Bachelor-Praktikum (INF 105)

Vorbesprechung

Tobias Werner M.Sc.
tobias.werner@uni-bayreuth.de
http://www.ai3.uni-bayreuth.de

```
template <class O, typename F, typename A, typename B>
    class scope_guard_object_2 : public scope_guard_base
{
    friend class scope_guard_base;

    public:

        ~scope_guard_object_2() NOTHROW
            { execute_unless_dismissed(*this); }

        static scope_guard_object_2<O, F, A, B> create
            (O &guard_object, F guard_function, const A param_a, const B param_b)
        {
            return scope_guard_object_2<O, F, A, B>
                (guard_object, guard_function, param_a, param_b);
        }

    private:

        scope_guard_object_2
            (O &guard_object, F guard_function, const A param_a, const B param_b)
            :
            guard_object_(guard_object),
            guard_function_(guard_function),
            param_a_(param_a),
            param_b_(param_b)
        { }

        void execute() NOTHROW
            { (guard_object_.*guard_function_)(param_a_, param_b_); }

        O &guard_object_;
        F guard_function_;
        const A param_a_;
        const B param_b_;
};
```

Überblick

Voraussetzungen
- Konzepte der Programmierung (INF 107)
- Algorithmen und Datenstrukturen (INF 109)

Aufgabe
- Acht Aufgaben mit fester Abarbeitungsreihenfolge
- Pro Aufgabe 3 Wochen Bearbeitungszeit einplanen
- Aufgaben stehen geschlossen im eLearning zur Verfügung

Arbeitsaufwand
- 180h (6 LP)
- 3 Tage Vollzeit pro Aufgabe

Betreuung
- Wöchentliche Zusammenkunft im CIP
- Klärung von Fragen zur Aufgabenstellung
- Beantwortung von allgemeinen Fragen zu C++
- Bei Bedarf: Einführungskurs C++

Bewertung
- Individuelles Testat zu jeder Aufgabe
- Alle Testate müssen bestanden werden
- Testatpunkte ergeben individuelle Note
- Ausschlussforderungen beachten!

Anmeldung
- Verbindliche Anmeldung am Lehrstuhl
- Verbindliche Anmeldung auf CampusOnline
- Anmeldung im eLearning-Kurs

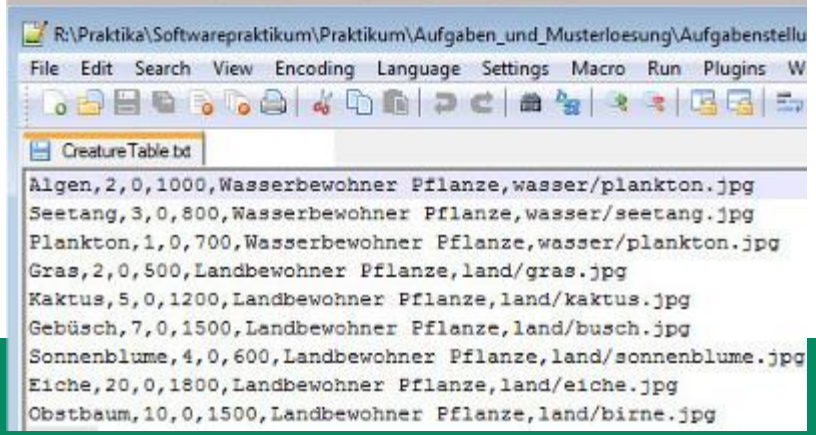
Aufgabenstellung Biosim

Aufgaben

1. Textdatei mit Kreatureigenschaften lesen
2. Bilddateien für Grafikausgabe laden
3. Benutzeroberfläche erstellen
4. Zufallslandschaft erzeugen und darstellen
5. Kreaturen plazieren und zeichnen
6. Pfadfindung mit A* implementieren
7. KI mit endlichen Automaten realisieren
8. Testplan aufstellen und abarbeiten

Rahmenbedingungen

- Erstellung einer Gesamtsoftware
- Programmiersprache C++
- Freie Wahl Compiler / Bibliotheken
- **Keine Gruppenarbeit!**



Bewertungsformular

Bachelorpraktikum BioSim — BEWERTUNGSBOGEN

Angewandte Informatik 3, Universität Bayreuth
Tobias Werner, M.Sc.

Name	Mat-ID	Semester	Testate	1	2	3	4	5	6	7	8
Punkte	0	1	2	Termine							

Funktionale Anforderungen

Funktionalität	Aufgabenstellung nicht umgesetzt	Aufgabenstellung im Wesentlichen umgesetzt	Aufgabenstellung vollständig umgesetzt								
Fehlerarmut	offensichtliche Laufzeit- oder Compilezeit-Fehler	Laufzeit-Fehler nur bei Spezialfällen	keine Laufzeitfehler selbst bei intensiven Tests								
Speicherverwaltung	offensichtliche Speicherlecks	Speicherlecks nur bei Spezialfällen	keine Speicherlecks								
Fehlerbehandlung	keine Fehlerbehandlung	provisorische Fehlerbehandlung	Ausnahmebehandlung mit Fehlersicherheitsgarantien								

Nichtfunktionale Anforderungen

Übergabekonventionen	keine Beachtung sinnvoller Übergabekonventionen	manchmal korrekte Nutzung von Referenzen, Zeigern	durchweg korrekte Nutzung von Zeigern und Referenzen								
Const-Korrektheit	keine Verwendung von const	manchmal Nutzung von const (Parameter, Member, Methoden)	durchweg korrekte Nutzung von const								
Datenstrukturen	keine effizienten Datenstrukturen oder Algorithmen	manchmal Nutzung effizienter Datenstrukturen und Algorithmen	durchweg Nutzung effizienter Datenstrukturen und Algorithmen								
Schlichtheit	keine Nutzung von Paradigmen, umständliches Vorgehen	seltene Paradigmen-Nutzung, geradliniges Vorgehen	intensive Paradigmen-Nutzung, z.B. RAII, unique_ptr, move								

Wartbarkeits-Anforderungen

Lesbarkeit	beliebige oder inkonsistente Namenskonventionen	manchmal inkonsistente oder unaussagekräftige Benennungen	konsistente, sprechende, sinnvolle Benennungen								
Formatierung	Code ist unformatiert	wenige zu lange Zeilen, inkonsistente Formatierung	konsistente Formatierung, passende Zeilenumbrüche								
Strukturierung	Quellcode unstrukturiert ohne Aufteilung in Klassen, Methoden	wenige zu große oder zu kleine Funktionen, Klassen, Dateien	übersichtliche Aufteilung in Funktionen, Klassen, Dateien								
Dokumentation	keine oder fehlerhafte Dokumentation	Dokumentation zu wenig, zu viel, oder ungenau	Dokumentation ist genau und hat korrekten Umfang								

Testplan-Anforderungen

Umfang	insgesamt wenige Testfälle, Testplan nicht durchgeführt	wenige Testfälle pro Kategorie, insgesamt genügend Testfälle	sehr umfangreicher Testplan	
Abdeckung	Abdeckung nur von trivialem Programmverhalten	Berücksichtigung von typischen Fehlersituationen	intensive Abdeckung selbst von seltenen Fehlersituationen	
Präzision	Durchführen von Testfällen nach Beschreibung unzuverlässig	Durchführen von Testfällen im Regelfall möglich	sehr präzise und spezifische Formulierung aller Testfälle	
Struktur	unstrukturierter Testplan	Angabe von erwartetem und beobachtetem Verhalten	aufeinander aufbauende Tests, Nennung von Voraussetzungen	

Notenskala	1,0	1,3	1,7	2,0	2,3	2,7	3,0	3,3	3,7	4,0	n.b.
max	176 ≥	154 ≥	140 ≥	126 ≥	112 ≥	98 ≥	84 ≥	70 ≥	56 ≥	42 ≥	28 ≥

Summe

Ausschlussanforderung: bei 0 Punkten Funktionalität, Fehlerarmut, Testplan-Umfang in einem Testat ist das BSc-Praktikum nicht bestanden

Gesamt Note

Nutzen für die Teilnehmer

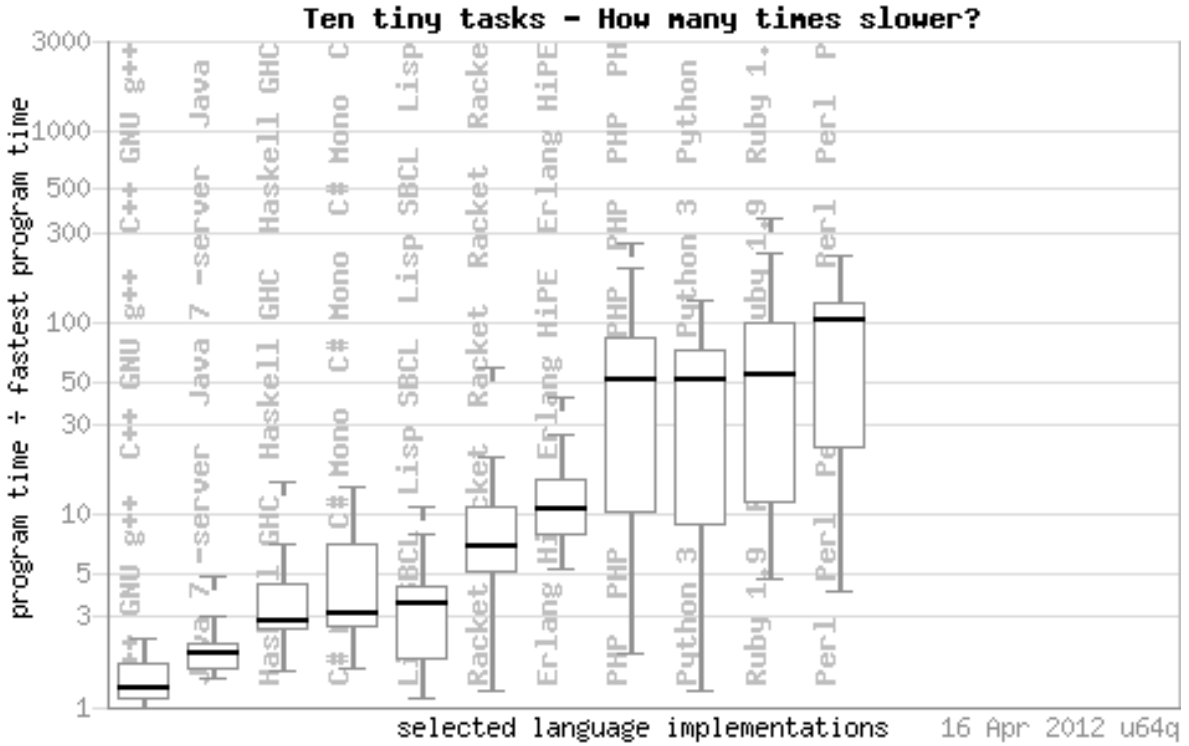
- Individuelle Programmiererfahrung sammeln
- Alltags-Probleme des Programmierens kennen lernen
- Eine weitere Programmiersprache erlernen
- Verschiedene Werkzeuge kennen lernen und einsetzen
- Entwicklung eines abgeschlossenen Gesamt-Systems
- Vorbereitung auf eventuelle Projekte und Abschlußarbeiten am Lehrstuhl

Warum C++? (1)

Geschwindigkeit bei datenintensiven Anwendungen

- Manuelle Speicherverwaltung
- Offline-Kompilierung und -Optimierung
- Kompilierung für spezielle Zielarchitektur
- ...

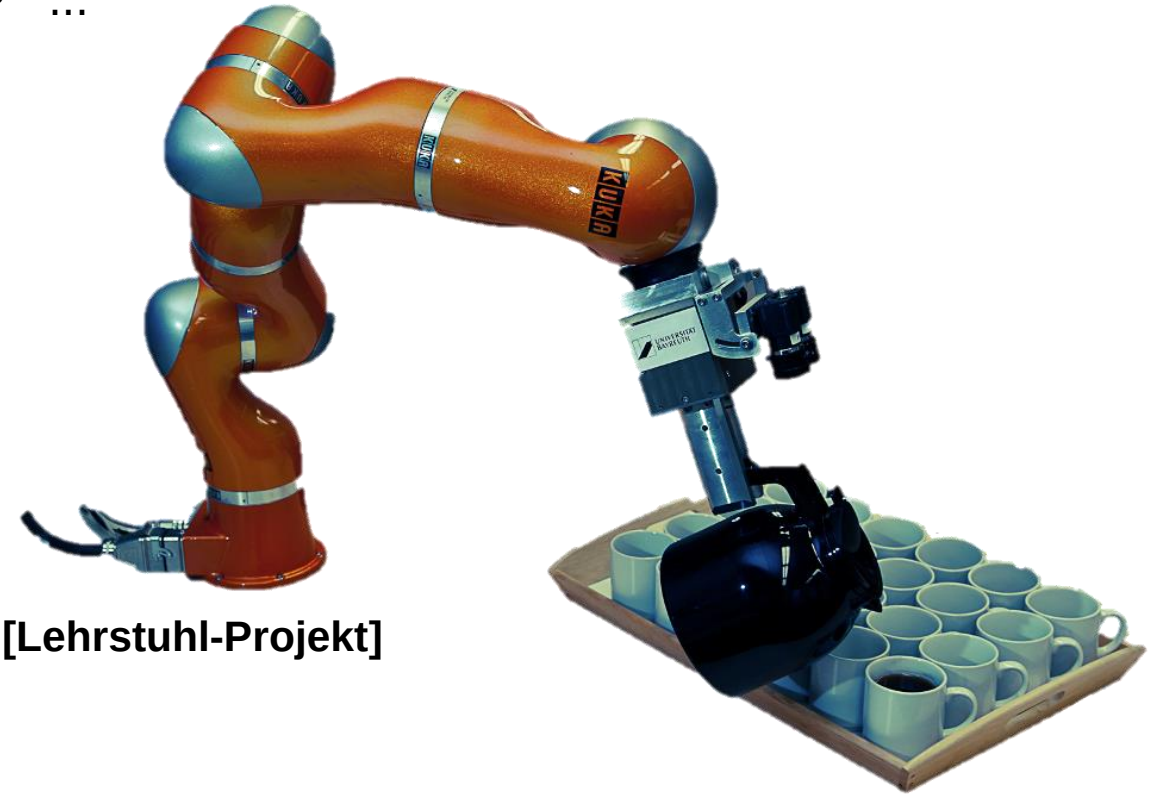
[http://shootout.alioth.debian.org]



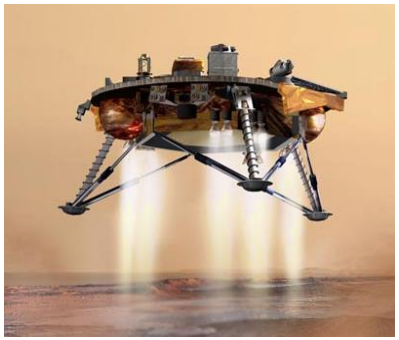
Warum C++? (2)

Hardware-nahe Programmierung

- Direkter Zugriff auf native Bibliotheken: Grafik, Ein- und Ausgabe, Netzwerk
- Zugriff auf Inline-Assembler
- Echtzeitfähigkeit
- ...



[Lehrstuhl-Projekt]



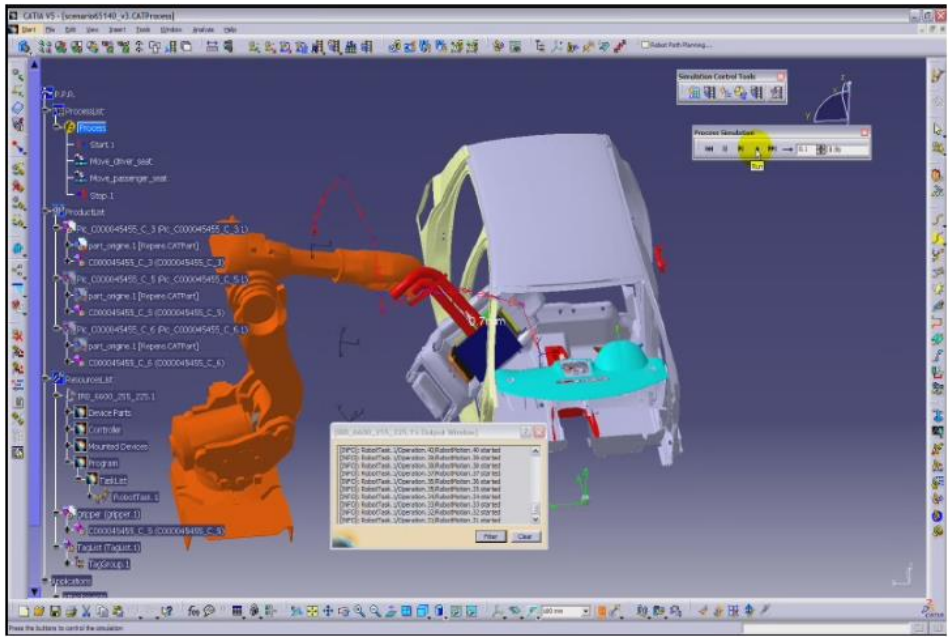
[http://code.nasa.gov/cfe]

[www.elderscrolls.com]

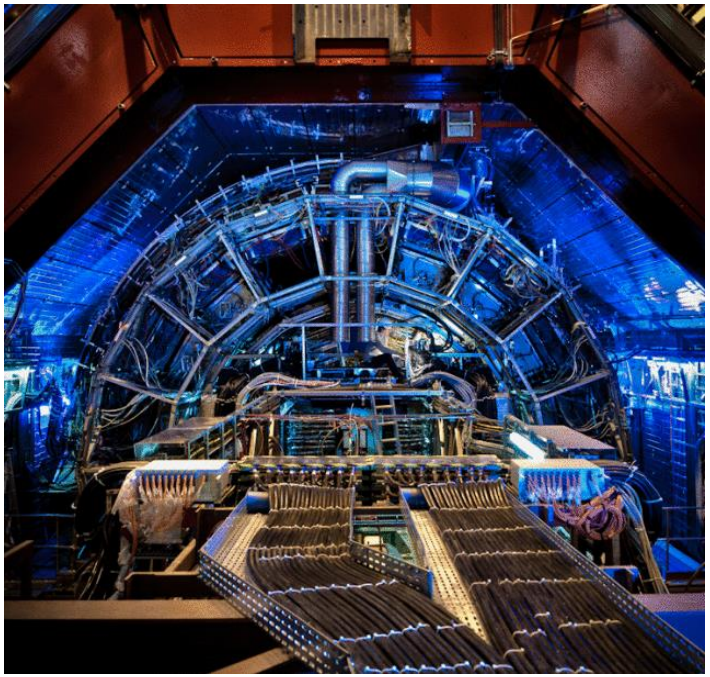
Warum C++? (3)

Moderne Programmierparadigmen

- RAII (Resource-Acquisition-is-Initialization)
- Kompilierzeit-Templates
- Erweiterte statische Typsicherheit
- ...



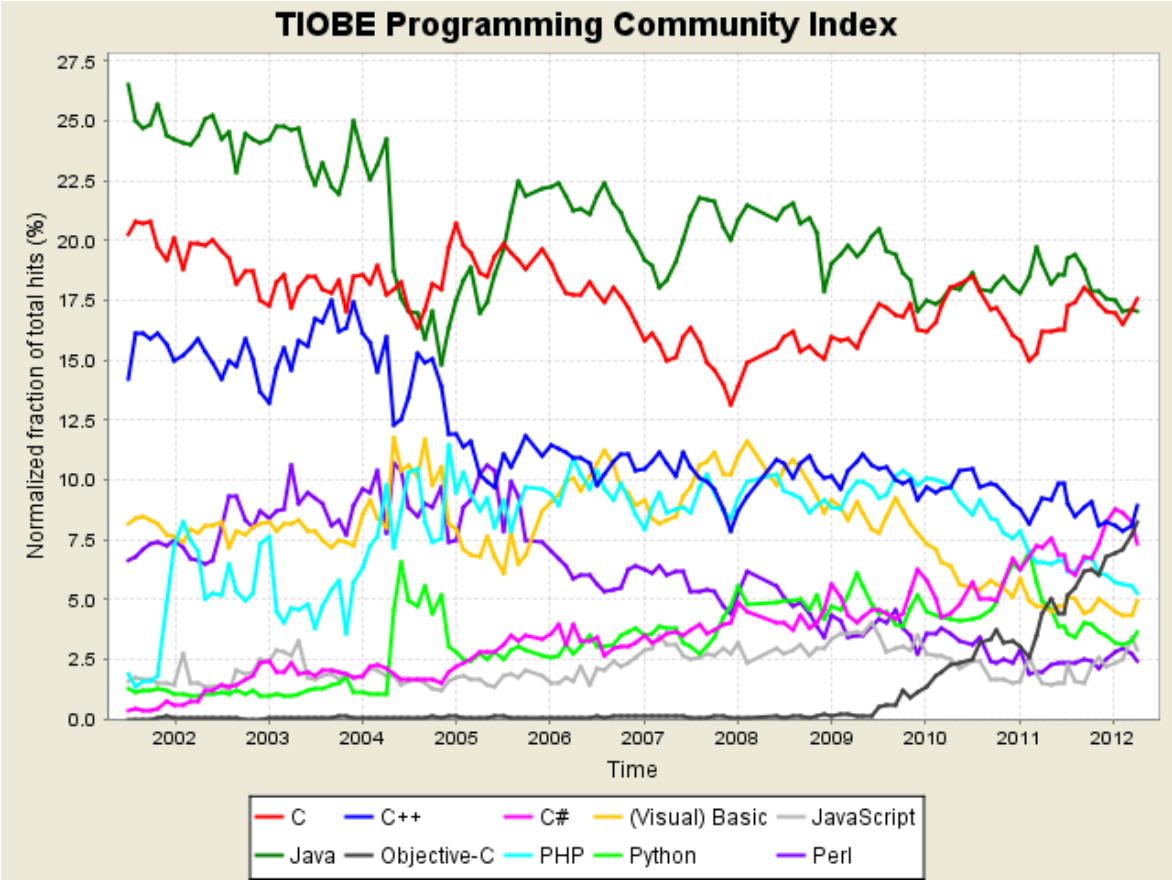
[http://media.3ds.com]



[http://cdsweb.cern.ch/record/1436153]

Warum C++? (4)

Verbreitungsgrad C und C++



[http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html]

Weiteres Vorgehen

Anmeldung

- Anmeldung im eLearning-Kurs
- **Verbindliche Anmeldung am Lehrstuhl**
  → Nächste Woche, selbe Zeit, selber Ort
  → Alternativ bis dahin per eMail an tobias.werner@uni-bayreuth.de
  → Benötigte Daten: Name, Matrikelnummer, Studiengang, Semester
- Teilnahmebeschränkung: 12 Personen!
  → Losverfahren bei Überschreitung
  → Gewährleistet intensive Betreuung

CIP-Termin

- Abstimmung per Doodle-Umfrage im eLearning
  → Abstimmung läuft bis Ende nächster Woche
- **Aufgabenbearbeitung vor den CIP-Treffen**
  → **CIP-Treffen nur für Fragen und Testate nutzen!**
- Erstes CIP-Treffen in der darauffolgenden Woche
  → Kurze Einführung in C++ / VisualStudio