Zweiter Milestone: KI-Wettbewerb

Patrick Schäfer

patrick.schaefer@hu-berlin.de

Agenda

- Heute
  - Kickoff Zweiter Milestone
  - Vortrag durch Dennis Wagner: Reinforcement Learning

- Montag, 14.01., **13:15 Uhr:**
  - Erstes Turnier

Anforderungskatalog

LOG

Playback

JSON

Server
(2-5 Spieler)

1. Meilenstein
Observer-GUI

2. Meilenstein
KI-Spieler /
Wettbewerb

Erinnerung: Roadmap

- End of First Milestone:
  - Team with most votes: Set up binary for AI challenge
  - Finalize first milestone: every team has to submit a binary for playbacks of playthrough logs
  - Familiarize with API for AI development

Ziele im 2ten Milestone

- Wir stellen eine Benchmark-KI zur Verfügung

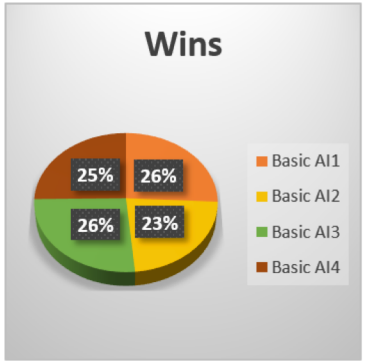
- Ziele im 2ten Milestone:
  1. Eine erste lauffähige KI wurde bis zum 18.1. abgegeben
  2. Eure KI erreicht 60 Punkte im Durchschnitt in 10 Spielen gegen unsere KI
  3. Eure KI erreicht 80 Punkte im Durchschnitt in 10 Spielen gegen unsere KI
  4. Eure KI erreicht 100 Punkte im Durchschnitt in 10 Spielen gegen unsere KI oder alternativ gewinnt in 7 von 10 Spielen

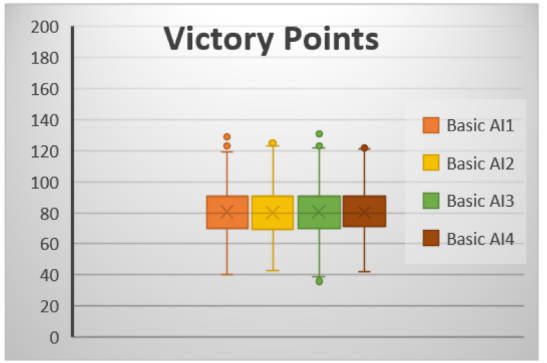
- Ein Akzeptanzkriterium gilt nur dann als erfüllt, wenn
  - eure KI nicht länger als 5 Minuten für ein Spiel gegen unsere KI benötigt

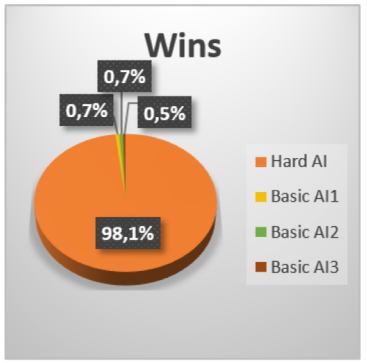
- Aktive Teilnahme ist Scheinkriterium. Daher neue Chance, sich einzubringen.

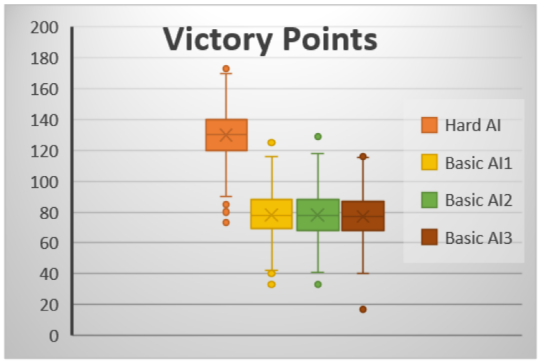
Wiederholung: Batch-Modus

- Batch-Modus:
  - Neue KIs wiederholt (zB 10 Mal) gegen alte KIs antreten lassen
  - Messen der Siegpunkte
- Workflow: Spiel gegen KI
  - Neue Ideen (KIs) implementieren
  - Test im Batchmodus
  - Änderungen ggf. übernehmen









https://connect.unity.com/p/articles-boardgame-ai

Batch-Modus

- Wir stellen euch eine Benchmark-KI und ein Script für den Batch-Modus zur Verfügung
- Das Skript simuliert mehrere Durchläufe und bestimmt den Sieger
- Ihr könnt eure KI einbinden, indem ihr ein “run_ki.sh“ Skript schreibt
- Aufruf:

  ./run_test.py 10 ./run_ki.sh ./run_ki2.sh

Batch-Modus – Live Demo

Batch-Modus

Running KI Tests
Repetition: | | 100.0%
Completed games: 10, Failed games: 0

==========(Black(Java_KI_B))==========
AVG Longest Route: 23.9
AVG Score: 84.8
Winrate: 60.0%.
AVG Claimed Tickets: 2.6
AVG Not Claimed Tickets: 0.4

==========(Blue(RouteFocusedAgent))==========
AVG Longest Route: 23.3
AVG Score: 91.8
Winrate: 40.0%.
AVG Claimed Tickets: 2.1
AVG Not Claimed Tickets: 0.0

The overall winner is Black(Java_KI_B) .

KI-Challenge: Ablauf

- Wöchentliche Gruppen- und Einzelabgabe-Turniere, Zeitlimit 5 Minuten pro Spiel

a) Gruppen-Turnier im 1-on-1 K.O.-System:

- Jeweils eine KI jedes Teams spielt gegen eine zufällige KI eines anderen Teams
- 4 Paarungen: 2 Halbfinale, Finale und Spiel um Platz 3
- Der Gewinner wird im Batch-Modus mit 10 Wiederholungen bestimmt
- Clients müssen vorab abgegeben werden
- Zunehmende Punkte für erfolgreiche Teilnahme und Platzierung
  - 14.1.: 1-1-2, 21.1.: 1-2-3, 28.1.: 2-3-5, 4.2.: 3-5-8, 11.2.: 5-8-13
- 0 Punkte, wenn KI wiederholt abstürzt oder Spiel nicht terminiert
- Wir visualisieren montags jeweils ein exemplarisches Spiel pro Partie

b) Free-For-All: Live-Turnier

- Einzelne Mitglieder eines Teams, Scrum-Master, Kursleiter usw.
- Anzahl Paarungen abhängig von Anzahl Abgaben
- Keine Abgabe
- Ihr meldet euch live mit (eurem Notebook und) eurer KI am Spiele-Server an

KI-Challenge: Abgabe

- Abgabe für Gruppen-Turnier jeweils bis Freitag Mitternacht:
  - Abgabe erfolgt über HU-BOX: https://hu.berlin/ttr_ai
  - Ladet ein ZIP für euer Team (z.B. team_blau.zip) hoch mit
    - eurer KI,
    - einem Startscript und
    - einer Anleitung, wie man den Client baut
  - Die Abgabe wird bzgl. der Erfüllung der Akzeptanzkriterien getestet und nimmt am Wettbewerb teil
- Keine Abgabe für Einzelturniere

Server v0.94

- Diffs:
  - Spieler können beim Join nun auch eine Farbe übergeben
    `Join KI Player Black`
  - Bugfixes
- Startparameter:
  - java -jar ttr-server-0.94.jar {Players} {ExportLogFile} {Seed}

Mögliche Strategien

- Destination Hungry Agent (DHA)
  - Ziehe maximale Anzahl überlappender Zielkarten bis Anzahl Train-Tokens erreicht wurde
  - Berechne Steiner Tree
  - Baue Strecken

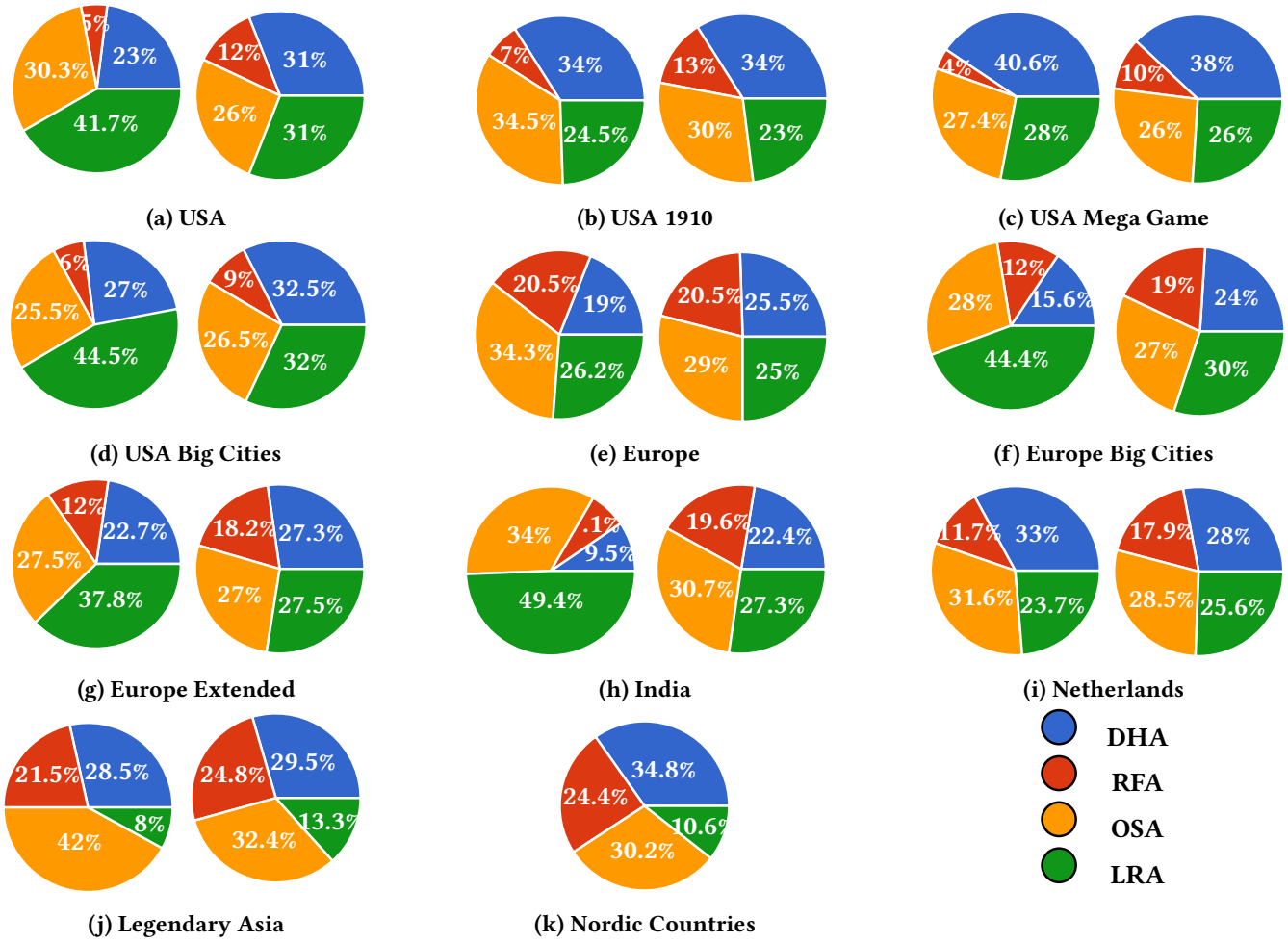
- Route Focused Agent (RFA)
  - Behalte alle 3 initialen Zielkarten
  - Erfülle alle Zielkarten
  - Baue anschließend lange Strecken

Mögliche Strategien

- One Step Agent (OSA)
  - Ziehe möglichst überlappende Zielkarten
  - Sobald Zielkarten erfüllt sind, ziehe neue Zielkarten

- Long Route Agent (LRA)
  - Behalte alle initialen Zielkarten
  - Aber: Baue nur Strecken der Länge >= 3
  - Sobald alle Zielkarten erfüllt wurden: baue nur noch lange Strecken

Welche Strategien funktionieren gut?



Destination Hungry Agent (DHA)
First keep drawing tickets,
then build Steiner Tree

Route Focused Agent (RFA)
Fulfill all initial tickets,
then build long routes

One Step Agent (OSA)
First fulfill held tickets,
then draw new tickets

Long Route Agent (LRA)
First fulfill held tickets,
but Focus on building long
(>=3) routes

http://julian.togelius.com/Silva2017AI.pdf

Weiterführende Links

- Vorlesung „General Game Playing“ von Sebastian Wandelt im WS 13/14: https://hu.berlin/vl_game
- Ticket To Ride Strategies: http://julian.togelius.com/Silva2017AI.pdf
- Ticket to Ride in Java mit implementierter (aber unkommentierter) Minimax- und Monte-Carlo-Baumsuche https://github.com/cooijmanstim/hobo
- Reddit-Thread mit allerlei nützlichen Informationen und Erfahrungswerten: https://www.reddit.com/r/gameai/comments/44l6w7/building_an_ai_for_a_multiplayer_turnbased_board/