

# Real-time Graphics

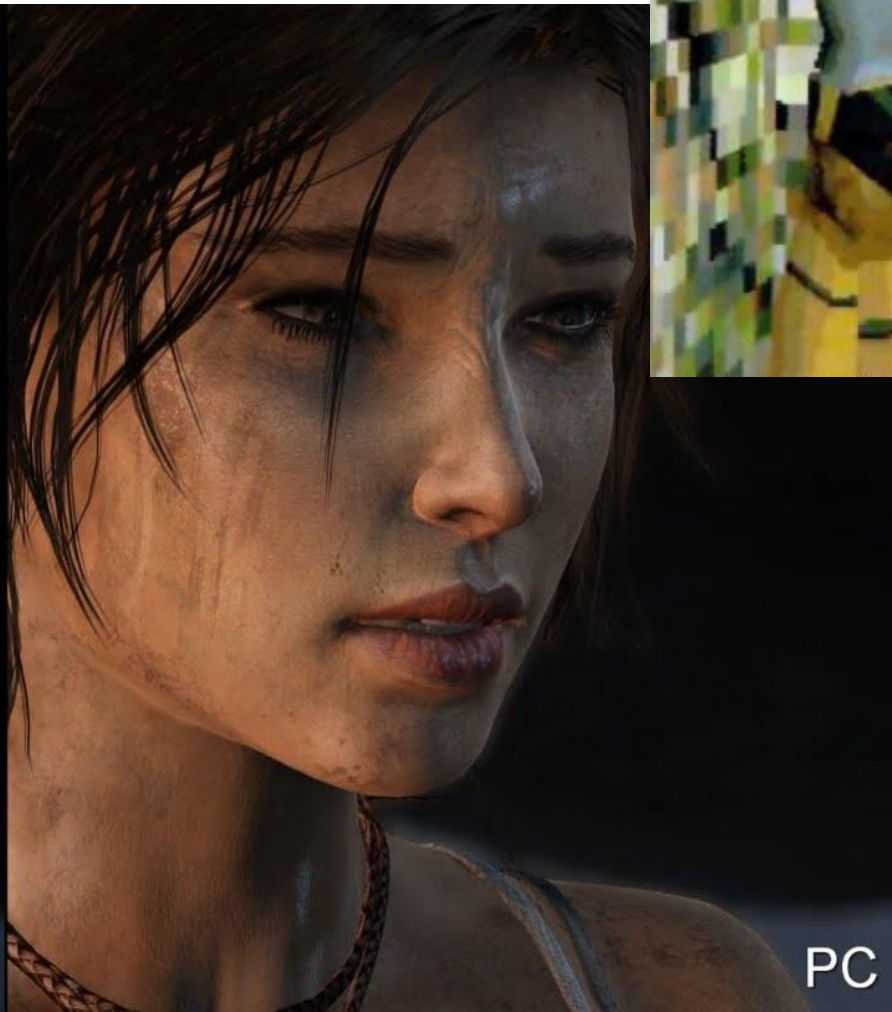
Andrej Mihálik  
mihalik@sccg.sk

16

# Motivation

- ▶ Rendering - visualization of 3D scene, geometry + material + effects
- ▶ Real-time - 60 frames per second, maintain constant rate
- ▶ Close approximation of reality
- ▶ Usage: games, games, games, scientific visualizations, interactive presentations
- ▶ Inclusion in web browsers (e.g. WebGL), cell phones (e.g. OpenGL ES), ...

# Motivation

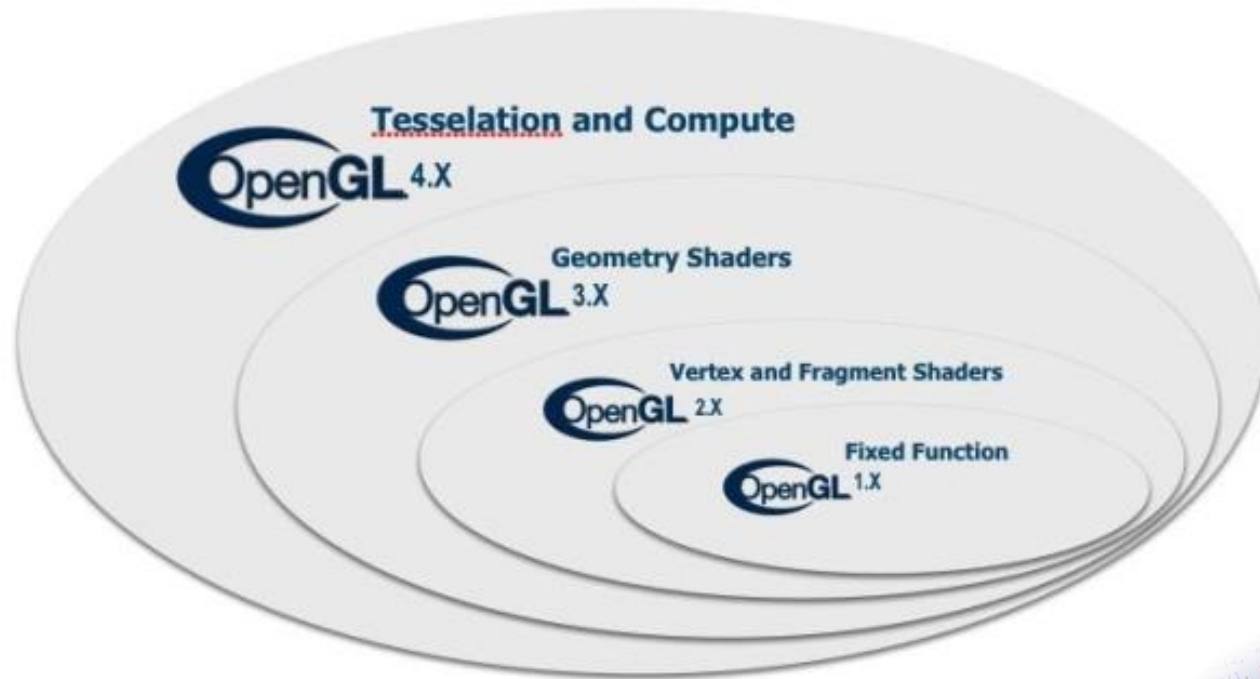




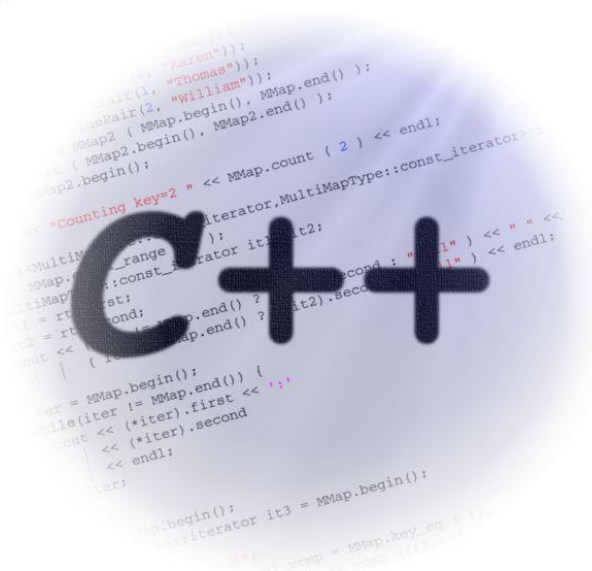
# Motivation



# Demonstrations & project



Visual C#



# Prerequisites

- ▶ Linear algebra, geometry
- ▶ Computer graphics
- ▶ Programming language - C, C++, C#, Java, Python, ...
- ▶ Willing to learn something new and exciting
- ▶ Lots of time

# Lecture plan

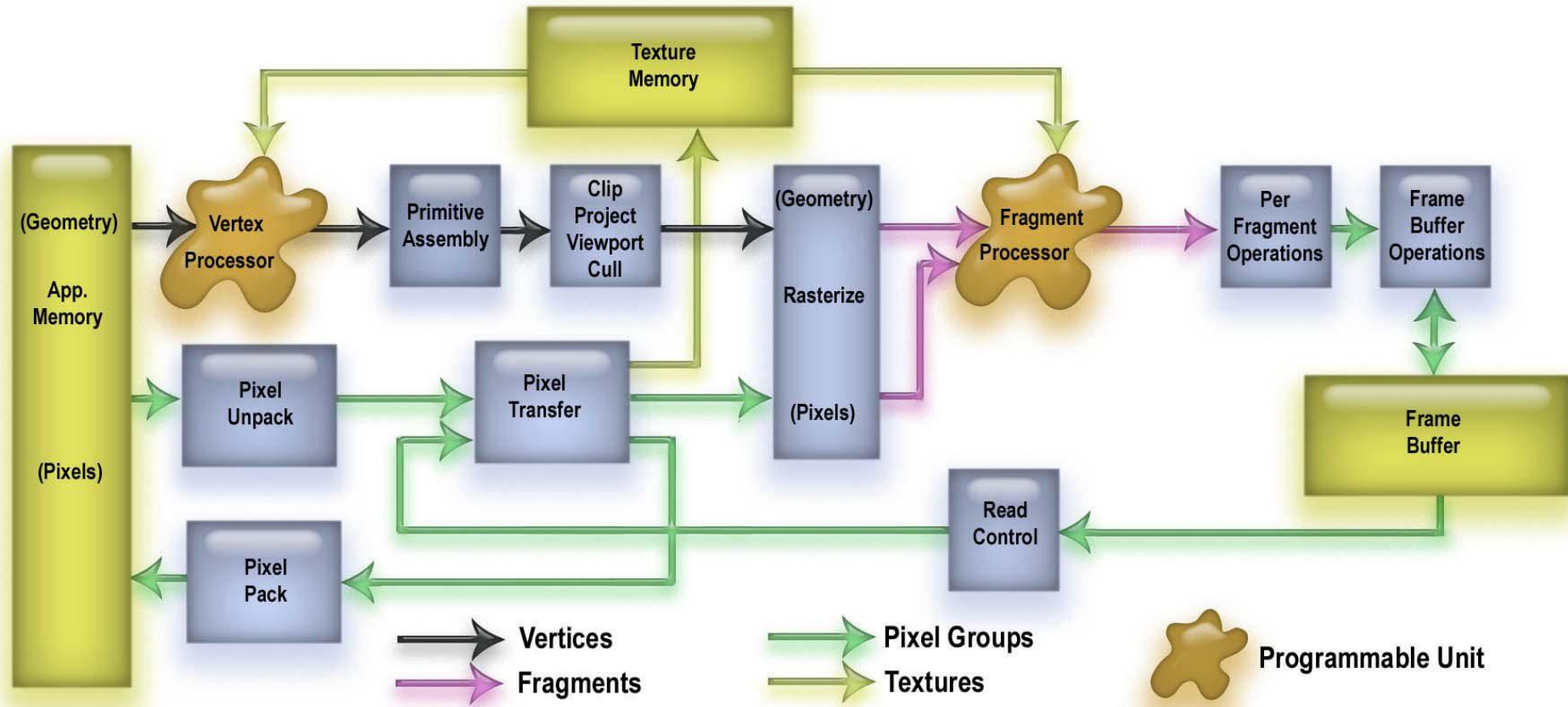
- ▶ Graphics pipeline, VBO, FBO, GLSL
- ▶ Shading, texturing
- ▶ Global illumination, shadows
- ▶ Reflections, refractions
- ▶ Optimization, culling techniques, collision detection, LODs, curves, terrains
- ▶ Post-processing, image based rendering
- ▶ GPGPU, raytracing
- ▶ Volume rendering
- ▶ Non-photorealistic rendering

# Graphics pipeline

- ▶ Based on architecture of graphics cards
- ▶ Processing of geometry
- ▶ Input = geometry and its properties
- ▶ Output = pixels
- ▶ OpenGL = API for setting pipeline parts and inserting geometry
- ▶ Fixed parts, programmable parts

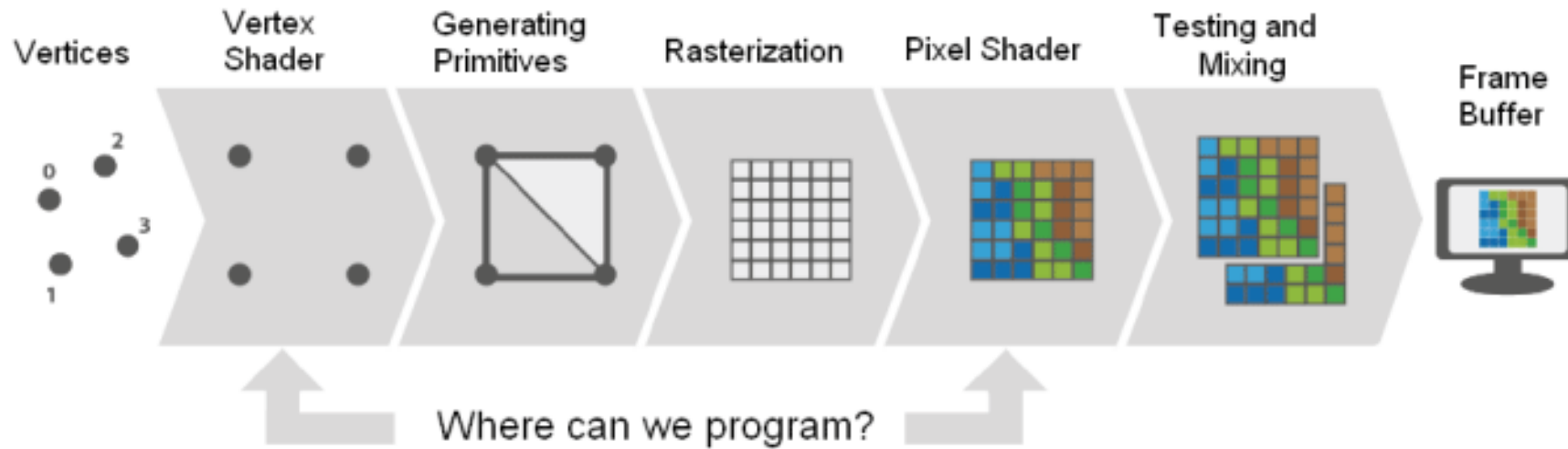


# Graphics pipeline



# Graphics pipeline

## OpenGL 2.0 Graphics Pipeline



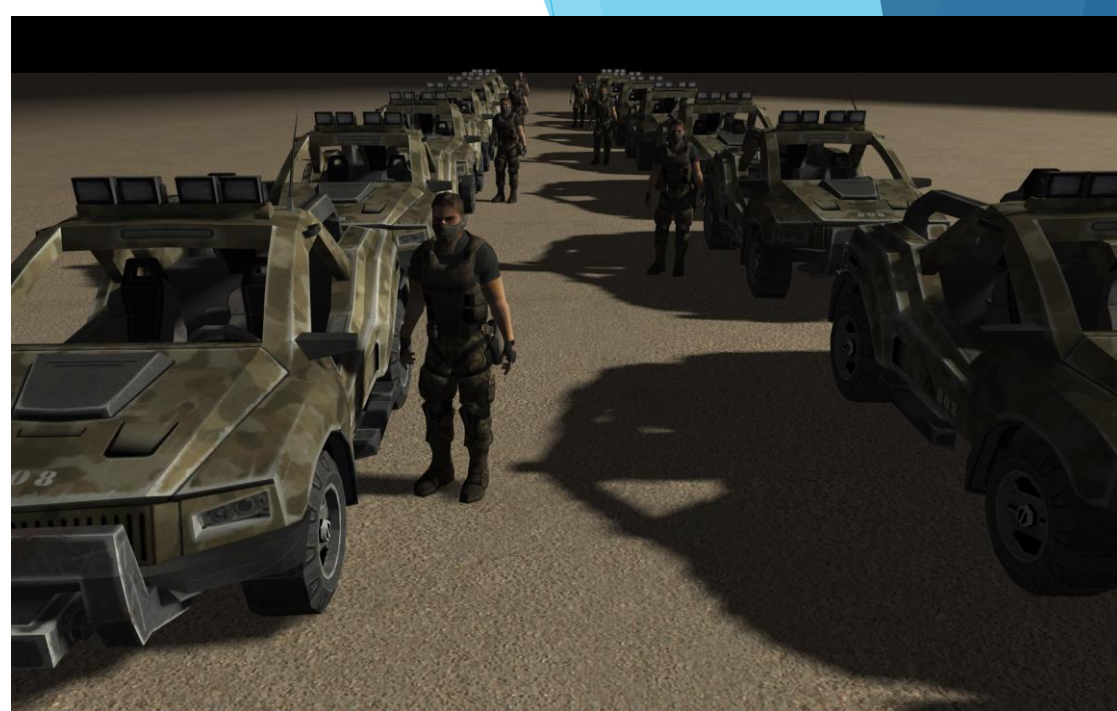
# Shading, textures

## ► Improving visual quality





# Shadows





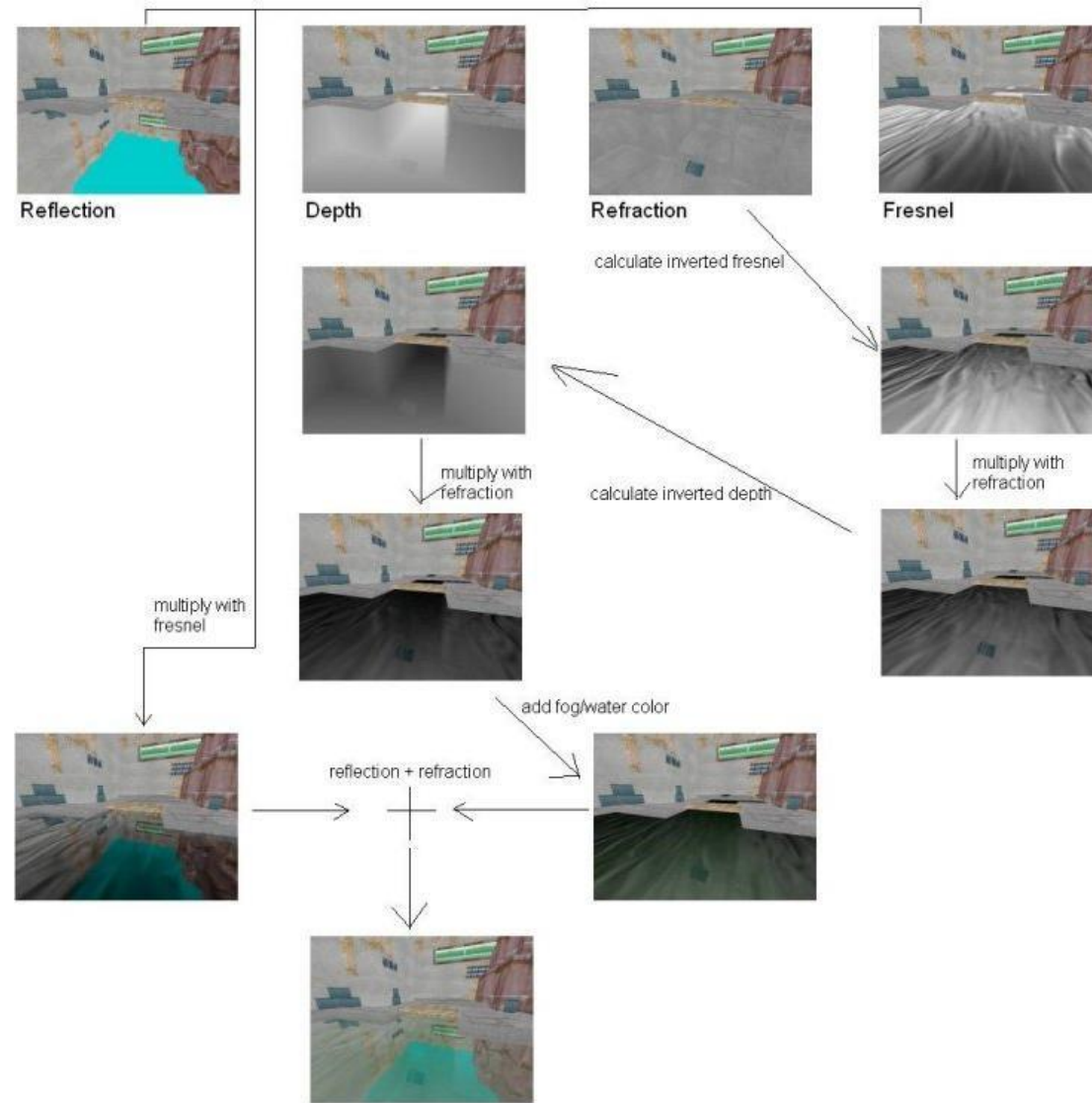
# Global Illumination

- ▶ ambient occlusion

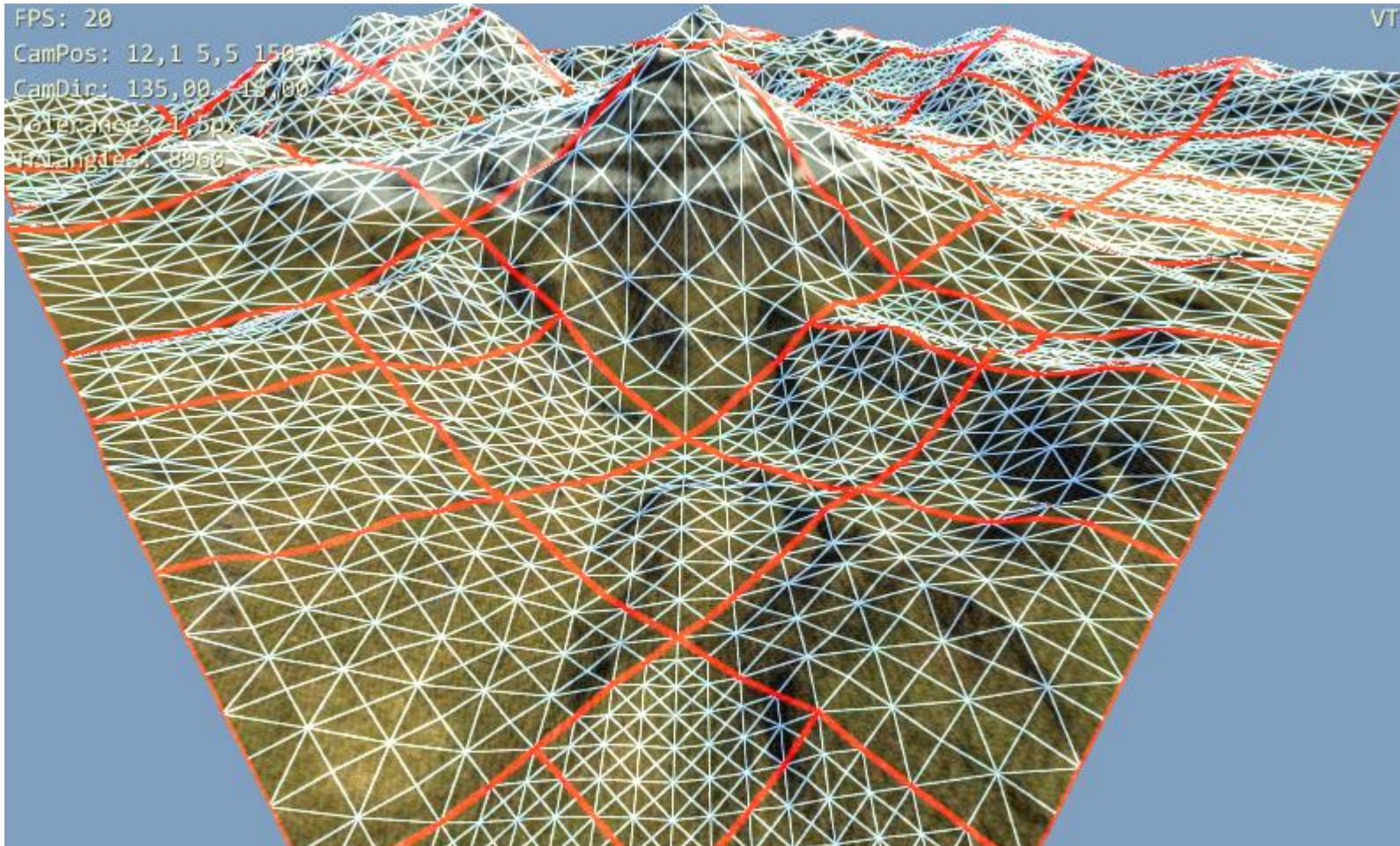




# Reflections

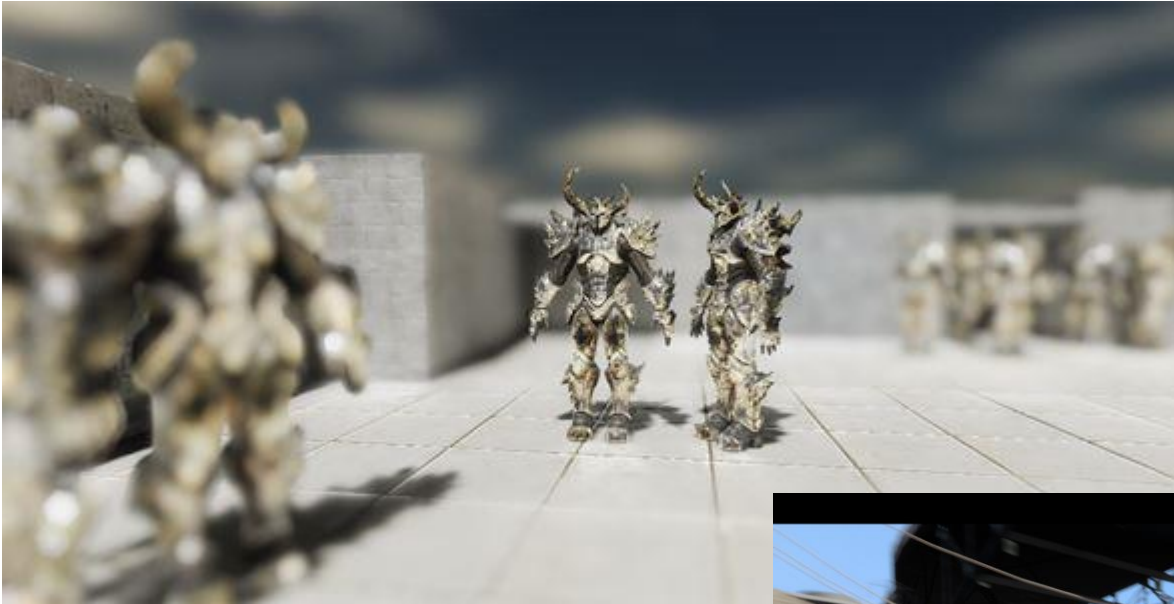


# Terrain, LOD

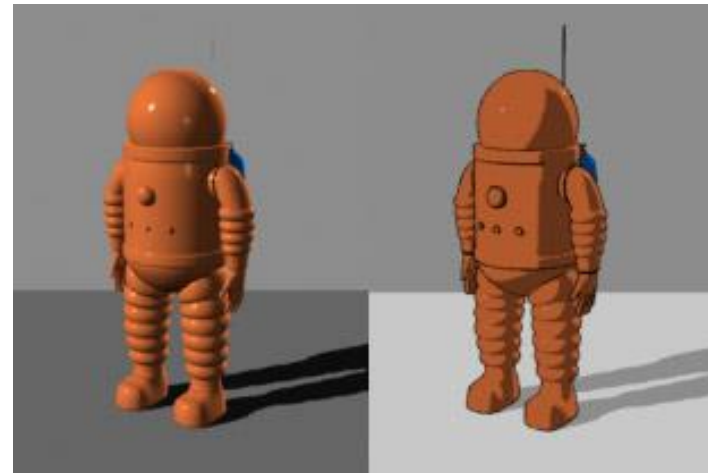




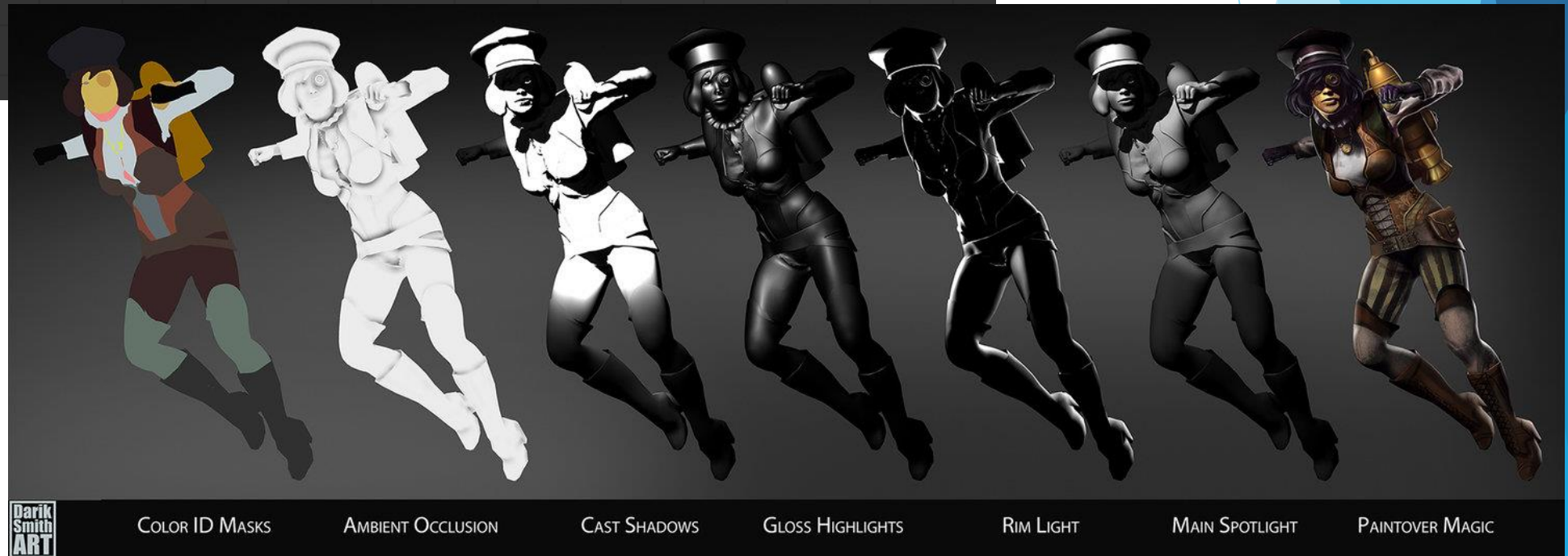
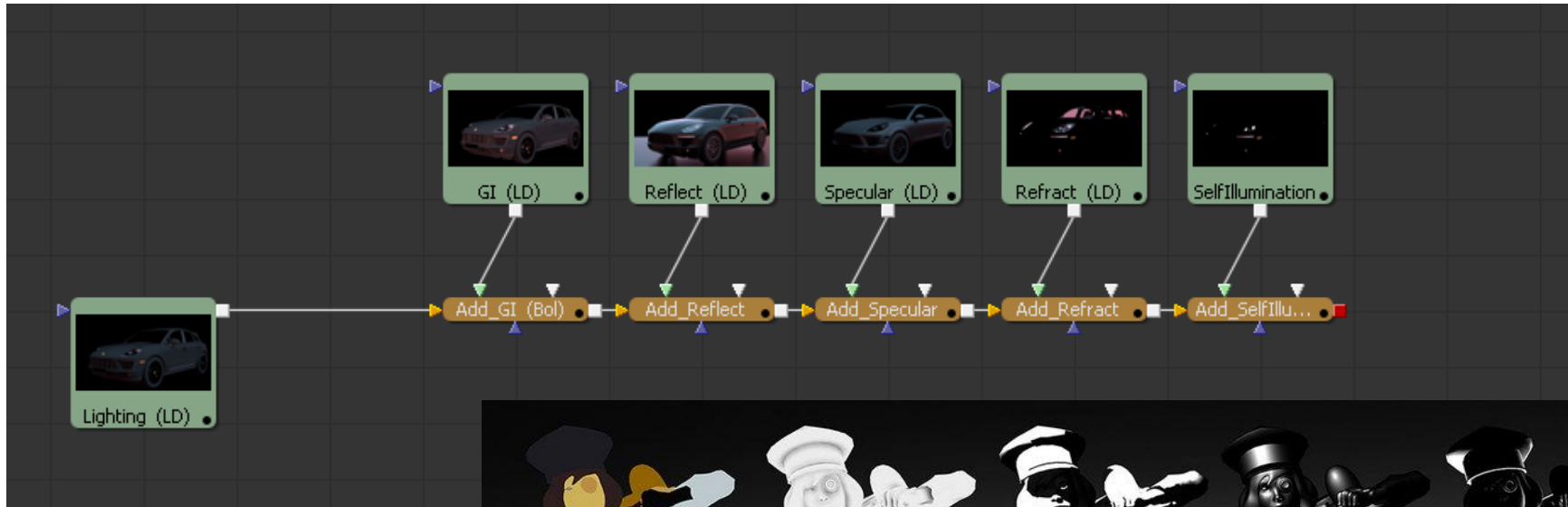
# Post-processing



# Non-photorealistic rendering



# Compositing, multipass rendering





# Project

- ▶ Project is demo program that uses OpenGL and GLSL for visualization of scene
- ▶ Necessary conditions:
  - ▶ Loading of scene and all objects from external file
  - ▶ At least 4 animated objects, including necessary animated camera and 1 light
  - ▶ All objects should be textured and rendered using shaders
  - ▶ At least 3 light sources (point + directional)
  - ▶ 2 basic per-pixel lighting methods (phong, phong+normal mapping), switched in real-time
  - ▶ At least 3 different shader programs (vertex+fragment shader)
  - ▶ Rendering to texture, Shadows

# Project

- ▶ Pick 3 additional packages of effects:
  - ▶ Using geometry shader for generating subdivision surfaces
  - ▶ Displacement mapping, Terrain rendering with LOD
  - ▶ Depth of field, Motion blur
  - ▶ Screen space ambient occlusion
  - ▶ HDR rendering of sun, Lens flare, Bloom effects
  - ▶ Parallax, bump, relief mapping
  - ▶ Reflection and refraction on water surface
  - ▶ Particle system for waterfall and fire visualization, particles update and rendered using shaders
  - ▶ Volume rendering of clouds, volumetric effects (smoke, fog, light volumes)
  - ▶ Toon, cell shading, Oren-Nayar & Cook-Torrance per-pixel lighting

# Rating

- ▶ Project: 70% - everything on time, complexity, fulfilled conditions
- ▶ Oral exam: 30% (min. 15%) - understanding of the topics from the lesson