Arbeitsjournal

# 21.06.2016

|  |  |
| --- | --- |
| Erstes UML | Pascal Honegger |
| Zweites (leserliches) UML | Alain Keller |
| Erstes GUI mit leerem Schachbrett | Pascal Honegger |

# 28.06.2016

|  |  |
| --- | --- |
| GUI um Schachfigur erweitert | Pascal Honegger |
| Logik «Feld besetzt» implementiert | Alain Keller |
| Implementation | Pair-Programming (Pascal & Alain) |
| Arbeitsjournal | Pascal Honegger |

Erfahrungen

# Damenproblem & Backtracking

Wir hatten in diesem Modul den ersten Kontakt mit dem Damenproblem. Die Herausforderung an diesem Problem ist das sogenannte Backtracking. Wir hatten zuvor noch nie mit Backtracking zu tun gehabt. Es war ein gutes und forderndes Projekt, mit welchem wir erste Erfahrungen mit dem Backtracking sammeln konnten. Zu beginnt dachten wir, dass die Implementation viel komplexer sein wird als bei einem «normalen» Algorithmus. Wir waren daher erstaunt, dass wir die Implementation schnell implementieren konnten.

# Multi-Threading

Da wir dem User konstant Feedback geben möchten, müssen wir das Ausführen unserer Berechnungen verlangsamen. Dazu haben wir die Methode «Thread.Sleep» benutzt. Das Problem an diesem Ansatz ist, dass der Main GUI-Thread blockiert wird. Daher erstellen wir für die Berechnung der Schritte einen neuen Thread. Die «Race-Condition»-Probleme haben wir über das Blockieren des User-Inputs umgangen. Lediglich JavaFX hat noch Probleme beim Aktualisieren eines GUI-Elementes ausserhalb des GUI-Threads, was aber umgangen werden konnte.

# Streams

Um einfacher nach gewissen Kriterien in einer Liste zu Filtern benutzten wir Streams. Ein ähnliches Konzept kannten wir schon aus C# (LINQ). Wir brauchten Streams z.B. um herauszufinden, ob eine Dame in der gleichen Spalte ist. Mit unserer Erfahrung aus C# ging das relativ gut, wir merkten jedoch, dass Streams noch nicht ganz so ausgereift ist wie LINQ.

# JavaFX

In diesem Projekt haben wir unsere ersten Erfahrungen mit JavaFX gesammelt. Dabei ist uns vor allem aufgefallen, dass viele ähnlich ist wie in C# & XAML. Der Hauptunterschied liegt bei den Bindings und dem Initialisieren. In C# sind die Bindings und das Verbinden einer Klasse mit dem Controller viel automatischer gelöst.

# Erweiterungen

Unser Programm kann nur von null aus die erste Position finden, und dann die nächsten. Als einziges "Feature" haben wir eine TextBox implementiert, in welcher man das Intervall, in welchem die Damen platziert werden. Wir hätten noch die Möglichkeit umsetzen können, dass der User Damen selber platzieren kann, und dann mit diesen nach einer Lösung geschrieben werden. Ein weiteres Feature wäre ein File gewesen, in welchen alle gefunden Lösungen gespeichert werden.