

# Entwicklung eines Frameworks für Graphenspiele

## Praxis der Software-Entwicklung

INSTITUT FÜR THEORETISCHE INFORMATIK · FAKULTÄT FÜR INFORMATIK

Thomas Bläsius  
Andreas Gemsa  
Tamara Mchedlidze  
Dorothea Wagner  
ITI Wagner



## 1. Organisatorisches

## 2. Ablauf

# Vorstellung der Teilnehmer

Das sind wir...



Thomas Bläsius



Andreas Gemsa



Tamara Mchedlidze

Wer seid ihr?

- Name
- Vorkenntnisse

Zwei Module:

- Praxis der Softwareentwicklung – *PSE* (Pr.Nr. 529)
- Teamarbeit in der Software-Entwicklung – *TSE*  
(Pr.Nr.455)

## Zwei Module:

- Praxis der Softwareentwicklung – *PSE* (Pr.Nr. 529)
  - Teamarbeit in der Software-Entwicklung – *TSE* (Pr.Nr.455)
- 
- Prüfungsanmeldung über QISPOS
  - Anmeldezeitraum 23. April – 18. Mai  
Danach keine An- und Abmeldung mehr möglich.  
⇒ Aussteiger bekommen 5.0

# Ziele

# Programmieraufgaben



≈ 200 LOC



# Ziele

## Programmieraufgaben



≈ 200 LOC

## Windows Vista

≈ 50.000.000 LOC



# Ziele

## Programmieraufgaben

≈ 200 LOC

Paint.NET	≈ 36.000 LOC
Mozilla Firefox	≈ 100.000 LOC
Mozilla Thunderbird	≈ 500.000 LOC
mySQL	≈ 1.000.000 LOC
KDE core	≈ 4.200.000 LOC
Linux kernel 3.2	≈ 15.000.000 LOC
Windows Vista	≈ 50.000.000 LOC

# Ziele

Programmieraufgaben

≈ 200 LOC

PSE – GraphGames Framework

≈ 10.000 LOC

Paint.NET

≈ 36.000 LOC

Mozilla Firefox

≈ 100.000 LOC

Mozilla Thunderbird

≈ 500.000 LOC

mySQL

≈ 1.000.000 LOC

KDE core

≈ 4.200.000 LOC

Linux kernel 3.2

≈ 15.000.000 LOC

Windows Vista

≈ 50.000.000 LOC

# Ziele

- Umsetzung eines vollständigen Softwareprojekts nach dem aktuellen Stand der Softwaretechnik

- Umsetzung eines vollständigen Softwareprojekts nach dem aktuellen Stand der Softwaretechnik

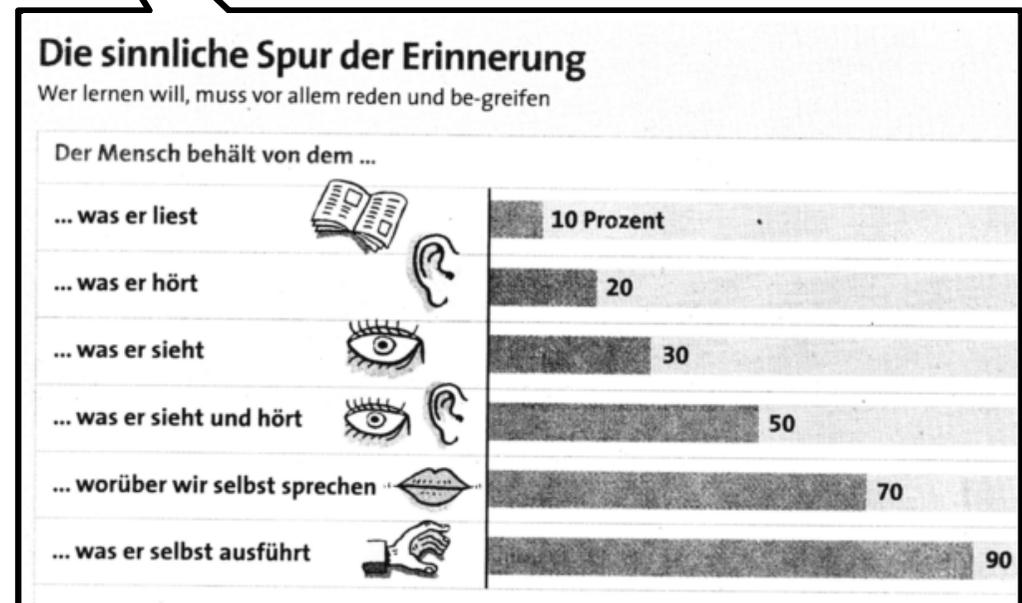
- Vorgehen nach Phasenmodell
  - Pflichtenheft
  - Entwurf
  - Implementierung
  - Validierung
  - Systemabnahme

# Ziele

- Umsetzung eines vollständigen Softwareprojekts nach dem aktuellen Stand der Softwaretechnik
- Praktische Umsetzung von Verfahren des Softwareentwurfs und der Qualitätssicherung

# Ziele

- Umsetzung eines vollständigen Softwareprojekts nach dem aktuellen Stand der Softwaretechnik
- Praktische Umsetzung von Verfahren des Softwareentwurfs und der Qualitätssicherung



# Ziele

- Umsetzung eines vollständigen Softwareprojekts nach dem aktuellen Stand der Softwaretechnik
- Praktische Umsetzung von Verfahren des Softwareentwurfs und der Qualitätssicherung
- Implementierungskompetenz

# Ziele

- Umsetzung eines vollständigen Softwareprojekts nach

der Stufe 1: Schüler

```
10 PRINT "HELLO WORLD"
```

```
20 END
```

- Praxis der Software-Entwicklung

- Implementierung

# Ziele

- Umsetzung eines vollständigen Softwareprojekts nach

der Stufe 2: Das erste Studienjahr  
program Hello(input, output)

- Pra  
Sof  
end.  
writeln('Hello World')

- Imp

- Umsetzung eines vollständigen Softwareprojekts nach der Stufe 4: Der erste Job

```
#include <stdio.h>
```

```
void main(void)
```

- Praxis  
Software

```
{  
char *message[] = {"Hello ", "World"};
```

```
int i;
```

- Implementierung

```
for(i = 0; i < 2; ++i)
```

```
printf("%s", message[i]);
```

```
printf("\n");
```

```
}
```

- Umsetzung eines vollständigen Softwareprojekts nach der Stufe 5: Erfahrener Software-Entwickler

- Praxis Software

- Implementierung

```
#include <iostream.h>
#include <string.h>
class string
{
private:
int size;
char *ptr;
public:
string() : size(0), ptr(new char('\0')) {};
string(const string &s) : size(s.size)
{
ptr = new char[size + 1];
strcpy(ptr, s.ptr);
};
string()
{
delete [] ptr;
};
friend ostream& operator <<(ostream &, const string &);
string& operator=(const char *);
ostream &operator<<(ostream &stream, const string &s)
{
return(stream << s.ptr);
};
};
```

```
string& string::operator=(const char *chrs)
{
if (this != &chrs)
{
delete[] ptr;
size = strlen(chrs);
ptr = new char[size + 1];
strcpy(ptr, chrs);
}
return *this;
}
int main(void)
{
string str;
str = "Hello World";
cout << str << endl;
return 0;
}
```

- Umsetzung eines vollständigen Softwareprojekts nach  
der Stufe 12: Mittleres Management

```
mail -s "Hello, world." bob@b12
```

- **Praxis** Bob, could you please write me a program  
**Software** that prints "Hello world." on the screen?

- **Imp** I need it by tomorrow.

# Ziele

- Umsetzung eines vollständigen Softwareprojekts nach dem aktuellen Stand der Softwaretechnik
- Praktische Umsetzung von Verfahren des Softwareentwurfs und der Qualitätssicherung
- Implementierungskompetenz
- Teamarbeit

# Ziele

- Umsetzung eines vollständigen Softwareprojekts nach dem aktuellen Stand der Softwaretechnik
- Praktische Umsetzung von Verfahren des Softwareentwurfs und der Qualitätssicherung
- Implementierungskompetenz
- Teamarbeit
- Präsentation

# Anforderungen

- **aktive Beteiligung** in **allen** Phasen

# Anforderungen

- **aktive Beteiligung** in **allen** Phasen
- Anwesenheit an **wöchentlichen Terminen** mit Betreuer

# Anforderungen

- **aktive Beteiligung** in **allen** Phasen
- Anwesenheit an **wöchentlichen Terminen** mit Betreuer
- Verwendung eines **Versionsverwaltungssystems**

# Anforderungen

- **aktive Beteiligung** in **allen** Phasen
- Anwesenheit an **wöchentlichen Terminen** mit Betreuer
- Verwendung eines **Versionsverwaltungssystems**
- Abgabe aller geforderten Dokumente
  - Vorabversion am Vortag des wöchentlichen Treffens
  - Fristgerechte Abgabe der finalen Versionen
  - Abgabe als **pdf**
  - Empfehlung:  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

# Anforderungen

- **aktive Beteiligung** in **allen** Phasen
- Anwesenheit an **wöchentlichen Terminen** mit Betreuer
- Verwendung eines **Versionsverwaltungssystems**
- Abgabe aller geforderten Dokumente
  - Vorabversion am Vortag des wöchentlichen Treffens
  - Fristgerechte Abgabe der finalen Versionen
  - Abgabe als **pdf**
  - Empfehlung:  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
- Kolloquium nach jeder Phase (Termine nach Vereinbarung)
  - Präsentation: Phasen-Ergebnisse, **Prüfungsgespräch**

# Anforderungen

- **aktive Beteiligung** in **allen** Phasen
- Anwesenheit an **wöchentlichen Terminen** mit Betreuer
- Verwendung eines **Versionsverwaltungssystems**
- Abgabe aller geforderten Dokumente
  - Vorabversion am Vortag des wöchentlichen Treffens
  - Fristgerechte Abgabe der finalen Versionen
  - Abgabe als **pdf**
  - Empfehlung:  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
- Kolloquium nach jeder Phase (Termine nach Vereinbarung)
  - Präsentation: Phasen-Ergebnisse, **Prüfungsgespräch**
- Gesamtnote ergibt sich aus
  - Qualität abgegebener Dokumente
  - Kolloquien
  - Qualität des Projekts

## 1. Organisatorisches

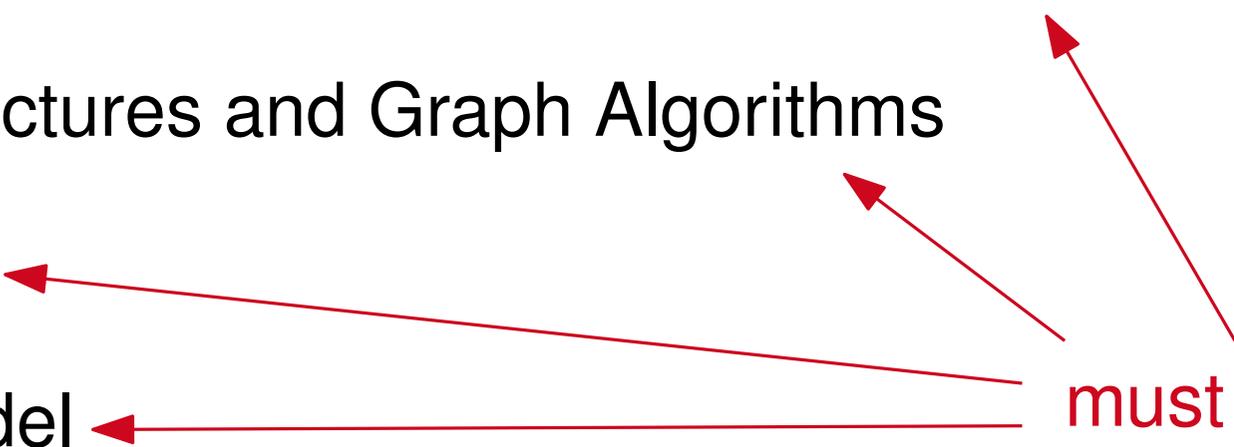
## 2. Ablauf

# Your Task

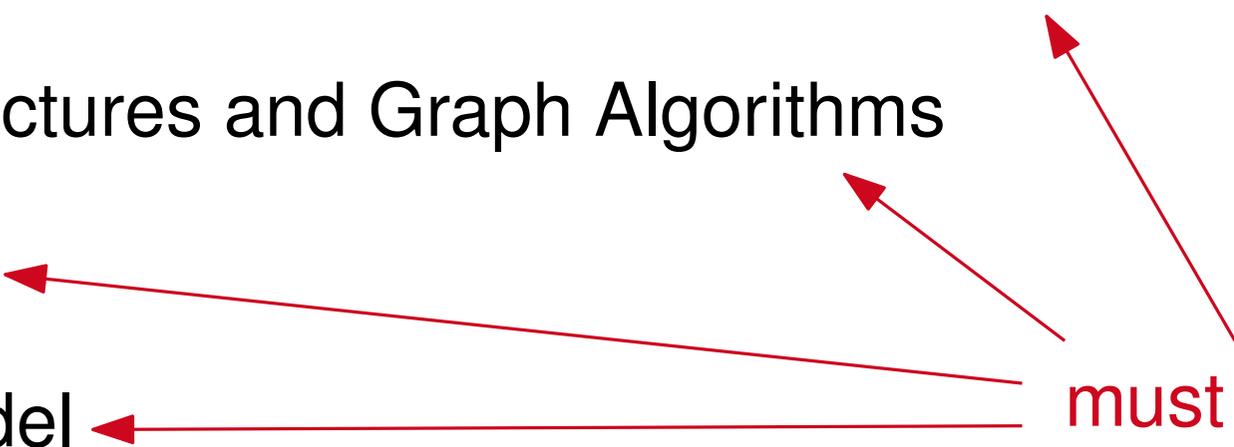
# Your Task

- Game Specification – general game class? metalanguage?
- Data Structures and Graph Algorithms
- Graphics
- Input Model

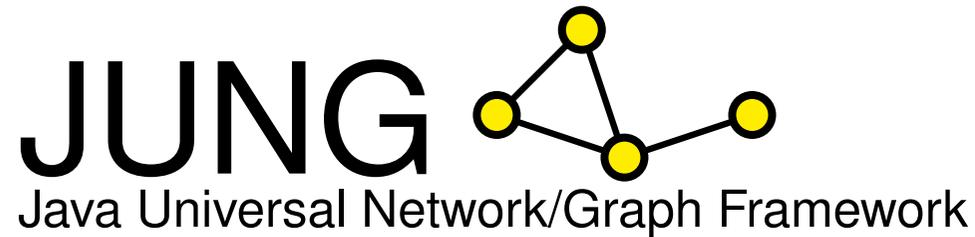
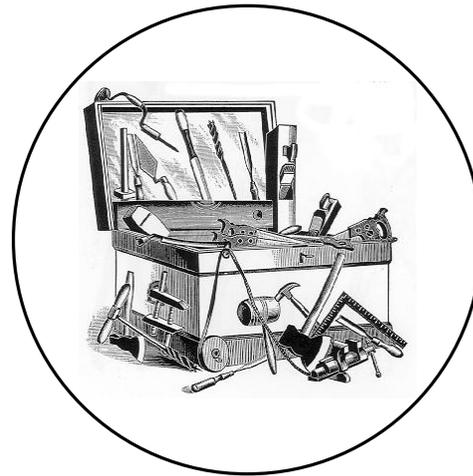
# Your Task

- Game Specification – general game class? metalanguage?
  - Data Structures and Graph Algorithms
  - Graphics
  - Input Model
- must have
- 

# Your Task

- Game Specification – general game class? metalanguage?
  - Data Structures and Graph Algorithms
  - Graphics
  - Input Model
  - Artificial Intelligence
  - Network
  - Tutorial
- must have
- 

# Werkzeuge



# Geplanter Ablauf

KW	Inhalt	Abgabedatum
16+17	Vorbesprechung	
18	Projektbeginn	
17–20	Pflichtenheft	20.05.2012
21–24	Entwurf	17.06.2012
25–28	Implementierung	15.07.2012
(2)	Klausurpause nach Wahl	
29–33	Validierung	19.08.2012
34	Interne Abnahme	
37	Abschlussveranstaltung	

Webseite:

■ [i11www.iti.uka.de](http://i11www.iti.uka.de) → *Lehre* → *Praxis der Softwareentwicklung*

- Vorbereitung:
  - Wöchentlichen Termin vereinbaren (direkt im Anschluß!)
  - Aufgabenstellung studieren
  - Versionsverwaltungssystem einrichten (GIT/SVN)
  - Daten sichten, Werkzeuge kennenlernen:
    - \* Rational Rose/Umbrello, eclipse, CodeCover, jUnit
    - \* JUNG, Java2D

- Vorbereitung:
  - Wöchentlichen Termin vereinbaren (direkt im Anschluß!)
  - Aufgabenstellung studieren
  - Versionsverwaltungssystem einrichten (GIT/SVN)
  - Daten sichten, Werkzeuge kennenlernen:
    - \* Rational Rose/Umbrello, eclipse, CodeCover, junit
    - \* JUNG, Java2D
  
- Ablauf der Phasen:
  - Wöchentliche Treffen mit Betreuer
  - Vor jeder Phase: Schriftliche Aufgabenvereinbarung
  - Phase endet mit Abgabe des jeweiligen Dokuments

- Vorbereitung:
  - Wöchentlichen Termin vereinbaren (direkt im Anschluß!)
  - Aufgabenstellung studieren
  - Versionsverwaltungssystem einrichten (GIT/SVN)
  - Daten sichten, Werkzeuge kennenlernen:
    - \* Rational Rose/Umbrello, eclipse, CodeCover, junit
    - \* JUNG, Java2D
- Ablauf der Phasen:
  - Wöchentliche Treffen mit Betreuer
  - Vor jeder Phase: Schriftliche Aufgabenvereinbarung
  - Phase endet mit Abgabe des jeweiligen Dokuments

Arbeitsaufwand ca. **2 Tage/Woche**