

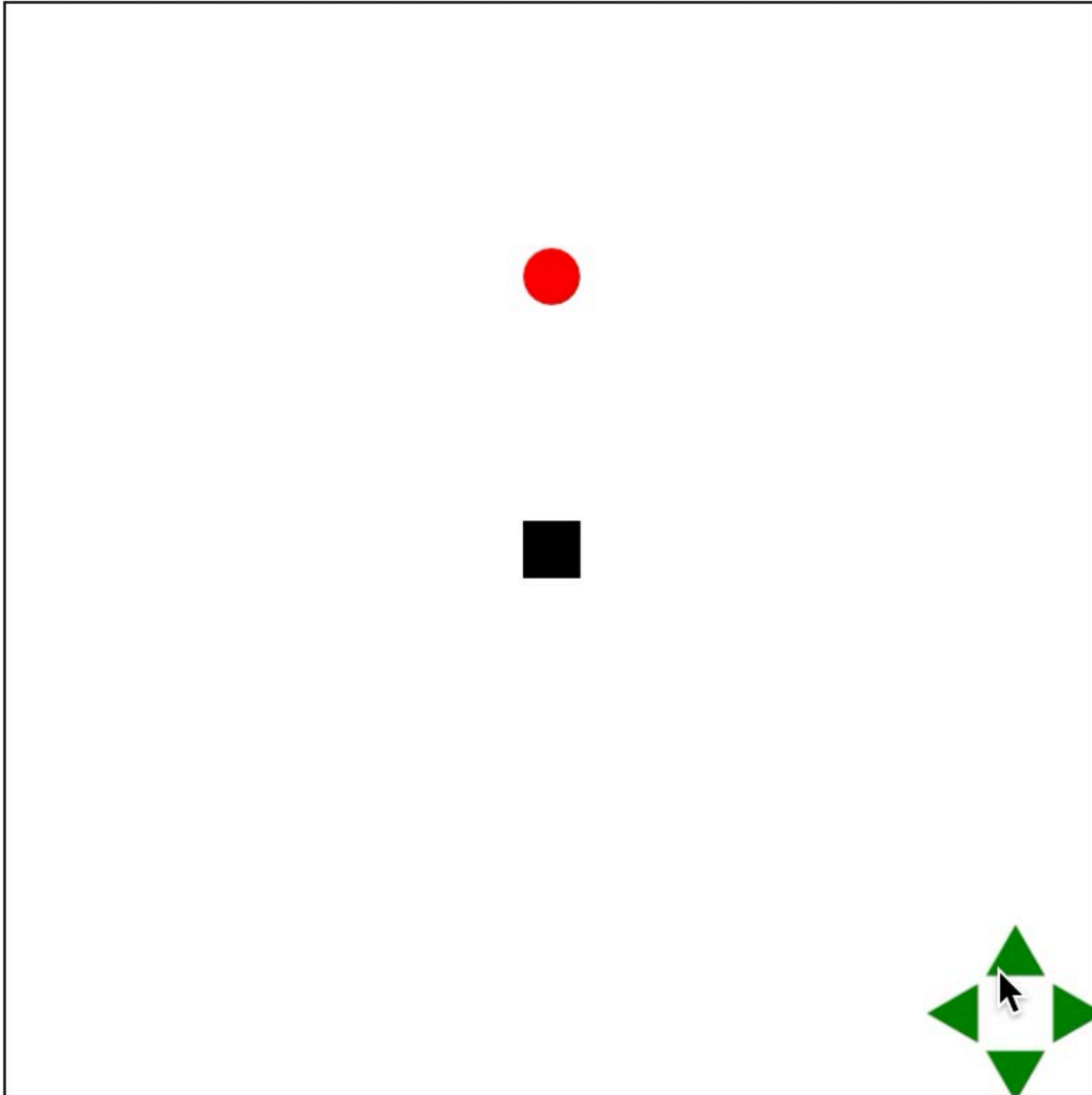


Programmieren
lernen mit Python

Snake-Spiel (1/3)

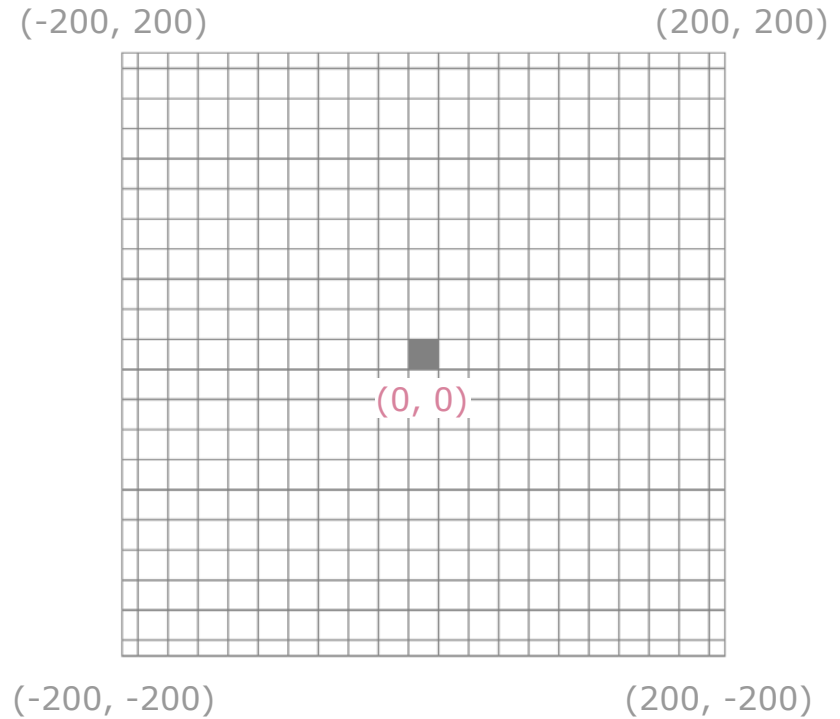
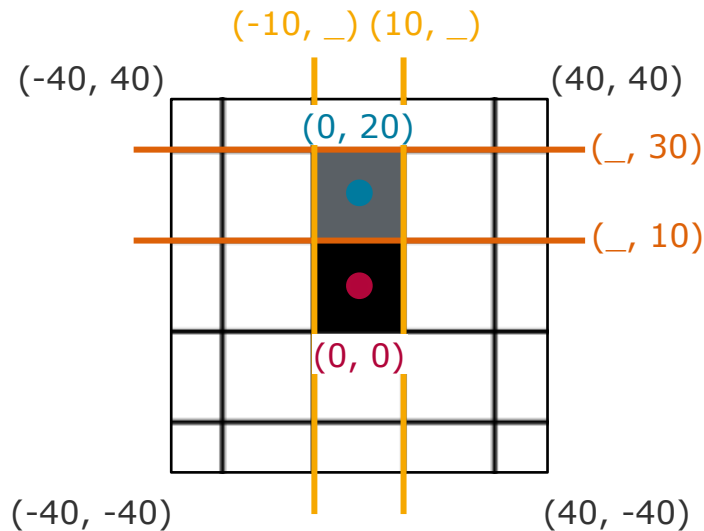
Kira Grammel, Nina Ihde, Sebastian Serth & Selina Reinhard
Hasso-Plattner-Institut
Universität Potsdam

Das Snake-Spiel



Schlängenkopf

```
1 from turtle import *
2
3
4 shape("square")
5 color("black")
6 penup()
```

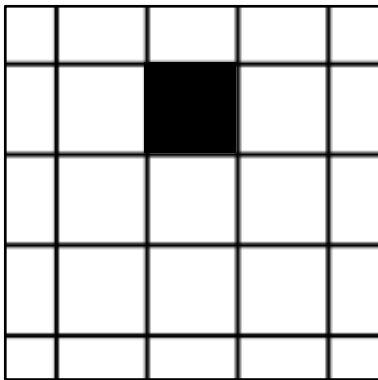


Bewegung im Raster

```

1 from turtle import *
2
3
4 shape("square")
5 color("black")
6 penup()
7
8 goto(0, 20)

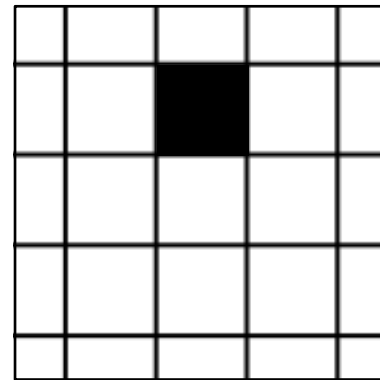
```



```

1 from turtle import *
2
3
4 shape("square")
5 color("black")
6 penup()
7 speed(0)
8 goto(0, 20)

```



Schlängenkopf

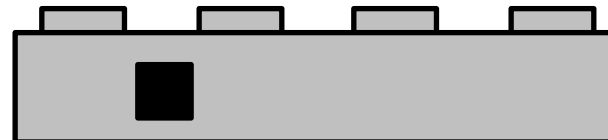
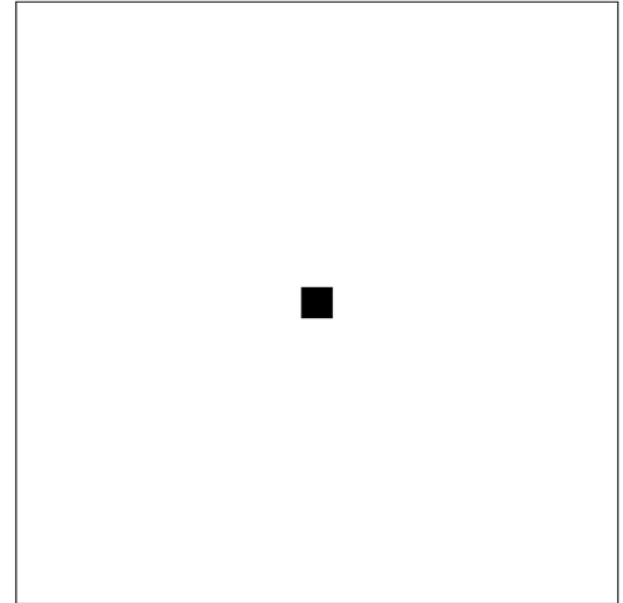
```
1 from turtle import *
2
3
4 shape("square")
5 color("black")
6 penup()
7 speed(0)
8 goto(0, 0)
9 direction = "stop"
```



Bewegung initial
auf "stop" setzen

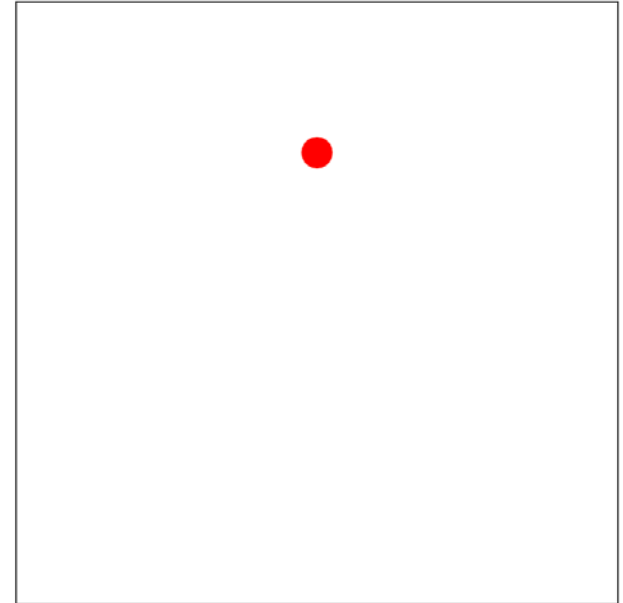
Schlängenkopf

```
1 from turtle import *
2
3
4 shape("square")
5 color("black")
6 penup()
7 speed(0)
8 goto(0, 0)
9 direction = "stop"
```



Schlange und Essen

```
1 from turtle import *
2
3
4 shape("square")
5 color("black")
6 penup()
7 speed(0)
8 goto(0, 0)
9 direction = "stop"
10
11
12 shape("circle")
13 color("red")
14 penup()
15 speed(0)
16 goto(0, 100)
```

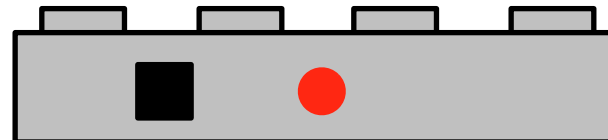
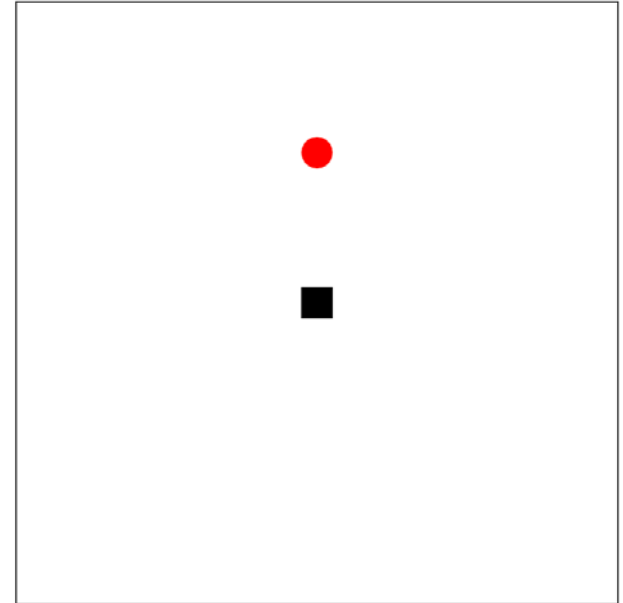


Schlange und Essen

```

1 from turtle import *
2
3 kopf = Turtle()
4 kopf.shape("square")
5 kopf.color("black")
6 kopf.penup()
7 kopf.speed(0)
8 kopf.goto(0, 0)
9 kopf.direction = "stop"
10
11 essen = Turtle()
12 essen.shape("circle")
13 essen.color("red")
14 essen.penup()
15 essen.speed(0)
16 essen.goto(0, 100)

```

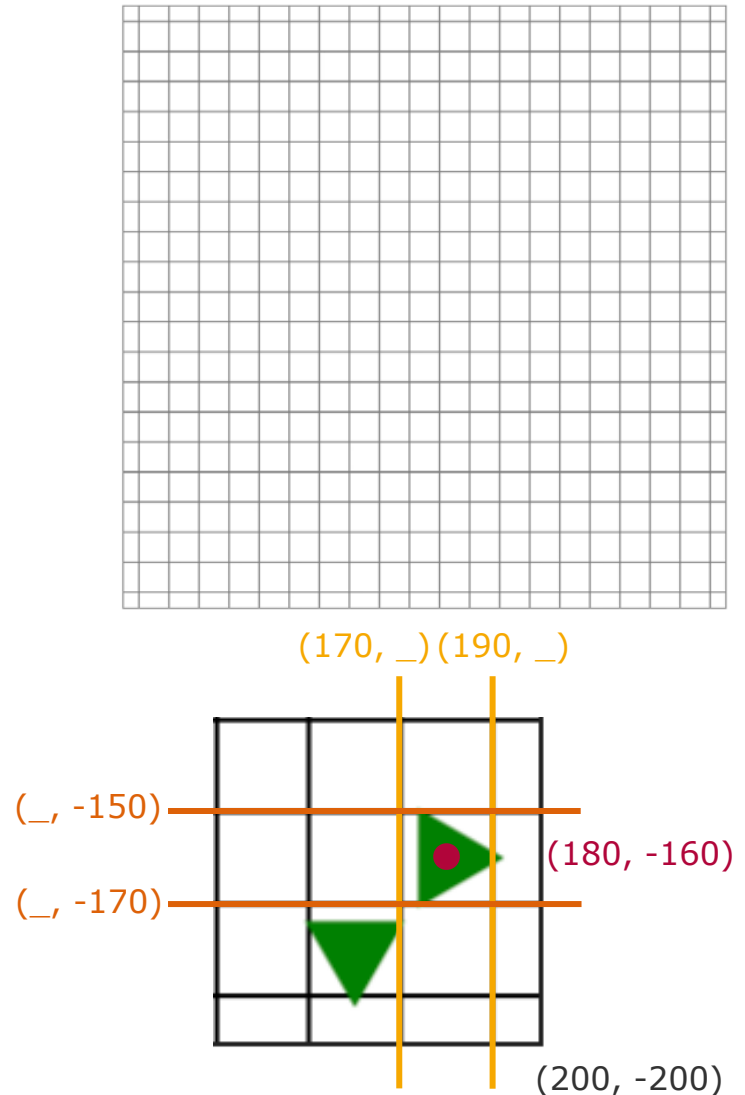


Grafische Steuerung

```

1  from turtle import *
2
3  rechts = Turtle()
4  rechts.shape("triangle")
5  rechts.color("green")
6  rechts.speed(0)
7  rechts.penup()
8  rechts.goto(180, -160)
9
10
11 unten = Turtle()
12 unten.shape("triangle")
13 unten.color("green")
14 unten.right(90)
15 unten.speed(0)
16 unten.penup()
17 unten.goto(160, -180)

```

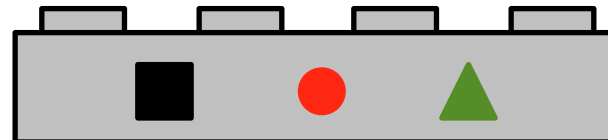
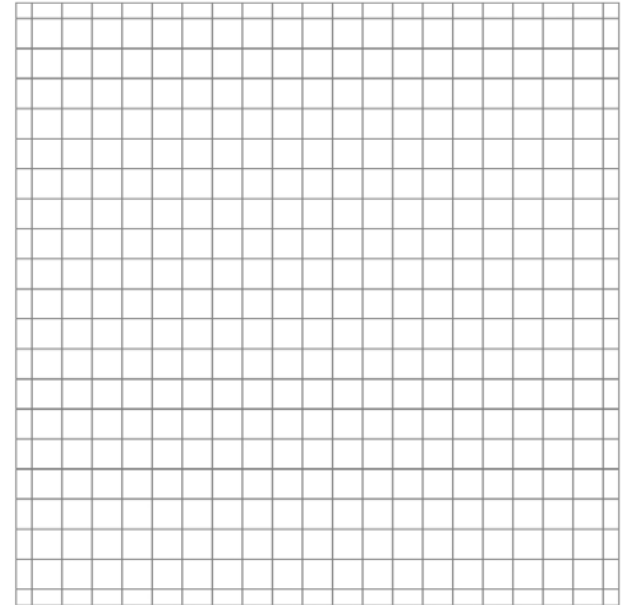


Grafische Steuerung

```

1 from turtle import *
2
3 rechts = Turtle()
4 rechts.shape("triangle")
5 rechts.color("green")
6 rechts.speed(0)
7 rechts.penup()
8 rechts.goto(180, -160)
9
10
11 unten = Turtle()
12 unten.shape("triangle")
13 unten.color("green")
14 unten.right(90)
15 unten.speed(0)
16 unten.penup()
17 unten.goto(160, -180)

```

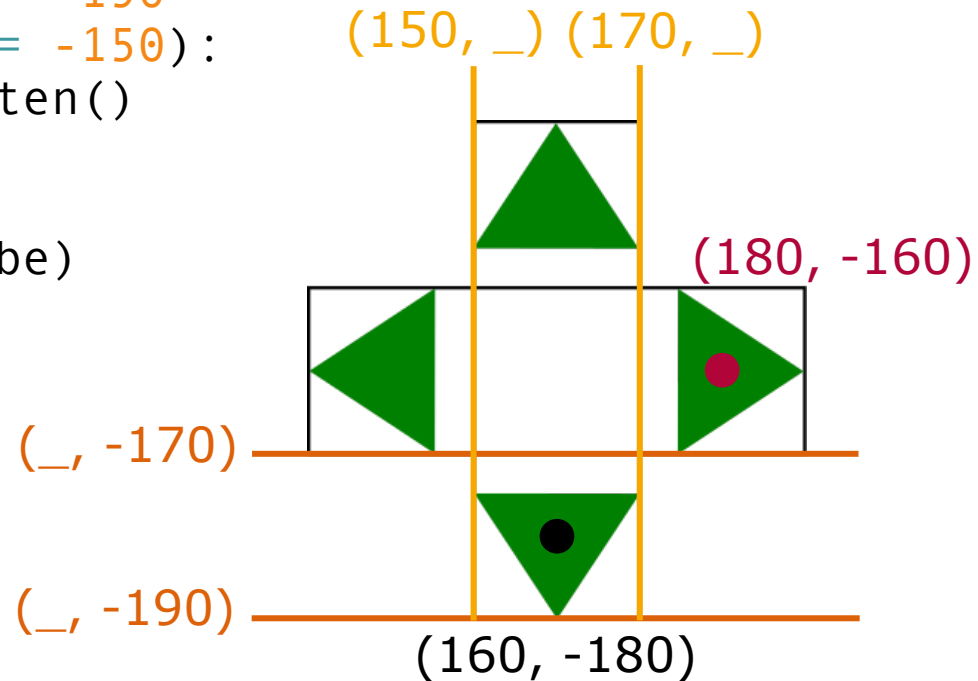


Steuerungseingabe

```

1 from turtle import *
2
3 def interpretiere_eingabe(x, y):
4     if (x >= 150 and x <= 170
5         and y >= -190 and y <= -170):
6         nach_unten_ausrichten()
7     elif (x >= 170 and x <= 190
8           and y >= -170 and y <= -150):
9         nach_rechts_ausrichten()
10    # ...
11 onclick(interpretiere_eingabe)

```



Steuerungseingabe

```

1 from turtle import *
2
3 def interpretiere_eingabe(x, y):
4     if (x >= 150 and x <= 170
5         and y >= -190 and y <= -170):
6         nach_unten_ausrichten()
7     elif (x >= 170 and x <= 190
8           and y >= -170 and y <= -150):
9         nach_rechts_ausrichten()
10    # ...
11 onclick(interpretiere_eingabe)

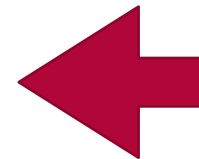
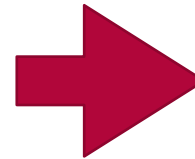
```



onclick(interpretiere_eingabe)

Richtungsänderung

```
1 def nach_unten_ausrichten():
2     if kopf.direction != "up":
3         kopf.direction = "down"
4
5 def nach_rechts_ausrichten():
6     if kopf.direction != "left":
7         kopf.direction = "right"
8
9 def nach_links_ausrichten():
10    if kopf.direction != "right":
11        kopf.direction = "left"
12
13 def nach_oben_ausrichten():
14    if kopf.direction != "down":
15        kopf.direction = "up"
```



Bewegung des Schlangenkopfes

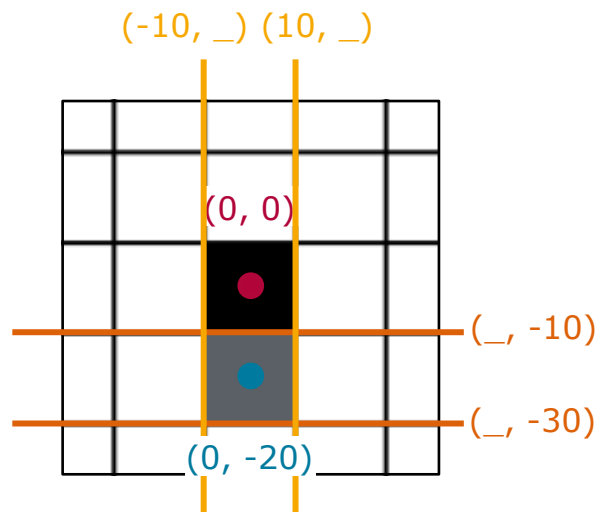
```

1 def kopf_bewegen():
2     if kopf.direction == "down":
3         y = kopf.ycor()
4         kopf.sety(y - 20)
5
6     if kopf.direction == "right":
7         x = kopf.xcor()
8         kopf.setx(x + 20)
9     # ...

```

`kopf.ycor()`
gibt y-Koordinate
vom Kopf zurück

`kopf.sety(wert)`
setzt y-Koordinate
vom Kopf auf `wert`



Bewegung des Schlangenkopfes

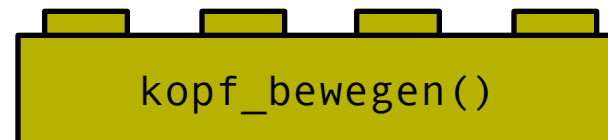
```

1 def kopf_bewegen():
2     if kopf.direction == "down":
3         y = kopf.ycor()
4         kopf.sety(y - 20)
5
6     if kopf.direction == "right":
7         x = kopf.xcor()
8         kopf.setx(x + 20)
9     # ...

```

`kopf.ycor()`
gibt y-Koordinate
vom Kopf zurück

`kopf.sety(wert)`
setzt y-Koordinate
vom Kopf auf `wert`

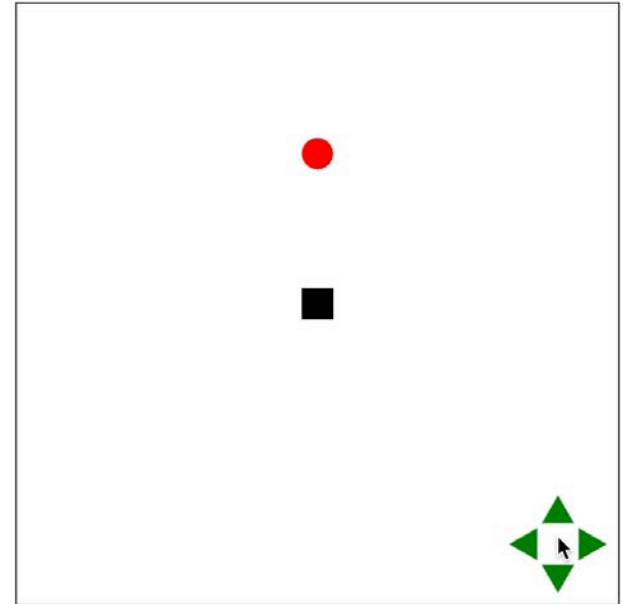


Steuerungseingabe

```

1 from turtle import *
2
3 def interpretiere_eingabe(x, y):
4     if (x >= 150 and x <= 170
5         and y >= -190 and y <= -170):
6         nach_unten_ausrichten()
7     elif (x >= 170 and x <= 190
8           and y >= -170 and y <= -150):
9         nach_rechts_ausrichten()
10    # ...
11    kopf_bewegen()
12 onclick(interpretiere_eingabe)

```



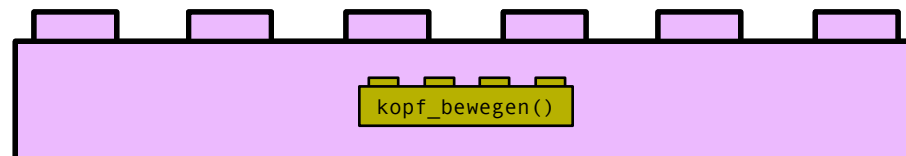
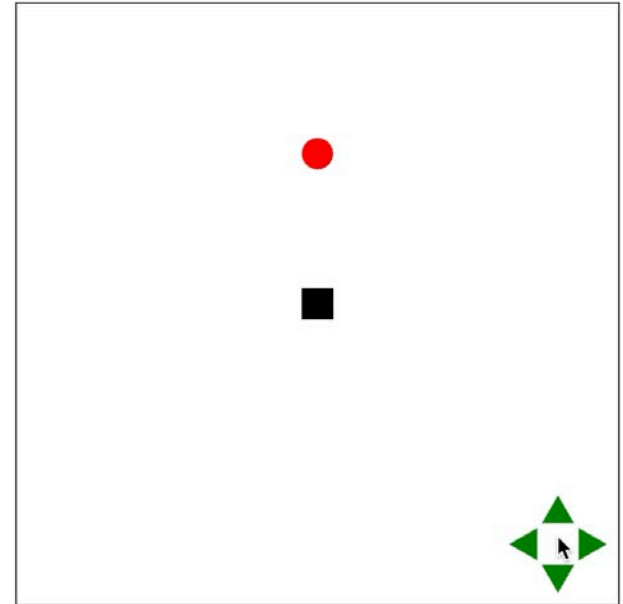
onclick(interpretiere_eingabe)

Steuerungseingabe

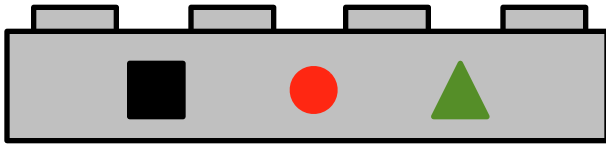
```

1 from turtle import *
2
3 def interpretiere_eingabe(x, y):
4     if (x >= 150 and x <= 170
5         and y >= -190 and y <= -170):
6         nach_unten_ausrichten()
7     elif (x >= 170 and x <= 190
8           and y >= -170 and y <= -150):
9         nach_rechts_ausrichten()
10    # ...
11    kopf_bewegen()
12 onclick(interpretiere_eingabe)

```



- Verschiedene Grafiken:



- Wichtige Funktionsaufrufe:

