

Rotierende LED Anzeige



Maturaarbeit Kantonsschule Sargans

Loris Graf, 4NP

Betreuung:
Herr Thomas Büsser

Eingereicht am:
5. Januar 2015

Zusammenfassung

Themenumschreibung

Ich möchte eine mehrfarbige Anzeige entwerfen, welche Texte und Bilder scheinbar in die Luft schreiben kann. Dazu sollen schnell rotierende LEDs mithilfe eines Mikrocontrollers angesteuert werden.

Zentrale Fragestellung

Lassen sich mit heutiger LED-Technik mehrfarbige Anzeigen realisieren, welche alleine durch Rotieren verschiedenfarbiger LEDs den Eindruck eines Gesamtbildes erzeugen?

Welche Umdrehungszahl ist nötig, um dem menschlichen Auge ein flimmerfreies Bild zu suggerieren, ab wann werden einzelne dunkle Abschnitte sichtbar?

Vorwort

Motivation

Vor etwa drei Jahren habe ich in einem Elektronikladen ein witziges Gadget entdeckt: Ein kleiner USB Ventilator, an dessen Rand sich 6 kleine rote LEDs befanden. Sobald der Ventilator eine gewisse Geschwindigkeit erreicht hatte, schrieb er einen von ca. fünf möglichen Sätzen in die Luft. Ich war so fasziniert von diesem Gerät, dass ich es unbedingt nachbauen und um weitere Funktionen erweitern wollte. Leider hatte ich damals noch keine Erfahrung mit Mikrocontrollern und hatte wenig Ahnung, wie ich das Projekt angehen sollte. So realisierte ich das Vorhaben nicht sofort. Es blieb jedoch ein Vorhaben, welches ich unbedingt einmal umsetzen wollte. Vor etwa einem halben Jahr entdeckte ich einige günstige Angebote für Mikrocontroller und tauchte in deren Welt ein. Als ich mich für ein Maturaarbeitsthema entscheiden musste, war mir sofort klar: Ich wollte unbedingt mein Wissen über Mikrocontroller erweitern. Es sollte eindrücklich, lehrreich und vor allem spannend sein. Da erinnerte ich mich wieder an die LED Anzeige, sie erfüllte alle von mir gesetzten Kriterien. Ich war von Anfang an hochmotiviert.

Kantonale Mittelschule Uri

MARUSENKOSPHERE

Programm zur Lösung und Visualisierung der
MarusenkoSphere

Marcel Würsten

Maturaarbeit 2014

Begleitperson: lic. en cienc. N. Cantà



Inhaltsverzeichnis

1	Einleitung	2
2	MarusenkoSphere	2
3	Lösungsalgorithmus	4
3.1	Weg zum Lösungsalgorithmus	4
3.2	1. Lösungsalgorithmus	5
3.3	2. Lösungsalgorithmus	5
4	Grafische Benutzeroberfläche	6
4.1	Das Controlpanel	6
4.2	Wahl der Grafikbibliothek	7
4.3	Portabilität mit LWJGL	8
4.4	Darstellung der Kugel	9
4.5	Algorithmus zum Zeichnen der runden Kugel	10
4.6	Animation der Kugel	12
4.6.1	Matrizen rechnen	12
4.6.2	Koordinatentransformation	13
4.6.3	Rotationsmatrix	13
4.6.4	Anwendung bei der Animation	14
4.7	Editor	14
4.7.1	Benutzerinteraktion	15
4.7.2	Baryzentrische Koordinaten	16
5	Speicherung der Kugel	17
5.1	Die Datenstruktur	17
5.2	Modulo	19
6	Schlusswort	20
7	Dank	21
8	Literaturverzeichnis	22
9	Abbildungsverzeichnis	26
10	Glossar	27
11	Selbständigkeitserklärung	30
12	Anhang	31

Abstract

Diese Arbeit beschäftigt sich mit der MarusenkoSphere und wie sie mit dem Computer in möglichst wenigen Schritten gelöst werden kann. Dafür wurde mit *Java* und dem *OpenGL-Binding* LWJGL ein Programm geschrieben, welches die MarusenkoSphere simuliert. In dieser Simulation kann eine Kugel zufällig generiert werden oder der Benutzer kann in einem integrierten Editor den Stand der eigenen Kugel erfassen. Die dreidimensionale Kugel kann dabei von allen Seiten frei besichtigt werden. Dem Lösungsprozess ist dank einer Animation einfach zu folgen und die Lösungsalgorithmen werden klar dargestellt. In der Simulation lösen zwei geschriebene Algorithmen die Kugel. Dabei wird immer der Algorithmus mit dem kürzeren Lösungsweg gewählt. Neben der grafischen Oberfläche sind diese Algorithmen der grösste Teil dieser Arbeit.

1 Einleitung

Als die Frage bezüglich des Themas für die Maturaarbeit aufkam, war für mich schnell klar, dass ich etwas programmieren möchte. Die Suche nach einem geeigneten Thema war jedoch schwierig. Man findet kaum etwas, wofür es noch kein Programm gibt, zumindest was im Rahmen einer Maturaarbeit bearbeitet werden könnte.

Nach den Sommerferien 2013 brachte Frau N. Canta eine MarusenkoSphere aus ihren Ferien in Spanien mit. Ich war sofort fasziniert von der Kugel. Als die Themenwahl für die Maturaarbeit aktuell wurde, erinnerte ich mich daran und entschloss mich dies als Thema zu nehmen.

Circa 8400 Zeilen oder bei Schriftgrösse 12 etwa 280 A4 Seiten Programmcode sind das Resultat des geschriebenen Programmes. Das Ziel dieser Arbeit war ein Computerprogramm zu entwickeln, welches die MarusenkoSphere lösen kann. Hierzu musste ein Lösungsalgorithmus gefunden und implementiert werden. Damit dieser Lösungsalgorithmus visuell geprüft werden konnte, erhielt das Programm eine grafische Benutzeroberfläche, welche dem Betrachter die Kugel visualisiert. Das Programm kann sowohl zufällige, als auch vom Benutzer eingegebene Kugeln lösen und darstellen.

2 MarusenkoSphere

Die MarusenkoSphere ist ein 3D-Puzzle welches vom Ukrainer Aleksandr Marusenko erfunden wurde. Die Teile der Kugel können dabei in alle 3 Dimensionen gedreht werden (siehe Abbildung 1). Zusätzlich kann man noch die Polkappen um die Polachse drehen. Die Kugel besteht aus 54 Teilen, wovon 32 an der Oberfläche zu sehen sind (24 Dreiecke und 8 Verbindungsstücke) [1]. Die Kugel beinhaltet somit etwa $2.3 \cdot 10^{30}$ mögliche Positionen. Im Vergleich dazu, der 3×3 Rubik's Cube hat $4.3 \cdot 10^{19}$ Möglichkeiten. Der Rubik's Cube, ein mechanisches 3D-Puzzle, hat sich seit Mitte der 1970er Jahre zu einem der beliebtesten Spielzeuge entwickelt. Über 350 Millionen Exemplare wurden davon bis heute verkauft [2]. Dies nahm der Ukrainer Aleksandr Marusenko

“Bitcoin Casino”

The development of a next generation gaming platform.

Ilian Vogels, 5B

Academic Year 2014-2015

School: Gymnasium am Münsterplatz Basel

Supervisor: Mr. Helmar Meier

Second examiner: Mr. Bruno Hümmer

Submission date: 20th of October 2014

0 Preface

By writing this dissertation and the associated project, I was challenged to develop a working web application using my own programming and writing skills. It was also an opportunity to work with and learn about several new programming languages and software which will be discussed later in this essay. The first time I heard about Bitcoin, I was astonished by the concept and possibilities of this digital currency, so I've decided I want to use this currency for this project. On the internet, I've seen several examples of Bitcoin gambling sites, but they all lack the functionality of playing against a real player without any fees. Therefore I came up with the idea of creating a complete web application for users to enjoy an interactive gaming experience using the Bitcoin currency.

I'd like to thank Mr. Meier for supporting me with this project by giving me feedback of my ideas and motivating me during the development process of the project and while I was writing this dissertation.

My dad has also been a great help with testing the platform while in development. He played several games with me on the platform, and has reported several bugs, which I am very thankful for.



DER SPRUNG IN DIE PROGRAMMIERWELT

Maturarbeit 2015

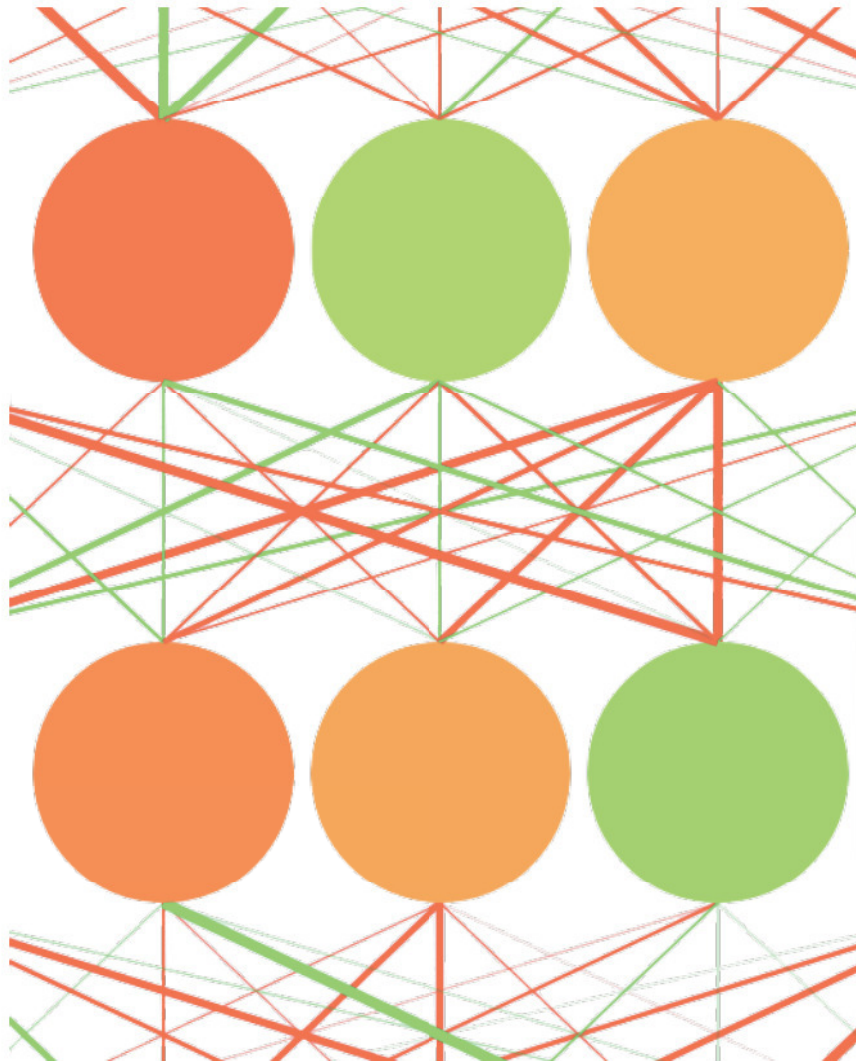
Laure Ciernik, Klasse 6c
Betreuer Dr. Markus Bott
Literargymnasium Rämibühl

Abstract

Das Internet hat unser aller Leben verändert. Schon Jugendliche und Kinder lernen früh, wie man eine Suchmaschine, also einen Browser¹, wie Internet Explorer, Firefox, Safari oder ähnliche, benützt. Suchmaschinen erlauben einem in die Welt des Internets einzutauchen, Inhalte zu öffnen und unzählige verschiedene Websites einzusehen. Doch was alles steckt eigentlich hinter einer einzelnen Website? Wie kommt es dazu, dass Websites überhaupt so aussehen und wie entstehen sie? Diese Maturarbeit behandelt diese Fragestellungen anhand der Erarbeitung einer Internet-Plattform für eine Mädchenpfadfinderabteilung (Abteilung Agua in der Stadt Zürich) und bietet die Möglichkeit einen Blick in die Kunst des Webprogrammierens zu werfen.

¹ Definition (<http://wirtschaftslexikon.gabler.de/Definition/browser.html#definition>): „Der Browser ist ein Programm zur grafischen Darstellung der Inhalte des World Wide Web (WWW)“; Datum: 08.12.2014

Künstliche neuronale Netze in einfachen Spielen



Maturaarbeit von Mattia Humbel und Linus Geiser
Alte Kantonsschule Aarau, Abteilung G4E

Betreuungsperson: Lukas Wampfler

Abgegeben am 15.10.2014

3 Problemstellung

Die vorliegende Arbeit befasst sich mit dem Gebiet der künstlichen neuronalen Netze. In diesem Untergebiet der künstlichen Intelligenz geht es darum, inspiriert durch natürliche Netzwerke von Neuronen, lernende Strukturen zu schaffen. Ein Beispiel eines natürlichen neuronalen Netzes ist unser Gehirn, welches täglich hohe Leistungen erbringt.

Dieses Gebiet ist aus vielen Gründen faszinierend. Jeden Tag arbeiten weltweit mehrere Milliarden Gehirne mehr oder weniger einwandfrei und trotzdem ist von ihrer Funktionsweise noch vieles unbekannt. Deshalb werden auch millionenschwere Projekte zur Erforschung der Lernvorgänge im Gehirn gestartet. Eines davon ist das Human Brain Project der Europäischen Union, an welchem die ETH Lausanne (EPFL) mitarbeitet. [1]

Gleichzeitig stellen künstliche neuronale Netze auch eine Alternative zu klassischen Rechnern dar. Ein herkömmlicher Computer arbeitet seriell. Das heisst, er hat nur eines oder wenige Rechenwerke, welche in einem sehr hohen Tempo viele kleine Arbeiten erledigen. Im Gegensatz dazu arbeitet ein neuronales Netz parallel. Es gibt also viele kleine Rechenzentren, welche einfache Arbeiten erledigen. Da neuronale Netze in etlichen Bereichen, so zum Beispiel den kognitiven Fähigkeiten, Computern weit überlegen sind, verspricht man sich viel von netzwerkartigen Rechenstrukturen. [2:25f] [3:3ff]

Das Themengebiet der künstlichen Intelligenz und besonders der künstlichen neuronalen Netzwerke wird zunehmend auch in Zeitschriften und Zeitungsartikeln thematisiert (siehe zum Beispiel [4]). Auf diesem Weg haben wir auch die ersten Male davon erfahren. Nach etwas tieferer Beschäftigung mit dem Thema und ersten Versuchen erschienen uns künstliche neuronale Netze spannend für eine Untersuchung in Form einer Maturarbeit.

Unsere anhaltende Faszination begründet sich wohl in der Breite und Tiefe des Themas. Es gibt etliche verschiedene Netztypen und Trainingsverfahren, die weitgehend eigenständig sind. Dabei gibt es noch viele offene Fragen, welche zur Erforschung locken.

Eine weitere Motivation für unsere Arbeit war das dabei entstehende Produkt, ein Programm, mit dem man sich aktiv beschäftigen kann und das den Erfolg der Arbeit zeigt.

Das Lernen soll im Rahmen von einfachen Spielen stattfinden. Der Grund dafür ist die Anschaulichkeit. Gibt es bei vielen Problemen nur den Ausgang ‚das Netz hat das Problem gelöst‘ oder ‚das Netz hat das Problem nicht gelöst‘, ist bei Spielen eine problemlos visualisierbare Abstufung vorhanden. Das Resultat des Lernens kann so einfach gezeigt und verstanden werden. Ausserdem handelt es sich nicht nur um eine Aufgabestellung, deren Lösung schon vorgegeben ist, sondern eher um eine Knobelaufgabe, bei der Möglichkeiten gesucht werden müssen.

Gerade wegen der Vielfalt des Themengebiets mussten wir uns aber für diese Arbeit beschränken. Wir legten fest, nur den Netztyp des **Perceptrons** mit dem **Backpropagation**-Lernalgorithmus untersuchen zu wollen. Warum es genau dieser Typ ist und was darunter zu verstehen ist, wird weiter erklärt werden. Zusätzlich musste das ausgewählte Spiel im Vergleich zu anderen relativ simpel sein, um dem neuronalen Netz das Lernen zu ermöglichen. Die Wahl fiel schliesslich auf *Vier Gewinnt*.

Am Anfang der Arbeit werden im Kapitel *Ziele und Fragestellungen* die für die Arbeit gesetzten Ziele vorgestellt. Das Kapitel *Theoretische Grundlagen* führt in das Thema der neuronalen Netze und deren Lernverfahren ein, sowie in für das Programmieren wichtige Themen. Im Kapitel *Vorgehen, Methoden, Material* wird anschliessend die Behandlung der Fragestellungen beschrieben. Die Ergebnisse werden im Kapitel *Resultate* vorgestellt und anschliessend in der *Diskussion* ausgewertet. Das darauffolgende Kapitel ist ein *Fazit* zur Arbeit und dem dahintersteckenden Prozess.

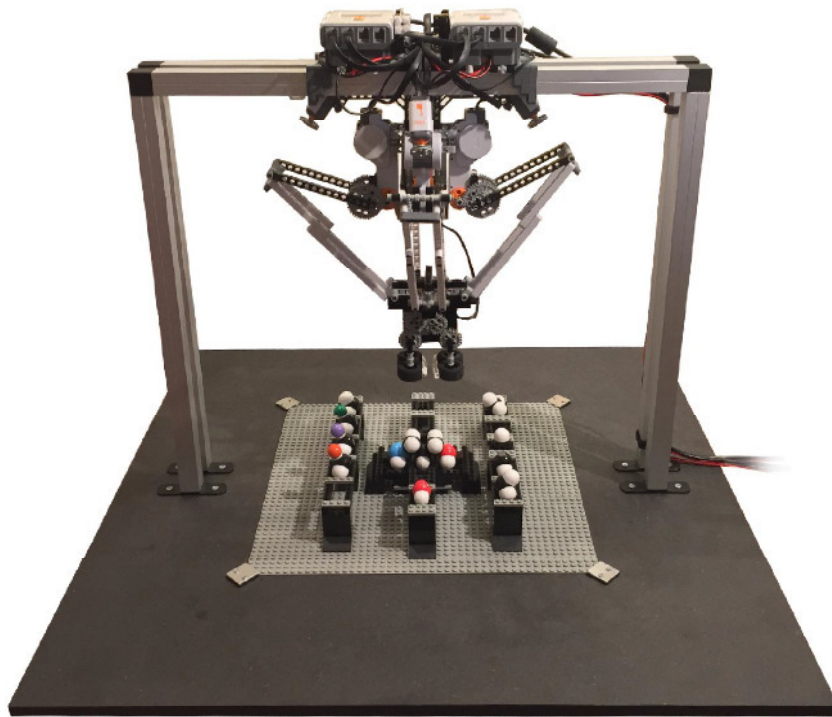
Wer sich einen ersten Überblick verschaffen will, findet nach dem Fazit eine kurze *Zusammenfassung* der Arbeit. Darauf folgt das *Abbildungs- und Tabellenverzeichnis* und ein *Glossar*, in welchem wichtige Begriffe erklärt werden. Stösst man während der Lektüre dieser Arbeit auf fett gedruckte Fachausdrücke, so sind diese im Glossar zu finden. Begriffe werden jedoch jeweils nur beim ersten Erscheinen hervorgehoben. An den verwendeten Quellen Interessierte finden vor dem *Anhang* eine *Bibliographie*.

Der Anhang enthält Anmerkungen zur *Notation*, ein *Beispiel* für das Ergebnis der Trainingsfunktion, die *Anti-Plagiat-Erklärung* und eine *CD*.

Die CD beinhaltet das erstellte Programm und den Quellcode, sowie die Datei mit den Trainingsbeispielen, die später erwähnt wird. Diese Inhalte dienen dazu, Bezüge auf das Programm verständlicher zu machen und die *Resultate* zu ergänzen.

ΔELTAFOUR

Ein Delta-Roboter setzt Molekülmodelle zusammen



Maturitätsarbeit im Themengebiet der Mechatronik

von Victor Klemm, Klasse 6e

Kantonsschule Hohe Promenade, Gymnasium, Zürich

Schuljahr 2014/2015

Betreuer:

Michael Liebich

Korreferentin:

Dr. Elisabeth Ruh

1. Vorwort

Technische Konstruktionen haben mich schon immer fasziniert. Das LEGO System mit seiner Vielfalt an Bauelementen hat sich bisher in jedem Alter geradezu als ideal angeboten. Das Konstruieren und Modifizieren von Mechanismen, Maschinen und Fahrzeugen machte und macht mir immer noch ausnehmend viel Spass.

Weihnachten 2007 bescheerte mir das LEGO Mindstorms NXT-Set. Mit zehn Jahren war ich noch zu jung, um das volle Potential eines programmierbaren Mini-Computers mit Motoren und Sensoren zu erfassen.

Vor zwei Jahren (2012) wurden mein Klassenkollege Dominik Mannhart (der die Faszination für Roboter, Maschinen und vor allem auch für LEGO mit mir teilt) und ich von unserem Mathematiklehrer Dr. Stefan Grieder auf einen Robotik-Wettbewerb von Helveticrobot hingewiesen, bei dem man mit einem innerhalb von 24 Stunden selbstkonstruierten LEGO-Roboter verschiedene Parcours absolvieren musste. Als Team haben wir sowohl im Januar 2013 als auch 2014 in Chur bei diesem spannenden Wettbewerb mitgemacht. Dabei habe ich das vielfältige Potential des NXTs neu entdeckt. Darüber hinaus hat es mir ausserordentlich Spass gemacht, einen Roboter selber zu konzipieren, zu konstruieren und zu programmieren, um eine vorgegebene Aufgabe zu bewältigen.

Die Passion für LEGO und die Erfahrungen in Chur gaben den endgültigen Impuls, das Thema für meine Maturitätsarbeit an der Kantonsschule Hohe Promenade Zürich aus dem Themenbereich der Mechatronik zu wählen.



Abb. 1: Robotik-Wettbewerb in Chur 2014

2. Einleitung

Das Ziel meiner Arbeit ist, einen Delta-Roboter zu konzipieren, zu konstruieren und zu programmieren. Der Roboter soll mittels eines Computers gesteuert werden, welcher mit dem Roboter verbunden ist und mit diesem interagiert. Der Benutzer soll auf dem Computer über ein GUI¹ Moleküle auswählen können, die der Roboter dann automatisch aus verschiedenen, vorgegebenen Molekül-Teilen zu einem Molekülmodell zusammenbaut. Dieses Ziel habe ich am 4. Juli 2014 festgelegt. Anfangs musste ich mich mit folgenden Themen auseinandersetzen:

- Wie baue ich einen stabilen und funktionstüchtigen Delta-Roboter?
- Wie soll der Roboter die Molekül-Teile greifen und wieder absetzen?
- Wie programmiere ich den Roboter und wie soll das GUI aussehen?
- Welche Art von Molekülmodell verwende ich? Stelle ich es allenfalls selber aus bemalten Holz- oder Styroporkugeln her ?
- Wie baue ich eine Plattform und ein passendes Träger-Gestell, damit der Roboter die Molekül-Teile von oben greifen kann?

Genauere Spezifikationen, wie z.B. wie viele und welche Moleküle der Roboter am Schluss bauen soll, liess ich offen, um sie nach Bedarf auf die Funktionalität des Roboters anpassen zu können.

In dieser Dokumentation wird der Entwicklungsprozess beschrieben, werden die Probleme thematisiert und das Endprodukt erklärt.

¹GUI steht für Graphical User Interface,
auf Deutsch: graphische Benutzeroberfläche

Multi-Agenten-Pfadplanung
Künstliche Intelligenz am Beispiel der
Pfadplanung für mehrere Agenten

Jonas Passweg 5B

Betreuungsperson: Christian Datzko

Koreferentin: Gabriele Röger

8. Dezember 2014

1 Vorwort

Ich möchte gerne meine Matura-Arbeit mit der Frage beginnen, die mir oft gestellt wurde: Wieso mag ich Informatik? Um dies zu beantworten, muss ich früh in meine Kindheit zurückschauen. Ich weiss noch, dass ich nach dem ersten sportbedingten Muskelkater dachte, dass ich diese Schmerzen jedes Mal nach dem Sport wieder bekommen würde. Als ich dann später wieder Sport machte, aber weniger Muskelkater bekam, war ich verwundert. Auch sagte meine Mutter, ich sei ein Kind gewesen, das gerne Zahlen gehabt hat. Bei Autos hätte ich mich immer auf die Schildernummer anstatt wie alle anderen Knaben auf die Automarke konzentriert. Ich wurde mir der Vorliebe nach logischem Denken und der Kommunikation mit Zahlen bewusst und wählte deshalb das Naturwissenschaftliche Gymnasium Kirschgarten.

In dieser Schule konnte ich anderen Schülern beim Programmieren zuschauen und fand eine Welt, die meinem Denken entgegenkam. Denn, bei gleicher Eingabe kommt in der Welt der Informatik die gleiche Ausgabe heraus. Dies fand ich sehr interessant und die Möglichkeit selbst etwas zu schaffen, zog mich vollends in die Welt der Informatik. Ich konzentrierte mich darauf, wie ein Computer zu denken, damit dieser meine Befehle korrekt ausführen kann.

Umso spannender fand ich deshalb die Idee, einen Computer mit Künstlicher Intelligenz zu versehen, damit er "wie ein Mensch" denken würde. Natürlich musste ich feststellen, dass das mit der Künstlichen Intelligenz nicht einfach ist, doch das machte es für mich noch spannender. Denn bisher konnten nur Programmierer Computern Befehle geben, eine Künstliche Intelligenz hingegen könnte mit Menschen in einfach verständlicher Sprache kommunizieren. So war für mich die Künstliche Intelligenz ein Computersystem, das wie ein Mensch zwischen verschiedenen Möglichkeiten, mit einer bestimmten Strategie, entscheiden kann.

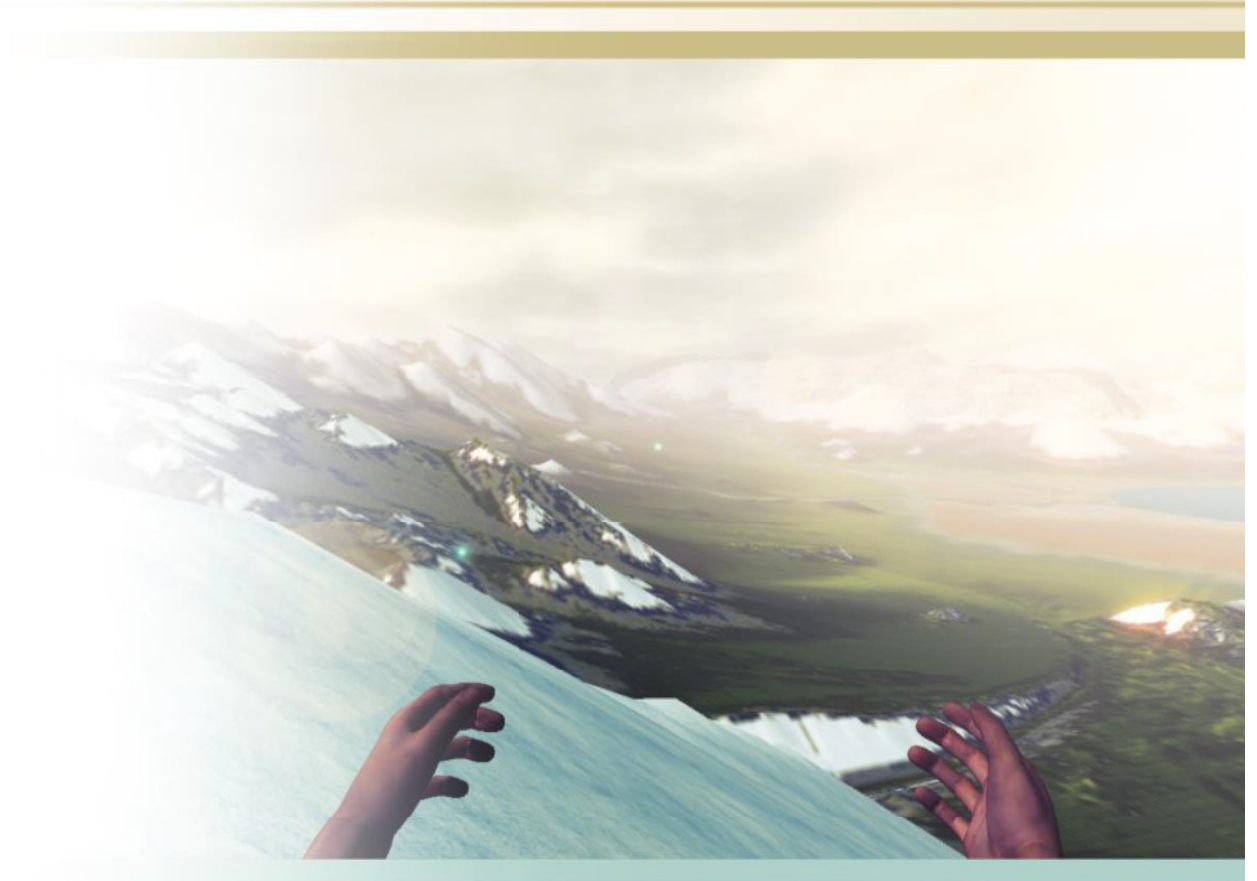
Auch spielte das Interesse für und die Angst vor einem Roboter, der für oder anstatt der Menschheit entscheidet, eine Rolle. Deshalb wählte ich die Multi-Agenten-Pfadplanung für meine Matura-Arbeit. Schliesslich überzeugte mich die Idee, dass eine Kommunikation innerhalb eines Computersystems stattfinden würde, da diese strikt gewissen Regeln folgen muss.

Ich möchte meine Matura-Arbeit mit einem Zitat beginnen, das ich von mehreren Personen (mehrheitlich Informatiker) gehört hatte und mich dazu anregte, die Künstliche Intelligenz unter die Lupe zu nehmen:

*„A year spent in artificial intelligence is enough to make one believe in God.“
Perlis (1982, S. 12)*

Ich will dieses Zitat hier nicht interpretieren, doch habe ich mir viele Gedanken darüber gemacht und über die Verbindung zwischen der Künstlichen Intelligenz und einem noch komplexeren System nachgedacht.

DIE ENTWICKLUNG EINES DREIDIMENSIONALEN OPEN-WORLD SURVIVAL GAMES MIT DER GAME-ENGINE UNITY



**MATURAARBEIT VON ELIAS KÜNG
IM FACH INFORMATIK
BETREUUNGSPERSON: R. RIEDERER
KORREFERENT: R. STEIGER
KANTONSSCHULE SCHAFFHAUSEN
2. DEZEMBER 2014**

ABSTRACT

Im Rahmen meiner Maturaarbeit habe ich eine First Playable Demo Version¹⁹ meines Spiels "MERCILESS – Witnessing Survivors " produziert. Dazu verwendete ich die Game Engine¹ Unity und eine grosse Anzahl anderer Softwares. Das Modellieren⁷, Texturieren¹⁰ und die Animation¹⁴ von Spielobjekten wurde hauptsächlich in 3ds Max vorgenommen, während der programmiertechnische Teil in MonoDevelop entwickelt wurde. Alle benötigten Objektverhalten wurden entweder als Java-²⁶ oder C#-Scripts³² verfasst.

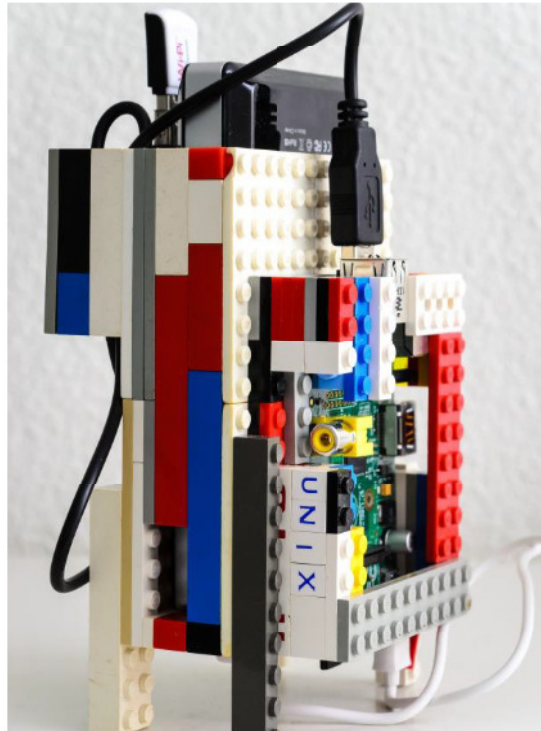
Die Spielgeschichte von "MERCILESS" findet in einer kurzen Warmzeit vor etwa 15'000 Jahren mitten in der Region Schaffhausen statt. Die Geschichte handelt von einem im Kesslerloch beheimateten, steinzeitlichen Jäger. Seine Stammesippe wurde durch den Angriff eines fremden Volkes, den Drakemern, ausgelöscht. Schon bald macht er Bekanntschaft mit einer Schicksalsgefährtin und gemeinsam rächen sie sich an den Bewohnern des mörderischen Stammes.

Das Spiel enthält Funktionen wie das Speichern und Laden von Daten, die Herstellung von Objekten an gewünschten Standorten im Spiel, das Jagen von Tieren, das Sammeln und Einnehmen von überlebenswichtigen Ressourcen und etliches mehr.

Sehr oft sah ich mich während meiner Arbeit mit Problemen konfrontiert. Viele unter ihnen waren lösbar, andere bis zum jetzigen Zeitpunkt noch nicht. Eine der heikelsten Problematiken innerhalb meiner Spielentwicklung war die ausgesprochen hohe Systembelastung des Computers durch "MERCILESS" und das im Extremfall daraus resultierende Stocken des Games. Neben den allgemein systemfreundlichen Vorgehensweisen und Verhaltensregeln stünden eigentlich aber allerhand Werkzeuge für einen flüssigeren Spielverlauf zur Verfügung. Dieser Sachverhalt wird im Diskussionsteil noch genauer untersucht.

Indoor Positioning

Development of an indoor location determination system using WLAN
fingerprinting



Maturaarbeit im Fach Physik / Elektrotechnik

Kantonsschule Sectal

Kurzzeitgymnasium

Eingerichtet von Yannick Strümpfer

Referent: Dipl. phys. ETH, Christoph Kalt

Korreferent: Dr., lic. phil. II, Marc Meyer

Schuljahr 2014/15

Baldegg, 14. Oktober 2014

1. Introduction

The idea to design an indoor positioning system is actually over two years old. Back then I dreamed of a system that could create a three dimensional X-ray scan of my room in order to tell me where to find certain objects I could not find anymore. Struggling to decide on an appropriate subject for my Matura thesis, I remembered this vision and decided to implement a system to locate objects indoors. Of course the actual positioning works totally different than in my imagination. However, the motivation is the same. I personally tend to forget and lose things wherever I am and I suppose I am not the only one. I believe that this is just human, at least to a certain extent. To me it is always the most agonizing if I lose something in our own house. Knowing that you are so close to whatever you are searching but not being able to find it makes me go mad from time to time. With my Matura thesis I therefore want to create a positioning system in an indoor surrounding to help me solve this problem and show the possibilities we have to compensate for our innate imperfection.

Over the last few decades computer technology has gradually facilitated our life and contributes to solving several task in our daily routine. One of the great achievements was the establishment of the Global Positioning System (GPS). Its applications, such as car navigation or global tracking of GPS devices seem essential nowadays. Yet other forms of location determination is subject of today's research, which is due to the fact that GPS positioning is not applicable in indoor environments. The reason for this is that the satellite's GPS signals are blocked by walls and its resolution does not suffice.¹ There are several signal transmission techniques that, however, enable indoor location determination, for instance Bluetooth, Infrared, RFID, Ultrasonic or WLAN, the last of which is used in this project owing to the low effort. Precisely, this positioning is based on a method called *fingerprinting*, which is specialized for indoor use. Unlike traditional location services such as GPS, fingerprinting takes characteristic fingerprints at selected locations and stores them into a database before running a location determination. During positioning you take a fingerprint at the unknown location and compare it to the fingerprints in the database, returning the fingerprint associated to a known location that matches best.

The aim of this thesis is to develop a location determination system that can locate a target on the first floor of the Kantonsschule Seetal. As a target we use a Raspberry Pi, which is a credit-

¹ Han, et al. 2009, p. 1

card-sized Linux computer. The Raspberry Pi is the transmitter that is attached to the object we want to find (comparable to a GPS transmitter). In the course of this thesis a few crucial questions need to be answered. What hard- and software is required and also affordable? Is there a need for an additional infrastructure or can an existing one be used? Which form of signal transmission is best suited? How do transmitter and receiver communicate and how should the user interface look like? Which allocation method combined with a mathematical algorithm to calculate the position achieves the highest accuracy?

In recent years many research groups around have been developing new indoor positioning systems. The first system that was based on Wi-Fi fingerprinting was the RADAR system by Microsoft (2000). It achieves an accuracy of 2-3 meters and uses a deterministic algorithm to find out the position.² A company called Ekahau has even developed a complete product for sale. The Ekahau Real Time Location System is a combination of active RFID technology and WLAN fingerprinting. Assets or people that should be tracked are marked with a Wi-Fi tag. This product is already used in healthcare and education.³ The most significant work for this thesis was done by Moustafa Youssef with his dissertation "*Horus: A WLAN based indoor location determination system*". This system is based on a probabilistic algorithm, the Horus algorithm, which has proven to be one of the most accurate ones.⁴ Therefore, the Horus algorithm will be used to perform the positioning in this project. Additionally, the insight about WLAN fingerprinting and its implementation provided by this dissertation is invaluable to my matura thesis.

In *Chapter 2* the basic knowledge about the topics related to this project will be explained. *Chapter 3* is about the setup of hard- and software. It describes all fundamental requirements to develop and run the positioning system. The focus of *Chapter 4* is the issue of reading network information from the wireless adapter and the problems that occurred. In *Chapter 5*, which is the main part of this thesis, every single part of the positioning system and how they work together will be described. *Chapter 6* presents the most important results of the project. *Chapter 7* discusses the results and evaluates the project also analyzing possible improvements. This chapter also reflects upon the problems during the thesis and how they were solved. *Chapter 8* is a short conclusion and shows some future prospects.

² Han, et al. 2009, p. 1

³ Ekahau 2014

⁴ Youssef 2004, p. 19



+Add Struktur

Maturaarbeit



Krispin - 2014-12-31 06:57:01.0

SchulApp

Eine Android- App für die Verwaltung und Gestaltung von Kursen

*Kantonsschule Olten
Maturaarbeit
05.01.2015
von Krispin Wandel 4aM
Betreuungsperson:
Claude Vonlanthen*

Vorwort

Ich habe mich für das Programmieren einer Schul-App deshalb entschieden, da ich mir so noch weitere Programmierkenntnisse aneignen konnte. Zudem hatte ich ständig den Wunsch, nicht immer meine dicken Bücher in die Schule mitnehmen zu müssen. Auch wollte ich einen Weg finden, wie ich immer einen geordneten Überblick über meine Schulnoten oder Kopien erhalte. Auch war stets das Anliegen vorhanden, den Stoff nicht verpassen zu müssen, wenn man einmal krank war. Doch auch die Vorstellung, dass Kinder, die im Moment keine Bildungsinstitution geniessen können, durch meine App eines Tages die Möglichkeit hätten, an Kursen teilnehmen zu können, die ganz über das Internet laufen würden, war mir ein grosser Antrieb.

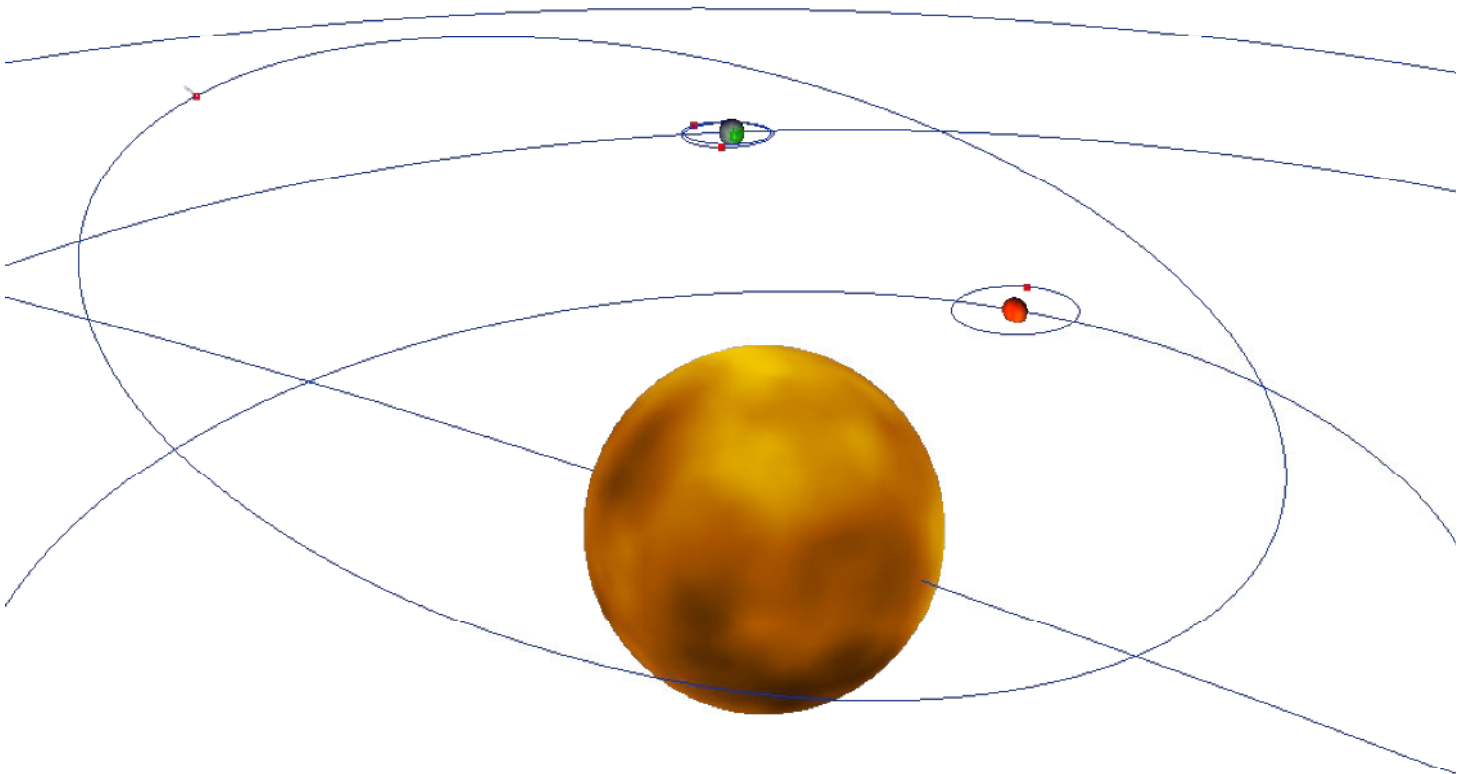
Bedanken möchte ich mich hiermit auch bei Nils Wandel, der mir eine sehr grosse Hilfe war.

Abstract

Meine Aufgabe bestand darin, eine Android App zu entwickeln, die eine Verbindung zum Server aufbauen kann, um mit diesem zu kommunizieren und fähig ist, Strukturen bzw. Kurse zu erstellen. Einen Messenger einzubauen war Zusatz. Das grosse Risiko bei dem Projekt war für mich, dass ich nicht einschätzen konnte, wie umfangreich bzw. wie viel Zeit es im Endeffekt benötigen würde, diese App mit den entsprechenden Anforderungen fertigzustellen. Die grösste Herausforderung lag darin, sich eine sinnvolle Struktur für die App zu überlegen.

Computersimulation eines zufällig generierten Systems
aus Massenpunkten programmiert in Java

GRAVITY



Autor: Martin Zumsteg

Betreuungsperson: Christian Datzko

Co-Referent: Matthias Liebendörfer

Vorwort

Von Anfang an war die Möglichkeit, für meine Maturaarbeit ein Programm zu schreiben eine meiner Favoriten. Zu der Zeit hatte ich bereits eine grosse Erfahrung mit Java, weshalb ich diese Sprache gewählt hatte. Ich hatte auch gerade begonnen C zu lernen, habe mich aber dagegen entschieden meine Arbeit darin zu schreiben. Ich bedaure diese Entscheidung nicht, aber vielleicht hätte ich mich ein Jahr später anders entschieden. Danach war es nicht schwer, ein Thema zu finden. Im Physikunterricht hatten wir gerade das Thema der Schwerkraft beendet, und ich fand es schade, dass wir dieses Thema nur theoretisch und als Ergänzung zu den Kreiselkräften behandelten. Dazu kam, dass ich in der Zeit sowieso vor hatte, ein Programm auf diesem Gebiet zu schreiben. Deswegen war ich mit den Möglichkeiten bereits vertraut und konnte einschätzen, dass dies eine spannende Arbeit werden würde.

Ich hatte aber auch einige andere Ideen, unter anderen die folgenden:

- Ein Programm welches chemische Verbindungen darstellen kann und gewisse Operationen mit diesen ausführen (wie z.B. die Verbrennungsreaktion oder eine Säure-Base-Reaktion). Ich wählte diese Option nicht, weil ich mich mehr für Physik interessiere und weil ich nicht so viel für eine Matur angemessenes hätte herausholen können.
- Ein 3D-Animiertes Spiel zu schreiben. Das war eigentlich gar keine richtige Idee, weil ich schon anfangs wusste, dass meine Modelle und die Geräuschkulisse, die ich ebenfalls hätte erstellen müssen, nicht wirklich gut herausgekommen wären. Es war mehr etwas, das ich gerne getan hätte, aber wusste, dass es nicht realistisch ist.

Die Wahl der Einheiten im Programm war auch eine wichtige Frage. Der Einfachheit halber stimmen diese mit denen in der Realität überein. Das bedeutet, dass die Physik korrekt beachtet wird, jedoch musste ich aufgrund der Tatsache, dass niemand gerne das Programm eine Jahr am Stück beobachtet, die Verhältnisse anders wählen. Also, wenn eines der Systeme in der Realität existieren würde, würde es sich sehr ähnlich verhalten, aber es würde nicht existieren. Das perfekte Beispiel ist, dass die standardmässig generierte Singularität etwa die Masse unserer eigenen Sonne hat, ein Stern ist eine Million Mal leichter und die Planeten nochmals etwa eine Million.

3k

Entwicklung eines Computerspiels

Maturarbeit von Julian Dunskus, 6e



Kantonsschule Hohe Promenade, Gymnasium, Zürich

Schuljahr 2014/15

Referent: Stefan Müller

Korreferent: Michael Liebich

1 Anfang

1.1 Motivation

Schon als ich 9 Jahre alt war, zeigte mein Vater mir "Scratch" (www.scratch.mit.edu), eine auf Anfänger im Programmieren ausgelegte Plattform, um einfache Anwendungen und Spiele zu entwickeln, indem man farbige Blöcke zusammenzieht, um so den Code visuell aufzubauen. Ich war von Scratch begeistert, da ich endlich selber das machen konnte, was ich schon lange bewundert hatte. Später kaufte er mir ein Buch, "Java mit Eclipse für Kids" von Hans-Georg Schumann, was die Grundlagen von Java kinderfreundlich erklärt. Durch dieses Buch lernte ich die Syntax von Java kennen, hörte aber wieder auf mit Java, da ich Scratch zugänglicher und ansprechender fand.

Im September 2010 zeigte mir ein Freund "Minecraft", ein inzwischen sehr bekanntes Spiel, was einem eine Welt aus 1x1x1m-Blöcken bereitstellt, in der man gegen Monster kämpfen, Bäume fällen, Stein minen, Höhlen erforschen und alles erdenkliche bauen kann. Minecraft wurde mein absolutes Lieblingsspiel. Das wichtigste an dieser ganzen Geschichte ist aber, dass Minecraft in Java programmiert ist, und dass man, wenn man das Spiel mit einem Werkzeug decompiliert hat, den Code bearbeiten kann. Dadurch kann man neue Blöcke hinzufügen, verrückte Sachen mit der Mechanik anstellen und alles ändern. Von diesen Möglichkeiten hingerissen tauchte ich in den Code ein und konnte mein durch Scratch und das Buch angeeignete Wissen anwenden. Ich schaute auf YouTube Anleitungen, die mir beibrachten, wie ich bestimmte Sachen tun konnte, zum Beispiel einen neuen Block hinzuzufügen. Durch diese Auseinandersetzung mit dem Code erlernte ich die Mehrheit meines heutigen Programmierkönnens. Schliesslich fing ich parallel dazu an, auch eigene Spiele und Anwendungen in Java zu programmieren.

1.2 Themenwahl

Bei den Präsentationen der Maturarbeiten wurde mir schnell klar, dass ich als Maturarbeit etwas programmieren wollte. Mir war aber noch nicht klar, ob ich ein Spiel oder eine sonstige Anwendung entwickeln wollte, bis ich während einer Mathestunde die Idee zu 3k hatte. Ich zeichnete dann gleich eine Skizze vom fertigen Spiel. Als ich wieder zuhause war, setzte ich diese Skizze dann gleich in Photoshop um (Abbildung 1). Ich schickte dann eine Anfrage an meinen damaligen Chemielehrer, Michael Liebich, der aber schon besetzt war und mich nicht noch dazunehmen konnte. Er leitete die Anfrage aber für mich weiter an zwei Mathematiklehrer, von denen Stefan Müller sich meiner Arbeit annahm. Nach unserer ersten Besprechung machte ich mich gleich an die Arbeit und kreierte die Dateien, die ich für dieses Spiel brauchen würde.



Darstellung von Fraktalen auf dem Computer

Joel Schmitz

9. Februar 2015

Betreuer: Karlheinz Briz

Abschlussarbeit

Atelierschule Zürich

Klasse 12c

Einleitung

Fragestellung: *Wie können Fraktale¹ visuell 2D und 3D als auch musikalisch dargestellt werden?*

In dieser Arbeit geht es um die Vertiefung und Erkundung des Gebiets der Fraktale. Einerseits ist dies ein Bereich der Mathematik, welches die Einarbeitung in iterierte Funktionen, komplexe Zahlen und Quaternionen verlangt. Andererseits braucht es zur Darstellung auch gute Kenntnisse im Programmieren. Bei der Einarbeitung in die noch unbekanntem mathematischen Aspekte des Themas haben mir unter Anderen die zwei Bücher *Hypercomplex Iterations: Distance Estimation and Higher Dimensional Fractals* [1] und *The Science of Fractal Images* [2] geholfen.

Eine grosse Motivation bildete für mich vor allem die visuelle Ästhetik und ungeahnte Vielfalt der fraktalen Muster. Weiterhin inspirierte mich eben diese Ästhetik zur Erkundung der Möglichkeiten der Generierung von Musik mittels Fraktalen.

Da ich mich schon seit einer Weile mit dem Thema Fraktale beschäftige und ich bereits mehrere Möglichkeiten, Fraktale zu visualisieren, ausprobiert habe, wollte ich in dieser Arbeit diese Ansätze vertiefen, um weitere Möglichkeiten zu finden, Fraktale auf effektive Weise abzubilden. Des Weiteren wollte ich versuchen, Fraktale nicht nur bildlich, sondern auch musikalisch darzustellen, d.h. mithilfe von Fraktalen Musik zu erzeugen.

Diese Arbeit ist in 4 Kapitel aufgeteilt. In Kapitel 1 werden kurz Fraktale im Allgemeinen diskutiert und beschrieben. In Kapitel 2 wird die Darstellung von Fraktalen in der komplexen Zahlenebene betrachtet, also Fraktale im zweidimensionalen Raum, wobei ein besonderer Fokus auf Mandelbrot- und Julia-Mengen gelegt wird. Auch werden hier die verschiedenen Möglichkeiten, Fraktale mit möglichst geringem Rechenaufwand ästhetisch schön darzustellen, erkundet. In Kapitel 3 wird zunächst eine kurze Einführung in die allgemeine Wiedergabe von 3D-Szenen gegeben. Anschliessend geht es um die verschiedenen Methoden der Darstellung von Fraktalen im dreidimensionalen Raum und um 3D-Schnitte durch im Vierdimensionalen generierte Fraktale. Dabei wird eine Verallgemeinerung der Julia-Menge aus Kapitel 2 mithilfe von Quaternionen erreicht. Weiterhin werden kaleidoskopische IFS Fraktale beschrieben. In Kapitel 4 wird versucht mit verschiedenen fraktalen Methoden interessante Musik zu erzeugen, namentlich mit Lindenmeyer-Systemen, Perlin Noise und IFS Fraktalen. Dieses Kapitel ist unabhängig von den anderen, sodass auch gleich in dieses Kapitel eingestiegen werden kann.

¹Geometrische Objekte die interessante Formen bilden. Für eine ausführliche Erklärung des Begriffs siehe Kapitel 1

simulup

Entwicklung einer weltweiten
Projektmanagement Plattform

Kantonsschule Frauenfeld
Patrick Egli
2014

Maturaarbeit im Fach Informatik
Abgabetermin: 20.10.2014
Betreuer: Matthias Bolli

1 Einleitung

1.1 Werdegang als Programmierer

Vor ca. 10 Jahren erzählte mir mein Onkel Robert Ott von seinem Berufsleben als Informatiker. Mit diesem Thema konnte er mich sehr faszinieren und ich wollte gleich darauf ebenfalls Programmieren lernen. Mein Onkel kaufte mir ein erstes Buch „Spiele programmieren mit Flash MX: 12 ActionScript-Workshops“. Ich begann sofort das Buch zu lesen, allerdings war das Verstehen dieses Buches sehr schwierig zu diesem Zeitpunkt, weil ich noch sehr jung war. Nach ungefähr einem weiteren Jahr kaufte ich ein neues Buch „C++ für Spieleprogrammierer“. Als ich dieses Buch gelesen habe, hatte ich eine erste Grundlage im Bereich der Programmierung geschaffen. In den folgenden Jahren habe ich mich mit weiteren Programmier- und Skriptsprachen beschäftigt, C#, PHP, Java, JavaScript, MySQL, CSS. Ich programmiere nun leidenschaftlich seit 10 Jahren. Während dieser Zeit habe ich sehr viele Erfahrungen gesammelt.

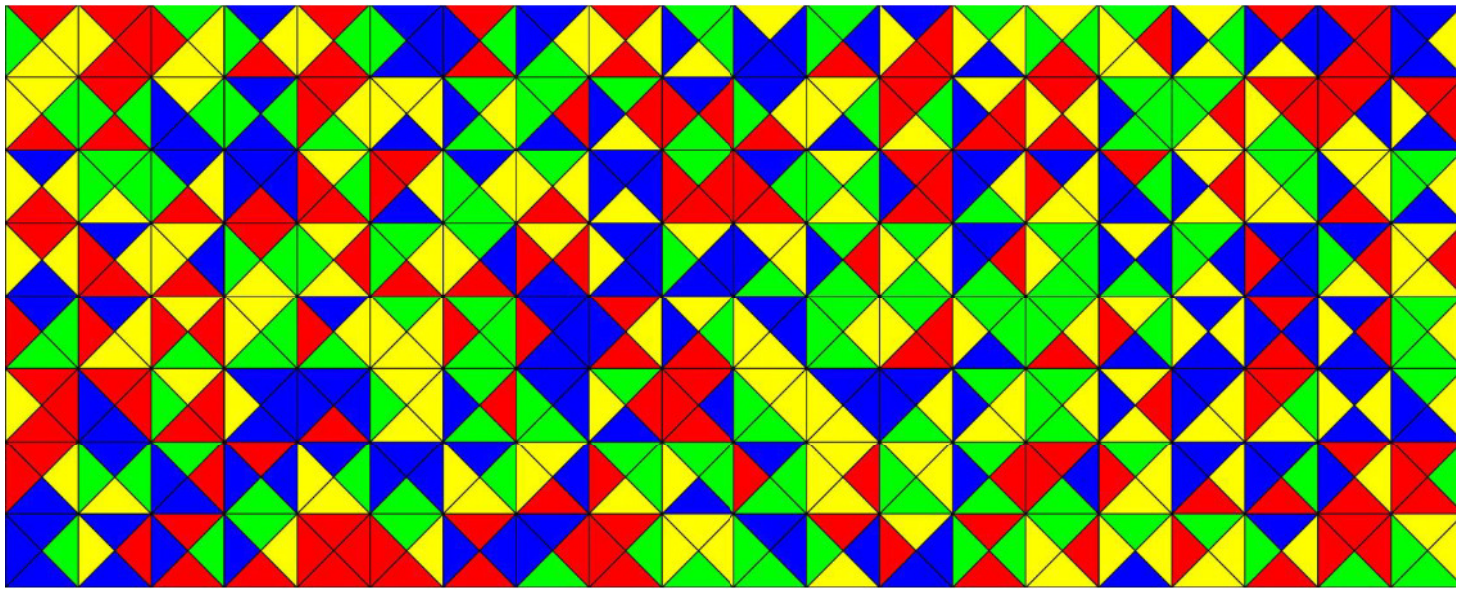
1.2 Motivation und Vision

Vor ungefähr zwei Jahren kam ich auf die Idee eine Webplattform zu programmieren, bei welcher Gruppen von Benutzern Daten teilen können. Zu diesem Zeitpunkt nannte ich die Plattform „GroupShare“. Allerdings musste ich feststellen, dass die Domain GroupShare.com bereits reserviert war. Nach langem Nachdenken kam ich auf den Namen SimulUp. Dieser Name besteht aus zwei Teilen. Der erste Teil dieses Namens „Simul“ bedeutet auf lateinisch gemeinsam. Der zweite Teil „Up“ ist das Präfix für das englische Wort update, auf Deutsch aktualisieren.

Als ich nach längerer Entwicklungszeit festgestellt habe, dass Filesharing Plattformen oftmals mit Gerichtsprozessen verwickelt sind, habe ich den Inhalt der Plattform etwas angepasst. Ich kam auf die Idee eine Plattform zu entwickeln, bei der Gruppen von Benutzern Projekte planen können.

Viele Projekte werden ohne saubere Planung durchgeführt. Für viele Projektverantwortliche ist das Aufsetzen einer Projektplanung zu aufwendig. Mit grosser Wahrscheinlichkeit hätten solche Projekte, welche ohne Planung durchgeführt worden sind, wesentlich besser resultiert. Mit SimulUp biete ich eine Webplattform, welche dieses Problem möglicherweise bzw. teilweise lösen könnte. SimulUp ist besonders für kleine bis mittelgrosse Projekte geeignet.

Als Programmierer habe ich den grossen Wunsch eine Plattform zu entwickeln, welche anderen Menschen helfen kann. Es macht mich glücklich, wenn ich anderen Menschen das Leben vereinfachen kann. Zu sehen, wie Benutzer meine Plattform benutzen ist ein gutes Gefühl.

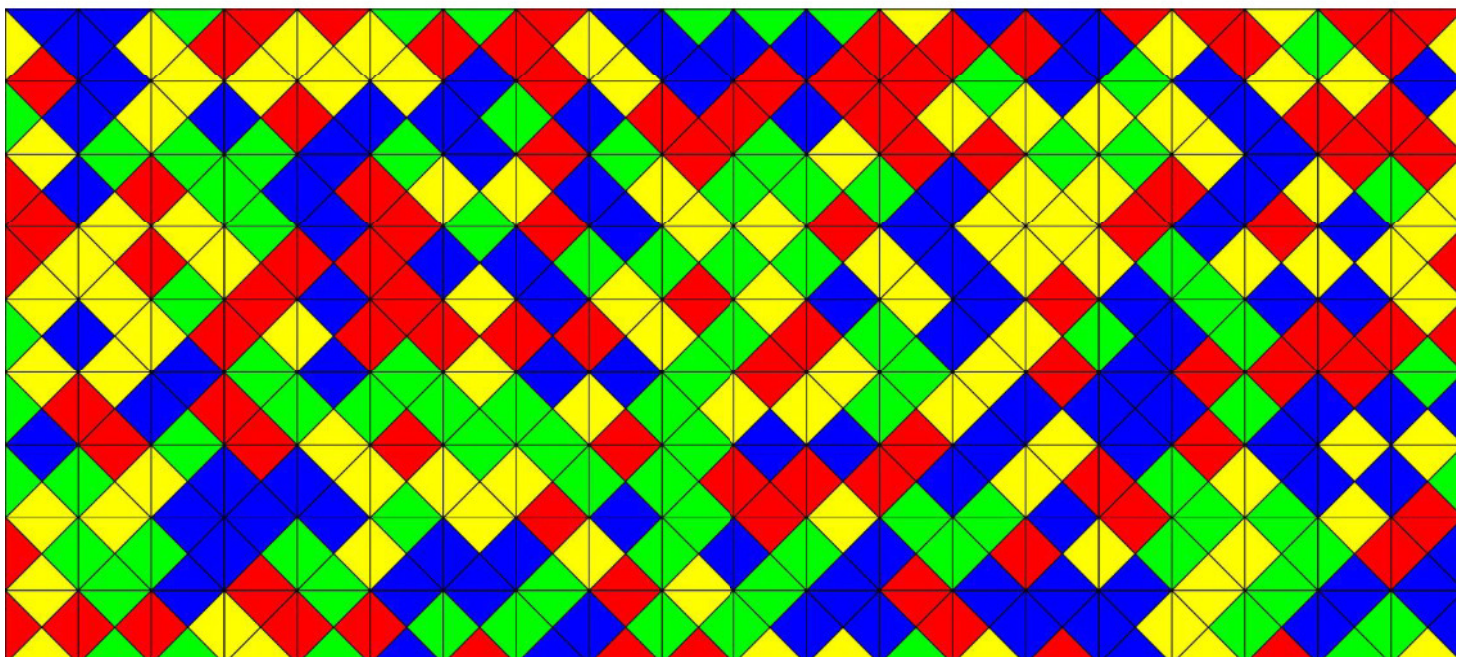


Simulated Annealing

Programmierung und Analyse eines numerischen Optimierungsverfahrens am Beispiel quadratischer Dominosteine

Maturand: Noah Fehr, Kantonsschule Limmattal, Herbst/ Winter 2014

Betreuer: Harald Pierhöfer



1 Einleitung

1.1 Ausgangslage¹

Betrachten wir ein einfaches kombinatorisches Puzzlespiel: Wir haben verschiedene quadratische Dominosteine zur Verfügung, die in einer Holzkiste auf 8×8 Positionen gesetzt werden dürfen. Anfangs mögen diese Steine wie in Abb. 1 gezeigt angeordnet sein. Wie dort zu sehen ist, haben die Steine vier Seiten mit den Farben Gelb, Grün, Blau und Rot.

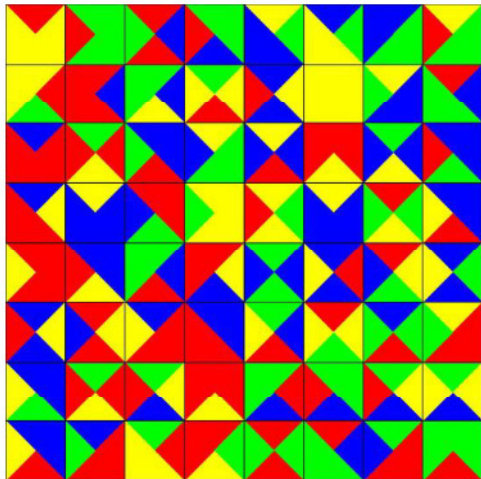


Abbildung 1 Ausgangslage des Dominospiels: die Steine liegen zufällig in der Kiste

Bei diesem Puzzle dürfen wir die Steine beliebig umsetzen und drehen. Unser Ziel ist es, möglichst viele gleichfarbige benachbarte Flächen zu erhalten: Für jedes solcher Paare erhalten wir einen Punkt. Es gibt genau $2n(n - 1)$ benachbarte Flächen (wobei n die Anzahl der Blöcke pro Reihe beim quadratisch aufgebauten Spielfeld ist), so dass wir in diesem Beispiel auf keinen Fall mehr als 112 Punkte erreichen können.

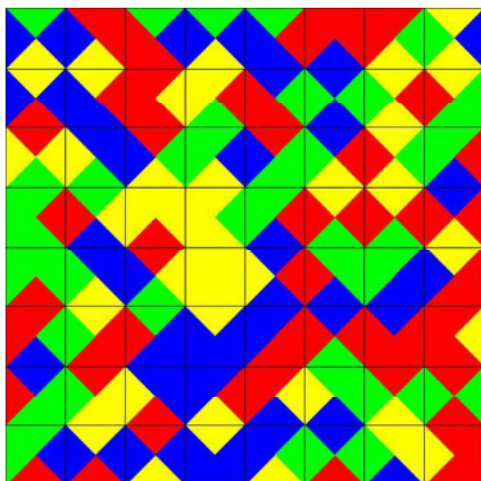


Abbildung 2 Lösung des Dominospiels: wir erreichen 112 Paare

Die Lösung dieses Dominospiels ist nur für relativ kleine Eingaben n und beschränkter Anzahl

¹Adaptiert nach Taschenbuch der Algorithmen, Kapitel 43, Peter Rossmanith (RWTH Aachen)

Farben mit kombinatorischen Methoden noch exakt lösbar. Ein Verfahren, das geeignet ist, eine gute Lösung für viele solcher Fälle, d.h. wo die Lösung nicht mehr durch Probieren aller Möglichkeiten und Backtracking gefunden werden kann, ist *Simulated Annealing*.

1.2 Definitionen

1.2.1 Wikipedia

Simulated Annealing (simulierte Abkühlung) ist ein heuristisches² Optimierungsverfahren. Das Verfahren wird zum Auffinden einer approximativen Lösung von Optimierungsproblemen eingesetzt, die durch ihre hohe Komplexität das vollständige Ausprobieren aller Möglichkeiten und einfache mathematische Verfahren ausschließen. Grundidee ist die Nachbildung eines Abkühlungsprozesses, etwa beim Glühen in der Werkstoffkunde. Nach Erhitzen eines Metalls sorgt die langsame Abkühlung dafür, dass die Atome ausreichend Zeit haben, sich zu ordnen und stabile Kristalle zu bilden. Dadurch wird ein energieärmer Zustand nahe am Optimum erreicht.

Übertragen auf das Optimierungsverfahren entspricht die Temperatur einer Wahrscheinlichkeit, mit der sich ein Zwischenergebnis der Optimierung auch verschlechtern darf. Der Metropolis Algorithmus (s. unten) ist die Grundlage der simulierten Abkühlung. Im Gegensatz zu einem Lokale-Suche-Algorithmus kann das Verfahren ein lokales Optimum wieder verlassen und ein besseres finden.

Das Problem des simulierten Ausglühens kann man sich graphisch verdeutlichen: Angenommen, man sucht in einer zweidimensionalen Landschaft den (global) tiefsten Punkt. Die Landschaft selbst besteht aus vielen unterschiedlich tiefen Dellen. Die einfache Suchstrategie (suche den nächsten tiefsten Punkt) entspricht dem Verhalten einer Kugel, welche in dieser Landschaft ausgesetzt wird. Sie rollt zum nächsten lokalen Minimum und bleibt dort. Bei der simulierten Abkühlung wird der Kugel immer wieder ein Stoß versetzt, der mit zunehmender Abkühlung jedoch schwächer wird. Dieser ist idealerweise stark genug, um die Kugel aus einer flachen Delle (lokales Minimum) zu entfernen, reicht aber nicht aus, um aus dem globalen Minimum zu fliehen.

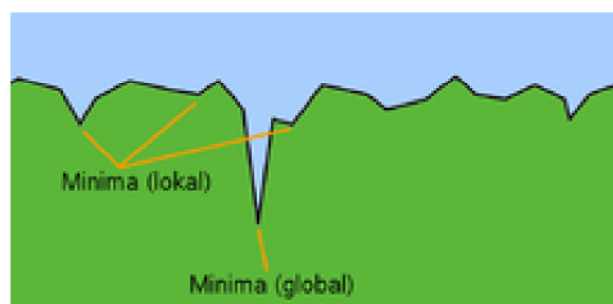
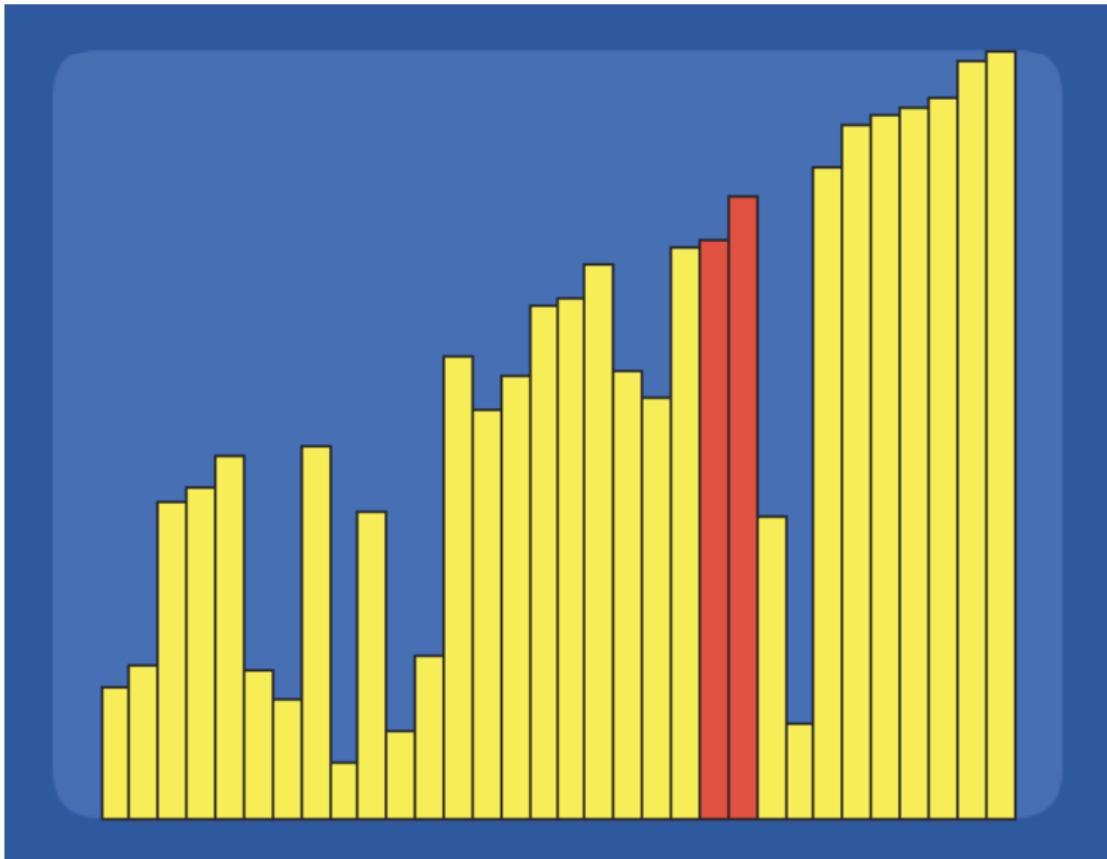


Abbildung 3 Graphische Darstellung einer Landschaft, in der ein globales Minimum gefunden werden soll

²Heuristik bezeichnet die Kunst, mit begrenztem Wissen, d.h. bei unvollständigen Informationen und wenig Zeit zu guten Lösungen zu kommen (G. Gigerenzer und P. M. Todd mit der ABC Research Group: Simple heuristics that make us smart. Oxford University Press, New York 1999)

Graphische Darstellung von Sortieralgorithmen



Joris Diem

Kantonsschule Limmattal
Schuljahr 2014/15

Betreuer: H. Pierhöfer

1 Einleitung

Durch das Computerspiel „Minecraft“¹ habe ich erste Erfahrungen mit dem Programmieren gemacht. Dabei habe ich gemerkt, dass es mich interessierte und ich mich gerne weiter mit diesem Thema auseinandersetzen wollte. Darum gab mir mein Vater ein Dossier über die Einführung in „Processing“², eine Programmiersprache. Dies hat mir so gut gefallen, dass ich angefangen habe, winzige Programme und Minispiele zu schreiben. Dieses Interesse ist geblieben, bis ich ein Thema für die Maturitätsarbeit auswählen musste, denn ich wollte immer noch etwas mit Informatik machen. In einem Gespräch mit meinem Betreuer haben wir eine Idee gefunden: Das Ziel meiner Arbeit soll sein, Sortieralgorithmen graphisch darzustellen, also die einzelnen Schritte der Sortiervorgänge aufzuzeigen.

In der folgenden Arbeit erkläre ich, was ein Sortieralgorithmus ist und auszeichnet, gebe Beispiele dazu, zeige wie mein Programm aufgebaut ist und wie ich dabei vorgegangen bin.

¹ Siehe <https://minecraft.net>, 03.01.15

² Siehe <https://processing.org>, 03.01.15

ANDROID APPLIKATION ZUR PERIODIZITÄT DER ELEMENTE

Erstellung einer Android Applikation zur Förderung des Verständnisses und Vertrautheit mit den periodischen Eigenschaften des Periodensystems der Elemente

Eine Maturitätsarbeit von

Marco Job, Bonstetten

unter Betreuung von

Dr. Harald Piehörer, Uster

Kantonsschule Limmattal

In der Luberzen 39

8902 Urdorf

Frühlingssemester 2013/2014 und Herbstsemester 2014/2015

Stichworte: Android, Applikation, Periodizität, Periodensystem, Elemente

Zusammenfassung

Diese Maturitätsarbeit zeigt die Konzipierung, die Programmierung und das Design einer Android-Applikation zur Periodizität der Elemente. Periodische Eigenschaften sind Eigenschaften, die regelmäßig und wiederkehrend sind. Sie sind deshalb vorhersehbar und abschätzbar. Diese Eigenschaft wurde sich in dieser Applikation zunutze gemacht, um ein Spielprinzip zu schaffen. Der Nutzer soll einen Zahlenwert zu einer periodischen Eigenschaft in einem Periodensystem der Elemente zuordnen. Durch den resultierenden Lerneffekt soll der Nutzer die periodischen Eigenschaften Ionisierungsenergie, Elektronenaffinität, Elektronegativität, Atomradius abschätzen können. Um anfänglich eine Basis zum Verständnis der chemischen Theorie zu schaffen, ist ein Theorie-teil zu allen Eigenschaften verfügbar. Alle Zahlenwerte, sowie ein Periodensystem mit Elementen kann dargestellt werden. Die Applikation ist vollständig auf Englisch und Deutsch verfügbar, um möglichst viele Nutzer anzusprechen.

Die Applikation wurde mit Processing programmiert. Processing ist eine objektorientierte Programmiersprache ähnlich Java.

In einem ersten Schritt wurde die chemische Theorie erarbeitet, um das Konzept hinter der Periodizität zu verstehen. Mit dem gewonnenen Wissen wurde im zweiten Teil die Applikation programmiert. Die wichtigsten Funktionen der Applikation sind die Spielmodi zu den genannten periodischen Eigenschaften, der Theorie-teil, das interaktive Periodensystem mit allen periodischen Zahlenwerten und die Statistik. Der letzte Schritt war es ein einheitliches, thematisch passendes Design zu erstellen und auf die gesamte Applikation anzuwenden. Die Grundform des Periodensystems wurde als Grundlage für das Design genommen und streckt sich durch alle Funktionen. Dieses Prinzip schafft einen fließenden Umgang mit der Applikation.

Nach Beendigung der Programmierung musste die Applikation so bearbeitet werden, damit aus dem Quelltext (*Text der Anwendung in der Programmiersprache*) eine signierte APK-Datei (von engl. *Android Application Package*) entsteht. Nur signierte APK-Dateien werden vom Google Play Store (*Plattform auf welcher Android-Applikationen angeboten werden*) akzeptiert und können dann veröffentlicht werden. Schlussendlich wurde ein Anzeigebild erstellt, um die Applikation auf der Android-Plattform und im Google Playstore zu repräsentieren. Dem Eintrag im Google Play Store wurde eine kurze Beschreibung und Bilder hinzugefügt, um das Prinzip der Anwendung zu wiedergeben. Die Anwendung ist unter dem Namen *Elementum* für Android-Plattformen verfügbar.

Maturitätsarbeit 2015

René Zurbrugg

C6c, KZO Wetzikon

Betreuende Lehrperson:

Albert Kern



Skatepedia 2014

1 Einleitung

Im Rahmen meiner Maturarbeit habe ich eine App programmiert. Der hier vorliegende schriftliche Teil befasst sich zuerst mit technischen Aspekten (Wie schreibe ich eine App? Was ist eine Activity?) und beschreibt danach die Applikation selbst (Wie ist sie aufgebaut? Wie publiziert man eine App?).

Den leicht modifizierten Quellcode der gesamten Applikation finden Sie im Anhang.

Die schriftliche Arbeit richtet sich bewusst an technisch interessierte Leser, sollte jedoch besonders im zweiten Teil für andere ebenfalls interessant sein. Falls Ihnen ein technischer Begriff unbekannt sein sollte, befindet sich am Ende der Arbeit ein Glossar mit einer kurzen Definition.

1.1 Motivation

Skaten ist schon seit meiner Kindheit eines meiner grössten Hobbies und ich engagiere mich sehr für diese Sportart.

Seit vier Jahren bin ich deshalb bemüht, einen Skatepark in Wetzikon aufzubauen - bis jetzt leider noch mit mässigem Erfolg, obwohl wir vor mehreren Jahren erfolgreich eine Petition mit 1000 Unterschriften übergeben haben. Daneben bin ich Präsident des Vereins

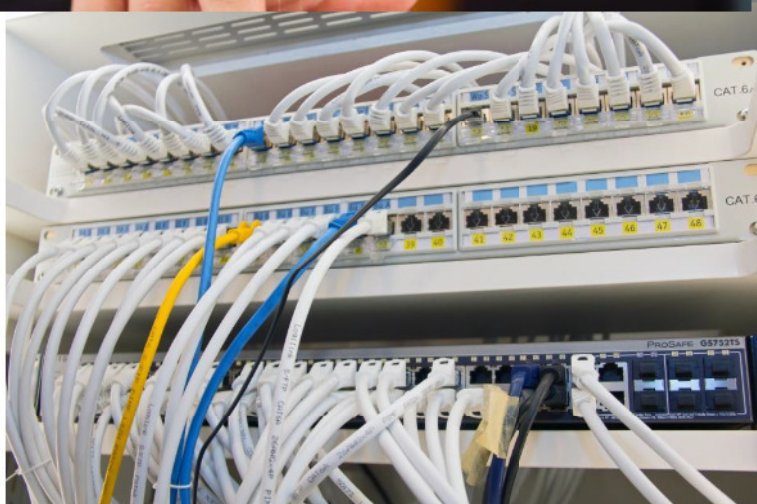
„Skatepark Wetzikon“. Die Homepage unseres Vereines (www.skateparkwetzikon.ch) kam mir sehr gelegen, da ich für meine Arbeit einen Server benötigte. So laufen sämtliche Serveranfragen (Kommentarfunktion, Bewertungsfunktion etc.) über diesen Server.

Meine Faszination für Informatik habe ich schon seit langem. Mit 15 Jahren machte ich erste Erfahrungen mit dem Programmieren. Zuerst während zwei Monaten in C, danach wechselte ich jedoch zu Java. Mit Android hatte ich bereits etwas Erfahrung, jedoch waren meine Fähigkeiten sehr begrenzt. Ausserdem arbeite ich an Wochenenden und/oder in den Sommerferien gelegentlich bei einer Informatikfirma.

Als Thema meiner Maturarbeit kam mir die Idee, diese beiden Hobbys zu verbinden. Zuerst wollte ich ein Skatelehrbuch mit Bildern machen, danach eine schriftliche Arbeit zum Thema Sicherheit und Hacking. Das eine war mir zu praktisch, das andere zu theoretisch. Deshalb entschied ich mich letztlich für eine Mischung aus Informatik und Skaten. Die Idee von Skatepedia war geboren.



Intelligentes Wohnen Heimvernetzung



Verfasser:
Yannick Huber
Klasse 4f

Betreuende Lehrperson:
Martin Friedli-Hofmann

Abgabetermin:
5. Januar 2015

Hausautomation, Smart Home, intelligentes Wohnen, integrierte Heimvernetzung – ganz egal, welchen Begriff man verwendet – es sind Schlagworte für Bequemlichkeit, Wohnkomfort, Strom- und Temperaturüberwachung, Multimediagenuss und vieles mehr. Zukunftsvisionen, die immer näher an die Gegenwart heranrücken und plötzlich realistisch sind.

Die vorliegende Maturitätsarbeit beschäftigt sich mit folgenden Fragen: Wie muss ein Netzwerk in bestehenden Häusern auf- oder ausgebaut werden, damit eine intelligente Haussteuerung möglich ist? Wie wird dem Medienkonsum des 21. Jahrhunderts Rechnung getragen? Wo liegen die Probleme des „Plug and Play“ und der Kommunikation zwischen den einzelnen Geräten? Wie kann man Abläufe im Haushalt automatisieren? Wie kann ein Haus aus der Ferne überwacht werden?

Im ersten Teil dieser Maturitätsarbeit soll diesen Fragen theoretisch nachgegangen werden. Im zweiten Teil zeigt das Fallbeispiel Praxis, wie ein Local Area Network in einem bestehenden Einfamilienhaus eingebaut wird und welche Arbeitsschritte dazu nötig sind.

Die theoretischen und praktischen Erkenntnisse zeigen, dass die Computer- und Netzwerktechnologie sehr weit standardisiert ist. Die Geräte sind ausgereift; die Preise angemessen. Sie können auch selber installiert werden.

Im Gegensatz dazu der Bereich der Hausautomation, der aufgrund der kleinen Hersteller und deren Produkte sehr unübersichtlich ist. Die zahlreichen Standards sind noch nicht in der Breite etabliert. Ein Eigenbau kann nur mit breitem Wissen, grossem zeitlichen Aufwand und entsprechenden Finanzen erfolgreich umgesetzt werden.

Zurzeit bieten Komplettlösungen, installiert durch einen Fachmann, die besseren Systeme. Langfristig haben Produkte beim Schweizer Kunden nur Erfolg, wenn sie einwandfrei arbeiten, universell kompatibel und preislich realistisch sind. Ebenso müssen sie punkto Sicherheit und Privatsphäre absolute Transparenz bieten.

Kantonsschule Zug – 2014-2015

Artificial life

Der Versuch, einfaches Verhalten von Lebewesen zu simulieren

Die Maturaarbeit von Irvin Hostettler

Betreut von Mohamed Kubba-von Jüchen

1. Einleitung

«Wenn es Götter gäbe, wie hielte ich's aus, kein Gott zu sein!» Friedrich Nietzsche
in «Also sprach Zarathustra»

Die Erschaffung von Leben, welche eine Vision! Eine Thematik, angesiedelt zwischen der eigenen Identitätsfrage und der Obskurität unserer Umwelt. Die Vision provoziert Fragen nach der Grenze unserer Schaffenskraft. Erschaffen-Können setzt auch immer ein tiefes Verständnis voraus. Der Wunsch, selber Leben erschaffen zu können, situiert sich hiermit als eine substantielle Schnittmenge menschlicher Existenz.

So ist es nicht weiter verwunderlich, dass künstliches Leben eine grosse Faszination auf die Menschen ausübt. Immer wieder fand die Thematik Eingang in Sagen, Legenden und Geschichten. Ein literarisches Denkmal setzte Mary Shelleys dem künstlichen Leben im Roman Frankenstein (1818).

Seit der Erfindung des Computers verfügen wir nun über ein Werkzeug, welches es uns ermöglichen könnte, künstliches Leben auch als Intelligenz zu erschaffen. So wurde aus reiner Fiktion ein naturwissenschaftlicher Forschungsbereich.

Während der Ideenfindung für eine Maturaarbeit begann ich mich schnell für diesen Bereich der Wissenschaften zu interessieren. Die Idee, dass aus statischen Rechenregeln organisches Verhalten entstehen kann, faszinierte mich. So kam auch bald die Idee, ein Programm zu entwickeln, welches gewisse Eigenschaften eines Lebewesens simuliert.

Diese Arbeit soll einen Überblick über meine Maturaarbeit geben. Sie soll über Entstehung, Aufbau, Funktion und Ergebnis meiner Simulation berichten und sich auch mit der Realitätsnähe meines Projekts befassen. Ich werde dabei, wenn immer möglich, auf technische Details verzichten. Technisch Interessierten sei insbesondere der Anhang A.2 ans Herz gelegt.

Inhaltlich ist diese Maturaarbeit durch mehrere Absätze strukturiert. Zunächst werden in einem Abschnitt Aufbau, Funktion und Ergebnis der Simulation kurz umrissen. Der nächste Abschnitt befasst sich detailliert mit den grundlegenden Konzepten und dem Aufbau der Simulation. Danach wird die Oberfläche bzw. deren Bedienung erläutert. Ein weiterer Abschnitt behandelt die Ergebnisanalyse und die Frage betreffend des Realitätsbezugs. Nach der Simulation werde ich deren Entstehungsprozess reflektieren, die grössten Probleme der Entwicklung darstellen und einen Ausblick auf mögliche Verbesserungen und Erweiterungen des Programms geben. Ein Fazit rundet die Arbeit und deren Beobachtungen ab.

Ich möchte an dieser Stelle einen grossen Dank an meine Betreuungslehrperson, Herrn Mohamed Kubba-von Jüchen, aussprechen. Vielen herzlichen Dank für die Unterstützung und Führung in fachlichen wie auch organisatorischen Belangen!

Rubik's Cube

-

Bau einer Lösungsmaschine

Dokumentation



Betreuer:

Herr L. Turi, dipl. math.

Autoren:

Alex Huber, 6h
Daniel Meier, 6h

1. Zusammenfassung

In dieser Dokumentation ist die Entstehung dieser Maturitätsarbeit beschrieben. Es wird auf den Bau der Lösungsmaschine für den Rubik's Cube, das Lösungsprogramm, die Verbindung zwischen dem Computer und der Maschine und verschiedene Lösungsmethoden für den Rubik's Cube eingegangen. Zusätzlich werden auch noch die Schwierigkeiten und Probleme der automatischen Farberkennung aufgezeigt. Alle in dieser Arbeit erwähnten Programme und Quelltexte können unter folgendem Link eingesehen werden:

« <https://www.dropbox.com/sh/bmoofm77ce1y303/AAC3-SN7tu3VaXplfmlA9Cgia?dl=0> »

2. Einleitung

2.1. Entstehung dieser Maturitätsarbeit

Der Rubik's Cube ist ein alter Klassiker, den wir (Alex und Daniel) bereits als Primarschüler kannten, aber noch nicht lösen konnten. Der Würfel faszinierte uns und wir kauften, sobald wir den Würfel lösen konnten, verschiedene Variationen des Rubik's Cube. Zu Beginn der fünften Klasse im Gymnasium merkten wir, dass wir die gleiche Faszination für den Würfel empfinden. Wir versuchten uns an schnelleren Lösungsmethoden und kauften uns auch bessere und schneller drehende Würfel (Speedcubes).

Beide hatten schon Videos von Maschinen gesehen, die den Würfel lösen konnten, aber wir hatten keine Ahnung, wie diese funktionierten. Als wir uns für die Maturitätsarbeit einschreiben mussten, kamen wir auf die Idee, eine solche Maschine zu entwickeln und zu programmieren. Anfangs waren wir uns nicht sicher, ob wir überhaupt unser Ziel erreichen können, aber je länger wir uns mit diesem Thema befassten, desto überzeugter wurden wir, dass wir erfolgreich sein können.

2.2. Fragestellung

Das Ziel dieser Maturitätsarbeit ist, eine Lösungsmaschine für den Rubik's Cube zu bauen. Das heisst, dass man einen verdrehten Würfel in eine Maschine stellen kann, die den Würfel so dreht, dass er am Schluss gelöst ist. Wir hatten drei Hauptprobleme zu lösen: Das erste war, ein Programm zu schreiben, das nach Eingabe des verstellten Würfels eine Zugabfolge liefert, die den Würfel löst. Das zweite Problem besteht darin, eine Maschine zu bauen, die den Würfel drehen kann. Das letzte Problem ist die Ansteuerung der Maschine mit einem Computer bzw. die Verbindung zwischen dem Lösungsprogramm und der Maschine. Der Schwerpunkt dieser Arbeit liegt im Erarbeiten eines Computerprogramms, das den Würfel innert nützlicher Frist mit möglichst wenigen Drehungen lösen kann. Der Zustand des Würfels wird manuell eingegeben und je nach Zeit noch automatisiert.

Automatische Erkennung eines Captchas



Maturaarbeit von Dan Kluser
Betreut durch Mohamed Kubba-von Jüchen
Kantonsschule Zug, Schuljahr 2014/2015

1. Vorwort

Informatik hat mich schon immer interessiert. Darum habe ich mich sehr auf die Maturaarbeit gefreut. Es bietet mir die Gelegenheit, etwas zu tun, was mir Spass macht und die Arbeit fliesst sogar noch in mein Maturazeugnis ein.

Schnell war klar für mich, dass ich mich einem eher schwierigen Projekt widmen will - ich wollte eine Herausforderung. So habe ich mich entschieden, ein Programm zu schreiben, welches ein Captcha automatisch erkennen kann.

Diese Maturaarbeit ist für alle Leser. Es ist kein Vorwissen in der Informatik nötig. Ich möchte mich an dieser Stelle bei meiner Betreuungsperson, Herrn Kubba, für die äusserst gute und effiziente Betreuung bedanken.

2. Einführung

Die Abkürzung Captcha steht für „**C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part“. Im Namen ist also gleich das Ziel eines Captchas genannt. Man will mit Captchas automatisiert einen Menschen von einer Maschine unterscheiden können.

Es gibt viele verschiedene Arten von Captchas. Diese Maturaarbeit befasst sich aber nur mit den klassischen Captchas, bei denen ein Text auf einem Bild abgebildet ist und der Nutzer diesen abtippen muss.



Man sieht ein klassisches, visuelles Captcha und ein Textfeld, wo der Nutzer den auf dem Bild abgebildeten Text abtippen muss. (Abbildung 2.1, viacar.ch)

Das Ziel dieser Arbeit ist es, diesen Test mit einem Computerprogramm zu lösen.

SpaceWars

The background of the entire page is a dark space scene featuring a dense field of grey, irregularly shaped asteroids of various sizes. In the upper left, a portion of a galaxy with a bright, glowing core and spiral arms is visible. A small, blue and red spaceship is positioned in the center-right, appearing to fly through the asteroid field.

Entwicklung eines Mehrspieler-
Weltraum-Shooters

Maturaarbeit von

Alexander Türler

Fabian Lyck

Oktober 2014

Gymnasium Neufeld, Bern

Betreut von Markus Jordi

15Mc

2 Abstract

Diese Arbeit beschreibt die Entwicklung eines modernen Mehrspieler-Computerspiels. Es ging nicht darum, ein Spiel mit gewohnter Fülle an Inhalt zu entwickeln. Vielmehr ging es darum grundlegende Prinzipien zu erforschen und diese als Elemente in der Entwicklung eines Computerspiels aufzuzeigen.

Das eigentliche Spiel ist im Weltraum situiert. Der Spieler kontrolliert ein Raumschiff und kann sich damit frei im Raum bewegen. Damit das funktioniert, muss er jedoch physikalische Gesetze berücksichtigen: Der Spieler muss bei der Konstruktion seines Raumschiffes auf die Positionierung der Antriebe und des Schwerpunktes achten, da sein Schiff sonst schnell unkontrolliert zu rotieren beginnt. Zu seiner Unterstützung steht dem Spieler hier ein System zur Stabilisierung (Stability Augmentation System, SAS) des Raumschiffes zur Verfügung¹, welches kleinere Rotationsmomente automatisch ausgleichen kann. Des weiteren muss der Spieler den Antrieb seines Raumschiffes vorsichtig nutzen, da er sich sonst plötzlich in einem ungewünschten Orbit befindet.

Doch damit es erst einmal soweit kommen kann, stellen sich einige grundlegende Fragen: Wie und wo werden die Daten des Spielers gespeichert?² Wie sieht ein Raumschiff für den Computer aus?³ Wie designt man ein Raumschiff?⁴ Wie synchronisiert man die Positionen von Hunderten von Asteroiden?⁵ Was gibt es für Möglichkeiten Gravitation zu simulieren?⁶ Wie findet man zuverlässig Fehler in einem Programm?⁷ Wie funktioniert eine Verfolgungsrakete im Weltraum?⁸ Diese Fragen sind, wie der Inhalt dieser Arbeit, nur eine Zusammenfassung der spannendsten Probleme, die sich in der Spielentwicklung stellen.

¹siehe Kapitel 5.9 Schiffsstabilisierung und -rotation per erweitertem SAS

²siehe Kapitel 5.4 Dateistruktur und Sicherheitsrisiken

³siehe Kapitel 5.8 Abstraktion von Raumschiffen

⁴siehe Kapitel 7 Erstellung von Modellen

⁵siehe Kapitel 5.3 Statussynchronisation im Netzwerk

⁶siehe Kapitel 5.12 Gravitationsberechnung und Spielfeldgenerierung

⁷siehe Kapitel 8 Softwaretests

⁸siehe Kapitel 5.1 Automatische Steuerung der Verfolgungsraketen