Real-time Graphics

Andrej Mihálik

mihalik@sccg.sk

I6

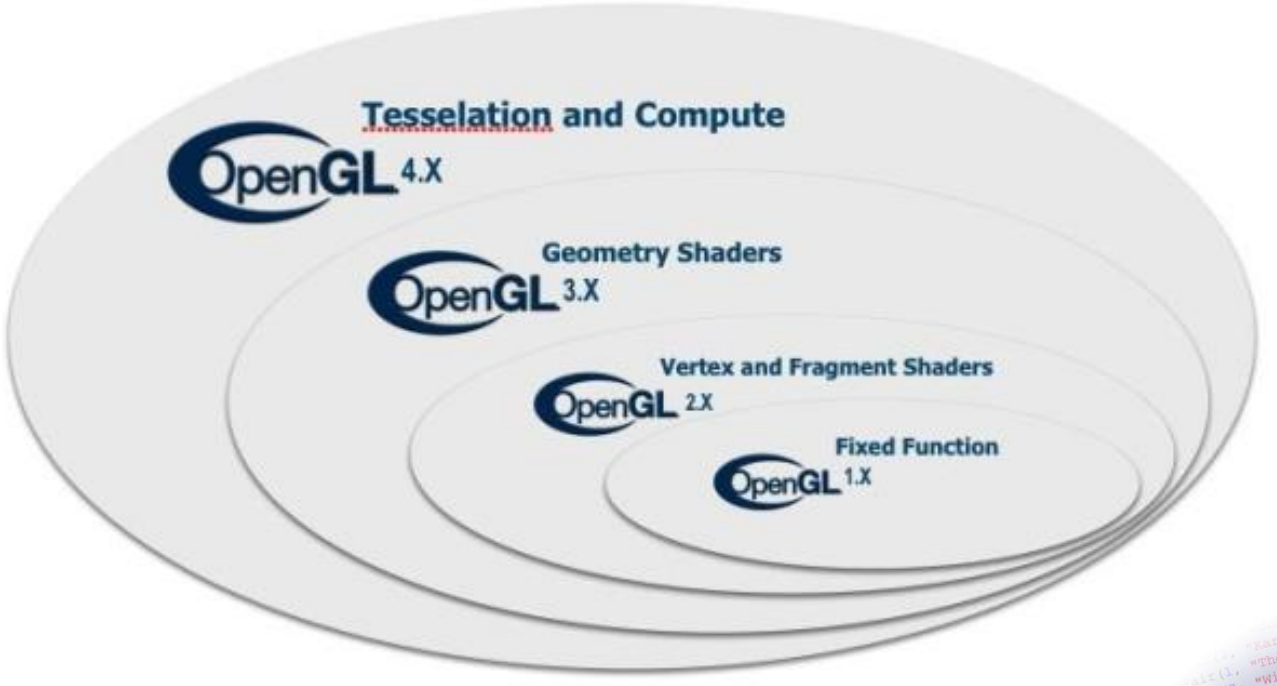
Motivation

- 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

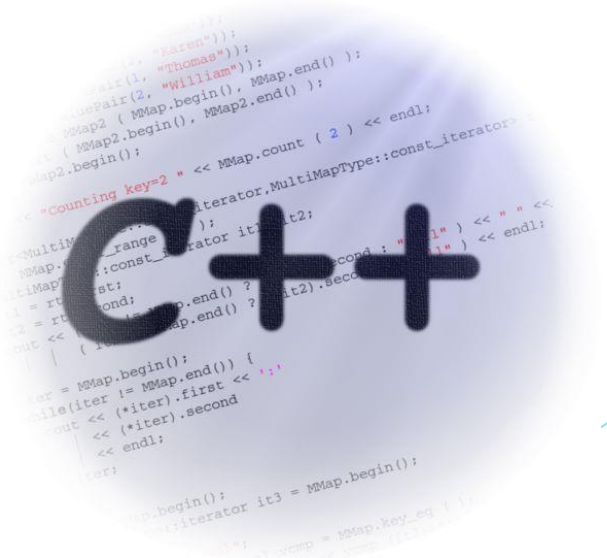
Motivation

Demonstrations & project



OpenGL|ES

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

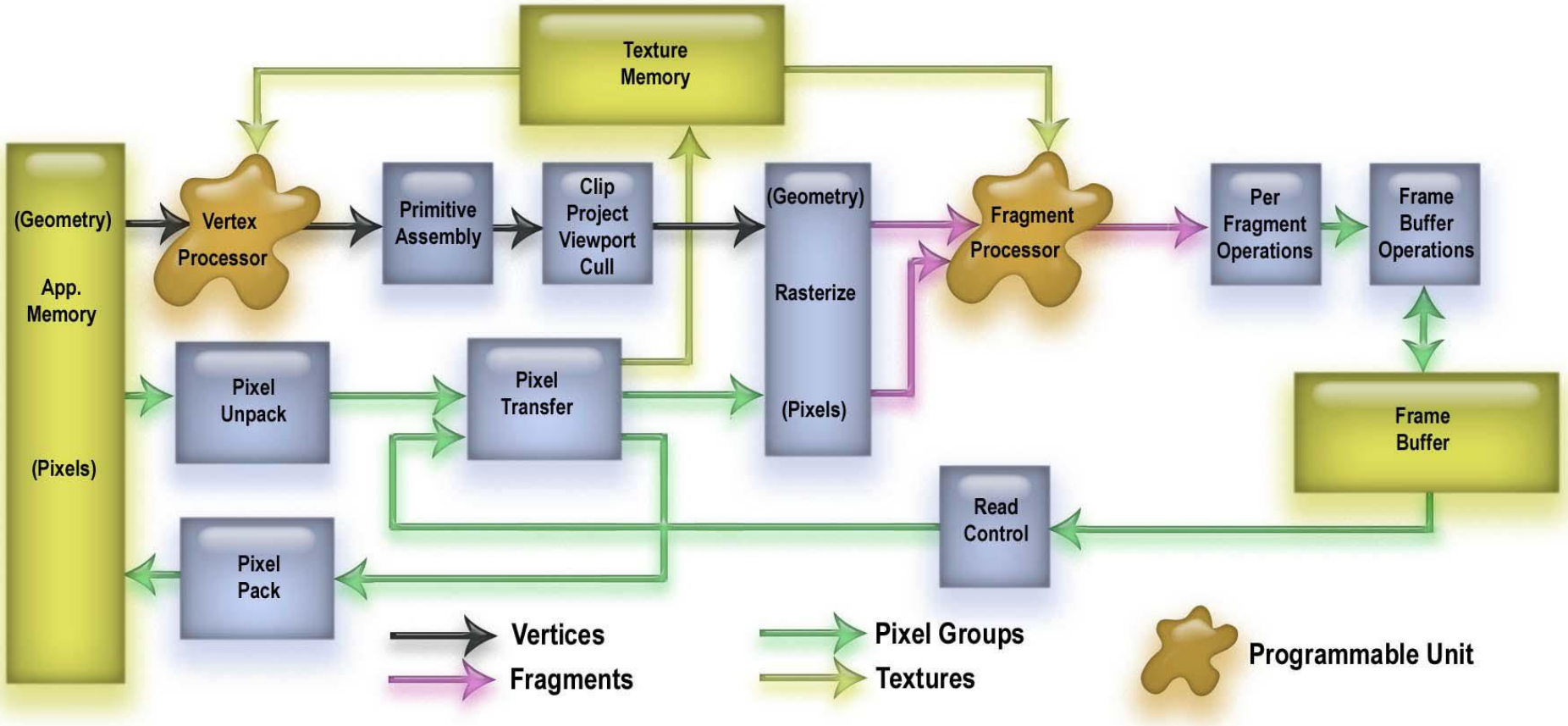
Course plan

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

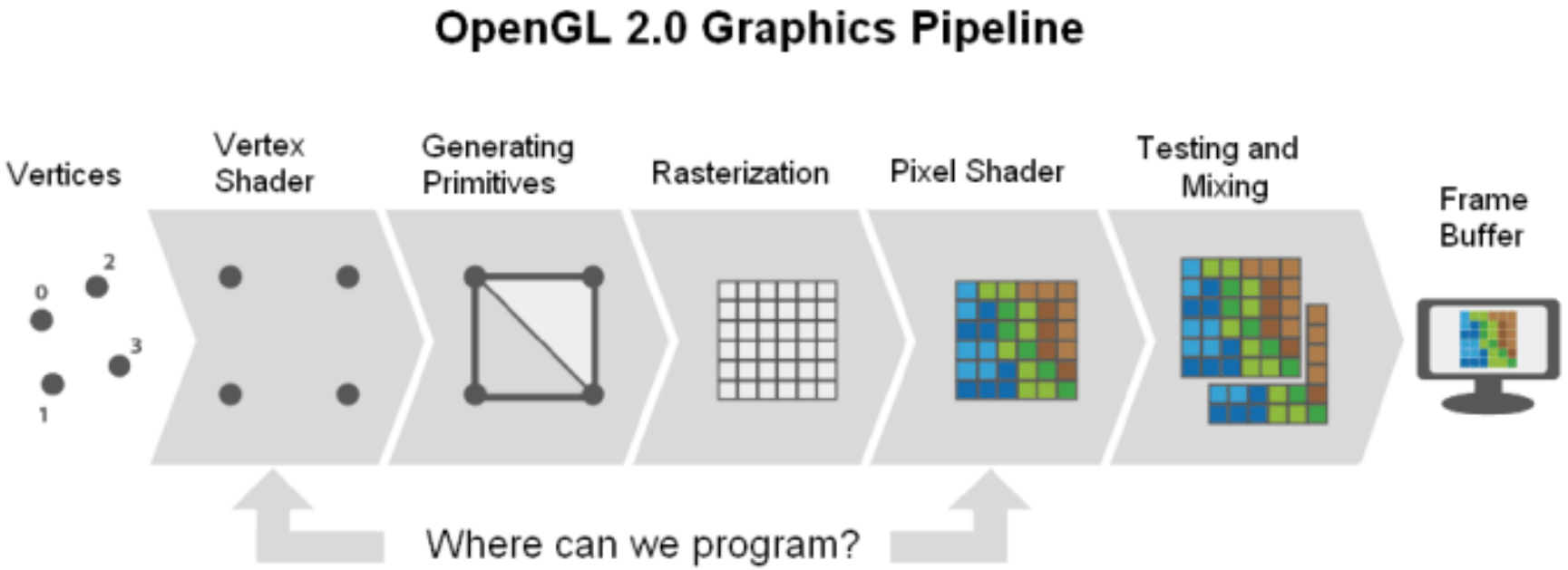
Graphics pipeline

- 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



Shading, textures

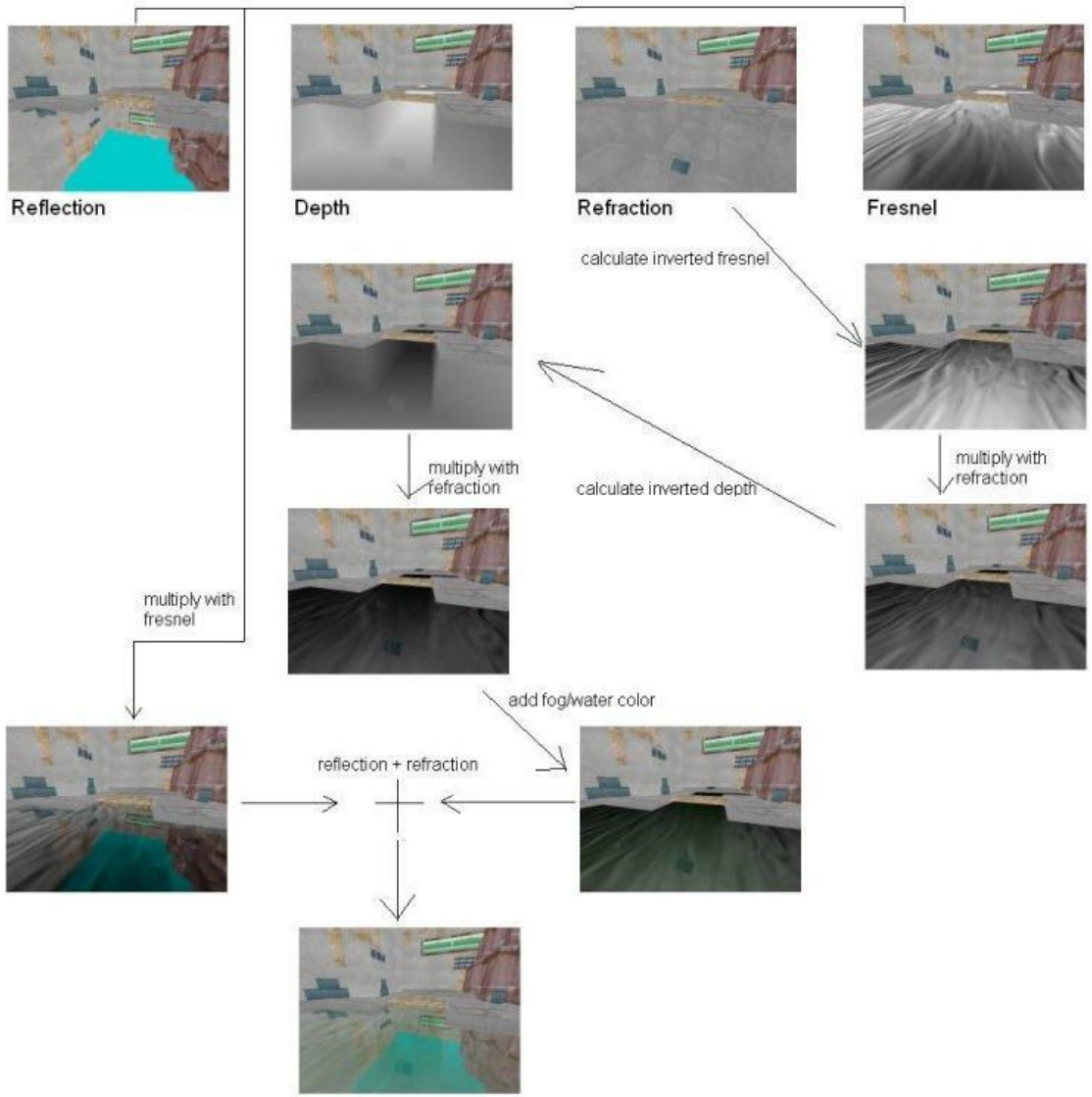
- Improving visual quality

Shadows

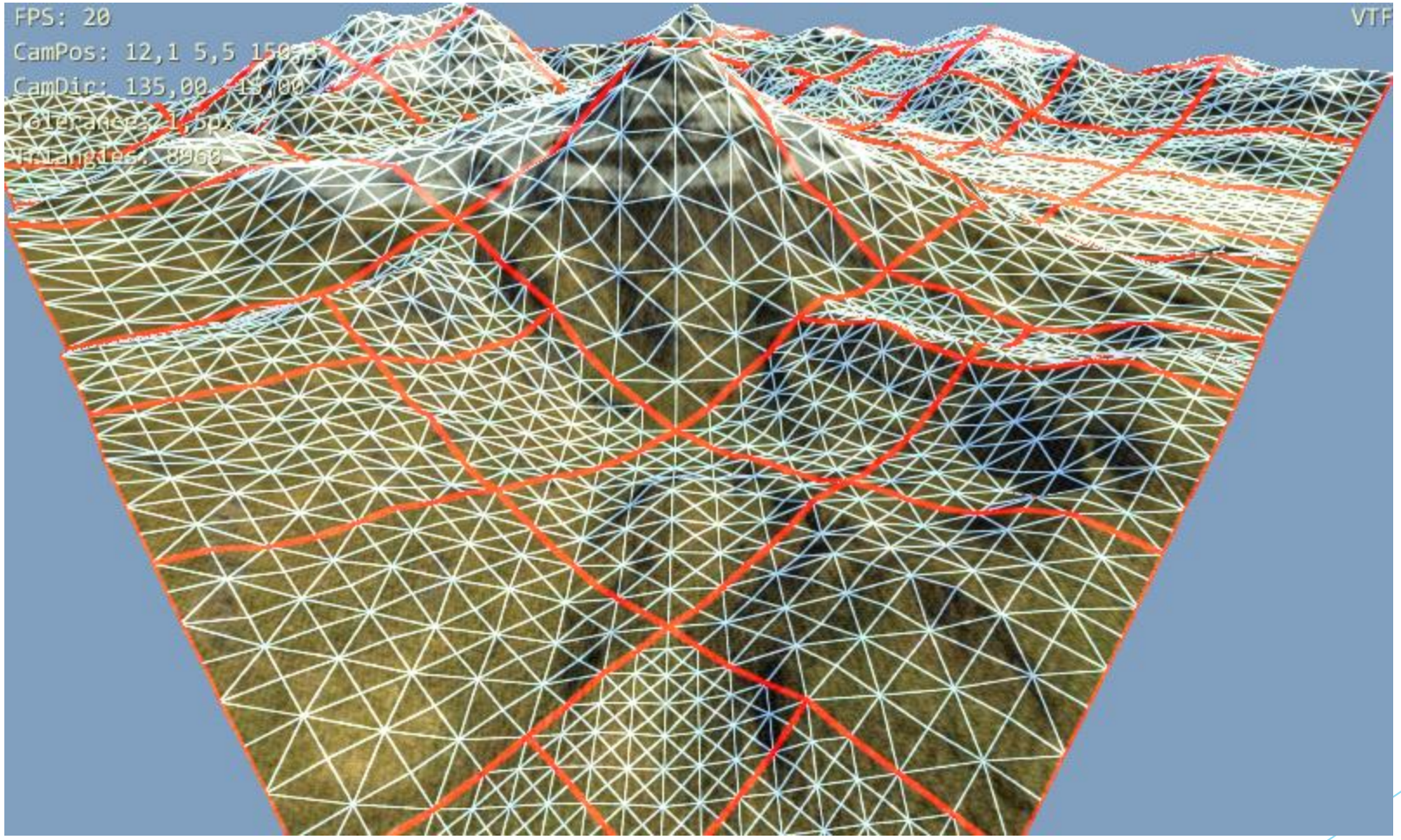
Global Illumination

- ambient occlusion

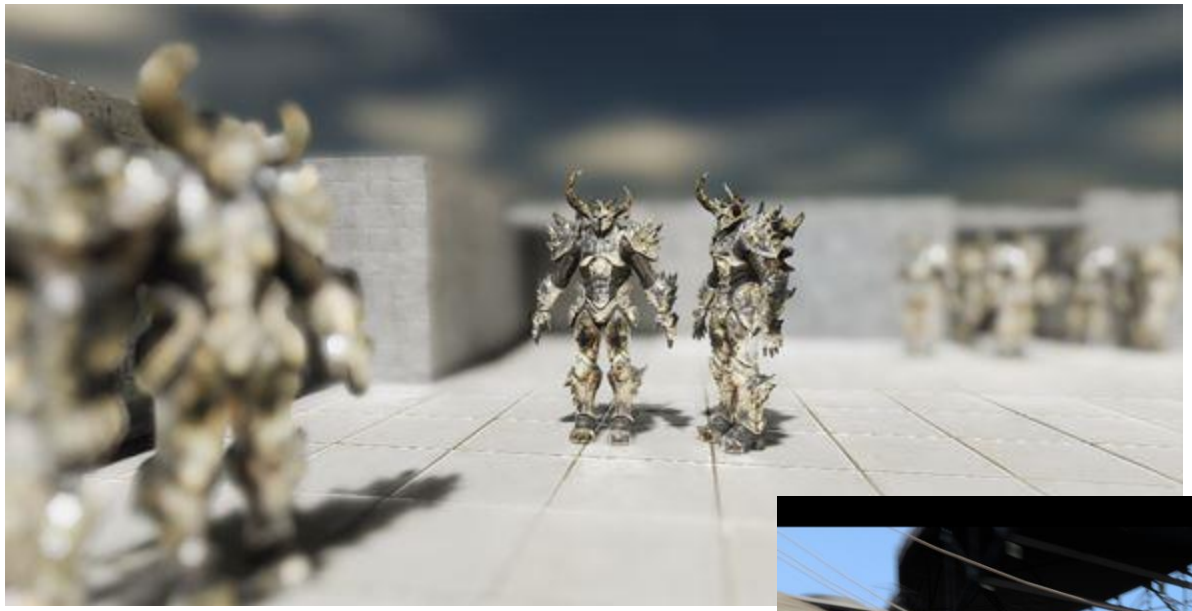
Reflections



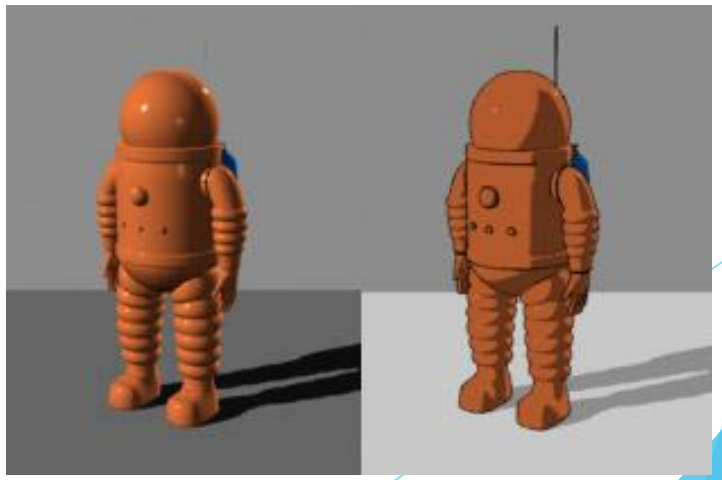
Terrain, LOD



Post-processing



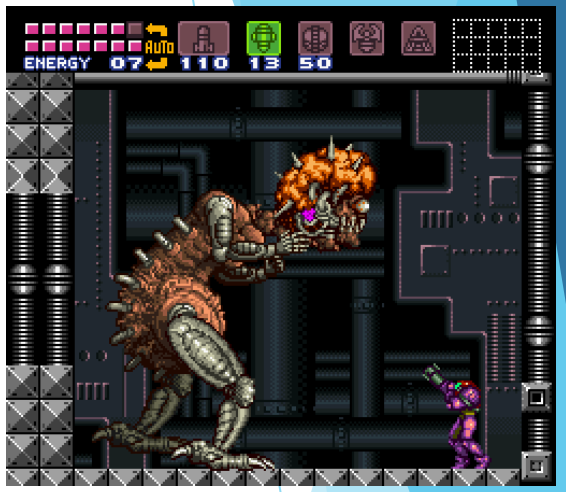
Non-photorealistic rendering



Ageing of graphical styles

Metal Gear Solid: The Twin Snakes (2004)

The Legend of Zelda: The Wind Waker (2002)



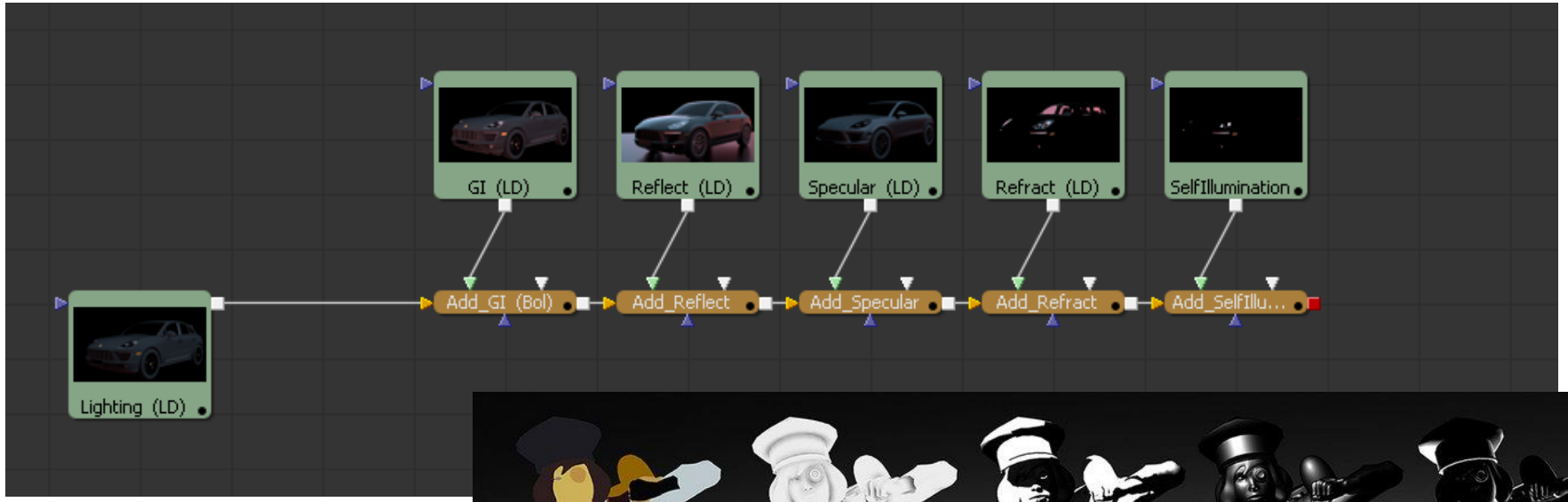
Super Metroid (1994)

Metal Gear Solid V (2015)

Splatoon 2 (2017)

Axiom Verge (2014)

Compositing, multipass rendering



Project

- Project is demo program that uses OpenGL and GLSL for visualization of scene
- Necessary conditions:
  - Loading of 4 objects from external file.
  - At least 1 moving/animated object.
  - Moving camera.
  - All objects should be textured and rendered using shaders
  - At least 3 light sources (point + directional)
  - At least 3 different shader programs (vertex+fragment shader)
  - Rendering to texture or shadows

Project

- Pick 1 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 or fire visualization
  - 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