

Programmieren in C

SS 2019

Vorlesung 9, Dienstag 25. Juni 2019
(Parsing, Graphics)

Prof. Dr. Peter Thiemann
Lehrstuhl für Programmiersprachen
Institut für Informatik
Universität Freiburg

Blick über die Vorlesung heute

■ Organisatorisches

- Erfahrungen mit dem Ü7
- Tutorengespräch
- Temperaturen

intplane + Dateien

Erinnerung + Erklärung

Aktuelles

■ Inhalt

- Recursive Descent Parsing
- Grafik

Prinzip + Beispiel JSON0

OpenGL Secrets #2

- **Ü8: weiterer Aufbau für Game of Life**

- Zusammenfassung / Auszüge
 - 5 × mehr zu OpenGL in der Vorlesung
 - 4 × Aufgaben diesmal gut machbar
 - 2 × Aufgaben gut und verständlich \
 - 2 × Probleme mit OpenGL auf dem eigenen Rechner
 - 2 × Aufgaben interessant waren

■ Einzelkommentare

- Konnte ein Segfault mit Hilfe von gdb finden
- Sprengt ETCS-Rahmen komplett
- Schwer, einen Test für Aufgabe 3 (erfahrungen-h) zu schreiben

Ironie?

- Schöne Aufgaben
- Schwammige Aufgabenstellungen
- Angenehme, lehrreiche Aufgaben
- Will kein OpenGL in Aufgaben mehr

Können wir nicht erfüllen, aber im Abschlussprojekt wird es eine Option ohne Graphik geben

■ Bearbeitungszeiten / Statistik

- Diesmal viele Erfahrungen (39)

Wohl, weil in Aufgabe gebraucht 😊

- Durchschnitt 3,8 Stunden

Aber der Durchschnitt sagt nicht viel!

- Median 9-10 Stunden

Unser Ziel: ca 8 Stunden

- Maximum 2x24 Stunden, 5 Personen \geq 15h

Uns mal kontaktieren, wo das Problem liegt

Tutorengespräch

- Gesprächstermin mit einem Tutor machen!
 - Obligatorisch für die Studienleistung
 - Wir wollen wissen ob Sie die Übungen selbst machen oder ob es ein [Mechanical Turk](#) ist
 - **Frist zur Terminvereinbarung: bis 2.7. vor der Vorlesung**
 - Die Tutoren schicken diese Woche (noch ein letztes Mal) eine Aufforderung
 - Kein Tutorengespräch --- keine Studienleistung

Aktuelles: Temperaturen (1/2)

■ **Battle for the thermostat: Gender and the effect of temperature on cognitive performance**

- Betrachtet den Einfluss der Temperatur auf die kognitive Leistung in Abhängigkeit vom Geschlecht
- Bei steigenden Temperaturen werden Frauen besser in Mathe und Sprachen
- Bei Männern umgekehrt, aber geringere Verschlechterung
- Kein Einfluss auf kognitive Reflektion
- Also: Zur Steigerung der Produktivität bei gemischter oder rein weiblicher Belegschaft sollte die Raumtemperatur höher als bisher gewählt werden

– Originalartikel:

<https://doi.org/10.1371/journal.pone.0216362>

Aktuelles: Temperaturen (2/2)

■ Beispiel für CRT (cognitive reflection test)

- Ein Schläger und ein Ball kosten zusammen 1.10€.
 - Der Schläger kostet 1€ mehr als der Ball.
 - Wieviel kostet der Ball?
-
- Kennzeichnend für CRT: die intuitive Lösung ist falsch

■ Motivation: JSON

- JSON (JavaScript Object Notation) ist ein textuelles Datenaustauschformat
- Es ist für Mensch und Maschine einfach zu lesen und zu schreiben
- JSON repräsentiert Objekte, Felder und primitive Daten
- **Wir wollen eine Teilmenge von JSON einlesen (**parsen**).**

■ Das JSON0 Format

- Einfaches Beispiel

```
{ "fruit": "Apple", "size": "Large", "color": "Red" }
```

- Geschachteltes Beispiel

```
{ "host": "localhost",  
  "port": 3030,  
  "public": "../public/",  
  "paginate": { "default": 10, "max": 50 },  
  "mongodb": "mongodb://localhost:27017/api"  
}
```

■ Definition JSON0

Object ::=

{ }

{ Members }

Members ::=

Member

Member , Members

Member ::=

String : Value

Value ::=

Number

String

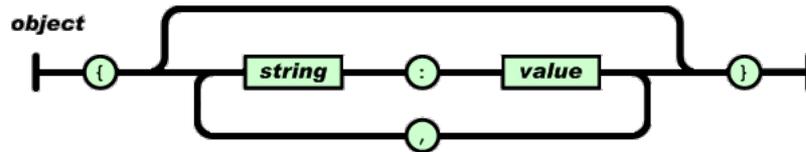
Object

Object, Members,
Member, String, Value,
Number sind **Variable**

{ } , : sind
Terminale, Zeichen, die
in der Eingabe so
vorkommen müssen

Übereinanderstehende
Zeilen sind Alternativen

■ Graphische Darstellung der Regeln für Object



- Railroad Diagram
- Jeder Durchlauf von links nach rechts liefert ein gültiges Objekt
- Dabei werden Zeichen ausgegeben und Variable „aufgerufen“
- Einlesen (parsen) geht genauso, nur dass die Zeichen gelesen und verglichen werden
- Richtschnur: eine (ggf rekursive) Funktion pro Variable

■ Überlesung von Leerzeichen

- #include <ctype.h>
- Enthält Klassifikationsfunktionen für char
 - isspace(), isalpha(), isdigit(), ...
- Überlesen durch eine while-Schleife:

```
int skip_ws (FILE * f) {  
    int c = fgetc(f);  
    while (isspace( c)) {  
        c = fgetc(f);  
    }  
    return c;    // return first non-space character  
}
```

■ Zurückstellen von Zeichen

- Oft muss das erste Zeichen bereits gelesen sein, um zu entscheiden, wie es weitergeht
- Beispiel: `parse_value()`
- In diesem Fall muss manchmal das Zeichen wieder zurückgestellt werden

`int ungetc(int char, FILE *stream)`

- Die nächste Leseoperation erhält das zurückgestellte Zeichen.

Hinweise zum Übungsblatt

- OpenGL Secrets #2

- Alles zu stdio

https://www.tutorialspoint.com/c_standard_library/stdio_h.htm

- Alles zu ctype

https://www.tutorialspoint.com/c_standard_library/ctype_h.htm