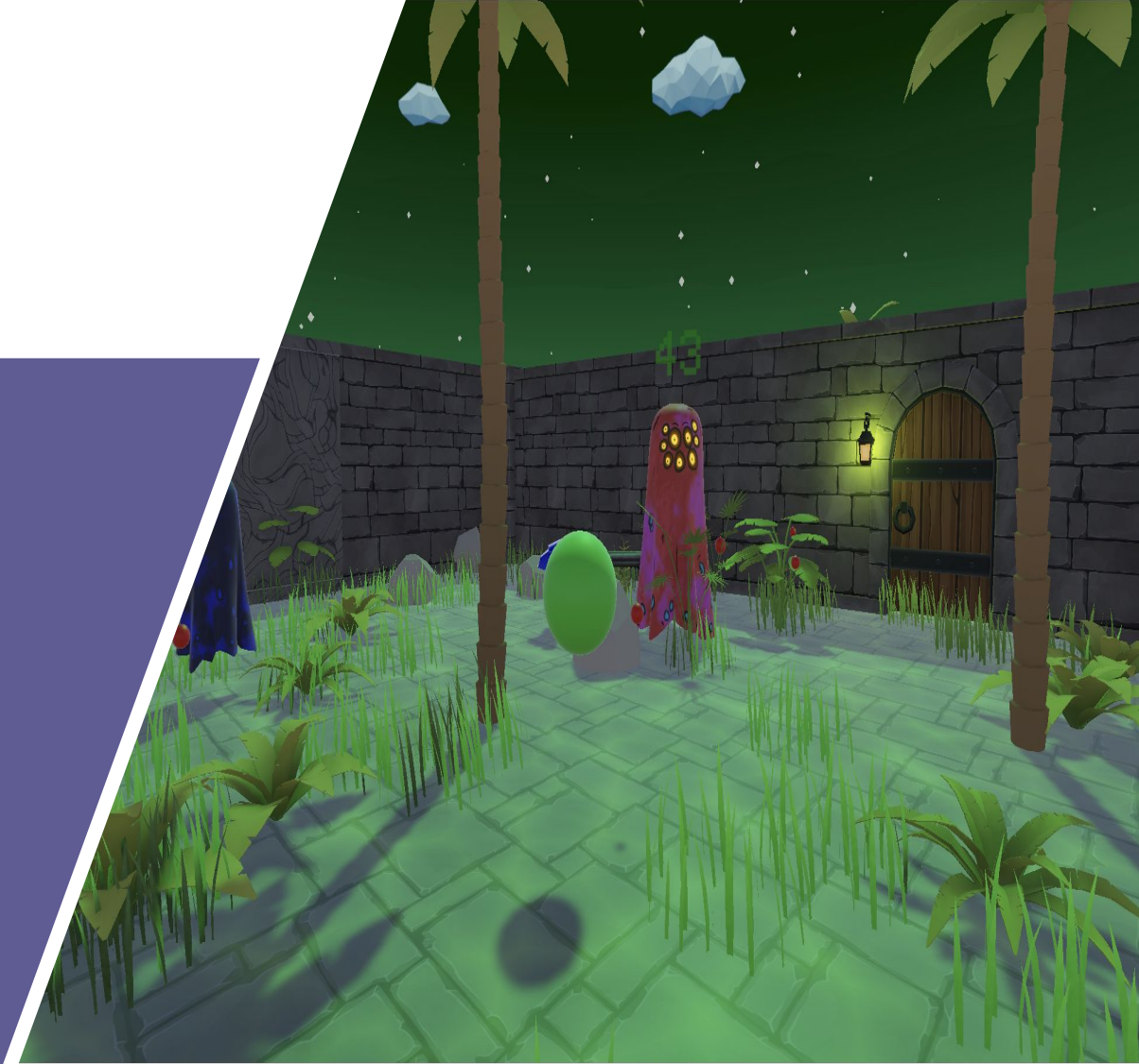


Projektarbeit 24/25

# Entwicklung einer KI- und Computergrafik-gestützten Spielumgebung mit Unity

Haitham El Euch, Thorben Meiswinkel, Arman Niaruhi, Peter Kletschka

24. September 2025



# Inhaltsverzeichnis

Einleitung

Organisation & Kommunikation

Spielprinzip & Spielflow

Aufbau & Logik des Spiels

Bedienung & UI

Live-Demo

Herausforderungen & Lösungen

Mögliche Erweiterung

Fazit & Links

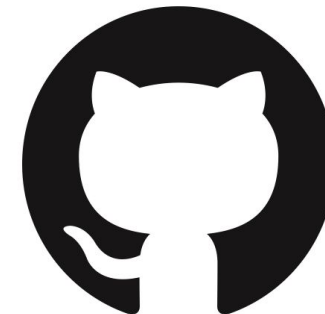
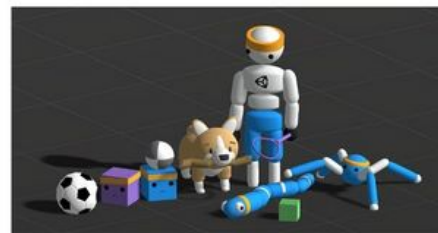
# Einleitung

# Einleitung

- Dungeon-Crawler mit KI, Minispielen und prozeduralen Levels
- Bewegung auf einer Ebene, prozedural generiert.
- Tech: Unity 6 (URP), C#, GitHub, Blender, ML-Agents (ONNX), Sentis/Tensor.
- Ziel: spielbarer Prototyp mit deterministischem Seed-Run.

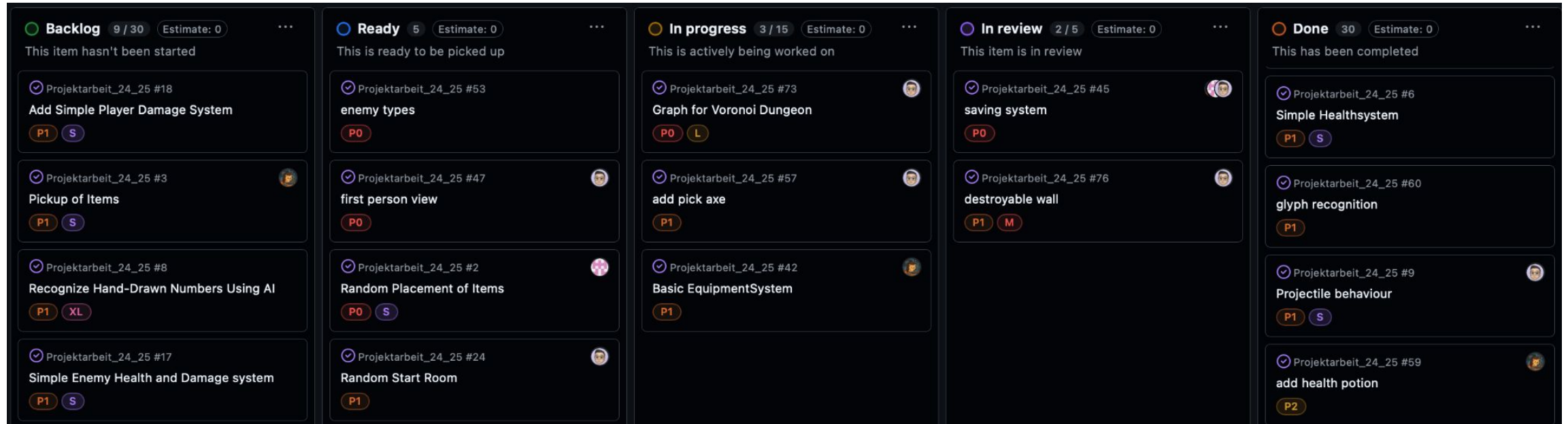


## ML-Agents



# Organisation & Kommunikation

# Organisation & Kommunikation



# Spielprinzip & Spielflow

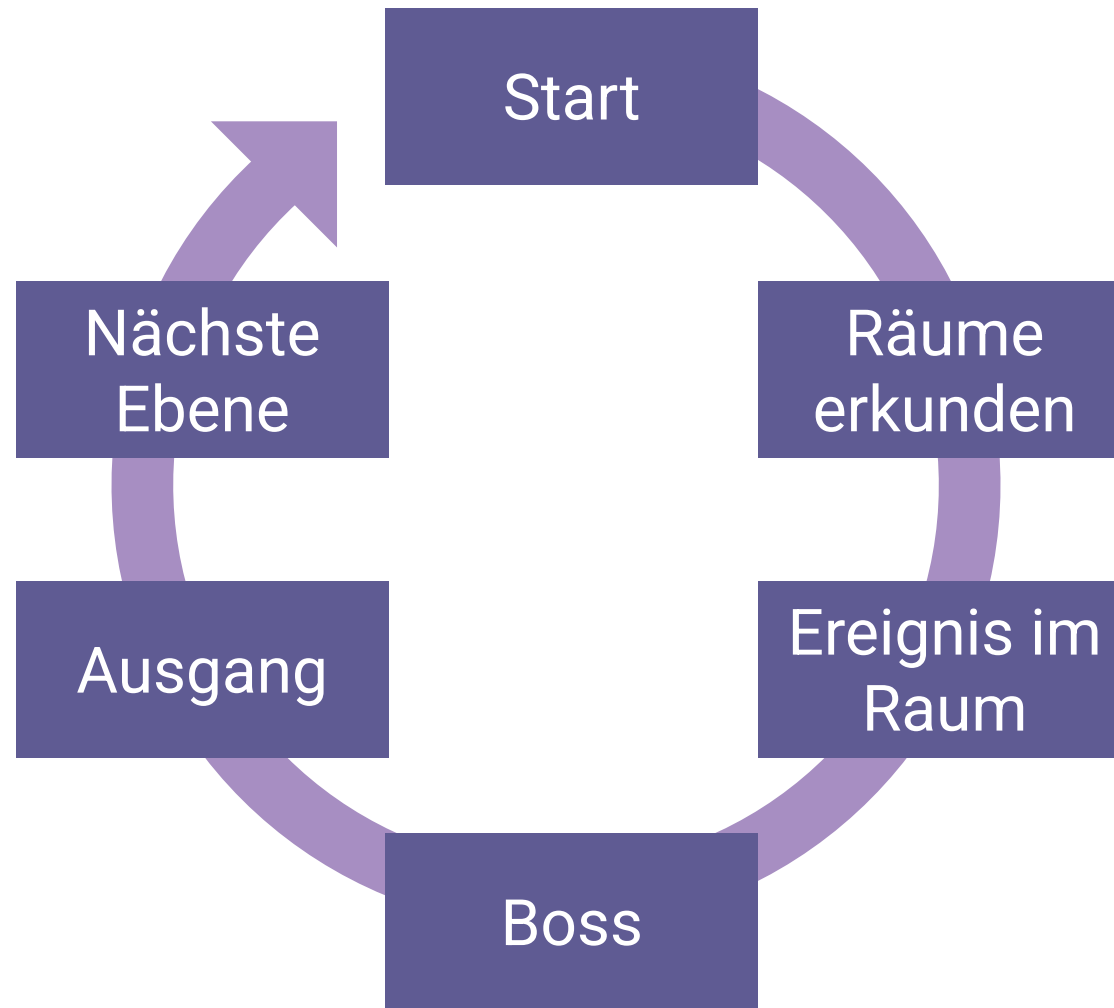
# Spielprinzip

Prinzip:

- möglichst weit kommen
- Ausrüstung & Belohnungen sammeln
- Minispiele lösen
- immer schwierigerer Boss besiegen



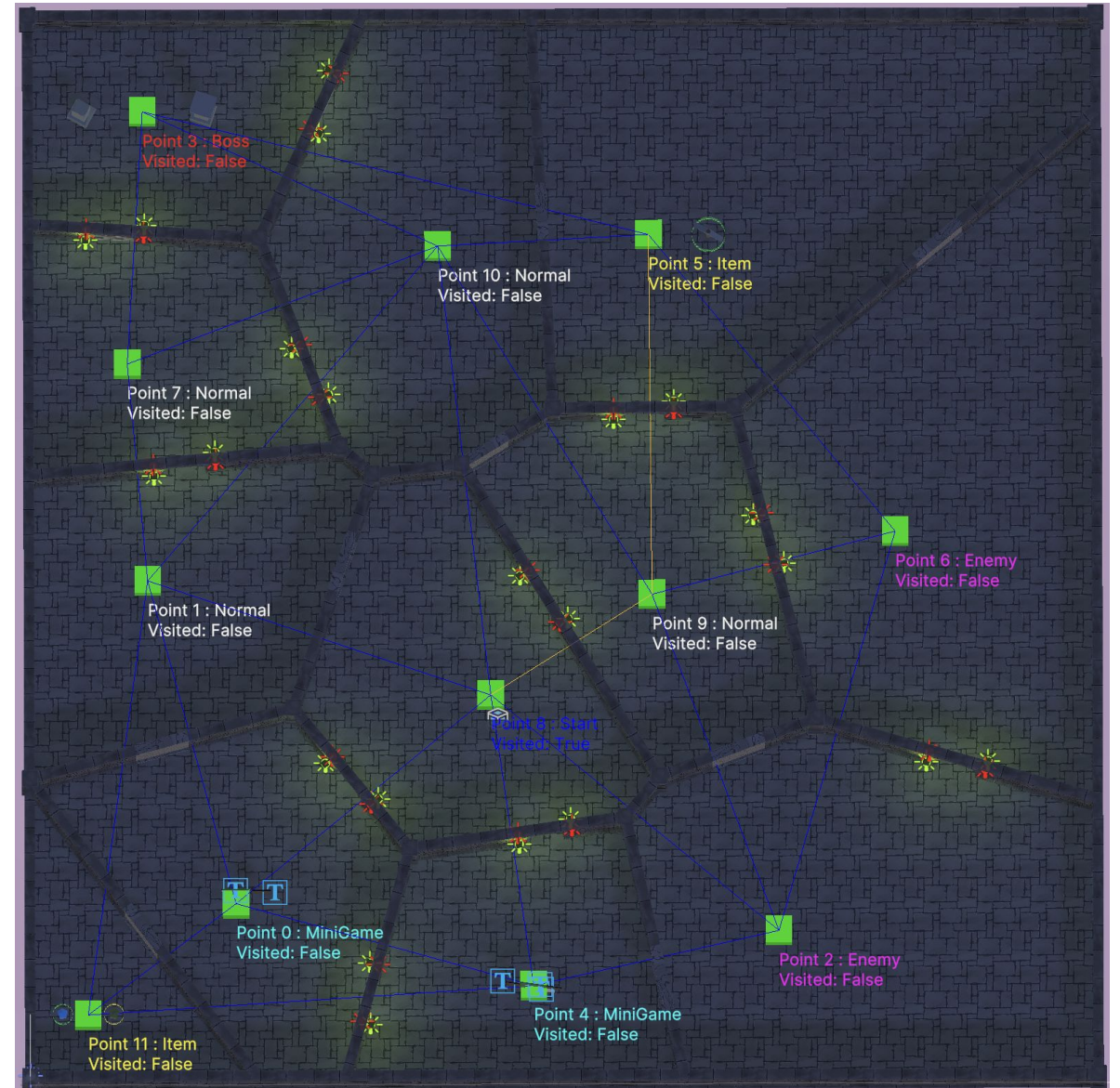
# Spielflow



# Aufbau & Logik des Spiels

# Dungeon-Generierung

- Voronoi aus Delaunay/Bowyer-Watson -> Dungeon-Graph (stabile Raum-/Kanten-IDs)
- Türkandidaten nur auf langen Segmenten, kurze Kanten gefiltert
- garantierter Pfad vom Startraum zu einem Itemraum
- Bossraumtüren durch Nebelwände hervorgehoben
- Alternative zu Türen, zerstörbare Wände
- Manche Räume sind nur durch zerstörbare Wände erreichbar



# Verteilung & Platzierung

## Verteilung

- Verteilung anhand von Liste der Wahrscheinlichkeiten von Elementen
- Pflichtelemente werden vor zufälligen Elementen herausgegeben

## Platzierung

- Elemente innerhalb von Inkreis des Raumes
- kreisförmig um Mittelpunkt

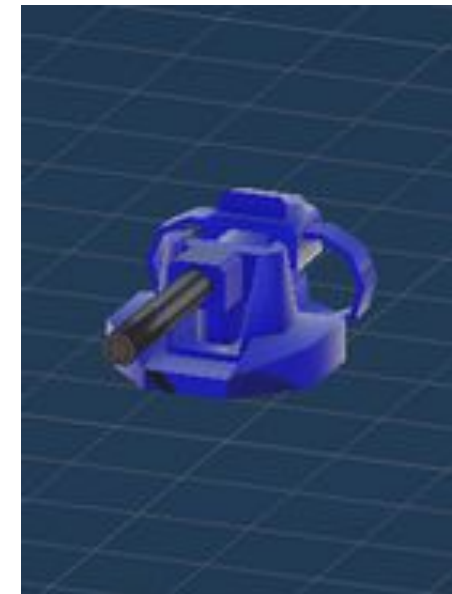
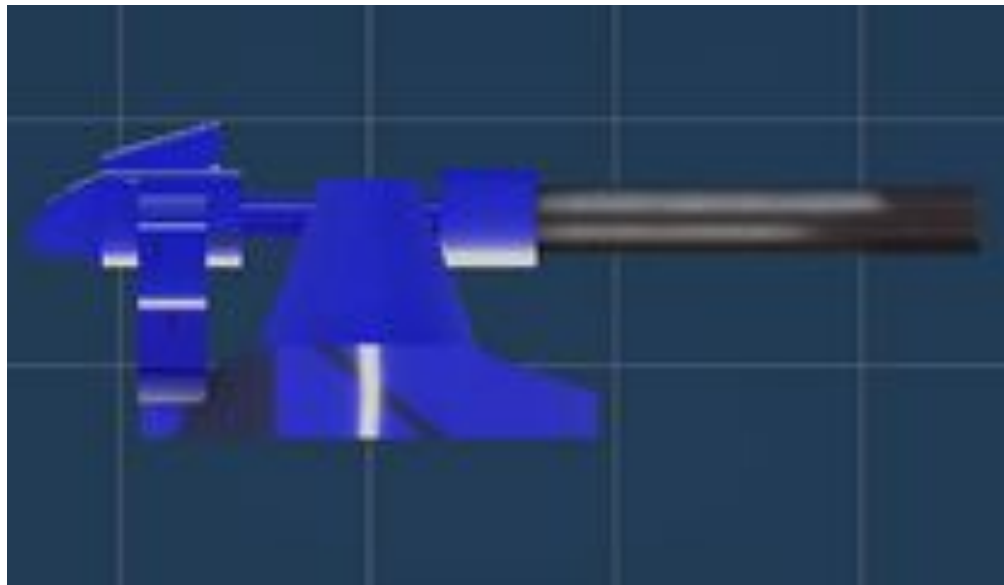
# Gegner & KI

- Nahkampfgegner
  1. Ghost
  2. Ministalker
  3. Bigstalker



# Gegner & KI

- Fernkampfgegner
  1. Drone
  2. Shooter



# Gegner & KI

- Boss-Gegner

Finder



Spawner (Drone)



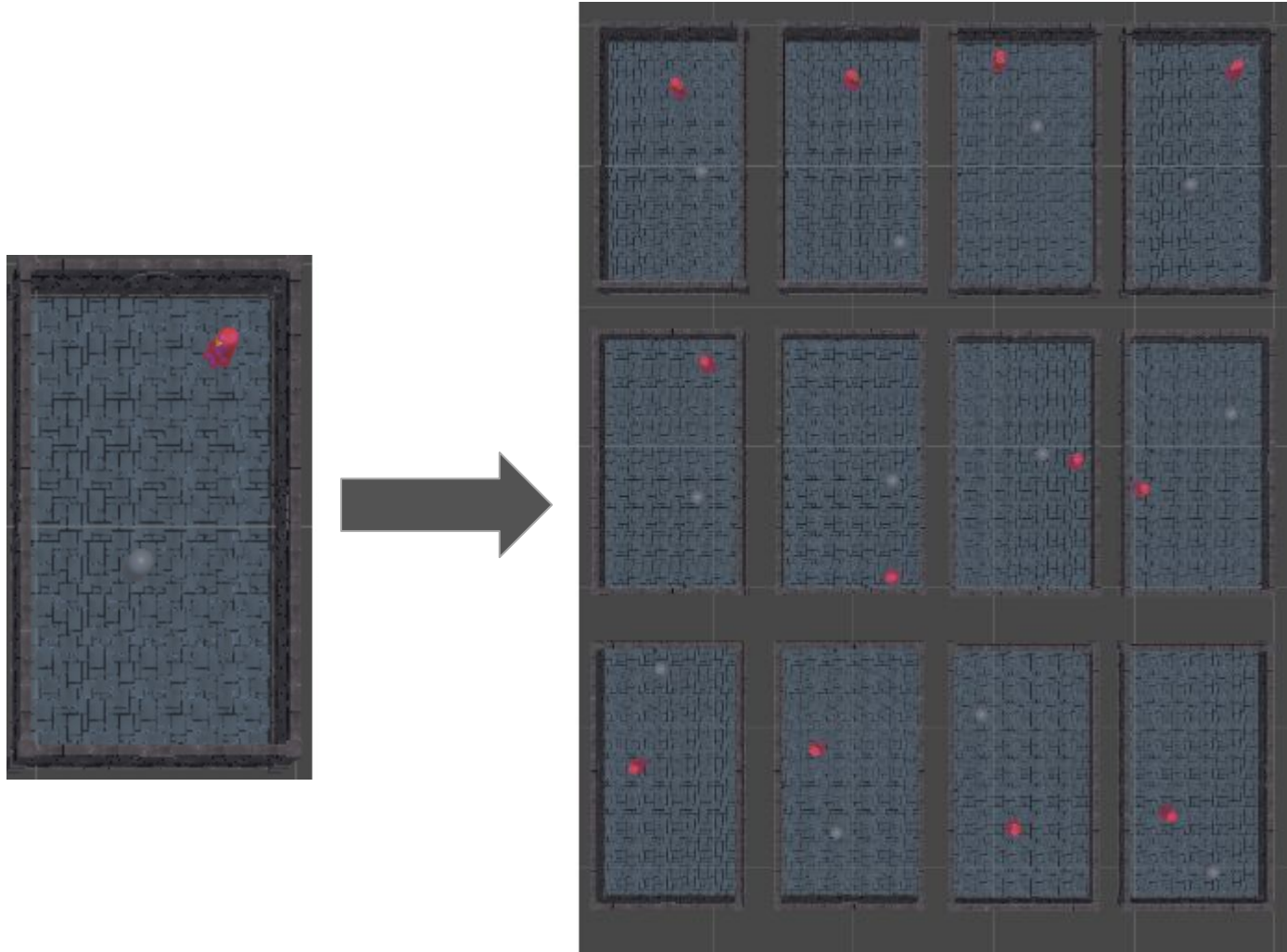
Spawner (Stalker)





# Gegner & KI

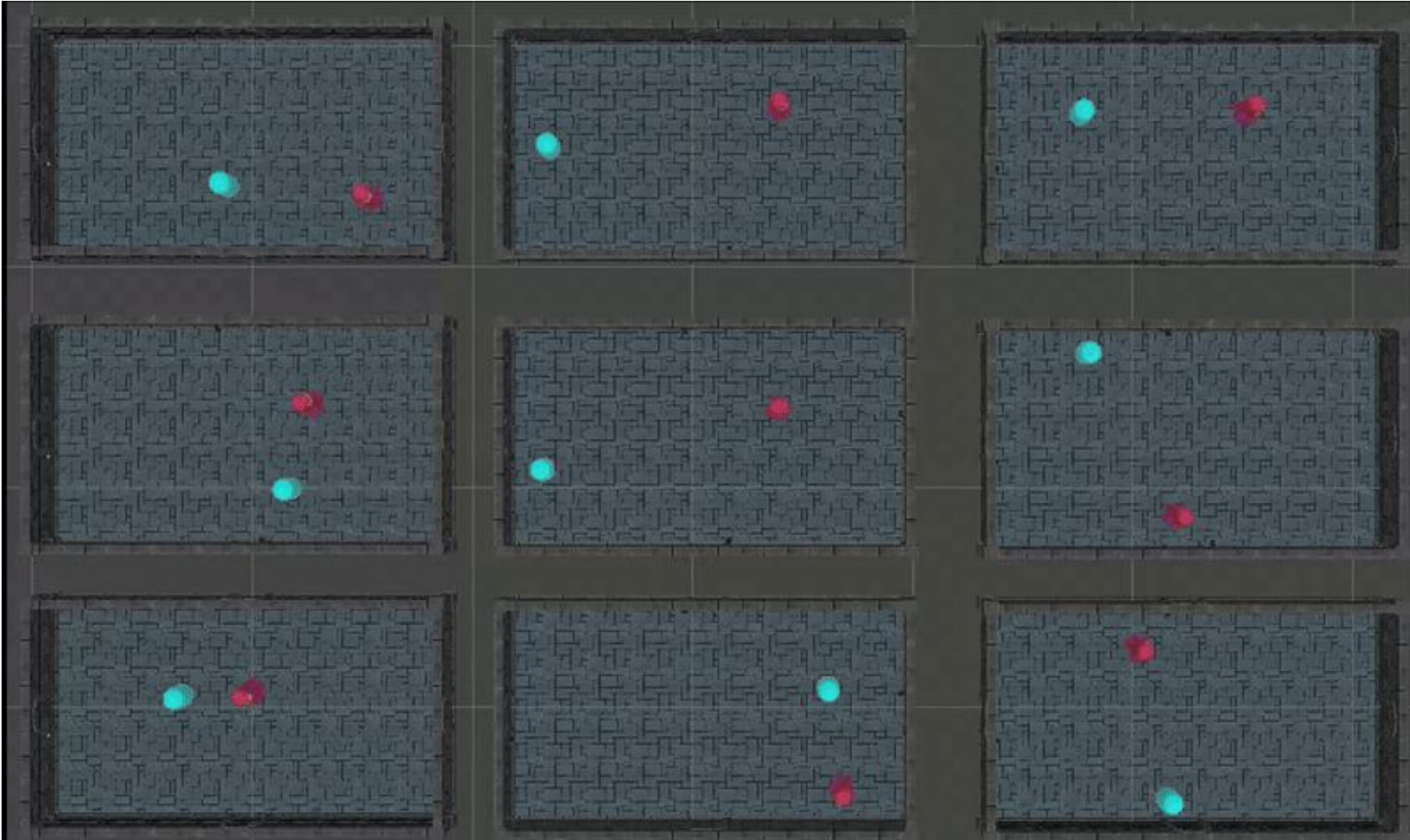
- Wie trainiert? ML-Agent
- Reinforcement-Learning





# Gegner & KI

>> 10x










# Gegner & KI

Type	Leben	Schaden	Geschwindigkeit	Spawn-Intervall
Player	100	5	7	-
BigStalker	50	15	3	-
MiniStalker	20	8	7	-
Shooter	100	-	-	-
Drone	60	-	-	-
Ghost	30	5	6	-
Finder	50	3	10	-
Spawnling1	500	20	5	5
MiniHunter	20	6	6	-
Spawnling2	500	20	5	4
Drone (Hunter)	30	-	-	-

# Minispiele (Glyphs)

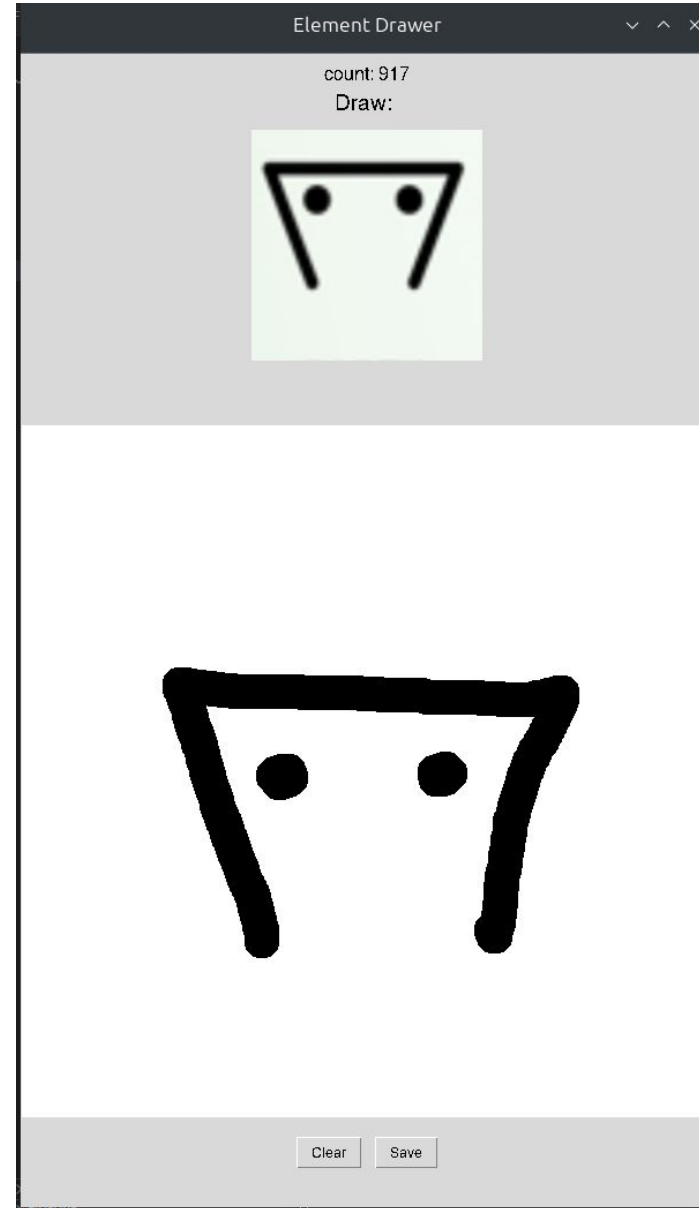
- Insgesamt 8 Glyphs

Glyph name	Air	Fire	Water	Earth	Energy	Time	Power	Light
Glyph sign								

- Unser eigener Datensatz in python

# Minispiele (Glyphs)

- TKinter in python
- Zeigt das Programm ein Glyph Zeichnen
- Alle Bilder automatisch in 28x28
- Trainiere CNN-Modelle für unsere Klassifizierung.
- Das Modell wird im ONNX-Format gespeichert.
- In Unity anwenden



# Minispiele (Glyphs)



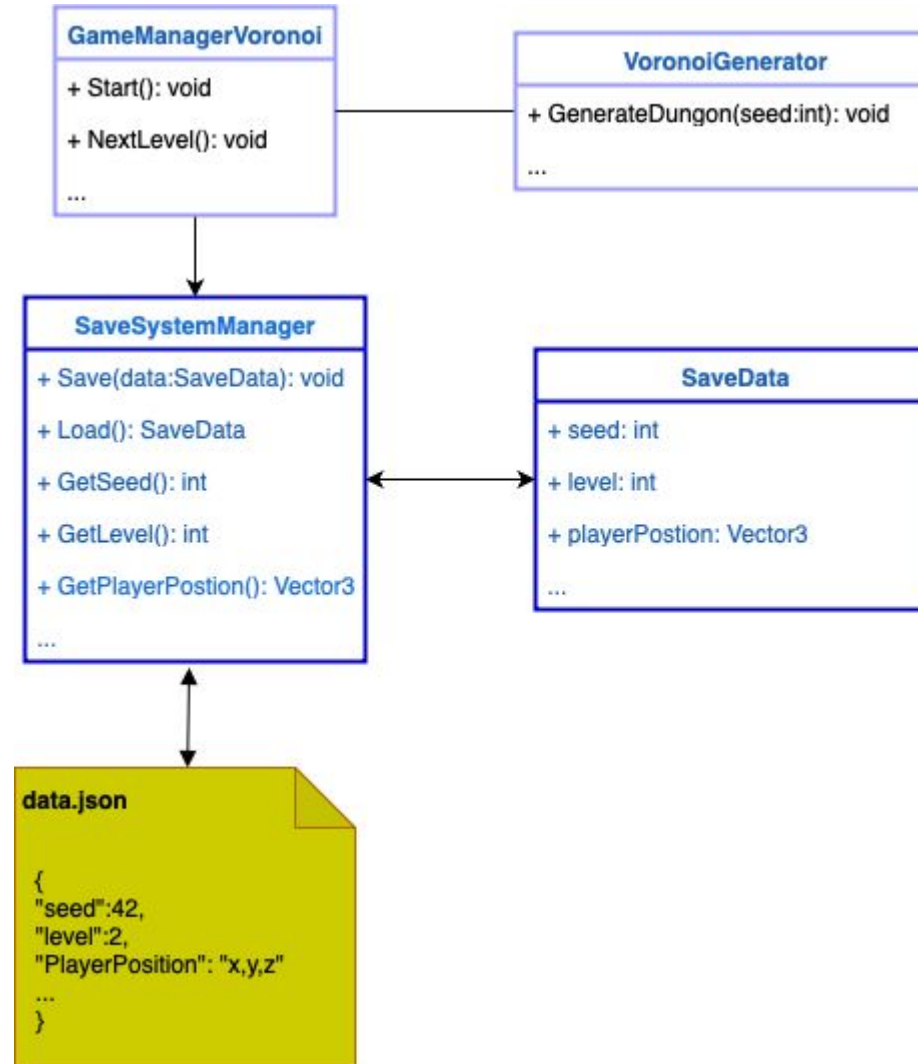
# Minispiele (Digit)

- Fragen-Pool: 10 Rätsel zu lösen
- ONNX von Digit- Erkenner
- 10 Fragen zu beantworten
- Preis: Der Eine Ring.



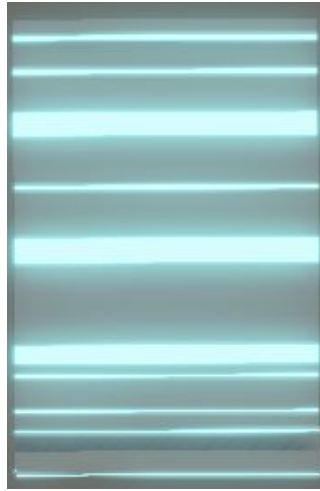


# Saving-System



# Shader

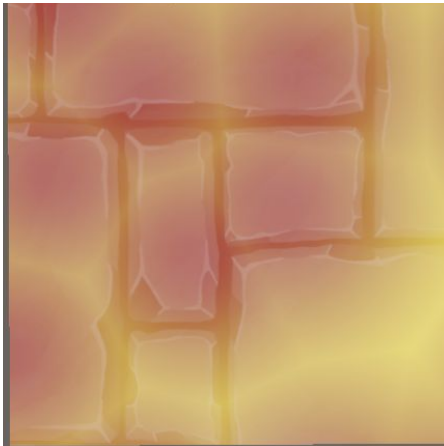
Emission Shader



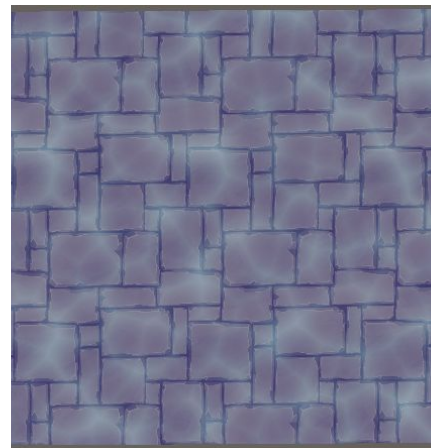
Fog Shader



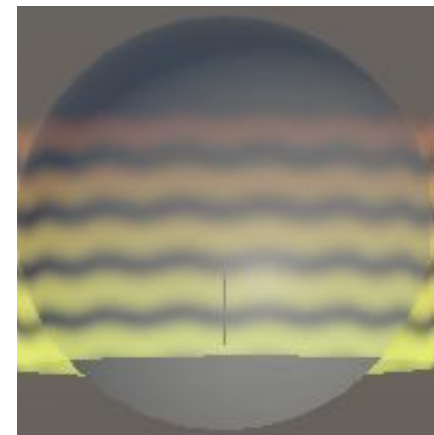
Twirl Shader



Ground Shader



Ground Shader

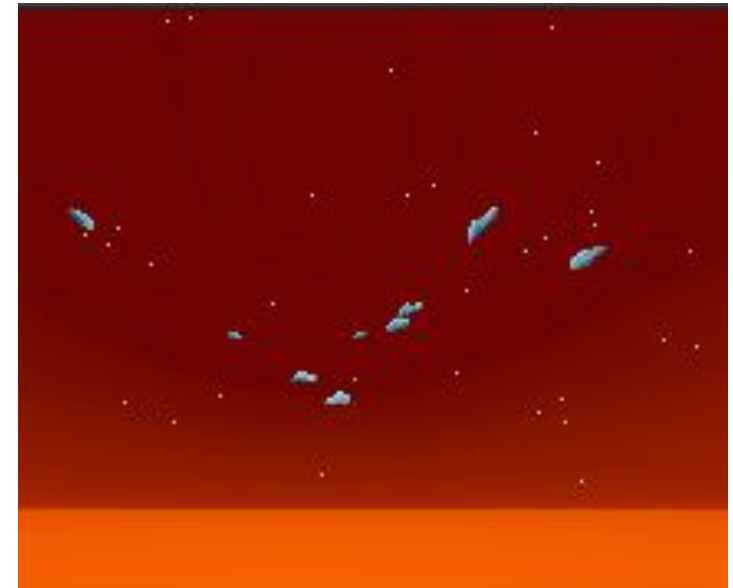


Aura Shader



# Beleuchtung

- Anpassung an Stil des Bodens, als auch der Skybox



# Bedienung & UI

# Steuerung

Aktion	Taste
Bewegen	W A S D
Umschauen	Maus bewegen
Schießen	Leertaste
Inventar öffnen oder schließen	I
Im Inventar ausgewähltes Item benutzen oder ausrüsten	E
Im Inventar ausgewähltes Item entfernen	O
Karte ein oder aus	M
Pause-Menü öffnen oder schließen	P
Minispielansicht betreten am Zeichenboard	G
Minispielansicht verlassen zurück zur First Person Ansicht	Tab
Auf dem Zeichenboard zeichnen	Linke Maustaste halten
Zeichen auf dem Zeichenboard auswerten	Rechte Maustaste
Zeichenfläche leeren	C
Spitzhacken Schlag an brüchiger Wand ausführen	Rechte Maustaste

# Inventar



Health 100 (x1)



Energy (x1)



Earth (x1)



Water (x1)



Pickaxe (x1)









































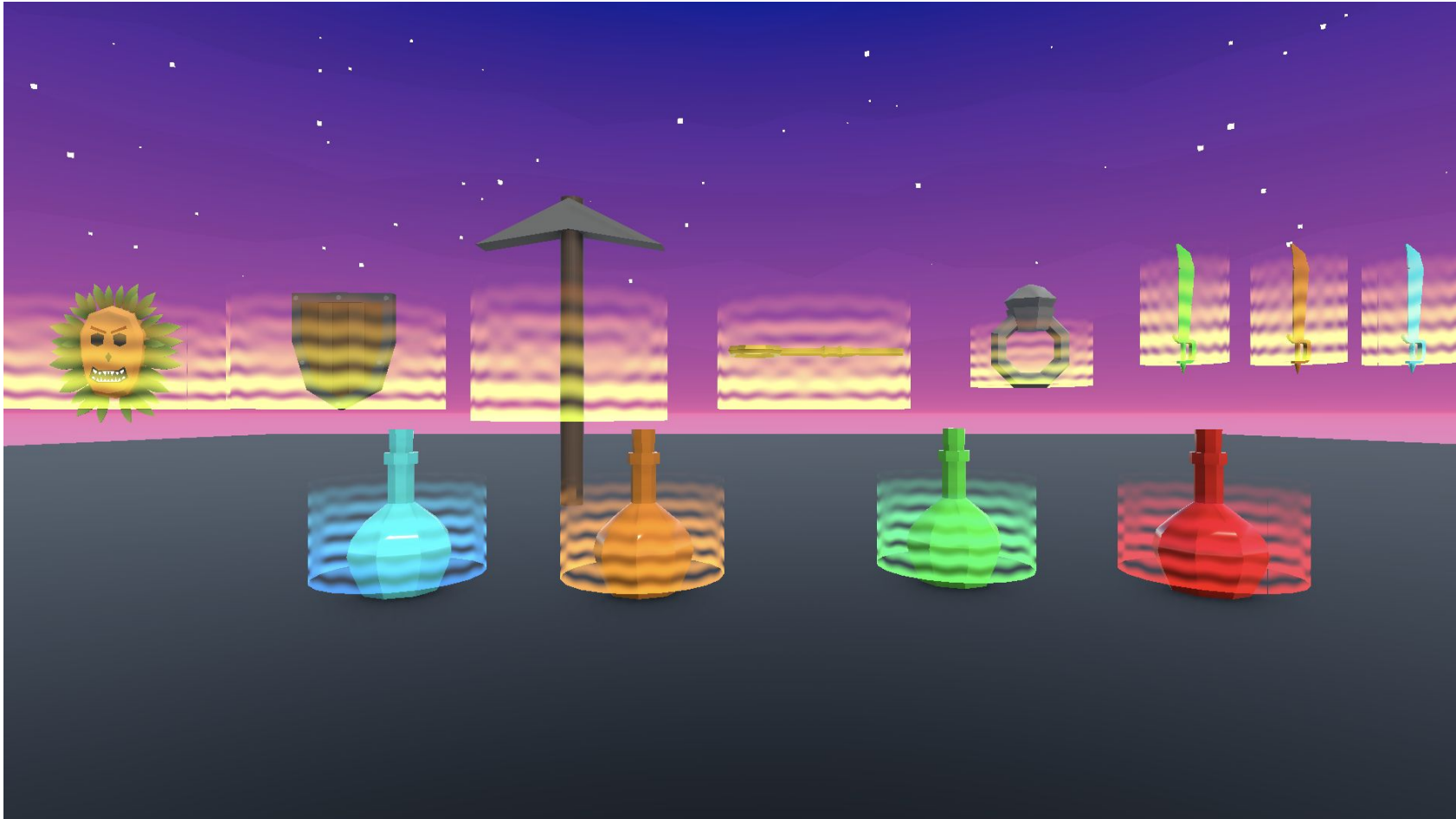


Health: 100

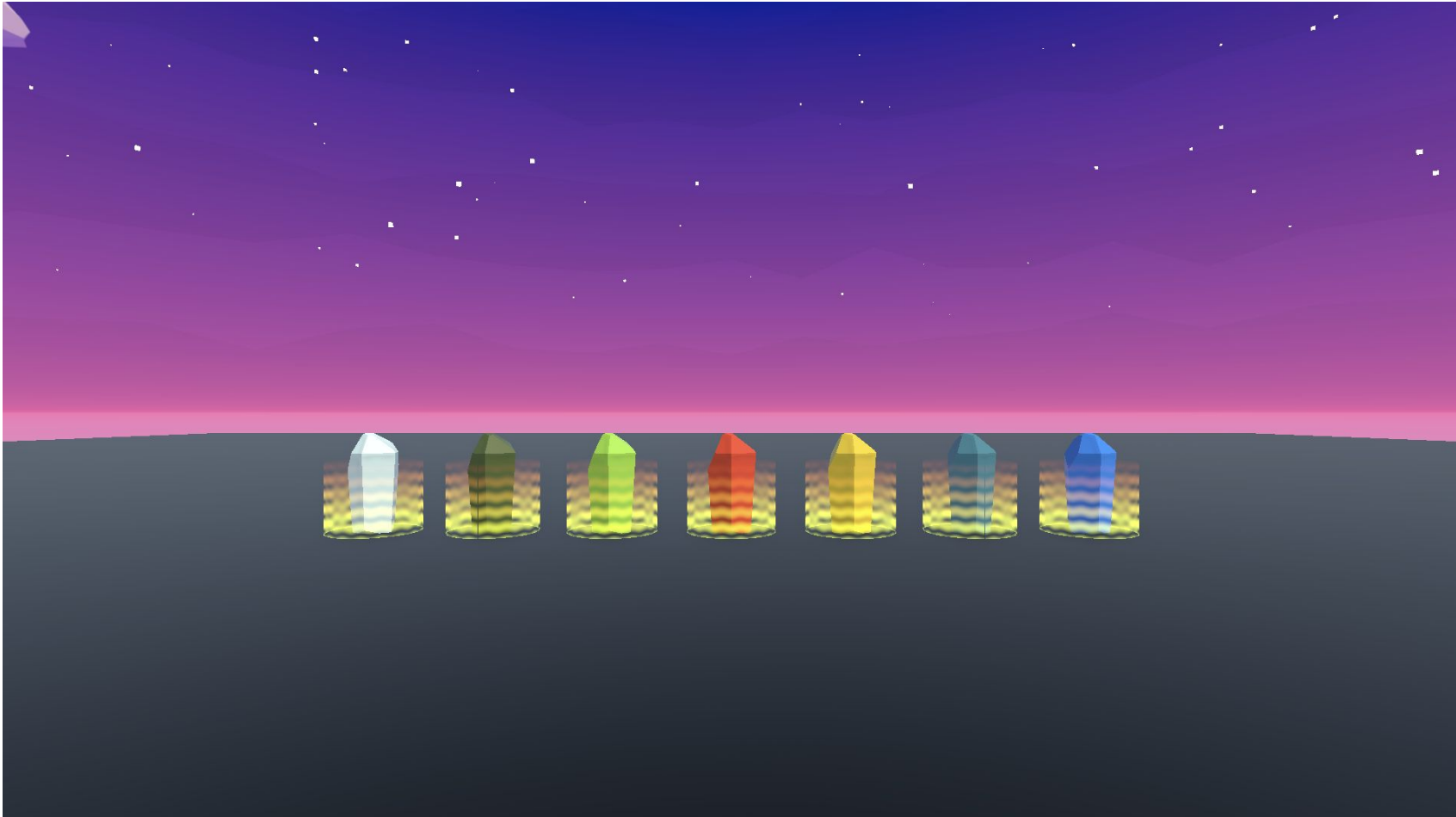
Damage:5

Speed:7

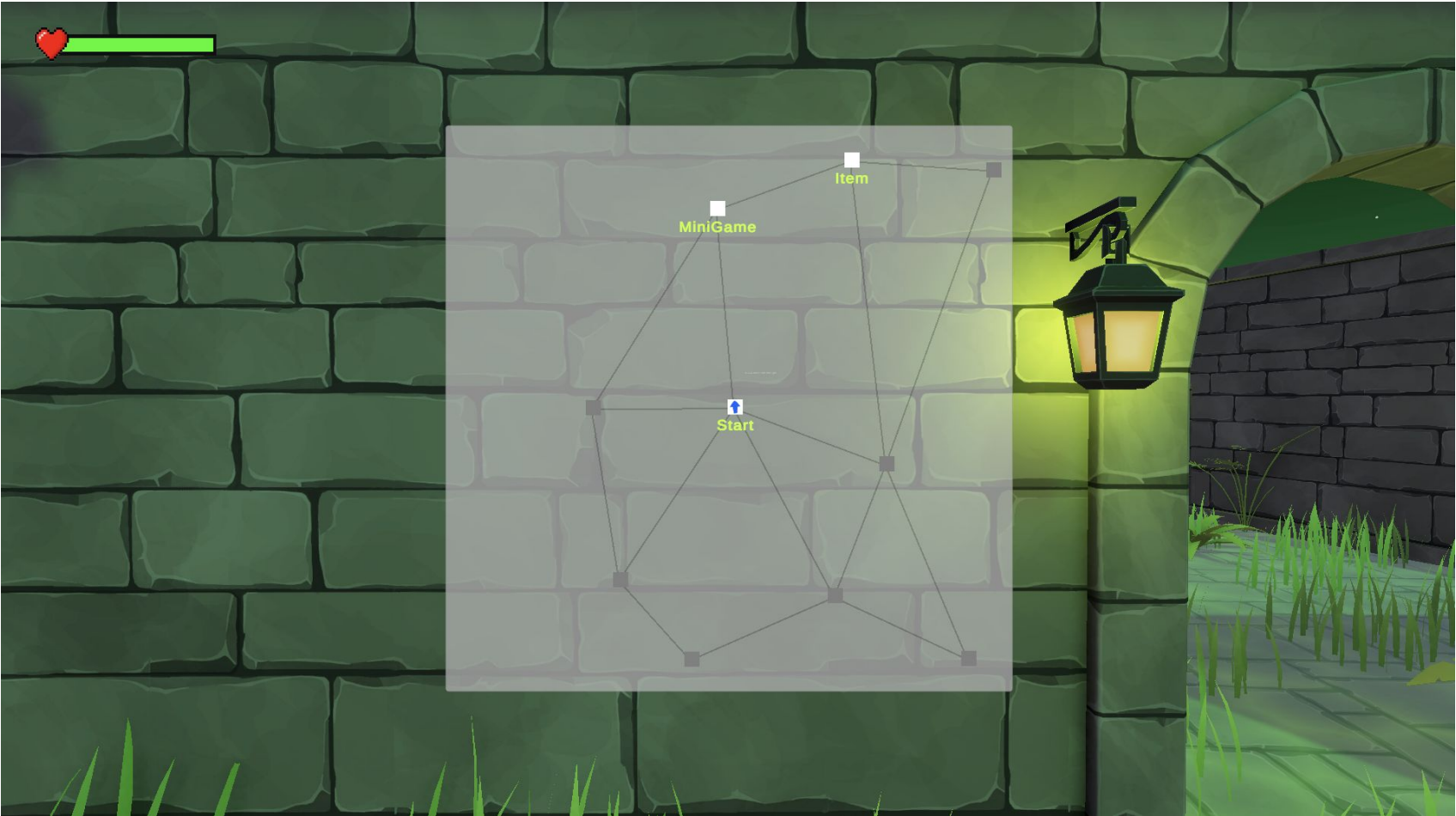
# Items



# Items



# Karte



# Startbildschirm





# Pause-Menü



# Game Over-Bildschirm

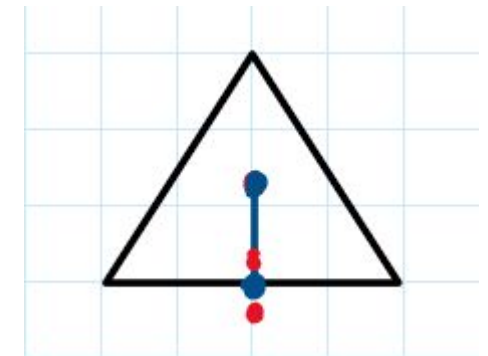
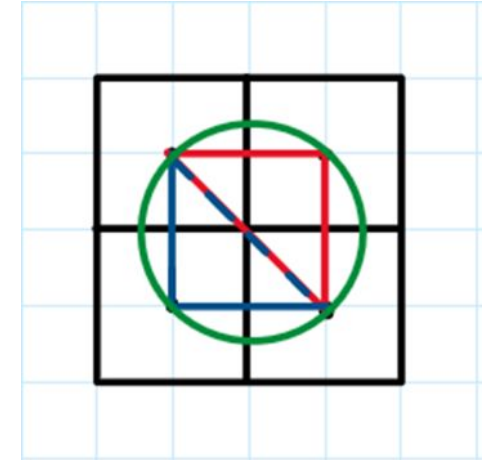


**Live-Demo**

# Herausforderungen & Lösungen

# Herausforderungen & Lösungen

- Voronoi-Heuristik -> Wir erlauben einen 4ten Punkt genau auf dem Umkreis
- Richtung der Voronoi-Kanten in die Unendlichkeit
- Falsche Generierung aufgrund von Float-Vergleichen -> Annäherung
- Lade der Kamera erst nach dem Laden der Szene
- Kollision bei Türen -> verkleinern der Hitbox und Check für die Kantenlänge
- Gebrochene Referenzen -> Editor-Regeln & kleine Schritte



**Mögliche Erweiterung**

# Mögliche Erweiterung

- Item Platzierung verbessern
- Andere Minispiele
- Mehrere Gegner Fähigkeiten
- Dungeon erweitern
- Sound Effekte
- Third-Person-Ansicht
- Multiplayer
- ...

# Fazit & Links



## Fazit & Links

- Viel gelernt
- Spaß gemacht
- Spieleentwicklung aufwendig

**Github:** [https://github.com/haithameleuch/Projektarbeit\\_24\\_25](https://github.com/haithameleuch/Projektarbeit_24_25)

# Vielen Dank für eure Aufmerksamkeit!

## Q&A