Tutorial Processing **P3D**



im WPF Simulation für Spiele

Prof. Dr. Wolfgang Konen TH Köln



Wolfgang Konen – Spiele + Simulation – WS2024

```
void setup() {
    size(200,200,P3D);
}
```

Wann P3D benutzen?

- ... wenn man in 3D zeichnen will
- \blacksquare ... wenn man schneller sein will (OpenGL \rightarrow Grafikprozessor)
- wenn man in 2D zeichnet, aber bestimmte 3D-Effekte braucht (Shading, Light, ...)

Zeigen: <u>ex_rotate.pde</u>



P3D Koordinatensystem

Technology Arts Sciences TH Köln

Koordinatensystem:

ist linkshändig (!)





```
void setup() {
    size(300,300,P3D);
}
```

```
void draw() {
   background(20);
   stroke(150);
   fill(50,0,200);
   translate(width/4,height/4,0);
   rect(0,0,200,100);
}
```

- translate verschiebt Koordinatensystem → wirkt also auf alle nachfolgenden Befehle (!)
- rect zeichnet Rechteck bezogen auf den neuen Ursprung (0,0,0) in die XY-Ebene mit UL-Ecke in (0,0) und (w,h) = (200,100)



```
void setup() {
  size(300,300,P3D);
}
                                     rotateX rotiert um die
                                       aktuelle (!) X-Achse
void draw() {
                                     rotateY rotiert um die
  background(20);
                                       aktuelle (!) Y-Achse
  stroke(150);
  fill(50,0,200);
  translate(width/4, height/4, 0);
                                       Andern Sie den Winkel für
                                     rotateX(0);
                                        rotateX auf "45°" und für
  rotateY(PI/4);
                                        rotateY auf "0°"! Wohin
  rect(0,0,200,100);
                                        bewegt sich die
}
                                       Rechteckunterkante?
```



P3D: "Flying Rects ..."

```
int z;
void setup() {
   size(300,300,P3D);
}
```

}

```
void draw() {
   background(20);
   stroke(150);
   fill(50,0,200);
   translate(width/4,height/4,z);
   rotateX(0);
   rotateY(0);
   rect(0,0,200,100);
   z++;
```

Wieso verschwindet das Rechteck schließlich?

Ändern Sie den Winkel für rotateY auf "45°"! Können Sie die leichte Andersartigkeit des Verschwindens des Rechtecks erklären?

Ändern Sie z++ auf z-um! Schrumpft das Rechteck schließlich auf einen Punkt zusammen?



Interaktive Slider, Buttons, ...

```
import controlP5.*;
ControlP5 cp5;
int sliderVal = 100;
```





Damit die Slider nicht durch translate() usw. beeinflusst werden:



- Zeichnen Sie im 300x300 P3D-Mode ein (130x50)-Rechteck mit UL in der Mitte, blau mit hellgrauem Rand.
- Fügen Sie zwei Slider hinzu, mit denen Sie das Rechteck um die Xund um die Y-Achse im Range [0,90°] drehen können.
- Verschieben Sie die UL-Corner an andere Stellen. Entspricht das Resultat einer 90°-Drehung Ihren Erwartungen?
- Was passiert bei translate NACH rotateX und rotateY?



Lösung: rotateX + rotateY via Slider

```
import controlP5.*;
ControlP5 cp5;
float angleX = 0, angleY=0;
void setup() {
  size(300,300,P3D);
  cp5 = new ControlP5(this);
  cp5.setAutoDraw(false);
  textSize(14);
  cp5.addSlider("angleX", 0,
PI/2, angleX, 50, 50, 200, 10);
  cp5.addSlider("angleY", 0,
PI/2, angleY, 50, 75, 200, 10);
```

```
void draw() {
  background(20);
  cp5.draw();
  fill(220);
  text("angleX="+
     nf(angleX,1,3),100,250,0);
  text("angleY="+
     nf(angleY,1,3),100,275,0);
  stroke(150);
  fill(50,0,200);
  translate(width/2, height/2, 0);
  rotateX(angleX);
  rotateY(angleY);
  rect(0, 0, 130, 50);
```



10

```
void setup() {
  size(500,500,P3D);
void draw() {
  background(20);lights();
  pushMatrix();
  translate(130, height/2, 0);
  rotateY(1.25); rotateX(-0.4);
  fill(200); noStroke();
 box(100);
  popMatrix();
```

```
pushMatrix();
translate(500, height*0.35, -200);
noFill(); stroke(255);
sphere(280);
popMatrix();
```

```
box(100) zeichnet einen
```

- Würfel der Länge 100 mit Zentrum bei (0,0,0)
- **sphere(280)** zeichnet eine Kugel mit Radius 280 und Zentrum bei (0,0,0)
- pushMatrix() legt eine Kopie des Koordinatensystems auf den Stack



P3D: Custom 3D Shapes

beginShape(); vertex(x,y,z); vertex(0,1,2); endShape(); Beispiele f
ür <u>beginShape</u> in Processing Reference zeigen





Übung: Custom 3D Shapes

- Zeichnen Sie im 500x500 P3D-Mode eine "Buchstütze" bestehend aus 2 Dreiecken, die an einer Kante rechtwinklig aneinanderstoßen
- Fügen Sie zwei Slider hinzu, mit denen Sie die Buchstütze um die X- und um die Y-Achse im Range [0,90°] drehen können.
- Geben Sie jedem Vertex eine unterschiedliche fill-Color. Was passiert?



Lösung: Custom 3D Shapes

```
import controlP5.*;
ControlP5 cp5;
float angleX = 0, angleY=0;
void setup() {
  size(500,500,P3D);
  cp5 = new ControlP5(this);
  cp5.setAutoDraw(false);
  textSize(14);
  cp5.addSlider("angleX",0,
PI/2, angleX, 50, 50, 200, 10);
  cp5.addSlider("angleY",0,
PI/2, angleY, 50, 75, 200, 10);
                               }
```

```
void draw() {
 background(20);
  cp5.draw();
  translate(3*width/8,
            2.5*height/8, 0);
  rotateX(angleX);
  rotateY(angleY);
  scale(1.5);
  noStroke();
 beginShape(TRIANGLES);
  fill(0,0, 50);vertex(
                         0,100, 0);
                         0,100,100);
  fill(0,0,100);vertex(
  fill(0,0,230);vertex(100,100,100);
                         0, 0, 0);
  fill(0, 50,0);vertex(
  fill(0,230,0);vertex(
                         0,100,100);
  fill(0,150,0);vertex(
                         0, 100, 0);
  endShape();
```



P3D: Texturen





Übung: Texturen

- Zeichnen Sie ein Parallelogramm mit oberer und unterer Kantenlänge 200, obere um 75 nach links verschoben, und Höhe 250.
- Texturieren Sie es mit dem Bild "berlin-1.jpg", aber so, dass der 'Alex' weiterhin aufrecht steht (!)
- Was bewirkt der zusätzliche Befehl rotateY(map(mouseX, 0, width, -PI, PI));





Lösung: Texturen



P3D: Kamera

Technology Arts Sciences TH Köln

Die Kamera "sieht" alles innerhalb ihres Frustums (Pyramidenstumpf).



etwas genauer:

lookAt bezeichnet einen Punkt, der fixiert werden soll. 

Wolfgang Konen – Spiele + Simulation – WS2024

Technology

P3D: Kamera

```
void setup() {
  size(640, 360, P3D);
}
void draw() {
  background(100);
  camera(mouseX, height/2, (height/2)/tan(PI/6),
         width/2, height/2, 0, 0, 1, 0);
  translate(width/2, height/2, -100);
  stroke (255);
  noFill();
  box(200);
}
                        Weitere "Kamera"-Befehle:
                        perspective()
                        ortho()
```

Bei Bedarf anschauen in "Find in Reference"



Übung: Kamera

- Verändern Sie die Kamerasteuerung so
 - dass mouseY eine veränderte Blickhöhe bewirkt (mit Fixation)
 - dass mouseX eine horizontale Vorbeifahrt ("Panning") bewirkt, ohne Fixation.



Lösung: Kamera

```
void setup() {
   size(640, 360, P3D);
}
void draw() {
   background(100);
   camera(mouseX, mouseY, (height/2)/tan(PI/6),
        mouseX, height/2, 0, 0, 1, 0);
   translate(width/2, height/2, -100);
   stroke(255);
   noFill();
   box(200);
```

P3D: Licht

- lights() default light
- ambientLight(R,G,B)
- directionalLight(R,G,B,nx,ny,nz) certain direction, but location ininfinity (like the sun)
- spotLight(R,G,B, x, y, z, nx, ny, nz, angle, concentration) certain light source emitting a directional cone with angle 'angle'
- pointLight(R,G,B,x,y,z) a spotLight with a 180 degree cone
- IghtSpecular(R,G,B) interacts with specular(R,G,B) and directionalLight(), see below

with

- R,G,B = RGB- or HSV-values, depending on <u>colorMode()</u>,
- x,y,z = position of light

nx,ny,nz = direction of light (a vector with same direction as light rays)

Lighting commands need to be in the draw routine (!)

Lighting commands affect only objects created after them in the code.

Wolfgang Konen – Spiele + Simulation – WS2024

```
void setup() {
  size(200, 200, P3D);
}
void draw() {
  background(0);
  translate(100, 100, 0);
  if (mousePressed) {
    lights();
  }
  noStroke();
  fill(255);
  sphere(50);
}
```



```
void setup() {
  size(200, 200, P3D);
}
void draw() {
  background(0);
  translate(100, 100, 0);
  if (mousePressed) {
    lightSpecular(255,200,0);
    directionalLight(204,204,204,
          (width/2-mouseX)/10,
          (height/2-mouseY)/10,-2);
  }
  specular(0,200,200);
  noStroke();
  fill(255);
  sphere(50);
}
```

- lightSpecular bietet rotes und grünes Licht an
- specular sagt, dass die Kugel nur grünes und blaues Licht reflektiert

also grün:



directionalLight in Referenz

nachschlagen



P3D: Licht und Glanzlicht

Technology Arts Sciences TH Köln

```
void setup() {
  size(250, 250, P3D);
}
void draw() {
  background(100);
  noStroke();
  fill(0, 51, 102);
  lightSpecular(255, 255, 255);
  directionalLight (204,204,204,
                    0, +2, -2);
  translate(80, 125, 0);
  specular(255,255,255);
  shininess(5.0);
  sphere(60);
  translate(110, 0, 0);
  specular(204,102,0);
  shininess(1);
  sphere (50);
}
```

- shininess(5) erzeugt für linke Kugel ein Glanzlicht
- shininess(1) erzeugt für rechte Kugel matte Oberfläche (def.)





P3D bietet einen erleichterten Zugang zu (fast) allen OpenGLBefehlen

- Man beachte: Koordinatensystem linkshändig (initial sind positive z-Werte auf Betrachter zu, negative von ihm weg)
- Das geometrische Verständnis der Befehle translate(), rotateX() usw. in ihrer Wirkung auf das Koordinatensystem ist bedeutsam für das Schreiben korrekter P3D-Programme.
- Custom Shapes (beginShape() ... endShape()) können schnell mit Texturen versehen werden.
- Kameraeinstellungen sind relativ simpel über camera() machbar. Library peasycam ist eine interessante Alternative, belegt (blockiert) aber die Maustasten mit der Kamerasteuerung.

