

Groker

Erzeugt von Doxygen 1.8.10

Mit Okt 14 2015 18:15:17



# Inhaltsverzeichnis

<b>1</b>	<b>Hauptseite</b>	<b>1</b>
<b>2</b>	<b>Verzeichnis der Namensbereiche</b>	<b>3</b>
2.1	Liste aller Namensbereiche . . . . .	3
<b>3</b>	<b>Hierarchie-Verzeichnis</b>	<b>5</b>
3.1	Klassenhierarchie . . . . .	5
<b>4</b>	<b>Klassen-Verzeichnis</b>	<b>7</b>
4.1	Auflistung der Klassen . . . . .	7
<b>5</b>	<b>Datei-Verzeichnis</b>	<b>9</b>
5.1	Auflistung der Dateien . . . . .	9
<b>6</b>	<b>Dokumentation der Namensbereiche</b>	<b>11</b>
6.1	ai-Namensbereichsreferenz . . . . .	11
6.2	game_wrapper-Namensbereichsreferenz . . . . .	11
6.3	Paket org . . . . .	11
6.4	Paket org.pixelgaffer . . . . .	11
6.5	Paket org.pixelgaffer.turnierserver . . . . .	11
6.6	Paket org.pixelgaffer.turnierserver.groker . . . . .	11
6.7	Paket org.pixelgaffer.turnierserver.groker.ai . . . . .	12
6.8	Paket org.pixelgaffer.turnierserver.groker.logic . . . . .	12
<b>7</b>	<b>Klassen-Dokumentation</b>	<b>13</b>
7.1	_player Strukturreferenz . . . . .	13
7.1.1	Ausführliche Beschreibung . . . . .	13
7.1.2	Dokumentation der Datenelemente . . . . .	13
7.1.2.1	gewonneneChips . . . . .	13
7.1.2.2	letzterEinsatz . . . . .	13
7.2	_result Strukturreferenz . . . . .	13
7.2.1	Ausführliche Beschreibung . . . . .	14
7.2.2	Dokumentation der Datenelemente . . . . .	14
7.2.2.1	chips . . . . .	14

7.2.2.2	output	14
7.3	ai.Ai Klassenreferenz	14
7.3.1	Dokumentation der Elementfunktionen	14
7.3.1.1	einsatz(self)	14
7.3.1.2	process(self, ownWonChips, enemyWonChips, ownChips, enemyChips)	14
7.4	org.pixelgaffer.turnierserver.groker.ai.AiDaten Klassenreferenz	14
7.4.1	Dokumentation der Datenelemente	15
7.4.1.1	gewonneneChips	15
7.4.1.2	letzterEinsatz	15
7.5	game_wrapper.GameWrapper Klassenreferenz	15
7.5.1	Dokumentation der Elementfunktionen	15
7.5.1.1	add_output(self, d, o)	15
7.5.1.2	process(self, d)	15
7.5.1.3	update(self, updates)	15
7.6	org.pixelgaffer.turnierserver.groker.ai.GrokerAi Klassenreferenz	15
7.6.1	Ausführliche Beschreibung	16
7.6.2	Beschreibung der Konstruktoren und Destruktoren	16
7.6.2.1	GrokerAi(String[] args)	16
7.6.3	Dokumentation der Elementfunktionen	16
7.6.3.1	einsatz(AiDaten du, AiDaten gegner)	16
7.6.3.2	update(String answer)	17
7.7	GrokerAi Klassenreferenz	17
7.7.1	Ausführliche Beschreibung	17
7.7.2	Beschreibung der Konstruktoren und Destruktoren	18
7.7.2.1	GrokerAi(Wrapper *wrapper)	18
7.7.3	Dokumentation der Elementfunktionen	18
7.7.3.1	calc(Player *me, Player *enemy)=0	18
7.7.3.2	crash(const char *reason)	18
7.7.3.3	out()	18
7.7.3.4	readOutput()	18
7.7.3.5	surrender()	18
7.8	org.pixelgaffer.turnierserver.groker.logic.GrokerAiObject Klassenreferenz	18
7.9	org.pixelgaffer.turnierserver.groker.GrokerGameState Klassenreferenz	19
7.9.1	Beschreibung der Konstruktoren und Destruktoren	19
7.9.1.1	GrokerGameState()	19
7.9.2	Dokumentation der Elementfunktionen	19
7.9.2.1	applyChanges(GrokerResponse response, Ai ai)	19
7.9.2.2	applyChanges(GrokerUpdate changes)	19
7.9.2.3	calculatePoints()	19
7.9.2.4	clearChanges(Ai ai)	19

7.9.2.5	<a href="#">getChanges(Ai ai)</a>	19
7.9.3	<a href="#">Dokumentation der Datenelemente</a>	19
7.9.3.1	<a href="#">chips</a>	20
7.9.3.2	<a href="#">output</a>	20
7.9.3.3	<a href="#">wonChips</a>	20
7.10	<a href="#">org.pixelgaffer.turnierserver.groker.logic.GrokerLogic Klassenreferenz</a>	20
7.10.1	<a href="#">Dokumentation der Elementfunktionen</a>	20
7.10.1.1	<a href="#">aiTimeout()</a>	20
7.10.1.2	<a href="#">createGameState()</a>	20
7.10.1.3	<a href="#">createUserObject(Ai ai)</a>	20
7.10.1.4	<a href="#">gameFinished()</a>	20
7.10.1.5	<a href="#">lost(Ai ai)</a>	20
7.10.1.6	<a href="#">parse(String string)</a>	20
7.10.1.7	<a href="#">sendFirstRenderData()</a>	21
7.10.1.8	<a href="#">setup()</a>	21
7.10.1.9	<a href="#">update()</a>	21
7.11	<a href="#">org.pixelgaffer.turnierserver.groker.logic.GrokerRenderData Klassenreferenz</a>	21
7.11.1	<a href="#">Beschreibung der Konstruktoren und Destruktoren</a>	21
7.11.1.1	<a href="#">GrokerRenderData(GrokerGameState state, String name1, String name2)</a>	21
7.11.1.2	<a href="#">GrokerRenderData(String name1, String name2)</a>	21
7.11.2	<a href="#">Dokumentation der Datenelemente</a>	21
7.11.2.1	<a href="#">chips</a>	21
7.11.2.2	<a href="#">output</a>	21
7.11.2.3	<a href="#">wonChips</a>	21
7.12	<a href="#">org.pixelgaffer.turnierserver.groker.GrokerResponse Klassenreferenz</a>	21
7.12.1	<a href="#">Beschreibung der Konstruktoren und Destruktoren</a>	22
7.12.1.1	<a href="#">GrokerResponse(String s)</a>	22
7.12.2	<a href="#">Dokumentation der Datenelemente</a>	22
7.12.2.1	<a href="#">chips</a>	22
7.12.2.2	<a href="#">output</a>	22
7.13	<a href="#">org.pixelgaffer.turnierserver.groker.GrokerUpdate Klassenreferenz</a>	22
7.13.1	<a href="#">Dokumentation der Elementfunktionen</a>	22
7.13.1.1	<a href="#">toString()</a>	22
7.13.2	<a href="#">Dokumentation der Datenelemente</a>	22
7.13.2.1	<a href="#">enemyWonChips</a>	22
7.13.2.2	<a href="#">ownWonChips</a>	22
7.14	<a href="#">org.pixelgaffer.turnierserver.groker.SimplePlayer Klassenreferenz</a>	22
7.14.1	<a href="#">Beschreibung der Konstruktoren und Destruktoren</a>	23
7.14.1.1	<a href="#">SimplePlayer(String[] args)</a>	23
7.14.2	<a href="#">Dokumentation der Elementfunktionen</a>	23

7.14.2.1	einsatz(AiDaten du, AiDaten gegner)	23
7.14.2.2	main(String[] args)	23
<b>8</b>	<b>Datei-Dokumentation</b>	<b>25</b>
8.1	mainpage.h-Dateireferenz	25
8.2	/home/dominic/git/Groker/simpleplayer/C/CAi.c-Dateireferenz	25
8.2.1	Dokumentation der Funktionen	25
8.2.1.1	calc(Player *me, Player *enemy, OutputBuffer *out)	25
8.3	/home/dominic/git/Groker/simpleplayer/C/CppAi.cxx-Dateireferenz	25
8.4	/home/dominic/git/Groker/simpleplayer/Java/org/pixelgaffer/turnierserver/groker/SimplePlayer.java-Dateireferenz	25
8.5	/home/dominic/git/Groker/simpleplayer/Python/ai.py-Dateireferenz	26
8.6	/home/dominic/git/Groker/src/main/c/grokerWrapper.c-Dateireferenz	26
8.6.1	Dokumentation der Funktionen	26
8.6.1.1	grokerMainLoop(Wrapper *w, GROKER_CALLBACK(callback))	26
8.6.1.2	parsePlayer(char *s)	26
8.7	/home/dominic/git/Groker/src/main/c/grokerWrapper.h-Dateireferenz	26
8.7.1	Makro-Dokumentation	27
8.7.1.1	GROKER_CALLBACK	27
8.7.1.2	GROKER_MAIN	27
8.7.2	Dokumentation der benutzerdefinierten Typen	28
8.7.2.1	Player	28
8.7.2.2	Result	28
8.7.2.3	Spieler	28
8.7.3	Dokumentation der Funktionen	28
8.7.3.1	__c_crash(Wrapper *w, const char *reason)	28
8.7.3.2	__c_surrender(Wrapper *w)	28
8.7.3.3	grokerMainLoop(Wrapper *w, GROKER_CALLBACK(callback))	28
8.7.3.4	parsePlayer(char *s)	28
8.8	/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/ai/AiDaten.java-↔ Dateireferenz	28
8.9	/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/ai/GrokerAi.java-↔ Dateireferenz	28
8.10	/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/GrokerGameState.java-Dateireferenz	29
8.11	/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/GrokerResponse.java-Dateireferenz	29
8.12	/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/GrokerUpdate.java-↔ Dateireferenz	29
8.13	/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/logic/GrokerAiObject.java-Dateireferenz	29
8.14	/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/logic/GrokerLogic.java-Dateireferenz	30

---

8.15	<a href="#">/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/logic/GrokerRender↵</a>	
	<a href="#">Data.java-Dateireferenz</a> . . . . .	30
8.16	<a href="#">/home/dominic/git/Groker/src/main/python/game_wrapper.py-Dateireferenz</a> . . . . .	30
<b>Index</b>		<b>31</b>





# Kapitel 1

## Hauptseite

Groker ist eine KI-Aufgabe des 34. Bwlnf Runde 1. Dabei setzen die KIs immer gleichzeitig eine bestimmte Anzahl an Chips. Wenn sich die Gebote der beiden KIs nicht mehr als 5 Chips unterscheiden, dürfen beide KIs ihren Einsatz behalten. Ansonsten darf nur die KI mit dem geringeren Einsatz ihre Chips behalten. Eine KI gewinnt, wenn sie mehr als 100 Chips mehr als der Gegner hat. [Offizielle Aufgabenstellung](#)

Die Implementation einer KI ist in verschiedenen Programmiersprachen möglich. Generell sind die sogenannten Simple Player, die eine minimale KI sind, gut dokumentiert. Hier sind ein paar nützliche Doku-Links und Hinweise für verschiedenen Programmiersprachen (Go wird soweit ich weiß von doxygen nicht unterstützt):

### C++

- [GrokerAi](#)
- [\\_player](#)

Um eine KI in C++ zu schreiben, muss die Klasse [GrokerAi](#) erweitert werden und mithilfe des Makros

```
GROKER_MAIN (ClassName)
```

eine main-Methode erstellt werden.

### C

Um eine KI in C zu schreiben muss eine Funktion

```
int xxx (Player *me, Player *enemy, OutputBuffer *out)
```

implementiert werden und mithilfe des Makros

```
GROKER_MAIN (xxx)
```

eine main-Methode erstellt werden, die die KI startet.

### Java

- [org.pixelgaffer.turnierserver.groker.ai.GrokerAi](#)

Um eine KI in Java zu schreiben, muss die Klasse [org.pixelgaffer.turnierserver.groker.ai.GrokerAi](#) erweitert werden und in der main-Methode muss eine neue Instanz davon mit den Argumenten der main-Methode erstellt und die start()-Methode aufgerufen werden.

## Python

- [ai.AI](#)

## Kapitel 2

# Verzeichnis der Namensbereiche

### 2.1 Liste aller Namensbereiche

Liste aller Namensbereiche mit Kurzbeschreibung:

ai	11
game_wrapper	11
org	11
org.pixelgaffer	11
org.pixelgaffer.turnierserver	11
org.pixelgaffer.turnierserver.groker	11
org.pixelgaffer.turnierserver.groker.ai	12
org.pixelgaffer.turnierserver.groker.logic	12



## Kapitel 3

# Hierarchie-Verzeichnis

### 3.1 Klassenhierarchie

Die Liste der Ableitungen ist -mit Einschränkungen- alphabetisch sortiert:

_player . . . . .	13
_result . . . . .	13
ai.AI . . . . .	14
org.pixelgaffer.turnierserver.groker.ai.AiDaten . . . . .	14
GrokerAi . . . . .	17
org.pixelgaffer.turnierserver.groker.logic.GrokerRenderData . . . . .	21
org.pixelgaffer.turnierserver.groker.GrokerResponse . . . . .	21
org.pixelgaffer.turnierserver.groker.GrokerUpdate . . . . .	22
Ai	
org.pixelgaffer.turnierserver.groker.ai.GrokerAi . . . . .	15
org.pixelgaffer.turnierserver.groker.SimplePlayer . . . . .	22
AIObject	
org.pixelgaffer.turnierserver.groker.logic.GrokerAiObject . . . . .	18
AIWrapper	
game_wrapper.GameWrapper . . . . .	15
GameState	
org.pixelgaffer.turnierserver.groker.GrokerGameState . . . . .	19
TurnBasedGameLogic	
org.pixelgaffer.turnierserver.groker.logic.GrokerLogic . . . . .	20



## Kapitel 4

# Klassen-Verzeichnis

### 4.1 Auflistung der Klassen

Hier folgt die Aufzählung aller Klassen, Strukturen, Varianten und Schnittstellen mit einer Kurzbeschreibung:

<a href="#">_player</a>	Diese Klasse speichert die für die KI sichtbaren Daten einer KI . . . . .	13
<a href="#">_result</a>	Diese Klasse repräsentiert das Ergebnis einer Runde . . . . .	13
<a href="#">ai.AI</a>	. . . . .	14
<a href="#">org.pixelgaffer.turnierserver.groker.ai.AiDaten</a>	. . . . .	14
<a href="#">game_wrapper.GameWrapper</a>	. . . . .	15
<a href="#">org.pixelgaffer.turnierserver.groker.ai.GrokerAi</a>	Dies ist die Mutterklasse aller Groker KIs . . . . .	15
<a href="#">GrokerAi</a>	Dies ist die Mutter-Klasse von jeder Groker-KI . . . . .	17
<a href="#">org.pixelgaffer.turnierserver.groker.logic.GrokerAiObject</a>	. . . . .	18
<a href="#">org.pixelgaffer.turnierserver.groker.GrokerGameState</a>	. . . . .	19
<a href="#">org.pixelgaffer.turnierserver.groker.logic.GrokerLogic</a>	. . . . .	20
<a href="#">org.pixelgaffer.turnierserver.groker.logic.GrokerRenderData</a>	. . . . .	21
<a href="#">org.pixelgaffer.turnierserver.groker.GrokerResponse</a>	. . . . .	21
<a href="#">org.pixelgaffer.turnierserver.groker.GrokerUpdate</a>	. . . . .	22
<a href="#">org.pixelgaffer.turnierserver.groker.SimplePlayer</a>	. . . . .	22





## Kapitel 5

# Datei-Verzeichnis

### 5.1 Auflistung der Dateien

Hier folgt die Aufzählung aller Dateien mit einer Kurzbeschreibung:

<a href="#">mainpage.h</a>	25
/home/dominic/git/Groker/simpleplayer/C/ <a href="#">CAi.c</a>	25
/home/dominic/git/Groker/simpleplayer/C/ <a href="#">CppAi.cxx</a>	25
/home/dominic/git/Groker/simpleplayer/Java/org/pixelgaffer/turnierserver/groker/ <a href="#">SimplePlayer.java</a>	25
/home/dominic/git/Groker/simpleplayer/Python/ <a href="#">ai.py</a>	26
/home/dominic/git/Groker/src/main/c/ <a href="#">grokerWrapper.c</a>	26
/home/dominic/git/Groker/src/main/c/ <a href="#">grokerWrapper.h</a>	26
/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/ <a href="#">GrokerGameState.java</a>	29
/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/ <a href="#">GrokerResponse.java</a>	29
/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/ <a href="#">GrokerUpdate.java</a>	29
/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/ai/ <a href="#">AiDaten.java</a>	28
/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/ai/ <a href="#">GrokerAi.java</a>	28
/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/logic/ <a href="#">GrokerAiObject.java</a>	29
/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/logic/ <a href="#">GrokerLogic.java</a>	30
/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/logic/ <a href="#">GrokerRenderData.java</a>	30
/home/dominic/git/Groker/src/main/python/ <a href="#">game_wrapper.py</a>	30



## Kapitel 6

# Dokumentation der Namensbereiche

### 6.1 ai-Namensbereichsreferenz

#### Klassen

- class [AI](#)

### 6.2 game\_wrapper-Namensbereichsreferenz

#### Klassen

- class [GameWrapper](#)

### 6.3 Paket org

#### Pakete

- package [pixelgaffer](#)

### 6.4 Paket org.pixelgaffer

#### Pakete

- package [turnierserver](#)

### 6.5 Paket org.pixelgaffer.turnierserver

#### Pakete

- package [groker](#)

### 6.6 Paket org.pixelgaffer.turnierserver.groker

## Pakete

- package [ai](#)
- package [logic](#)

## Klassen

- class [GrokerGameState](#)
- class [GrokerResponse](#)
- class [GrokerUpdate](#)
- class [SimplePlayer](#)

## 6.7 Paket org.pixelgaffer.turnierserver.groker.ai

### Klassen

- class [AiDaten](#)
- class [GrokerAi](#)

*Dies ist die Mutterklasse aller Groker Kls.*

## 6.8 Paket org.pixelgaffer.turnierserver.groker.logic

### Klassen

- class [GrokerAiObject](#)
- class [GrokerLogic](#)
- class [GrokerRenderData](#)

# Kapitel 7

## Klassen-Dokumentation

### 7.1 `_player` Strukturreferenz

Diese Klasse speichert die für die KI sichtbaren Daten einer KI.

```
#include <grokerWrapper.h>
```

#### Öffentliche Attribute

- `int letzterEinsatz`  
*Der Einsatz der KI in der letzten Runde.*
- `int gewonneneChips`  
*Die Anzahl der Chips, die die KI gewonnen hat.*

#### 7.1.1 Ausführliche Beschreibung

Diese Klasse speichert die für die KI sichtbaren Daten einer KI.

Dies ist der Einsatz der KI der letzten Runde sowie die Anzahl der von dieser KI gewonnenen Chips.

#### 7.1.2 Dokumentation der Datenelemente

##### 7.1.2.1 `int _player::gewonneneChips`

Die Anzahl der Chips, die die KI gewonnen hat.

##### 7.1.2.2 `int _player::letzterEinsatz`

Der Einsatz der KI in der letzten Runde.

Die Dokumentation für diese Struktur wurde erzeugt aufgrund der Datei:

- `/home/dominic/git/Groker/src/main/c/grokerWrapper.h`

### 7.2 `_result` Strukturreferenz

Diese Klasse repräsentiert das Ergebnis einer Runde.

```
#include <grokerWrapper.h>
```

## Öffentliche Attribute

- int [chips](#)
- const char \* [output](#)

### 7.2.1 Ausführliche Beschreibung

Diese Klasse repräsentiert das Ergebnis einer Runde.

Dabei wird die Anzahl der Chips, die die KI setzt, und die Ausgabe der KI gespeichert.

### 7.2.2 Dokumentation der Datenelemente

#### 7.2.2.1 int \_result::chips

#### 7.2.2.2 const char\* \_result::output

Die Dokumentation für diese Struktur wurde erzeugt aufgrund der Datei:

- </home/dominic/git/Groker/src/main/c/grokerWrapper.h>

## 7.3 ai.AI Klassenreferenz

### Öffentliche Methoden

- def [einsatz](#) (self)
- def [process](#) (self, ownWonChips, enemyWonChips, ownChips, enemyChips)

### 7.3.1 Dokumentation der Elementfunktionen

#### 7.3.1.1 def ai.AI.einsatz ( self )

Diese Funktion wird jede Runde aufgerufen, und gibt die Punkte zurück, die gesetzt werden sollen.

#### 7.3.1.2 def ai.AI.process ( self, ownWonChips, enemyWonChips, ownChips, enemyChips )

Diese Funktion wird nach jedem Schritt aufgerufen und ist dafür da, Daten zu verarbeiten.  
Es werden ownWonChips (gesamten Chips, die die eigene KI bekommen hat),  
enemyWonChips (gesamten Chips, die der gegner bekommen hat),  
ownChips (Chips, die man diese Runde gesetzt hat),  
enemyChips (Chips, die der Gegner diese Runde gesetzt hat) übergeben.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- </home/dominic/git/Groker/simpleplayer/Python/ai.py>

## 7.4 org.pixelgaffer.turnierserver.groker.ai.AiDaten Klassenreferenz

### Öffentliche Attribute

- int [letzterEinsatz](#)

*Der Einsatz in der letzten Runde.*

- int [gewonneneChips](#)

*Die gesamte Anzahl an gewonnenen Chips dieser KI.*

### 7.4.1 Dokumentation der Datenelemente

#### 7.4.1.1 int org.pixelgaffer.turnierserver.groker.ai.AiDaten.gewonneneChips

Die gesamte Anzahl an gewonnenen Chips dieser KI.

#### 7.4.1.2 int org.pixelgaffer.turnierserver.groker.ai.AiDaten.letzterEinsatz

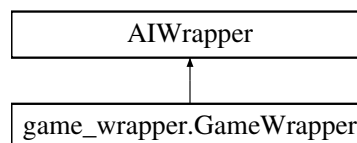
Der Einsatz in der letzten Runde.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- /home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/ai/[AiDaten.java](#)

## 7.5 game\_wrapper.GameWrapper Klassenreferenz

Klassendiagramm für game\_wrapper.GameWrapper:



### Öffentliche Methoden

- def [update](#) (self, updates)
- def [process](#) (self, d)
- def [add\\_output](#) (self, d, o)

### 7.5.1 Dokumentation der Elementfunktionen

#### 7.5.1.1 def game\_wrapper.GameWrapper.add\_output ( self, d, o )

#### 7.5.1.2 def game\_wrapper.GameWrapper.process ( self, d )

#### 7.5.1.3 def game\_wrapper.GameWrapper.update ( self, updates )

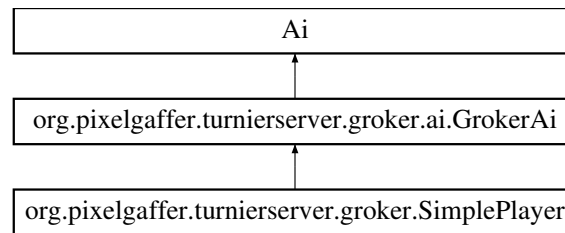
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- /home/dominic/git/Groker/src/main/python/[game\\_wrapper.py](#)

## 7.6 org.pixelgaffer.turnierserver.groker.ai.GrokerAi Klassenreferenz

Dies ist die Mutterklasse aller Groker KIs.

Klassendiagramm für org.pixelgaffer.turnierserver.groker.ai.GrokerAi:



## Öffentliche Methoden

- **GrokerAi** (String[] args)  
*Dies instantiiert die KI.*
- abstract int **einsatz** (AiDaten du, AiDaten gegner)  
*Diese Methode soll den Einsatz dieser KI zurückgeben.*

## Geschützte Methoden

- String **update** (String answer)

### 7.6.1 Ausführliche Beschreibung

Dies ist die Mutterklasse aller Groker KIs.

Jede KI muss diese Klasse erweitern. Nach der Instantiierung muss die start() methode aufgerufen werden, damit die KI anfängt, auf Nachrichten von der Spiellogik zu hören.

Rufe surrender() auf um ihne Grund aufzugeben Rufe crash(Throwable t) oder crash(String reason) auf, um zu signalisieren, dass es einen crash gab. Dies hat den gleichen Effekt wie surrender(), nur dass du die Fehlermeldung auf der Webseite ausgegeben bekommst.

### 7.6.2 Beschreibung der Konstruktoren und Destruktoren

#### 7.6.2.1 org.pixelgaffer.turnierserver.groker.ai.GrokerAi.GrokerAi ( String[] args ) [inline]

Dies instantiiert die KI.

Vergiss nicht start() aufzurufen!

Parameter

<i>args</i>	Die in main(String[] args) übergebenen Kommandozeilenargumente
-------------	--

### 7.6.3 Dokumentation der Elementfunktionen

#### 7.6.3.1 abstract int org.pixelgaffer.turnierserver.groker.ai.GrokerAi.einsatz ( AiDaten du, AiDaten gegner ) [abstract]

Diese Methode soll den Einsatz dieser KI zurückgeben.

Parameter



<i>du</i>	Die <a href="#">AiDaten</a> dieser KI der letzten Runde
<i>gegner</i>	Die <a href="#">AiDaten</a> der Gegner-KI der letzten Runde

### Rückgabe

Eine Zahl  $\geq 1$ , der Einsatz dieser KI für diese Runde

7.6.3.2 `String org.pixelgaffer.turnierserver.groker.ai.GrokerAi.update ( String answer ) [inline], [protected]`

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- `/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/ai/GrokerAi.java`

## 7.7 GrokerAi Klassenreferenz

Dies ist die Mutter-Klasse von jeder Groker-KI.

```
#include <grokerWrapper.h>
```

### Öffentliche Methoden

- [GrokerAi](#) (Wrapper \*wrapper)
- virtual int [calc](#) (Player \*me, Player \*enemy)=0  
*Wird jede Runde aufgerufen.*
- const char \* [readOutput](#) ()  
*Gibt den Inhalt des aktuellen Output Buffers zurück und leert diesen.*

### Geschützte Methoden

- void [crash](#) (const char \*reason)  
*Lässt die KI mit Angabe eines Grundes abstürzen.*
- void [surrender](#) ()  
*Lässt die KI ohne Angabe eines Grundes aufgeben.*
- std::stringstream & [out](#) ()  
*Gibt ein `std::stringstream` zurück, an das die KI ihre Ausgaben schicken kann.*

### 7.7.1 Ausführliche Beschreibung

Dies ist die Mutter-Klasse von jeder Groker-KI.

Sie muss im Kontruktor genau ein Argument, `Wrapper*`, entgegennehmen, und an diese Klasse weitergeben. Zudem muss die pure virtual Methode `calc` überschrieben werden. Diese Methode wird jede Runde aufgerufen. Zudem enthält diese Klasse eine Methode `names out`, die ein `std::stringstream` zurückgibt, an das die KI ihre Ausgabe schicken kann. **Alles, was die KI an `std::cout` sendet, geht verloren!!!**. Der Name dieser Klasse gehört als Argument an das `GROKER_MAIN` Makro.

## 7.7.2 Beschreibung der Konstruktoren und Destruktoren

7.7.2.1 `GrokerAi::GrokerAi ( Wrapper * wrapper ) [inline],[explicit]`

## 7.7.3 Dokumentation der Elementfunktionen

7.7.3.1 `virtual int GrokerAi::calc ( Player * me, Player * enemy ) [pure virtual]`

Wird jede Runde aufgerufen.

Gibt die Anzahl der Chips, die die KI einsetzt, zurück.

Parameter

<i>me</i>	Die eigene KI.
<i>enemy</i>	Die gegnerische KI.

Rückgabe

Die Anzahl an Chips, die die KI setzen will.

7.7.3.2 `void GrokerAi::crash ( const char * reason ) [inline],[protected]`

Lässt die KI mit Angabe eines Grundes abstürzen.

Parameter

<i>reason</i>	Der Grund, aus dem die KI abstürzt.
---------------	-------------------------------------

7.7.3.3 `std::stringstream& GrokerAi::out ( ) [inline],[protected]`

Gibt ein `std::stringstream` zurück, an das die KI ihre Ausgaben schicken kann.

7.7.3.4 `const char* GrokerAi::readOutput ( ) [inline]`

Gibt den Inhalt des aktuellen Output Buffers zurück und leert diesen.

7.7.3.5 `void GrokerAi::surrender ( ) [inline],[protected]`

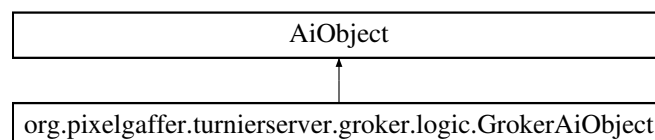
Lässt die KI ohne Angabe eines Grundes aufgeben.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [/home/dominic/git/Groker/src/main/c/grokerWrapper.h](#)

## 7.8 org.pixelgaffer.turnierserver.groker.logic.GrokerAiObject Klassenreferenz

Klassendiagramm für `org.pixelgaffer.turnierserver.groker.logic.GrokerAiObject`:

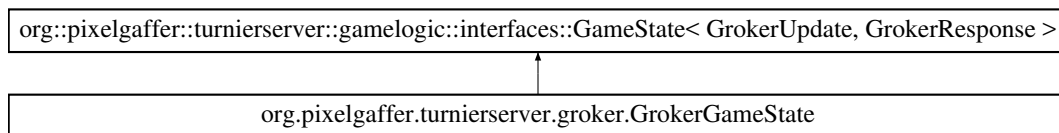


Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/logic/GrokerAiObject.java](#)

## 7.9 org.pixelgaffer.turnierserver.groker.GrokerGameState Klassenreferenz

Klassendiagramm für org.pixelgaffer.turnierserver.groker.GrokerGameState:



### Öffentliche Methoden

- [GrokerGameState](#) ()
  - [GrokerUpdate](#) [getChanges](#) (Ai ai)
  - void [clearChanges](#) (Ai ai)
  - void [applyChanges](#) ([GrokerResponse](#) response, Ai ai)
  - void [calculatePoints](#) ()
- Wird nur auf dem Server verwendet, bitte in der AI nicht aufrufen.*
- void [applyChanges](#) ([GrokerUpdate](#) changes)

### Öffentliche Attribute

- int[] [wonChips](#)
- int[] [chips](#)
- String[] [output](#)

#### 7.9.1 Beschreibung der Konstruktoren und Destruktoren

7.9.1.1 `org.pixelgaffer.turnierserver.groker.GrokerGameState.GrokerGameState ( )` `[inline]`

#### 7.9.2 Dokumentation der Elementfunktionen

7.9.2.1 `void org.pixelgaffer.turnierserver.groker.GrokerGameState.applyChanges ( GrokerResponse response, Ai ai )` `[inline]`

7.9.2.2 `void org.pixelgaffer.turnierserver.groker.GrokerGameState.applyChanges ( GrokerUpdate changes )` `[inline]`

7.9.2.3 `void org.pixelgaffer.turnierserver.groker.GrokerGameState.calculatePoints ( )` `[inline]`

Wird nur auf dem Server verwendet, bitte in der AI nicht aufrufen.

7.9.2.4 `void org.pixelgaffer.turnierserver.groker.GrokerGameState.clearChanges ( Ai ai )` `[inline]`

7.9.2.5 `GrokerUpdate org.pixelgaffer.turnierserver.groker.GrokerGameState.getChanges ( Ai ai )` `[inline]`

#### 7.9.3 Dokumentation der Datenelemente

7.9.3.1 `int [] org.pixelgaffer.turnierserver.groker.GrokerGameState.chips`

7.9.3.2 `String [] org.pixelgaffer.turnierserver.groker.GrokerGameState.output`

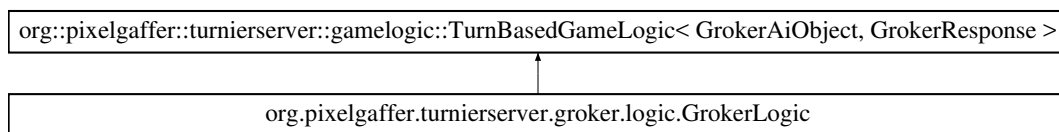
7.9.3.3 `int [] org.pixelgaffer.turnierserver.groker.GrokerGameState.wonChips`

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- `/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/GrokerGameState.java`

## 7.10 org.pixelgaffer.turnierserver.groker.logic.GrokerLogic Klassenreferenz

Klassendiagramm für `org.pixelgaffer.turnierserver.groker.logic.GrokerLogic`:



### Öffentliche Methoden

- void `lost` (Ai ai)
- float `aiTimeout` ()

### Geschützte Methoden

- Object `update` ()
- GameState<?, GrokerResponse> `createGameState` ()
- void `setup` ()
- GrokerAiObject `createUserObject` (Ai ai)
- void `gameFinished` ()
- void `sendFirstRenderData` ()
- GrokerResponse `parse` (String string) throws ParseException

### 7.10.1 Dokumentation der Elementfunktionen

7.10.1.1 `float org.pixelgaffer.turnierserver.groker.logic.GrokerLogic.aiTimeout ( )` `[inline]`

7.10.1.2 `GameState<?, GrokerResponse> org.pixelgaffer.turnierserver.groker.logic.GrokerLogic.createGameState ( )` `[inline]`, `[protected]`

7.10.1.3 `GrokerAiObject org.pixelgaffer.turnierserver.groker.logic.GrokerLogic.createUserObject ( Ai ai )` `[inline]`, `[protected]`

7.10.1.4 `void org.pixelgaffer.turnierserver.groker.logic.GrokerLogic.gameFinished ( )` `[inline]`, `[protected]`

7.10.1.5 `void org.pixelgaffer.turnierserver.groker.logic.GrokerLogic.lost ( Ai ai )` `[inline]`

7.10.1.6 `GrokerResponse org.pixelgaffer.turnierserver.groker.logic.GrokerLogic.parse ( String string )` throws `ParseException` `[inline]`, `[protected]`

7.10.1.7 void org.pixelgaffer.turnierserver.groker.logic.GrokerLogic.sendFirstRenderData ( ) [inline],  
[protected]

7.10.1.8 void org.pixelgaffer.turnierserver.groker.logic.GrokerLogic.setup ( ) [inline], [protected]

7.10.1.9 Object org.pixelgaffer.turnierserver.groker.logic.GrokerLogic.update ( ) [inline], [protected]

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- /home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/logic/[GrokerLogic.java](#)

## 7.11 org.pixelgaffer.turnierserver.groker.logic.GrokerRenderData Klassenreferenz

### Öffentliche Methoden

- [GrokerRenderData](#) ([GrokerGameState](#) state, String name1, String name2)
- [GrokerRenderData](#) (String name1, String name2)

### Öffentliche Attribute

- HashMap< String, Integer > [chips](#)
- HashMap< String, Integer > [wonChips](#)
- HashMap< String, String > [output](#)

### 7.11.1 Beschreibung der Konstruktoren und Destruktoren

7.11.1.1 org.pixelgaffer.turnierserver.groker.logic.GrokerRenderData.GrokerRenderData ( [GrokerGameState](#) state, String  
name1, String name2 ) [inline]

7.11.1.2 org.pixelgaffer.turnierserver.groker.logic.GrokerRenderData.GrokerRenderData ( String name1, String name2 )  
[inline]

### 7.11.2 Dokumentation der Datenelemente

7.11.2.1 HashMap<String, Integer> org.pixelgaffer.turnierserver.groker.logic.GrokerRenderData.chips

7.11.2.2 HashMap<String, String> org.pixelgaffer.turnierserver.groker.logic.GrokerRenderData.output

7.11.2.3 HashMap<String, Integer> org.pixelgaffer.turnierserver.groker.logic.GrokerRenderData.wonChips

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- /home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/logic/[GrokerRenderData.java](#)

## 7.12 org.pixelgaffer.turnierserver.groker.GrokerResponse Klassenreferenz

### Öffentliche Methoden

- [GrokerResponse](#) (String s)

## Öffentliche Attribute

- int [chips](#)
- String [output](#)

### 7.12.1 Beschreibung der Konstruktoren und Destruktoren

7.12.1.1 `org.pixelgaffer.turnierserver.groker.GrokerResponse.GrokerResponse ( String s )` `[inline]`

### 7.12.2 Dokumentation der Datenelemente

7.12.2.1 `int org.pixelgaffer.turnierserver.groker.GrokerResponse.chips`

7.12.2.2 `String org.pixelgaffer.turnierserver.groker.GrokerResponse.output`

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- `/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/GrokerResponse.java`

## 7.13 `org.pixelgaffer.turnierserver.groker.GrokerUpdate` Klassenreferenz

### Öffentliche Methoden

- String [toString](#) ()

### Öffentliche Attribute

- int [ownWonChips](#)
- int [enemyWonChips](#)

### 7.13.1 Dokumentation der Elementfunktionen

7.13.1.1 `String org.pixelgaffer.turnierserver.groker.GrokerUpdate.toString ( )` `[inline]`

### 7.13.2 Dokumentation der Datenelemente

7.13.2.1 `int org.pixelgaffer.turnierserver.groker.GrokerUpdate.enemyWonChips`

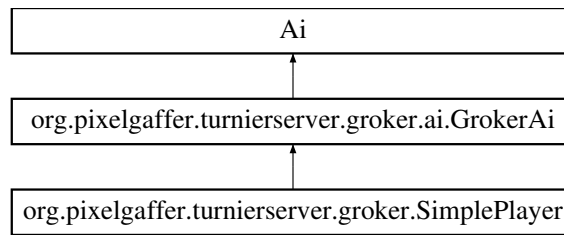
7.13.2.2 `int org.pixelgaffer.turnierserver.groker.GrokerUpdate.ownWonChips`

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- `/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/GrokerUpdate.java`

## 7.14 `org.pixelgaffer.turnierserver.groker.SimplePlayer` Klassenreferenz

Klassendiagramm für `org.pixelgaffer.turnierserver.groker.SimplePlayer`:



### Öffentliche Methoden

- [SimplePlayer](#) (String[] args)
- int [einsatz](#) ([AiDaten](#) du, [AiDaten](#) gegner)

### Öffentliche, statische Methoden

- static void [main](#) (String[] args)

### Weitere Geerbte Elemente

#### 7.14.1 Beschreibung der Konstruktoren und Destruktoren

7.14.1.1 `org.pixelgaffer.turnierserver.groker.SimplePlayer.SimplePlayer ( String[] args )` [inline]

#### 7.14.2 Dokumentation der Elementfunktionen

7.14.2.1 `int org.pixelgaffer.turnierserver.groker.SimplePlayer.einsatz ( AiDaten du, AiDaten gegner )` [inline]

7.14.2.2 `static void org.pixelgaffer.turnierserver.groker.SimplePlayer.main ( String[] args )` [inline], [static]

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- `/home/dominic/git/Groker/simpleplayer/Java/org/pixelgaffer/turnierserver/groker/SimplePlayer.java`





## Kapitel 8

# Datei-Dokumentation

### 8.1 mainpage.h-Dateireferenz

### 8.2 /home/dominic/git/Groker/simpleplayer/C/CAi.c-Dateireferenz

```
#include "grokerWrapper.h"
```

#### Funktionen

- `int calc (Player *me, Player *enemy, OutputBuffer *out)`  
*Diese Methode wird jede Runde aufgerufen.*

#### 8.2.1 Dokumentation der Funktionen

##### 8.2.1.1 `int calc ( Player * me, Player * enemy, OutputBuffer * out )`

Diese Methode wird jede Runde aufgerufen.

Player ist ein struct mit den beiden int-Werten `letzterEinsatz` und `gewonneneChips`. Der return-Wert ist der Einsatz der KI. Zur Ausgabe stehen die folgenden Funktionen zur Verfügung:

- `append (out, "text")`: Gibt "text" aus
- `appendi(out, 123456)`: Gibt 123465 aus
- `appendd(out, 123.45)`: Gibt 123.45 aus

### 8.3 /home/dominic/git/Groker/simpleplayer/C/CppAi.cxx-Dateireferenz

### 8.4 /home/dominic/git/Groker/simpleplayer/Java/org/pixelgaffer/turnierserver/groker/↵ SimplePlayer.java-Dateireferenz

#### Klassen

- class `org.pixelgaffer.turnierserver.groker.SimplePlayer`

## Pakete

- package [org.pixelgaffer.turnierserver.groker](#)

## 8.5 /home/dominic/git/Groker/simpleplayer/Python/ai.py-Dateireferenz

### Klassen

- class [ai.AI](#)

### Namensbereiche

- [ai](#)

## 8.6 /home/dominic/git/Groker/src/main/c/grokerWrapper.c-Dateireferenz

```
#include "output.h"
#include "grokerWrapper.h"
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
```

### Funktionen

- [Player](#) \* [parsePlayer](#) (char \*s)  
*Diese Methode parst den String s zu einem pointer auf eine Player-Struktur.*
- char \* [grokerMainLoop](#) (Wrapper \*w, [GROKER\\_CALLBACK](#)(callback))  
*Dies ist die MainLoop von Grooker.*

### 8.6.1 Dokumentation der Funktionen

#### 8.6.1.1 char\* grokerMainLoop ( Wrapper \* w, GROKER\_CALLBACK(callback) )

Dies ist die MainLoop von Grooker.

Sie wartet auf Daten vom Server, leitet diese an die eigentliche KI weiter, und sendet diese Daten zurück.

#### 8.6.1.2 Player\* parsePlayer ( char \* s )

Diese Methode parst den String s zu einem pointer auf eine Player-Struktur.

## 8.7 /home/dominic/git/Groker/src/main/c/grokerWrapper.h-Dateireferenz

```
#include "wrapper.h"
#include <stdlib.h>
#include <sstream>
```

## Klassen

- struct `_player`  
*Diese Klasse speichert die für die KI sichtbaren Daten einer KI.*
- struct `_result`  
*Diese Klasse repräsentiert das Ergebnis einer Runde.*
- class `GrokerAi`  
*Dies ist die Mutter-Klasse von jeder Groker-KI.*

## Makrodefinitionen

- `#define GROKER_CALLBACK(name) Result* (* name) (Player*, Player*)`
- `#define GROKER_MAIN(clazzname)`

## Typdefinitionen

- `typedef struct _player Player`
- `typedef struct _player Spieler`
- `typedef struct _result Result`

## Funktionen

- `Player* parsePlayer (char *s)`  
*Diese Methode parst den String `s` zu einem pointer auf eine `Player`-Struktur.*
- `char* grokerMainLoop (Wrapper *w, GROKER_CALLBACK(callback))`  
*Dies ist die MainLoop von Groker.*
- `void __c_crash (Wrapper *w, const char *reason)`
- `void __c_surrender (Wrapper *w)`

### 8.7.1 Makro-Dokumentation

8.7.1.1 `#define GROKER_CALLBACK( name ) Result* (* name) (Player*, Player*)`

8.7.1.2 `#define GROKER_MAIN( clazzname )`

Wert:

```
clazzname *__ai = 0; \
\
Result* __callback (Player *me, Player *enemy) \
{ \
    Result *r = (Result*) malloc(sizeof(Result)); \
    r->chips = __ai->calc(me, enemy); \
    r->output = __ai->readOutput(); \
    return r; \
} \
\
int main (int argc, char **argv) \
{ \
    Wrapper *w = globalInit(argc, argv); \
    __ai = new clazzname (w); \
    grokerMainLoop(w, __callback); \
    globalCleanup(&w); \
    delete __ai; \
    return 0; \
}
```

## 8.7.2 Dokumentation der benutzerdefinierten Typen

8.7.2.1 `typedef struct _player Player`

8.7.2.2 `typedef struct _result Result`

8.7.2.3 `typedef struct _player Spieler`

## 8.7.3 Dokumentation der Funktionen

8.7.3.1 `void __c_crash ( Wrapper * w, const char * reason )`

8.7.3.2 `void __c_surrender ( Wrapper * w )`

8.7.3.3 `char* grokerMainLoop ( Wrapper * w, GROKER_CALLBACK(callback) )`

Dies ist die MainLoop von Grooker.

Sie wartet auf Daten vom Server, leitet diese an die eigentliche KI weiter, und sendet diese Daten zurück.

8.7.3.4 `Player* parsePlayer ( char * s )`

Diese Methode parst den String `s` zu einem pointer auf eine `Player`-Struktur.

## 8.8 [/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/ai/Ai](#)↵ Daten.java-Dateireferenz

### Klassen

- class [org.pixelgaffer.turnierserver.groker.ai.AiDaten](#)

### Pakete

- package [org.pixelgaffer.turnierserver.groker.ai](#)

## 8.9 [/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/ai/](#)↵ GrokerAi.java-Dateireferenz

### Klassen

- class [org.pixelgaffer.turnierserver.groker.ai.GrokerAi](#)

*Dies ist die Mutterklasse aller Groker Kls.*

### Pakete

- package [org.pixelgaffer.turnierserver.groker.ai](#)

## 8.10 /home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/Groker↔ GameState.java-Dateireferenz

### Klassen

- class [org.pixelgaffer.turnierserver.groker.GrokerGameState](#)

### Pakete

- package [org.pixelgaffer.turnierserver.groker](#)

## 8.11 /home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/Groker↔ Response.java-Dateireferenz

### Klassen

- class [org.pixelgaffer.turnierserver.groker.GrokerResponse](#)

### Pakete

- package [org.pixelgaffer.turnierserver.groker](#)

## 8.12 /home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/Groker↔ Update.java-Dateireferenz

### Klassen

- class [org.pixelgaffer.turnierserver.groker.GrokerUpdate](#)

### Pakete

- package [org.pixelgaffer.turnierserver.groker](#)

## 8.13 /home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/logic/↔ GrokerAiObject.java-Dateireferenz

### Klassen

- class [org.pixelgaffer.turnierserver.groker.logic.GrokerAiObject](#)

### Pakete

- package [org.pixelgaffer.turnierserver.groker.logic](#)

## 8.14 [/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/logic/↔](#) GrokerLogic.java-Dateireferenz

### Klassen

- class [org.pixelgaffer.turnierserver.groker.logic.GrokerLogic](#)

### Pakete

- package [org.pixelgaffer.turnierserver.groker.logic](#)

## 8.15 [/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/logic/↔](#) GrokerRenderData.java-Dateireferenz

### Klassen

- class [org.pixelgaffer.turnierserver.groker.logic.GrokerRenderData](#)

### Pakete

- package [org.pixelgaffer.turnierserver.groker.logic](#)

## 8.16 [/home/dominic/git/Groker/src/main/python/game\\_wrapper.py-Dateireferenz](#)

### Klassen

- class [game\\_wrapper.GameWrapper](#)

### Namensbereiche

- [game\\_wrapper](#)

# Index

/home/dominic/git/Groker/simpleplayer/C/CAi.c, 25  
/home/dominic/git/Groker/simpleplayer/C/CppAi.cxx, 25  
/home/dominic/git/Groker/simpleplayer/Java/org/pixelgaffer/turnierserver/groker/↔  
SimplePlayer.java, 25  
/home/dominic/git/Groker/simpleplayer/Python/ai.py, 26  
/home/dominic/git/Groker/src/main/c/grokerWrapper.c, 26  
/home/dominic/git/Groker/src/main/c/grokerWrapper.h, 26  
/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/↔  
GrokerGameState.java, 29  
/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/↔  
GrokerResponse.java, 29  
/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/↔  
GrokerUpdate.java, 29  
/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/ai/↔  
AiDaten.java, 28  
/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/ai/↔  
GrokerAi.java, 28  
/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/logic/↔  
GrokerAiObject.java, 29  
/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/logic/↔  
GrokerLogic.java, 30  
/home/dominic/git/Groker/src/main/java/org/pixelgaffer/turnierserver/groker/logic/↔  
GrokerRenderData.java, 30  
/home/dominic/git/Groker/src/main/python/game\_↔  
wrapper.py, 30  
\_\_c\_crash  
grokerWrapper.h, 28  
\_\_c\_surrender  
grokerWrapper.h, 28  
\_player, 13  
gewonneneChips, 13  
letzterEinsatz, 13  
\_result, 13  
chips, 14  
output, 14  
add\_output  
game\_wrapper::GameWrapper, 15  
ai, 11  
ai.AI, 14  
ai::AI  
einsatz, 14  
process, 14  
aiTimeout  
org::pixelgaffer::turnierserver::groker::logic::↔  
GrokerLogic, 20  
applyChanges  
org::pixelgaffer::turnierserver::groker::Groker↔  
GameState, 19  
CAi.c  
calc, 25  
calc  
CAi.c, 25  
GrokerAi, 18  
calculatePoints  
org::pixelgaffer::turnierserver::groker::Groker↔  
GameState, 19  
chips  
\_result, 14  
org::pixelgaffer::turnierserver::groker::Groker↔  
GameState, 19  
org::pixelgaffer::turnierserver::groker::Groker↔  
Response, 22  
org::pixelgaffer::turnierserver::groker::logic::↔  
GrokerRenderData, 21  
clearChanges  
org::pixelgaffer::turnierserver::groker::Groker↔  
GameState, 19  
crash  
GrokerAi, 18  
createGameState  
org::pixelgaffer::turnierserver::groker::logic::↔  
GrokerLogic, 20  
createUserObject  
org::pixelgaffer::turnierserver::groker::logic::↔  
GrokerLogic, 20  
einsatz  
ai::AI, 14  
org::pixelgaffer::turnierserver::groker::Simple↔  
Player, 23  
org::pixelgaffer::turnierserver::groker::ai::GrokerAi, 16  
enemyWonChips  
org::pixelgaffer::turnierserver::groker::Groker↔  
Update, 22  
GROKER\_CALLBACK  
grokerWrapper.h, 27  
GROKER\_MAIN  
grokerWrapper.h, 27  
game\_wrapper, 11  
game\_wrapper.GameWrapper, 15  
game\_wrapper::GameWrapper  
add\_output, 15  
process, 15

- update, 15
- gameFinished
  - org::pixelgaffer::turnierserver::groker::logic::↔  
GrokerLogic, 20
- getChanges
  - org::pixelgaffer::turnierserver::groker::Groker↔  
GameState, 19
- gewonneneChips
  - \_player, 13
  - org::pixelgaffer::turnierserver::groker::ai::AiDaten,  
15
- GrokerAi, 17
  - calc, 18
  - crash, 18
  - GrokerAi, 18
  - org::pixelgaffer::turnierserver::groker::ai::GrokerAi,  
16
  - out, 18
  - readOutput, 18
  - surrender, 18
- GrokerGameState
  - org::pixelgaffer::turnierserver::groker::Groker↔  
GameState, 19
- grokerMainLoop
  - grokerWrapper.c, 26
  - grokerWrapper.h, 28
- GrokerRenderData
  - org::pixelgaffer::turnierserver::groker::logic::↔  
GrokerRenderData, 21
- GrokerResponse
  - org::pixelgaffer::turnierserver::groker::Groker↔  
Response, 22
- grokerWrapper.c
  - grokerMainLoop, 26
  - parsePlayer, 26
- grokerWrapper.h
  - \_\_c\_crash, 28
  - \_\_c\_surrender, 28
  - GROKER\_CALLBACK, 27
  - GROKER\_MAIN, 27
  - grokerMainLoop, 28
  - parsePlayer, 28
  - Player, 28
  - Result, 28
  - Spieler, 28
- letzterEinsatz
  - \_player, 13
  - org::pixelgaffer::turnierserver::groker::ai::AiDaten,  
15
- lost
  - org::pixelgaffer::turnierserver::groker::logic::↔  
GrokerLogic, 20
- main
  - org::pixelgaffer::turnierserver::groker::Simple↔  
Player, 23
- mainpage.h, 25
- org, 11
- org.pixelgaffer, 11
- org.pixelgaffer.turnierserver, 11
- org.pixelgaffer.turnierserver.groker, 11
- org.pixelgaffer.turnierserver.groker.ai, 12
- org.pixelgaffer.turnierserver.groker.ai.AiDaten, 14
- org.pixelgaffer.turnierserver.groker.ai.GrokerAi, 15
- org.pixelgaffer.turnierserver.groker.GrokerGameState,  
19
- org.pixelgaffer.turnierserver.groker.GrokerResponse, 21
- org.pixelgaffer.turnierserver.groker.GrokerUpdate, 22
- org.pixelgaffer.turnierserver.groker.logic, 12
- org.pixelgaffer.turnierserver.groker.logic.GrokerAi↔  
Object, 18
- org.pixelgaffer.turnierserver.groker.logic.GrokerLogic,  
20
- org.pixelgaffer.turnierserver.groker.logic.Groker↔  
RenderData, 21
- org.pixelgaffer.turnierserver.groker.SimplePlayer, 22
- org::pixelgaffer::turnierserver::groker::GrokerGame↔  
State
  - applyChanges, 19
  - calculatePoints, 19
  - chips, 19
  - clearChanges, 19
  - getChanges, 19
  - GrokerGameState, 19
  - output, 20
  - wonChips, 20
- org::pixelgaffer::turnierserver::groker::GrokerResponse
  - chips, 22
  - GrokerResponse, 22
  - output, 22
- org::pixelgaffer::turnierserver::groker::GrokerUpdate
  - enemyWonChips, 22
  - ownWonChips, 22
  - toString, 22
- org::pixelgaffer::turnierserver::groker::SimplePlayer
  - einsatz, 23
  - main, 23
  - SimplePlayer, 23
- org::pixelgaffer::turnierserver::groker::ai::AiDaten
  - gewonneneChips, 15
  - letzterEinsatz, 15
- org::pixelgaffer::turnierserver::groker::ai::GrokerAi
  - einsatz, 16
  - GrokerAi, 16
  - update, 17
- org::pixelgaffer::turnierserver::groker::logic::GrokerLogic
  - aiTimeout, 20
  - createGameState, 20
  - createUserObject, 20
  - gameFinished, 20
  - lost, 20
  - parse, 20
  - sendFirstRenderData, 20
  - setup, 21
  - update, 21



org::pixelgaffer::turnierserver::groker::logic::Groker↔  
     RenderData  
     chips, 21  
     GrokerRenderData, 21  
     output, 21  
     wonChips, 21  
 out  
     GrokerAi, 18  
 output  
     \_result, 14  
     org::pixelgaffer::turnierserver::groker::Groker↔  
         GameState, 20  
     org::pixelgaffer::turnierserver::groker::Groker↔  
         Response, 22  
     org::pixelgaffer::turnierserver::groker::logic::↔  
         GrokerRenderData, 21  
 ownWonChips  
     org::pixelgaffer::turnierserver::groker::Groker↔  
         Update, 22  
  
 parse  
     org::pixelgaffer::turnierserver::groker::logic::↔  
         GrokerLogic, 20  
 parsePlayer  
     grokerWrapper.c, 26  
     grokerWrapper.h, 28  
 Player  
     grokerWrapper.h, 28  
 process  
     ai::AI, 14  
     game\_wrapper::GameWrapper, 15  
  
 readOutput  
     GrokerAi, 18  
 Result  
     grokerWrapper.h, 28  
  
 sendFirstRenderData  
     org::pixelgaffer::turnierserver::groker::logic::↔  
         GrokerLogic, 20  
 setup  
     org::pixelgaffer::turnierserver::groker::logic::↔  
         GrokerLogic, 21  
 SimplePlayer  
     org::pixelgaffer::turnierserver::groker::Simple↔  
         Player, 23  
 Spieler  
     grokerWrapper.h, 28  
 surrender  
     GrokerAi, 18  
  
 toString  
     org::pixelgaffer::turnierserver::groker::Groker↔  
         Update, 22  
  
 update  
     game\_wrapper::GameWrapper, 15  
     org::pixelgaffer::turnierserver::groker::ai::GrokerAi,  
         17

org::pixelgaffer::turnierserver::groker::logic::↔  
     GrokerLogic, 21  
  
 wonChips  
     org::pixelgaffer::turnierserver::groker::Groker↔  
         GameState, 20  
     org::pixelgaffer::turnierserver::groker::logic::↔  
         GrokerRenderData, 21