**Projektarbeit Informatik**

Projektdokumentation   
LaserChess V1.2



Website: <http://stocyr.github.com/LaserChess/>

Dozent: Ivo Oesch

Autoren: Marcel Bärtschi, Jascha Haldemann, Nicola Käser, Cyril Stoller

2 Semester 2012

**Zusammenfassung**

**Inhaltsverzeichnis**

1 Einleitung 1-1

2 Planungsvorgehen 2-2

3 Realisation 3-3

3.1 Main 3-3

3.2 Spiel 3-3

3.3 Logik 3-3

3.4 Grafik 3-3

4 Testvorgehen 4-4

5 Verbesserungen/Zukunft oder so… 5-5

6 Dokumentationsvorgehen 6-6

7 Spielanleitung 7-7

8 Schlusswort 8-8

Anhang 8-9

# Einleitung

Aufgabenstellung:

Innerhalb von 8 Wochen planen und realisierten wir ein ‘kleines‘ C-Programm. Am Ende der 8 Wochen hielten wir eine kleine Kurzpräsentation um unsere Vorgehensweise und Resultate aufzuzeigen.

Da wir das von uns definierten Pflichtenheft schon früh erfüllt hatten, fügten wir mehrere kleine Verbesserungen und Ergänzungen (Eastereggs) über das ganze Semester hinweg hinzu.

Dieses Projekt zählt 25% zur Modulqualifikation des 2. Semesters. Der Abgabetermin ist der 25.06.2012.

Die Idee LaserChess entstand ursprünglich von dem Brettspiel KHET 2.0 (Abbildung 1). Wir haben aber lediglich das Spielprinzip übernommen. Die Regeln, sowie Grundaufstellungen und Figuren passten wir unseren Bedürfnissen an.

In dieser Dokumentation gehen wir näher auf die Entwicklung von LaserChess ein. Wir betrachten die einzelnen Planungs-und Realisationsschritte, sowie die Resultate. Näheres zur Software von LaserChess ist in der Softwaredokumentation zu finden.



Abbildung 1: KHET 2.0

# Planungsvorgehen

Als die Spielidee nach einer erstaunlich kurzen Suche gefunden war, und das Spielprinzip von KHET verinnerlicht wurde, machten wir uns als erstes ans Pflichtenheft (Anhang 1) ran. Im Pflichtenheft definierten wir unsere Spielregeln, die Spielumgebung, der Spielablauf und der Grafikumfang der Beta-Version. Mit diesem Pflichtenheft konnten wir eine ungefähre Zeitplanung (Anhang 2) erstellen. Dann folgte die erste Softwareanalyse. Um uns einen Überblick über den Spieleablauf zu schaffen, erstellten wir ein Flowchart (Abbildung 2). Aus diesem, eine Grobe Modulaufteilung und zwar in: Grafik, Spiel, Logik und main (LaserChess). Um die Module dann in seine Funktionen aufzuspalten und diese unter uns aufzuteilen, verwendeten wir das Prinzip mit den CRC und erstellten daraus ein ‘Structured Designe‘ (Anhang 3). Wider Erwarten, funktionierte das Prinzip aussergewöhnlich gut. Innerhalb von 1 oder 2 Lektionen hatte jeder von uns mehrere Kärtchen mit schnittstellendefinierten Funktionen in den Händen, welche man individuell bearbeiten konnte. Als unser Spiel mit der Planung erstmals vorstellbar wurde, bemerkten wir auch diverse Kleinigkeiten, welche wir im Pflichtenheft ergänzen mussten. Etwas verspätet, aber zum Glück nicht vergessen, haben wir dann das Styleguideline (Anhang 4) definiert.

Abbildung 2: Flowchart

# Realisation

Gitbash und Erklärung der Moule?…

Abbildung 3: Gitbash

## Main

## Spiel

## Logik

## Grafik

Im Grafikmodul werden hauptsächlich alle Grafischen Elemente gezeichnet oder eingefügt (draw\_playground, draw\_figure, usw…). Auch diverse Umrechnungen, welche auch andere Module benutzen wie z.B. die Umrechnung der Pixel auf unsere definierten Spielfelder (pixel\_to\_map) und umgekehrt (map\_to\_pixel), oder die Initialisierung aller Bilder.

# Testvorgehen

Zuerst überprüften wir die Stabilität des Programms und überprüften, dass keine ‘Magic-Numbers‘ mehr vorhanden sind. Wir skalierten das gesamte Spielfeld herunter und veränderten andere Parameter, wie z.B. diverse Farben. Abbildung 4 zeigt, dass auch nach den Veränderungen noch alles noch in Ordnung war.



Abbildung 4: Parameter-Tuning

Als zweiten Schritt, zogen wir einen Schlussstrich und beschlossen den damals aktuellen Stand des Programms als Version 1.0 zu deklarieren. Dazu erstellten wir ein Testprotokoll und führten damit einen Blackbox-test durch. Wir notierten Fehlerquellen und Unschönheiten und fassten diese in einem Fazit zusammen.

Insgesamt führten wir 3 Blackbox-tests durch und zwar mit den Versionen: 1.0, 1.1 und dem Endprodukt 1.2. Diese sind im Anhang 3 zu finden.

{5nake}

# Dokumentationsvorgehen

Arbeitsjournal/Laborjournal:

Im Allgemeinen versuchten wir ein Laborjournal (Anhang 2) mit den Täglichen Entscheidungen zu führen. Da wir aber vorwiegend individuell bei uns Zuhause arbeiteten, war es schnell vergessen. Wir beschlossen daher, einfach den Verlauf unserer commits von Gitbash zusammenzufassen. So garantieren wir auch, dass nichts vergessen wurde.

Zeitplan:

Der nach dem Pflichtenheft erstellte Zeitplan (Anhang 1) wurde Wöchentlich (jeden Freitag) während der Projektphase kontinuierlich mit dem am Anfang eingeplanten Soll-Wert verglichen. Die Änderungen die über das Semester hinweg getätigt wurden, wurden im Zeitplan nicht mehr vermerkt.

# Spielanleitung

## Hauptmenü

Beim Start von LaserChess sieht man zuerst das Hauptmenü (Abbildung 5) mit dem Titel und ein Willkommen. Falls das Programm nicht gewöhnlich mit der .exe Datei geöffnet wurde, folgt eine kurze Mitteilung mit der Art der Öffnung. Möglich wäre in Eclipse oder mit vorgegebenen Argumenten (in V1.2 aber auskommentiert).

Zum Öffnen eines Modes oder einer anderen Funktion muss die dazugehörige Zahl (1 bis 5) eingegeben und mit Enter bestätig werden.

1 Öffnet ein normales Spiel mit der Standard-Aufstellung

2 Öffnet den Platziermodus, in welchem man die Figuren nach Wunsch selbst platzieren kann

3 Öffnet nach Eingabe und Bestätigung mit Enter der gewünschten Textdatei (z.B. Aufstellung2.txt) eine vordefinierte Aufstellung

4 Schaltet sowohl den Intro-sound wie auch den gesamten Spiele-sound aus oder ein

5 Schliesst das Programm LaserChess (auch mit dem Schliessen-button von Windows oben rechts möglich)

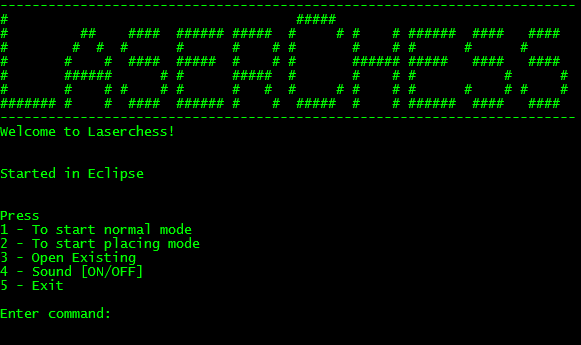


Abbildung 5: Hauptmenü LaserChess

## Normaler Spielablauf

Die vordefinierte Aufstellung besteht aus einem roten und blauen Team mit je einem Laser (oben links / unten rechts), einem König (Stern), einer Mauer und vier Spiegeln.

Rot beginnt immer und hat die Möglichkeit eine seiner Figuren entweder zu drehen oder um ein Feld zu verrücken. Dazu muss die gewünschte Figur angeklickt werden. Es erscheinen Fokusfelder (Abbildung 6), welche die möglichen Verschiebungen anzeigt. Durch klicken auf eines dieser Fokusfelder verschiebt sich die Figur. Dreht man aber mit dem Mausrad nach oben, so dreht sich die Figur um 90° gegen den Uhrzeigersinn. Dreht man das Mausrad nach unten, gibt das eine Drehung um 90° mit dem Uhrzeigersinn.

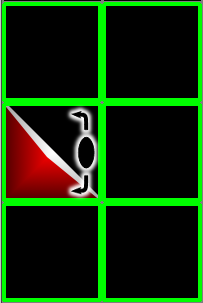


Abbildung 6: Fokusfelder

Um eine angewählte Figur wieder abzuwählen kann man entweder nochmals auf die gleiche Figur klicken oder einfach eine andere Figur anwählen.

Nach jeder Drehung oder Verschiebung ist ein Zug beendet und die Kanone des Spielers des beendeten Zuges feuert automatisch seinen Laser ab. Dies kann nicht verhindert werden!

Der Laser kann an den Spiegeln abprallen, einen König (auch den Eigenen!) treffen und das Spiel beenden oder in eine Wand oder Mauer schiessen und den Zug damit endgültig beenden.

Trifft der Laser aber einen Spiegel im Rücken, so wird dieser zerstört und ist für das fortlaufende Spiel verloren.

Die Spieler wechseln sich nach jedem Laserabschuss ab.

Ziel ist es, durch vorausschauendes und “laserisches“ Denken den gegnerischen König mit einem Laser zu treffen.

## Platziermodus

Im Platziermodus erscheint nur ein leeres Spielfeld. Es gilt nun die einzelnen Figuren nach einer vordefinierten Reihenfolge zu setzten.

Zuerst werden die Figuren mit einem Klick der linken Maustaste auf dem gewünschten Feld platziert und dann mittels Mausrad in die richtige Richtung gedreht und mit einem weiteren Klick der linken Maustaste bestätigt. Es muss also für jede Figur zweimal geklickt werden.

Reihenfolge:

1. Roter König
2. Blauer König
3. Rote Kanone
4. Blaue Kanone
5. Rote Mauer
6. Blaue Mauer
7. Abwechselnd ein roter- und ein blauer Spiegel (je 4)

(Die folgende Figur ist jeweils auch im Hauptmenü-Fenster zu sehen)

{5gewinnt}

# Schlusswort

Dieses Informatikprojekt hat uns im grossen und ganzen sehr Spass gemacht. Wir hätten nicht gedacht, dass es uns in diesem Ausmass okkupiert. Natürlich machte dann das Testen und Dokumentieren weniger Spass. Nichts desto trotz; war es eine gute Erfahrung. Vor allem in Bezug auf die Teamarbeit bei einer solch grösseren Arbeit.

Fazit: Wir haben alle zwingend vorausgesetzten Funktionalitäten des Pflichtenhefts erreicht. Auch einige der zusätzlich geplanten Funktionen, wie blablabla.

Zum Schluss hoffen wir, dass dem Leser den Inhalt der fragwürdigen Klammen in der Projektdokumentation aufgefallen sind und sich einmal kurz die Zeit nimmt unsere ‘Spiele‘ genauer unter die Lupe zu nehmen.

# Anhang

1. Zeitplanung
2. Laborjournal
3. Testprotokolle (V1.0, 1.1, 1.2)