pyenguin - Zeichnen mit Python Teil III

1 Bewegung

Erinnerung: Der Koordinatenursprung `(0,0)` liegt in der Mitte eines Objektes. Eine Änderung der x-Koordinate, die größer `0` ist (z.B. `5`), ist eine Bewegung nach rechts. Eine negative Änderung bedeutet eine Bewegung nach links (z.B. `-5`).

Eine Änderung der y-Koordinate, die größer `0` ist, ist eine Bewegung nach **unten!**.
Dies ist zunächst verwirrend, allerdings ist dies bei vielen Computerprogrammen so.
Eine negative Änderung bedeutet demzufolge eine Bewegung nach oben!

```
1 box = Rechteck(10, 10, 50, 50, ROT) # Ändert die x- und >
   ↪ y-Koordinate um 5 Pixel
2 box.aendere_position(5, 5)
```

Dies ist also eine Bewegung um jeweils 5 Pixel nach rechts unten!

1.1 Objekte positionieren

Es gibt verschiedene Möglichkeiten Objekte, wie Kreise, Rechtecke, Bilder, etc.. zu platzieren:

1.1.1 Mit links, rechts, oben, unten:

Bei der Definition wird meistens die x-, y-Koordinate mit angegeben. Diese kann aber auch noch leicht nachträglich geändert werden, indem der Abstand zum linken, rechten, oberen oder unteren Rand angegeben wird.

1.1.2 Die Objekte-Mitte festlegen:

Man kann auch die Objektmitte direkt setzen:

1.2 Die x-,y-Koordinaten ändern:

1.2.1 Relative Änderung

Die Position um einen Wert ändern. .
box = Rechteck(10, 10, 50, 50, ROT) # Ändert die x- und y-Koordinate um 5 Pixel box.aendere_position(5, 5)

1.2.2 Absolute Position setzen

Die Position auf einen Wert setzen. .
box = Rechteck(10, 10, 50, 50, ROT) # Setzt die x- und y-Koordinate auf die gegebenen Werte box.setze_position(50, 50)

Hinweis: Diese Bewegungsmöglichkeiten sind ähnliche wie die in Scratch. verhält sich genau wie und .
verhält sich genau wie und .

1.3 Objekte zentrieren:

Objekte können horizontal, vertikal oder in beide Richtungen zentriert werden.

1.4 Geschwindigkeit festlegen und bewegen

Man kann Objekte auch mithilfe von bewegen. Dabei wird das Objekt um die mit definierte Distanz (hier: 6 Pixel nach rechts und 6 Pixel nach unten).
box = Rechteck(10, 10, 50, 50, ROT) # Die x- und y-Geschwindigkeit setzen box.setze_geschwindigkeit(6,6)
# Bewegt die Box bei jedem Aufruf von bewege() um 6 Pixel box.bewege()

1.5 Position abrufen

1.5.1 x- und y-Koordinaten

Man kann die x- und y-Koordinaten eines Objektes ganz einfach abfragen
Box an der Stelle 10x10 box = Rechteck(10, 10, 50, 50, ROT) # x- und y-Koordinate abfragen und ausgeben print(box.x, box.y)

1.5.2 Abstand zum Rand

Genau wie man den Abstand zum linken, rechten, oberen oder unteren Rand setzen kann, kann man ihn auch abfragen. Das Gleiche gilt auch für die Objektmitte.

1.5.3 Größe des umgebenden Rechtecks

Wie zuvor beschrieben, ist jedes Objekt von einem unsichtbaren Rechteck eingegrenzt. Man kann auch dessen Breite und Höhe abfragen.

2 Aufgaben

1. Bewege ein Objekt mit den verschiedenen Funktionen über die Spielfläche.
2. Frage die Position von Objekten und derren Abstand zum Rand ab. Ändert sich die Postion wie erwartet, wenn du sie bewegst? Z.B. sollte sich die x-Koordinate um 20 ändern, wenn aufrufst.