



# Programmierpraktikum 2007

Einführungsveranstaltung

Montag, 4 Apr 7

[ullrich@informatik.uni-koeln.de](mailto:ullrich@informatik.uni-koeln.de)



## Features

### 1. Projektweite Features

1.1. Aufbau der Welt

1.2. Items

1.3. Etc

### 2. Features des Clients

2.1. GUI

2.2. Levelverwaltung

2.3. Kommunikation

Nächste Sitzung (11 Apr 7)

- Features Server

- Gruppenbildung

- Aufgabe für Meilenstein 1



# 1. Projektweite Features

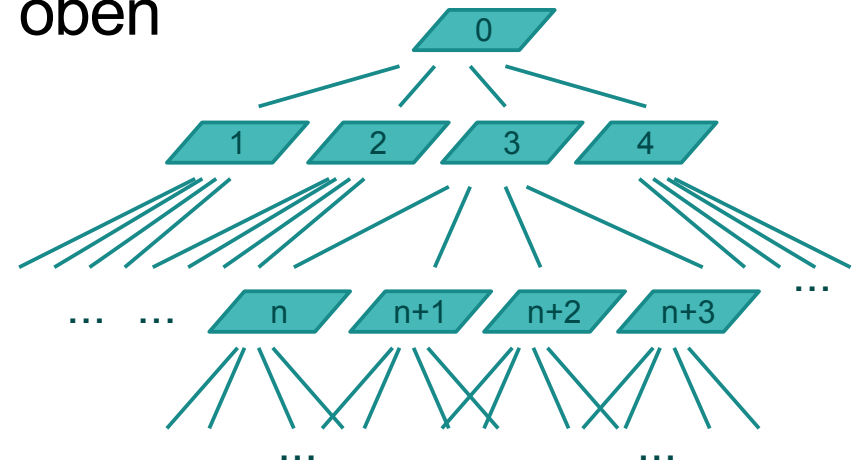
## 1.1. Aufbau der Welt

- Levelbaum -
- Aufbau der Levels -
- Spielcharakter -



## Aufbau der Spielwelt – Levelbaum

- › Level 0 auf Ebene 0 (= Startpunkt für neue Spieler)
- › Vervierfachung auf jeder Ebene
- › Vier Leitern nach unten, eine nach oben
- › Je tiefer, umso schwieriger
  - › Monster gefährlicher
  - › Funde wertvoller
  - › Goldhaufen größer
- › Auf Ebene 0 oder 1: „Kneipe“ als Treffpunkt





## Aufbau der einzelnen Level – Bodenkacheln

- Aufgebaut aus 1x1-Meter „Fliesen“
- 2D-Feld von  $m \cdot (2n+1)$  Bodenfliesen, zB 63x63

Warum  $(2n+1)$ ? → Irrgärten lassen sich mit ungerader Anzahl Kacheln besser aufziehen (s. Literatur)

Warum  $m \cdot \dots$ ? → Level wird abwechslungsreicher durch mehrere Bereiche/Räume



## Aufbau der einzelnen Level – Bodenkacheln II

Der Boden der einzelnen Level bleibt jeweils unveränderlich

Bestehend aus:

- Wänden
- Freier Boden
- Wasserflächen
- etc

→ Bilden Struktur des Levels



## Aufbau der einzelnen Level – Bodenkacheln III

Darauf befinden sich

- › Die eigene Spielfigur („Charakter“)
- › Gegenstände („Items“)
- › Spielfiguren anderer Spieler
- › Monster, andere Non Player Characters (NPCs)
- › Türen, offen/geschlossen
- › Treppen/Leitern, auf/ab

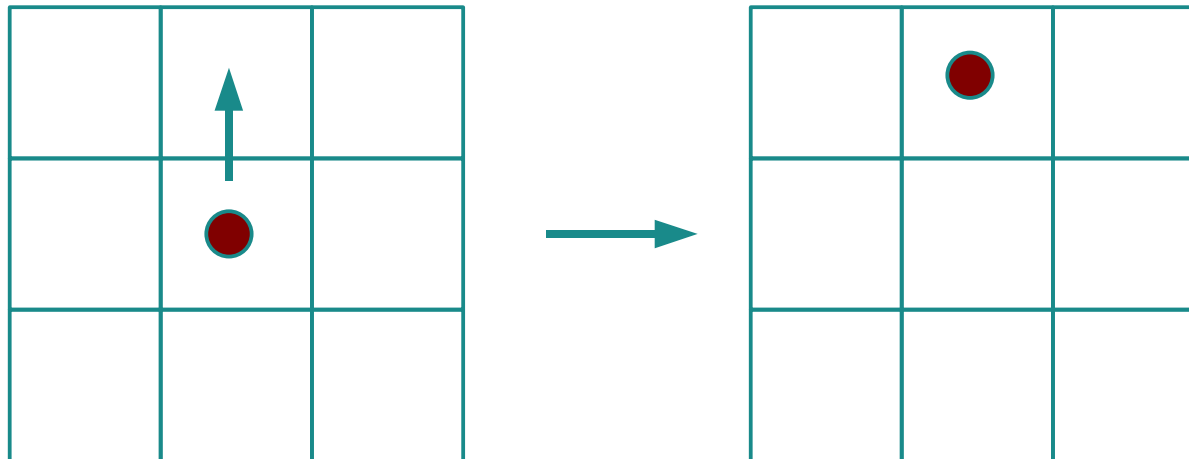


## Aufbau der einzelnen Level – Bodenkacheln IV

Bewegung der Figuren erfolgt (logisch) kachelweise

→ Kacheln sind für uns atomisch

Darstellerisch ist sanftes Hinübergleiten machbar, falls gewünscht.





## Spieler/Charakter

Allgemeine Attribute:

- ID
- Name
- Graphik
- Inventar (Liste von Gegenständen)

Details: RPG-Paper



## Spieler/Charakter II

Eigentliche Charakterwerte:

- Erfahrungspunkte (ExPts, Gewinn durch Schlagen von Monstern, Erfüllung von Aufträgen)
- Lebenspunkte (HitPts)
- Stärke
- Intelligenz
- ...
- Völlegefühl



## Spieler/Charakter III

Charakter bleibt bestehen:

- Beim Abmelden des Clients von Server in DB gespeichert und aus der Welt entfernt
- Beim Wiedereinloggen an alter Stelle wieder eingeführt
- Falls von Monster besiegt oder verhungert, etc: Löschen

→ Nicht unbedingt beste Art, aber recht einfach



# 1. Projektweite Features

## 1.2. Items



## Items – Allgemein

- › Items sind
  - › Objekte der Spielewelt,
  - › die der Spieler mitnehmen, benutzen, essen, etc, kann.
  
- › Items werden vom Levelgenerator verteilt
- › Verschiedene Klassen von Items liegen herum

Beschreibung einiger Items → zuerst ausführlich, dann knapper



## Items – Nahrung

Typ	Nährwert
<i>Kompaktration</i>	200
<i>Obst</i>	50
<i>Pizza</i>	300

„use“ <item>:  $v\ddot{o}lle = v\ddot{o}lle + n\ddot{a}hrwert$

Spielersschritt:  $v\ddot{o}lle = v\ddot{o}lle - 1$

>500: „Überfressen!“, Nichts geht mehr!

Bei 0: „Hunger!“

Bei -100: Bauchkrämpfe: HitPts - -

Bei -200: Leider verhungert! †



## Items – Gold

Zweitwichtigster Itemtyp:

*Gold!*

Goldhaufen geeigneten Werts

- › liegen herum (vom Levelgenerator verteilt),
- › werden von Monstern nach Exitus hinterlassen.

→ Nicht logisch hinterfragbar, da Tradition!



## Items – Waffen/Rüstungen

Einige Eigenschaften:

- Typ (Schwert, Handgranate, etc – Hängt vom Spieluniversum ab)
- Schadensstärke
- Reichweite (Nah-/Fernwaffen)
- Zustand (Nutzt bei Gebrauch ab; evtl. in Werkstatt richten lassen?)
- ...

Spieler kann mehrere Waffen/Rüstungen dabei haben, eine ist ausgezeichnet als aktuell verwandt

Details: RPG-Paper



## Items – Weitere Typen

Eher kriegerisch:

- › Helme/Hüte
- › Stiefel
- › Schilde

Eher zivil:

- › Lampen/Fackeln (für dunkle Räume)
- › Bücher
- › Edelsteine



## Items – Weitere Typen

Zauberkrum:

- Zaubertränke (Heilung, Gift, Gestank, Superkräfte, etc)
- Zaubersprüche (Licht, Teleportation, etc)

Details: RPG-Paper

Items, aber zur Levelstruktur gehörig:

- Türen offen/geschlossen
- Leitern

**Entwarnung:** Wenn Itemsystem erstmal implementiert, sind zusätzliche Typen simpel integrierbar, da sie sich im Groben alle gleich verhalten.



## Items – Weitere Typen: Schätze

Tief, tief unten verstecken sich große Schätze:

- Antwort auf die Frage „ $P \stackrel{?}{=} NP$ “
- Lösung für das SAT-Problem in  $O(N)$
- Buch: „Die Promotion in 30 Tagen“



# 1. Projektweite Features

## 1.3. Etc

- Monster -
- Kämpfe -
- Zauberei -
- Aufträge -



## Monster

- Typen und Eigenschaften
- Verhalten



## Monster – Typen und Eigenschaften

Klassische Monstertypen:

- › Dicke Ratten
- › Daleks
- › Gefräßige Plapperkäfer von Traal
- › Drachen, etc, etc → abhängig vom Spieluniversum

Eigenschaften:

- › Lebenspunkte
- › Stärke
- › Mitgeführtes Gold

Details: RPG-Paper



## Monster – Typen und Eigenschaften II

Wird ein Monster besiegt, dann

- › bekommt der Spieler Erfahrungspunkte dazu und
- › das Monster lässt Goldhaufen fallen.

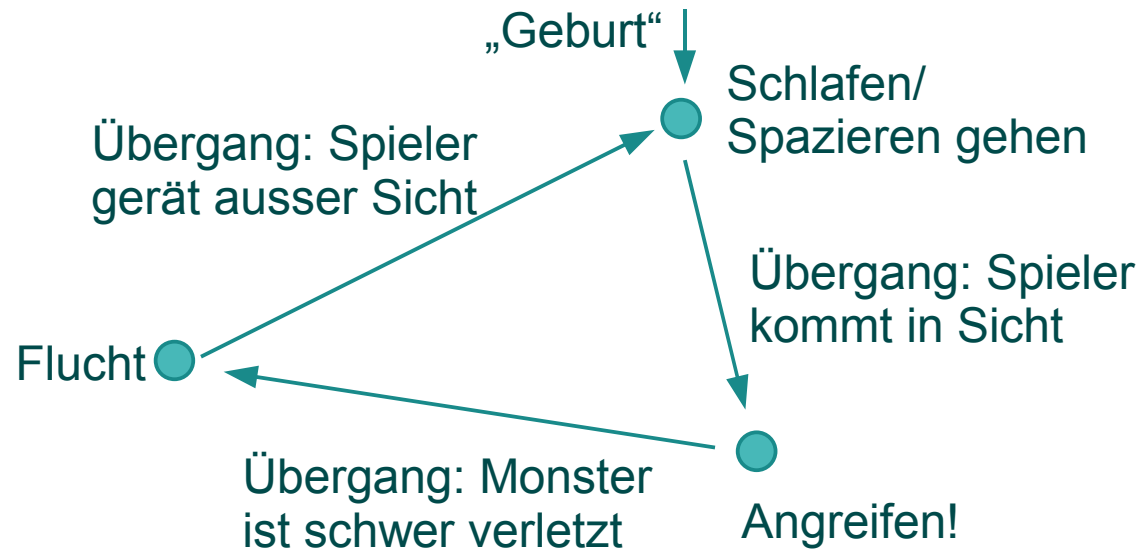


## Monster – Verhalten

Das Verhalten der Monster wird auf zwei Ebenen bestimmt:

- › Strategische Ebene
- › Taktische Ebene

Strategische Ebene: Abbildung ähnlich wie DEA: *state machine*





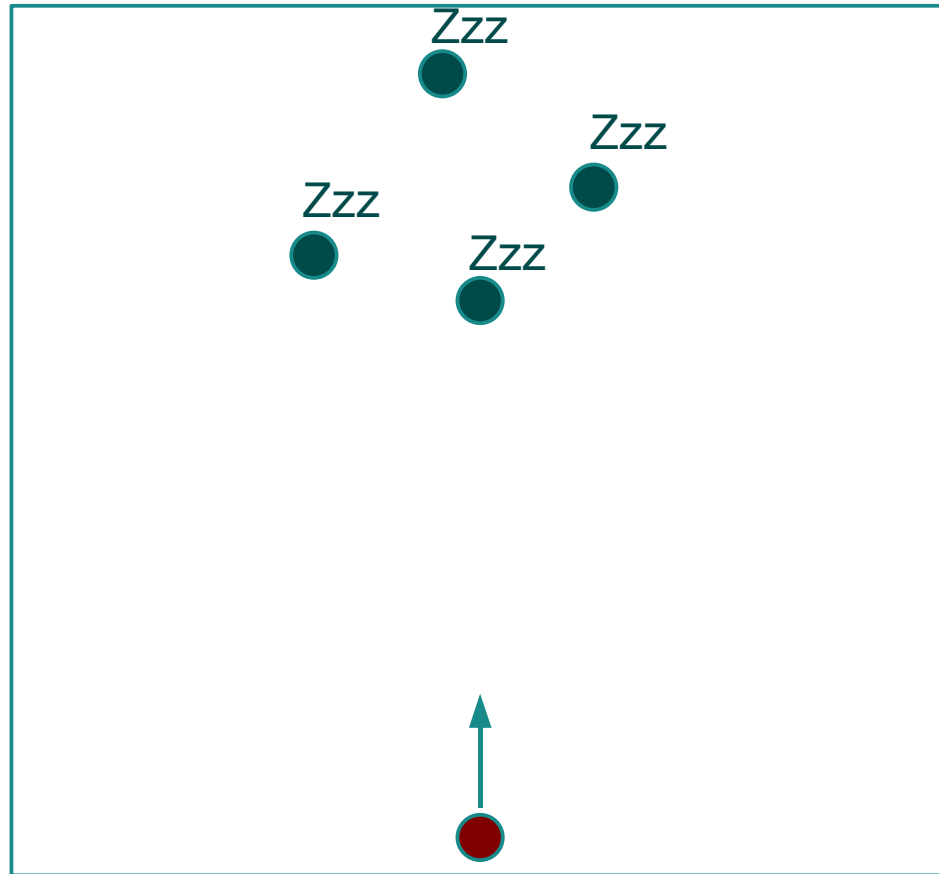
## Monster – Verhalten II

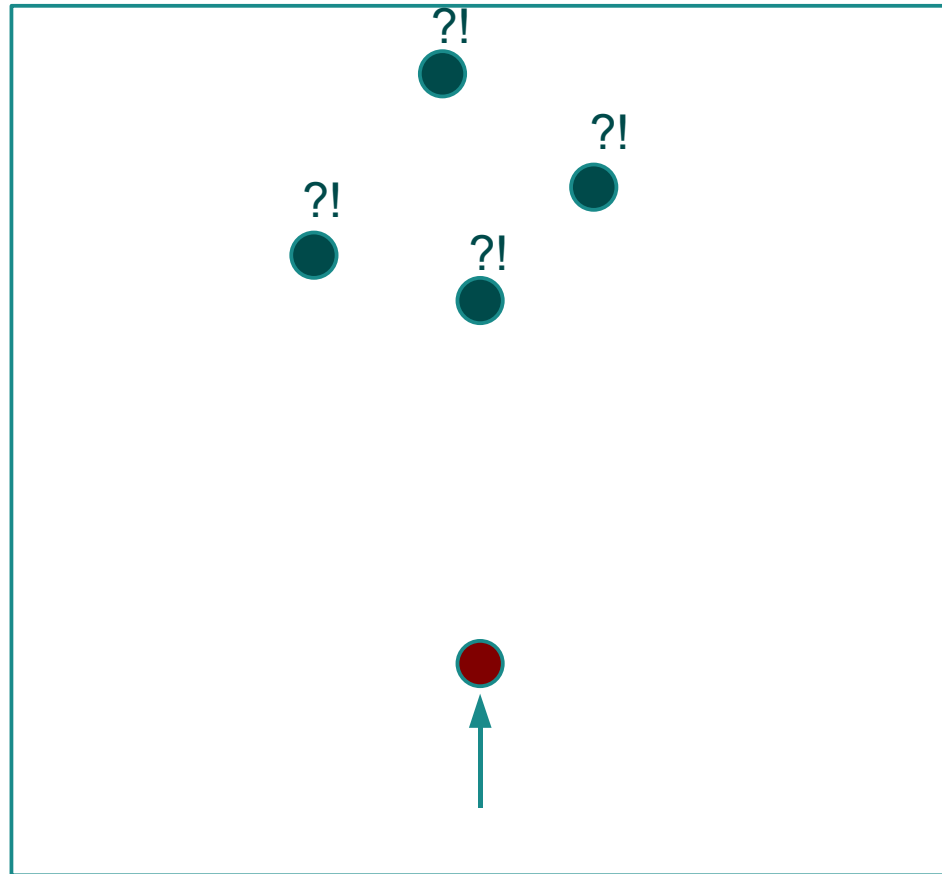
Taktische Ebene: Welche Algorithmen verwenden wir in welcher Situation?

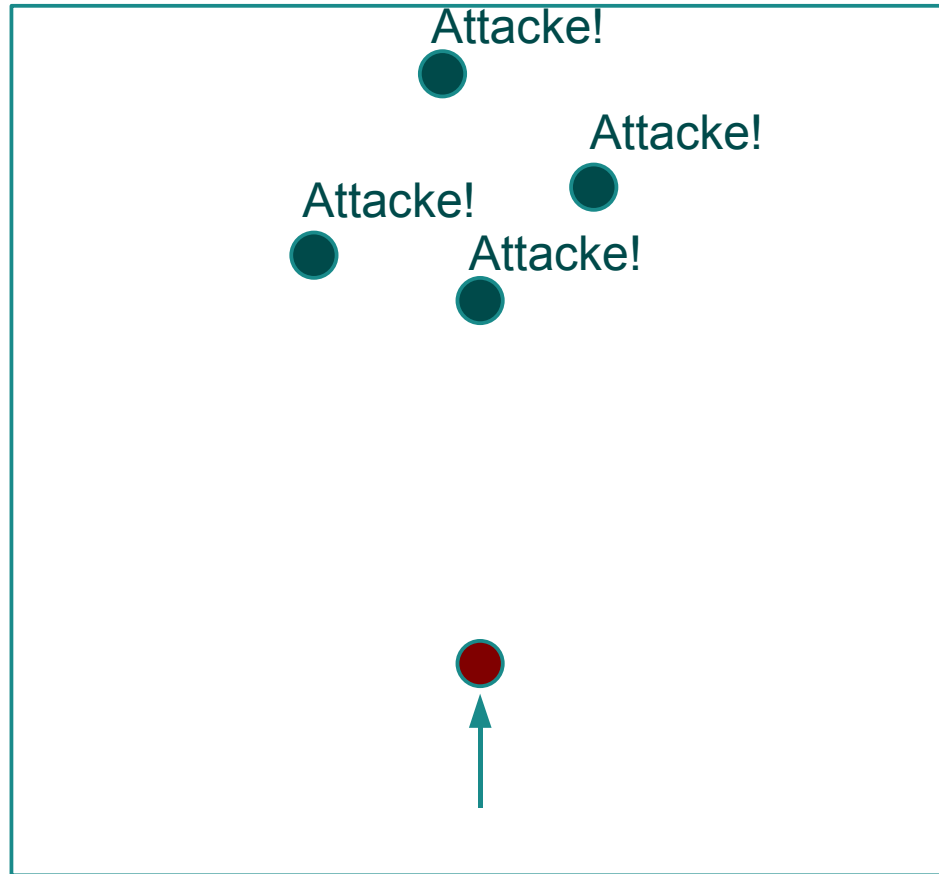
- › Spazieren gehen: Zufälliges Umherlaufen → kein Algorithmus
- › Panik: Blind vom Spieler weg laufen → in Ecke treiben möglich  
→ kein Algorithmus
- › Angreifen: Finden eines kürzesten Weges → Dijkstra/A\*-Algorithmus

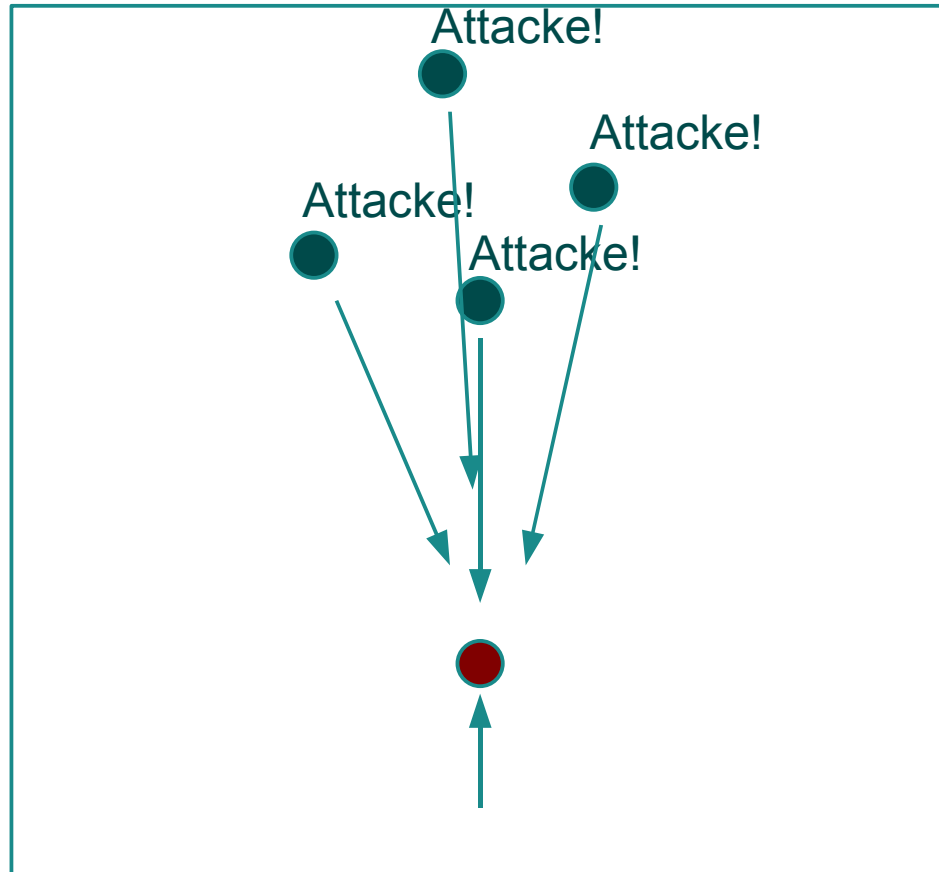
Einfaches Schema, aber einige Sonderfälle müssen beachtet werden.

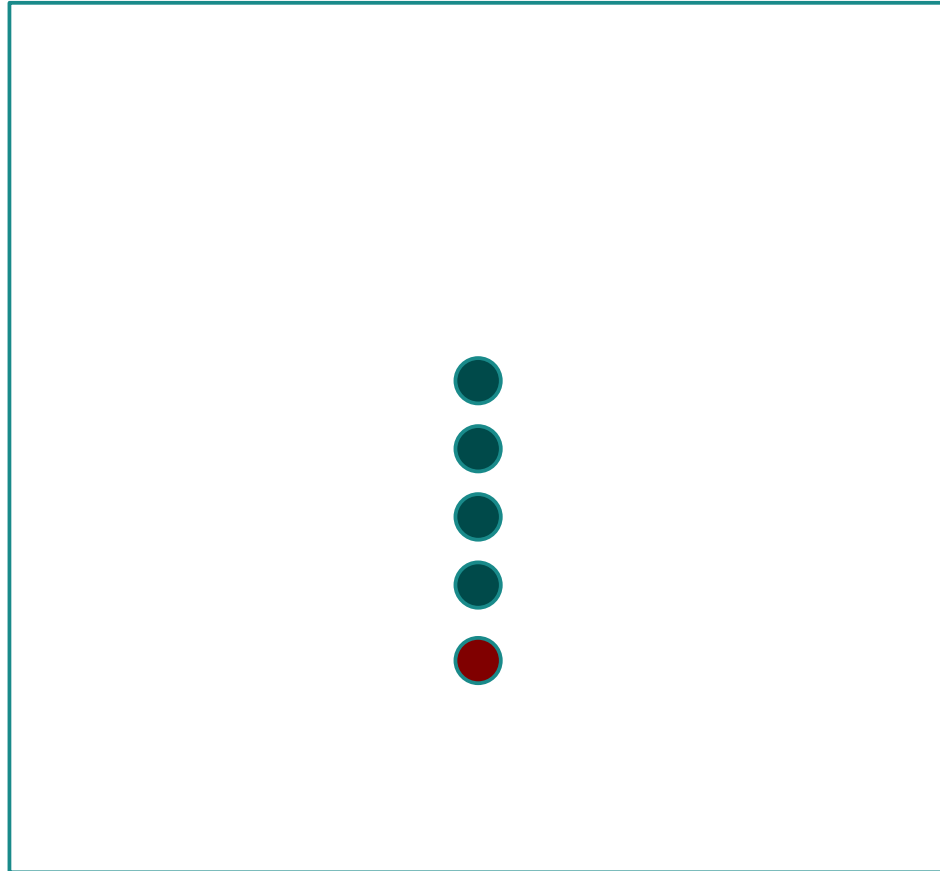
Sonst: Dämliches Verhalten.







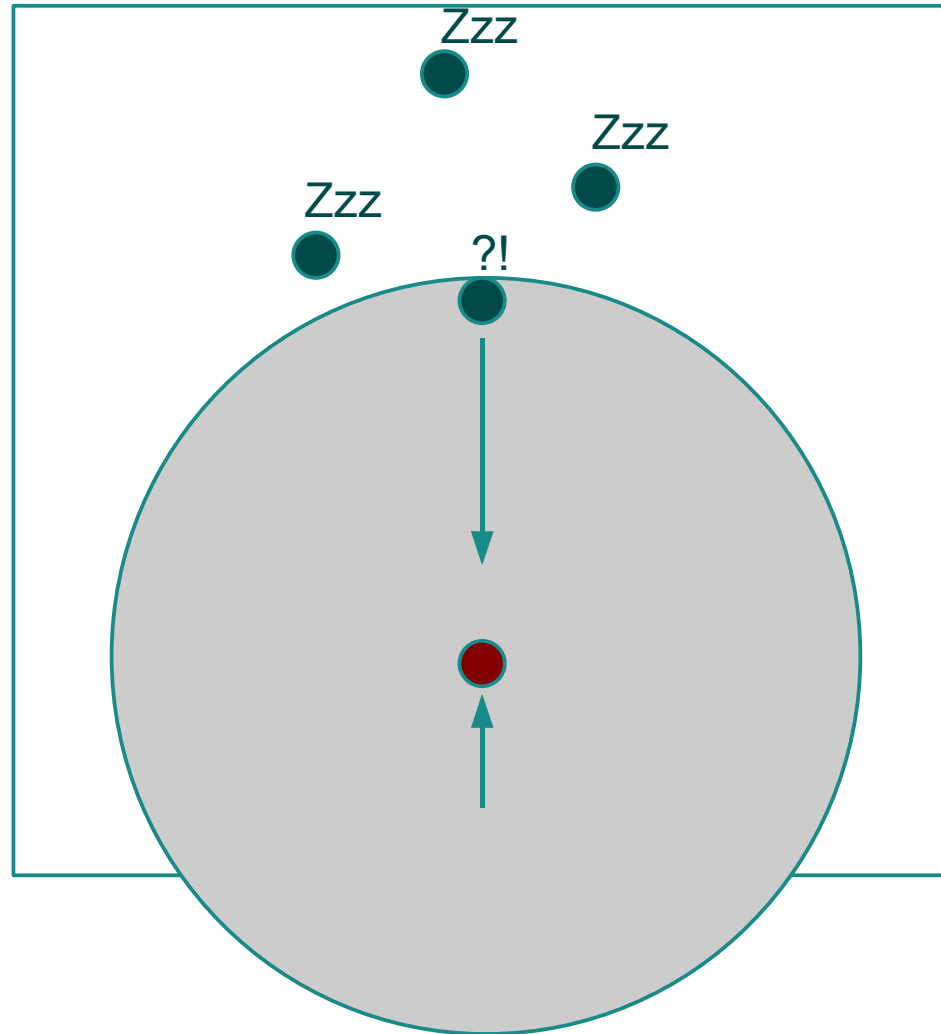


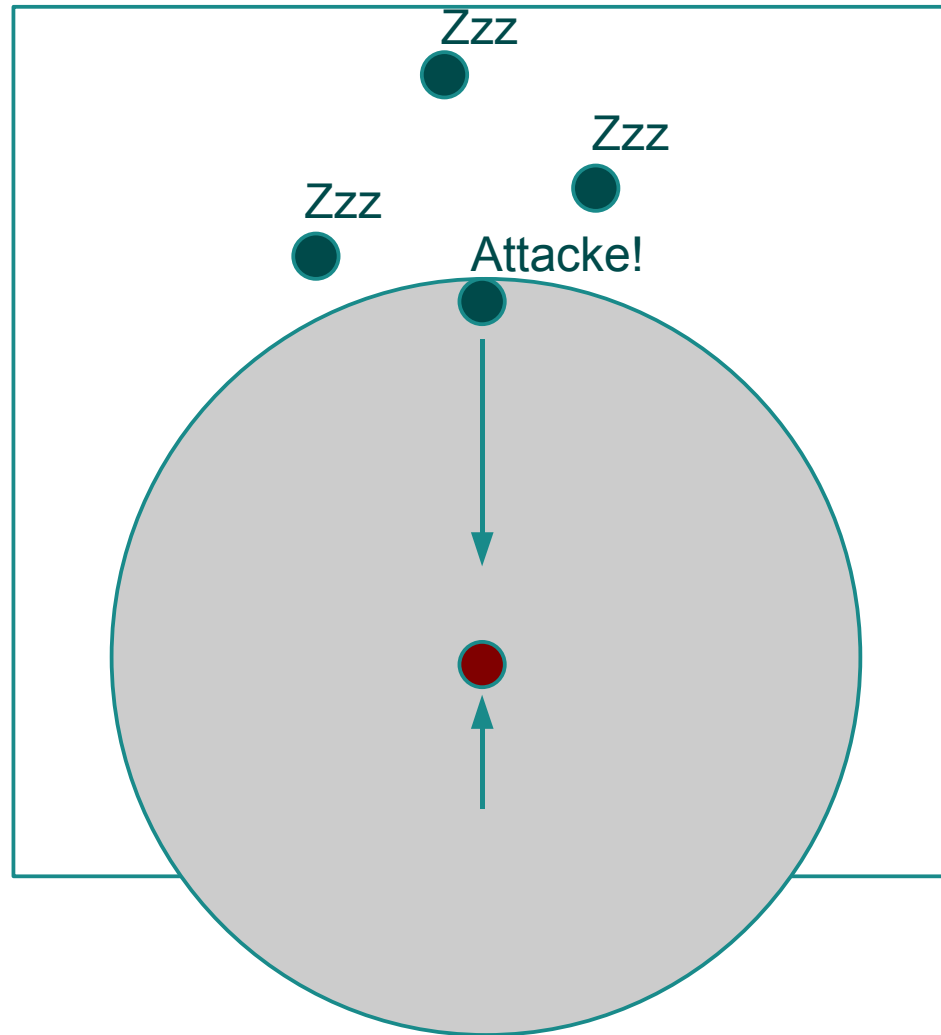


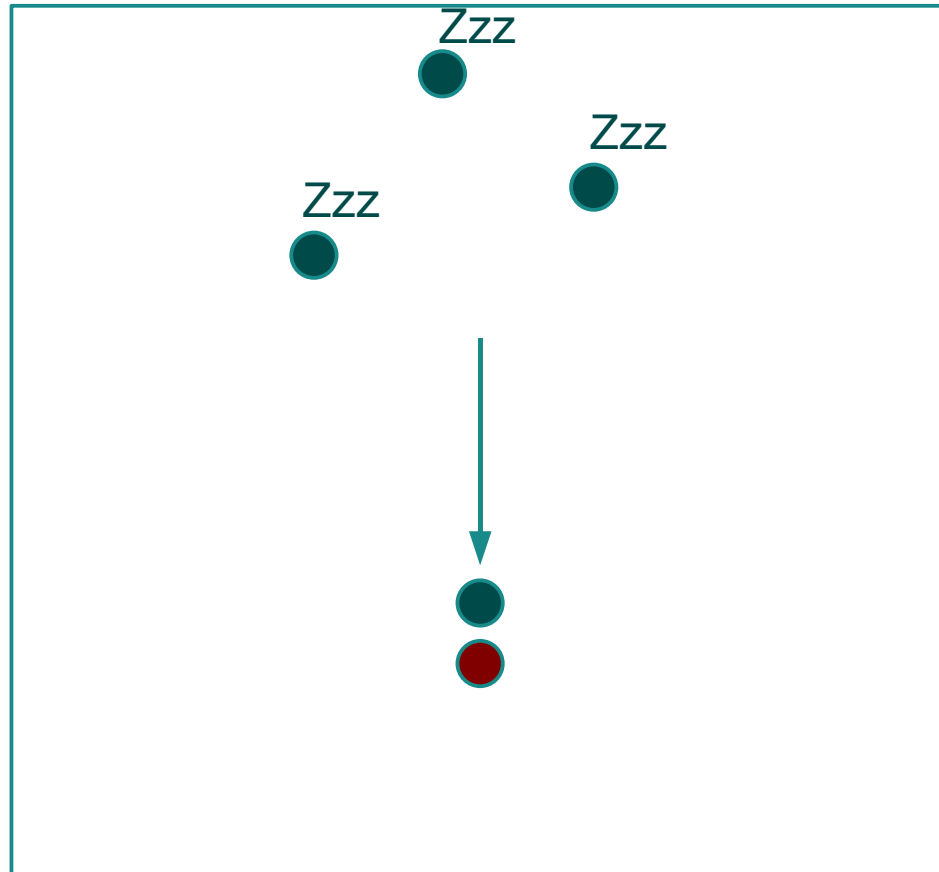
„Alle Wachen zu mir!“ - Effekt



Das ist dämlich!

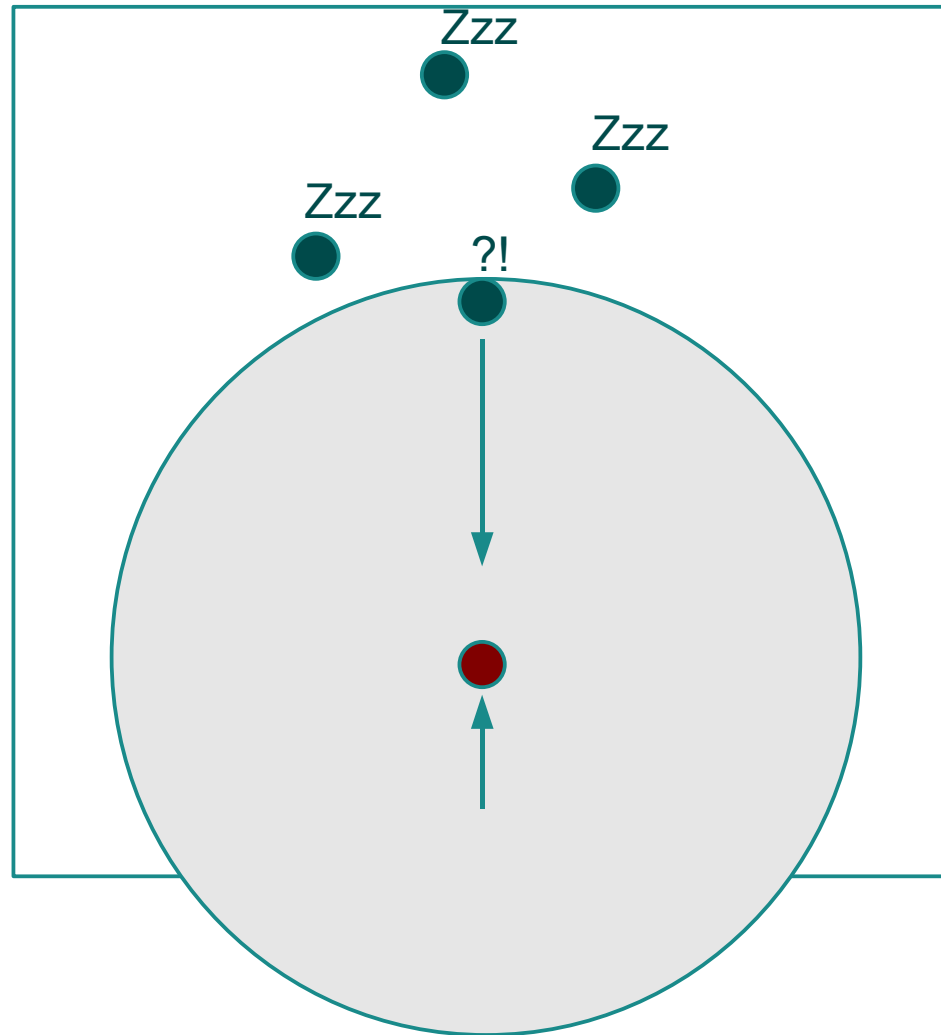


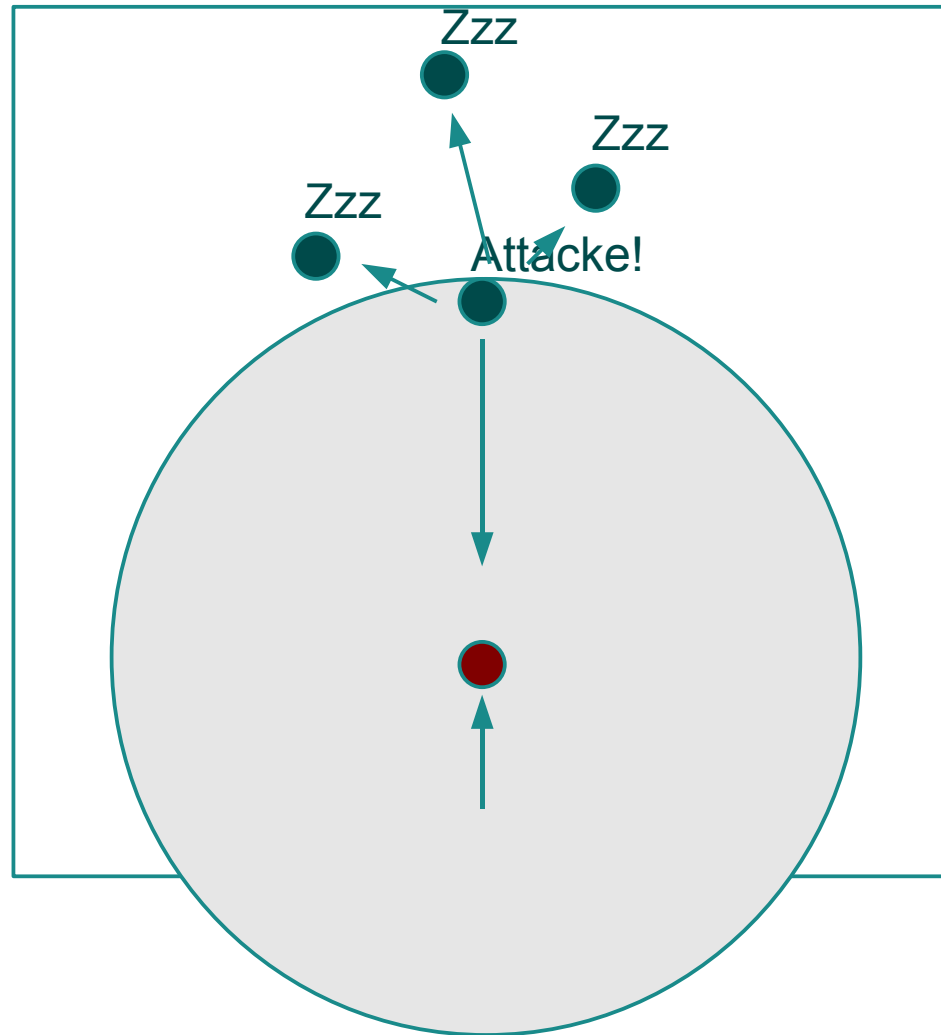


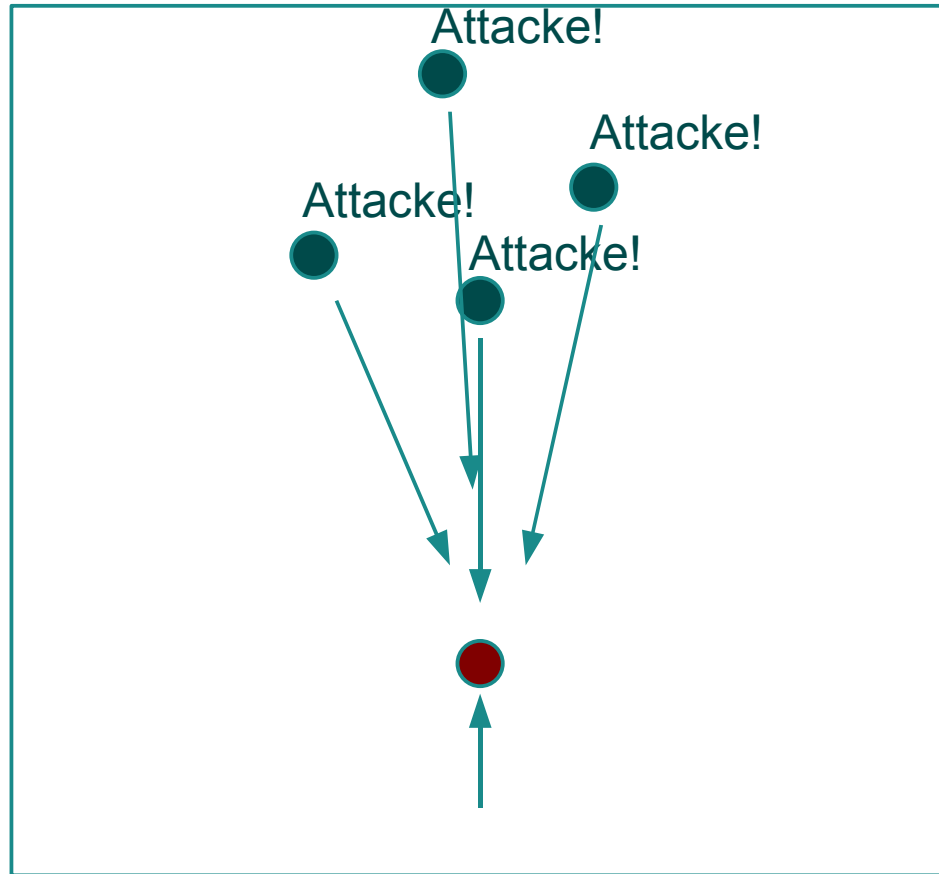


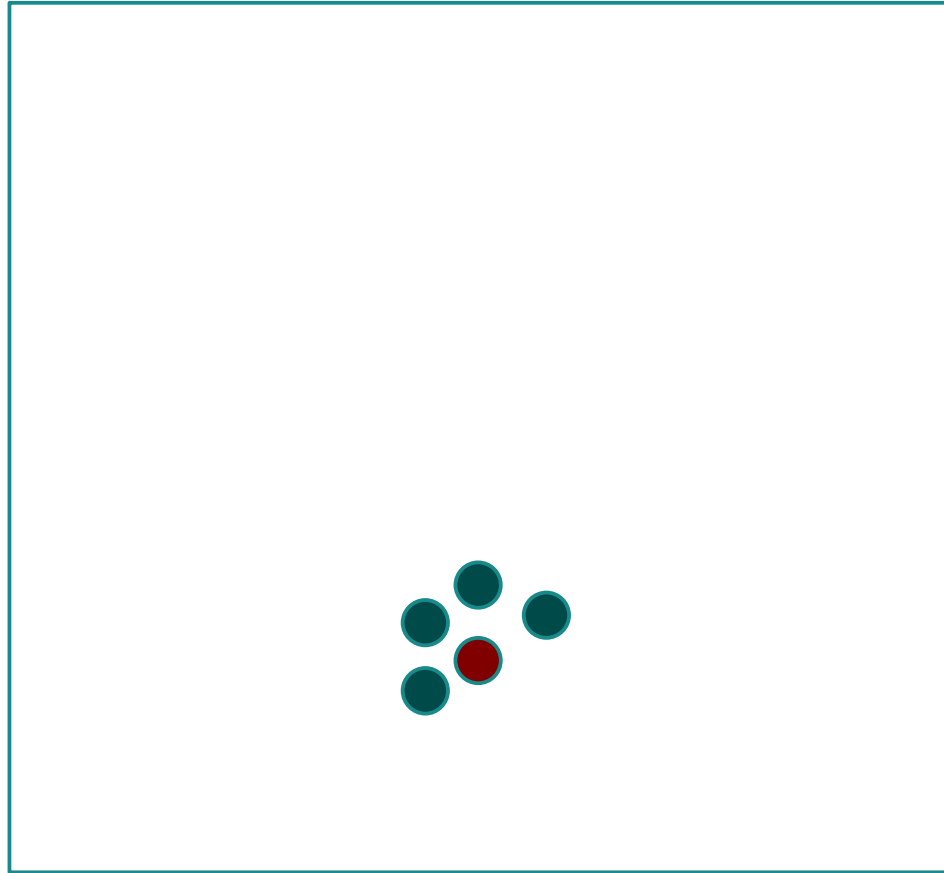


Das ist auch dämlich!











Das ist clever!

→ Unsere Monster sollten sich so verhalten!



## Kampfsystem

→ Verweis auf Paper

Paper definiert sehr einfachen (zu einfachen?) Mindestsatz,  
Enthusiasten können gern einen Überbau basteln (und dokumentieren!)

## Magiesystem

→ dito.



## Händler

NPCs, die in bestimmten Levels einen „Laden“ (=Wandrechteck mit Tür drin) betreiben:

- Verkaufen Güter
- Kaufen Güter (zum halben Preis)

Händlertypen:

- Wirte/Kneipiers
- Quest-Händler
- Allg. Ausstattung (Lampen, Rüstung, etc.)

Ist Handel einmal implementiert, ist Hinzufügen neuer Typen sehr einfach.



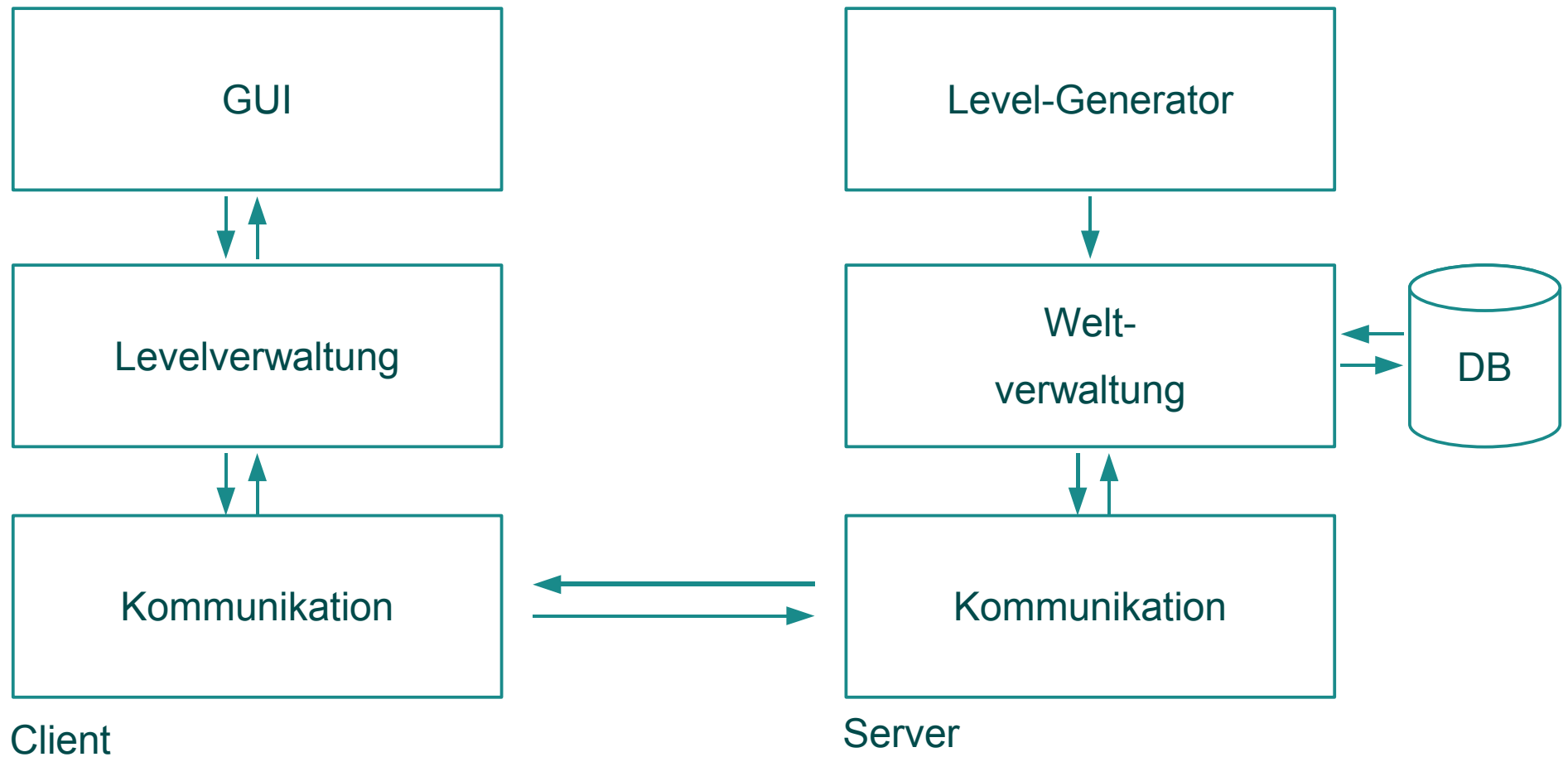
## 2. Features des Clients

### 2.1. GUI

- Darstellung -
- Steuerung -
- Technik -



## Architektur





## Darstellung

2D-Darstellung des sichtbaren Levelausschnitts mit Spieler in der Mitte:

- Flach ist etwas einfacher,
- Isometrische Ansicht ist etwas hübscher.

Mehraufwandsschätzung: ca. 1h Papier und Stift, ca. 2h Implementierung

Darstellung einer Minimap:

- Darstellung des kompletten Levels im Miniaturformat
- Erleichtert die Übersicht



## Darstellung II

Darstellung des Inventars:

- Items, die Charakter in seiner Tasche mit sich herumträgt
- Aktuell verwandte Items wie Waffe, Rüstung, Hut, Stiefel.

Darstellung von Charakterwerten (HitPts, ExPts, etc.)

Darstellung eines Nachrichtenfensters:

- Nachrichten des Client-Engines („Oliver isst die Pizza. (+200NW)“)
- Chat-Nachrichten anderer Spieler im aktuellen Level
- Flüster-Chat-Nachrichten direkt benachbarter Spieler



## Steuerung

Abfrage von Maus und Tastatur



Paper: Graphikausgabe und  
Maussteuerung

Charakteraktionen:

- Bewegen
- Monster angreifen per dagegen anrennen

Item-Aktionen:

- Aufnehmen eines Items per „Überlaufen“
- „Use“-Aktion: Öffnen, Schließen, Aufessen, Anwenden, etc.
- Ablegen eines Items im Level



## Steuerung II

### Chataktionen:

- Normal chatten – Für alle Spieler im aktuellen Level mit zu lesen
- Benachbarten Spielern etwas zu flüstern

### Spielsteuerung:

- Anmeldung: Fragen nach Charaktername und Passwort
- Abmelden und beenden

## Technik: Double Buffering

Bild zeichnen – Erster Versuch:

- Maloberfläche („Surface“) im Fenster besorgen
- Altes Bild löschen
- Bodenkacheln malen
- Items malen
- Spielfiguren malen

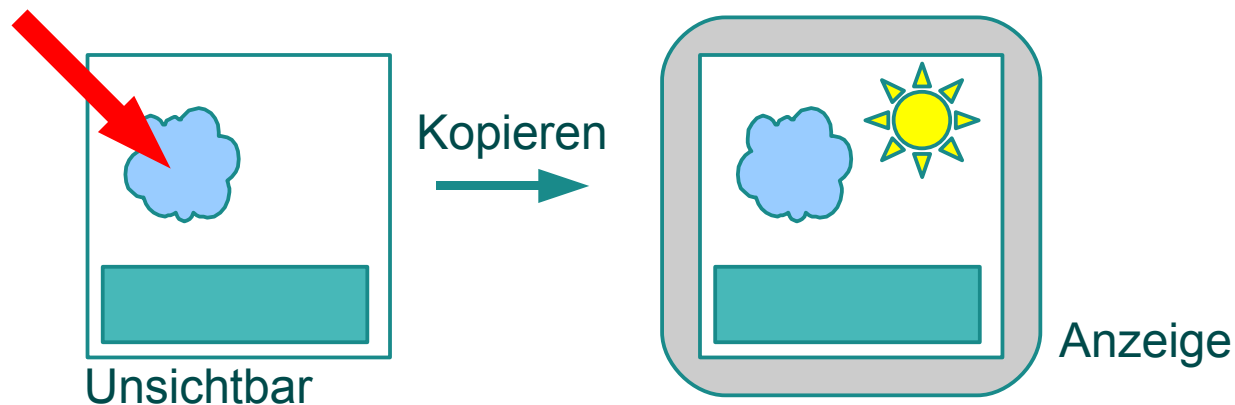


Problem: Beim Bildschirmaufbau kann zugeschaut werden - Flackern!

## Technik: Double Buffering II

### Lösung: Double Buffering

- Zwei Surfaces:
  - eins für die aktuelle Anzeige (den aktuellen „Frame“),
  - auf dem anderen finden im Hintergrund Zeichenaktivitäten statt.
- Wenn fertig gezeichnet, in einem Rutsch in die Anzeige kopieren



Anzahl der Zeichen- und Kopiervorgänge pro Sekunde: *Framerate* (gerne >15)



## 2. Features des Clients

### 2.2. Levelverwaltung



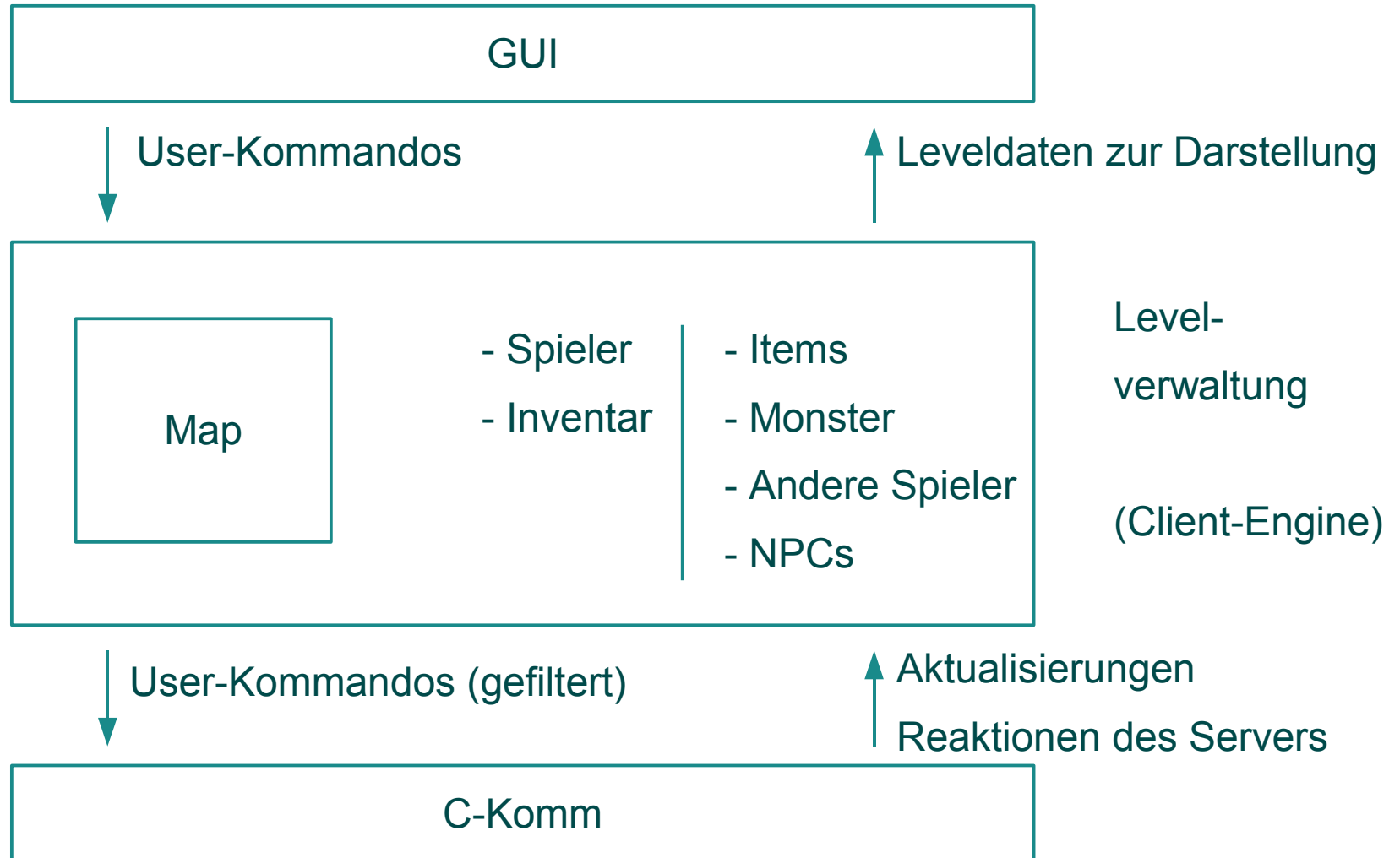
## Levelverwaltung

- Verwaltet aktuellen Zustand des aktuellen Levels
- Prüft Spielerkommandos auf Konsistenz
- Reicht Spielerkommandos weiter an Kommunikationsmodul

Gutes Datenmodell für die Levels ist sehr wichtig.

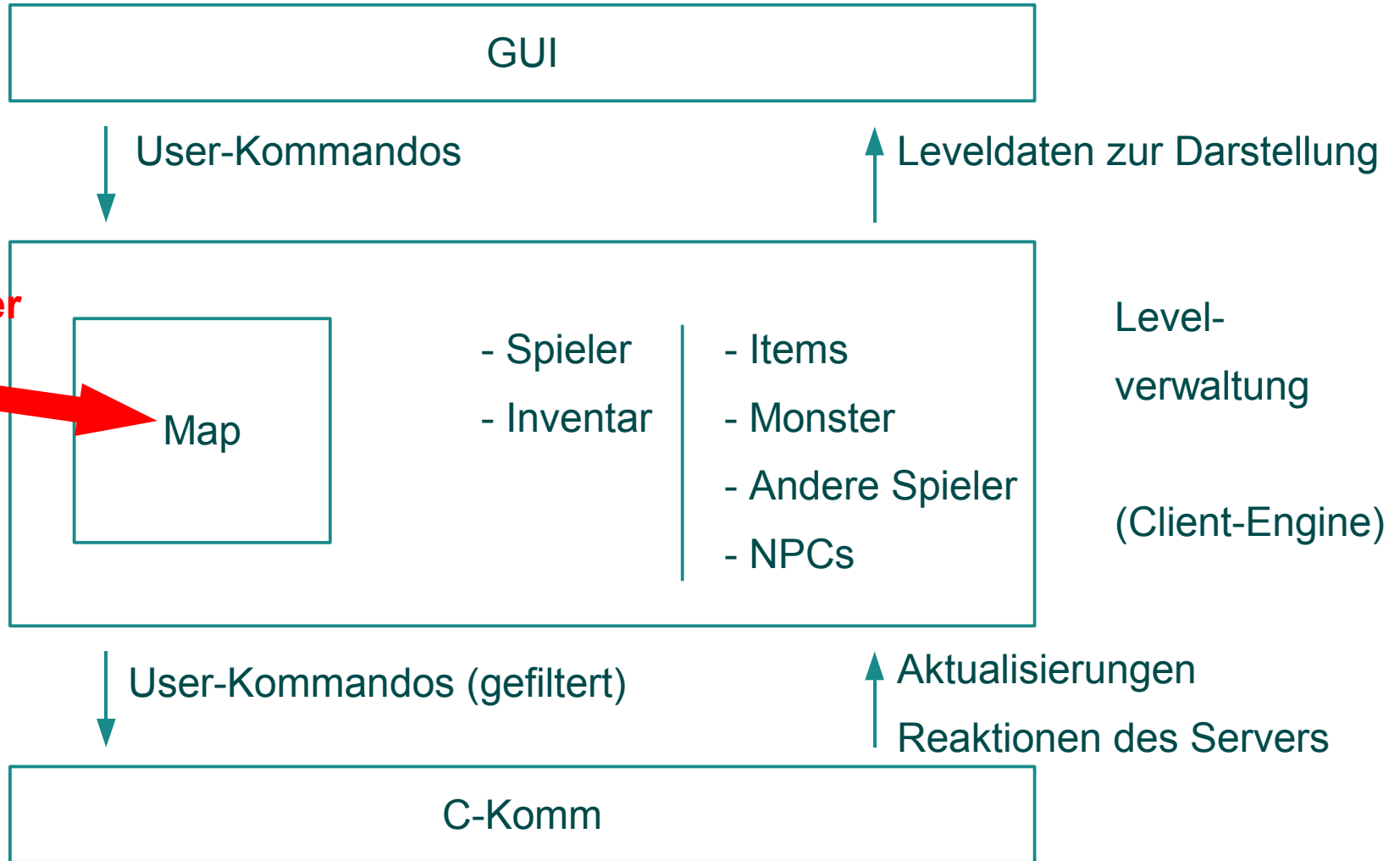


## Levelverwaltung II





## Levelverwaltung IIb



**Benutzen Sie hier kein char[][]!  
Benutzen Sie eigene Klassen,  
Chars tragen nicht!**





## 2. Features des Clients

### 2.3. Kommunikation



## Kommunikation – Clientseite

- Sessionmanagement (An-/Abmelden)
- Empfängt Leveldaten und Aktualisierungen
- Sendet Userdaten an den Server

Die grundlegenden Messageklassen sind von uns implementiert worden und stehen für Sie zur Verfügung. Dazu gehörige Dokumentation ist in der Mache.



## **„Hausaufgabe“ für nächste Woche Mittwoch:**

1. Bekommen Sie das mit der Gruppenaufteilung hin!
2. Lesen Sie die Papers!



**Vielen Dank.**

[ullrich@informatik.uni-koeln.de](mailto:ullrich@informatik.uni-koeln.de)