



University of British Columbia CPSC 414 Computer Graphics

Hierarchies, Display Lists

- recap: matrix hierarchies, stacks
- display lists

Week 3, Monday 15 Oct 03

© Tamara Munzner

1

News

- Reminders
 - check web page often for updates
 - index, schedule
 - check newsgroup
- News
 - last week's slides posted as PDF
 - project 0 solutions posted

Week 3, Monday 15 Oct 03

© Tamara Munzner

2

Readings

- Week 1
 - Chap 1, graphics overview
- Week 2
 - Chap 2, graphics programming
 - Chap 4, geometry and transformations
- Week 3
 - Chap 9, hierarchical modeling
 - Chap 3, input and interaction

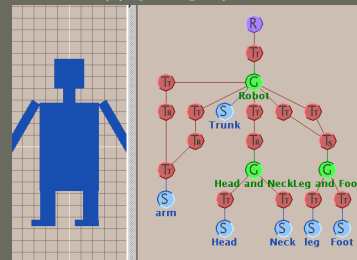
Week 3, Monday 15 Oct 03

© Tamara Munzner

3

Transformation Hierarchies recap

- hierarchies don't fall apart when changed
- transforms apply to graph nodes beneath



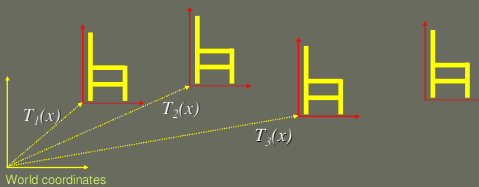
Week 3, Monday 15 Oct 03

© Tamara Munzner

4

Matrix Stacks recap

- Push and pop matrix stack
 - avoid computing inverses or incremental xforms
 - avoid numerical error



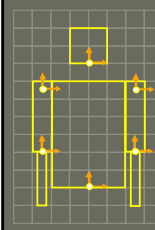
Week 3, Monday 15 Oct 03

© Tamara Munzner

5

Transformation Hierarchies recap

- Example



```
glTranslate3f(x,y,0);
glRotatef(theta,0,0,1);
DrawBody();
glPushMatrix();
glTranslate3f(0,7,0);
DrawHead();
glPopMatrix();
glPushMatrix();
glTranslate(2.5,5.5,0);
glRotatef(theta,0,0,1);
DrawUArm();
glTranslate(0,-3.5,0);
glRotatef(theta,0,0,1);
DrawLArm();
glPopMatrix();
... (draw other arm)
```

Week 3, Monday 15 Oct 03

© Tamara Munzner

6

Hierarchies Discussion

- advantages
 - often good control knobs
 - maintain structural constraints
- limitations
 - expressivity: not always the best controls
 - can't do closed kinematic chains
 - keep hand on hip
 - can't do other constraints
 - e.g. don't walk through walls

Week 3, Monday 15 Oct 03

© Tamara Munzner

7

Single Parameter: simple

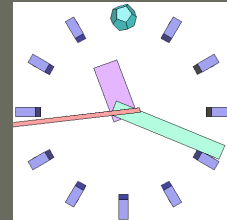
- parameters as functions of other params
 - clock: control all hands with seconds s

$$m = s/60, h = m/60,$$

$$\theta_{s} = (2 \pi s) / 60,$$

$$\theta_{m} = (2 \pi m) / 60,$$

$$\theta_{h} = (2 \pi h) / 60$$



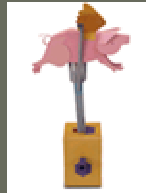
Week 3, Monday 15 Oct 03

© Tamara Munzner

8

Single Parameter: complex

- mechanisms not easily expressible with affine transforms



<http://www.flying-pig.co.uk>

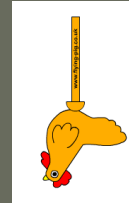
Week 3, Monday 15 Oct 03

© Tamara Munzner

9

Single Parameter: complex

- mechanisms not easily expressible with affine transforms



<http://www.flying-pig.co.uk/mechanisms/pages/irregular.html>

Week 3, Monday 15 Oct 03

© Tamara Munzner

10



University of British Columbia
CPSC 414 Computer Graphics

Display lists

Week 3, Monday 15 Oct 03

© Tamara Munzner

11

Display Lists

- Concept:
 - If multiple copies of an object are required, it can be compiled into a display list:

```
glNewList( listId, GL_COMPILE );
glBegin( ... );
... // geometry goes here
glEndList();
// render two copies of geometry offset by 1 in z-direction:
glCallList( listId );
glTranslatef( 0.0, 0.0, 1.0 );
glCallList( listId );
```

Week 3, Monday 15 Oct 03

© Tamara Munzner

12

Display Lists

- Advantages:
 - More efficient than individual function calls for every vertex/attribute
 - Can be cached on the graphics board (bandwidth!)
 - Display lists exist across multiple frames
 - Represent static objects in an interactive application

Week 3, Monday 15 Oct 03

© Tamara Munzner

13

Display Lists

- Example: 36 Snowmen
<http://www.lighthouse3d.com/opengl/displaylists>

efficiency issues

Week 3, Monday 15 Oct 03

© Tamara Munzner

14

drawSnowMan

```
void drawSnowMan() {  
    // Draw Eyes  
    glColor3f(1.0f, 1.0f, 1.0f);  
    glPushMatrix();  
    glColor3f(0.0f, 0.0f, 0.0f);  
    glTranslatef(0.05f, 0.10f, 0.18f);  
    glutSolidSphere(0.05f, 10, 10);  
    glTranslatef(-0.1f, 0.0f, 0.0f);  
    glutSolidSphere(0.05f, 10, 10);  
    glPopMatrix();  
  
    // Draw Body  
    glTranslatef(0.0f, 0.75f, 0.0f);  
    glutSolidSphere(0.75f, 20, 20);  
  
    // Draw Head  
    glTranslatef(0.0f, 1.0f, 0.0f);  
    glutSolidSphere(0.25f, 20, 20);  
  
    // Draw Nose  
    glColor3f(1.0f, 0.5f, 0.5f);  
    glRotatef(0.0f, 1.0f, 0.0f, 0.0f);  
    glutSolidCone(0.08f, 0.5f, 10, 2);  
}
```

Week 3, Monday 15 Oct 03

© Tamara Munzner

15

Snowmen: no lists

```
// Draw 36 Snowmen  
for(int i = -3; i < 3; i++)  
    for(int j = -3; j < 3; j++) {  
        glPushMatrix();  
        glTranslatef(i*10.0, j*10.0);  
        // Call the function to draw a snowman  
        drawSnowMan();  
        glPopMatrix();  
    }
```

Week 3, Monday 15 Oct 03

© Tamara Munzner

16

Display Lists

- Example: 36 Snowmen
<http://www.lighthouse3d.com/opengl/displaylists>

benchmarks of 36K polygons

– 55 FPS: no display list

Week 3, Monday 15 Oct 03

© Tamara Munzner

17

Snowmen: display lists

```
GLuint createDL() {  
    GLuint snowManDL;  
    // Create the id for the list  
    snowManDL = glGenLists(1);  
    // start list  
    glNewList(snowManDL, GL_COMPILE);  
    // call the function that contains the rendering commands  
    drawSnowMan();  
    // endList  
    glEndList();  
    return(snowManDL); }
```

Week 3, Monday 15 Oct 03

© Tamara Munzner

18

Snowmen: display lists

```
// Draw 36 Snowmen
for(int i = -3; i < 3; i++)
    for(int j=-3; j < 3; j++) {
        glPushMatrix();
        glTranslatef(i*10.0,0,j * 10.0);
        // Call the function to draw a snowman
        glCallList(Dlid);
        glPopMatrix();
    }
```

Week 3, Monday 15 Oct 03

© Tamara Munzner

19

Display Lists

- Example: 36 Snowmen
<http://www.lighthouse3d.com/opengl/displaylists>
benchmarks of 36K polygons
 - 55 FPS: no display list
 - 153 FPS: 1 snowman display list, called 36x

Week 3, Monday 15 Oct 03

© Tamara Munzner

20

Snowmen: one big list

```
GLuint createDL() {
    GLuint snowManDL;
    snowManDL = glGenLists(1);
    glNewList(snowManDL, GL_COMPILE);
    for(int i = -3; i < 3; i++)
        for(int j=-3; j < 3; j++) {
            glPushMatrix();
            glTranslatef(i*10.0,0,j * 10.0);
            drawSnowMan();
            glPopMatrix();
        }
    glEndList();
    return(snowManDL);
}
```

Week 3, Monday 15 Oct 03

© Tamara Munzner

21

Display Lists

- Example: 36 Snowmen
<http://www.lighthouse3d.com/opengl/displaylists>
benchmarks of 36K polygons
 - 55 FPS: no display list
 - 153 FPS: 1 snowman display list, called 36x
 - 108 FPS: single 36 snowman display list

Week 3, Monday 15 Oct 03

© Tamara Munzner

22

Snowmen: nested lists

```
GLuint createDL() {
    GLuint snowManDL, loopDL;
    snowManDL = glGenLists(1);
    loopDL = glGenLists(1);
    glNewList(snowManDL, GL_COMPILE);
    drawSnowMan();
    glEndList();
    glNewList(loopDL, GL_COMPILE);
    for(int i = -3; i < 3; i++)
        for(int j=-3; j < 3; j++) {
            glPushMatrix();
            glTranslatef(i*10.0,0,j * 10.0);
            glCallList(snowManDL);
            glPopMatrix();
        }
    glEndList();
    return(loopDL);
}
```

Week 3, Monday 15 Oct 03

© Tamara Munzner

23

Display Lists

- Example: 36 Snowmen
<http://www.lighthouse3d.com/opengl/displaylists>
benchmarks of 36K polygons
 - 55 FPS: no display list
 - 153 FPS: 1 snowman display list, called 36x
 - 108 FPS: single 36 snowman display list
 - 153 FPS: nested display lists

Week 3, Monday 15 Oct 03

© Tamara Munzner

24

Animating

- redraw entire screen each frame
 - clear to background color
 - traverse entire transformation stack
 - draw all geometry
- problem: flicker
 - <http://www.cs.unc.edu/~davemc/Class/136>
- solution: double buffering

Week 3, Monday 15 Oct 03

© Tamara Munzner

25

Double buffering

- two buffers, front and back
 - while front is on display, draw into back
 - when drawing finished, swap the two
- GLUT
 - `glutInitDisplayMode(GLUT_DOUBLE);`
 - `glutSwapBuffers();`
- avoids flicker, but could be slow

Week 3, Monday 15 Oct 03

© Tamara Munzner

26