SceneKit

Torsten Kammer

@zcochrane

Problem: 3D ist schwer

```
glVertexAttribPointer(0, 3, GL_FLOAT,
GL_FALSE, (GLsizei) mesh.stride, (GLvoid
*) mesh.offsetForPosition);
```

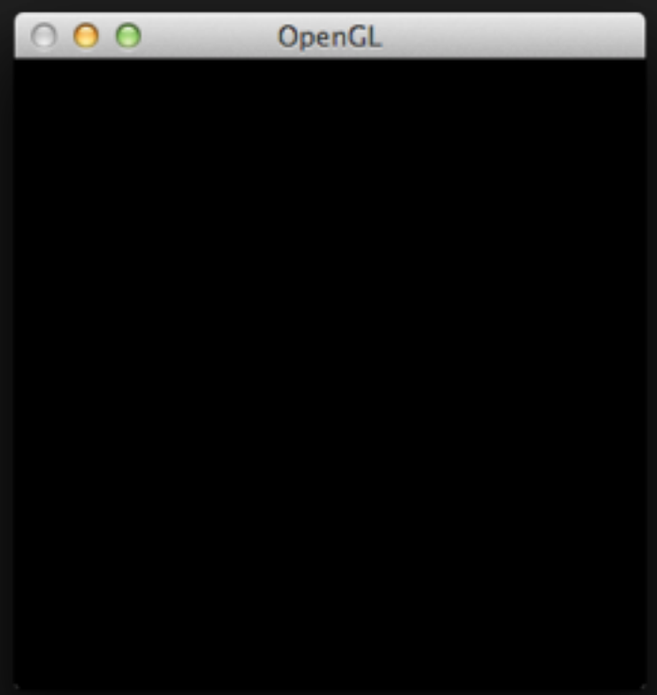
Problem: 3D ist schwer

```
glBindTexture(GL_TEXTURE_2D, textureNames[finishedTextures]);
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, width, height, 0,
GL_RGBA, GL_UNSIGNED_BYTE, NULL);
glFramebufferTexture2D(GL_FRAMEBUFFER, GL_COLOR_ATTACHMENT0,
GL_TEXTURE_2D, textureNames[finishedTextures], 0);
glBindRenderbuffer(GL_RENDERBUFFER, depthRenderbuffer);
glRenderbufferStorage(GL_RENDERBUFFER, GL_DEPTH_COMPONENT24,
width, height);
glFramebufferRenderbuffer(GL_FRAMEBUFFER, GL_DEPTH_ATTACHMENT,
GL_RENDERBUFFER, depthRenderbuffer);
```

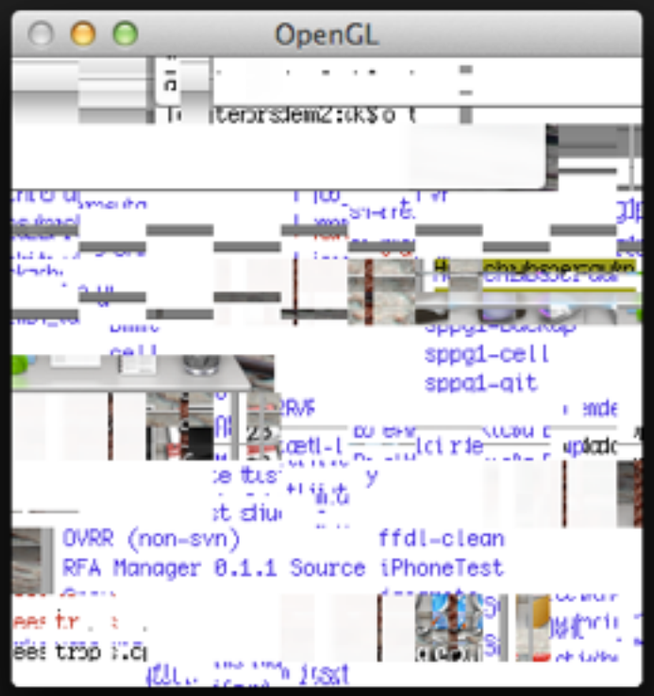
Problem: 3D ist schwer

```
/*
 * This is essentially identical to DiffuseLightmapBump3, but the specular color is not always white; instead it is read from its own texture.
 */
#version 150

in vec4 outColor;
in vec2 outTexCoord;
in vec3 positionWorld;
in mat3 tangentToWorld;

out vec4 screenColor;

uniform sampler2D diffuseTexture;
uniform sampler2D lightmapTexture;
uniform sampler2D bumpTexture;
uniform sampler2D bump1Texture;
uniform sampler2D bump2Texture;
uniform sampler2D maskTexture;

uniform sampler2D specularTexture;

struct Light {
	vec4 diffuseColor;
	vec4 specularColor;
	vec4 direction;
};

layout(std140) uniform LightData {
	vec4 cameraPosition;
	vec4 ambientColor;
	Light lights[3];
} lightData;

uniform RenderParameters {
	float bumpSpecularGloss;
	float bumpSpecularAmount;
	float bump1UVScale;
	float bump2UVScale;
} parameters;

layout(std140) uniform AlphaTest {
	uint mode; // 0 - none, 1 - pass if greater than, 2 - pass if less than.
	float reference;
} alphaTest;

void main()
{
	// Find diffuse texture and do alpha test.
	vec4 diffuseTexColor = texture(diffuseTexture, outTexCoord);
	if ((alphaTest.mode == 1U && diffuseTexColor.a <= alphaTest.reference) || (alphaTest.mode == 2U && diffuseTexColor.a >= alphaTest.reference))
		discard;

	// Separate specular color
	vec4 specularColor = texture(specularTexture, outTexCoord);

	// Base diffuse color
	vec4 diffuseColor = diffuseTexColor * outColor;

	// Calculate normal
	vec4 normalMap = texture(bumpTexture, outTexCoord);
	vec4 detailNormalMap1 = texture(bump1Texture, outTexCoord * parameters.bump1UVScale);
	vec4 detailNormalMap2 = texture(bump2Texture, outTexCoord * parameters.bump2UVScale);
	vec4 maskColor = texture(maskTexture, outTexCoord);

	vec3 normalFromMap = (normalMap.rgb + detailNormalMap1.rgb * maskColor.r + detailNormalMap2.rgb * maskColor.g) * 2 - 1;
```

Problem: 3D ist schwer



Problem: 3D ist schwer



Trend: Das wird schlimmer!

Trend: Das wird schlimmer!

Klassisches OpenGL ≤ 2.1	Einfach	z.B. Immediate Mode Fixed Function
	Mittel	z.B. Vertex Arrays Shader mit Standarduniforms
	Schwer	z.B. VAOs + VBOs Shader mit eigenen Uniforms

Trend: Das wird schlimmer!

Klassisches OpenGL ≤ 2.1	Einfach	z.B. Immediate Mode Fixed Function
	Mittel	z.B. Vertex Arrays Shader mit Standarduniforms
	Schwer	z.B. VAOs + VBOs Shader mit eigenen Uniforms
Modernes OpenGL ≥ 3.0	Schwer	z.B. VAOs + VBOs Shader für alles
	Noch Schwerer	z.B. Neue Shadertypen, Uniform Buffer

Lösung: SceneKit

- Stellt Scene Graph bereit
  - Alles, was sichtbar ist, ist ein eigenes Objekt.
  - Aussehen, Position usw. sind properties
  - Details des Renderns sind verborgen

Randnotizen

- OS X 10.8
- Basiert auf OpenGL
- Unterstützt COLLADA
  - Offenes Dateiformat für 3D-Dateien
- Version 1.0
  - Sehr eingeschränkte Dokumentation

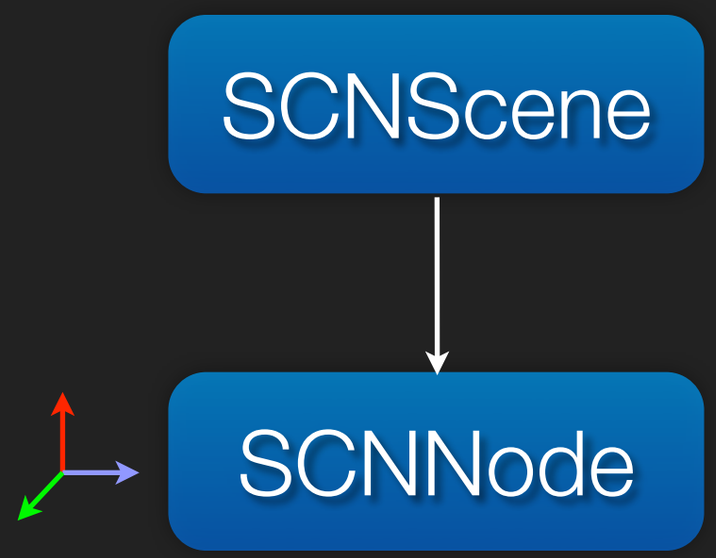
Anwendung

- SCNScene erzeugen
- Mit SCNView verbinden
- Optional:
  - Eigene Geometrie hinzufügen
  - Material ändern
  - Animieren

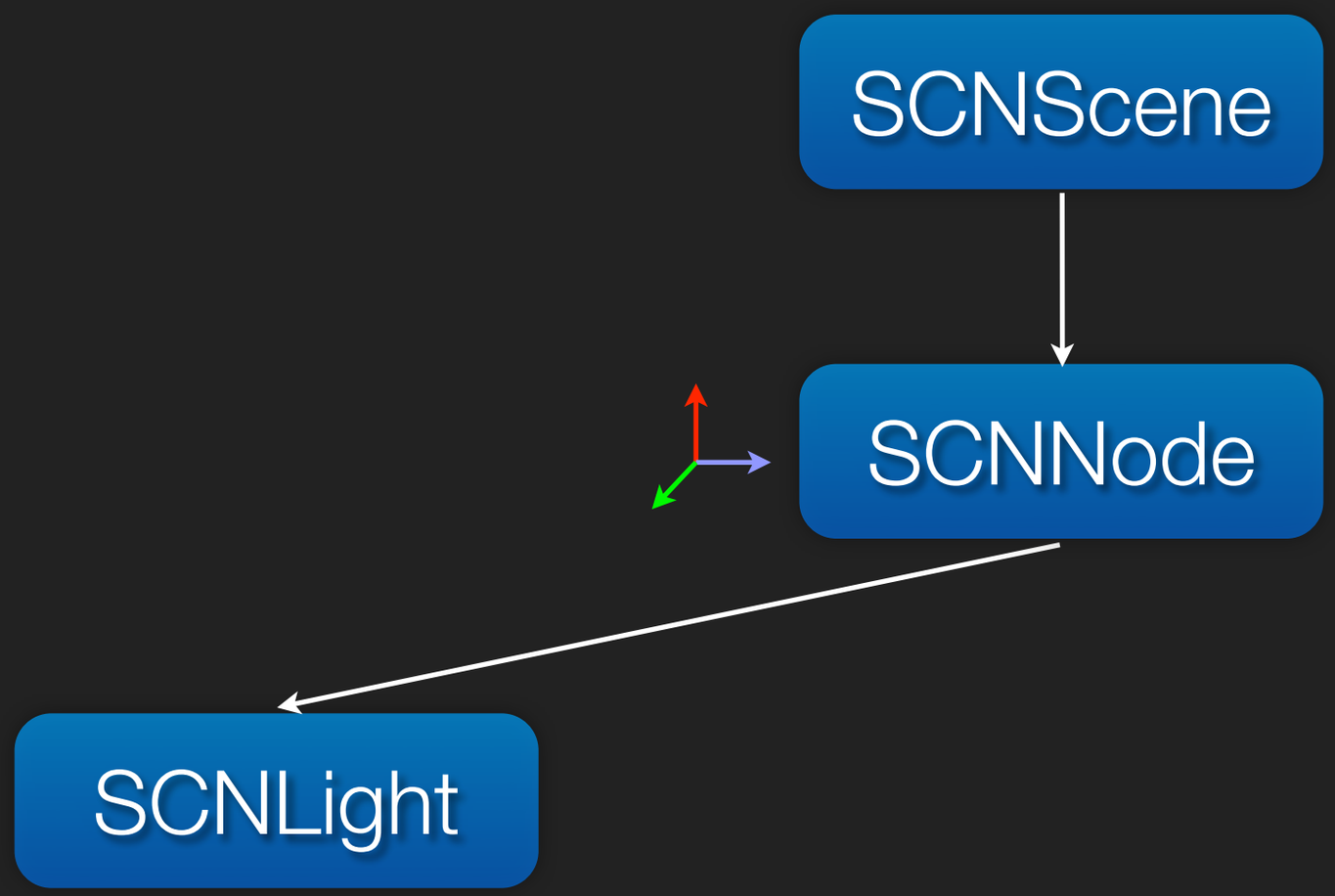
Wichtige Klassen

SCNScene

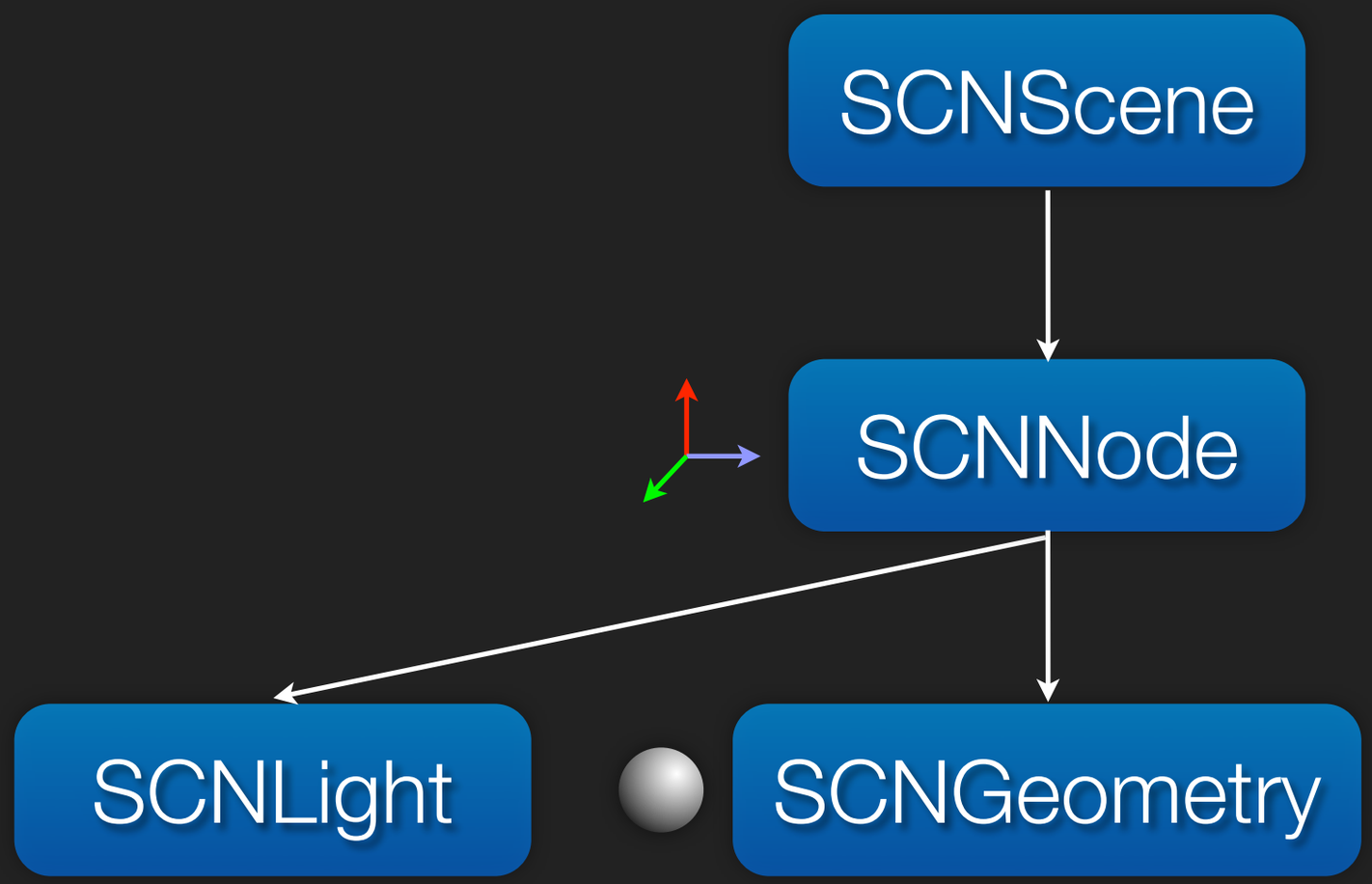
Wichtige Klassen



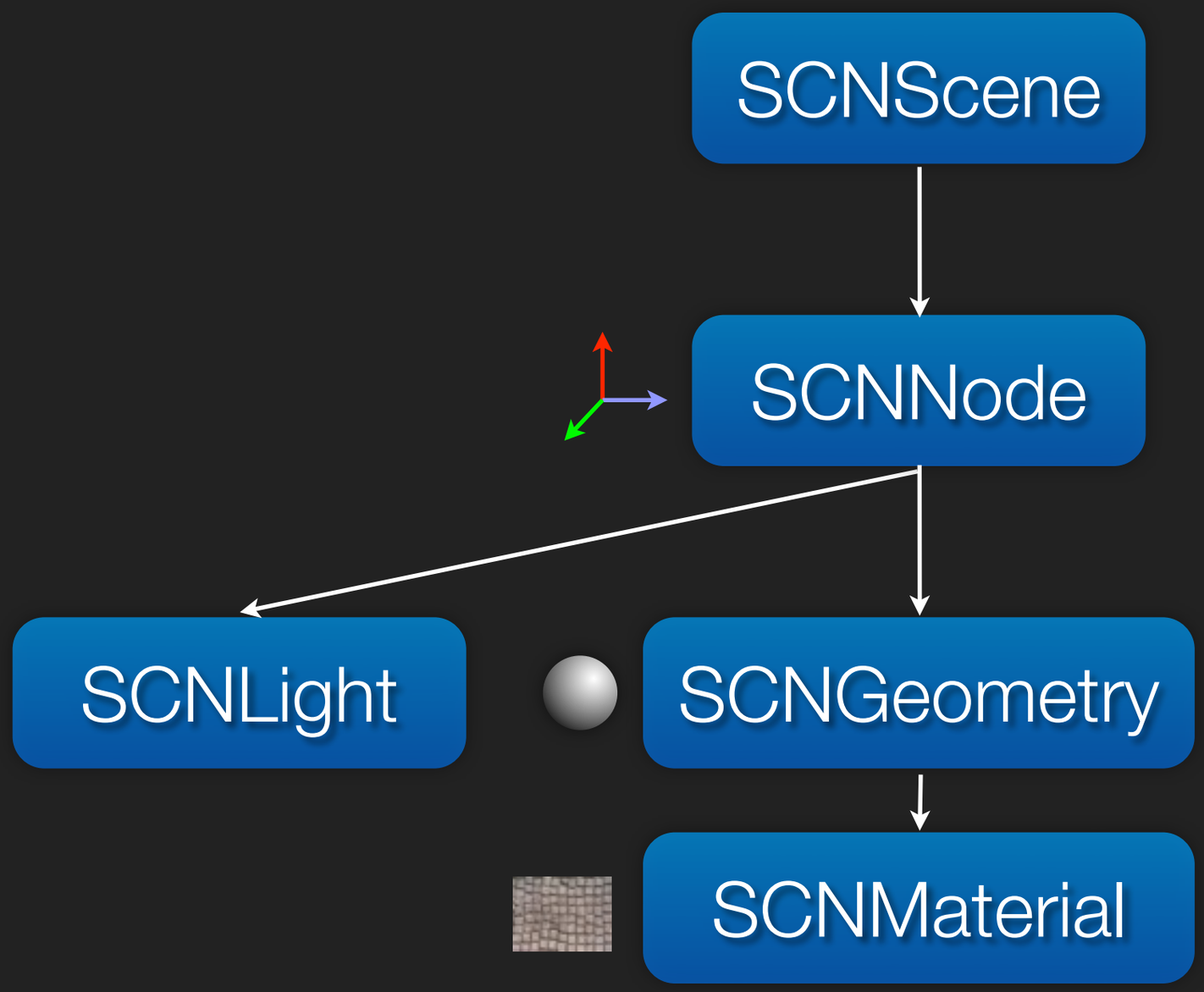
Wichtige Klassen



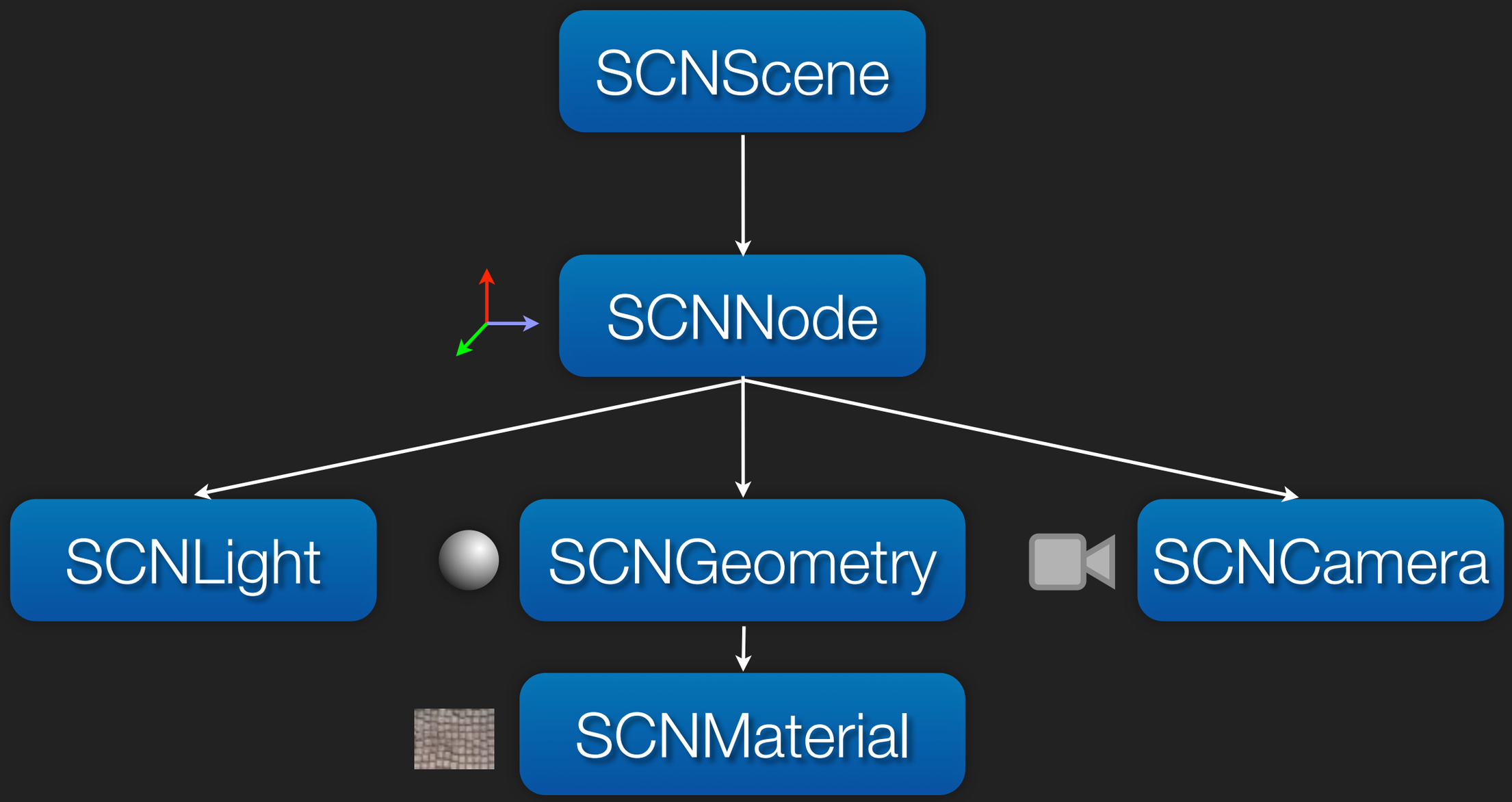
Wichtige Klassen



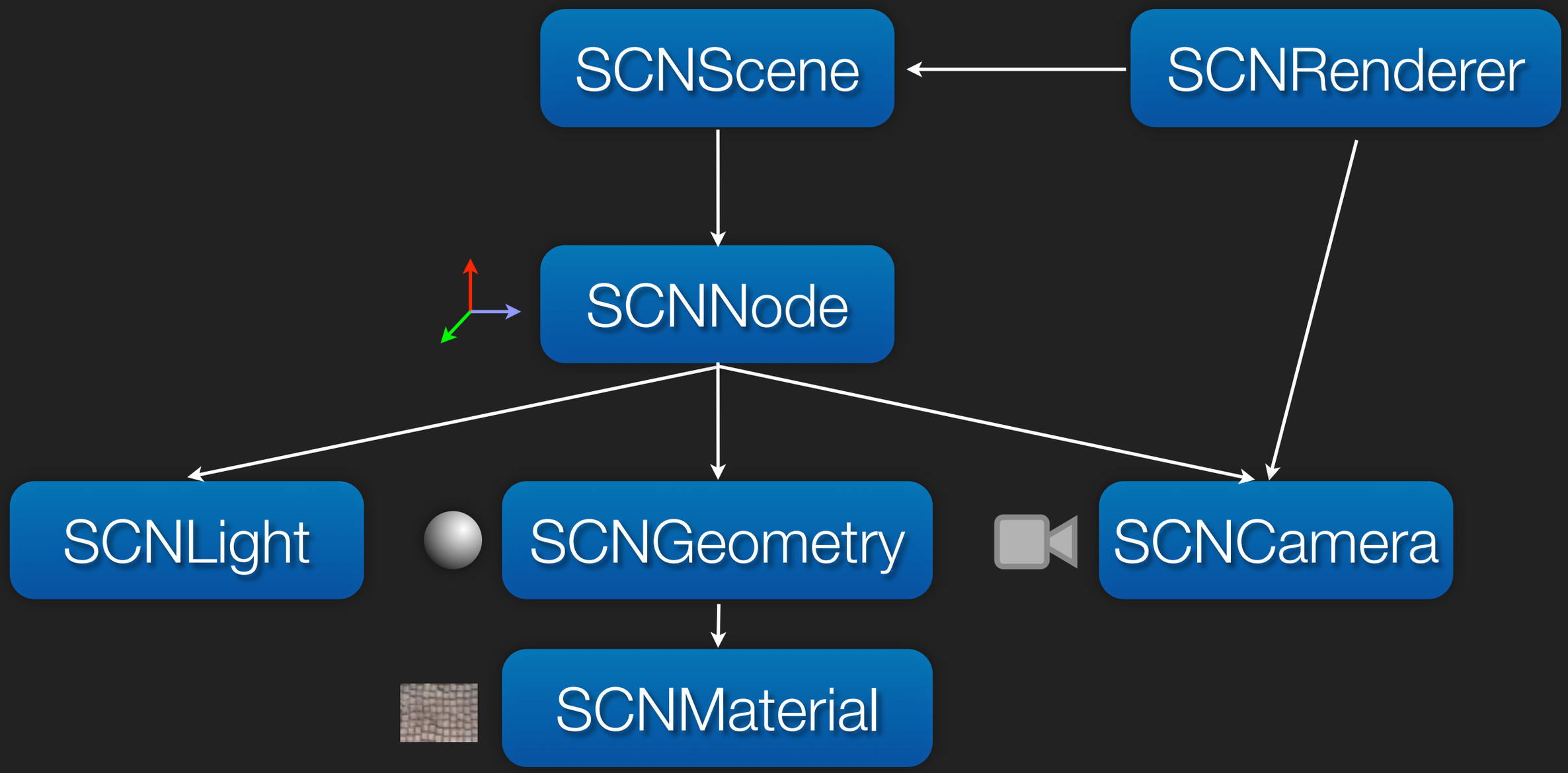
Wichtige Klassen



Wichtige Klassen



Wichtige Klassen



Eigene Geometrie

- Für jede Komponente: SCNGeometrySource
- Für jeden Teil mit eigenem Material: SCNGeometryElement
- Kombiniert zu SCNGeometry

Eigene Shader

Eigene Shader

- Wüsste ich auch gerne!

Eigene Shader

- Wüsste ich auch gerne!
- Speziell:
  - Welcher Typ
  - Wie werden Texturen geladen?
  - Welches Mesh wird gerendert?