BlockgamePrg4.0

1.5

Erzeugt von Doxygen 1.7.6.1

Don Mai 31 2012 15:04:50

Inhaltsverzeichnis

Kapitel 1

Blockgame PRG4

Autor

Dominik Breu, Nils Moh und Simon Scholl

1.1 Gedanken zum Entwurf

Die Spielsteine werden in Grundsteine (basicstones) zerlegt, da die Entfernung von -Linien sonst zu aufwändig wäre. Der aktuelle Spielstein und das Spielfeld mit bereits gefallenen Steinen werden getrennt gespeichert, um die Kollisionserkennung zu erleichtern. Dazu wird einfach für alle Grundsteine des Spielsteins getestet ob die vorgesehenen Positionen noch frei sind. Das Spielfeld speichert auch einen zwei Grundsteine breiten Rand, um Rotation und Kollisionen ohne Arrayzugriffsprobleme zu lösen.

Bei der internen Definition des Spielsteins stellt sich die Frage nach der Darstellung als Matrix der belegten Positionen oder als Liste der einzelnen Grundsteine. Witr haben uns für eine Liste von Grundsteinen entschieden, weil sie eine problemlose Nutzung einer Rotationsmatrix erlaubt und die Einzeldefinition der gedrehten Steine überflüssig macht.

Realisiert wurden die Spielsteine über eine Vererbung, d.h. Es gibt einen Komplexstein mit den notwendigen Methoden und Kindsteine welche jeweils eine eigene konkrete Form haben.

1.2 Bedienung

Das Spiel startet automatisch. Mit den Pfeiltasten wird das Spiel gesteuert.

Die Pfeil-nach-links-Taste verschiebt den Spielstein um eine Position nach links.

Die Pfeil-nach-unten-Taste erhöht die Fallgeschwindigkeit des Spielsteins.

Die Pfeil-nach-oben-Taste dreht den Spielstein um 90 Grad mit dem Uhrzeigersinn.

Die Taste F2 startet ein neues Spiel, falls das letzte beendet wurde. Den selben Effekt

hat die Auswahl des Menüpunktes "New" im Untermenü "Game".

Das Spiel wird beendet durch Klicken auf den X-Button in der oberen rechten Ecke.

1.3 Spielsteine

Die Spielsteine entsprechen den klassischen Tetrissteinen (vgl. http://de.wikipedia.org/wiki/Tetris). Zusätzlich gibt es noch einen Plus-förmigen Stein, einen planetenförmigen Stein mit Mond zur Demonstration der Rotationsfunktion und einen quadratischen disjunkten Stein zur Demonstration der Flexibilität unserer Steinimplementierung. Jeder Grundstein kann dabei eine eigene Farbe haben.

1.4 Kompatibilitätshinweis

Aus nicht nachvollziehbaren Gründen funktioniert die Darstellung der Highscoreliste mit dem Sun JDK 1.7 nicht immer. Mit Version 1.6 funktioniert die Darstellung hingegen tadellos unter Windows und Linux.

1.5 Warum sind bestimmte Funktionalitäten enthalten?

In erster Linie wurden die gestellten Anforderungen erfüllt. Darueber hinaus wurde eine musikalsche Untermalung eingebaut, weil sie essentiell für die Spielatmosphäre ist. Die flexible Spielsteindarstellung mit einzelnen Grundsteinfarben erlaubt große Freiheiten in der künstlerischen Gestaltung.

1.6 Auf welche Funktionalitäten haben sie warum verzichtet?

Auf die Implementierung eines Netzwerkmodus bzw. einer Mehrspielerunterstützung haben wir verzichtet, weil die nötige Entwicklungszeit nicht verfügbar war und der - Spielspass in keinem günstigen Verhältnis zum Aufwand steht.

1.7 Welche Erweiterungsmöglichkeiten sehen Sie?

Eine benutzersteuerbare Musikauswahl wäre relativ leicht umzusetzen, weil die - Abspielfunktionalität bereits vorhanden ist. Ein grafischer Editor zum Erstellen eigener Spielsteine würde die Anpassbarkeit für den Benutzer fördern und die Flexibilität der Steindarstellung mehr zur Geltung bringen.

1.8 Verbesserungen beim nächsten Projekt

Bei einer Wiederholung des Projektes würden wir die zeitraubenden Versuchs die - Canvasklasse aus der Übung zu verwenden weglassen. Sie unterstützt in der vorliegenden Form nur eine Darstellungsfläche und scheitert bei der Anzeige eines zusätzlichen Vorschaufensters.

Kapitel 2

Verzeichnis der Namensbereiche

2.1 Pakete

Hier folgen die Pakete mit einer Kurzbeschreibung (wenn verfügbar):

Kapitel 3

Klassen-Verzeichnis

3.1 Klassenhierarchie

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

Kapitel 4

Klassen-Verzeichnis

4.1 Auflistung der Klassen

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

Kapitel 5

Datei-Verzeichnis

5.1 Auflistung der Dateien

Hier folgt die Aufzählung aller Dateien mit einer Kurzbeschreibung:

Kapitel 6

Dokumentation der Namensbereiche

6.1 Paket controller

Klassen

- class CHighScoreEntry
- class CHighScores

  enthält einen Vektor mit den Einträgen
- class JHighscoreFileIO
- class Logic

6.2 Paket game

Klassen

- class Config
- class Game

6.3 Paket gui

Klassen

- class AdvancedGui

6.4 Paket stone

Klassen

- class Basicstone
- class Complexstone
- class IStone
- class JStone
- class LStone
- class MoonStone
- class NStone
- class OStone
- class PlusStone
- class SStone
- class TStone
- class ZStone

6.5 Paket world

Klassen

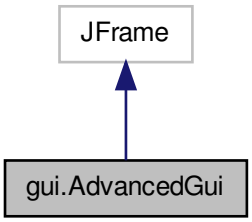
- class Spielfeld

Kapitel 7

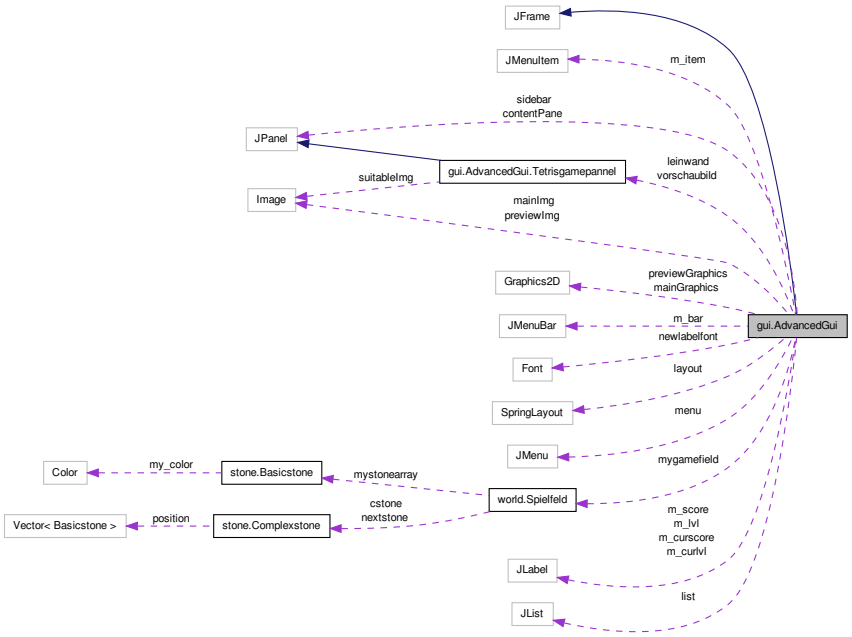
Klassen-Dokumentation

7.1 gui.AdvancedGui Klassenreferenz

Klassendiagramm für gui.AdvancedGui:



Zusammengehörigkeiten von gui.AdvancedGui:



Klassen

- class Tetrisgamepannel

Öffentliche Methoden

- AdvancedGui (Spielfeld _mygamefield, Logic _mylogic, KeyListener _KeyListener, ActionListener _ActionListner)
- void init ()
- synchronized void zeichnen ()
- synchronized void zeichnen_preview ()
- synchronized void highscoredialog ()

Öffentliche Attribute

- JMenuItem m_item

  Menüeintrag in.

Geschützte Attribute

- Spielfeld mygamefield

  Verweiß auf die worldkomponente innherhalb des MVC patern.
- Logic mylogic

  Verweiß auf die Controler komponte innerhalb des MVC pattern.

Private Attribute

- Tetrisgamepannel leinwand

  Das spielfeld auf dem die setine gezeichnet werden linkes element.
- Tetrisgamepannel vorschaubild

  Das voirschaubild des nächstensteins rechts obben mitte.
- JPanel sidebar

  vater panel für
- JPanel contentPane

  vater pannel für alle JPanels und elemente innerhalb von AdvancedGui
- JLabel m_lvl

  level anzeige
- JLabel m_curlvl

  aktueler level
- JLabel m_score

  punkte anzeige
- JLabel m_curscore

  aktuelle punktzahl
- Graphics2D mainGraphics

  zeichenobjekt für
- Graphics2D previewGraphics

  zeichenobjekt für
- Image previewImg

  zeichenfläche für
- Image mainImg

  zeichenfläche für
- JList list = new JList()

  liste der highscore auf datei und aktuel higscores
- JMenu menu

  Menü für unser spiel.
- JMenuBar m_bar

  Menüleiste für unser Spiel.
- SpringLayout layout

  Das Layout für unser Sideannel.
- Font newlabelfont

  Der font für unser sidepannel.
- CHighScores hightemp

  die higscoreliste für unser Spiel

Statische private Attribute

- static final long serialVersionUID = 1L

  muss halt so
- static int ratiox = 20

  größe des steins in x richtung
- static int ratioy = 20

  größe des steins in y richtung
- static int spielfeldbreite = game.Config.s_FieldWidth

  breite des spielfeldes
- static int spielfeldhoehe = spielfeldbreite ∗ 2

  höhe des Spielfeldes
- static int fensterbreite = (spielfeldbreite ∗ ratiox)+302

  breite des fensters
- static int fensterhoehe = (spielfeldhoehe ∗ ratioy) +25

  höhe des fesnters

7.1.1 Ausführliche Beschreibung

Implementierung des Viewobjektes nach MVC Pattern. Enthält alle Elemente die dafür notwendig sind, eine solche View zu erstellen.

Definiert in Zeile 36 der Datei AdvancedGui.java.

7.1.2 Beschreibung der Konstruktoren und Destruktoren

7.1.2.1 gui.AdvancedGui.AdvancedGui (Spielfeld *_mygamefield,* Logic *_mylogic,* KeyListener *_KeyListener,* ActionListener *_ActionListner*)

Erstellt ein JFrame der das Tetrisspiel als Fenster im Betriebsystem repräsentiert.

Parameter

_mygamefield	Verweis auf Model im MVC-Pattern
_mylogic	Verweise auf Control im MVC-Pattern
_KeyListener	Anzuhaengender Listener, damit GUI-Ereignisse den Spielthread erreichen koennen, darf null sein
_ActionListner	Anzuhaengender Listener, damit GUI-Ereignisse den Spielthread erreichen koennen, darf null sein

Definiert in Zeile 80 der Datei AdvancedGui.java.

Benutzt gui.AdvancedGui.contentPane, gui.AdvancedGui.fensterbreite, gui.AdvancedGui.fensterhoehe, gui.AdvancedGui.hightemp, gui.AdvancedGui.leinwand, gui.AdvancedGui.list, gui.AdvancedGui.m_curlvl, gui.AdvancedGui.m_curscore, gui.AdvancedGui.m_lvl, gui.AdvancedGui.m_score, gui.AdvancedGui.mygamefield, gui.-

AdvancedGui.mylogic, gui.AdvancedGui.ratiox, gui.AdvancedGui.ratioy, gui.AdvancedGui.sidebar, gui.AdvancedGui.spielfeldbreite, gui.AdvancedGui.spielfeldhoehe und gui.AdvancedGui.vorschaubild.

7.1.3 Dokumentation der Elementfunktionen

7.1.3.1 synchronized void gui.AdvancedGui.highscoredialog ( )

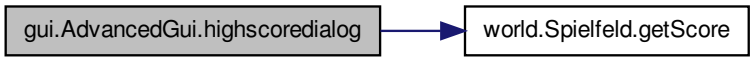
Synchronisierte Funktion um nach Spielende die Highscores anzuzeigen und bei - Erreichen der Highscoreliste den Spielernamen abzufragen

Definiert in Zeile 277 der Datei AdvancedGui.java.

Benutzt world.Spielfeld.getScore(), gui.AdvancedGui.list, gui.AdvancedGui.mygamefield und gui.AdvancedGui.mylogic.

Wird benutzt von game.Game.run().

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



Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:

7.1.3.2 void gui.AdvancedGui.init ( )

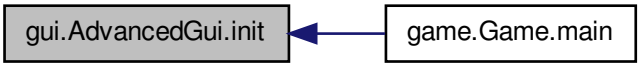
Diese Funktion dient dazu alle Grafiken und Bilder zu erzeugen, die wir brauchen um das Spiel auf den Bildschirm bringen zu können. Dies kann nicht im Konstruktor passieren, da die objekte schon alle fertig erstellt sein müssen.

Definiert in Zeile 196 der Datei AdvancedGui.java.

Benutzt gui.AdvancedGui.fensterbreite, gui.AdvancedGui.fensterhoehe, gui.AdvancedGui.leinwand, gui.AdvancedGui.mainGraphics, gui.AdvancedGui.mainImg, gui.AdvancedGui.previewGraphics, gui.AdvancedGui.previewImg, gui.AdvancedGui.Tetrisgamepannel.suitableImg und gui.AdvancedGui.vorschaubild.

Wird benutzt von game.Game.main().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.1.3.3 synchronized void gui.AdvancedGui.zeichnen ( )

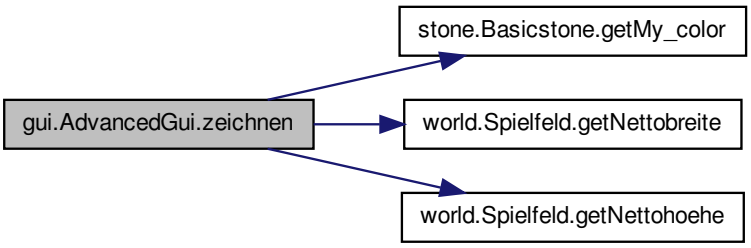
Synchronisierte Funktion um das Spielfeld und den momentanen Spielstein auf unsere Spielfeldzeichenflaeche zu schreiben

Definiert in Zeile 209 der Datei AdvancedGui.java.

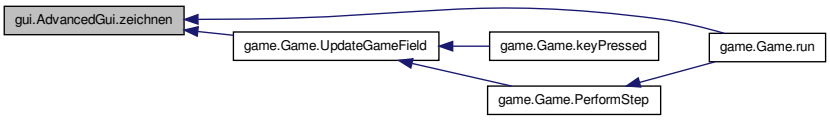
Benutzt world.Spielfeld.cstone, stone.Basicstone.getMy_color(), world.Spielfeld.getNettobreite(), world.Spielfeld.getNettohoehe(), world.Spielfeld.Level, gui.AdvancedGui.m_curlvl, gui.AdvancedGui.m_curscore, gui.AdvancedGui.mainGraphics, gui.AdvancedGui.mygamefield, world.Spielfeld.mystonearray, stone.Complexstone.position, gui.AdvancedGui.ratiox, gui.AdvancedGui.ratioy und world.Spielfeld.score.

Wird benutzt von game.Game.run() und game.Game.UpdateGameField().

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



Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.1.3.4 synchronized void gui.AdvancedGui.zeichnen_preview ( )

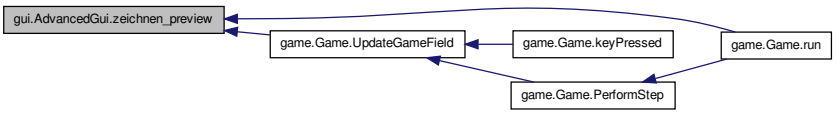
Synchronisierte Funktion, die den folgenden Spielstein auf unsere Zeichenflaeche (-Vorschaufeld) zeichnet

Definiert in Zeile 254 der Datei AdvancedGui.java.

Benutzt gui.AdvancedGui.mygamefield, world.Spielfeld.nextstone, stone.Complexstone.-position, gui.AdvancedGui.previewGraphics, gui.AdvancedGui.ratiox, gui.Advanced-Gui.ratioy, stone.Complexstone.xpos und stone.Complexstone.ypos.

Wird benutzt von game.Game.run() und game.Game.UpdateGameField().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.1.4 Dokumentation der Datenelemente

7.1.4.1 JPanel gui.AdvancedGui.contentPane [private]

vater pannel für alle JPanels und elemente innerhalb von AdvancedGui

Definiert in Zeile 43 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.AdvancedGui().

7.1.4.2 int gui.AdvancedGui.fensterbreite = (spielfeldbreite * ratiox)+302 [static, private]

breite des fensters

Definiert in Zeile 61 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.AdvancedGui() und gui.AdvancedGui.init().

7.1.4.3 int gui.AdvancedGui.fensterhoehe = (spielfeldhoehe * ratioy) +25 [static, private]

höhe des fesnters

Definiert in Zeile 62 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.AdvancedGui() und gui.AdvancedGui.init().

7.1.4.4 CHighScores gui.AdvancedGui.hightemp [private]

die higscoreliste für unser Spiel

Definiert in Zeile 70 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.AdvancedGui().

7.1.4.5 SpringLayout gui.AdvancedGui.layout [private]

Das Layout für unser Sideannel.

Definiert in Zeile 68 der Datei AdvancedGui.java.

7.1.4.6 Tetrisgamepannel gui.AdvancedGui.leinwand [private]

Das spielfeld auf dem die setine gezeichnet werden linkes element.

Definiert in Zeile 40 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.AdvancedGui() und gui.AdvancedGui.init().

7.1.4.7 JList gui.AdvancedGui.list = new JList() [private]

liste der highscore auf datei und aktuel higscores

Definiert in Zeile 64 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.AdvancedGui() und gui.AdvancedGui.highscoredialog().

7.1.4.8 JMenuBar gui.AdvancedGui.m_bar [private]

Menüleiste für unser Spiel.

Definiert in Zeile 66 der Datei AdvancedGui.java.

7.1.4.9 JLabel gui.AdvancedGui.m_curlvl [private]

aktueler level

Definiert in Zeile 47 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.AdvancedGui() und gui.AdvancedGui.zeichnen().

7.1.4.10 JLabel gui.AdvancedGui.m_curscore [private]

aktuelle punktzahl

Definiert in Zeile 49 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.AdvancedGui() und gui.AdvancedGui.zeichnen().

7.1.4.11 JMenuItem gui.AdvancedGui.m_item

Menüeintrag in.

Siehe auch

menu

Definiert in Zeile 67 der Datei AdvancedGui.java.

Wird benutzt von game.Game.main().

7.1.4.12 JLabel gui.AdvancedGui.m_lvl [private]

level anzeige

Definiert in Zeile 46 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.AdvancedGui().

7.1.4.13 JLabel gui.AdvancedGui.m_score [private]

punkte anzeige

Definiert in Zeile 48 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.AdvancedGui().

7.1.4.14 Graphics2D gui.AdvancedGui.mainGraphics [private]

zeichenobjekt für

Siehe auch

leinwand

Definiert in Zeile 51 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.init() und gui.AdvancedGui.zeichnen().

7.1.4.15 Image gui.AdvancedGui.mainImg [private]

zeichenfläche für

Siehe auch

previewGraphics

Definiert in Zeile 55 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.init().

7.1.4.16 JMenu gui.AdvancedGui.menu [private]

Menü für unser spiel.

Definiert in Zeile 65 der Datei AdvancedGui.java.

7.1.4.17 Spielfeld gui.AdvancedGui.mygamefield [protected]

Verweiß auf die worldkomponente innherhalb des MVC patern.

Definiert in Zeile 44 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.AdvancedGui(), gui.AdvancedGui.highscoredialog(), gui.AdvancedGui.zeichnen() und gui.AdvancedGui.zeichnen_preview().

7.1.4.18 Logic gui.AdvancedGui.mylogic [protected]

Verweiß auf die Controler komponte innerhalb des MVC pattern.

Definiert in Zeile 45 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.AdvancedGui() und gui.AdvancedGui.highscoredialog().

7.1.4.19 Font gui.AdvancedGui.newlabelfont [private]

Der font für unser sidepannel.

Definiert in Zeile 69 der Datei AdvancedGui.java.

7.1.4.20 Graphics2D gui.AdvancedGui.previewGraphics [private]

zeichenobjekt für

Siehe auch

vorschaubild

Definiert in Zeile 52 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.init() und gui.AdvancedGui.zeichnen_preview().

7.1.4.21 Image gui.AdvancedGui.previewImg [private]

zeichenfläche für

Siehe auch

mainGrahics

Definiert in Zeile 54 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.init().

7.1.4.22 int gui.AdvancedGui.ratiox = 20 [static, private]

größe des steins in x richtung

Definiert in Zeile 57 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.AdvancedGui(), gui.AdvancedGui.zeichnen() und gui.AdvancedGui.zeichnen_preview().

7.1.4.23 int gui.AdvancedGui.ratioy = 20 [static, private]

größe des steins in y richtung

Definiert in Zeile 58 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.AdvancedGui(), gui.AdvancedGui.zeichnen() und gui.AdvancedGui.zeichnen_preview().

7.1.4.24 final long gui.AdvancedGui.serialVersionUID = 1L [static, private]

muss halt so

Definiert in Zeile 39 der Datei AdvancedGui.java.

7.1.4.25 JPanel gui.AdvancedGui.sidebar [private]

vater panel für

Siehe auch

vorschaulbild und für die level und higscore anzeige

Definiert in Zeile 42 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.AdvancedGui().

7.1.4.26 int gui.AdvancedGui.spielfeldbreite = game.Config.s_FieldWidth [static, private]

breite des spielfeldes

Definiert in Zeile 59 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.AdvancedGui().

7.1.4.27 int gui.AdvancedGui.spielfeldhoehe = spielfeldbreite * 2 [static, private]

höhe des Spielfeldes

Definiert in Zeile 60 der Datei AdvancedGui.java.

Wird benutzt von gui.AdvancedGui.AdvancedGui().

7.1.4.28 Tetrisgamepannel gui.AdvancedGui.vorschaubild [private]

Das voirschaubild des nächstensteins rechts obben mitte.

Definiert in Zeile 41 der Datei AdvancedGui.java.

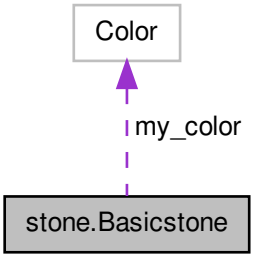
Wird benutzt von gui.AdvancedGui.AdvancedGui() und gui.AdvancedGui.init().

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- src/gui/AdvancedGui.java

7.2 stone.Basicstone Klassenreferenz

Zusammengehörigkeiten von stone.Basicstone:



Öffentliche Methoden

- Basicstone (Color my_color, int _xpos, int _ypos)
- Basicstone (Basicstone element)
- Color getMy_color ()
- void setMy_color (Color my_color)
- boolean isat (int x, int y)

Öffentliche Attribute

- int xpos = 0
- int ypos = 0
- boolean ispartoffullline
- boolean isdirty

Private Attribute

- Color my_color = new Color(255, 255, 255)

7.2.1 Ausführliche Beschreibung

einzelner Grundstein (Grundquadrat) des Spielfeldes

Definiert in Zeile 8 der Datei Basicstone.java.

7.2.2 Beschreibung der Konstruktoren und Destruktoren

7.2.2.1 stone.Basicstone.Basicstone (Color *my_color,* int *_xpos,* int *_ypos*)

Konstuktor des atomaren Steins der Grundlage für alle Steine ist

Parameter

my_color	farbe des steins
_xpos	x position
_ypos	y position

Definiert in Zeile 38 der Datei Basicstone.java.

Benutzt stone.Basicstone.isdirty, stone.Basicstone.ispartoffullline, stone.Basicstone.-setMy_color(), stone.Basicstone.xpos und stone.Basicstone.ypos.

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

7.2.2.2 stone.Basicstone.Basicstone (Basicstone *element*)

Kopierkonstruktor für den Grundstein

Parameter

existierender	Grundstein der zu kopieren ist

Definiert in Zeile 51 der Datei Basicstone.java.

Benutzt stone.Basicstone.ispartoffullline, stone.Basicstone.my_color, stone.Basicstone.-xpos und stone.Basicstone.ypos.

7.2.3 Dokumentation der Elementfunktionen

7.2.3.1 Color stone.Basicstone.getMy_color ( )

Gibt die Farbe des Grundsteins zurück

Rückgabe

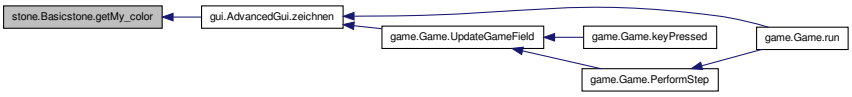
Farbe des steins

Definiert in Zeile 63 der Datei Basicstone.java.

Benutzt stone.Basicstone.my_color.

Wird benutzt von gui.AdvancedGui.zeichnen().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.2.3.2 boolean stone.Basicstone.isat (int *x*, int *y*)

Gibt zurück, ob ein Stein an der gegebenen Position vorhanden ist

Parameter

x	X Position
y	Y Position

Rückgabe

false falls der Grundstein nicht an diese Stelle liegt, true false der Grundstein die gegebenen Koordinaten belegt

Definiert in Zeile 86 der Datei Basicstone.java.

Benutzt stone.Basicstone.xpos und stone.Basicstone.ypos.

7.2.3.3 void stone.Basicstone.setMy_color (Color *my_color*)

Setzt die Farbe des Grundsteins

Parameter

my_color	zu setzende Farbe des Steins

Definiert in Zeile 75 der Datei Basicstone.java.

Benutzt stone.Basicstone.my_color.

Wird benutzt von stone.Basicstone.Basicstone() und world.Spielfeld.recolorlines().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:

7.2.4 Dokumentation der Datenelemente

7.2.4.1 boolean stone.Basicstone.isdirty

Flag ob der Stein "dirty" ist

Definiert in Zeile 30 der Datei Basicstone.java.

Wird benutzt von stone.Basicstone.Basicstone().

7.2.4.2 boolean stone.Basicstone.ispartoffullline

Flag ob der Stein teil einer vollen Linie ist

Definiert in Zeile 26 der Datei Basicstone.java.

Wird benutzt von stone.Basicstone.Basicstone() und world.Spielfeld.recolorlines().

7.2.4.3 Color stone.Basicstone.my_color = new Color(255, 255, 255) [private]

Farbe des Grundsteins

Definiert in Zeile 14 der Datei Basicstone.java.

Wird benutzt von stone.Basicstone.Basicstone(), stone.Basicstone.getMy_color() und stone.Basicstone.setMy_color().

7.2.4.4 int stone.Basicstone.xpos = 0

Position auf der X ebene des spielfelds

Definiert in Zeile 18 der Datei Basicstone.java.

Wird benutzt von stone.Basicstone.Basicstone() und stone.Basicstone.isat().

7.2.4.5 int stone.Basicstone.ypos = 0

Position auf der Y ebene des spielfelds

Definiert in Zeile 22 der Datei Basicstone.java.

Wird benutzt von stone.Basicstone.Basicstone() und stone.Basicstone.isat().

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- src/stone/Basicstone.java

7.3 controller.CHighScoreEntry Klassenreferenz

Öffentliche Methoden

- int getScore ()
- CHighScoreEntry (String name_, int score_)
- String toString ()

Private Attribute

- String name
- int score

7.3.1 Ausführliche Beschreibung

Ein Datensatz der Highscores aus Spielername und Punktzahl

Definiert in Zeile 8 der Datei CHighScoreEntry.java.

7.3.2 Beschreibung der Konstruktoren und Destruktoren

7.3.2.1 controller.CHighScoreEntry.CHighScoreEntry (String *name_*, int *score_*)

Erzeugt einen Highscoreeintrag aus Name und zugehöriger Punktzahl

Parameter

*name_*	Name der Spielerin oder des Spielers
*score_*	Die erreichte Punktzahl

Definiert in Zeile 22 der Datei CHighScoreEntry.java.

7.3.3 Dokumentation der Elementfunktionen

7.3.3.1 int controller.CHighScoreEntry.getScore ( )

Definiert in Zeile 13 der Datei CHighScoreEntry.java.

7.3.3.2 String controller.CHighScoreEntry.toString ( )

Definiert in Zeile 28 der Datei CHighScoreEntry.java.

7.3.4 Dokumentation der Datenelemente

7.3.4.1 String controller.CHighScoreEntry.name [private]

Definiert in Zeile 10 der Datei CHighScoreEntry.java.

7.3.4.2 int controller.CHighScoreEntry.score [private]

Definiert in Zeile 11 der Datei CHighScoreEntry.java.

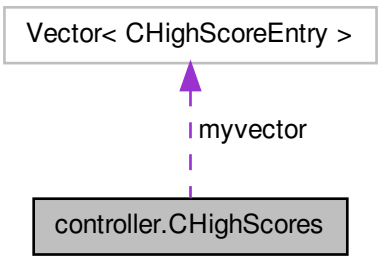
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- src/controller/CHighScoreEntry.java

7.4 controller.CHighScores Klassenreferenz

enthält einen Vektor mit den Einträgen

Zusammengehörigkeiten von controller.CHighScores:



Öffentliche Methoden

- Vector< CHighScoreEntry > getHighScoreVector ()
- CHighScores ()
- int getLowestScore ()
- void enterScore (String s, int newscore)

Private Attribute

- int MAXENTRIES
- int numentries
- int lowestscore
- Vector< CHighScoreEntry > myvector = new Vector<CHighScoreEntry>()

7.4.1 Ausführliche Beschreibung

enthält einen Vektor mit den Einträgen

Repräsentiert die vorhandenen Highscores

Definiert in Zeile 12 der Datei CHighScores.java.

7.4.2 Beschreibung der Konstruktoren und Destruktoren

7.4.2.1 controller.CHighScores.CHighScores ( )

Definiert in Zeile 24 der Datei CHighScores.java.

7.4.3 Dokumentation der Elementfunktionen

7.4.3.1 void controller.CHighScores.enterScore (String *s*, int *newscore*)

Trägt Datensatz in die Highscoreliste ein, falls die Daten eine Platzierung ergeben

Parameter

s	Name des Spielers oder der Spielerin
newscore	Punktzahl

Definiert in Zeile 55 der Datei CHighScores.java.

7.4.3.2 Vector<CHighScoreEntry> controller.CHighScores.getHighScoreVector ( )

Definiert in Zeile 19 der Datei CHighScores.java.

7.4.3.3 int controller.CHighScores.getLowestScore ( )

gibt die kleinste Punktzahl zurück, die für einen Highscoreeintrag qualifiziert

Rückgabe

kleinste Punktzahl

Definiert in Zeile 45 der Datei CHighScores.java.

7.4.4 Dokumentation der Datenelemente

7.4.4.1 int controller.CHighScores.lowestscore [private]

Definiert in Zeile 16 der Datei CHighScores.java.

7.4.4.2 int controller.CHighScores.MAXENTRIES [private]

Definiert in Zeile 14 der Datei CHighScores.java.

7.4.4.3 Vector<CHighScoreEntry> controller.CHighScores.myvector = new Vector<CHighScoreEntry>() [private]

Definiert in Zeile 17 der Datei CHighScores.java.

7.4.4.4 int controller.CHighScores.numentries [private]

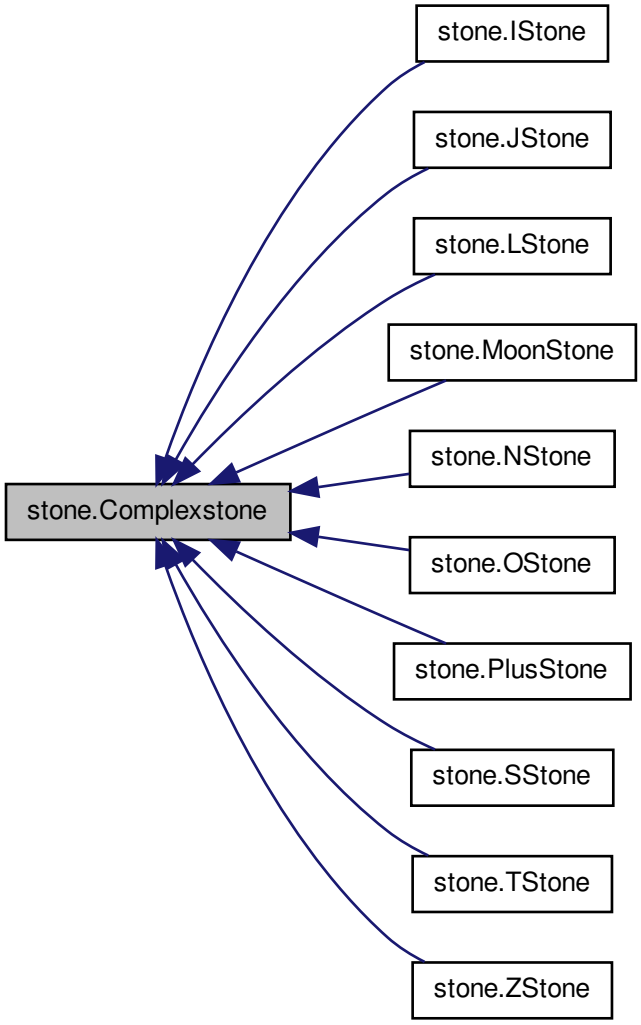
Definiert in Zeile 15 der Datei CHighScores.java.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

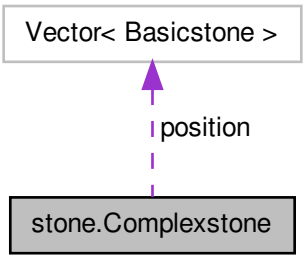
- src/controller/CHighScores.java

7.5 stone.Complexstone Klassenreferenz

Klassendiagramm für stone.Complexstone:



Zusammengehörigkeiten von stone.Complexstone:



Öffentliche Methoden

- Complexstone (int xpos_, int ypos_)
- Complexstone (Complexstone cstone)
- int howManyRectanglesUp ()
- int getComplexstonepartatxandypos (int x, int y)
- void down ()
- void up ()
- void left ()
- void right ()
- void rotate ()
- String toString ()

Öffentliche Attribute

- Vector< Basicstone > position
- int xpos
- int ypos

Private Attribute

- int hmru

7.5.1 Ausführliche Beschreibung

Die Basisklasse für einen komplexen Spielstein . Dieser setzt sich aus mehreren - Grundsteinen Basicstone zusammen, welche im Vector position gespeichet werden. Konkrete Spielsteinformen erben davon.

Definiert in Zeile 11 der Datei Complexstone.java.

7.5.2 Beschreibung der Konstruktoren und Destruktoren

7.5.2.1 stone.Complexstone.Complexstone (int *xpos_*, int *ypos_*)

Konstrukter für eine Spielstein

Parameter

*xpos_*	X-Position des Spielsteins
*ypos_*	Y-Position des Spielsteins

Definiert in Zeile 18 der Datei Complexstone.java.

Benutzt stone.Complexstone.hmru, stone.Complexstone.position, stone.Complexstone.-xpos und stone.Complexstone.ypos.

7.5.2.2 stone.Complexstone.Complexstone (Complexstone *cstone*)

Kopier Konsturktor für den Spielstein der aus Basicstone zusammen gestzt wird

Parameter

cstone	Stein der zu Kopieren ist

Definiert in Zeile 29 der Datei Complexstone.java.

Benutzt stone.Complexstone.position, stone.Complexstone.xpos und stone.Complexstone.-ypos.

7.5.3 Dokumentation der Elementfunktionen

7.5.3.1 void stone.Complexstone.down ( )

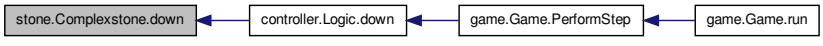
Methode um den Stein ungeprüft um eine Position nach unten zu bewegen

Definiert in Zeile 94 der Datei Complexstone.java.

Benutzt stone.Complexstone.position und stone.Complexstone.ypos.

Wird benutzt von controller.Logic.down().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.5.3.2 int stone.Complexstone.getComplexstonepartatxandypos (int *x*, int *y*)

Liefert die Rotkomponente des Steins an der übergebenen Position

Parameter

x	X-Position
y	Y-position

Rückgabe

liefert Rotkomponente des Grundsteins oder -1 falls da keiner an der gegeben - Stelle steht

Definiert in Zeile 78 der Datei Complexstone.java.

Benutzt stone.Complexstone.position.

Wird benutzt von world.Spielfeld.toString().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:

7.5.3.3 int stone.Complexstone.howManyRectanglesUp ( )

Methode die die maximaleGröße vom Drehpunkt nach oben berechnet

Rückgabe

Größe nach oben vom Drehpunkt aus

Definiert in Zeile 60 der Datei Complexstone.java.

Benutzt stone.Complexstone.hmru, stone.Complexstone.position und stone.-Complexstone.ypos.

7.5.3.4 void stone.Complexstone.left ( )

Methode um den Stein ungeprüft um eine Position nach links zu bewegen

Definiert in Zeile 116 der Datei Complexstone.java.

Benutzt stone.Complexstone.position und stone.Complexstone.xpos.

Wird benutzt von controller.Logic.left().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:

7.5.3.5 void stone.Complexstone.right ( )

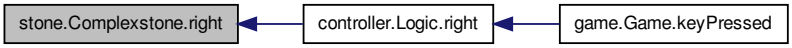
Methode um den Stein ungeprüft um eine Position nach rechts zu bewegen

Definiert in Zeile 127 der Datei Complexstone.java.

Benutzt stone.Complexstone.position und stone.Complexstone.xpos.

Wird benutzt von controller.Logic.right().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.5.3.6 void stone.Complexstone.rotate ( )

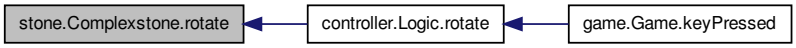
Methode um den Stein um 90 Grad um seine rotationsachse zu drehen

Definiert in Zeile 138 der Datei Complexstone.java.

Benutzt stone.Complexstone.position, stone.Complexstone.xpos und stone.Complexstone.-ypos.

Wird benutzt von controller.Logic.rotate().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.5.3.7 String stone.Complexstone.toString ( )

Generiert einen String aus dem Complexstein zu Testzwecken

Definiert in Zeile 153 der Datei Complexstone.java.

Benutzt stone.Complexstone.position.

7.5.3.8 void stone.Complexstone.up ( )

Methode um den Stein ungeprüft eine Position nach oben zu bewegen

Definiert in Zeile 105 der Datei Complexstone.java.

Benutzt stone.Complexstone.position und stone.Complexstone.ypos.

7.5.4 Dokumentation der Datenelemente

7.5.4.1 int stone.Complexstone.hmru [private]

Maximale Größe des Steins nach oben vom Drehpunkt aus

Definiert in Zeile 55 der Datei Complexstone.java.

Wird benutzt von stone.Complexstone.Complexstone() und stone.Complexstone.howManyRectanglesUp().

7.5.4.2 Vector<Basicstone> stone.Complexstone.position

Speicher für die Grundsteine die den Spielstein bilden

Definiert in Zeile 43 der Datei Complexstone.java.

Wird benutzt von stone.Complexstone.Complexstone(), stone.Complexstone.down(), controller.Logic.down(), stone.Complexstone.getComplexstonepartatxandypos(), stone.Complexstone.howManyRectanglesUp(), stone.IStone.IStone(), stone.JStone.JStone(), stone.Complexstone.left(), stone.LStone.LStone(), world.Spielfeld.maymove(), stone.MoonStone.MoonStone(), stone.NStone.NStone(), stone.OStone.OStone(), stone.PlusStone.PlusStone(), stone.Complexstone.right(), stone.Complexstone.rotate(), stone.SStone.SStone(), stone.Complexstone.toString(), stone.TStone.TStone(), stone.Complexstone.up(), gui.AdvancedGui.zeichnen(), gui.AdvancedGui.zeichnen_preview() und stone.ZStone.ZStone().

7.5.4.3 int stone.Complexstone.xpos

X-Position des Steins

Definiert in Zeile 47 der Datei Complexstone.java.

Wird benutzt von stone.Complexstone.Complexstone(), stone.IStone.IStone(), stone.JStone.JStone(), stone.Complexstone.left(), stone.LStone.LStone(), stone.MoonStone.-

MoonStone(), stone.NStone.NStone(), stone.OStone.OStone(), stone.PlusStone.PlusStone(), stone.Complexstone.right(), stone.Complexstone.rotate(), stone.SStone.SStone(), stone.TStone.TStone(), gui.AdvancedGui.zeichnen_preview() und stone.ZStone.ZStone().

7.5.4.4 int stone.Complexstone.ypos

Y- Position des Steins

Definiert in Zeile 51 der Datei Complexstone.java.

Wird benutzt von stone.Complexstone.Complexstone(), stone.Complexstone.down(), stone.Complexstone.howManyRectanglesUp(), stone.IStone.IStone(), stone.JStone.JStone(), stone.LStone.LStone(), stone.MoonStone.MoonStone(), stone.NStone.NStone(), stone.OStone.OStone(), stone.PlusStone.PlusStone(), stone.Complexstone.rotate(), stone.SStone.SStone(), stone.TStone.TStone(), stone.Complexstone.up(), gui.AdvancedGui.zeichnen_preview() und stone.ZStone.ZStone().

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- src/stone/Complexstone.java

7.6 game.Config Klassenreferenz

Öffentliche, statische Methoden

- static int GetFieldHeight ()
- static int GetFieldWidth ()
- static int Getadjustedstepwaittime (int lvl)

Statische öffentliche Attribute

- static int s_FieldWidth = 20
- static int s_linesperlevel = 1

Statische private Attribute

- static int s_FieldHeight = s_FieldWidth ∗ 2

7.6.1 Ausführliche Beschreibung

Klasse mit allen Konstanten, damit diese an einer Stelle verändert werden können

Definiert in Zeile 7 der Datei Config.java.

7.6.2 Dokumentation der Elementfunktionen

7.6.2.1 static int game.Config.Getadjustedstepwaittime (int *lvl*) [static]

Funktion die je nach Level eine unterschiedliche Wartezeit bis der Stein eine Position eine Position weiterfällt

Parameter

lvl	Level für das die Wartezeit gesucht wird

Rückgabe

Wartezeit in Millisekunden

Definiert in Zeile 40 der Datei Config.java.

Wird benutzt von game.Game.GetStepWaitTime().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:

7.6.2.2 static int game.Config.GetFieldHeight ( ) [static]

Funktion die Höhe des Spielfeldes in Grundsteinen zurückliefert

Rückgabe

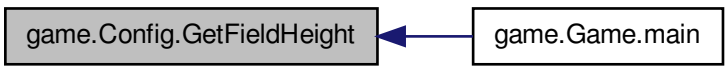
Höhe des spielfeldes

Definiert in Zeile 18 der Datei Config.java.

Benutzt game.Config.s_FieldHeight.

Wird benutzt von game.Game.main().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.6.2.3 static int game.Config.GetFieldWidth ( ) [static]

Funktion die die Breite des Spielfeldes in Grundsteinen zurückliefert

Rückgabe

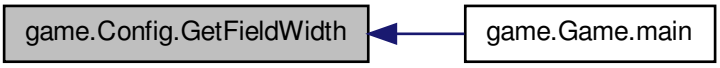
Breite des spielfelds

Definiert in Zeile 27 der Datei Config.java.

Benutzt game.Config.s_FieldWidth.

Wird benutzt von game.Game.main().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.6.3 Dokumentation der Datenelemente

7.6.3.1 int game.Config.s_FieldHeight = s_FieldWidth * 2 [static, private]

Definiert in Zeile 10 der Datei Config.java.

Wird benutzt von game.Config.GetFieldHeight().

7.6.3.2 int game.Config.s_FieldWidth = 20 [static]

Definiert in Zeile 9 der Datei Config.java.

Wird benutzt von game.Config.GetFieldWidth().

7.6.3.3 int game.Config.s_linesperlevel = 1 [static]

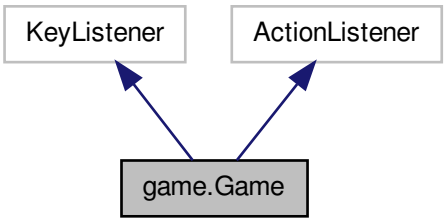
Definiert in Zeile 12 der Datei Config.java.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

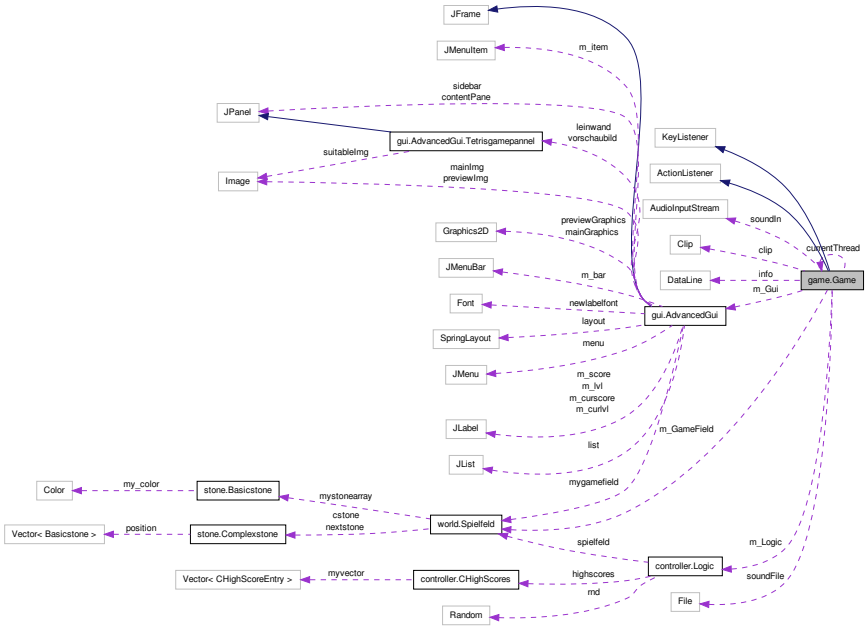
- src/game/Config.java

7.7 game.Game Klassenreferenz

Klassendiagramm für game.Game:



Zusammengehörigkeiten von game.Game:



Öffentliche Methoden

- Game (Spielfeld spf, Logic log, AdvancedGui gu)

- void run ()
- int GetStepWaitTime ()
- void SetStoneFalledDown ()
- void keyPressed (KeyEvent e)
- void UpdateGameField ()
- void newGameforce ()
- void keyReleased (KeyEvent e)
- void keyTyped (KeyEvent e)
- void actionPerformed (ActionEvent e)

Öffentliche, statische Methoden

- static void main (String[] args)

Private Methoden

- synchronized void PerformStep ()
- boolean IsEnded ()
- synchronized void ActionDown ()

Private, statische Methoden

- static void play ()

Private Attribute

- boolean m_IsEnded = false
- boolean m_StoneFallDown = false

Statische private Attribute

- static Game currentThread
- static Spielfeld m_GameField
- static Logic m_Logic
- static AdvancedGui m_Gui
- static File soundFile
- static Clip clip
- static AudioInputStream soundIn
- static DataLine.Info info

7.7.1 Ausführliche Beschreibung

Übergreifende Klasse, die das MVC-Pattern verdrahtet. Ggf. würde man hier einzelne Komponenten wie das Frontend auswechseln.

Definiert in Zeile 98 der Datei Game.java.

7.7.2 Beschreibung der Konstruktoren und Destruktoren

7.7.2.1 game.Game.Game (Spielfeld *spf,* Logic *log,* AdvancedGui *gu*)

Konstruktor für das Hauptklasse des Spiels

Parameter

spf	Zeiger auf das spielfeld

Siehe auch

m_Gamefiled

Parameter

log	Zeiger auf die Logik

Siehe auch

ma_Logic

Parameter

gu	Zeiger auf die GUI

Siehe auch

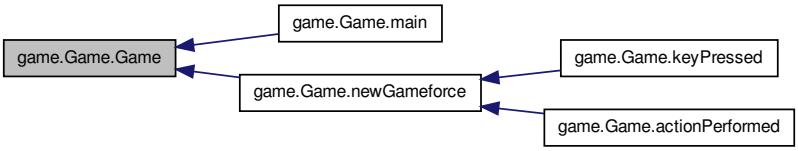
m_Gui

Definiert in Zeile 114 der Datei Game.java.

Benutzt game.Game.m_GameField, game.Game.m_Gui und game.Game.m_Logic.

Wird benutzt von game.Game.main() und game.Game.newGameforce().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.7.3 Dokumentation der Elementfunktionen

7.7.3.1 synchronized void game.Game.ActionDown ( ) [private]

Funktion die die Fallgeschwindigkeit der Steine von normal auf schnell umstellt

Definiert in Zeile 204 der Datei Game.java.

Benutzt game.Game.m_StoneFallDown.

Wird benutzt von game.Game.keyPressed().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:

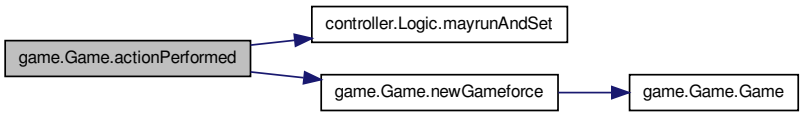
7.7.3.2 void game.Game.actionPerformed (ActionEvent *e*)

Funktion die dem Menüpunkt New im Menü Game eine Funktionalität ermöglicht

Definiert in Zeile 361 der Datei Game.java.

Benutzt game.Game.m_Logic, controller.Logic.mayrunAndSet() und game.Game.newGameforce().

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



7.7.3.3 int game.Game.GetStepWaitTime ( )

Funktion die die Wartezeit zwischen den einzelnen Schritten abruft, beachtet ggf. schnelles Fallen

Rückgabe

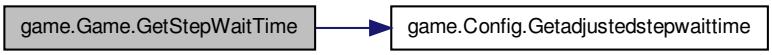
Wwartezeit zwischen den einzeln Steps, beachtet Schnelles Fallen und das aktuelle Level

Definiert in Zeile 184 der Datei Game.java.

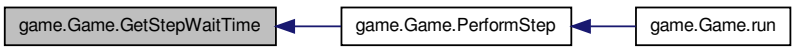
Benutzt game.Config.Getadjustedstepwaittime(), world.Spielfeld.Level, game.Game.m_GameField und game.Game.m_StoneFallDown.

Wird benutzt von game.Game.PerformStep().

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



Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.7.3.4 boolean game.Game.IsEnded ( ) [private]

Funktion die den Status des Spiels abruft

Rückgabe

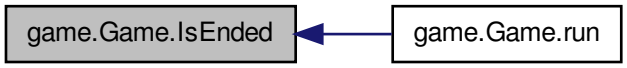
ob das Spiel beendet ist, false wenn das Spiel noch läuft

Definiert in Zeile 196 der Datei Game.java.

Benutzt game.Game.m_IsEnded.

Wird benutzt von game.Game.run().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



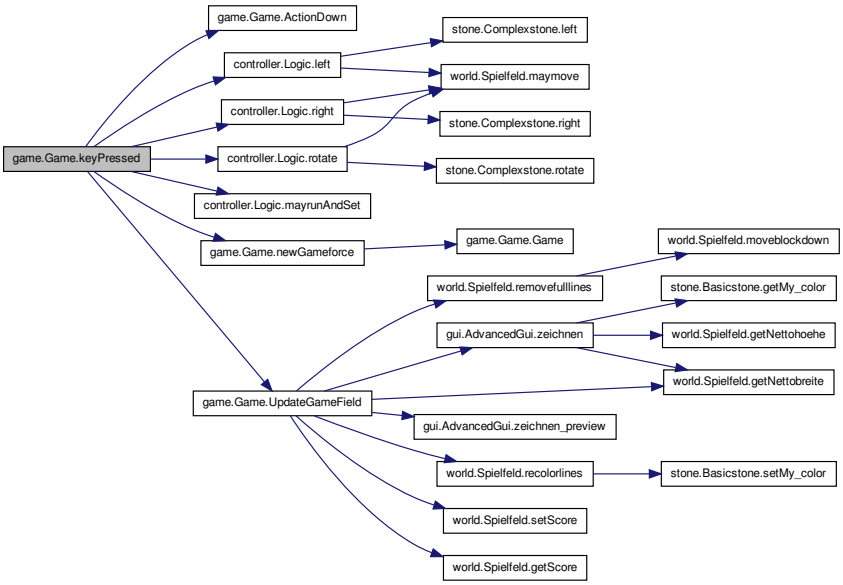
7.7.3.5 void game.Game.keyPressed (KeyEvent *e*)

Zuordnung der Reaktion auf einzelne Tasten

Definiert in Zeile 243 der Datei Game.java.

Benutzt game.Game.ActionDown(), controller.Logic.left(), game.Game.m_Logic, controller.Logic.mayrunAndSet(), game.Game.newGameforce(), controller.Logic.right(), controller.Logic.rotate() und game.Game.UpdateGameField().

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



7.7.3.6 void game.Game.keyReleased (KeyEvent *e*)

Definiert in Zeile 307 der Datei Game.java.

7.7.3.7 void game.Game.keyTyped (KeyEvent *e*)

Definiert in Zeile 312 der Datei Game.java.

7.7.3.8 static void game.Game.main (String[] *args*) [static]

Hauptfunktion, die alle Einstellungen trifft und ein Spiel startet

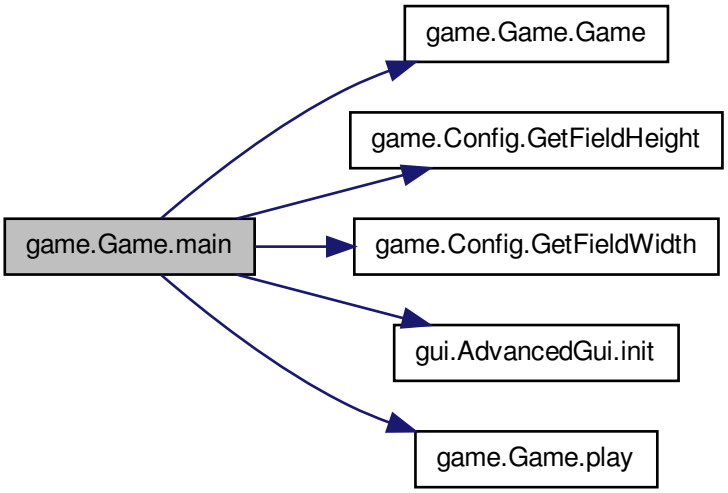
Parameter

args	

Definiert in Zeile 222 der Datei Game.java.

Benutzt game.Game.currentThread, game.Game.Game(), game.Config.GetFieldHeight(), game.Config.GetFieldWidth(), gui.AdvancedGui.init(), game.Game.m_GameField, game.Game.m_Gui, gui.AdvancedGui.m_item, game.Game.m_Logic und game.Game.play().

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



7.7.3.9 void game.Game.newGameforce ( )

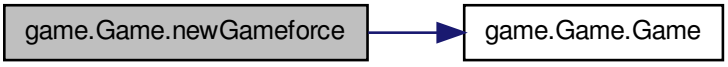
Funktion die ein neues Spiel startet, insbesondere eine neue Gameinstanz erstellt, die Keylistener neu verdrahtet und den Thread startet

Definiert in Zeile 298 der Datei Game.java.

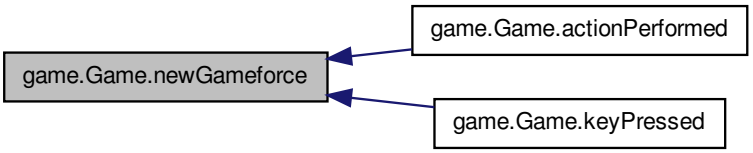
Benutzt game.Game.currentThread, game.Game.Game(), game.Game.m_GameField, game.Game.m_Gui und game.Game.m_Logic.

Wird benutzt von game.Game.actionPerformed() und game.Game.keyPressed().

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



Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.7.3.10 synchronized void game.Game.PerformStep ( ) [private]

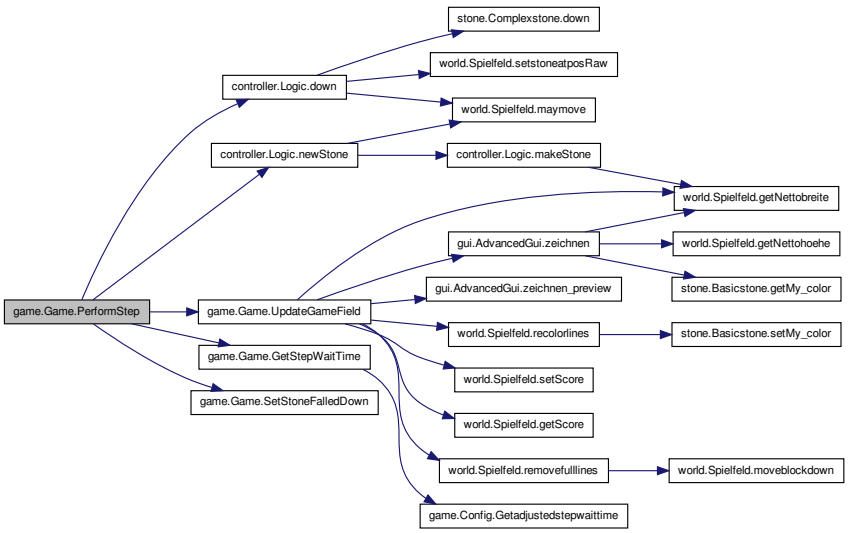
Funktion die den stein um eine Position fallen lässt und dazwischen wartet

Definiert in Zeile 158 der Datei Game.java.

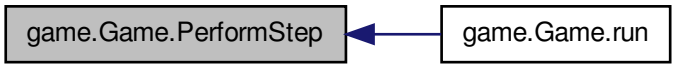
Benutzt controller.Logic.down(), game.Game.GetStepWaitTime(), game.Game.m_IsEnded, game.Game.m_Logic, controller.Logic.newStone(), game.Game.SetStoneFalledDown() und game.Game.UpdateGameField().

Wird benutzt von game.Game.run().

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



Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.7.3.11 static void game.Game.play ( ) [static, private]

Funktion die den Hintergrundsound abspielt

Definiert in Zeile 325 der Datei Game.java.

Benutzt game.Game.clip, game.Game.info, game.Game.soundFile und game.Game.soundIn.

Wird benutzt von game.Game.main().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:

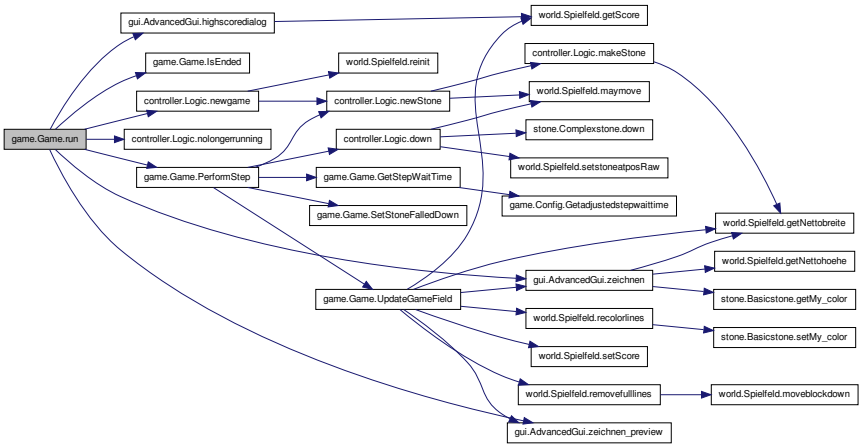
7.7.3.12 void game.Game.run ( )

Die überladene run-Methode aus Thread um unseren eigen Thread zu erzeugen

Definiert in Zeile 124 der Datei Game.java.

Benutzt gui.AdvancedGui.highscoredialog(), game.Game.IsEnded(), game.Game.m_Gui, game.Game.m_IsEnded, game.Game.m_Logic, game.Game.m_StoneFallDown, controller.Logic.newgame(), controller.Logic.nolongerrunning(), game.Game.PerformStep(), gui.AdvancedGui.zeichnen() und gui.AdvancedGui.zeichnen_preview().

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



7.7.3.13 void game.Game.SetStoneFalledDown ( )

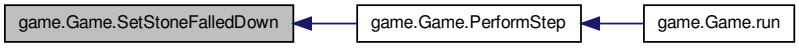
Funktion die einträgt, dass der Stein nicht mehr fällt und er somit das untere Ende des Spielfeld erreicht hat

Definiert in Zeile 212 der Datei Game.java.

Benutzt game.Game.m_StoneFallDown.

Wird benutzt von game.Game.PerformStep().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.7.3.14 void game.Game.UpdateGameField ( )

Funktion die das Spielfeld aktualisiert

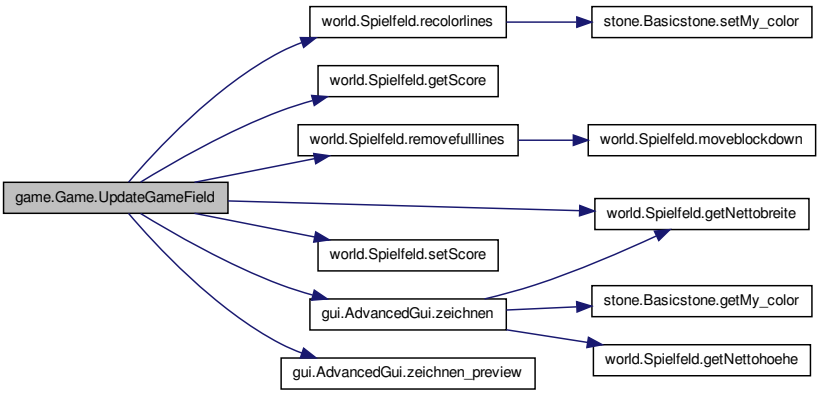
Definiert in Zeile 279 der Datei Game.java.

Benutzt world.Spielfeld.getNettobreite(), world.Spielfeld.getScore(), game.Game.m_GameField, game.Game.m_Gui, world.Spielfeld.nextstone, world.Spielfeld.-

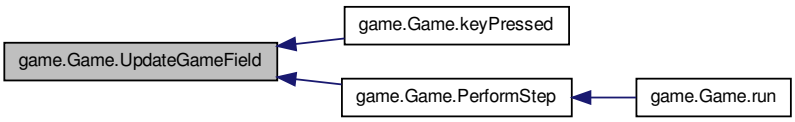
recolorlines(), world.Spielfeld.removefulllines(), world.Spielfeld.setScore(), gui.-AdvancedGui.zeichnen() und gui.AdvancedGui.zeichnen_preview().

Wird benutzt von game.Game.keyPressed() und game.Game.PerformStep().

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



Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.7.4 Dokumentation der Datenelemente

7.7.4.1 Clip game.Game.clip [static, private]

Definiert in Zeile 318 der Datei Game.java.

Wird benutzt von game.Game.play().

7.7.4.2 Game game.Game.currentThread [static, private]

Definiert in Zeile 100 der Datei Game.java.

Wird benutzt von game.Game.main() und game.Game.newGameforce().

7.7.4.3 DataLine.Info game.Game.info [static, private]

Definiert in Zeile 320 der Datei Game.java.

Wird benutzt von game.Game.play().

7.7.4.4 Spielfeld game.Game.m_GameField [static, private]

Definiert in Zeile 101 der Datei Game.java.

Wird benutzt von game.Game.Game(), game.Game.GetStepWaitTime(), game.Game.main(), game.Game.newGameforce() und game.Game.UpdateGameField().

7.7.4.5 AdvancedGui game.Game.m_Gui [static, private]

Definiert in Zeile 103 der Datei Game.java.

Wird benutzt von game.Game.Game(), game.Game.main(), game.Game.newGameforce(), game.Game.run() und game.Game.UpdateGameField().

7.7.4.6 boolean game.Game.m_IsEnded = false [private]

Definiert in Zeile 104 der Datei Game.java.

Wird benutzt von game.Game.IsEnded(), game.Game.PerformStep() und game.Game.run().

7.7.4.7 Logic game.Game.m_Logic [static, private]

Definiert in Zeile 102 der Datei Game.java.

Wird benutzt von game.Game.actionPerformed(), game.Game.Game(), game.Game.keyPressed(), game.Game.main(), game.Game.newGameforce(), game.Game.PerformStep() und game.Game.run().

7.7.4.8 boolean game.Game.m_StoneFallDown = false [private]

Definiert in Zeile 105 der Datei Game.java.

Wird benutzt von game.Game.ActionDown(), game.Game.GetStepWaitTime(), game.Game.run() und game.Game.SetStoneFalledDown().

7.7.4.9 File game.Game.soundFile [static, private]

Definiert in Zeile 317 der Datei Game.java.

Wird benutzt von game.Game.play().

7.7.4.10 AudioInputStream game.Game.soundIn [static, private]

Definiert in Zeile 319 der Datei Game.java.

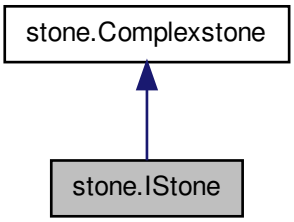
Wird benutzt von game.Game.play().

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

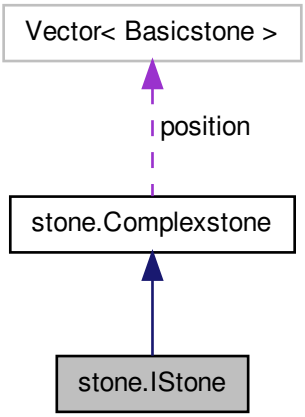
- src/game/Game.java

7.8 stone.IStone Klassenreferenz

Klassendiagramm für stone.IStone:



Zusammengehörigkeiten von stone.IStone:



Öffentliche Methoden

- IStone (int xpos_, int ypos_)

7.8.1 Ausführliche Beschreibung

Stein in Form eines senkrechten blauen Balkens aus 4 Grundsteinen

Definiert in Zeile 10 der Datei IStone.java.

7.8.2 Beschreibung der Konstruktoren und Destruktoren

7.8.2.1 stone.IStone.IStone (int *xpos_*, int *ypos_*)

Der Konstruktor für einen senkrechten Stein aus 4 Grundsteinen

Parameter

*xpos_*	X-Position des Mittelpunktes des neuen Steins
*ypos_*	Y-Position des Mittelpunktes des neuen Steins

Definiert in Zeile 18 der Datei IStone.java.

Benutzt stone.Complexstone.position, stone.Complexstone.xpos und stone.Complexstone.-ypos.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- src/stone/IStone.java

7.9 controller.JHighscoreFileIO Klassenreferenz

Öffentliche, statische Methoden

- static synchronized void toDisk (CHighScores chs)
- static synchronized CHighScores fromDisk ()

7.9.1 Ausführliche Beschreibung

Manager zum Lesen und Schreiben des ganzen CHighScores-Objektes based on http://www2.math.uni-wuppertal.de/~axel/skripte/oop/oop27_6.html

Definiert in Zeile 11 der Datei JHighscoreFileIO.java.

7.9.2 Dokumentation der Elementfunktionen

7.9.2.1 static synchronized CHighScores controller.JHighscoreFileIO.fromDisk ( ) [static]

liest ein CHighScores-Objekt aus der Datei highscores.ser

Rückgabe

Highscoresobjekt, welches von der Festplatte gelesen wurde oder null

Definiert in Zeile 36 der Datei JHighscoreFileIO.java.

7.9.2.2 static synchronized void controller.JHighscoreFileIO.toDisk (CHighScores *chs*) [static]

Schreibt ein CHighScores-Objekt in die Datei highscores.ser

Parameter

- *Highscoresobjekt,welches*	auf die Festplatte geschrieben werden soll

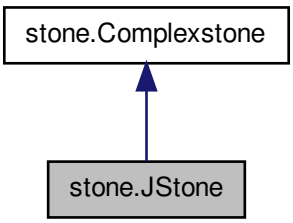
Definiert in Zeile 18 der Datei JHighscoreFileIO.java.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

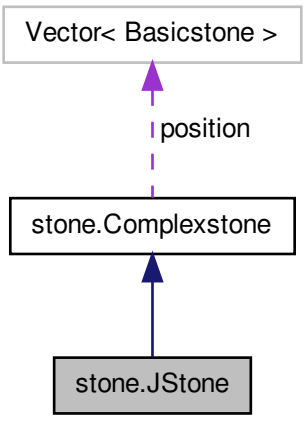
- src/controller/JHighscoreFileIO.java

7.10 stone.JStone Klassenreferenz

Klassendiagramm für stone.JStone:



Zusammengehörigkeiten von stone.JStone:



Öffentliche Methoden

- JStone (int xpos_, int ypos_)

7.10.1 Ausführliche Beschreibung

Stein in Form eines J-förmigen Steins

Definiert in Zeile 10 der Datei JStone.java.

7.10.2 Beschreibung der Konstruktoren und Destruktoren

7.10.2.1 stone.JStone.JStone (int *xpos_*, int *ypos_*)

Der Konstruktor für einen J-förmigen Stein aus 3 senkrechten Grundsteinen und einem links unten daneben

Parameter

*xpos_*	X-Position des Mittelpunktes des neuen Steins
*ypos_*	Y-Position des Mittelpunktes des neuen Steins

Definiert in Zeile 18 der Datei JStone.java.

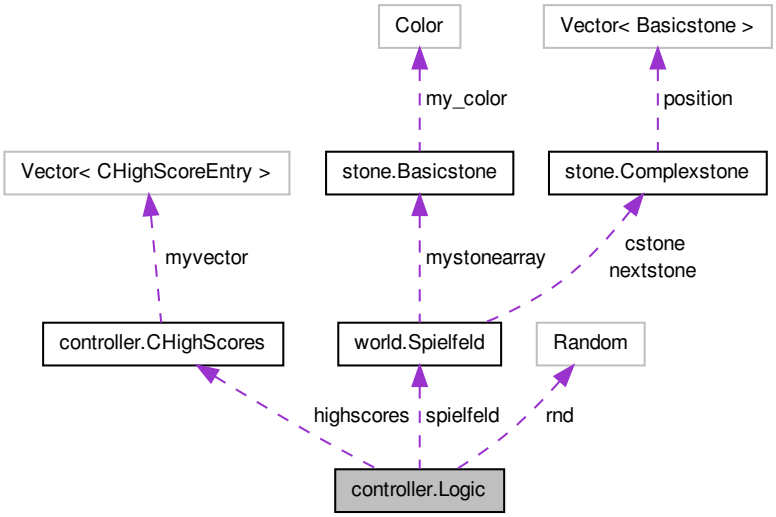
Benutzt stone.Complexstone.position, stone.Complexstone.xpos und stone.Complexstone.-ypos.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- src/stone/JStone.java

7.11 controller.Logic Klassenreferenz

Zusammengehörigkeiten von controller.Logic:



Öffentliche Methoden

- Logic (Spielfeld myspielfeld)
- synchronized boolean mayrunAndSet ()
- synchronized void nolongerrunning ()
- boolean newStone ()
- boolean down ()
- void left ()
- void right ()
- void rotate ()
- void newgame ()

Öffentliche Attribute

- CHighScores highscores = new CHighScores()

Paketattribute

- Spielfeld spielfeld
- Random rnd

Private Methoden

- Complexstone makeStone (int rndst)

Private Attribute

- boolean gamerunning = false

7.11.1 Ausführliche Beschreibung

Logikklasse ist Controllerkomponente aus dem MVC-Pattern. Sie kennt das Spielfeld und stellt Funktionen zur geprüften Verschiebung der Steine zur Verfügung.

Definiert in Zeile 23 der Datei Logic.java.

7.11.2 Beschreibung der Konstruktoren und Destruktoren

7.11.2.1 controller.Logic.Logic (Spielfeld *myspielfeld*)

Konstruktor für die Logicklasse des spieles

Parameter

myspielfeld	spiefeld für das der Controler die spielzüge kontrolieren soll

Definiert in Zeile 44 der Datei Logic.java.

Benutzt controller.Logic.rnd und controller.Logic.spielfeld.

7.11.3 Dokumentation der Elementfunktionen

7.11.3.1 boolean controller.Logic.down ( )

Methode um den Stein nach unten zu bewegen. Wenn er ganz unten angekommen ist, wird der Stein zerstört und in den Hintergrund kopiert.

Rückgabe

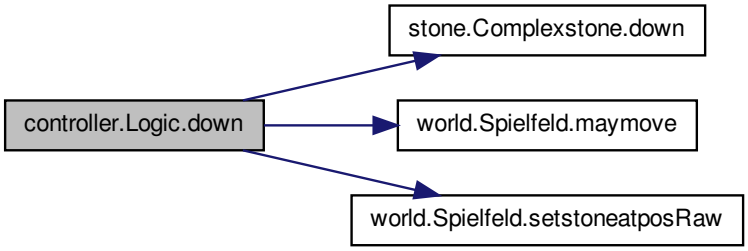
true kann stein nach unten bewegen false kann stein nicht bewegen

Definiert in Zeile 133 der Datei Logic.java.

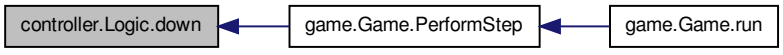
Benutzt world.Spielfeld.cstone, stone.Complexstone.down(), world.Spielfeld.maymove(), stone.Complexstone.position, world.Spielfeld.setstoneatposRaw() und controller.Logic.spielfeld.

Wird benutzt von game.Game.PerformStep().

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



Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.11.3.2 void controller.Logic.left ( )

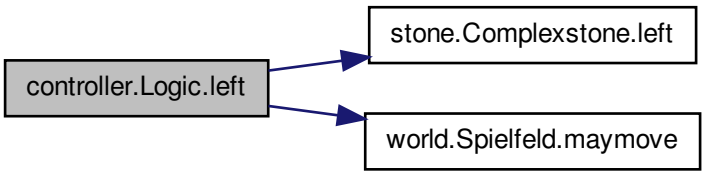
Bewegt Stein nach links, falls dort freier Platz ist

Definiert in Zeile 160 der Datei Logic.java.

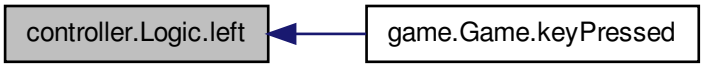
Benutzt world.Spielfeld.cstone, stone.Complexstone.left(), world.Spielfeld.maymove() und controller.Logic.spielfeld.

Wird benutzt von game.Game.keyPressed().

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



Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.11.3.3 Complexstone controller.Logic.makeStone (int *rndst*) [private]

erzeugt eine der Steinformen, namentlich die mit der uebergebenen Nummer oder die der Modulonummer falls der Parameter zu gross ist

Parameter

rndst	Nummer des gewünschten Steins im Bereich 0..9

Rückgabe

neu angelegter Stein der angeforderten Form

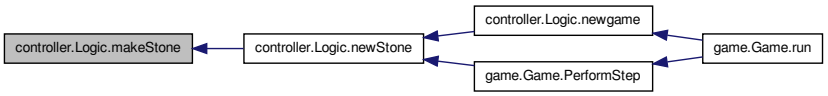
Definiert in Zeile 104 der Datei Logic.java.

Benutzt world.Spielfeld.getNettobreite() und controller.Logic.spielfeld.

Wird benutzt von controller.Logic.newStone().

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

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.11.3.4 synchronized boolean controller.Logic.mayrunAndSet ( )

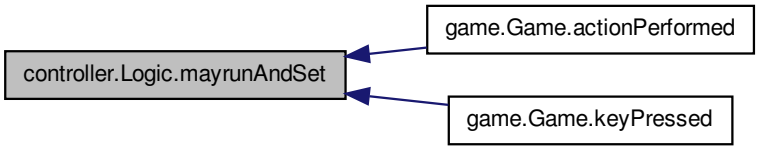
threadsichere Abfragemethode ob ein Spiel stattfindet falls nicht wird das Flag gesetzt

Definiert in Zeile 55 der Datei Logic.java.

Benutzt controller.Logic.gamerunning.

Wird benutzt von game.Game.actionPerformed() und game.Game.keyPressed().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.11.3.5 void controller.Logic.newgame ( )

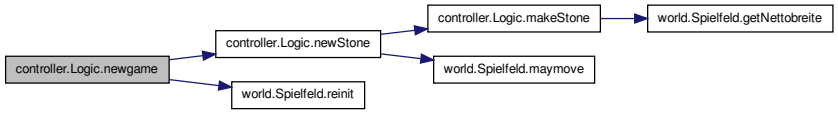
Reinitialisiert existierendes Spielfeld

Definiert in Zeile 200 der Datei Logic.java.

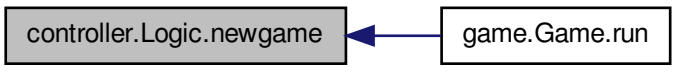
Benutzt controller.Logic.newStone(), world.Spielfeld.reinit() und controller.Logic.-spielfeld.

Wird benutzt von game.Game.run().

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



Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.11.3.6 boolean controller.Logic.newStone ( )

Methode zum erzeugen eines neuen Spielsteines. Die Form wird zufällig ausgewählt.

Rückgabe

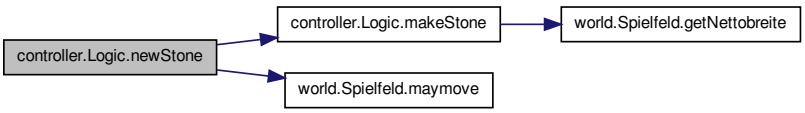
true falls neuer stein angelegt werden kann false falls nicht möglich

Definiert in Zeile 79 der Datei Logic.java.

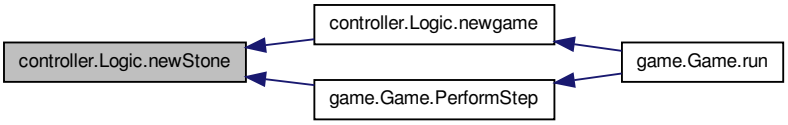
Benutzt world.Spielfeld.cstone, controller.Logic.makeStone(), world.Spielfeld.-maymove(), world.Spielfeld.nextstone, controller.Logic.rnd und controller.Logic.-spielfeld.

Wird benutzt von controller.Logic.newgame() und game.Game.PerformStep().

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



Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.11.3.7 synchronized void controller.Logic.nolongerrunning ( )

threadsichere Rücksetzmethode wenn das Spiel endet

Definiert in Zeile 68 der Datei Logic.java.

Benutzt controller.Logic.gamerunning.

Wird benutzt von game.Game.run().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:

7.11.3.8 void controller.Logic.right ( )

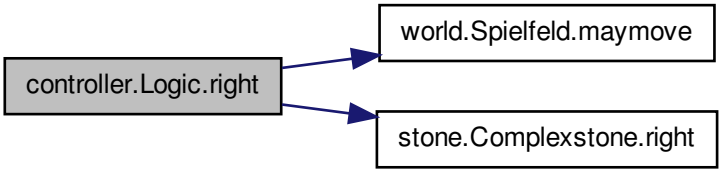
Bewegt Stein nach rechts, falls da freier Platz ist

Definiert in Zeile 173 der Datei Logic.java.

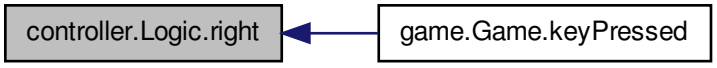
Benutzt world.Spielfeld.cstone, world.Spielfeld.maymove(), stone.Complexstone.right() und controller.Logic.spielfeld.

Wird benutzt von game.Game.keyPressed().

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



Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.11.3.9 void controller.Logic.rotate ( )

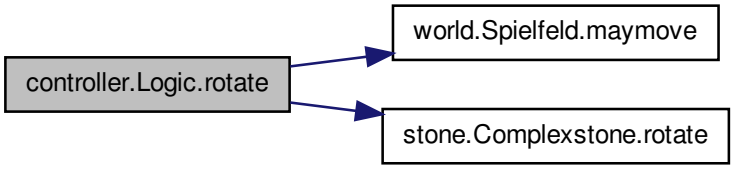
Rotiert den Stein um seine Rotationsachse, falls da freier Platz ist

Definiert in Zeile 186 der Datei Logic.java.

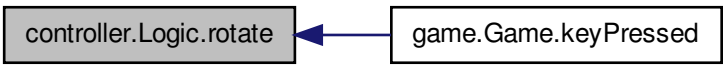
Benutzt world.Spielfeld.cstone, world.Spielfeld.maymove(), stone.Complexstone.-rotate() und controller.Logic.spielfeld.

Wird benutzt von game.Game.keyPressed().

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



Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.11.4 Dokumentation der Datenelemente

7.11.4.1 boolean controller.Logic.gamerunning = false [private]

Flag ob gerade ein Spiel stattfindet, verhindert mehrfaches Starten eines Spiels

Definiert in Zeile 30 der Datei Logic.java.

Wird benutzt von controller.Logic.mayrunAndSet() und controller.Logic.nolongerrunning().

7.11.4.2 CHighScores controller.Logic.highscores = new CHighScores()

Definiert in Zeile 25 der Datei Logic.java.

7.11.4.3 Random controller.Logic.rnd [package]

Zufallsgenerator dient zum Erzeugen der Zufallszahlem für die newStone() methode zu erzeugen

Definiert in Zeile 39 der Datei Logic.java.

Wird benutzt von controller.Logic.Logic() und controller.Logic.newStone().

7.11.4.4 Spielfeld controller.Logic.spielfeld [package]

Variable um das Spielfeld kontroliert zu speichern

Definiert in Zeile 35 der Datei Logic.java.

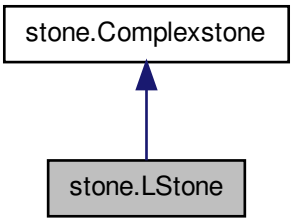
Wird benutzt von controller.Logic.down(), controller.Logic.left(), controller.Logic.Logic(), controller.Logic.makeStone(), controller.Logic.newgame(), controller.Logic.newStone(), controller.Logic.right() und controller.Logic.rotate().

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

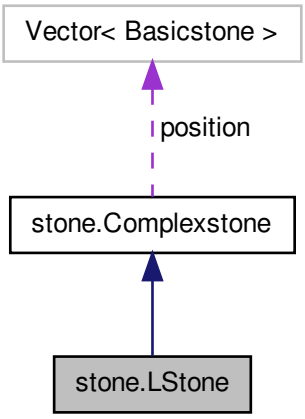
- src/controller/Logic.java

7.12 stone.LStone Klassenreferenz

Klassendiagramm für stone.LStone:



Zusammengehörigkeiten von stone.LStone:



Öffentliche Methoden

- LStone (int xpos_, int ypos_)

7.12.1 Ausführliche Beschreibung

Stein in L-Form

Definiert in Zeile 11 der Datei LStone.java.

7.12.2 Beschreibung der Konstruktoren und Destruktoren

7.12.2.1 stone.LStone.LStone (int *xpos_*, int *ypos_*)

Der Konstruktor für einen L-förmigen Stein aus 3 senkrechten Grundsteinen und einem rechts unten daneben

Parameter

*xpos_*	X-Position des Mittelpunktes des neuen Steins
*ypos_*	Y-Position des Mittelpunktes des neuen Steins

Definiert in Zeile 20 der Datei LStone.java.

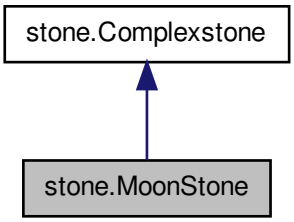
Benutzt stone.Complexstone.position, stone.Complexstone.xpos und stone.Complexstone.-ypos.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

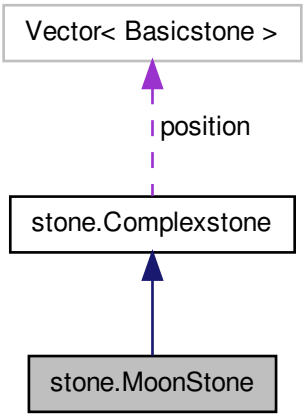
- src/stone/LStone.java

7.13 stone.MoonStone Klassenreferenz

Klassendiagramm für stone.MoonStone:



Zusammengehörigkeiten von stone.MoonStone:



Öffentliche Methoden

- MoonStone (int xpos_, int ypos_)

7.13.1 Ausführliche Beschreibung

Stein in Form eines blauen Kreuzes mit einem gelben Grundstein (Mond) in einer Ecke

Definiert in Zeile 10 der Datei MoonStone.java.

7.13.2 Beschreibung der Konstruktoren und Destruktoren

7.13.2.1 stone.MoonStone.MoonStone (int *xpos_*, int *ypos_*)

Der Konstruktor für einen Mondstein

Parameter

*xpos_*	X-Position des Mittelpunktes des neuen Steins
*ypos_*	Y-Position des Mittelpunktes des neuen Steins

Definiert in Zeile 18 der Datei MoonStone.java.

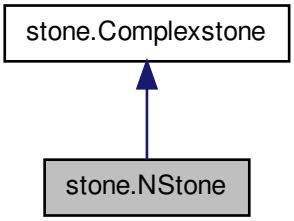
Benutzt stone.Complexstone.position, stone.Complexstone.xpos und stone.Complexstone.-ypos.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

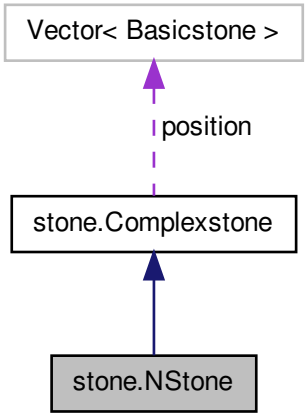
- src/stone/MoonStone.java

7.14 stone.NStone Klassenreferenz

Klassendiagramm für stone.NStone:



Zusammengehörigkeiten von stone.NStone:



Öffentliche Methoden

- NStone (int xpos_, int ypos_)

7.14.1 Ausführliche Beschreibung

Disjunkter Stein aus 4 Grundsteinen, die die Ecken eines Quadrats bilden

Definiert in Zeile 12 der Datei NStone.java.

7.14.2 Beschreibung der Konstruktoren und Destruktoren

7.14.2.1 stone.NStone.NStone (int *xpos_*, int *ypos_*)

Der Konstruktor für einen Stein aus 4 voneinander getrennten, quadratisch angeordneten Elementen

Parameter

*xpos_*	X-Position des Mittelpunktes des neuen Steins
*ypos_*	Y-Position des Mittelpunktes des neuen Steins

Definiert in Zeile 20 der Datei NStone.java.

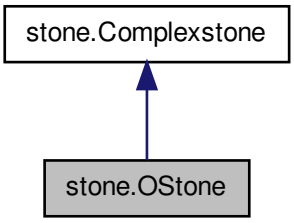
Benutzt stone.Complexstone.position, stone.Complexstone.xpos und stone.Complexstone.ypos.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

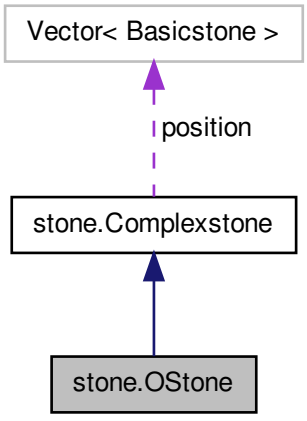
- src/stone/NStone.java

7.15 stone.OStone Klassenreferenz

Klassendiagramm für stone.OStone:



Zusammengehörigkeiten von stone.OStone:



Öffentliche Methoden

- OStone (int xpos_, int ypos_)

7.15.1 Ausführliche Beschreibung

Stein in Form eines Quadrates 2x2

Definiert in Zeile 12 der Datei OStone.java.

7.15.2 Beschreibung der Konstruktoren und Destruktoren

7.15.2.1 stone.OStone.OStone (int *xpos_*, int *ypos_*)

Der Konstruktor für einen quadratischen 2x2 großen Stein

Parameter

*xpos_*	X-Position des Mittelpunktes des neuen Steins
*ypos_*	Y-Position des Mittelpunktes des neuen Steins

Definiert in Zeile 21 der Datei OStone.java.

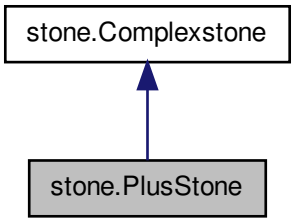
Benutzt stone.Complexstone.position, stone.Complexstone.xpos und stone.Complexstone.-ypos.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

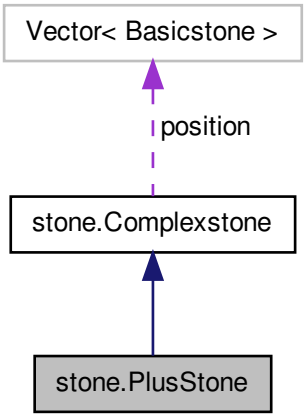
- src/stone/OStone.java

7.16 stone.PlusStone Klassenreferenz

Klassendiagramm für stone.PlusStone:



Zusammengehörigkeiten von stone.PlusStone:



Öffentliche Methoden

- PlusStone (int xpos_, int ypos_)

7.16.1 Ausführliche Beschreibung

Stein in Form eines Pluszeichens aus insgesamt 5 Grundsteinen

Definiert in Zeile 12 der Datei PlusStone.java.

7.16.2 Beschreibung der Konstruktoren und Destruktoren

7.16.2.1 stone.PlusStone.PlusStone (int *xpos_*, int *ypos_*)

Der Konstruktor für einen plusförmigen Stein aus 3 senkrechten Grundsteinen und jeweils links und rechts in der Mitte einem daneben

Parameter

*xpos_*	X-Position des Mittelpunktes des neuen Steins
*ypos_*	Y-Position des Mittelpunktes des neuen Steins

Definiert in Zeile 20 der Datei PlusStone.java.

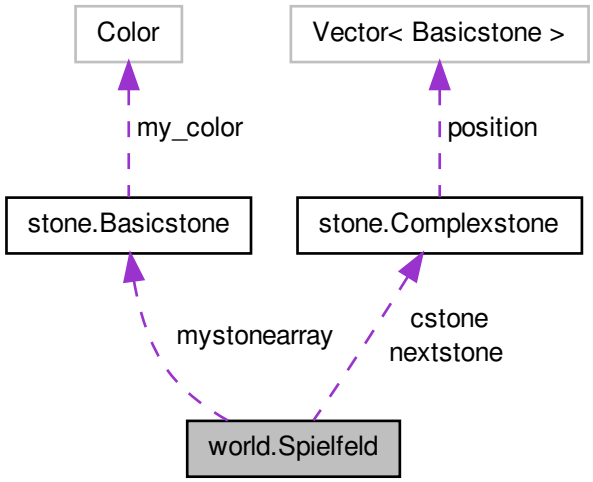
Benutzt stone.Complexstone.position, stone.Complexstone.xpos und stone.Complexstone.-ypos.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- src/stone/PlusStone.java

7.17 world.Spielfeld Klassenreferenz

Zusammengehörigkeiten von world.Spielfeld:



Öffentliche Methoden

- int getScore ()
- void setScore (int score)
- int getNettohoehe ()
- int getNettobreite ()
- Basicstone getstoneatpos (int x, int y)
- void setstoneatposRaw (int x, int y, Basicstone b)
- int recolorlines (Color linecolor)
- int removefulllines ()
- void moveblockdown (int ytoerased)
- Spielfeld (int nettobreite, int nettohoehe)

- String toString ()
- boolean maymove (Complexstone cs)
- void reinit ()

Öffentliche Attribute

- Basicstone[][] mystonearray
- Complexstone cstone
- Complexstone nextstone
- int score
- int Level

Private Attribute

- int hoehe
- int nettohoehe
- int breite
- int nettobreite
- int linestillnxtlvl

7.17.1 Ausführliche Beschreibung

Definiert in Zeile 13 der Datei Spielfeld.java.

7.17.2 Beschreibung der Konstruktoren und Destruktoren

7.17.2.1 world.Spielfeld.Spielfeld (int *nettobreite,* int *nettohoehe*)

Konstuktor für das Spielfeld

Parameter

nettobreite	Breite des spielfeldes ohne Rahmen in Grundsteinen
nettohoehe	Höhe des Spielfeldes ohne Rahmen in Grundsteinen

Definiert in Zeile 177 der Datei Spielfeld.java.

Benutzt world.Spielfeld.breite, world.Spielfeld.hoehe, world.Spielfeld.Level, world.-Spielfeld.linestillnxtlvl, world.Spielfeld.mystonearray, world.Spielfeld.nettobreite, world.-Spielfeld.nettohoehe und world.Spielfeld.score.

7.17.3 Dokumentation der Elementfunktionen

7.17.3.1 int world.Spielfeld.getNettobreite ( )

Liefert die Spielfelds ohne Rand in Grundsteinen

Rückgabe

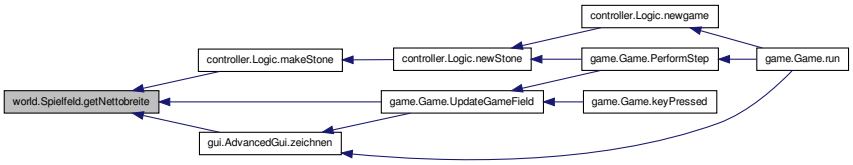
Breite ohne Rand

Definiert in Zeile 78 der Datei Spielfeld.java.

Benutzt world.Spielfeld.nettobreite.

Wird benutzt von controller.Logic.makeStone(), game.Game.UpdateGameField() und gui.AdvancedGui.zeichnen().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.17.3.2 int world.Spielfeld.getNettohoehe ( )

Liefert die Höhe des Spielfelds ohne Rand in Grundsteinen zurück

Rückgabe

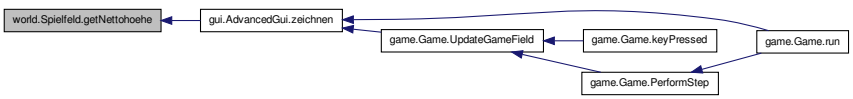
Höhe ohne Rand

Definiert in Zeile 70 der Datei Spielfeld.java.

Benutzt world.Spielfeld.nettohoehe.

Wird benutzt von gui.AdvancedGui.zeichnen().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.17.3.3 int world.Spielfeld.getScore ( )

Rückgabe

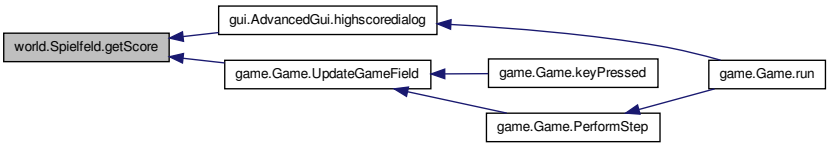
liefert den Punktestand

Definiert in Zeile 57 der Datei Spielfeld.java.

Benutzt world.Spielfeld.score.

Wird benutzt von gui.AdvancedGui.highscoredialog() und game.Game.UpdateGameField().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.17.3.4 Basicstone world.Spielfeld.getstoneatpos (int *x*, int *y*)

Liefer den Grundstein an der übergeben Position zurück

Parameter

x	X Position
y	Y position

Rückgabe

Grundstein an der Position x,y bzw. null

Definiert in Zeile 89 der Datei Spielfeld.java.

Benutzt world.Spielfeld.mystonearray.

7.17.3.5 boolean world.Spielfeld.maymove (Complexstone *cs*)

Prüft ob ein Stein an die gegebene Position kopiert werden könnte, d.h. ob der Platz im Array fest ist

Parameter

cs	Stein der bewegt werden soll

Rückgabe

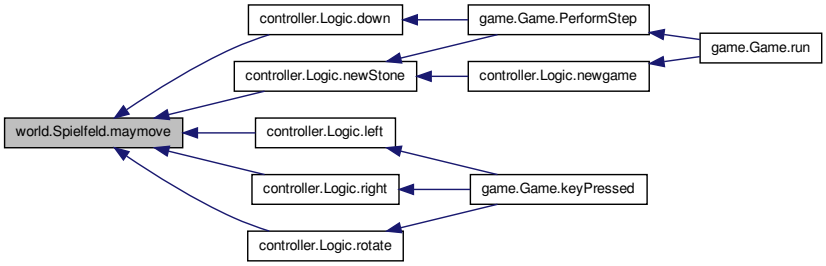
true falls der Stein an die inhärente Position passt, sonst false

Definiert in Zeile 251 der Datei Spielfeld.java.

Benutzt world.Spielfeld.mystonearray und stone.Complexstone.position.

Wird benutzt von controller.Logic.down(), controller.Logic.left(), controller.Logic.newStone(), controller.Logic.right() und controller.Logic.rotate().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.17.3.6 void world.Spielfeld.moveblockdown (int *ytoerased*)

Kopiert den Rest des Hintergrundarray y Zeilen nach unten

Parameter

Zeilenzahl	die nach unten kopiert werden sollen

Definiert in Zeile 161 der Datei Spielfeld.java.

Benutzt world.Spielfeld.mystonearray und world.Spielfeld.nettobreite.

Wird benutzt von world.Spielfeld.removefulllines().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:

7.17.3.7 int world.Spielfeld.recolorlines (Color *linecolor*)

Findet volle Linien

Parameter

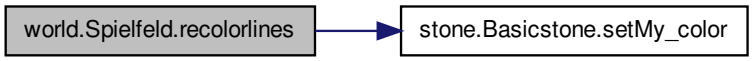
linecolor	Wunschfarbe für die vollen Linien

Definiert in Zeile 109 der Datei Spielfeld.java.

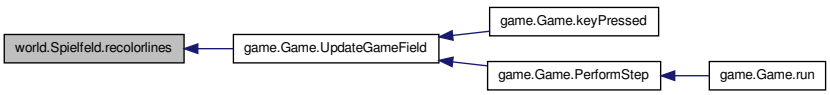
Benutzt stone.Basicstone.ispartoffullline, world.Spielfeld.mystonearray, world.Spielfeld.-nettobreite, world.Spielfeld.nettohoehe und stone.Basicstone.setMy_color().

Wird benutzt von game.Game.UpdateGameField().

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



Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.17.3.8 void world.Spielfeld.reinit ( )

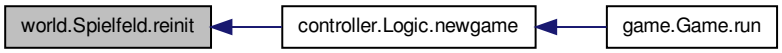
Löscht alle Steine(Grundsteine und Complexstones) aus dem spielfeld so das ein neues Spiel in dem Spielfeld laufen kann.

Definiert in Zeile 265 der Datei Spielfeld.java.

Benutzt world.Spielfeld.mystonearray, world.Spielfeld.nettobreite, world.Spielfeld.-nettohoehe, world.Spielfeld.nextstone und world.Spielfeld.score.

Wird benutzt von controller.Logic.newgame().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.17.3.9 int world.Spielfeld.removefulllines ( )

Löscht die vollen Linien aus dem Hintergrundarray

Rückgabe

Anzahl der gelöschten Linien

Definiert in Zeile 136 der Datei Spielfeld.java.

Benutzt world.Spielfeld.Level, world.Spielfeld.linestillnxtlvl, world.Spielfeld.moveblockdown(), world.Spielfeld.mystonearray, world.Spielfeld.nettobreite, world.Spielfeld.nettohoehe und world.Spielfeld.score.

Wird benutzt von game.Game.UpdateGameField().

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

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:

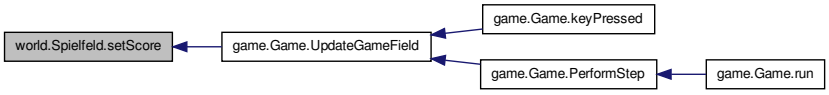
7.17.3.10 void world.Spielfeld.setScore (int *score*)

Definiert in Zeile 61 der Datei Spielfeld.java.

Benutzt world.Spielfeld.score.

Wird benutzt von game.Game.UpdateGameField().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.17.3.11 void world.Spielfeld.setstoneatposRaw (int *x*, int *y*, Basicstone *b*)

Setzt an die Position einen Basistein

Parameter

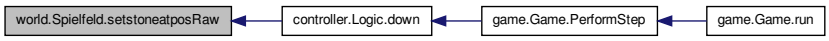
x	X position
y	Y position
b	Der Stein, welcher in das Hintergrundarray geschrieben werden soll

Definiert in Zeile 100 der Datei Spielfeld.java.

Benutzt world.Spielfeld.mystonearray.

Wird benutzt von controller.Logic.down().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



7.17.3.12 String world.Spielfeld.toString ( )

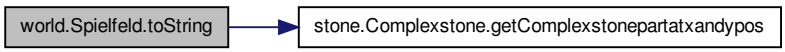
Ausgabemethode für die Spielfeldlkasse zu Testzwecken

Definiert in Zeile 217 der Datei Spielfeld.java.

Benutzt world.Spielfeld.breite, world.Spielfeld.cstone, stone.Complexstone.getComplexstonepartatxandypos(), world.Spielfeld.hoehe, world.Spielfeld.mystonearray

und world.Spielfeld.score.

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



7.17.4 Dokumentation der Datenelemente

7.17.4.1 int world.Spielfeld.breite [private]

Breite des Spielfelds mit Rahmen

Definiert in Zeile 27 der Datei Spielfeld.java.

Wird benutzt von world.Spielfeld.Spielfeld() und world.Spielfeld.toString().

7.17.4.2 Complexstone world.Spielfeld.cstone

Momentaner Spielstein der sich aus Einzelsteinen zusammensetzt

Definiert in Zeile 39 der Datei Spielfeld.java.

Wird benutzt von controller.Logic.down(), controller.Logic.left(), controller.Logic.newStone(), controller.Logic.right(), controller.Logic.rotate(), world.Spielfeld.toString() und gui.AdvancedGui.zeichnen().

7.17.4.3 int world.Spielfeld.hoehe [private]

Höhe des Soielfelds mit Kollisionserkennungsrahmen in Grundsteinen

Definiert in Zeile 19 der Datei Spielfeld.java.

Wird benutzt von world.Spielfeld.Spielfeld() und world.Spielfeld.toString().

7.17.4.4 int world.Spielfeld.Level

Definiert in Zeile 49 der Datei Spielfeld.java.

Wird benutzt von game.Game.GetStepWaitTime(), world.Spielfeld.removefulllines(), world.Spielfeld.Spielfeld() und gui.AdvancedGui.zeichnen().

7.17.4.5 int world.Spielfeld.linestillnxtlvl [private]

Definiert in Zeile 50 der Datei Spielfeld.java.

Wird benutzt von world.Spielfeld.removefulllines() und world.Spielfeld.Spielfeld().

7.17.4.6 Basicstone [][] world.Spielfeld.mystonearray

Hintergrundarray um die einzelnen Steine nachdem sie runtergefallen sind zu speichern und um den Kollisionserkennungsrand zu realisieren

Definiert in Zeile 35 der Datei Spielfeld.java.

Wird benutzt von world.Spielfeld.getstoneatpos(), world.Spielfeld.maymove(), world.Spielfeld.moveblockdown(), world.Spielfeld.recolorlines(), world.Spielfeld.reinit(), world.Spielfeld.removefulllines(), world.Spielfeld.setstoneatposRaw(), world.Spielfeld.Spielfeld(), world.Spielfeld.toString() und gui.AdvancedGui.zeichnen().

7.17.4.7 int world.Spielfeld.nettobreite [private]

Breite des Spielfelds ohne Rahmen

Definiert in Zeile 31 der Datei Spielfeld.java.

Wird benutzt von world.Spielfeld.getNettobreite(), world.Spielfeld.moveblockdown(), world.Spielfeld.recolorlines(), world.Spielfeld.reinit(), world.Spielfeld.removefulllines() und world.Spielfeld.Spielfeld().

7.17.4.8 int world.Spielfeld.nettohoehe [private]

Höhe des Spielfelds ohne Kollissionserkennungsrahmen in Grundsteinen

Definiert in Zeile 23 der Datei Spielfeld.java.

Wird benutzt von world.Spielfeld.getNettohoehe(), world.Spielfeld.recolorlines(), world.Spielfeld.reinit(), world.Spielfeld.removefulllines() und world.Spielfeld.Spielfeld().

7.17.4.9 Complexstone world.Spielfeld.nextstone

Definiert in Zeile 43 der Datei Spielfeld.java.

Wird benutzt von controller.Logic.newStone(), world.Spielfeld.reinit(), game.Game.UpdateGameField() und gui.AdvancedGui.zeichnen_preview().

7.17.4.10 int world.Spielfeld.score

Spielstand in Punkten

Definiert in Zeile 47 der Datei Spielfeld.java.

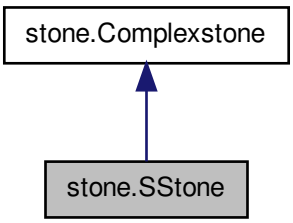
Wird benutzt von world.Spielfeld.getScore(), world.Spielfeld.reinit(), world.Spielfeld.removefulllines(), world.Spielfeld.setScore(), world.Spielfeld.Spielfeld(), world.Spielfeld.toString() und gui.AdvancedGui.zeichnen().

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

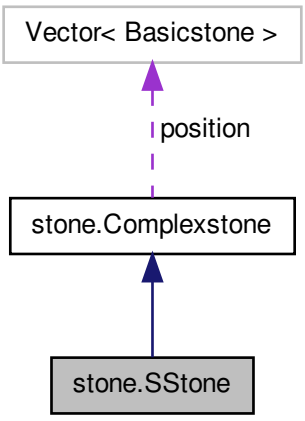
- src/world/Spielfeld.java

7.18 stone.SStone Klassenreferenz

Klassendiagramm für stone.SStone:



Zusammengehörigkeiten von stone.SStone:



Öffentliche Methoden

- SStone (int xpos_, int ypos_)

7.18.1 Ausführliche Beschreibung

Stein in Form eines S

Definiert in Zeile 12 der Datei SStone.java.

7.18.2 Beschreibung der Konstruktoren und Destruktoren

7.18.2.1 stone.SStone.SStone (int *xpos_*, int *ypos_*)

Der Konstruktor für einen S-förmigen Stein aus 2 waagerechten Grundsteinen und 2 waagerechten Grundsteinen eine Position nach links versetzt in dern nächsten Zeile

Parameter

*xpos_*	X-Position des Mittelpunktes des neuen Steins
*ypos_*	Y-Position des Mittelpunktes des neuen Steins

Definiert in Zeile 20 der Datei SStone.java.

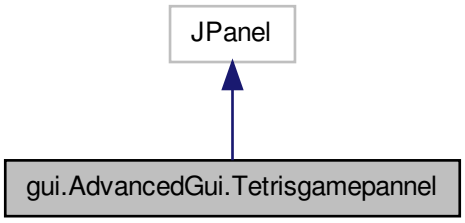
Benutzt stone.Complexstone.position, stone.Complexstone.xpos und stone.Complexstone.-ypos.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

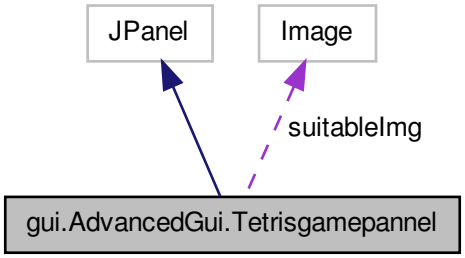
- src/stone/SStone.java

7.19 gui.AdvancedGui.Tetrisgamepannel Klassenreferenz

Klassendiagramm für gui.AdvancedGui.Tetrisgamepannel:



Zusammengehörigkeiten von gui.AdvancedGui.Tetrisgamepannel:



Öffentliche Methoden

- void paintComponent (Graphics g)

Private Attribute

- Image suitableImg

  Der Zeichenpuffer speziell für diese Instanz.

Statische private Attribute

- static final long serialVersionUID = 1L

  damit alle funktionen eines jpanel implementiert sind

7.19.1 Ausführliche Beschreibung

Eigene Zeichenfläche mit eigenem Buffer um mehrere unabhaengige Zeichenflächen zu unterstützen

Definiert in Zeile 298 der Datei AdvancedGui.java.

7.19.2 Dokumentation der Elementfunktionen

7.19.2.1 void gui.AdvancedGui.Tetrisgamepannel.paintComponent (Graphics *g*)

Funktion von JPanel - überladen damit wir eigene Grafiken zeichen können

Definiert in Zeile 308 der Datei AdvancedGui.java.

Benutzt gui.AdvancedGui.Tetrisgamepannel.suitableImg.

7.19.3 Dokumentation der Datenelemente

7.19.3.1 final long gui.AdvancedGui.Tetrisgamepannel.serialVersionUID = 1L [static, private]

damit alle funktionen eines jpanel implementiert sind

Definiert in Zeile 303 der Datei AdvancedGui.java.

7.19.3.2 Image gui.AdvancedGui.Tetrisgamepannel.suitableImg [private]

Der Zeichenpuffer speziell für diese Instanz.

Definiert in Zeile 301 der Datei AdvancedGui.java.

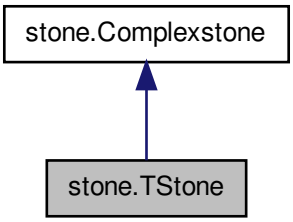
Wird benutzt von gui.AdvancedGui.init() und gui.AdvancedGui.Tetrisgamepannel.paint-Component().

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

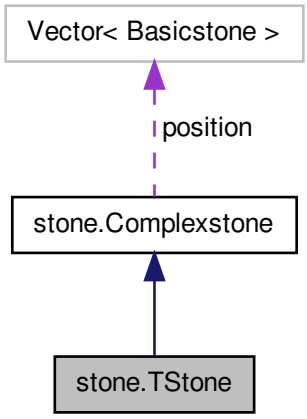
- src/gui/AdvancedGui.java

7.20 stone.TStone Klassenreferenz

Klassendiagramm für stone.TStone:



Zusammengehörigkeiten von stone.TStone:



Öffentliche Methoden

- TStone (int xpos_, int ypos_)

7.20.1 Ausführliche Beschreibung

Stein in Form eines T

Definiert in Zeile 11 der Datei TStone.java.

7.20.2 Beschreibung der Konstruktoren und Destruktoren

7.20.2.1 stone.TStone.TStone (int *xpos_*, int *ypos_*)

Der Konstruktor für einen T-förmigen Stein aus 3 waagerechten Grundsteinen und einem in der Mitte darunter

Parameter

*xpos_*	X-Position des Mittelpunktes des neuen Steins
*ypos_*	Y-Position des Mittelpunktes des neuen Steins

Definiert in Zeile 19 der Datei TStone.java.

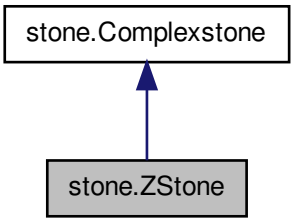
Benutzt stone.Complexstone.position, stone.Complexstone.xpos und stone.Complexstone.ypos.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

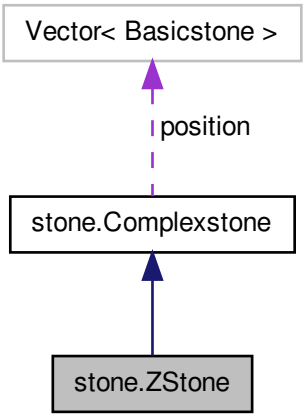
- src/stone/TStone.java

7.21 stone.ZStone Klassenreferenz

Klassendiagramm für stone.ZStone:



Zusammengehörigkeiten von stone.ZStone:



Öffentliche Methoden

- ZStone (int xpos_, int ypos_)

7.21.1 Ausführliche Beschreibung

Stein in Form eines Z

Definiert in Zeile 11 der Datei ZStone.java.

7.21.2 Beschreibung der Konstruktoren und Destruktoren

7.21.2.1 stone.ZStone.ZStone (int *xpos_*, int *ypos_*)

Der Konstruktor für einen Z-förmigen Stein aus 2 waagerechten Grundsteinen und 2 waagerechten Grundsteinen eine Position nach rechts versetzt in der nächsten Zeile

Parameter

*xpos_*	X-Position des Mittelpunktes des neuen Steins
*ypos_*	Y-Position des Mittelpunktes des neuen Steins

Definiert in Zeile 20 der Datei ZStone.java.

Benutzt stone.Complexstone.position, stone.Complexstone.xpos und stone.Complexstone.-ypos.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- src/stone/ZStone.java

Kapitel 8

Datei-Dokumentation

8.1 src/controller/CHighScoreEntry.java-Dateireferenz

Klassen

- class controller.CHighScoreEntry

Pakete

- package controller

8.2 src/controller/CHighScores.java-Dateireferenz

Klassen

- class controller.CHighScores

  enthält einen Vektor mit den Einträgen

Pakete

- package controller

8.3 src/controller/JHighscoreFileIO.java-Dateireferenz

Klassen

- class controller.JHighscoreFileIO

Pakete

- package controller

8.4 src/controller/Logic.java-Dateireferenz

Klassen

- class controller.Logic

Pakete

- package controller

8.5 src/game/Config.java-Dateireferenz

Klassen

- class game.Config

Pakete

- package game

8.6 src/game/Game.java-Dateireferenz

Klassen

- class game.Game

Pakete

- package game

8.7 src/gui/AdvancedGui.java-Dateireferenz

Klassen

- class gui.AdvancedGui
- class gui.AdvancedGui.Tetrisgamepannel

Pakete

- package gui

8.8 src/stone/Basicstone.java-Dateireferenz

Klassen

- class stone.Basicstone

Pakete

- package stone

8.9 src/stone/Complexstone.java-Dateireferenz

Klassen

- class stone.Complexstone

Pakete

- package stone

8.10 src/stone/IStone.java-Dateireferenz

Klassen

- class stone.IStone

Pakete

- package stone

8.11 src/stone/JStone.java-Dateireferenz

Klassen

- class stone.JStone

Pakete

- package stone

8.12 src/stone/LStone.java-Dateireferenz

Klassen

- class stone.LStone

Pakete

- package stone

8.13 src/stone/MoonStone.java-Dateireferenz

Klassen

- class stone.MoonStone

Pakete

- package stone

8.14 src/stone/NStone.java-Dateireferenz

Klassen

- class stone.NStone

Pakete

- package stone

8.15 src/stone/OStone.java-Dateireferenz

Klassen

- class stone.OStone

Pakete

- package stone

8.16 src/stone/PlusStone.java-Dateireferenz

Klassen

- class stone.PlusStone

Pakete

- package stone

8.17 src/stone/SStone.java-Dateireferenz

Klassen

- class stone.SStone

Pakete

- package stone

8.18 src/stone/TStone.java-Dateireferenz

Klassen

- class stone.TStone

Pakete

- package stone

8.19 src/stone/ZStone.java-Dateireferenz

Klassen

- class stone.ZStone

Pakete

- package stone

8.20 src/world/Spielfeld.java-Dateireferenz

Klassen

- class world.Spielfeld

Pakete

- package world