

PV: Theorieteil

Projektgruppe 17

Aufgabe 4:

Aufbau:

Ziel des Spiels ist es eine KI zu programmieren, die durch ihre Fähigkeit länger als alle anderen Schlangen in der Partie zu bleiben gewinnt. Eine Battlesnake Partie wird erstellt durch Auswahl eines Spielmodus, der Größe der verwendeten rechteckigen Spielfläche und einer Selektion von bis zu acht KI-Schlangen, die sich auf ihrer Gitterstruktur bewegen werden. An dieser Stelle wird sich auf die Beschreibung der Eigenschaften und Regeln des Standardmodus beschränkt.

Zu Beginn der Runde wird jede Schlange in ihrer Gänze auf einer der acht festgelegten Startpositionen platziert. Diese sind so angeordnet, dass die Manhattan-Distanz zwischen zwei beliebigen Positionen immer gerade ist. Durch diese Anforderung wird gewährleistet, dass im Spielverlauf jeder Kopf mit jedem anderen Kopf kollidieren kann. Damit zusätzlich eine gleichmäßige Verteilung möglich ist, muss die Seitenlänge der Spielfläche ungerade (≥ 5) sein. Die Auswahl wurde auf drei Größen beschränkt: 7x7, 11x11, 19x19. Jeder Schlange gehört eine Nahrungsanzeige (Wert von 0 bis 100) an, die zu Beginn aufgefüllt ist. Abschließend werden Nahrungseinheiten ungefähr gleichmäßig im Verhältnis zu der Anzahl der erzeugten Schlangen platziert, womit das Spiel vollständig aufgebaut ist.

Ablauf einer Partie:

Schlangen sind gezwungen sich zu jedem Spielzug in eine von vier Himmelsrichtungen zu bewegen und scheiden aus der Partie unter zwei Bedingungen aus: ihr Nahrungswert fällt auf null oder ihr Kopf kollidiert mit einem Hindernis. Es gibt zwei Hindernistypen: die Umrandung der Spielfläche und Schlangen. Somit kann ein Ausscheiden auch durch Eigenkollision herbeigeführt werden. Bei der Kollision von zwei Kopfsegmenten tritt eine Sonderregel in Kraft: die kürzere von beiden Schlangen scheidet aus (bei gleicher Länge scheiden beide aus). Durch Kollision mit Nahrung wird die Nahrungsanzeige vollständig aufgefüllt und durch jeden Schritt um eins verringert.

Ablauf eines Spielzuges:

Jeder Schlange werden sämtliche Informationen des Spielstatus übertragen (Objektpositionen, Nahrungswerte aller Schlangen, ...). Den Servern wird eine begrenzte Zeit gewährt ihren nächsten Schritt zu berechnen und zu übertragen. Falls dies nicht rechtzeitig geschieht, wird die jeweilige Schlange in die Richtung bewegt, in die sie sich in dem letzten Zug bewegt hat (nach Unten, falls erster Zug). Generell erfolgt der Zugzwang 500 Millisekunden nach Versenden des Spielstatus.

Der Spielstatus wird für alle Schlangen gleichzeitig aktualisiert:

- Schlangenpositionen anpassen:
 - Kopf an neuer Position erzeugen
 - alten Kopf zu normalem Körpersegment ändern
 - Schwanzspitze entfernen (falls sich keine Körpersegmente überlagern)
 - Nahrungswert $\text{--} 1$
- alle Schlangen auf Nahrungsfeldern essen:
 - Nahrungswert auf 100 setzen
 - neues Körpersegment über Schwanzspitze platzieren
 - Nahrung entfernen
- kollidierte / verhungerte (Nahrungsw. $\text{==} 0$) Schlangen entfernen

Sensoren/Aktuatoren:

Battlesnake ist ein Spiel, an dem bis zu acht Agenten gleichzeitig teilnehmen können. Auch wenn Agenten im eigentlichen durch ihre Agentenfunktion definiert sind, lassen sich diese hier im Sinne der Spiels ohne Beschränkung der Allgemeinheit mit ihren Schlangenköpfen gleichsetzen. Jeder Agent verfügt über ein global state eye, was bedeutet, dass sie vollständige Kenntnis über den "Schnappschuss" der aktuellen Spielsituation besitzen. Die Richtung, die sie an das Spiel übertragen, ist ihr einziger Aktuator und somit ihre einzige Möglichkeit die Spielumgebung zu beeinflussen. Da das Genick der Schlange stets eine der vier Richtungen blockiert (außer am Anfang), lässt sich die Entscheidung des Spielzugs auf drei Möglichkeiten reduzieren.

Umgebung:

Die Spielumgebung ist:

- vollständig observabel \iff global state eye
- nicht-deterministisch:
 - Handlung anderer Agenten
 - Spawnfrequenz und -verteilung der Nahrung
- sequentiell:
 - Muster im Verhalten anderer Agenten
 - \longrightarrow Gewichtung berechneter Zustandsräume
- statisch: Agentenaktuatoren werden simultan ausgewertet
- diskret: ca. 500ms Stillstand vor sofortiger Bewegungsaktualisierung
- Multi-Agenten im Wettbewerbssituation

Agenten können sich einen Vorteil verschaffen, indem auf das wahrscheinlichste Verhalten anderer Schlange spekuliert wird. Es lässt sich mit Sicherheit ausschließen, dass sich unter den Schlangen einfache (ernstzunehmende) Reflexagenten befinden, da die Anzahl möglicher Spielzustände zu groß ist. Bei Zustandsbehafteten Reflex-Agenten lassen sich bereits Muster herauslesen. Zum Beispiel laufen Schlangen möglicherweise nie unmittelbar in Hindernisse oder bewegen sich immer zur nächsten Nahrungsquelle. Vielleicht ändert sich auch das Verhalten von Schlangen, wenn eine aggressivere Spielweise eingeschlagen wird. Bereits damit lässt sich der zu berechnende Zustandsbaum verkleinern bzw. gewichten. Somit fließt ein Explorationsproblem in die Nutzenfunktion ein.

Nutzen:

Da das Ziel von Battlesnake generell nicht das Gewinnen von einzelnen Partien ist, sondern das Erklimmen einer Rangliste (bzw. ähnlich in Turnierformaten), entscheidet ein rationaler Agent bei Zurateziehen einer guten Nutzenfunktion nicht, welcher Zug die Gewinnchance maximiert, sondern welcher Zug den Erwartungswert für den Punktestand maximiert. Dies gilt natürlich nicht, falls Ranglistenpunkte nicht gestaffelt nach Reihenfolge des Ausscheidens vergeben oder Turnierformate mit Soforteliminierung ausgetragen werden. Variablen die in die Nutzenfunktionen einfließen können sind zum Beispiel:

- Anzahl der Gegner
- Bewegungsfreiraum
- Nahrungsspawnrate
- Länge
- Länge in Relation zu Anderen
- wahrscheinliche / unwahrscheinliche Gegneraktionen (Zustandswahrscheinlichkeit)
- Ping