

Farbmischung

- ▶ Farbmischung zwischen halbdurchlässigen Oberflächen und dem Hintergrund (Anwendung von Alpha-Werten)

glEnable, **glDisable** mit GL_BLEND

glBlendEquation mit GL_FUNC_ADD (Standardwert),
GL_FUNC_SUBSTRACT, GL_FUNC_REVERSE_SUBSTRACT,
GL_MIN, GL_MAX

glBlendFunc(int *sFactor*, int *dFactor*), Tabelle mit Argumenten:

<http://www.glprogramming.com/red/chapter06.html>



Alpha-Wert (Glas) = 0.7, (Teapot) = 1
(Milchglas, lässt 30% Licht durch)
`gl.glBlendFunc(GL.GL_SRC_ALPHA,
GL.GL_ONE_MINUS_SRC_ALPHA);`



Alpha-Wert (Alles) = 1.0
`gl.glBlendFunc(GL.GL_ONE,
GL.GL_ONE);`



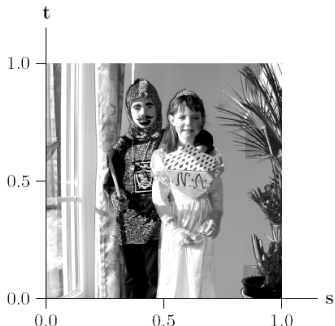
Alpha-Wert (Alles) = 1.0
`gl.glBlendEquation(GL.GL_FUNC_REVERSE_SUBSTRACT);
gl.glBlendFunc(GL.GL_ONE,
GL.GL_ONE);`

Texture-Mapping

- ▶ Eingesetzt, um die Oberfläche von Objekten mit einer Struktur zu überziehen → realist. Bilder
- ▶ Texturkoordinaten spezifizieren wie die Textur auf die Geometrie abgebildet wird

glTexCoord*

```
gl.glBegin(GL.GL_QUADS);  
gl.glTexCoord2f(0.0f, 0.0f);  
gl.glVertex2f(-5.0f, -5.0f);  
gl.glTexCoord2f(1.0f, 0.0f);  
gl.glVertex2f(5.0f, -5.0f);  
gl.glTexCoord2f(1.0f, 1.0f);  
gl.glVertex2f(5.0f, 5.0f);  
gl.glTexCoord2f(0.0f, 1.0f);  
gl.glVertex2f(-5.0f, 5.0f);  
gl.glEnd();
```



(Nischwitz *et al.*, 2011)

Erzeugen eines Textur-Objektes - Beispiel

```
...
Texture textureTree;

public void init(GLAutoDrawable drawable) {
    ...
    textureTree = TextureIO.newTexture(new File("images/poplar.png")); }
}

public void display(GLAutoDrawable drawable) {
    ...
    drawTree(gl);
}

void drawTree(GL gl) {
    textureTree.enable();
    textureTree.bind();
    ...
    gl.glBegin(GL.GL_QUADS);
    gl.glTexCoord2f(0.0f, 0.0f);
    gl.glVertex2f(-0.45f, 0.0f);
    gl.glTexCoord2f(1.0f, 0.0f);
    gl.glVertex2f(0.45f, 0.0f);
    gl.glTexCoord2f(1.0f, 1.0f);
    gl.glVertex2f(0.45f, 1.3f);
    gl.glTexCoord2f(0.0f, 1.0f);
    gl.glVertex2f(-0.45f, 1.3f);
    gl.glEnd();

    textureTree.disable();
}
}
```

- Für optimale Ergebnisse, Breite und Höhe als Zweierpotenz



Textur-Filter

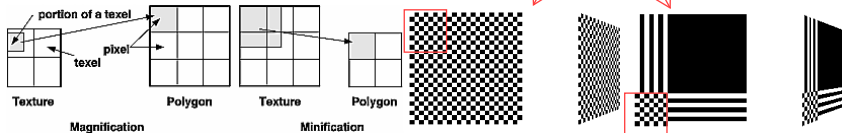
► Vergrößerung, Verkleinerung

```
gl.glTexParameteri(GL.GL_TEXTURE_2D, GL.GL_TEXTURE_MAG_FILTER, GL.GL_LINEAR);  
gl.glTexParameteri(GL.GL_TEXTURE_2D, GL.GL_TEXTURE_MIN_FILTER, GL.GL_LINEAR);
```

► Texture Wraps

```
gl.glTexParameteri(GL.GL_TEXTURE_2D, GL.GL_TEXTURE_WRAP_S, GL.GL_REPEAT);  
gl.glTexParameteri(GL.GL_TEXTURE_WRAP_T, GL.GL_REPEAT);
```

```
gl.glTexParameteri(GL.GL_TEXTURE_2D, GL.GL_TEXTURE_WRAP_S, GL.GL_CLAMP);  
gl.glTexParameteri(GL.GL_TEXTURE_WRAP_T, GL.GL_CLAMP);
```



www.glprogramming.com/red/chapter09.html

Nebel

► Atmosphärische Effekte → mehr Realismus

glFog*

name	param	Bedeutung
GL_FOG_MODE	GL_EXP	exponentiell abnehmende Sichtbarkeit, d.h. konstante Nebeldichte (11.2).
	GL_EXP2	exponentiell quadratisch abnehmende Sichtbarkeit, d.h. zunehmende Nebeldichte (11.4).
	GL_LINEAR	linear abnehmende Sichtbarkeit, d.h. hyperbolisch zunehmende Nebeldichte (11.5).
GL_FOG_COLOR	$\mathbf{g}_n = (R_n, G_n, B_n, A_n)^T$	Zeiger auf eine Nebelfarbe (nur in der Vektor-Form des Befehls möglich)
GL_FOG_DENSITY	a	Nebeldichte bei exp- und exp2-Funktion
GL_FOG_START	z_{start}	Entfernung, bei der ein linear ansteigender Nebel beginnt
GL_FOG_END	z_{end}	Entfernung, ab der ausschließlich die Farbe des linear ansteigenden Nebels zu sehen ist
GL_FOG_COORDINATE_SOURCE	GL_FRAGMENT_DEPTH	Entfernung (z -Wert) des Fragments vom Augenpunkt, wie sie im z -Buffer steht.
	GL_FOG_COORDINATE	z -Wert des Fragments, linear interpoliert aus den mit <code>glFogCoordfv()</code> an den Vertices festgelegten z -Werten.



exp



exp2



linear

(Nischwitz *et al.*, 2011)

Transparenz, Nebel - Beispiel

Anmerkungen:

Blend-Befehle stehen ausschließlich für Farbmischungs-zwecke;
fürs Verwerfen von transparenten Fragmenten (hier: mit dem Alpha-Wert = 0) ist der Alpha-Test da

```
// do not draw the transparent parts of the texture  
gl.glEnable(GL.GL_BLEND);  
gl.glBlendFunc(GL.GL_SRC_ALPHA, GL.GL_ONE_MINUS_SRC_ALPHA);
```

```
// perform alpha-test, draw only if alpha > 0  
gl.glEnable(GL.GL_ALPHA_TEST);  
gl.glAlphaFunc(GL.GL_GREATER, 0);
```

```
// draw tree  
...
```

```
gl.glDisable(GL.GL_ALPHA_TEST);  
gl.glDisable(GL.GL_BLEND);
```

```
float[] fogColor = {1.0f, 1.0f, 1.0f, 1.0f};
```

```
gl.glEnable(GL.GL_FOG);
```

```
gl.glFogi(GL.GL_FOG_MODE, GL.GL_EXP);  
gl.glFogf(GL.GL_FOG_DENSITY, 0.05f);  
gl.glFogfv(GL.GL_FOG_COLOR, fogColor, 0);
```

