

Simulation zur Weltraumpiraterie

Entwickler:

Tim Beier

Forschungsfrage

Eine Potenzialeinschätzung unterschiedlicher Regelwerke für Computerspiele anhand einer exemplarischen Simulation für Weltraumpiraterie

Eignen sich adaptive Agenten für Computerspiele? Die rudimentäre Untersuchung eines Fallbeispiels

Untersuchung

Es findet eine Aufwandseinschätzung adaptiver Regelwerke und ihres Nutzen gegenüber statischer Regelwerke für Agenten statt. Bei der Aufwandseinschätzung wird die Qualität des Regelwerks in Kontrast zum Arbeitsaufwand und der benötigten Rechenzeit gesetzt. Es ist auch möglich Spieler gegen die Regelwerke antreten zu lassen(Wofür bleibt fraglich).

Software:

Engine: Unity 3D

Sprache: C#

IDE: JetBrains - Rider

Dokumentation: atom

Welt

Der Weltraum wird als Umfeld gewählt, da es einen sehr freien Raum repräsentiert und vielseitig

interpretiert werden kann.

Piraten werden als Entitäten/Agenten gewählt, da sie ein Konkurrenzverhalten inne halten aber dennoch Kooperationen möglich sind.

Die Piraten starten auf einem Rasterfeld aus quadratischen Feldern und haben eine begrenzte Menge an Aktionen.

Aktionen

- Nichts tun: Notwendig, um falsche Aktion aus Zugzwang zu vermeiden
- Schießen: Abschüsse von Feinden inkludieren das Plündern und gewähren generische Ressourcen.
- Bewegen: Notwendig, sobald keine Feinde mehr in Reichweite sind
- Scannen: Gibt Aufschluss über Feinde und Allierte, sowie freie Felder in Reichweite

Regelwerke

Non-Adaptiv

- 1. Wenn Umgebung unbekannt -> Scannen
Feinden und Alliierte(inklusive sich selbst) in Reichweite und freie Felder sind bekannt
- 2. Es gibt keine Feinde -> Auf freies Feld bewegen oder nicht bewegen. (Zufall?)
- 3. Es gibt mindestens genausoviele Alliierte wie Feinde -> Schießen
- 4. Es gibt mehr Feinde als Alliierte -> Auf freies Feld fliehen

Adaptiv

Übergänge zwischen allen Aktionen mittels Markov und Q-Werten möglich.

Eigene Matritzen für: Verhältnis von Freundanzahl - Feindanzahl -> (<0, 0, >0)

Aktionsmatrix: (-1, 0, +1, Unwissen?)

- Schießen -> Schießenmatrix(angrenzende Felder: V-Werte)
- Scannen -> Errechnet für eine Runde Freund-Feind Verhältnis
 - Gibt eine Runde lang Boni auf Schießenmatrix & Bewegenmatrix abhängig für Felder mit Feind/Freund:
 - Schießenmatrix: Bonus auf Feld von Feind durch Paramerer $v_schießenFeind$ (Startwert 0)
 - Bewegenmatrix: Bonus auf Feld von Feind durch Paramerer $v_bewegenFeind$ (Startwert 0)
 - Schießenmatrix: Bonus auf Feld von Freund durch Paramerer $v_schießenFreund$ (Startwert 0)
 - Bewegenmatrix: Bonus auf Feld von Freund durch Paramerer $v_bewegenFreund$ (Startwert 0)
- Bewegen -> Bewegenmatrix(angrenzende Felder: V-Werte)
- Nichts tun

Offene Fragen

Welche Algorithmen eignen sich dynamisch genug für adaptive Regelwerke?

Konzept in Stichpunkten

Ausgangssituation:

- Eine begrenzte, rechteckige Spielwelt (z.B. 8x6 Felder)
- Die Welt ist mit Piraten verschiedener Fraktionen bevölkert(2-x Fraktionen)
- Rundenbasiert entscheiden sich Piraten für ihre Aktion(feste Reihenfolge)

- Regelmäßig fliegen Transportschiffe durch die Welt

Ziel/Selbstorganisation

- Piraten wollen sich eine maximale Menge an Ressourcen aneignen.
(Ihre Ressourcenaneignung optimieren)

Belohnungsszenario

- Abschuss eines Schiffes -> +Ressourcen

Spielregeln:

- Schiffe lagern Ressourcen (100 max)
- Schiffe verlieren mit der Zeit Ressourcen (1 pro Tick)
- Besitzt ein Schiff keine Ressourcen mehr (0), so wird es zerstört
- Schiffe erhalten Ressourcen beim Zerstören von Schiffen
- Schiffe können schießen und verursachen Schaden in Ressourcen
- Schiffe besitzen eine begrenzte Sichtweite
- Ein Schiff, dass sich auf ein besetztes Feld bewegt erleidet Schaden(25) und bewegt sich nicht

Variablen

- Sichtweite(angrenzend)
- Scanner(angrenzend)
- Schießreichweite(angrenzend)
- Schaden(25)

Optionales

- Schiffe mit 100 Ressourcen spalten sich in 2 Schiffe mit je 25 Ressourcen (Spawn auf freiem, angrenzendem Feld und restliche Ressourcen werden aufgeteilt)
Neue Schiffe kopieren ihre Werte von ihrem Erschaffer (Fraktion, v-werte)
-> Erfordert,dass Belohnungsszenarios des Klons für den Vater gelten, damit es nicht
als Ressourcenverlust zählt sondern als Investition.
- Schildaktion, welche 1-3 Runden Schaden/einen Treffer abfängt
- Schiffe hinterlassen nicht passierbare Wracks
- Das Zerstören von Wracks gibt Ressourcen(50)
- Schiffe verursachen Flächenschaden bei Zerstörung
- Neue Schiffe kosten 50 Ressourcen und Zeitpunkt wird vom Agenten gewählt
- Schiffe können die Stärke des Feindes einschätzen
- Waffensysteme
- Upgrades

- Variable: Waffenschaden
- Variable: Geschwindigkeit
- Variable: Zielgenauigkeit
- Variable: Schätzgenauigkeit

Visualisierung:

Anzahl:

- Agenten zu Spielstart
- aktuelle Agenten
- aktuelle Agenten nach Fraktion
- spezifischer Aktionen
- Ressourcen zu Spielstart
- Ressourcen im Spiel
- Graph plugin?