ARLAB

ME/CprE/ComS 557

Computer Graphics and Geometric Modeling

Introduction

August 25, 2015

Rafael Radkowski

VRAC|HCI

IOWA STATE UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Rafael Radkowski

ARLAB

1998-2003	Study of mechanical engineering at the University of Paderborn, emphasis to product development.
2003-2006	PhD student at graduate school „Automatic Configuration in Open Systems“ in Paderborn
2006	Graduation at the University of Paderborn, Prof. Gausemeier
seit 2006	Post-Doc in the department of Product Development, Heinz Nixdorf Instituts, Prof. Gausemeier
now	at VRAC

Iowa State University
Virtual Reality Applications Cen
1620 Howe Hall
Ames, IA, 50011

Phone: +1 (515) 294 5880
Fax: +1 (515) 294 5530
E-Mail: rafael@iastate.edu

Lectures (in Germany)

- Virtual and Augmented Reality
- Seminar Virtual Reality
- Computer Graphics, Part 1
- Computer Integrated Manufacturing

VRAC|HCI IOWA STATE UNIVERSITY OF SCIENCE AND TECHNOLOGY

What do you expect from this course?

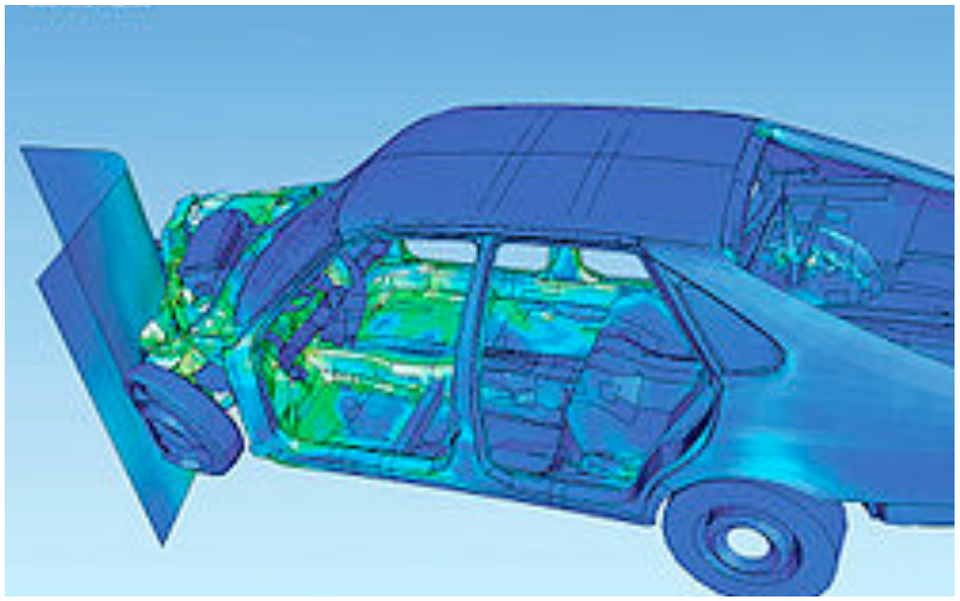
What is Computer Graphics?

Computer Graphics

ARLAB

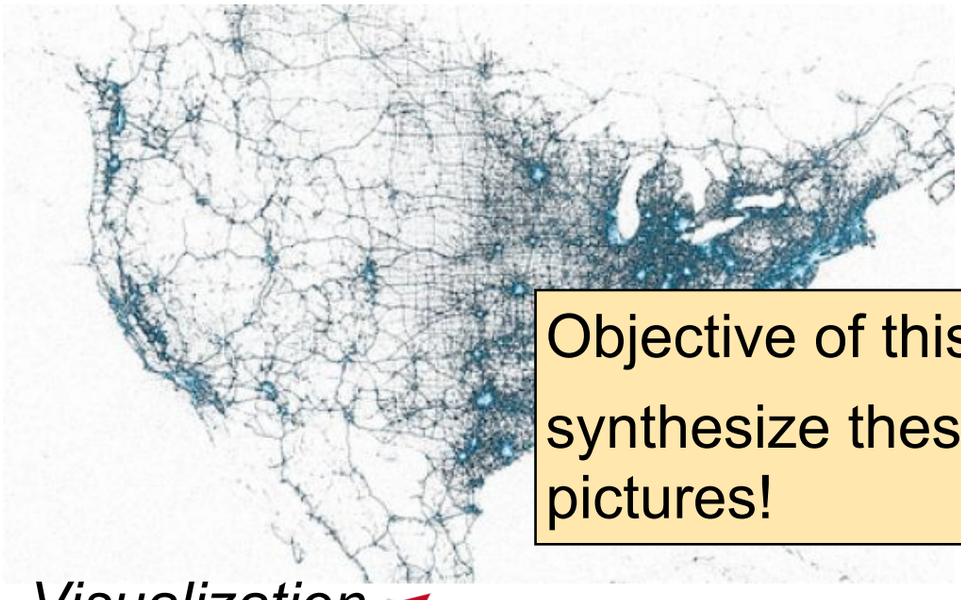
Computer Graphics is a research area that deals with the generation of synthetic images.

Games



Engineering

Movies



Objective of this course:

synthesize these pictures!

Visualization

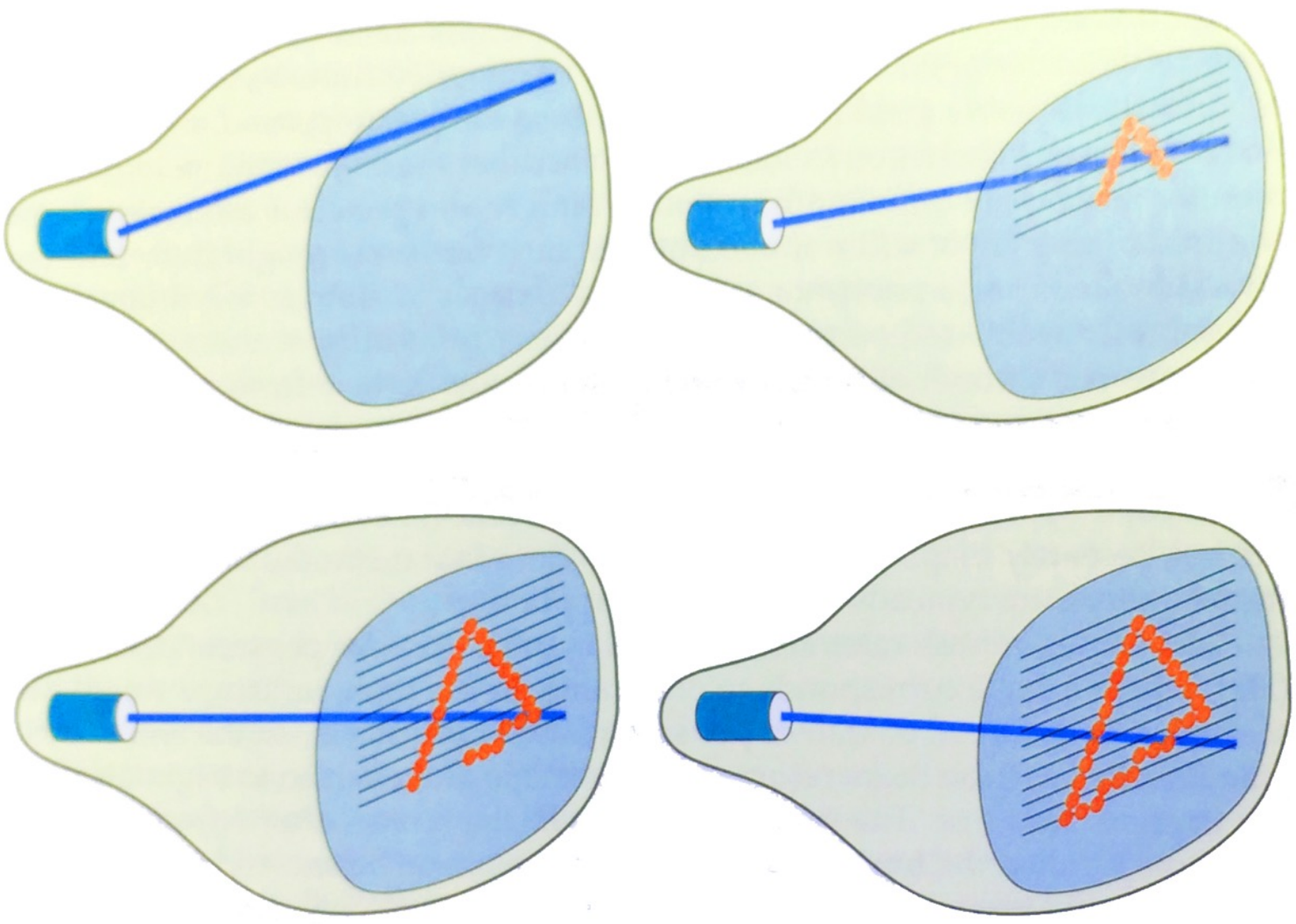
VRAC|HCI

IOWA STATE UNIVERSITY
OF SCIENCE AND TECHNOLOGY

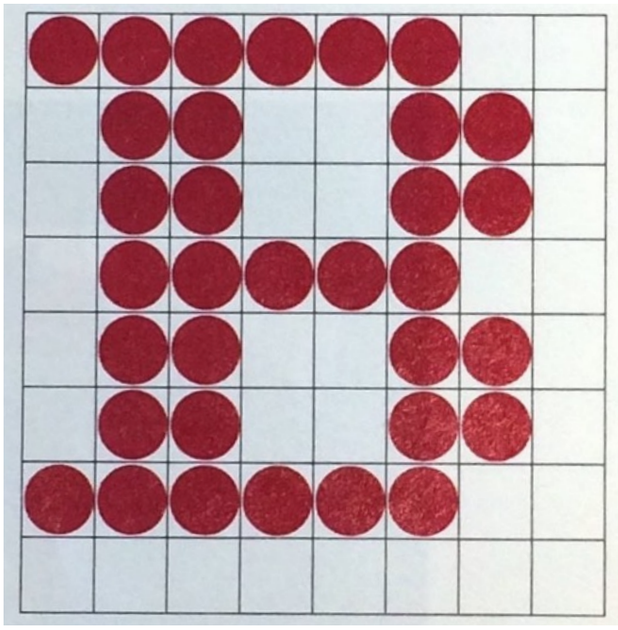
Image Generation

ARLAB

Modern displays follow a scan-line concept
(that's only have of the truth).



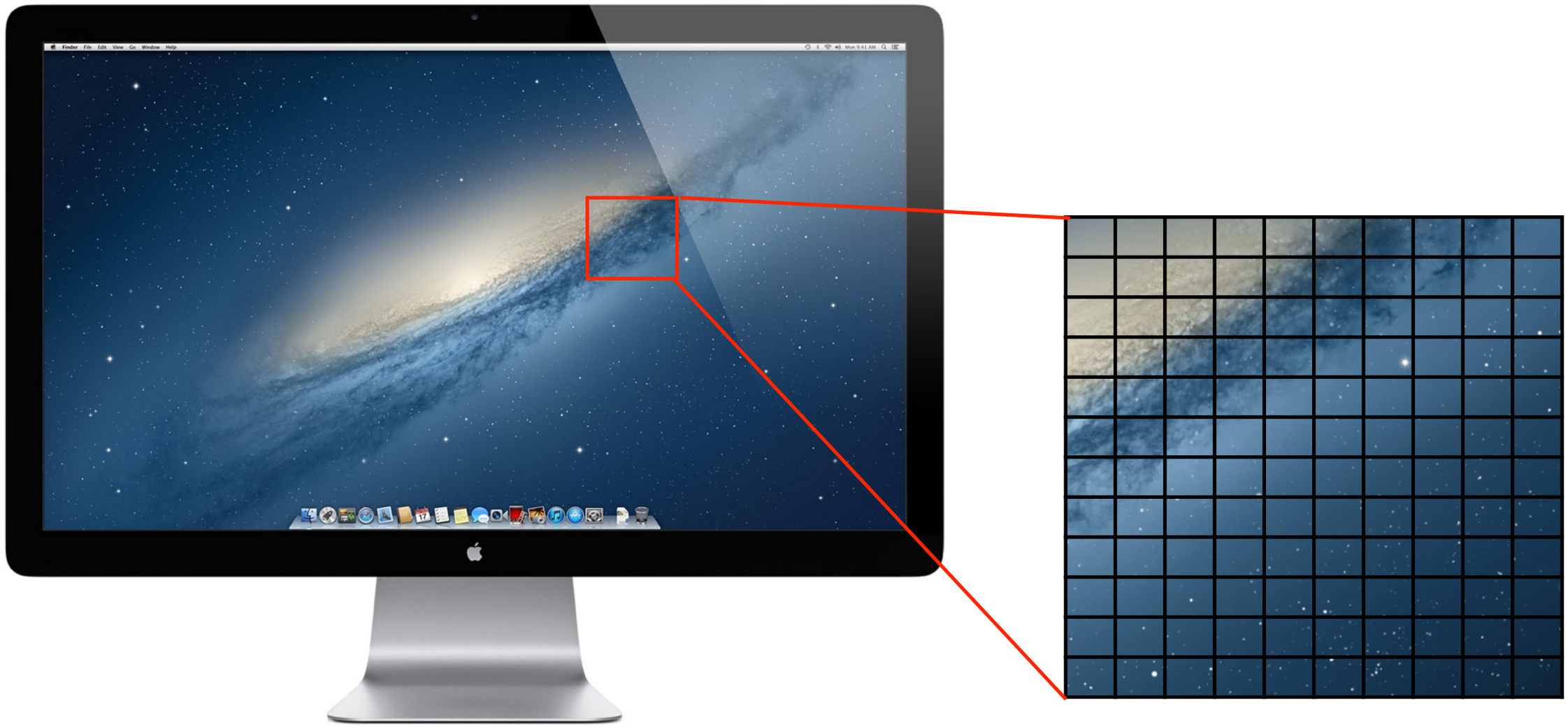
Display system



Output: dots

Image Generation

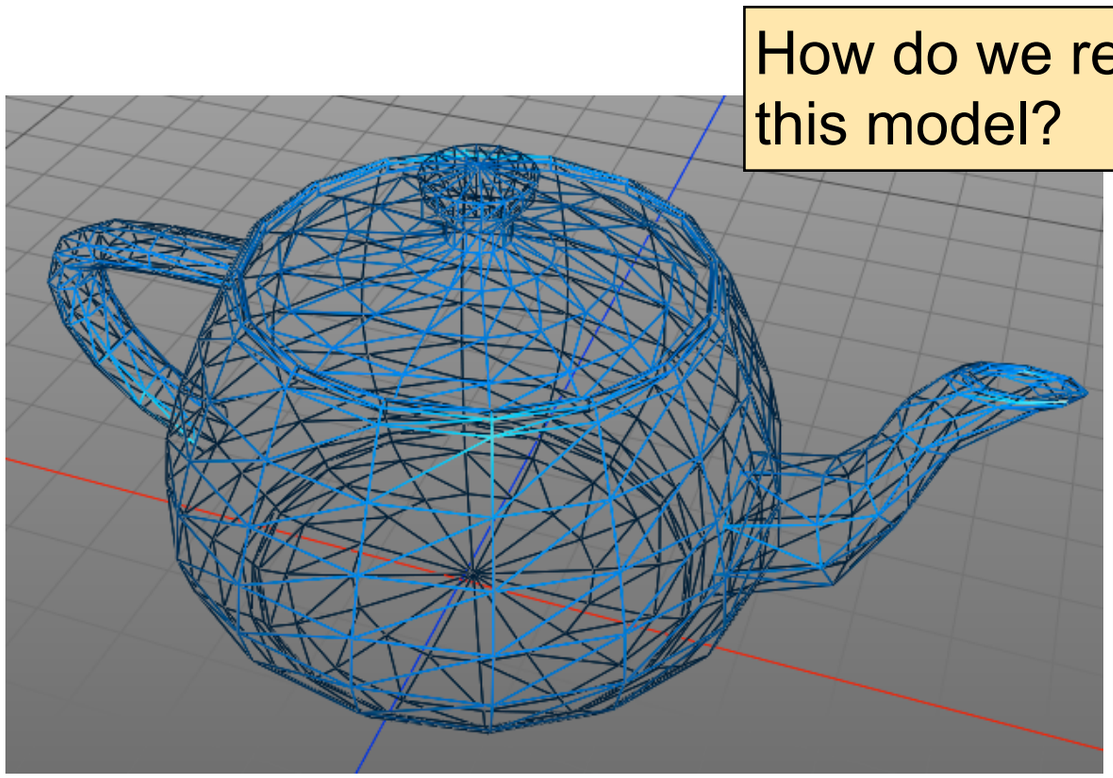
Modern displays follow a scan-line concept
(that's only have of the truth).



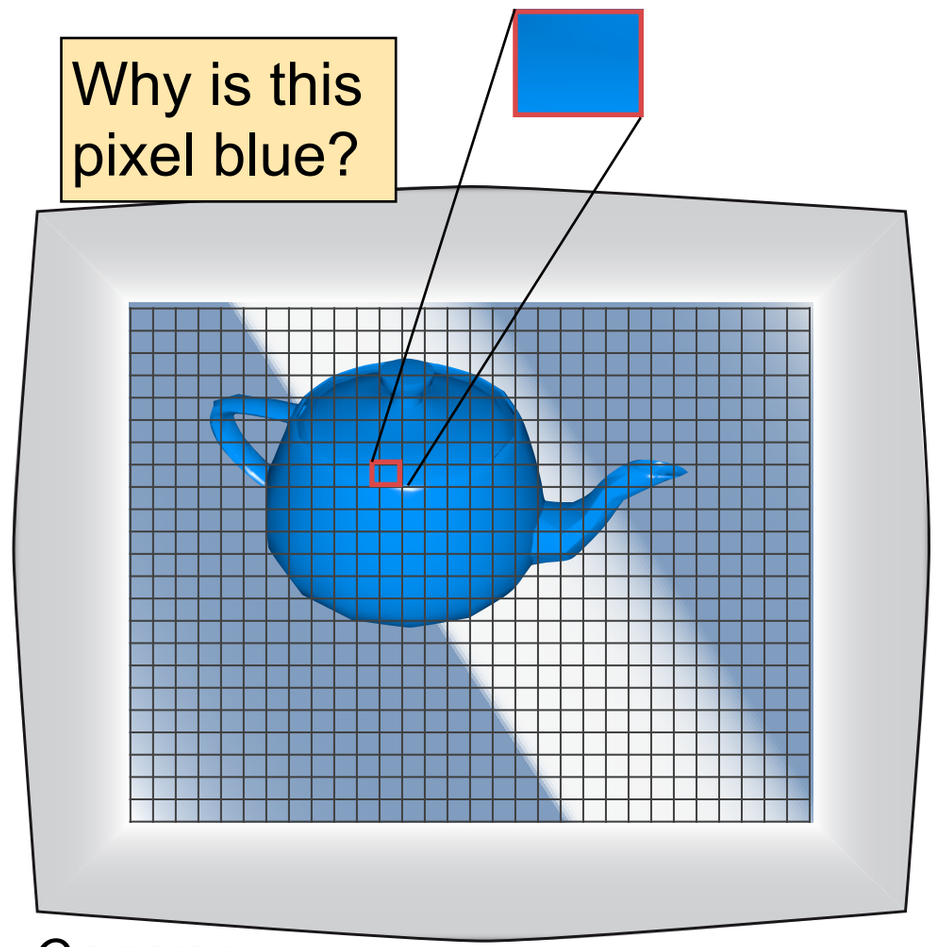
Computer Graphics is ....

ARLAB

... determine a location of an object at screen at its color.



Somewhere inside the computer



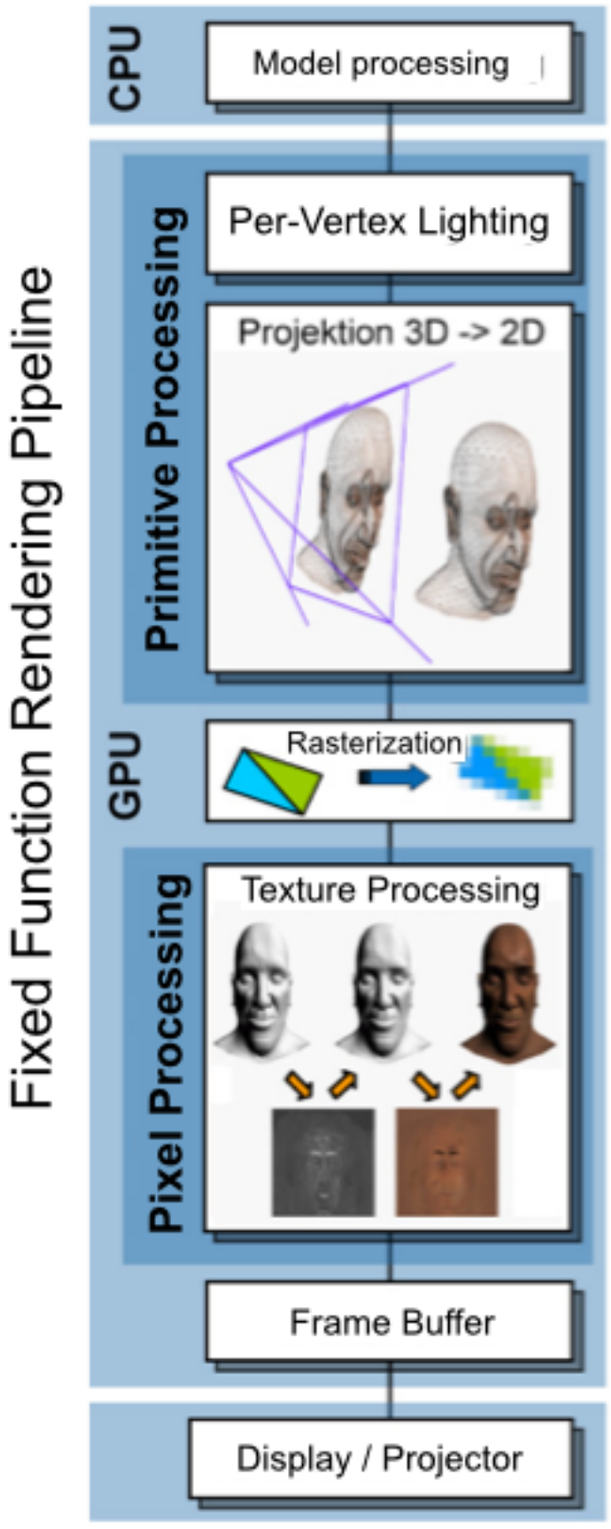
On screen

The objective of computer graphics is to determine a color for each pixel (picture element) of the display considering the object that is stored in computer memory. This is denoted by the term rendering: create an image from a model.

VRAC|HCI

IOWA STATE UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Computer Graphics



Graphics cards

Graphics cards implement a pipeline model: a set of functions are invoked in a "fixed" sequence

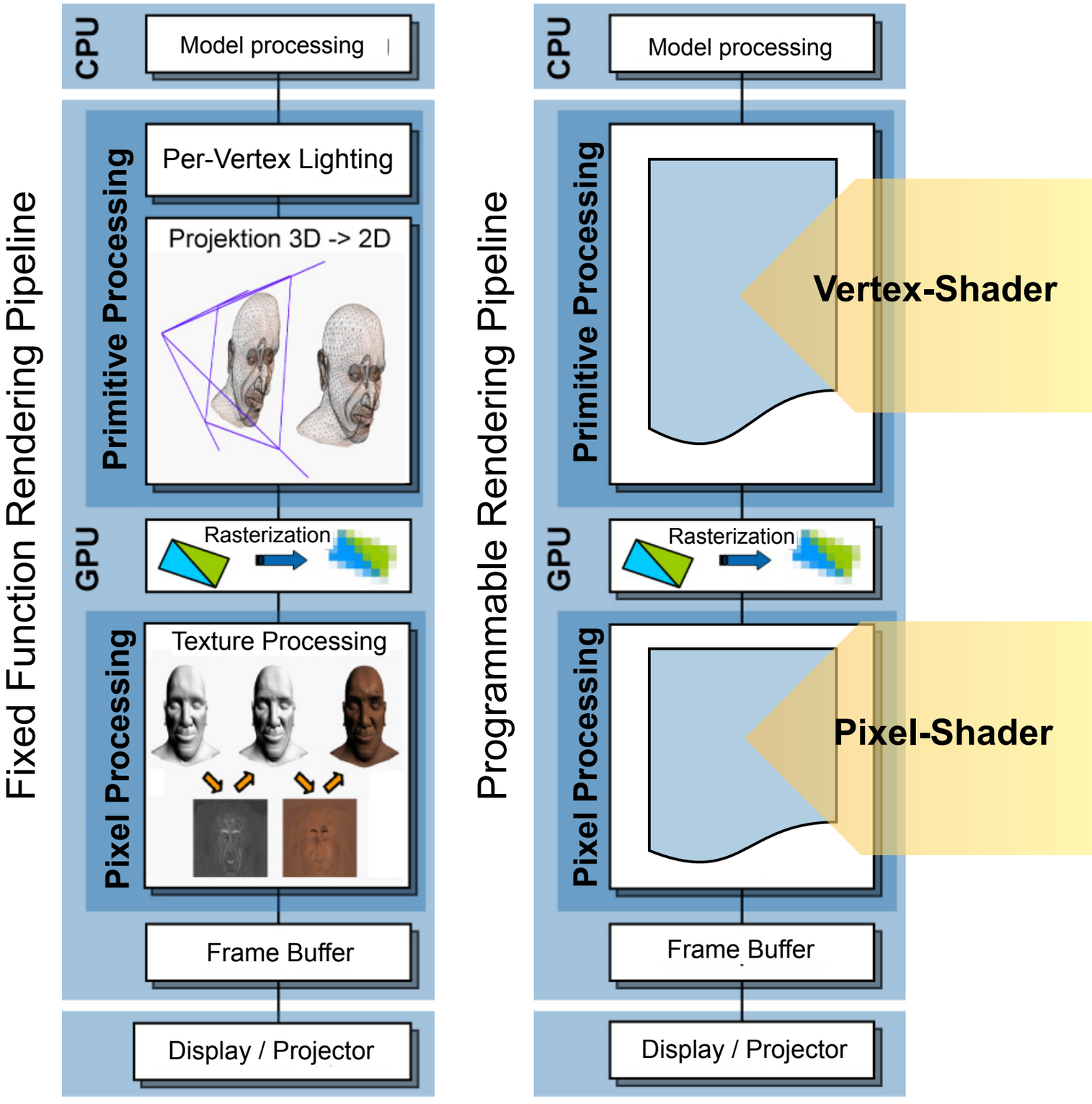
We distinguish primitive processing and pixel processing.

The course will

a) introduce the steps of this pipeline model

b) explain the API to invoke / implement the functions.

Concept of GPUs



The **fixed function rendering pipeline** implements all necessary calculations (transformation & lighting, rasterization, shading, etc.) via hardware circuits to generate 3D graphic on display.

The **programmable rendering pipeline** uses free-programmable logic processors. Thus, they can be used to implement a vast amount of visual effects which goes beyond the capabilities of the fixed function rendering pipeline. The programmer can decide on his/her own, which function need to be implement to realize a distinct effect.

The **Vertex-Shader** is used to manipulate the vertices of a 3D model and to carry out per-vertex lighting calculations.

The **Pixel (Fragment)-Shader** facilitates the manipulation of single pixels.

The **Geometry-Shader** enables the programmer to create new vertices and

Application Programming Interfaces

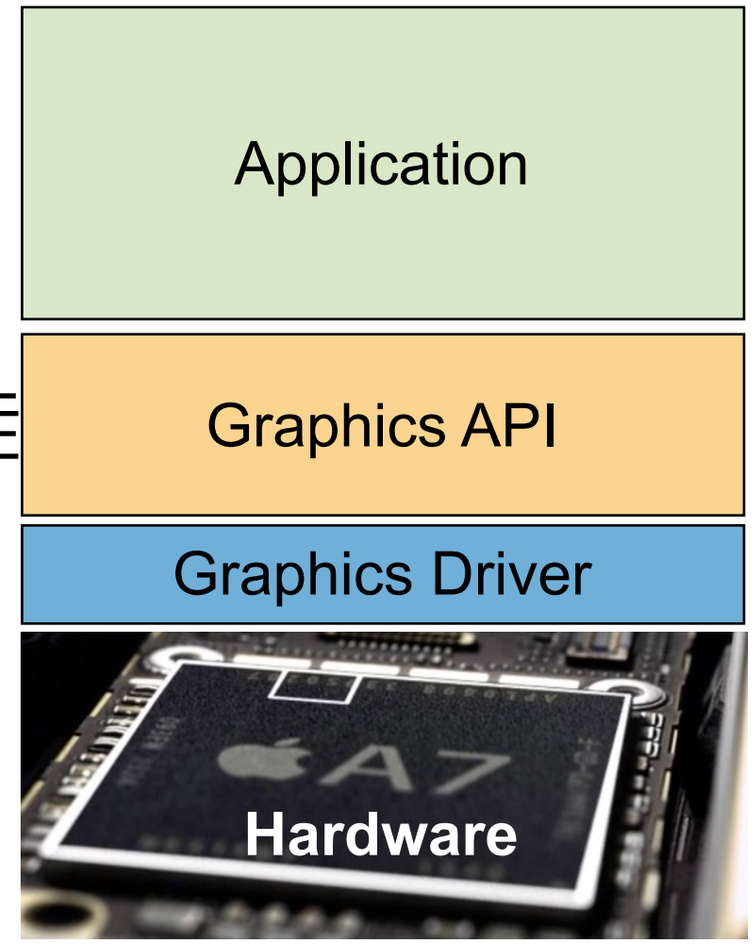
ARLAB

OpenGL®

Microsoft®
DirectX®

Metal

Vulkan™
release: later this year



The graphics API connects your application with the functions provided by the graphics card.

It allows one to program graphics functions independent from the hardware

Application Programming Interfaces

ARLAB

We can also write code that runs on the graphics hardware

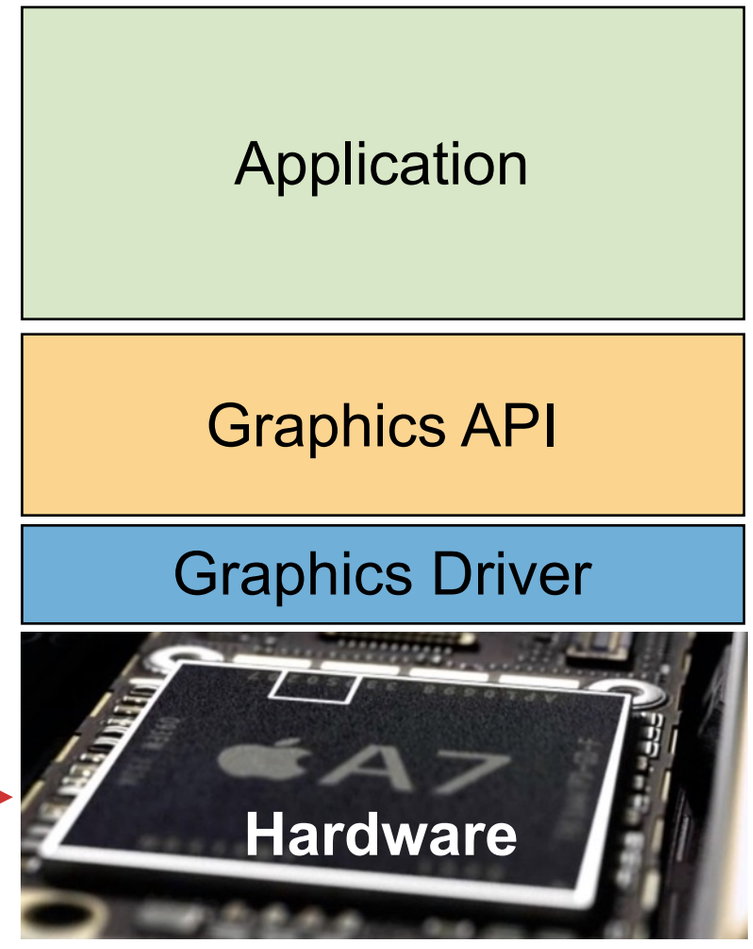
OpenGL®

Shading Language

Cg

Microsoft® DirectX®

High Level Shading Language (HLSL)



And others

OpenGL|ES™

WebGL

The graphics API connects your application with the functions provided by the graphics card.

It allows one to program graphics functions independent from the hardware

Topics

ARLAB

Content

- 3D model representation
- Transformation of objects
- 3D viewing
- Color and material
- Lighting
- Shading
- Textures
- Interaction
- Animation

Teaching Outcomes

The students will learn to create computer graphics applications and to leverage the underlying techniques to meet their application demand.

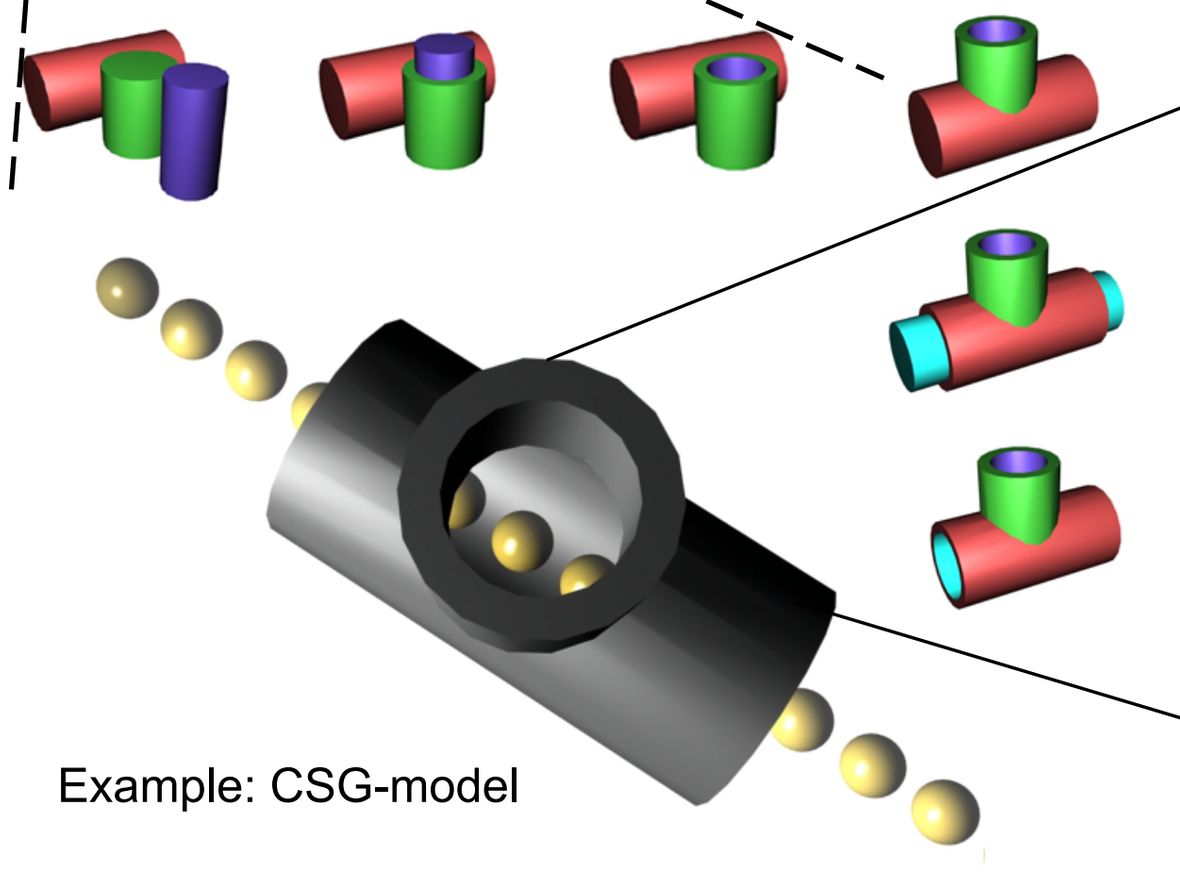
VRAC|HCI IOWA STATE UNIVERSITY OF SCIENCE AND TECHNOLOGY

Model Representation

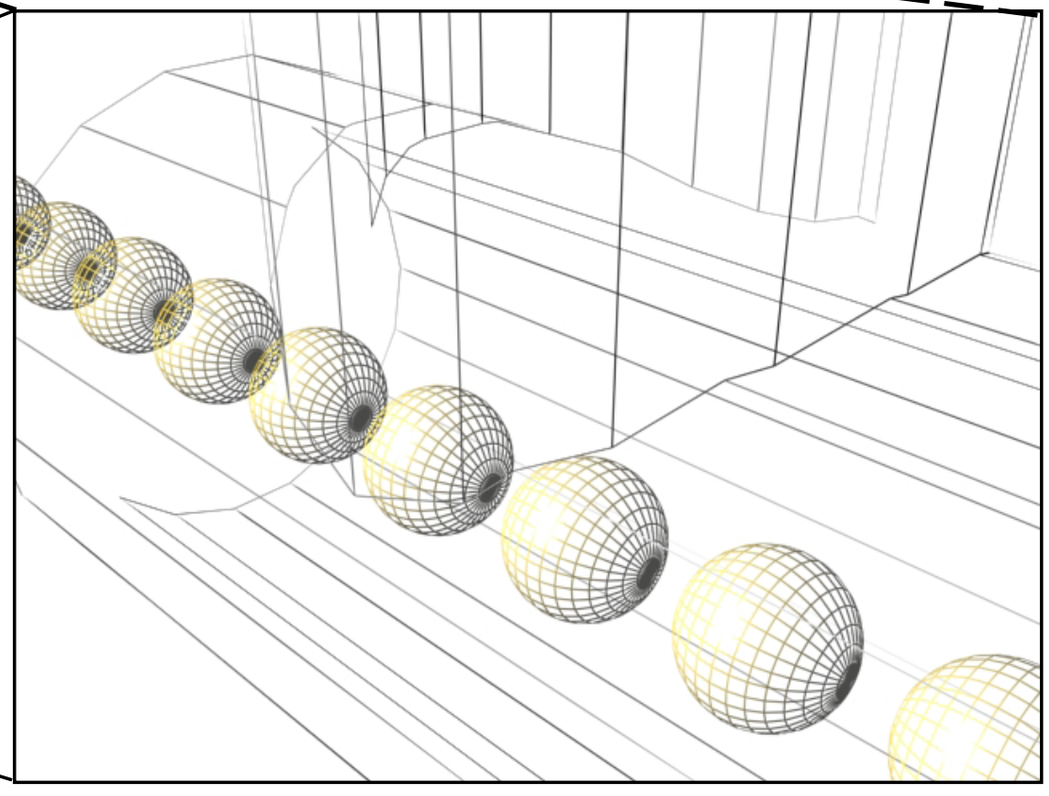
ARLAB

Model type

- Volume model
- Surface model
- Primitive model
- Parametric model
- Voxel model



Example: CSG-model



Wireframe rendering of a polygon-model

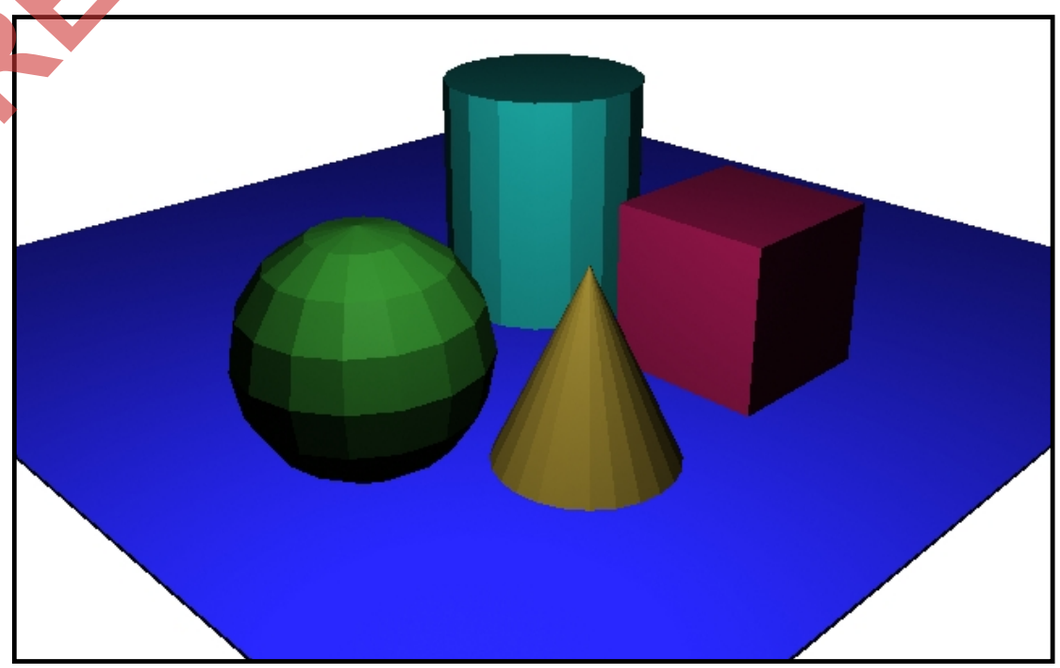
CSG: Constructive Solid Geometry

VRAC|HCI

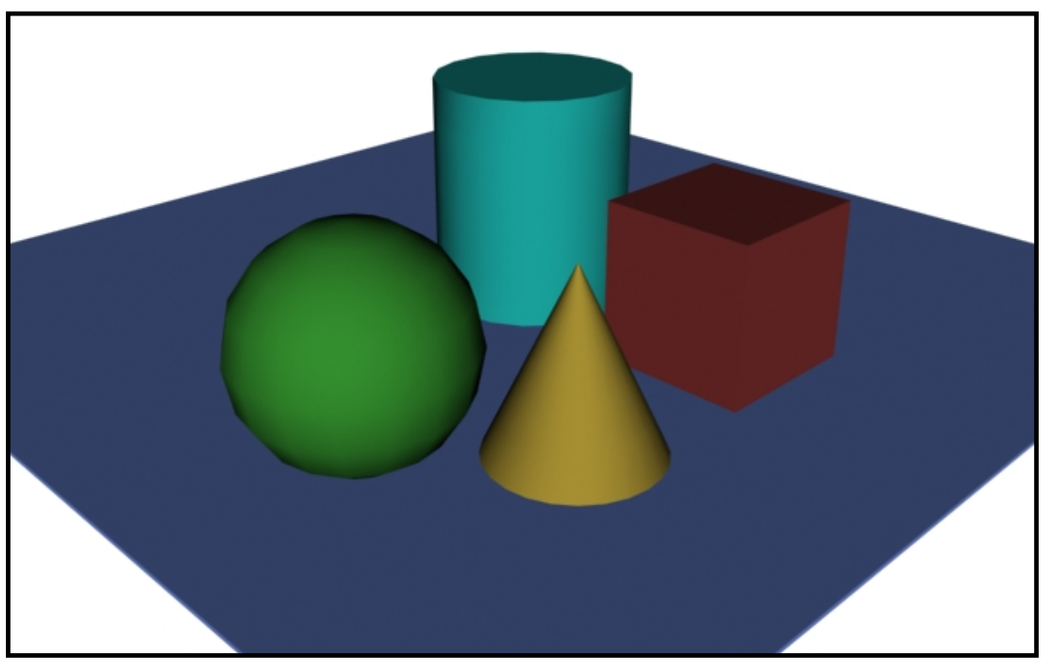
IOWA STATE UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Shading

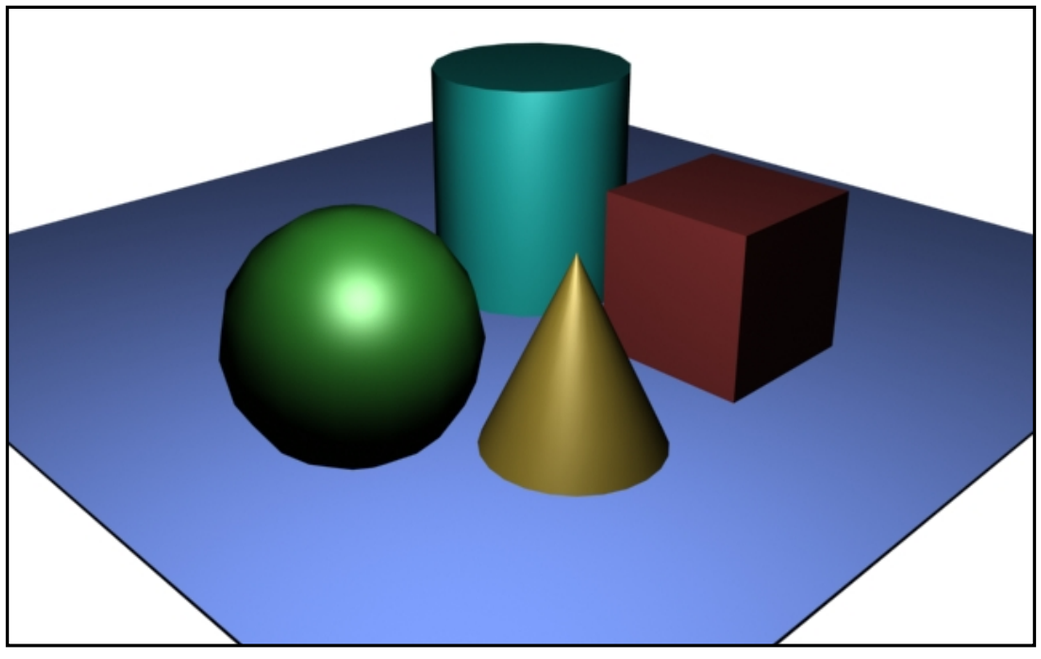
ARLAB



Flat Shading



Gouraud Shading

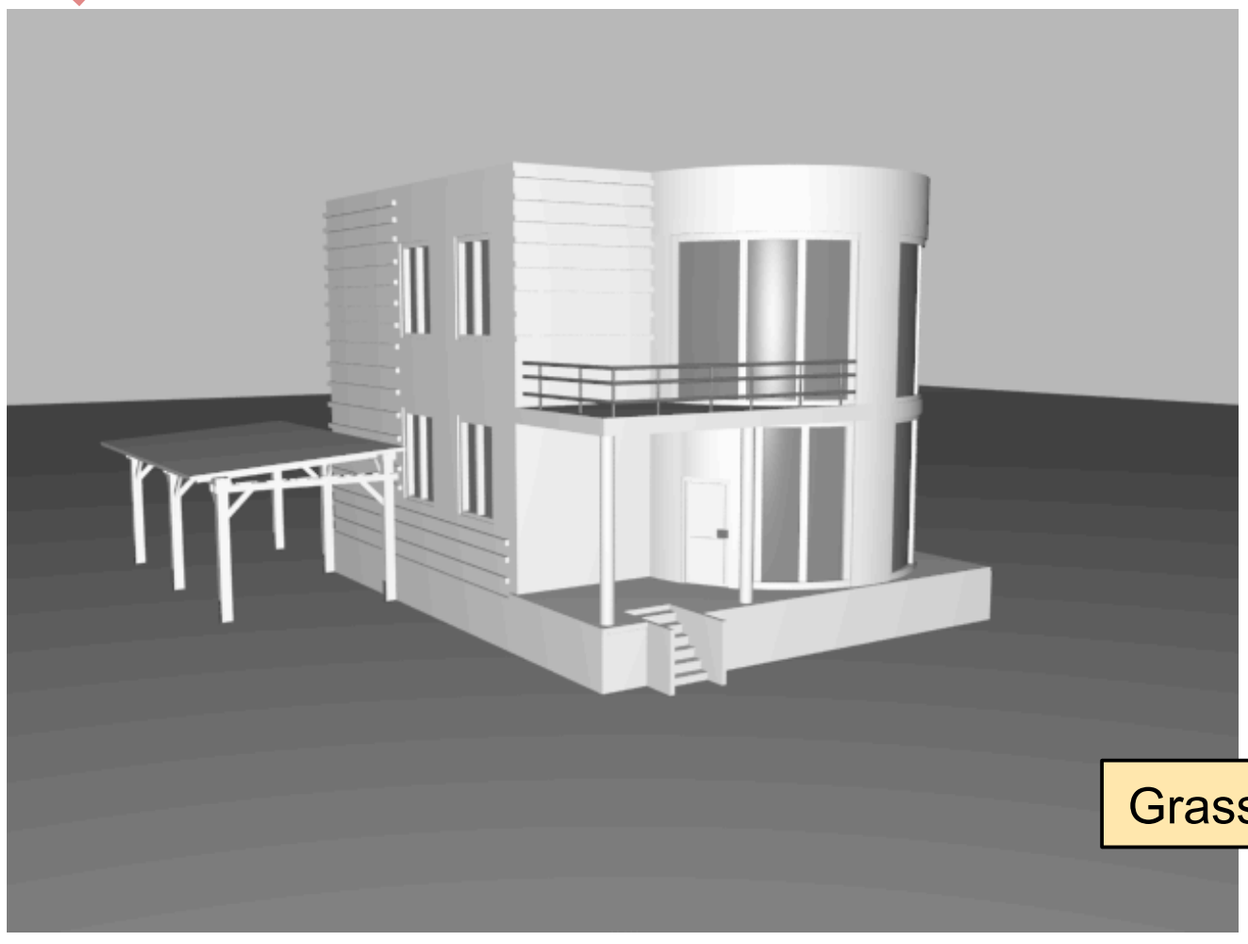


Phong Shading

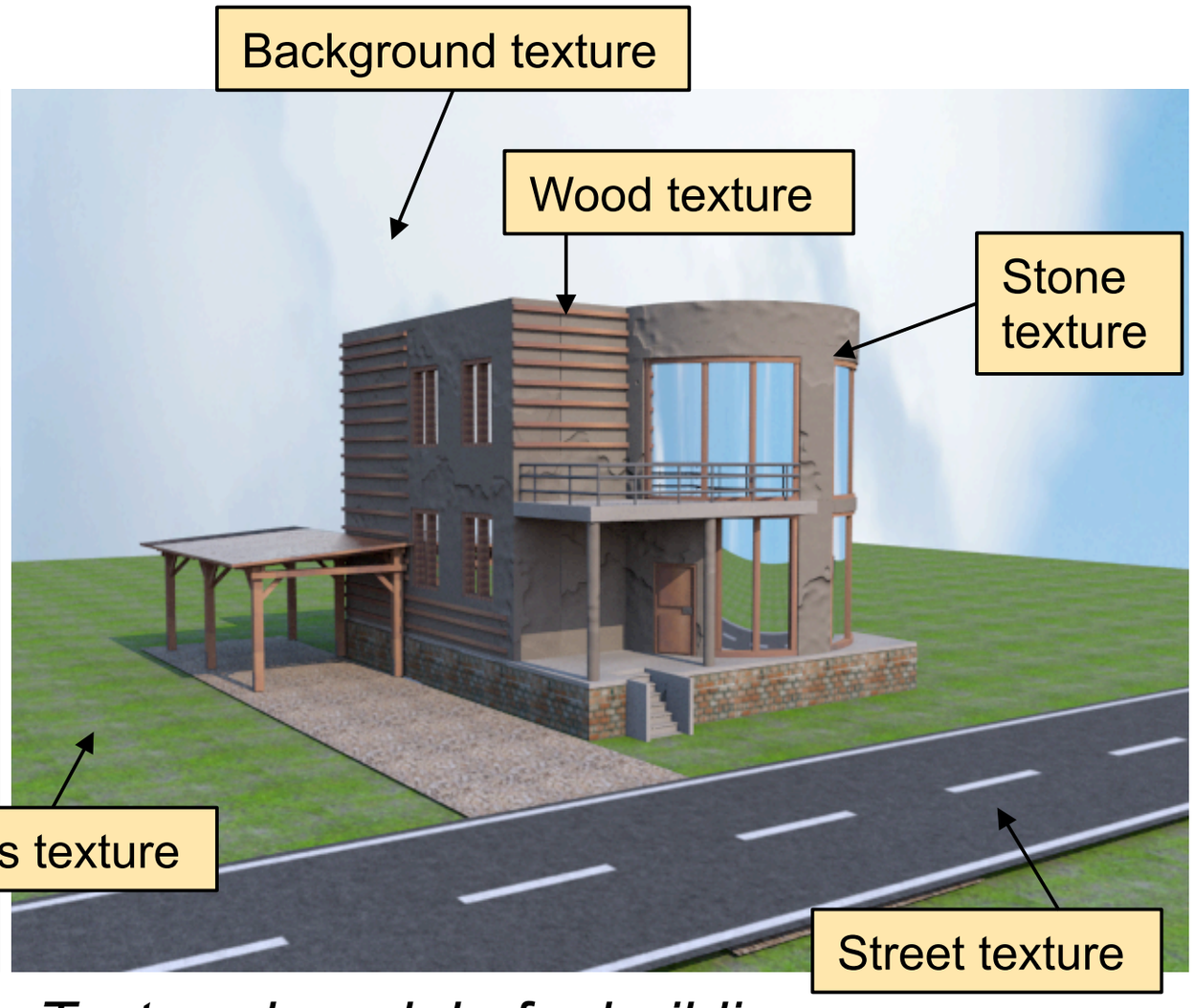
Shading is the process that fills the primitives (e.g., polygons, triangles, quads etc.) with color. The task is to determine a color value for each pixel on screen.

Texture Mapping

ARLAB



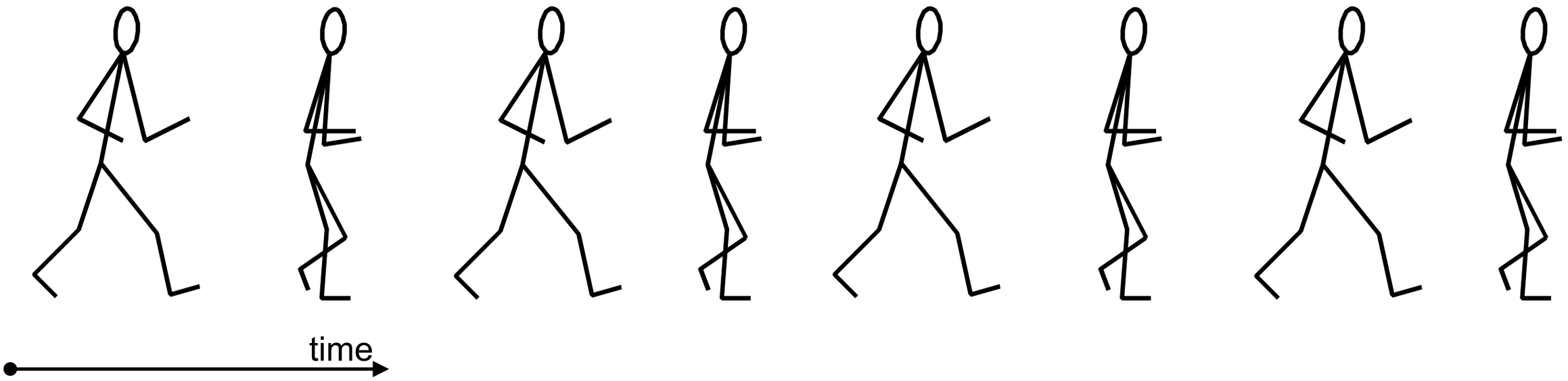
Plain 3D model of a building

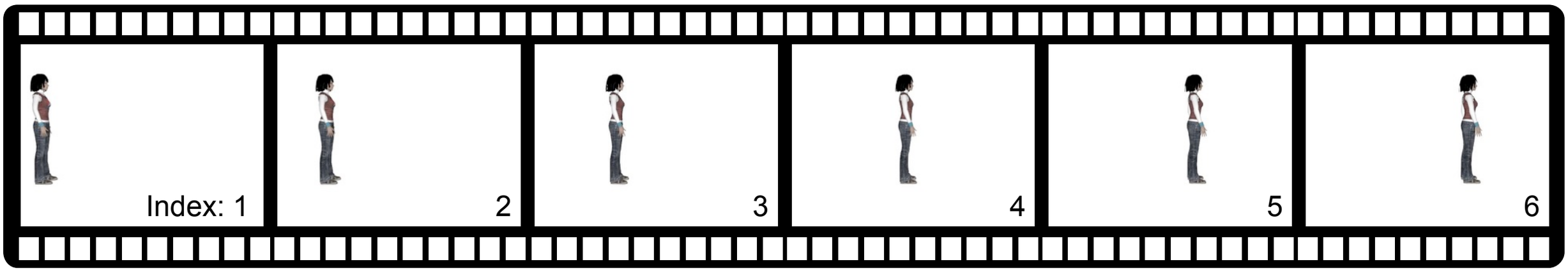


Textured model of a building

What is interactive or real time computer graphics?

Real Time Computer Graphics





A sequence of images, generated in display refresh rates to simulate smooth transition between single image.

ARLAB

We will do things!!!!

VRAC|HCI

IOWA STATE UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Software Tools

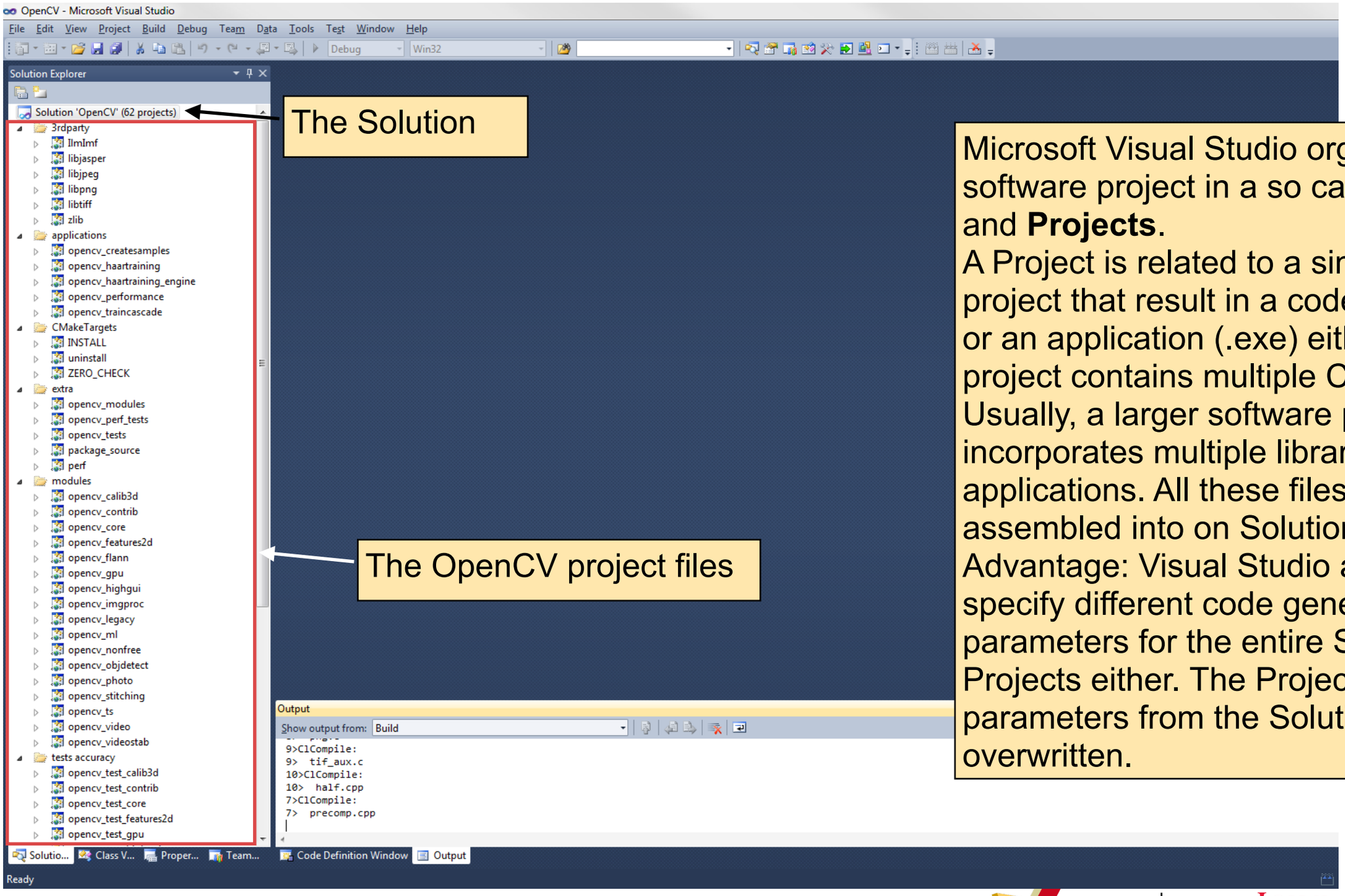
Software Tools

- Microsoft Visual Studio or XCode (Mac OS X)
- Windows: Microsoft Windows SDK
- CMake
- Git repository (GitHub, Bitbucket)
- SourceTree or GitHub
- OpenGL
  - GLEW
  - GLM
- GLSL

Microsoft Visual Studio (Professional / Express)

ARLAB

MS Visual Studio organization



Microsoft Visual Studio organize a software project in a so called **Solution** and **Projects**.
A Project is related to a single code project that result in a code library (.lib) or an application (.exe) either. Each project contains multiple C/C++ files. Usually, a larger software project incorporates multiple library files and applications. All these files are assembled into on Solution.
Advantage: Visual Studio allows to specify different code generation parameters for the entire Solution or the Projects either. The Projects inherit the parameters from the Solution if not overwritten.

Microsoft Visual Studio main view

Visual Studio

ARLAB

Visual Studio Express

Visual Studio Express editions provide free tools to develop applications for a specific platform, such as Windows Universal Platform applications, web sites, and Windows desktop applications.

New edition available

Visual Studio Community has all the features of Express and more, **and is still free** for individual developers, open source projects, academic research, education, and small professional teams.

Download Community 2015

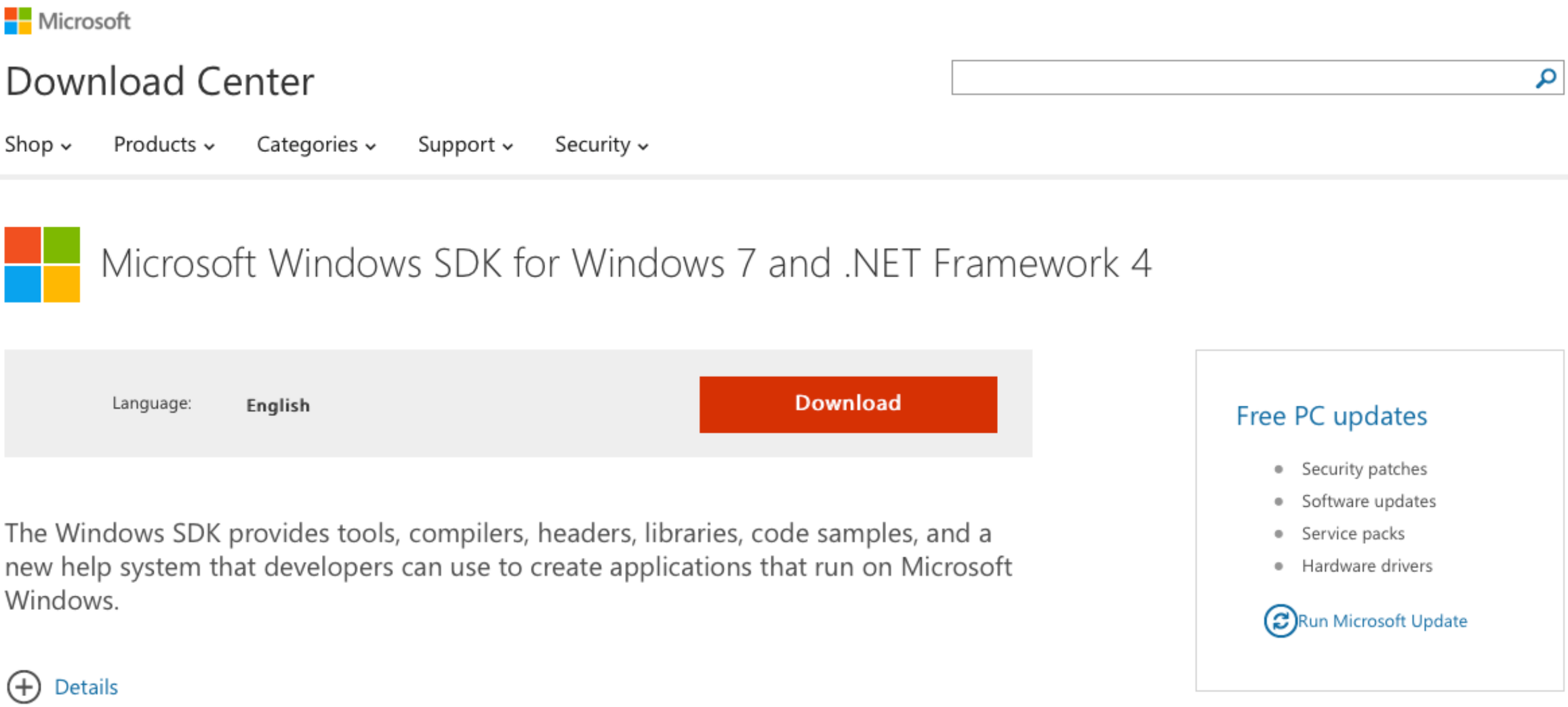
Learn more >

https://www.visualstudio.com/en-us/products/visual-studio-express-vs.aspx

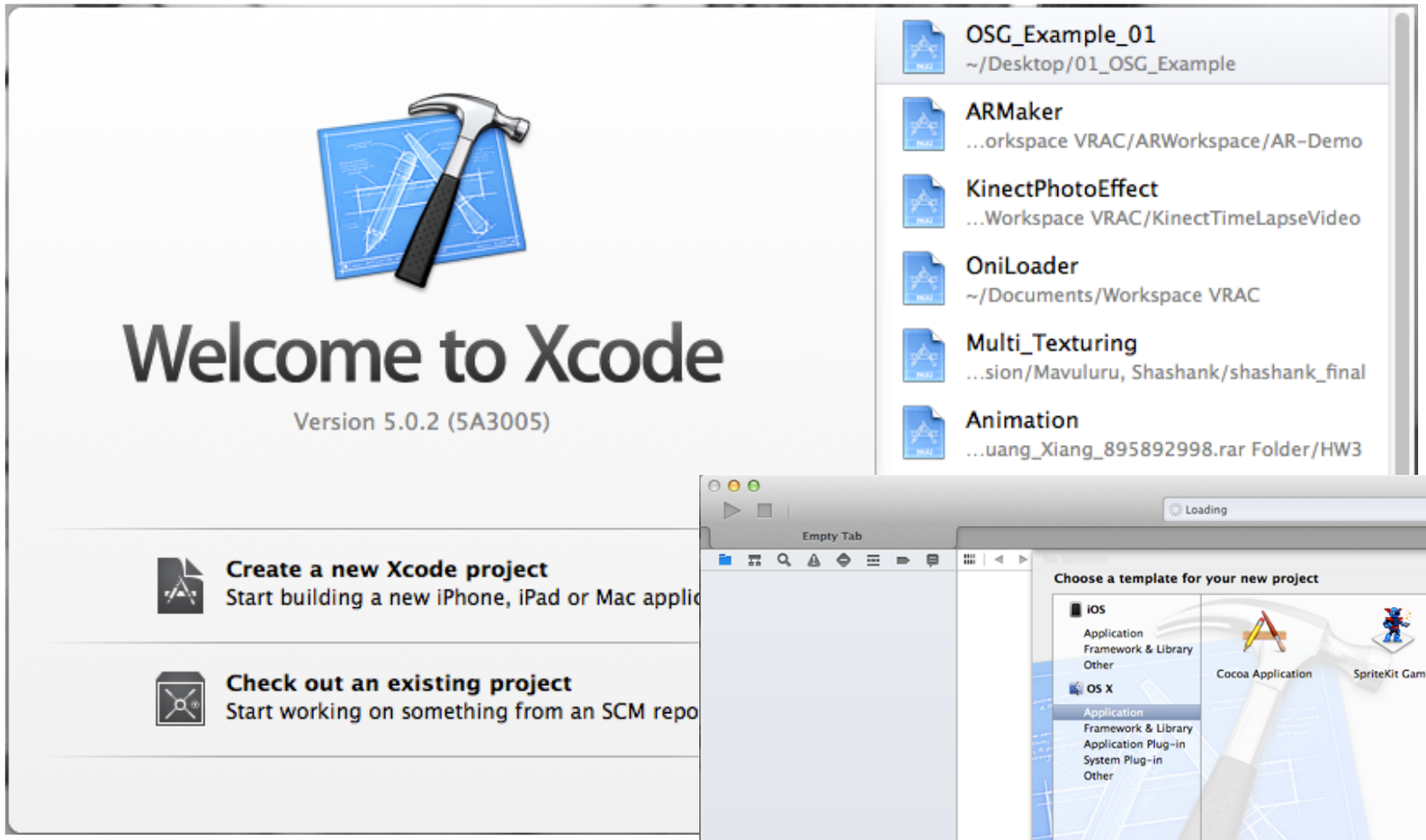
Microsoft Windows SDK

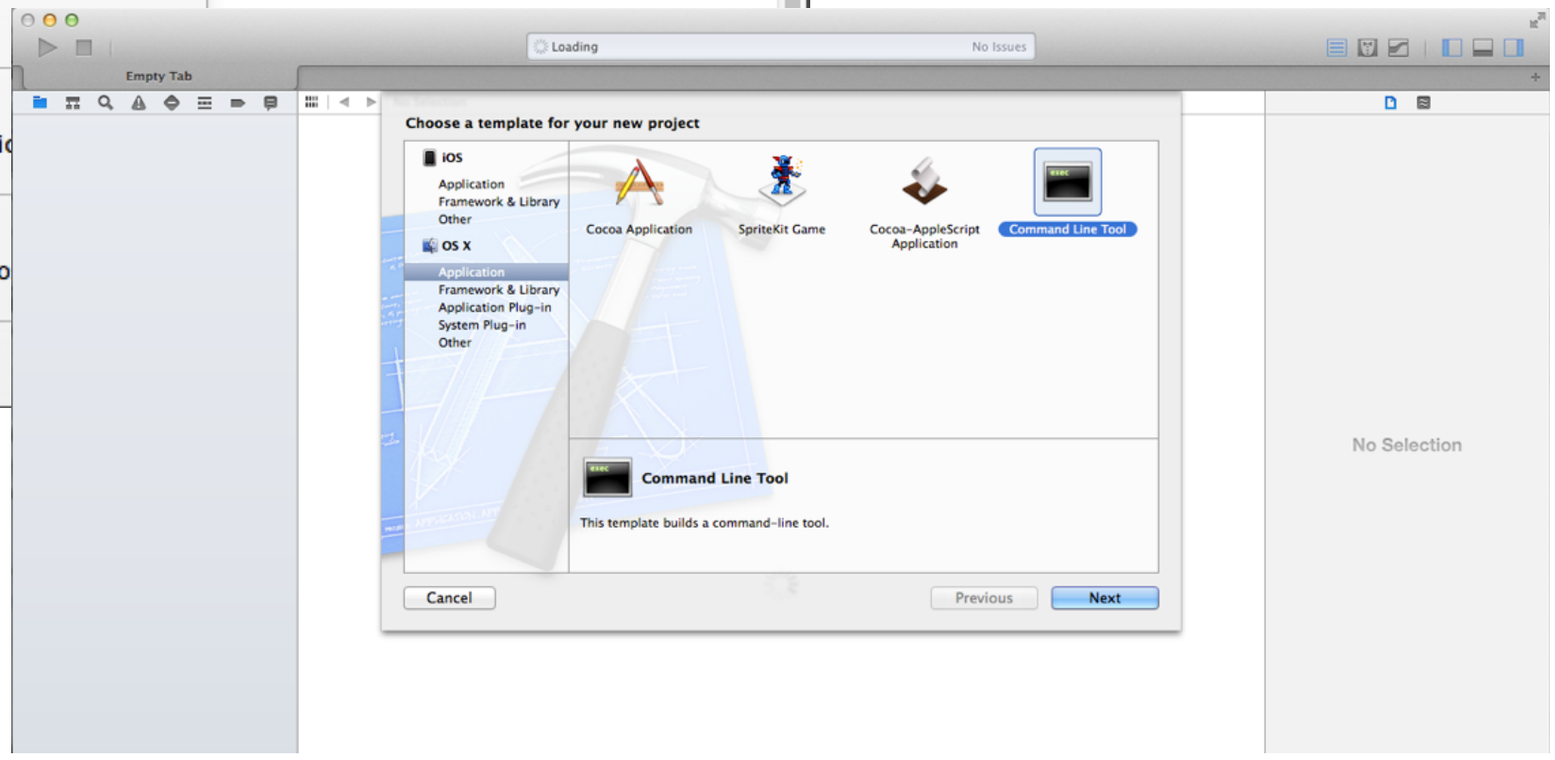
The Windows SDK provides and update for Windows that allows one to develop code on a computer with Windows operating system. You can download it from the MS download center.

Important: you need to download the correct version for your system. The download center helps!



XCode on Mac





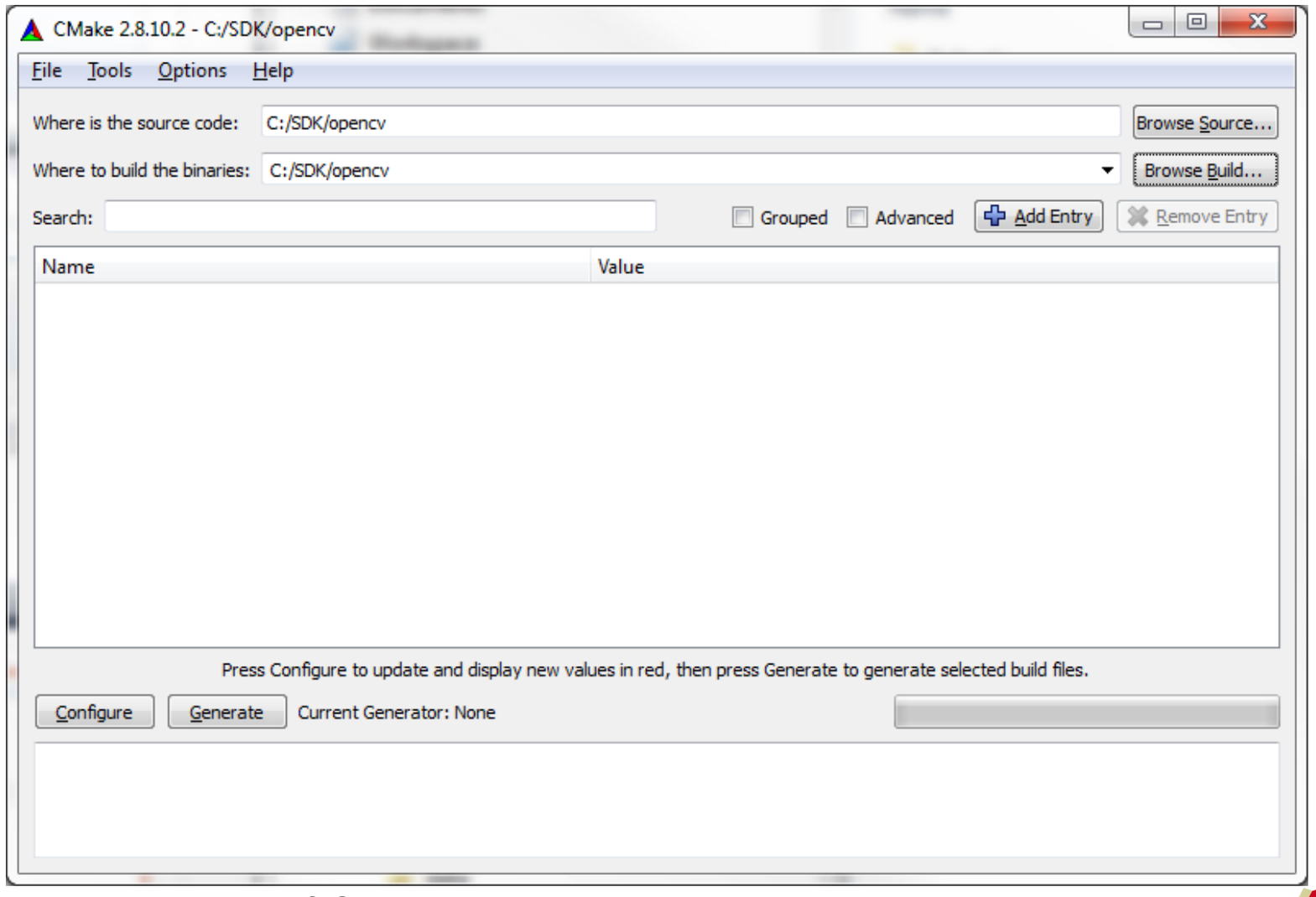
Download it from the Apple Store

XCode main window

VRAC|HCI

IOWA STATE UNIVERSITY
OF SCIENCE AND TECHNOLOGY

CMake is a cross-platform, open-source build system, designed to build, test and package software. CMake is used to control the software compilation process using simple platform and compiler independent configuration files. CMake generates native makefiles and workspaces that can be used in the compiler environment of your choice.



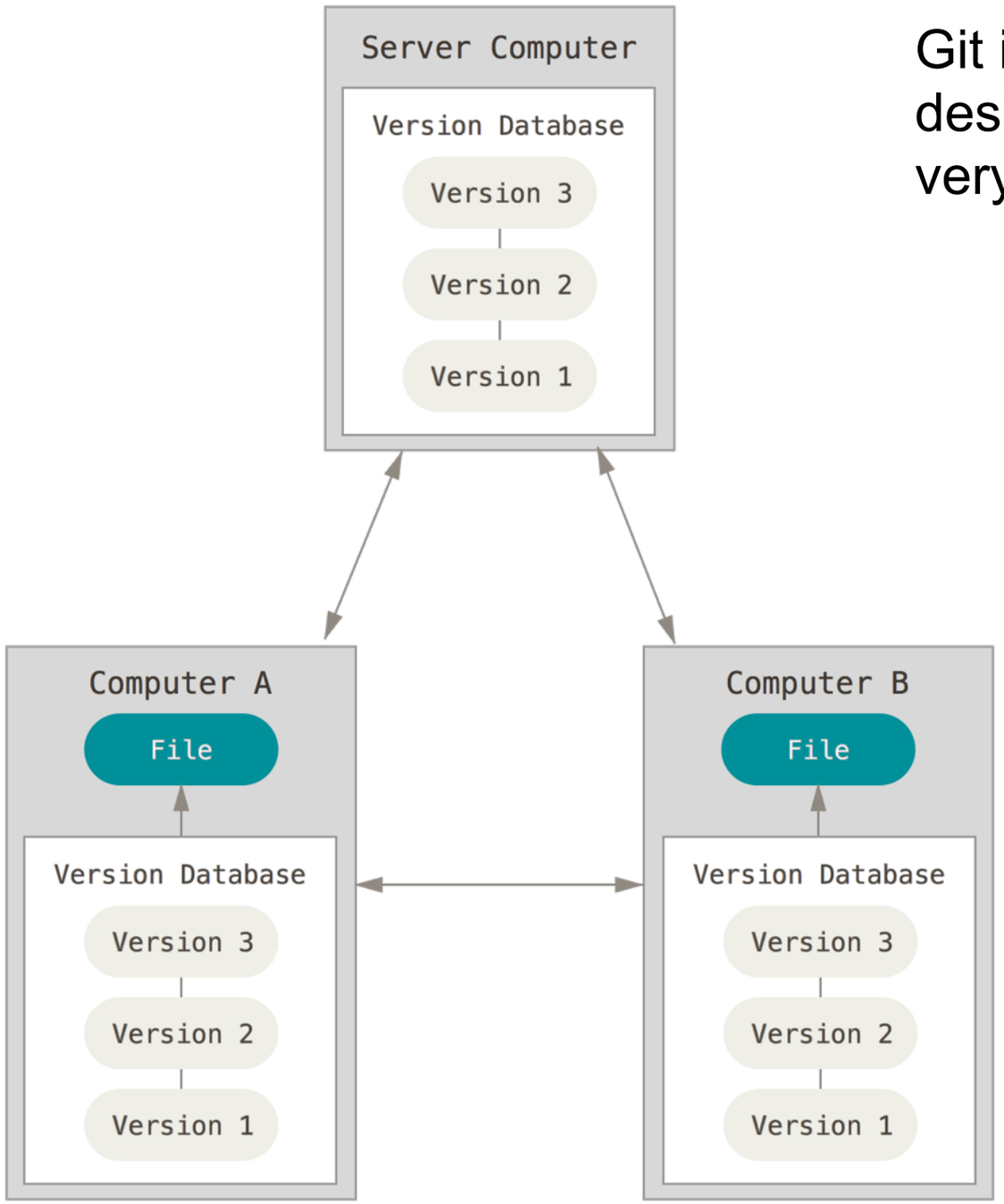
Main Window of CMake

Download the latest version!

http://www.cmake.org

VRAC|HCI

IOWA STATE UNIVERSITY
OF SCIENCE AND TECHNOLOGY

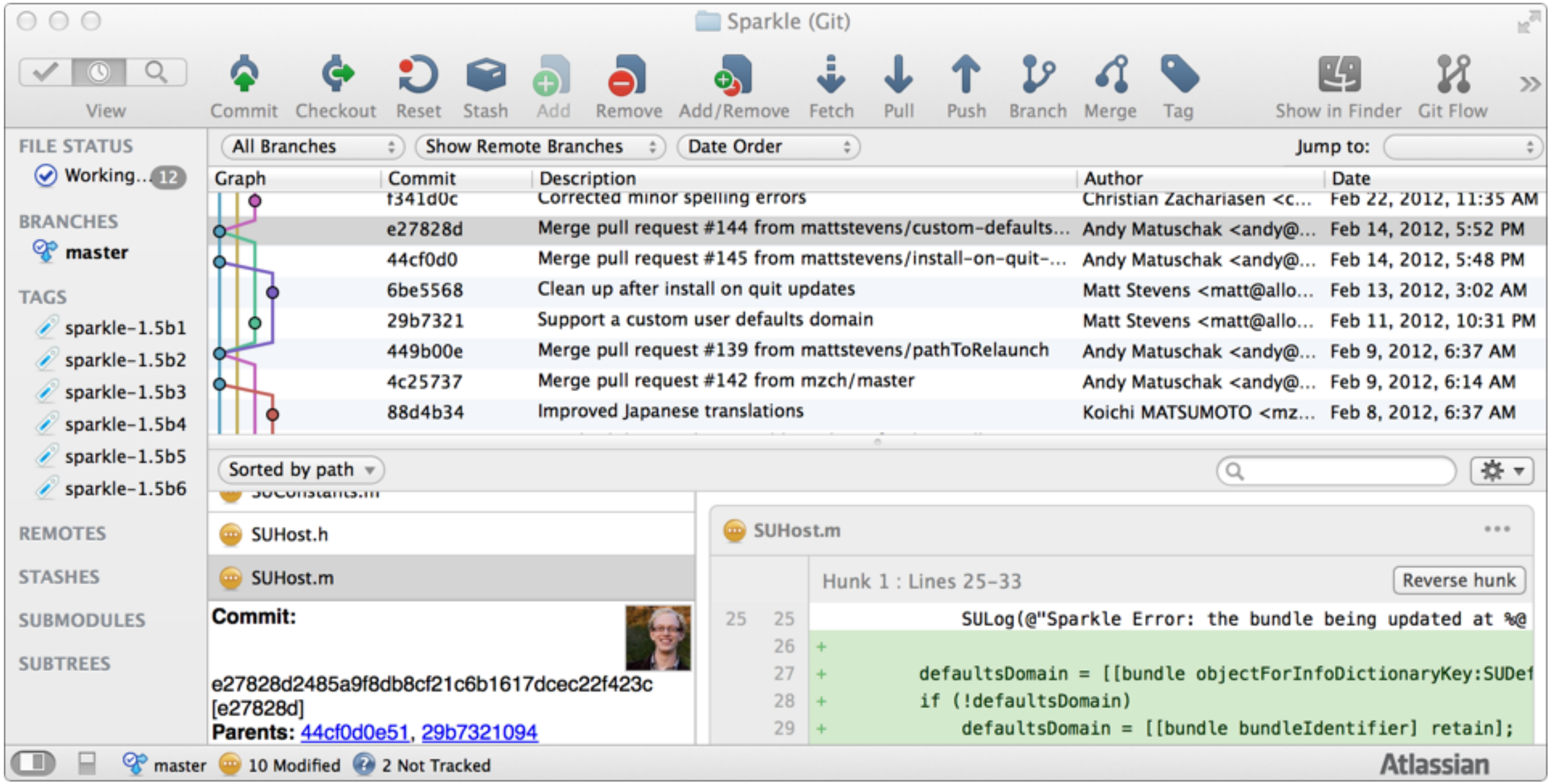


Git is a distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

https://git-scm.com

SourceTree

SourceTree is a graphical git client for Mac and Windows.



SourceTree main window

https://www.sourcetreeapp.com

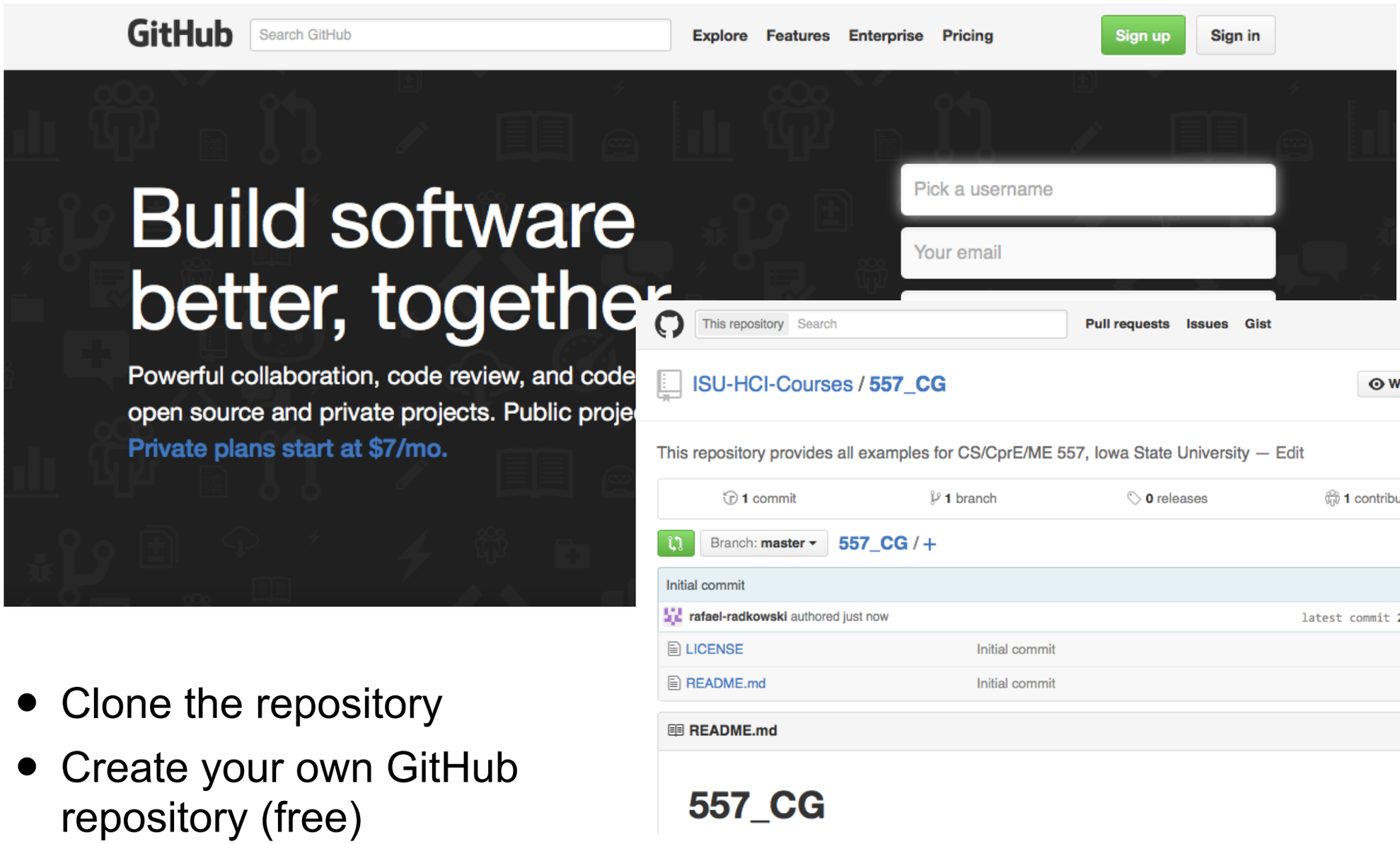
VRAC|HCI

IOWA STATE UNIVERSITY
OF SCIENCE AND TECHNOLOGY

GitHub

All examples for this class can be cloned from a git repository hosted by GitHub

https://github.com/ISU-HCI-Courses/557_CG



- Clone the repository
- Create your own GitHub repository (free)

VRAC|HCI

IOWA STATE UNIVERSITY
OF SCIENCE AND TECHNOLOGY

OpenGL (Graphics Library) is a cross-language application programming interface (API) **specification** for rendering of 2D and 3D computer graphics. Every API (or function call) interacts with the Graphics Processing Unit (GPU) to achieve rendering.

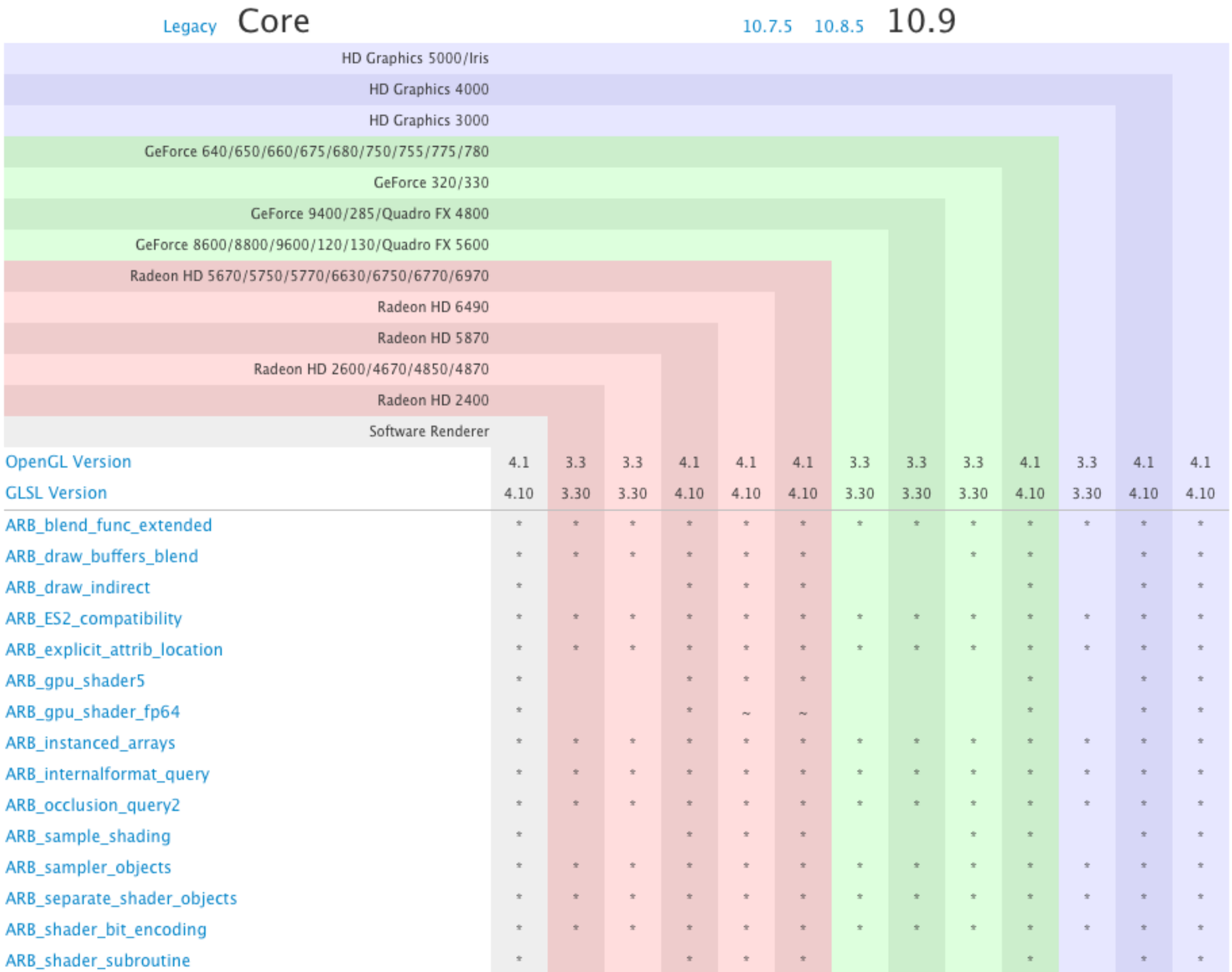
OpenGL was developed by Silicon Graphics Inc. The first version was released in 1992. Today, it is managed by the non-profit consortium Khronos Group.

Latest release OpenGL V4.5, Release on August 11, 2014

- About 150 distinct commands to specify the objects and operations needed to produce interactive three-dimensional applications.
- Geometric primitives like points, lines, and polygons.
- Implemented in C\C++ (Wrappers for Java and other languages are also available).
- No window control system
- OpenGL installation depends on the vendor
  - Windows (1.1, frozen and not supported by MS)
  - Mac OS X (4.1, OS X Maverick)

https://www.opengl.org

OpenGL Functionality



OpenGL Version	4.1	3.3	3.3	4.1	4.1	4.1	3.3	3.3	3.3	4.1	3.3	4.1	4.1
GLSL Version	4.10	3.30	3.30	4.10	4.10	4.10	3.30	3.30	3.30	4.10	3.30	4.10	4.10
ARB_blend_func_extended	*	*	*	*	*	*	*	*	*	*	*	*	*
ARB_draw_buffers_blend	*	*	*	*	*	*			*	*		*	*
ARB_draw_indirect	*			*	*	*				*		*	*
ARB_ES2_compatibility	*	*	*	*	*	*	*	*	*	*	*	*	*
ARB_explicit_attrib_location	*	*	*	*	*	*	*	*	*	*	*	*	*
ARB_gpu_shader5	*			*	*	*				*		*	*
ARB_gpu_shader_fp64	*			*	~	~				*		*	*
ARB_instanced_arrays	*	*	*	*	*	*	*	*	*	*	*	*	*
ARB_internalformat_query	*	*	*	*	*	*	*	*	*	*	*	*	*
ARB_occlusion_query2	*	*	*	*	*	*	*	*	*	*	*	*	*
ARB_sample_shading	*			*	*	*			*	*		*	*
ARB_sampler_objects	*	*	*	*	*	*	*	*	*	*	*	*	*
ARB_separate_shader_objects	*	*	*	*	*	*	*	*	*	*	*	*	*
ARB_shader_bit_encoding	*	*	*	*	*	*	*	*	*	*	*	*	*
ARB_shader_subroutine	*			*	*	*				*		*	*

GLEW:
The OpenGL Extension Wrangler Library

ARLAB

http://glew.sourceforge.net

GLEW is an OpenGL implementation which provides access to the latest OpenGL driver functions

Latest Release: 1.13.0

Download
Usage
Building
Installation
Source Generation
Credits & Copyright
Change Log
GitHub
Project Page
Bug Tracker

Last Update: 08-10-15

OpenGL
GitHub
SOURCEFORGE.NET

The OpenGL Extension Wrangler Library

The OpenGL Extension Wrangler Library (GLEW) is a cross-platform open-source C/C++ extension loading library. GLEW provides efficient run-time mechanisms for determining which OpenGL extensions are supported on the target platform. OpenGL core and extension functionality is exposed in a single header file. GLEW has been tested on a variety of operating systems, including Windows, Linux, Mac OS X, FreeBSD, Irix, and Solaris.

Downloads

GLEW is distributed as source and precompiled binaries.
The latest release is 1.13.0[08-10-15]:

Source	ZIP | TGZ
Binaries	Windows 32-bit and 64-bit

An up-to-date copy is also available using git:

- github
  `git clone https://github.com/nigels-com/glew.git glew`
- Sourceforge
  `git clone git://git.code.sf.net/p/glew/code glew`

Unsupported snapshots are also available:

- glew-20150805.tgz
- glew-20150124.tgz

Supported Extensions

The latest release contains support for OpenGL 4.5 and the following extensions:

- OpenGL extensions

VRAC|HCI IOWA STATE UNIVERSITY OF SCIENCE AND TECHNOLOGY

http://glm.g-truc.net/0.9.4/

OpenGL Mathematics provides - the name states it - mathematical support for graphics.

GLM GLI
0.9.7 0.9.6 0.9.5 0.9.4 0.9.3 0.9.2 0.9.1 0.9.0

OpenGL Mathematics

GLSL + Optional features = OpenGL Mathematics (GLM)
A C++ mathematics library for graphics programming

Download GLM 0.9.4.6
20/09/2013

Manual
API Documentation
Code Samples
Code Snapshot

Release Notes
Downloads
Repository
Feedback

GLSL Specification
GLSL Man Pages

OpenGL Mathematics (GLM) is a header only C++ mathematics library for graphics software based on the OpenGL Shading Language (GLSL) specification and released under the MIT license.

This library provides classes and functions designed and implemented following as strictly as possible the GLSL conventions and functionalities so that when a programmer knows GLSL, he knows GLM as well, making it really easy to use.

This project isn't limited to GLSL features. An extension system provides extended capabilities: matrix transformations, quaternions, half-based types, random number generation, procedural noise functions, etc.

GLM ensures interoperability with third party libraries, SDKs and OpenGL; replacing advantageously the deprecated matrix functions. It is a good candidate for software rendering (Raytracing / Rasterisation), image processing, physic simulations and any context that requires a simple and convenient mathematics library.

It is a platform independent library with no dependence to external libraries even OpenGL. GLM is written in C++98 but can take advantage of C++11 when available. It supports the following compilers:

- Clang 2.6 and higher
- CUDA 3.0 and higher

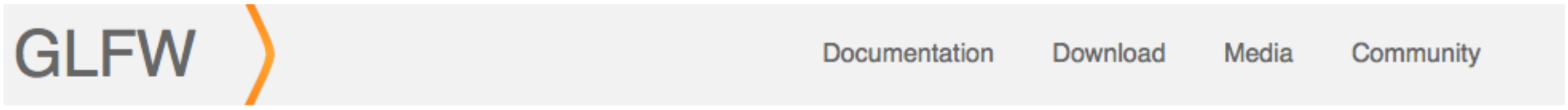
VRAC|HCI IOWA STATE UNIVERSITY OF SCIENCE AND TECHNOLOGY

GLFW

ARLAB

http://www.glfw.org/index.html

GLFW is a packages that allows us to create windows.



ToDo

Integrated Development Environment:

- Download and install Microsoft Visual Studio Express (or XCode on Mac)
- Windows: download and install the Microsoft Windows SDK.

Software

- Download CMake
- Download and install SourceTree

Software management

- Create a GitHub account
- Read the Git documentation online and learn how it works: https://git-scm.com/doc

Software Development Kits

- Download (and install) GLEW
- Download (and install) GLM
- Download (and install) GLFW

For Mac Users

ARLAB

Mac users can use homebrew to brew the Software Development Kits GLEW, GLM, and GLFW.

Follow the installation documentation.

ARLAB

Organization

VRAC|HCI

IOWA STATE UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Organization

ARLAB

Grading
The grading depends on homework results (40%), a final project (40%) and group presentations in class (20%).

Homework Projects:
The homework assignments incorporate the writing of code to solve computer graphics problems. The outcomes preparing students for the final project and can be used for it. The homework tasks must be solved individually.

Final Project
The final project is a programing exercise: students have to individually define, to plan, and to implement a computer graphics application. Every student has to propose an application topic and a software design.

Time and Location

Time: Tuesday and Thursday, 2:10 p.m. - 3:30 p.m.

Location: Howe 1242

Content

ARLAB

Web https://github.com/rafael-radkowski/HCI-557-CG

Git https://github.com/rafael-radkowski/HCI-557-CG.git

rafael-radkowski / **HCI-557-CG**

Watch 0 Star 0 Fork 0

Code repository for HCI / CS / ME / CpE 557. This repository provides example code and some documentation.

4 commits **1** branch **0** releases **1** contributor

Branch: **master** HCI-557-CG / +

Another additional ignore-file.

rafael-radkowski authored 3 hours ago latest commit `ac68fecb39`

01_HelloOpenGL	Initial push of the GLTest application and the packages required for 557	3 hours ago
SDK	Added ignore files	3 hours ago
.gitignore	Another additional ignore-file.	3 hours ago
LICENSE	Initial commit	11 days ago
README.md	Initial commit	11 days ago

README.md

HCI-557-CG

Code

Issues 0

Pull requests 0

Pulse

Graphs

HTTPS clone URL

https://github.com/rafael

You can clone with HTTPS or Subversion.

Clone in Desktop

Download ZIP

ERSITY

Course Strcture

ARLAB

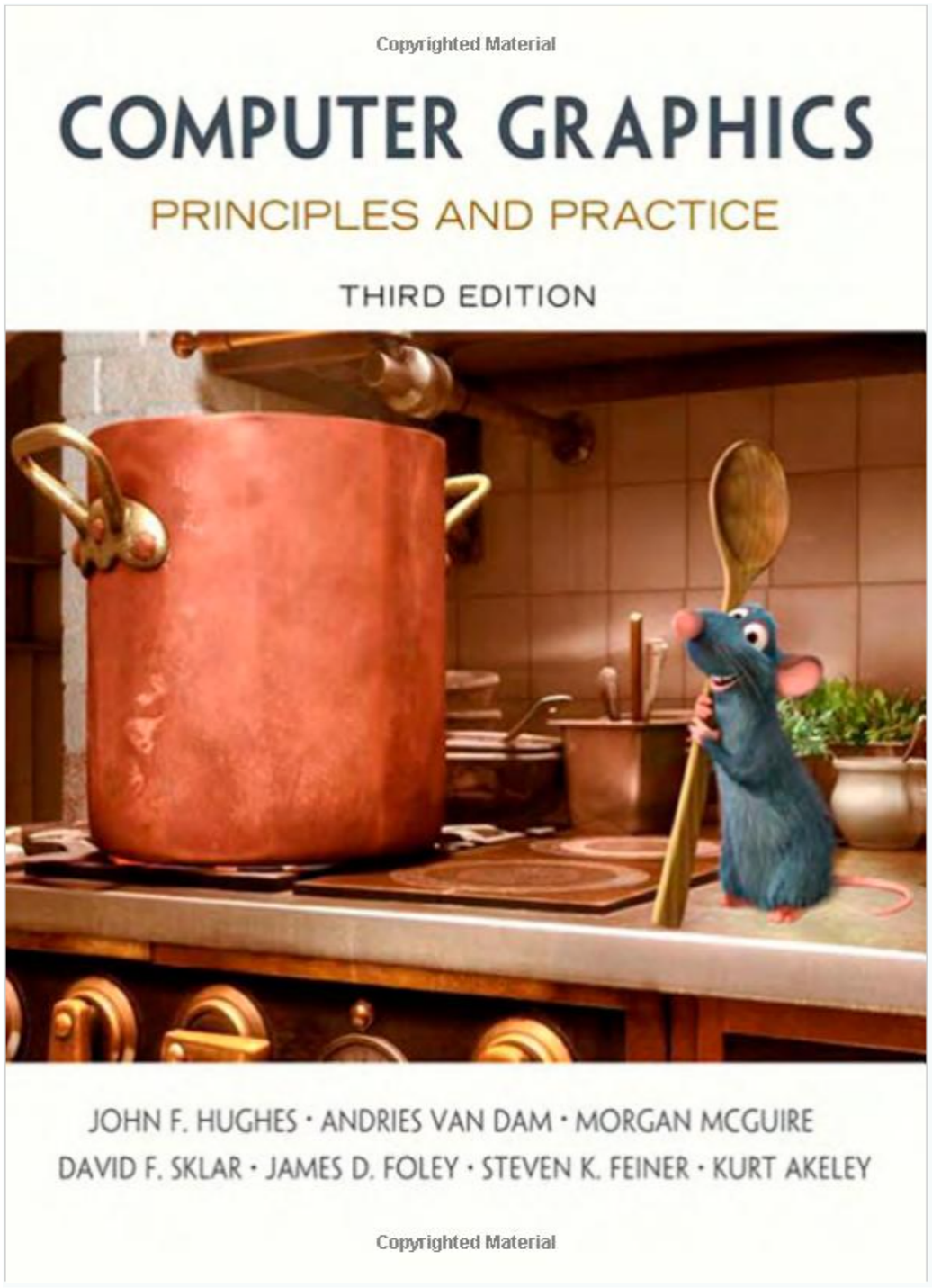
Rendering Techniques — 2 week

Interactive Computer Graphics — 3 weeks
- Animations
- Collision Detection
- Interaction / Navigation

Computer Graphics Fundamentals — 7 weeks
- Window Management
- Modeling
- Transformation
- Viewing in 3D
- Lighting & Shading
- Texturing

Programming Concepts — 2 weeks
- C/C++ / Object Oriented programming
- GLSL programming

Books

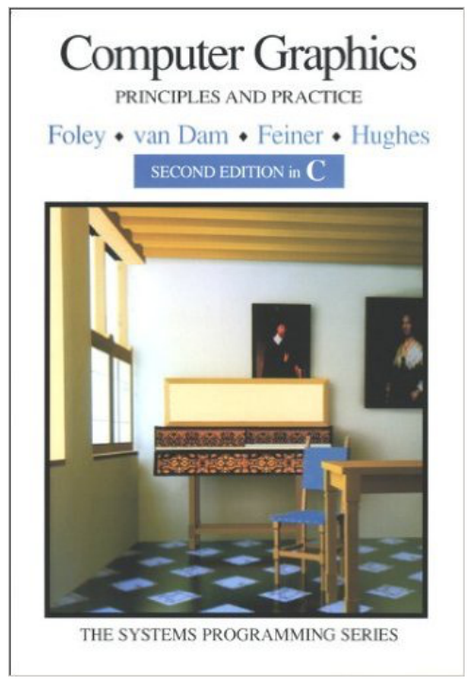


Hughes, van Dam, McGuire, Sklar, Foley, Feiner, Akeley: Computer Graphics: Principles and Practice (3rd Edition) 3rd Edition

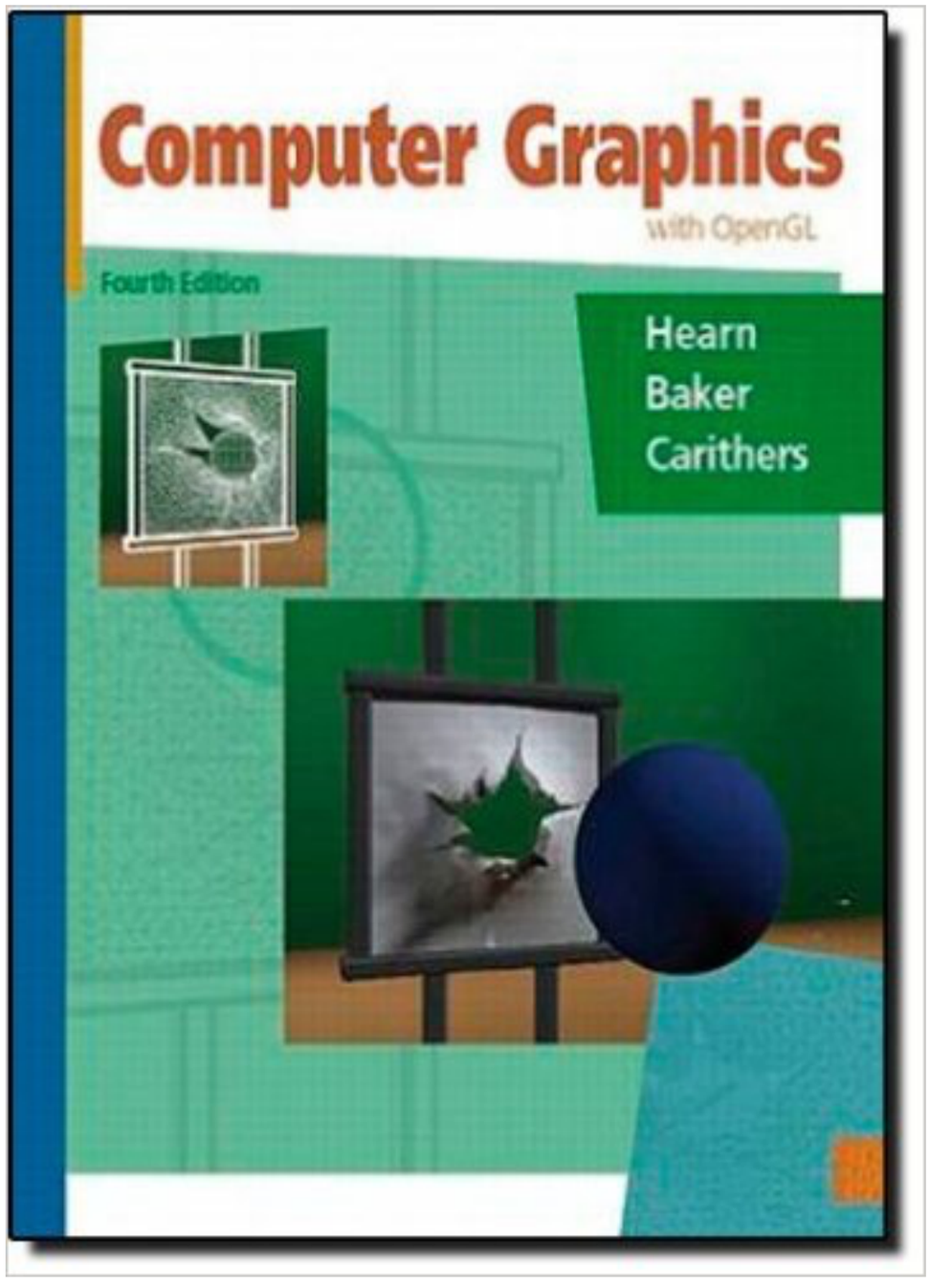
The course covers parts of this book only.

Online access via ISU digital library

Predecessor is considered as the bible of Computer Graphics!



Books



Hearn, Baker, Carithers:

Computer Graphics with Open GL, 4th Edition

- Theory is very good!
- The books uses an outdated version of OpenGL

Thank you!

Questions

Rafael Radkowski, Ph.D.
Iowa State University
Virtual Reality Applications Center
1620 Howe Hall
Ames, Iowa 50011, USA

+1 515.294.5580

rafael@iastate.edu
http://arlab.me.iastate.edu

www.linkedin.com/in/rradkowski

ARLAB

VRAC|HCI

IOWA STATE UNIVERSITY
OF SCIENCE AND TECHNOLOGY