications)

But JavaFX is limited in complexity; like Tkinter there are limits to the visual complexity of the User interface which is a result of JavaFX being older than a lot of newer GUI libraries, also the JavaFX community is very small in comparison to Tkinter and QT which means there is less available guides and tutorials, there aren’t many forums to which users can ask and answer question regarding JavaFX, there is a somewhat sizeable overhead due to the Java runtime environment and virtual machine which everything is ran on and therefore the final application would be much larger in storage size and ram usage in comparison to other tools.

Qt Framework with C++:

Qt is an open-source framework that is built on C++ and uses C++ to create enterprise level Graphical user interface applications, it is widely used in the tech industry because of its extensive inclusion of tools for creating visually stunning applications for a range of platforms; including Desktop, Mobile, and Automobiles.

The benefits to creating the artefact with Qt are the opportunity to create an enterprise level complex interface which has the potential for multi-faceted functionality and usability, Qt unlike JavaFX and Tkinter can create modern-esque applications which are aesthetically pleasing through the importing of complex multimedia and allowing the creation of vector graphics and immensely customizable widgets. Additionally, QT has many packages and can create applications for a range of platforms and operating systems hence achieving one of the objectives of the project, furthermore QT has a decently sized forum where users can ask questions and provide solutions and information to problems and help troubleshoot.

(Summerfield. M, 2010, Advanced QT Programming….)

However, Qt has a large performance and storage overhead resulting in the application being harder to use and run, hence affecting accessibility. There is a steep learning curve to using Qt which means that it will be harder to make a start on the development phase and that more time will have to allocated to the development and the testing procedure to ensure by the end of the project there is a functioning artefact which isn’t a guarantee, additionally Qt is also very complex to use with excessive functionality and in my personal opinion is over-engineered and sophisticated for this project to meet the current aims and objectives.

Currently, I am leaning towards using Tkinter and Python due to its high capability despite having low hardware requirements, hence making for an appropriate and accessible solution.

In regards to the usage of other tools which will not be used in the main application development of the solution, there are many online tools available for prototyping the GUI like Figma, which is free to use and allows for creating very detailed prototypes which can then be easily implemented hence why it is my resource of choice for the prototype phase, for creating UML diagrams such as the use case and class diagrams, I will use SmartDraw which is another free to access web tool which allows for easy and free creation.

**Project Outline:**

The proposed project methodology is a Waterfall approach, and the reason’s supporting this choice are there is a fixed deadline for which the project needs to be completed by, the deliverables of the project are already defined and immutable, there is no commercial userbase, therefore the agile approach of constant revision and adapting to user feedback is unnecessary. Additionally, during the artefacts construction the waterfall method is flexible to allow for cohesion between development and testing.

The initial plan for the project is to segregate tasks into 2 categories: Major and minor, tasks which are deemed to be major will have a maximum of 2 weeks dedicated to them and minor tasks 1 week, however as the project progresses time allocation will be more flexible. The critical path for the project is:

* Start collecting sources of information relevant to the subject matter and that which is relevant contextually and make a note of these sources for later use as part of the Bibliography.
* Start reviewing the sources for the Literature review, begin by skim reading and trying to identify what is pertinent and then delve deeper into the pertinent information.
* Research about behavioural modification methods to treat addiction and identify the chosen method to implement in the artefact.
* Investigate the various tools available to create the GUI application on Desktop, evaluate the options and make a choice to use and research the documentation for said choice.
* Research and define an evaluation method.
* Start of the prototyping phase; experiment with the GUI’s creation and develop a prototype like a Wireframe to plan the look of the GUI, the widgets that will be used and where they will be located.
* Development of Use cases and class diagrams, identifying the required functionality and actors.
* Development of back-end classes and functions needed by the application.
* Back-end code review
* Development of the front-end GUI of the application
* Front-end code review
* Development of User Guide
* Ongoing regression testing of the application GUI and Back end
* Implementing unit tests for GUI and Back end
* Fixing Bugs and Final code review
* Development of documentation explaining the system’s functionality
* Developing a FAQ / Tutorial inside the application
* Evaluation of the project and whether the aims and objectives were met, scrutinizing the design, the code, and the choices made. Evaluation will involve a Self-evaluation of me using the application to control my Video game habits over a 2-week period.
* End of project reflections; review feedback from the evaluation, discussing what could have been done differently and what I would do if given another opportunity.

The critical path for the project is: reviewing literature to further understand the difficulties of addictions; the difficulties being the causes and the effectiveness of available treatments. The next stage in the path is researching the available tools to utilise in the creation of the User interface and back-end classes, followed by the creation of a Wireframe/Prototype which will feed into the Front-end development of the application, followed by the Unit testing stage. The success of the unit tests will decide whether more development is needed, and finally the evaluation of the projects design choices and whether the objectives were met will be the final stage of the critical path. Failure or delay in completion in any stage of the critical path will detrimentally affect the project, thereby each critical stage task will be allocated 2 weeks for completion and depending on progress further time may be allocated.

Each task is typically linked to a task above it, for example, the creation of the applications Graphical user interface cannot be completed without the production of the Wireframe of the application as it is the basis of the design and without it a functional GUI is unlikely to be produced, additionally the Unit tests cannot be implemented before the Test plan is designed and identified, the literature review cannot occur without the collection of sources to create the bibliography. Except for the first task, each task has a pre-requisite task that must be completed which shows the project is structured sequentially and tasks build on top of one another, hence every task mentioned in the Project outline will produce an output be it a prototype or an idea, that will then be used in another task as an input. A further example of this is the back end development of the application which involves the creation of the classes, the identifying of the relevant functions needed as well as the implementation cannot be done without developing the use case and class diagrams in the pre development phase.

Given that my project does not make use of many resources besides free and widely available tools, there is unlikely to be any resource shortfalls or delays caused by inaccessibility of resources.

**GANTT Chart:**

Pictured below is a Gantt chart detailing the project, the tasks with the shorter sized bar indicate a Minor task where 1 week of time is allocated for, the larger bars indicate a major task where 2 weeks of time is allocated for said task. The tasks are colour coded to group them together, for example: Green represents development tasks, Red represents testing, purple evaluation, and blue planning.

A graph with multiple colored lines

Description automatically generated with medium confidence

**Sustainability Analysis:**

Given my project does not utilise any real-world resources as the final artefact is a Software application, there are no environmental or sustainability concerns. With that said, I will optimise the application to use limited hardware resources so there is minimal power draw while using it.

**Project resources checklist:**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Will a specialist laboratory PC be required for the project?  If YES, please specify below: | YES | NO |
| 2 | If the answer to 1 is Yes, will you require any special hardware installed/attached?  If YES, please specify below: | YES | NO |
| 3 | Do you require access to specialist systems and/or hardware (Database Servers, Network Hardware, Robotics, etc)?  If YES, please specify below: | YES | NO |
| 4 | Will you require specific software?  If YES, please specify below: | YES | NO |
| 5 | Are there other resources required, not covered above?  If YES, please specify below: | YES | NO |

**Ethics checklist:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ethical Issues** | | | |
| **1** | Will your work involve human subjects, e.g., survey, interview, experiment, etc. This includes friends and family. | YES | NO |

**Risk register:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Associated Risk | Mitigation Process | Likelihood | Severity |
| Utilising information from Literature review that is un-verified or inaccurate to derive conclusions. | High | Being thorough and using only verified academic papers and articles in reputable Scientific journals or organisations. | Medium | High |
| Poorly designed GUI which affects accessibility and usability. | Medium | Plan the design of the GUI prior to building it, by using Wireframes to layout the buttons and widgets. | High | High |
| Security Vulnerabilities in the Application allowing for unintended access | High | Sanitise any user input, make sure to only take input from user if it is necessary, employ defensive programming strategies | Low | High |
| The project does not meet it’s intended objectives and aims | Medium | Constantly refer to the established aims and objectives for feedback and guidance with building the project. | Medum | Medium |
| Project is unfinished and no functional artefact is developed | Low | Ensure a schedule is developed and followed and that tasks and time is managed properly to allow for a functional artefact at the end. | Low | High |
| Badly written code resulting in Severe issues resulting in a non-functioning product. | Medium | Implementation of testing methods such as Regression testing and unit tests to test functions are implemented and correctly and detect bugs to patch. | Medium | High |
| Missing functionality which is a result of limitations of ability. | Medium | Thorough planning and review of the aims and objectives to ensure everything that is required is implemented correctly. | Medium | Medium |
| Badly designed unit test cases which allow for Bugs to go unnoticed. | High | Prior to starting development create an exhaustive list of all the required functionality to derive all the required tests. | Medium | Medium |

Reference List/ Bibliography:

* (Moore. A.D, 2018, Python GUI Programming with Tkinter: Develop responsive and powerful GUI)
* (Clarke. J, Connors. J, Bruno. E.J, 2009, JavaFX: Developing rich internet applications)
* (Summerfield. M, 2010, Advanced QT Programming…. )
* (Heather. N, 2017, Rethinking Addiction, British Psychology Society)
* (Griffiths. M, 1995, Technological Addictions, Clinical Psychology Forum)
* (Griffiths. M, 1993, Fruit machine addictions in Adolescence)
* (Savci. M, Aysan. F, 2017, Technological addictions and social connectedness: …)
* (Sherer. J, Levounis . P, 2022, Technological Addictions)
* (Vohs and Baumeister, 2009, Addiction and Free Will)
* (Kant. H, 2009, What Neurobiology cannot tell us about addiction)
* (Clay. SW, Allen. J, Parran. T, 2008, A review of addiction, Postgraduate Medicine)
* (Kadden. M.R, 1994, Cognitive behavioural approaches to Alcoholism Treatment)
* (Oxford University Press, 2020)