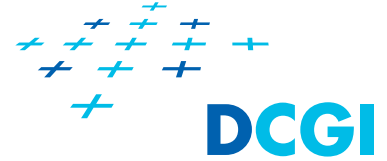


What you must learn for exam

GLUT



Event driven programming

(events, event handlers, event loop,...)

Events in graphics application

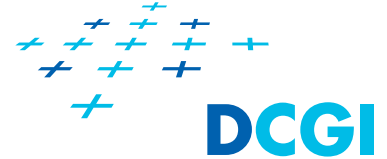
window (reshape, draw)

mouse (click / release, drag, passive move)

keyboard (key press / release, ASCII and special keys)

timer (idle, timer)

Shaders I - programs



Programmable blocks in raster graphics pipeline

Vertex shader (VS)

Fragment shader (FS)

How to pass programs to shaders

How to connect their inputs and outputs?

attributes

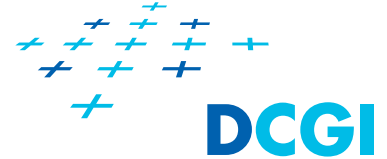
built in variables

(gl_VertexID, gl_PointSize, gl_Position, gl_FragCoord, gl_PointCoord)

user defined variables

- GLSL data types, swizzling

Shaders II - data



Passing vertex data to GPU

- attribute variables, uniforms

- buffers (VBO, VAO,...)

Buffer organization

- geometry as sequence x indexed vertices

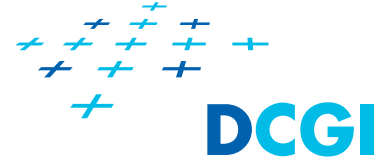
- order of vertex attributes in buffers
(sequential block, interleaved)

Drawing loop

Primitives

Back face culling

Transformations I



Fundamental concepts

point, vector, bases, coordinate frames, coordinates, ...

Matrices representing transformations

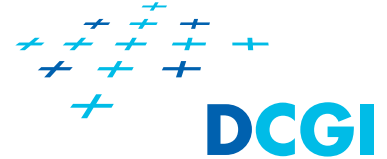
- Recognize the matrix type (linear, affine, projection) ...debug
- Derive/write matrices for: $R_x, R_y, R_z, S, T, V_p, P$ (explain $1/w$)
reference frames (change of: lookAt, on curve;
transf. with respect to: axes, virtual trackball)

Combination of transformations

- Order of matrix multiplications, decomposition of TL
- Order of transformations in OpenGL

Limitations of matrices: gimbal lock, interpolation of rotations

Transformations II



Fundamental concepts [done / today]
point, vector, bases, coordinate frames, coordinates, ...

Matrices representing transformations

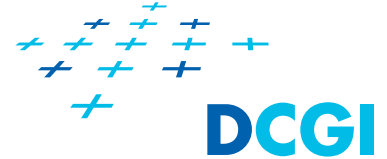
- Recognize the matrix type (linear, affine, projection) ...debug
- Derive/write matrices for: $R_x, R_y, R_z, S, T, V_p, P$ (explain $1/w$)
reference frames (change of: lookAt, on curve;
transf. with respect to: axes, virtual trackball)

Combination of transformations

- Order of matrix multiplications, decomposition of TL
- Order of transformations in OpenGL

Limitations of matrices: gimbal lock, interpolation of rotations

Lighting



Illumination (lighting) model (for point in the scene)

- Lambertian diffuse illumination model
- Phong empirical illumination model
complete equation and equations for each component
ambient, diffuse, specular, and emissive

Normals

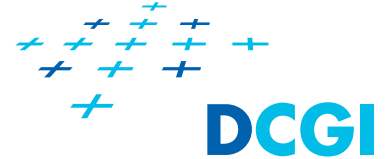
- usage in graphics, computation for a triangle mesh
- normal matrix for non-rigid transformations

Shading models (flat, Gouraud, Phong)

Lights types: directional and positional – point light, reflector

Gamma correction

Textures



- What is texture? What can texture be used for?
- Direct and inverse texture mapping
Why do we use the inverse mapping?
- Texture coordinates for texture bound to object
 - assignment during modelling
(inverse parametrization, intermediate map surfaces)
 - perspective correct interpolation in runtime
- Texture of environment reflected on object (env. mapping)
- In OpenGL
 - Texture object, data, parameters (min, max filtering)
 - Application of texture in FS (fragment color computation)

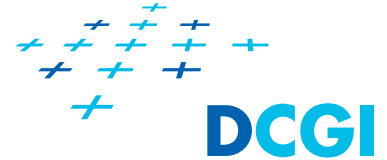
Fog and Antialiasing

Fog based on distance to camera

Problem of aliasing (reasons, examples)

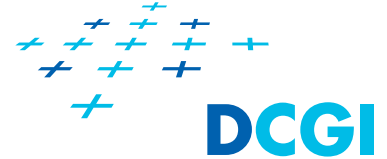
Antialiasing methods (coverage, multisampling,
accumulation)

Pipeline



- Complete OpenGL 3.1 pipeline
 - fixed & programmable blocks
 - Which block performs what?
(Geometric transformations, rasterization, clipping, visibility, color, texturing, ...)
- Video memory layers
 - usage by other pipeline blocks
 - how to write to each layer

Interaction



Mouse input of logical values

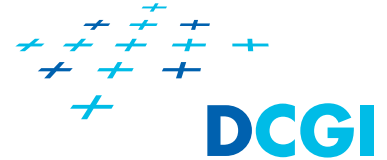
(conversion of device coordinates to logical ones)

world coordinates (inverse transformation chain -unProject)

object selection (pick \rightarrow id)

rotation (mapped to axes and virtual trackball)

Curves I



Ways of representing curves and their pros and limits

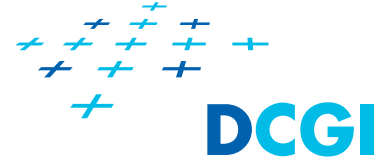
Matrix notation of parametric curves

Interpolating and approximating curves

Connectivity, maximum connectivity for the curve of degree n ,
degradation of connectivity for repeated control points and
repeated values in knot vector

Curves: Fergusson, Catmull-Rom,
Bézier (de Casteljau alg.),
B-spline – uniform (Coons),
non-uniform,
non-uniform rational (NURBS)
(Cox – de Boor alg.),

Curves II



Matrix notation of parametric curves

Knot vector, vector of weights

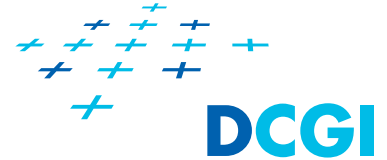
Degradation of connectivity

for repeated control points and
repeated values in knot vector

B-spline curves:

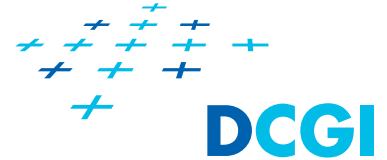
uniform (Coons),
non-uniform, Cox – de Boor alg.
non-uniform rational (NURBS)

Quaternions



- Quaternions
 - Variants of notation
 - Fundamental operations (norm, inverse, conjugate, product,...)
 - Representation of vectors as pure quaternions
 - Unit quaternions
- Rotation and orientation
 - Meaning (difference between orientation and rotation)
 - Representation of orientation by unit quaternions
 - Performing of rotation using quaternions (sandwich product)
 - Rotation by interpolation of orientations ((n)lerp,slerp)
 - Principle of conversion (quaternion \leftrightarrow 3x3 matrix)

Rendering



- Principles of visibility computation
- Local x global methods for computation of illumination
- Notion about fundamentals of
 - Radiosity
 - Ray casting, ray-tracing, path-tracing
 - Light tracing
 - Bi-directional path tracing, photon mapping