

Schnittstelle zu GUI

UML-Diagramm mit allen für die GUI relevanten Attribute und Methoden:

Spielfeld
- pinn - kontrolle - ziel - anzahlPinns - anzahlReihen - anzahlFarben - aktiveReihe
+ pinnFarbe() + kontrolle() + wertung()

Attribute:

Name	Datentyp	Beschreibung
pinn	int[Position][Reihe]*	Array mit farbigen (1 bis „anzahlFarben“) oder keinen(0) Pinns
kontrolle	int[Position][Reihe]*	Array mit jeweils weißen (1) schwarzen (2) oder keinen (0) Pinns
ziel	int[Position]*	Array mit farbigen (1 bis „anzahlFarben“)
anzahlPinns	int	Anzahl der verwendeten Pinns
anzahlReihen	int	Anzahl der verwendeten Reihen
anzahlFarben	int	Anzahl der verwendeten Farben
aktiveReihe	int	Aktuell aktive Reihe*

* alle Positions- bzw. Reihennummern beginnen bei 0 und Enden bei „anzahlPinns“-1 bzw. „anzahlReihen“-1!

** Farbe „0“ steht immer für einen leeren Steckplatz!

Konstruktor:

Eingabe	int – Level (Zahl von 1 bis 3 ggf. 1 bis 4)
Beschreibung	Bereitet das Spiel gemäß Levelangabe vor: Level 1: 4 Pinns, 8 Reihen, 6 Farben Level 2: 5 Pinns, 9 Reihen, 6 Farben Level 3: 6 Pinns, 10 Reihen, 6 Farben (Level 4: 4 Pinns, 8 Reihen, 6 Farben)

Methoden:

Name	pinnFarbe()
Eingabe	int – Pinnposition innerhalb der Reihe (beginnend mit 0) int – Reihe (beginnend mit 0)
Rückgabe	–
Beschreibung	Wird bei Klicken auf einen leeren Steckplatz bzw. Pinn aufgerufen. Falls sich der Pinn in der aktuell aktiven Reihe befindet, wird seine Farbe (Zahl außer 0) eins weiter geschaltet. Dabei kann ein Pinn nicht wieder auf 0 (leerer Steckplatz) zurückgesetzt werden. Falls sich der Pinn außerhalb der aktiven Reihe befindet, passiert nichts.

Name	kontrolle()
Eingabe	–
Rückgabe	true – Spiel zu Ende (sowohl gewonnen als auch wenn Zahl der Versuche ausgeht) false – Spiel geht weiter
Beschreibung	Wird bei Klicken auf den Kontrollbutton aufgerufen. Falls die aktuell aktive Reihe voll ist (ganze Reihe mit Pinnns belegt, keine leeren Steckplätze), wird sie ausgewertet und weiße (1) und schwarze (2) Kontrollpinns für diese Reihe gesetzt. Falls die Reihe nicht voll ist, geschieht nichts. Es wird rückgemeldet, ob das Spiel nach der Kontrolle beendet ist.

Name	wertung()
Eingabe	–
Rückgabe	int – erreichte Punktzahl
Beschreibung	Wird am Spielende aufgerufen und gibt die erreichte Punktzahl zurück, die an den Highscore weitergereicht wird.

Falls benötigt stehen get-Methoden für alle oben genannten Attribute zur Verfügung:

Name	Eingabe	Rückgabe
getPinn	–	int[][] – Alle Pinns
getPinn	int – Pinnposition	int[] – Pinns der aktiven Reihe
getPinn	int – Pinnposition int – Reihe	int – Pinn an Position
getKontrolle	–	int[][] – Alle Pinns
getKontrolle	int – Pinnposition	int[] – Pinns der aktiven Reihe
getKontrolle	int – Pinnposition int – Reihe	int – Pinn an Position
getZiel	–	int[] – Alle Pinns
getZiel	int – Pinnposition	int – Pinn an Position
getAnzahlPinns	–	int – Anzahl der Pinns
getAnzahlReihen	–	int – Anzahl der Reihen
getAnzahlFarben	–	int – Anzahl der Farben
getAktiveReihe	–	int – Aktuell aktive Reihe