

Analyse- und Designmodell

Gruppe 1: Wiesn-Run

Übersicht

- Spielidee
- Menü
- Level
- Klassen

Spielidee: Wiesen-Run

- Oktoberfest-Grafik
- Figur fix, Welt bewegt sich
- Optionale Module
 - Bewegungsparallaxe
 - Soundausgabe
- Steuerung:
 - rechts laufen
 - springen: entweder senkrecht oder schräg vorne
 - werfen horizontal

Ziel des Spiels

Punkte sammeln durch

- Gegner töten
- Strecke zurücklegen
- Items sammeln
- Level mit möglichst hohem “Pegel” beenden

Spielerstatus

- Pegel: +/- über Powerups und Zeit
 - Ansporn, Level schnell zu beenden
- Punkte
- Gesundheit

Powerups

- Bier: + Pegel, +Gesundheit, +Munition
- Hendl: - Pegel, +Gesundheit

Gegner

- Türsteher: Schaden durch Kontakt
- Betrunkener Tourist: wirft Masskrüge
- Endgegner im letzten Level
- Treffer reduzieren Gesundheit
 - Auswertung durch Kollisionskontrolle und Event-Handling
 - nach Treffern kurze Unverwundbarkeit

Angriffsarten

- Player:
 - auf Gegner springen, Masskrug werfen
- Gegner:
 - gegen Spieler laufen, Masskrug werfen

Menü

- graphisch
- Steuerung über Tastatur
- Startmenü
- Statistik
 - getötete Gegner, gesammelte Items, Zeit
 - Namenseingabe
- Highscore
 - wird in Textdatei gespeichert



WILLKOMMEN ZUM

Wiesn Run

Pack ma's!
Highscores
Servus!

Levels

- Weg zum Zelt (Hauptstraße)
 - nur Hindernisse, keine Gegner
- Hacker-Zelt
 - Türsteher als Gegner
- Käfer-Zelt
 - masskrugwerfende Touristen, mehr Türsteher
 - Endgegner am Ende



0001670



Physik

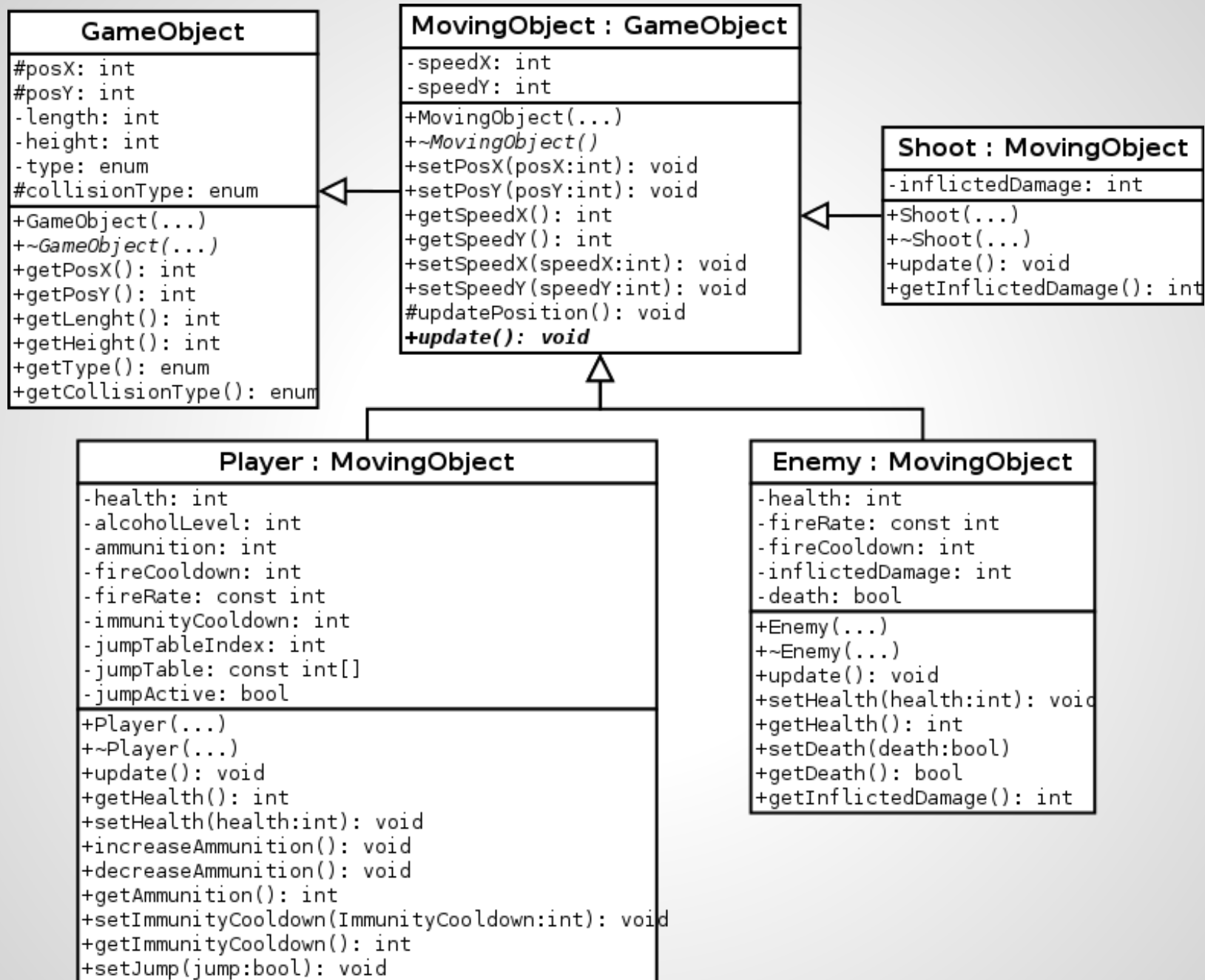
- Bewegungsberechnung
- Kollisionskontrolle
 - keine Floats, nur Integer
- Sprünge über Lookup-Table

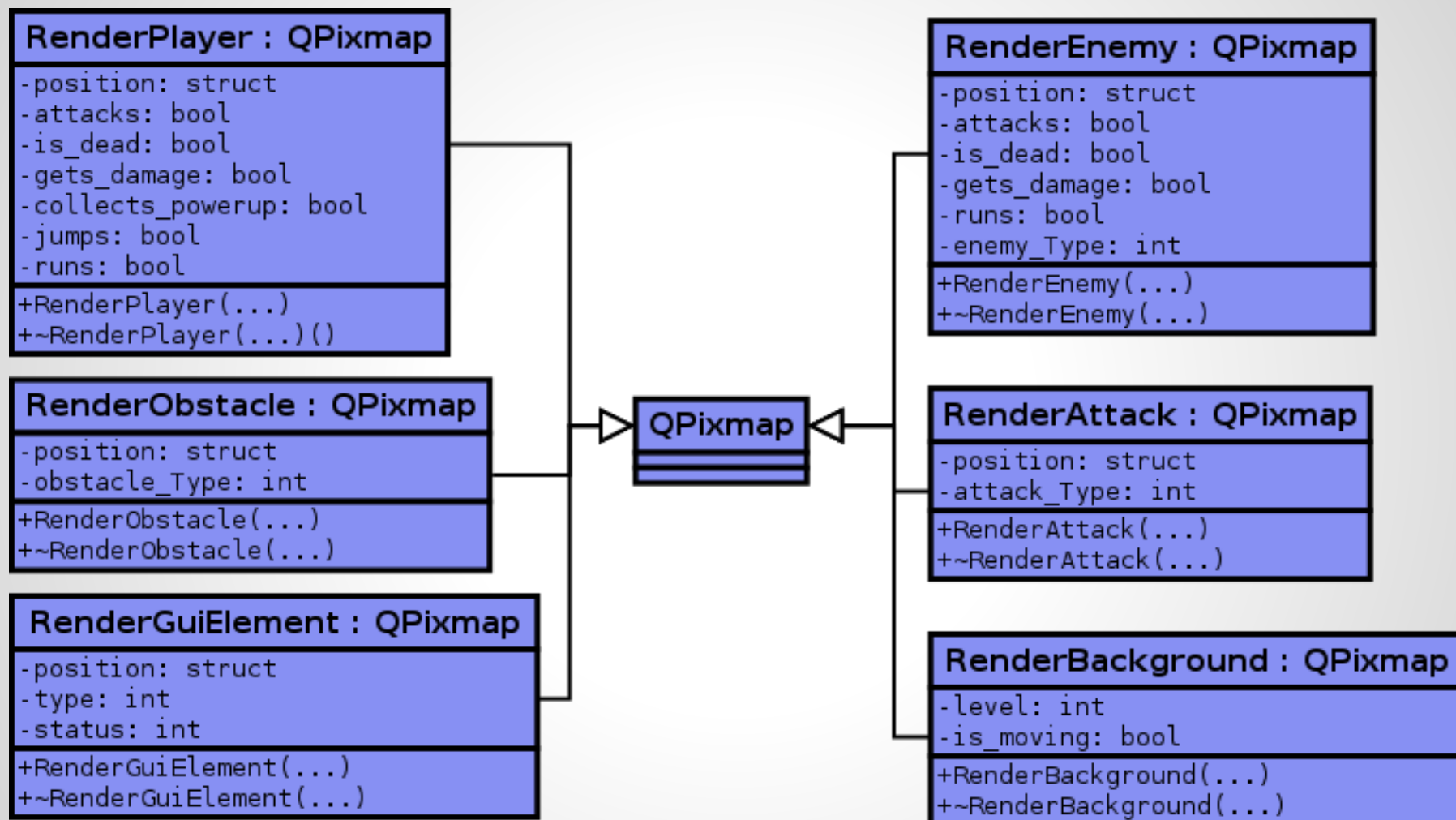
Klassen

Game

```
-worldObjects: QSortedList GameObject
-levelInitial: QSortedList GameObject
-levelSpawn: QSortedList GameObject
-scores: struct
-stepSize: const int
+eventsToHandle: QList struct
-playerObjPointer: *Player
+states: QMultiHash
-keyInputs: Input

+Game(...)
+~Game(...)
+start(): void
+getStepSize(): int
-appendWorldObjects(): void
-reduceWorldObjects(): void
-evaluateInput(): void
-calculateMovement(): void
-detectCollision(): void
-correctMovement(): void
-handleEvents(): void
-renderGraphics(&worldObjects:QSortedList gameObject
                position:int): void
-sound(states:QMultiHash): void
-gameEnd()
```





Input
-keyevents: QSet -keyactions: QSet
+Input(...) +~Input(...) -keyeventsFilter(&event:QEvent) -update_keyactions() +get_keyactions(): QSet

AudioControl
-audiostates: QMultiHash
+AudioControl(...) +~AudioControl(...) +add(states:QMultiHash) -check(states:QMultiHash) -start(states:QMultiHash) -stop(states:QMultiHash) -update()

Audio
+Audio(...) +~Audio(...) +setSource() +setVolume() +setPlaytype() +startPlaying() +stopPlaying() +checkPlaying(): bool

Fragen?

