Up Go The Living

Entwicklungsprozess

Frederik, Leandro

TU Dortmund

14. Juli 2020

Zielsetzung

Zielsetzung

- Dungeon Crawl
  - Navigation durch Dungeon mit Gegnern, Gegenständen etc.
  - Bewältigung des Dungeons durch Powerups
- Rogue-like
  - prozedural generierte Level und rundenbasierte Spielweise
  - eine Runde: Level erfolgreich/erfolglos beenden (Sieg/ Game Over)

Problemstellung

- Dungeongenerierung: Welches PCG-Verfahren?
  - L-Systems
- Raumlayouts innerhalb von Dungeon (Hindernisse etc.) ?
  - Variabilität
- Größe Itempool/Gegnerpool?
  - Großer Bestandteil des Spiels
- Typen von Items/Gegnern: Welches KI-Verfahren?
  - Steering Behaviors

Schwerpunkte

L-Systems

Anforderungen an die Dungeongenerierung

- Ebenen generieren mit n Räumen
- Genau ein Boss- und ein Loot-Raum
- Alle Räume sind gleich groß und rechteckig
- Benachbarte Räume können, müssen aber nicht verbunden sein
- Kreise sollen möglich sein
- Jeder Raum muss vom Startraum aus erreichbar sein

L-Systems (Lindenmayer Systems)

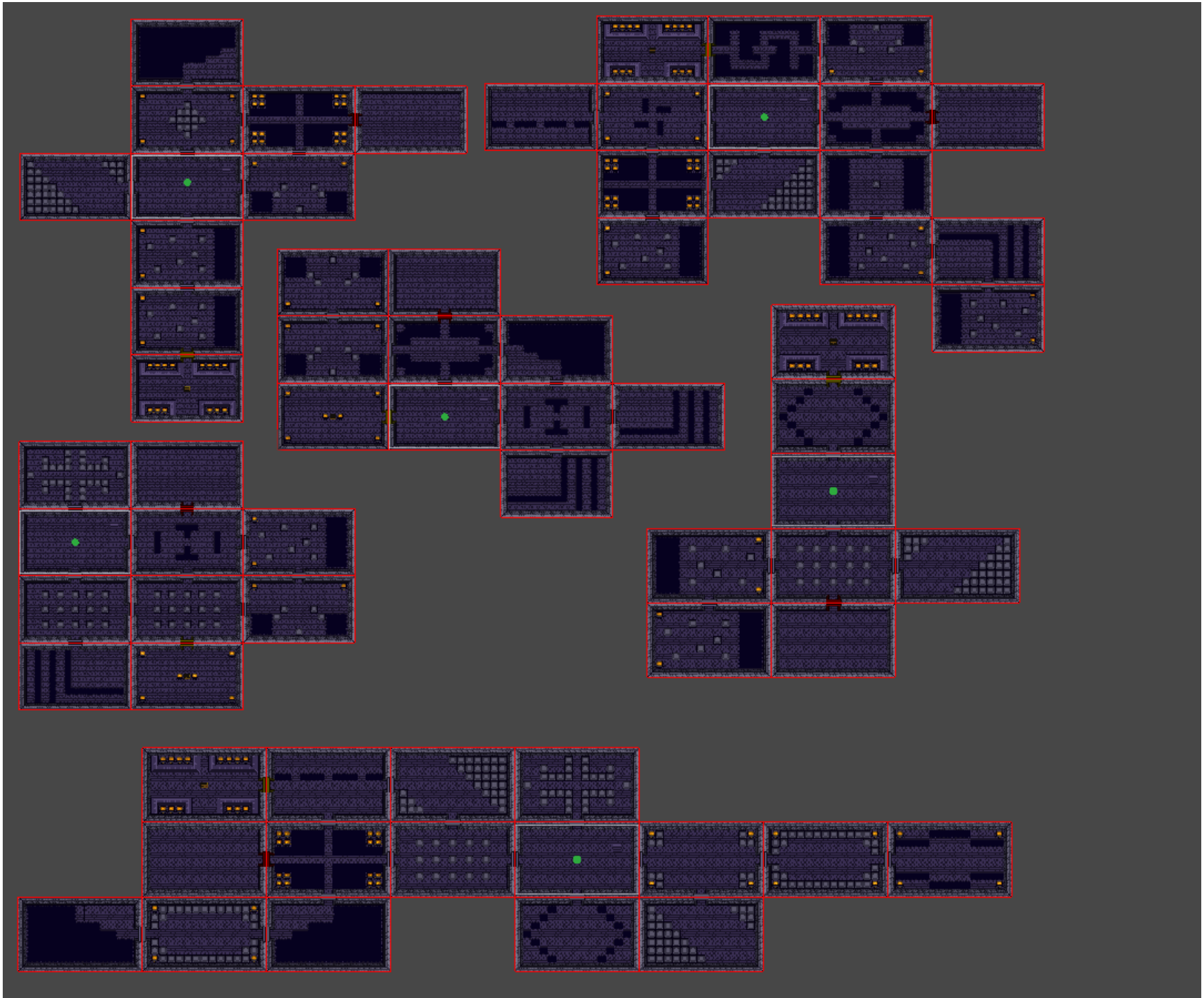
- Raumstrukturen werden über Grammatiken erzeugt
- Alphabet: S, L, B, 1, 2, 3, 4
- Startsymbol: S
- Ersetzungsregeln zweidimensional

Regeln

- Anfügeregeln: z.B.: S: S-3, 2: 2-4 2: 2-L
- Verschiebunsgregeln: z.B.: B: 2-B, L: 3-L
- Sonderzwangsregeln: z.B.: 2: 3-B, 3: 4-L
- Räume können nur erzeugt werden, wenn dort noch kein Raum ist und wenn die zulässige Türenanzahl nicht überschritten wird (1 bei L und B, 4 bei S)
- Genau n-2 nicht L oder B Räume
- Regeln werden inaktiv, wenn Randbedingungen nicht gelten
- Generierte Räume verbinden sich wenn erlaubt mit benachbarten Räumen
- Regeln haben Gewichte um die Ebenen interessanter zu machen

Generierungsprozess

- Wähle zufällig einen bestehenden Raum aus
- Prüfe welche Regeln darauf anwendbar sind
- Wende eine zufällige dieser Regeln an, wiederhole das ganze
- Anderen Raum auswählen wenn keine Regel anwendbar ist
- Abbruch wenn kein Raum anwendbare Regeln besitzt
- Wähle für jeden Raum ein zufälliges Raum-Preset für diesen Raumtyp
- Erstelle die Räume als entsprechende Spielszenen



Steering Behaviors

Anforderungen an Gegner

- Gegner sollen unterschiedliches Verhalten zeigen können (Fliehend, Zielsuchend, Wandernd etc.)
- Verhalten soll änderbar sein (z.B. durch Items)

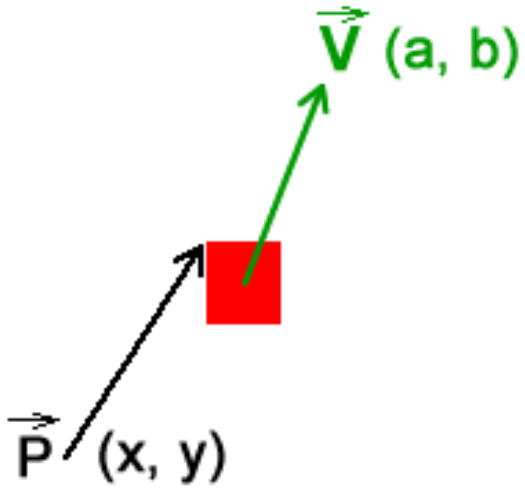
- Steering Behaviors

Steering Behaviors - Grundlagen

- realistische Bewegungsmuster für Gegner
- Konzept basiert auf Vektoren
  - bekanntes Konstrukt aus der Mathematik
  - Verzicht auf (komplexere) globale Berechnungen
- leicht zu verstehen
- trotzdem sind komplexe Bewegungen realisierbar

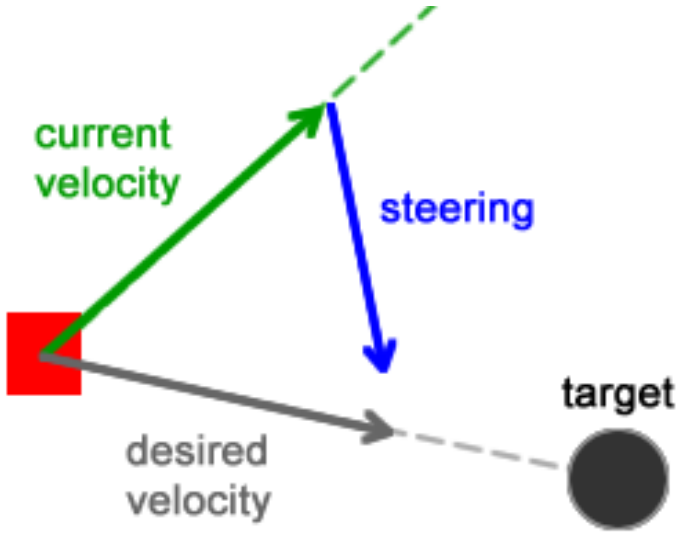
Steering Behaviors

- Positionsvektor $\vec{P}$
- Geschwindigkeitsvektor $\vec{V}$
- Bewegung = Position + Geschwindigkeit



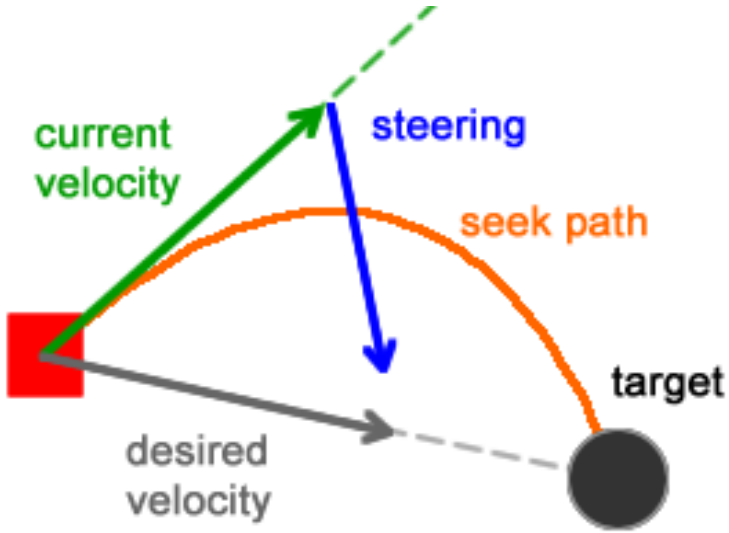
Steering Behaviors (2)

- Hinzunahme von angestrebter Geschwindigkeit
- Steuerungsvektor ist Resultat aus momentaner und angestrebter Geschwindigkeit
- Steuerung = Angestrebte Geschwindigkeit - Momentane Geschwindigkeit

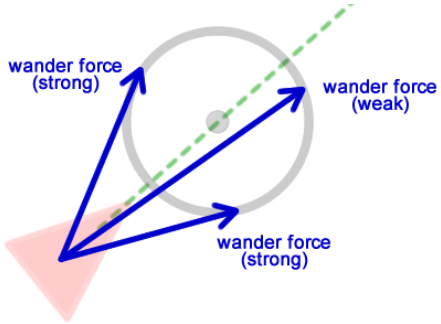


Steering Behaviors (2)

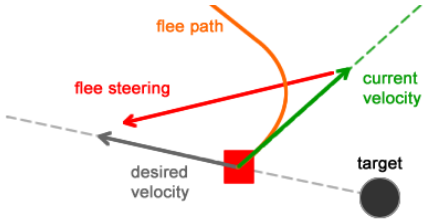
- Hinzunahme von angestrebter Geschwindigkeit
- Steuerungsvektor ist Resultat aus momentaner und angestrebter Geschwindigkeit
- Steuerung = Angestrebte Geschwindigkeit - Momentane Geschwindigkeit



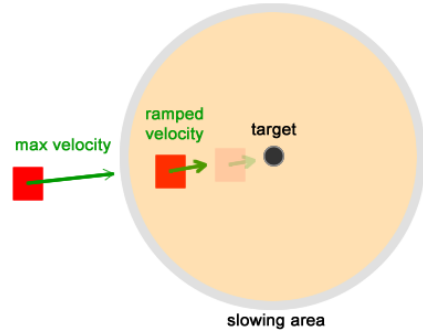
Steering Behaviors (3)



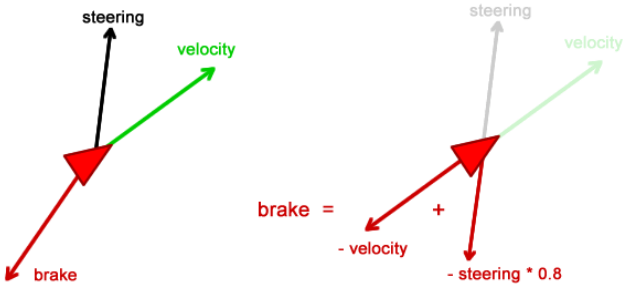
Wander



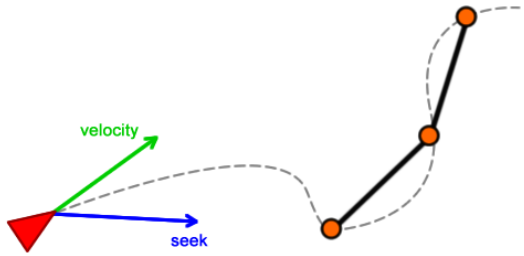
Flee



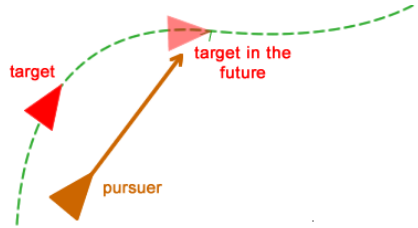
Arrival



Brake

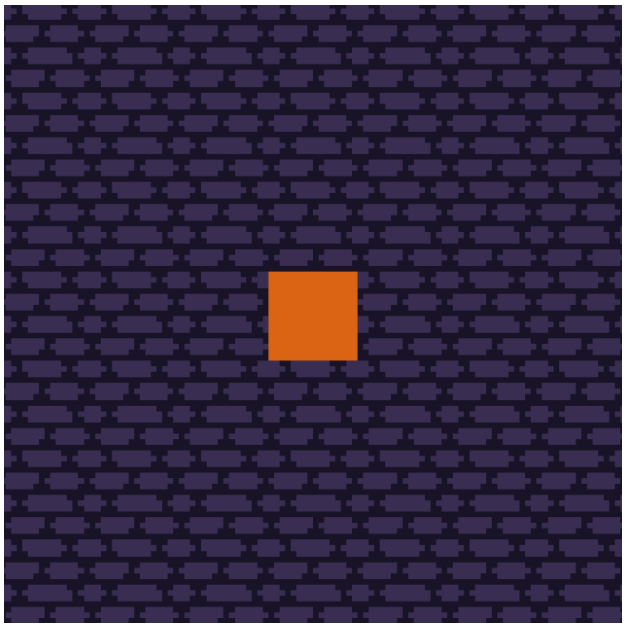


Path Following

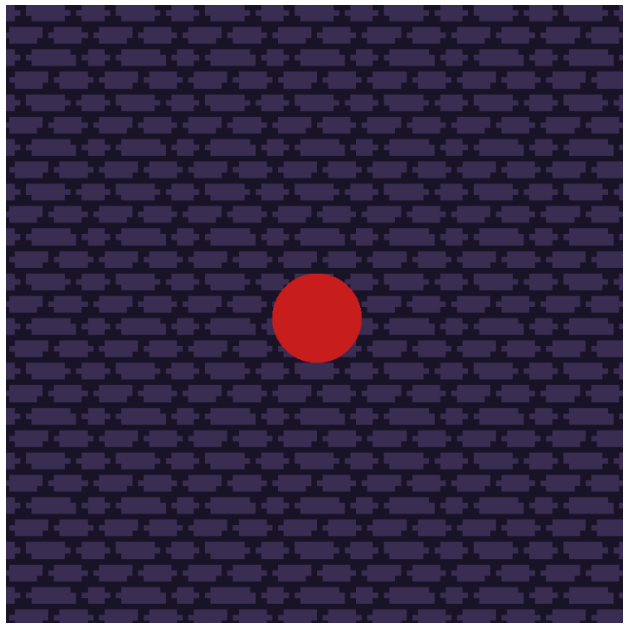


Pursuit

Dürfen wir vorstellen...?



Der Clotty (Wander)



Der Globin (Seek)

Gameplay-Elemente