University of Liverpool

Computer Science with a Year in Industry BSc (Hons)

G403

Year In Industry Dissertation

Author:
N Aishah B M Senin
(200912462)

Project advisor:
Dr. Prudence Wong

March 24, 2016

Abstract

This project is being created for the University of Liverpool under the supervision of Dr. Prudence Wong and Dr. Igor Popatov with the aim of creating an application that serves an educational purpose to the students in the Computer Science department. The intentions of this application is to display the algorithmic paradigms with the use of animations. This program is easily accessible through the use of any up-to-date Internet browsers.

Contents

Chapter 1

Introduction

1.1 Project Description

This project primarily focuses on the animation of different types of commonly used algorithms. The benefit this, is to let users to understand how algorithms work in general. Learning about what algorithms are and how they work is essential for students who are studying computer science, and those who are interested in understanding on designing and building the architecture of an efficient software. Since this project is meant to be educational, the target audience of the software will be students studying computer science, or at least have an interest on how software are made efficient. In this project, the scope will revolve around the animations within the three main algorithmic paradigms, such as the greedy method, divide and conquer, and dynamic programming. Of course, some of the sorting algorithms will be included as well, as these types of algorithms are considered the essence, which means that it needs to be understood in order to understand wholly on the purpose of algorithms.

The primary purpose of this project is to develop a software that displays animations in which shows different types of algorithms within the three main paradigms and sorting works in general. From the program, the users are able to select an algorithm they wish to learn. From here, the user can choose, within a designated range, can either enter their own set of inputs, or generate a random set of inputs for the algorithm to take in. After which, the algorithm will use these inputs and work accordingly by displaying the animation of the algorithm selected on the program.

1.2 Project Aims

The primary objective to this project is simply to make difficult algorithms to be easily understood. To achieve such feat, there would be a need to use animations. These animations act as a visual aids for the students to learn the algorithms from a different perception. As a computer science student myself, I believe that using visual tools such as the animations would certainly enhance the students' learning experience.

In is generally known that algorithms is one of the challenging topics within the computer science field, and is difficult for some students to grasp at times. Making this another aim of the project is to build this product as a tool for assistance, where students use the program to help them understand further about the project.

1.3 Project Objectives

1.4 Project Challenges

1.5 Solutions

1.6 Effectiveness and Success of the Project

Chapter 2

Design

This chapter primarily consists of the plan for the construction of the *AlgoAnimation* program. The design documentation includes the list of requirements that is expected of the program, system flowchart, database designs, code designs, sequence diagrams, and the designs of the animations that currently exists in the program.

2.1 Database Design

2.1.1 Benefits in retrieving relevant data

Initially, I thought that there wasn't a real reason for having a database system connected to this project, mainly because I could've kept the data in the model of the code. However, during the initial part of the implementation phase, I have realised that there was a need to add a database system into the project. Reasons being that, I needed a way to store the details of each algorithm, such as its name, type, form type, (more details in the database design below). I found that doing this way is easier for me to retrieve relevant data about the algorithms displayed on the page by using LINQ queries instead of sifting through the list of algorithms.

2.1.2 Benefits in scalability

I have also found out that there were additional benefits for doing this way as well, such as the scalability of adding new algorithm animations into the existing program. In a certain circumstance where the developer decides to add a new algorithm into the program, s/he would only need to connect to the database, and add the relevant details of the algorithm, and the program will easily take them into an account and add the newly added algorithm into the program. When the developer already done so, s/he would just need to add the actual animation program into the code, possibly add a few additional models on top of the existing ones, and done.

2.1.3 Conceptual Design

2.1.4 Logical Design

2.1.5 Physical Design

2.2 Data Dictionary

2.3 Algorithm Code Design

2.4 Code Structure Design

2.4.1 UML Class Diagram

2.4.2 Sequence Diagram

2.5 Storyboarding

2.5.1 Selecting an algorithm to animate

2.6 Front-End Interface Design

2.6.1 Main page

2.6.2 Form page

2.6.3 Animation page

Chapter 3

Realisation

3.1 Technologies used

3.1.1 The MVC Architecture

3.1.2 Database Connectivity Technology and the use of Models

3.2 Component Implementation

3.2.1 Database Connection Manager

3.2.2 Code generation from SQL to LINQ

Chapter 4

Evaluation

4.1 Criticisms on the animations

4.2 Project Weakness

4.3 Stakeholder Feedback