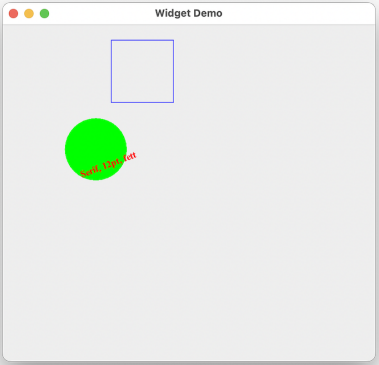
Einführung in Graphics und Java 2D

Carsten Gips (FH Bielefeld)

GUIs mit Java



Demo: java2d.simplegame.J2DTeaser

Einführung in die Java 2D API

Swing-Komponenten erben von `javax.swing.JComponent`:

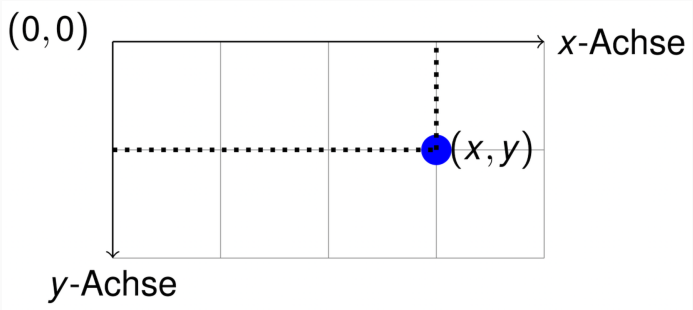
```
public void paintComponent(Graphics g)
```

- Wird durch Events aufgerufen
- Oder "von Hand" mit `void repaint()`

Objekt vom Typ `Graphics` stellt graphischen Kontext dar

=> **Methode überschreiben und auf der GUI malen**

Java2D Koordinatensystem



- Koordinatensystem lokal zum Graphics-Objekt
- Einheiten in Pixel(!)

Einfache Objekte zeichnen

Methoden von `java.awt.Graphics` (Auswahl):

```
public void drawLine(int x1, int y1, int x2, int y2)
public void drawRect(int x, int y, int width, int height)
public void fillRect(int x, int y, int width, int height)
public void drawOval(int x, int y, int width, int height)
public void fillOval(int x, int y, int width, int height)
```

Vorher Strichfarbe setzen: `Graphics.setColor(Color color)`:

- Farb-Konstanten in `java.awt.Color`: `RED`, `GREEN`, `WHITE`, ...
- Ansonsten über Konstruktor, beispielsweise als RGB:

```
public Color(int r, int g, int b)  // Rot/Grün/Blau, Werte zw. 0 und 255
```

Demo: java2d.SimpleDrawings

Fonts und Strings

Fonts über Font-Klasse einstellen: `Graphics.setFont(Font font);`

```
public Font(String name, int style, int size)
```

`Graphics` kann Strings "zeichnen":

```
public void drawString(String str, int x, int y);
```

Vorher Font und Farbe setzen!

Demo: java2d.SimpleFonts

Einfache Polygone definieren

Polygone zeichnen: `Graphics.drawPolygon(Polygon p)`:

```
public Polygon()
public Polygon(int[] xPoints, int[] yPoints, int points)
public void addPoint(int x, int y)
```

Polygone mit Farbe füllen: `Graphics.fillPolygon(Polygon p)`

Vorher Farbe setzen!

Demo: java2d.SimplePoly

Ausblick I: Umgang mit Bildern

```
BufferedImage img = ImageIO.read(new File("DukeWave.gif"));

boolean Graphics.drawImage(Image img, int x, int y, ImageObserver observer);
```

Ausblick II: *Graphics2D* kann noch mehr ...

```
Graphics g;
Graphics2D g2 = (Graphics2D) g;
```

=> `Line2D`, `Rectangle2D`, ...

- Strichstärken, Strichmuster
- Clippings
- Transformationen: rotieren, ...
- Zeichnen in Bildern, Rendern von Ausschnitten
- ...

Spiele mit Bewegung

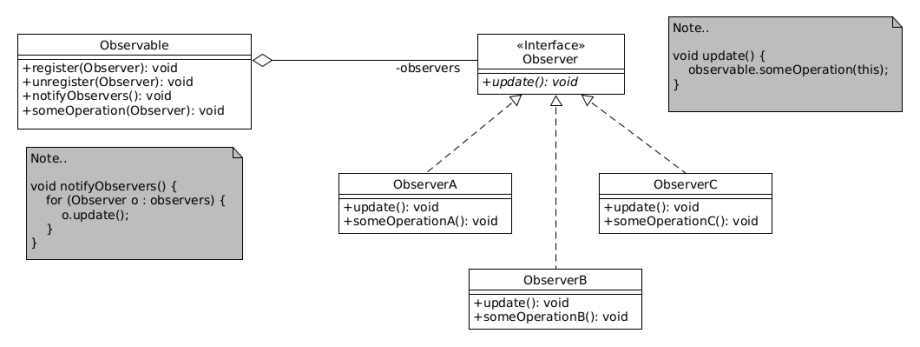
Beobachtung: `paintComponent()` schreibt `Graphics`-Objekt komplett neu!

Idee: Je Zeitschritt:

1. Position der Objekte neu berechnen
2. Weitere Berechnungen: Kollision, Interaktion, Angriff, ...
3. Objekte mit `paintComponent()` neu in GUI zeichnen

Hinweis: Zentrale Struktur vs. Observer-Pattern

Erinnerung: Observer Pattern



Hinweis auf Push- vs. Pull-Modell

Spielobjekte als Observer (Listener)

```
abstract class GameObject {
    abstract void move();
    abstract void paintTo(Graphics g); // entspricht Observer#update()
}

class GameRect extends GameObject {
    int x, y, deltaX;
    void move() { x += deltaX; }
    void paintTo(Graphics g) {
        g.drawRect(x, y, 80, 80);
    }
}
```

Weitere evtl. nützliche Methoden:

- Check auf Kollision
- Methode zum Umdrehen der Bewegungsrichtung

Oberfläche zusammenbauen

1. Spielfeld von `JPanel` ableiten: Observable
2. Observer registrieren: Liste mit Spiel-Objekten anlegen
3. `paintComponent()` vom Spielfeld überschreiben
   - für alle Observer (Spiel-Objekte) `paintTo()` aufrufen
4. Hauptschleife für Spiel:
   - Taktgeber (Zeit, Interaktion)
   - Je Schritt `move()` für alle Observer aufrufen
   - Weitere Berechnungen (Kollisionen, Interaktionen, ...)
   - `Spielfeld.repaint()` aufrufen => Neuzeichnen mit `paintComponent()`

Demo: java2d.simplegame.J2DTeaser

Wrap-Up

- Java2D: Swing-Komponenten zeichnen mit `paintComponent()` auf `Graphics`
- `Graphics`: Methoden zum Zeichnen von Linien, Rechtecken, Ovalen, Text ...
  - Koordinatensystem: Ursprung links oben!
  - Geom. Primitive und Text werden in ausgewählter Zeichenfarbe gerendert
    - Rechtecke, Ovale, Polygone auch als "gefüllte" Variante
  - Mehr Möglichkeiten: `Graphics2D`

- Spiel: Game-Loop
  - Bewege Objekte: Rechne neue Position aus
  - Interagiere: Angriffe, Sammeln, ...
  - Zeichne Objekte neu

LICENSE