

BeepBoop - The Robot Game

Pawel Rasch, Tim Runge

12. Juli 2018

- Strategy game
- Collect resources
- Build robots
- Program Robots to collect resources

Voraussetzungen:

- . Projekt-Requirements nicht vollständig spezifiziert
 - . Großer Umfang der Domain
-

Aufteilung:

- . detaillierte Aufteilung in Tasks/Subtasks - google-spreadsheet
- . Vorantreibung der Umsetzung in Sprints
- . häufige Wechsel der Zuständigkeit

Design-Paradigmen:

. Rapid-Prototyping

explorativ, experimentell, evolutionär

... technische und fachliche" Klassen möglichst weit modularisieren und ausdifferenzieren in der Explorationsphase (evtl. auch nur intern, innerhalb einer Klasse)

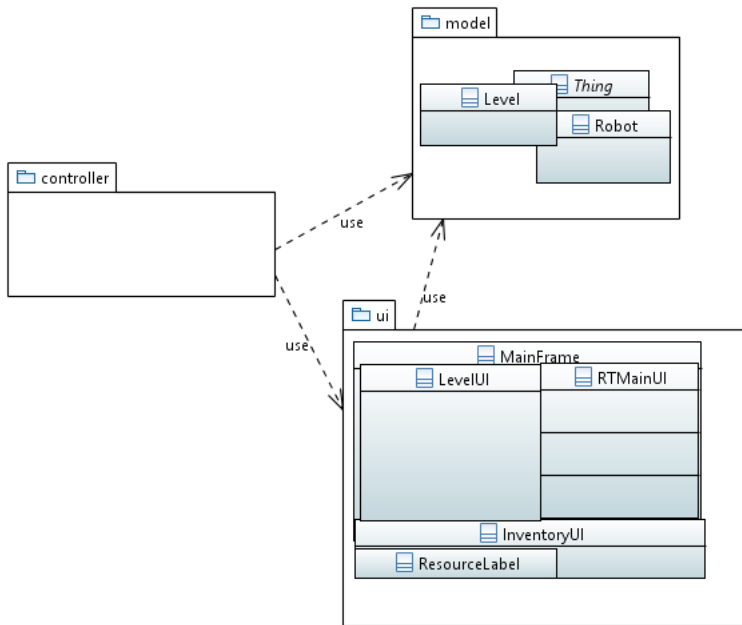
. DomainDrivenDesign

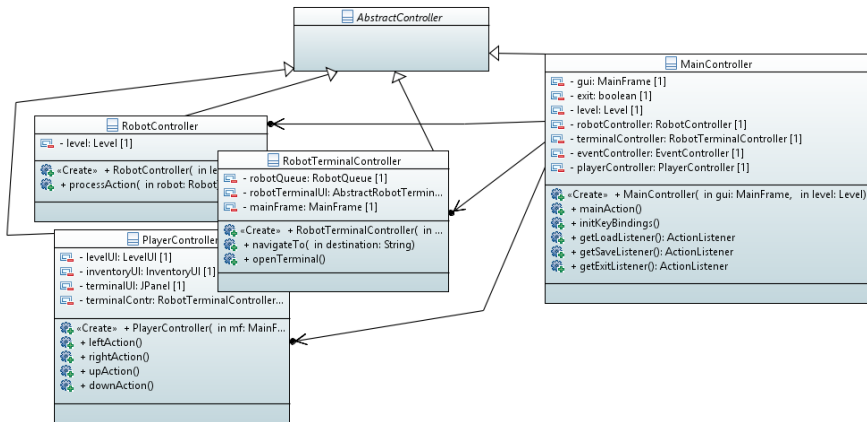
... möglichst differenziertes Model, wird im Verlauf des Prototypings Busammengefaltet"

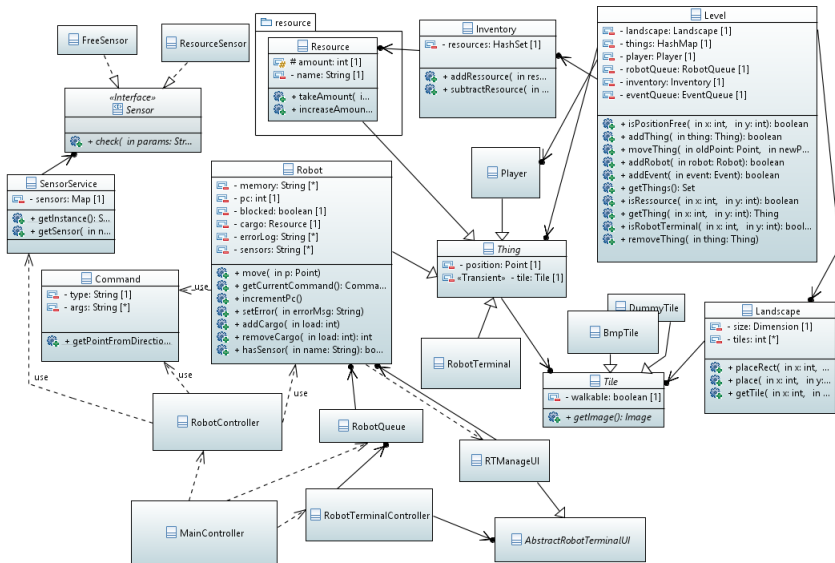
. MVC - Umsetzung oft durch Refactorings

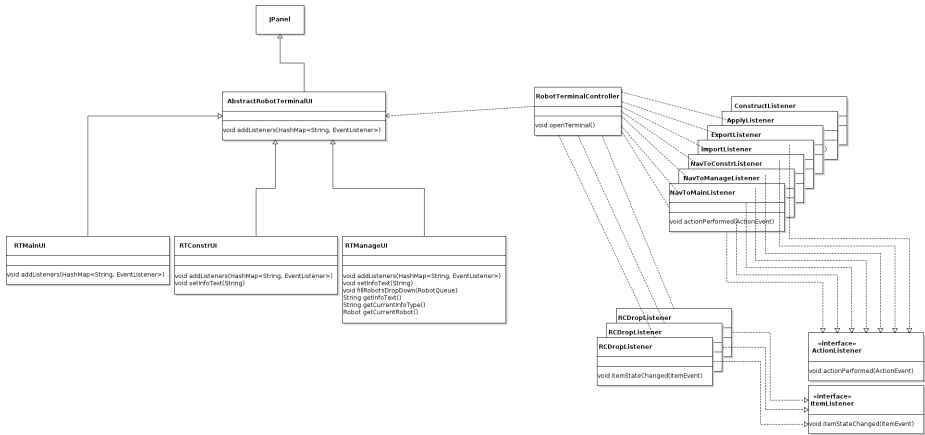
. TestDrivenDesign

verworfen









Probleme:

- Listener-Struktur / Swing-Event-thread (Lösung: Mutexe? Strukturänderung?)
- Umfang - zu groß, Game Design nicht abgeschlossen
- Test First
- Aufgabenteilung: ständig wechselnde Zuständigkeiten

Algorithmen:

- Command parsing und processing
- ...

Design-Pattern:

- Observer-Pattern
- Factory-Pattern
- Singleton-Pattern
- Strategy-Pattern - ausgebaut, unpassend

Selbstkritik:

- zu anspruchsvoll - unangemessener Umfang
- feature-f**king
- division of concerns

