

Objektorientiertes Programmieren mit GPanel-Grafik

Das Projekt MeinPark, VI

Aufgabe 13: Tastatursteuerung

Der Schneemann hat keine Chance, die Sonne gewinnt immer. Das soll nun geändert werden, indem der Schneemann mit Hilfe der Pfeiltasten vom Benutzer gesteuert wird.

- a) Hierfür brauchst du zunächst in der Klasse `Schneemann` eine Methode

```
public void bewege(int nachX, int nachY)
```

die den Schneemann löscht und an der Position `(nachX|nachY)` neu zeichnet.

- b) Um den Schneemann mit der Tastatur zu steuern, muss in der Klasse `MeinPark` permanent geprüft werden, ob eine Pfeiltaste gedrückt wurde. Wenn ja, muss sich der Schneemann um (zum Beispiel) drei Pixel in die entsprechende Richtung bewegen.

Erweitere dafür deine Klasse `MeinPark` um eine Methode

```
private void bewegeSchneemannFallsTasteGedrueckt()
```

die genau, wie z.B. die Methode zur Bewegung der Sonne, immer wieder aufgerufen wird. Diese muss (vorausgesetzt dein Schneemannobjekt heißt `snowman`) folgenden Code beinhalten:

```
int keyCode = myGPanel.getKeyCode();
int x = snowman.getPosX();
int y = snowman.getPosY();
switch (keyCode) {
    case KeyEvent.VK_LEFT:
        snowman.bewege(x-1,y);
        break;
    case KeyEvent.VK_RIGHT:
        snowman.bewege(x+1,y);
        break;
    case KeyEvent.VK_UP:
        snowman.bewege(x,y-1);
        break;
    case KeyEvent.VK_DOWN:
        snowman.bewege(x,y+1);
        break;
    ...
}
```

Informiere dich in der Java-Dokumentation

<https://docs.oracle.com/javase/7/docs/api/>

über die Konstanten und Methoden der Klasse `KeyEvent` im Package `java.awt.event` und vervollständige den Code.

Teste dein Programm. Die Blumen sollten vom Schneemann nicht gelöscht werden.

Objektorientiertes Programmieren mit GPanel-Grafik

Das Projekt MeinPark, VI

Aufgabe 14: Sieg oder Niederlage

Jetzt kann der Schneemann zwar fliehen, aber noch immer nicht gewinnen. Dies soll geschehen, wenn er eine bestimmte Blume (z.B. die gelbe „Sonnenblume“) pflückt. „Pflücken“ soll dabei bedeuten, dass die Distanz zwischen dem Mittelpunkt der mittleren Kugel des Schneemanns und dem unteren Ende des Blumenstängels kleiner wird, als der Radius der mittleren Kugel.

- a) Überprüfe nach jeder Schneemannbewegung die Distanz zwischen Schneemann und (zum Beispiel) der gelben Blume. Ist sie kleiner als der Radius der mittleren Schneemannkugel, soll das Bild „eingefroren“ werden.

Fehlt nur noch, dass am Ende des Spiels ausgegeben wird, wer gewonnen hat. Dazu kann die Methode `text` der Klasse `GPanel` benutzt werden, z.B.:

```
myGPanel.text(600,300,"Die Sonne hat gewonnen");
```

Importierst du die Klasse `java.awt.Font`, kann z.B. mit

```
myGPanel.font(new Font( "Verdana", Font.BOLD, 20 ));
```

die Schriftart bzw. die Schriftgröße verändert werden. (Genauere Informationen liefert die Dokumentation der Klasse `Font` im Package `java.awt`.)

- b) Ändere dein Programm zum Schluss nur noch so ab, dass der Sieger auf dem `GPanel` ausgegeben wird, sobald er feststeht.

Aufgabe 15: Erweiterungen

Lust auf mehr? Das Projekt ist nach Belieben erweiterbar. Zum Beispiel könnte man

- vor dem Start des Spiels eine Spielanleitung anzeigen – das Spiel beginnt dann erst nach einem Mausklick
- einen 2-Player-Modus einbauen, indem man auch die Sonne mit der Tastatur steuern lässt
- verschiedene Level einbauen, die sich z.B. in der Geschwindigkeit der Sonne oder des Schneemanns unterscheiden
- den Schneemann erst gewinnen lassen, wenn er z.B. drei beliebige Blumen eingesammelt hat
- einen Timer einbauen, so dass der Schneemann nur eine bestimmte Zeit zur Verfügung hat, um die Blume zu pflücken
- Sound einbauen

Du hast sicher noch bessere Ideen!