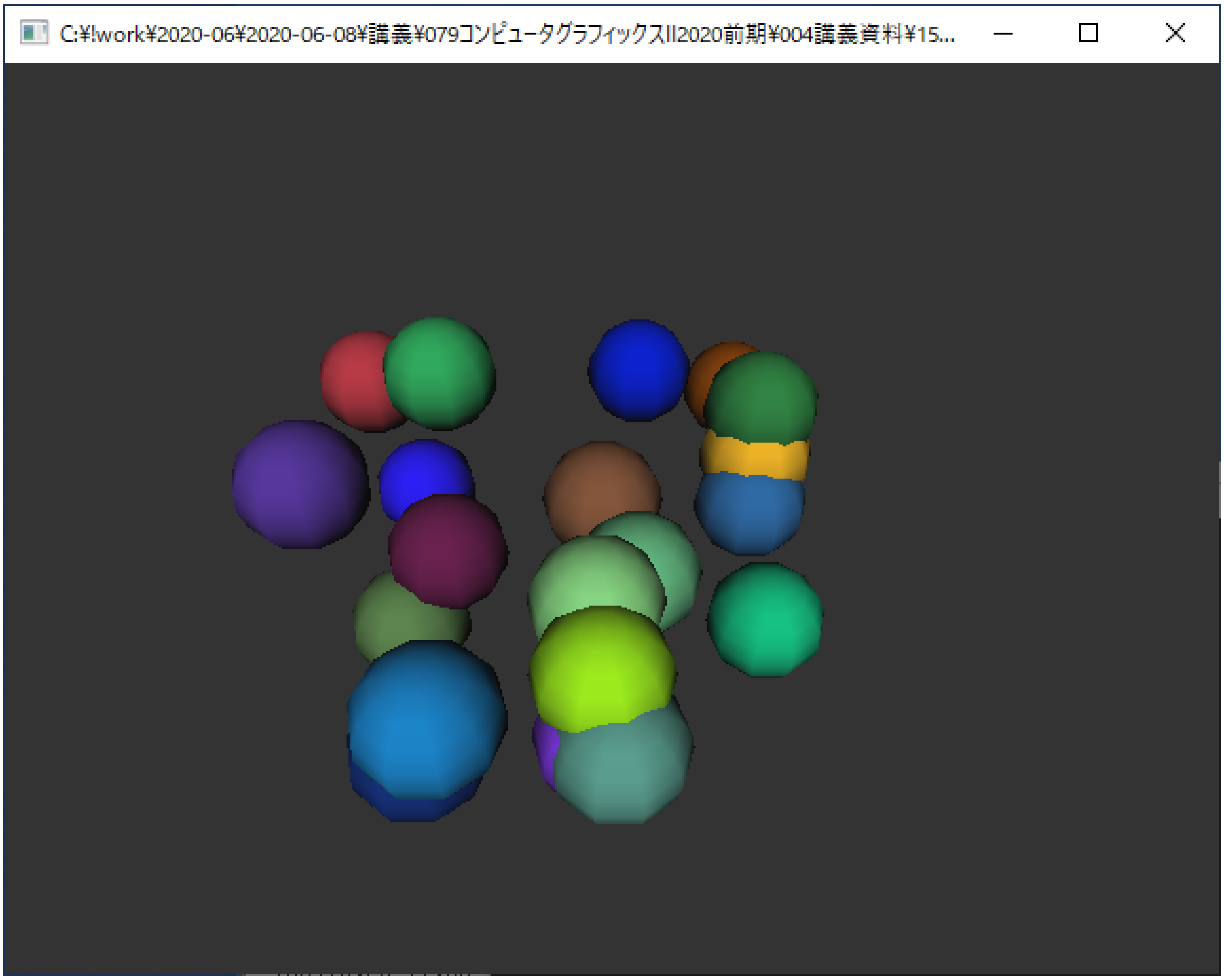
物体を取得

- マウスでクリックした物体を取得する


C:¥!work¥2020-06¥2020-06-08¥講義¥079コンピュータグラフィックスII2020前期¥004講義資料¥15...

サンプルソースコード

```
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include <GL/glut.h>

int winw, winh;
const int BALLNUM = 20;
double ballpos[BALLNUM][3];
float ballcol[BALLNUM][4];

void myDraw()
{
    int i;

    glClearColor(0.2f, 0.2f, 0.2f, 1.0f);
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glEnable(GL_DEPTH_TEST);
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);

    gluPerspective(60.0, (double)winw / (double)winh, 0.1, 100.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(0.0, -10.0, 10.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0);

    glInitNames();

    for (i = 0; i < BALLNUM; i++) {
        glPushName(i);
        glPushMatrix();
        glTranslated(ballpos[i][0], ballpos[i][1], ballpos[i][2]);
        glMaterialfv(GL_FRONT, GL_DIFFUSE, ballcol[i]);
        glutSolidSphere(1.0, 10, 10);
        glPopMatrix();
        glPopName();
    }
}

void myDisplay()
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    myDraw();
    glutSwapBuffers();
}
```

サンプルソースコード

```
void myMouseFunc(int button, int state, int x, int y)
{
    const int SELECTMAX = 256;
    unsigned int selection[SELECTMAX];
    int viewport[4];
    unsigned int* ptr;
    int hits;
    int names;
    int i;
    int j;
    double z;
    double minz;
    int id;
    int minid;

    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
        glSelectBuffer(SELECTMAX, selection);
        glRenderMode(GL_SELECT);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        glGetIntegerv(GL_VIEWPORT, viewport);
        gluPickMatrix(x, winh - y, 1, 1, viewport);
        myDraw();
        glFlush();
        hits = glRenderMode(GL_RENDER);

        minid = -1;
        minz = 1.0;
        ptr = selection;
        for (i = 0; i < hits; i++) {
            names = *ptr;
            ptr++;
            z = (double)*ptr/(double)0xFFFFFFFF;
            ptr++;
            ptr++;
            if (names > 0) id = *ptr;
            else id = -1;
            for (j = 0; j < names; j++) ptr++;
            if (z < minz) {
                minz = z;
                minid = id;
            }
        }
        if (minid >= 0 && minid < BALLNUM) {
            ballcol[minid][0] = (float)rand() / (float)RAND_MAX;
            ballcol[minid][1] = (float)rand() / (float)RAND_MAX;
            ballcol[minid][2] = (float)rand() / (float)RAND_MAX;
        }

        glutPostRedisplay();
    }
}
void myKeyboard(unsigned char key, int x, int y)
{
    if (key == 0x1B) exit(0);
}

void myReshape(int width, int height)
{
    winw = width;
    winh = height;
    glViewport(0, 0, winw, winh);
}
```

サンプルソースコード

```
int main(int argc, char* argv[])
{
    int i;

    srand((unsigned)time(NULL));
    for (i = 0; i < BALLNUM; i++) {
        ballpos[i][0] = 10.0 * (double)rand() / (double)RAND_MAX - 5.0;
        ballpos[i][1] = 10.0 * (double)rand() / (double)RAND_MAX - 5.0;
        ballpos[i][2] = 4.0 * (double)rand() / (double)RAND_MAX - 2.0;
        ballcol[i][0] = (float)rand() / (float)RAND_MAX;
        ballcol[i][1] = (float)rand() / (float)RAND_MAX;
        ballcol[i][2] = (float)rand() / (float)RAND_MAX;
        ballcol[i][3] = 1.0f;
    }

    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE | GLUT_DEPTH);
    glutInitWindowSize(640, 480);
    glutInitWindowPosition(0, 0);
    glutCreateWindow(argv[0]);
    glutKeyboardFunc(myKeyboard);
    glutMouseFunc(myMouseFunc);
    glutReshapeFunc(myReshape);
    glutDisplayFunc(myDisplay);
    glutMainLoop();
    return 0;
}
```