

Besprechung 5.12.2018

Tagesordnung

- Tagesordnung bestätigen
- Kennenlernen je nach Bedarf ~ 10 min
- Kollaborationsvereinbarung ~ 25 min
 - Ziele
 - Arbeitszeit
 - Verhaltenscodex
- Technologie ~ 30 min
- Aufgabenverteilung ~ 25 min

Präambel

Ziel dieses Meetings ist, entweder auf einen gemeinsamen Nenner zu kommen und konkrete Aufgaben zu definieren **oder** weiterhin getrennte Wege zu gehen.

Ziel

Jeder erklärt kurz seine persönlichen Erwartungen und es wird ein gemeinsames Ziel verabschiedet. Vorschlag Generell:

Wir wollen durch effektives kollaboratives Zusammenarbeiten und Ausnutzung der gegebenen Ressource mit moderatem Zeitaufwand die SAT Challenge der Vorlesung Formale System gewinnen. Unser Vorteil gegenüber anderen Teams ist die strukturierte, organisierte und dadurch effiziente Arbeitsweise im Vergleich mit den anderen studentischen Teams. Martins Ziel:

Mir persönlich ist es auch nicht so wichtig zu "gewinnen" oder gut abzuschneiden, ich möchte eher etwas an meiner Programmierung arbeiten und finde das Thema recht interessant.

Zeitaufwand

Jeder nennt seine persönlichen Vorstellungen, damit wir mit den gegebenen Ressourcen das bestmögliche Ergebnis erzielen können.

Verhaltenscodex

- Kollaboration ist unsere Stärke
 - Entscheidungen sind bindend, d.h. sie werden unabhängig von der persönlichen Meinung umgesetzt
- Absolute Verlässlichkeit
 - Lieber Aufgaben zurückweisen als Deadlines nicht einhalten

→ Wir haben nur 2 Wochen und können uns keine Verzögerung leisten (Aber:)

- Dualismus von Kritik:
 - a. nur konstruktive, rationale Kritik ist zulässig
 - b. Kritik ist eine Chance, kein Ärgernis
- Anerkennung von Verantwortlichkeit: Rechenschaft und Entscheidungshoheit
- Explizit ist besser als Implizit.
 - Explizite Kommunikation ist besser als Implizite Kommunikation **Aber:** Was geklärt ist wird nicht wiederholt
 - sämtlicher Code muss kommentiert und für jeden verständlich sein (Stichwort: Kollaboration)
 - Source Code Englisch, docs in Deutsch
 - Abkürzungen, bei denen nicht sicher ist ob sie allen bekannt sind, sind nicht zu verwenden
- Jedes Feature muss durch einen **schriftlichen** Vorher/Nachher Vergleich bezüglich der Performance begründet werden.

Programmers waste enormous amounts of time thinking about, or worrying about, the speed of noncritical parts of their programs, and these attempts at efficiency actually have a strong negative impact when debugging and maintenance are considered. We should forget about small efficiencies, say about 97% of the time: **premature optimization is the root of all evil**. Yet we should not pass up our opportunities in that critical 3%.

— Donald Knuth, Paper: "Structured Programming With Go To Statements"

- Kürze ist in Ordnung, die Existenz und Einsehbarkeit des Nachweis ist relevant

Technologie

- Statements zum Solver
- Statements zur Programmiersprache
- Wahl des Kommunikationsmittels
- Verfahren mit offenen Technologieentscheidungen
 - im Notfall Mehrheitsbeschluss (Betonung von strukturiertem und kollaborativen Vorgehen)

Konkrete Organisation

Zeitplan

Woche 1: Erarbeiten der Features

Woche 2: Zusammenbringen und Feinschliff

Aufgabenverteilung

Hohe Priorität

- Recherche und Auswahl des Solvers, erarbeiten der korrekten Flags
- Encodings und Verbesserungen
- Präprozessor: implementieren oder nutzen
- Planung
 - Zuweisung und Rechenschaft für das allgemeine Zusammenarbeiten (Prinzip der Verantwortlichkeit)
 - zweigeteilte Tasks
 - präzise und nicht überschneidende Tasks definieren
- Anforderungen des Wettbewerbs überprüfen
 - Annahme: Wir dürfen alles verwenden. Als quantitatives Kriterium wird dabei die Zeit vom Programmstart per Kommandozeile bis zur Ausgabe der Lösung auf der Kommandozeile gewertet.
- Tipps von den Tutoren koordinieren
 - Kontaktperson innerhalb der Gruppe

geringe Priorität

- automatisches Profiling
 - hängt vom Zeitbedarf für die Implementierung ab
- Logikminimierer nachschauen

Wenn Richard und Georgs Source Code übernommen werden sollte

- Listen durch vorallokierte ndarrays ersetzen