

Dreiecke zeichnen mit OpenGL ES 1.1

Philipp Klaus Krause

June 18, 2019

- OpenGL ist eine Graphikchnittstelle, die effiziente Graphikausgabe ermöglicht.
- Verschiedene Varianten. Neuere (und Vulkan) sind flexibler, aber auch komplizierter (Shader, etc).
- OpenGL ES für kleinere Systeme, auch Mobilgeräte
- OpenGL ES 1.1 ist die aktuelle Version von OpenGL ES ohne Unterstützung für Shader.
- Wir verwenden OpenGL ES 1.1 in der Vorlesung (zumindest erstmal) für 2D-Graphik.

Quadrat aus zwei Dreiecken (aus dem Beispielprogramm)

```
const GLfloat square_corners[] = {  
0.0f, 0.0f,  
0.8f, 0.0f,  
0.0f, 0.8f,  
0.8f, 0.8f,  
};  
  
...  
glVertexPointer(2, GL_FLOAT, 0, square_corners);  
glDrawArrays(GL_TRIANGLE_STRIP, 0, 4);
```

Datentypen

OpenGL (ES) verwendet eigene Datentypen, per `typedef` in `gl.h`. Diese haben teils andere Mindestgrößen als die C-Datentypen entsprechenden Namens. Für uns wichtig sind:

- `GLubyte`
- `GLint`
- `GLuint`
- `GLenum`
- `GLsizei`
- `GLfloat`

glGetError

```
GLenum glGetError(void);
```

- Gibt den ersten seit dem letzten Aufruf von `glGetError()` bemerkten Fehler zurück.
- Falls kein Fehler bemerkt wurde, wird `GL_NO_ERROR` zurückgegeben.

glVertexPointer

```
void glVertexPointer(GLint size, GLenum type,  
    GLsizei stride, const void *pointer);
```

- Setzt den Zeiger auf die Koordinaten der Ecken.
- `size` gibt an, wie viele Einträge es pro Ecke gibt. Gültige Werte sind 2 (verwenden wir für 2D-Graphik), 3 und 4.
- `type` gibt den Datentyp an. `GL_FLOAT` für `GLfloat`.
- `stride` gibt an, wie viel Abstand zwischen aufeinanderfolgenden Einträgen im Feld ist (0 falls diese direkt aufeinanderfolgen).
- `pointer` zeigt auf die Koordinaten (die in einem Feld liegen).

glColor4f

```
void glColor4f(Glfloat red, Glfloat green, Glfloat blue,  
              Glfloat alpha);  
void glColor4ub(GLubyte red, GLubyte green, GLubyte blue,  
              GLubyte alpha);
```

- Setzt die aktuelle Farbe der Ecken (Bereich aller Komponenten: $[0, 1]$ bzw. $[0, 255]$).
- Damit kann man nur einfarbige Dreiecke zeichnen.
- red, green, blue wie üblich (additive Farbmischung).
- alpha: Opaquität (1 – Transparenz).

```
void glDrawArrays(GLenum mode, GLint first, GLsizei count);
```

- Zeichnet Dreiecke.
- **mode** gibt an, wie die Koordinaten als Ecken von Dreiecken zu verstehen sind. Bei `GL_TRIANGLE_STRIP` geben jeweils drei aufeinanderfolgende Koordinaten die Ecken eines Dreiecks an (die letzten beiden Ecken eines Dreiecks sind also die ersten beiden des nächsten).
- **first** gibt an, wo die erste zu zeichnende Ecke im Feld liegt.
- **count** gibt an, wie viele Ecken zum Zeichnen von Dreiecken verwendet werden sollen.

Quadrat aus zwei Dreiecken (aus dem Beispielprogramm)

```
const GLfloat square_corners[] = {  
0.0f, 0.0f,  
0.8f, 0.0f,  
0.0f, 0.8f,  
0.8f, 0.8f,  
};  
  
...  
glVertexPointer(2, GL_FLOAT, 0, square_corners);  
glDrawArrays(GL_TRIANGLE_STRIP, 0, 4);
```