



Greenfoot

Axel Schmolitzky

Arbeitsbereich Softwaretechnik
Department Informatik, Universität Hamburg



Agenda

- ◆ Vorstellungsrunde
- ◆ Hintergrund von Greenfoot: BlueJ
- ◆ Greenfoot Demo
- ◆ Greenfoot theoretisch
- ◆ Greenfoot Szenarien
- ◆ Greenfoot interaktiv
- ◆ Greenfoot praktisch
- ◆ Abschluss, Feedbackrunde





Vorstellungsrunde

- ◆ Name?
- ◆ Hamburger/in oder zugereist?
- ◆ Erfahrung mit der Lehre objektorientierter Programmierung?
- ◆ Erfahrung mit BlueJ?
- ◆ Warum ausgerechnet hier?



Hintergrund: BlueJ



Vorteile:

- ◆ Einfache Entwicklungsumgebung
- ◆ Direkte Interaktion mit Klassen und Objekten
- ◆ Visualisierung der statischen Struktur eines Projektes
- ◆ Große Verbreitung (mehrere hundert Einrichtungen weltweit)
- ◆ Gute Diskussionsliste
- ◆ Frei verfügbar
- ◆ Gutes Lehrbuch vorhanden (Barnes/Kölling)





Hintergrund: BlueJ



Nachteile:

- ◆ Ohne Lehrbuch nicht leicht einsetzbar
- ◆ Kaum Visualisierung von (interagierenden) Objekten
- ◆ Trennung zwischen BlueJ und Java unscharf



Greenfoot

Einstiegsdemo



Greenfoot theoretisch

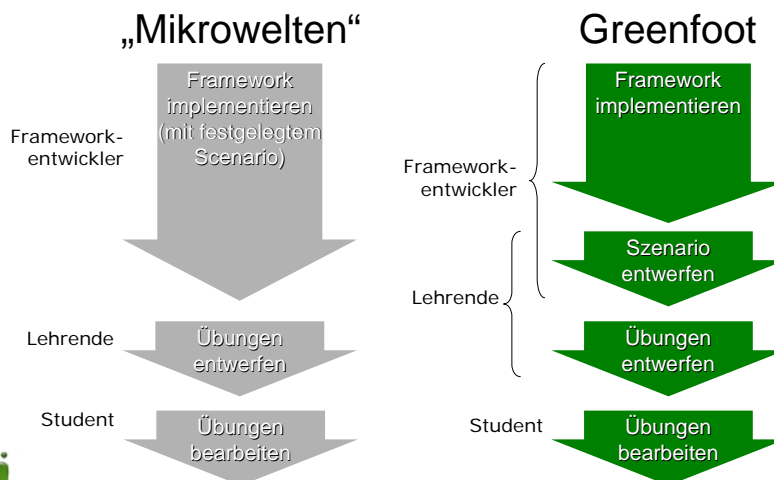
- ◆ Greenfoot ist keine Mikrowelt!

Greenfoot ist...

- ◆ ein Meta-Framework für Mikrowelten
 - Basiert auf der Idee von *Discrete Event Simulation*
 - Ermöglicht flexible Szenarien
- ◆ animierte Ausführung
- ◆ Interaktion mit visualisierten Objekten
- ◆ eine vollwertige IDE (mit Editor, Debugger, etc.)



Rollenverteilung





Entkoppelte Szenarien

- ◆ Die Szenarien sind vom Framework zur Animation entkoppelt.
- ◆ Dies führt zu...
 - einfacherem Entwurf neuer Szenarien
 - mehr Szenarien
 - individuellen Szenarien (für spezielle Zielgruppen)



Greenfoot Szenarien

weitere Demos





Greenfoot interaktiv

- ◆ Grundidee: **Aufgaben gemeinsam lösen.**
- ◆ Ich zeige ein Szenario und stelle Aufgaben darin.
- ◆ Lösungsvorschläge kommen aus dem Plenum.
- ◆ Ich moderiere die Diskussion.
- ◆ Ich implementiere für alle sichtbar das, was entschieden wird.



1. Aufgabe

- ◆ Wombats sollen sich nicht immer nach links drehen, wenn sie auf ein Hindernis treffen.
- ◆ Stattdessen sollen sie sich in eine zufällig gewählte Richtung wenden.
- ◆ Was ist wo zu tun?





2. Aufgabe

- ◆ Neben Wombats und Blättern sollen auch Steine unser Szenario bevölkern.
- ◆ Es soll deshalb eine weitere Klasse für Steine definiert werden.



3. Aufgabe

- ◆ Wombats sollen nicht einfach über Steine hinweg laufen!



Zwischenfazit

- ◆ Bei entsprechend gewählten Szenarien können von sehr einfachen Problemen bis hin zu „harten Nüssen“ Aufgaben gestellt werden.
- ◆ Diese werden im Quelltext der beteiligten Akteursklassen gelöst.
- ◆ Teilweise müssen neue Akteursklassen definiert werden.
- ◆ Es können auch in der Welt-Klasse Methoden implementiert werden.
- ◆ Es stehen die kompletten Möglichkeiten von Java und BlueJ zur Verfügung.



Möglichkeiten für Lehrende

- ◆ Lehrende wählen/schreiben/teilen Szenarien
- ◆ Typisches Szenario:
 - Robot demo: 4 Klassen, 260 Zeilen Quelltext (gesamt)
 - Ants: 6 Klassen, 600 Zeilen
- ◆ Kontext ist flexibel wählbar

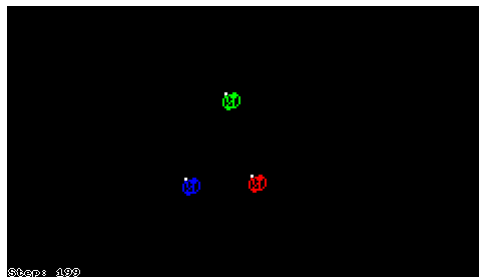
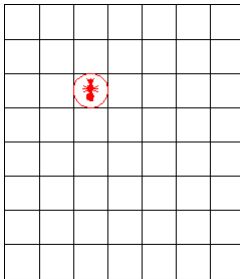


Greenfoot praktisch

- ◆ Selbst aktiv werden: ein eigenes Szenario bauen!
- ◆ Einfaches Szenario: **Langtons Ameise**
 - Eine Ameise bewegt sich auf einer zweidimensionalen Ebene mit schwarzen und weißen Feldern.
 - Wenn sie auf einem **weißen** Feld steht, dreht sie sich um 90 Grad nach **links** und geht ein Feld vor.
 - Wenn sie auf einem **schwarzen** Feld steht, dreht sie sich um 90 Grad nach **rechts** und geht ein Feld vor.
 - Nachdem die Ameise ein Feld verlassen hat, wechselt dieses seine Farbe.



Langtons Ameise





Langtons Ameise: Umsetzen!

Zeit zum Basteln: 20 bis 30 Minuten



Entwurfsziele für OOP-Lernumgebungen

Visualisierung von
Objektzustand und
Verhalten

integrierte Werkzeuge

direkte Interaktion /
Experimentieren

akkurate
Repräsentation
objektorientierter
Konzepte

Standard-
Programmierstil

flexible Szenarien





Turtle Graphics

Visualisierung von
Objektzustand und
Verhalten

integrierte Werkzeuge

direkte Interaktion /
Experimentieren

akkurate
Repräsentation
objektorientierter
Konzepte

Standard-
Programmierstil

flexible Szenarien



Karel Family

Visualisierung von
Objektzustand und
Verhalten

integrierte Werkzeuge

direkte Interaktion /
Experimentieren

akkurate
Repräsentation
objektorientierter
Konzepte

Standard-
Programmierstil

flexible Szenarien



BlueJ

Visualisierung von
Objektzustand und
Verhalten

integrierte Werkzeuge

direkte Interaktion /
Experimentieren

akkurate
Repräsentation
objektorientierter
Konzepte

Standard-
Programmierstil

flexible Szenarien



Greenfoot

Visualisierung von
Objektzustand und
Verhalten

integrierte Werkzeuge

direkte Interaktion /
Experimentieren

akkurate
Repräsentation
objektorientierter
Konzepte

Standard-
Programmierstil

flexible Szenarien



Zusammenfassung

- ◆ Zielt auf Sek. II und Einstiegskurse an Hochschulen
- ◆ Sehr visuell orientiert
- ◆ Ermöglicht interaktives Experimentieren
- ◆ Flexibel
- ◆ Unterstützung für Lehrende
- ◆ Leichter Übergang zu BlueJ



Abschluss

- ◆ Feedbackrunde
 - Diskussion: Einschätzung von Greenfoot
 - Beurteilung dieses Workshops

