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

# 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.

# 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 unsere Erfahrung aus C# ging das relativ gut, wir merkten jedoch, dass Streams noch nicht ganz so ausgereift ist wie Linq.

# JavaFX

# 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 den 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.