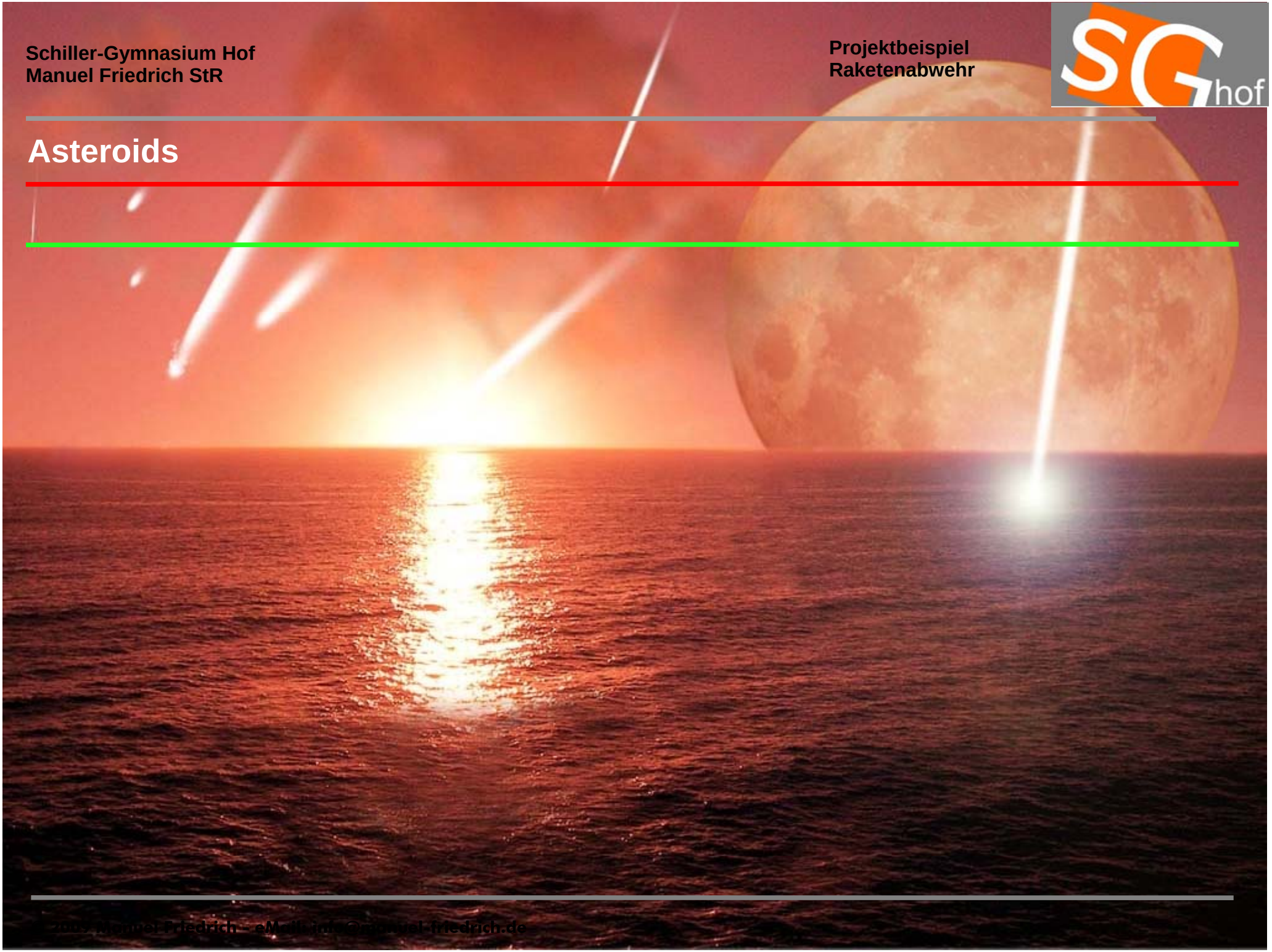
Asteroids



Wir planen eine Software

Lastenheft für ein Spiel

Asteroiden fallen auf die Erde und müssen, um die Erde zu retten, mithilfe von Raketen beschossen werden.

Im ersten Teil der Software-Entwicklung sollen nur die Raketen modelliert und umgesetzt werden.

Mit der Maus klickt man auf den Bildschirm und von der Erdoberfläche startet eine Rakete. Diese bewegt sich bis zum Ziel. Dort angekommen verschwindet die Rakete.

Mehrere Mausklicks hintereinander führt dazu, dass mehrere Raketen starten.

Wir planen eine Software

Pflichtenheft „Asteroids“

Gespielt wird in einem Fenster 800 x 800 Pixel groß. Mit der Maus kann jeder Punkt im Fenster angeklickt werden. Daraufhin startet eine Rakete auf einer linearen Bahn vom den Koordinaten Start(400,800) bis zum Ziel.

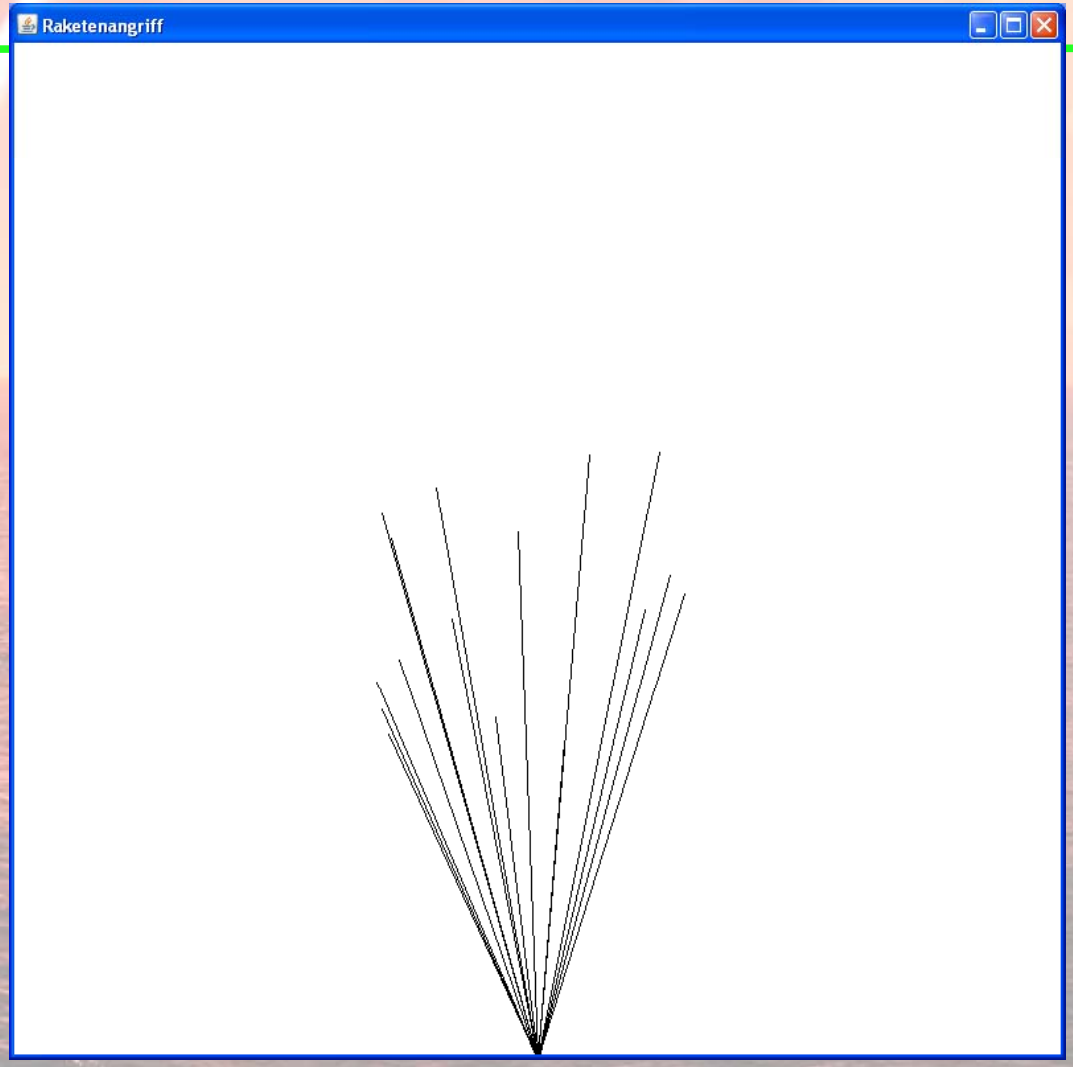
Die Geschwindigkeit der Rakete beträgt 10 Pixel pro Zeittakt. Die aktuelle Position der Rakete wird dargestellt als Linie zwischen Start und aktueller Position.

Das Ziel gilt als erreicht, wenn die Rakete in einem Zeittakt näher als 20 Pixel zum Ziel entfernt ist.

Dann wird die Rakete und ihre graphische Darstellung gelöscht.

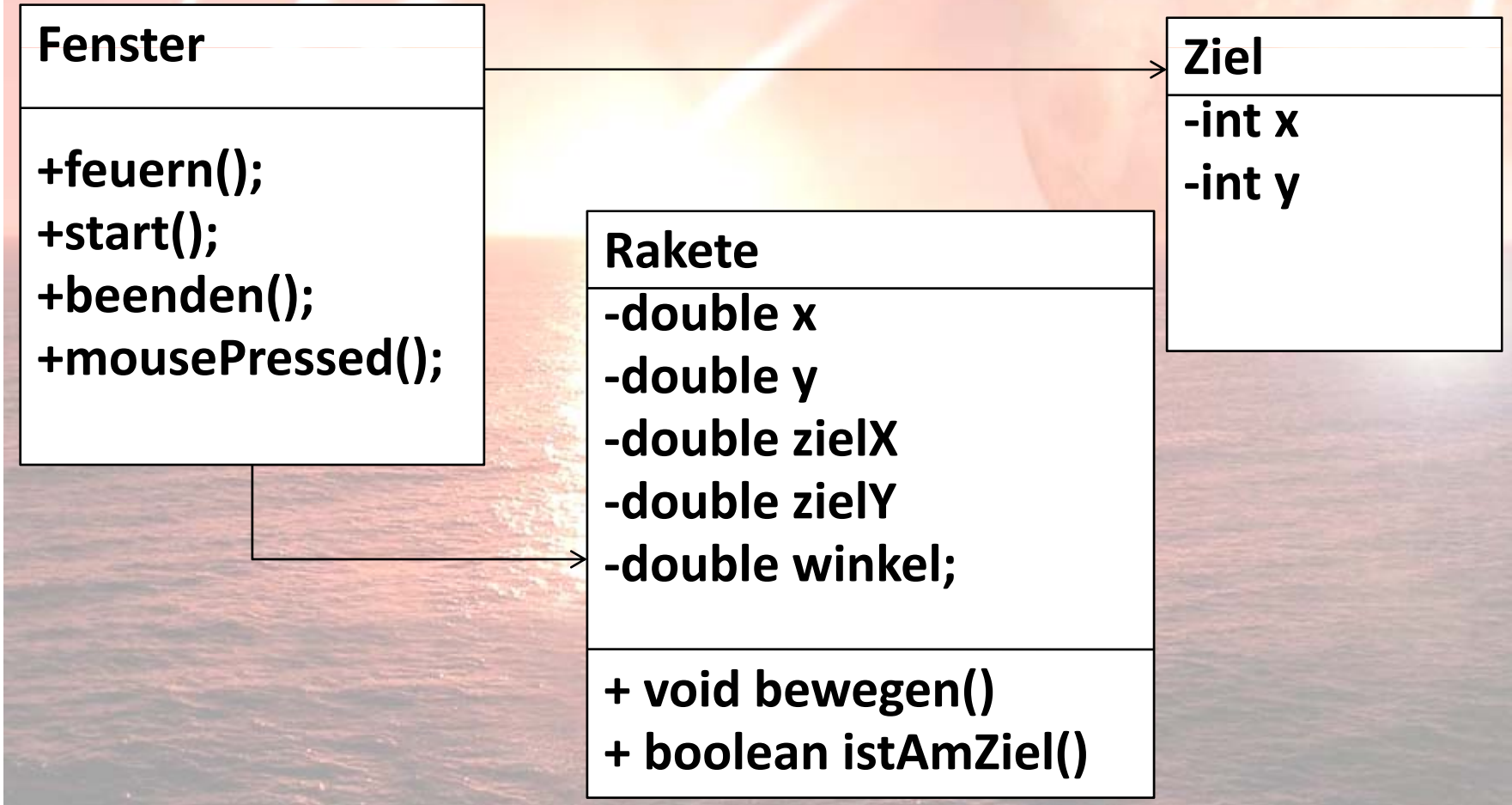
Wir planen eine Software

Skizze



Wir planen eine Software

Klassendiagramm



Wir planen eine Software

Klassendiagramm

Fenster

+feuern();
+start();
+beenden();
+mouseMoved(…);
+mousePressed(…);

Wenn mouseMoved(…), verändern sich die Zielkoordinaten der nächsten Rakete.

Wir planen eine Software

Klassendiagramm

Fenster

+feuern();
+start();
+beenden();
+mouseMoved(…);
+mousePressed(…);

Wenn mousePressed(), dann wird die Methode feuern() ausgeführt. Diese erzeugt eine neue Rakete mit dem Ziel der aktuellen Zielposition und fügt diese Rakete der ArrayList hinzu!

To do: Irgendwo muss gesteuert werden, dass die Raketen aus der ArrayList gelöscht werden, wenn sie ihr Ziel erreicht haben!!!

Wir planen eine Software

Klassendiagramm

Fenster

+feuern();
+start();
+beenden();
+mouseMoved(…);
+mousePressed(…);

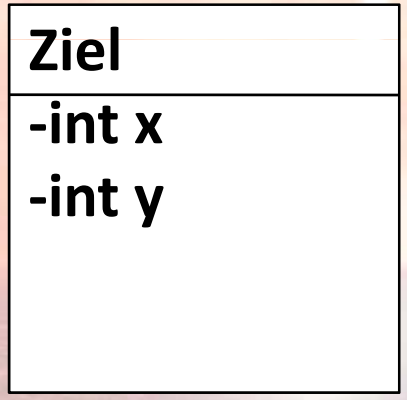
Die Methode start() führt eine Endlosschleife aus while(true).

Im Zeittakt werden alle Raketen bewegt und dann das Fenster neu gezeichnet. → paint-Methode

Wir planen eine Software

Klassendiagramm

Nur die beiden Attribute x und y und die dazugehörenden verändernden und sondierenden Methoden.



Wir planen eine Software

Klassendiagramm

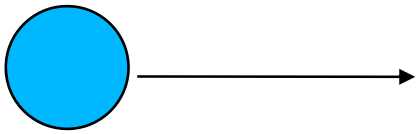
Rakete
-double x -double y -double zielX -double zielY -double winkel;
+ void bewegen() + boolean istAmZiel()

istAmZiel ist relativ einfach:

```
If ((Math.abs(x-zielX)<20)
&& (Math.abs(y-zielY)<20)
return true; else return
false;
```

Exkurs: Animation: Wie der Ball fliegt

Ausgangslage

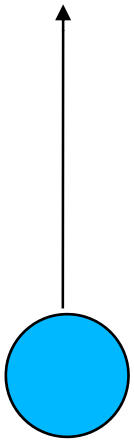


Hat der Ball 0 Grad Richtung so fliegt er in jedem Zeittakt um 10 Pixel nach rechts. Die Y-Koordinate bleibt gleich.

x=x+10;

Animation: Wie der Ball fliegt

Ausgangslage

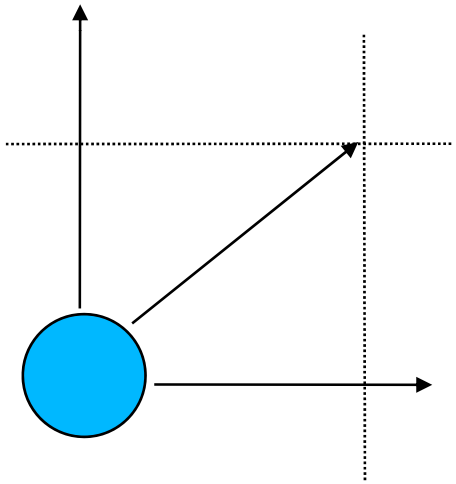


Hat der Ball 90 Grad Richtung so fliegt er in jedem Zeittakt um 10 Pixel nach oben. Die X-Koordinate bleibt gleich.

y=y-10;

Animation: Wie der Ball fliegt

Ausgangslage

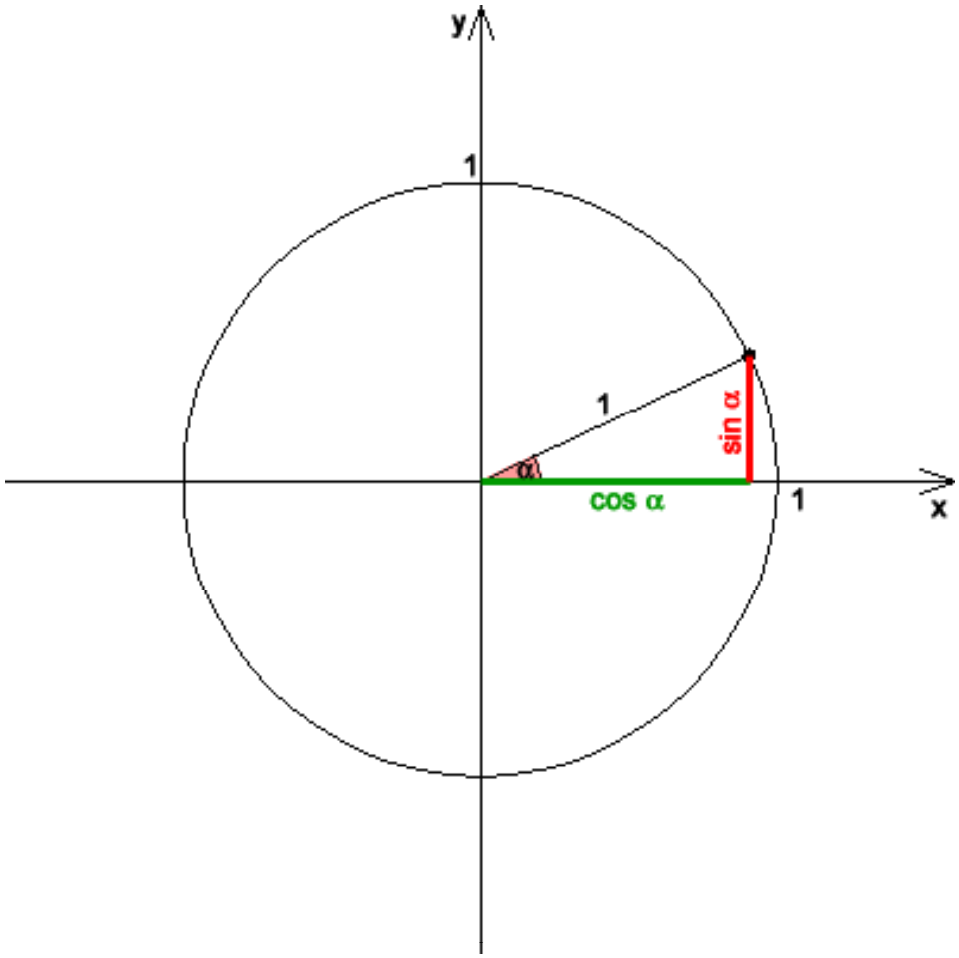


Hat der Ball 45 Grad Richtung so fliegt er weniger als 10 nach rechts und weniger als 10 nach oben...

„Wenn ich nicht mehr weiter weiß, dann mal ich mir 'nen Einheitskreis“

Animation: Wie der Ball fliegt

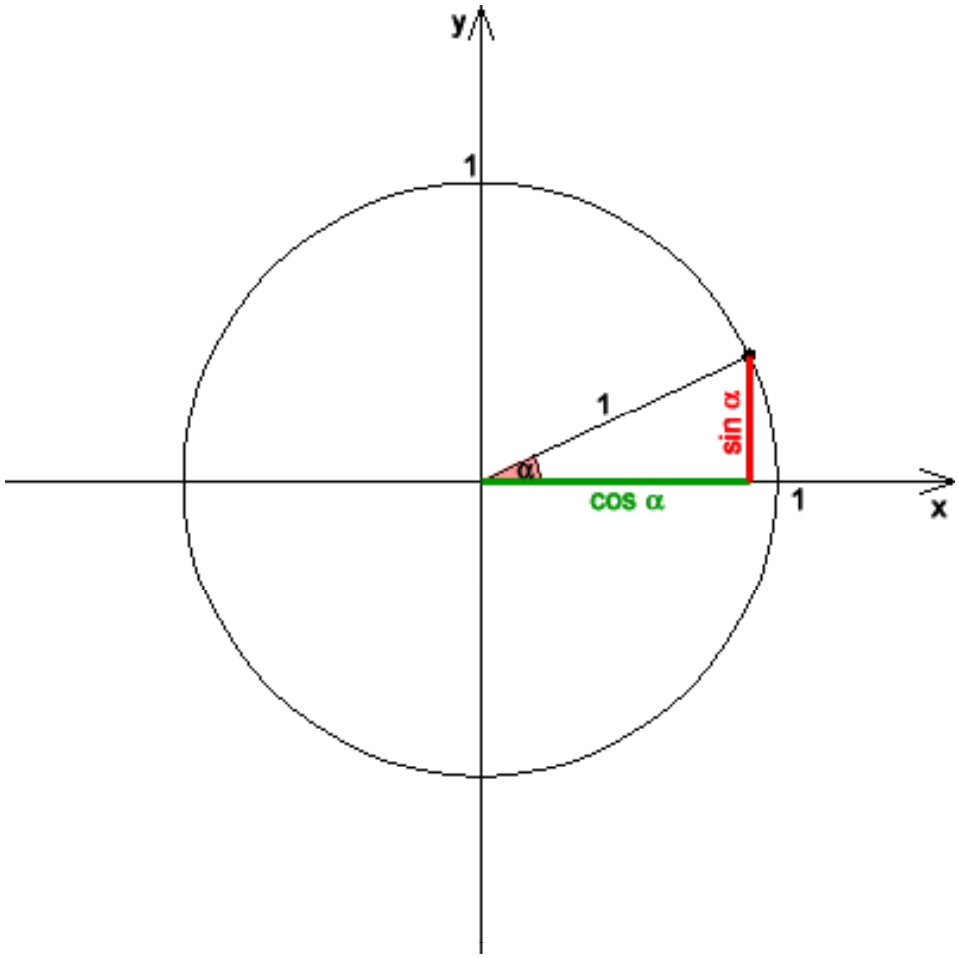
Sinus und Cosinus




```
x=x+cos(a)*v;
y=y-sin(a)*v;
```

Animation: Wie der Ball fliegt

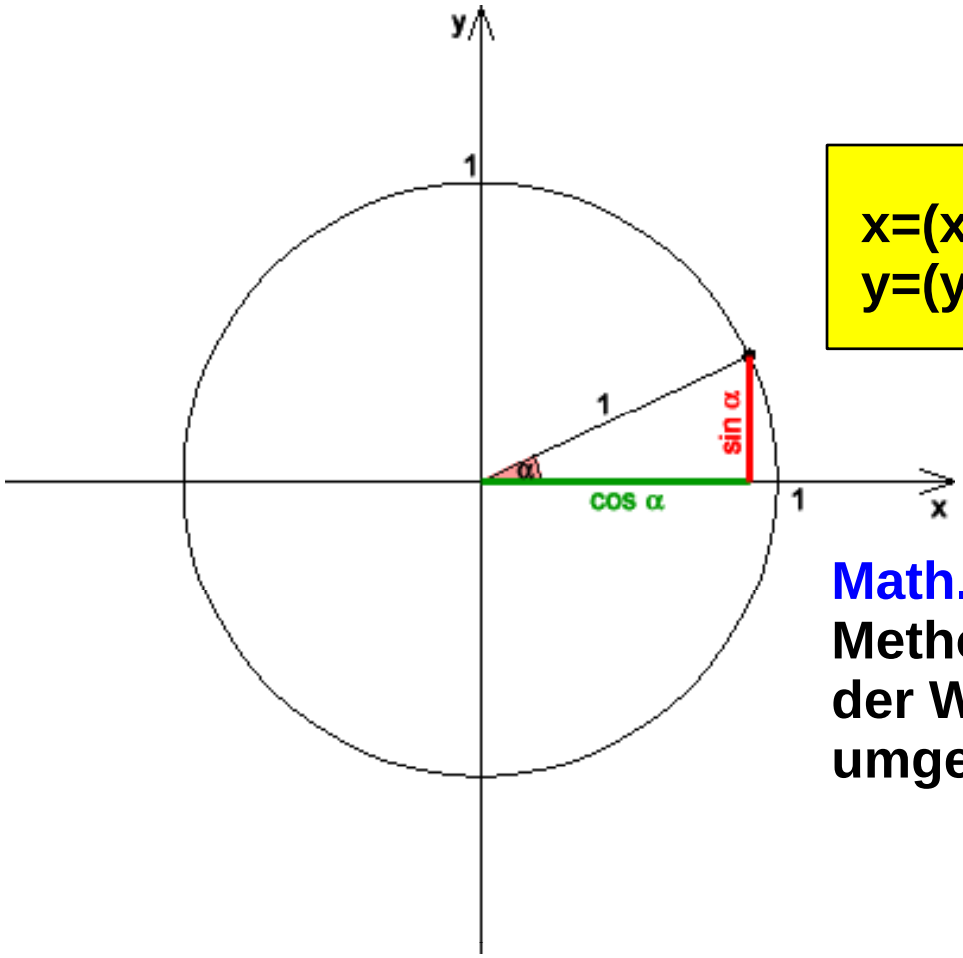
In einer Wiederholungsschleife




```
while (ende==false){
  for (int i=0; i<raketen.size(); i=i+1)
     raketen.get(i).bewegen();
  }
  repaint();
  wait(50);
}
```

Animation: Wie der Ball fliegt

Sinus und Cosinus in Java




```
x=(x+(Math.cos(Math.toRadians(a))*v));
y=(y-(Math.sin(Math.toRadians(a))*v));
```

Math.cos(a) und Math.sin(a) sind die Methoden zur Berechnung. Allerdings muss der Winkel noch ins Bogenmaß umgerechnet werden mit Math.toRadians(a);

Asteroids

Grau ist alle Theorie!!!

Los geht‘s...