DistanzSpiel

Erzeugt von Doxygen 1.8.9.1

Son Jun 21 2015 22:49:06

Inhaltsverzeichnis

1	Hier	archie-\	/erzeichnis	1
	1.1	Klasse	nhierarchie	1
2	Klas	sen-Ve	rzeichnis	3
	2.1	Auflist	ung der Klassen	3
3	Date	ei-Verze	ichnis	5
	3.1		ung der Dateien	5
_		_		_
4	Klas		kumentation	7
	4.1	Feld K	lassenreferenz	7
		4.1.1	Ausführliche Beschreibung	9
		4.1.2	Beschreibung der Konstruktoren und Destruktoren	9
			4.1.2.1 Feld	9
			4.1.2.2 Feld	9
			4.1.2.3 Feld	9
		4.1.3	Dokumentation der Elementfunktionen	9
			4.1.3.1 delStein	9
			4.1.3.2 getBesetzt	9
			4.1.3.3 getGast	9
			4.1.3.4 getPos	10
			4.1.3.5 setStein	10
		4.1.4	Dokumentation der Datenelemente	10
			4.1.4.1 besetzt	10
			4.1.4.2 gast	10
			4.1.4.3 pos	10
	4.2	GUI KI	assenreferenz	10
		4.2.1	Ausführliche Beschreibung	12
		4.2.2	Beschreibung der Konstruktoren und Destruktoren	12
			4.2.2.1 GUI	12
		4.2.3	Dokumentation der Elementfunktionen	12
			4.2.3.1 Spieler	12

iv INHALTSVERZEICHNIS

		4.2.3.2	zeichneAnleitung	12
		4.2.3.3	zeichneSpielfeld	12
		4.2.3.4	zeichneZug	12
	4.2.4	Dokumen	tation der Datenelemente	12
		4.2.4.1	brett	12
		4.2.4.2	Klsw	13
4.3	KI Klas	senreferen	nz	13
	4.3.1	Ausführlic	che Beschreibung	15
	4.3.2	Beschreib	oung der Konstruktoren und Destruktoren	15
		4.3.2.1	KI	15
		4.3.2.2	$\sim\!\!KI\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$	15
	4.3.3	Dokumen	tation der Elementfunktionen	15
		4.3.3.1	getBrett	15
		4.3.3.2	getTeam	15
		4.3.3.3	mergeStrategie	16
		4.3.3.4	mergeStrategie	17
		4.3.3.5	nexZug	18
		4.3.3.6	seachBestZug	18
	4.3.4	Dokumen	tation der Datenelemente	19
		4.3.4.1	abrett	19
		4.3.4.2	anzstrat	19
		4.3.4.3	$dv \dots $	19
		4.3.4.4	nZug	19
		4.3.4.5	strat	19
		4.3.4.6	$t \ \dots $	19
4.4	Koenig	Klassenre	ferenz	20
	4.4.1	Beschreib	oung der Konstruktoren und Destruktoren	22
		4.4.1.1	Koenig	22
		4.4.1.2	Koenig	22
	4.4.2	Dokumen	tation der Elementfunktionen	22
		4.4.2.1	setGeffangen	22
		4.4.2.2	ziehenach	22
4.5	Possition	on Struktur	referenz	23
	4.5.1	Ausführlic	che Beschreibung	24
	4.5.2	Beschreib	oung der Konstruktoren und Destruktoren	24
		4.5.2.1	Possition	24
		4.5.2.2	Possition	24
	4.5.3	Dokumen	station der Elementfunktionen	24
		4.5.3.1	operator==	24
	4.5.4	Dokumen	station der Datenelemente	24

INHALTSVERZEICHNIS

		4.5.4.1 x	24
		4.5.4.2 y	24
4.6	SfH Kla	ssenreferenz	24
	4.6.1	Ausführliche Beschreibung	27
	4.6.2	Beschreibung der Konstruktoren und Destruktoren	27
		4.6.2.1 SfH	27
		4.6.2.2 ~SfH	27
	4.6.3	Dokumentation der Elementfunktionen	27
		4.6.3.1 bewerten	27
4.7	SfK Kla	ssenreferenz	28
	4.7.1	Ausführliche Beschreibung	31
	4.7.2	Beschreibung der Konstruktoren und Destruktoren	31
		4.7.2.1 SfK	31
		4.7.2.2 SfK	31
		4.7.2.3 ~SfK	31
	4.7.3	Dokumentation der Elementfunktionen	31
		4.7.3.1 bewerten	31
4.8	SpielBr	ett Klassenreferenz	33
	4.8.1	Beschreibung der Konstruktoren und Destruktoren	34
		4.8.1.1 SpielBrett	34
		4.8.1.2 SpielBrett	34
		4.8.1.3 ~SpielBrett	35
	4.8.2	Dokumentation der Elementfunktionen	35
		4.8.2.1 getFeld	35
		4.8.2.2 getSchwarz	35
		4.8.2.3 getWeis	36
		4.8.2.4 initBrett	36
	4.8.3	Dokumentation der Datenelemente	36
		4.8.3.1 Brett	36
		4.8.3.2 dimmension	36
		4.8.3.3 schwarz	36
		4.8.3.4 weis	36
4.9	SrH Kla	ssenreferenz	36
	4.9.1	Ausführliche Beschreibung	39
	4.9.2	Beschreibung der Konstruktoren und Destruktoren	39
		4.9.2.1 SrH	39
		4.9.2.2 ~SrH	39
	4.9.3	Dokumentation der Elementfunktionen	39
		4.9.3.1 bewerten	39
4.10	SsK Kla	ssenreferenz	40

vi INHALTSVERZEICHNIS

	4.10.1	Ausführliche Beschreibung	42
	4.10.2	Beschreibung der Konstruktoren und Destruktoren	42
		4.10.2.1 SsK	42
		4.10.2.2 SsK	42
		4.10.2.3 ∼SsK	42
	4.10.3	Dokumentation der Elementfunktionen	42
		4.10.3.1 bewerten	42
		4.10.3.2 posSicher	43
	4.10.4	Dokumentation der Datenelemente	43
		4.10.4.1 gegner	43
		4.10.4.2 gZuege	43
4.11	Stein K	lassenreferenz	43
	4.11.1	Ausführliche Beschreibung	46
	4.11.2	Beschreibung der Konstruktoren und Destruktoren	46
		4.11.2.1 Stein	46
		4.11.2.2 Stein	46
		4.11.2.3 ~Stein	47
	4.11.3	Dokumentation der Elementfunktionen	47
		4.11.3.1 getGeffangen	47
		4.11.3.2 getid	47
		4.11.3.3 getMteam	47
		4.11.3.4 getOrt	47
		4.11.3.5 setFrei	47
		4.11.3.6 setGeffangen	47
		4.11.3.7 setOrt	47
		4.11.3.8 ziehenach	47
		4.11.3.9 zuege	48
	4.11.4	Dokumentation der Datenelemente	48
		4.11.4.1 geffangen	48
		4.11.4.2 id	48
		4.11.4.3 mteam	49
		4.11.4.4 ort	49
4.12	Strateg	ie Klassenreferenz	49
	4.12.1	Ausführliche Beschreibung	51
	4.12.2	Beschreibung der Konstruktoren und Destruktoren	51
		4.12.2.1 Strategie	51
		4.12.2.2 Strategie	51
		4.12.2.3 ~Strategie	51
	4.12.3	Dokumentation der Elementfunktionen	51
		4.12.3.1 bewerten	51

INHALTSVERZEICHNIS vii

		4.12.3.2	getmZuege	52
		4.12.3.3	getWert	52
		4.12.3.4	getZuege	52
		4.12.3.5	nexZug	52
4	1.12.4	Dokumen	ntation der Datenelemente	52
		4.12.4.1	aZuege	52
		4.12.4.2	brett	52
		4.12.4.3	$h1 \ \ldots \ldots \ldots \ldots \ldots$	52
		4.12.4.4	h2	52
		4.12.4.5	h3	52
		4.12.4.6	$k \ \ldots \ldots \ldots \ldots \ldots \ldots$	52
		4.12.4.7	mZuege	52
		4.12.4.8	nZug	52
		4.12.4.9	team	52
		4.12.4.10	wert	53
4.13 T	Геат К	lassenrefe	erenz	53
4	1.13.1	Ausführlic	che Beschreibung	55
4	1.13.2	Beschreit	oung der Konstruktoren und Destruktoren	55
		4.13.2.1	Team	55
		4.13.2.2	Team	55
		4.13.2.3	\sim Team	55
4	1.13.3	Dokumen	ntation der Elementfunktionen	55
		4.13.3.1	distanzen	55
		4.13.3.2	getBrett	56
		4.13.3.3	getFarbe	56
		4.13.3.4	getGegner	56
		4.13.3.5	getSieg	56
		4.13.3.6	getStein	56
		4.13.3.7	setGegner	56
		4.13.3.8	setSieg	56
4	1.13.4	Dokumen	station der Datenelemente	56
		4.13.4.1	brett	56
		4.13.4.2	Farbe	57
		4.13.4.3	gegner	57
		4.13.4.4	helfer1	57
		4.13.4.5	helfer2	57
		4.13.4.6	helfer3	57
		4.13.4.7	koenig	57
		4.13.4.8	Sieg	57
4.14 L	Jser Kl	assenrefe	renz	57

viii INHALTSVERZEICHNIS

		4.14.1	Ausführliche Beschreibung	57
		4.14.2	Beschreibung der Konstruktoren und Destruktoren	57
			4.14.2.1 User	58
			4.14.2.2 ~User	58
		4.14.3	Dokumentation der Elementfunktionen	58
			4.14.3.1 Graphik	58
	4.15	zug Str	ukturreferenz	58
		4.15.1	Ausführliche Beschreibung	60
		4.15.2	Beschreibung der Konstruktoren und Destruktoren	60
			4.15.2.1 zug	60
			4.15.2.2 zug	60
			4.15.2.3 zug	60
		4.15.3	Dokumentation der Elementfunktionen	60
			4.15.3.1 operator<	60
			4.15.3.2 operator=	60
			4.15.3.3 operator==	60
		4.15.4	Dokumentation der Datenelemente	60
			4.15.4.1 stein	60
			4.15.4.2 wert	60
			4.15.4.3 zpos	60
			4.15.4.4 zu	61
5	Date	i-Dokur	nentation	63
Ĭ	5.1			63
	0	5.1.1		63
		0.1.1		63
	5.2	Feld h-		64
	5.3			64
	5.4			65
	5.5			67
	5.6			68
	5.7			70
	0.7	5.7.1		71
		0.7.1		71
	5.8	Koenia		71
	0.0	5.8.1		72
				72
	5.9	main.c		72
		5.9.1	•	73
			5.9.1.1 main	74

INHALTSVERZEICHNIS ix

5.10	Main.h-Dateireferenz	74
5.11	Possition.h-Dateireferenz	74
	5.11.1 Makro-Dokumentation	74
	5.11.1.1 POSSITION_H	74
5.12	SfH.cpp-Dateireferenz	75
5.13	SfH.h-Dateireferenz	76
5.14	SfK.cpp-Dateireferenz	77
5.15	SfK.h-Dateireferenz	78
5.16	SpielBrett.cpp-Dateireferenz	80
	5.16.1 Makro-Dokumentation	81
	5.16.1.1 SPIELBRETT_C	81
5.17	SpielBrett.h-Dateireferenz	81
5.18	SrH.cpp-Dateireferenz	82
5.19	SrH.h-Dateireferenz	83
5.20	SsK.cpp-Dateireferenz	85
5.21	SsK.h-Dateireferenz	85
5.22	Stein.cpp-Dateireferenz	87
	5.22.1 Makro-Dokumentation	88
	5.22.1.1 STEIN_C	88
5.23	Stein.h-Dateireferenz	88
5.24	Strategie.cpp-Dateireferenz	89
5.25	Strategie.h-Dateireferenz	90
5.26	Team.cpp-Dateireferenz	92
	5.26.1 Makro-Dokumentation	92
	5.26.1.1 TEAM_C	92
5.27	Team.h-Dateireferenz	92
5.28	User.cpp-Dateireferenz	94
	5.28.1 Makro-Dokumentation	94
	5.28.1.1 USER_C	94
5.29	User.h-Dateireferenz	94
5.30	zug.h-Dateireferenz	95
Index		97

Kapitel 1

Hierarchie-Verzeichnis

1.1 Klassenhierarchie

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

Feld	
GUI	
KI	13
Possition	
SpielBrett	
Stein	
Koenig	20
Strategie	49
SfH	24
SfK	
SrH	
SsK	
Team	53
User	57
zug	58

2 Hierarchie-Verzeichnis

Kapitel 2

Klassen-Verzeichnis

2.1 Auflistung der Klassen

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

Feld	
GUI	
KI	
Koenig	
Possition	
SfH	
SfK	
SpielBrett	
SrH	36
SsK	
Stein	
Strategie	
Team	
User	57
7110	58

4 Klassen-Verzeichnis

Kapitel 3

Datei-Verzeichnis

3.1 Auflistung der Dateien

Hier folgt die Aufzählung aller Dateien mit einer Kurzbeschreibung:

Feld.cpp	63
Feld.h	64
GUI.cpp	64
GUI.h	65
Kl.cpp	67
Kl.h	68
Koenig.cpp	70
Koenig.h	71
main.cpp	72
Main.h	74
Possition.h	74
SfH.cpp	75
SfH.h	76
SfK.cpp	77
SfK.h	78
SpielBrett.cpp	80
SpielBrett.h	81
SrH.cpp	82
SrH.h	83
SsK.cpp	85
SsK.h	85
Stein.cpp	87
Stein.h	88
Strategie.cpp	89
Strategie.h	90
Team.cpp	92
Team.h	92
User.cpp	94
User.h	94
zua h	95

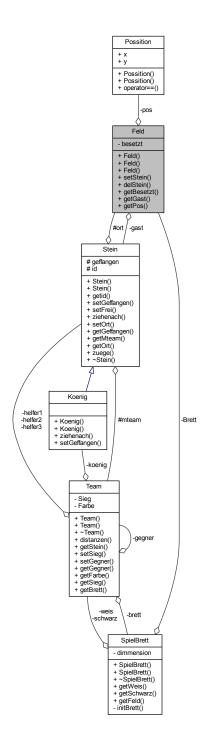
6 Datei-Verzeichnis

Kapitel 4

Klassen-Dokumentation

4.1 Feld Klassenreferenz

Zusammengehörigkeiten von Feld:



Öffentliche Methoden

- Feld ()
- Feld (short nx, short ny)
- Feld (Feld &f)
- void setStein (Stein *newstein)
- void delStein ()

4.1 Feld Klassenreferenz 9

- bool getBesetzt ()
- Stein * getGast ()
- Possition getPos ()

Private Attribute

- · bool besetzt
- · Possition pos
- Stein * gast =nullptr

4.1.1 Ausführliche Beschreibung

class Feld Diese Klasse Symbolisiert ein Feld auf einem Spielbrett.

4.1.2 Beschreibung der Konstruktoren und Destruktoren

```
4.1.2.1 Feld::Feld()
```

4.1.2.2 Feld::Feld (short nx, short ny)

Feld Konstruktor

Parameter

in	nx	x Koordinaten des Feldes
in	ny	y Koordinaten des Feldes

```
4.1.2.3 Feld::Feld ( Feld & f )
```

4.1.3 Dokumentation der Elementfunktionen

```
4.1.3.1 void Feld::delStein ( )
```

delStein Loescht Zeiger auf Gast Setzt besetzt auf false

4.1.3.2 bool Feld::getBesetzt ()

Get the value of besetzt.

Rückgabe

the value of besetzt.

4.1.3.3 Stein * Feld::getGast ()

getGast

Rückgabe

Gibt einen Pointer auf den Gast zurueck.

4.1.3.4 Possition Feld::getPos()

getPos

Rückgabe

the value of pos.

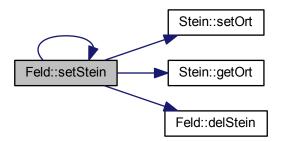
4.1.3.5 void Feld::setStein (Stein * newstein)

Setzt Stein auf das Feld und Markiert das Feld als Besetzt. Falls das Feld besetzt ist, werden die Gaeste/Steine getauscht.

Parameter

[in/out] *newstein pointer auf den zu setzenden Stein.

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



4.1.4 Dokumentation der Datenelemente

- 4.1.4.1 bool Feld::besetzt [private]
- **4.1.4.2 Stein*** Feld::gast =nullptr [private]
- **4.1.4.3 Possition Feld::pos** [private]

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

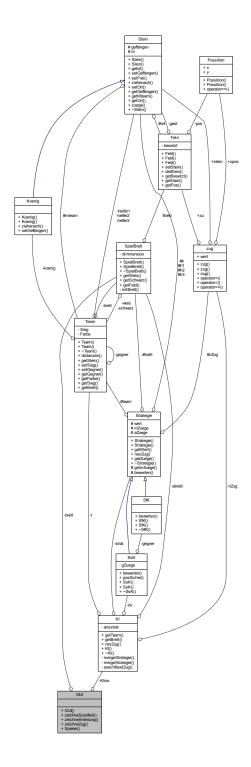
- Feld.h
- Feld.cpp

4.2 GUI Klassenreferenz

#include <GUI.h>

4.2 GUI Klassenreferenz 11

Zusammengehörigkeiten von GUI:



Öffentliche Methoden

- GUI (SpielBrett *br, KI *ki)
- void zeichneSpielfeld (int zug, int spieler)
- void zeichneAnleitung ()
- void zeichneZug (int zug, int spieler, int zeile, int spalte)
- void Spieler (bool farbe, int zug, int spieler)

Private Attribute

- SpielBrett * brett
- KI * KIsw

4.2.1 Ausführliche Beschreibung

class GUI Die Abkürzung GUI steht für graphical user interface bzw. grafische Benutzeroberfläche. Durch diese Klasse wird dem Benutzer eine grafische Oberfläche zur Verfügung gestellt, über die er mit dem Programm interagierien kann. Alle Benutzereingaben erfolgen ausschließlich über die Tastatur. Unterstützend wird die Strucktur der erwarteten Eingabe in Klammern mit angegeben. Sollte dennoch der Benutzer eine Falsch Eingabe tätigen, so wird er darauf hingewiesen und kann seine Eingabe nach 3 Sekunken wiederholen. Die Darstellung des Spielfeldes und wichtiger Spielparameter erfolgt in der Windows Konsole über ANSI-Zeichen.

4.2.2 Beschreibung der Konstruktoren und Destruktoren

4.2.2.1 GUI::GUI (SpielBrett * br, KI * ki)

4.2.3 Dokumentation der Elementfunktionen

4.2.3.1 void GUI::Spieler (bool farbe, int zug, int spieler)

zeichne Zug(zug, spieler, zeile, spalte) Mit dieser Funktion wird dem Spieler alle zulässigen Züge des ausgewählten Steins angezeigt.

Parameter

[int]	zug gibt den aktuellen Zug an
[int]	spieler gibt an, welcher Spieler gerade am Zug ist
[int]	zeile Zeile des ausgewählten Steins
[int]	spalte Spalte des ausgewählten Steins

4.2.3.2 void GUI::zeichneAnleitung ()

zeichneSpielfeld(zug,spieler) Mit dieser Funktion wird das Spielfeld grafisch für den Spieler aufbereitet.

Parameter

[int]	zug gibt den aktuelle Zug an
[int]	spieler gibt an, welcher Spieler an Zug ist

4.2.3.3 void GUI::zeichneSpielfeld (int zug, int spieler)

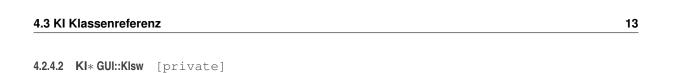
GUI Diese Funktion ist ein Konstructor für eine Instanz von der Klasse Spielbrett.

4.2.3.4 void GUI::zeichneZug (int zug, int spieler, int zeile, int spalte)

zeichenAnleitung() Diese Funktion gibt dem Benutzer Auskunft über die Spielregeln.

4.2.4 Dokumentation der Datenelemente

4.2.4.1 SpielBrett* **GUI::brett** [private]



Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

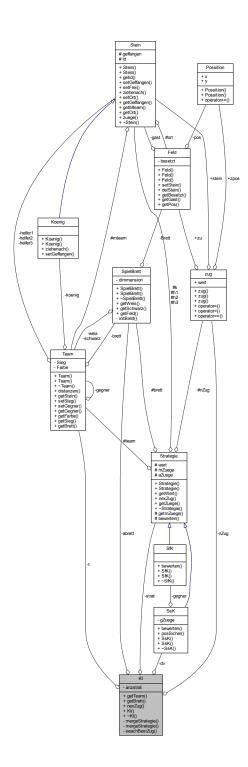
• GUI.h

• GUI.cpp

4.3 KI Klassenreferenz

#include <KI.h>

Zusammengehörigkeiten von KI:



Öffentliche Methoden

- Team & getTeam ()
- SpielBrett & getBrett ()
- void nexZug ()
- KI (Team &t)
- virtual ∼KI ()

4.3 KI Klassenreferenz 15

Private Methoden

- std::vector< zug > mergeStrategie (Strategie *st1, std::vector< zug > st2Zuege)
- std::vector< zug > mergeStrategie (Strategie *st1, Strategie *st2)
- void seachBestZug ()

Private Attribute

- Team & t
- · SpielBrett & abrett
- Strategie * strat [anzstrat]
- SsK * dv
- zug nZug

Statische, private Attribute

• static const int anzstrat =4

4.3.1 Ausführliche Beschreibung

class KI Ist eine Klasse die aus den möglichen Spielzügen den besten auswählt. Sie ist mit zusätzlichen Strategien erweiterbar.

4.3.2 Beschreibung der Konstruktoren und Destruktoren

```
4.3.2.1 KI::KI ( Team & t )
```

KI Konstruktor

Parameter

ſ	in out	t	Referenz auf das Team, das gesteuert werden soll
ı	III, Out	ι	Helerenz auf das feath, das gestedert werden son

```
4.3.2.2 KI::∼KI( ) [virtual]
```

4.3.3 Dokumentation der Elementfunktionen

4.3.3.1 SpielBrett & KI::getBrett ()

getBrett

Rückgabe

Referenz auf das Spielbrett

4.3.3.2 Team & KI::getTeam ()

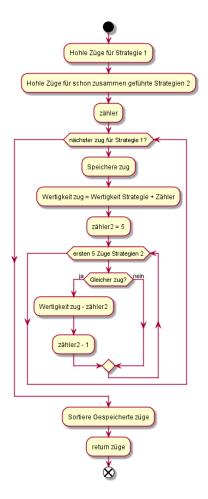
getTeam

Rückgabe

Referenz auf das gesteuerte Team

4.3.3.3 std::vector< zug > Kl::mergeStrategie (Strategie * st1, std::vector< zug > st2Zuege) [private]

mergeStrategie Vereint zwei Strategien und führt die Wertigkeiten zusammen. Je kleiner die Wertigkeits-Zahl desto besser ist der zug.



Parameter

in	st1	Pointer auf eine Strategie
in	st2Zuege	Vector mit Zügen.

Rückgabe

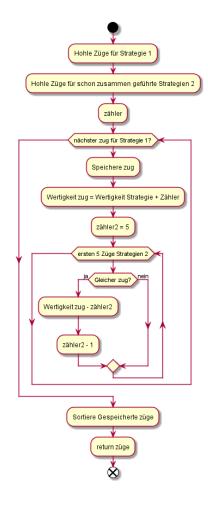
Vector mit nach wertigkeit sortierten zügen.

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



4.3 KI Klassenreferenz

- 4.3.3.4 std::vector < zug > KI::mergeStrategie (Strategie * st1, Strategie * st2) [private]
 - mergeStrategie Vereint zwei Strategien und fürt die Wertigkeiten zusammen. Je kleiner die wertigkeits Zahl desto besser ist der Zug.



Parameter

in	st1	Pointer auf eine Strategie.
in	st2	Pointer auf eine Strategie.

Rückgabe

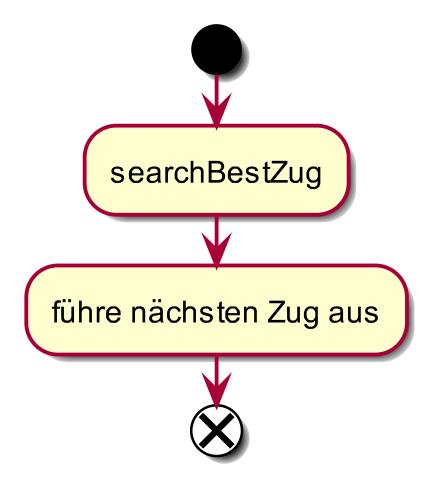
Vector mit nach Wertigkeit sortierten Zügen.

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:

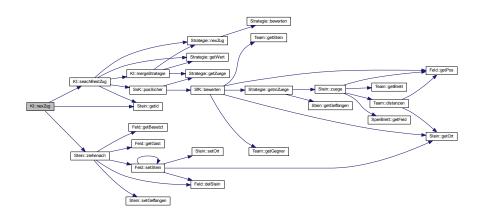


4.3.3.5 void KI::nexZug ()

nexZug Führt den nächsten Zug aus



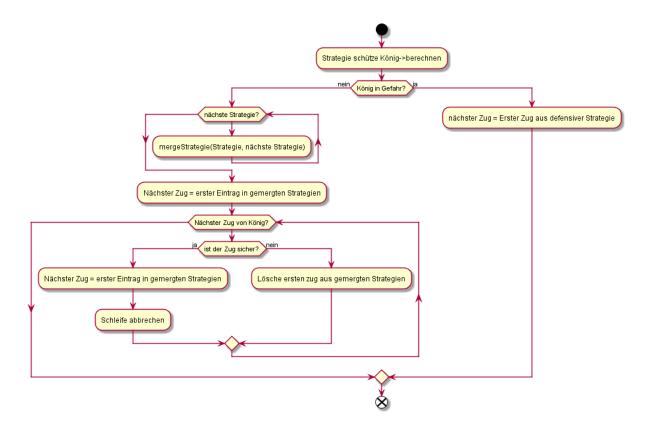
Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



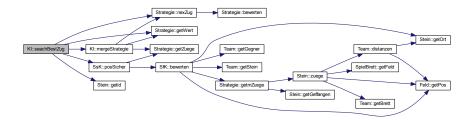
4.3.3.6 void KI::seachBestZug() [private]

seachBestZug Wählt aus den zusammengefürten Strategien den besten Zug aus. Und stelt sicher das der König nicht in Gefahr ist bzw. kommt.

4.3 KI Klassenreferenz 19



Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



- 4.3.4 Dokumentation der Datenelemente
- 4.3.4.1 SpielBrett& Kl::abrett [private]
- **4.3.4.2 const int KI::anzstrat =4** [static], [private]
- **4.3.4.3 SsK*** **Kl**::**dv** [private]
- 4.3.4.4 zug Kl::nZug [private]
- **4.3.4.5 Strategie*** Kl::strat[anzstrat] [private]
- 4.3.4.6 Team& KI::t [private]

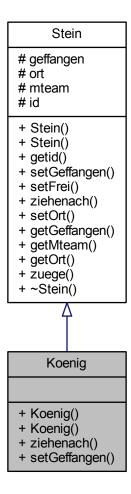
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- Kl.h
- Kl.cpp

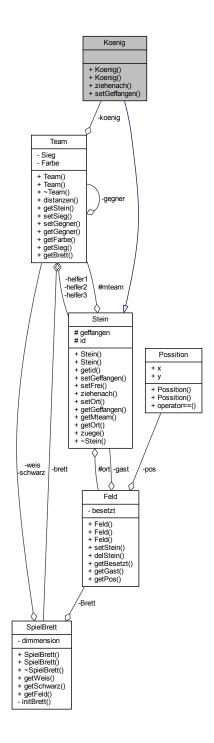
4.4 Koenig Klassenreferenz

#include <Koenig.h>

Klassendiagramm für Koenig:



Zusammengehörigkeiten von Koenig:



Öffentliche Methoden

- · Koenig ()
- Koenig (int id, Feld *startplatz, Team *mt)
- virtual bool ziehenach (Feld *ziehe) override
- virtual void setGeffangen () override

Weitere Geerbte Elemente

4.4.1 Beschreibung der Konstruktoren und Destruktoren

```
4.4.1.1 Koenig::Koenig ( )
```

```
4.4.1.2 Koenig::Koenig ( int id, Feld * startplatz, Team * mt )
```

4.4.2 Dokumentation der Elementfunktionen

```
4.4.2.1 void Koenig::setGeffangen() [override], [virtual]
```

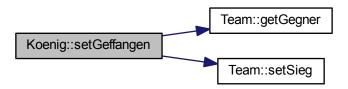
ziehenach(Feld) Mit dieser Methode hat der König die Möglichkeit seine Helfer zu befreien und sich auf dem Spielfeld zu bewegen.

Parameter

```
[Feld] *ziehe hierbei handelt sich um einen Pointer auf das Feld, das er springen soll
```

Erneute Implementation von Stein.

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:

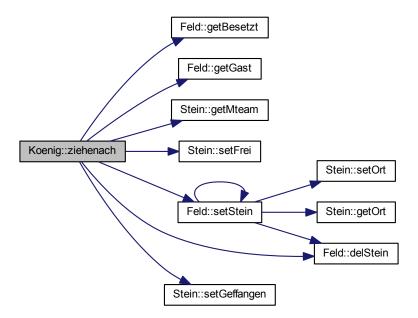


```
4.4.2.2 bool Koenig::ziehenach ( Feld * ziehe ) [override], [virtual]
```

Implementiert den Startplatz des Königs.

Erneute Implementation von Stein.

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



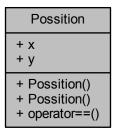
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- · Koenig.h
- Koenig.cpp

4.5 Possition Strukturreferenz

#include <Possition.h>

Zusammengehörigkeiten von Possition:



Öffentliche Methoden

- Possition (short int x, short int y)
- Possition ()
- bool operator== (const Possition &p) const

Öffentliche Attribute

- short int x
- short int y

4.5.1 Ausführliche Beschreibung

struct Position

4.5.2 Beschreibung der Konstruktoren und Destruktoren

```
4.5.2.1 Possition::Possition ( short int x, short int y ) [inline]
```

- 4.5.2.2 Possition::Possition() [inline]
- 4.5.3 Dokumentation der Elementfunktionen
- 4.5.3.1 bool Possition::operator== (const Possition & p) const [inline]
- 4.5.4 Dokumentation der Datenelemente
- 4.5.4.1 short int Possition::x
- 4.5.4.2 short int Possition::y

Die Dokumentation für diese Struktur wurde erzeugt aufgrund der Datei:

· Possition.h

4.6 SfH Klassenreferenz

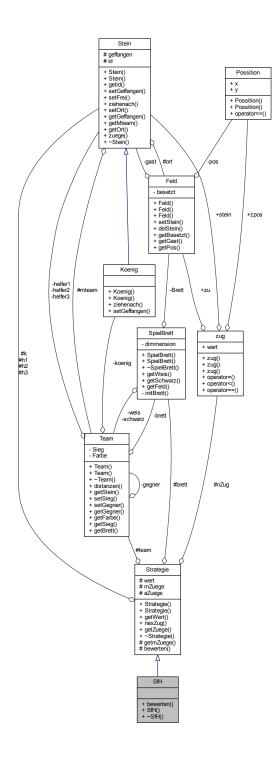
#include <SfH.h>

4.6 SfH Klassenreferenz 25

Klassendiagramm für SfH:



Zusammengehörigkeiten von SfH:



Öffentliche Methoden

- virtual void bewerten () override
- SfH (Team &team, SpielBrett &b)
- virtual ∼SfH ()

4.6 SfH Klassenreferenz 27

Weitere Geerbte Elemente

4.6.1 Ausführliche Beschreibung

class SfH (Strategie fange Helfer) implementiert die Methode bewerten();

Diese Strategie sorgt dafür, dass die gegnerischen Helfer festgesetzt/gefangen werden. Ein festgesetzter/gefangener Helfer stellt insofern keine Bedrohung mehr dar, bis er wieder vom Koenig befreit wird. Dies gilt es durch andere Strategien zu verhindern.

Parameter

&team	Referenz auf Instanz von Team
&b	Referenz auf Instanz von SpielBrett

4.6.2 Beschreibung der Konstruktoren und Destruktoren

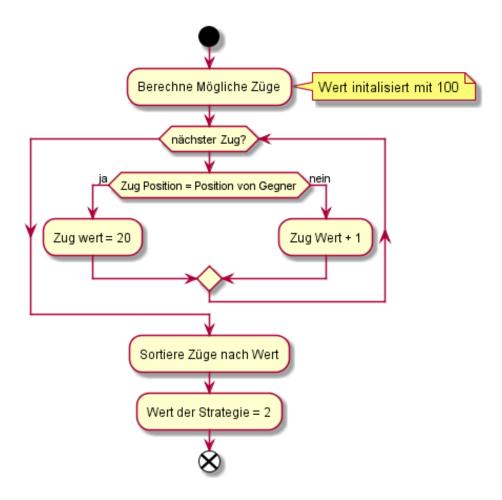
```
4.6.2.1 SfH::SfH ( Team & team, SpielBrett & b )
```

```
4.6.2.2 SfH::∼SfH() [virtual]
```

4.6.3 Dokumentation der Elementfunktionen

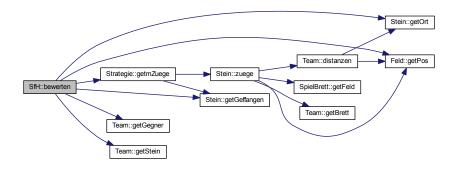
```
4.6.3.1 void SfH::bewerten() [override], [virtual]
```

bewerten() Bewertet mögliche Züge nach der Möglichkeit gegnerische Helfer zu fangen.



Implementiert Strategie.

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- SfH.h
- SfH.cpp

4.7 SfK Klassenreferenz

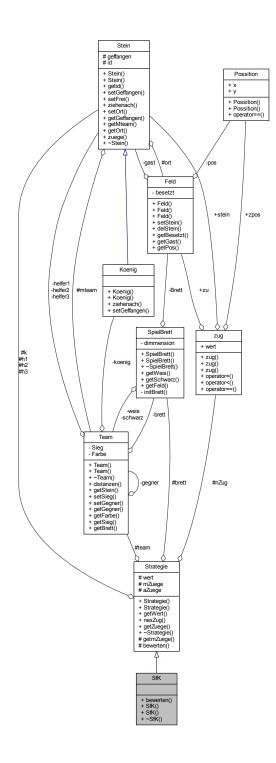
#include <SfK.h>

4.7 SfK Klassenreferenz 29

Klassendiagramm für SfK:



Zusammengehörigkeiten von SfK:



- virtual void bewerten () override
- SfK (Team &team, SpielBrett &b)
- SfK ()
- virtual ∼SfK ()

4.7 SfK Klassenreferenz 31

Weitere Geerbte Elemente

4.7.1 Ausführliche Beschreibung

class SfK (Strategie fange König) Ist eine Ableitung der abstrakten Klasse Strategie.

Diese Strategie sorgt dafür, dass sich die Spielfiguren dem gegnerischen König nähern, um in festsetzen/gefangen nehmen zu können. Ein festgesetzter/gefangener König bedeutet das Spielende. Ein Sieg wird erzielt, sobald der gegnerische König festgesetzt/gefangen ist.

Überschreibt/implementiert die Methode bewerten();

Parameter

&team	Referenz auf Instanz von Team
&b	Referenz auf Instanz von SpielBrett

4.7.2 Beschreibung der Konstruktoren und Destruktoren

```
4.7.2.1 SfK::SfK ( Team & team, SpielBrett & b )
```

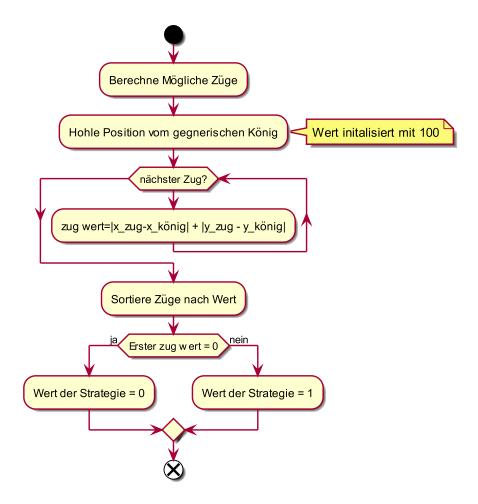
```
4.7.2.2 SfK::SfK()
```

```
4.7.2.3 SfK::∼SfK() [virtual]
```

4.7.3 Dokumentation der Elementfunktionen

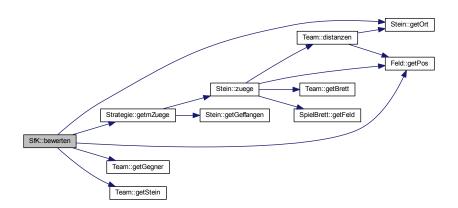
```
4.7.3.1 void SfK::bewerten() [override], [virtual]
```

bewerten() Bewertet mögliche Züge nach der Möglichkeit gegnerischen König zu fangen.



Implementiert Strategie.

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



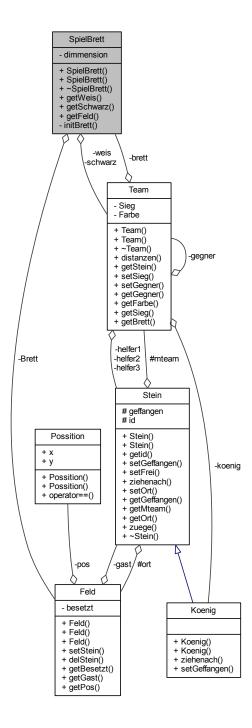
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- SfK.h
- · SfK.cpp

4.8 SpielBrett Klassenreferenz

#include <SpielBrett.h>

Zusammengehörigkeiten von SpielBrett:



- SpielBrett ()
- SpielBrett (const SpielBrett &sb)

- ∼SpielBrett ()
- Team * getWeis () const
- Team * getSchwarz () const
- Feld * getFeld (int x, int y) const

Private Methoden

• void initBrett ()

Private Attribute

- Feld ** Brett =nullptr
- Team * schwarz =nullptr
- Team * weis =nullptr

Statische, private Attribute

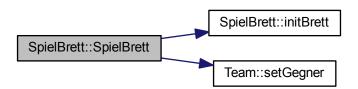
• static const short int dimmension = 8

4.8.1 Beschreibung der Konstruktoren und Destruktoren

4.8.1.1 SpielBrett::SpielBrett ()

initBrett() Erzeugt das 8x8 großes Feld.

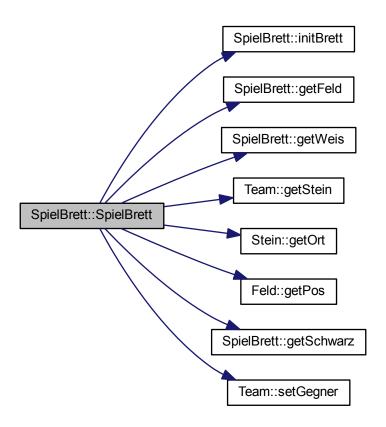
Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



4.8.1.2 SpielBrett::SpielBrett (const SpielBrett & sb)

SpielBrett() Beinhaltet die Startaufstellung.

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



4.8.1.3 SpielBrett:: \sim SpielBrett ()

SpielBrett (const SpielBrett &sb) Verweißt auf die Pointer, der einzelnen Spielsteine.

4.8.2 Dokumentation der Elementfunktionen

4.8.2.1 Feld * SpielBrett::getFeld (int x, int y) const

getSchwarz() Kennzeichnet die schwarzen Steine.

Rückgabe

schwarz

 $\textbf{4.8.2.2} \quad \textbf{Team} * \textbf{SpielBrett::getSchwarz} \, (\quad) \, \textbf{const}$

getWeis() Kennzeichnet die weißen Steine.

Rückgabe

weiß

```
4.8.2.4 void SpielBrett::initBrett() [private]

4.8.3.1 Folda: SpielBrett::Brett = nullntr [private]
```

4.8.3.1 Feld SpielBrett::Brett =nullptr** [private]

4.8.3.2 const short int SpielBrett::dimmension = 8 [static], [private]

4.8.3.3 Team* SpielBrett::schwarz =nullptr [private]

4.8.3.4 Team * SpielBrett::weis =nullptr [private]

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- SpielBrett.h
- SpielBrett.cpp

4.9 SrH Klassenreferenz

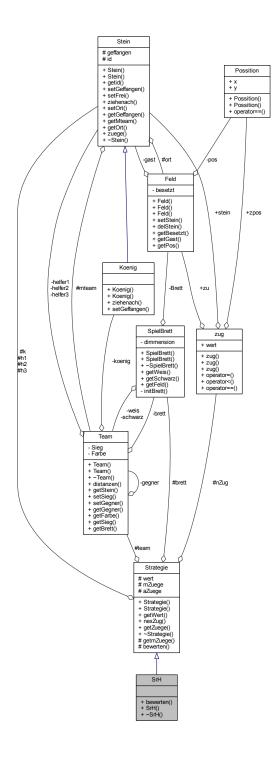
#include <SrH.h>

4.9 SrH Klassenreferenz 37

Klassendiagramm für SrH:



Zusammengehörigkeiten von SrH:



- virtual void bewerten () override
- SrH (Team &team, SpielBrett &b)
- virtual ∼SrH ()

4.9 SrH Klassenreferenz 39

Weitere Geerbte Elemente

4.9.1 Ausführliche Beschreibung

class SrH (Strategie rette Helfer) Ist eine Ableitung der abstrakten Klasse Strategie.

Diese Strategie sorgt dafuer, dass der Koenig teameigene festgesetzte/gefangene Helfer befreit. Dies tut er allerdings nach Moeglichkeit erst dann, wenn sie sich auch in unmittelbarer Umgebung befinden, da der Koenig selber eine sehr defensive Rolle im Spielverlauf einnimmt.

Ueberschreibt/implementiert die Methode bewerten();

Parameter

&team	Referenz auf Instanz von Team
&b	Referenz auf Instanz von SpielBrett

4.9.2 Beschreibung der Konstruktoren und Destruktoren

4.9.2.1 SrH::SrH (Team & team, SpielBrett & b)

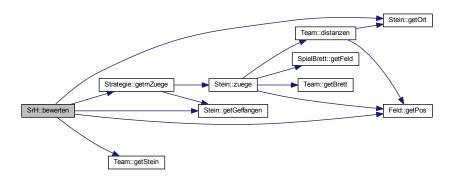
4.9.2.2 SrH::∼SrH() [virtual]

4.9.3 Dokumentation der Elementfunktionen

4.9.3.1 void SrH::bewerten() [override], [virtual]

Implementiert Strategie.

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- SrH.h
- SrH.cpp

4.10 SsK Klassenreferenz

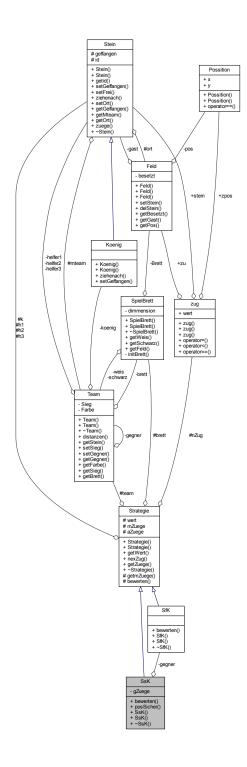
#include <SsK.h>

Klassendiagramm für SsK:



4.10 SsK Klassenreferenz 41

Zusammengehörigkeiten von SsK:



- virtual void bewerten () override
- bool posSicher (Possition p)
- SsK (Team &team, SpielBrett &b)
- SsK ()
- virtual ∼SsK ()

Private Attribute

- · SfK gegner
- std::vector < zug > gZuege

Weitere Geerbte Elemente

4.10.1 Ausführliche Beschreibung

class SsK (Strategie schuetze Koenig) Ist eine Ableitung der abstrakten Klasse Strategie.

Diese Strategie sorgt dafuer, dass der teameigene Koenig vor festsetzen/gefangen nehmen durch feindliche Spielfiguren geschuetzt wird. Zu beobachten ist hierbei das fangen von gegnerischen Spielfiguren, sobald sie dem König zu nahe kommen. Auch der Koenig selber nimmt ein sehr defensives Verhalten an und haelt sich von den Gegnern fern, um ein fruehzeitiges Ableben zu verhindern.

Ueberschreibt/implementiert die Methode bewerten();

Parameter

&team	Referenz auf Instanz von Team
&b	Referenz auf Instanz von SpielBrett

4.10.2 Beschreibung der Konstruktoren und Destruktoren

```
4.10.2.1 SsK::SsK ( Team & team, SpielBrett & b )
```

4.10.2.2 SsK::SsK()

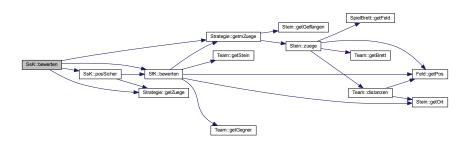
4.10.2.3 SsK::∼SsK() [virtual]

4.10.3 Dokumentation der Elementfunktionen

```
4.10.3.1 void SsK::bewerten() [override], [virtual]
```

Implementiert Strategie.

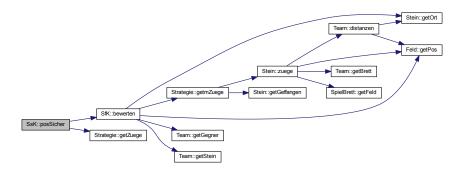
Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



4.11 Stein Klassenreferenz 43

4.10.3.2 bool SsK::posSicher (Possition p)

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



4.10.4 Dokumentation der Datenelemente

4.10.4.1 SfK SsK::gegner [private]

4.10.4.2 std::vector<zug> SsK::gZuege [private]

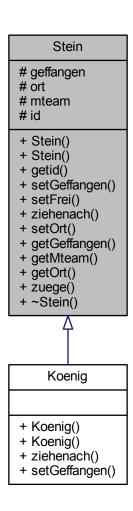
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- SsK.h
- SsK.cpp

4.11 Stein Klassenreferenz

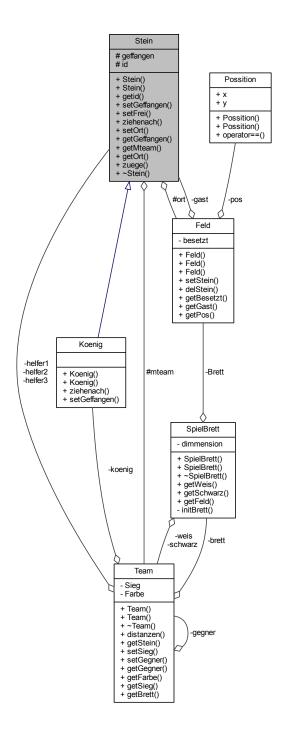
#include <Stein.h>

Klassendiagramm für Stein:



4.11 Stein Klassenreferenz 45

Zusammengehörigkeiten von Stein:



- Stein ()
- Stein (int id, Feld *startplatz, Team *mt)
- int getid () const
- virtual void setGeffangen ()
- void setFrei ()

- virtual bool ziehenach (Feld *ziehl)
- void setOrt (Feld *o)
- bool getGeffangen ()
- Team * getMteam ()
- Feld * getOrt ()
- std::vector< Feld * > zuege ()
- virtual ∼Stein ()=default

Geschützte Attribute

- bool geffangen =false
- Feld * ort =nullptr
- Team * mteam =nullptr
- · const int id

4.11.1 Ausführliche Beschreibung

class Stein

Jedes Team besitzt drei Helfer. Sie können sich auf dem Spielfeld bewegen, festgesetzt (gefangen) werden, gegnerische Spielfiguren festsetzen, indem man sie ganz einfach auf das vom Gegner besetzte Feld schickt und in Verbindung mit dem teameigenen KĶnig kĶnnen sie auch selber befreit werden, sollte der Gegner sie gefangen genommen haben. Jede Spielfigur und damit auch jeder Helfer, bekommt bei Spielbeginn einen Platz mittels Pointern zugewiesen. Die Spielfigur-ID und die Spielfeld-ID bestimmen also, welche Spielfigur von welchem Team sich wo im Feld befindet.

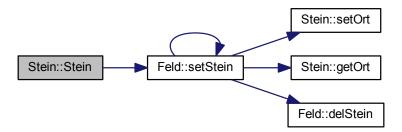
4.11.2 Beschreibung der Konstruktoren und Destruktoren

```
4.11.2.1 Stein::Stein ( )
```

Konstruktor

```
4.11.2.2 Stein::Stein ( int id, Feld * startplatz, Team * mt )
```

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



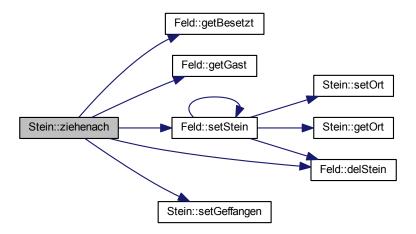
4.11 Stein Klassenreferenz 47

```
4.11.2.3 virtual Stein::∼Stein() [virtual], [default]
4.11.3
        Dokumentation der Elementfunktionen
4.11.3.1 bool Stein::getGeffangen ( )
getGeffangen() Die Funktion beschreibt, ob der Stein gefangen ist oder nicht.
Rückgabe
     the value of gefangen
4.11.3.2 int Stein::getid ( ) const
getid() getid() Diese Funktion sagt aus, ob es sich hierbei um weiß oder schwarz handelt.
Rückgabe
     id der Instanz
4.11.3.3 Team * Stein::getMteam ( )
4.11.3.4 Feld * Stein::getOrt ( )
4.11.3.5 void Stein::setFrei()
setFrei() Setzt den Stein frei Setzt gefangen -> false
4.11.3.6 void Stein::setGeffangen() [virtual]
setGefangen() Setzt den Stein gefangen. gefangen -> true
Erneute Implementation in Koenig.
4.11.3.7 void Stein::setOrt ( Feld * o )
4.11.3.8 bool Stein::ziehenach ( Feld * ziehl ) [virtual]
setOrt Rückt auf das übergebene Feld.
Parameter
                     gibt die neue Position an
             [Feld]
```

Erzeugt am Son Jun 21 2015 22:49:06 für DistanzSpiel von Doxygen

Erneute Implementation in Koenig.

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



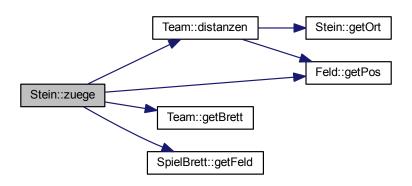
4.11.3.9 std::vector< Feld *> Stein::zuege ()

zuege() Die Funktion Zuege ermittelt alle möglichen Züge und gibt diese als Vector zurück.

Rückgabe

Feld zue

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



4.11.4 Dokumentation der Datenelemente

4.11.4.1 bool Stein::geffangen =false [protected]

4.11.4.2 const int Stein::id [protected]

- **4.11.4.3 Team*** **Stein::mteam =nullptr** [protected]
- **4.11.4.4 Feld* Stein::ort =nullptr** [protected]

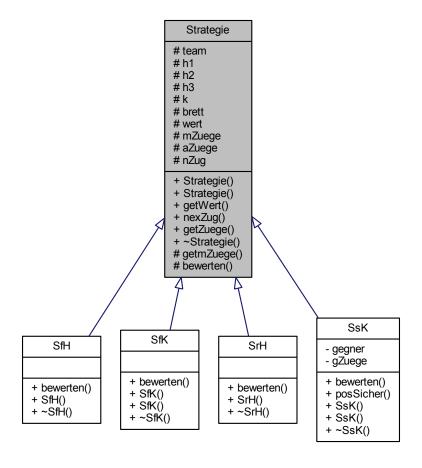
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- · Stein.h
- Stein.cpp

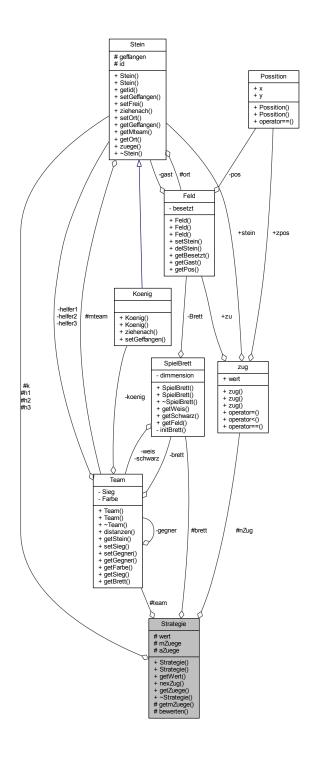
4.12 Strategie Klassenreferenz

#include <Strategie.h>

Klassendiagramm für Strategie:



Zusammengehörigkeiten von Strategie:



- Strategie (Team &team, SpielBrett &b)
- Strategie ()
- int getWert () const
- zug nexZug ()
- std::vector < zug > getZuege () const
- virtual ∼Strategie ()

Geschützte Methoden

- void getmZuege (std::vector < zug > &zuege)
- virtual void bewerten ()=0

Geschützte Attribute

- · Team & team
- Stein & h1
- · Stein & h2
- Stein & h3
- · Stein & k
- SpielBrett & brett
- · int wert
- std::vector< zug > mZuege
- std::vector< zug > aZuege
- zug nZug

4.12.1 Ausführliche Beschreibung

class Strategie

Abstrakte Klasse zur Erzeugung von speziellen Zug-Strategien.

Als Strategien sind jene Funktionen gemeint, welche neben der Bewegung im Feld, zusätzlich auch dafür sorgen, dass es zu einer Sieg/Niederlage Situation kommt. Sie stellen die Möglichkeiten dar, welche die Spielfiguren in den jeweiligen Momenten besitzen. Die Bewertung erfolgt in Echtzeit.

Wir programmierten 4 Strategien ein. Jede der 4 Strategien ist eine Vererbung dieser Klasse.

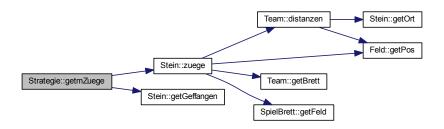
4.12.2 Beschreibung der Konstruktoren und Destruktoren

```
4.12.2.1 Strategie::Strategie ( Team & team, SpielBrett & b )
4.12.2.2 Strategie::Strategie ( )
4.12.2.3 Strategie::~Strategie ( ) [virtual]
4.12.3 Dokumentation der Elementfunktionen
4.12.3.1 void Strategie::bewerten ( ) [protected], [pure virtual]
```

Implementiert in SfK, SfH, SsK und SrH.

4.12.3.2 void Strategie::getmZuege (std::vector< zug > & zuege) [protected]

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



- 4.12.3.3 int Strategie::getWert () const
- 4.12.3.4 std::vector < zug > Strategie::getZuege () const
- 4.12.3.5 zug Strategie::nexZug ()

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



- 4.12.4 Dokumentation der Datenelemente
- **4.12.4.1 std::vector**<**zug**> **Strategie::aZuege** [protected]
- **4.12.4.2 SpielBrett& Strategie::brett** [protected]
- **4.12.4.3 Stein& Strategie::h1** [protected]
- **4.12.4.4 Stein & Strategie::h2** [protected]
- **4.12.4.5 Stein & Strategie::h3** [protected]
- **4.12.4.6 Stein & Strategie::k** [protected]
- **4.12.4.7 std::vector**<**zug**> **Strategie::mZuege** [protected]
- **4.12.4.8 zug Strategie::nZug** [protected]
- 4.12.4.9 Team& Strategie::team [protected]

4.12.4.10 int Strategie::wert [protected]

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

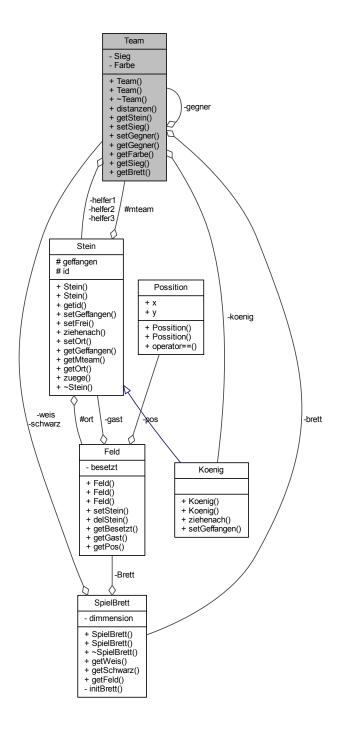
• Strategie.h

• Strategie.cpp

4.13 Team Klassenreferenz

#include <Team.h>

Zusammengehörigkeiten von Team:



- Team (SpielBrett *br, bool f, Feld *s1, Feld *s2, Feld *s3, Feld *k, Team *g)
- Team ()=default
- virtual ~Team ()
- void distanzen (const Stein &anfrage, int *arr)
- · Stein & getStein (int id) const

- void setSieg (bool new_var)
- void setGegner (Team *new_var)
- Team * getGegner () const
- bool getFarbe () const
- bool getSieg ()
- SpielBrett * getBrett () const

Private Attribute

- Stein * helfer1 =nullptr
- Stein * helfer2 =nullptr
- Stein * helfer3 =nullptr
- Koenig * koenig =nullptr
- bool Sieg =false
- Team * gegner =nullptr
- SpielBrett * brett =nullptr
- bool Farbe =false

4.13.1 Ausführliche Beschreibung

class Team

4.13.2 Beschreibung der Konstruktoren und Destruktoren

```
4.13.2.1 Team::Team (SpielBrett * br, bool f, Feld * s1, Feld * s2, Feld * s3, Feld * k, Team * g = nullptr)
```

Erzeugt Team.

```
4.13.2.2 Team::Team( ) [default]
4.13.2.3 Team::~Team( ) [virtual]
```

4.13.3 Dokumentation der Elementfunktionen

4.13.3.1 void Team::distanzen (const Stein & anfrage, int * arr)

distanzen() Trägt x und y Distanzen der "Anderen" Steine in einem Array ein. Array muss 6 Felder besitzen und vom Typ Integer sein.

Parameter

in	&anfrage	: Stein, [out] *arr : int array[6]

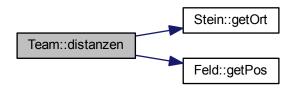
Rückgabe

unsigned short

Parameter

anfrage

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



```
4.13.3.2 SpielBrett * Team::getBrett ( ) const
```

4.13.3.3 bool Team::getFarbe () const

4.13.3.4 Team * Team::getGegner () const

Gibt Pointer auf Gegnerisches Team aus.

4.13.3.5 bool Team::getSieg ()

4.13.3.6 Stein & Team::getStein (int id) const

getStein Gibt Referenz auf Stein mit übergebener ID zurück, bei falschen IDŽs wird Referenz auf Koenig zurückgegeben. 1-3 -> Helfer 4 -> Koenig

Parameter

in	id	: int

Rückgabe

&Stein

4.13.3.7 void Team::setGegner (Team * new_var)

Setze Gegnerisches Team

4.13.3.8 void Team::setSieg (bool new_var)

Set the value of Sieg

Parameter

new_var	the new value of Sieg

4.13.4 Dokumentation der Datenelemente

4.13.4.1 SpielBrett* Team::brett =nullptr [private]

4.14 User Klassenreferenz 57

```
4.13.4.2 bool Team::Farbe =false [private]
4.13.4.3 Team* Team::gegner =nullptr [private]
4.13.4.4 Stein* Team::helfer1 =nullptr [private]
4.13.4.5 Stein * Team::helfer2 =nullptr [private]
4.13.4.6 Stein * Team::helfer3 =nullptr [private]
4.13.4.7 Koenig* Team::koenig =nullptr [private]
4.13.4.8 bool Team::Sieg =false [private]
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- · Team.h
- · Team.cpp

4.14 User Klassenreferenz

#include <User.h>

Zusammengehörigkeiten von User:

+ User() + ~User() + Graphik()

Öffentliche Methoden

- User ()
- virtual ∼User ()
- void Graphik ()

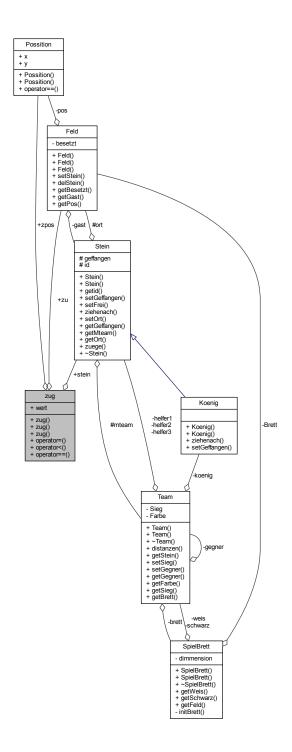
4.14.1 Ausführliche Beschreibung

class User

4.14.2 Beschreibung der Konstruktoren und Destruktoren

4.14.2.1	User::User()
Empty C	onstructor
4.14.2.2	User::~User() [virtual]
Empty D	estructor
4.14.3	Dokumentation der Elementfunktionen
4.14.3.1	void User::Graphik ()
	umentation für diese Klasse wurde erzeugt aufgrund der Dateien:
DIE DORG	amentation für diese Masse wurde erzeugt aufgründ der Dateien.
• Us	ser.h
• Us	ser.cpp
4.15	zug Strukturreferenz
#inclu	ide <zug.h></zug.h>

Zusammengehörigkeiten von zug:



- zug ()=default
- zug (Feld *z, Stein *s)
- zug (const zug &z)
- zug & operator= (const zug &z)
- bool operator< (const zug &z) const
- bool operator== (const zug &z) const

Öffentliche Attribute

- Feld * zu =nullptr
- Stein * stein =nullptr
- int wert =100
- Possition zpos

4.15.1 Ausführliche Beschreibung

struct Zug Daten Struktur die einen Spiel-Zug Symbolysiert.

4.15.2 Beschreibung der Konstruktoren und Destruktoren

```
4.15.2.1 zug::zug( ) [default]
4.15.2.2 zug::zug( Feld * z, Stein * s ) [inline]
```

4.15.3 Dokumentation der Elementfunktionen

4.15.2.3 zug::zug(const zug & z) [inline]

```
4.15.3.1 bool zug::operator < ( const zug & z ) const [inline]
```

Kleiner als Operator Vergleicht Zuege nach Wertigkeit;

Parameter

Z

Rückgabe

```
4.15.3.2 zug& zug::operator=(const zug & z) [inline]
```

```
4.15.3.3 bool zug::operator== ( const zug & z ) const [inline]
```

Vergleichs-Operator Vergleicht Zuege auf gleiche Ziel-Position

Parameter

Z

Rückgabe

4.15.4 Dokumentation der Datenelemente

4.15.4.1 Stein* zug::stein =nullptr

4.15.4.2 int zug::wert =100

4.15.4.3 Possition zug::zpos

4.15.4.4 Feld* zug::zu =nullptr

Die Dokumentation für diese Struktur wurde erzeugt aufgrund der Datei:

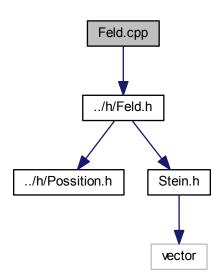
• zug.h

Kapitel 5

Datei-Dokumentation

5.1 Feld.cpp-Dateireferenz

#include "../h/Feld.h" Include-Abhängigkeitsdiagramm für Feld.cpp:



Makrodefinitionen

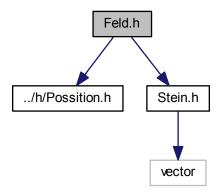
• #define STEIN_C

5.1.1 Makro-Dokumentation

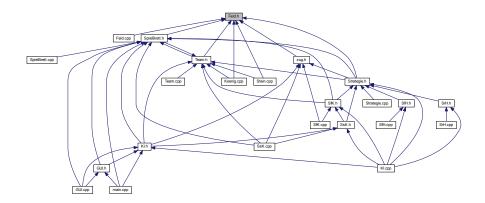
5.1.1.1 #define STEIN_C

5.2 Feld.h-Dateireferenz

#include "../h/Possition.h"
#include "Stein.h"
Include-Abhängigkeitsdiagramm für Feld.h:



Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:



Klassen

class Feld

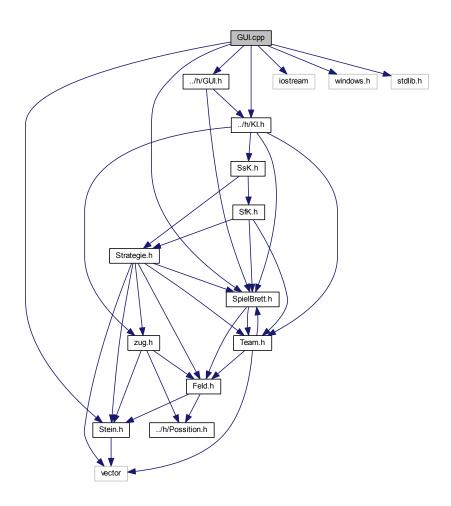
5.3 GUI.cpp-Dateireferenz

#include "../h/GUI.h"

5.4 GUI.h-Dateireferenz 65

```
#include <iostream>
#include <windows.h>
#include <stdlib.h>
#include "../h/KI.h"
#include "../h/Spielbrett.h"
#include "../h/Stein.h"
```

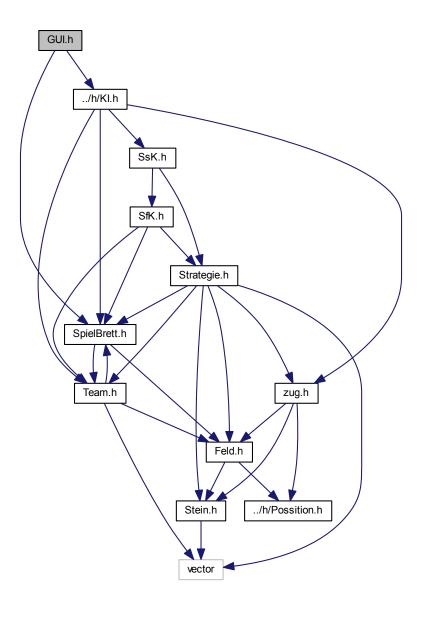
Include-Abhängigkeitsdiagramm für GUI.cpp:



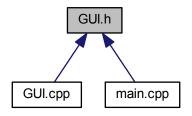
5.4 GUI.h-Dateireferenz

```
#include "SpielBrett.h"
#include "../h/KI.h"
```

Include-Abhängigkeitsdiagramm für GUI.h:



Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:



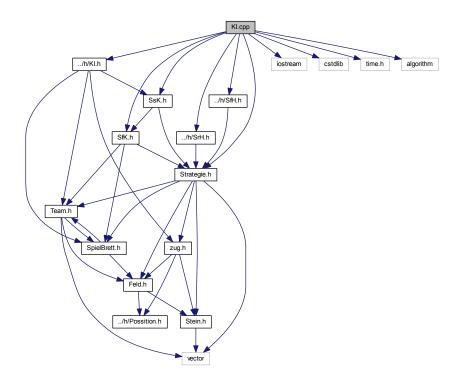
Klassen

• class GUI

5.5 Kl.cpp-Dateireferenz

```
#include "../h/KI.h"
#include "../h/SfK.h"
#include "../h/SsK.h"
#include "../h/SfH.h"
#include "../h/SrH.h"
#include <iostream>
#include <cstdlib>
#include <time.h>
#include <algorithm>
```

Include-Abhängigkeitsdiagramm für Kl.cpp:

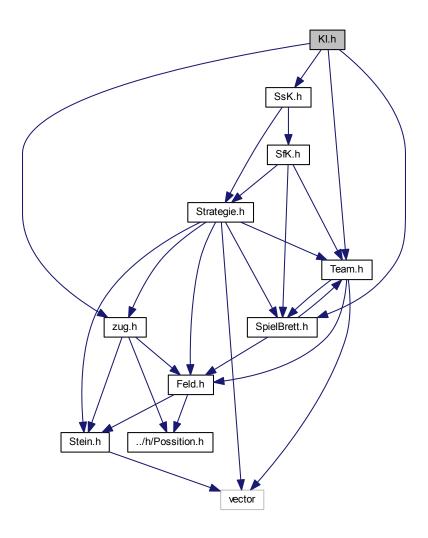


5.6 Kl.h-Dateireferenz

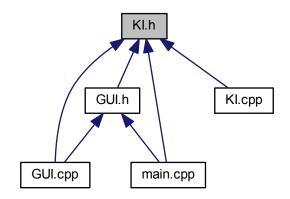
```
#include "Team.h"
#include "SpielBrett.h"
#include "zug.h"
#include "SsK.h"
```

5.6 Kl.h-Dateireferenz 69

Include-Abhängigkeitsdiagramm für Kl.h:



Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:



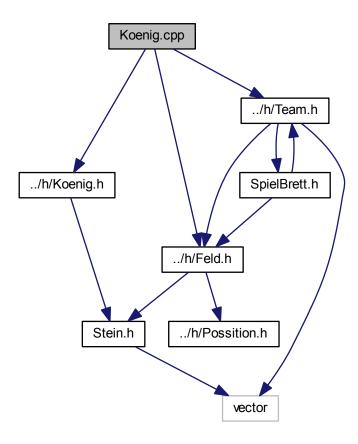
Klassen

• class KI

5.7 Koenig.cpp-Dateireferenz

```
#include "../h/Koenig.h"
#include "../h/Feld.h"
#include "../h/Team.h"
```

Include-Abhängigkeitsdiagramm für Koenig.cpp:



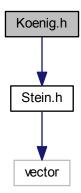
Makrodefinitionen

- #define KOEING_C
- 5.7.1 Makro-Dokumentation
- 5.7.1.1 #define KOEING_C

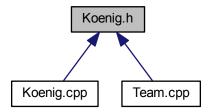
5.8 Koenig.h-Dateireferenz

#include "Stein.h"

Include-Abhängigkeitsdiagramm für Koenig.h:



Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:



Klassen

class Koenig

Makrodefinitionen

• #define KOENIG_H

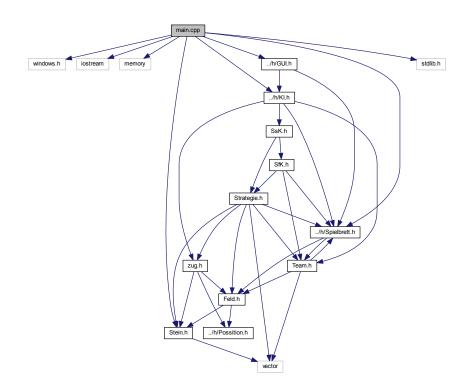
5.8.1 Makro-Dokumentation

5.8.1.1 #define KOENIG_H

5.9 main.cpp-Dateireferenz

#include <windows.h>

```
#include <iostream>
#include <memory>
#include "../h/Spielbrett.h"
#include <stdlib.h>
#include "../h/KI.h"
#include "../h/GUI.h"
#include "../h/Stein.h"
Include-Abhängigkeitsdiagramm für main.cpp:
```



Funktionen

• int main (int _argc, char *argv[])

5.9.1 Dokumentation der Funktionen

5.9.1.1 int main (int _argc, char * argv[])

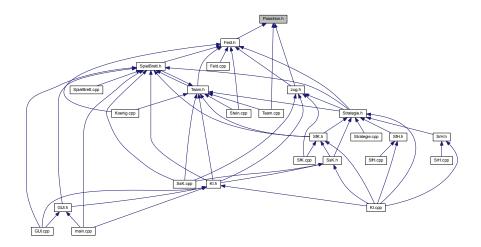
Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



5.10 Main.h-Dateireferenz

5.11 Possition.h-Dateireferenz

Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:



Klassen

• struct Possition

Makrodefinitionen

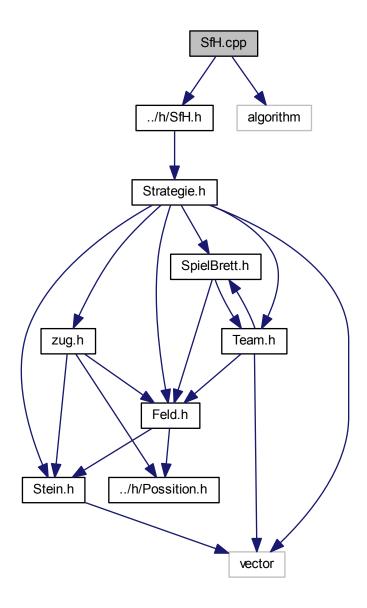
• #define POSSITION_H

5.11.1 Makro-Dokumentation

5.11.1.1 #define POSSITION_H

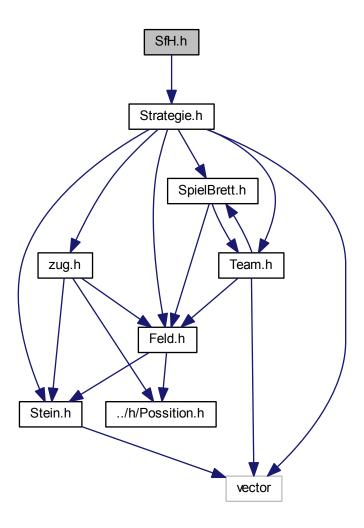
5.12 SfH.cpp-Dateireferenz

#include "../h/SfH.h"
#include <algorithm>
Include-Abhängigkeitsdiagramm für SfH.cpp:

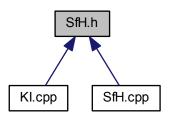


5.13 SfH.h-Dateireferenz

#include "Strategie.h"
Include-Abhängigkeitsdiagramm für SfH.h:



Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:



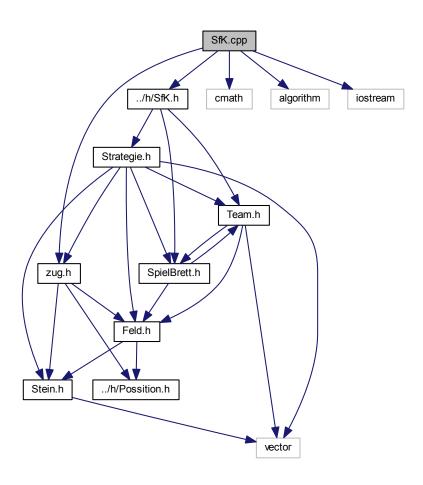
Klassen

• class SfH

5.14 SfK.cpp-Dateireferenz

```
#include "../h/SfK.h"
#include <cmath>
#include <algorithm>
#include "../h/zug.h"
#include <iostream>
```

Include-Abhängigkeitsdiagramm für SfK.cpp:

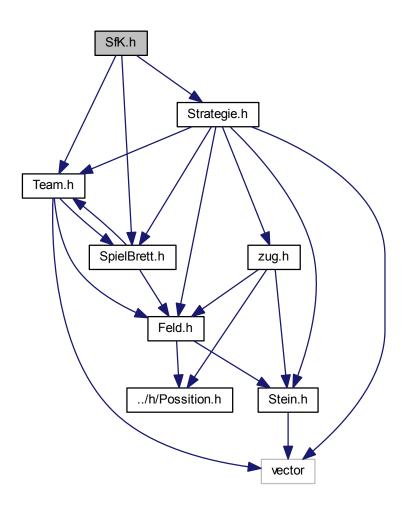


5.15 SfK.h-Dateireferenz

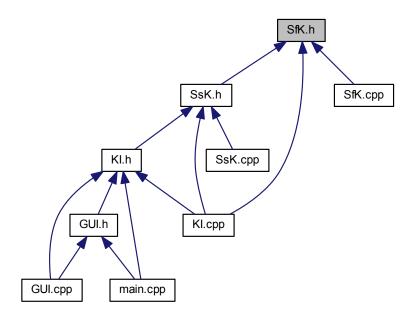
#include "Team.h"
#include "SpielBrett.h"
#include "Strategie.h"

5.15 SfK.h-Dateireferenz 79

Include-Abhängigkeitsdiagramm für SfK.h:



Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:



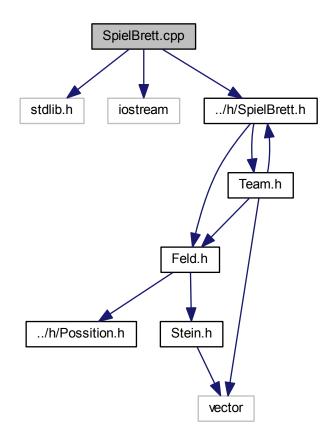
Klassen

• class SfK

5.16 SpielBrett.cpp-Dateireferenz

#include <stdlib.h>
#include <iostream>
#include "../h/SpielBrett.h"

Include-Abhängigkeitsdiagramm für SpielBrett.cpp:



Makrodefinitionen

• #define SPIELBRETT_C

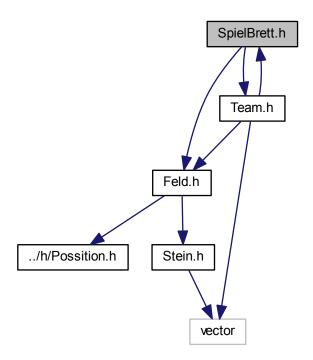
5.16.1 Makro-Dokumentation

5.16.1.1 #define SPIELBRETT_C

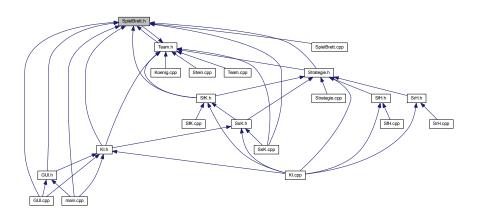
5.17 SpielBrett.h-Dateireferenz

```
#include "Feld.h"
#include "Team.h"
```

Include-Abhängigkeitsdiagramm für SpielBrett.h:



Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:



Klassen

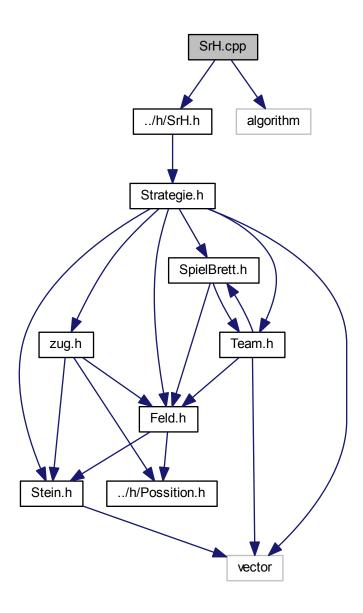
• class SpielBrett

5.18 SrH.cpp-Dateireferenz

#include "../h/SrH.h"

5.19 SrH.h-Dateireferenz 83

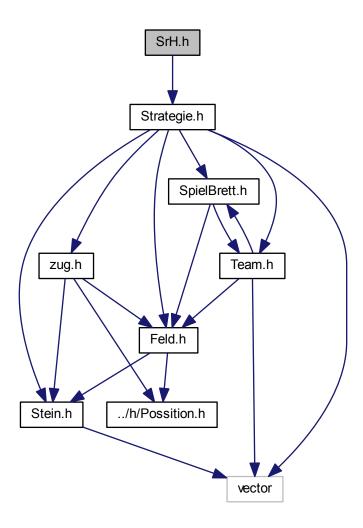
#include <algorithm>
Include-Abhängigkeitsdiagramm für SrH.cpp:



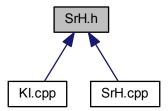
5.19 SrH.h-Dateireferenz

#include "Strategie.h"

Include-Abhängigkeitsdiagramm für SrH.h:



Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:



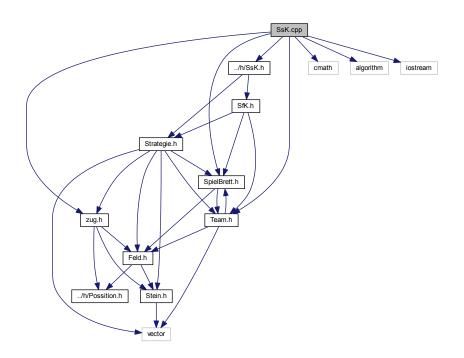
Klassen

• class SrH

5.20 SsK.cpp-Dateireferenz

```
#include "../h/SsK.h"
#include <cmath>
#include <algorithm>
#include "../h/zug.h"
#include "../h/SpielBrett.h"
#include "../h/Team.h"
#include <iostream>
```

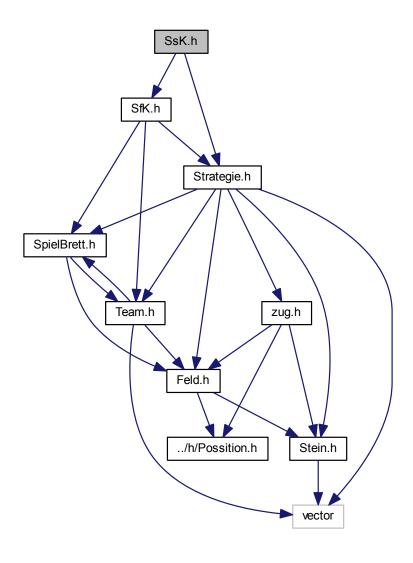
Include-Abhängigkeitsdiagramm für SsK.cpp:



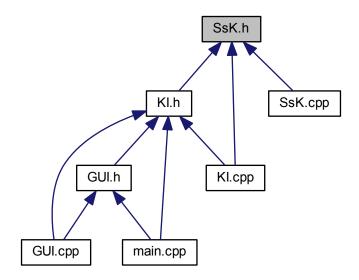
5.21 SsK.h-Dateireferenz

```
#include "Strategie.h"
#include "SfK.h"
```

Include-Abhängigkeitsdiagramm für SsK.h:



Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:



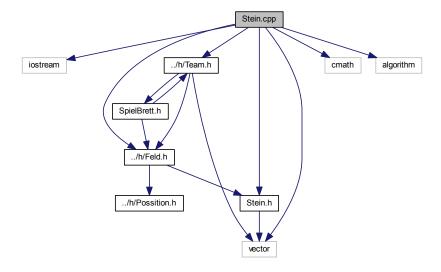
Klassen

• class SsK

5.22 Stein.cpp-Dateireferenz

```
#include <iostream>
#include "../h/Feld.h"
#include "../h/Team.h"
#include "../h/Stein.h"
#include <cmath>
#include <vector>
#include <algorithm>
```

Include-Abhängigkeitsdiagramm für Stein.cpp:



Makrodefinitionen

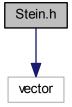
• #define STEIN_C

5.22.1 Makro-Dokumentation

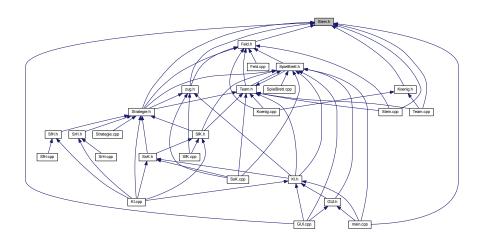
5.22.1.1 #define STEIN_C

5.23 Stein.h-Dateireferenz

#include <vector>
Include-Abhängigkeitsdiagramm für Stein.h:



Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:



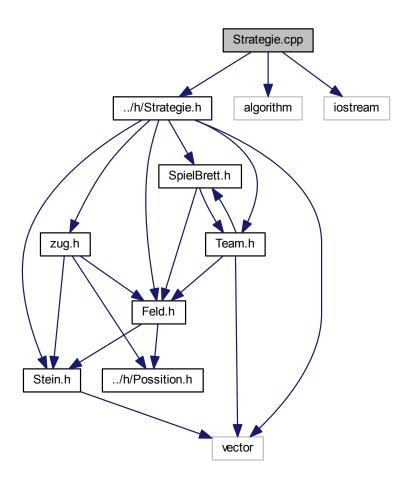
Klassen

• class Stein

5.24 Strategie.cpp-Dateireferenz

```
#include "../h/Strategie.h"
#include <algorithm>
#include <iostream>
```

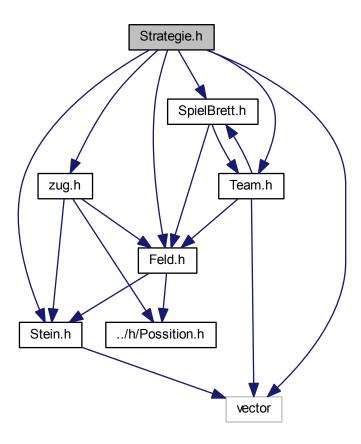
Include-Abhängigkeitsdiagramm für Strategie.cpp:



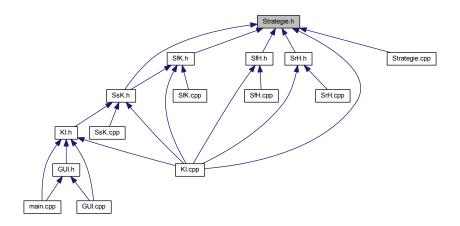
5.25 Strategie.h-Dateireferenz

```
#include <vector>
#include "Feld.h"
#include "SpielBrett.h"
#include "Team.h"
#include "Stein.h"
#include "zug.h"
```

Include-Abhängigkeitsdiagramm für Strategie.h:



Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:

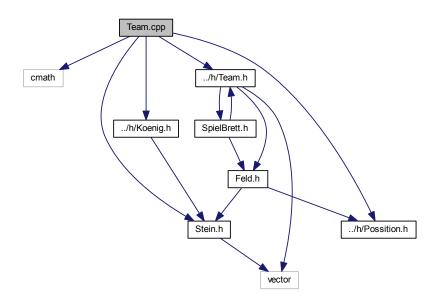


Klassen

· class Strategie

5.26 Team.cpp-Dateireferenz

```
#include <cmath>
#include "../h/Team.h"
#include "../h/Stein.h"
#include "../h/Koenig.h"
#include "../h/Possition.h"
Include-Abhängigkeitsdiagramm für Team.cpp:
```



Makrodefinitionen

• #define TEAM_C

5.26.1 Makro-Dokumentation

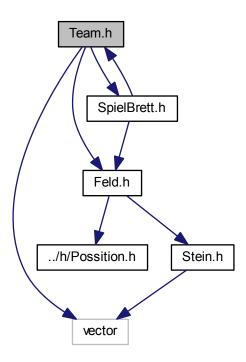
5.26.1.1 #define TEAM_C

5.27 Team.h-Dateireferenz

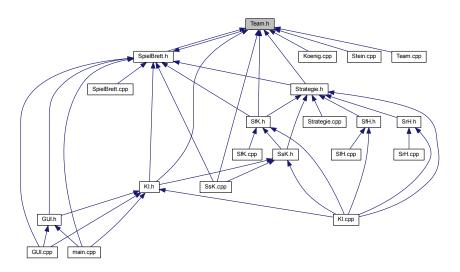
```
#include <vector>
#include "Feld.h"
#include "SpielBrett.h"
```

5.27 Team.h-Dateireferenz 93

Include-Abhängigkeitsdiagramm für Team.h:



Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:

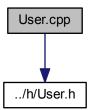


Klassen

• class Team

5.28 User.cpp-Dateireferenz

#include "../h/User.h" Include-Abhängigkeitsdiagramm für User.cpp:



Makrodefinitionen

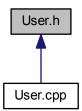
• #define USER_C

5.28.1 Makro-Dokumentation

5.28.1.1 #define USER_C

5.29 User.h-Dateireferenz

Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:



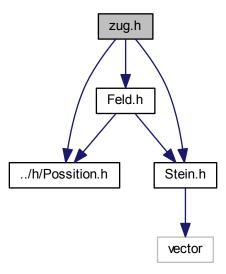
Klassen

• class User

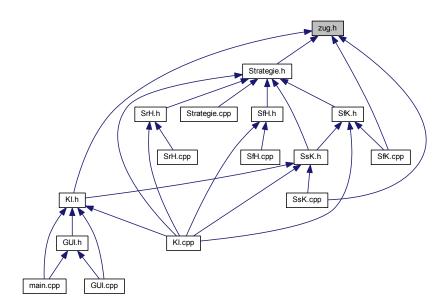
5.30 zug.h-Dateireferenz

#include "Feld.h"
#include "Stein.h"
#include "Possition.h"

Include-Abhängigkeitsdiagramm für zug.h:



Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:



Klassen

• struct zug

Index

~KI	KI, 19
KI, 15	
~SfH	Farbe
SfH, 27	Team, 56
\sim SfK	Feld, 7
SfK, 31	besetzt, 10
\sim SpielBrett	delStein, 9
SpielBrett, 35	Feld, 9
~SrH	gast, 10
SrH, 39	getBesetzt, 9
~SsK	getGast, 9
SsK, 42	getPos, 9
~Stein	pos, 10
Stein, 46	setStein, 10
~Strategie	Feld.cpp, 63
Strategie, 51	STEIN_C, 63
~Team	Feld.h, 64
Team, 55	
~User	GUI, 10
User, 58	brett, 12
,	GUI, 12
aZuege	Klsw, 12
Strategie, 52	Spieler, 12
abrett	zeichneAnleitung, 12
KI, 19	zeichneSpielfeld, 12
anzstrat	zeichneZug, 12
KI, 19	GUI.cpp, 64
	GUI.h, 65
besetzt	gZuege
Feld, 10	SsK, 43
bewerten	gast
SfH, 27	Feld, 10
SfK, 31	geffangen
SrH, 39	Stein, 48
SsK, 42	gegner
Strategie, 51	SsK, 43
Brett	Team, 57
SpielBrett, 36	getBesetzt
brett	Feld, 9
GUI, 12	getBrett
Strategie, 52	KI, 15
Team, 56	Team, 56
	getFarbe
delStein	Team, 56
Feld, 9	getFeld
dimmension	SpielBrett, 35
SpielBrett, 36	getGast
distanzen	Feld, 9
Team, 55	getGeffangen
dv	Stein, 47

98 INDEX

getGegner	nexZug, 17
Team, 56	seachBestZug, 18
getMteam	strat, 19
Stein, 47	t, 19
getOrt	KI.cpp, 67
Stein, 47	Kl.h, 68
getPos	Klsw
Feld, 9	GUI, 12
getSchwarz	KOEING_C
SpielBrett, 35	Koenig.cpp, 71
getSieg	KOENIG H
Team, 56	Koenig.h, 72
getStein	Koenig, 20
Team, 56	Koenig, 22
getTeam	setGeffangen, 22
KI, 15	ziehenach, 22
getWeis	koenig
SpielBrett, 35	Team, 57
getWert	Koenig.cpp, 70
Strategie, 52	KOEING_C, 71
getZuege	Koenig.h, 71
Strategie, 52	KOENIG H, 72
-	ROLINIA_II, 72
getid	mZuege
Stein, 47	•
getmZuege	Strategie, 52
Strategie, 51	main
Graphik	main.cpp, 73
User, 58	main.cpp, 72
	main, <mark>73</mark>
h1	Main.h, 74
Strategie, 52	mergeStrategie
h2	KI, 15, 16
Strategie, 52	
-	mteam
h3	Stein, 48
Strategie, 52	7
helfer1	nZug
Team, 57	KI, 19
helfer2	Strategie, 52
Team, 57	nexZug
helfer3	KI, 17
Team, 57	Strategie, 52
10411, 07	
id	operator<
	zug, 60
Stein, 48	
initBrett	operator=
SpielBrett, 36	zug, 60
	operator==
k	Possition, 24
Strategie, 52	zug, 60
KI, 13	ort
∼KI, 15	Stein, 49
abrett, 19	J.J., 10
anzstrat, 19	POSSITION H
	-
dv, 19	Possition.h, 74
getBrett, 15	pos
getTeam, 15	Feld, 10
KI, 15	posSicher
mergeStrategie, 15, 16	SsK, 42
nZug, 19	Possition, 23
- g, -	· · · · · · · · · · · ·

INDEX 99

operator==, 24 Possition, 24 x, 24 y, 24 y, 24 Possition.h, 74 PosSiTION_H, 74 SrH., 39 PosSiTION_H, 74 SrH.cpp, 82 SrH.cpp, 82 SrH.cpp, 83 Stein.cpp, 81 Stein.cpp, 83 Stein.cpp, 88 Stein.cpp, 88 SeachBestZug KI, 18 setFrei Stein, 47 setGeffangen Koenig, 22 Stein, 47 setGeffangen Koenig, 22 stein, 47 setGeffangen Koenig, 22 setOrt Team, 56 setOrt Stein, 47 setGeffanger Team, 56 setOrt Stein, 47 setStein Stein, 47 setStein Stein, 47 setStein Stein, 47 setGeffangen Koenig, 22 Stein, 47 setGeffangen Koenig, 28 Stein, 49 Stein, 48 Stein, 48 Stein, 49 S		
Possition, 24	operator==, 24	SrH, 36
x, 24 y, 24 SrH, 39 SrH, 31 SrEIN_C SpielBrett, 26 SeachBestZug SeachBestZug Srein, 47 SetGeffangen Geffangen, 48 SetGeffangen Koenig, 22 Stein, 47 SetGeffangen Geffangen, 47 SetGeffangen Geffangen Geffangen, 47 SetGeffangen Geffangen, 47 SetGeffangen, 47 SetGeffangen Geffangen, 47 SetGeffangen Geffangen, 47 SetGeffangen, 47 SetGeffangen, 47 SetGeffangen, 47 SetGeffangen, 47 SetSien, 47 SetGeffangen, 47 SetSien, 47 SetSien, 47 SetSien, 48 SetIrei, 47 SetSien, 48 SetIrei, 47 SetSien, 48 SetIrei, 47 SetGeffangen, 47 SetSien, 48 SetIrei, 48 SetIrei, 48 SetIrei, 48 SetIrei, 49 SetIrei, 47 SetOrt, 4	•	\sim SrH, 39
Possition.h, 74	x, 24	bewerten, 39
Possition.h, 74 SrH.cpp, 82 POSSITION_H, 74 SrH.h, 83 SPIELBRETT_C ~SsK, 40 SpielBrett.cpp, 81 —SsK, 42 Stein.cpp, 63 gegner, 43 Stein.cpp, 88 posSicher, 42 Schwarz SsK, 42 SpielBrett, 36 SsK.cpp, 85 seachBestZug SsK.h, 85 Kl, 18 Stein, 43 setFrei ~Stein, 43 setGeffangen geffangen, 48 koenig, 22 getfangen, 48 stein, 47 getfangen, 49 steGegner getOrt, 47 getGeffangen, 47 getid, 47 stedGegner getOrt, 47 stedGegner getTeri, 47 setGegner getTeri, 47 setGeffangen, 47 getTeri, 47 setSlein, 47 setStein, 49 setSlein, 47 setStein, 49 setSlei, 40 setStein, 49 setSlei, 41 setStein, 49 setSlei, 10 setTrei, 47 setSlei, 10 setTeri, 47 setSlein, 49 <td></td> <td></td>		
POSSITION_H, 74 SPIELBRETT_C	-	
SPIELBRETT_C SsK, 40 SpielBrett.cpp, 81 bewerten, 42 STEIN_C gZuege, 43 Feld cpp, 63 gegner, 43 Stein.cpp, 88 posSicher, 42 Schwarz SsK, 42 SpielBrett, 36 SsK.cpp, 85 seachBestZug SsK., 85 Ki, 18 Stein, 43 setFrei Stein, 43 Stein, 47 Stein, 46 setGeffangen getGaffangen, 48 Koenig, 22 getMeam, 47 Stein, 47 getMeam, 47 setGegner getid, 47 Team, 56 getFrei, 47 setStein mteam, 48 Team, 56 setFrei, 47 setStein setGeffangen, 47 setStein setGeffangen, 47 setStein setGeffangen, 47 setStein, 47 setGeffangen, 47 setStein, 47 setGeffangen, 47 setStein, 48 setGeffangen, 47 setStein, 48 setGeffangen, 47 setStein, 48 setGeffangen, 47 setStein, 48		• •
SPIELBRETT_C ~SsK, 42 bewerten, 42 bewerten, 42 STEIN_C gZuege, 43 Feld.cpp, 63 gegner, 43 Stein.cpp, 88 soscathes. schwarz SsK, 42 SpielBrett, 36 SsK, 42 seachBestZug SsK, h, 85 Kl, 18 Stein, 43 setFrei ~Stein, 46 stein, 47 geffangen, 48 Koenig, 22 getdefangen, 47 Stein, 47 getdefangen, 47 setGeffangen getdefangen, 47 setOrt getid, 47 setSieg getid, 47 getid, 47 getid, 47 getid, 47 getid, 47 setSien, 47 setGeffangen, 48 setStein, 47 setGeffangen, 47 setStein, 47 setGeffangen, 47 setStein, 47 setGeffangen, 47 setStein setGeffangen, 47 setStein setGeffangen, 47 setStein, 49 setFrei, 47 setH, 27 zuge, 48 stein, 27 zuge, 60	_ /	•
SpielBrett.cpp, 81 STEIN_C Feld.cpp, 63 Stein.cpp, 88 schwarz SpielBrett, 36 seachBestZug KI, 18 setFrei Stein, 47 setGeffangen Koenig, 22 Stein, 47 setGegner Team, 56 setOrt Stein, 47 setSieg Team, 56 setStein Feld, 10 SH, 24 ~SIH, 27 bewerten, 27 SH, 27 Sewerten, 27 SHL, 27 Sewerten, 31 SK, 31 SK, 31 SK, 31 SKK, 31 SKK	SPIELBRETT_C	
STEIN_C gZuege, 43 Feld.cpp, 83 gegner, 43 Schwarz posSicher, 42 SpielBrett, 36 SsK.cpp, 85 seachBestZug SsK.cpp, 85 Kl, 18 Ssk., 42 SetFrei Stein, 43 Stein, 47 Stein, 43 setGeffangen getGeffangen, 47 Koenig, 22 getMteam, 47 Stein, 47 getOrt, 47 setGegner getOrt, 47 feam, 56 id, 48 setOrt meam, 48 feid, 10 setFrei, 47 setStein setGeffangen, 47 setStein setGeffangen, 47 setStein, 49 setFrei, 47 setStein, 49 setFrei, 47 setStein, 49 setFrei, 47 setStein, 49 setFrei, 47 setStein, 49 stein, 48 stein, 49 stein, 48 stein, 49 stein SfK, 28 Stein, 88 strategie, 51 strategie, 51 setStein, 49 strategie, 51	SpielBrett.cpp, 81	,
Feld.cpp, 63	STEIN_C	
Stein.cpp, 88 schwarz SpielBrett, 36 seachBestZug KI, 18 setFrei Stein, 47 setGeffangen Koenig, 22 Stein, 47 setGegner Team, 56 setOrt Stein, 47 setSieg Team, 56 setStein Feld, 10 SfH, 24 ~SIH, 27 bewerten, 27 SfH, 27 Stein, 27 SfH, 28 Stein, 31 bewerten, 31 SfK, 31 SfK	Feld.cpp, 63	
Schwarz SpielBrett, 36 seachBestZug KI, 18 setFrei Stein, 47 setGeffangen Koenig, 22 Stein, 47 setGegner Team, 56 setOrt Stein, 47 setSieg Team, 56 setStein Feld, 10 SfH, 24 ~SfH, 27 bewerten, 27 SfH, 27 Sewerten, 27 SfK, 28 ~SfK, 31 Sewerten, 31 SfK, 28 ~SfK, 31 SfK, 31 SfK, 29 Team, 57 SpielBrett, 35 Brett, 36 dimmension, 36 getFeld, 35 getSchwarz, 35 getSchwarz, 35 getSelBrett, 34 weis, 36 SpielBrett, 34 SpielBrett, 34 weis, 36 SpielBrett, 34 SpielBrett, 34 weis, 36 SpielBrett, 34 SpielBrett, 36 SpielBrett, 37 SpielBrett, 34 weis, 36 SpielBrett, 34 weis, 36 SpielBrett, 36 SpielBrett, 34 weis, 36 SpielBrett, 36 SpielBrett, 37 SpielBrett, 37 SpielBrett, 34 weis, 36 SpielBrett, 35 SpielBrett, 36 SpielBrett, 34 weis, 36 SpielBrett, 34 weis, 36 SpielBrett, 35 SpielBrett, 36 SpielBrett, 36 SpielBrett, 37 SpielBrett, 37 SpielBrett, 38 SpielBrett, 39 SpielBrett, 30 SpielBrett, 30 SpielBrett, 30 SpielBrett, 30 SpielBrett, 30 SpielBrett, 31 SpielBrett, 32 SpielBrett, 33 SpielBrett, 34 SpielBrett, 35 SpielBrett, 35 SpielBrett, 35 SpielBrett, 36 SpielBrett, 37 SpielBrett, 38 SpielBrett, 39 Spi	Stein.cpp, 88	
SpielBrett, 36 SsK.cpp, 85 seachBestZug Kl, 18 setFrei Stein, 43 setGeffangen geffangen, 48 Koenig, 22 getMteam, 47 setGegner getId, 47 Team, 56 id, 48 setOrt mteam, 47 setSieg setFrei, 47 Team, 56 setFrei, 47 setSieg setFrei, 47 Team, 56 setFrei, 47 setStein setGeffangen, 47 setStein setGeffangen, 47 setStein setGeffangen, 47 setStein setGeffangen, 47 setStein, 46 ziehenach, 47 setStein, 46 ziehenach, 47 setStein, 46 ziehenach, 47 setStein, 48 ziehenach, 47 setStein, 48 stein, 29 stH, 27 stein sewerten, 27 stein stH, 27 stein sewerten, 31 stint, 8 stK, 28 Strategie, 58 strategie, 51 strategie, 51 <tr< td=""><td>schwarz</td><td>•</td></tr<>	schwarz	•
seachBestZug SsK.h, 85 kl, 18 SetErrei Stein, 47 SetGeffangen Koenig, 22 getfangen, 48 Stein, 47 getGeffangen, 47 getGegner getOrt, 47 Team, 56 getOrt, 47 setOrt id, 48 stein, 47 mteam, 48 ort, 49 setFrei, 47 setStein setFrei, 47 setStein setGeffangen, 47 setStein setGeffangen, 47 setStein setGeffangen, 47 setStein, 46 setOrt, 47 setStein setGeffangen, 47 setStein, 48 setFrei, 47 setStein, 49 stein.cpp, 87 SfK, 28 Strategie, 51 sfK, 21 strategie, 51 skx, 19 Strategie, 51 skx, 29 setTein, 52 <td>SpielBrett, 36</td> <td></td>	SpielBrett, 36	
KI, 18 Stein, 47 setGeffangen Koenig, 22 Stein, 47 setGegner Team, 56 setOrt Stein, 47 setSieig Team, 56 setStein Feld, 10 SH, 24 ∼SfH, 27 bewerten, 27 SfH, 27 bewerten, 27 SfH, 31 bewerten, 31 SfK, 31 SfR, 35 SpielBrett, 35 Seg Team, 57 SpielBrett, 35 Seg Team, 57 SpielBrett, 35 Seg Team, 57 SpielBrett, 35 SegetSchwarz, 35 SegetSchwarz, 35 SegetSchwarz, 35 SegetSchwarz, 35 SegetSchwarz, 36 SegetSchwarz, 36 SegetSchwarz, 36 SegetSchwarz, 36 SegetSpielBrett, 34 weis, 36 SpielBrett, 34 weis, 36 SpielBrett, 52 SpielBrett, 52 SpielBrett, 54 SpielBrett, 55 SpielBrett, 57 SpielBrett, 58 SpielBrett, 58 SpielBrett, 58 SpielBrett, 58 SpielBrett, 59 SpielBrett, 81	seachBestZug	• •
sett-rei Stein, 47 setGeffangen geffangen, 48 koenig, 22 getGeffangen, 47 stein, 47 getMteam, 47 getOrt, 47 getid, 47 setOrt id, 48 stein, 47 mteam, 48 setSieg setFrei, 47 Team, 56 setFrei, 47 setStein setGeffangen, 47 Feld, 10 setTrei, 47 SfH, 24 ziehenach, 47 ~SfH, 27 zuege, 48 Stein, 46 ziehenach, 47 zuege, 48 stein SfH, 27 zug, 60 SfH, 27 zug, 60 SfH, 28 Stein, pp, 87 SfK, 28 Stein, pp, 87 SfK, 31 strat bewerten, 31 strat SfK, 31 strategie, 49 ~SfKh, 78 Strategie, 51 Sieg aZuege, 52 Team, 57 bewerten, 51 SpielBrett, 33 getVert, 52 getVert, 52 getZuege, 52 getWert, 52 <	KI, 18	
Stein, 47 geffangen, 48 setGeffangen getGeffangen, 47 getDeffangen, 47 getMteam, 47 getDrt, 47 getOrt, 47 setOrt id, 48 setSieg ort, 49 Team, 56 setFrei, 47 setSieg setFrei, 47 Team, 56 setFrei, 47 setStein setGeffangen, 47 setStein setStein setStein setStein setStein setStein setStein set	setFrei	•
setCeeflangen getGeffangen, 47 setGegner getOtt, 47 setOrt getId, 47 setSein, 47 getId, 47 setSieg getId, 47 Team, 56 setFrei, 47 setSieg ort, 49 Team, 56 setFrei, 47 setStein setGeffangen, 47 Feld, 10 Stein, 46 Stein, 47 setOrt, 47 Stein, 46 Stein, 46 x-SfH, 27 zuege, 48 stein, 46 Stein, 46 x-SfH, 27 zuege, 48 stein, 47 stein, 46 x-SfH, 27 zuege, 48 Stein, 47 stein, 47 x-SfH, 27 stein, 47 x-SfH, 27 stein, 47 x-SfH, 27 stein, 48 x-SfK, 31 stein, 49 x-SfK, 31 strategie, 49 x-SfK, 31 strategie, 49 x-SfK, 31 strategie, 51 x-Sieg azuege, 52 bewerten, 51 brett, 52 getWeit, 52 <	Stein, 47	
Koenig, 22 getMteam, 47 setGegner getJ, 47 Team, 56 id, 48 setOrt mteam, 48 setSieg ort, 49 Team, 56 setFrei, 47 setStein setGeffangen, 47 Feld, 10 setOrt, 47 SfH, 24 stein, 46 ~SfH, 27 ziehenach, 47 bewerten, 27 zug, 60 SfH, 27 stein SfH, 27 zug, 60 SfH, 28 STEIN_C, 88 ~SfK, 31 strat bewerten, 31 strat SfK, 31 strat SfK, 31 strategie, 49 ~SfK, 18 strategie, 49 ~SfK, 19 Strategie, 51 SfK, 78 strategie, 51 Sieg aZuege, 52 Team, 57 bewerten, 51 SpielBrett, 35 getZuege, 52 getWeit, 52 getZuege, 52 getSchwarz, 36 getZuege, 52 spielBrett, 36 nZuege, 52 spielBrett, 34 nZuege	setGeffangen	
Stein, 47 getWitteam, 47 setGegner getVrt, 47 Team, 56 id, 48 setOrt mteam, 48 StestSieg ort, 49 Team, 56 setFrei, 47 setStein setGeffangen, 47 Feld, 10 stein, 46 SitH, 27 stein, 46 bewerten, 27 zuege, 48 SfH, 27 stein SfH, 28 Stein, 98 SfK, 31 stein bewerten, 27 stein SfK, 31 stratl k1, 19 Strategie, 48 SfK, 31 stratl k1, 19 Strategie, 49 SfK, 31 strategie, 49 SfK, 78 -Strategie, 51 Sieg aZuege, 52 Team, 57 bewerten, 51 SpielBrett, 35 <t< td=""><td>Koenig, 22</td><td></td></t<>	Koenig, 22	
setGegner gettoft, 47 Team, 56 id, 48 setOrt mteam, 48 Stein, 47 ort, 49 setSteig setFrei, 47 setStein setGeffangen, 47 Feld, 10 setOrt, 47 SfH, 24 ziehenach, 47 vewerten, 27 zuege, 48 SfH, 27 stein.cpp, 87 SfH, 28 Stein.cpp, 87 SfK, 31 strateje, 51 SfK, 28 Strategie, 49 SfK, 31 strategie, 49 SfK, 31 Strategie, 51 SfK, 31 Strategie, 51 SfK, 31 Strategie, 51 SpielBrett, 33 petWerten, 52 SpetWert, 52 getWerten, 52 SpielBrett, 36 k, 52 mcxZug, 52 s	_	•
Team, 56 setOrt Stein, 47 setSieg Team, 56 setStein Feld, 10 SiH, 24 SfH, 27 bewerten, 27 SiH, 28 Sitin, 76 Sitin, 76 Sitin, 76 Sitin, 76 Sitin, 76 Sitin, 76 Sitin, 28 Sitin, 28 Sitin, 31 SiK, 31 SiK, 31 SiK, 31 SiK, 31 SiK, 31 SiK, 31 Sixat Sixa		•
setOrt Id, 48 Stein, 47 mteam, 48 setSieg ort, 49 setStein setFrei, 47 setOtt, 47 setOrt, 47 SiH, 24 Stein, 46 ~SfH, 27 zuege, 48 SiH, 27 stein SiH, 76 Stein.cpp, 87 SiK, 28 STEIN_C, 88 ~SfK, 31 Stein.h, 88 bewerten, 31 strat SfK, 31 Strategie, 49 Sieg, 49 ~Strategie, 49 Team, 57 Strategie, 51 SpielBrett, 33 brett, 52 getWert, 52 getWert, 52 getWeit, 52 getZuege, 52 bewerten, 51 brett, 52 getWeis, 35 h1, 52 initBrett, 36 getZuege, 51 schwarz, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett.cpp, 80 Strategie, 51 SpielBrett.th, 81 wert, 52 SpielBrett.h, 81 wert, 52	-	-
Stein, 47 mteam, 48 setSieg setFrei, 47 ream, 56 setFrei, 47 setStein setGeffangen, 47 Feld, 10 Stein, 46 SiH, 24 Stein, 46 ~SfH, 27 ziehenach, 47 bewerten, 27 stein SfH, 27 stein SfK, 28 STEIN_C, 88 Stein, 49 Strategie, 49 SfK, 31 KI, 19 Strategie, 51 strategie, 51 Sieg aZuege, 52 SpielBrett, 35 petWert, 52 getWert, 52 <t< td=""><td></td><td></td></t<>		
setSieg Team, 56 setStein Feld, 10 SfH, 24		mteam, 48
Team, 56 setStein Feld, 10 SfH, 24 SfH, 27 bewerten, 27 SfH, 27 SfK, 31 Stein Cpp, 87 SfK, 28 SfK, 31 Stein, 46 Stein Cpp, 87 SfK, 31 Stein, 76 Stein. Cp, 87 SfK, 31 Stein. A8 Stein. A8 Stein. Stein Stei		· ·
setStein setGeffangen, 47 Feld, 10 setOrt, 47 SfH, 24 Stein, 46 ~SfH, 27 ziehenach, 47 bewerten, 27 stein SfH, 27 stein SfH, 27 stein SfH, 27 stein SfH, 28 STEIN_C, 88 ~SfK, 31 Stein.h, 88 bewerten, 31 strat SfK, 31 Strategie, 49 SfK.cpp, 77 Strategie, 51 Sieg aZuege, 52 Team, 57 bewerten, 51 SpielBrett, 33 getWert, 52 Team, 57 bewerten, 51 SpielBrett, 36 getZuege, 52 Gimmension, 36 getWert, 52 getKuege, 51 h1, 52 getSchwarz, 35 h2, 52 getWeis, 35 h3, 52 initBrett, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett, 34 n2ug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SpielBrett.h, 81 wert, 52 SpielBrett.h, 81 wert,	· ·	
Feld, 10 SfH, 24		setGeffangen, 47
SfH, 24 Stein, 46 ~SfH, 27 ziehenach, 47 bewerten, 27 stein SfH, 27 stein SfH, 27 zug, 60 SfH, 1, 76 Stein.cpp, 87 SfK, 28 STEIN_C, 88 ~SfK, 31 stratt bewerten, 31 KI, 19 SfK.cpp, 77 Strategie, 49 Sieg aZuege, 52 Team, 57 bewerten, 51 SpielBrett, 33 brett, 52 ~SpielBrett, 35 getWert, 52 Brett, 36 getZuege, 52 dimmension, 36 getmZuege, 51 getFeld, 35 h1, 52 getWeis, 35 h1, 52 initBrett, 36 k, 52 schwarz, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SpielBrett.h, 81 wert, 52 Spieler Strategie.cpp, 89		setOrt, 47
~SfH, 27 ziehenach, 47 bewerten, 27 stein SfH, 27 stein SfH, 27 zug, 60 SfH, 76 Stein.cpp, 87 SfK, 28 STEIN_C, 88 ~SfK, 31 Stein.h, 88 bewerten, 31 kI, 19 SfK.cpp, 77 Strategie, 49 SfK.h, 78 ~Strategie, 51 Sieg aZuege, 52 Team, 57 bewerten, 51 SpielBrett, 33 getWert, 52 Brett, 36 getWert, 52 dimmension, 36 getZuege, 52 dimmension, 36 getmZuege, 52 getSchwarz, 35 h2, 52 getWeis, 35 h3, 52 initBrett, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SpielBrett.h, 81 wert, 52 SpielBrett.h, 81 wert, 52 SpielBrett.p, 89		Stein, 46
bewerten, 27 SfH, 27 SfH, 27 SfH, 27 SfH, 27 SfH, 27 SfH, 27 Stein.cpp, 75 SfH, 76 SfK, 28 STEIN_C, 88 Stein.h, 88 Strategie, 51 SpielBrett, 31 SpielBrett, 32 SpielBrett.pie.h, 81 SpielBrett.pie.h, 81 SpielBrett.pie.h, 81 SpielBrett.pie.h, 89		ziehenach, 47
SfH, 27 stein SfH.cpp, 75 zug, 60 SfH.h, 76 Stein.cpp, 87 SfK, 28 STEIN_C, 88 ~SfK, 31 Stein.h, 88 bewerten, 31 kI, 19 SfK.cpp, 77 Strategie, 49 SfK.h, 78 ~Strategie, 51 Sieg aZuege, 52 Team, 57 bewerten, 51 SpielBrett, 33 brett, 52 ~SpielBrett, 35 getWert, 52 Brett, 36 getZuege, 52 dimmension, 36 getmZuege, 51 getFeld, 35 h1, 52 getSchwarz, 35 h2, 52 getWeis, 35 h3, 52 initBrett, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 wert, 52 SpielBrett.h, 81 wert, 52 Spieler Strategie.cpp, 89		zuege, 48
SfH.cpp, 75 zug, 60 SfH.h, 76 Stein.cpp, 87 SfK, 28 STEIN_C, 88 ~SfK, 31 strat bewerten, 31 KI, 19 SfK.cpp, 77 Strategie, 49 SfK.h, 78 ~Strategie, 51 Sieg aZuege, 52 Team, 57 bewerten, 51 SpielBrett, 33 brett, 52 ~SpielBrett, 35 getWert, 52 Brett, 36 getZuege, 52 dimmension, 36 getmZuege, 51 getFeld, 35 h1, 52 getSchwarz, 35 h2, 52 getWeis, 35 h3, 52 initBrett, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 team, 52 Spieler Strategie.cpp, 89		stein
SfH.h, 76 Stein.cpp, 87 SfK, 28 STEIN_C, 88 ~SfK, 31 Stein.h, 88 bewerten, 31 Strat SfK, 31 KI, 19 SfK.cpp, 77 Strategie, 49 SfK.h, 78 ~Strategie, 51 Sieg aZuege, 52 Team, 57 bewerten, 51 SpielBrett, 33 brett, 52 ~SpielBrett, 36 getZuege, 52 dimmension, 36 getmzuege, 51 getFeld, 35 h1, 52 getSchwarz, 35 h2, 52 getWeis, 35 h3, 52 initBrett, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 team, 52 Spieler Strategie.cpp, 89		zug, <mark>60</mark>
SfK, 28 STEIN_C, 88 ~SfK, 31 Stein.h, 88 bewerten, 31 strat SfK, 31 KI, 19 SfK.cpp, 77 Strategie, 49 ~Sfk.h, 78 ~Strategie, 51 Sieg aZuege, 52 Team, 57 bewerten, 51 SpielBrett, 33 brett, 52 ~SpielBrett, 35 getWert, 52 getHel, 36 getZuege, 52 dimmension, 36 getmZuege, 51 getFeld, 35 h1, 52 getWeis, 35 h2, 52 initBrett, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 team, 52 Spieler Strategie.cpp, 89		Stein.cpp, 87
∼SfK, 31 Stein.h, 88 bewerten, 31 strat SfK, 31 KI, 19 SfK.cpp, 77 Strategie, 49 ∼SfK.h, 78 ∼Strategie, 51 Sieg aZuege, 52 Team, 57 bewerten, 51 SpielBrett, 33 brett, 52 ∞SpielBrett, 35 getWert, 52 Brett, 36 getZuege, 52 dimmension, 36 getmZuege, 51 getFeld, 35 h1, 52 getSchwarz, 35 h2, 52 getWeis, 35 h3, 52 initBrett, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 team, 52 SpielBrett.h, 81 wert, 52 Spieler Strategie.cpp, 89		STEIN_C, 88
bewerten, 31		Stein.h, 88
SfK.cpp, 77 Strategie, 49 SfK.h, 78 ~Strategie, 51 Sieg aZuege, 52 Team, 57 bewerten, 51 SpielBrett, 33 brett, 52 ~SpielBrett, 35 getWert, 52 Brett, 36 getZuege, 52 dimmension, 36 getmZuege, 51 getFeld, 35 h1, 52 getSchwarz, 35 h2, 52 getWeis, 35 h3, 52 initBrett, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 team, 52 SpielBrett.h, 81 wert, 52 Spieler Strategie.cpp, 89		
SfK.cpp, 77 Strategie, 49 SfK.h, 78 ~Strategie, 51 Sieg aZuege, 52 Team, 57 bewerten, 51 SpielBrett, 33 brett, 52 ~SpielBrett, 35 getWert, 52 Brett, 36 getZuege, 52 dimmension, 36 getmZuege, 51 getFeld, 35 h1, 52 getSchwarz, 35 h2, 52 getWeis, 35 h3, 52 initBrett, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 team, 52 SpielBrett.h, 81 wert, 52 Spieler Strategie.cpp, 89		KI, 19
SfK.h, 78 ~Strategie, 51 Sieg aZuege, 52 Team, 57 bewerten, 51 SpielBrett, 33 brett, 52 ~SpielBrett, 35 getWert, 52 Brett, 36 getZuege, 52 dimmension, 36 getmZuege, 51 getFeld, 35 h1, 52 getWeis, 35 h2, 52 initBrett, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 team, 52 SpielBrett.h, 81 wert, 52 Spieler Strategie.cpp, 89		
Sieg aZuege, 52 Team, 57 bewerten, 51 SpielBrett, 33 brett, 52 ~SpielBrett, 35 getWert, 52 Brett, 36 getZuege, 52 dimmension, 36 getmZuege, 51 getFeld, 35 h1, 52 getSchwarz, 35 h2, 52 getWeis, 35 h3, 52 initBrett, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 wert, 52 Spieler Strategie.cpp, 89	• • •	
Team, 57 bewerten, 51 SpielBrett, 33 brett, 52 ~SpielBrett, 35 getWert, 52 Brett, 36 getZuege, 52 dimmension, 36 getmZuege, 51 getFeld, 35 h1, 52 getSchwarz, 35 h2, 52 getWeis, 35 h3, 52 initBrett, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 team, 52 SpielBrett.h, 81 wert, 52 Spieler Strategie.cpp, 89		-
SpielBrett, 33 brett, 52 ~SpielBrett, 35 getWert, 52 Brett, 36 getZuege, 52 dimmension, 36 getmZuege, 51 getFeld, 35 h1, 52 getSchwarz, 35 h2, 52 getWeis, 35 h3, 52 initBrett, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 wert, 52 Spieler Strategie.cpp, 89	<u> </u>	
SpielBrett, 35 getWert, 52 Brett, 36 getZuege, 52 dimmension, 36 getmZuege, 51 getFeld, 35 h1, 52 getSchwarz, 35 h2, 52 getWeis, 35 h3, 52 initBrett, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 wert, 52 Spieler Strategie.cpp, 89		•
Brett, 36 getZuege, 52 dimmension, 36 getmZuege, 51 getFeld, 35 h1, 52 getSchwarz, 35 h2, 52 getWeis, 35 h3, 52 initBrett, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 team, 52 SpielBrett.h, 81 wert, 52 Spieler Strategie.cpp, 89	•	
dimmension, 36 getmZuege, 51 getFeld, 35 h1, 52 getSchwarz, 35 h2, 52 getWeis, 35 h3, 52 initBrett, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 team, 52 SpielBrett.h, 81 wert, 52 Spieler Strategie.cpp, 89	·	_
getFeld, 35 h1, 52 getSchwarz, 35 h2, 52 getWeis, 35 h3, 52 initBrett, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 team, 52 SpielBrett.h, 81 wert, 52 Spieler Strategie.cpp, 89		
getSchwarz, 35 getWeis, 35 initBrett, 36 schwarz, 36 SpielBrett, 34 weis, 36 SpielBrett.cpp, 80 SPIELBRETT_C, 81 SpielBrett.h, 81 SpielBrett.spp, 89 Strategie.cpp, 89		
getWeis, 35 initBrett, 36 schwarz, 36 schwarz, 36 SpielBrett, 34 weis, 36 SpielBrett.cpp, 80 SPIELBRETT_C, 81 SpielBrett.h, 81 SpielBrett.h, 81 Spieler Strategie.cpp, 89	_	
initBrett, 36 k, 52 schwarz, 36 mZuege, 52 SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 team, 52 SpielBrett.h, 81 wert, 52 Spieler Strategie.cpp, 89		
schwarz, 36 mZuege, 52 SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 team, 52 SpielBrett.h, 81 wert, 52 Spieler Strategie.cpp, 89		
SpielBrett, 34 nZug, 52 weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 team, 52 SpielBrett.h, 81 wert, 52 Spieler Strategie.cpp, 89		
weis, 36 nexZug, 52 SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 team, 52 SpielBrett.h, 81 wert, 52 Spieler Strategie.cpp, 89	•	-
SpielBrett.cpp, 80 Strategie, 51 SPIELBRETT_C, 81 team, 52 SpielBrett.h, 81 wert, 52 Spieler Strategie.cpp, 89	•	-
SPIELBRETT_C, 81 team, 52 SpielBrett.h, 81 wert, 52 Spieler Strategie.cpp, 89		-
SpielBrett.h, 81 wert, 52 Spieler Strategie.cpp, 89	•	
Spieler Strategie.cpp, 89		
•	•	
Gui, 12 Strategie.n, 90	•	- · · ·
	GUI, 12	Strategie.n, 90

100 INDEX

t			Stein, 47
·	KI, 19	zpos	
TEA	-		zug, 60
T	Team.cpp, 92	zu	CO
Tean		71100	zug, 60
	~Team, 55	zueg	Stein, 48
	brett, 56	7110	
	distanzen, 55	zug,	
	Farbe, 56 gegner, 57		operator<, 60 operator=, 60
	getBrett, 56		operator==, 60
	getFarbe, 56		stein, 60
	getGegner, 56		wert, 60
	getSieg, 56		zpos, 60
	getStein, 56		zu, 60
	helfer1, 57		zug, 60
	helfer2, 57	zug.l	n, <mark>95</mark>
	helfer3, 57		
	koenig, 57		
	setGegner, 56		
	setSieg, 56		
	Sieg, 57		
	Team, 55		
team			
_	Strategie, 52		
Tean	1.cpp, 92		
_	TEAM_C, 92		
lean	n.h, 92		
USE	B C		
OOL	User.cpp, 94		
User			
000.	\sim User, 58		
	Graphik, 58		
	User, 57		
User	.cpp, 94		
	USER_C, 94		
User	.h, 94		
weis			
	SpielBrett, 36		
wert	Otrata via 50		
	Strategie, 52		
	zug, 60		
Χ			
•	Possition, 24		
у			
	Possition, 24		
zeich	neAnleitung		
	GUI, 12		
zeich	nneSpielfeld		
: -'	GUI, 12		
zeicr	nneZug		
zioha	GUI, 12 enach		
	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		

Koenig, 22