




Übersicht:

- ▶ Was sind Vektoren?
 - ▶ Wo werden Vektoren eingesetzt?
 - ▶ Vektorgrafik
 - ▶ Wie bin ich vorgegangen
 - ▶ Was war leicht/schwierig?
 - ▶ Code/vorführung
 - ▶ Fragen?
- 

Was sind Vektoren?

- ▶ Mathematisches Objekt mit Richtung/Länge
- ▶ Notiert werden sie übereinander dargestellt:

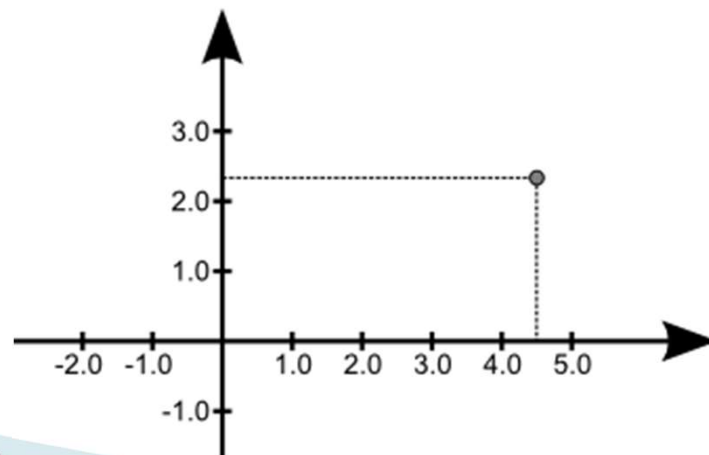
$$\vec{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

- ▶ Parallele Vektoren mit der gleichen Länge und gleichen Orientierung sind der selbe Vektor, egal wo im Koordinaten System eingetragen.



Wo werden Vektoren eingesetzt?

- ▶ Wenn Koordinaten von 2D- oder 3D-Formen in der Computergrafik mathematisch definiert werden, geschieht dies idealerweise durch reelle Zahlen.
- ▶ Ein Punkt in einem 2D-Koordinatensystem kann mit 2 reellen Zahlen definiert werden:



Wo werden Vektoren eingesetzt?

- ▶ Vektorräume(Linearer Raum) Algebra. Struktur
- ▶ Durchbruch zum Verständnis der Welt im 20. Jahrhundert.
 - Riemann'schen Geometrie (Relativitätstheorie)
 - Hilbertraum–Tensorprodukt (Quantentheorie)
- ▶ Mathematik (Geometrie, Funktionentheorie, die Theorie von Operatoren, Interpolation von Funktionen: Newton, Leibniz und : alles Vektorräume



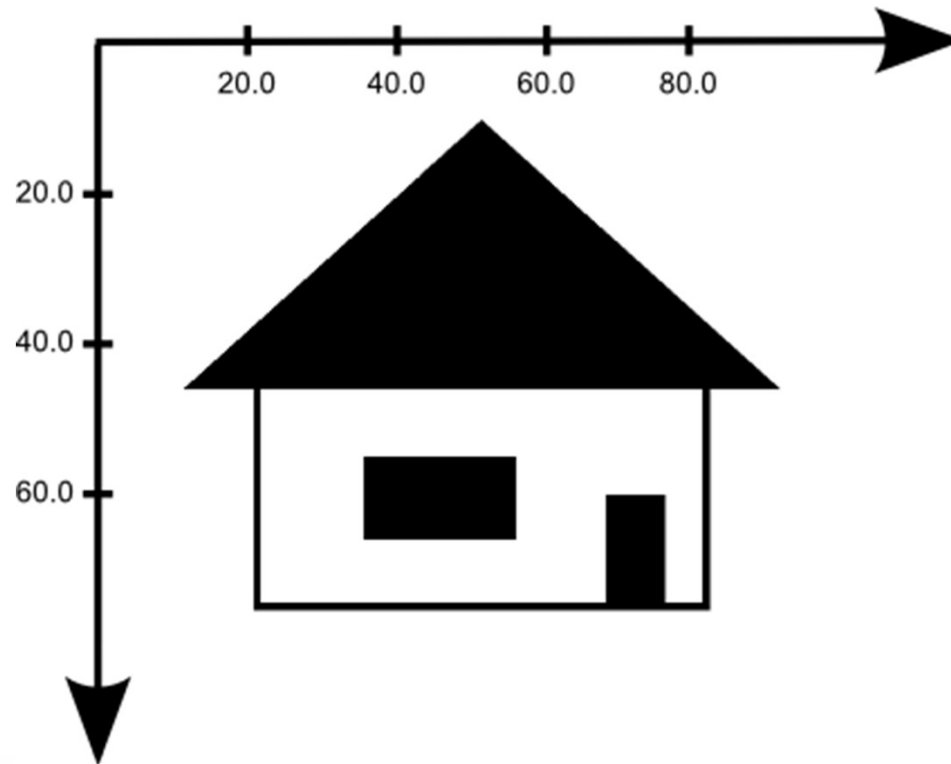
Vektorgrafik

- ▶ Eine Repräsentation von 2D- oder 3D-Formen mit reellen Koordinaten wird als Vektorgrafik bezeichnet.
- ▶ Der Begriff ergibt sich, weil die Koordinaten als Vektor geschrieben werden kann:

$$\mathbf{p} = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 4.5 \\ 2.33 \end{pmatrix}$$

Vektorgrafik

- ▶ In der Vektorgrafik werden die Objekte durch Kombination von Basisobjekten (wie z.B. Linie, Rechteck, Kreis, Dreieck, Polygon) beschrieben



Vektorgrafik

```
public void paint(Graphics g) {  
    Graphics2D g2d = (Graphics2D) g; // cast to Graphics2D  
    g2d.scale(10.0, 10.0); // try different scale factors  
    g2d.draw(new Rectangle2D.Double(20.75, 45.0, 60.0, 30.00) );  
    g2d.fill(new Rectangle2D.Double(67.33, 60.0, 8.0, 15.00) );  
    g2d.fill(new Rectangle2D.Double(35.00, 55.0, 20.3, 11.00) );  
    GeneralPath triangle = new GeneralPath();  
    triangle.moveTo(10.75, 46.0); // first point  
    triangle.lineTo(90.75, 46.0); // move left  
    triangle.lineTo(50.75, 10.0); // move to tip  
    triangle.closePath();  
    g2d.fill(triangle);  
}  
}
```


Wie bin ich vorgegangen

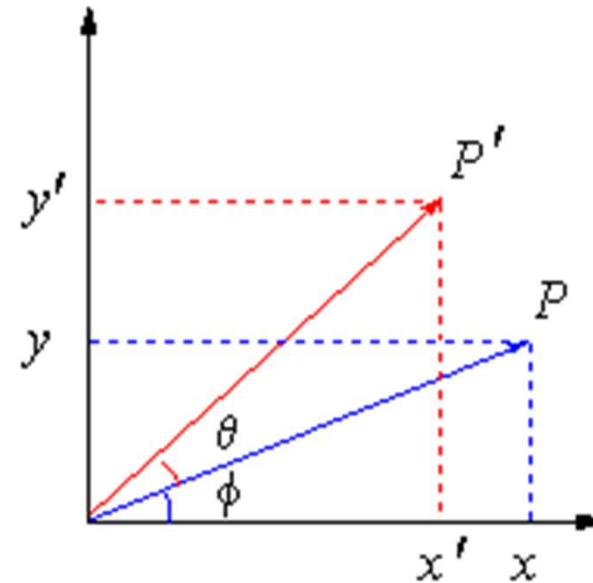
- ▶ Über Vektoren/Vektorgrafik Informiert
- ▶ Formeln Herausgeschrieben
- ▶ Test-Programm Geschrieben
- ▶ Verbesserung des Codes
- ▶ Endprodukt Programm Geschrieben



Formel

- Der zu $P(x,y)$ gehörender Vektor(blau) wird mit dem Winkel θ zu einer neuen Position $P'(\text{rot})$ gedreht:

$$\begin{aligned}x' &= x \cos \theta - y \sin \theta \\y' &= x \sin \theta + y \cos \theta\end{aligned}$$



Was war leicht/schwierig?

Leicht	Schwierig
Formel suchen	Fehler im Code Finden
Formel Funktion Programmieren	Flyer
Zeitplan einhalten	

Code

```
/*includes*/
#include <stm32f10x.h>
#include "TouchP0P1.h"
#include "math.h"
/*Variablen*/
double Figur[125] = {
//   x   y   z
-20,-80, 10, // 1. Vektor
 20,-80, 10, // 2. Vektor
 20,-20, 10, // 3. Vektor
 60,-20, 10, // 4. Vektor
 60, 20, 10, // 5. Vektor
 20, 80, 10, // 6. Vektor
-20, 80, 10, // 7. Vektor
-60, 20, 10, // 8. Vektor
-60,-20, 10, // 9. Vektor
-20,-20, 10, // 10. Vektor
```

```
double Vektor_1_x, Vektor_1_y,
Vektor_2_x, Vektor_2_y,
Vektor_3_x, Vektor_3_y,
Vektor_4_x, Vektor_4_y,
Vektor_5_x, Vektor_5_y,
Vektor_6_x, Vektor_6_y,
Vektor_7_x, Vektor_7_y,
Vektor_8_x, Vektor_8_y,
Vektor_9_x, Vektor_9_y,
Vektor_10_x, Vektor_10_y,
```

```
void pixel(int farbe)
{
  Linie(Vektor_1_x,Vektor_1_y,Vektor_2_x,Vektor_2_y,farbe);
  Linie(Vektor_2_x,Vektor_2_y,Vektor_3_x,Vektor_3_y,farbe);
  Linie(Vektor_3_x,Vektor_3_y,Vektor_4_x,Vektor_4_y,farbe);
  Linie(Vektor_4_x,Vektor_4_y,Vektor_5_x,Vektor_5_y,farbe);
  Linie(Vektor_5_x,Vektor_5_y,Vektor_6_x,Vektor_6_y,farbe);
  Linie(Vektor_6_x,Vektor_6_y,Vektor_7_x,Vektor_7_y,farbe);
  Linie(Vektor_7_x,Vektor_7_y,Vektor_8_x,Vektor_8_y,farbe);
  Linie(Vektor_8_x,Vektor_8_y,Vektor_9_x,Vektor_9_y,farbe);
  Linie(Vektor_9_x,Vektor_9_y,Vektor_10_x,Vektor_10_y,farbe);
  Linie(Vektor_10_x,Vektor_10_y,Vektor_1_x,Vektor_1_y,farbe);
```


Code

```
int main(void)
{
    InitTouchScreen();
    clearScreen(WHITE);

    while(1)
    {
        Vektor(0.0125,0.0125,0.025,20);
        pixel(BLACK);
        delay();
        pixel(WHITE);
    }
}
```

```
void Vektor(double X,double Y, double Z, int Vektorenanzahl)
{
    int increment = 0;
    double Xt,Yt,Zt;
    double Zan = Z;
    double Yan = Y;
    double Xan = X;

    int Counter = 1;

    while(Counter <= (Vektorenanzahl + 1))
    {
        Yt = (Figur[increment+1] * cos(Xan)) - (Figur[increment+2] * sin(Xan));
        Zt = (Figur[increment+1] * sin(Xan)) + (Figur[increment+2] * cos(Xan));

        Figur[increment+1] = Yt;
        Figur[increment+2] = Zt;

        Xt = (Figur[increment] * cos(Yan)) - (Figur[increment+2] * sin(Yan));
        Zt = (Figur[increment] * sin(Yan)) + (Figur[increment+2] * cos(Yan));

        Figur[increment] = Xt;
        Figur[increment+2] = Zt;

        Xt = (Figur[increment] * cos(Zan)) - (Figur[increment+1] * sin(Zan));
        Yt = (Figur[increment] * sin(Zan)) + (Figur[increment+1] * cos(Zan));

        Figur[increment] = Xt; // plot x
        Figur[increment+1] = Yt; // plot y

        switch(Counter)
        {
            case(1):{Vektor_1_x = Xt;Vektor_1_y = Yt;break;}
            /**/
            case(40):{Vektor_40_x = Xt;Vektor_40_y = Yt;break;}
        }

        increment = increment + 3;
        Counter++;
    }
}
```

Fragen?

