

CS 523: Computer Graphics, Spring 2009

Shape Modeling

OpenGL picking

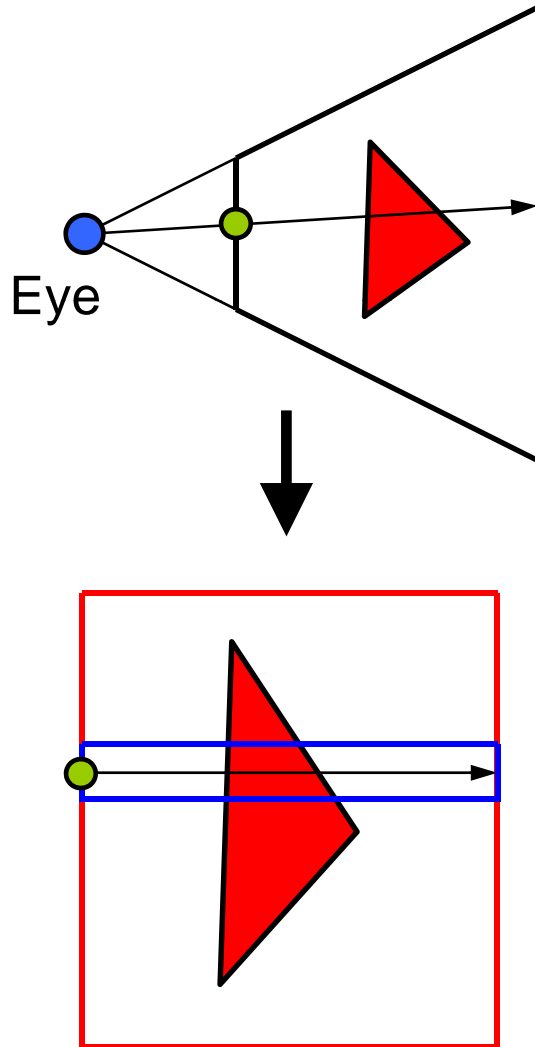
Lasso implementation by projection

OpenGL Picking

- Avoids traversing the model manually
- Picking can be done on multiple levels
- Uses graphics hardware (Framebuffer)
- Is performed in image space
 - Pickable primitives are rastered (with an ID)
 - `gluPickMatrix(...)` only looks at small region around provided $[x,y]$ image space coords
 - Provides „hit“ objectIDs and depths in $[0, 2^{32}-1]$

OpenGL Picking

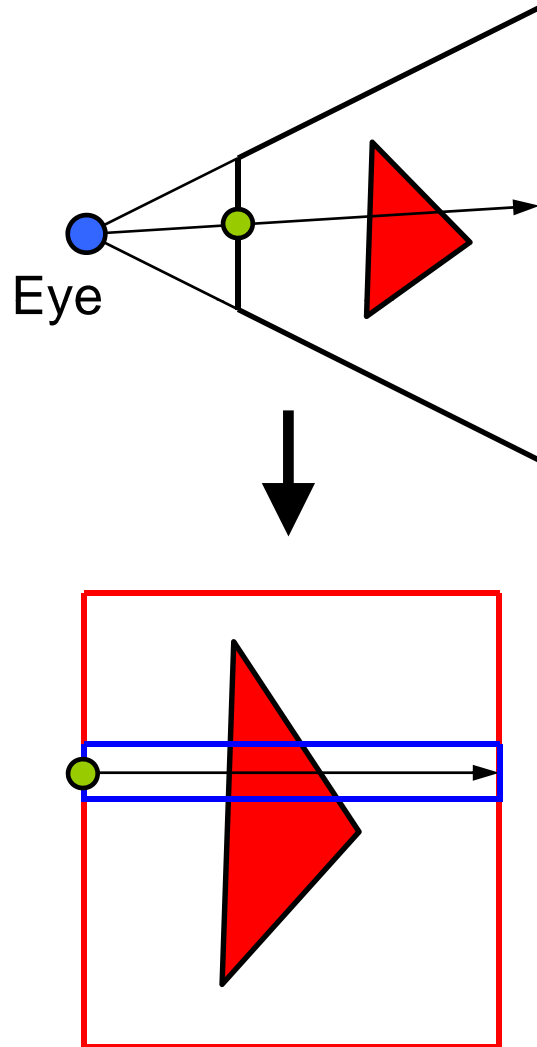
Code example



```
GLuint hitbuf[100];
glSelectBuffer(100, hitbuf);
glRenderMode(GL_SELECT);
glInitNames();
glPushName(0);
glMatrixMode(GL_PROJECTION);
glPushMatrix();
glLoadIdentity();
gluPickMatrix(...);
gluPerspective(...);
glMatrixMode(GL_MODELVIEW);
For each node
    glLoadName(name);
    transfAndDrawNode(...);
int hits =
glRenderMode(GL_RENDER);
```

OpenGL Picking

Code example



```
// process the hitbuffer data from the picking step
void processHits(int hits, GLuint * ptr) {
    GLuint names;

    double closest=1e8;

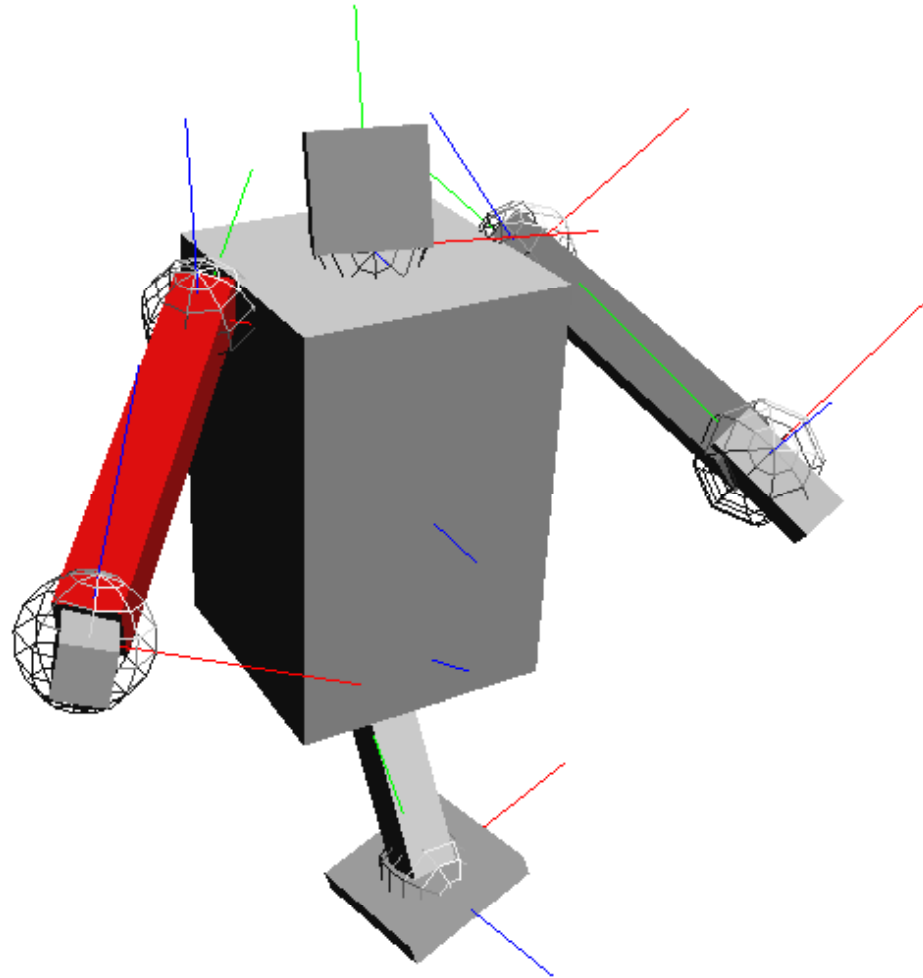
    for (int i = 0; i < hits; i++)
    {
        /* for each hit */
        names = *ptr; ++ptr;
        double z1 = (double) *ptr/0x7fffffff; ++ptr;
        double z2 = (double) *ptr/0x7fffffff; ++ptr;
        if (z1 < closest) {
            closest = z1;
            selected = nodes[*ptr];
        }
        ptr += names;
    }
}
```

n	z1	z2	id ₀	...	id _{n-1}	n	z1	z2	id ₀	...
---	----	----	-----------------	-----	-------------------	---	----	----	-----------------	-----

- Here: flat hierarchy, so there is always only one name for each hit object

OpenGL Picking

Demo + Code



Simple Projection

- Create image space mask
 - *Render* tessellated polyline into framebuffer, and read back into an array (`gluTess` in OpenGL redbook Appendix C)
 - *Paint* the mask onto screen
- Project all primitives (e.g. vertices) to image space using `gluProject(...)` and check if they are inside the lasso/paintbrush mask
 - If yes, add to (or subtract from) selected primitives

