

An Interactive Web Application to motivate learning of programming

GYOKAY ALI

Banner ID 000762233

Supervisor: **YASMINE ARAFA**

Final Year Individual Project COMP1682

A dissertation submitted in partial fulfilment of the University of Greenwich undergraduate degree programme
BSc (Hons) Software Engineering

18th April 2016

Word count: **11,919 (not counting preamble and appendices)**

ABSTRACT

Programming is the language of the future and preparing students to this future is crucial, because there is no other solution other than learning to program. This project has a mission to investigate and help to understand the key motivating factors affecting learning of programming among first year undergraduate students. Why students are afraid of programming in such level? Is it too difficult or just lack of motivation? What is the solution to this problem?

The outcome of this piece of paper finds an answer to these aforementioned questions by conducting a deep research into past literature by credible authors and sources who tried to solve or highlighting the same problem. Different approaches to motivate students into programming will be discussed in detail. The most substantial factors will be taken under serious consideration and applied into a real application.

As a result an interactive web application will be developed to support this project and hopefully it will find a place in practice to motivate students to program in a more fun and engaging way.

ACKNOWLEDGEMENTS

I would like to take this opportunity to show great appreciation to all the support and help given to me during this project's lifecycle.

I would like to thank my supervisor Dr Yasmine Arafa for her commitment, advices and support and co-supervisor Dr Markus Wolf for his excellent lectures in ASP.NET that helped me a lot.

Also, I am very grateful for the support of my family and my girlfriend.

CONTENTS

| | | |
|-------|---|----|
| 1 | Introduction | 7 |
| 1.1 | Background Information | 7 |
| 1.2 | Project Scope..... | 7 |
| 1.3 | Project Aim..... | 8 |
| 1.4 | Project Objectives | 8 |
| 1.5 | Project Deliverables | 9 |
| 1.6 | Project Methodology..... | 9 |
| 2 | The Feasibility Study..... | 10 |
| 3 | Literature Review | 11 |
| 3.1 | Introduction..... | 11 |
| 3.2 | Difficulty of learning to program..... | 11 |
| 3.3 | Motivation..... | 13 |
| 3.3.1 | Key factors of motivation | 13 |
| 3.3.2 | Different approaches to motivate students..... | 15 |
| 3.4 | Conclusion | 17 |
| 4 | Legal, Social, Ethical and Professional Issues and Considerations | 19 |
| 4.1 | Legal | 19 |
| 4.2 | Social..... | 19 |
| 4.3 | Ethical | 20 |
| 4.4 | Professional..... | 20 |
| 5 | Review of Existing Products | 21 |
| 5.1 | Introduction..... | 21 |
| 5.2 | Product 1 - CodeCombat..... | 21 |
| 5.2.1 | Review | 21 |
| 5.2.2 | Appearance | 22 |
| 5.3 | Product 2 – CodinGame | 22 |
| 5.3.1 | Review | 22 |
| 5.3.2 | Appearance | 23 |
| 5.4 | Product 3 – CodeMonkey | 24 |
| 5.4.1 | Review | 24 |
| 5.4.2 | Appearance | 25 |
| 5.5 | Key issues to use in the design and implementation | 25 |
| 5.6 | Conclusions | 26 |
| 6 | Human Computer Interaction | 27 |

| | | |
|--------------|--|----|
| 6.1 | Introduction..... | 27 |
| 6.2 | HCI..... | 27 |
| 6.3 | Accessibility..... | 27 |
| 6.4 | Usability | 28 |
| 7 | Requirements Analysis..... | 29 |
| 7.1 | Introduction..... | 29 |
| 7.2 | Target audience..... | 29 |
| 7.3 | Statement of requirements..... | 29 |
| 7.3.1 | Functional requirements..... | 30 |
| 7.3.2 | Non-functional requirements | 30 |
| 8 | Design of The Product..... | 32 |
| 8.1 | Entity Relationship Diagram (ERD) | 32 |
| 8.2 | UML Use-Case Diagram | 33 |
| 8.3 | UML Class Diagram..... | 34 |
| 8.4 | UML Sequence Diagrams | 35 |
| 8.4.1 | Take Quiz SD..... | 35 |
| 8.4.2 | Play Game SD..... | 36 |
| 8.5 | Site-Map Diagram..... | 37 |
| 8.6 | Initial Sketches..... | 37 |
| 8.7 | Prototypes..... | 37 |
| 9 | Development of The Product..... | 38 |
| 9.1 | Introduction..... | 38 |
| 9.2 | The Process of Implementation..... | 38 |
| 9.3 | Changes made to Prototype C..... | 39 |
| 9.4 | Technical problems | 40 |
| 9.5 | Extra about Prototype 3..... | 40 |
| 10 | Evaluation | 41 |
| 10.1 | Critical evaluation of the Process..... | 41 |
| 10.2 | Critical evaluation of the Product | 42 |
| 10.3 | Self-evaluation | 42 |
| 10.4 | Future development..... | 42 |
| 11 | Conclusion | 43 |
| 12 | References..... | 44 |
| APPENDIX A - | Project Proposal | 48 |
| APPENDIX B - | Online Questionnaire | 56 |
| APPENDIX C - | Existing Products Imagery..... | 65 |
| APPENDIX D - | Low Fidelity Wireframe Sketches | 74 |

| | | |
|--------------|-------------------------|----|
| APPENDIX E - | Prototypes | 78 |
| APPENDIX F - | Testing | 84 |
| | White-box Testing..... | 84 |
| | Black-box Testing | 86 |
| APPENDIX G - | Borrowed code..... | 87 |

1 INTRODUCTION

1.1 Background Information

In the recent years there has been a huge demand of recruiting IT professionals worldwide to fill the hundreds of thousands positions, even millions currently available, a problem which continues to grow much more intensively. It is expected by 2020 the demand to show "*increases ranging from 28% to 32%*" (Thibodeau, 2012). This is not due to the lack of IT graduates or specialists, but due to lack of quality and competent IT professionals. Quality is very substantial requirement, especially in the software industry.

To produce quality and competent individuals, who will be ready to start an immediate career after graduation, organizations and universities has to cooperate with each other to find a solution or improve the already available ones. This is a rapidly changing environment. The solution is to keep students motivated all the way through and inject them with the idea that there is nothing to be afraid of in programming. A solution, which is to teach and help them improve and develop the newly acquired knowledge from the lectures, on their own pace. This must be achieved in an engaging, interactive and fun way, because the beginners get bored very easily after seeing their first error on the screen. This is where they get lost and confused.

One of the deliverables of this project was to develop an interactive web application that will help motivating the beginners of this area of study. The application will be developed according to current standards of the industry with emphasis on HCI to make it an interactive approach to learning. Accessibility and usability issues will be considered very carefully to make the application available to anyone willing to tackle the difficulties they are facing in programming. The product is also intended to support the findings of the research study.

1.2 Project Scope

The purpose of this project, is firstly to conduct a research into the field, to identify the key factors that mostly motivate the youth of this era to learn programming. Also to investigate in detail and compare similar products already available, with greater attention to HCI, to acquire a better understanding of the concept and to give a better user experience with the intended final product. Prior to development process, some research into Accessibility and Usability is mandatory, to achieve a quality product that adheres to standards. After all this an interactive web application will be developed to support the arguments highlighted in the Literature review of the report. A good process will result in a good product.

The proposed web application will let the students to practice and learn in depth a chosen programming language on their own. After an online questionnaire was sent out to First Year Undergraduate students of the CMS Department in the University of Greenwich, the exact programming language to be used in the product was finally decided, which is C# (C Sharp). The results can be clearly seen from the summary of responses (see Appendix B). A detailed discussion on this will be available in the next chapters.

Which Programming language would you like to learn next or most?



The application will involve some quizzes and small games, which will make the time spent more fun and motivating. Also there will be implemented a score and progress functionality. At first everything will be very easy for the beginners, but it will get more difficult as the student progress through the learning environment by collecting points.

If the project meets all of the requirements and deliver all components, it will be deemed as successful one.

1.3 Project Aim

The aim of this project is to develop an innovative learning application/website for Year 1 CIS student to learn and practice programming basics by “doing”. The long term goal is to motivate learners.

1.4 Project Objectives

A detailed set of objectives were set at the beginning of the project. Please refer to the Project Proposal (see Appendix A) for full version. Below is a summarization of the main objectives:

- Conduct a thorough research in the relevant field and write an Investigation report consisting of: Literature review, Review of existing products and Technical review.
- Produce the Design documentation consisting of separate documents in detail: Statement of requirements (Requirements elicitation, Questionnaire results review, Functional and Non-functional requirements); UML and ERD design; User Interface design.
- Product development: Web application and SQL Database
- Produce Test documentation as a result of thorough Black box, White box and Unit testing
- Write Self-evaluation and Product and Process evaluation
- Conclude the project with a Reflection report

1.5 Project Deliverables

Two products will be delivered at the end of this project. A result of systematic investigation and knowledge gathering through reading and analysing quality literature.

- An Interactive Web Application
- Final Year Project Thesis

1.6 Project Methodology

As the life of the current project is unpredictable and open to changes many times, this will require the Rapid Application Development (RAD) Prototyping approach using the ‘MoSCoW’ methodology to control the process in an evolutionary way. Building an initial prototype as early as possible will be a very significant step. Also an iterative concept will be required in order to test and get feedback from the end-user.

The reason for choosing RAD is due to the fact that it is time and cost effective methodology, which puts more emphasis on development to deliver a quality end product, rather spending time on planning. It is completely opposite of the Waterfall model, which puts more emphasis on planning. The RAD Prototyping is suitable in this case, because it allows going back to make corrections to the requirements as knowledge is gathered along the way of development.

The evolutionary methodology ‘MoSCoW’ will be used to identify the high priority and low priority requirements. In this way requirements are divided into two: high level requirements (MUST and SHOULD) and low level requirements (COULD and WOULD). More attention will be given to the high level requirements as they will be enough to have a functioning prototype delivered. And if time permits the low level requirements will be taken under consideration and implemented to have a fully functioning faultless quality product. But this is not the main goal, as the future of the project is unpredictable.

Drawbacks may occur due to the nature of RAD, which are reduced scalability, reduced features, limited functionality as the developed application starts as a prototype and evolves during the development life cycle into a finished product. More than one prototype will be developed along the way to ensure quality and better functionality. In this case three prototypes will be delivered, each one an evolved version of the previous one.

Adjustments would be made if required during the length of the project. Monitoring the plan at all times will reduce the risks of fail and keep up with deadlines (see Appendix A).

2 THE FEASIBILITY STUDY

An email that contains short description, aim and purpose of the study, with a link to the online questionnaire, was sent out to Year 1 CMS students by the supervisor of this course. The results were a little bit disappointing at first, because only 31 students entered their answers to the questionnaire. Nevertheless, the summary of responses by the students were quite enough to prove the feasibility of this study.

The results (see Appendix B) show that 61.3% of the students identified themselves as ‘Beginners’ in programming, also with 19.4% and 6.5% respectively as ‘Intermediate’ and ‘Advanced’ programmers. Only 4 students (12.9%) were identified as ‘No knowledge’ (no previous experience). As the aim of the final product is to motivate mainly beginners, the study acquired good results.

When asked ‘Which Programming languages do you know already?’ responses show that most of the students (72.4%) are already familiar with ‘Java’. This is due to the fact that most of the First Year students are enrolled on computer programming courses with ‘Java’. A third of them (37.9%) also familiar with ‘C#’, a knowledge and experience more likely gained through college or self-study. As these two languages are the most common among students, the rest of the answers will be ignored. Ignored because students shared that they would like to practice and learn ‘C#’ the most (51.6%), followed by ‘Java’ (29%).

Most of the students (25.8%) were not confident and sure with their programming skills, which reveals a serious problem. Hopefully the end product will be able to motivate them in their studies. To mention, a reason for motivation to the author of this project, was the positive answers of the students who mostly liked the idea of the project. This means that they believe in the success of this project.

Another area to note is students approach positively the idea of using interactive learning environment with games (87.1%) answering with ‘Yes’ and small quizzes (48.4%). They believe that this approach will help them to get motivated in programming.

To conclude due to tight deadlines only one language will be available to practice at this point, which is ‘C#’, the champion with 51.6%. Moreover more emphasis will be on implementing simple games to grab the attention of the students and if time permits some quizzes will be added as well. Future improvements to the learning environment could be the added in feature to practice more than one language (e.g. C#, Java, JavaScript and etc.) and availability of more interactive content not limited to only one or two.

Please check the questionnaire and summary of responses available at (see Appendix B).

3 LITERATURE REVIEW

3.1 Introduction

This literature review, a part of the report is intended to discuss the findings of an extensive and deep investigation. Including to that knowledge gathered by the analysis of existing research studies carried out by quality academics and professionals. The topics covered by them are quite wide in range such as e-learning, robotics, issues in programming, learning motivation, pedagogy and other problems of the field relevant to this project.

The author will be addressing the problem and solution to the problem divided into few sections, and finally conclude his arguments and findings with a summary of the problems and solutions investigated and analysed. This will include the difficulties of first year undergraduate students facing in programming and the underlying reasons. The key factors of motivation among students will be the primary topic of this review, which will also help to form and develop the end product of the project. Hence before moving to the key factors that help in motivation, firstly will be looked at the demotivating factors. What is so frustrating or bad in the curriculum or the system that cannot motivate the students and meet their expectations? Their expectations are very important point to look at and discuss.

A summarized list of key motivation factors and concepts to be discussed and considered in the following sections are: use of e-learning environments, e-assessment, and web based interactive solutions, use of visualization and animations, Virtual Reality (VR), programming robots, programming games and learning styles.

3.2 Difficulty of learning to program

“Programming” is a complicated business (Jenkins, 2001). Jenkins is not the only one who argues that learning programming is very difficult and not the only one who proposes few possible approaches to solve this problem. The different approaches trying to find solution to this issue are yet to come in the next section, in detail. Moreover in nearly all research papers used for this literature review, the difficulty of programming was the first thing to point out by the authors. Everything about programming was *“recognized as a difficult endeavour”* (Kelleher and Pausch, 2007), especially for first year students who are novice and cannot assimilate the overwhelming material and abstract concepts, that the traditional approach to teaching offers to them (Kuo-En Chang et al., 2000).

Regardless of their many efforts, programming teachers also find themselves in struggle to teach the important aspects of the material and grab the attention of the students (Drumond et al., 2014). The authors also indicate that every little problem or stress reduces student’s enthusiasm to learn.

According to Hwang et al., “*coding to solve problems*” on its own as a single activity is not good enough to develop students’ cognitive learning (Hwang et al., 2008). They suggested applying multiple activities in proper, following each other from simple to advanced concepts. Coding to solve problems is a common approach, however as mentioned by the authors single activities can start a feeling of monotony and ineffectiveness in students. Again the traditional programming and the way it is taught to students is not engaging, relevant and fun. Markham and King indicated in their conference paper that examples as introduction to variables, expressions, conditionals and etc., and their use in assessments are “*purely academic, designed to practice the concepts taught*” (Markham and King, 2010). They conclude with the argument that this approach is no longer suitable for attracting the youth of this era into computer programming courses. Moreover the traditional approach cannot do much to prevent the high dropout rates of first year undergraduate students. Apparently the debate on practicing the best teaching methodologies will never end (Markham and King, 2010).

Some examples used in teaching programming are either trivial or not specific to the real world. For instance, Anderson et al., gave examples with the “Fibonacci sequence” and “hello world!” and tried to show how some students may fail to understand the application of similar examples, beyond computing (Anderson et al., 2015). The authors also concluded that students must work on real life problems in order to prevent this kind of false impression. To program the aforementioned examples could be easy, however the idea of programming itself becomes difficult, as long as the students don’t know why they are doing and learning something in particular. Some could think of it as a waste of time and may decide to skip that lesson, even abandon the course as a whole. A concern resulting in “misbeliefs” such as “programming is difficult”.

Jenkins is not the only who claimed that programming is difficult, as mentioned previously (Jenkins, 2001). Lahtinen et al. also supported his arguments and added that the difficulty is in the nature of programming (Lahtinen et al., 2005). The authors believed the “*lack of personal instruction*” in students was another reason. Programming is a skill, supported by knowledge or even by aptitude. Why not! It requires correct understanding and dedication (time) to grasp the abstract concepts in its nature. Lahtinen et al. argued in their research paper that the most important reason to face difficulties in programming, is because students new to programming don’t know how to apply what they have learnt in class. Students have to be encouraged to “*learning by doing*” by their teachers (Lahtinen et al., 2005). Practical exercises and assignments are a key to acquire deeper understanding of programming.

A research carried out by Butler and Morgan, included surveying 150 students from their organization, ended up with unsurprising results – a high percentage of the students did have difficulties in implementing high level programming concepts, especially challenged with Object-oriented programming (OOP) (Butler and Morgan, 2007). As a result this showed that nearly all novice programmers don’t feel comfortable with conceptual material. Esteves and Mendes strongly agreed with

the previous argument and clearly stated the difficulties faced with conceptual aspects in their conference paper (Esteves and Mendes, 2004).

Gomes and Mendes summarized a list of possible determining factors of student failure. Factors such as: Static format of learning versus dynamic behaviour of a program; Abstract concepts hard to visualize; Emphasis on theoretical knowledge and memorization; Limited access to the individual support of their teachers; Different skillset; Lack of visual representation to non-English speaking students (Gomes and Mendes, 2007). To add on this some important factors of failure and demotivation were identified by several researchers: lack of study, practice, determination and time, wrong learning strategy, subject difficulty, unsuitable teaching method and imbalanced treatment (Hawi, 2010; Hijon-Neira et al., 2014). Other factors that affect the learning of students were proposed by Jenkins too, in his substantial piece of work: multiple skills, multiple processes, the language, and educational novelty, interest, reputation and image, and pace (Jenkins, 2001).

“Although young people interact with digital media all of the time, few of them can create their own games, animations, or simulations. It’s as if they can “read” but not “write”” (Resnick et al., 2009).

To rectify all these problems there needs to be introduced means of motivation. By means it was meant either one of the following possible solutions or a combination of them: PBL, robots, e-learning environments (web or desktop), e-assessment environments, interactive visualisation tools and games. The following section is dedicated to discuss some of these methodologies used as a way to motivate students in different organizations.

To finalize this section, Jenkins concluded his arguments with *“Programming is learned by programming, not from books”* (Jenkins, 2001), thus calling for a new approach into teaching and learning of programming.

3.3 Motivation

This section is intended to discuss the key factors of motivation, among students enrolled on computer programming courses, identified by quality researchers. The author will expose his findings to explain the key factors of motivation, as well as possible approaches and methodologies to enhance students' learning process. The main goal found in all research papers was to motivate and attract students into programming. Moreover to prepare them to the future real life of a Software Engineer, by developing their soft skills, in addition to technical.

3.3.1 Key factors of motivation

Law et al. in their hypothetical research paper proposed that the motivation of students and the factors affecting them are strongly correlated with students' efficacy (Law et al., 2010). They introduced two

types of motivation: “*intrinsic*” and “*extrinsic*”. To identify a set of motivation factors the authors categorized the two types as:

Intrinsic:

- “*individual attitude and expectation*“
- “*challenging goals*“

Extrinsic:

- “*clear direction*“
- “*reward and recognition*“
- “*punishment*“
- “*social pressure and competition*“

(Law et al., 2010)

In their research study, Law et al. set two hypotheses out of three, to check the correlation level of students’ efficacy against each motivation factor. To summarize students who valued ‘intrinsic’ or ‘extrinsic’ factors showed higher degree of efficacy (Law et al., 2010). Undoubtedly there is a strong relationship between efficacy and performance, which as a result showed that students with higher level of efficacy would perform better in learning computer programming. Drumond et al. also proposed the use of ‘intrinsic’ type of motivation, with strategies related to games, activities to simulate logical thinking and letting students set their own objectives (Drumond et al., 2014).

Interesting results of a study conducted by Cummins et al., using robotic-based learning in programming courses with novice students, revealed several factors to have a positive effect on motivation: ‘absolute integration’, ‘same programming environment’, ‘simplicity of programming’, ‘robust communication’ and ‘robust hardware’ (Cummins et al., 2008). Factors, which present comfortable environment to students as they would like to see, as well as helping develop their soft skills. Other researchers also agreed with this and concluded that students’ learning capabilities were enhanced when they work in comfortable and usable environment (Feldgen and Clua, 2004; Jurado et al., 2012). Lahtinen et al. surveyed students in their organization and ended up with interesting answers by students, who shared that studying and programming by themselves, ‘learning by doing’ and interesting examples were key motivation for them (Lahtinen et al., 2005). But the authors noted that the one has to be careful when judging student opinions, because they contradicted teachers’ opinions in the same research study. Feldgen and Clua also indicated two types of learners: ‘analytic’ and ‘global’ learners (Feldgen and Clua, 2004). Their piece of work was quite significant to take into account.

“*We know that people learn best, and enjoy it most, when they are working on personally-meaningful projects*” (Resnick et al., 2009).

An approach applied by Binas and Pietrikova in their organization showed significant improvement and better motivation to students. The factors adopted were:

- Letting students program at the soonest
- Not using unfamiliar concepts, prior teaching
- Teaching step by step
- Using attractive examples
- Teaching how to debug their programs
- Sharing the solution of difficult problems
- Avoiding teaching irrelevant material

(Binas and Pietrikova, 2014)

To conclude Jenkins and Lykke et al. shared the same vision in their quality pieces of work and indicated the substantial significance of the different motivation types ‘extrinsic’, ‘intrinsic’, ‘social’ and ‘achievement’. (Jenkins, 2001; Lykke et al., 2014). The latter publication also stated that the ‘comfort level’ of students is one of the strongest factors in motivation (Lykke et al., 2014).

3.3.2 Different approaches to motivate students

This subsection was aimed to discuss the available methodologies and approaches recommended by credible sources, which have significant effect on motivating students and attracting them more into computer programming. The findings are not limited to one good solution, but encouraging to combine few or more to suit the needs of the students.

Third hypothesis set by Law et al. encouraged the use of facilitative e-learning systems, due to positive linkage with the students’ learning efficacy (Law et al., 2010). Others agreed with his statements too. Another similar approach to motivate programming was introduced using ‘Wanda’, a tool to create card-based games encouraging the use of algorithms, and improved the algorithmic thinking of students (Drumond et al., 2014). A challenging and competitive environment that can be used by teachers to teach their students the basic concepts of algorithms. Survey conducted by Drumond et al. gathered significantly helpful data about students’ experience with ‘Wanda’. About 90% of the students claimed that this kind of learning environment was very interesting and fun, and they wanted to see similar activities in other courses too (Drumond et al., 2014). This proves that the use of interesting and engaging methodologies in the course could have big influence on students’ interests.

A study carried out by Kuo-En Chang et al., introduced the ‘completion strategy’, which encouraged the use of examples in programming (Kuo-En Chang et al., 2000). This approach was used to let students program by completing, modifying and extending the given examples. According to Kuo-En Chang et al., this strategy was very beneficial in motivating students. Other activities like ‘gap filling’ and ‘peer assessment’ were found to be very helpful in the cognitive development of students, if there was an assisting tool (Hwang et al., 2008). The authors conducted interviews with students and the derived

information showed that activities like ‘coding to solve problems’ and ‘peer assessment’ enabled to improve students’ problem solving and evaluation skills (Hwang et al., 2008).

Yet another approach introduced by Ball et al. was the use of ‘LEGO Mindstorms’, which aimed to prove that Computer Science could be exciting and fun, despite the wrong perceptions of people, e.g. being ‘boring’ and difficult (Ball et al., 2012). ‘LEGO Mindstorms’ was accepted by the authors as a creative and innovative way to encourage and motivate children and students, by showing them the importance of working as a team and solving problems (Ball et al., 2012; Cummins et al., 2008). Similarly a research by Markham and King exposed the positive effect of robots in programming motivation, and how working with them was fun and satisfactory (Markham and King, 2010). Other than motivation to learn, ‘robot-based’ learning was stated as a very effective way to improve students’ problem solving, team working and communication skills. Anderson et al. also added the effect of using ‘robotics’ in education, including other ones like ‘media computation’ and ‘animations’ (Anderson et al., 2015). They strongly suggested the use of real life applicable projects and to keep deadlines flexible.

“When creating curricula based on games, they need to be sure that the games they choose really are able to motivate students of both genders” (Kelleher and Pausch, 2007). Diversity is a substantial aspect to consider. That’s why in ‘game-based’ learning teachers have to consider both genders, and create environments suitable for both sides. For instance, if a ‘game-based’ learning environment was proposed, which included only games in the sort of ‘first-person shooters’ and ‘sports’, as a consequence we would put barriers in front of the female learners. Another game-based learning environment for novice programmers was also introduced, called ‘Greenfoot’, with significant results in motivating students and facilitated learning. The study results indicated that more than 60% of the students were satisfied with this approach and 65% rated positively its usefulness, effectiveness and productiveness (Hijon-Neira et al., 2014).

Other approaches that might help to increase the motivation, involved the use of ‘concurrent programming models’ (Leavens, 2008), and ‘pedagogical (elementary) programming patterns’ (Nunes de Barros et al., 2005). However the arguments given to these approaches did not fulfil the expectations of the author and were not measured as very helpful in this particular case study.

Esteves and Mendes in their study indicated the significant benefits of using ‘animation based simulation’ to reduce the difficulties of students (Esteves and Mendes, 2004). According them this kind of approach could help to interact better with program’s dynamics and let student learn on their own pace. Novice students into programming frequently face problems while using the complicated Integrated Development Environments (IDE), as they are professional tools and meant for advanced programmers. That’s why the introduction of learning environments with pedagogical goals could be more suitable to help students learn (Esteves and Mendes, 2004).

Yet again another product that was praised by students with positive comments was ‘AulaWeb’, a “*Web Assisted Self-assessment*” application (Garcia-Beltran and Martinez, 2006). The authors stated that this approach is not to assess the students, but motivate and encourage them. Moreover self-assessment objects e.g. quizzes help to “evaluate their own understanding and misconception” (Matthews et al., 2014). Respectively an Intelligent Tutoring System (ITS) proposed by others, was described as an ideal way of Computer-Assisted Learning (CAL) approach, which guided and helped students in their learning process to acquire the best skills (Jurado et al., 2012).

The difficulties of learning to program and the need to find a proper motivation method were considered and discussed by nearly all researchers so far. One more way to attain this was by introducing a web-based ‘Intelligent assistant’, with exact goals similar to the previous examples (Konecki et al., 2015). Similarly an interesting approach to motivate students was the use of Virtual Reality (VR) in computer programming courses (Chandramouli et al., 2014). The authors described it as a fun-based learning in 3D environment, which could help to reduce the cognitive overload on students.

The author also analysed an interesting comparison study by Lykke et al., which had for purpose to compare different approaches and their effectiveness against each other (Lykke et al., 2014). Three approaches were put in comparison: Project Based Learning (PBL), PBL + LEGO Mindstorms (robot-based approach), and the traditional way of teaching. The results of the study indicated there wasn’t too much difference between each and their effectiveness was questioned (Lykke et al., 2014).

Jiau et al. clarified the difference between traditional game-based assignments and commercial games, and indicated why the commercial ones are so successful (Jiau et al., 2009). They took similar approach in their study, and observed the improvements that traditional approach wouldn’t have achieved. The key was to develop competitive games, where “*outcomes are largely under the control of the students*” (Jiau et al., 2009). This is called ‘game-based metrics approach’. All student are familiar with games, therefore understand a problem better this way, as if they were trying to win the game (Feldgen and Clua, 2004). Apparently game-based assignments and game-based metrics are one of the best ways to motivate students today.

Simple, but meaningful conclusion to this section –

“A little bit of programming can go a long way” (Resnick et al., 2009).

3.4 Conclusion

To conclude, the investigation process of extensive amount of quality literature and their analysis let the student to obtain significantly in-depth understanding of the problem stated. The gathering of proven

arguments and applying them to form the literature review of this project, did help the student to improve his critical-thinking, self-study, evaluation, writing and research skills. The credible sources mostly agreed with the difficulties students are facing in learning and understanding the conceptual aspects in programming. Motivating students is the key to teach them programming.

Throughout the whole process of reviewing the findings, the motivation problem of students in learning programming, the source of these problems and what we can do to solve them were clearly stated in detail. The investigation resulted in with the general fact that programming will be always ‘difficult’, however if we approach the problem with carefully planned methodologies and apply them, there will be significant changes in the learning process of students. After the investigation the student can tell that, students need to work in their comfortable environments in order to learn and practice programming. This means the curriculum has to put efforts to understand students’ expectations from a learning activity or methodology. This argument was proven to be correct in the literature review, indicating the fact that students learn the best in their own way, the way they feel more comfortable (e.g. playing games and following strategy instructions).

Several factors of motivation were discussed in the review, e.g. intrinsic, extrinsic, social, achievement and others. Each student has their own way of learning, therefore the factors that motivate them vary. For instance some could have only one way to motivate themselves, but also some others could have a mix of different types of motivation. Also many different approaches were proposed to help student motivation: web and desktop-based e-learning environments, web-based self-assessment environments, intelligent assistive environments, visualization and animal tools, game-based learning, robot-based learning, Virtual Reality (VR) and pedagogically oriented development environments. Each of the mentioned approaches had proven benefits in motivating students to learn programming. Other benefits of using these ways of teaching were indicated as: reduced drop-out rates, increased success in assignments, increase in the knowledge, problem-solving and critical-thinking skills of the students.

Finally the investigation showed that the most appropriate way to motivate students in this case would be the use of interactivity with ‘game-based approach’, especially it is substantial to take into account the ‘game-based metrics approach’. Overall this was a very useful and helpful process of learning.

4 LEGAL, SOCIAL, ETHICAL AND PROFESSIONAL ISSUES AND CONSIDERATIONS

4.1 Legal

This project's final product will be developed only for the needs of Year 1 undergraduate students, so it will not require any authorization from the university (as long as they are over 18). Therefore there won't be any underage student involved in the process of requirements gathering and testing. But as the requirements gathering involves handing questionnaires to collect relevant information, a permission by the Research Ethics Committee will be required. Also the questionnaire forms must be designed and written in compliance with the Data Protection Act (1998). Any personal data must be handled and stored appropriately according to the Data Protection Act (1998). For the purpose registration of users to the intended web application would only involve adding username and password, where the passwords will be stored into the database in an encrypted format with the help of an algorithm (hashing). This would prevent to some extend security risks that may arise due to users or external parties with bad intentions.

The project will involve comparing similar products available (functionality and design). All the material (images, sounds and content) to be used in the web application must comply with the Copyright, Designs and Patents Act (1988). A suitable way to comply with the aforementioned regulation would be creating the imagery content of the intended product personally with the use of 3rd party tools, such as Adobe Photoshop CS6, considering a professional and creative design. Any textual content available should be derived from the project itself or written in self-words. However as the creation of audio content would be very difficult, they could be kept at the bare minimum or not used at all. Also Copyright, Designs and Patents Act (1988) involves proper citing and referencing of all the sources used in the Literature Review or simply the whole project.

It is clear that not complying with these laws might result in from light to heavy punishment or even banned if decided by the faculty due to plagiarism.

4.2 Social

Kelleher and Pausch (2007) in their substantial piece of work indicated the significance of gender diversity and its association with students' motivation. Therefore all contents must be created for the learning environment with particular attention to be unisex. Other social issues may arise if religion, sex, gender and age are requisite for the users to provide when they register. To prevent such a bad issue none of the aforementioned details will be asked from the users. Also if the intended product was to be used

also by non-English speaking students, further adjustments should take in place. But currently it will be available only to learners from the UK, so this won't be a problem.

4.3 Ethical

The privacy aspects will be considered very strictly and an encryption will be required when retrieving and saving data to the database. Also unauthorized access to content and personal data, will be eliminated by denying the access of a user if their login details are not correct or are not registered with the system. To prevent misuse and going against the law, at the end of the development all collected data will be destroyed. Other ethical issues that have to be take into account could be about the accessibility and usability of the web application, as well as cross-platform compatibility and interoperability.

4.4 Professional

A professional manner will be followed by taking under consideration standards and guidelines by well-established bodies such as BCS (2011) and IEEE (2012). Proper planning and management of the project will prove this. Four sections from the BCS code of conduct are particularly important to bear in mind: *Public Interest, Duty to Relevant Authority, Duty to Profession and Professional Competence and Integrity* (BCS, 2011).

5 REVIEW OF EXISTING PRODUCTS

5.1 Introduction

This chapter is intended to review existing products in the market available for free or fee, which are similar to this project's final product. By similar it was meant having the same principles, approach and aim, including running on the same platform (web-based). Only the most appealing and well-known to the audience products will be reviewed. Because an investigation on existing products ended up with too many results, which would have required too much time to test and review, if we consider the tight deadlines of this project. As a result three products were reviewed, which are pretty interactive and game-based with the purpose to ease and motivate learning of programming. The chosen web-based products are: 'CodeCombat', 'CodinGame' and 'CodeMonkey'. Each product will be reviewed in a separate section.

| Product Name | URL to Website |
|-------------------|---|
| CodeCombat | https://codecombat.com/ |
| CodinGame | https://www.codingame.com/start |
| CodeMonkey | https://www.playcodemonkey.com/ |

5.2 Product 1 - CodeCombat

5.2.1 Review

CodeCombat is a web application with the aim to teach programming with 'game-based' approach. There are no lessons, but instructions and quests (assignments) to follow in the game. At first the levels are pretty easy to complete, but as the player continues to progress the game difficulty increases too. As you progress the need for better problem-solving and algorithmic-thinking skills increase too. CodeCombat is doing great in motivating learners, as the players have good control over the game, e.g. being clan member, editing avatar, play desired level, choice of programming language or contributing to the project itself, which is open source. Also there is a feature specially added for teachers' use, where they can create a class for their students and teach them programming in a fun way.

Currently the available programming languages to learn and practice are as follows: Python, JavaScript, CoffeScript, Clojure and Lua. Unfortunately the choice of available languages is limited to scripting ones. From my personal experience with this learning environment, Python was particularly easy to start coding. Surprisingly there are wide range of languages to choose, to be exact more than 50. This is excellent as it will allow anyone from anywhere in the world to learn programming in their native language, if they don't know or prefer English. CodeCombat heavily relies on open source code and high

quality services, with more than 40 open source languages and technologies used in the development process of the web application. The core languages are CoffeeScript, Jade, Sass and Markdown. One more thing very important to note is that the code editor used is called ‘Ace’, which is very customizable, open source and embeddable in any web environment.

CodeCombat is a very interactive learning environment, which uses good visualization and animations in the games, therefore very attractive option especially for the younger audience. Navigation is pretty easy and it is obvious where each button links to. At first this application may seem complicated for new comers, but it gets very easy to use after few tries. Overall CodeCombat presents very interesting, attractive and interactive environment to learn programming, while having fun. Very similar to ‘game-based metrics’ approach. Also their source code is available on GitHub. Not suitable for mobile devices.

5.2.2 Appearance



For more images please refer to Appendix C.

5.3 Product 2 – CodinGame

5.3.1 Review

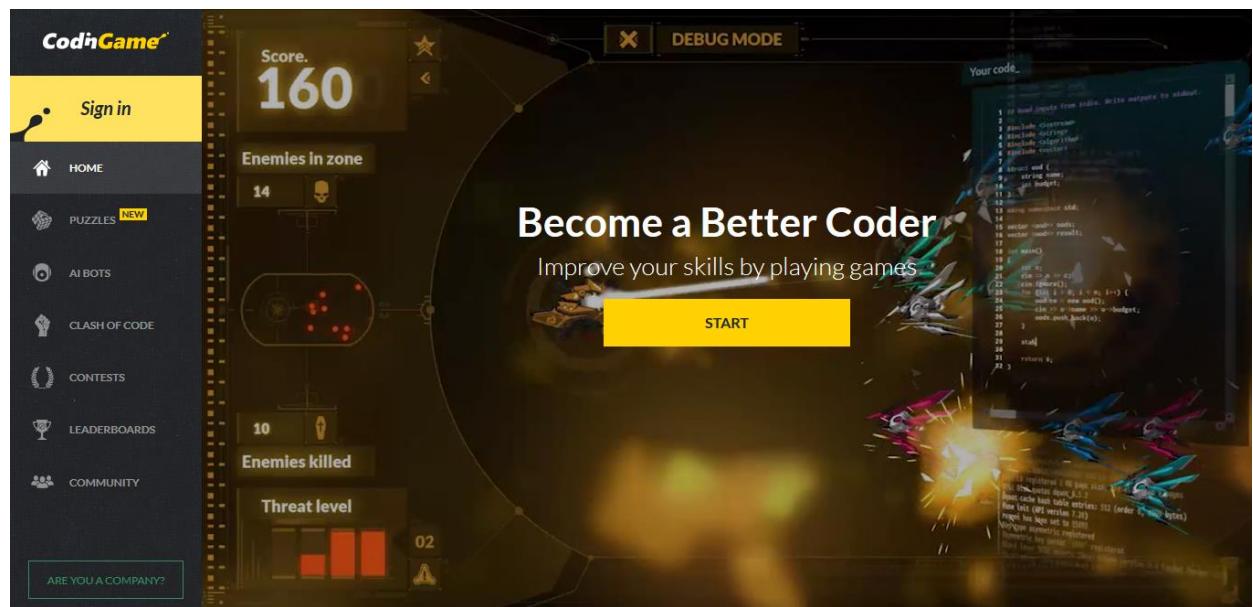
CodinGame is a web-based innovative learning environment with ‘game-based’ approach. They have the goal to help learning and improving programming skills of the coders, no matter if they are qualified professionals or just starting to program. There are many features in CodinGame and the games are categorized in four levels of difficulty – easy, medium, hard and very hard. So they present a challenging environment to the programmers. The games are called puzzles and you play them in three different modes – Single player (solving algorithms), Optimization (getting the highest possible score), Code golf (achieving the shortest possible code). Also another feature called AI Bots, allows the programmers to build smart bots to ‘fight their opponents’. CodinGame is a very challenging environment, because the

players also have the option to participate in challenges with other players to solve algorithmic problems as fast as possible. Something very interesting to mention is the option to participate in contests with great prizes for the top 10 programmers. Finally if a player becomes consistent and very good in solving problems, they have the chance to be offered a real job, which is the greatest motivation to start learning programming.

There are more than 20 programming languages to practice, which makes it suitable for anyone. Unfortunately only English and French are available to choose from, which is very disappointing. What about someone who wants to learn programming, but doesn't speak either of those languages? This is an issue that needs more attention. Again the code editor in CodinGame is 'Ace', which is important to note for later investigation on its use. However this product is not open source and it is not available to the community, whereas CodeCombat is and heavily reliant on community's help for improvements.

Navigation is very straightforward, however the application is neither easy nor difficult to use. CodinGame unfortunately has a bit more confusing interface than CodeCombat, where the game, instructions, code editor and test cases are in the same page. Basically it is overcrowded with too many features. Still a very good learning environment to practice programming, especially for skilful programmers. CodinGame could be a bit difficult for newcomers into the field. Not suitable for mobile devices.

5.3.2 Appearance



For more images please refer to Appendix C.

5.4 Product 3 – CodeMonkey

5.4.1 Review

CodeMonkey is an online game for teaching and learning basic programming concepts. It is intended for the very young audience from the age of 9 and up, but it is also suitable for older learners with no previous knowledge or little experience in programming. CodeMonkey follows the ‘game-based’ approach to motivate the learners. Basic concepts are covered throughout the learning process by playing the game, e.g. objects, function calls, arguments, loops, variables, arrays, for loops, function definitions, until loops, if and if-else conditions and boolean operators. The game is very simple and entertaining, where the main goal is to control the monkey to walk and catch the bananas. The players are encouraged to get 3 stars as achievement for each game level (e.g. 1 – to catch the banana; 2 – use of all concepts taught to complete the game; 3 – short code and to the point). Other than playing the basic game, the learners who have active subscription have the option to use features like ‘challenge builder’ and ‘skill challenges’. Moreover teachers and organizations can use CodeMonkey to create classes and teach their students programming in an interactive and fun way, but again subscription is required.

The game is translated into 18 languages, making it accessible to a wider audience. The only programming language used by the learners is CoffeeScript, which compiles to JavaScript and resembles the way we write in English, due to its simple syntax. Looks like the open source code editor ‘Ace’ helps many of the similar environment used to practice and learn programming. ‘Ace’ is used by CodeMonkey too.

Unfortunately this learning environment is not available for mobile devices too. But we have to bear in mind that, coding in a mobile device wouldn’t be very pleasant and easy. Programming is better done using computers, so a web-based application is very suitable for this purpose. CodeMonkey is designed very attractively and ‘fun’, also it is very easy to navigate and use. Overall a very good product for beginners to learn programming in a very entertaining and engaging way.

5.4.2 Appearance



Our Programs

For more images please refer to Appendix C.

5.5 Key issues to use in the design and implementation

After the investigation and review of current successful products several issues were taken under consideration. Hopefully applying or avoiding few of them, will have a positive impact on the intended final product. Key issues and aspect to consider are:

Positive:

- The overall structure must be consistent;
- The navigation must be available on all pages and easy to navigate;
- Access to home page from other pages must be as easy as one click;
- Straightforward registration and login;
- Any text included must be conformant to standards (size, colour, font and etc.);
- Any imagery and animation should be considered carefully (relevance, copyright and etc.);
- Any sound/music could be attractive, but should be considered carefully (irritation, copyright);
- The colour scheme used must be chosen carefully (accessibility, standards and etc.);
- Simpler layout and content would be helpful;
- The product must be easy to use;
- The products must be accessible;
- The product must be interactive;
- The product could use the ‘game-based’ approach;
- The product could have more than one language or way to teach;
- The use of open source technologies would be helpful;

Negative:

- Avoid inconsistency;
- Avoid content that may cause confusion and irritation;
- Avoid over using imagery, animations and sounds;

5.6 Conclusions

To conclude all of the products were developed to teach and help learning to program by introducing interactive game-based environments and were presented up to good standard. They all have the same goal – to motivate learning and make programming fun and engaging. This is achieved by solving problems to complete challenging, competitive and fun games. Learning to code, while playing games is a wonderful idea especially for the younger audience.

Again programming must be easy and fun, not difficult and boring. Aforementioned examples really succeeded in attracting children and adults into programming, no matter of their experience and future career intentions. Programming definitely shouldn't be perceived as a way to career in the IT industry, but more as an essential skill and a way to improve a person's problem-solving, critical-thinking and other soft skills. Basically being able to write program and think analytically shouldn't mean the person has to work as a programmer.

We can categorize the reviewed products according their difficulty level as: CodeMonkey more suitable for beginners, CodeCombat for beginners and intermediate programmers and finally CodinGame for experienced and advanced programmers.

6 HUMAN COMPUTER INTERACTION

6.1 Introduction

This section is intended to discuss the significance of Human Computer Interaction (HCI) in software development, its positive effects on the product and key issues arising when it is not used according to set of rules and well-established standards. Additional attention will be paid to accessibility and usability concerns, which are also very important aspects of HCI. Hopefully the literature used here and the information gathered would have a great impact on the final product's design. This is essential because the web application of this project is expected to be interactive, easy to use and adherent to standards.

6.2 HCI

The interaction of users with interactive computer systems is referred to as Human Computer Interaction (HCI), which plays great role in the design and implementation phase of a given product. The product/system can be anything from mobile, desktop and web application up to embedded systems in some devices. If something is poorly designed or just hard to use, then the users won't be very pleased to use it and they may even ignore it after few tries. So this is where HCI come into action to make things simple and easy to use and understand. However a system with a simple design look, doesn't always mean that a simple technology is behind of it, but actually more complex implementation and set of technologies needed to build it (Mathew et al., 2011). Therefore functionality and usability are combined to make up a well-balanced and successful system. Usability of a given product means how successful it is in utilizing those functionalities easily and faultlessly.

Mathew et al. (2011) also indicated in their conference paper that there are three factors to be considered when designing an interface, such as physical, cognitive and affective factors. Hearing, touching and seeing are the physical aspects, so these are achieved with the use of devices such as touch screen, keyboard, mouse an etc. Especially very helpful for people with disabilities, who experience most things by touching and listening. Customizability and adaptability of features such as navigation bar and menu lists according the needs of the user is result of the cognitive factors. And finally affective are those factors, which include human's emotions and satisfactions. This can be a result of a system with pleasant and easy to use design, as well as not irritating and forcing the user for particular tasks (Mathew et al., 2011).

6.3 Accessibility

Accessibility is a practice, which eliminates the barriers of people with disabilities and allows them to use a particular software as healthy one would do. Especially designing accessible Web sites and applications nowadays has become almost mandatory, although there are still organizations that are not

fully conformant with this standard, due to the need to make visually attractive products, especially with HTML5. Unfortunately this is not too good to hear and another reason to this problem could be the time required to train and familiarize the staff with this standard. Yes it is a standard supported by the W3C Web Accessibility Initiative (WAI) and there is an extensive documentation on their web site. Also there are loads of plugins for the browsers we use today, to help with the validation of our web pages on the go automatically. Personally I have been using similar tools as well, one for validating XHTML1.1 documents and another one to validate the accessibility of the web sites I have created so far. Also I think adding the logos in the page footer provided by W3C would be nice, which are available for different validation purposes, and would give the user the feeling that the page/site they are visiting is very professional and thought about humanity and show that the Web should not have barriers between people and computers. Hopefully in the future more companies will try their best to adopt this standard and we will see more examples than today



The web accessibility covers all people with disabilities to help their access to the web such as visual, auditory, cognitive, physical, speech and neurological disabilities. As more accessible becomes a product the more people with disabilities will be able to contribute and feel effective. However we should not think about making our products accessible only because it will help disabled people, but also the ones without disabilities. The benefits of this can be seen also when there is a slow internet connection, physical problems due to injury or just due to aging and people become more limited with their daily tasks. As a result everyone benefits from web sites designed conforming to accessibility standards, as they make the Web way more flexible than we think actually.

6.4 Usability

The user interface of a system is a medium in the communication between a human and computer. A good design consequently leads to more effective, easier and less mistaking interaction for the user in this communication. The interface should meet user's expectations and previous experience, as a result the usability of a system is perceived as a quality term (Gong, 2009). Good usability requires designers particularly to pay attention to understand the cognition of the user, provide approaches to make the learning as easy as possible as well as by providing hints and feedback with instructions for particular tasks, which really makes it user-centric. Moreover the user interface has to be effective to allow the user to spend as little as possible time performing a given task and satisfy them by knowing their wishes, needs, minds and experience from the very beginning to the end (Gong, 2009). And as a result all these factors all together help the user interface to be more usable.

7 REQUIREMENTS ANALYSIS

7.1 Introduction

In order to successfully complete the design phase and finally move on to product development, firstly we have to carefully gather and analyse the requirements. For the purpose a questionnaire was created to get information about students' expectations from this type of product and measure their experience and skills in programming. The results from the questionnaire were already discussed to approve the feasibility of the project (see Chapter 2), as well as the further actions to be carried out. For a detailed view, please refer to the questionnaire and summary of responses (see Appendix B).

Also a significant amount of information was derived by reviewing existing products, which aimed to motivate learners and ease learning of programming by 'game-based' approach' (see Chapter 5). The whole review process was very helpful to determine the positive and negative issues arising, and to identify the functional and non-functional requirements.

7.2 Target audience

The target audience of the end product could be anyone, as it is a web-based learning environment and will be accessible from any location, as long as they have access to computer and Internet. Mainly the product is intended to be used by Year 1 students enrolled on computer programming courses. Students who experience difficulties in learning programming and where the traditional way of teaching cannot motivate them. Also the product will be suitable for both male and female students, in order to have gender diversity.

7.3 Statement of requirements

The statement of requirements consists of two subsections: Functional requirements (FR) to show what the product is intended to achieve and Non-functional requirements (NFR) to show how they will be achieved and possible constraints that may arise. Each of the requirements will be categorized as 'high' or 'low' level priority using the MoSCoW methodology. Requirements with high priority will be indicated with 'MUST' or 'SHOULD' as they are the most significant ones to be achieved. Also requirements with low priority, which are not necessarily significant to achieve, but might be considered if time permits will be indicated with 'COULD' or 'WOULD'. A list of functional and non-functional requirements is below:

7.3.1 Functional requirements

| FR | Description | MoSCoW |
|-----|---|--------|
| 1. | User to register successfully | MUST |
| 2. | User to login successfully using their ID and Password | MUST |
| 3. | User login details to be kept in a Cookie and Session | MUST |
| 4. | User must have access to learning material after authorization | MUST |
| 5. | User to logout from the system successfully | MUST |
| 6. | System to retrieve data from MS SQL Server database | MUST |
| 7. | System to alter data in MS SQL Server database | MUST |
| 8. | System to insert data into MS SQL Server database | MUST |
| 9. | User must be able to navigate to desired pages and default content | MUST |
| 10. | User must be able to choose practice activity (programming quiz or game) | MUST |
| 11. | User to start a quiz | MUST |
| 12. | User to play a programming game | MUST |
| 13. | System to generate questions in a random order | SHOULD |
| 14. | User to check personal progress (level, experience and results) | SHOULD |
| 15. | System to keep progress data in MS SQL Server database | SHOULD |
| 16. | User to check solutions | SHOULD |
| 17. | User to change account password | SHOULD |
| 18. | Bugs in games and quizzes | COULD |
| 19. | User to find information in F.A.Q. section | COULD |
| 20. | User to get in touch with the admin/owner/developer (email or phone) | COULD |
| 21. | User to penetrate and threaten the system using hacking tricks (injection tricks) | WON'T |

7.3.2 Non-functional requirements

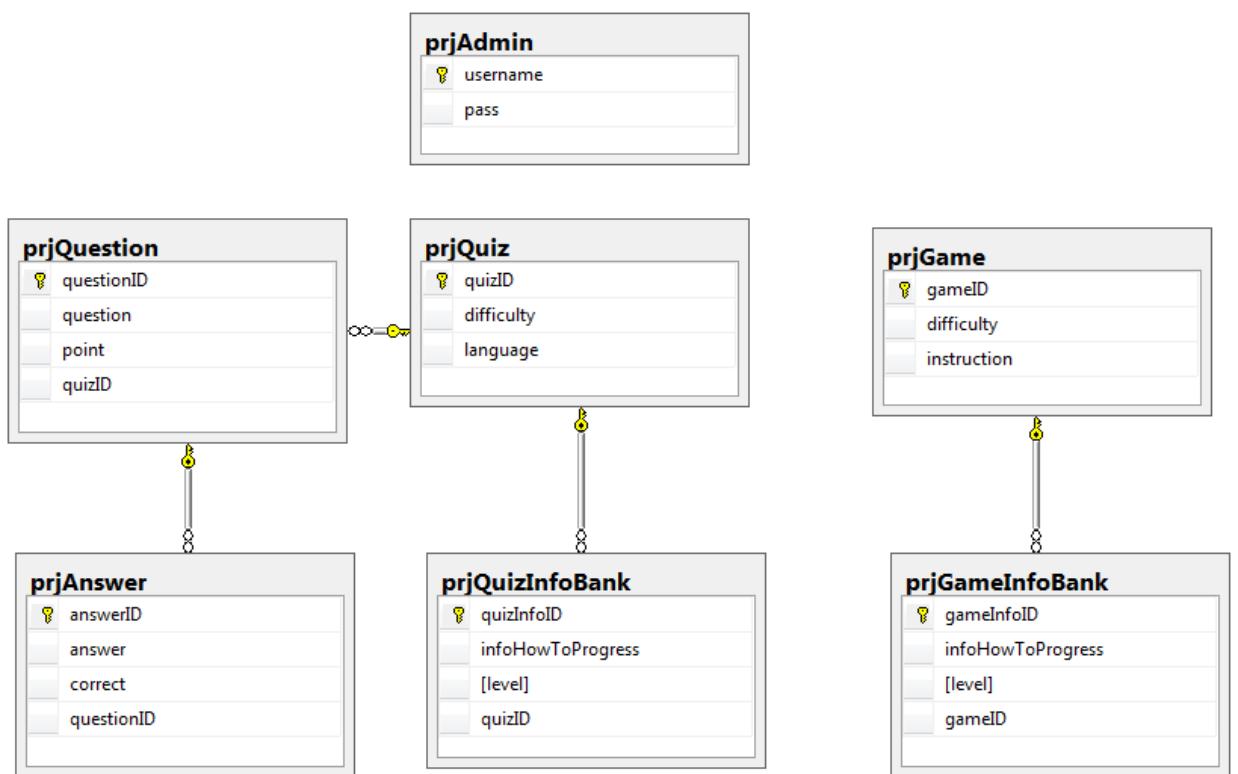
| NFR | Description | MoSCoW |
|-----|--|--------|
| 1. | Usable – The application must have user-centric design and must be kept clear and simple. The application must be easy to use and navigate through pages back and forward. | MUST |
| 2. | Easy to learn - Consistent interface, with additional instructions for advanced tasks and actions | MUST |
| 3. | Easy to understand – Simple and unambiguous content | MUST |
| 4. | Engaging – Must be pleasant, comforting and satisfying to use (interactive content and games). | MUST |
| 5. | Security – Unauthorized access and manipulation to the system must be prevented. | MUST |
| 6. | Privacy – Personal information must be protected and data to be stored securely. | MUST |

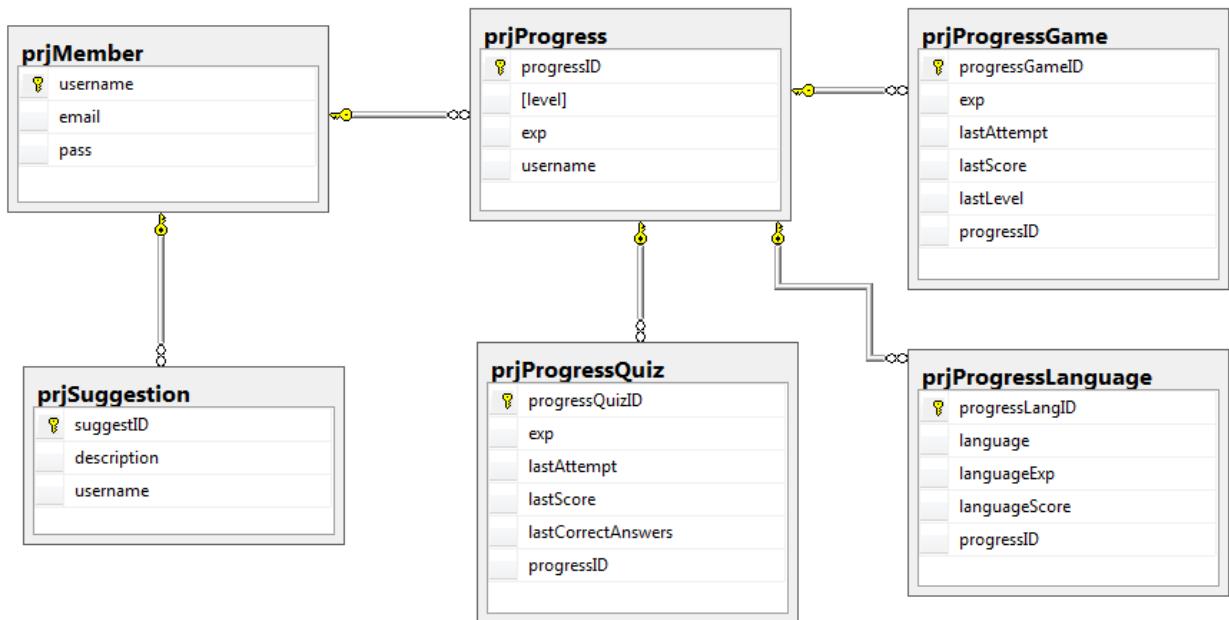
| | | |
|-----|---|--------|
| | (e.g. email and password) | |
| 7. | Accessible - The user should be able to reach the website from any browser and platform. Also any user with disabilities should be able to use the application. (appropriate colour scheme, compliant to standards) | SHOULD |
| 8. | Performance – The application should perform well. (loading, request/response). Also the application should be effective and efficient. | SHOULD |
| 9. | Error tolerant – Errors prevented and recovered caused by user's interactions | SHOULD |
| 10. | Reliable – The application must work to a certain extent to satisfy user's needs | SHOULD |

8 DESIGN OF THE PRODUCT

8.1 Entity Relationship Diagram (ERD)

Entity Relationship Diagrams are an important aspect of the design phase and the initial step to development. Especially to have a fully functional database driven application, the database and its structure must adhere to well supported and up-to-date standards. Therefore as the student had previous experience with database design and use of RDBMS tools, all tables were normalized and created in 3rd Normal Form. This methodology is very important, because it eliminates ambiguity, as well as redundant and repetitive occurrence of the data stored in the database. Only few of the tables did not require normalization as they were too simple to do anything with. Moreover being in 3rd Normal Form means that any data processed through the application is very likely not to cause problems while reading or writing data. This also means less coding, for particular tasks required in the application.

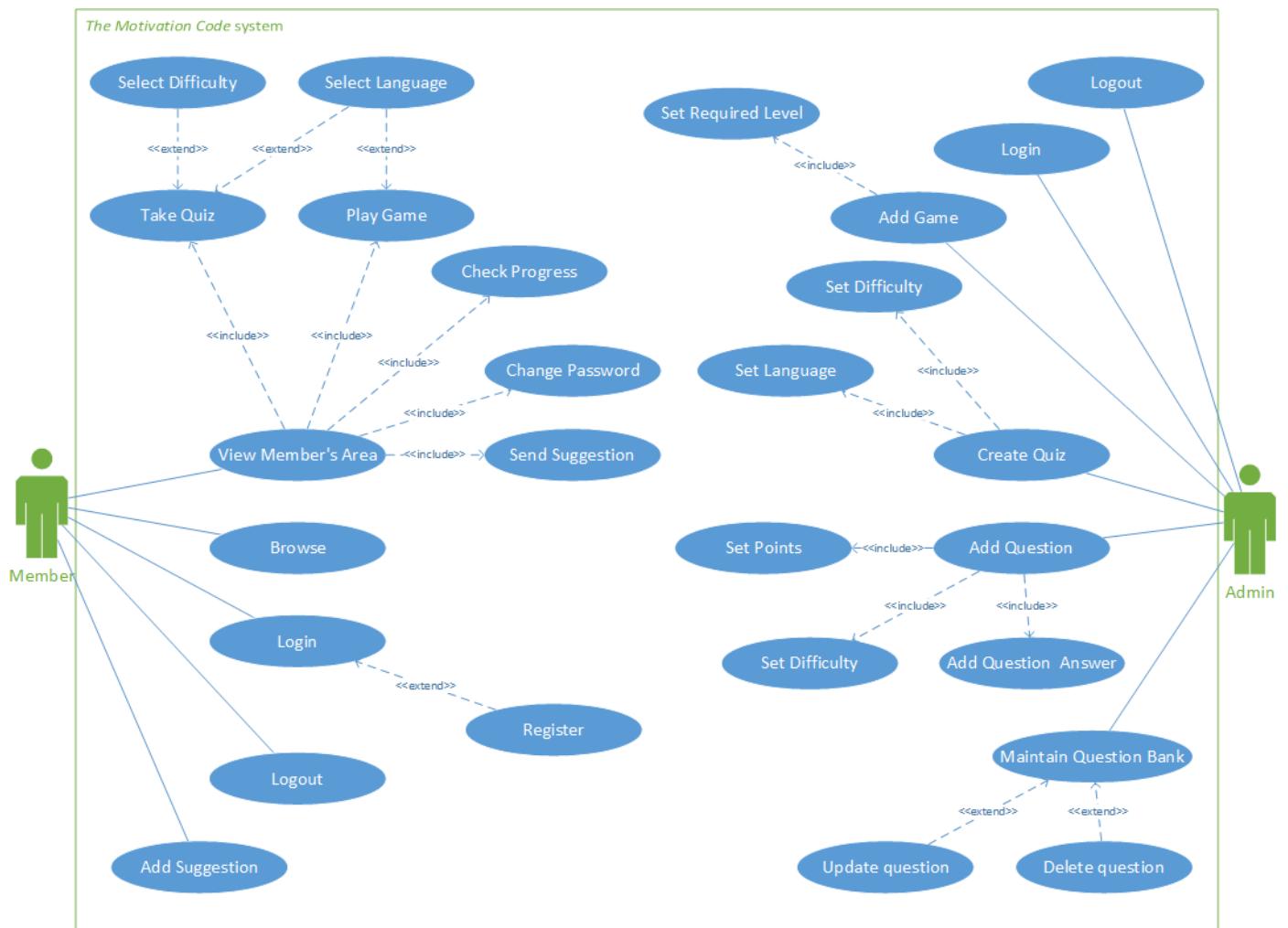




8.2 UML Use-Case Diagram

Primary Actor: Member

Secondary Actor: Admin



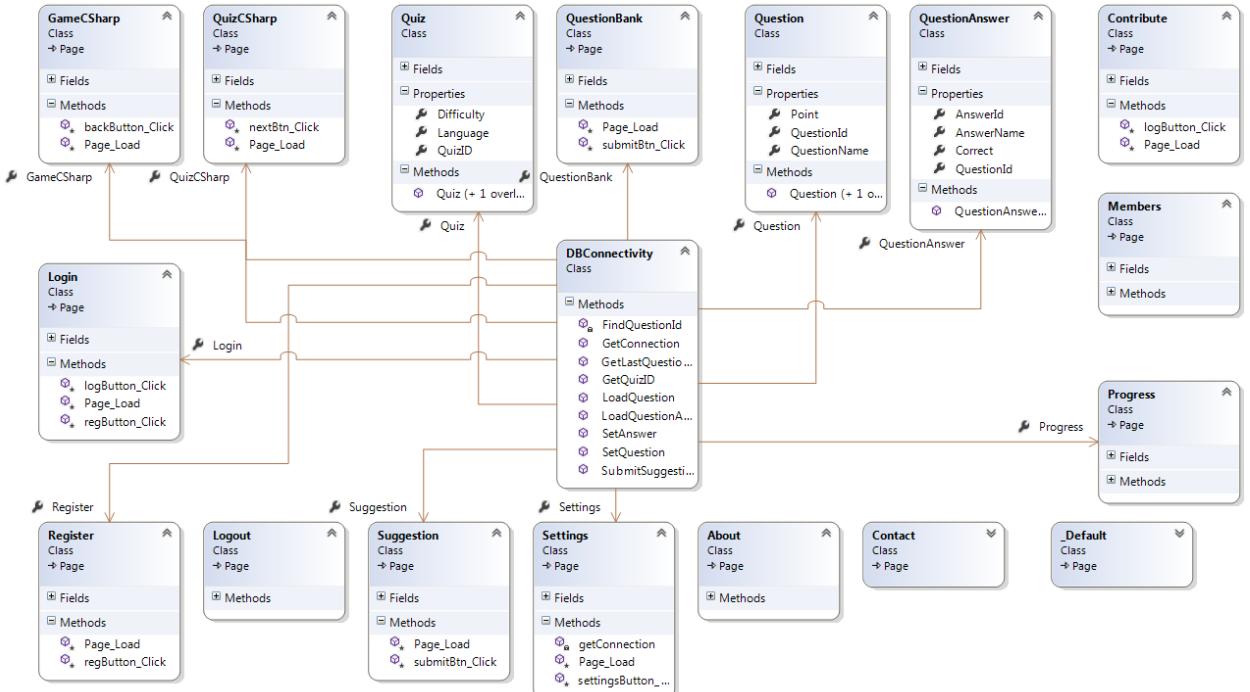
The Use Case diagram above illustrates the actions available to both Primary and Secondary actors. It is an overall overview how different users interact through the web application. Both Members and Admin can log in and log out from the application, except Members have to register using their details in order to authenticate, otherwise access to main content will be rejected by the system.

Members may browse and select specific features in the member's home page. Those features/functions include taking a quiz, play a programming game, submit a suggestions/ideas to the system, display personal progress or just amend account details.

The Admin is sort of a secondary user, however has an important role, who is able to add new games to the system according their difficulty, create new quizzes, which includes adding questions and their answers, difficulty and points awarded for success in each question. The Admin is also responsible to maintain the Question bank by updates and removal of unnecessary questions.

8.3 UML Class Diagram

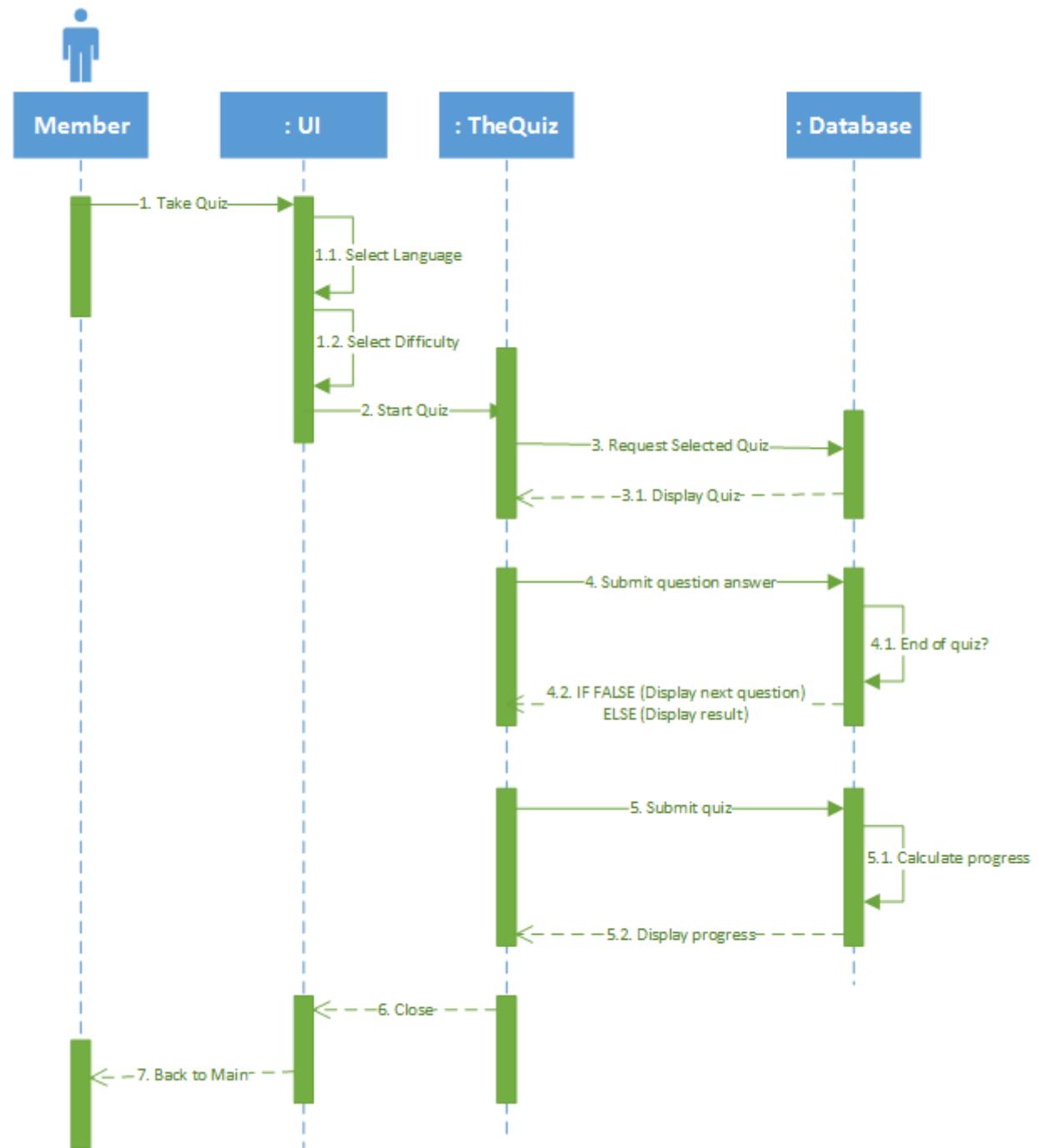
The Class diagram below, also referred to as structural diagram, represents product's static overview. Moreover it is an overview of the relationships and code dependencies between classes.



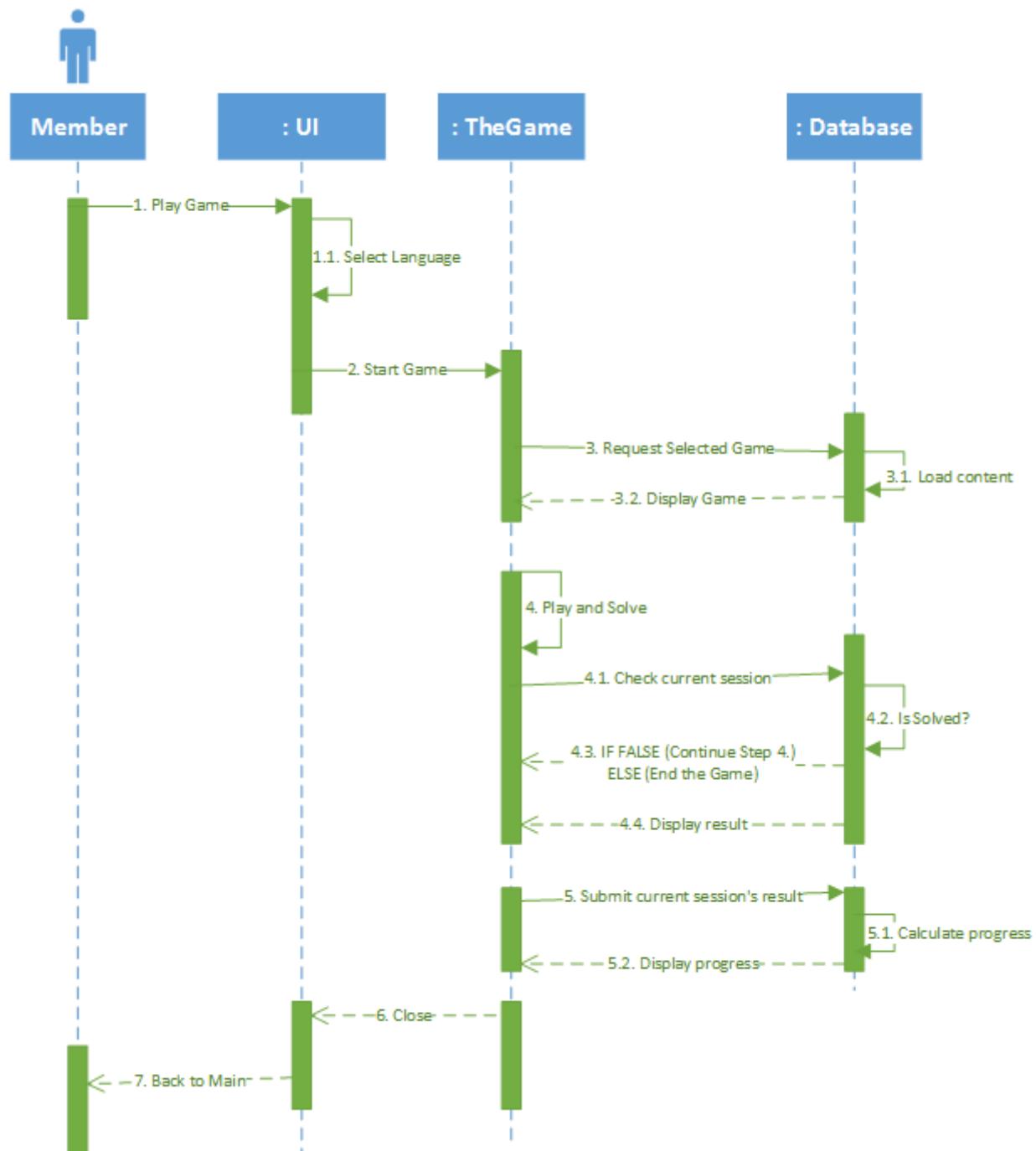
8.4 UML Sequence Diagrams

UML Sequence diagrams are also very important part of the software development life-cycle. They represent how processes operate with each other and in what order (arranged in time sequence). For the purpose only two Sequence diagrams were created as they are the main important aspect of the whole system. One illustrating the process of taking a quiz by the user (Take Quiz SD) and the other one similar but for playing a programming game (Play Game SD). Other Use Cases were not depicted, due to the fact that they are just too simple to put in a diagram and also there isn't much of interaction.

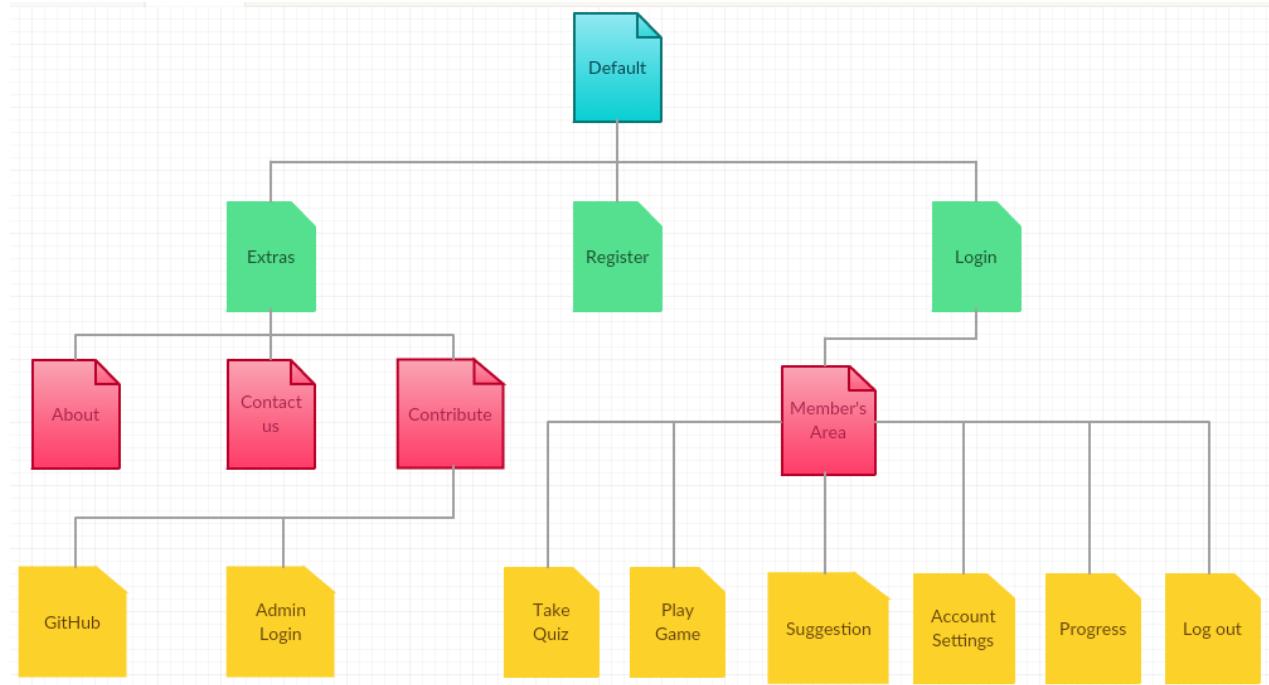
8.4.1 Take Quiz SD



8.4.2 Play Game SD



8.5 Site-Map Diagram



8.6 Initial Sketches

For the initial design of the product in order to obtain a clear and proper structure a Low Fidelity Wireframe sketching was conducted, which consists of several simpler sketches done by hand. The expected looks of most of the pages to be designed/created in the development phase are available at Appendix D. This approach really clarified how the first few prototypes should look like, which helped a lot because the coding of the front-end was significantly less time consuming than expected.

8.7 Prototypes

Thanks to the previously created Low Fidelity Wireframe sketches (Appendix D) this stage was really straightforward to accomplish. In total three prototypes were created to form the final product:

1. Prototype A – This initial prototype only consists of the Default page with some simpler content achieved using only CSS, HTML5 and some images gathered from the Web.
2. Prototype B – This is an improved version of the previous Prototype A, which has most of the required pages with added functionality from the back-end using .NET C# and uses MS SQL Server database to read and write data. All images were created using Adobe Photoshop CS6.
3. Prototype C – This is the final version of the intended product of the project. This version fills the gaps found in Prototype B. Design-wise it is not very different from the previous version.

As part of the Rapid Application Development (RAD) approach Prototyping was really useful as it allowed the student to make improvements and changes on top of the previous prototype, as well as moving back and forward through the development phase. Prototypes available at Appendix E.

9 DEVELOPMENT OF THE PRODUCT

9.1 Introduction

This section is intended to discuss the whole process of developing and forming the final version of the web application, which will be also referred to as prototype. Starting from simple Low Fidelity Wireframe sketches up to a working product, not fully but still up to some extent, the process involved lots of reading, investigation and applying the newly acquired knowledge into a real product. The product was developed as ASP.NET Web Forms application with C#, HTML5, JavaScript and MS SQL Server database.

9.2 The Process of Implementation

The first thing to do was to build the database to be used at server side for data handling of inputs and outputs. All tables were normalized on the go in 3rd Normal Formal prior coding, to prevent going back at some point in the future and alter the tables again and again. This also could have caused time consuming changes to the back-end logic of the web application, after every change to the database. That's why it was really important to build the database correctly at the first. Another good plus of using normalized tables is that it allows you to add new tables without messing with the old ones, if new features were to be added to the prototype. Moreover I would like to explain a good practice that I started doing in the last few years, while using C#. It can be done in any language, but I just can't stop doing it if I have to build a project in Visual Studio, which is the greatest IDE in my opinion. My approach to data writing to and reading from the database is to create a separate class just to hold all database methods used in the application. This prevents duplicate code, when the same function is required in few place, which happens a lot, as well as this leads to maintainable code as each method is available only at single place. Everyone is familiar how complicated and messy could be a database method.

Thanks to my previously created Low Fidelity Wireframe sketches, done by hand and the Use Case diagrams from design phase of the project really helped me to figure out the initial design of the product with its initial navigation and layout structure, including where to put the dummy content such as text and imagery. I decided to use a template provided by *OS Templates* which consisted of CSS and HTML5 documents structured specially to have a responsive layout adapting to screen size. Only the structure of HTML document has been used for this document and unnecessary boiler plate code was removed to meet my needs. I did know that if I do the design myself it would have consumed too much time and probably with very simple design that wouldn't have met the requirements of this project such as ease of use, understandability, learnability and simplicity, which leads to an appealing to use application.

The first Prototype A included only the development of the front-end for testing purposes and included the aforementioned activities. After this major improvements have been made to Prototype B, which included the pages Register, Login, Log out, Settings and Progress. Building these Web Forms did not consume too much time and were pretty easy to accomplish as they only required input validation and database connectivity to write and read data. The possibility of SQL Injection was taken into account and to keep the database secure from third party manipulations parameterized queries have been used for this purpose.

Prototype C, which is also the final version involved the stages of implementing the non-trivial functionalities required in order to have a complete and interactive web application. The information gathered from the Literature review and the outcomes have led to the idea of using interactive games to engage and motivate the learners of programming. Moreover the findings from the existing products review showed that all those products did use very challenging and interactive games, which required the use of code editor and writing code to complete certain tasks in the games and proceed to the next level. Interestingly the code editor used in all reviewed products was ACE code editor, which is open source and available at GitHub. This is how I decided to combine the use of games and the open source code editor. However as I didn't have experience at all in game programming, inevitably I had to use game developed by others. Although it is a simple game developed in JavaScript, it seemed to me way complicated to create by myself. Also the code editor unfortunately did not have a large documentation to learn from or maybe it was due to my inexperience, not sure. As a result I couldn't manage to link the game with the code editor and to run codes from the application to achieve the main objective of my project. This issues of course will be discussed in the final section in more detail.

9.3 Changes made to Prototype C

I was really worried about the project demonstration as I did not have fully functioning product as I expected, however the understanding of my tutors and their great feedback and suggestions on my work helped me get motivated and to not lose enthusiasm no matter of what is the result at end. Their feedback resulted in few more features to be added that I hadn't thought about:

1. Added page for suggestions/ideas to be submitted by the user. Because it is always good to hear others' opinions and meet their needs.
2. Added separate page for login to be used by the administrator.
3. Added a link to my personal GitHub account, which in real life would have been the open source code of project. This is what CodeCombat does and continuously improving itself with the help of the community.
4. Added a page where the administrator could add new questions to the question bank.

9.4 Technical problems

The main problem apparently was underestimating the project and the expected final product. As I mentioned before I did not have previous experience in game development, as well as using open source code to meet my needs, which ended up with a failure. Unfortunately these 2 tasks consumed my whole time while trying to understand and complete and link them to each other. Due to bad time management I couldn't complete the quiz functionality as well, where the quiz uses JSON objects to get the questions, instead of reading from the database. Although I have the database tables required for all these 3 features, none of them use the database functions provided.

9.5 Extra about Prototype 3

Picked color pairs that are complaint with WCAG 2.0(AA) for the web application using the Accessibility Color Wheel at <http://gmazzocato.altervista.org/colorwheel/wheel.php>. People with vision deficiencies: deuteranopia, protanopia and tritanopia shouldn't have problem using the application. Exception could be made for all images used as most of them use gradient background colours. All of the imagery available in the web application have been created by myself with the help of Adobe Photoshop CS6, with gradient background colours as mentioned before and interesting text fonts, that you hardly would see in another application. Those 2 features make up to some extent an accessible, easy to use and fun environment for learning of programming. In my opinion one won't regret that the application looks very appealing and interactive at first sight.

10 EVALUATION

This sections contains:

- Critical evaluation of the Process
- Critical evaluation of the Product
- Self-evaluation
- Future development

10.1 Critical evaluation of the Process

Overall it has been a very exhaustive and extensive process of planning, investigation, reviewing, learning, design and development. All this process feels like a real life project of the real Software Engineer, because it covers widely used methodologies and techniques used in today's IT industry. Now it is clear why we have been studying courses such as System Building 1, Systems Development Project, and Software Engineering Management in the last 3 years. We used to not understand the point of studying these courses every year, because each course's outline was more or less the same.

Despite being very long and time consuming the process seems finally to achieve its objectives to some extent. With minor faults due to lack of time I feel the Testing, Design and HCI sections could have been more detailed and accurate. Especially I am not very confident about the Black box testing that I have provided, which is due to the fact that we have never done in any project this kind of testing. It is indeed very simple testing technique, however it wasn't clear how I should approach it. And I feel embarrassed that couldn't do it appropriately. Also would have been nice if I could write more about HCI, because it is very extensive topic, as well as to give more detail about accessibility and usability considerations. The design phase consists of many diagrams and drawings, where most of them seem quite good and detailed. Unfortunately cannot say the same about the UML Class diagram and I always had this issue of not knowing how the diagrams must be created the correct way since Year 1.

Fortunately the literature review and existing products review are up to standard with wide range of relevant information, which address the issues of the research study and find appropriate answers to them.

My initial planning for the project was quite good, however when things started to go wrong it was really hard to keep up with the plan and it just became a chaos. This is one of the reasons of poor achievement in some of the tasks. And I believe I should have been more careful with my time management.

10.2 Critical evaluation of the Product

I would evaluate the product by its pros and cons:

Pros:

1. Normalized database tables in 3rd NF.
2. The web application has a responsive design adapting itself to screen size.
3. Usability is considered: simple, easy to use, understandable, learnable, consistent and efficient.
4. Conforming to standards: such as W3C WAI.
5. Secure handling of personal data.

Cons:

1. Game not developed by myself.
2. Quiz not developed by myself; except the questions.
3. Code editor does not have a functionality.
4. User progress not implemented.
5. Quiz does not read the questions from the database.
6. Poor functionality of the question bank.

As it can be seen the final product has its own advantages and disadvantages. If you ask could you fix the cons of the application if I were given more time, probably it would be ‘No’. I believe this is due to overambitious project topic and implementation proposed by myself. Taking into account that I wasn’t competent to manage to develop this product, I really wanted to do it. My final year project wouldn’t be nice if it was something simple and meaningless or something that even can’t solve a real life problem. I just wanted to challenge myself and understood I’m yet to learn more and more.

10.3 Self-evaluation

Personally I believe this project taught me how to conduct an investigation in the correct way; how to process the findings and use the important aspects of a given source; how to conduct academic referencing; how to plan a project step by step; improved my critical thinking; improved the way I propose arguments; also to be self-teaching and self-reliant in my studies.

10.4 Future development

Of course future development of this project and its product is very likely to be a success as long as it is managed well and developed by more competent engineers. I am saying this because 2 or 3, nor few technologies will be enough to complete such a complex system. According to my observations ‘CodeCombat’ for instance was developed with the use of more than 20 open-source programming languages and technologies. So it is not an easy task, however it is possible and would be very pleasant to see a new programming learning environment in the market.

11 CONCLUSION

This final year project has been a challenging learning experience, which has taught the student loads of valuable skills in software project management and development. It was especially helpful to represent the software development life cycle from the very beginning until finalizing the developed product.

Thanks to this project the student shouldn't face a problem in his future career as a Software Engineer, who will definitely see the benefits of this exhaustive process of investigation and learning. The slightest experience can go a long way and hopefully will improve on top of this in the near future, while working on real life projects at work.

To conclude every piece of work included in this project will be a vital step into Software Engineering. With its ups and downs the final year at the University of Greenwich will be an unforgettable and valuable experience for the student.

12 REFERENCES

Anderson, R., Ordóñez, R., Pham, P. and Tribelhorn, B. (2015) A Data Programming CS1 Course, In *SIGCSE '15 Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, ACM New York, NY, USA, pp. 150-155, [online] Available at: <https://homes.cs.washington.edu/~mernst/pubs/data-programming-sigcse2015.pdf> (Accessed 19 February 2016).

Ball, C., Moller, F. and Pau, R. (2012) The Mindstorms effect: A gender analysis on the influence of LEGO Mindstorms in computer science education, In *Proceedings of the 7th Workshop in Primary and Secondary Computing Education*, ACM, pp. 141-142, [online] Available at: <http://cs.swan.ac.uk/~csfm/Pubs/wipsce12.pdf> (Accessed 20 February 2016).

BCS (2011) Code of Conduct for BCS Members, [online] Available at: <http://www.bcs.org/upload/pdf/conduct.pdf> (Accessed 18 April 2016).

Binas, M. and Pietrikova, E. (2014) Useful recommendations for successful implementation of programming courses, In *Emerging eLearning Technologies and Applications (ICETA), 2014 IEEE 12th International Conference on*, IEEE, pp. 397 - 401.

Butler, M. and Morgan, M. (2007) Learning challenges faced by novice programming students studying high level and low feedback concepts, In *Providing choices for learners and learning. Proceedings ascilite Singapore 2007*, pp. 99-107, [online] Available at: <http://www.ascilite.org/conferences/singapore07/procs/butler.pdf> (Accessed 19 February 2016).

Chandramouli, M., Zahraee, M. and Winer, C. (2014) A Fun-Learning Approach to Programming: An Adaptive Virtual Reality (VR) Platform to Teach Programming to Engineering Students, In *Electro/Information Technology (EIT), 2014 IEEE International Conference on*, IEEE, pp. 581 - 586.

Copyright, Designs and Patents Act 1988 (1988) Copyright, Designs and Patents Act 1988 [online] Available at: http://www.legislation.gov.uk/ukpga/1988/48/pdfs/ukpga_19880048_en.pdf (Accessed 18 April 2016).

Cummins, J., Azhar, M. and Sklar, E. (2008) *Using Surveyor SRV-1 Robots to Motivate CS1 Students*, AAAI, pp. 23-27, [online] Available at: <https://www.aaai.org/Papers/Workshops/2008/WS-08-02/WS08-02-005.pdf> (Accessed 20 February 2016).

Data Protection Act 1998 (1998) Data Protection Act 1998, London: HMSO, [online] Available at:

http://www.legislation.gov.uk/ukpga/1998/29/pdfs/ukpga_19980029_en.pdf (Accessed 18 April 2016).

Drumond, R., Brandao, A. and Salles, C. (2014) Wanda: A Framework to Develop Card Based Games to Help Motivate Programming Students, *Computer Games and Digital Entertainment (SBGAMES), 2014 Brazilian Symposium on*, pp. 158-164.

Esteves, M. and Mendes, A. (2004) A simulation tool to help learning of object oriented programming basics, In *Frontiers in Education, 2004. FIE 2004. 34th Annual*, IEEE, pp. F4C-7-12 Vol. 2.

Feldgen, M. and Clua, O. (2004) Games As A Motivation For Freshman Students To Learn Programming, In *34th ASEE/IEEE Frontiers in Education Conference*, IEEE, pp. S1H-11-S1H-16.

Garcia-Beltran, A. and Martinez, R. (2006) Web Assisted Self-assessment in Computer Programming Learning Using AulaWeb, *Int. J. Engng Ed.*, 22(5), pp. 1063-1069, [online] Available at: http://www.ijee.ie/articles/Vol22-5/18_ijee1790.pdf (Accessed 20 February 2016).

Gomes, A. and Mendes, A. (2007) An environment to improve programming education, In *Proceedings of the 2007 International Conference on Computer Systems and Technologies*, ACM, p. 88, [online] Available at:

https://www.researchgate.net/publication/220795506_An_environment_to_improve_programming_education (Accessed 19 February 2016).

Gong, C. (2009) Human-computer interaction: The usability test methods and design principles in the human-computer interface design, In *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, IEEE, pp. 283 - 285, [online] Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5234724 (Accessed 18 April 2016).

Hawi, N. (2010) Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course, *Computers & Education*, 54(4), pp. 1127-1136.

Hijon-Neira, R., Velazquez-Iturbide, A., Pizarro-Romero, C. and Carrico, L. (2014) Serious games for motivating into programming, In *Frontiers in Education Conference (FIE), 2014 IEEE*, IEEE, pp. 1-8.

Hwang, W., Wang, C., Hwang, G., Huang, Y. and Huang, S. (2008) A web-based programming learning environment to support cognitive development, *Interacting with Computers*, 20(6), pp. 524-534.

IEEE (2012) Publications & Standards, [online] Available at:

http://www.ieee.org/publications_standards/index.html (Accessed 18 April 2016).

- Jenkins, T. (2001) The motivation of students of programming, *SIGCSE Bull.*, 33(3), pp. 53-56.
- Jiau, H., Chen, J. and Ssu, K. (2009) Enhancing Self-Motivation in Learning Programming Using Game-Based Simulation and Metrics, *IEEE Trans. Educ.*, 52(4), pp. 555-562.
- Jurado, F., Redondo, M. and Ortega, M. (2012) eLearning standards and automatic assessment in a distributed eclipse based environment for learning computer programming, *Comput Appl Eng Educ*, 22(4), pp. 774-787.
- Kelleher, C. and Pausch, R. (2007) Using storytelling to motivate programming, *Communications of the ACM*, 50(7), p. 58.
- Konecki, M., Kadoic, N. and Piltaver, R. (2015) Intelligent assistant for helping students to learn programming, In *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015 38th International Convention on*, IEEE, pp. 924 - 928.
- Kuo-En Chang, Bea-Chu Chiao, Sei-Wang Chen, and Rong-Shue Hsiao, (2000) A programming learning system for beginners-a completion strategy approach, *IEEE Trans. Educ.*, 43(2), pp. 211-220.
- Lahtinen, E., Ala-Mutka, K. and Järvinen, H. (2005) A study of the difficulties of novice programmers, *SIGCSE Bull.*, 37(3), p. 14.
- Law, K., Lee, V. and Yu, Y. (2010) Learning motivation in e-learning facilitated computer programming courses, *Computers & Education*, 55(1), pp. 218-228.
- Leavens, G. (2008) Use concurrent programming models to motivate teaching of programming languages, *ACM SIGPLAN Notices*, 43(11), p. 93.
- Lykke, M., Coto, M., Mora, S., Vandel, N. and Jantzen, C. (2014) Motivating programming students by problem based learning and LEGO robots, In *Global Engineering Education Conference (EDUCON), 2014 IEEE*, IEEE, pp. 544 - 555.
- Markham, S. and King, K. (2010) Using personal robots in CS1: experiences, outcomes, and attitudinal influences. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education (ITiCSE '10)*, New York, NY, USA, ACM, pp. 204-208, [online] Available at: http://ims.mii.lt/ims/konferenciju_medziaga/ITiCSE'10/docs/p204.pdf (Accessed 19 February 2016).

Mathew, A., Hajj, A. and Abri, A. (2011) Human-Computer Interaction (HCI): An overview, In Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on, IEEE, pp. 99 - 100, [online] Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5953178 (Accessed 18 April 2016).

Matthews, R., Hin, H. and Choo, K. (2014) Learning Object to Enhance Introductory Programming Understanding: Does The Size Really Matter?, *TOJET: The Turkish Online Journal of Educational Technology*, 13(1), pp. 174-183, [online] Available at: <http://www.tojet.net/articles/v13i1/13115.pdf> (Accessed 20 February 2016).

Nunes de Barros, L., Paula dos Santos Mota, A., Valdivia Delgado, K. and Megumi Matsumoto, P. (2005) A tool for programming learning with pedagogical patterns, In *Proceedings of the 2005 OOPSLA workshop on Eclipse technology eXchange (eclipse '05)*, ACM, pp. 125-129, [online] Available at: <http://dl.acm.org/citation.cfm?id=1117722&dl=ACM&coll=DL&CFID=582778302&CFTOKEN=77452920> (Accessed 20 February 2016).

Resnick, M., Maloney, J., Monroy-Hernandes, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B. and Kafai, Y. (2009) Scratch: Programming for all, *Communications of the ACM*, 52(11), pp. 60-67, [online] Available at: <http://web.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf> (Accessed 19 February 2016).

Thibodeau, P. (2012) IT jobs will grow 22% through 2020, says U.S., *Computerworld*, [online] Available at: <http://www.computerworld.com/article/2502348/it-management/it-jobs-will-grow-22--through-2020--says-u-s-.html> (Accessed 19 February 2016).

APPENDIX A - Project Proposal

An Interactive Web Application to motivate learning of programming

Gyokay Ali

BSc (Hons) Software Engineering

000762233

1 Overview

The project will investigate the different ways of learning programming as well as the key motivating factors affecting learning among first year undergraduate CIS students. Nowadays Software Engineering is one of the most sought after professions all over the world, therefore as future IT specialists the students should possess very good applicable practical skills, in addition to the theoretical knowledge. Without an adequate motivation it is typically very difficult for new students to develop good programming skills, which requires a lot of practice.

Today's fast-changing environment requires to be made continuous improvements to the way programming is learnt and taught, as well as the introduction of effective learning frameworks. It is known that currently technology is the main factor of engagement in learning and academic achievement of the youth.

Repetitive failure and other obstacles that students face throughout the process of learning programming, are some of the main factors of losing interest and enthusiasm. This problem calls for special attention to the way computer programming is taught and learnt, that would enhance students' learning motivation (Jenkins, 2001).

The proposed web application will let the students to practice a chosen programming language (Java, C#, SQL or JavaScript) on their own. The exact programming language is not decided yet, but it will be clear after a questionnaire is sent out to the students. The application will involve some quizzes and small games, which will make the time spent more fun and motivating. Also there will be implemented a score and progress functionality. At first everything will be very easy for the beginners, but it will get more difficult as the student progress through the learning environment by collecting points.

Apparently the development of the application or at least a working prototype will not be an easy task. However I believe that I have the necessary technology skills in order to have a successful final product. The courses Web & Application Development, Advanced Programming and Database Applications Technologies from 2nd Year were very pragmatic and useful. The skills required for this project's application are ASP.Net with C#. SQL, OO concepts and little bit of HTML and CSS. First class grades acquired from the courses above motivate me that this project will be a successful one.

In order to evaluate the effectiveness of this application, it will be tested by students to see if it would have any impact on their learning and if it helps to get them more confident in programming.

Keywords: learning, motivation, e-learning, programming, self-study, education, interactivity

2 Aim

The aim of this project is to develop an innovative learning application/website for Year 1 CIS student to learn programming basics by “doing”. The long term goal is to motivate learners.

3 Objectives

3.1 An Investigation Report [16.0]

3.1.1 The investigation report will consist of literature review and review of existing similar products.

3.1.2 Literature Review [11.0]

3.1.2.1 Search for articles, journals and books [2.0]

3.1.2.2 Read the related material [7.0]

3.1.2.3 Write and finish off the Literature review [2.0]

3.1.3 Existing Products Review [5.0]

3.1.3.1 Search for similar products [1.0]

3.1.3.2 Test, compare and evaluate the products [2.0]

3.1.3.3 Write and finish off the Product review [2.0]

3.2 Design documentation [14.0]

3.2.1 Several design documents will be produced as a result. It includes the statement of requirements, series of design diagrams and the user interface design.

3.2.2 Statement of Requirements [4.0]

3.2.2.1 Create a questionnaire [1.0]

3.2.2.2 Analyse results from the questionnaire [1.0]

3.2.2.3 Gather and write the requirements [2.0]

3.2.3 UML Diagrams [3.0]

3.2.3.1 Design Use-Case diagrams [1.5]

3.2.3.2 Design Class diagrams [1.5]

3.2.4 Entity Relationship Diagram (ERD) [1.0]

3.2.5 User Interface Design [6.0]

3.3 Product Implementation [17.5]

3.3.1 This objective involves creating and populating the database to be used and development of the product.

3.3.2 Database Implementation [3.5]

3.3.2.1 Create the tables [2.0]

3.3.2.2 Apply normalization rules [1.0]

3.3.2.3 Populate the tables [0.5]

3.3.3 Web application development [14.0]

3.4 Testing [7.5]

3.4.1 The testing documentation involves testing the web application as well as the web application

3.4.2 Database Testing [3.0]

3.4.2.1 Create test plan [1.0]

3.4.2.2 Test the database [0.5]

3.4.2.3 Review the results [0.5]

3.4.2.4 Fix if necessary [1.0]

3.4.3 Web application Testing [4.5]

3.4.3.1 Create test plan [1.0]

3.4.3.2 Test the web application [1.0]

3.4.3.3 Review the results [0.5]

3.4.3.4 Fix if necessary [2.0]

3.5 An Evaluation Report [5.0]

3.6 A Reflection Report [5.0]

Project estimated duration [65.0]

4 Legal, Social, Ethical and Professional

This project's final product will be developed only for the needs of Year 1 undergraduate students, so it will not require any authorization from the university (as long as they are over 18). Therefore there won't be any underage student involved in the process of requirements gathering and testing. But as the requirements gathering involves handing questionnaires to collect relevant information, a permission by the Research Ethics Committee will be required. Also the questionnaire forms must be designed and written in compliance with the Data Protection Act (1998).

The confidentiality of sensitive data must be signed off by the end user. The privacy aspects will be considered and an encryption will be required when retrieving and saving data to the database. To prevent misuse and going against the law, at the end of the development all collected data will be destroyed.

The project will involve comparing similar products available (functionality and design). All the material (images, sounds and content) to be used in the web application must comply with the Copyright, Designs and Patents Act (1988). Also this involves properly citing and referencing all the sources used while writing the Literature Review or simply the whole project.

To prevent any social issues it will not be required from the users to enter their religion, sex, gender and age, when they register.

A professional manner will be followed by taking under consideration BCS and IEEE standards. Properly planning and managing the project will prove this.

5 Planning (see appendix A)

As the life of the current plan is unpredictable and open to changes (probably many times) at some point, this will require the Rapid Application Development (RAD) Prototyping approach using the 'MoSCoW' methodology to control the process in an evolutionary way. Building an initial prototype as early as possible will be a very significant step. Also an iterative concept will be required in order to test and get feedback from the end-user.

In general, RAD approaches to software development put less emphasis on planning tasks and more emphasis on development. In contrast to the waterfall model, which emphasizes rigorous specification and planning, RAD approaches emphasize the necessity of adjusting requirements in reaction to knowledge gained as the project progresses (Wikipedia, The Free Encyclopedia, 2015).

Using UML will be a big part of the design phase. Its role is to show the actions of the users involved in the system and help to create the prototype by simply designing the Use Case and Class diagrams.

The project objectives will be managed by creating a Gantt chart in MS Project. Adjustments would be made if required during the length of the project. Monitoring this plan at all times will reduce the risks of fail and keep up with deadlines.

6 Initial References

Websites:

BCS, (2011) Code of Conduct, [Online], Accessed on 24 Oct 2015,
<http://www.bcs.org/category/6030#profession>

Data Protection Act 1998, (1998) London: HMSO, [Online], Accessed on 24 Oct 2015,
<http://www.legislation.gov.uk/ukpga/1998/29>

IEEE, (2012) Publications & Standards, [Online], Accessed on 24 Oct 2015,
http://www.ieee.org/publications_standards/index.html

Copyright, Designs and Patents Act 1988, (1988), [Online], Accessed on 24 Oct 2015,
<http://www.legislation.gov.uk/ukpga/1988/48/contents>

Rapid application development. (2015, October 15). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:10, October 24, 2015, from https://en.wikipedia.org/w/index.php?title=Rapid_application_development&oldid=685812708

Conference Papers:

Jenkins, T. (2001). The motivation of students of programming. In *Proceedings of ITiCSE 2001: The 6th annual conference on innovation and technology in computer science education* (pp. 53–56).

Appendix A – Gantt Chart

| | | | | |
|--|-------------------------------------|---------|--------------|--------------|
| | ▲ Investigation Report | 17 days | Mon 02/11/15 | Mon 23/11/15 |
| | ▲ Literature Review | 12 days | Mon 02/11/15 | Mon 16/11/15 |
| | Search for reading material | 2 days | Mon 02/11/15 | Tue 03/11/15 |
| | Read the related material | 7 days | Wed 04/11/15 | Thu 12/11/15 |
| | Write Literature Review | 2 days | Fri 13/11/15 | Sun 15/11/15 |
| | ▲ Existing Product Review | 5 days | Tue 17/11/15 | Mon 23/11/15 |
| | Search for similar products | 1 day | Tue 17/11/15 | Tue 17/11/15 |
| | Test, compare and evaluate findings | 2 days | Wed 18/11/15 | Thu 19/11/15 |
| | Write the Existing Products Review | 2 days | Fri 20/11/15 | Mon 23/11/15 |

| Task Mode | Task Name | Duration | Start | Finish |
|-----------|-----------------------------|----------|--------------|--------------|
| ➡ | ▫ Design Documentation | 14 days | Tue 24/11/15 | Fri 11/12/15 |
| ➡ | ▫ Statement of Requirements | 4 days | Tue 24/11/15 | Fri 27/11/15 |
| ➡ | Create a questionnair | 1 day | Tue 24/11/15 | Tue 24/11/15 |
| ➡ | Analyse results | 1 day | Wed 25/11/15 | Wed 25/11/15 |
| ➡ | Write requirement | 2 days | Thu 26/11/15 | Fri 27/11/15 |
| ➡ | ▫ UML Diagrams | 3 days | Mon 30/11/15 | Wed 02/12/15 |
| ➡ | Design Use Case diagrams | 1.5 days | Mon 30/11/15 | Tue 01/12/15 |
| ➡ | Design Class diagram | 1.5 days | Tue 01/12/15 | Wed 02/12/15 |
| ➡ | ERD | 1 day | Thu 03/12/15 | Thu 03/12/15 |
| ➡ | User Interface Design | 6 days | Fri 04/12/15 | Fri 11/12/15 |

| Task Mode | Task Name | Duration | Start | Finish |
|-----------|-----------------------------|-----------|--------------|--------------|
| ➡ | ▫ Product Implementation | 17.5 days | Mon 14/12/15 | Tue 05/01/16 |
| ➡ | ▫ Database Implementation | 3.5 days | Mon 14/12/15 | Thu 17/12/15 |
| ➡ | Create the tables | 2 days | Mon 14/12/15 | Tue 15/12/15 |
| ➡ | Apply normalization | 1 day | Wed 16/12/15 | Wed 16/12/15 |
| ➡ | Populate tables | 0.5 days | Thu 17/12/15 | Thu 17/12/15 |
| ➡ | Develop the Web Application | 14 days | Thu 17/12/15 | Tue 05/01/16 |
| ➡ | ▫ Testing | 7.5 days | Tue 05/01/16 | Thu 14/01/16 |
| ➡ | ▫ Database testing | 3 days | Tue 05/01/16 | Fri 08/01/16 |
| ➡ | Create test plans | 1 day | Tue 05/01/16 | Wed 06/01/16 |
| ➡ | Test the database | 0.5 days | Wed 06/01/16 | Wed 06/01/16 |
| ➡ | Review results | 0.5 days | Thu 07/01/16 | Thu 07/01/16 |
| ➡ | Fix if necessary | 1 day | Thu 07/01/16 | Fri 08/01/16 |

| Task Mode | Task Name | Duration | Start | Finish |
|-----------|----------------------------|----------|--------------|--------------|
| ▶ | Web Application testing | 4.5 days | Fri 08/01/16 | Thu 14/01/16 |
| ▶ | Create test plan | 1 day | Fri 08/01/16 | Mon 11/01/16 |
| ▶ | Test the web application | 1 day | Mon 11/01/16 | Tue 12/01/16 |
| ▶ | Review the results | 0.5 days | Tue 12/01/16 | Tue 12/01/16 |
| ▶ | Fix if necessary | 2 days | Wed 13/01/16 | Thu 14/01/16 |
| ▶ | Evaluation Report | 5 days | Fri 15/01/16 | Thu 21/01/16 |
| ▶ | Reflection Report | 5 days | Fri 22/01/16 | Thu 28/01/16 |
| ★? | Project Deliverables | | | |
| ▶ | Demonstration of Prototype | 1 day? | Mon 07/12/15 | Mon 07/12/15 |
| ▶ | Initial Contextual Report | 1 day? | Fri 18/12/15 | Fri 18/12/15 |
| ▶ | Interim Report | 1 day? | Sat 13/02/16 | Sat 13/02/16 |
| ▶ | Demonstration and Viva | 6 days? | Mon 07/03/16 | Mon 14/03/16 |
| ▶ | Final Report Upload | 1 day? | Mon 18/04/16 | Mon 18/04/16 |
| ★? | Other Course Deliverables | | | |
| ▶ | WAD CW1 | 20 days | Tue 10/11/15 | Sat 05/12/15 |
| ▶ | WAD CW2 | 10 days? | Sun 27/12/15 | Thu 07/01/16 |
| ▶ | SEM CW | 10 days? | Tue 10/11/15 | Sat 21/11/15 |

| 08 Nov '15 | 22 Nov '15 | 06 Dec '15 | 20 Dec '15 | 03 Jan '16 | 17 Jan '16 | 31 Jan '16 | 14 Feb '16 | 28 Feb '16 | 13 Mar '16 | 27 Mar '16 | 10 Apr '16 |
|-----------------------------|------------|------------|------------|-----------------------------|------------|-------------------------------|------------|----------------|-----------------|-----------------|------------|
| Software Development | | | | | | | | | | | |
| Sun 01/11/15 - Mon 18/04/16 | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| Investigation Report | | | | Design Documentation | | Product Implementation | | Testing | Evaluati | Reflecti | |
| Mon 02/11/15 - Mon 23/11/15 | | | | Tue 24/11/15 - Fri | | Mon 14/12/15 - Tue 05/01/16 | | Tue | Fri | Fri | Demonstr |
| | | | | | | | | | | | Mon |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | </td | | |

APPENDIX B - Online Questionnaire

1 Questionnaire Form

*Required

How good do you think you are in Programming? *

Even if it is only one language, that would be fine.

- No knowledge
- Beginner
- Intermediate
- Advanced
- Expert

Which Programming languages do you know already? *

- Java
- C#
- JavaScript
- Other:

Could you rate your level of confidence with the language/s you have chosen above? *

Please choose 5, if you say "I don't know" or "I am not sure".

0 1 2 3 4 5 6 7 8 9 10

Bad/Struggling Very good/No problems

Which Programming language would you like to learn next or most? *

- Java
- C#
- JavaScript
- Other:

Do you have struggle with any of the following? *

- Operators and Expressions
- Conditional Statements
- Loops
- Arrays
- Other:

How do you think the traditional teaching of programming might be improved?

Do you think little games implemented in such application will make you more motivated to learn/practice a Programming language? *

- Yes
- No

Do you think quizzes in such application will make you more motivated to learn/practice a Programming language? *

- Yes
- No

What sort of game would be interesting to see in such environment? *

Any particular one that is not mentioned here? Please fill the box for the purpose.

- Puzzle with drag and drop
- Crossword puzzle
- Question challenge
- Labyrinth
- Card game

Any particular game that you would like to see, which is not mentioned above?

What puts you off mostly when you are to study/practice programming? *

- I am being distracted by other things
- It is not fun
- Lack of time
- I don't see point in learning programming
- The material taught is not relevant to my expectations
- Programming seems very complicated to me
- Other:

Which of the following online learning/self-study environments have you used? *

- AppInventor
- Scratch
- CODE.org
- CodinGame
- CodeCombat
- Other:

Could you rate your experience with the learning environment/s you have chosen above? *

Were they helpful and motivating to you or not? Please chose 5 if you say "I don't know" or "I am not sure".

0 1 2 3 4 5 6 7 8 9 10

Not at all Very helpful/motivating

What do you like the most about the learning experience in the environments you chose above?

What was the worst about them?

What would you like to see that wasn't included? How do you think they could be improved to enhance your learning experience?

Do you like the idea and goal of this project? *

0 1 2 3 4 5

No Yes

Would you have any suggestions to improve?

2 Consent Form

**A Study carried out by Gyokay Ali,
Final Year BSc Software Engineering student at the University of Greenwich**

The aim of this project is to develop an innovative learning application/website for Year 1 CIS students to learn programming basics by “doing”. The long term goal is to motivate learners. The proposed web application will let the students to practice a chosen programming language on their own. The exact programming language and motivating methods are not decided yet, but it will be clear after this questionnaire. Basically, help me to help you.

Please read the disclaimer carefully and if you accept them you can move on to the questions. If you have any queries about the questionnaire or just want to know the results and final product at the end, please feel free to contact me at ag306@gre.ac.uk and I will get back to you ASAP.

DISCLAIMER

Thank you for agreeing to participate in this study, which is being conducted by Gyokay Ali, a student at the University of Greenwich, as part of the project for their BSc Software Engineering. The project is supervised by Dr Yasmine Arafa.

The Questionnaire:

- As part of this study, you will be asked to fill out a questionnaire.
- The purpose of the study is to collect data about students' experiences with various programming languages. To check what is a potential method that will motivate them to learn or practice one. And to get opinion and suggestions from the students, if have any possible.
- While you are under no obligation to answer any of the questions we would appreciate if you could provide an answer to all of the questions.
- The entire process should take about 5 minutes at the most of your time.
- If at any time you wish to withdraw from the study, you can do so.

What we will do with your answers:

- The answers you provide will be analysed and the data will be stored.
- We will keep the data for research purposes only. Some of the data we have gathered may be published as part of the project report.
- All data gathered will be destroyed on completion of the project.

For further information about this study please contact:

Gyokay Ali

Email: ag306@gre.ac.uk

Dr Yasmine Arafa

Faculty of Architecture, Computing and Humanities

University of Greenwich,

Old Royal Naval College, London SE10 9LS

Tel: 020 8331 8388

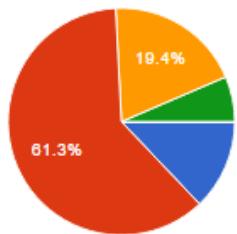
Email: Y.Arafa@gre.ac.uk

Thank you for taking the time to complete this questionnaire

3 Summary of Responses

Figure 1

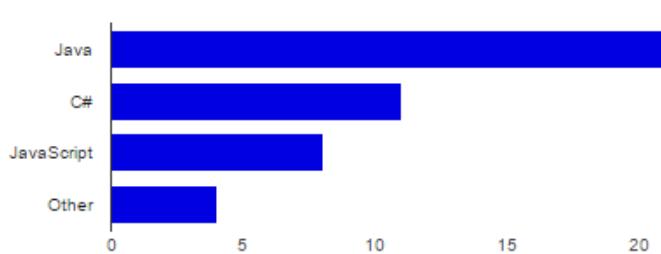
How good do you think you are in Programming?



| | | |
|--------------|----|-------|
| No knowledge | 4 | 12.9% |
| Beginner | 19 | 61.3% |
| Intermediate | 6 | 19.4% |
| Advanced | 2 | 6.5% |
| Expert | 0 | 0% |

Figure 2

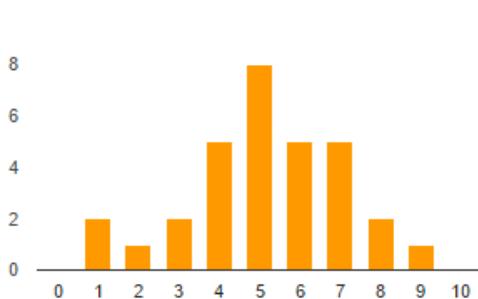
Which Programming languages do you know already?



| | | |
|------------|----|-------|
| Java | 21 | 72.4% |
| C# | 11 | 37.9% |
| JavaScript | 8 | 27.6% |
| Other | 4 | 13.8% |

Figure 3

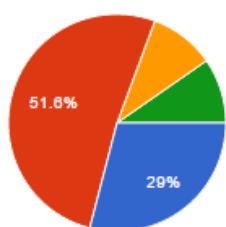
Could you rate your level of confidence with the language/s you have chosen above?



| | | |
|------------------------|----|-------|
| Bad/Struggling: | 0 | 0% |
| 1 | 2 | 6.5% |
| 2 | 1 | 3.2% |
| 3 | 2 | 6.5% |
| 4 | 5 | 16.1% |
| 5 | 8 | 25.8% |
| 6 | 5 | 16.1% |
| 7 | 5 | 16.1% |
| 8 | 2 | 6.5% |
| 9 | 1 | 3.2% |
| Very good/No problems: | 10 | 0% |

Figure 4

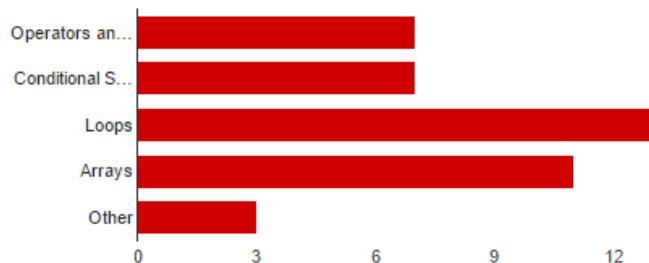
Which Programming language would you like to learn next or most?



| | | |
|------------|----|-------|
| Java | 9 | 29% |
| C# | 16 | 51.6% |
| JavaScript | 3 | 9.7% |
| Other | 3 | 9.7% |

Figure 5

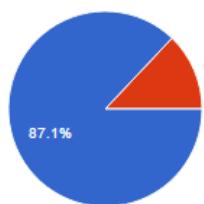
Do you have struggle with any of the following?



| | | |
|---------------------------|----|-------|
| Operators and Expressions | 7 | 30.4% |
| Conditional Statements | 7 | 30.4% |
| Loops | 13 | 56.5% |
| Arrays | 11 | 47.8% |
| Other | 3 | 13% |

Figure 6

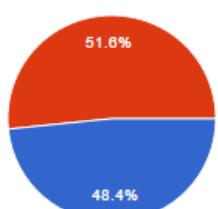
Do you think little games implemented in such application will make you more motivated to learn/practice a Programming language?



| | | |
|-----|----|-------|
| Yes | 27 | 87.1% |
| No | 4 | 12.9% |

Figure 7

Do you think quizzes in such application will make you more motivated to learn/practice a Programming language



| | | |
|-----|----|-------|
| Yes | 15 | 48.4% |
| No | 16 | 51.6% |

Figure 8

What sort of game would be interesting to see in such environment?

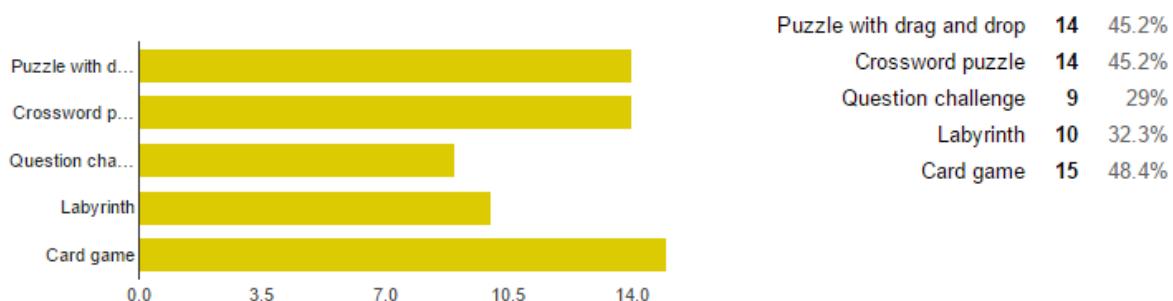


Figure 9

What puts you off mostly when you are to study/practice programming?

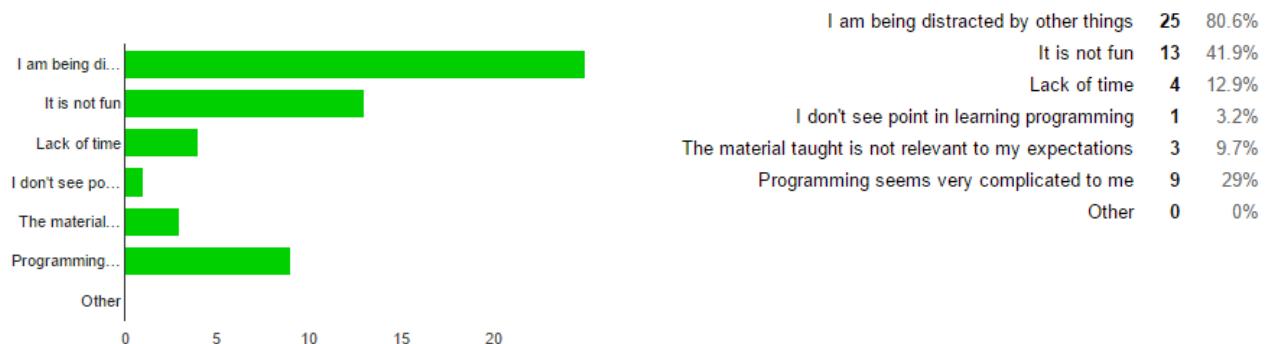


Figure 10

Which of the following online learning/self-study environments have you used?

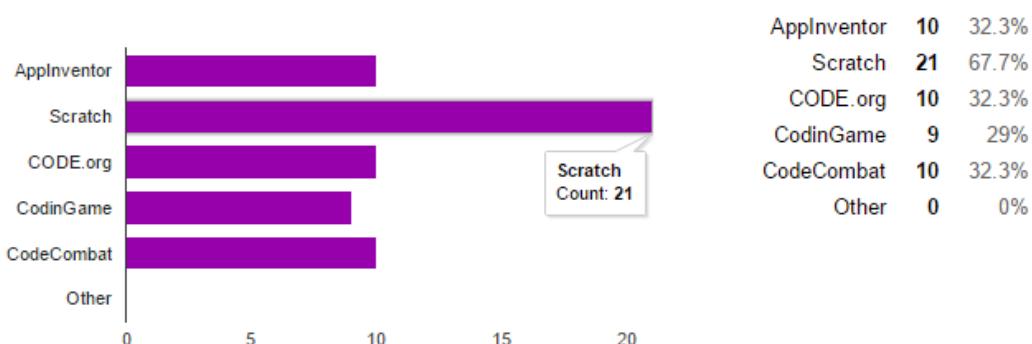


Figure 11

Could you rate your experience with the learning environment/s you have chosen above?

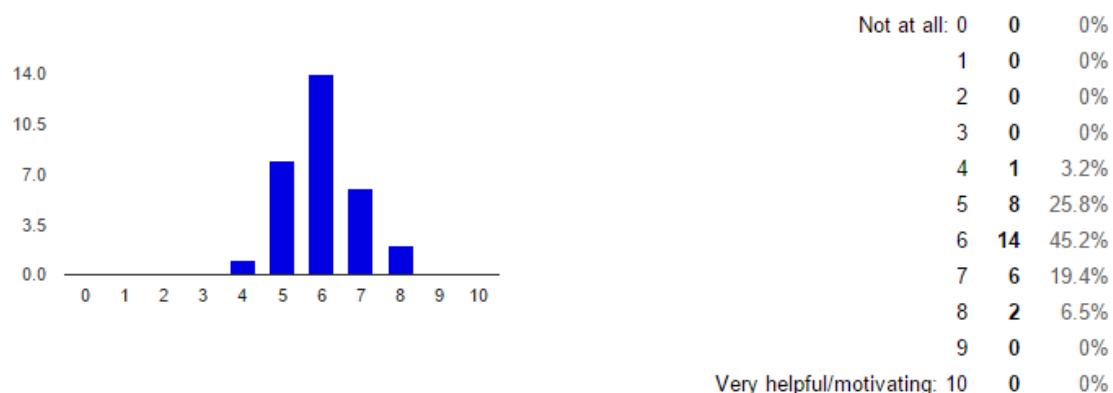
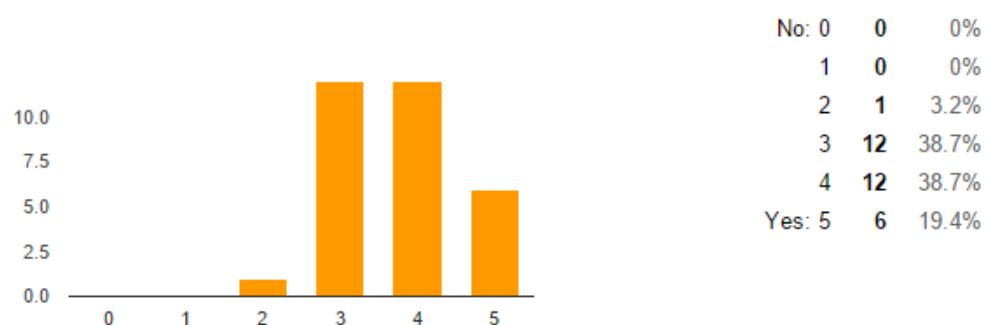


Figure 12

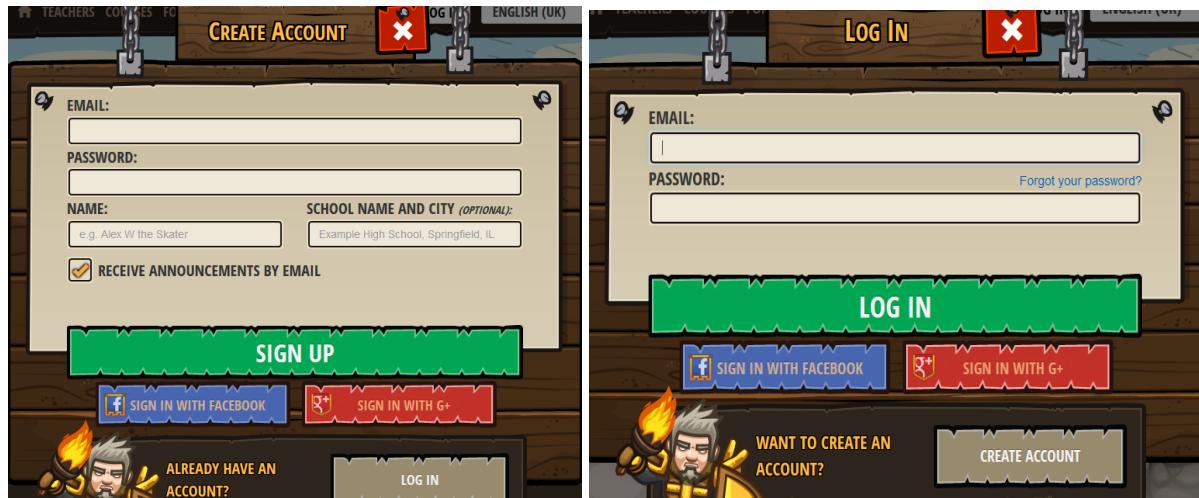
Do you like the idea and goal of this project?



APPENDIX C - Existing Products Imagery

1 CodeCombat

Registration / Login



Community

CODECOMBAT COMMUNITY

Check out the ways you can get involved below and decide what sounds the most fun. We look forward to working with you!

LEVEL EDITOR

Use the CodeCombat [Level Editor](#) to create and edit levels. Users have created levels for their classes, friends, hackathons, students, and siblings. If create a new level sounds intimidating you can start by forking one of ours!

THANG EDITOR

We call units within the game 'thangs'. Use the [Thang Editor](#) to modify the CodeCombat source artwork. Allow units to throw projectiles, alter the direction of an animation, change a unit's hit points, or upload your own vector sprites.

ARTICLE EDITOR

See a mistake in some of our docs? Want to make some instructions for your own creations? Check out the [Article Editor](#) and help CodeCombat players get the most out of their playtime.

FIND US ON THESE SITES

CONTRIBUTE TO THE PROJECT

Levels



Hero Inventory / Instructions

This screenshot shows the inventory and a specific level goal. On the left, the inventory panel shows a character equipped with a 'SIMPLE SWORD'. The sword has a damage of 6 (12.0 DPS) and grants the 'attack' skill. The inventory also lists 'RESTRICTED' and 'LOCKED' items. On the right, the 'Goals' panel for 'FIRE DANCING' shows a castle interior with fireballs falling from above. The goal is to 'Dodge the fireballs forever. Under 4 statements.' A blue arrow points to the character, and a red circle highlights a fireball. Below the panel, a large green 'START LEVEL' button is visible.

Gameplay

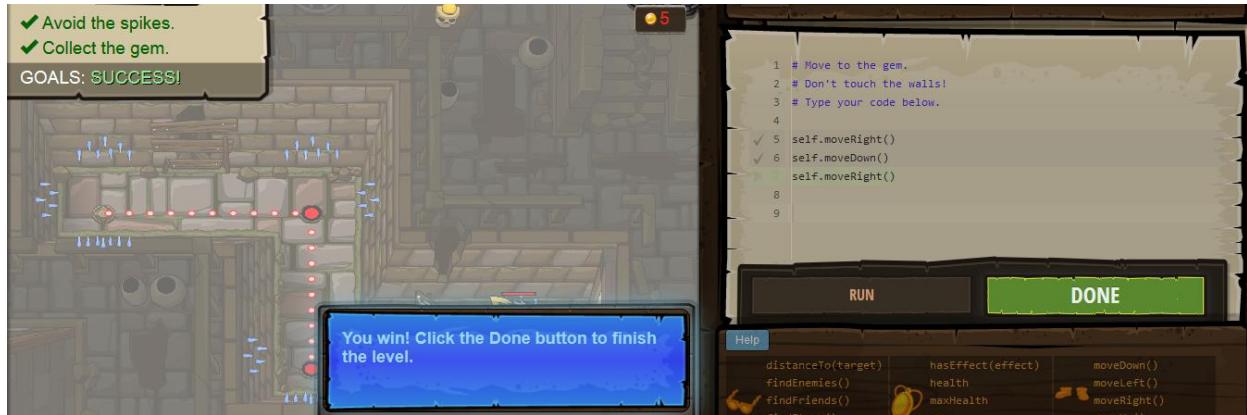
This screenshot shows the game in progress. The hero, Tharin, is in a castle corridor dodging fireballs. The top left shows the 'LEVEL FIRE DANCING' and 'GOALS FAILING' status. The top right shows the 'GAME MENU' and 'CURRENT METHOD PLAN'. The bottom right shows the code editor with a 'while(true)' loop. The 'RUN' button is at the bottom right of the code editor. The bottom left shows the hero's stats: THARIN with 183 health.

```

1 // Code normally executes in the order it's written.
2 // Loops repeat a block of code multiple times.
3
4 while(true) {
5     this.moveRight();
6     // Add a moveLeft command to the loop here
7 }
8
9

```

Correct code



Victory / Survey



2 CodinGame

Instructions 1

The main area displays the text "Choose a programming language". Below it, a message says "You can choose a language you already know or you can also learn a new language here. Ever tried "Go"?"

To the right, a sidebar lists programming languages: C#, C++, Java, Javascript, and Python3. A dropdown menu is open over the Python3 option.

At the bottom right, a chat log shows a message from a user named "P" that says "[P] has joined #codingame". The log also includes a list of users who have joined: !MaximeCG, !Nonofr, @CGBot, SERGE_, AbiTl, AdamTrnka, and Add_nickname.

At the bottom left, there is a "PREVIOUS STEP" button.

Instructions 2

Time to code!

OK, let's tell your program to shoot the closest Alien. Copy/Paste this code at the right place in the code editor.

```
if (dist1 < dist2) {
    Console.WriteLine(enemy1);
} else {
    Console.WriteLine(enemy2);
```

GOT IT!

[← PREVIOUS STEP](#)

```
YOUR CODE | C# | C
5 static void Main(string[] args)
6 {
7     // game loop
8     while (true)
9     {
10        string enemy1 = Console.ReadLine(); // name of enemy 1
11        int dist1 = int.Parse(Console.ReadLine()); // distance to enemy 1
12        string enemy2 = Console.ReadLine(); // name of enemy 2
13        int dist2 = int.Parse(Console.ReadLine()); // distance to enemy 2
14
15        // Write an action using Console.WriteLine()
16
17        // Enter the code here
18
19        if (dist1 < dist2) {
20            Console.WriteLine(enemy1);
21        } else {
22            Console.WriteLine(enemy2);
23        }
24    }
}
```

Gameplay / Code editor

Onboarding



16/16

BACK

FRIENDS

Best score N/A

```
YOUR CODE | C# | C
5 static void Main(string[] args)
6 {
7     // game loop
8     while (true)
9     {
10        string enemy1 = Console.ReadLine(); // name of enemy 1
11        int dist1 = int.Parse(Console.ReadLine()); // distance to enemy 1
12        string enemy2 = Console.ReadLine(); // name of enemy 2
13        int dist2 = int.Parse(Console.ReadLine()); // distance to enemy 2
14
15        // Write an action using Console.WriteLine()
16
17        // Enter the code here
18
19        if (dist1 < dist2) {
20            Console.WriteLine(enemy1);
21        } else {
22            Console.WriteLine(enemy2);
23        }
24    }
}
```

Test cases

| | | |
|----|-----------------|-----------------|
| 01 | Imminent danger | ▶ PLAY TESTCASE |
|----|-----------------|-----------------|

Actions

▶ PLAY ALL TESTCASES

✓ SUBMIT

Results

LAST REPORT

FINAL SCORE 100%

SETTINGS

RESULTS

HINTS

FORUM

BACK

Make your solution available to the community PUBLISH

Unlocked achievements

| | |
|---|--------------------------|
|  | Onboarding Master +15 CP |
|  | 50% Onboarding +15 CP |

HISTORY

Community

COMMUNITY SOLUTIONS Learn new languages and tricks from others.

What's next?

TRY THE NEXT GAME Power of Thor

CLASH OF CODE It's time for a coding match!

CODE A BOT Fight in the bot arena of "Back to the Code".

Puzzles and descriptions

SINGLE PLAYER OPTIMIZATION CODE GOLF

The Descent
Searching in a list.
★ 0/50
⌚ 0/2

Skynet: the Chasm
Conditions.
★ 0/50
⌚ 0/2

Temperatures
Searching in a list.
★ 0/50
⌚ 0/2

Mars Lander - Level 1
Conditions, Speed regulation.
★ 0/75
⌚ 0/3

Skynet: the Chasm
Conditions.

0%
FACEBOOK TWITTER GOOGLE+ FIND FRIENDS +

SOLVE IT

Description
You wouldn't think of anything easier than making a terminator bike jump over a hole between two sides of a bridge, would you? It's all about finding the right balance between acceleration and deceleration...
Anyway, the resistance is counting on you: you must capture this bike!

Programming AI Bot

Back to the Code - Multi game

YOUR CODE | C# | ▾

```

12  int i;
13  string[] inputs;
14  int opponentCount = int.Parse(Console.ReadLine()); // Opponent count
15
16  // game loop
17  while (true)
18  {
19      int gameRound = int.Parse(Console.ReadLine());
20      inputs = Console.ReadLine().Split(' ');
21      int x = int.Parse(inputs[0]); // Your x position
22      int y = int.Parse(inputs[1]); // Your y position
23      int backInTimeLeft = int.Parse(inputs[2]); // Remaining back in time
24      for (int i = 0; i < opponentCount; i++)
25      {
26          inputs = Console.ReadLine().Split(' ');
27          int opponentX = int.Parse(inputs[0]); // X position of the opponent
28          int opponentY = int.Parse(inputs[1]); // Y position of the opponent
29          int opponentBackInTimeLeft = int.Parse(inputs[2]); // Remaining back in time
30      }
31  }
32
33 }
```

Players

| | | | |
|---------|---------|------------|------------|
| | | + | + |
| Gyokito | Default | Add player | Add player |
| ★ N/A | ★ N/A | DELETE | |

Actions

- ▶ PLAY MY CODE
- ⟳ REPLAY IN SAME CONDITIONS
- ✓ SUBMIT

Console output

Challenges for advanced players / programmers

The screenshot shows a user profile for 'Gyokito' with 0 notifications and a 'Chat' option. Social sharing buttons for Facebook, Twitter, Google+, and friends are at the top. A sidebar on the left includes links for HOME, PUZZLES, AI BOTS, CLASH OF CODE (which is selected), CONTESTS, LEADERBOARDS, and COMMUNITY. The main area features a large, stylized title 'CLASH OF CODE' with a subtitle 'Challenge your pals on short coding battles.' Below the title are four boxes: 'Flash 5mn' (green), 'Coffee Break 10mn' (teal), 'Reflection 20mn' (blue), and 'Private clash' (gray).

Contests

The screenshot shows a user profile for 'Gyokito' with 0 notifications and a 'Chat' option. A sidebar on the left includes links for HOME, PUZZLES, AI BOTS, CLASH OF CODE, CONTESTS (selected), LEADERBOARDS, and COMMUNITY. It also has a 'ARE YOU A COMPANY?' button. The main section shows the 'NEXT CONTEST' (8 days away, one multiplayer game, free registration, open to all). To the right, there's a 'Win prizes' section with three categories: 1st Place (Pack PS4 + Star Wars Battlefront), 2nd Place (LEGO Mindstorms: Star Wars Droid Developer Kit), and 3rd Place (Darth Vader Toaster). There's also a 'Win your T-Shirt' section for global ranking.

3 CodeMonkey

Information

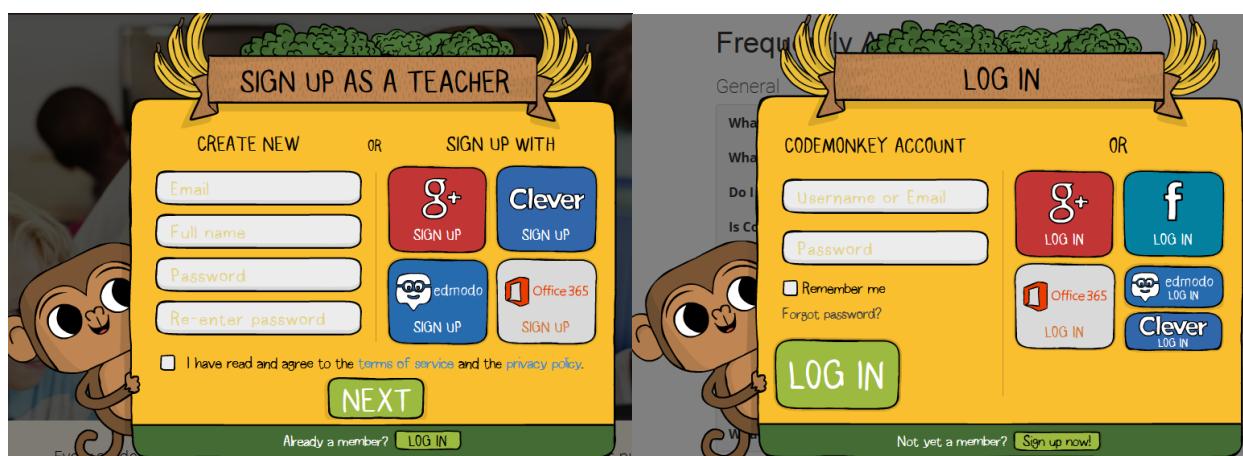
Our Programs

Click to find out more:

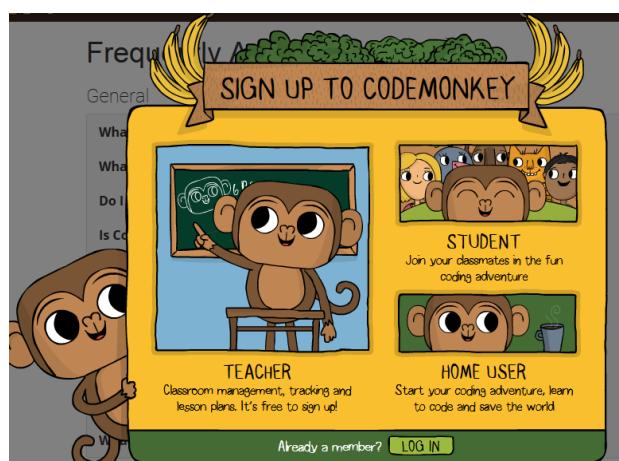


| | | |
|---|---|--|
| Home User + Access to all challenges. | Teacher + Access to all challenges. + Up to 30 users. + Student tracking. | School + Access to all challenges. + Up to 180 users. + Student tracking. + Online support. |
|---|---|--|

Registration / Login



More info before sign up



Gameplay and coding

The image consists of three vertically stacked screenshots from the CodeMonkey game.

Top Screenshot: CHALLENGE #0. A boy character says: "Oh no! We must get our bananas back! Press **RUN!** to get the monkey moving!" An "OK" button is at the bottom right. The status bar shows "1 step 15".

Middle Screenshot: CHALLENGE #0 COMPLETE! A monkey wearing a party hat and holding a banana is shown with three yellow stars. A speech bubble says "Nice clicking!". A "REPLAY" button is on the right. The status bar shows "X 1".

Bottom Screenshot: CHALLENGE #7. A boy character says: "You can also use **turn** with a number. Like **turn 45** or **turn -90**". An "OK" button is at the bottom right. The status bar shows "1 turn 30" and "2 step 14".

Codemonkey

CHALLENGE #60

```

1 for b in bananas
2 ... turnTo b
3 ... step distanceTo b

```

RUN! C

step turn turnTo distanceTo say for

Codemonkey

CHALLENGE #89

```

1 chase = (t) ->
2 ... until near t
3 turnTo t
4 step 1
5 safeCollect = (m) ->
6 ... until cat.sleeping()
7 ..... wait()
8 ..... chase m
9 ..... grab()
10 ..... chase pile
11 ..... drop()
12 ...
13 safeCollect match

```

STOP C

step turnTo distanceTo grab drop function until
near sleeping wait chase safeCollect

Codemonkey

CHALLENGE #107

```

1 for b in bananas
2 goat.goto b
3 goat.hit()
4
5 # if else here:
6 |

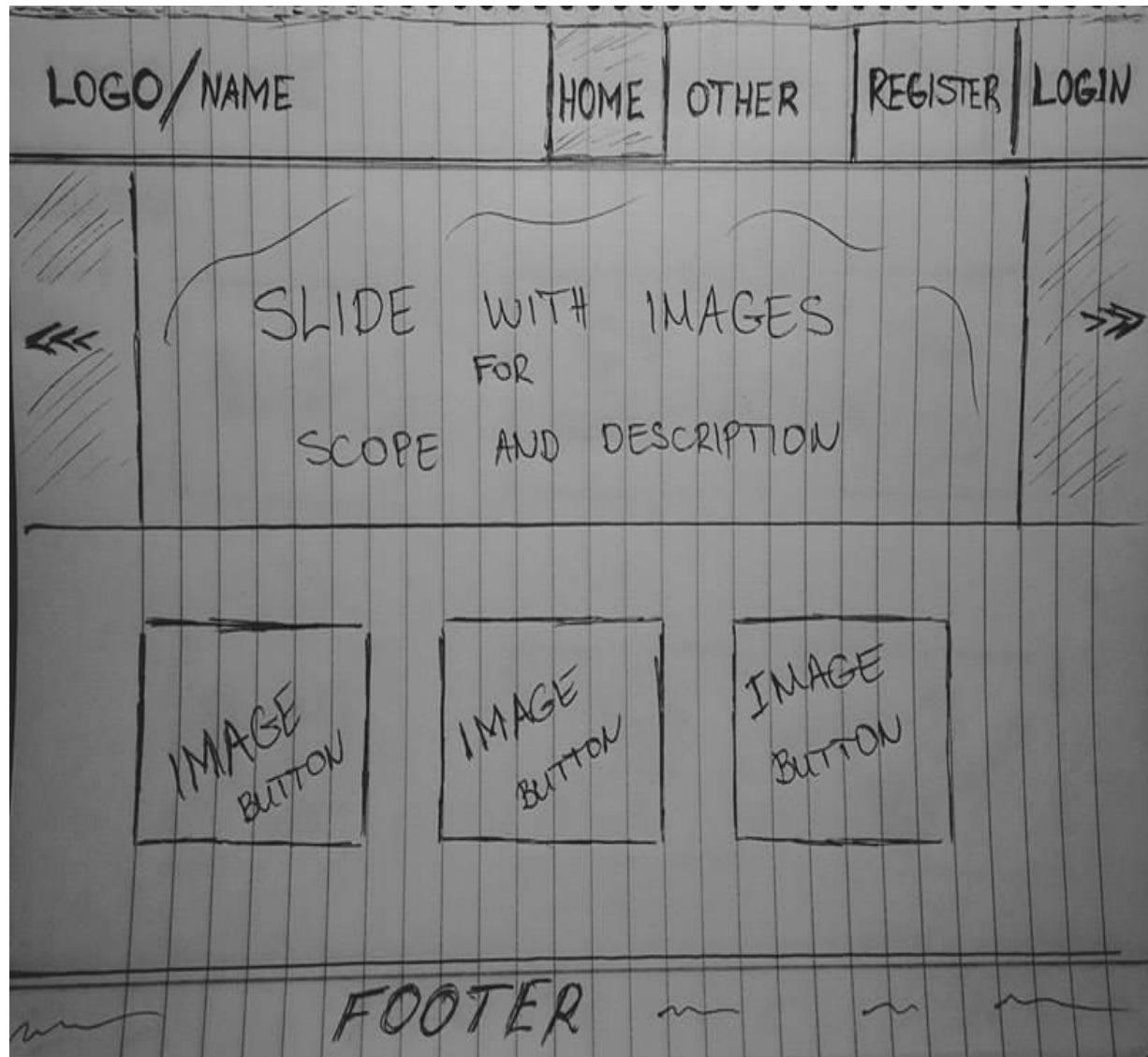
```

RUN! C

say goto hit frozen green if else

APPENDIX D - Low Fidelity Wireframe Sketches

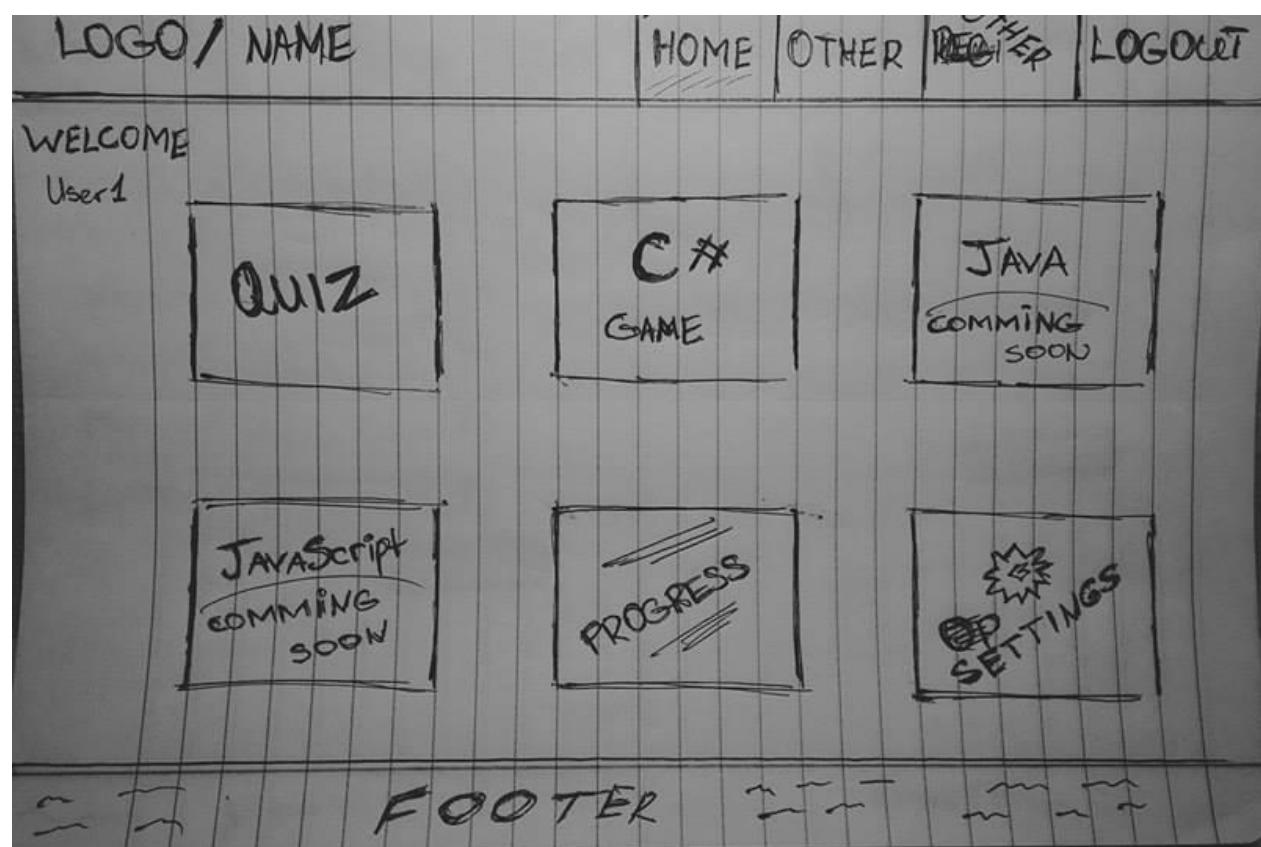
Home / Default page view



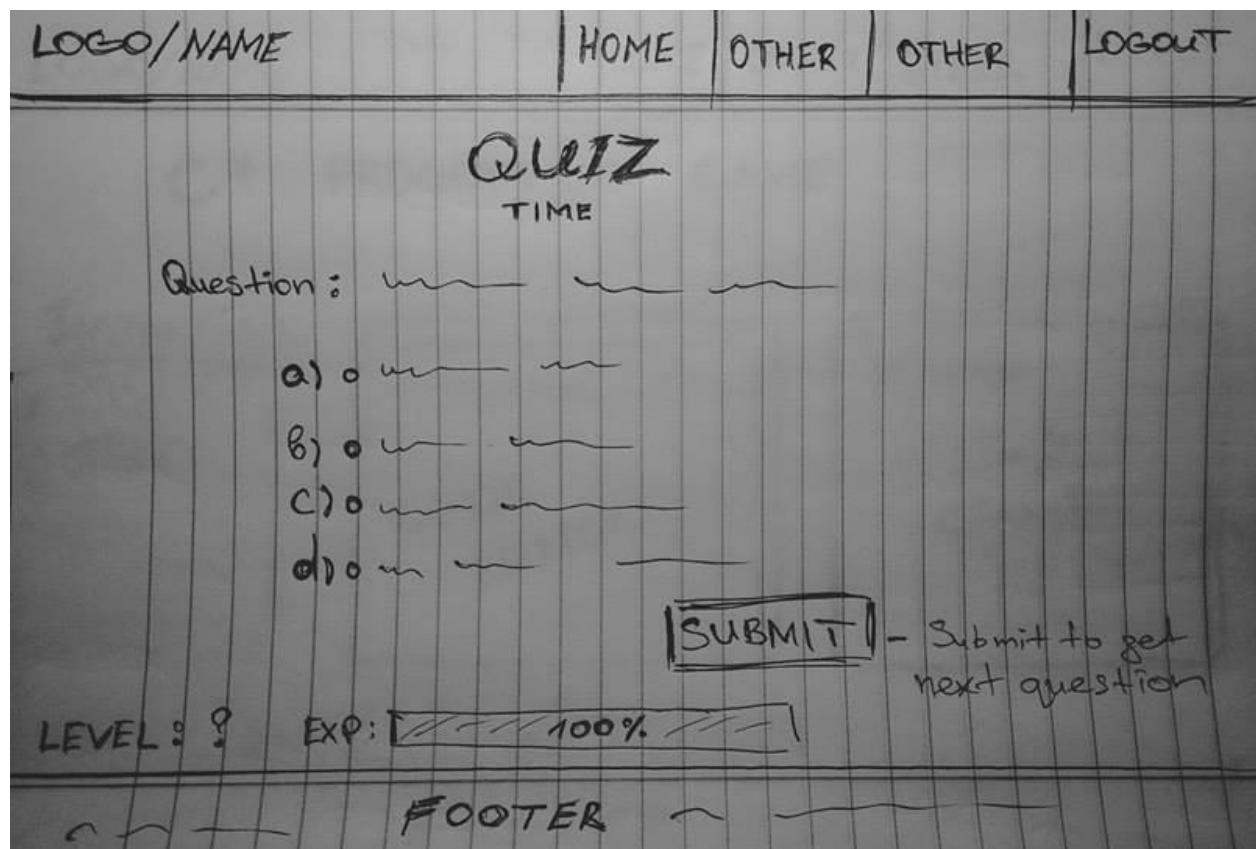
Register / Login page view

| | | | | | |
|------------------|-------|-------|--------------|-------|-------|
| LOGO/NAME | OTHER | OTHER | LOGO/NAME | OTHER | OTHER |
| REGISTER | | | LOGIN | | |
| username: | []* | | username: | []* | |
| email : | []* | | password: | []* | |
| password : | []* | | | | |
| conf. password : | []* | | | | |
| Submit | | | Login | | |
| FOOTER | | | FOOTER | | |

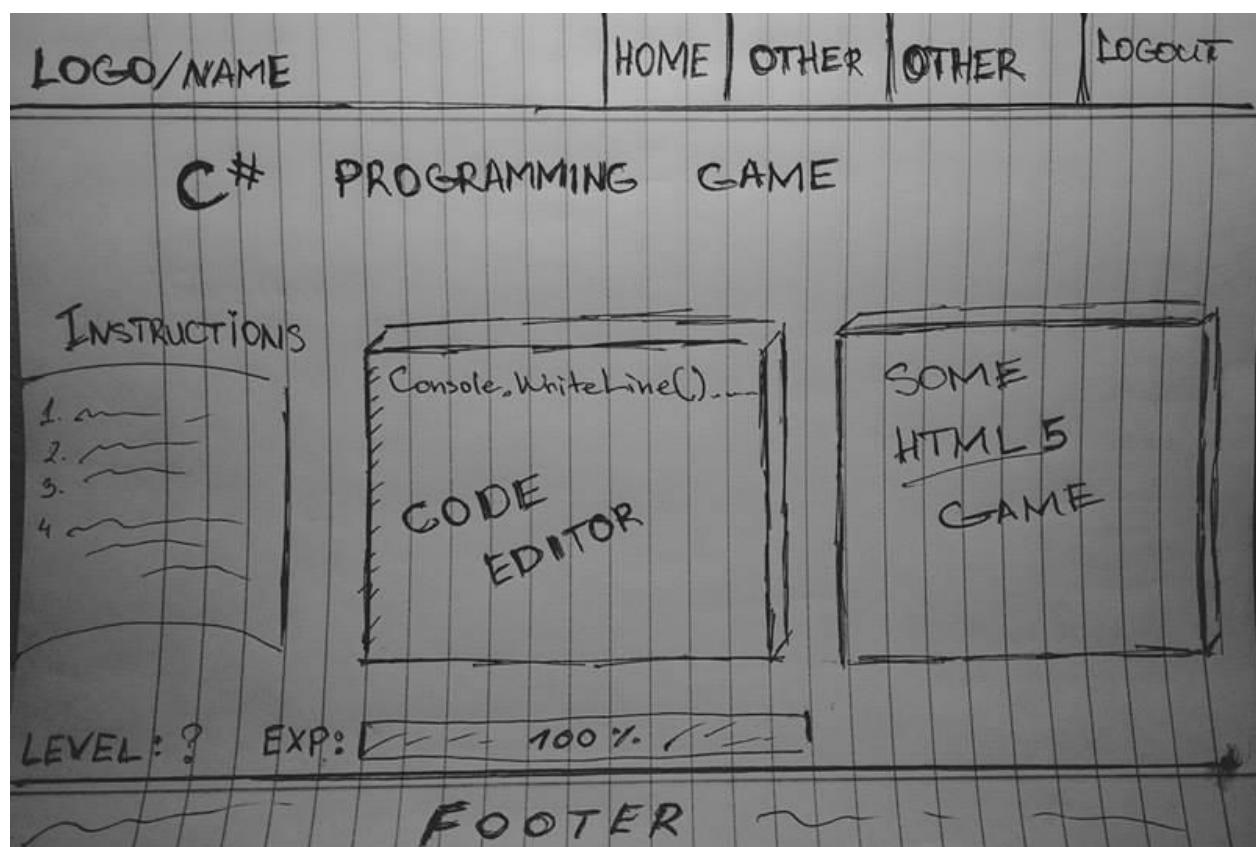
Member's area page view

| | | | | |
|------------------|--|-------|---------------------------------|--------|
| LOGO / NAME | HOME | OTHER | RE MEMBER | LOGOUT |
| WELCOME User1 |  | | | |
| FOOTER | | | | |

Quizzes page view



Games page view



Progress / Settings page view

| | | | | | | | |
|---|------|-------|--------|--|------|-------|--------|
| LOGO/NAME | HOME | OTHER | LOGOUT | LOGO/NAME | HOME | OTHER | LOGOUT |
| PROGRESS | | | | SETTINGS | | | |
| LEVEL : 5 EXP : 59 %. OTHER : - - - | | | | <u>CHANGE PASSWORD ?</u> OLD PASSWORD : <input type="text"/> NEW PASSWORD : <input type="text"/> CONFIRM PASSWORD : <input type="text"/> <input type="button" value="Submit"/> | | | |
| < BACK | | | | FOOTER | | | |

APPENDIX E - Prototypes

1 PROTOTYPE A

Default Page

THE MOTIVATION CODE

Home Start Programming Extras Register Login

Programming is fun!

FEEL BORED? LET'S GET MOTIVATED!

Quizzes Learn C# while playing games Learn Java while playing games Learn JS while playing games

Learn C# while playing games Learn Java while playing games Learn JS while playing games More coming soon...

DETAILS

Final Year Project
Developed by
Gyokay Ali
BSc (Hons) Software Engineering
University of Greenwich

QUICK LINKS

[Home Page](#)
[Start Programming](#)
[Contact Us](#)

PROJECT AIM

The aim of this project was to develop an innovative learning application for Year 1 CIS students to learn programming basics by "doing". The long term goal is to motivate learners.

FOLLOW US ON SOCIAL MEDIA

f t in g+ s t

2 PROTOTYPE B

Default page

THE MOTIVATION CODE

Home My Area ▾ Programming ▾ Extras Register Login

Learn and practice programming while playing fun games!

Welcome to The Motivation Code

...

Start now

FEEL BORED? LET'S GET MOTIVATED!

test your knowledge

Programming QUIZ

Take a Quiz

C# Programming

Learn C# while playing games

Java Programming

Coming soon

Learn Java while playing games

JavaScript Programming

Coming soon

Learn JS while playing games

Login page

LOGIN

Please enter your login details

Username

Password

LOGIN Not registered?

Registration page

REGISTRATION

Please fill in your details below

Username

Email

Password

Confirm password

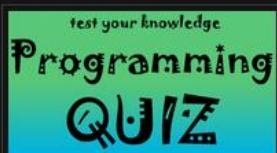
SUBMIT

Members' My Area page

THE MOTIVATION CODE

Home My Area Programming Extras Account

MAYBE YOU SHOULD START WITH A SHORT QUIZ? OR JUST DIVE INTO PROGRAMMING...



Take a Quiz



Learn C# while playing games

Coming soon!



Other



Check progress



Account settings

DETAILS

Final Year Project
Developed by
[Gyokay Ali](#)

QUICK LINKS

[Home Page](#)
[About](#)

PROJECT AIM

The aim of this project was
to develop an innovative
learning application for Year 1

FOLLOW US ON SOCIAL MEDIA



3 PROTOTYPE C (Final)

This will contain only the features added on top of Prototype B

Programming Games page



Share your suggestions/ideas page

PLEASE SHARE WITH US YOUR IDEAS AND SUGGESTIONS!

SUBMIT

Settings page (change password)

SETTINGS

You can change your password using the form below

Old password

New password

New password again

SUBMIT

Quiz page



\$125,000
\$64,000
\$32,000
\$16,000
\$8,000
\$4,000
\$2,000
\$1,000
\$500
\$300
\$200
\$100

\$0.00

Which of the following converts a type to a Boolean value, where possible in C#?

A: ToBoolean C: ToChar
B: ToSingle D: ToDateTime

Next question

Which of the following converts a type to a small floating point number in C#?

A:ToInt64 C: ToSingle
B: ToSByte D:ToInt32

End of game (wrong answer)

Game Over!

Contribute page (GitHub + Admin login)

WOULD YOU LIKE TO CONTRIBUTE TO US?



Fork in our source code

WOULD YOU LIKE TO ADD NEW QUESTIONS TO OUR QUESTION BANK?

You need to login with a privileged account. Don't have? Please get in [touch with us!](#)

Username

Password

LOGIN

Adding questions to Question Bank by the Admin

ADD QUESTIONS

Select question difficulty

Select programming language

Question:

Points given:

Answers:
Only one correct answer please!

1: correct ?

2: correct ?

3: correct ?

4: correct ?

SUBMIT

APPENDIX F - Testing

White-box Testing

| Test case | Test description | Expected result | Actual result is same as expected (YES/NO) |
|-----------|---|--|--|
| 1 | Run the application | Should run and display the Default page | Yes |
| 2 | Navigate to 'My Area' without authentication | Should not allow the user to access the page and redirect to Login | Yes |
| 3 | Navigate to 'Take Quiz' without authentication | Should not allow the user to access the page and redirect to Login | Yes |
| 4 | Navigate to 'C# Game' without authentication | Should not allow the user to access the page and redirect to Login | Yes |
| 5 | Navigate to 'About' without authentication | Should allow access to the page and will display About | Yes |
| 6 | Navigate to 'Contact us' without authentication | Should allow access to the page and will display Contact us | Yes |
| 7 | Navigate to 'Contribute without authentication | Should allow access to the page and will display Contribute | Yes |
| 8 | Navigate to 'Contribute/GitHub' without authentication | Should allow access to the page and will navigate to github.com | Yes |
| 9 | Navigate to 'Contribute/Admin' without authentication | Should allow access to the page and display error message | Yes |
| 10 | Navigate to 'Contribute/Admin' and enter random login details | Should allow access to the page and display error message | Yes |
| 11 | Navigate to 'Contribute/Admin' and enter correct Admin login details | Should allow access to the page and display Question Bank | Yes |
| 12 | At 'QuestionBank' select question difficulty, language, write question and fill in the answer. Specify correct answer | Should allow the Admin to add the question to the Question Bank and save to database | Yes |
| 13 | At 'QuestionBank' leave a blank text | Should not allow the question to be | Yes |

| | | | |
|----|---|--|----------------------|
| | field | added to the Question Bank and will be displayed error message | |
| 14 | At ‘QuestionBank’ don’t specify the correct answer (all checkboxes unchecked) | Should not allow the question to be added to the Question Bank and will be displayed error message | Yes |
| 15 | Navigate to ‘Register’ and leave all text fields blank | Should not allow the user to register and display the required fields | Yes |
| 16 | Navigate to ‘Register’ and enter 2 different passwords | Should not allow the user to register as the passwords do not match | Yes |
| 17 | Navigate to ‘Register’ and enter email in incorrect format | Should not allow the user to register and display error message | Yes |
| 18 | Navigate to ‘Register’ and acceptable account details | Should register the user and navigate to Login | Yes |
| 19 | Navigate to ‘Login’ and leave all text fields blank | Should not allow the user to login and display the required fields | Yes |
| 20 | Navigate to ‘Login’ and enter incorrect data | Should not allow the user to login and display error message | Yes |
| 21 | Navigate to ‘Login’ and enter correct data | Should allow the user to login and navigate to My Area | Yes |
| 22 | Navigate to ‘Share you ideas’ and leave blank text field | Should not allow to submit the suggestion made by the user | Yes |
| 23 | Navigate to ‘Share you ideas’ and submit some random data | Should allow the user to submit the suggestion and save to database | Yes |
| 24 | Navigate to ‘GameCSharp’ and play the game using arrow keys | Should allow the user to access the page and play only the game using arrows | Yes |
| 25 | Navigate to ‘GameCSharp’ and play the game using programming code | Should allow the user to access the page and play the game using code | No / Not implemented |
| 26 | Navigate to ‘TheQuiz’ and start the quiz | Should allow the user to access the page and start the quiz | Yes |
| 27 | In the quiz make a wrong selection | Should end the quiz | Yes |
| 28 | In the quiz make a correct selection | Should allow the user to progress to the next question | Yes |
| 29 | Navigate to “Progress” and check current level and experience | Should allow the user to access the page and display user’s progress details | No/Not implemented |
| 30 | Navigate to ‘Settings’ and leave text | Should not allow the user to change | Yes |

| | | | |
|-----------|---|---|-----|
| | fields blank | their password and display error | |
| 31 | Navigate to 'Settings and enter not matching passwords' | Should not allow the user to change their password and display error | Yes |
| 32 | Navigate to 'Settings and expected data' | Should allow the user to change their password and navigate back to MyArea | Yes |
| 33 | Navigate to 'Logout' and exit from the application | Should allow the user to exit from the application and destroy all sessions | Yes |

Black-box Testing

| Test case | Test description | Expected result | True/False (Explain) |
|-----------|--|--|--|
| 1 | Go to Register and enter your details. | Successfully registered. | True (Redirected to Login page) |
| 2 | Go to Login and use your newly created details | Successfully authenticated | True (Redirected to MyArea page) |
| 3 | Go to Settings and change password | Password updated | True (Redirected to MyArea page) |
| 4 | Go to Progress and check user progress | Up-to-date user progress details displayed | False (Nothing displayed) |
| 5 | Go to Share Your Ideas and submit suggestion | Successfully submitted | True (Success message displayed) |
| 6 | Go to Contribute and login to Admin area Username: admin1 Password: admin1 | Successfully authenticated | True (Redirected to QuestionBank page) |
| 7 | Add new question at QuestionBank | Successfully added | True (Success message displayed) |
| 8 | Go to C# Game and play the game | Successfully played | True (Played only by keyboard arrow keys) |
| 9 | At C# Game run some code | Successfully run | False (Codes not accepted. Does not run) |
| 10 | Go to Quiz and answer the questions | Successfully taken | True (Correct answers led to next question. Incorrect end of quiz.) |

APPENDIX G - Borrowed code

1. HTML5 Game with JavaScript – Viking and Goblins

Reference:

Hackett, M. (2011) How to make a simple HTML5 Canvas game - Lost Decade Games, *Lostdecadegames.com*, [online] Available at: <http://www.lostdecadegames.com/how-to-make-a-simple-html5-canvas-game/> (Accessed 18 April 2016).

Source code:

```
// Create the canvas
var canvas = document.createElement("canvas");
var ctx = canvas.getContext("2d");
canvas.width = 512;
canvas.height = 480;
canvas.style.left = "45%";
canvas.style.top = "60%";
canvas.style.position = "absolute";
document.body.appendChild(canvas);

// Background image
var bgReady = false;
var bgImage = new Image();
bgImage.onload = function () {
    bgReady = true;
};
bgImage.src = "Game-Goblin/images/background.png";

// Hero image
var heroReady = false;
var heroImage = new Image();
heroImage.onload = function () {
    heroReady = true;
};
heroImage.src = "Game-Goblin/images/hero.png";

// Monster image
var monsterReady = false;
var monsterImage = new Image();
monsterImage.onload = function () {
    monsterReady = true;
};
monsterImage.src = "Game-Goblin/images/monster.png";

// Game objects
var hero = {
    speed: 256 // movement in pixels per second
};
var monster = {};
var monstersCaught = 0;

// Handle keyboard controls
var keysDown = {};
```

```

addEventListener("keydown", function (e) {
    keysDown[e.keyCode] = true;
}, false);

addEventListener("keyup", function (e) {
    delete keysDown[e.keyCode];
}, false);

// Reset the game when the player catches a monster
var reset = function () {
    hero.x = canvas.width / 2;
    hero.y = canvas.height / 2;

    // Throw the monster somewhere on the screen randomly
    monster.x = 32 + (Math.random() * (canvas.width - 64));
    monster.y = 32 + (Math.random() * (canvas.height - 64));
};

// Update game objects
var update = function (modifier) {
    if (38 in keysDown) { // Player holding up
        hero.y -= hero.speed * modifier;
    }
    if (40 in keysDown) { // Player holding down
        hero.y += hero.speed * modifier;
    }
    if (37 in keysDown) { // Player holding left
        hero.x -= hero.speed * modifier;
    }
    if (39 in keysDown) { // Player holding right
        hero.x += hero.speed * modifier;
    }

    // Are they touching?
    if (
        hero.x <= (monster.x + 32)
        && monster.x <= (hero.x + 32)
        && hero.y <= (monster.y + 32)
        && monster.y <= (hero.y + 32)
    ) {
        ++monstersCaught;
        reset();
    }
};

// Draw everything
var render = function () {
    if (bgReady) {
        ctx.drawImage(bgImage, 0, 0);
    }

    if (heroReady) {
        ctx.drawImage(heroImage, hero.x, hero.y);
    }

    if (monsterReady) {
        ctx.drawImage(monsterImage, monster.x, monster.y);
    }
};

```

```

}

// Score
ctx.fillStyle = "rgb(250, 250, 250)";
ctx.font = "24px Helvetica";
ctx.textAlign = "left";
ctx.textBaseline = "top";
ctx.fillText("Goblins caught: " + monstersCaught, 32, 32);
};

// The main game loop
var main = function () {
  var now = Date.now();
  var delta = now - then;

  update(delta / 1000);
  render();

  then = now;

  // Request to do this again ASAP
  requestAnimationFrame(main);
};

// Cross-browser support for requestAnimationFrame
var w = window;
requestAnimationFrame = w.requestAnimationFrame || w.webkitRequestAnimationFrame || w.msRequestAnimationFrame || w.mozRequestAnimationFrame;

// Let's play this game!
var then = Date.now();
reset();
main();

```

2. JavaScript game – Millionaire

Available at: <https://github.com/aaronnech/Who-Wants-to-Be-a-Millionaire>

Source code:

```
/**  
 * Edits the number prototype to allow money formatting  
 *  
 * @param fixed the number to fix the decimal at. Default 2.  
 * @param decimalDelim the string to delimitate the non-decimal  
 *         parts of the number and the decimal parts with. Default  
 *         ".."  
 * @param breakdDelim the string to delimitate the non-decimal  
 *         parts of the number with. Default ","  
 * @return returns this number as a USD-money-formatted String  
 *         like this: x,xxx.xx  
 */  
Number.prototype.money = function(fixed, decimalDelim, breakDelim){  
    var n = this,  
        fixed = isNaN(fixed = Math.abs(fixed)) ? 2 : fixed,  
        decimalDelim = decimalDelim == undefined ? "." : decimalDelim,  
        breakDelim = breakDelim == undefined ? "," : breakDelim,  
        negative = n < 0 ? "-" : "",  
        i = parseInt(n = Math.abs(+n || 0).toFixed(fixed)) + "",  
        j = (j = i.length) > 3 ? j % 3 : 0;  
    return negative + (j ? i.substr(0, j) +  
        breakDelim : "") + i.substr(j).replace(/(\d{3})(?=\d)/g,  
        "$1" + breakDelim) +  
        (fixed ? decimalDelim + Math.abs(n -  
i).toFixed(fixed).slice(2) : "");  
}  
  
/**  
 * Plays a sound via HTML5 through Audio tags on the page  
 *  
 * @require the id must be the id of an <audio> tag.  
 * @param id the id of the element to play  
 * @param loop the boolean flag to loop or not loop this sound  
 */  
startSound = function(id, loop) {  
    soundHandle = document.getElementById(id);  
    if(loop)  
        soundHandle.setAttribute('loop', loop);  
    soundHandle.play();  
}  
  
/**  
 * The View Model that represents one game of  
 * Who Wants to Be a Millionaire.  
 *  
 * @param data the question bank to use  
 */  
var MillionaireModel = function(data) {  
    var self = this;  
  
    // The 15 questions of this game  
    this.questions = data.questions;
```

```

// Boolean whether mute is on or off
this.muting      = false;

// A flag to keep multiple selections
// out while transitioning levels
this.transitioning = false;

// The current money obtained
this.money = new ko.observable(0);

// The current level(starting at 1)
this.level = new ko.observable(1);

// The three options the user can use to
// attempt to answer a question (1 use each)
this.usedFifty = new ko.observable(false);
this.usedPhone = new ko.observable(false);
this.usedAudience = new ko.observable(false);

self.mute = function(){
    this.muting = !this.muting;
    var muting = this.muting;

    var src = $("#mute").attr("src");
    $("#mute").attr("src", $("#mute").attr("altimg"));

    $("#mute").attr("altimg", src)

    $("audio").each(function(){
        $(this).prop('muted', muting);
    });
}

// Grabs the question text of the current question
self.getQuestionText = function() {
    return self.questions[self.level() - 1].question;
}

// Gets the answer text of a specified question index (0-3)
// from the current question
self.getAnswerText = function(index) {
    return self.questions[self.level() - 1].content[index];
}

// Uses the fifty-fifty option of the user
self.fifty = function(item, event) {
    if(self.transitioning)
        return;
    $(event.target).fadeOut('slow');
    var correct = this.questions[self.level() - 1].correct;
    var first = (correct + 1) % 4;
    var second = (first + 1) % 4;
    if(first == 0 || second == 0) {
        $("#answer-one").fadeOut('slow');
    }
    if(first == 1 || second == 1) {

```

```

        $("#" + answer).fadeOut('slow');
    }
    if(first == 2 || second == 2) {
        $("#" + answer).fadeOut('slow');
    }
    if(first == 3 || second == 3) {
        $("#" + answer).fadeOut('slow');
    }
}

// Fades out an option used if possible
self.fadeOutOption = function(item, event) {
    if(self.transitioning)
        return;
    $(event.target).fadeOut('slow');
}

// Attempts to answer the question with the specified
// answer index (0-3) from a click event of elm
self.answerQuestion = function(index, elm) {
    if(self.transitioning)
        return;
    self.transitioning = true;
    if(self.questions[self.level() - 1].correct == index) {
        self.rightAnswer(elm);
    } else {
        self.wrongAnswer(elm);
    }
}

// Executes the procedure of a correct answer guess, moving
// the player to the next level (or winning the game if all
// levels have been completed)
self.rightAnswer = function(elm) {

    startSound('rightsound', false);
    //var bgcss = ($("#" + elm).toggleClass('correct'))
    $("#" + elm).toggleClass('correct')
    setTimeout(function(){
        self.money($(".active").data('amt'));
        if(self.level() + 1 > 15) {
            $("#game").fadeOut('slow', function() {
                $("#game-over").html('You Win!');
                $("#game-over").fadeIn('slow');
            });
        } else {
            $("#question-answer-block").fadeOut('fast',
function() {
                $("#question-answer-block").fadeIn('slow');

                self.level(self.level() + 1);
                var bgcss = $("#" + elm).toggleClass('correct');
                $("#answer-one").show();
                $("#answer-two").show();
                $("#answer-three").show();
                $("#answer-four").show();
            });
        }
    });
}

```

```

        self.transitioning = false;
    })
}
}, 1000)

}

// Executes the procedure of guessing incorrectly, losing the
game.
self.wrongAnswer = function(elm) {

    startSound('wrongsound', false);
    $("#" + elm).css('background', 'red')
    setTimeout(function(){
        $("#game").fadeOut('slow', function() {
            $("#game-over").html('Game Over!');
            $("#game-over").fadeIn('slow');
            self.transitioning = false;
        });
    }, 1000)
}

// Gets the money formatted string of the current won amount of
money.
self.formatMoney = function() {
    return self.money().money(2, '.', ',');
}
};

// Executes on page load, bootstrapping
// the start game functionality to trigger a game model
// being created
$(document).ready(function() {
    $.getJSON("Game-Millionaire/questions2.json", function (data) {
        for(var i = 1; i <= data.games.length; i++) {
            $("#problem-set").append('<option value="' + i + '">' + i
+ '</option>');
        }
        $("#pre-start").show();
        $("#start").click(function() {
            var index = $('#problem-set').find(":selected").val() -
1;
            ko.applyBindings(new
MillionaireModel(data.games[index]));
            $("#pre-start").fadeOut('slow', function() {
                startSound('background', true);
                $("#game").fadeIn('slow');
            });
        });
    });
});
});

```

3. HTML5 and CSS Template

Reference:

Anon (n.d.) Leoratic Website Template | Free Website Templates | OS Templates, *Os-templates.com*, [online] Available at: <http://www.os-templates.com/free-website-templates/leoratic> (Accessed 18 April 2016).

Info:

Template Name: Leoratic

Author: OS Templates

Author URI: <http://www.os-templates.com/>

Licence: Free to use under our free template licence terms

Licence URI: <http://www.os-templates.com/template-terms>

4. ACE code editor

Reference:

Anon (2010) Ace - The High Performance Code Editor for the Web, *Ace.c9.io*, [online] Available at: <https://ace.c9.io/#nav=about> (Accessed 18 April 2016).

GitHub: <https://github.com/ajaxorg/ace>