Proseminar Softwaretechnik

Game Programming Patterns

Florian Ege Thomas Witte

25. Oktober 2017

Lernziele

- Wissenschaftliches Arbeiten
- Schreiben einer Ausarbeitung
- Präsentation und Vortrag
- Verständnis für Software-Architekturpattern entwickeln

Vorbedingungen

- Proseminar (Unbenotet)
- Informatik Studiengänge
- Kenntnis englischer Sprache
- **Motivation**

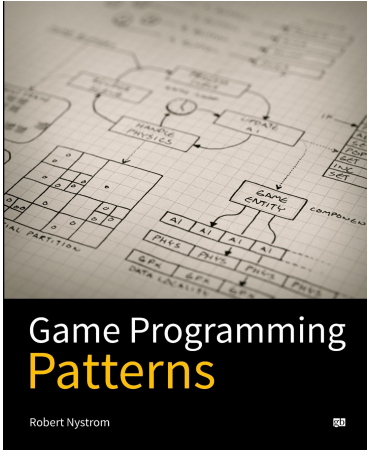
Anforderungen

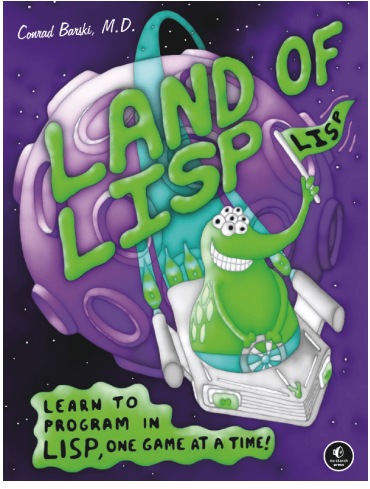
- Einhaltung der Deadlines
- Gute Ausarbeitung (ca. 5 Seiten)
- Motivierte Präsentation (20 Minuten)
- körperliche sowie geistige Anwesenheit

Hinweise

- Verbindliche Anmeldung mit Abgabe des ersten Meilensteins
- Abbruch danach $\Rightarrow$ nicht bestanden

Themen


Game Programming
Patterns
Robert Nystrom

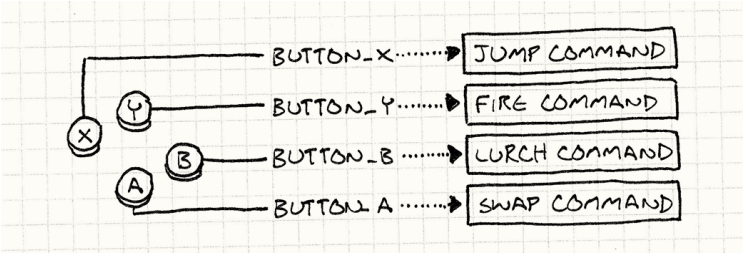

Conrad Barski, M.D.
LAND OF LISP
LISP
LEARN TO PROGRAM IN LISP, ONE GAME AT A TIME!

Themen

- Command / Lambda
- Prototype
- Singleton
- State
- Sequencing Patterns
- Component
- Optimization Patterns
- Generic Programming
- Functional Style
- Lazy Programming
- Domain-Specific Languages
- Bytecode

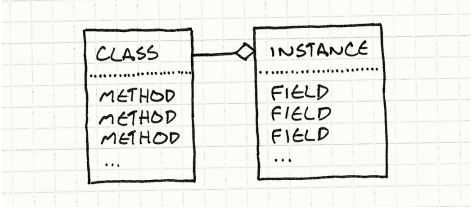
Command / Lambda

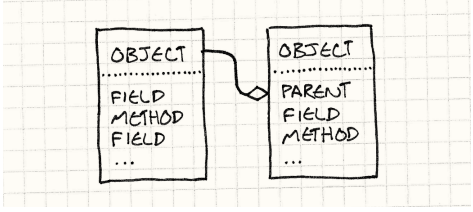
- Befehle als Objekt auffassen und z.B. in Warteschlangen zu verwalten.



Prototype

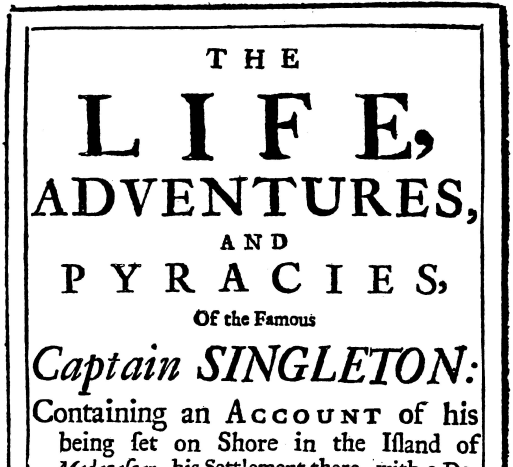
- Objekte nicht aus einem Klassentemplate instanziieren, sondern als Klon eines Prototypen.





Singleton

- Einzigartige Ressourcen – und den Zugriff darauf – durch ein globales Objekt verwalten.



THE

LIFE,

ADVENTURES,

AND

PYRACIES,

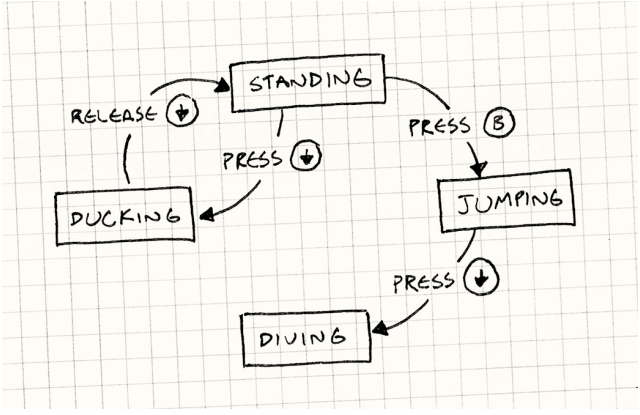
Of the Famous

Captain SINGLETON:

Containing an ACCOUNT of his being ſet on Shore in the Iſland of

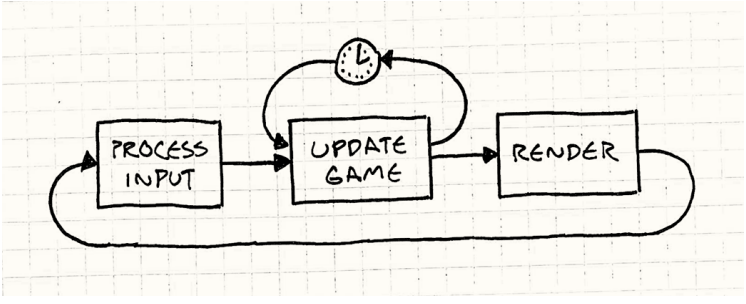
State

- Komplexes Verhalten durch Zustandswechsel des Systems ermöglichen.



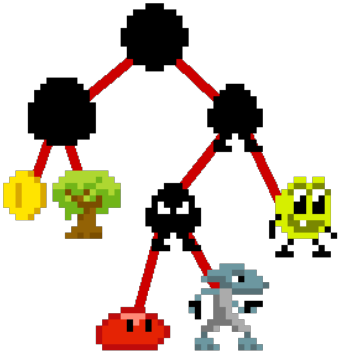
Sequencing Patterns

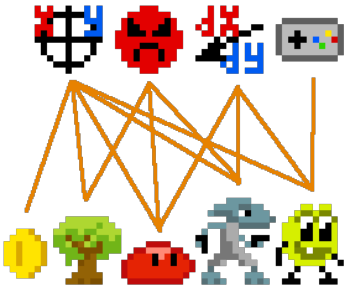
- „Gleichzeitige“ Änderungen/Aktionen innerhalb eines einzelnen Threads ermöglichen.



Component

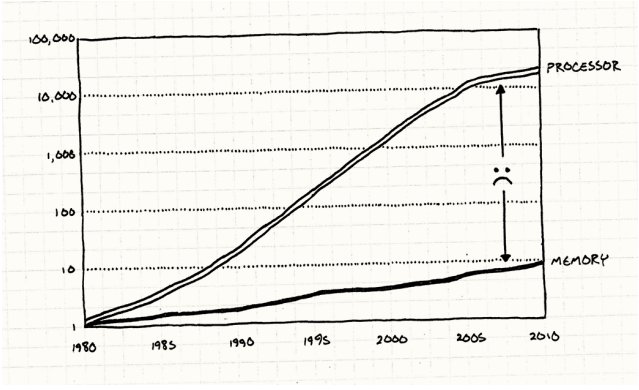
- Objekte erstellen, die viele Teilbereiche der Anwendung betreffen ohne starke Kopplung.





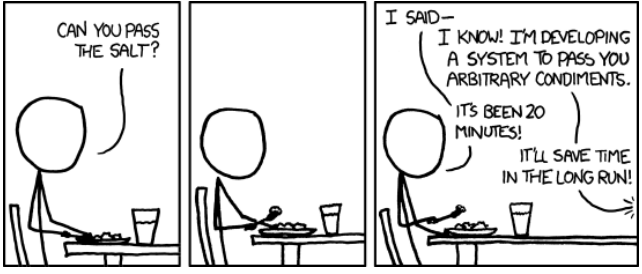
Optimization Patterns

- Objektzugriffe und Speicher optimieren um die Ausführungsgeschwindigkeit zu erhöhen.



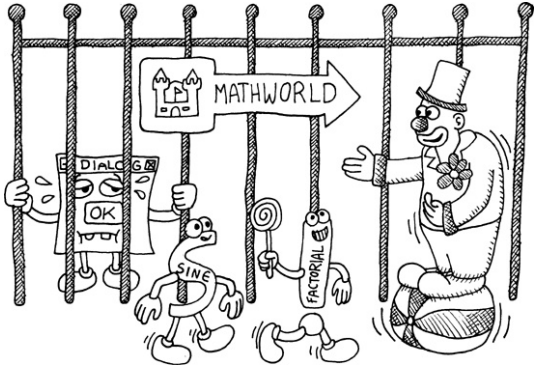
Generic Programming

- Funktionen werden allgemein entworfen, um für mehrere Datentypen und -Strukturen verwendet werden zu können.



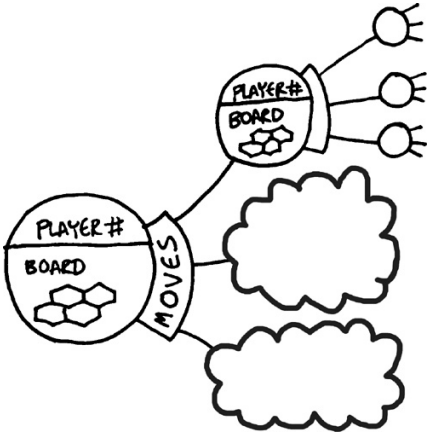
Functional Style

- "a style of programming where we write all of our code using functions."



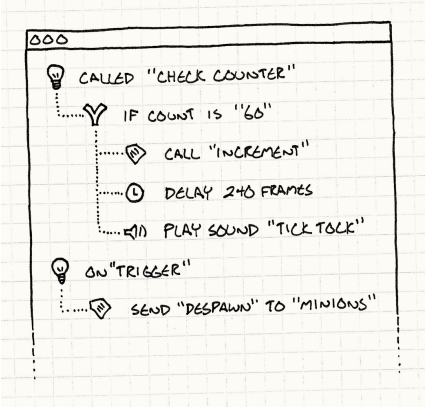
Lazy Programming / Dirty Flag

- Nur tatsächlich benötigte Werte berechnen um Speicher und Zeit zu sparen.



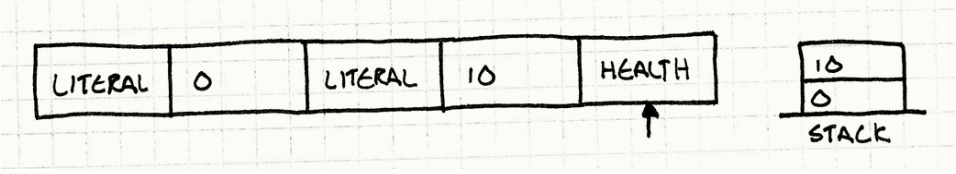
Domain-Specific Languages

- Spezialisierte Sprachen entwickeln um Teilprobleme leichter zu lösen.



Bytecode

- Maschinennahen Bytecode verwenden um die Performance gegenüber Interpretern zu steigern.



Themen

- Command / Lambda
- Prototype
- Singleton
- State
- Sequencing Patterns
- Component
- Optimization Patterns
- Generic Programming
- Functional Style
- Lazy Programming
- Domain-Specific Languages
- Bytecode

Zeitplan

25.10.2017	Einführungsveranstaltung
02.11.2017	Vortrag wiss. Arbeit / Einführung LaTeX
08.11.2017	Lightning Talks / **Abgabe** Gliederung
19.11.2017	**Abgabe** 1. Version
03.12.2017	**Abgabe** 2. Version (vollständig)
10.12.2017	**Abgabe** Peer Review
07.01.2018	**Abgabe** 3. Version
TBD	Vortrag Präsentationstechniken
28.01.2018	**Abgabe** des Entwurfs der Folien
04.02.2018	**Abgabe** finale Version
TBD	Vorträge

Wie geht's weiter?

- Vortrag wiss. Arbeiten, Mi 2.11. 16:00
- Einarbeitung bis 8.11.
- Gliederung (Überschriften, Stichpunkte, Quellen)
- Lightning-Talks (3 min, 1 Folie)
- Verbindliche Anmeldung