

Analyse- und Designmodell

Gruppe 1: Wiesn-Run

Übersicht

- Spielidee, Ziel
 - Spielerstatus, Powerups, Gegnerarten, Angriffe
- Levels
- Menüs
- Klassen

Allgemeines: Wiesn-Run

- Oktoberfest-Grafik
- Figur fix, Welt bewegt sich
- 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 Eventhandling
 - nach Treffern kurze Unverwundbarkeit

Angriffsarten

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

Levels

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

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!



0001670



Physik

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

Klassen

Game

GameObject

 MovingObject

 Player

 Enemy

 Shoot

Grafik

Input

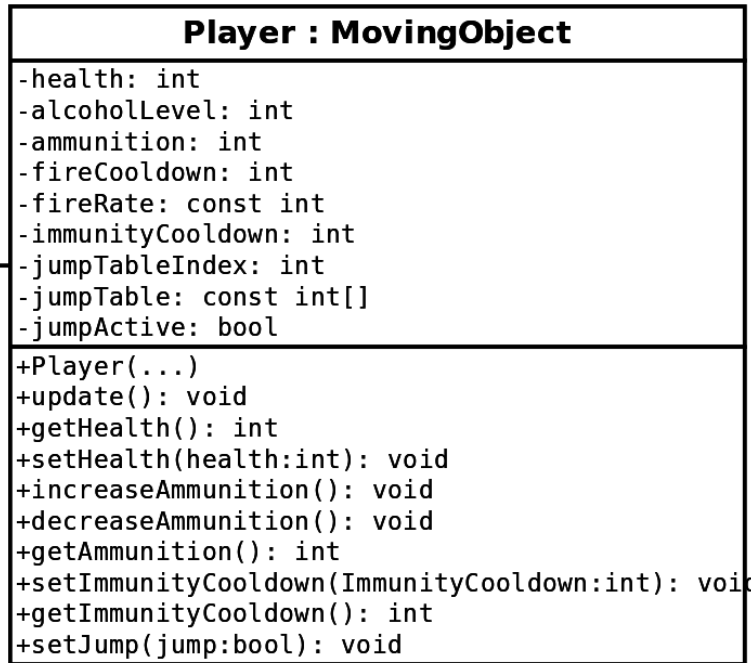
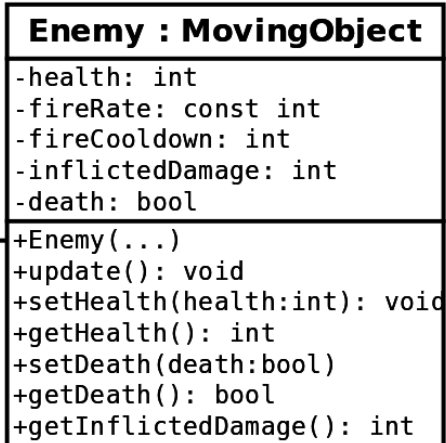
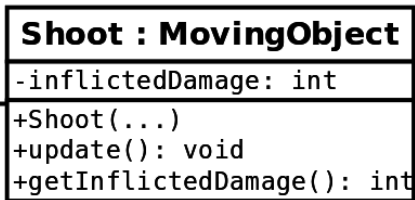
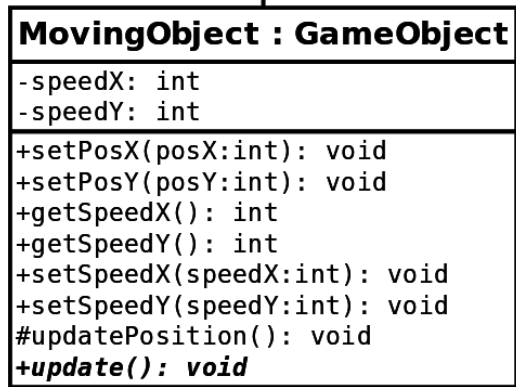
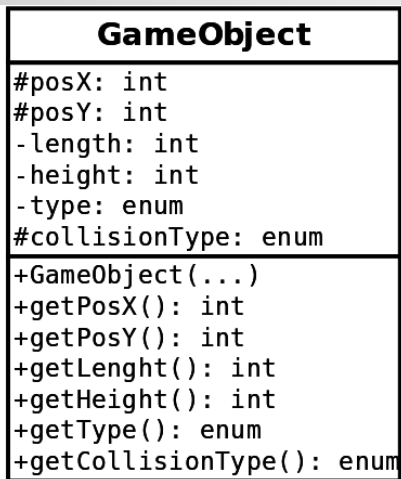
Audio

game

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

- +start(): void
- +getStepSize(): int
- appendWorldObjects(): void
- reduceWorldObjects(): void
- evaluateInput(): void
- calculateMovement(): void
- detectCollision(): void
- correctMovement(): void
- handleEvents(): void

- position:int): void
- sound(states:QMultiHash): void
- gameEnd()



render_player : QPixmap

- position: struct
- attacks: bool
- is_dead: bool
- gets_damage: bool
- collects_powerup: bool
- jumps: bool
- runs: bool

render_enemy : QPixmap

- position: struct
- attacks: bool
- is_dead: bool
- gets_damage: bool
- runs: bool
- enemy_Type: int

render_obstacle : QPixmap

- position: struct
- obstacle_Type: int

render_attack : QPixmap

- position: struct
- attack_Type: int

render_gui_element : QPixmap

- position: struct
- type: int
- status: int

render_background : QPixmap

- level: int
- is_moving: bool

Input
-keyevents: QSet -keyactions: QSet
-update_keyactions() +get_keyactions(): QSet

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

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

Fragen?

Zusatz-Folien

Grafik-Beispiele

Player

Gegner

Masskrug

Status: Gesundheit, Pegel, Zeit, Punkte

Hintergrund

Menü

Menü - Start

- Spiel starten: Startet Level 1
- Highscores: zeigt Liste mit den zehn besten Punktezahlen und Namen an
- Beenden: Schließt Fenster

Menü - Highscores

- Zeigt die 10 besten Punktezahlen an
- nach Levelabschluss Namens eingabe
- Highscores werden mit Namen in Textdatei gespeichert
- zurück-Button

Grafik

QTGraphics Framework

Background: QTGraphicspixmapitem