

# Algorithm Master - Documentation

Elina Naymushina

Nikolaj Hensch

Léon Kaiser

6. Juni 2011

# Inhaltsverzeichnis

|  |          |
|--|----------|
| <b>1</b>                               | <b>2</b> |
| 1.1 Vorwort . . . . .                  | 2        |
| 1.2 Bedienung . . . . .                | 3        |
| 1.3 Projektverlauf . . . . .           | 7        |
| 1.3.1 Entwürfe . . . . .               | 8        |
| 1.4 Technische Umsetzung . . . . .     | 10       |
| 1.4.1 GUI-Verwaltung . . . . .         | 10       |
| 1.4.2 Algorithmen-Verwaltung . . . . . | 11       |
| 1.5 Fazit . . . . .                    | 12       |
| 1.5.1 Android-SDK . . . . .            | 12       |
| 1.5.2 Sonstige Tools . . . . .         | 13       |

# Kapitel 1

## 1.1 Das Vorwort zum Projekt

Wir haben die Android Applikation Algorythm-Master zur grafischen Darstellung diverser Scheduling-verfahren entwickelt.

Das Projekt umfasste eine Gesamtarbeitszeit von ca. 80 Stunden was bei unserer 5er Gruppe 16 Arbeitsstunden pro Person entspricht.

Organisatorischer Grundpfeiler waren die Mittwochstreffen, die bis auf ein paar Ausnahmen regelmäßig stattfanden, sowie der Einsatz unseres SVN-Systems.

Der Hauptgrund warum wir uns für die Android Plattform entschieden haben war, dass ein Gruppenmitglied bereits tiefgreifende Erfahrungen mit dieser Technologie besaß, auf der wir sehr gut aufbauen konnten.

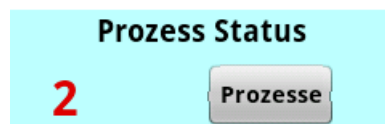
Da die Android-Plattform in diversen Versionen vorliegt, mussten wir durch diverse Diskussionen und Tests die richtige Version ermitteln. Maßgeblich waren Kriterien wie z.b. Verbreitungsgrad, API-Umfang sowie die Vermeidung von Versionskonflikten. Unsere Applikation wurde schließlich für die Android-Version 2.2 entwickelt, da diese zurzeit den größten Verbreitungsgrad besitzt.

## 1.2 Bedienung des Alogrithm Masters

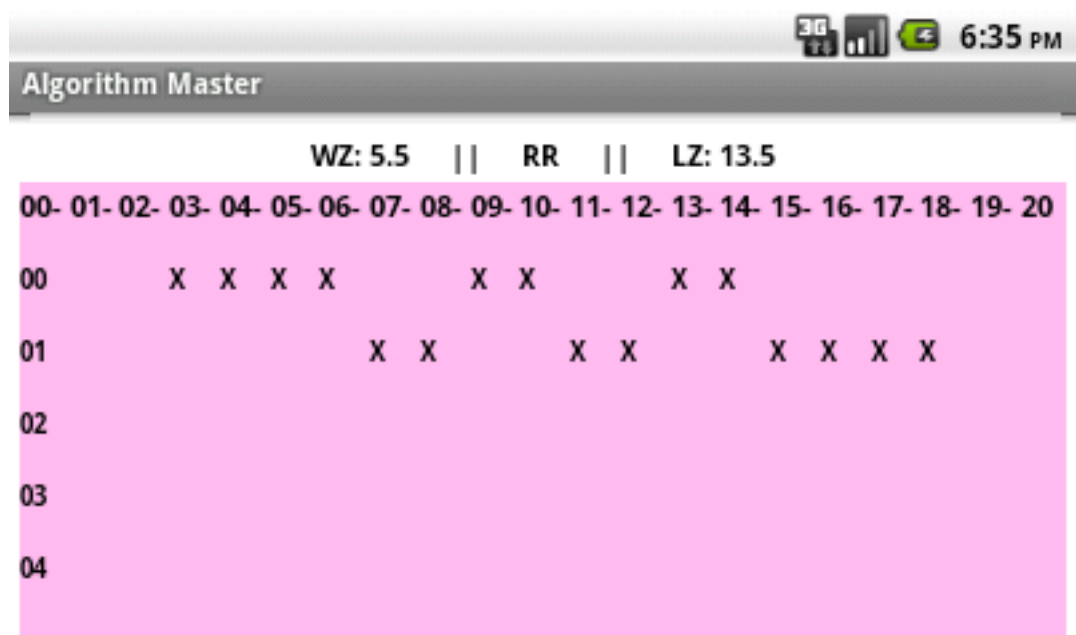
Man startet die Applikation über das Android Applikations Panel in welchem auch diverse andere Android Apps abgelegt sind.



Der Startbildschirm erscheint und mit einem Fingerdruck auf die Schaltfläche „To the Algorithms“ gelangt man ins Hauptmenü. In der oberen Hälfte sieht man die unterstützten Algorithmen und in der unteren kann die Anzahl der Prozesse festgelegt werden, welche standardmäßig mit 2 Prozessen festgelegt sind.

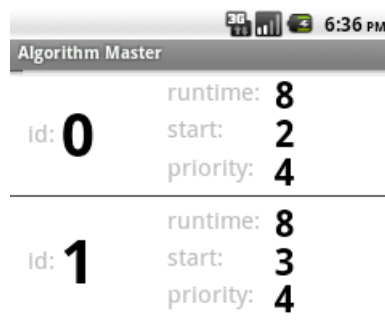


Nach dem man die bevorzugte Prozessanzahl festgelegt hat (dazu später mehr), kann man mit einem einfachen Fingerdruck den bevorzugten Algorithmus auswählen und das grafische Schaubild seiner Arbeitsweise wird angezeigt.



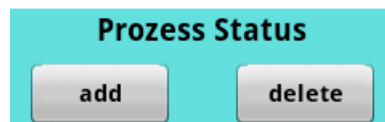
Auf der Y-Achse befinden sich die durchnummerierten Prozesse und auf der X-Achse befindet sich die Timeline. Jeder laufende Prozess, wird durch einen oder mehrere „X“-Blöcke in einer Zeile symbolisiert. In der oberen Zeile kann die Wartezeit (WZ) sowie die Laufzeit (LZ) des gewählten Algorithmus eingesehen werden. Mit dem „Back“ Button des Mobiltelefons, kann man zurück zum Hauptmenü gelangen bzw. die Applikation komplett verlassen.

Möchte man nun mehrere Prozesse hinzufügen, muss man im Hautmenü die Schaltfläche „Prozesse“ anklicken.

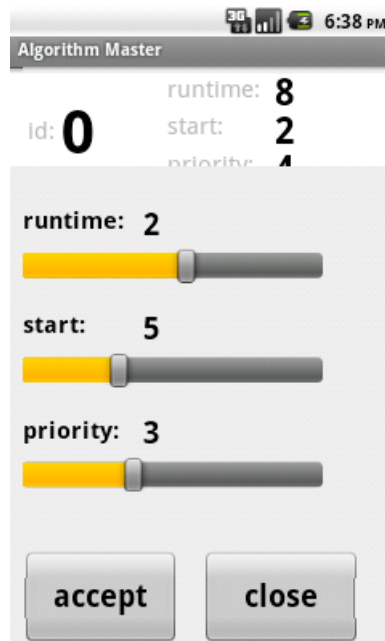


The screenshot shows the 'Algorithm Master' app interface. At the top, there is a status bar with '3G', signal strength, battery, and the time '6:36 PM'. Below the title bar, there is a table listing processes. The first process has an ID of 0, a runtime of 8, a start time of 2, and a priority of 4. The second process has an ID of 1, a runtime of 8, a start time of 3, and a priority of 4.

|              | runtime: |                    |
|--------------|----------|--------------------|
| Id: <b>0</b> | 8        | start: <b>2</b>    |
|              |          | priority: <b>4</b> |
| Id: <b>1</b> | 8        | start: <b>3</b>    |
|              |          | priority: <b>4</b> |



Nun befindet man sich in dem Prozessmenü. Mit einem Fingerdruck auf die Schaltfläche „Add“, erscheinen 3 Regler mit dem man die Laufzeit (runtime), Startzeit (start) sowie die Priorität (priority) festlegen kann.



Möchte man nun einen Prozess löschen, so drückt man zuerst auf „delete“ und drückt anschließend auf den zu löschenden Prozess. Möchte man das Menü verlassen, betätigt man den „Back“ Button des Mobiltelefons.

## 1.3 Projektverlauf

Zuerst mussten wir (Elina, Jan-Paul, Zaher, Léon, Nikolaj und Lena) als einzelne Personen zu einer Gruppe zusammenfinden. Bevor wir das Projekt starten konnten, mussten wir uns zunächst auf eine Plattform einigen. Unsere Wahl fiel dabei auf das Betriebssystem Android, da ein Gruppenmitglied von uns schon Erfahrung bei der Implementierung damit hatte. Als nächsten Schritt wurden Untergruppen gebildet sowie die Aufgaben und Deadlines zugeteilt.

Alle Gruppenmitglieder haben sich zudem das Android SDK mit dazugehörigem Emulator installiert. Die dadurch gewonnen Erfahrungen, befinden sich im Fazit-Teil dieser Dokumentation.

Da wir bereits einen erfahrenen Android-Entwickler in unserer Mitte hatten, war dieser die erste Anlaufstelle für Fragen und Vorschläge. Dennoch mussten alle anderen Mitglieder, sich am Anfang unseres Projektes in die Materie einarbeiten.

Unser erstes Gruppentreffen fand am 8. April statt und wir haben uns eine Übersicht über das Projekt verschafft.

| <b>Datum</b> | <b>Aufgaben</b>                                       |
|--------------|---|
| 8.April      | Gruppe kennen lernen, Plattform wählen                |
| 13.April     | Aufgaben aufteilen, Algorithmen präsentieren          |
| 20.April     | Klassen definieren, die GUI festlegen                 |
| 4.Mai        | Begin der Implementierung                             |
| 18.Mai       | Implementieren  |
| 25.Mai       | Deadline der Implementierung, Begin der Dokumentation |
| 8.Juni       | Vorbereitung der Endergebnisse                        |

Beim zweiten Treffen haben wir die Aufgaben verteilt und für bestimmte Bereiche die Zuständigen festgelegt. Da wir lediglich Algorithmen zu implementieren hatten, haben wir ausgemacht, dass sich jeder einen Algorithmus aussucht und diesen der Gruppe beim nächsten Treffen präsentiert (siehe Tabelle).

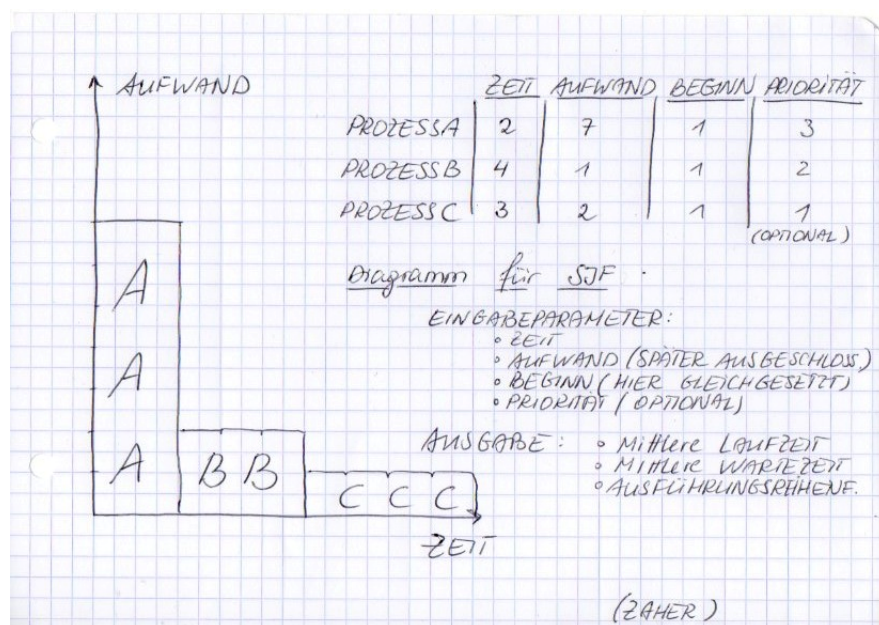
| <b>Name</b> | <b>Aufgabenfeld</b>                            |
|-------------|--|
| Elina       | SJF, Protokollieren von Treffen, Dokumentation |
| Nikolaj     | SRTF, Entwurf, Präsentation                    |
| Jan-Paul    | RR, Implementierung der Algorithmen            |
| Zaher       | LJF, Entwurf, Implementierung von Android-GUI  |
| Léon        | LRTF, Management, Dokumentation, Testen        |



### 1.3.1 Entwürfe

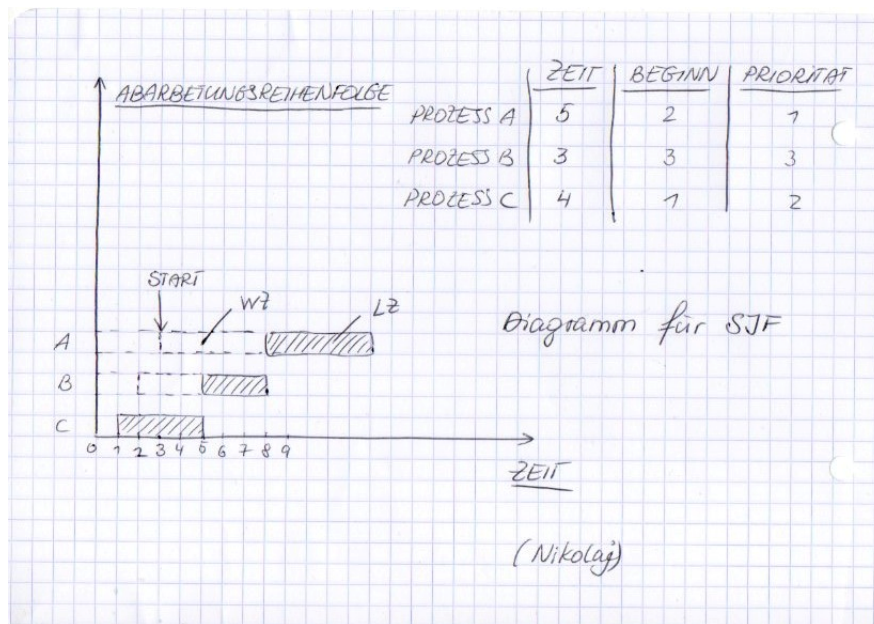
Bevor wir unsere finale Darstellungsform gefunden hatten, haben wir diverse Prototypen entworfen. Dabei viel uns auf, dass kein einheitliches Verständnis dafür bestand, was konkret zur Darstellung der Prozesse benötigt wird und was nicht.

Im ersten Schaubild haben wir die Zeit und den benötigten Aufwand dargestellt. Der Begriff „Aufwand“ ist aber je nach Kontext sehr unterschiedlich interpretierbar. Zu diesem Zeitpunkt haben wir uns auch auf weitere Eingabeparameter geeinigt (siehe unteres Bild).

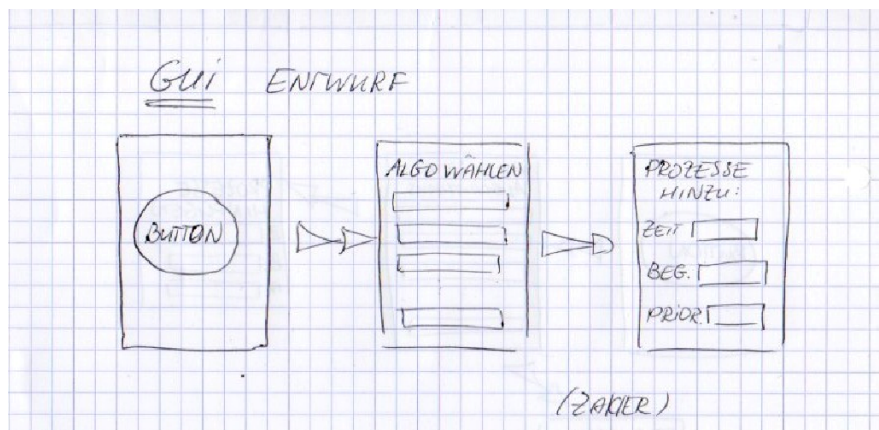


Das obige Bild zeigt unser erstes Modell welches heftig diskutiert wurde.

Schließlich hat uns Nikolaj seine eigene Version vorgestellt, welche nur noch drei Eingabeparameter benötigte. Des weiteren war die grafische Darstellung der Ausführungsreihenfolge deutlich übersichtlicher. Dieser Entwurf schaffte es schließlich auch in die finale Version.



Danach präsentierte Zaher uns seinen Entwurf der GUI, die ebenfalls implementiert wurde.

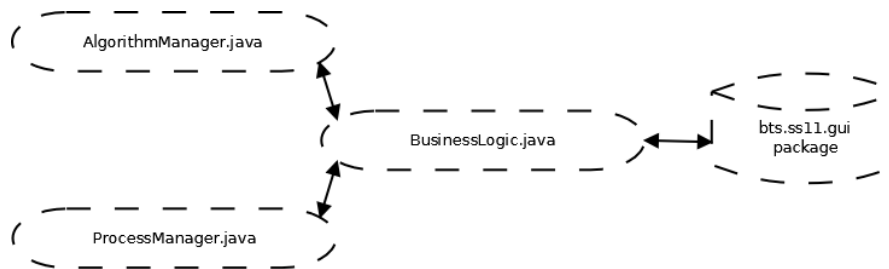


## 1.4 Technische Umsetzung

Hier legen wir zwei Themen unserer Technische Umsetzungen da.

### 1.4.1 GUI-Verwaltung

Im unteren Bild sieht man das MFC-Modell unserer Applikation. Die BusinessLogic nimmt die Interaktionen zwischen dem Kernprogramm und der GUI an und gibt diese an die jeweils andere Gegenstelle weiter. So wird beispielsweise die aktuelle Anzahl, Laufzeiten sowie Wartezeiten der Prozesse aus dem Processmanager geladen und an die GUI zur Anzeige weiter gegeben. Kommen nun User-Interaktionen von der GUI, so vermittelt die BusinessLogic diese zu den jeweiligen Programmteilen, die für diese zuständig sind. Dies könnte z.b. der AlgorithmManager sein, der den ausgewählten Algorithmus mithilfe der vorher festgelegten Prozesse berechnet.

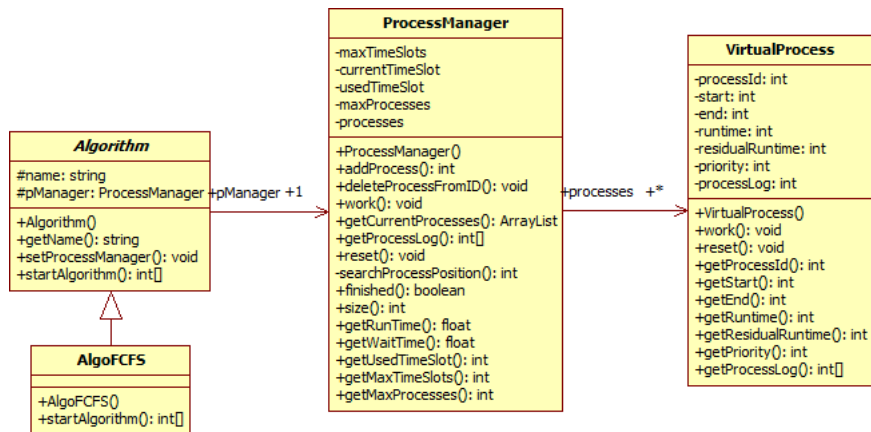


## 1.4.2 Algorithmen-Verwaltung

Um eine möglichst leichte Implementierung sowie Einbindung der Algorithmen zu garantieren, so wurde eine Oberklasse namens „ProcessManager“ geschrieben. Diese Klasse verwaltet die Algorithmen und beinhaltet alle Methoden die ein Algorithmus zur Demonstration seiner Arbeitsweise benötigt. Einige dieser Methoden sind zum Beispiel `work()`, `getCurrentProcess()`, `isfinished()` etc.

Die Algorithmen bekommen eine Instanz des ProcessManager's übergeben und erhalten auf diese Weise die Methoden sowie die vorher festgelegten Prozesse mit ihren Attributen (Laufzeit, Startzeit und Priorität).

Mithilfe dieses kleinen Frameworks wurde die Implementierung der einzelnen Algorithmen sehr vereinfacht und standardisiert.



## 1.5 Fazit

Und zu guter-letzt unser Fazit bezüglich des verwendeten Android SDKs sowie weiterer Tools.

### 1.5.1 Android-SDK

Die Installation der Andoird-Plattform ist einfach und auch für Anfänger gut umsetzbar. Leider besteht keine Verbindung zwischen dem Android-SDK und diverser IDEs wie z.b. Eclipse oder Netbeans.

Diese benötigen zunächst Plugins von Drittherstellern, wodurch es immer wieder zu Versionskonflikten kommen kann. Nach erfolgreicher Installation des SDKs, Erstellung eines Virtuellen Mobiltelefons (welches der Emulator benötigt) sowie der Installation eines Plugins zur Einbindung in die jeweilige IDE, ist das gröbste geschafft.

Nun können Programme entwickelt und in dem Android Emulator ausgeführt werden. Dies erfordert jedoch viel Zeit und Nerven von Seiten des Entwicklers. Grund dafür ist die lange Wartezeit die das starten des Emulators benötigt, welche je nach Hardware zwischen 2 bis 6-Minuten beträgt. Hat man nun diese Zeit abgesessen ist man sehr gut beraten ihn nicht wieder leichtfertig zu schließen, da man ihn sonst erneut starten müsste was die eben erwähnte Zeit in Anspruch nimmt.

Auch wenn ein Debugger vorhanden ist, besitzt dieser relativ wenige Features und besitzt aufgrund der Emulation manchmal immense Zeitverzögerungen.

Möchte man nun das Programm im voller Pracht genießen oder testen muss eine .apk Datei generiert und auf den Emulator gestartet werden. Dieser Prozess läuft nachdem man auf dem „Play-button“ von Eclipse geklickt hat, weitgehend automatisch, dennoch dauert dieser ca. 30 Sekunden.

Alles in allem erfüllt das Android-SDK seinen Zweck, ist aber nicht zuletzt wegen seines mangelhaftem Lieferumfangs auf diverse Plugins angewiesen, welches zahlreiche Versionskonflikte bedeuten kann. Des weiteren entspricht die Arbeitsgeschwindigkeit dem Java

Standard, welche die Geduld von C bzw. C++ Programmierern auf eine harte Probe stellt.

### 1.5.2 Sonstige Tools

Wir hatten neben Eclipse und dem Android-SDK noch ein SVN-System im Einsatz, welches den beiden Hauptentwicklern sowie dem Tester sehr gute Dienste geleistet hat. Auch wenn es mittlerweile besser Lösungen auf diesem Gebiet gibt bot sich SVN an, da Jan-Paul bereits einen SVN Server für seine Privaten Projekte verwendete.

Die Erlernung von LaTeX war einfacher als erwartet, jedoch ist der Konfigurationsaufwand enorm. Dennoch bietet es viele schöne features wie z.b. das auslagern von Texten, des weiteren kommt es in LaTeX Dokumenten nie zu „rätselhaften“ Problemen bezüglich der Formatierung, da man einen kompletten Einblick in den Quellcode besitzt. Dies gibt einem eine neue Dimension der Kontrolle und Transparenz. Außerdem gilt auch für LaTeX die bekannte „Configs are easier to config“ Regel da man sich nicht durch hunderte Menüs klicken muss, um irgendeinen Hacken oder Button zu finden (oder auch nicht). Google führt einen in so gut wie alle Fällen zur richtigen Lösung des eigenen Problems.