Software Projects

Listing and description of software projects I created in context of my studies and in my private time

Subject: Software projects

Author: Maurice Tollmien – maurice.tollmien@gmail.com

Last modification: March 4, 2016

Contents

1 Introduction

In context of job applications and interviews, it is not always clear, what skills and capabilities the candidate (in this case - me) has and where the main technical competence lies.
During the last years, it became apparent, that a simple listing of relevant projects with short desciptions often circumvents and avoids such uncertainties.

With those projects, often linked with the correspondent repositories, I hope to create an impression of my strengths and capabilities.

Most of my more relevant and meaningful projects are uploaded or developed using the following GitHub repository:

`https://github.com/MauriceGit/`

The list of exercises and projects is not complete and rather serves to create a valid impression of the technical side of myself. Mostly the listed projects were created in context of my studies of computer science (Bachelor and Master). Several others I build in my private time.

2 Computer Graphics and Virtual Reality

2.1 Basic Computer Graphics Projects

2/3D programs in C, using the OpenGL interface and pipeline. Specific projects vary from different lighting computations in 3D, shadow volumes and transparancy to higher order spline interpolation.
Later projects include physically approximated calculations for sphere movement on spline surfaces, collision detection and particle simulations to give some examples.
Most later implementations make extended use of the OpenGL shader language GLSL for vertex, fragment and compute shaders.

2.2 Virtual Reality Project

The main goal of this project was to implement software to make a head mounted display compatible to Linux. This included reverse engineering the USB bitstream and interpret it as Quaternions for further use in arbitrary software. The final state of this program was a C library which could be used in any program using the OpenGL pipeline to create a real 3D experience using the head mounted display.

2.3 Procedural Modelling

Creation of a 3D model (from scratch) of a self picked item by use of the Autodesk Softimage software. The resulting object file could then be included into further projects.

This project (including screenshots) is uploaded on my GitHub account:

`https://github.com/MauriceGit/Procedural_Modeling_Spyder`

2.4 Cloth Simulation

This project was created entirely in my private time. It implements basic cloth simulation using a particle based approach with virtual springs for structure. The cloth interacts with different objects like spheres, quads, planes, as well as virtual wind.
This project (including screenshots and further descriptions) is uploaded on my GitHub account:

`https://github.com/MauriceGit/Cloth_Simulation`

2.5 Water Simulation with Reflections/Refractions

Implementation of a pressure based water simulation, internally represented as a height field. The surroundings are procedurally generated using a self implemented Perlin noise.
The reflections and refractions are entirely computed in screen space, using the OpenGL fragment shader.

This project (including screenshots and further descriptions) is uploaded on my GitHub account:

`https://github.com/MauriceGit/Water_Simulation`

2.6 Particle Simulation on GPU

Implementation of a particle simulation (Euler integration) by using the OpenGL compute shader. All computations, regarding the particles, are done by the GPU entirely. Leading to impressive speed and particle count.

Further information and source code is uploaded on my GitHub account:

`https://github.com/MauriceGit/Partikel_accelleration_on_GPU`

2.7 Screen Space Ambient Occlusion (SSAO)

A private project, implementing a version of screen space ambient occlusion. Adjustments to the ambient occlusion are possible while running the sample. The complete implementation of SSAO is done using the OpenGL fragment shader.

Further information and source code is uploaded on my GitHub account:

`https://github.com/MauriceGit/Screen_Space_Ambient_Occlusion`

2.8 Environment Mapping

As extension to the practical water simulation project (see 2.5 „Water Simulation with Reflections/Refractions“), I presented a seminar on how environment mapping could be implemented and realized in virtual environments. Mostly featuring screen space reflections and parallax corrected cubemaps.

2.9 Delaunay and Voronoi Partition of Images

As a private project, I implemented a software which takes an image, extracts points according to the local image contrast and triangulates them with a Delaunay triangulation (implemented completely from scratch). The triangulation is then converted to voronoi regions. The output are images with the delaunay/voronoi regions colored in the mean color of the particular region.

Further information and source code is uploaded on my GitHub account:

`https://github.com/MauriceGit/Delaunay_Triangulation`

3 Artificial Intelligence

3.1 Game AI (Bachelor Thesis)

In the course of my Bachelor thesis, I created a dissertation in a subdiscipline of artificial intelligence with a complex board game (Arimaa[1]) as subject. Next to the theoretical approach, I implemented most of the relevant algorithms such as alpha-beta pruning, transposition tables, Zobrist hashing, move ordering and parallelization.
The dissertation won an award for innovation (Wedeler Hochschulpreis).

The dissertation is published on the official Arimaa homepage:

`http://arimaa.com/arimaa/papers/MauriceTollmien/Thesis.pdf`

The source code of the AI is uploaded and available on my GitHub account:

`https://github.com/MauriceGit/Bachelor-Thesis-Arimaa`

3.2 Pure Logical Agent

Design and implementation of a pure logical agent for the game of Wumpus World, which dynamically acts according to the game situation. The resulting logical expression is then solved, using a SAT solver (Picosat[2]).
The implementation is functional and written in Haskell. The project is uploaded on my GitHub account:

`https://github.com/MauriceGit/AI-Logical-Agent`

3.3 Artificial Neural Network

This project was created for university and proceeded in my private time. It contains a from-scratch implementation of a backpropagation neural network for letter recognition. Several extensions are implemented, like gradient descent and dynamic hidden layer count and size.
In the end it turns out to be on the same level of effectiveness and efficiency as common backpropagation neural networks like PyBrain[3].

The implementation as well as a presentation are uploaded on my GitHub account:

`https://github.com/MauriceGit/Artificial_Neural_Network_Character_Classification`

[1] `http://arimaa.com/arimaa/`
[2] `https://hackage.haskell.org/package/picosat`
[3] `http://pybrain.org/`

4 IT Security

4.1 Seminar Paper on Compiler Backdoors

Preparing and presenting a seminar paper on backdoors in software implementation, especially in a compiler itself.
The seminar paper is written in english and published on the website of the university:

`http://www.fh-wedel.de/~gb/seminare/ws2012/tollmien_backdoors.pdf`

4.2 Security Engineering Project

Implementation of a small program in C and cracking/decompiling the implementation of a fellow student with use of common tools like hexdump, gdb and objdump.

4.3 Cryptography Project

Extended workshop on cryptographic methods and algorithms. The workshop covered the practical approach as well as the mathematical background of common cryptographic encryption, hashing and signature algorithms.
Included subjects in this workshop were for example: DES, GOST, AES, message integrity/authenticity, random number generators, number theory, asynchrone techniques like Diffie-Hellman, RSA, elliptic curve cryptography and one way accumulators.

Every task and correspondent solution is uploaded on my GitHub account:

`https://github.com/MauriceGit/Cryptography_Workshop`

4.4 Secure File Encryption

As a private project I implemented an interactive secure file encryption in Python. It alows encryption/decryption, read mode and write mode (using Linux ‚less‘ and ‚nano‘ tools). The encryption consists of hashing the password and AES.

The complete implementation can be viewed under the following link:

`https://github.com/MauriceGit/Secure_File_Encryption`

4.5 Mental Poker

In this project a group of students (me included) extended the official Bouncy Castle[4] Crypto library with the SRA Algorithm which can be used for mental poker. The main difference to the common RSA algorithm is its commutative behaviour regarding the encryption and decryption.

The SRA implementation is published under the following link:

`https://github.com/mervyn-mccreight/mentalpoker`

5 Programming Contests

5.1 Shortest path - 2013

I took part in a programming contest with the goal, to find the shortest path through a set of points. I used a mesh triangulation (Delaunay) from computer graphics and was able to reach the 3rd rank overall.

5.2 2048 AI - 2014

I participated in a programming contest with the main goal to implement a fast and efficient AI for the game 2048[5]. I reused most of my AI algorithms from my bachelor thesis (see 3.1 „Game AI (Bachelor Thesis)“) and was able to reach the 2nd rank overall.

The complete source code is uploaded on my GitHub account:

`https://github.com/MauriceGit/2048_Game_AI`

5.3 SameGame AI - 2015

I participated in the programming contest for implementing an efficient AI for the game SameGame[6]. My implementation used a Monte-Carlo tree search. I was able to reach the 3rd rank overall.

The complete source code is uploaded on my GitHub account:

`https://github.com/MauriceGit/Programming_Contest_2015`

[4] `https://www.bouncycastle.org/`
[5] `https://gabrielecirulli.github.io/2048/`
[6] `https://de.wikipedia.org/wiki/SameGame`

5.4 Organization of the Programming Contest 2016

This year's programming contest will be organized by a fellow student and myself. The problem definition is not yet public and we are right now working on the server and middleware structure for bot testing and visualization.
The topic will be about writing an AI for an Agar[7]-like environment.
The software is still under construction but runnable and can be viewed with the following links.
Server (Handles middlewares and complete game logic):

`https://github.com/hpatjens/Programmierwettbewerb-Server`

Middleware (Manage bot in/output and connects to server):

`https://github.com/hpatjens/Programmierwettbewerb-Middleware`

Gui (Connects to server and visualizes the game with all connected bots)

`https://github.com/hpatjens/Programmierwettbewerb-Gui`

6 Other Projects

6.1 Android App

In cooperation with the wildlife park Eekholt[8] we developed an Android App as a navigation help for visitors.

6.2 Lua Compiler

As part of a team of several students, we developed a Lua interpreter from scratch in Java. My part was mostly the code generation (Generate assembly like code from an abstract syntax tree) but was involved in decision making of the runtime environment and parser/scanner development as well.

6.3 Distributed Systems in Erlang

Partly for university, partly as a private project, I developed a few algorithms for distributed systems (like bully algorithm, time control, mail delivery, ...) in Erlang.

The implementation of the bully algorihm is uploaded on my GitHub account:

`https://github.com/MauriceGit/Distributed_Systems_Bully_Algorithm`

[7] `http://agar.io/`
[8] `http://www.wildpark-eekholt.de/`