

Simulator für wiederholte Spiele

Michel Bodé, Max Braun, Sophie Bräuniger, Simon Hügel, Zezhong Tong

1. Einführung
 - 1.1 Grundlagen
 - 1.2 Motivation
2. Planungsfragen
3. Umsetzung
4. Demonstration

Worum geht es?

- Spieltheorie
- Wiederholte Spiele → Stufenspiele
- Suche nach Gleichgewichten

Worum geht es?

- Spieltheorie
 - Wiederholte Spiele → Stufenspiele
 - Suche nach Gleichgewichten

Worum geht es?

- Spieltheorie
- Wiederholte Spiele → Stufenspiele
- Suche nach Gleichgewichten

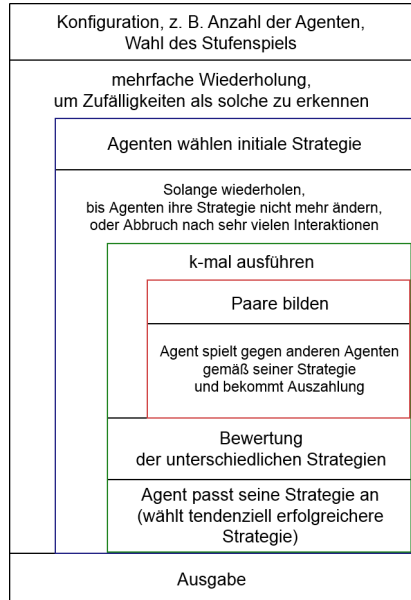
Worum geht es?

- Spieltheorie
- Wiederholte Spiele \rightarrow Stufenspiele
- Suche nach Gleichgewichten

Wofür?

Ausgangspunkt:

Wiederholung
Anpassungszyklus
Runde



Welche zentralen Planungsfragen ergeben sich daraus?

- Können sinnvolle Vereinfachungen vorgenommen werden?
- Welches Verhalten soll durch die Strategien abbildbar sein?
- Wie wird Erfolg gemessen?
- usw.

Außerdem:

- Wie arbeiten wir?

Welche zentralen Planungsfragen ergeben sich daraus?

- Können sinnvolle Vereinfachungen vorgenommen werden?
- Welches Verhalten soll durch die Strategien abbildbar sein?
- Wie wird Erfolg gemessen?
- usw.

Außerdem:

- Wie arbeiten wir?

Welche zentralen Planungsfragen ergeben sich daraus?

- Können sinnvolle Vereinfachungen vorgenommen werden?
- Welches Verhalten soll durch die Strategien abbildbar sein?
- Wie wird Erfolg gemessen?
- usw.

Außerdem:

- Wie arbeiten wir?

Welche zentralen Planungsfragen ergeben sich daraus?

- Können sinnvolle Vereinfachungen vorgenommen werden?
- Welches Verhalten soll durch die Strategien abbildbar sein?
- Wie wird Erfolg gemessen?
- usw.

Außerdem:

- Wie arbeiten wir?

Welche zentralen Planungsfragen ergeben sich daraus?

- Können sinnvolle Vereinfachungen vorgenommen werden?
- Welches Verhalten soll durch die Strategien abbildbar sein?
- Wie wird Erfolg gemessen?
- usw.

Außerdem:

- Wie arbeiten wir?

Welche zentralen Planungsfragen ergeben sich daraus?

- Können sinnvolle Vereinfachungen vorgenommen werden?
- Welches Verhalten soll durch die Strategien abbildbar sein?
- Wie wird Erfolg gemessen?
- usw.

Außerdem:

- Wie arbeiten wir?

Können sinnvolle Vereinfachungen vorgenommen werden?

↔ z.B. Art der erlaubten Spiele

Können sinnvolle Vereinfachungen vorgenommen werden?

↪ z.B. Art der erlaubten Spiele

Spiele in Normalform:

		Spieler 2	
		<i>K</i>	<i>D</i>
Spieler 1	<i>K</i>	$-1, -1$	$-3, 0$
	<i>D</i>	$0, -3$	$-2, -2$

Spiele in Normalform:

		Spieler 2	
		<i>K</i>	<i>D</i>
Spieler 1	<i>K</i>	$-1, -1$	$-3, 0$
	<i>D</i>	$0, -3$	$-2, -2$

Spiele erstellen:

Name:

Spieler1 / Spieler2	Kooperation		Defektion
Kooperation	<input type="text" value="-3"/> <input type="button" value="↑"/> <input type="button" value="↓"/>	/	<input type="text" value="-2"/> <input type="button" value="↑"/> <input type="button" value="↓"/> <input type="text" value="-1"/> <input type="button" value="↑"/> <input type="button" value="↓"/> / <input type="text" value="0"/> <input type="button" value="↑"/> <input type="button" value="↓"/>
Defektion	<input type="text" value="1"/> <input type="button" value="↑"/> <input type="button" value="↓"/> / <input type="text" value="2"/> <input type="button" value="↑"/> <input type="button" value="↓"/>		<input type="text" value="3"/> <input type="button" value="↑"/> <input type="button" value="↓"/> / <input type="text" value="4"/> <input type="button" value="↑"/> <input type="button" value="↓"/>

Beschreibung:

Verwaltung:

Falke-Taube
Feiglingsspiel
Gefangenendilemma
Hirschjagd
Kampf der Geschlechter
Treppenspiel
Vertrauensspiel

Falke-Taube

Spieler1 / Spieler2	Kooperation	Defektion
Kooperation	3/3	0/10
Defektion	10/0	-5/-5

Beschreibung:

Spiel bearbeiten

Spiel löschen

neues Spiel

Welches Verhalten soll durch die Strategien abbildbar sein?

↔ Spielverhalten und Anpassungsverhalten

Welches Verhalten soll durch die Strategien abbildbar sein?

↔ Spielverhalten und Anpassungsverhalten

Kombinierte Strategien:

Name:	<input type="text" value="Gruppenloyal"/>		
Priorität	Bedingungen		Strategie
1.	<input type="text" value="Eigene Gruppe"/> ▼	<input type="button" value="Parameter setzen"/>	<input type="text" value="Immer kooperieren"/> ▼
2.	<input type="text" value="Reicher als ich"/> ▼	<input type="button" value="Parameter setzen"/>	<input type="text" value="Nie kooperieren"/> ▼
<input type="button" value="Bedingung hinzufügen"/>		<input type="button" value="letzte Bedingung entfernen"/>	
Keine Bedingung:	<input type="text" value="Tit-for-Tat Gruppe"/> ▼		
Beschreibung:	<input type="text"/>		

Gemischte Strategien:

Name:		<input type="text" value="Gut oder boese"/>
Wahrscheinlichkeit		Kombinierte Strategie
<input type="text" value="0.5"/>		<input type="text" value="Gut"/> ▼
<input type="text" value="0.5"/>		<input type="text" value="Boese"/> ▼
<input type="button" value="Kombinierte Strategie hinzufügen"/>		<input type="button" value="letzte Kombinierte Strategie entfernen"/>
Beschreibung:		
<input type="text"/>		

Die Strategien auf einen Blick:

Basisstrategien \subset kombinierte Strategien \subset gemischte Strategien

+ Bedingungen

+ Zufallskomponente

Anpassung der Strategien durch Anpassungsalgorithmen:

Replicator Dynamic Rank	▼
Mixed Linear Interpolation	
Mixed Sum	
Rank Percentage	
Replicator Dynamic Rank	
Replicator Dynamic Score	

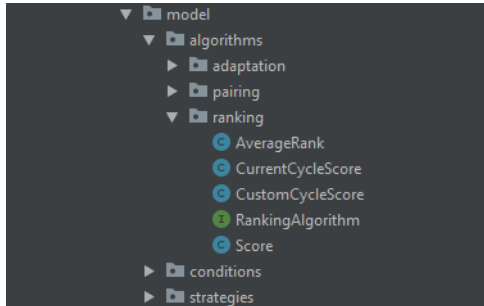
Wie wird Erfolg gemessen?

↔ Erfolg eines Spielers vs. Erfolg einer Strategie

Wie wird Erfolg gemessen?

↔ Erfolg eines Spielers vs. Erfolg einer Strategie

Erfolg eines Spielers durch Bewertungsalgorithmen bemessen:



Wann ist eine Strategie erfolgreich?

↔ Wenn sie selten angepasst wird!

Wann ist eine Strategie erfolgreich?

↪ Wenn sie selten angepasst wird!

Arbeitsweise:

- Dezentral mit wöchentlichen Treffen
- Programmierung in Java
- Arbeitsteilung (M-V-C)
- Versionskontrolle (Git, GitHub)
- Testgetriebene Entwicklung (continuous integration)
- Code Reviews (Pull Request)

Arbeitsweise:

- Dezentral mit wöchentlichen Treffen
- Programmierung in Java
- Arbeitsteilung (M-V-C)
- Versionskontrolle (Git, GitHub)
- Testgetriebene Entwicklung (continuous integration)
- Code Reviews (Pull Request)

Arbeitsweise:

- Dezentral mit wöchentlichen Treffen
- Programmierung in Java
- Arbeitsteilung (M-V-C)
- Versionskontrolle (Git, GitHub)
- Testgetriebene Entwicklung (continuous integration)
- Code Reviews (Pull Request)

Arbeitsweise:

- Dezentral mit wöchentlichen Treffen
- Programmierung in Java
- Arbeitsteilung (M-V-C)
- Versionskontrolle (Git, GitHub)
- Testgetriebene Entwicklung (continuous integration)
- Code Reviews (Pull Request)

Arbeitsweise:

- Dezentral mit wöchentlichen Treffen
- Programmierung in Java
- Arbeitsteilung (M-V-C)
- Versionskontrolle (Git, GitHub)
- Testgetriebene Entwicklung (continuous integration)
- Code Reviews (Pull Request)

Arbeitsweise:

- Dezentral mit wöchentlichen Treffen
- Programmierung in Java
- Arbeitsteilung (M-V-C)
- Versionskontrolle (Git, GitHub)
- Testgetriebene Entwicklung (continuous integration)
- Code Reviews (Pull Request)

Arbeitsweise:

- Dezentral mit wöchentlichen Treffen
- Programmierung in Java
- Arbeitsteilung (M-V-C)
- Versionskontrolle (Git, GitHub)
- Testgetriebene Entwicklung (continuous integration)
- Code Reviews (Pull Request)

Die Entwicklung in Zahlen:

- 1.075 Commits
- 153 geschlossene Pull requests
- 22 geschlossene Issues
- 80.382 hinzugefügte Zeilen (via GitHub)
- 38.229 gelöschte Zeilen (via GitHub)
 Δ 42.153 Zeilen
- 92% Testabdeckung im Model
- 90+% Testabdeckung im Controller

Die Entwicklung in Zahlen:

- 1.075 Commits
- 153 geschlossene Pull requests
- 22 geschlossene Issues
- 80.382 hinzugefügte Zeilen (via GitHub)
- 38.229 gelöschte Zeilen (via GitHub)
 Δ 42.153 Zeilen
- 92% Testabdeckung im Model
- 90+% Testabdeckung im Controller

Die Entwicklung in Zahlen:

- 1.075 Commits
- 153 geschlossene Pull requests
- 22 geschlossene Issues
- 80.382 hinzugefügte Zeilen (via GitHub)
- 38.229 gelöschte Zeilen (via GitHub)
 Δ 42.153 Zeilen
- 92% Testabdeckung im Model
- 90+% Testabdeckung im Controller

Die Entwicklung in Zahlen:

- 1.075 Commits
- 153 geschlossene Pull requests
- 22 geschlossene Issues
- 80.382 hinzugefügte Zeilen (via GitHub)
- 38.229 gelöschte Zeilen (via GitHub)
 Δ 42.153 Zeilen
- 92% Testabdeckung im Model
- 90+% Testabdeckung im Controller

Die Entwicklung in Zahlen:

- 1.075 Commits
- 153 geschlossene Pull requests
- 22 geschlossene Issues
- 80.382 hinzugefügte Zeilen (via GitHub)
- 38.229 gelöschte Zeilen (via GitHub)
 Δ 42.153 Zeilen
- 92% Testabdeckung im Model
- 90+% Testabdeckung im Controller

Die Entwicklung in Zahlen:

- 1.075 Commits
- 153 geschlossene Pull requests
- 22 geschlossene Issues
- 80.382 hinzugefügte Zeilen (via GitHub)
- 38.229 gelöschte Zeilen (via GitHub)
 - △ 42.153 Zeilen
- 92% Testabdeckung im Model
- 90+% Testabdeckung im Controller

Die Entwicklung in Zahlen:

- 1.075 Commits
- 153 geschlossene Pull requests
- 22 geschlossene Issues
- 80.382 hinzugefügte Zeilen (via GitHub)
- 38.229 gelöschte Zeilen (via GitHub)
 - Δ 42.153 Zeilen
- 92% Testabdeckung im Model
- 90+% Testabdeckung im Controller

Die Entwicklung in Zahlen:

- 1.075 Commits
- 153 geschlossene Pull requests
- 22 geschlossene Issues
- 80.382 hinzugefügte Zeilen (via GitHub)
- 38.229 gelöschte Zeilen (via GitHub)
 Δ 42.153 Zeilen
- 92% Testabdeckung im Model
- 90+% Testabdeckung im Controller

Die Entwicklung in Zahlen:

- 1.075 Commits
- 153 geschlossene Pull requests
- 22 geschlossene Issues
- 80.382 hinzugefügte Zeilen (via GitHub)
- 38.229 gelöschte Zeilen (via GitHub)
 Δ 42.153 Zeilen
- 92% Testabdeckung im Model
- 90+% Testabdeckung im Controller

in progress...

Vielen Dank für Ihre Aufmerksamkeit!