

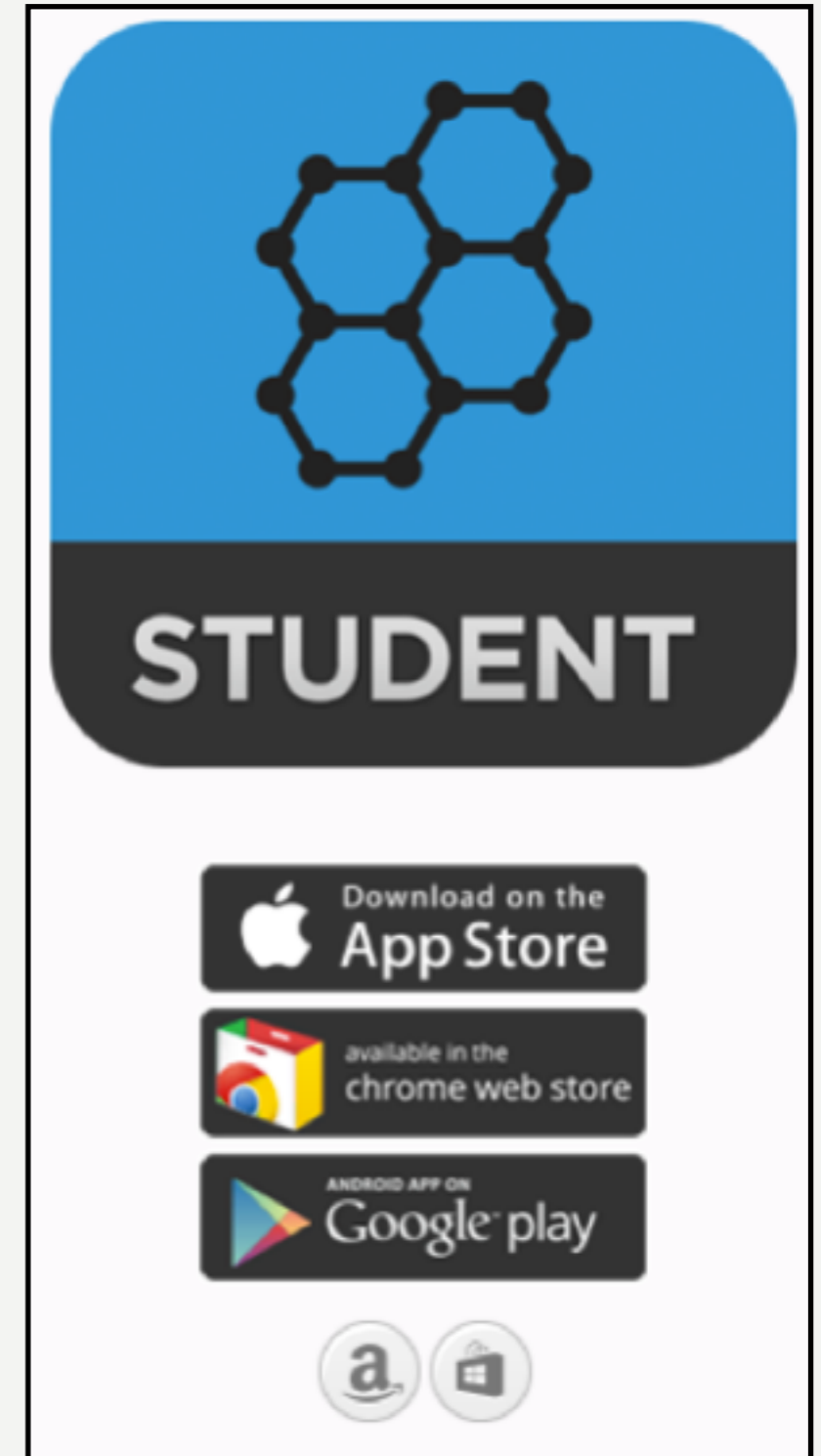
# Einführung in die Programmierung für Nebenfach Medieninformatik

Beat Rossmly, Michael Kirsch



- Eure Mitarbeit ist uns wichtig!
- Installiert euch dazu die kostenlose App „Socrative Student“ auf Eurem Smartphone oder nutzt das Webinterface unter <http://b.socrative.com/login/student/>
- Anonymer Login über den Raumnamen:

***MSMJOKRQ***



# Grafikobjekte

Verwendung von Java2D, Canvas und der  
Paint-Methode

Beat Rossmay, Michael Kirsch

Quiztime :)

## Quiz: Was bedeutet die folgende Code-Zeile?

```
public class Spielfenster extends JFrame {...}
```

A: Die Klasse Spielfenster implementiert das Interface JFrame

B: Spielfenster ist ein primitiver Datentyp

C: JFrame erweitert die Klasse Spielfenster

D: Die Klasse Spielfenster erbt von JFrame und besitzt somit alle Eigenschaften und Methoden eines JFrame

## Quiz: Was bedeutet die folgende Code-Zeile?

```
public class Spielfenster extends JFrame {...}
```

A: Die Klasse Spielfenster implementiert das Interface JFrame

B: Spielfenster ist ein primitiver Datentyp

C: JFrame erweitert die Klasse Spielfenster

D: Die Klasse Spielfenster erbt von JFrame und besitzt somit alle Eigenschaften und Methoden eines JFrame



## Quiz: Welche der folgenden Aussagen bzgl. JPanel im Kontext von Java Swing sind/ist richtig?

A: JPanels werden beim Bau von Fertighäusern verwendet

B: Ein JPanel ist eine Komponente, die mit weiteren Komponenten „bestückt“ werden kann

C: JFrame und JPanel sind das gleiche

D: Ein JPanel kann einen JFrame enthalten



## Quiz: Welche der folgenden Aussagen bzgl. JPanel im Kontext von Java Swing sind/ist richtig?

A: JPanels werden beim Bau von Fertighäusern verwendet

B: Ein JPanel ist eine Komponente, die mit weiteren Komponenten „bestückt“ werden kann

C: JFrame und JPanel sind das gleiche

D: Ein JPanel kann einen JFrame enthalten





## Quiz: Welche Aussage bzgl. KeyListener bei Java Swing sind/ist falsch?

A: KeyListener ermöglichen die Abfrage von allen mögliche Tastatureingaben

B: KeyListener können „nur“ Zustände bei Tastendrücken erkennen

C: KeyListener können ebenfalls bei JButtons verwendet werden

D: KeyListener besitzen als Parameter ein Key-Event



## Quiz: Welche Aussage bzgl. KeyListener bei Java Swing sind/ist falsch?

A: KeyListener ermöglichen die Abfrage von allen mögliche Tastatureingaben

B: KeyListener können „nur“ Zustände bei Tastendrücken erkennen

C: KeyListener können ebenfalls bei JButtons verwendet werden

D: KeyListener besitzen als Parameter ein Key-Event



## I. Einleitung

1. Wie geht es weiter?

## II. Theorie

1. Der this-Operator
2. Keylistener (2)
3. Die paint-Methode
4. Threads & Animationen
5. Try/Catch

## III. Anwendung

1. Praxis in Eclipse

## IV. Verknüpfung

1. Tutorials

## V. Ausblick

1. Nächste Vorlesung
2. Übung

## Einführung in die Programmierung

Processing	
Typen und Operatoren	Kontroll-Strukturen
Klassen und Objekte	Gültigkeit und Konventionen
Methoden	Arrays
Konstruktoren	Eingaben-verarbeitung
Animationen	...

Java	
Grundlagen aus Processing	Swing
Objekte/Klassen	Interfaces
...	Vererbung

# Einleitung

## Wie geht es weiter?



- Bisher wurde das Spielfeld erstellt und Tasteneingaben verwertet
- Dieses Mal: Paint-Methode, Threads und Fehlerbehandlung



# Theorie



- **this**-Operator
- Mit dem **this**-Operator kann auf die Klasse zugegriffen werden, die den Code enthält. Somit ist es möglich innerhalb dieser Klasse Attribute und Methoden der Klasse selbst zu verwenden.

```
16 public class Spielfenster extends JFrame { // Spielfenster leitet von JFrame ab u
17
18     Spielfeld spielfeld;
19     JLabel score;
20
21     public Spielfenster() { // In dem Konstruktor werden alle Konfigurationen von
22
23         spielfeld = new Spielfeld(); // Ein neues Objekt vom Typ Spielfeld wird e
24         score = new JLabel("Aktueller Spielstand 0 : 0 "); // Ein neues Objekt vo
25
26         score.setFont(new Font("Calibri", Font.BOLD, 20)); // Setzt den Font der
27         score.setForeground(new Color(0x0EE00)); // Setzt die Farbe auf Grün bzw.
28
29         spielfeld.setFocusable(true); // Ermöglicht, dass die Keylistener auf die
30
31         this.setSize(new Dimension(600, 600)); // Setzt die Größe des Spielfenste
32         this.setLocation(500, 100); // Setzt die Position des Spielfensters
33         spielfeld.setBackground(new Color(0x212121)); // Setzt die die Hintergrun
34
```



- KeyListener ermöglichen die Abfrage von Tastatureingaben
- Verwendung von KeyListener in zwei Varianten
  - Implementierung des Interfaces *KeyListener*
  - Implementierung in einer *anonymen inneren Klasse*
- Damit ein Fenster Key-Events abfragen kann, muss es im *Fokus* stehen  
`spielfeld.setFocusable(true); //Aktivierung der „Fokusierbarkeit“`
- KeyListener können die folgenden drei Events abfragen



```
public void keyTyped(KeyEvent e)
public void keyPressed(KeyEvent e)
public void keyReleased(KeyEvent e)
```



- Um Formen wie Kreise, Rechtecke usw. darstellen zu können, wird die „paint-Methode“ benötigt
- In ihr wird beschrieben, was “gemalt” werden soll
- Beispiel: Ein roter Kreis an der Position (x=300, y=300) mit der Größe 20x20 Pixel

@Override

```
public void paint(Graphics g) {  
    super.paint(g);  
    Graphics2D g2d = (Graphics2D) g;  
    g2d.setColor(Color.red);  
    g2d.fillOval(300, 300, 20, 20);  
}
```





- Besonderheit: Um die Inhalte erneut zu zeichnen, muss sie aufgerufen werden

```
repaint(); //Erneut die paint-Methode aufrufen
```

- Um Änderungen bei den Inhalten zu ermöglichen, können Variablen verwendet werden

```
public void paint(Graphics g) {  
    super.paint(g);  
    Graphics2D g2d = (Graphics2D) g;  
    g2d.setColor(Color.red);  
    g2d.fillOval(ballXpos, ballYpos, 20, 20);  
}
```



- Um Bewegung in die gezeichneten Elemente zu bekommen, benötigen wir Threads
- Threads sind Unterprogramme in unserem Hauptprogramm
- D.h. das Unterprogramm kümmert sich um die Animation. Das Hauptprogramm um alles übrige (Tasteneingaben, Steuerung, Punktstand)
- Um mit Threads zu arbeiten benötigen wir zwei Dinge
  - Das Interface *Runnable*
  - Die *Run*-Methode

- Das Interface *Runnable* ermöglicht uns die *Run-Methode* zu implementieren

```
public class Spielfeld extends JPanel implements KeyListener, Runnable {...}
```

- Die Run-Methode wird als Thread (Unterprogramm) ausgeführt

```
@Override
```

```
public void run() {  
    //Animation  
    int offset = 1;  
    while (true) {  
        ballXpos = ballXpos + offset;  
        ballYpos = ballYpos + offset;  
        repaint();  
    }  
}
```

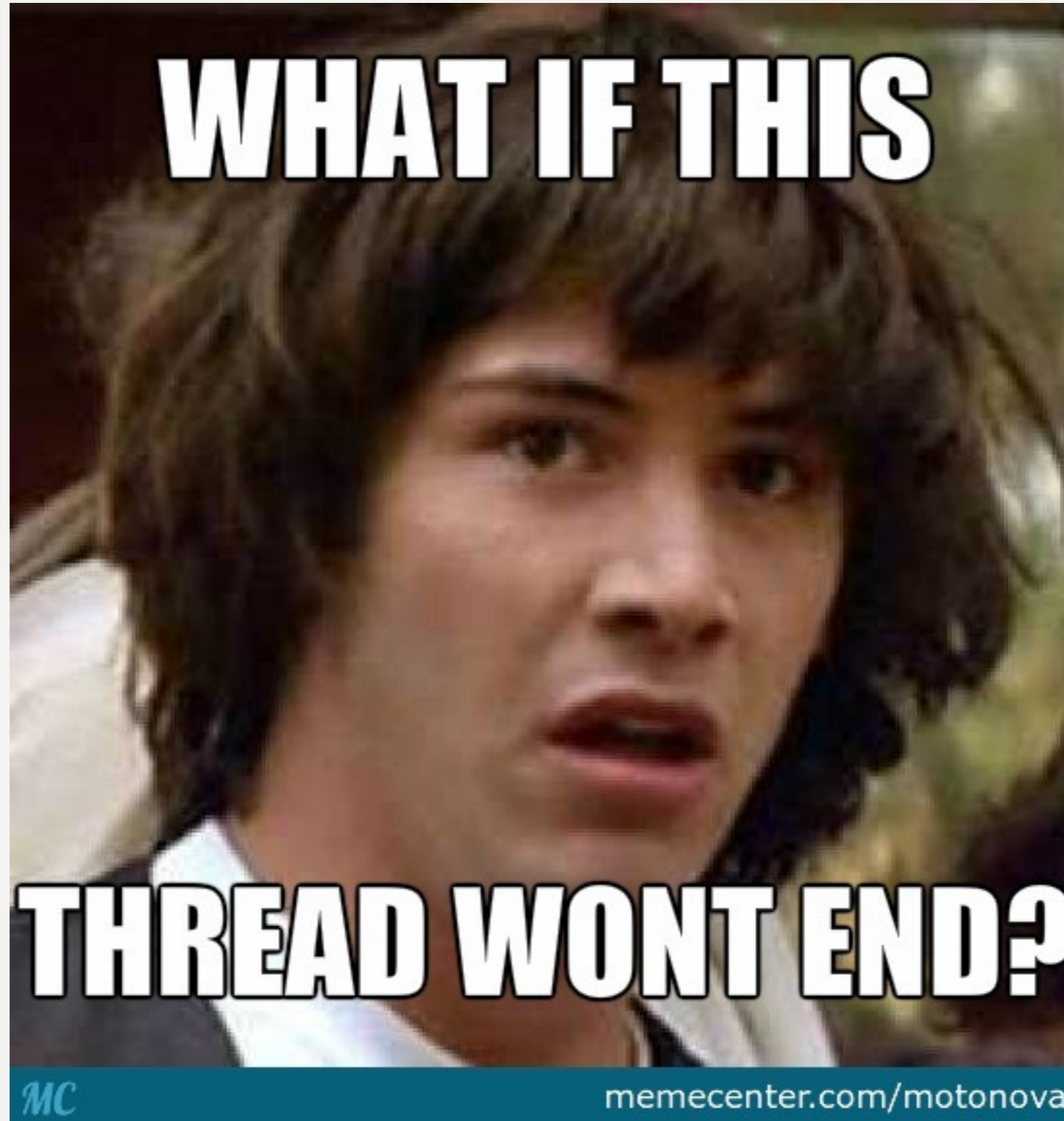




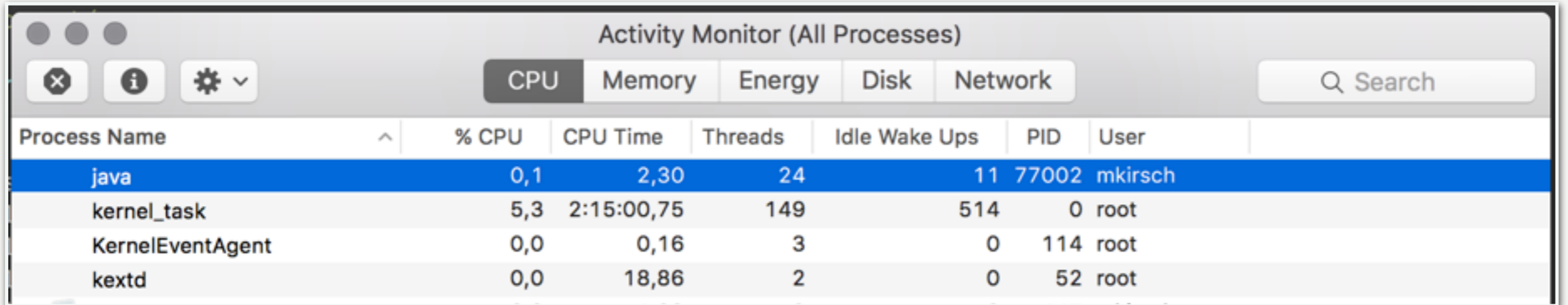
- Threads können auch „schlafen gelegt werden“, wobei die „Schlafzeit“ in Millisekunden angegeben wird.

```
@Override
public void run() {
    //Animation
    int offset = 1;
    while (true) {
        ballXpos = ballXpos + offset;
        ballYpos = ballYpos + offset;
        repaint();
        Thread.currentThread().sleep(5);
    }
}
```





- Threads sind Unterprogramme des Hauptprogramms. In unserem Fall ist der Animations-Thread Teil von Java Pong.
- Wird das Spiel Java Pong beendet, so wird auch der Thread selbst beendet



The screenshot shows the Activity Monitor window with the CPU tab selected. The 'java' process is highlighted in blue. The table below summarizes the data visible in the screenshot.

Process Name	% CPU	CPU Time	Threads	Idle Wake Ups	PID	User
java	0,1	2,30	24	11	77002	mkirsch
kernel_task	5,3	2:15:00,75	149	514	0	root
KernelEventAgent	0,0	0,16	3	0	114	root
kextd	0,0	18,86	2	0	52	root

- “Java” ist der Prozess von unserem Spiel, der wiederum 24 weitere Threads besitzt. Einer davon ist unser Animations-Thread.



- Mittels Try/Catch-Blöcken, können problematische Codefragmente „abgesichert“ werden

```
@Override
public void run() {
    int offset = 1;

    while (true) {

        try { //Versuche das Folgende...
            ballXpos = ballXpos + offset;
            ballYpos = ballYpos + offset;
            repaint();
            Thread.currentThread().sleep(5);
        } catch (InterruptedException e) { //Wenn es schief geht, hier weiter...
            e.printStackTrace();
        }
    }
}
```



# Anwendung

- Hands on...



# Verknüpfung

- Bei vielen Internetseiten findet ihr die behandelten Themen weiter und ausführlicher erklärt
- Eine Auswahl:
  - Brotcrunsher  
(<https://www.youtube.com/playlist?list=PL71C6DFDDEF73835C2>)
  - Java-Tutorials  
(<http://www.java-tutorial.org/swing.html>)
  - Head First  
<http://shop.oreilly.com/product/9780596009205.do>

# Ausblick



- In den Übungen werden die Inhalte der heutigen Vorlesung vertieft
  - Spielelemente (Kreis, Rechtecke) zeichnen
  - Ball von Programm automatisch bewegen lassen
  - Ball auf Spielfeld halten (Grenzen)



# Fragen?



Vielen Dank für Eure  
Aufmerksamkeit