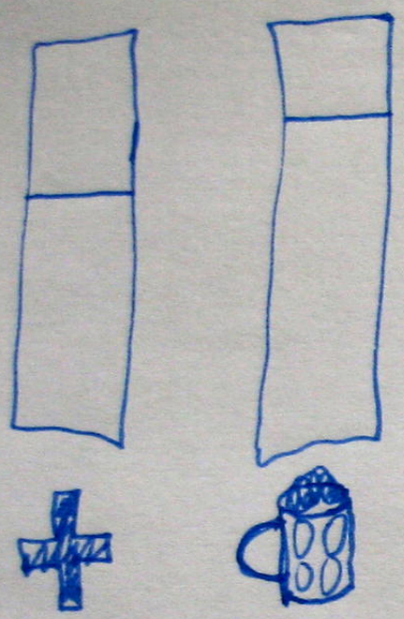
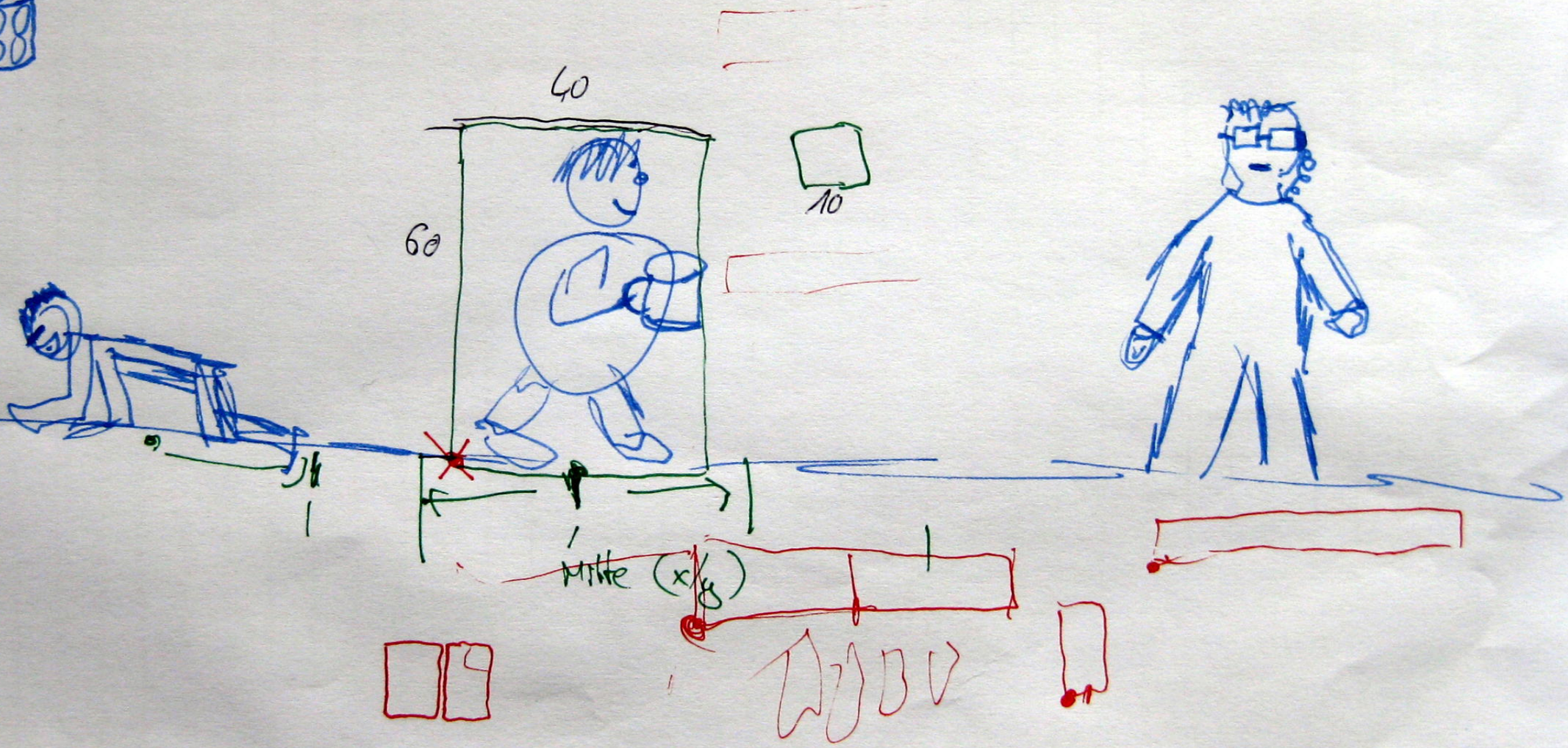


1 -> Ende

velches

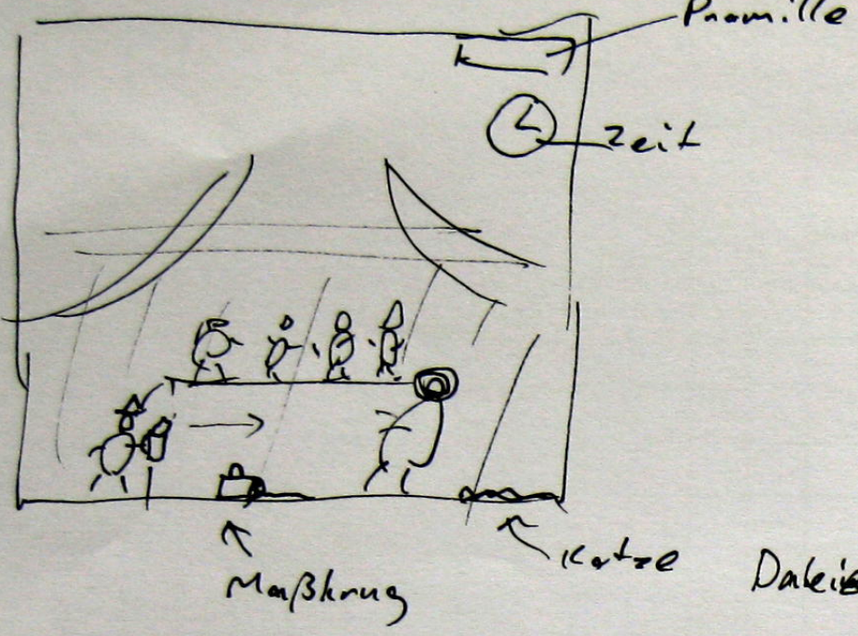
Score:



(x/y)

Länge x

Länge y



OO Analyse
- welche Objekte
- Interaktion

OO Design
- Klassenbeziehungen

Sound
- Transformationen
- Ausgabe
- Dateien
- Anpassung

Grafik
- Files
- Bewegung
- Ausgabe
- Licht

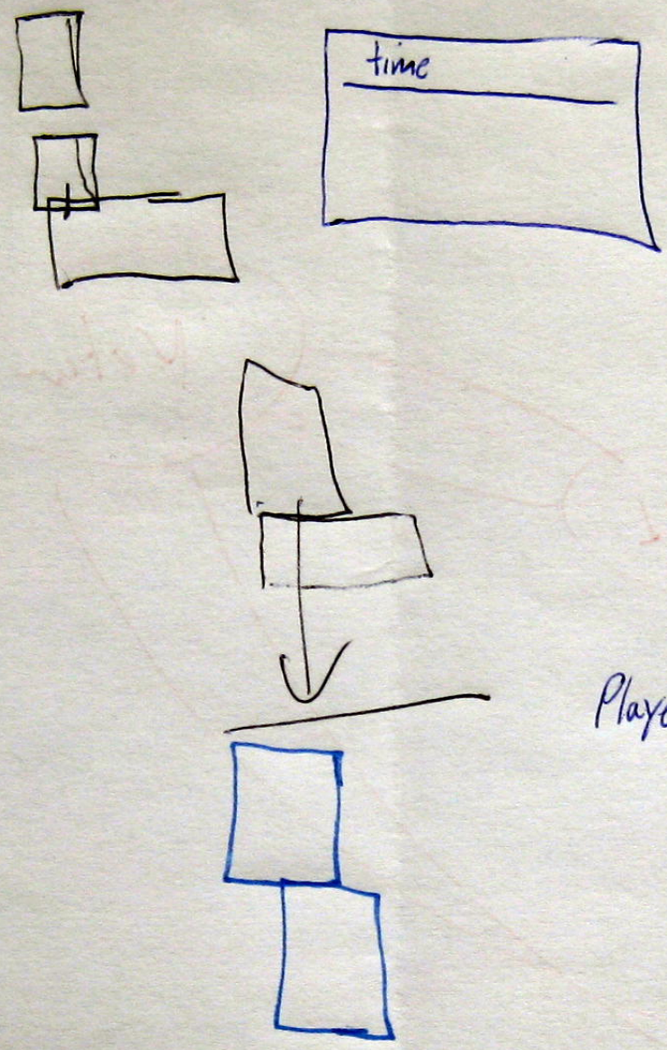
Input
- Tastatur
- Maus?

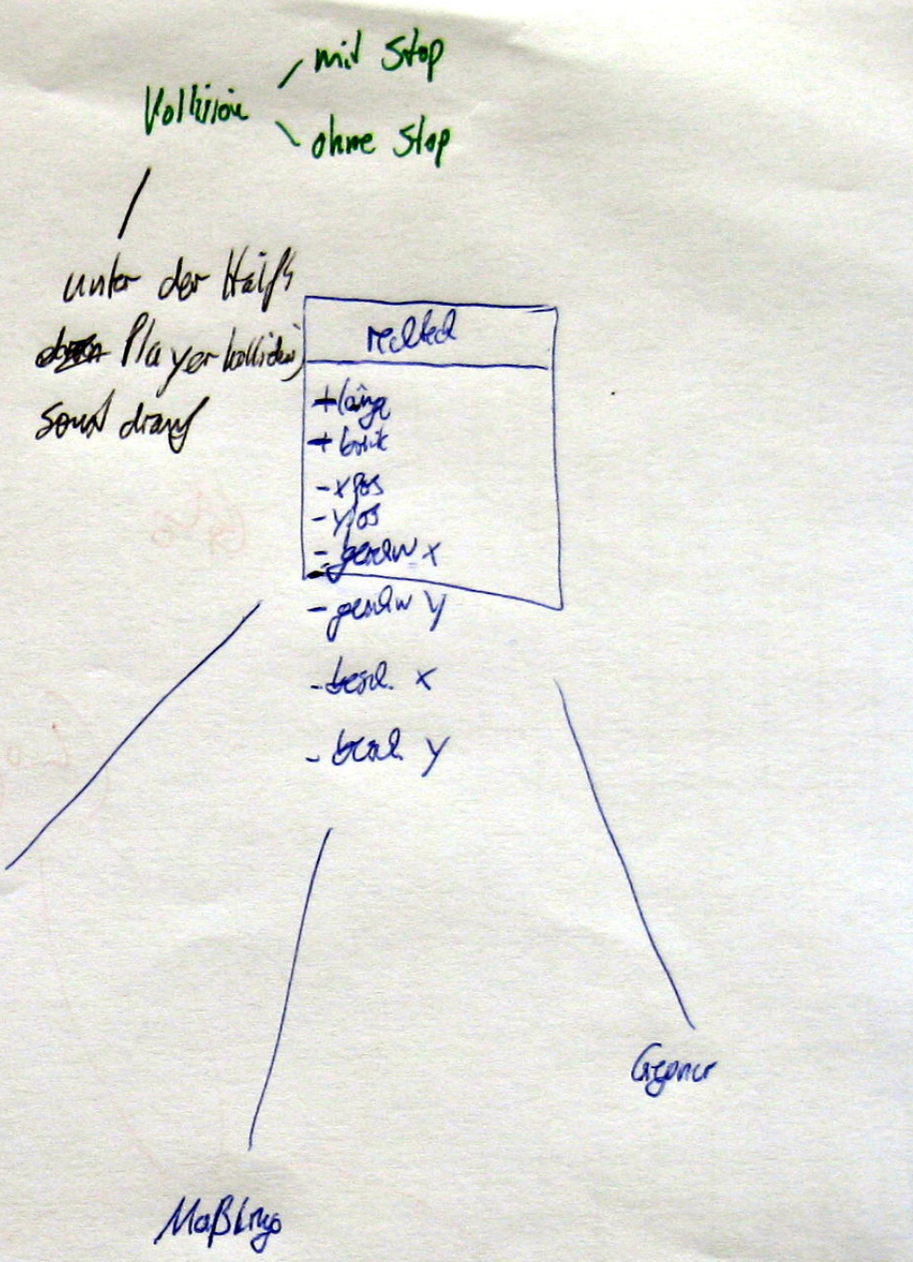
Gamelogik
- Leben / Score
- Zeit
- Gameloop
  - Objekte
  - Score
  - Collision
  - Move
- Objekte erstellen
- Level definition
- Start
- Stop
- Save
- Menü

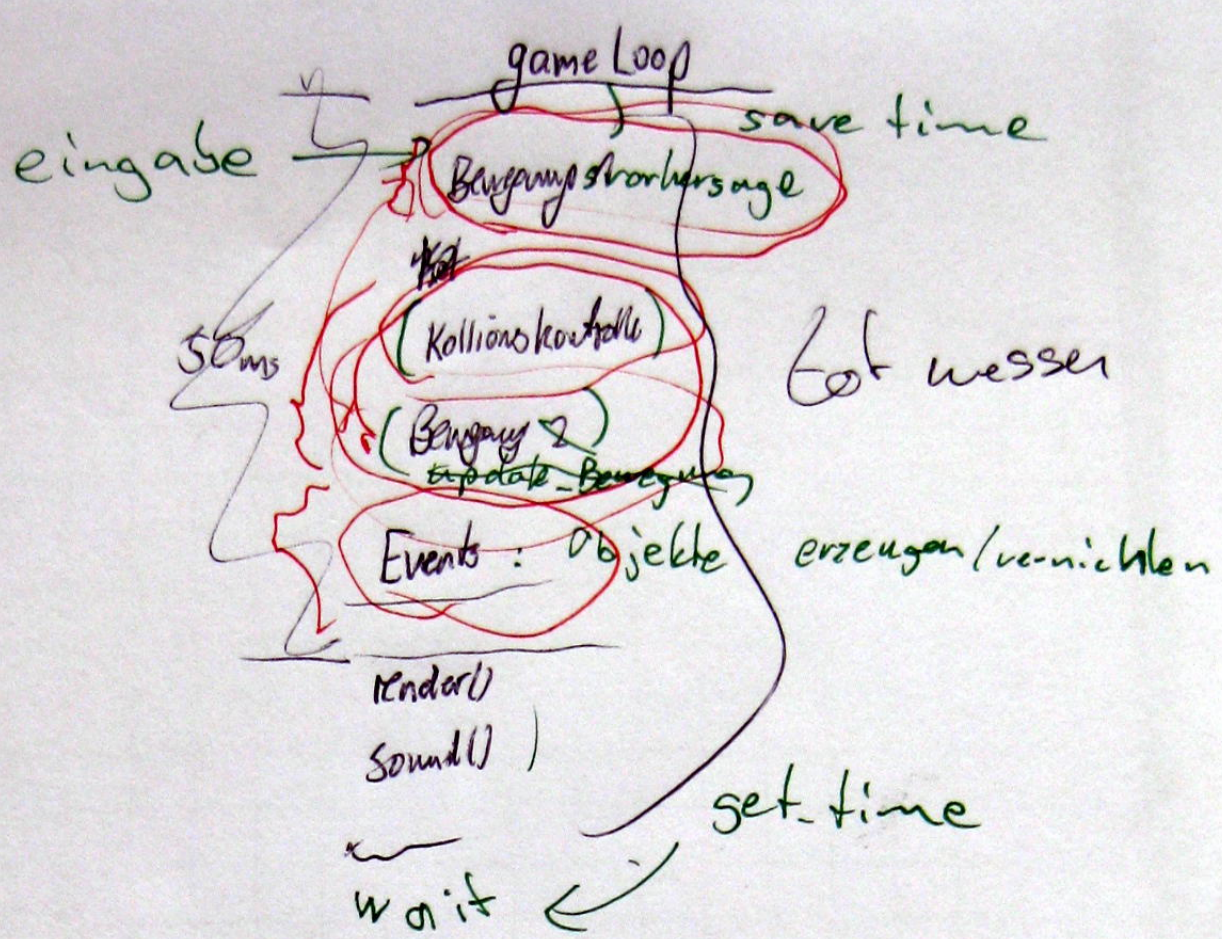
Anforderungen

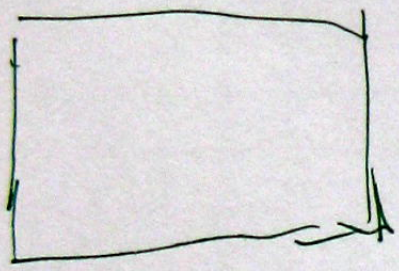
- Kollisionskontrolle und Lösung
- „Bewegung" in der Welt
- Aufrufen & Abarbeiten von Events
- Infos in kond. Rate zur Verfügung stellen
- Punkte / Leben / Status / Zustand berechnen
- Online laden/generieren der Welt

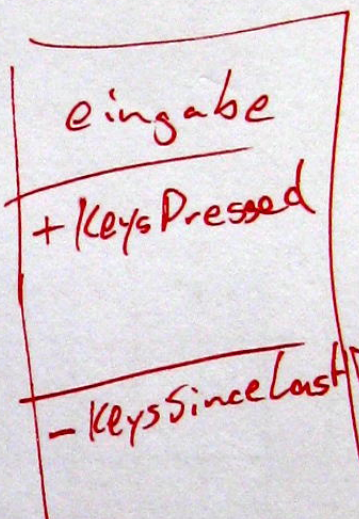
- (Netzwerk)
- (KI)
- Multithreading











Game

level welt: Liste mit allen Objekten im Level

welt..

aktuellen geladenen Obj.

Schüsse

unbewegte
- wird einmal erstellt u. dann nur noch Obj. gelöscht

bewegte
- muss immer sortiert werden
- Player ist Element
→ nur Element vorher und danach überprüfen

Kollision

check_for_collision (Rechteck)

Rechteck

Weltobjekt:
QSortedList (sortiert nach xpos)

an Anfang Hardgecoded
QSortedList (xpos)

Bewegt
- Gegner
- Schüsse
- Player

Unbewegt
- Obstacles
- Power-Ups
- Player

Initialisiert Feld

Dynamisch Spawnen
- Gegner

gameObject(__)

+get...
+set...

- länge // x
- höhe // y
- xpos } Bezug
- ypos } links unten

- xgeschw.
- ygeschw

Game
+ ~~gameStand~~ Liste bearbeiten
+ ~~Bewegungsvorhersage~~
- Kollisionskontrolle
- ~~Bewegt~~
- ~~Event~~
- weltBewegt
- weltUnbewegt
- Level ~~Initialisierung~~
- LevelSpawn
- HighScore

QSortedList an Grafik

Did an Sound

+Eingabe
+SaveGame
+MaxLength

Fripur

Player : gameObject
- Score
- Leben
- Pegel
- powerup.

~~Master~~
Schuss : gameObject
- Spawnzeit
-

Gegner1 : gameObject
- Waffentyp
- todeszeitpunkt