

Analyse- und Designmodell

WILLKOMMEN ZUM

Wiesn Run

Pack ma's!
Highscores
Servus!

Analyse- und Designmodell

WILLKOMMEN ZUM

Wiesn Run

Pack ma's!

Highscores

Servus!

Spielidee: Wiesen-Run

- Spielfigur: Bayer der durchs Oktoberfest irrt
- Fähigkeiten Spielfigur:
 - Bewegung: In Richtung rechter Spielrand
 - Angriff: Springen senkrecht oder schräg vorwärts
 - Angriff: Horizontales Werfen von Bierkrügen
- Gegner:
 - Türsteher: Schaden durch Kontakt
 - Betrunkener Tourist: wirft Masskrüge

Spielerzustand

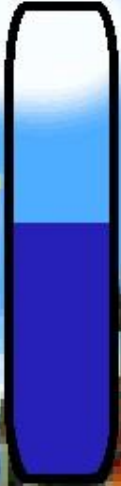
- Gesundheit:
 - + Hendl
 - + Bier
 - Gegnerkontakt
 - Bierkrugtreffer
- Alkoholpegel:
 - + Bier
 - Hendl
 - Zeit

Levels

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



0001670



Ziel des Spiels

Score-Punkte sammeln durch

- Außer Kraft gesetzte Gegner
- Zurückgelegte Strecke
- Gesammelte Powerups
- Hohen Alkohol Pegel

Optionale Module

- Bewegungsparallaxe
- Soundausgabe
- Ziel: Möglichst viele Spielfunktionen selbst programmieren

Menüs

- Graphische Implementierung
- Steuerung über Tastatur
- Startmenü
 - Spiel Starten
 - Spielstatistik
 - Spiel Beenden
- Spielstatistikmenü
 - Score, getötete Gegner, gesammelte Powerups, Zeit
 - wird in Textdatei gespeichert
 - neue Score: Namenseingabe am Ende vom Spiel

Physik

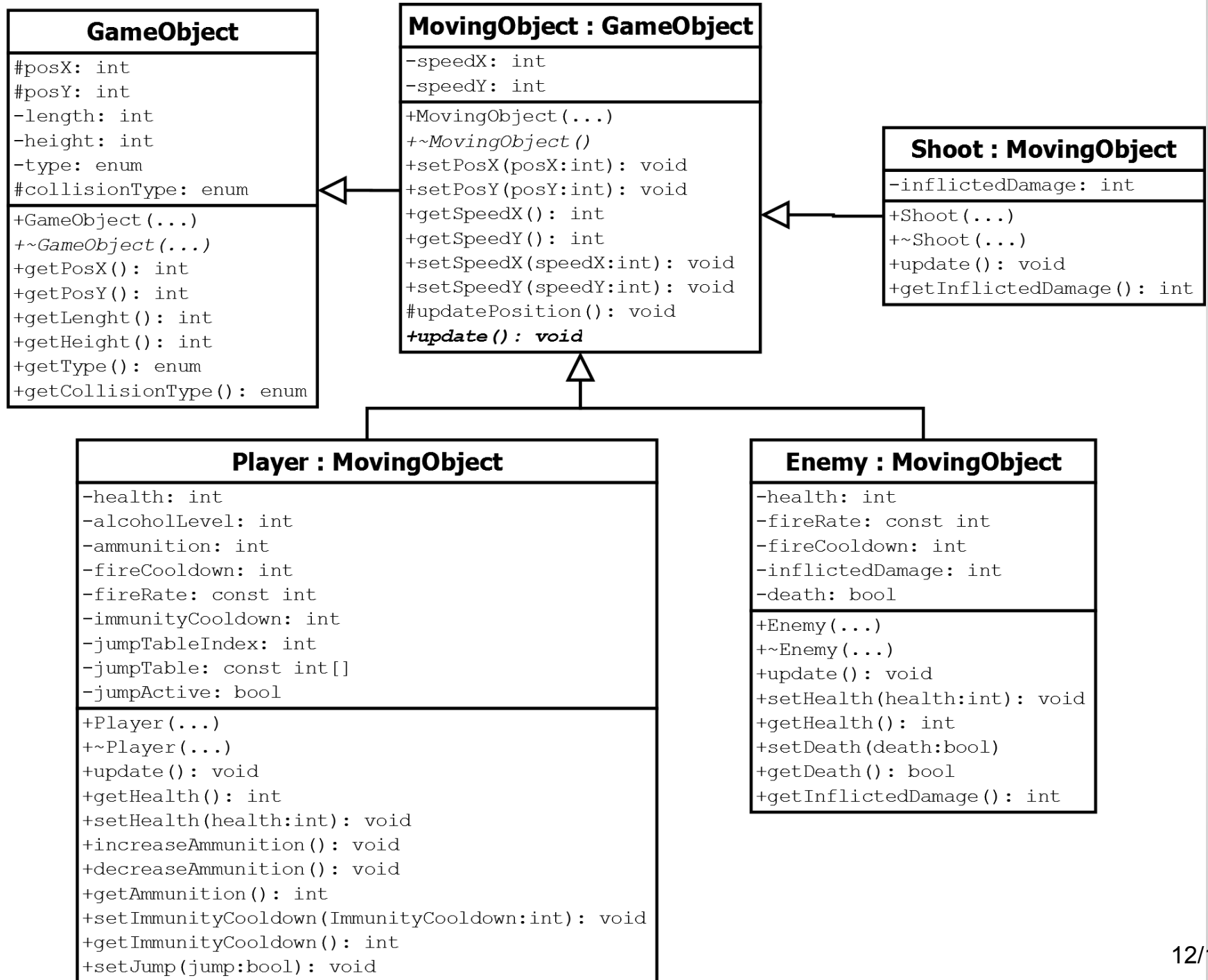
- Koordinaten und Geschwindigkeiten in Integern
- Bewegungsberechnung durch einfache Integration der Geschwindigkeit
- Kollisionskontrolle und Korrektur der Bewegungen
- Sprünge über Lookup-Table

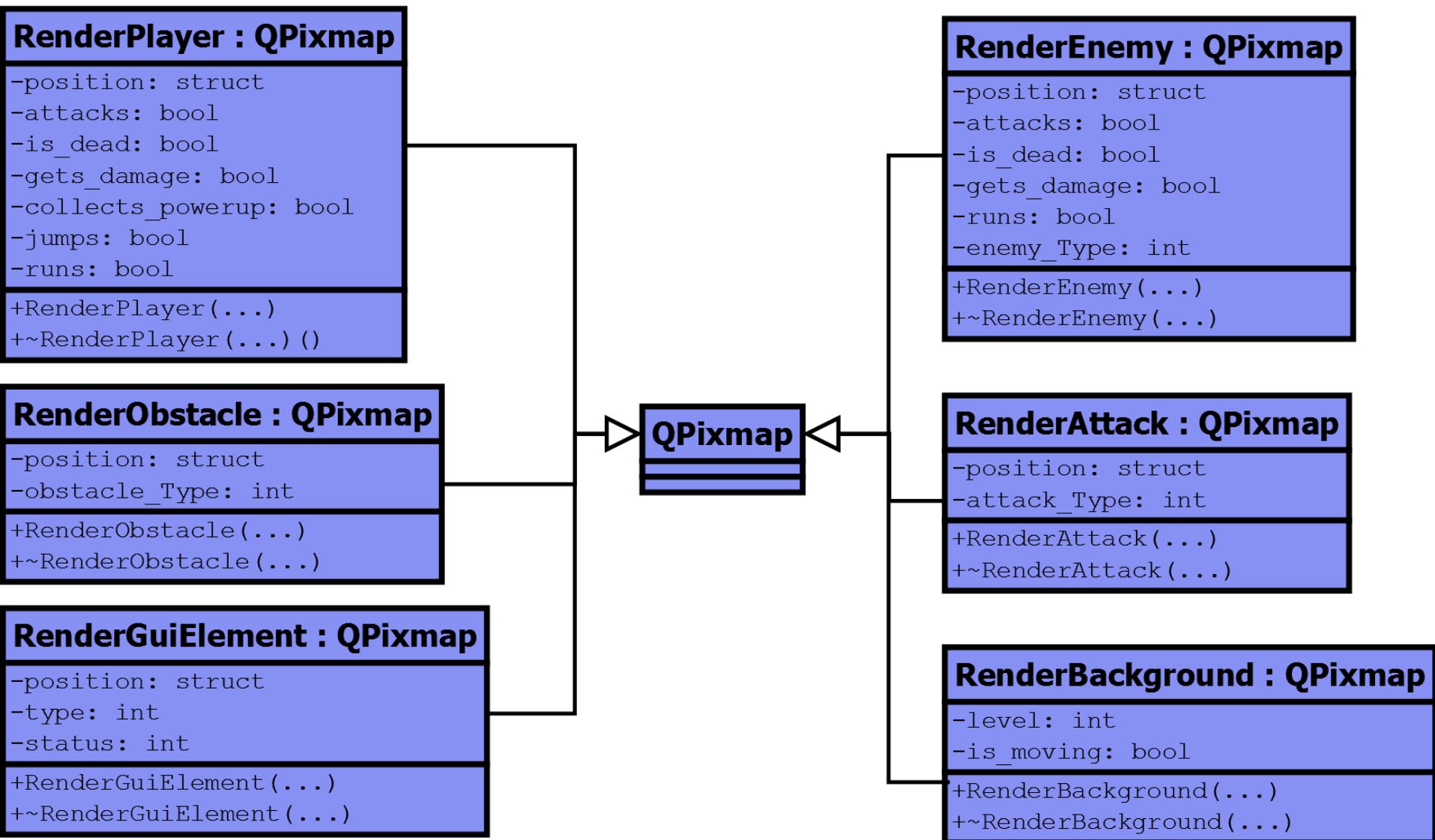
Klassen

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?

