

CT Projekt: Raycasting engine (Hundefels 2D)

Christian Korn

20.10.2021 - 11.01.2022

Inhaltsverzeichnis

1	Ziele	2
1.1	Muss-Ziele	2
1.2	Soll-Ziele	2
1.3	Kann-Ziele	2
2	Mathematische Funktionsweise	3
2.1	Bewegung	3
2.1.1	Drehung	3
2.1.2	Laufen	3
2.2	Raycasting	3
3	Programmaufbau	4
4	Steuerung	5
4.1	Command-Line Argumente	5
4.2	Levelerstellung	5
4.3	Bewegung	5
4.3.1	Translation (Laufen)	5
4.3.2	Rotation	5
4.4	UI	6

1 Ziele

1.1 Muss-Ziele

Wenn diese Ziele nicht erreicht werden, wird das Projekt als Fehlschlag angesehen.

- Anzeigen eines 2D Levels in 2,5D (Raycasting Methode)
- Bewegungsfreiheit im Level (Translation und Rotation)

1.2 Soll-Ziele

Diese Ziele müssen nicht unbedingt erreicht werden, sind aber für einen vollen Erfolg nötig.

- Laden von Leveln aus Dateien
- Anzeigen von anderen Objekten im Level (z.B. Gegner, Items)
- Kollisionserkennung

1.3 Kann-Ziele

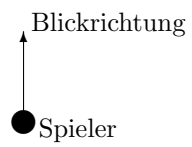
Diese Ziele sind nicht nötig, können aber nach Vollendung der Höheren Ziele in Angriff genommen werden.

- Gegner KI
- Schießen
- Sprites
- Texturen für Wände
- visuelle Effekte (view bobbing, Blutspritzer)

2 Mathematische Funktionsweise

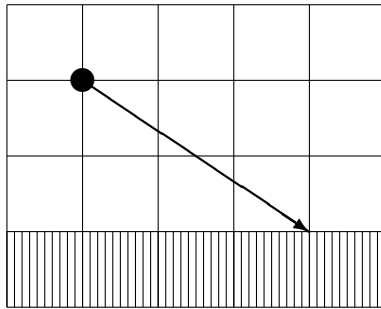
2.1 Bewegung

2.1.1 Drehung

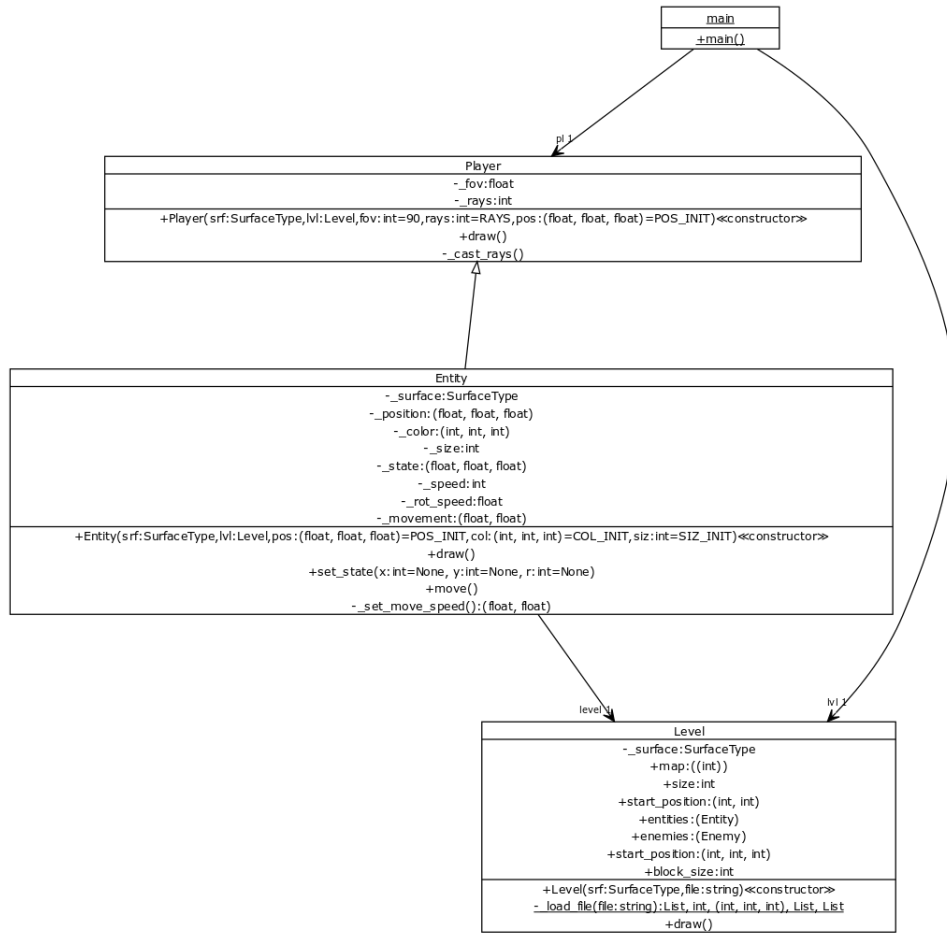


2.1.2 Laufen

2.2 Raycasting



3 Programmaufbau



CREATED WITH YUML

4 Steuerung

4.1 Command-Line Argumente

- `-h --help` zeigt die CLI Argumente und beendet das Programm.
- `-l --level` lädt das angegebene Level oder die angegebene Level Datei.
- `--fov` ändert den Blickwinkel (angegeben in Grad) Standardwert ist 90°.
- `--rays` ändert die horizontale Auflösung (Anzahl der gesendeten Strahlen) Standardwert ist 90. Höhere Werte können die Leistung beeinträchtigen.

4.2 Levelerstellung

Level werden im JSON-Format gespeichert.

- `"map": [[int]]` Die Map: 1 entspricht einer Wand, 0 Leerraum. Die Map muss quadratisch sein (ansonsten crasht das Programm)
- `"size": int` Die Größe der Map. Muss dem tatsächlichen Wert entsprechen.
- `"start_pos": [x: int, y: int, r: int]` Die Startposition des Spielers. X und y sind Werte zwischen 0 und 512, sie geben die Position in Pixeln an. R ist zwischen 0 und 360 und ist die Drehung in Grad.
- `"entities": [], "enemies": []` Enthalten aktuell keine Werte und werden für zukünftigen Gebrauch freigehalten.

4.3 Bewegung

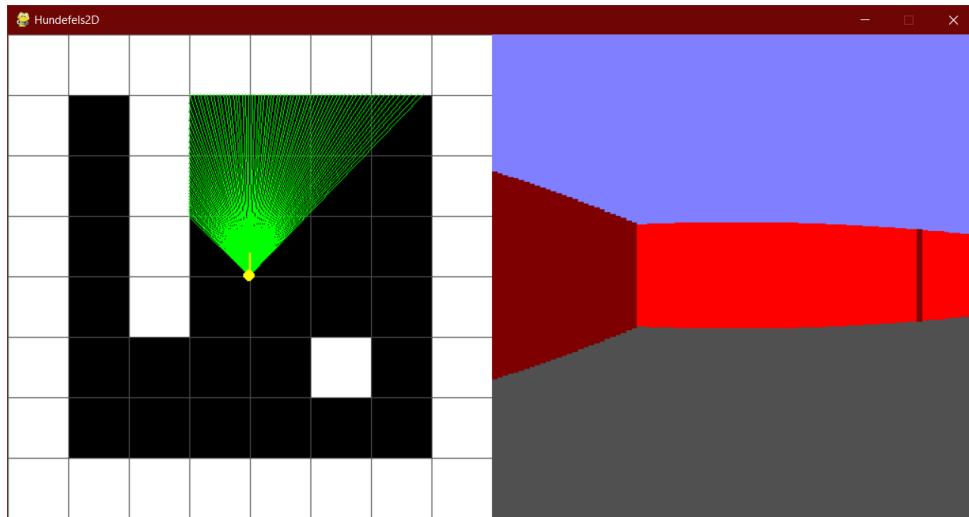
4.3.1 Translation (Laufen)

- Vorwärts: 'W'
- Links: 'A'
- Rückwärts: 'S'
- Rechts: 'D'

4.3.2 Rotation

- Links: linke Pfeiltaste (\leftarrow)
- Rechts: rechte Pfeiltaste (\rightarrow)

4.4 UI



Literatur

- [1] 3DSage: “Make Your Own Raycaster Part 1”
<https://youtu.be/gYRrGTC7GtA>
Quellcode verfügbar unter
https://github.com/3DSage/OpenGL-Raycaster_v1
- [2] Pygame tutorial: