BOSCH
Technik fürs Leben

DHBW
Duale Hochschule
Baden-Württemberg

Konzept und Implementierung einer KI für das Rundenstrategiespiel Pummelz

Dokumentation

Bachelor of Science

des Studiengangs Informatik

an der Dualen Hochschule Baden-Württemberg Stuttgart

von

Jona Krumrein, Ruben Hartenstein

19.05.2022

Bearbeitungszeitraum	07.02.-19.05.2022
Matrikelnummer, Kurs	8366074, 2746235, TINF19ITA
Ausbildungsfirma	Robert Bosch GmbH, Stuttgart
Betreuer	Prof. Dr. Falko Kötter

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

1 Einleitung

Es ist ein in Unity und C# implementiertes Rundenstrategiespiel („Pummelz“) gegeben, in dem ähnlich wie beim Schach zwei Spieler mit blauen und roten Figuren gegeneinander antreten. Diese Spielfiguren haben dabei, jedoch unterschiedliche Lebens- und Angriffswerte und können sich anhand ihrer gegebenen Reichweite beliebig bewegen. Zusätzlich besitzen einige Einheiten Spezialfähigkeiten, welche durch unterschiedliche Ereignisse ausgelöst werden können.

Ziel des Programmentwurfs ist es, ein KI zu entwickeln, welche eine bereits vorhandene Greedy-KI in jeglichen Spielszenarien schlägt und eine sehr hohe Spielstärke erreicht. Zudem soll die KI in der Lage sein, auch mit unbekannten Spielfiguren und Szenarien zu spielen. Hierzu soll eine passender Entwurf erstellt und umgesetzt werden. Im nächsten Schritt sollen die konzeptionierten Algorithmen und Funktionen überarbeitet und mittels mehrerer Iterationsschritten verbessert werden. Zuletzt soll eine Evaluierung der implementierten KI sowie deren Spielstärke durchgeführt werden.

2 Entwurf

Zu Beginn des Entwurfs wurden sich die folgenden strategischen Grundlagen überlegt. Diese dienen als Grundlage der KI und ihr Handeln in bestimmten Situationen.

- Wenn ein Czaremir (König) im Spiel ist muss sein Überleben jede Runde garantiert werden, da sein Tod das Spiel beendet. Im Gegenzug sollte der gegnerische priorisiert angegriffen werden, um einen schnellen Sieg erzwingen zu können.
- Viel hilft viel: in jedem Zug muss mit jeder Figur angegriffen werden die kann. **Außnahme** sind Angriffe auf:
  Bummz wenn er einem selbst mehr Schaden als dem Gegner zufügt
  Chilly wenn er nicht „Oneshot“ ist
- Schaden den der Gegner austeilen kann minimieren:
  - Schaden auf einen Gegner zu konzentrieren lohnt sich mehr als Schaden auf mehrere Gegner zu verteilen, da so Schaden in der nächsten Runde vermieden werden kann.
  - Gegner müssen anhand ihrer Eigenschaften klassifiziert werden. Gegner die mehr Schaden austeilen sind früher zu töten, da auch hier Schaden in der nächsten Runde vermieden werden kann
  - Die KI soll unbedingt die Reichweite der eigenen Einheiten ausnutzen. So soll die Anzahl der gegnerischen Einheiten, die eigene Einheit mit großer Reichweite angreifen kann, gering gehalten werden.
  - Bummz sich in die Richtung der Gegner und weg von den eigenen Einheiten bewegen.
- Einheiten mit geringer Reichweite und hohen Lebenspunkten als „Tanks“ einsetzen, um andere Einheiten zu schützen.
- Einheiten passend gruppieren, sodass Einheiten mit Flächeneffekten (Buffy, Haley) möglichst viele Einheiten um sich stehen haben.

- **ABER:** Aufpassen, dass möglichst viele eigene Einheiten angreifen können, um keine Blockade zu erzeugen und Schaden zu „verlieren“

Als grundlegendes Konzept soll ein DefaultDecisionTree konzipiert werden, welcher anhand der Spielsituation eine Entscheidung über den Spielzug der jeweiligen Spielfigur trifft. Hierbei wird eine Unterscheidung zwischen einem Angriff und einer Bewegung getroffen. Der Angriffentscheidung liegen drei Abfragen zu Grunde: Kann eine gegnerische Einheit „geoneshotted“, getötet oder angegriffen werden. Diese werden in der angegeben Reiehenfolge durchlaufen. Beim Zutreffen des Ereignisses soll dann ein Angriff durchgeführt werden. Die Entscheidung bei mehreren möglichen Einheiten erfolgt über eine festgelegte Hierarchie.

Ein Bewegungszug hat die folgenden beiden Möglichkeiten: Wurden zuvor mehrere Züge ohne einen Angriff vollzogen, so läuft die Einheit aggressiv auf die nächste gegnerische Einheit zu. Ist dies nicht der Fall soll eine Standard-Bewegung ausgeführt werden, bei der die Einheit im besten Fall eine andere angreifen und von keiner gegenerischen Einheit angegriffen werden kann.

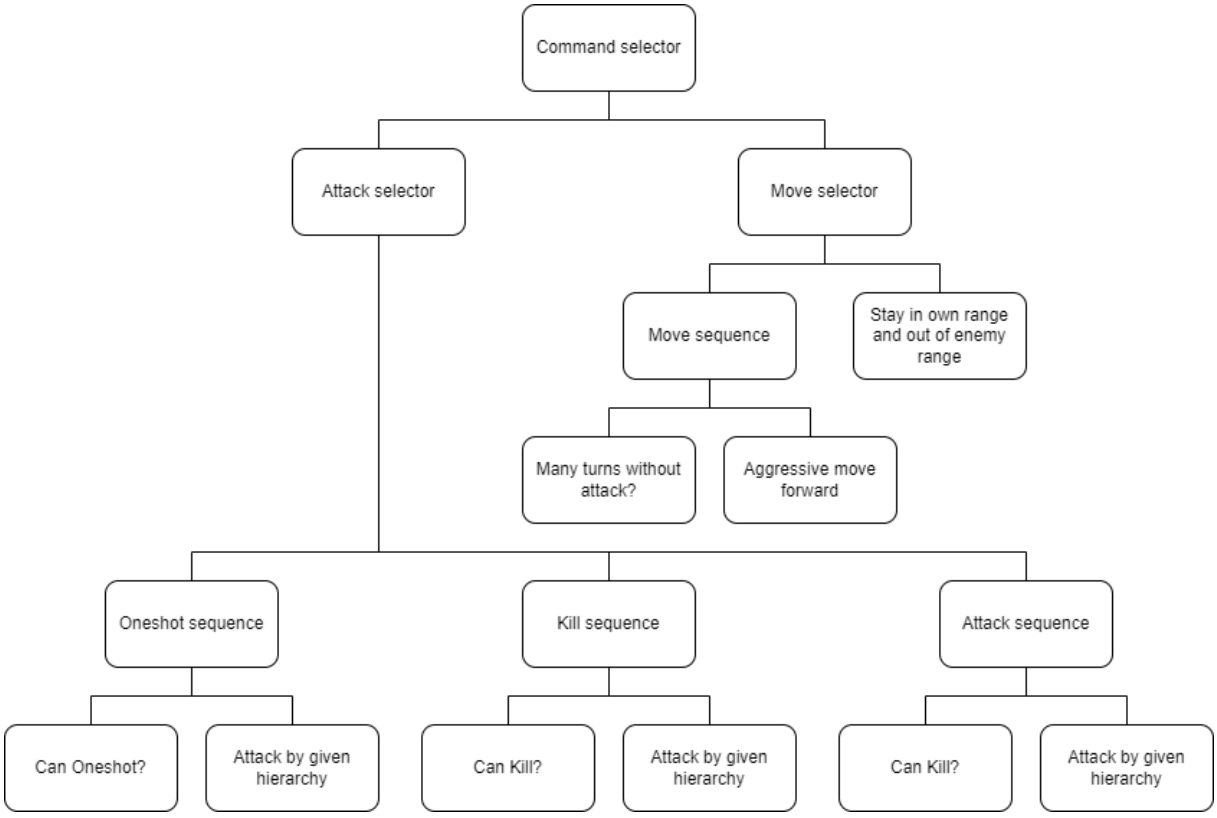

Abbildung 2.1: Konzept eines DecisionTrees zur Pummelz-Command-Entscheidung

Aufbauend auf diesem DefaultDecisionTree sollen weitere DecisionTrees erstellt werden, die auf die Eigenschaften und Spezialfähigkeiten der einzelnen Einheiten zugeschnitten sind und beim Zug der entsprechenden Einheit anstatt des DefaultTrees aufgerufen werden. So ist beispielsweise das Ziel von Bummz von den eigenen Einheiten weg in die gegnerischen zu Rennen. Sneip, Killy und Ängli hingegen sollen sich möglichst weit von gegnerischen Einheiten platzieren, um so wenig möglich Gefahr ausgesetzt zu sein.

3 Implementierung und Iterationsschritte

4 Evaluation

Anhang