**Developing a mobile CT game**

**Bebras App**

**Padraig Sean Connolly**

**13528563**

Final Year Project – **2016/2017**

B.Sc. Single Honours in

Computer Science/Computer Science and Software Engineering

****

Department of Computer Science

Maynooth University

Maynooth, Co. Kildare

Ireland

A thesis submitted in partial fulfilment of the requirements for the B.Sc.

Single Honours in Computer Science/Computer Science and Software

Engineering.

Supervisor: **Aiden Mooney***.*

Contents

[Declaration i](#_Toc445718342)

[Acknowledgements ii](#_Toc445718343)

[Abstract iii](#_Toc445718344)

[List of Figures iv](#_Toc445718345)

[List of Tables iv](#_Toc445718346)

[**Chapter one: Introduction** 1](#_Toc445718347)

[Summary 1](#_Toc445718348)

[1.1 Topic addressed in this project 1](#_Toc445718349)

[1.2 Motivation 1](#_Toc445718350)

[1.3 Problem statement 1](#_Toc445718351)

[1.4 Approach 1](#_Toc445718352)

[1.5 Metrics 2](#_Toc445718353)

[1.6 Project 2](#_Toc445718354)

[**Chapter two: Technical Background** 3](#_Toc445718355)

[Summary 3](#_Toc445718356)

[2.1 Topic material 3](#_Toc445718357)

[2.2 Technical material 3](#_Toc445718358)

[**Chapter three: The Problem** 4](#_Toc445718359)

[Summary 4](#_Toc445718360)

[3.1 Project UML documentation 4](#_Toc445718361)

[3.2 Problem analysis 4](#_Toc445718362)

[**Chapter four: The Solution** 5](#_Toc445718363)

[Summary 5](#_Toc445718364)

[Depending on your type of project, you may not need to include all of these: 5](#_Toc445718365)

[4.1 Analytical Work 5](#_Toc445718366)

[4.2 Architectural Level 5](#_Toc445718367)

[4.2 High Level 5](#_Toc445718368)

[E.g. Packages, Class Diagrams, etc. 5](#_Toc445718369)

[4.2 Low Level 5](#_Toc445718370)

[E.g. Method specifications, Algorithms, etc. 5](#_Toc445718371)

[4.2 Implementation 5](#_Toc445718372)

[**Chapter five: Evaluation** 6](#_Toc445718373)

[Summary 6](#_Toc445718374)

[5.1 Solution Verification 6](#_Toc445718375)

[E.g. use your equations to verify the correctness of your solution 6](#_Toc445718376)

[5.2 Software Design Verification 6](#_Toc445718377)

[5.3 Software Verification 6](#_Toc445718378)

[5.3.1 Your test approach (i.e. unit testing, sub-system testing, system testing) 6](#_Toc445718379)

[5.3.2 Your tests (e.g. scenarios, test cases, test data, etc.) 6](#_Toc445718380)

[5.3.3 Your test results 6](#_Toc445718381)

[5.3.4 An interpretation of the results 6](#_Toc445718382)

[5.4 Validation/Measurements 6](#_Toc445718383)

[5.4.1 Results 6](#_Toc445718384)

[5.4.2 Explanation of Results 7](#_Toc445718385)

[5.4.3 Analysis of Results 7](#_Toc445718386)

[5.4.4 Comparison with previous solutions (if relevant) 7](#_Toc445718387)

[**Chapter five: Conclusion** 8](#_Toc445718388)

[**Summary** 8](#_Toc445718389)

[**5.1** **Contribution to the state-of-the-art** 8](#_Toc445718390)

[**5.2** **Results discussion** 8](#_Toc445718391)

[**5.3** **Project Approach** 8](#_Toc445718392)

[**5.3** **Future Work** 8](#_Toc445718393)

[**References** 9](#_Toc445718394)

[**Appendices** 10](#_Toc445718395)

[Appendix 1 Schematic of the hardware associated with this project. 11](#_Toc445718396)

[Appendix 2 Code developed for this project. 12](#_Toc445718397)

[Appendix 3 UML Class, Use Case and sequence diagrams for this project. 13](#_Toc445718398)

[Appendix 4 Screen shots of the project implementation 14](#_Toc445718399)

## Declaration

I hereby certify that this material, which I now submit for assessment on the program of study as part of my B.Sc. Single Honours in Computer Science/Computer Science and Software Engineering qualification is *entirely* my own work and has not been taken from the work of others - save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Padraig Connolly Date: 01/02/2017

## Acknowledgements

I would like to thank the CoderDojo students and teachers for allowing me to take the time to test my application on real students of the optimum age group, which is the most valuable group of subjects to evaluate the use of the application. The data observed from this group of individuals gave me an understanding of how useful the application was and also gave me feedback on how well it performed and how it can be improved.

I would like to thank my fellow students in fourth year computer science in Maynooth University for their feedback, cooperation and of course friendship. In addition, I would like to express my gratitude to the staff of the Eolas Building for the use of the facilities.

Nevertheless, I am also grateful to the Mr Aidan Mooney, PhD, Computer Science and Software Engineering Lecturer and my project supervisor for his invaluable advice, feedback and above all else guidance.

I would like to thank my friends for accepting nothing less than excellence from me. Last but not the least, I would like to thank my family: my parents and to my brothers and sister for supporting me spiritually throughout writing this thesis and my life in general.

## Abstract

To develop a gaming application that will help improve the Computational Thinking skills in young children will be available across multiple devices including tablets, phone (iOS, Android) and on PC’s via a web application. The application will also hopefully provide insightful feedback that will alert children into thinking about the areas of logic they may need to improve on and why they may need to improve on it. The background of this project was based on the new Lithuanian competition introduced in 2004 known as Bebras, the competitions purpose was to promote Informatics (or Computer Science, or Computing) and Computational Thinking especially among teachers and pupils of all ages, but also to the public at large by extent. The big challenge of Bebras is to organise easily accessible and highly motivating online challenge in many countries. This challenge will prove much easier for a gaming application that can be played on multiple devices that is available on different marketplaces. Given a problem sheet from my supervisor, I started developing the game problems on a SDK known as App Game Kit, the reason being the choice for this software was due to the fact that it could easily port the project into the required Operating Systems that proved to be a major obstacle. As well as the fact that I had previously used the software before for personal reasons.

## List of Figures

[Figure 3‑1 UML class diagram overview for this project. 4](#_Toc445718606)

## List of Tables

[Table 2‑1 Table of interest: Aspect of your implementation 2](#_Toc445714278)

[Table 2‑2 Data sources used in your implementation 2](#_Toc445714279)

# **Chapter one: Introduction**

## Summary

This chapter will introduce the main topic proposed in this project, the motivation for me undertaking this project and why I undertook the project give my reasons and background, the natural problems that this task will have to overcome and my approach in how I will solve these problems. This chapter will also conclude how I evaluated my performance during this assignment and will list and describe the achievements made during this endeavour.

## 1.1 Topic addressed in this project

In everyday life people are dealt with problems that are solved through the fundamental use of logic. The definition of logic is stated by the Cambridge Dictionary as “a particular way of thinking, especially one that is reasonable and based on good judgment”. Seen as each different real world problem is unique and not all solutions to these problems are the same but the approach to a lot of real dilemmas will require a sense of reasoning and logic. This form of thinking is of course the foundations of many different and important fields today particularly computer science and Mathematics. Then it is of utmost importance to make sure that logic is a part of our education system and is applied to the problems given to today’s students. However, it has been noted of late that students that have come out of the education system cannot apply what they’ve learned from their studies to their field as problems that naturally erupt occur that can be solved easy with a logical way of thinking, particularly for those in the field of computer science and software engineering which is a very practical subject. To address this issue, A European Agency known as Bebras…

## 1.2 Motivation

Why would one care about the problem and the results?

Cite appropriate references in this section. Explain the high-level, abstract problem that your project addresses. Explain what you are trying to achieve in a way that leads naturally into the next section.

## 1.3 Problem statement

Describe the technical problem needed to be solved in your project. Note that most projects solve both a more abstract, high-level problem and a specific, technical problem: your problem statement is the detailed technical problem (your motivation should cover the more abstract high-level problem).

## 1.4 Approach

Summarise how you addressed solving the problem.

Provide an overview of how you analysed the problem, how you designed a solution, and how you evaluated your solution. (e.g. use of models, simulation, prototypes, real-world experiments, cases studies, etc.). What important variables did you control, ignore, or measure in your evaluation.

## 1.5 Metrics

Describe how you are going to evaluate your work.

## 1.6 Project

List, and briefly describe your significant achievements in the project (probably 3-5 of these in a typical project). If you have come up with any contributions

# **Chapter two: Technical Background**

## Summary

The purpose of this chapter is to show your depth and breadth of reading and understanding of the problem domain

## 2.1 Topic material

(Research material, if used, from published journals and conference proceedings; less academic publications, if required by the project, from other sources) – for example, what other work researchers have done already in this area, what results they have produced, what work has been done in related areas, what software already exists to solve this or similar problems, etc.

## 2.2 Technical material

(From any source: including books, websites) – for example, how to write a web server, how to use specific Java features, how to use Ajax, how to use UML to validate your design, etc.

NB: Note that material relating to the motivation or non-technical background should **NOT** go here, but rather in the introduction

Table 2‑1 Table of interest: Aspect of your implementation

|  |  |
| --- | --- |
| **Column description 1** | **Column description 2** |
| A | Text 1 |
| B | Text 2 |
| C | Text 3 |

Table 2‑2 Data sources used in your implementation

|  |  |  |
| --- | --- | --- |
| **Column description 1** | **Column description 2** | **Column description 3** |
| X | 22 | 33 |
| Y | 33 | 456 |
| Z | 17 | 22 |

# **Chapter three: The Problem**

## Summary

The purpose of this chapter is to clearly explain the technical problem and/or identify the user requirements.

## 3.1 Project UML documentation

Provide any model(s) of the problem (e.g. equations, ERD’s, UML Use Cases & Scenarios, Activity Diagrams, etc.)



Figure 3‑1 UML class diagram overview for this project.

## 3.2 Problem analysis

Provide any analysis of the problem, leading to a greater understanding

There should be no decisions made in this chapter

# **Chapter four: The Solution**

## Summary

The purpose of this chapter is to clearly identify, discuss, and justify the decisions you make

## Depending on your type of project, you may not need to include all of these:

## 4.1 Analytical Work

E.g. Equations, etc. that describe your solution

## 4.2 Architectural Level

E.g. Implementation Diagrams

## 4.2 High Level

## E.g. Packages, Class Diagrams, etc.

## 4.2 Low Level

## E.g. Method specifications, Algorithms, etc.

## 4.2 Implementation

Discuss anything interesting here; put full source code in an appendix or attachment

# **Chapter five: Evaluation**

## Summary

Chapter 5 describes……..

## 5.1 Solution Verification

## E.g. use your equations to verify the correctness of your solution

## 5.2 Software Design Verification

How did you show that your design worked properly?

Using a model of your solution. E.g. use UML interaction diagrams to verify each scenario.

## 5.3 Software Verification

How did you demonstrate your software worked properly?

If you have not tested your software, then you cannot rely on your results. Clearly describe:

### 5.3.1 Your test approach (i.e. unit testing, sub-system testing, system testing)

### 5.3.2 Your tests (e.g. scenarios, test cases, test data, etc.)

### 5.3.3 Your test results

### 5.3.4 An interpretation of the results

## 5.4 Validation/Measurements

How did you measure how well your solution solved the problem.

### 5.4.1 Results

### 5.4.2 Explanation of Results

### 5.4.3 Analysis of Results

### 5.4.4 Comparison with previous solutions (if relevant)

**Chapter five: Conclusion**

**Summary**

Chapter 5 identifies and discuss the implications of your work.

**5.1 Contribution to the state-of-the-art**

If you made a contribution to the state-of-the-art, clearly identify it here.

**5.2 Results discussion**

Discuss whether your results are general, potentially generalizable, or specific to a particular case. Identify threats to the validity of your results (e.g. limitations, risks introduced by your approach, etc.)

**5.3 Project Approach**

Discuss your project approach

**5.3 Future Work**

Discuss future work, based on what you have done (and not done)

# **References**

Action Research : A definition . (2015). Retrieved February 25, 2016, from http://valenciacollege.edu/faculty/development/tla/actionResearch/ARP\_softchalk/ARP\_softchalk\_print.html

Hammersley, M. (1993). On the teacher as researcher. In M. Hammersley (Ed.), *Educational Research: Volume One: Current Issues* (pp. 211–231). The Open University.

Jick, T. D. (1979). Mixing Qualitative and Quantitative Methods: Triangulation in Action. *Source: Administrative Science Quarterly Qualitative Methodology*, *24*(4), 602–611. Retrieved from http://www.jstor.org/stable/2392366

Kemmis, S. (1993). Action Research. In M. Hammersley (Ed.), *Educational Research: Volume One: Current Issues* (pp. 175–190). The Open University.

Kolb, D. (1984). *Experiential learning*. New Jersey: Prentice Hall.

McNiff, J., Lomax, P., & Whitehead, J. (2003). *You and Your Action Research Project* (2nd ed.). London & New York: London & New York.

**Appendices**

Include here all extra material, e.g. your source code, project management (optional) including: the task list, Gantt Chart diagrams (or equivalent), discussion of any significant deviations from plan, and how you managed them, discussion of what you would do differently if you repeated the project.

## Appendix 1 Schematic of the hardware associated with this project.

## Appendix 2 Code developed for this project.

## Appendix 3 UML Class, Use Case and sequence diagrams for this project.

|  |
| --- |
|  |
| Appendix 4 Screen shots of the project implementation |
|  |