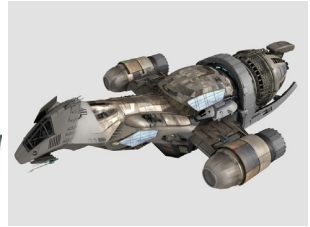
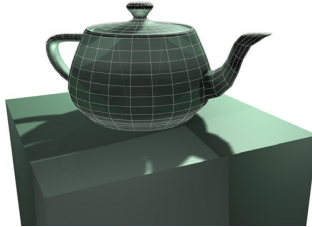


Grafikprogrammierung mit C++ und OpenGL

Seminar Introduction: Lectures, Hands-On, Collaborative Open Source Project



Stefan Buschmann, Daniel Limberger, Amir Semmo

Hasso-Plattner-Institut SoSe 2013 v10272016

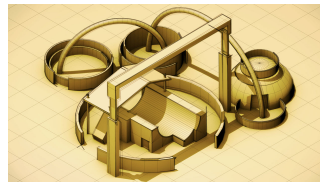
Agenda



1) Seminar Concept



2) Skill Survey



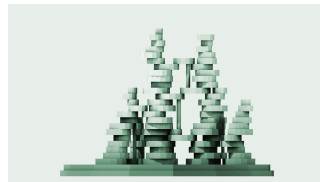
3) Seminar Structure



4) Collaborative Project



5) Your Participation



6) Getting Started



CONCEPT

You are going to

- get advanced C/C++ (2/3) and OpenGL (1/3) input.
- work on a collaborative, open source CG project.
- develop cross platform (GCC, MSVC, XCode, etc.).
- know how to handle Qt and CMake.
- assess, justify, discuss, and criticize source code.

You are going to

- get advanced C/C++ (2/3) and OpenGL (1/3) input.
- work on a collaborative, open source CG project.
- develop cross platform (GCC, MSVC, XCode, etc.).
- know how to handle Qt and CMake.
- assess, justify, discuss, and criticize source code.

You are going to

- get advanced C/C++ (2/3) and OpenGL (1/3) input.
- work on a collaborative, open source CG project.
- develop cross platform (GCC, MSVC, XCode, etc.).
- know how to handle Qt and CMake.
- assess, justify, discuss, and criticize source code.

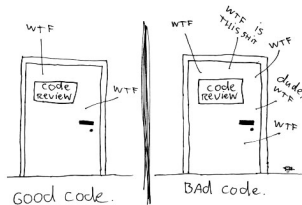
You are going to

- get advanced C/C++ (2/3) and OpenGL (1/3) input.
- work on a collaborative, open source CG project.
- develop cross platform (GCC, MSVC, XCode, etc.).
- know how to handle Qt and CMake.
- assess, justify, discuss, and criticize source code.

You are going to

- get advanced C/C++ (2/3) and OpenGL (1/3) input.
- work on a collaborative, open source CG project.
- develop cross platform (GCC, MSVC, XCode, etc.).
- know how to handle Qt and CMake.
- assess, justify, discuss, and criticize source code.

The ONLY valid MEASUREMENT
OF CODE QUALITY: WTFs/minute



You are going to

- get advanced C/C++ (2/3) and OpenGL (1/3) input.
- work on a collaborative, open source CG project.
- develop cross platform (GCC, MSVC, XCode, etc.).
- know how to handle Qt and CMake.
- assess, justify, discuss, and criticize source code.

This seminar is all **NEW** - new content, new slides, new project -
and You will (if so) participate in its **first pass**.

Your interests, ideas, and feedback are valueable for seminar quality and success.

This seminar is all **NEW** - new content, new slides, new project -
and You will (if so) participate in its **first pass**.

Your interests, ideas, and feedback are valueable for seminar quality and success.

The background of the slide is a repeating isometric pattern. It consists of a grid of light yellow squares, each containing a stylized green tree with a brown trunk. The trees are arranged in a staggered fashion across the grid. A white rectangular banner is positioned in the upper right area of the slide, containing the text 'SKILL SURVEY' in bold black capital letters.

SKILL SURVEY

Skill Survey 1/4 - C/C++ Experience



WHO has worked on more than two C/C++ **projects**?

WHO knows the meaning of more than half of the following C/C++ **keywords**?

```
1      auto break class const constexpr const_cast continue delete default
2      dynamic_cast enum explicit friend inline mutable namespace nullptr
3      operator overwrite private protected reinterpret_cast signed sizeof
4      static static_cast template typedef typename union using virtual void
```

WHO is familiar with more than half of the following **concepts**?

Unnamed Namespaces; References; Operator Overloading; Function Pointer; Multiple Inheritance; Pure Virtual; Smart-Pointer;
Memory Fragmentation; Run Time Type Information; Forward Declarations; Precompiled Headers; Macros; Include Guards;
Containers and Iterators; Lazy Initialization;

WHO has worked on at least one **platform independent** C/C++ project?

WHO has worked with **CMake**?

Skill Survey 1/4 - C/C++ Experience

WHO has worked on more than two C/C++ **projects**?

WHO knows the meaning of more than half of the following C/C++ **keywords**?

```
1      auto break class const constexpr const_cast continue delete default
2      dynamic_cast enum explicit friend inline mutable namespace nullptr
3      operator overwrite private protected reinterpret_cast signed sizeof
4      static static_cast template typedef typename union using virtual void
```

WHO is familiar with more than half of the following **concepts**?

Unnamed Namespaces; References; Operator Overloading; Function Pointer; Multiple Inheritance; Pure Virtual; Smart-Pointer;
Memory Fragmentation; Run Time Type Information; Forward Declarations; Precompiled Headers; Macros; Include Guards;
Containers and Iterators; Lazy Initialization;

WHO has worked on at least one **platform independent** C/C++ project?

WHO has worked with **CMake**?

Skill Survey 1/4 - C/C++ Experience

WHO has worked on more than two C/C++ **projects**?

WHO knows the meaning of more than half of the following C/C++ **keywords**?

```
1      auto break class const constexpr const_cast continue delete default
2      dynamic_cast enum explicit friend inline mutable namespace nullptr
3      operator overwrite private protected reinterpret_cast signed sizeof
4      static static_cast template typedef typename union using virtual void
```

WHO is familiar with more than half of the following **concepts**?

Unnamed Namespaces; References; Operator Overloading; Function Pointer; Multiple Inheritance; Pure Virtual; Smart-Pointer;
Memory Fragmentation; Run Time Type Information; Forward Declarations; Precompiled Headers; Macros; Include Guards;
Containers and Iterators; Lazy Initialization;

WHO has worked on at least one **platform independent** C/C++ project?

WHO has worked with **CMake**?

Skill Survey 1/4 - C/C++ Experience

WHO has worked on more than two C/C++ **projects**?

WHO knows the meaning of more than half of the following C/C++ **keywords**?

```
1      auto break class const constexpr const_cast continue delete default
2      dynamic_cast enum explicit friend inline mutable namespace nullptr
3      operator overwrite private protected reinterpret_cast signed sizeof
4      static static_cast template typedef typename union using virtual void
```

WHO is familiar with more than half of the following **concepts**?

Unnamed Namespaces; References; Operator Overloading; Function Pointer; Multiple Inheritance; Pure Virtual; Smart-Pointer; Memory Fragmentation; Run Time Type Information; Forward Declarations; Precompiled Headers; Macros; Include Guards; Containers and Iterators; Lazy Initialization;

WHO has worked on at least one **platform independent** C/C++ project?

WHO has worked with **CMake**?

Skill Survey 1/4 - C/C++ Experience

WHO has worked on more than two C/C++ **projects**?

WHO knows the meaning of more than half of the following C/C++ **keywords**?

```
1      auto break class const constexpr const_cast continue delete default
2      dynamic_cast enum explicit friend inline mutable namespace nullptr
3      operator overwrite private protected reinterpret_cast signed sizeof
4      static static_cast template typedef typename union using virtual void
```

WHO is familiar with more than half of the following **concepts**?

Unnamed Namespaces; References; Operator Overloading; Function Pointer; Multiple Inheritance; Pure Virtual; Smart-Pointer;
Memory Fragmentation; Run Time Type Information; Forward Declarations; Precompiled Headers; Macros; Include Guards;
Containers and Iterators; Lazy Initialization;

WHO has worked on at least one **platform independent** C/C++ project?

WHO has worked with **CMake**?

Skill Survey 1/4 - C/C++ Experience

WHO has worked on more than two C/C++ **projects**?

WHO knows the meaning of more than half of the following C/C++ **keywords**?

```
1      auto break class const constexpr const_cast continue delete default
2      dynamic_cast enum explicit friend inline mutable namespace nullptr
3      operator overwrite private protected reinterpret_cast signed sizeof
4      static static_cast template typedef typename union using virtual void
```

WHO is familiar with more than half of the following **concepts**?

Unnamed Namespaces; References; Operator Overloading; Function Pointer; Multiple Inheritance; Pure Virtual; Smart-Pointer; Memory Fragmentation; Run Time Type Information; Forward Declarations; Precompiled Headers; Macros; Include Guards; Containers and Iterators; Lazy Initialization;

WHO has worked on at least one **platform independent** C/C++ project?

WHO has worked with **CMake**?

```
1      int a = 13, b = 013;  
2      a ^= b ^= a ^= b;
```

```
1      const QString t("Hello_World");  
2      const char * c(t.toLocal8Bit().constData());
```

```
1      const float  f(const float  x) {  
2          return 3.f * x; }  
3      const double f(const double x) {  
4          return 2.  * x; }  
5  
6      f(2); // ?
```

```
1      void foo(int const * a, int * const b);
```

```
1      int a = 13, b = 013;  
2      a ^= b ^= a ^= b;
```

```
1      const QString t("Hello_World");  
2      const char * c(t.toLocal8Bit().constData());
```

```
1      const float  f(const float  x) {  
2          return 3.f * x; }  
3      const double f(const double x) {  
4          return 2.  * x; }  
5  
6      f(2); // ?
```

```
1      void foo(int const * a, int * const b);
```

```
1      int a = 13, b = 013;  
2      a ^= b ^= a ^= b;
```

```
1      const QString t("Hello_World");  
2      const char * c(t.toLocal8Bit().constData());
```

```
1      const float  f(const float  x) {  
2          return 3.f * x; }  
3      const double f(const double x) {  
4          return 2.  * x; }  
5  
6      f(2); // ?
```

```
1      void foo(int const * a, int * const b);
```

```
1      int a = 13, b = 013;  
2      a ^= b ^= a ^= b;
```

```
1      const QString t("Hello_World");  
2      const char * c(t.toLocal8Bit().constData());
```

```
1      const float  f(const float  x) {  
2          return 3.f * x; }  
3      const double f(const double x) {  
4          return 2.  * x; }  
5  
6      f(2); // ?
```

```
1      void foo(int const * a, int * const b);
```

```
1      int a = 13, b = 013;  
2      a ^= b ^= a ^= b;
```

```
1      const QString t("Hello_World");  
2      const char * c(t.toLocal8Bit().constData());
```

```
1      const float  f(const float  x) {  
2          return 3.f * x; }  
3      const double f(const double x) {  
4          return 2.  * x; }  
5  
6      f(2); // ?
```

```
1      void foo(int const * a, int * const b);
```

Skill Survey 3/4 - Graphics Experience



WHO has worked on more than two **projects** with graphics programming involved?

WHO knows which of the following functions are **not deprecated** in OpenGL 3.2?

1	Rotate Begin End GenLists Vertex MultiTexCoord LoadIdentity PushMatrix InterleavedArrays
2	MultTransposeMatrix Rotate Scale Translate TexGen Material Light ColorMaterial ShadeModel

WHO has used the GPU for **computational** stuff?

WHO is familiar with more than half of the following rendering **concepts**?

Fragment; Texel; Pixel; Image; Texture; Path Tracing; Immutable Texture; Vertex Buffer Object; Ray Casting; Transform Feedback; Instancing; Shader; Program; Pipeline; Atomic Counter; Texture Views; Bindless Texture;

WHO has seen at least one of these **guys**?

Skill Survey 3/4 - Graphics Experience



WHO has worked on more than two **projects** with graphics programming involved?

WHO knows which of the following functions are **not deprecated** in OpenGL 3.2?

1	Rotate Begin End GenLists Vertex MultiTexCoord LoadIdentity PushMatrix InterleavedArrays
2	MultTransposeMatrix Rotate Scale Translate TexGen Material Light ColorMaterial ShadeModel

WHO has used the GPU for **computational** stuff?

WHO is familiar with more than half of the following rendering **concepts**?

Fragment; Texel; Pixel; Image; Texture; Path Tracing; Immutable Texture; Vertex Buffer Object; Ray Casting; Transform Feedback; Instancing; Shader; Program; Pipeline; Atomic Counter; Texture Views; Bindless Texture;

WHO has seen at least one of these **guys**?

Skill Survey 3/4 - Graphics Experience



WHO has worked on more than two **projects** with graphics programming involved?

WHO knows which of the following functions are **not deprecated** in OpenGL 3.2?

1	Rotate Begin End GenLists Vertex MultiTexCoord LoadIdentity PushMatrix InterleavedArrays
2	MultTransposeMatrix Rotate Scale Translate TexGen Material Light ColorMaterial ShadeModel

WHO has used the GPU for **computational** stuff?

WHO is familiar with more than half of the following rendering **concepts**?

Fragment; Texel; Pixel; Image; Texture; Path Tracing; Immutable Texture; Vertex Buffer Object; Ray Casting; Transform Feedback; Instancing; Shader; Program; Pipeline; Atomic Counter; Texture Views; Bindless Texture;

WHO has seen at least one of these **guys**?

WHO has worked on more than two **projects** with graphics programming involved?

WHO knows which of the following functions are **not deprecated** in OpenGL 3.2?

1	Rotate Begin End GenLists Vertex MultiTexCoord LoadIdentity PushMatrix InterleavedArrays
2	MultTransposeMatrix Rotate Scale Translate TexGen Material Light ColorMaterial ShadeModel

WHO has used the GPU for **computational** stuff?

WHO is familiar with more than half of the following rendering **concepts**?

Fragment; Texel; Pixel; Image; Texture; Path Tracing; Immutable Texture; Vertex Buffer Object; Ray Casting; Transform Feedback; Instancing; Shader; Program; Pipeline; Atomic Counter; Texture Views; Bindless Texture;

WHO has seen at least one of these **guys**?

WHO has worked on more than two **projects** with graphics programming involved?

WHO knows which of the following functions are **not deprecated** in OpenGL 3.2?

1	Rotate Begin End GenLists Vertex MultiTexCoord LoadIdentity PushMatrix InterleavedArrays
2	MultTransposeMatrix Rotate Scale Translate TexGen Material Light ColorMaterial ShadeModel

WHO has used the GPU for **computational** stuff?

WHO is familiar with more than half of the following rendering **concepts**?

Fragment; Texel; Pixel; Image; Texture; Path Tracing; Immutable Texture; Vertex Buffer Object; Ray Casting; Transform Feedback; Instancing; Shader; Program; Pipeline; Atomic Counter; Texture Views; Bindless Texture;

WHO has seen at least one of these **guys**?

Skill Survey 3/4 - Graphics Experience

WHO has worked on more than two **projects** with graphics programming involved?

WHO knows which of the following functions are **not deprecated** in OpenGL 3.2?

1	Rotate Begin End GenLists Vertex MultiTexCoord LoadIdentity PushMatrix InterleavedArrays
2	MultTransposeMatrix Rotate Scale Translate TexGen Material Light ColorMaterial ShadeModel

WHO has used the GPU for **computational** stuff?

WHO is familiar with more than half of the following rendering **concepts**?

Fragment; Texel; Pixel; Image; Texture; Path Tracing; Immutable Texture; Vertex Buffer Object; Ray Casting; Transform Feedback; Instancing; Shader; Program; Pipeline; Atomic Counter; Texture Views; Bindless Texture;

WHO has seen at least one of these **guys**?



```
1      in vec3 a;  
2      varying vec3 b;  
3      uniform vec3 c;  
4      const vec3 d;  
5      out vec3 e;
```

```
1      gl_FragColor = mix(a, d, 0.2);
```

```
1      out vec4 a;  
2      layout(location = 0) out vec4 b;
```

```
1      vec2 n = noise2(fragID);  
2      vec4 t;  
3  
4      if(n < 0.333)  
5          t = texelFetch(sampler, n.yx);  
6      else  
7          t = texelFetch(sampler, n.xy);
```

```
1      in vec3 a;  
2      varying vec3 b;  
3      uniform vec3 c;  
4      const vec3 d;  
5      out vec3 e;
```

```
1      gl_FragColor = mix(a, d, 0.2);
```

```
1      out vec4 a;  
2      layout(location = 0) out vec4 b;
```

```
1      vec2 n = noise2(fragID);  
2      vec4 t;  
3  
4      if(n < 0.333)  
5          t = texelFetch(sampler, n.yx);  
6      else  
7          t = texelFetch(sampler, n.xy);
```

```
1      in vec3 a;  
2      varying vec3 b;  
3      uniform vec3 c;  
4      const vec3 d;  
5      out vec3 e;
```

```
1      gl_FragColor = mix(a, d, 0.2);
```

```
1      out vec4 a;  
2      layout(location = 0) out vec4 b;
```

```
1      vec2 n = noise2(fragID);  
2      vec4 t;  
3  
4      if(n < 0.333)  
5          t = texelFetch(sampler, n.yx);  
6      else  
7          t = texelFetch(sampler, n.xy);
```



```
1      in vec3 a;  
2      varying vec3 b;  
3      uniform vec3 c;  
4      const vec3 d;  
5      out vec3 e;
```

```
1      gl_FragColor = mix(a, d, 0.2);
```

```
1      out vec4 a;  
2      layout(location = 0) out vec4 b;
```

```
1      vec2 n = noise2(fragID);  
2      vec4 t;  
3  
4      if(n < 0.333)  
5          t = texelFetch(sampler, n.yx);  
6      else  
7          t = texelFetch(sampler, n.xy);
```

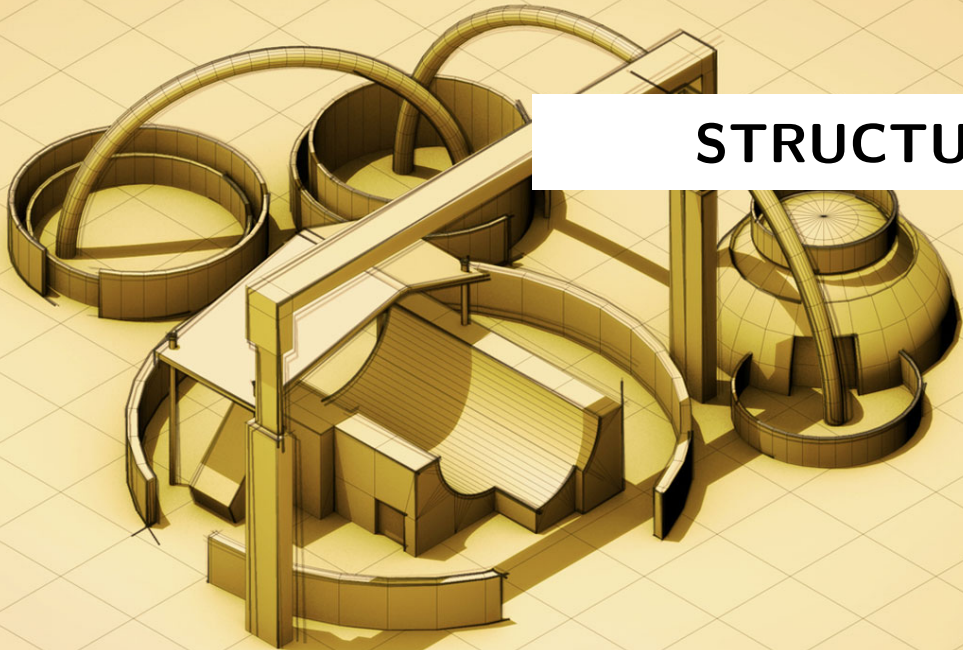
```
1      in vec3 a;  
2      varying vec3 b;  
3      uniform vec3 c;  
4      const vec3 d;  
5      out vec3 e;
```

```
1      gl_FragColor = mix(a, d, 0.2);
```

```
1      out vec4 a;  
2      layout(location = 0) out vec4 b;
```

```
1      vec2 n = noise2(fragID);  
2      vec4 t;  
3  
4      if(n < 0.333)  
5          t = texelFetch(sampler, n.yx);  
6      else  
7          t = texelFetch(sampler, n.xy);
```

STRUCTURE



April

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

May


M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

June

M	T	W	T	F	S	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

July

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

 Lecture

 Hands-On

■ 13 × Lecture

C++, CMake, OpenGL/GLSL, Qt, code riddles and assessments

■ Individual Tasks (2 - 3 × each)

Contribution to a collaborative open source project: CGSee

■ 7 × Hands-On

Programming sessions with everyone: working on individual task, joint inspection of and feedback for ideas and results.

■ Short Talks (3 + 2 min)

Consecutive short presentations of CGSee contributions.

April

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

May


M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

June

M	T	W	T	F	S	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

July

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

 Lecture

 Hands-On

■ 13 × Lecture

C++, CMake, OpenGL/GLSL, Qt, code riddles and assessments

■ Individual Tasks (2 - 3 × each)

Contribution to a collaborative open source project: *CGSee*

■ 7 × Hands-On

Programming sessions with everyone: working on individual task, joint inspection of and feedback for ideas and results.

■ Short Talks (3 + 2 min)

Consecutive short presentations of *CGSee* contributions.

April

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

May


M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

June

M	T	W	T	F	S	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

July

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

 Lecture

 Hands-On

■ 13 × Lecture

C++, CMake, OpenGL/GLSL, Qt, code riddles and assessments

■ Individual Tasks (2 - 3 × each)

Contribution to a collaborative open source project: *CGSee*

■ 7 × Hands-On

Programming sessions with everyone: working on individual task, joint inspection of and feedback for ideas and results.

■ Short Talks (3 + 2 min)

Consecutive short presentations of *CGSee* contributions.

April

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

May


M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

June

M	T	W	T	F	S	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

July

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

 Lecture

 Hands-On

■ 13 × Lecture

C++, CMake, OpenGL/GLSL, Qt, code riddles and assessments

■ Individual Tasks (2 - 3 × each)

Contribution to a collaborative open source project: *CGSee*

■ 7 × Hands-On

Programming sessions with everyone: working on individual task, joint inspection of and feedback for ideas and results.

■ Short Talks (3 + 2 min)

Consecutive short presentations of *CGSee* contributions.

April

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

May


M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

June

M	T	W	T	F	S	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

July

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

 Lecture

 Hands-On

■ 13 × Lecture

C++, CMake, OpenGL/GLSL, Qt, code riddles and assessments

■ Individual Tasks (2 - 3 × each)

Contribution to a collaborative open source project: *CGSee*

■ 7 × Hands-On

Programming sessions with everyone: working on individual task, joint inspection of and feedback for ideas and results.

■ Short Talks (3 + 2 min)

Consecutive short presentations of *CGSee* contributions.

Seminar Content

04/10 Introduction, project and process, issues for **individual tasks**

04/17 Initial **tasks assignment**, source introduction, and project planning

04/19 Review & hands-on

04/24 C++_1

05/01 nullptr

05/03 Review & hands-on

05/08 OpenGL_1

05/15 C++_2

05/17 Review & hands-on

05/22 Cross-Platform

05/29 Qt_1

05/31 Review & hands-On

06/05 OpenGL_2

06/12 CPP_3

06/14 Review & hands-On

06/19 OpenGL_3

06/26 C++_4

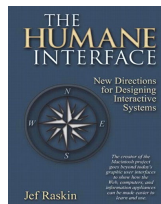
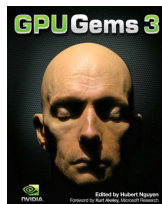
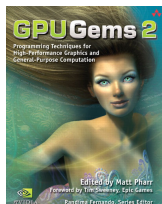
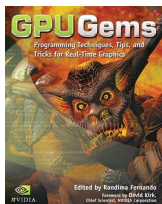
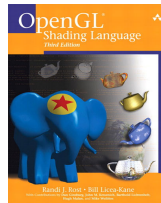
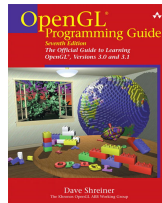
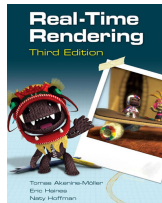
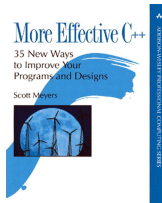
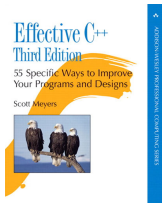
06/28 Review & hands-On

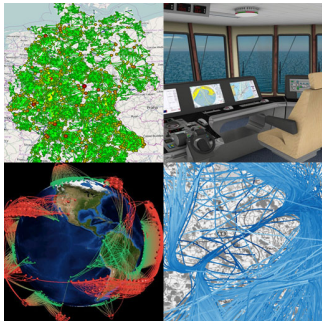
07/03 C++_5

07/10 **Buffer zone, discussion, feedback**

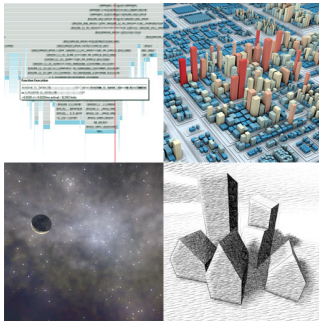
07/12 **CGSee evaluation, status review, future work, feedback**

Content References (Books)

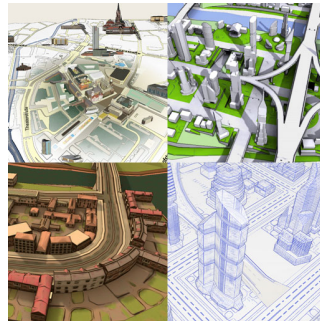




Stefan Buschmann

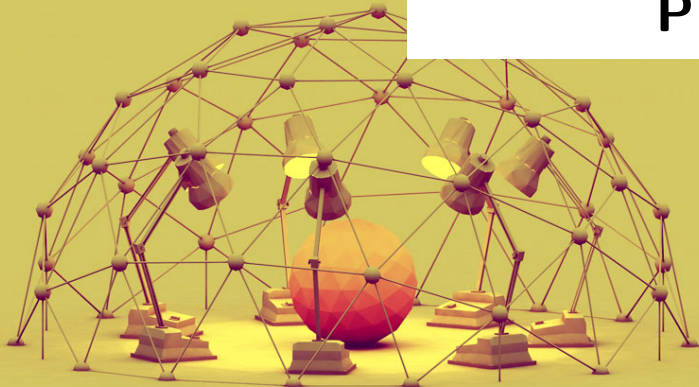


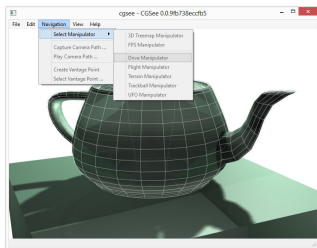
Daniel Limberger



Amir Semmo

PROJECT





CGSee

Working Title

Free open source viewer for computer graphics related data,
intended to become a Deep Exploration replacement.

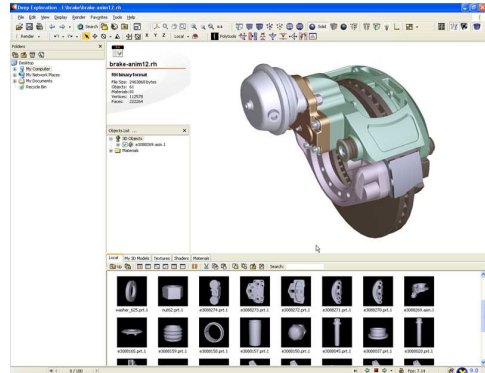
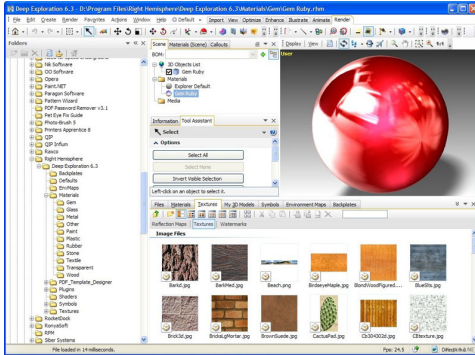
- Rendering, navigation, and conversion of 3D geometry
- 3D Geometry is exchanged and used in various areas of CG.
Movies, games, logos, CAD, evaluation of rendering techniques
- 3D file format hell
- 3D meshes vs. 3D scenes
- image based and procedural materials
- animation and dynamics

- Rendering, navigation, and conversion of 3D geometry
- 3D Geometry is exchanged and used in various areas of CG.
Movies, games, logos, CAD, evaluation of rendering techniques
- 3D file format hell
- 3D meshes vs. 3D scenes
- image based and procedural materials
- animation and dynamics

- Rendering, navigation, and conversion of 3D geometry
- 3D Geometry is exchanged and used in various areas of CG.
Movies, games, logos, CAD, evaluation of rendering techniques
- 3D file format hell
- 3D meshes vs. 3D scenes
- image based and procedural materials
- animation and dynamics

File Format	Type	Extension	File Format	Type	Extension
Windows Cursor File	Image	CUR	OpenFlight Scene Description Database	Scene	FLT
Windows Icon File	Image	ICO	OpenGL CPP Code	Scene	CPP
Windows Run Length Encoding Bitmap	Image	RLE	Points File	Scene	PTS
Autodesk 3D Studio	Scene	3DS	Polygon Model Format	Scene	PLY
Autodesk 3D Studio ASCII	Scene	ASC	Portable Graymap Graphics File	Scene	PGM
Autodesk 3D Studio Max ASCII	Scene	ASE	POV-Ray RAWTriangle Format	Scene	RAW
Autodesk 3D Studio Project	Scene	PRJ	Power Render Object	Scene	PRO
AOFF File Format	Scene	GEO	Protein Data Bank	Scene	PDB
Caligari trueSpace Object	Scene	COB	Quake Character Model I	Scene	MDL
Caligari trueSpace Scene	Scene	SCN	Quake Character Model II	Scene	MD2
CINEMA 4D	Scene	C4D	Quake Character Model III	Scene	MD3
DirectX Model	Scene	X	Quake Map	Scene	MAP
HTML Document with 3D		HTML	Quicktime 3D Metafile	Scene	3DM, 3DMF
Imagine Geometry	Scene	IOB	RAX Extended RAWTriangles	Scene	RAX
ISO G Code	Scene	ISO, NC	Rhinoceros 3D Model	Scene	3DM
LightWave 3D and Binary Object	Scene	LWO, LW	Right Hemisphere Binary	Scene	RH
LightWave Scene	Scene	LWS	Shockwave Web Page	Scene	HTML
Max File	Scene	MAX	SketchUp Document	Scene	SKP
Maya ASCII Scene	Scene	MA	Universal 3D	Scene	U3D
Maya Binary Scene	Scene	MB	VRML Worlds	Scene	WRL, VRML
Nendo	Scene	NDO	Wavefront Object	Scene	OBJ
Object File Format Vector Graphics	Scene	OFF	OpenFlight Scene Description Database	Scene	FLT
Open Inventor File	Scene	IV	OpenGL CPP Code	Scene	CPP

Deep Exploration - RIP



Right Hemisphere got acquired by SAP in 2011.

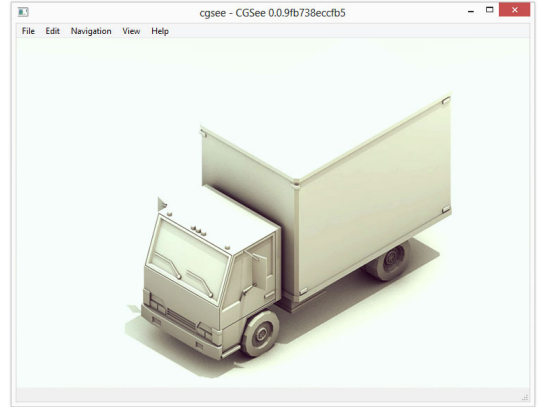
(Their products are now used for SAP Visual Enterprise)

Alternatives: 3D-Tool; Quick3D; Solid; Solid Works; Solid Edge Viewer; Rotor 3D Viewer; ...;

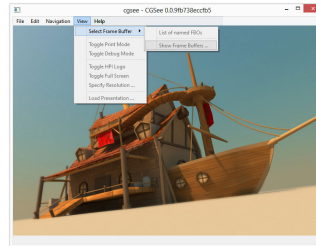
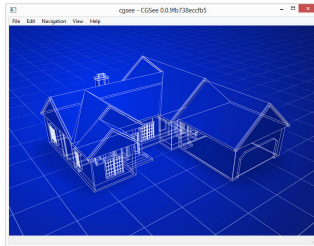
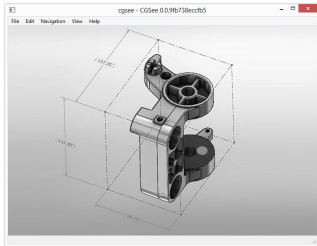
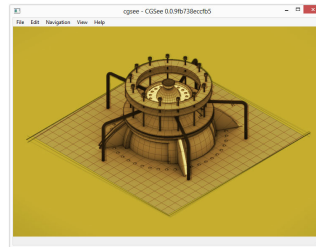
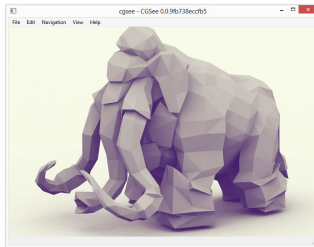
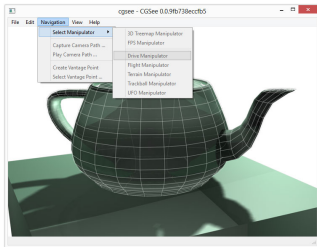
There are currently no similarly good alternatives: Either only CAD oriented, not free, few formats, bad export, bad navigation, old code base, bad UI, etc., ...

Requirements for **CGSee**:

- clean and minimal ui
- common formats for import and export
- basic manipulation capabilities
(e.g., mirroring, geometry fixes, etc.)
- efficient and beautiful rendering
- exploration tools
(e.g., filtering, navigation, scene structure)
- measurements
- ...



Inspiration



Third Parties, Resources, and Process



Revisioning with git
git-scm.com



Hosted on github (public)
github.com/cgcostume/cgsee



Style Guide by google
code.google.com/p/google-styleguide

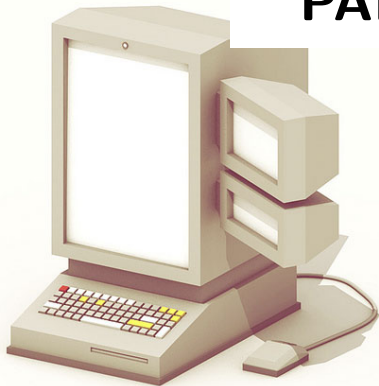


Project Setup with CMake
www.cmake.org



UI and Rendering with Qt5 and GLEW
qt-project.org & glew.sourceforge.net

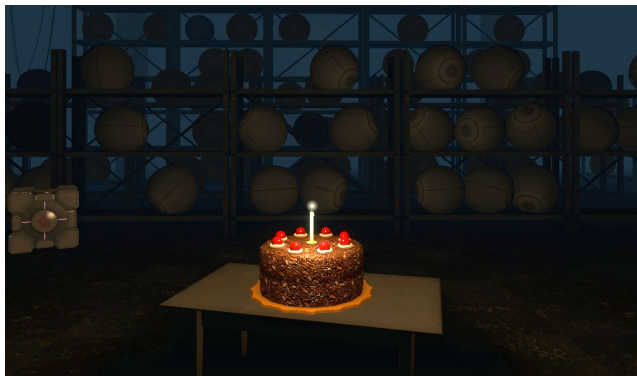
PARTICIPATION



Select/Propose at least two tasks. One in C++, the other in rendering with OpenGL.

For each task You

- 1 **propose** a solution for discussion (during review in hands-on).
- 2 **implement** the (probably refined) solution (2 to 4 weeks).
- 3 **present** your implementation (3 + 2 min).
- 4 pair programming if sufficient number of participants... (group making)



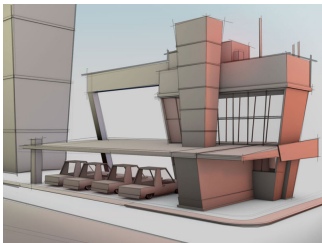
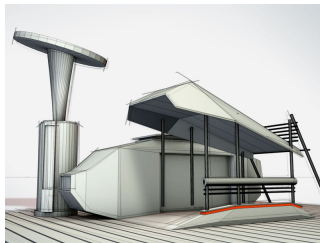
$\frac{1}{4}$ task **presentations** and $\frac{3}{4}$ task **implementations**

Actually it's $\frac{2}{8} + \frac{5}{8}$... with an added $\frac{1}{8}$ for **valuable participation**.

A pixelated green tree made of blocks, resembling a stylized evergreen or a tree made of stacked stones. The tree has a central trunk and several branches, each composed of stacked blocks. The overall shape is somewhat triangular, with a wider base and a narrower top. The blocks are in various shades of green, giving it a textured, pixelated appearance. The tree is positioned on the left side of the image, with its base resting on a flat, green surface.

GET STARTED

More Inspiration



1 Data Import and Export

- Image and 3D Formats
- Scene Graph and Matrices Provisioning
- Materials: Diffuse, Bump, Ambient

2 Efficient Scene Rendering

- View Frustum Culling
- Scene Graph Traversal / State Sorting
- Vertex Cache Optimization

3 Image Based Rendering

- Render To Texture, Antialiasing
- Provisioning of Common G-Buffers
- Shadows, Grid, Environments, Ground Plane
- Lighting, SSAO, Spherical Harmonics
- NPR/Stylization: Blue Print, Pencil

4 Interaction and Analysis

- Navigations: Trackball, Flight, Walk
- Bounding Boxes, Dimensions, Cross Sections, Explosion, Coloring
- Scene Graph and additional Information Display and Search
- Viewport Management and Transitions
- View Cube
- Measuring and Labeling: Angles, Distances, Ruler, Labels, Volumes

5 Manipulation and Creation

- Picking, Hovering, and Highlighting
- Selecting, Filtering, Editing (Gizmos)
- Geometry Fixes: Duplicate Vertices
- Camera Path Editor

And most importantly, **YOUR IDEAS!**

Issues (Excerpt)

1 Data Import and Export

- Image and 3D Formats
- Scene Graph and Matrices Provisioning
- Materials: Diffuse, Bump, Ambient

2 Efficient Scene Rendering

- View Frustum Culling
- Scene Graph Traversal / State Sorting
- Vertex Cache Optimization

3 Image Based Rendering

- Render To Texture, Antialiasing
- Provisioning of Common G-Buffers
- Shadows, Grid, Environments, Ground Plane
- Lighting, SSAO, Spherical Harmonics
- NPR/Stylization: Blue Print, Pencil

4 Interaction and Analysis

- Navigations: Trackball, Flight, Walk
- Bounding Boxes, Dimensions, Cross Sections, Explosion, Coloring
- Scene Graph and additional Information Display and Search
- Viewport Management and Transitions
- View Cube
- Measuring and Labeling: Angles, Distances, Ruler, Labels, Volumes

5 Manipulation and Creation

- Picking, Hovering, and Highlighting
- Selecting, Filtering, Editing (Gizmos)
- Geometry Fixes: Duplicate Vertices
- Camera Path Editor

And most importantly, **YOUR IDEAS!**

Issues (Excerpt)

1 Data Import and Export

- Image and 3D Formats
- Scene Graph and Matrices Provisioning
- Materials: Diffuse, Bump, Ambient

2 Efficient Scene Rendering

- View Frustum Culling
- Scene Graph Traversal / State Sorting
- Vertex Cache Optimization

3 Image Based Rendering

- Render To Texture, Antialiasing
- Provisioning of Common G-Buffers
- Shadows, Grid, Environments, Ground Plane
- Lighting, SSAO, Spherical Harmonics
- NPR/Stylization: Blue Print, Pencil

4 Interaction and Analysis

- Navigations: Trackball, Flight, Walk
- Bounding Boxes, Dimensions, Cross Sections, Explosion, Coloring
- Scene Graph and additional Information Display and Search
- Viewport Management and Transitions
- View Cube
- Measuring and Labeling: Angles, Distances, Ruler, Labels, Volumes

5 Manipulation and Creation

- Picking, Hovering, and Highlighting
- Selecting, Filtering, Editing (Gizmos)
- Geometry Fixes: Duplicate Vertices
- Camera Path Editor

And most importantly, **YOUR IDEAS!**

Issues (Excerpt)

1 Data Import and Export

- Image and 3D Formats
- Scene Graph and Matrices Provisioning
- Materials: Diffuse, Bump, Ambient

2 Efficient Scene Rendering

- View Frustum Culling
- Scene Graph Traversal / State Sorting
- Vertex Cache Optimization

3 Image Based Rendering

- Render To Texture, Antialiasing
- Provisioning of Common G-Buffers
- Shadows, Grid, Environments, Ground Plane
- Lighting, SSAO, Spherical Harmonics
- NPR/Stylization: Blue Print, Pencil

4 Interaction and Analysis

- Navigations: Trackball, Flight, Walk
- Bounding Boxes, Dimensions, Cross Sections, Explosion, Coloring
- Scene Graph and additional Information Display and Search
- Viewport Management and Transitions
- View Cube
- Measuring and Labeling: Angles, Distances, Ruler, Labels, Volumes

5 Manipulation and Creation

- Picking, Hovering, and Highlighting
- Selecting, Filtering, Editing (Gizmos)
- Geometry Fixes: Duplicate Vertices
- Camera Path Editor

And most importantly, **YOUR IDEAS!**

Issues (Excerpt)

1 Data Import and Export

- Image and 3D Formats
- Scene Graph and Matrices Provisioning
- Materials: Diffuse, Bump, Ambient

2 Efficient Scene Rendering

- View Frustum Culling
- Scene Graph Traversal / State Sorting
- Vertex Cache Optimization

3 Image Based Rendering

- Render To Texture, Antialiasing
- Provisioning of Common G-Buffers
- Shadows, Grid, Environments, Ground Plane
- Lighting, SSAO, Spherical Harmonics
- NPR/Stylization: Blue Print, Pencil

4 Interaction and Analysis

- Navigations: Trackball, Flight, Walk
- Bounding Boxes, Dimensions, Cross Sections, Explosion, Coloring
- Scene Graph and additional Information Display and Search
- Viewport Management and Transitions
- View Cube
- Measuring and Labeling: Angles, Distances, Ruler, Labels, Volumes

5 Manipulation and Creation

- Picking, Hovering, and Highlighting
- Selecting, Filtering, Editing (Gizmos)
- Geometry Fixes: Duplicate Vertices
- Camera Path Editor

And most importantly, **YOUR IDEAS!**

Issues (Excerpt)

1 Data Import and Export

- Image and 3D Formats
- Scene Graph and Matrices Provisioning
- Materials: Diffuse, Bump, Ambient

2 Efficient Scene Rendering

- View Frustum Culling
- Scene Graph Traversal / State Sorting
- Vertex Cache Optimization

3 Image Based Rendering

- Render To Texture, Antialiasing
- Provisioning of Common G-Buffers
- Shadows, Grid, Environments, Ground Plane
- Lighting, SSAO, Spherical Harmonics
- NPR/Stylization: Blue Print, Pencil

4 Interaction and Analysis

- Navigations: Trackball, Flight, Walk
- Bounding Boxes, Dimensions, Cross Sections, Explosion, Coloring
- Scene Graph and additional Information Display and Search
- Viewport Management and Transitions
- View Cube
- Measuring and Labeling: Angles, Distances, Ruler, Labels, Volumes

5 Manipulation and Creation

- Picking, Hovering, and Highlighting
- Selecting, Filtering, Editing (Gizmos)
- Geometry Fixes: Duplicate Vertices
- Camera Path Editor

And most importantly, **YOUR IDEAS!**

Issues (Excerpt)

1 Data Import and Export

- Image and 3D Formats
- Scene Graph and Matrices Provisioning
- Materials: Diffuse, Bump, Ambient

2 Efficient Scene Rendering

- View Frustum Culling
- Scene Graph Traversal / State Sorting
- Vertex Cache Optimization

3 Image Based Rendering

- Render To Texture, Antialiasing
- Provisioning of Common G-Buffers
- Shadows, Grid, Environments, Ground Plane
- Lighting, SSAO, Spherical Harmonics
- NPR/Stylization: Blue Print, Pencil

4 Interaction and Analysis

- Navigations: Trackball, Flight, Walk
- Bounding Boxes, Dimensions, Cross Sections, Explosion, Coloring
- Scene Graph and additional Information Display and Search
- Viewport Management and Transitions
- View Cube
- Measuring and Labeling: Angles, Distances, Ruler, Labels, Volumes

5 Manipulation and Creation

- Picking, Hovering, and Highlighting
- Selecting, Filtering, Editing (Gizmos)
- Geometry Fixes: Duplicate Vertices
- Camera Path Editor

And most importantly, **YOUR IDEAS!**

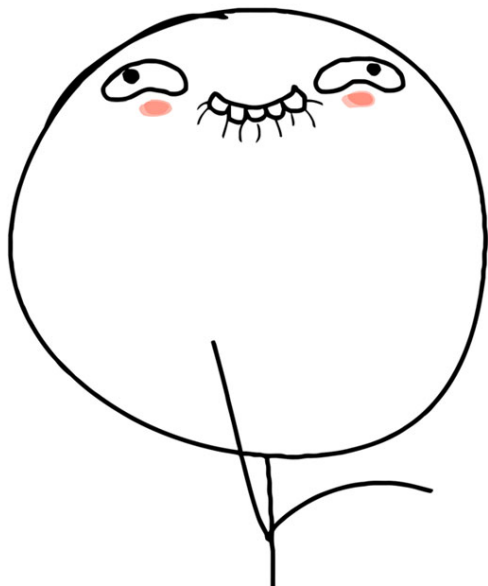
Getting Started on File Formats

3D Geometry File Formats:

- 3DS MAX (.3ds and .max)
- Autodesk (.fbx)
- Auto CAD (.dwg, .dxf)
- Alias WavefrontTM (.obj)
- Blender (.blend)
- Cinema 4D (.c4d)
- Collada (.dae)
- DirectX (.x)
- Lightwave (.lwo and .lws)
- OpenCTM (.ctm)
- QuakeTM (.mdl .md2 and .md3)
- SketchUp (.skp)
- Stanford Polygon PLY (.ply)
- Extensible 3D (.x3d)

2D Image File Formats:

- Windows Bitmap (.bmp)
- JPEG (.jpg)
- Portable RGB (.ppm)
- RAW (.raw)
- TARGA (.tga)
- PNG (.png)
- TIFF (.tiff)
- DXTC (.dds)



OH
STOP
IT, YOU.

Coming Next

- CGSee version control with Git.
- Building CGSee with CMake on gcc/msvc.
- CGSee source code introduction.
- Feature planning and issue assignment.
- Issue tracking in github.

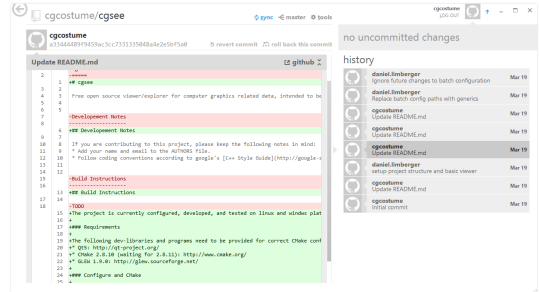


Image Rights

- Teaser 'House' by iStockphoto.com/Martin Fischer
- Teaser 'Tepaot' by Hay Kranen / CC-BY
- Teaser 'Firefly' by Alex Kung
<http://www.seansgallery.com/gallery/3dgraphics/serwip24.jpg>
- OpenGL Guys from OpenGL Quick Reference Card
<http://www.khronos.org/files/opengl43-quick-reference-card.pdf>
- Cake from Valve's game Portal
- Stanford Bunny in Mock by Jean-Christophe Naour (flickr)
- Solidworks Bounding Box from <http://cadcamstuff.com/>
- Boat by Toby Lewin at <http://www.tobylewin.com>
- WTFs/Minute by Tom Holwerda at <http://www.osnews.com/comics>
- Others by Timothy J. Reynolds