Electronics and Computer Science
Faculty of Physical and Applied Sciences
University of Southampton

Hannah Alice Short

12.12.17

A Graphical Programming E-Learning Platform Employing Gamification and Software Visualisation

Project supervisor: Dr Su White
Second examiner: Dr Paul Chappell

A project progress report submitted for the award of
BSc Computer Science

Abstract

Despite the growth in computing course enrolments over the past decade, there still exists an unfavourable stigma in society on adopting programming as a skill. Although it is a competence accompanied by multiple benefits, such as improving logical and technical abilities, many believe it is too complex and time-consuming to learn and generally exerts little appeal. It is necessary that an innovative method of integrating this skill into wider populations is found as companies become ever more dependent on individuals who possess software development knowledge.

This report gives an overview of the project to be completed, detailing the goals it aims to achieve and exploring the work and research that has already been carried out in the domains of e-Learning, gamification and software visualisation. Existing platforms are reviewed to evaluate what is currently available, the progress of the design and planning phases of the proposed application is discussed as well as the remaining work to be completed.

Contents

1 Project Goals

The overall goal of this project is to create an engaging e-Learning platform that is able to provide a basic knowledge of programming to beginners or simply individuals who are looking to reinforce their exiting knowledge. The sub-goals of this project are as follows:

1. **Ensure the platform accentuates software visualisation**
   Effective graphical representations will be used in order to illustrate concepts with the aim of minimising the quantity of theoretical information administered to the user
2. **Software clearly conveys fundamental programming and algorithmic concepts**
   The platform will endeavour to deliver content in two domains. Firstly, programming concepts including formatting, naming convention and decomposition as well as algorithmic concepts such as looping, conditions, data structures and types of searching and sorting algorithms. This is so a well-rounded knowledge can be provided and ensure good practise is taught.
3. **Deploy gamification strategies with an aim to drive participant motivation**
   Another factor that will be incorporated to boost user engagement will be gamification techniques such as rewards and milestones as these extrinsic values are likely to increase motivation
4. **Providing a suitable amount of guidance before allowing users to attempt tasks**
   It is important to allow users to have freedom to answer questions and try techniques for themselves after having followed tutorials so they are able to consolidate their knowledge and understanding. The sufficient amount of guidance given should be determined depending on the difficulty of the concept being taught

2 Background and Report of Literature Search

Although computer programming is widely recognised as a difficult competence to attain, it is one that is proven to be invaluable (Fesakis and Serafeim, 2009) and comes with an ever growing demand. The problem faced in the computing domain is discovering a way in which to advertise programming as an attractive skill to learn while teaching it credibly and conveying it in the simplest manner that can be achieved.

Literature Review

Causes of Negligence towards a Computing Education

Despite the growing need for computer science and software engineering skills in many of today's industries, a negative stereotype is still projected onto programming education. This stereotype warrants society deterring from feeling able to interact with code (Charters et al., 2014) and expanding their logical skill set.

Rogerson and Scott (2010) have investigated students whom have reported fearing learning to program and have concluded the main reasons behind this are the complex thinking process needed, the inability to adopt a reflective manner when working and the lack of personalisation in this sort of educational environment. For this reason, introductory programming courses have lead learners to face difficulties and deficiencies resulting in high dropout rates (Robins, Rountree and Rountree, 2003).

Another point that can be considered to contribute to the cause of negligence is the prominent gender imbalance in the technology sector. The proportion of women occupying technical computing jobs is low and appears unlikely to change due to conventions in popular culture (Abbate, 2012). This is also reflected in the learner demographic of computing.

Online Platform Engagement and MOOCs

Although the emergence of numerous online classroom environments over the last decade now means it is accessible and convenient to take up academic courses and expand skill portfolios, there is a fear that this has led to distanced learning and such environments provide a lack of interaction (McBrien, Jones and Cheng, 2009). Despite this concern, there have been results showing the positive correlation between learning on the web and engagement levels, students enrolling on online courses are more likely to possess a higher level of practicality and stronger personal growth. (Chen, Lambert and Guidry, 2010).

Massive Open Online Courses (MOOCs) enable people from any type of background to freely build upon their education at their own convenience, it is thought that the reason behind MOOC user engagement being rated so highly is due to the student's freedom to only pursue areas that reflect upon their personal interests and goals. It has been noted also that overall engagement is higher yet still when peers enrol together due to the increased social factor (Barak, Watted and Haick, 2016).

Learning Management Systems

As well as focusing on the content that is presented on e-Learning sites, the learning design and sequencing of activities should be reviewed. This aspect is covered by learning management systems, these are virtual environments that can include assignment management, discussion forums and sharing of files. There exists speculation that they are not widely used among student populations as alternative social software tools are favoured (Dalsgaard, 2006). Researchers deem an LMS to be the initial focal point when creating online platforms and courses as they can act as a basis in the delivery of instructions and participant outcomes. (Cavus, Uzunboylu and Ibrahim, 2007)

Application of Gamification Strategies

Gamification can be defined as 'the use of game design elements in non-game contexts' (Deterding et al., 2011, p.12), by developing an active learning environment for a student, their aspiration to be successful will be increased from new found engagement (Dicheva et al., 2014).

Pedreira et al. (2015) have evaluated a number of applications incorporating gamification and have identified that the following gamification elements should be implemented into such an application to make it effective: point-based systems, dashboards, levels and rewards. The increased use of gamification in e-Learning for higher education will not only raise levels of student satisfaction and motivation but also allow for a personalised learning environment to be created (Urh et al., 2015)

Christopher Su (2016) compares the classroom environment to that of an escape room in the sense that a student is forced to recall information, communicate ideas and work under pressure. This notion contributes to the structural overlap of games and education, the basis of gamification. He claims that when creating a platform utilising gamification, you should set initial content before game design methods are laid over the top.

Gamification guru Yu-Kai Chou (2014) has developed a framework, branded Octalysis. He argues that gamification should not be based on game design elements but rather the following eight core drives: Meaning, Accomplishment, Empowerment, Ownership, Social Influence, Scarcity, Unpredictability and Avoidance. He allocated each of these to an edge in the octagonal framework (shown below) and explains these can be divided into two groups in either one of two ways. Firstly, excluding meaning and avoidance, the octagon can be split vertically with drives representing extrinsic tendencies on the left and intrinsic tendencies on the right. The other way they can be grouped is by excluding ownership and social influence and splitting the octagon horizontally with 'white hat' gamification techniques on top and 'black hat' gamification techniques on the bottom. White hat techniques are positive and give a sense of growth, improvement, creativity and control while 'black hat' techniques are not so sustainable, a user will just execute an action to see what the outcome is without the belief it can deliver achievement.

Source: http://yukaichou.com/books

Software Visualisation

A comprehensive description of software visualisation can be given as 'the utilisation of graphical representations in form of text, pictures and animations' (Olsson, Mozellus and Collin, 2015, p. 442). Software visualisation can ameliorate the acceleration of programming pedagogy, enabling learners to obtain knowledge without the obligation to process large amounts of theoretical information. Pacione, Roper and Wood (2004) have highlighted the need for visual programming in object-oriented languages in particular due to the complexity of the interactions made in the code.

Olsson, Mozellus and Collin (2015) conducted a study in which they selected two techniques that were considered difficult for novice programmers to digest – for loops in an imperative language and the concepts of instantiation, inheritance and method calls relating to objects in an object oriented language. First year undergrads were divided and each allocated to one of the techniques, they were then divided again into control and experimental groups, the control groups were given a theoretical lecture explaining their concept whilst the lectures given to the experimental groups were reinforced with software visualisation. The results from this study found that 68% of the experimental group and 39% of the control group found that how they understood the dynamics of the technique were helped while 85% of the experimental group and 62% of the control group maintained a consistent understanding of where the execution of the code was focussed. These results show the scale the impact software visualisation can have in conveying seemingly complex ideas.

Petre and Blackwell (1999) discovered that mental imagery plays a big role in program design, expert programmers are found to make use of mental images so as to produce a clear simulation of program behaviour. This was also true when it came to software design, programmers would envisage structures and layouts in their mind before attempting to give a description or recording notes relating to what they were designing.

Graphical Programming

By using pictorial and graphical representations, a substantial level of information can be communicated, however this comes with the responsibility of recognising the way in which a taught concept, such as the execution of a program, will be perceived (Petre, 1995). Educating an individual via the means of graphical representations can offer an attractive alternative to simple text-based channel teaching methods.

Carlisle et al. (2005) designed RAPTOR, 'the Rapid Algorithmic Prototyping Tool for Ordered Reasoning' as they believed that when students were required to understand the flow of an algorithm, their attention was diverted instead to the complex syntax of the language. Thus compromising the development of their problem solving skills and hence showing the benefits of displaying a program graphically to direct focus onto solely the functionality of algorithms.

Existing Systems

I have explored a variety of already available e-Learning platforms in order to address their features and gauge what is effective in this type of an environment. Sites such as www.khanacademy.org (Khan Academy) and www.w3schools.com (W3Schools) provide in depth tutorials to help students learn and reinforce skills but both possess a lack of interactive elements.

The web-based game codecombat.com (Code Combat) requires players to write code in order to rank up through levels while also successfully teaching necessary foundations of computer science. It provides the interactivity that perhaps other platforms lack but is only directed at a child demographic.

Requirements Survey

I have created an online survey in order to elicit attitudes towards learning to program and also suggestions for the proposed application. This is composed of ten questions varying from Likert scale response questions such as 'how much easier do you find visual learning as opposed to learning through theory' to long answer response questions such as 'what are your initial thoughts on/how do you view the notion of programming?'. The full survey can be seen in Appendix 1. The design of this survey was influenced by John Creswell (2013), a professor specialising in educational research design.

3 Design and Planning

It is important to note that my design and planning work still remains in its initial stages, this is due to choosing to undertake my requirements survey over the Christmas period which will be influencing both the design and structure of my system. After I have collected the data, I will add to and refine my existing diagrams and wireframes and develop more realistic use cases.

Requirements

I have drawn up some initial requirements (Appendix 2) that the system will need to fulfil in order to be successful. Once I have gathered responses from potential users in my survey, I will alter and add to these so they become more representative of what users would like to see in such a platform.

Use Case Diagrams

I have created three prospective use cases for my system along with three accompanying simple use case diagrams to illustrate proposed user interactions with the system. These can be seen in appendix 3.

UI Design

I have preliminarily designed two of the main screens involved in the platform as basic wireframes in Balsamiq (Appendix 4), the first I have designed is the home screen, this is where users will be able to select options, minimally view their progress as well as access tutorials and corresponding quizzes to level up on the platform. The second is how a tutorial screen will be expected to look - program code will be stepped through and a user can opt to go back to a previous step or skip to the next depending on how they find the difficulty of the content.

Risk Assessment

I have produced a risk assessment (Appendix 5) to outline potential risks that could occur over the duration of my project. For each, I have estimated using a numbered scale the probability they will occur and the severity of the risk, from this I have calculated the product representing to what extent they would affect the success of the project if they were to occur. Mitigation and management methods have also been noted for each so there are recognised means of prevention as well as plans in place in the event of the risk arising.

Gantt Charts

To estimate the timings of tasks included within my project and to track the progress being made, I have used Gantt charts. Before developing the project brief, I made an initial Gantt chart (Appendix 7), highlighting all the tasks that needed to be achieved across the scope of the project and any intermediate deadlines. Upon writing my progress report, I have created a second Gantt chart (Appendix 8) where I have adjusted the timings of these tasks and updated the progress that has already been made.

4 Report on Technical Progress

Unfortunately, I have not made as much progress as I would have liked to have made up to this point in the project schedule. I believe the main cause of this is taking other modules this semester that are particularly coursework intensive and not completely accounting for this when initially allocating timings to the project. Next semester, the project will be taking up two module spaces and the additional modules I have chosen include little coursework in their assessment so I should have much more time to allocate to the remaining work.

Thus far I have completed the majority of my literature review, attempting to read from all the disciplines that my project is likely to cover, although as the process progresses, it is likely that I will read some additional literature to gain a more concrete background in areas such as graphical programming.

After this, I began looking into and reviewing existing systems in order to evaluate what works well and what could be changed for improvement in similar e-Learning platforms as well as producing my requirements survey in order to gather perspectives from individuals who have no or little experience in a programming environment.

I underestimated the time it would take to create my survey and complete an application for ethical approval and therefore this process has been pushed backwards with my data collection taking place over the Christmas break, which can be seen in my revised Gantt chart (Appendix 7).

I have created some initial simple use case diagrams and UI wireframes to form a basis for the design of my system although these will be enhanced and updated once my data has been analysed.

5 Plan of Remaining Work

I have opted to work over the Christmas period although the tasks I will be working on will be completed at a much slower rate. From my Gantt chart detailing the work I am to do from December until the deadline of the project (Appendix 7), it is shown that I plan to conduct my survey and data collection as well as finish the last few bits of the design and planning section so that upon returning, I am able to start the development phase of the project.

I have also opted to work over the exam period but again this will be leisurely due to having to accommodate for revision. I have made this decision as I was not content with the idea of dropping the project for a period of almost a month. My development stage will start during this time whereby I will begin developing the interfaces of my system. Hopefully the end of this stage of implementation will tie in with the end of the exam period and I can then start working on the back-end features such as handling question data and developing a point scoring system. I will be developing my platform in Java as this is the language I am most familiar with and will be using SQL for keeping a store of application questions and responses, these are recognised in the skills audit I conducted (Appendix 6). My software will be implemented using the MVC framework in order to have clean separation between my interfaces, data store, backend and logical components.

I plan to start the testing and evaluation phases a few weeks before the Easter break in order to dedicate the entire break to writing up my final report so it is ready for submission early in the case that any risks occur before the final deadline.

References

Abbate, J. (2012) *Recoding Gender: Women's Changing Participation in Computing*, Cambridge, MIT Press

Barak, M. Watted, A. Haick, H. (2016) Motivation to learn in massive open online courses: Examining aspects of language and social engagement, Computers & Education, Vol. 94, pp. 49-60, Available from: http://www.sciencedirect.com/science/article/pii/S0360131515300828 [Accessed 6th December 2017]

Carlisle, M. C. Wilson, T. A. Humphries, J. W. Hadfield, S. M. (2005) RAPTOR: a visual programming environment for teaching algorithmic problem solving, *Proceedings of the 36th SIGCSE technical symposium on Computer science education - SIGCSE '05*, Vol. 37, Issue: 1, pp. 176-180, Available from: https://dl.acm.org/citation.cfm?id=1047411 [Accessed 8th December 2017]

Cavus, N. Uzunboylu, H. Ibrahim, D. (2007) Assessing the success rate of students using a learning management system together with a collaborative tool in web-based teaching of programming languages, Journal of educational computing research, Vol. 36, Issue: 3, pp. 301-321, Available from: http://journals.sagepub.com/doi/pdf/10.2190/T728-G676-4N18-6871 [Accessed 9th December 2017]

Charters, P. Lee, M. J. Ko, A. J. Loksa, D. (2014) Challenging stereotypes and changing attitudes: the effect of a brief programming encounter on adults' attitudes toward programming, *Proceedings of the 45th ACM technical symposium on Computer science education – SIGCSE '14*, pp.653-658, Available from: https://dl.acm.org/citation.cfm?id=2538938 [Accessed 8th December 2017]

Chen, P. D. Lambert, A. D. Guidry, K. R. (2010) Engaging online learners: The impact of Web-based learning technology on college student engagement, *Computers & Education*, Vol. 54, Issue: 4, pp. 1222-1232, Available from: http://www.sciencedirect.com/science/article/pii/S0360131509003285 [Accessed 7th December 2017]

Chou, Y. (2014, February 28) Gamification to improve our world: Yu-kai Chou at TEDxLausanne [Video file]. Retrieved from http://yukaichou.com/chou-musings/tedx-gamification-change-world/

Code Combat, CodeCombat - Learn how to code by playing a game, Available from: https://codecombat.com/

Creswell, J. (2013) *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, SAGE Publications Inc.

Dalsgaard, C. (2006). Social software: E-learning beyond learning management systems. *European Journal of Open, Distance and ELearning*

Deterding, S. Khaled, R. Nacke, L. Dixon, D. (2011) Gamification: toward a definition, *Chi 2011*, pp. 12-15, Available from: http://gamification-research.org/wp-content/uploads/2011/04/02-Deterding-Khaled-Nacke-Dixon.pdf [Accessed 6th December 2017]

Dicheva, D. Irwin, K. Dichev, C. Talasila, S (2014) A course gamification platform supporting student motivation and engagement, *2014 International Conference on Web and Open Access to Learning, ICWOAL 2014*, Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7009214 [Accessed 9th December 2017]

Fesakis, G. Serafeim, K. (2009) Influence of the familiarization with "scratch" on future teachers' opinions and attitudes about programming and ICT in education, *Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education – ITiCSE '09*, Vol. 41, Issue: 3, pp.258-262, Available from: https://dl.acm.org/citation.cfm?id=1562957 [Accessed 8th December 2017]

Khan Academy, Khan Academy | Free Online Courses, Lessons & Practice, Available from: https://www.khanacademy.org/

McBrien, J. L. Jones, P. Cheng, R. (2009) Virtual Spaces: Employing a Synchronous Online Classroom to Facilitate Student Engagement in Online Learning, *The International Review of Research in Open and Distributed Learning*, Vol. 10, Issue: 3, pp. 1-17, Available from: http://www.irrodl.org/index.php/irrodl/article/view/605/1298 [Accessed 8th December 2017]

Olsson, M. Mozellus, P. Collin, J. (2015) Visualisation & Gamification of e-Learning & Programming Education, *The Electronic Journal of e-Learning*, Vol. 13, Issue: 6, pp. 441-454, Available from www.ejel.org [Accessed 21st September 2017]

Pacione, M. Roper, M. Wood, M. (2004) A novel software visualisation model to support software comprehension, *Proceedings - Working Conference on Reverse Engineering, WCRE*, pp. 70-79, Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1374307 [Accessed 21st September 2017]

Pedreira, O. Garcia, F. Brisaboa, N, Piattini, M. (2014) Gamification in software engineering – A systematic mapping, *Information and Software Technology*, Vol. 57, pp. 157-168, Available from: https://ac.els-cdn.com/S0950584914001980/ [Accessed 14th October 2017]

Petre, M. (1995) Why looking isn't always seeing: readership skills and graphical programming, *Communications of the ACM*, Vol. 38, Issue: 6, pp. 33-44, Available from: http://delivery.acm.org/10.1145/210000/203251/ [Accessed 8th December 2017]

Petre, M. Blackwell, A. F. (1999) Mental imagery in program design and visual programming, *International Journal of Human-Computer Studies*, Vol. 51, pp. 7-30, Available from: https://ac.els-cdn.com [Accessed 8th December 2017]

Robins, A. Rountree, J. Rountree, N. (2003) Learning and Teaching Programming: A Review and Discussion, *Computer Science Education*, Vol. 13, Issue: 2, pp. 137-172, Available from: https://doi.org/10.1076/csed.13.2.137.14200 [Accessed 8th December 2017]

Rogerson, C. Scott, E. (2010) The Fear Factor : How It Affects Students Learning to Program in a Tertiary Environment, *Journal of Information Technology Education*, Vol. 9, pp.147-171, Available from: https://www.learntechlib.org/p/111361/ [Accessed 6th December 2017]

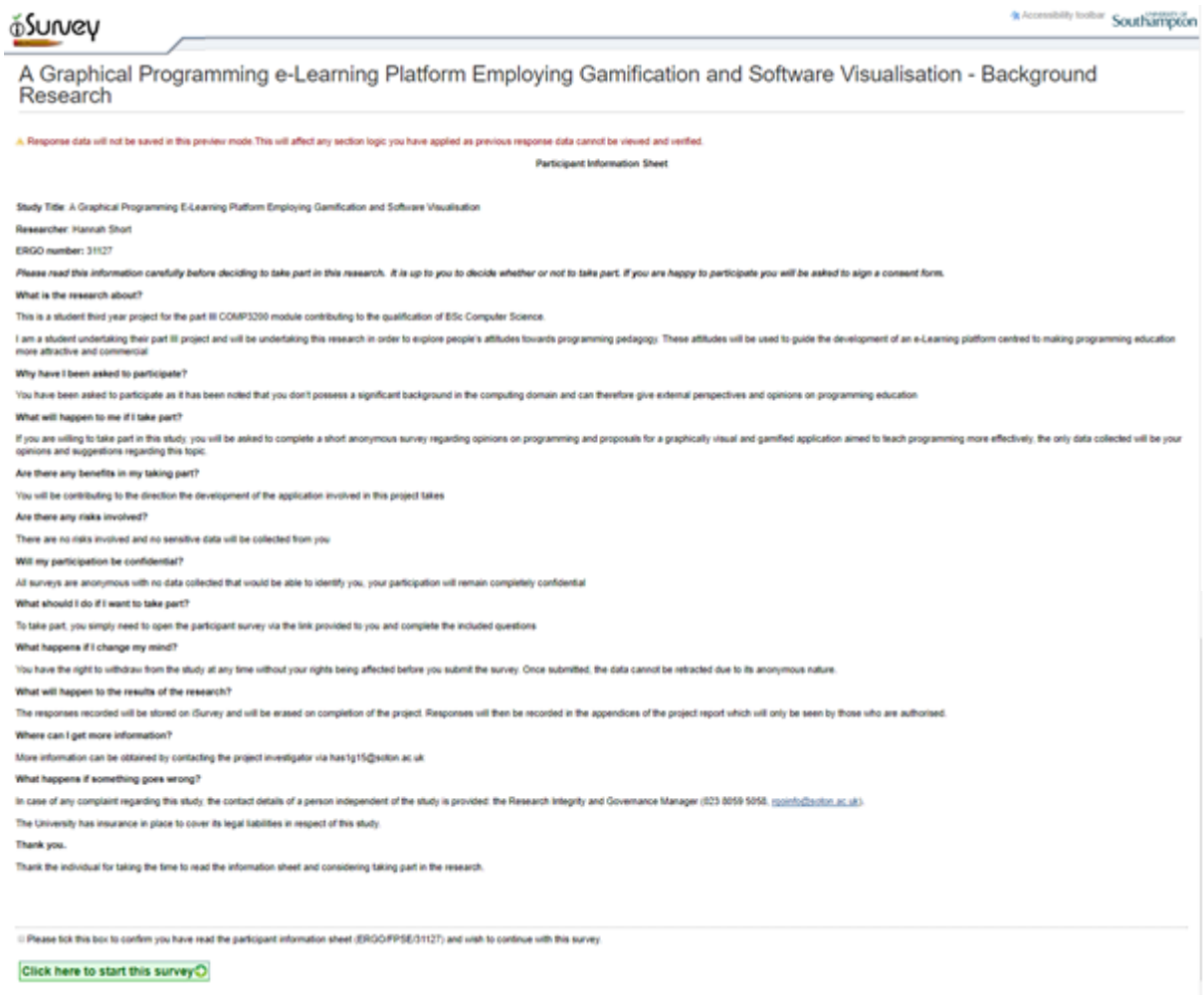
See, C. (2017, April 24) Gamification in Higher Education - TEDxCUHK [Video file]. Retrieved from https://www.youtube.com/watch?v=d8s3kZz1yQ4

Urh, M. Vukovic, G. Jereb, E. Pintar, R. (2015) The Model for Introduction of Gamification into E-learning in Higher Education, *Procedia - Social and Behavioral Sciences*, Vol. 197, pp. 388-397, Available from: https://ac.els-cdn.com/S1877042815041555/ [Accessed 7th December 2017]

W3Schools, W3Schools Online Web Tutorials, Available from: https://www.w3schools.com/

Appendices

Appendix 1: Survey Design



Survey

Accessibility toolbar University of Southampton

A Graphical Programming e-Learning Platform Employing Gamification and Software Visualisation - Background Research

⚠ Response data will not be saved in this preview mode.This will affect any section logic you have applied as previous response data cannot be viewed and verified.

Participant Information Sheet

Study Title: A Graphical Programming E-Learning Platform Employing Gamification and Software Visualisation

Researcher: Hannah Short

ERGO number: 31127

Please read this information carefully before deciding to take part in this research. It is up to you to decide whether or not to take part. If you are happy to participate you will be asked to sign a consent form.

What is the research about?

This is a student third year project for the part III COMP3200 module contributing to the qualification of BSc Computer Science.

I am a student undertaking their part III project and will be undertaking this research in order to explore people's attitudes towards programming pedagogy. These attitudes will be used to guide the development of an e-Learning platform centred to making programming education more attractive and commercial.

Why have I been asked to participate?

You have been asked to participate as it has been noted that you don't possess a significant background in the computing domain and can therefore give external perspectives and opinions on programming education.

What will happen to me if I take part?

If you are willing to take part in this study, you will be asked to complete a short anonymous survey regarding opinions on programming and proposals for a graphically visual and gamified application aimed to teach programming more effectively, the only data collected will be your opinions and suggestions regarding this topic.

Are there any benefits in my taking part?

You will be contributing to the direction the development of the application involved in this project takes

Are there any risks involved?

There are no risks involved and no sensitive data will be collected from you

Will my participation be confidential?

All surveys are anonymous with no data collected that would be able to identify you, your participation will remain completely confidential

What should I do if I want to take part?

To take part, you simply need to open the participant survey via the link provided to you and complete the included questions

What happens if I change my mind?

You have the right to withdraw from the study at any time without your rights being affected before you submit the survey. Once submitted, the data cannot be retracted due to its anonymous nature.

What will happen to the results of the research?

The responses recorded will be stored on iSurvey and will be erased on completion of the project. Responses will then be recorded in the appendices of the project report which will only be seen by those who are authorised.

Where can I get more information?

More information can be obtained by contacting the project investigator via has1g15@soton.ac.uk

What happens if something goes wrong?

In case of any complaint regarding this study, the contact details of a person independent of the study is provided: the Research Integrity and Governance Manager (023 8059 5058, rgoinfo@soton.ac.uk).

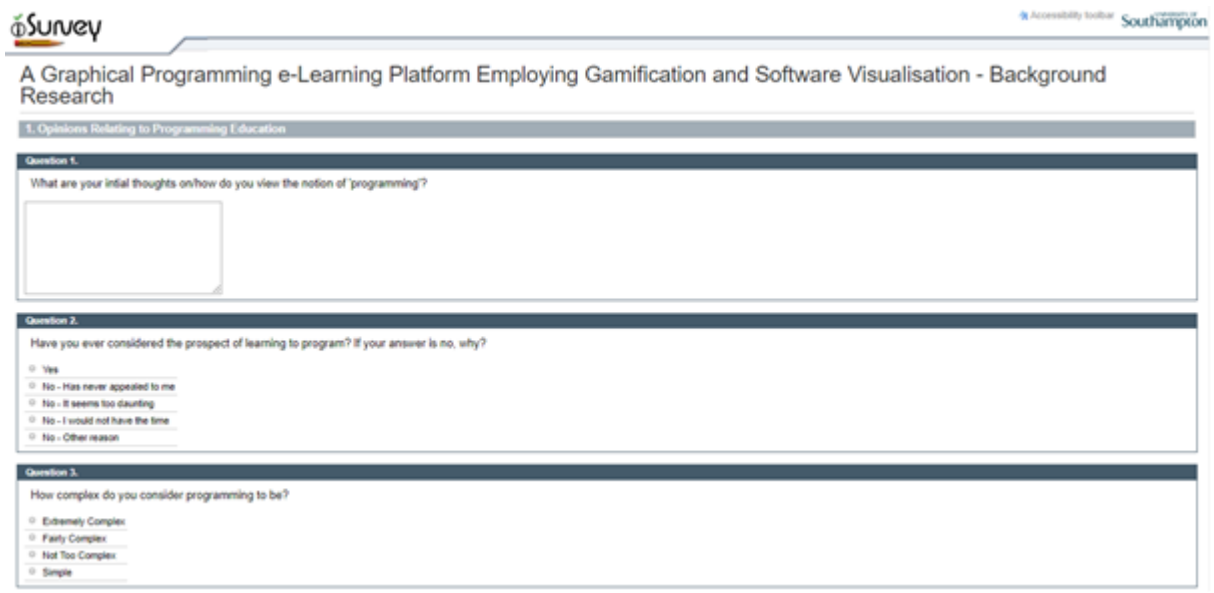
The University has insurance in place to cover its legal liabilities in respect of this study.

Thank you.

Thank the individual for taking the time to read the information sheet and considering taking part in the research.

☐ Please tick this box to confirm you have read the participant information sheet (ERGO/FPSE/31127) and wish to continue with this survey.

Click here to start this survey



Survey

Accessibility toolbar University of Southampton

A Graphical Programming e-Learning Platform Employing Gamification and Software Visualisation - Background Research

1. Opinions Relating to Programming Education

Question 1.

What are your initial thoughts on/how do you view the notion of 'programming'?

Question 2.

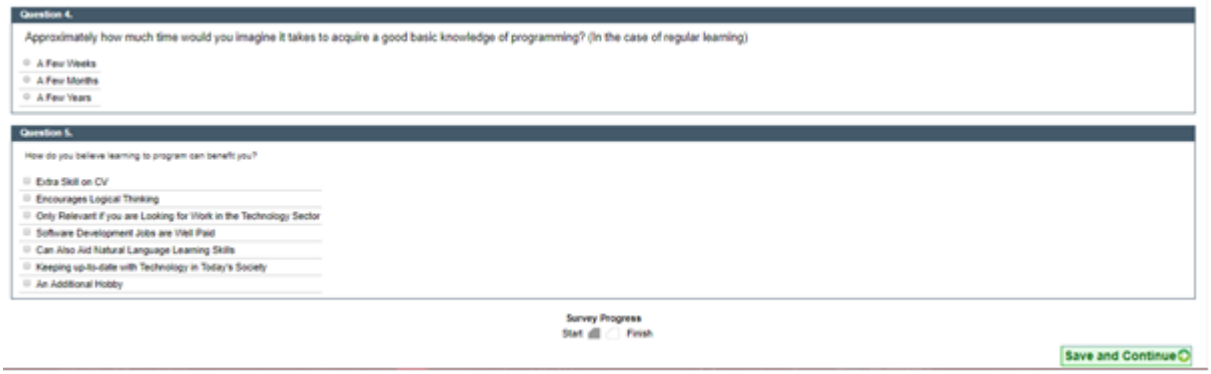
Have you ever considered the prospect of learning to program? If your answer is no, why?

- ○ Yes
- ○ No - Has never appealed to me
- ○ No - It seems too daunting
- ○ No - I would not have the time
- ○ No - Other reason

Question 3.

How complex do you consider programming to be?

- ○ Extremely Complex
- ○ Fairly Complex
- ○ Not Too Complex
- ○ Simple



Question 4.

Approximately how much time would you imagine it takes to acquire a good basic knowledge of programming? (In the case of regular learning)

- A Few Weeks
- A Few Months
- A Few Years

Question 5.

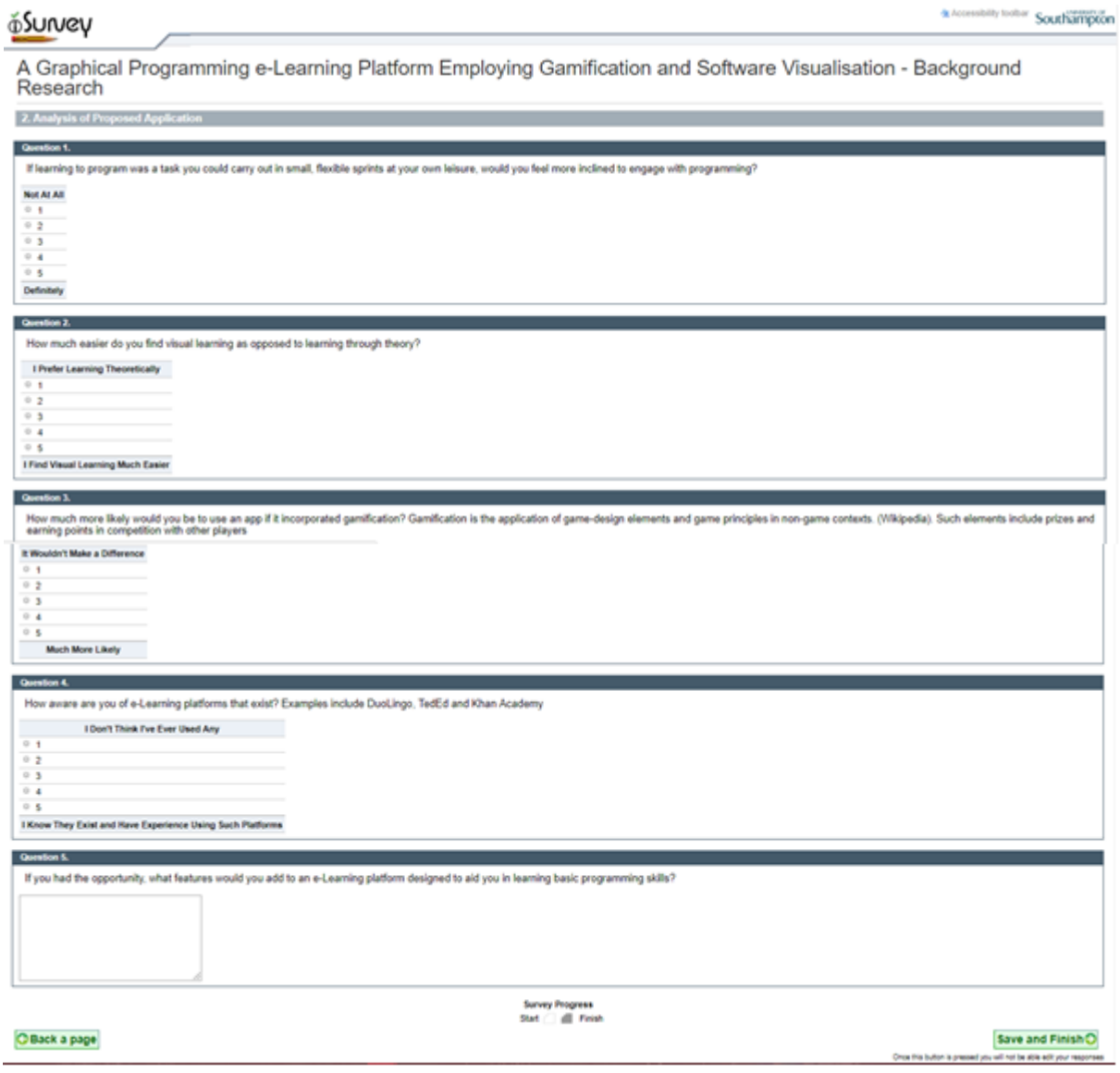
How do you believe learning to program can benefit you?

- Extra Skill on CV
- Encourages Logical Thinking
- Only Relevant if you are Looking for Work in the Technology Sector
- Software Development Jobs are Well Paid
- Can Also Aid Natural Language Learning Skills
- Keeping up-to-date with Technology in Today's Society
- An Additional Hobby

Survey Progress

Start Finish

Save and Continue



iSurvey

Accessibility toolbar University of Southampton

A Graphical Programming e-Learning Platform Employing Gamification and Software Visualisation - Background Research

2. Analysis of Proposed Application

Question 1.

If learning to program was a task you could carry out in small, flexible sprints at your own leisure, would you feel more inclined to engage with programming?

Not At All

- 1
- 2
- 3
- 4
- 5

Definitely

Question 2.

How much easier do you find visual learning as opposed to learning through theory?

I Prefer Learning Theoretically

- 1
- 2
- 3
- 4
- 5

I Find Visual Learning Much Easier

Question 3.

How much more likely would you be to use an app if it incorporated gamification? Gamification is the application of game-design elements and game principles in non-game contexts. (Wikipedia). Such elements include prizes and earning points in competition with other players

It Wouldn't Make a Difference

- 1
- 2
- 3
- 4
- 5

Much More Likely

Question 4.

How aware are you of e-Learning platforms that exist? Examples include DuoLingo, TedEd and Khan Academy

I Don't Think I've Ever Used Any

- 1
- 2
- 3
- 4
- 5

I Know They Exist and Have Experience Using Such Platforms

Question 5.

If you had the opportunity, what features would you add to an e-Learning platform designed to aid you in learning basic programming skills?

Survey Progress

Start Finish

Back a page

Save and Finish

Once this button is pressed you will not be able edit your responses

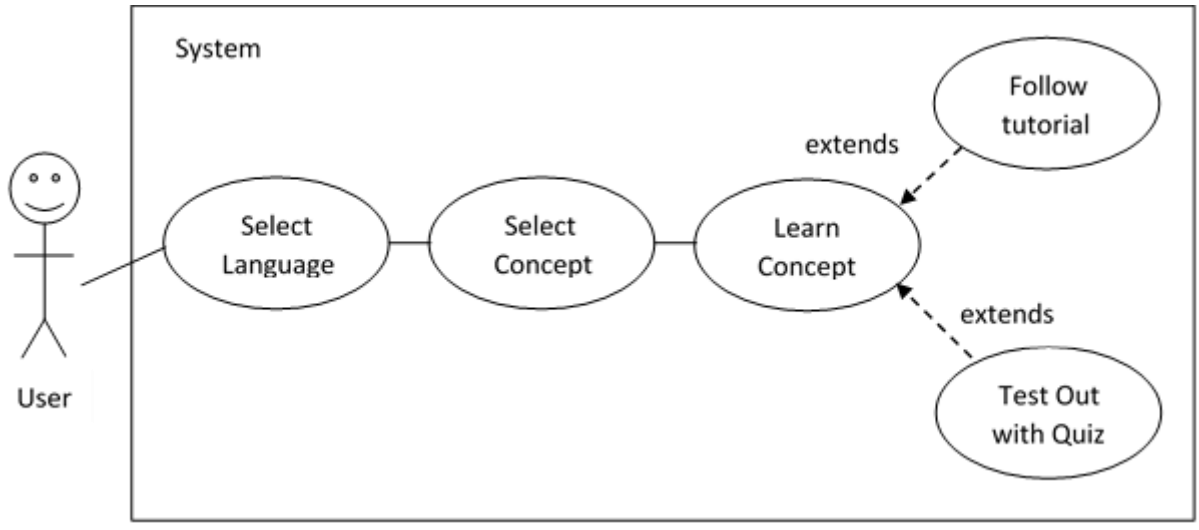
Appendix 2: Requirements Elicitation

1. Users must be able to view their statistics and progress
2. Users must be able to easily forward and rewind during tutorials
3. Users must be given unlimited time to complete test quizzes
4. Users must be able to configure settings such as colour schemes and resetting progress
5. The tutorial content should maximise the user's confidence when testing out of a skill
6. User data must be stored if they choose to exit the application
7. Users must be able to test straight out of a skill if they feel they can do so
8. Users must receive points from completing questions and assessments
9. Users must be able to improve their level based on the number of points they've earned
10. The graphics used should give the user a clear depiction of code execution

Appendix 3: Use Case Diagrams

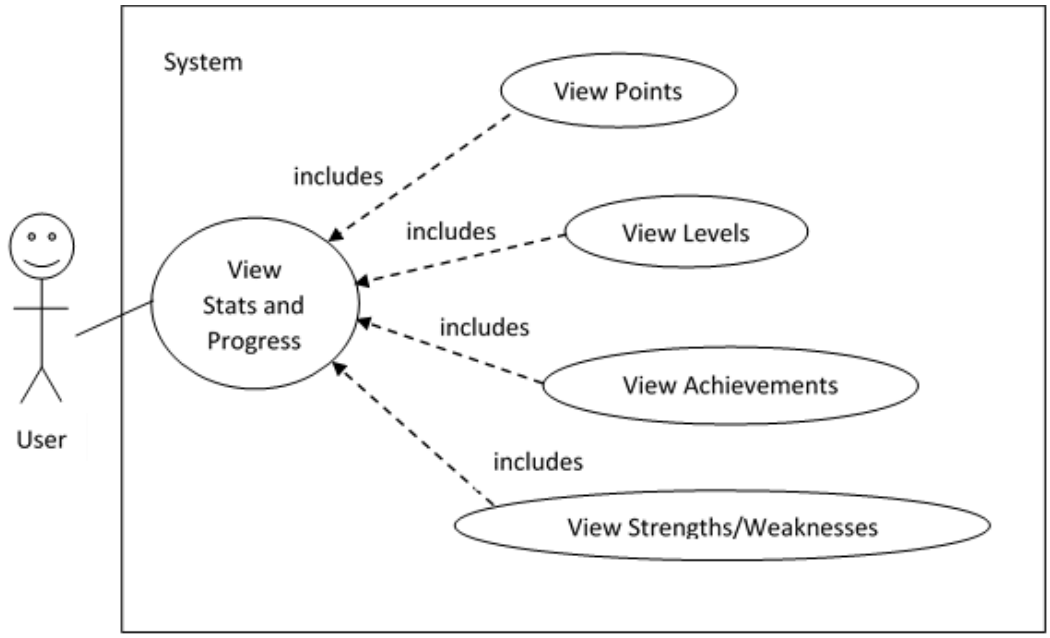
User wants to learn the concept of inheritance in Java:

- User opens home screen of platform
- User selects language as Java
- User selects skill as object orientation
- User views object orientation skill page
- User accesses tutorial on page
- User skips to inheritance section of tutorial
- User watches a graphical visualisation of inheritance with objects



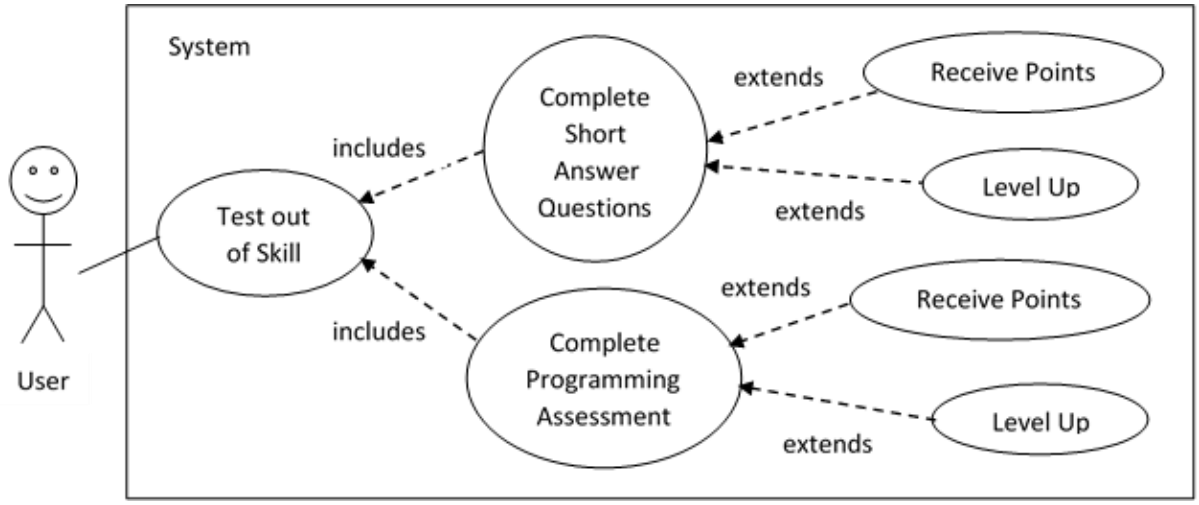
User wants to see the number of points they need to level up

- From home screen, user selects 'View Stats' button
- Stats for user are loaded and their current points are shown
- From progress bar, user can see how much more progress they need to make
- User starts to look for quizzes they can take to get to the next level



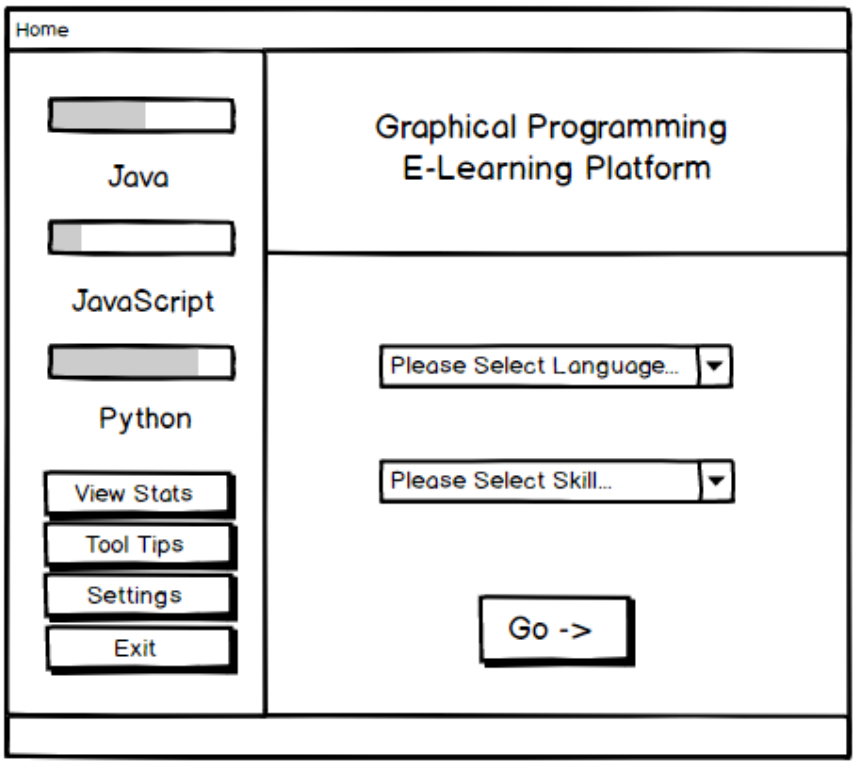
User wants to complete 'While loops in Python' in order to reach the next level

- After completing relevant tutorial, user selects 'take quiz' option
- User completes a set of short response questions
- User completes a short programming assessment
- User clicks finish
- Points from quiz are calculated
- If user has gained enough, they will receive a message saying they have levelled up
- Stats page will be updated to show new level

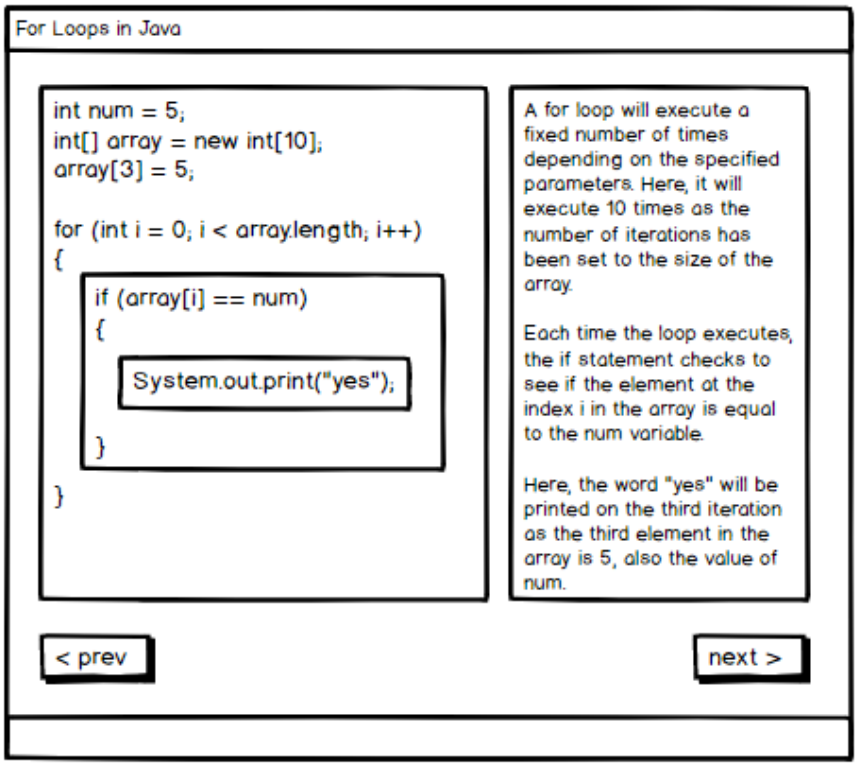


Appendix 4: User Interface Design

4.1 Home Screen



4.2 Tutorial Page



Appendix 5: Risk Assessment

Risk	Probability (1-5)	Severity (1-5)	Exposure (P*S)	Mitigation and Management Methods
Hardware Failure	1	3	**3**	I will be working predominantly on my laptop and desktop for the duration of this project. I will ensure they are protected and will not be running any unnecessary software. In the case of hardware failing, I will keep backups of all my project work using cloud-based solutions and external drives if required
Software Failure	1	2	**2**	I will be using software that I know is reliable such as IntelliJ for Java implementation and Balsamiq for drawing up wireframes. Should any of the software I am utilising fail, I will endeavour to use an alternative or use the same software on a different machine.
Data and Documentation Loss	1	5	**5**	All the work I produce will be stored either in the cloud or on a drive. My code and question store will all be hosted in a GitHub repository, my designs will be stored on cloud-based applications like Balsamiq and my documentation will be regularly saved on my laptop and desktop as well as backed up on an external drive which will be kept in an alternative location to the main hardware I am using. In the unlikely event of losing such data, I will attempt to access previous versions and if this is not possible, adjust my plan in order to accommodate for re-doing the affected piece of work

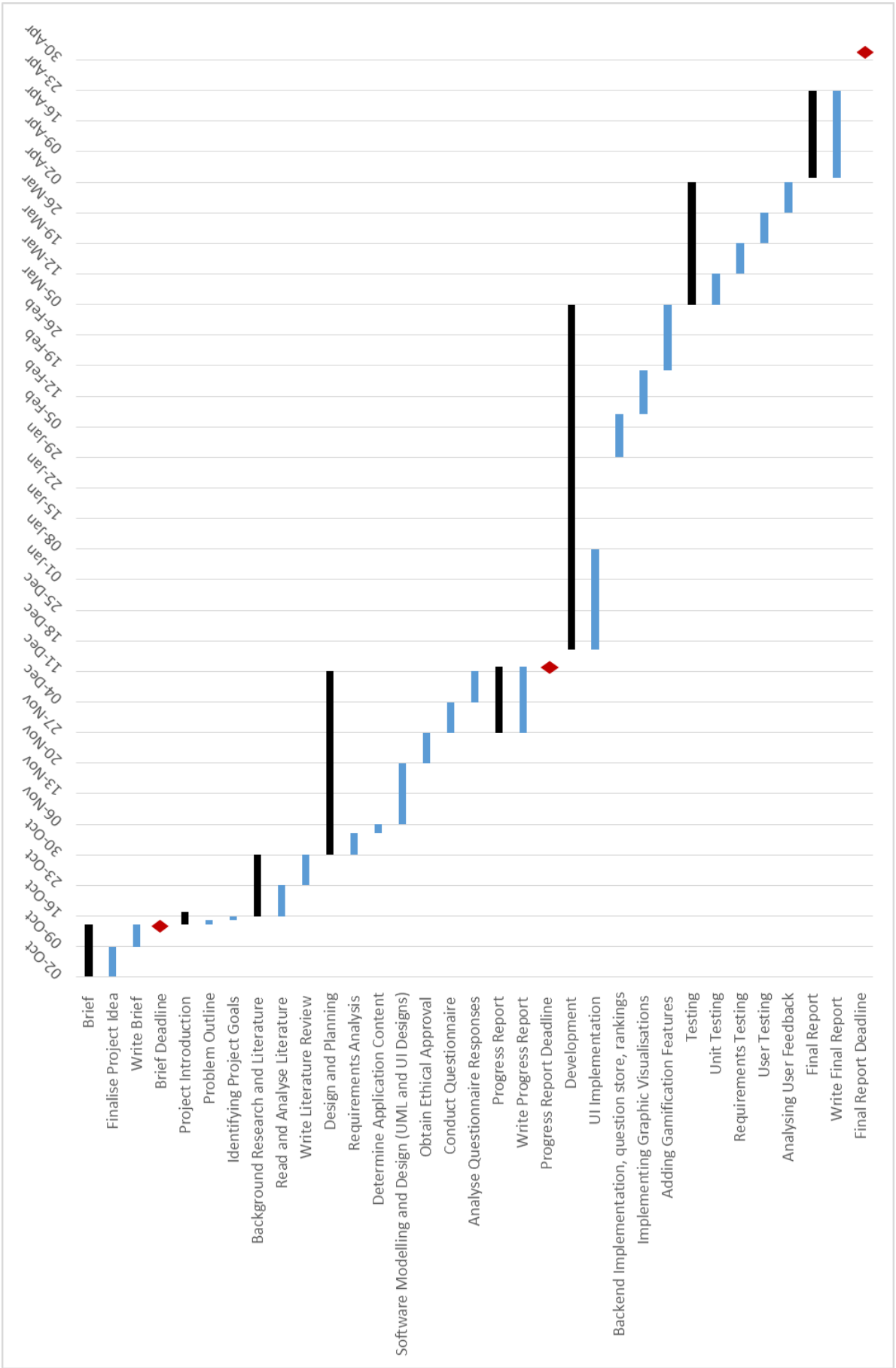
Minimal Participation in User Background and User Feedback Stages	4	3	**12**	The survey that I will be using to reinforce my background research will be shared on social media and through word of mouth in order to reach a wide audience and elicit the required number of participants. Regarding the feedback stage, I will take a smaller proportion of the initial participants meaning if somebody is not available, I can ask the next available candidate. However, if I find that I have a lack of participation, I will personally approach individuals to make certain they are aware of the study. Failing this, I will have a smaller sample of users in my project.
Sickness	2	3	**6**	Throughout the duration of the project, I will attempt to have a good daily routine and be mentally prepared for the forthcoming work. If I am marginally sick, I will try to carry on with the project and recognise that it will take longer to complete tasks. In extreme circumstances, I will contact my supervisor, tutor and ECS senior tutor to make them aware of my situation.
Supervisor Sickness	2	2	**4**	This is a risk that is out of my control but to ensure that exposure is as small as it can be, I will make sure I approach my supervisor when needed as early as possible. Alternatively, I can contact the COMP3200 module leader for project queries.

Failing to Meet Project Deadline	2	5	**10**	I will plan my time as efficiently as I can across the project period. I will also over allocate time in the case of something going wrong so I have ample time to rectify this. If I find that I am struggling to complete all the work I had set out to do coming up to the deadline, I will reduce the number of features in my application in order to have enough time to complete the report, this can then be evaluated in the report.
Struggling with Technical Aspects	2	3	**6**	I have conducted a skills audit to see where my technical weaknesses lie, I will allocate more time to these aspects as it is likely I will have to learn skills alongside carrying out the work If I find a technical aspect such as programming in a particular language too difficult, I will ask a peer on the course, seek support from the programming help desk or look for an alternative means of completing the task, i.e. using a different language
Struggling with Report Writing	3	3	**9**	Likewise, with technical aspects, report writing has been addressed in my skills audit, I have noted that it is not one of my stronger skills and I will therefore allocate a lot of time to writing the report. If I am struggling a great deal, I will ask peers for advice in structuring the report and ask people if they will proof-read what I have written to check for spelling and grammar errors

Appendix 6: Skills Audit

Skills	Rating (1-5)
Technical	
Java	4
SQLite	3
UML	4
UI Design	5
Survey Design	2
Testing	4
Communication & Analysis	
Report Writing	2
Reviewing Literature	4
Evaluation	3

Appendix 7: Initial Gantt Chart



Appendix 8: Revised Gantt Chart (From December)