

Games Engineering

Quake-Style Shooter from Scratch

Meilenstein 2

Bayer, Nico



DePaoli, Fabian

Eggers, Michael

Köhler, Benedikt

Rittenschober, Johann

Hochschule für angewandte Wissenschaften München

am 20. November 2024

Agenda

1. Was hat sich getan?

2. Demo & Level Design

3. Lightmapper

4. Audio

5. Gegner KI

6. Ausblick

7. Quellenangaben

Was hat sich getan?

- Input System
- In-Game-Console
- Lightmaps
- Türen reagieren korrekt auf Entities
- Pfade
- Bugfixes

Input System

Higher Level Game Code

```
...
CInputHandler* inputHandler = CInputHandler::Instance();
inputHandler->BindInputToActionName(SDLK_SPACE, "jump");
inputHandler->BindInputToActionName(SDLK_0, "equip_rocketlauncher");
inputHandler->BindInputToActionName(SDLK_w, "forward");
...
// Let player entity respond to input events
CInputDelegate::Instance()->SetReceiver(m_pPlayerEntity);
...
```

Entity implementiert Input Receiver Interface...

```
class Player : public BaseGameEntity, public IInputReceiver {
    ...
}
```

...um auf Input zu prüfen:

```
ButtonState jumpState = CHECK_ACTION("jump");
if (jumpState == ButtonState::PRESSED) {
    printf("I am jumping!\n");
}
```

Herausforderungen

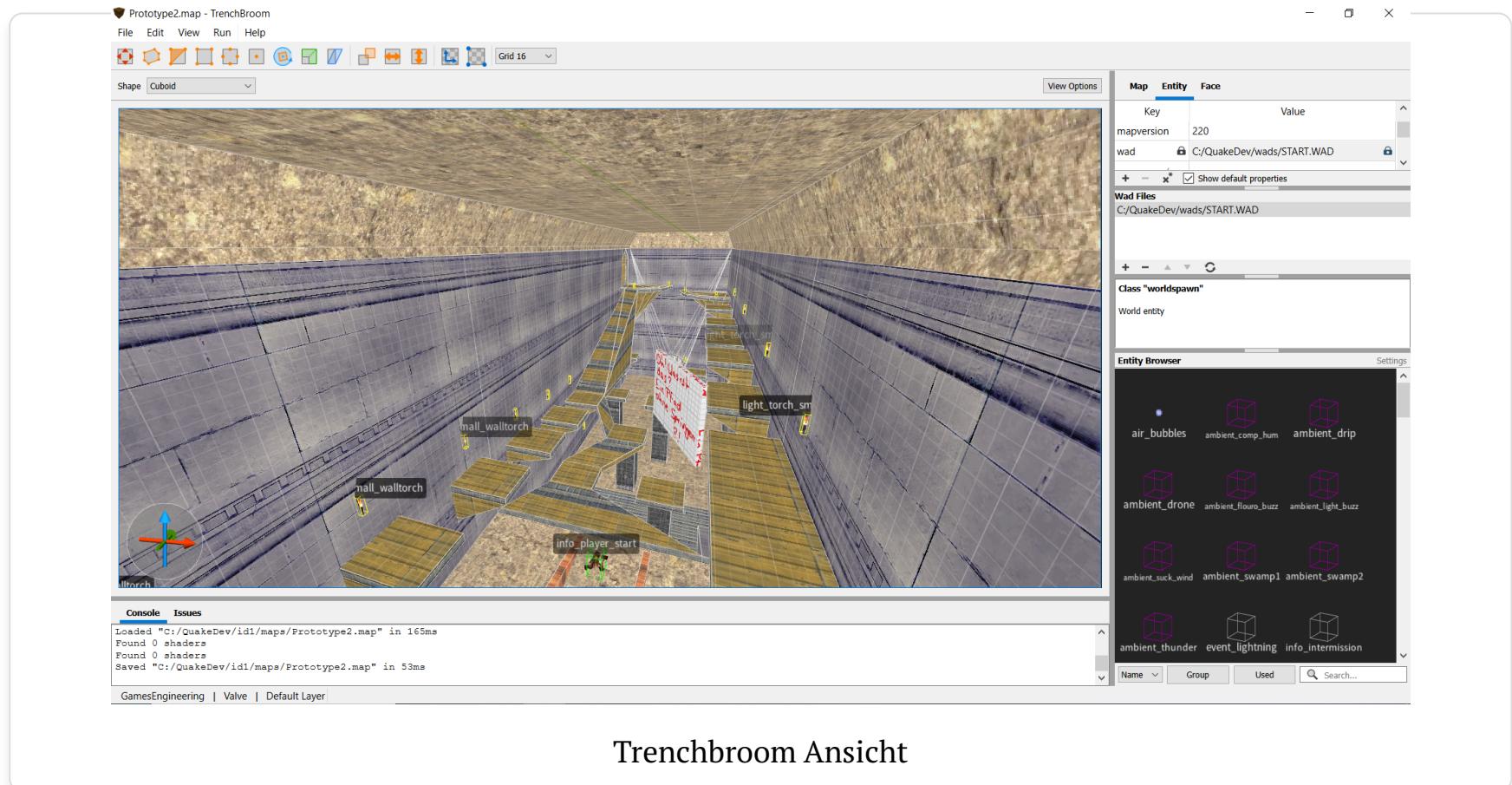
- Collision Detection: Schlechte Performance. **Alle** Entities prüfen jedes Frame auf Kollision mit **gesamter Welt**. Unterteilung in Octree würde Abhilfe schaffen.
- Collision Detection: Nicht framerate unabhängig.
- Animationssystem: Schlechte Performance. Das Aufbauen der aktuellen Pose ist im Moment sehr teuer. Wahrscheinlich schlechte Implementierung. Optimierung über Compute Shader wäre möglich.
- Entity System: Wir überlegen auf ein **Actor-Component** Modell wie in Unreal Engine umzustellen, um den Code innerhalb einer Entity etwas aufzuräumen gleichartiges Verhalten syntaktisch zu komprimieren.
- Engine unterstützt ausschließlich TGA-Files für Welttexturen.

Zu implementierende Features

- First Person Camera
- HUD Rendering für Lebensanzeige und Fadenkreuz
- Schießen auf Gegner
- Gegner AI weiter ausbauen (Navigation durch Welt, Interaktion mit Spieler)
- Eigene Gegner-Modelle
- Audio
- Lightmaps in Engine integrieren
- Memory Manager
- Virtuelles Filesystem integrieren: PhysicsFS (<https://icculus.org/physfs/>)

Demo & Level Design

Trenchbroom Ansicht



Trenchbroom Ansicht

Trim Sheets für die Texturen



Boden Edelstein



Trim Sheet Standard



Trim Sheet Stein



Trim Sheet Stone

Lightmapper

Ziele für Meilenstein 2

- Integration des Lightmappers
- Texturen
- Verbesserte UV Zuordnung
- Performance
- Lineare Textur-Filterung
- Renderer verbessern

Ziele für Meilenstein 2

- Integration des Lightmappers
- Texturen
- Verbesserte UV Zuordnung
- Performance
- Lineare Textur-Filterung
- Renderer verbessern

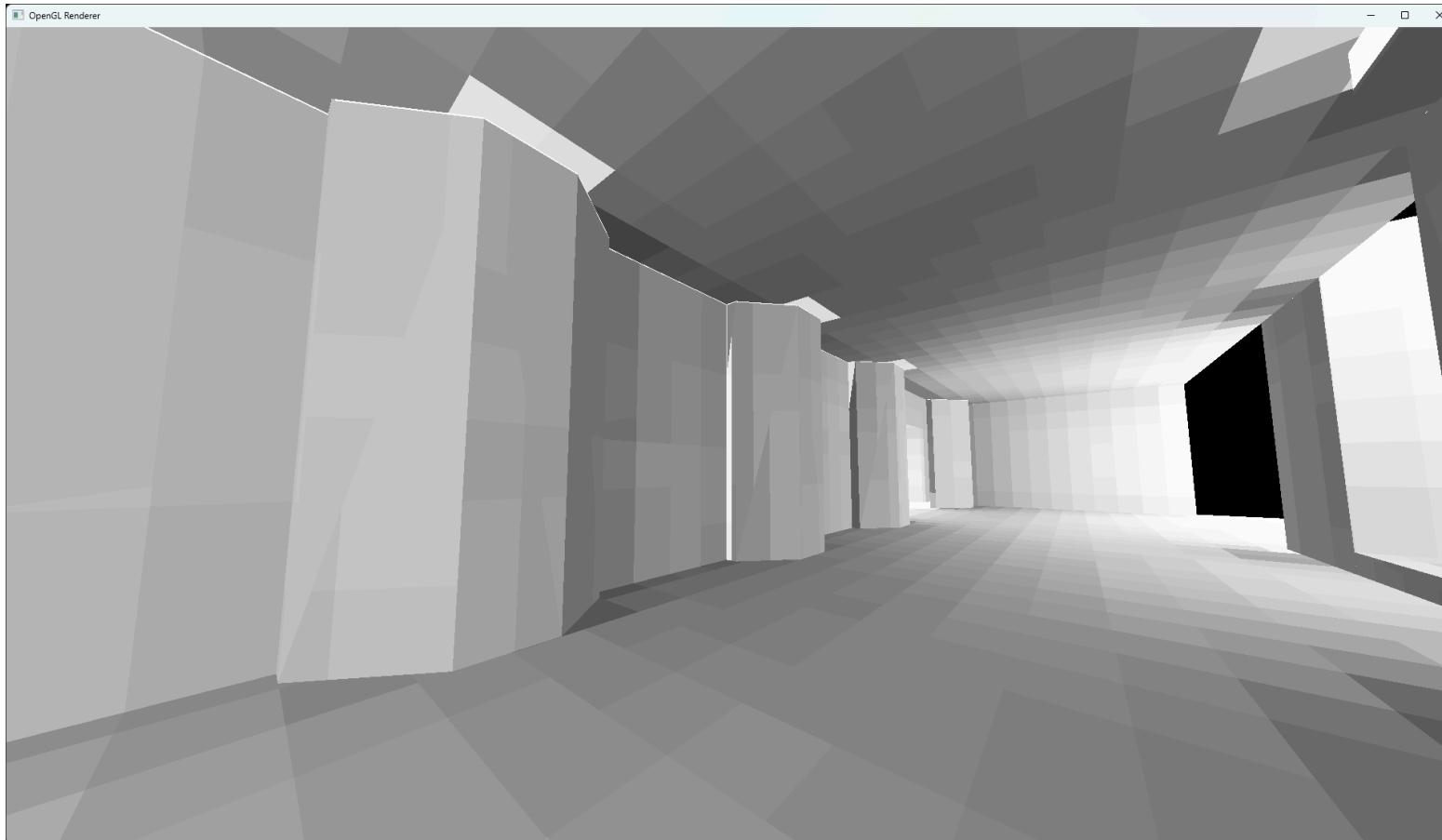
Ziele für Meilenstein 2

- Integration des Lightmappers
- Texturen
- Verbesserte UV Zuordnung
- Performance
- Lineare Textur-Filterung
- Renderer verbessern
- Usability
- Plattformübergreifende Kompatibilität
- Hemicube Platzierung

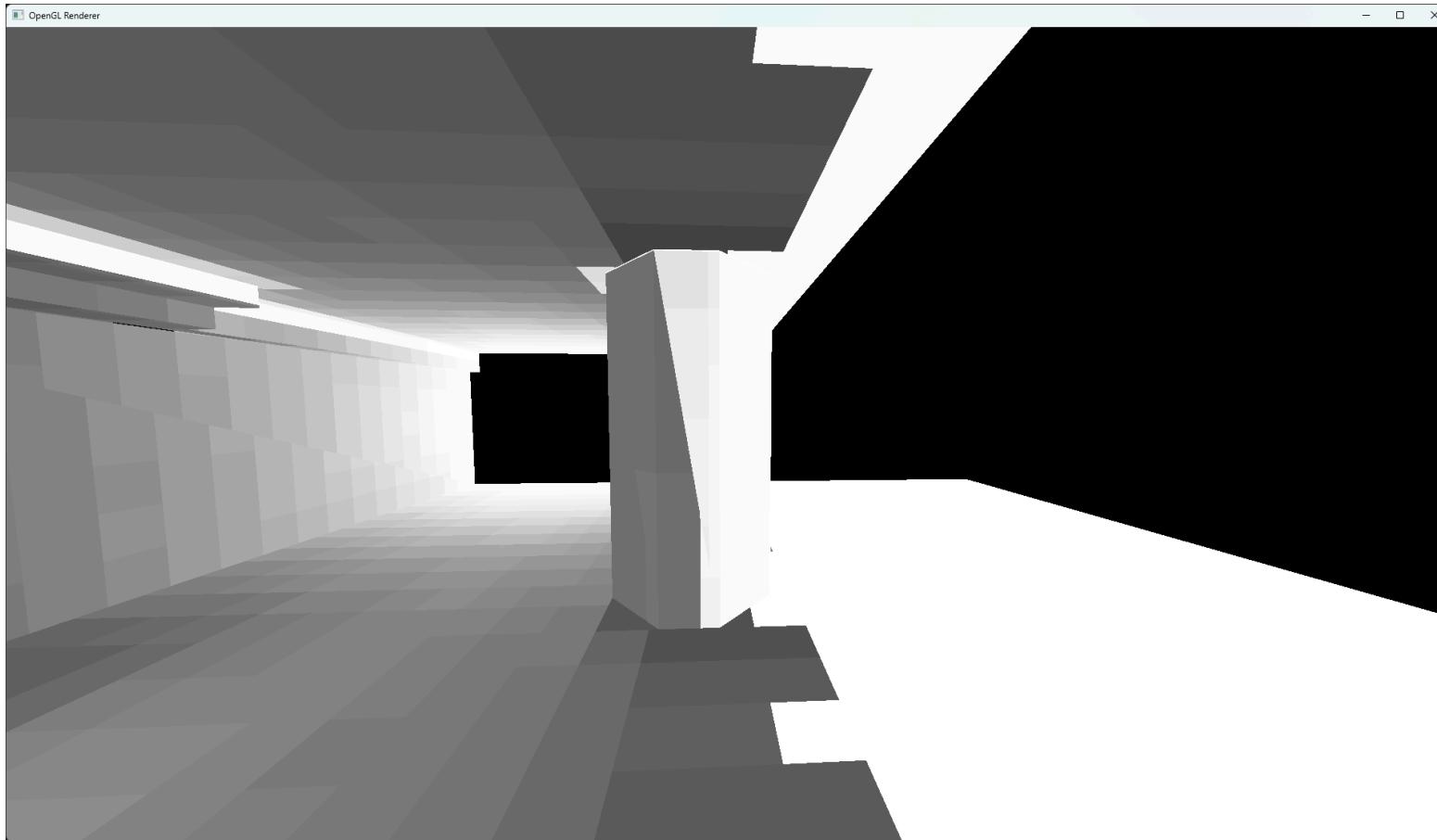
Ziele für Meilenstein 2

- Integration des Lightmappers
- Texturen
- Verbesserte UV Zuordnung
- Performance
- Lineare Textur-Filterung
- Renderer verbessern
- Usability
- Plattformübergreifende Kompatibilität
- **Hemicube Platzierung**

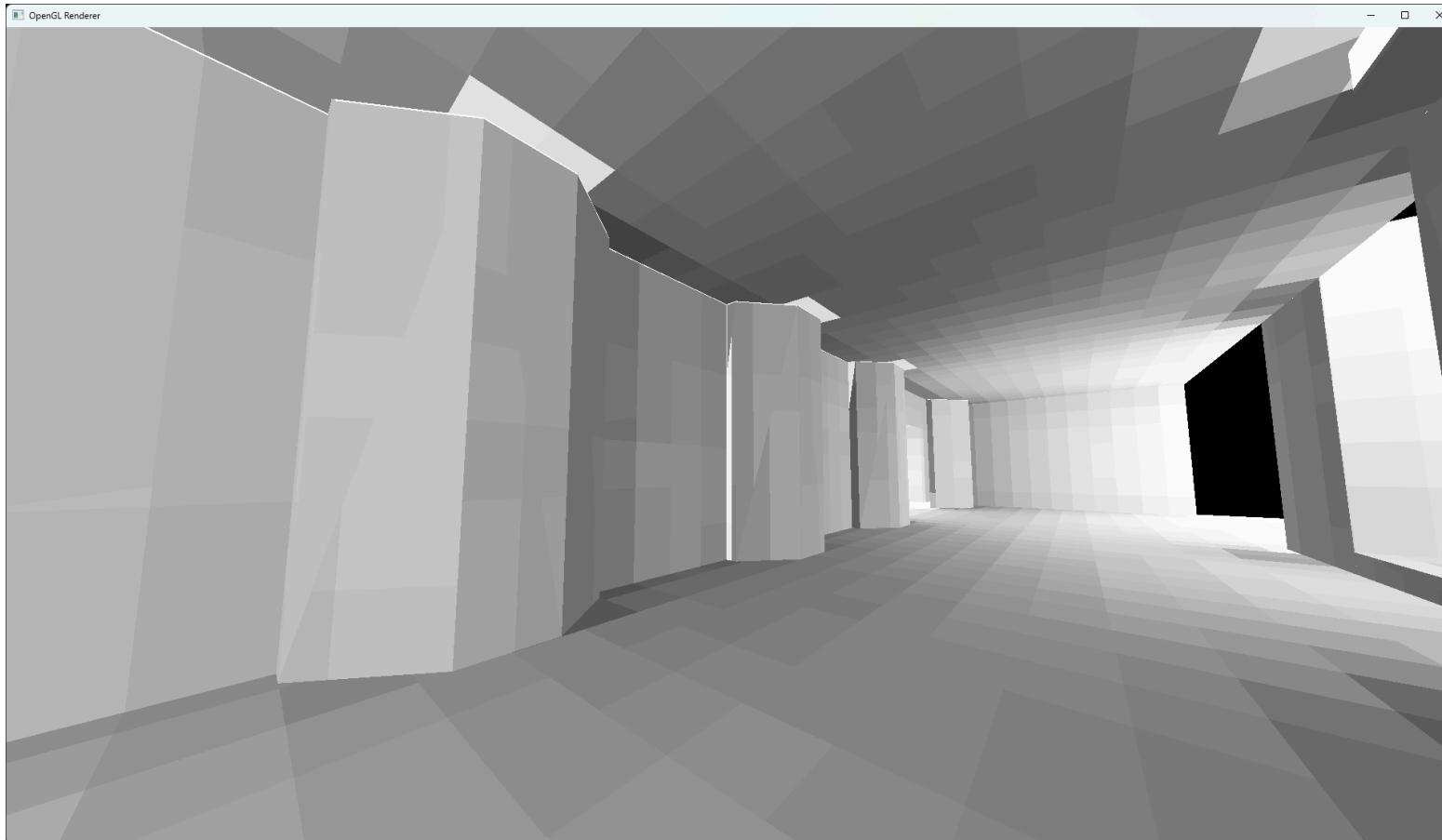
Hemicube Platzierung



Hemicube Platzierung



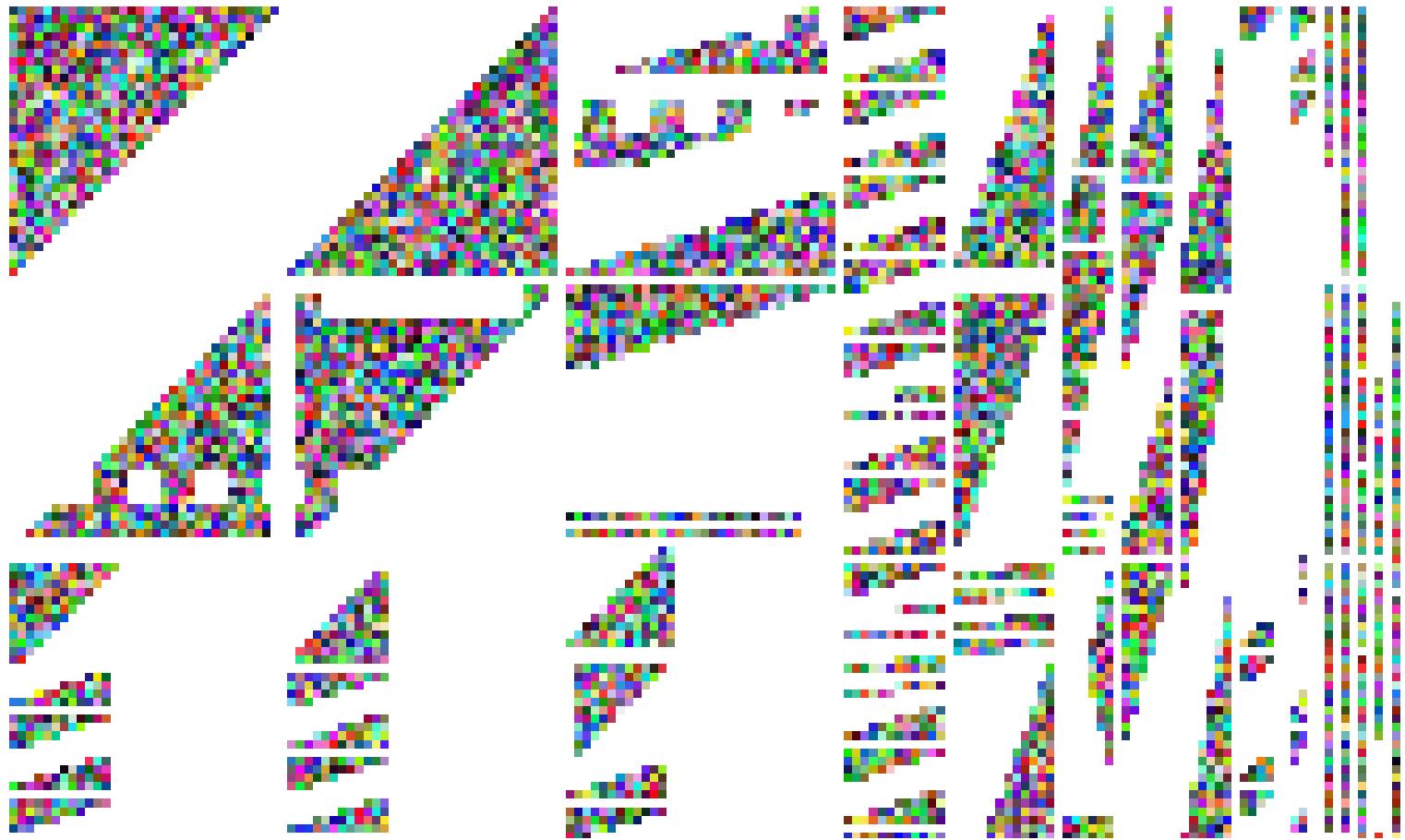
Hemicube Platzierung



Hemicube Platzierung



Hemicube Platzierung



Hemicube Platzierung



Hemicube Platzierung



Ziele für Meilenstein 3

- Patch Processing Pipeline
- Dynamic Patch Resolution
- Verbessertes UV-Mapping
- Bugfix Renderer

Audio

Audio (Library)

OpenAL

- + scheinbar sehr mächtig
- + speziell für 3D Game-Audio designed
- sehr Low-Level, API ähnlich wie OpenGL
- keine direkte 2D-Audio Unterstützung
- keine Einstiegs-Dokumentation vorhanden, nur Spezifikation

miniaudio

- + Feature-Umfang:
2d/3d audio, Effekte per Node-Graph System, Mixing, ...
- + Low-Level und High-Level API
- Dokumentation (und Code-Snippets) etwas schwer zu überblicken
- Ungenauigkeiten bzgl. Feature-Umfang

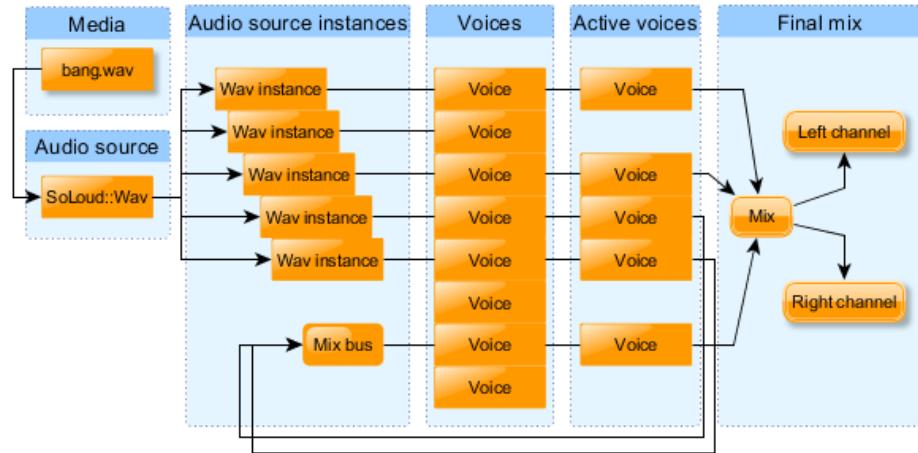
SoLoud

- + Dokumentation (und Code-Snippets) leicht zu verstehen
- + Feature-Umfang:
2d/3d audio, Effekte, Mixing, integrierter Synthesizer, ...
- + miniaudio als Backend enthalten
- ggf. langsamer / weniger optimiert als (lower-level) Alternativen
- evtl. eingeschränkt für weiter fortgeschrittene Use-Cases

Audio (Implementierung)

Notwendig (Basics):

- 2D Audio für Musik / Ambiente
- 3D Audio für positionierte Emitter (z.B. Gegner)
- Einbindung in System- / Entity-Events
(z.B. UI Clicks, Schuss-Trigger, Footsteps, etc.)



Nice to have:

- Spiel-Zustand für Sounds (z.B. Wechsel von 'idle' zu 'combat' Modus)
- Sound-Randomisierung (z.B. mehrere Samples, volume/pitch/effect Moodulationen, etc.)
- Prozedurale Sounds mit dynamischer Länge (z.B. Türen, bewegbare Plattformen, etc.)
- Räumliche Sound-Effekte (Räume, obstruction / occlusion / attenuation, etc.)

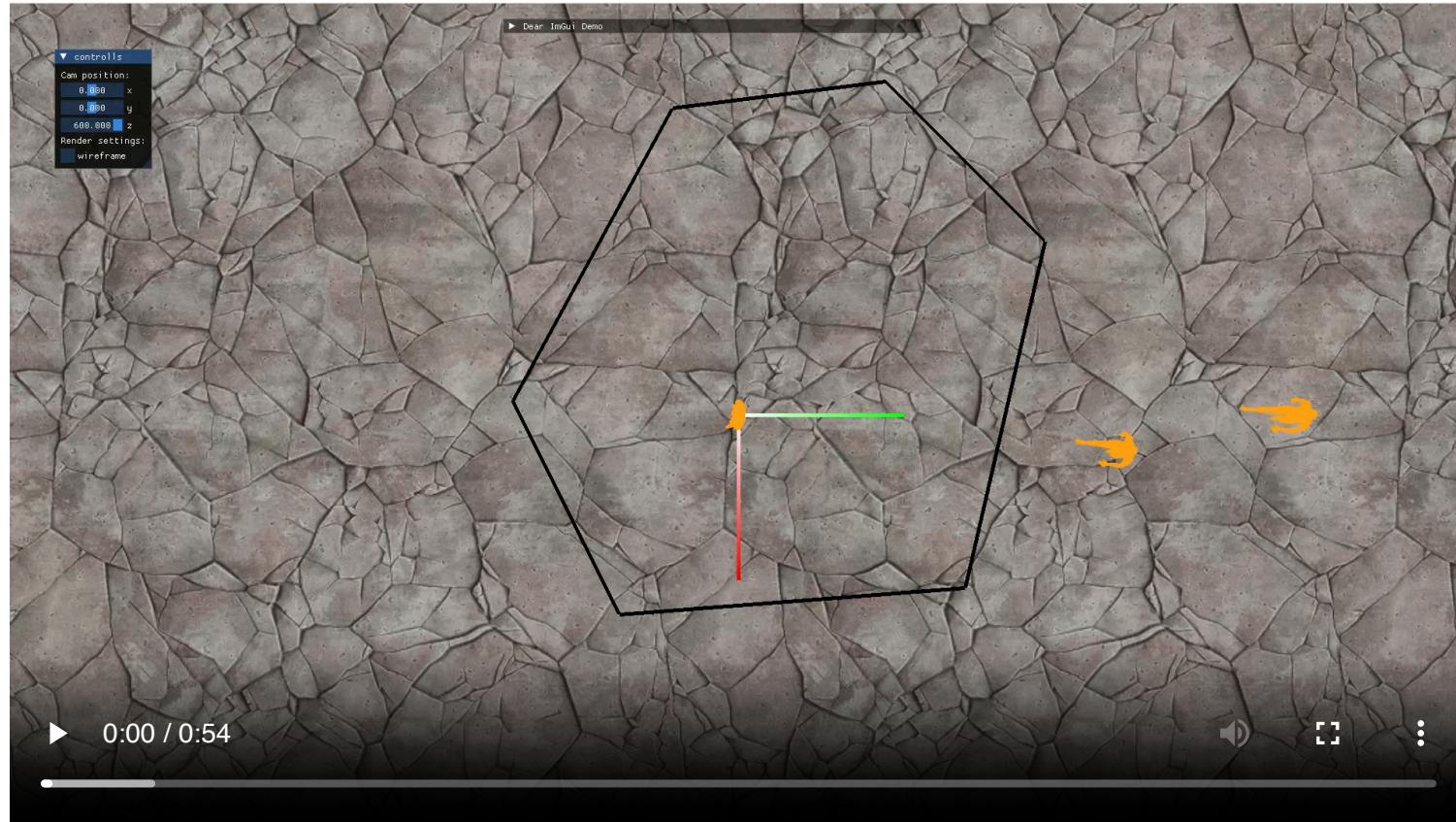
Gegner KI

Gegner KI

Was ist passiert?

- Random Walk
- Seek, Flee, Arrive
- Follow Path, Follow Waypoints

Das ist passiert



Steering Behaviours

Craig Reynolds beschreibt 3 Layer (1)

- Action Selection
- Steering
- Locomotion

Steering Force

Vereinfachtes Model um Steering Force zu berechnen:

$$\text{steeringForce} = \text{desiredVelocity} - \text{currentVelocity}$$

Wie wird Kraft zu Bewegung

2. Newtonsches Gesetz wird abgeleitet zu:

$$\mathbf{f} = \mathbf{ma}$$

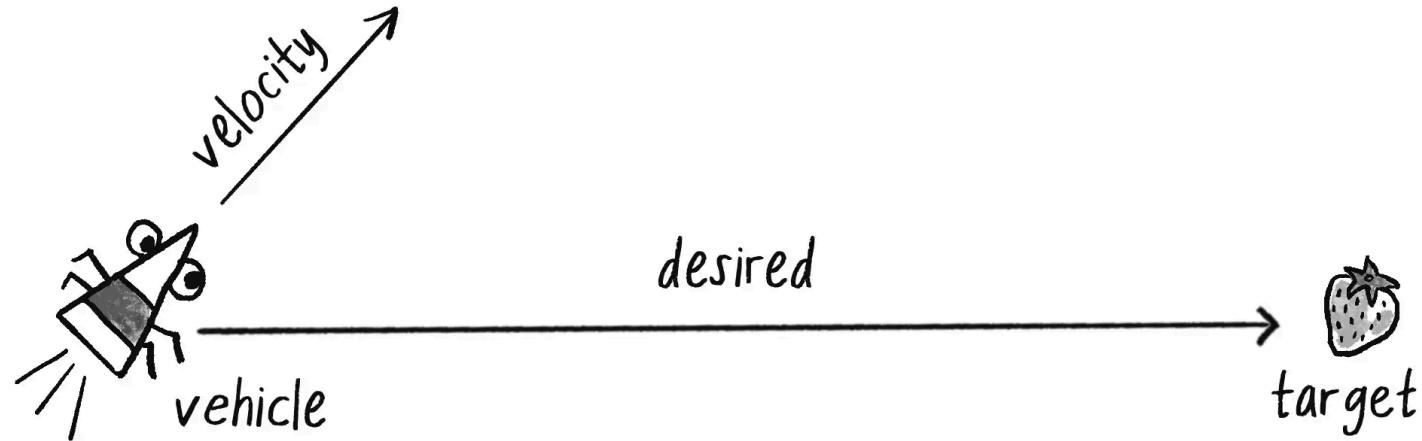
$$\mathbf{a} = \frac{\mathbf{f}}{m}$$

$$\rightarrow \mathbf{v}_{\text{new}} = \mathbf{v}_{\text{old}} + \mathbf{a}$$

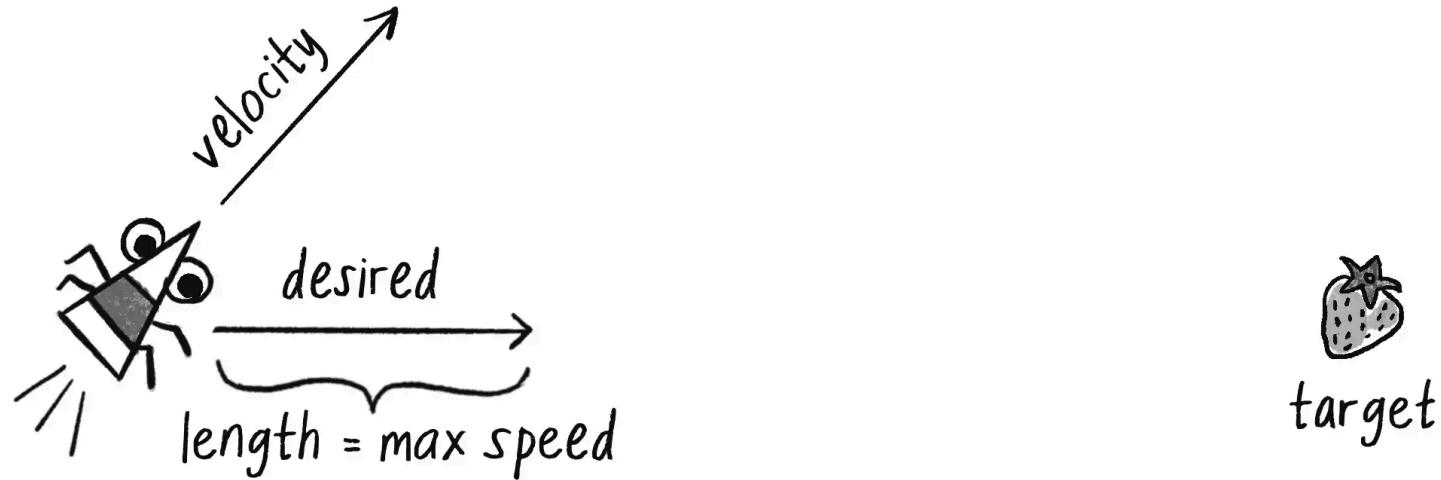
Steering Force



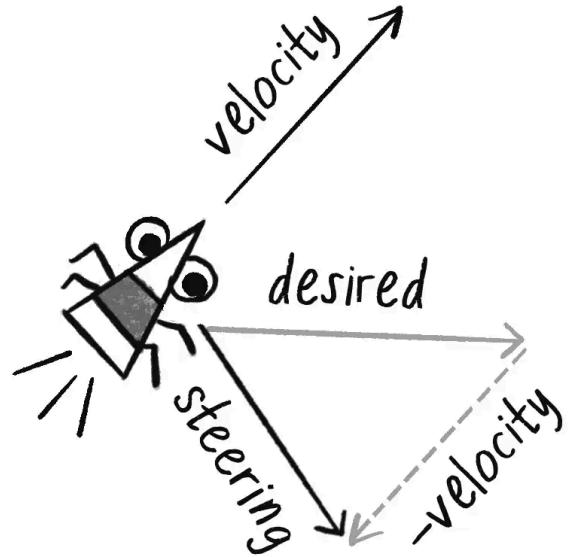
Steering Force



Steering Force



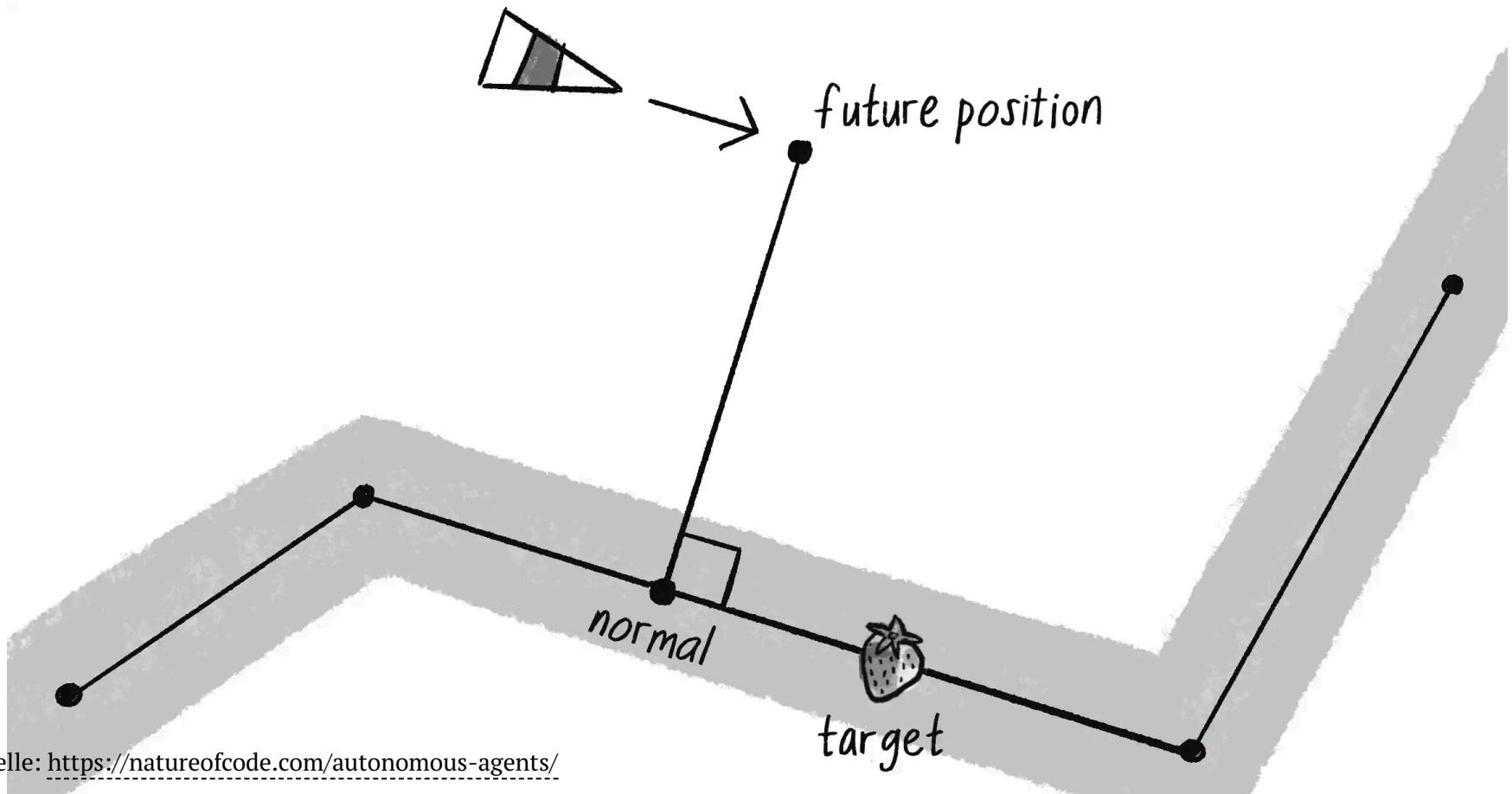
Steering Force



$$\text{steering} = \text{desired} - \text{velocity}$$



Pfad Verfolgung



Ausblick

- Bug Fixing
- State Machine & Steerign Behaviours zusammen bringen
- weitere State Machines (Verhalten) bauen

Ausblick

Quellenangaben

- 1. Reynolds CW. Flocks, Herds, and Schools: A Distributed Behavioral Model. Siggraph [Internet]. 1987; Available from: <https://www.red3d.com/cwr/papers/1987/SIGGRAPH87.pdf> (21)
- 2. Shiffman D. The Nature Of Code. No Starch Press®, Inc.; 2024. (24)