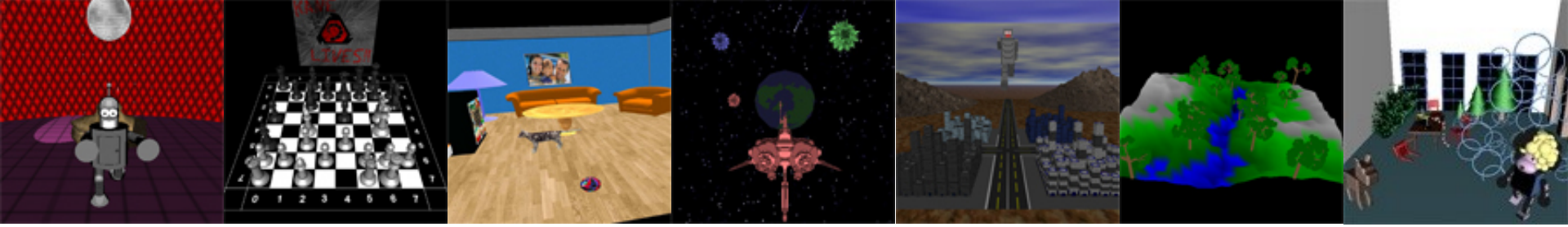


Computergrafik

Universität Bern
Herbst 2016

Assignments

- Programming assignments **every two weeks**
  - 6 in total, 10 points max each
- Theoretical assignments
  - 2 in total, 10 points max each
  - Schedule TBA

- **60 points in total required for the exam (=75%)**

Programming Assignments

- Use ExWi pool or your own computer
- Need support for OpenGL 3.2 or later
  - Update your graphics driver!
- Older Intel integrated graphics processors (before Sandy Bridge) do not currently support OpenGL 3.2!
- Use forum if problems arise

Programming Assignments

- Assignments and schedule on ILIAS
- Java base code and documentation on github
  https://github.com/mzwicker/Computergrafik-Basecode
- Turn-in electronically on ILIAS and demonstration to TA in ExWi pool
- Time-slot reservation for demonstration available on ILIAS
  - Takes place every 2nd Thursday afternoon

Demonstration to TA

- Show us the code you upload on ILIAS
  - We will check turn-ins on a random basis
  - Showing us different code will result in **late penalty (=50% of original score)**
- You should be able to demonstrate each part of the assignment individually without major changes to your code. Suggestions:
  - User interface
  - Config file
  - Separate projects
  - Global variable
  - ...

Programming Assignment 1

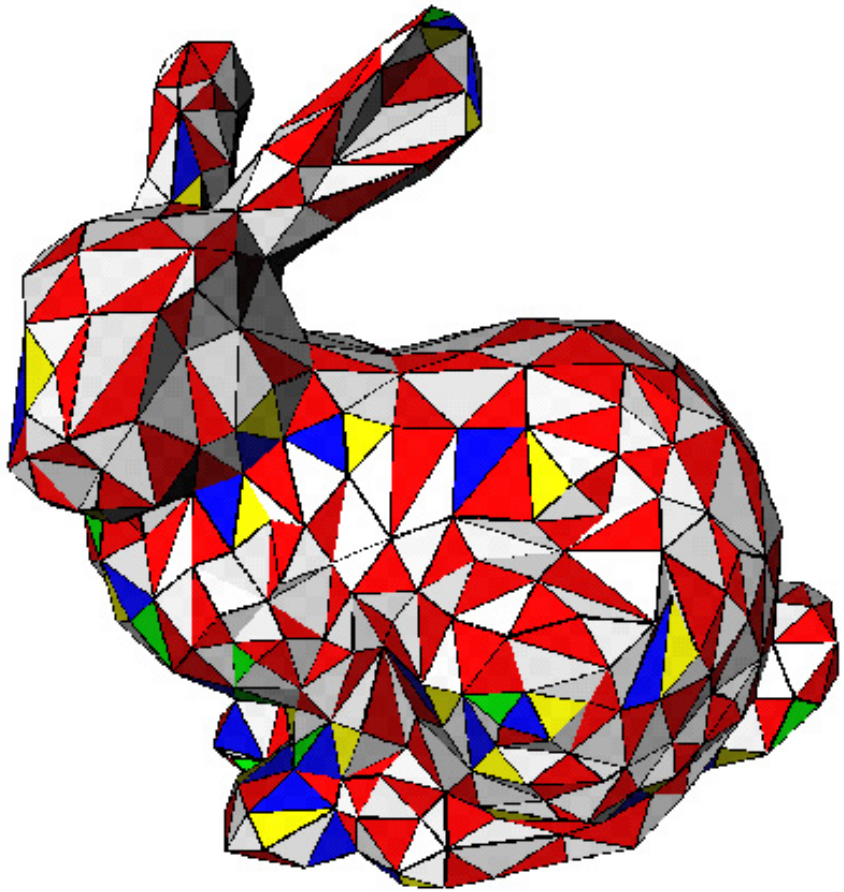
- Description available on ILIAS
- Turn-in electronically on ILIAS until **Thursday October 6th on 11:59**
- Demonstration on the same day between **12:15 to 14:00 in the ExWi pool**
  - Choose a time-slot on ILIAS under „*Sign up for Homework Presentation*”
  - Do not forget!

Programming Assignment 1

- Base Code
- Triangle Meshes
- Constructing Cylinders and Tori
- Concatenating Transformations

Triangle Meshes

- Data structure
  - Array of xyz vertex positions
  - Array of vertex indices
- Front and back facing triangles

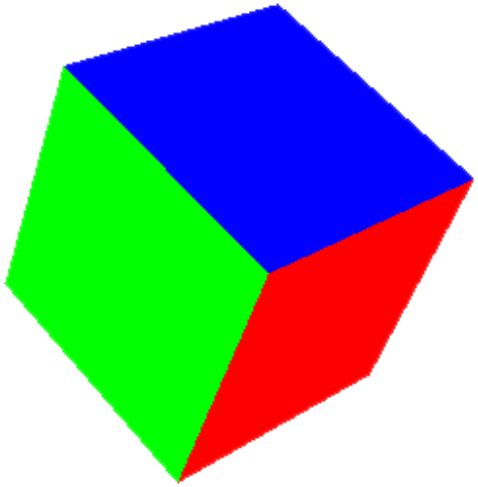


Triangle Meshes

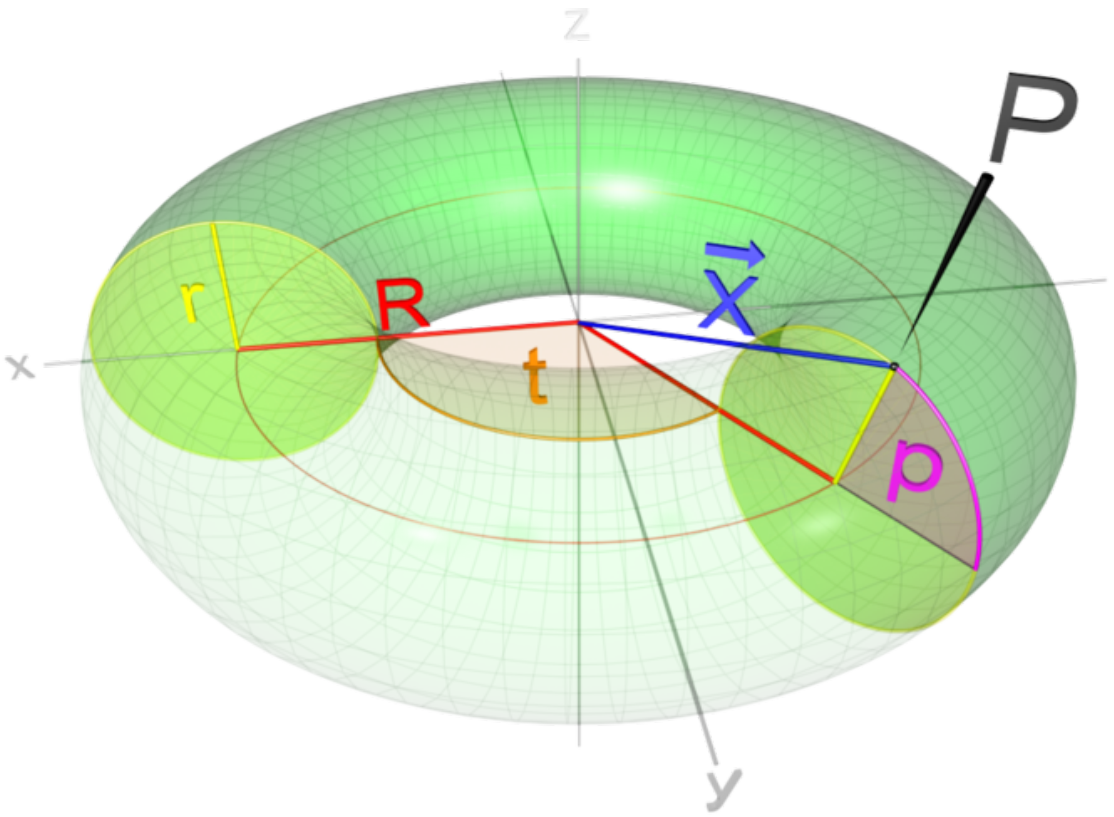
```
// The vertex positions of the cube
float v[] = {-1,-1,1, 1,-1,1, 1,1,1, -1,1,1,       // front face
             -1,-1,-1, -1,-1,1, -1,1,1, -1,1,-1,   // left face
             1,-1,-1, -1,-1,-1, -1,1,-1, 1,1,-1,    // back face
             1,-1,1, 1,-1,-1, 1,1,-1, 1,1,1,       // right face
             1,1,1, 1,1,-1, -1,1,-1, -1,1,1,       // top face
             -1,-1,1, -1,-1,-1, 1,-1,-1, 1,-1,1}; // bottom face

// The R,G,B vertex colors
float c[] = {1,0,0, 1,0,0, 1,0,0, 1,0,0,
             0,1,0, 0,1,0, 0,1,0, 0,1,0,
             1,0,0, 1,0,0, 1,0,0, 1,0,0,
             0,1,0, 0,1,0, 0,1,0, 0,1,0,
             0,0,1, 0,0,1, 0,0,1, 0,0,1,
             0,0,1, 0,0,1, 0,0,1, 0,0,1};

// The triangles (three vertex indices for each triangle)
int indices[] = {0,2,3, 0,1,2,         // front face
                 4,6,7, 4,5,6,         // left face
                 8,10,11, 8,9,10,      // back face
                 12,14,15, 12,13,14,   // right face
                 16,18,19, 16,17,18,   // top face
                 20,22,23, 20,21,22}; // bottom face
```



Constructing a Torus



$$x(t,p) = (R + r\cos p)\cos(t)$$
$$y(t,p) = (R + r\cos p)\sin(t)$$
$$z(t,p) = r\sin(p)$$

http://de.wikipedia.org/wiki/Torus

Concatenating Transformations

- Intuitive, math next time in class
- Example: Motion of object that rotates around some fixed point, while rotating around its center simultaneously