Project descriptions

A summary of development projects, both in college and private

Knut Lucas Andersen

January 28, 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 INF313 Management of Databases

- Level: Bachelor
- Project type: Assignment
- College/University: Buskerud University College
- Course: INF313 Management of Databases
- IDE, Editor, Tools: Visual Studio, Oracle SQL Developer, db4o[1], Prevayler (Bamboo.Prevalence)[2].
- Programming language(s): Oracle (first Assignment), Visual Studio/.NET C# (second Assignment)
- 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

In this course, we were given two mandatory assignments that were graded. Both assignments was based on solving the same task, but using different development tools. The assigment contained an UML-model to describe the expected architecture.

2.1.1 Oracle

A real estate agent wants an Oracle database for registering sales objects that contains sellers, buyers and bids. For each buyer/property, only one bid is stored, and the buyer is bound by the bid until the deadline for acceptance expires. Therefore, bids cannot be withdrawn. If the buyer wants to register a new bid (where the buyer already has an existing bid), the bid would be accepted only if:

- the acceptance deadline on the previous bid expired
- the new bid is higher (greater amount) regardless of the acceptance deadline[3]
- the new bid has the same amount, but later acceptance date

It is possible to sign up as stakeholder without submitting bids, but everyone who bid will automatically be registered as well as stakeholder (if they are not already there).

For a given buyer, it should be possible to get all current bids, ie. bids where the deadline has not yet expired. When checking for bids that expires the same

[1] Open source object database for Java and .NET developers.
[2] http://bbooprevalence.sourceforge.net/
[3] The acceptance deadline specified only by date - bid comes out throughout the day (until 24:00).

day, these should be returned with a the string "Urgent". In all other cases, it should return "Not urgent" - even if the time limit has expired. It should be possible to return all current bids (deadline has not expired) from a given buyer, which lists the propery information. There is always only one vendor for each property. If the property is co-owned, the registered name should contain the person representing the owner. When calling the to_string() function, all information stored in the given object should be returned.

2.1.2 Visual Studio/.NET C#

A real estate agent needs an object-oriented application for registration of properties and bids on these properties. An UML-model was given representing the architecture, along with the client side of the application (using Windows Forms as GUI). An assumption we could make was that all objects would fit in RAM simultaneously, and we were tasked to create two different version. Both versions should ensure persistence by using two different libraries, the first using db4o, the other Prevayler (Bamboo.Prevalence).

The architecture for the system was a three-layer model. Communcation went from the Menu (Windows Form; GUI) to a Controller class[4]. In the Controller class, data was checked/verified before being saved to file. If there were any errors, error messages were to be sent back to the user.

The specific entity classes was persistent, but not the control - and limit classes. Attributes marked with ID by users were to be identified for unique for each object that were stored.

There were also two notable differences that had to be taken into account with db40 and Prevayler. When persistence was secured with db40, the SortedList containing the properties was unnecessary and therefore not included. When using search (find-methods), the search had to be made directly in the object database. With Prevayler the search had to happen in the real estate list since it does not have an object database.

2.2 Bachelor Thesis: CleanMyFolder

- Level: Bachelor
- College/University: Buskerud University College
- Course: INF350 Bachelor Thesis
- IDE, Editor, Tools: Visual Studio/.NET C# 2010, UMLet[5] (to create UML diagram), PSPad (creating HTML help files), Microsoft HTML Help Workshop (for compiling help files)
- Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/C%23/Bachelor%20thesis

For selecting a Bachelor thesis, we had two options. The first was to do research and conduct a scientific study. The second was to develop a system based on a

[4] The Controller class mainly handled adding, deleting, updating, retrieving and searching for data provided by the user.

[5] http://www.umlet.com/

given problem/situation (the option I selected). My thesis was based on the following problem: "I want more space on the harddisk - which files do I delete?"

The delivered system was a file management system which could scan the computer for files and folders at a given location, and then list them in separate views. It was also planned to include the possibility to scan for identical duplicates and check for duplicates based in zip-files, but this was not included due to time-restraints.

A user may not be interested in small files, or perhaps files that were created as early as last week. Therefore the menu for initializing a scan includes several filtering options (include subfolders, file type, size and date). When a scan was completed, the user would be presented with the results, where the folders were listed in a TreeView and files in a DataGridView (presuming there were any results). The structure in the TreeView listed the parent folder (path selected) as the root, where every subfolder (and their subfolders) were listed as a child of that parent folder. When selecting a given folder, all the files in that folder was listed in the DataGridView (file name and - type, size, date created/modified/accessed).

In addition to getting a listing of folder(s) and file(s), the user could open, move or delete the selected folder/file(s). For deletion, there were two options; move to recycle bin, or delete permanently. However, if a filtered scan was ran, the folder deletion options was disabled. This was done to avoid the deletion of a non-empty folder (e.g. only textfiles were listed, but it also contained PDF-documents).

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

The goal of this course was to get an understanding on how computer graphics worked both in simulations and game development. There were a total of five programming assignments, were the last one was part of the course grade. The first two assignments were based on understanding the use of the XYZ-system, by using matrices and lines to draw a 3D-Cube and a spinning 3D-ball. The fourth assignment included use of particle effects and lighting/shaders.

In the third assignment, the task was to create a simulation of our solar system (this I did not manage to complete properly). The sun should be in the middle (and be the largest object), with the planets rotating around the sun (in addition to its own axis). The planets should also have moons with them, but the sizes and distances could be approximations (but one should attempt to make this as realistic as possible).

2.3.1 Graded Assignment: Car game

The graded assignment was a topic of our own choosing, but it required that we implemented at least one element of all the topics taught in this course. I chose to develop a car game, where I used the 3D-models I had developed in the 3D-modelling course (described in Section 7.4). This was a very simplistic game, where all the player could do was drive around in a "stadium" to gather flags (which gave the player points; static position). The flags gave the player points, but the score was not stored, and was reset when starting a new game. The player could not drive out of the stadium, nor could the player destroy or crash the car.

The game included music and sounds (e.g. screeching sound when breaking), which could be turned on/off (three different background musics were available). The camera followed the car, but the player could swap with camera being behind the car or in front (front-view look). The particle effects were present as exhaust from the car, and by using gears, the player could adjust the speed (1-4, and reverse).

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

2.4 Model Train registration

Project type: Private
Programming language(s): Java[6], Visual Studio/.NET C#.

One of my friends interests was model trains. As he started to get a larger collection, he wanted to get a better overview of what he owned, and also the possibility to create a simple wishlist to give to friends and family.

He wanted a separation between whether the model he registered was a train, a wagon or an accessory. Trains and wagons had nearly the same specifications, but accessories had a wider group, as these could be railroads, trees, figurines, houses, etc. Since he could own more than one wagon, instead of re-registering an existing wagon, he wanted to just update the amount of wagons owned. When creating a wishlist, he wanted to be able to save it to a textfile for printing. The wishlist should contain the models number/ID, name, sales location and price.

The finalised version ensured persistence by using Prevayler (Bamboo.Prevalence), which takes snapshots of the current state of the program, its objects and data. With Prevayler, he had no need to install any excess programs, and if there was a lot of useless data stored, it could easily be removed by deleting the existing file(s). In addition to desired requirements, the program also included the ability to search for models (based on number/ID, name or type). To ensure that models could be easily updated when listing registered models, these were presented in an editable DataGridView.

Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/C%23/ModellTog

[6] The project was originally developed using Java, but he was unable to make it run. It was therefore re-written using C#.

2.5 Multipurpose editor

- Project type: Private
- IDE, Editor, Tools: Visual Studio/.NET C#, ODBC (for MySQL)
- Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/C%23/MultiPurpose

At the time (2011-12), 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 could also be slightly time-consuming, so I created this simple program I named "MultiPurpose". 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 you are testing links, you can select which browser to open the link in (converted to links if the text contained 'http:/').
2. ReadMe Generator: Saved the text you entered to a text file. Options include overwriting existing file, adding asterics, and adding version number into the text.
3. MySQL Display: Establish a connection to a MySQL user account. After entering valid username, password and host/port, you could display the information available for the given users database(s). You could select a database, select a table, and run some queries against the selected 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.

It should be written a C ++ console application that solves the following tasks / problems / challenges: — 1. Calculate bullet volume. 2. Resolves an other quadratic equation. 3. Distance between geographical points. 4. Annuity. 5. Numbers text (not required, but is recommended to try out). — Elaboration / Tips:
Menypkt 1: User state radius and the program calculates (and prints) the volume. Menypkt 2: User state value for a, b and c and the program solves the equation. Menypkt 3: The program will calculate the distance between two geographical points on the earth's surface. Tip: Haver Its Formula. Menypkt 4: the program should bere gene and print a repayment plan for an annuity loan. User provides home loans, interest rates (in %) and the number of years (seen one installment = a year) for repayment. The program will calculate and print a repayment plan for the loan. If the loan is for $ 200,000, -, rate = 3.6% and the number of years / installments = 12. Menypkt 5: The user enters an integer program should write this as text (ex .: 31 = thirty, 1132 = EtTusenEtHundreOgTrettiTo). Should at least cover figures from 0-1000.

It will develop a simple system that will be used to sell tickets for showtimes. To make things simple it is only possible to manage a cinema show time. The program will be menu driven and will be used by the person who sells tickets for theater performance. — Basically, all the seats marked as vacant by means. drawn '*'. The hall can be represented by means. a two-dimensional table. Sold seat marked with the character '#'. Information about the seats that are vacant and sold should also be saved to a file. It operated with two prices: full price (eg. £90) and reduced prices (eg. £60). The menu will look something like this: — Actual performance:? 1. New performance 2. Buy Tickets 3. Show vacant seats 4. Display ticket sales for this performance 0. End — The "New performance" gives the user the opportunity to make clear to the new show at

the table (and file) that holds the free / sold seats "reset" - ie all seats marked as vacant. Here you can Also ask for the names of performances that in turn leads to the emergence of a name behind "Current Show: "in the menu above. The "Buy Tickets" allows the user to select the desired row and column. Price (1 of 2) must also recorded. Then the program asks for confirmation of the purchase. Selected seats marked as busy. — If the program terminates and is restarted, all information registered remembered. This means cinema-status is stored in a file. This "reset" through menu options 1. Option 3 "Show free seats" to print the saddle as shown above so that the user sees what is free and can recommend sites to buyers. Option 4 will show the number of tickets sold per day, income per row and total income on current performance.

Work through the document "Case: Bitmap" that is placed under uke16. Create your own 24-bit BMP / DIB file that you are testing with. — Once you've got the code, which is reviewed here to work as expected, expand the program with the following: Add function makes it possible to change one of the colors of all the pixels. Function should be made to change in an arbitrary color component (R, G or B) can be set to any color. — Function prototype should look like this: void change color (vector ¡Pixel¿ & pixel table, int component, int color) // component: 0 = R 1 = G 2 = B — Add a feature that makes the pixels in the image into grayscale following algorithm: For each pixel calculate an average value of r, g and b values. Then put the pixel r = g = b = average value. — Function prototype should look like this: void greyscale (vector ¡Pixel¿ & pixel table)

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.

Task Description
You have been commissioned by Chef S. Jeffsen on top Høgfjellshotel to create a program as he helps to keep track of nutritional information related to meals that he composes. Using the program it will be possible to compose dishes and get calculated corresponding nutrition information as energy in kcal (and / or Joule) and the content of proteins, fat, carbohydrates, vitamins, etc. — To achieve this we need to have access to nutrition information on foods. Here is a table in Excel format with nutritional values for more enn1300 goods where the amount stated as content per 100 grams of edible product. The program must have this information available to compose dishes. Chef envisions that he chooses x grams of vare1, y grams of vare2, z grams of vare3 etc. to compose a dish and then (or continuously) get printed total nutrient for the new court. The finished dish should be written to a recipe file (a separate file per dish). The recipe contains quantity (in grams) for each food and overall nutrition information (amount of protein). Food is selected by means of food number, ie. the user enters a food number (see table) to select a food item. The program will be menu driven and able to read and use information from an adjusted file

(see separate section below). — Report
It should be possible to extract "reports" from the system ie. information that is written to screen. The following shall be possible to retrieve the system: - All registered dishes / recipes listed on the screen. - Dishes with more than x (determined by user) g protein, fat or carbohydrates per 100g done right. - Dishes with more than x (determined by user) calories per 100g finished right. - Dishes with lots of vitamin D (compared to the daily recommended intake). - Dishes with much vitamin A (in relation to the daily recommended intake). - Dishes with much cholesterol (compared to the daily recommended intake). - ...It is expected that you use classes in the solution (but this is not an absolute requirement). Recipes should be stored in a structured way (files) to easily retrieve saved data on next run.

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 2.3 - 4.1, XML, SQLite

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

This was an introductory course to developing applications for Android. It consisted of four mandatory assignments and a graded assignment.

The first assignment was creating a conversion app for different measure systems (distance, volume, mass, temperature and time).

Within each of the main categories are also included examples of different devices that program to calculate from/to. The user begins by selecting the main category, then select the device to be counted from, unit to be counted and the numerical value to be converted. The program make the conversion and displays the result on the screen. To use Fragments and the application will use a different layout on small and large screens. On the big screen should be left of the screen, a list of categories that you can choose from. Right shall as corresponding subcategory to/from ETC. see you later. The program will also be multilingual ie minimum Norwegian and English. Examples: If the user selects area as a general category, one can convert from/to devices such as square millimeters, squarecentimeters, squaredecimeters, Square feet, squareinches, square yards, Acres, etc. Selected category Mass shall be converted from/to micrograms, milligrams, centigram, ounces, pounds etc. Selected temperature it should be possible to convert from/to Celsius, Fahrenheit or Kelvin.

It will develop an application (A) which can be used to manage information about books/book titles. The program will save the info. on an application associated text file. The application shall be (at least) the following activities (Activity classes): * An activity that offers an interface that displays a list of all registered titles. The user can choose a book title and this should then be returned from this activity. * An activity that allows you to register a new book title, this is stored on the file. * An activity that allows you to delete a selected book title. * Start-up activity should have a set of buttons or a menu that provides access to the other activities.

Activities in application A will then be used by another application (B). From this application, the user should make up the list which the activity of A, with titles and could choose and return selected title back to B. Selected title displayed in the application B. In addition, the user could choose to add a new title, using the activity of application A.

Every time it added a new book title sends a broadcast of application A. This should be captured by the application B and displayed by means of a Toast. The application B must therefore contain a broadcast receiver, receiving broadcasts from application A. Feel free to use Fragments where appropriate.

The task is to develop two Android applications, one that logs and stores information. about various "events" and one that serves as an interface to the log. — Application 1: Event Logs This application is by means of a BroadcastReceiver to capture and store info from system events such. receiving SMS, phone calls, GPS turned on or off, WiFi turned on or off, etc. All events are stored in a SQLite database. Each event should be associated with a category (eg. Telephony, network, positioning and the like, you determine categories and which events will be in the category). Each event will be stored with date, time, category, text, details (special parameters / values associated with the event,

eg. Who SMS comes from). When starting the application, a simple "Setting" activity appears which allows the user to decide which events (of a range of at least 5 different events from at least 3 different categories) to be logged. The application should also work so that when a kringkasing occurs, the current broadcast receiver start and store relevant information in the database. The application should not be started for it to work. The database must be "exposed" by means. a Content Provider when it will be available for Application 2 - LoggerUI. — Application 2: LoggerUI This will use info from database in app1 and display this in a simple Activity. Here user can select the category and display all events from this category (eg. from last night or the like). Here, use a ContentResolver.

The fourth assignment was 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. — It will develop an Android application that will monitor the temperature at different measurement stations used by southwest. A weather station has a unique number and name (eg. 84700, Narvik airport). Of applications shall include a service that periodically downloads weather data (observations) from yr.no and possible. Notify (using. Notification) if the temperature is outside set limits. In addition to service shall also contain an application Activity showing a list of measurement stations that the user wants to monitor. Beside the station name will last interrogated temperature displayed. The user must via a menu option to add (and possibly remove) monitor station. A weather station has a name, a measureNo and a URL. Different measuring stations can have the same URL.

After the service is started will periodically download temperature data. URL of giving forecasts for Narvik, pick out the temperature of the current station by means of XML parsing. The activity shall have the following menu options: Adding and deleting measuring stations. A weather station recorded with three values: place names, målernr and URL. Can be stored on file or in a SharedPreference.

Settings Here one should be able to set the interval to download weather data (eg. every hour, every two hours, every three hours, etc.). In addition, it should be possible to set upper and lower limit temperatures. If downloaded spot temperature is above/below these values, the service should give a notification. From the activity should also be possible to start and stop the service (should be visible "actions" in ActionBar'en). The service will continue to run in the background even if the event ends. Make sure the service starts when the device starts.

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): Java, 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.

Since users should be able to send messages to each other, they also needed to be notified if they had received a new message. This was accomplished by updating the link to the messages in the navigation bar with a number, where the number indicated how many new, unread messages they had received. Users could also 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 taking images of 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).

task1
The task is to model a Duplokloss (with 8 buds on top). It should look something like the following figures. See also the web for pictures of bricks to get the appropriate details. Like the real blocks is important. In my models depicted below, there are some shortcomings and you can do better. On the one hand should your name be in attendance for "Lego" that I have put on my. — Add an appropriate texture for the group as block appears authentic. Add in simple colors / textures that resemble blocks of reality. Texturing can take much time, but it is not the most important in the exercise, so consider how much time you spend on this.

Based on what is covered in Module 1 and 2 shall be used to model your vehicle as you outlined in Module 0 Exercise 1 Sketch. You must model with a view to future use so vehicle propulsion (wheels, belts, bone, etc.) can be animated at a later date. The same applies to the vehicle's opportunities for course change / control. — That all submissions should have roughly the same size ratio between the different vehicles and for subsequent scaling in such XNA / C#, go into the menu "Customize-¿ Units Setup" and select the "System Units Setup." Here you select the "1Unit = Centimeter" (press ok). Secondly, select the "Display Unit Scale" should be Metric and set to Centimeters. "Lighting Units" may be International. — Then draw a box with dimensions (xyz) 200,200,100 located in coordinates (XYZ) 0,0,0. This should be maximum (bounding box) for your model. The box is only to give the impression of how big your model like that about to be. The box allows you to hide (hide) and remove when needed.

Based on what is covered in Module 1 and 2 and 3 shall be used to model "race cart / vehicle trail" with trbiune and facilities to your vehicle that you modeled in "Module 2 Exercise 4 Vehicles". — You must model considering scaling the vehicle. If you use the same Unit setup as outlined in rehearsal "Module 2 Exercise 4 Vehicles", but changes the Display Units Scale to "Meters", then draw a box located to the (xyz) 0,0,0 with size 150m, 150m, 30m has a region that restricts "bounding box for the environment. — The box is only to give the impression of how big your model like that about to be. The box allows you to hide (hide) and remove when needed. Insert your vehicle onto the "path" after you have modeled the finished

Based on what is undergone in module 1..4 you should model a lavpolygon character (Avatar) by yourself. You assumes a tutor who came in an earlier version of 3dsmax, but we will turn the focus away from modulate a "Soldier" and using instead pictures of yourself to modeled one LowPoly model yourself after the existing this tutor. The tutor can be found by downloading tutorfiler from version 9sp2 of 3DSMax located at this link: http://fagweb.hin.no/axs/kilder/3dsmax_t.chm. REMEMBER, that after you have downloaded it you must right-click the file, select properties for the file and choose "unblock" or you will not see its contents. Once that's done, open the file and navigate to "Modelling" -¿ "Modeling a Low-Poly Character". Get help from someone to take pictures of yourself from the front and from the side. Make sure you do not go crazy when it comes to polygons, you should in turn right shape and animate it as a helper for the vehicle in the path you created in previous exercises. Use the same scale as the one you used to model vehicle. The avatar must fit the size of the vehicle.

Based on what is undergone in module 1..5 should you rig your avatar from previous practice with a skeleton. Please note that there is no talk of a biped here. You must use the "bones" and then the modifier "skin" after which you need to adjust "envelope" of the individual members / bones so that the skeleton motion moving the right parts of the figure. Add arm and fotrigger (as shown at the end of chapter 23 in the book) and Inverse Kinematics (IK) that you can use in animation.

Based on what is undergone in module 1..6 light the composite path, vehicle and character in a mood corresponding "evening sun" (see the book chapter 25). Rendering must function without error messages. This is an exercise in using bright spots. You should not take advantage of "Daylight System"