Project descriptions

A summary of development projects, both in college and private

Knut Lucas Andersen

January 27, 2017

Contents

1 Introduction

This document lists all the development projects I have been a part of, starting from my Bachelors degree until my completed Masters degree in Applied Computer Science. Each section is dedicated to a given programming language, the exceptions are the two last sections, Section 6 and 7. Projects uploaded to GitHub will also contain a link to its given repository, but group projects is not listed. For all projects, there will be a listing for project type (Assignment or Private), IDE, Editor, Tools and Programming language(s) that were used. Development projects from courses will also include the College/University, the Level (Bachelor or Master) and the Course it belongs to.

Repository: https://github.com/klAndersen?tab=repositories

2 Visual Studio: C#

2.1 Real-estate registration

- Level: Bachelor
- Project type: Assignment
- College/University: Buskerud University College
- Course: INF313 Management of Databases
- IDE, Editor, Tools:
- Oracle-Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/Databases/INF%20315%20-%20Oracle/Oblig1
- C#-Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/C%23/Oblig%20Inf315

This assignment was based on using Oracle to create object-types based on a given UML-diagram. It should store properties, bids (and these could not be withdrawn), buyers, real estate agents and the firm the agent belonged to. When calling the to_string, all property information should be returned. Bids should only be updated if they were bigger than the last, the date of acceptance for bid had expired, or if the date of bid acceptance was later than the first date given. One should be able to retrieve bids based on urgency (e.g. if it expired that same day). One should be able to retrieve all bids from a given buyer, but only for those whose deadline had not expired. There was also always only one seller for each property.

A slighty altered version of the the one listed above. It was divided into three parts, storing data in memory (RAM), db4o and Prevayler. We were given a UML-diagram of the architecture, which was using the three-layer-model. We were also given a pre-made GUI (Windows Forms), and needed only to create the object classes and Control class. The Control class should mainly be able to add, delete, update, retrieve and search for data provided by the user. This was developed using Visual Studio as IDE and C# as programming language.

2.2 Bachelor Thesis: CleanMyFolder

- Level: Bachelor
- College/University: Buskerud University College
- Course: INF350 Bachelor Thesis
- Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/C%23/Bachelor%20thesis

Tools used: Visual Studio 2010, C#. UMLet to create the UML diagram. The HTML files for the help was written using PSPad, and compiled with Microsoft HTML Help Workshop. Windows OS only.

The thesis was based on the following problem: ”I want more space on the harddisk - which files do I delete?" The main focus in this thesis was the development of a file management system that could scan the computer for folder and files at a given location, and search for duplicates (including zip-files).

Due to complexity with the ZIP-library in C#, and time restraints the zip-file and duplicate search was excluded (the idea was to look for folders/files that were exact duplicates, and replace these with a shortcut to the folder/file of the users selection). The finished system has the ability to search the whole hard drive or a given path, which then lists the folder(s) and file(s) based on set criterias (file type, size, date).

The folder(s) are listed in a TreeView, and the files in a DataGridview. By right clicking a folder, the user could opt to search the given folder using the same scan criterias, or open the location in Windows Explorer. The user also has the ability to move or delete a given folder or a selection of files.

2.3 ITE1605 Computer Graphics

- Level: Bachelor
- College/University: Narvik University College
- Programming language(s): C#/XNA
- Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/XNA/Datamaskingrafikk

Course using Visual Studio C# and XNA to create games and simulations. Assignments consisted of drawing a 3D-cube and a spinning 3D-ball using lines, matrices and the XYZ. There was also a task to create a representation of the planetary system (I didn't manage to properly complete that assignment), and also the use of particle effects and lighting/shaders. In the graded programming assignment I delivered a game based on the 3D-models developed in the 3D-modeling course.

2.3.1 Graded Assignment: Car game

The game was a simplistic car game, where you drove a car around in a "stadium", music could be turned on/off, and you got points for hitting the flags that was placed in static locations inside the stadium.

Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/XNA/Datamaskingrafikk/Karaktergivende%20oppgave

2.4 Model Train registration

Project type: Private
Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/C%23/ModellTog This was a program I developed for one of my friends. He had a thing for model trains, and wanted a program that gave him the ability to register trains, wagons, accessories and create a wish list.

Trains and wagons had almost the same specifications, so for these the same GUI controllers were used. The accessories had other specifications, as these could be railroads, trees, figurines, houses, etc.

For the wish list, he could register the model number, name, sales location and price. The program was originally developed in Java, but he had issues with running the program. It was therefore later re-written using Visual Studio and C#.

2.5 Multipurpose editor

- Project type: Private
- IDE, Editor, Tools:
- Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/C%23/MultiPurpose

At the time, I often experienced MySQL Workbench being slow at start-up, and sometimes I wanted to just a quick peek at the tables or data in the database. Creating Readme's for programs and systems developed could also be slightly time-consuming, so I created this simple program I named "MultiPurpose".

The program was written in C#, and used ODBC to connect to MySQL. This program had three functionalities:

1. Text Generator: When creating experience level requirements for games, setting these values can be quite time-consuming. The same can be said if you are testing a lot of "random" links with a set incremental numeric value. Here one can enter a start/end text, and append numbers (min, max, step). If one's testing links, you can select which browser to open the link in (converted to links if text contained 'http:/').
2. ReadMe generator which saved the text you entered to a text file. Options included overwriting existing file, adding asterics and version number into the text.
3. 3. Connecting to, and displaying information from a MySQL database, after the user had entered a username, password and host/port. Gives the ability to select a database, select a table, and run some queries against the database.

3 Visual Studio: C++

3.1 ITE1546 Programming C++

- Level: Bachelor
- Project type: Assignments
- College/University: Narvik University College
- Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/C%2B%2B/ITE1546%20Programmering%20i%20C%2B%2B

This was a pure introductory course, as it was targeted at the students studying electronics. Therefore there was no GUI, only command line in Visual Studio, using native C++. There were three mandatory assignments, and one graded.

The first assignment was mainly focusing on creating a menu-program that could execute some equations, and convert numbers to their textual value. The second was based on creating a cinema ticket system, which should only register one performance, sales data and vacant sales, etc, which was stored in a text file. If a new performance was registered, the old was deleted. The third was based on Bitmap pixel colour manipulation, where one should be able to alter the colour of all pixels or grey-scaling them all.

3.1.1 Graded assignment: recipeProgram

Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/C%2B%2B/ITE1546%20Programmering%20i%20C%2B%2B/Karaktergivende
The graded assignment consisted of creating a system that could store recipes, calculate a recipes nutrition data, list the data for a given nutrition, list recipes containing more than X nutritions, etc. The nutrition data was extracted from an Excel-file provided by the teacher.

3.2 Machine Learning library

- Level: Master
- Project type: Developed for re-usability in assignments
- College/University: Gjøvik University College
- Course: IMT4612 Machine Learning and Pattern Recognition 1
- Repository: https://github.com/klAndersen/Machine-Learning/tree/master/MachineLearning

This was written in Visual Studio 2013 using Managed C++, to use in the Machine Learning course I had. It is not a complete project, as it is mostly built on the algorithms taught in the course, and those that were asked for during the mandatory assignments. Aside from algorithms, it contains a Conversion class (string to numeric, naitive to Managed C++, vica versa, etc), a Matrix class, a class for statistics and exceptions.

3.3 IMT4641 Computational Forensics

- Level: Master
- Project type: Assignment
- College/University: Gjøvik University College
- IDE, Editor, Tools: Visual Studio, SQLite
- Repository: `https://github.com/klAndersen/IMT4641-Computational-Forensics`

This course was development only, where we were tasked with coming up with our own idea to develop a tool or system that was relevant to the track and courses we had. My project focused on creating a re-active tool (used after an incident has occurred) that analysed Android databases (SQLite). It was developed using Visual Studio C++, using Windows Forms as GUI.

I re-used parts of the graphical design that I used in my Bachelor thesis, since some qualities were similar. The tool lists the folders, SQLite database(s) and tables in a TreeView. In the Treeview, the folder is the parent, and the database(s) are the children. The table(s) in turn, are the child(ren) of the selected database. When a table was selected, the data was listed in a DatagridView. One could also search for a given entry in the listed treeview, where matches were highlighted in yellow.

4 Java and Android

4.1 Animal Registration

- Level: Bachelor
- Project type: Assignment
- College/University: Buskerud University College
- Course: INF240 and INF244 Object Oriented Programming
- IDE, Editor, Tools: Eclipse
- Programming language(s): Java
- Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/Java/Oblig2_Inf240_KnutLucasAndersen
- Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/Java/Oblig_Inf244_KnutLucasAndersen

In the two Java courses in my Bachelor, we had the same assignment, but with different requirements. The main focus was to register two different animals (hares and lynx). Common for both was the registration of their gender (male/female), length (Double), weight (Double), time of capture (day, month, year) and location (String). The ID was to be incremented, and start with the initial for the given animal (e.g. L1, L2, . . . , H1, H2, . . .). For the hares, colour (String) and type (Char) was to be registered. For the lynx, the length of their ears (Double).

If an animal was later re-captured, their data should be updated instead of re-registered. The program had the possibility to search for animals based on ID, show amounts of re-captures (based on the entered year), show amount of different hare captures (based on type), and an unsorted report of all first-time captures.

The first course was introductory level, so input/output was shown in command line, and data was stored in a text file. In the second course, the user was presented with a GUI that we had to code ourselves (not use a Designer), and the data was stored in a MySQL Database (using JDBC as database driver). Singleton was used to ensure that there was only one object maintaing the database connection. The main menu used JFrame, and the child windows used JDialog.

4.2 ITE1621 Applications for mobile and web

- Level: Bachelor
- Project type: Assignment
- College/University: Narvik University College
- IDE, Editor, Tools: Eclipse
- Programming language(s): Java, Android, XML

- Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/Android/ITE1621%20Applikasjoner%20for%20mobil%20og%20web

This was an introductory course to programming apps for Android. It consisted of four mandatory assignments and a graded assignment. The Android version that was targeted was Android 2.3 - 4.1, using Eclipse as IDE and Java as programming language.

The assignments consisted of creating a conversion app for different measure systems (distance, volume, mass, temperature and time), passing data from one application to another (broadcasting, event logging, settings and SQLite) and creating a weather notification app. The weather app downloaded XML data from a weather page in set intervals based on settings set by user. The data was then parsed and displayed to user, and the user could set a temperature limit (min/max). If the temperature was out of the set range, a notification should be shown to the user.

4.2.1 TracknHide

This project was part of the course grade, and consisted of developing an application that could show the different users location and route on a map. The project had two parts, an Android application and a Java web server.

The Android application used Google Cloud Messaging (GCM) to send and receive data, and Google Maps (API v1) to present a map. By using MapOverlay, users could also show the adress when clicking a location on the map, and switch between Street – and SateliteView. On the map, the route and current position of logged in users with shared position was shown. Users could also choose to store their own or others route, which could later then be re-drawn on the map. The position data was retrieved by using the devices GPS, but the amount of position data sent was set by the user (amount of time passed or distance moved).

The Java Web server was created using HTML, Java Servlets and JSP, running on Apache TomCat v7.0. It stored the user related information in a MySQL Database, and kept track over all the currently logged in users. The web pages was mainly targeted at an administrator, to give them the ability to view connected users, and disconnect if needed. The server was also the connection point for the users to share/receive position data and notifications via GCM. Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/Android/ITE1621%20Applikasjoner%20for%20mobil%20og%20web/Karaktergivende%20oppgave

4.3 TrackMyTeacher

- Level: Master
- Project type: Assignment
- College/University: NTNU
- Course: IMT5401 Advanced Course in Mobile Technology

- IDE, Editor, Tools: Android Studio, MySQL Workbench
- Programming language(s): Android, XML
- Repository: https://github.com/klAndersen/IMT5401-Mobile-Research

In this course I tried to develop an indoor tracking prototype app for Android. The purpose was to be able to locate teachers and professors on campus, by measuring the WiFi-signals at their current location (if they had enabled/allowed tracking of their current location).

The server side of the application was developed using PHP (because the college server only ran PHP), which mainly handled data transfer between the Android app and the MySQL database. When data from the database was needed, the server retrieved the data, added it to an array as JSON and printed it as HTML. This HTML was then converted to a JSONArray on the mobile device (as I stated in my report, it would probably have been much easier if one could have used Java Servlets, since one could overwrite the HTTP GET/POST methods).

To get test data, I went to different buildings, rooms and floors spread out on campus (20 rooms, signals measured in all 4 corners plus the middle of the room). What I found when looking at the gathered data was that the measurement for the WiFi-signal was set too low (value=5). Therefore this prototype was not finished, as I did not have the time to re-measure all the rooms to get more accurate signal data.

4.4 BeregnSnitt

- Project type: Private
- IDE, Editor, Tools: Eclipse
- Programming language(s): Java
- Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/Java/beregnSnitt

When nearing the end of my Bachelor degree, I wanted to know what my average grade was, to see what colleges and universities I could apply to in regards to taking a Master degree. This lead to the development of "BeregnSnitt". BeregnSnitt was a small, private Java project, which calculates an average score based on the grades the user has.

The user is presented with a GUI (Jframe and Jdialog), where s/he can enter the amount of A to E's achieved, the course points and the total points achieved (e.g. Bachelor = 180). I also added the possibility to add additional course grades, in case some courses varied (e.g. if the standard was 10, but one course had 3 points).

A Norwegian Java application I developed that calculates the average score of the grades achieved within a college. The main reason I created it was to see what colleges and universities I could apply to based on my current grade average. The user can enter the amount of A to E's achieved, the course points and the total points achieved (e.g. Bachelor = 180). One could also add additional grades (e.g. if some courses had different scoring). Then the user had to enter the grade and the course points.

4.5 Auto-reply for Android

- Project type: Private
- IDE, Editor, Tools: Eclipse, Android Studio
- Programming language(s): Java, Android, XML

This was a program I started on after learning Android. The reason I started working on it was because one often gets text messages or calls when you cannot reply or answer. Sometimes, people continue to text or call, and I therefore started working on what I call an Auto-reply app. The auto-reply is sent out as a text message (SM)S, which could either be random or a set topic.

The auto-reply could be sent to only to those marked for it, or apply to all in the phones contact list. The auto-reply was restricted by a delay, adjustable through the settings. When a call or SMS came in, if the person calling/texting was on the Auto-reply list, a timer would start. When the timer ran out, a check would occur to see if the SMS/lost call was unread. If so, a SMS would be sent out.

The SMS that were sent from the Auto-reply are random, but that was mostly because I created it for myself. I planned to later add some Machine Learning logic to it, so that it could reply somewhat intelligently to the SMS, instead of a random text.

The development started with using Java and Eclipse, but was later changed to using Android Studio as IDE.

5 Python

5.1 IMT4112 Global Software Development

- Level: Master
- Project type: Group, Assignment
- College/University: NTNU
- IDE, Editor, Tools: PyCharm, Arch Linux (OS)
- Programming language(s): Python

Group project course where the goal was to teach us how it was to work in a global team not having all members around, and also working on expanding an existing system. The task my group was given was to develop a multi-choice questionnaire plugin for the Learning Management System (LMS) Open edx.

Aside from creating the pages and functionality for creating, editing and displaying a questionnaire, we were given three tasks (the third was never started, and thus is excluded from this summary). In this questionnaire, students should be able to select their confidence level in regards to their submitted answers. Depending on the confidence level selected (0 = low, 1 = normal, 2 = high), their total score could be increased (or decreased if wrong answer was selected).

The second task was peer-review, where the students could select whether or not they thought the rest of the class found this to be a difficult question. Bonus points were then given to the students who selected the one that got the most selections.

Aside from handling administrative tasks (I was the Scrum Master in this project), the parts I worked on was the creation of answer alternatives, the answers submitted by the students and and grade for the submitted questionnaires (with confidence level). In the second task, I worked on the part related to difficulty level and updating of the students score. During this project, I used Arch Linux as OS, PyCharm as IDE and Python as programming language.

5.2 IMT5251 Advanced Project Work

- Level: Master
- College/University: NTNU
- IDE, Editor, Tools: PyCharm, Arch Linux (OS)
- Programming language(s): Python
- Repository: https://github.com/klAndersen/IMT5251_AdvProjWork

Preliminary work for my Master thesis, which was mostly focused on creating a simplistic prototype. The topic for my Master thesis was to develop a Chat Agent that could answer students programming questions, and also help them learn to be better at asking good questions. The Chat Agent was to be implemented as a module (an XBlock) in the Learning Management System (LMS)

Open edx. Python was used, as this is the programming language which the XBlocks are written in, using PyCharm as editor.

During this development, a simple chat interface was created which could connect to StackOverflow.com. One could ask questions, but it only returned the first question/answer it found on Stack Overflow. The prototype was also presented to two students that gave feedback and suggestions for features. The plan for further development was to implement AI in my Master thesis using a hybrid algorithm of Hidden-Markov-Model and Bayes Net.

5.3 Master thesis: Predicting coding question quality

- Level: Master
- College/University: NTNU
- Course: IMT4904 Master Thesis
- IDE, Editor, Tools: PyCharm, Arch Linux (OS)
- Programming language(s): Python 2.7, Python 3.4 (submitted version)
- Repository: https://github.com/klAndersen/IMT4904_MasterThesis_Code

During the presentation of the thesis topic, I was told that my topic was not just too large for a Master thesis, but even too large for a Ph.d. thesis. It therefore needed to be reduced drastically, which ended up with the topic being an attempt to predict question quality at Stack Overflow by looking at existing questions.

In my thesis, I used Arch Linux as an operative system since it was easier to develop for Python. I used MySQL as a database, which contained all the data from Stack Overflow (downloaded from StackExchange Archive). Data was extracted from the database into Python using a library called pandas, which also had the ability to export the data to a CSV file.

The questions were originally in HTML format, so they were therefore cleansed using Beautiful Soup 4 (bs4). To get the vocabulary over all the words used in all the selected questions, CountVectorizer from the machine learning library sklearn was used. Features were extracted and replaced with feature detectors, and these were then used to create a model for predicting the question quality.

6 Web development

6.1 Alumni Community

- Level: Bachelor
- Project type: Assignment
- College/University: Buskerud University College
- Course:
- IDE, Editor, Tools:
- Programming language(s):
- Repository: `https://github.com/klAndersen/Bachelor-development-projects/tree/master/Web%20Development/Inf%20268%20Utviklingsprosjekt`

This was a course that focused mainly on development for Web. In this course, everything was developed from scratch, without use of external frameworks or libraries. The goal was to develop an alumni community web page for the college. A requirement was that the delivered solution should include HTML, CSS3, JavaScript and PHP. For storing data, MySQL was selected as database. To have a local server running, Apache XAMPP was used. It was also a requirement that the pages should look as identical as possible regardless of the web browser that was used (I compared Chrome, Internet Explorer, Opera and Firefox).

The course was split into two semesters, and the development project into five parts:

1. Web page design/layout and policy
2. Access Control
3. Functionality and database model
4. Posts and user administration
5. Completion

The users were divided into the following user groups; registered, moderator and administrator. Adminstrators was able to appoint new moderators and edit users, with the same rights as moderators.

All registered users should be able to...

- ...register, login, update their profile and be able to change/request a new password (if forgotten).
- ...upload a profile picture, or select one of the existing ones that had previously been uploaded to the alumni page.
- ...join or leave networks created within the alumni community. Users should be able to message each other, and also see a notification when new messages has arrived.

- . . . see a list of the other registered users.

Moderators should be able to. . .

- . . . edit users profile (e.g. if inappropriate data was entered on their profile).
- . . . notify users they had breached the guidelines or block out/quarantine users (users should not be deleted).
- . . . create/edit/delete existing topic fields (e.g. events, area of expertise, etc).

In my project I mostly used PHP and HTML. Every page was created as a .php, as this allowed me to update the page layout in just one file, instead of having to update all the pages. The layout was then included at the location where it was to be used. It also simplified the use of function calls and checks to see if a user was logged in, and had the proper rights to view a given page. Database access was also handled via php, and JavaScript was not largely used (mostly to verify user input).

The following is a short summary of what the delivered version included. For new users, they had a registration form, that gave feedback both with JavaScript and PHP if an error occurred. If the user continued, even though one or more fields contained errors, the given field(s) would have a text marked in red next to it, explaining what was wrong. Users also got their own profile, which allowed them to update their information, alongside changing their password (users that weren't logged in could request a new password if they forgot it). Moderators also had the ability to edit user profiles, e.g. if any added something in-appropriate.

Users could also send messages to each other, and to ensure that the users would see if they had received a new message, the link to the messages in the navigation bar was updated with a number, indicating how many new, unread messages they had received. Users could message each other via their profile, or by searching for them from the member list (which listed all, or thosematching the search critera; name or e-mail). Users who broke the page policy could be put in quarantine. This rejected the users attempt to login, and giving them a message that they were quarantined. An information message could also be set, that would explain to the given user why they were quarantined.

Moderators could register campus, area of expertise and networks (e.g. degree, courses, etc). Moderators could also create events. All of the aforementioned could be updated or deleted. For events, only those active were displayed (but administrators and moderators could see all on the creation page). Events that were out of date was not showed, and this was also true for future events (based on start date for the given event).

6.2 Flea Market

- Level: Bachelor
- Project type: Assignment

- College/University: Buskerud University College
- Course:
- IDE, Editor, Tools:
- Programming language(s):
- Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/Web%20Development/Inf%20329%20XML%20-%20Oblig.%20oppg

A school band arranged yearly a flea market, to increase their income. This was a process lasting for months, starting with organization of the sales, to finally selling the various sales items. The flea market consists of several departements, several people are involved and also a lot of money. Income overview and control is a necessity to be able to plan next years flea market. Up until now, they used Excel sheets to keep an economical overview (participants sorted by departements). Every participant gets a bag of change, and can deliver income several times throughout the day. The participants on a Saturday is not necessarily the same as those on a Sunday (sales were only on Saturdays and Sundays). A daily overview over departement sales is registered, and a control is done to check the money delivered against the registered income.

The goal was to re-write this web-application to use XML as data storage, as the currently database in use (Access) gave compability issues depending on the version used. The assignment contained attachments showing screenshots of the Web-page, the Excel file and how the Access database looked. A requirement was that the file containing the database data should be in XML, and that this file should be controlled by using XSD (since the XML file could be altered manually). It was also a requirement to create at least four different web-pages to show how they could use the created XML-files to present and store new data.

6.3 IMT4003 Applied Computer Science Project

- Level: Master
- Project type: Group, Assignment
- College/University: NTNU
- IDE, Editor, Tools:
- Programming language(s):

In this course, our group worked on developing an A/B Test system. The following is a short project description, taken from our group report:

> ”The product is an Internet application, where the goal is to track user interaction. The developer/administrator is able to see the user interaction real time, and the interaction will also be stored in a database, to be shown for the developer/administrator later. This functionality is relevant when a person or a company wants to know the best possible user interface for a website. With this Internet

> application, you can present an interface for a user group, and then you can compare more interfaces with each other, to decide which one the user interacts with most efficient."

Various frameworks were used in this project, but I those I was directly involved with was the Java Servlets and MySQL. In the development part, I had the responsbility for the database model, the database and the database operations. Since we used Model-View-Controller (MVC), some of the programming I did was related to the interfaces and connections between the GUI, the Controller and the Database.

6.4 IMT4004 Integration Project

- Level: Master
- Project type: Assignment
- College/University: NTNU
- IDE, Editor, Tools:
- Programming language(s):
- Repository: https://github.com/klAndersen/IMT4004-Integration-Project

The original plan for this project was to create a 3D-simulation game that incorporated Darwins theory of evolution. The goal was to try helping students taking Machine Learning courses to learn an algorithm called "Genetic Algorithms". However, due to time restraints and complexity, this was changed and reduced to an unfinished web-page version (aimed at teaching Genetic Programming).

This version consisted of pages for students and teachers, where teachers could create new assignments, check student participation and adjust student scores based on their answers/progress. The students had access to creating and editing their profile, see game progress and score, and they could play a given game an unlimited number of times (but attempts and answers were logged in the database). After submission, the given game would no longer be playable.

The only game-element that was implemented was a LISP (reverse polish notation) game. The student was presented with an equation (e.g. 1 + (2 * 3) + (10 *10) + 30) and asked to write the LISP version of this. In addition, the student could see their answer represented as a tree-diagram (achieved by converting the input to a JSON-object, and passing it to the d3.js library).

The project was developed by using HTML5 and PHP, and included a data-visualization library called d3.js. D3.js was only used for creating the LISP tree, which was optional when the student were checking their answer. The data was stored in a MySQL database.

7 Others

Suggestions for sections:

- databases
- udk/unreal
- INF340 Information Systems in Business
- ITE1607 Game Design
- IMT4006 Intro to research on Web technologies
- IMT4032 Usability and Human Factors in Interaction Design
- IMT4072 Cross-media color reproduction
- IMT4122 Software Security Trends
- IMT4612 Machine Learning and Pattern Recognition 1: assignments
- anything else. . . ?

7.1 Teacher-Student questionnaire system

- Level: Bachelor
- Project type: Group, Assignment
- College/University: Buskerud University College
- Course: INF116 Object Oriented Programming
- IDE, Editor, Tools: Visual Studio, MySQL Workbench
- Programming language(s): Visual Studio/.NET Visual Basic (VB)

A two-part group project, where the first part was the teacher system and the second the system for the students. I was part of the group working with the teacher system (we were 3 people working on this), where the system should be able to create new questions for a given course, and set parameters for the evaluation. The teacher would need to get a statistical overview of students answers, and also be able to export/store these to a file. The tools used were Visual Studio IDE, Visual Basic as programming language (course requirement), and MySQL as database. The GUI was Windows Forms.

7.2 INF165 System Developing Project

- Level: Bachelor
- Project type: Group, Assignment
- College/University: Buskerud University College
- IDE, Editor, Tools: Microsoft Access, MySQL Workbench

This was a group project in a system development course. The task was to develop a system for hotel management, where the user should be able to check-in/out guests, see available rooms, room expenses and retrieve economical reports. The system was created by using Microsoft Access and MySQL as database. Microsoft Visio was used to create DataFlow Diagrams.

7.3 INF330 Flash Programming

- Level: Bachelor
- Project type: Group, Assignment
- College/University: Buskerud University College
- IDE, Editor, Tools: Adobe Flash CS 4, FlashDevelop
- Programming language(s): Flash, ActionScript 3.0

7.3.1 Space Invaders

A scroll-game like space invaders, where the player controls a space-ship. This was developed in Flash, using ActionScript 3 as scripting language. The topic for the development was chosen by the group. A short scenario context is that the player is a farmer kidnapped by aliens, and have to save the Earth (and his harvest).

I handled the design and development of the enemies, game settings and playing sound/music. The game contained up to six levels, where the player could select which to play (but level 2-6 had to be opened first by completing previous level. The player got points for destroying enemies (asteroids and aliens). After two minutes of play, the boss appeared. The game also contained a "Top 10" list displaying the scores for the players.

7.3.2 Tower of Knowlegde

Tower defence game meets quiz where the goal is to attack only the wrong answers. The topic was selected by the group. The player can put out none or more towers, and the player had to let the "correct" answer only pass.

Answers was shot by clicking the answer, presuming it was within the given towers range. Points were given for attacking enemies, and reduced for each answer passing the goal. The player was not told if the answer was correct until the game had ended, and when the game was over, the player was added to the Top 10 list presuming their score was high enough. My responsibilities in this project was the development of the main menu, reading (questions, answers, Top 10) and writing (Top 10) to file, sound/music.

The game was developed using Flash and ActionScript. The IDE was Adobe Flash CS 4 and FlashDevelop (actionscript). In addition, I also created a small program for the creation of questions and answers (since the variables was read from file using URLVariables), which was written in Visual Studio, C#.

This is how the file content looked like: *qst=question1?,question2?&alt=alt1,alt2&ans=ans1,ans2*

7.4 ITE1606 3D-Modelling

- Level: Bachelor
- Project type: Assignment
- College/University: Narvik University College
- IDE, Editor, Tools: 3DS Max
- Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/3D%20Modeling/ITE1606%203Dmodellering

This course was based on learning how to create 3D-models and how to work with a 3D-modeling tool, namely 3DS MAX. This was a purely introductory course, so I cannot claim to have any expertise after this course, other than a general understanding.

In the first two assignments, we were asked to create a sketch of a car, and create textures based on clothing, carpets, etc. The next four assignments were based on creating various models (a Lego block with our name on it using the texture we created), a car, a stadium for the car to drive on and a character representation of ourselves).

The last two assignments covered Skeletal movement (making the character able to move, and give it a short animation) and lighting (we had to simulate "evening sun" without using plug-ins or "daylight system").

I also had a course in Game Design, where we used UDK and Unreal to model a game world (the code was written using Unreal Script).