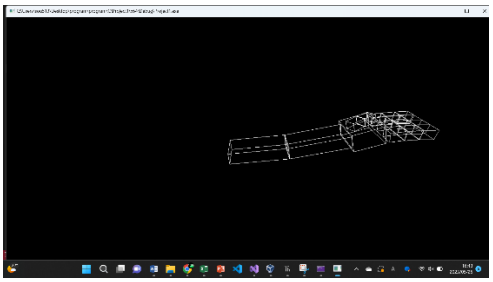
コンピュータグラフィック前半レポート

プログラムで工夫したところなど、考察

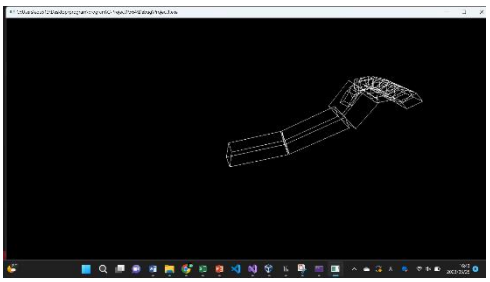
今回の課題では、ロボットアームを改造して手と指を作成した。出力としては作成した手を使ってじゃんけんの手（グー、チョキ、パー）を描画させた。工夫したところとしては２つある。１つ目は指の関節数で合計 14 もの関節を作ったところだ。指の関節を作るにあたって glPushMatrix()と glPopMatrix()をネスト構造にする必要性があり苦労した。そして、実際の手では親指のみ他の指と付いている向きが違うと考え、プログラム内では親指の第１関節までを glRotated()で回転させ調整した。２つ目の工夫したところは描画するところのプログラムで、time ライブラリをインポートしてプログラム実行後の経過時間（秒を）計測し、物体が回転して移動する時間を設定できるようにした。物体の回転では、回転する時間と fin_10 = (fin_10 - 3) % 360 の３の値を変更することで、時間あたりに回転する角度を増やして調整した。また、チョキの手やパーの手では指を開いて指の間に間隔を持たせたかったので各指の付け根にのみ２つ目の変数を設けて動かした。

今回の課題では２つの問題点があると考えた。１つ目の問題として、指の関節を 14 も作ったが、じゃんけんのような単調な動きではすべてを活かすことができなかったように感じた。なので後半のレポートではハンドシュミレータのような複雑なものを作るのもいいなと考えた。２つ目は、動きの滑らかさである。このプログラムでは先に説明したように time ライブラリを使って経過時間で動きを時間指定したのだが、time ライブラリの時間の単位で秒が最小であるためミリ秒単位での細かい時間指定ができなかった。そこでサンプリングタイムの値を大きくして描画の更新頻度を遅くしたのだが、そのせいで動きにカクつきが出てしまった。これも第２回目の課題では改善したいと思う。

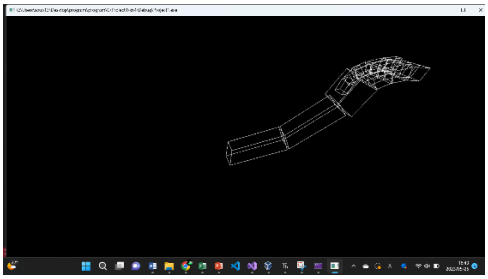
出力結果（動きがわかるように、数枚の絵を提示する）



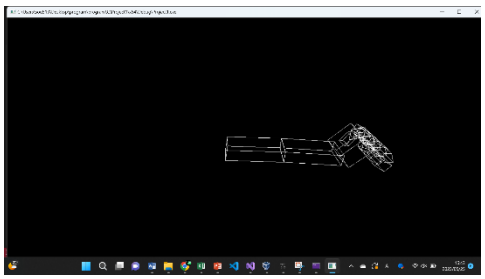
グー１



グー３



グー２



グー４

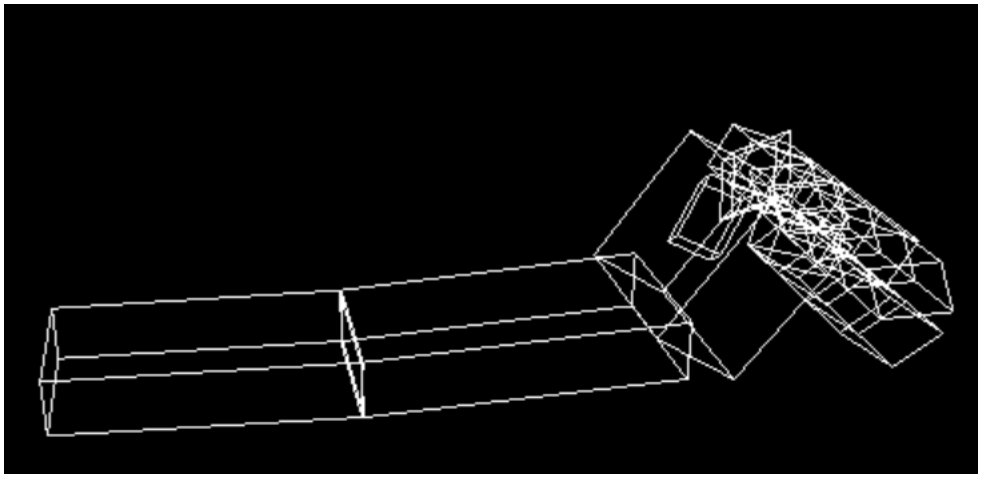
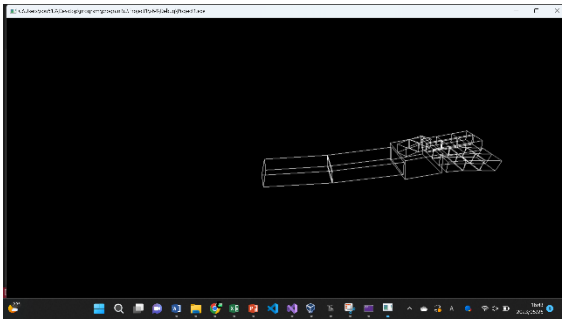
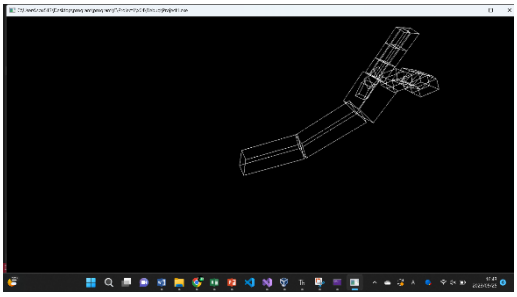


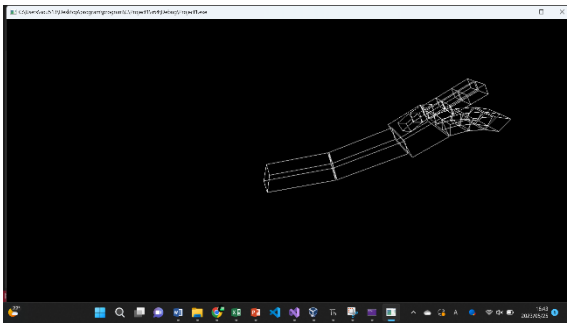
図 1 .グー



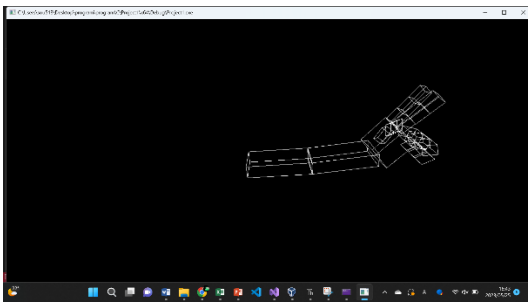
チョキ 1



チョキ 3



チョキ 2



チョキ 4

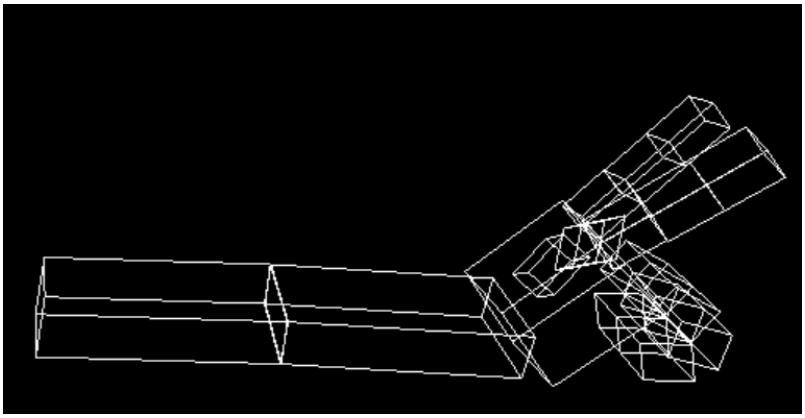
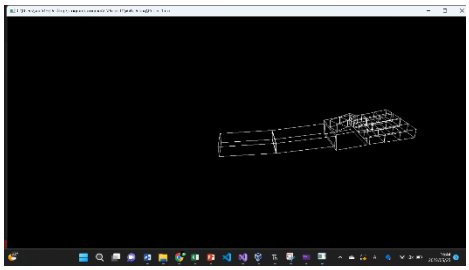
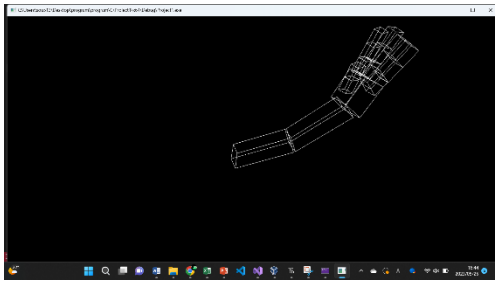


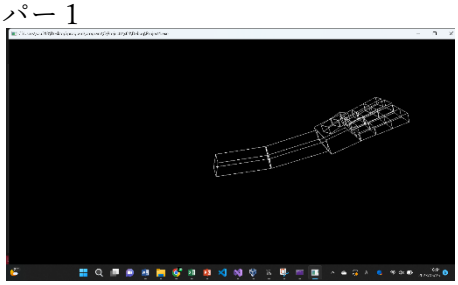
図 2.チョキ



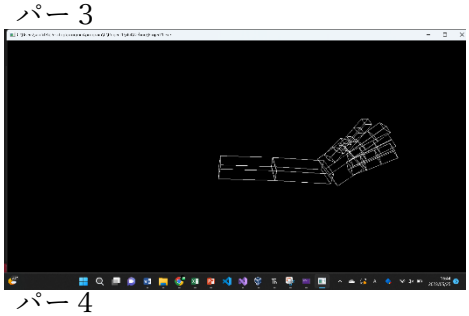
パー 1



パー 3



パー 2



パー 4

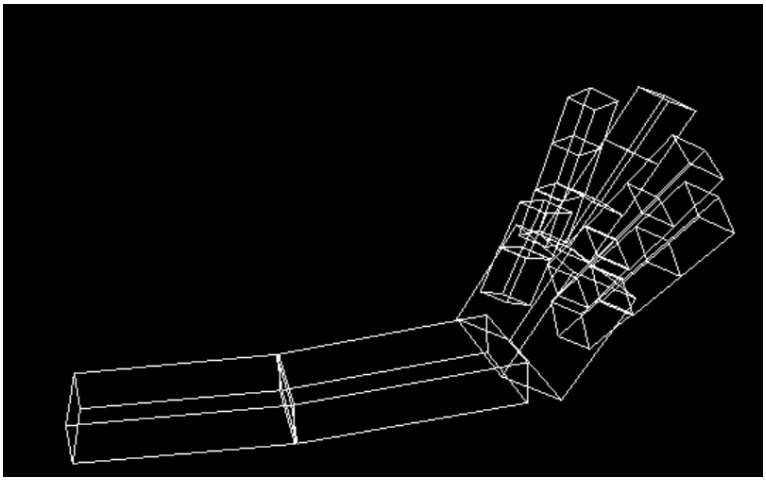


図 3.パー

4. ソース

```
/* List: p3-robot.c
 * Robot arm with two links and two joints.
 * Examination of world-local coordinates, modeling transfomation and
 * operation of matrix stack.
 */
#include <stdlib.h>
#include <GL/glut.h>
#include <math.h>
#include <time.h>
#include <stdio.h>
int     timenow=0,samplingTime = 200;
time_t start_time;
int rps_type;
static int shoulder = 0, elbow = 0,hand=0, fin_4 = 0, fin_5 = 0, fin_6 = 0,fin_7 = 0, fin_8 = 0, fin_9 = 0
, fin_10 = 0, fin_11 = 0 , fin_12 = 0 , fin_13 = 0, fin_14 = 0 , fin_15 = 0 , fin_16 = 0, fin_17 = 0,fin_4_y=0,fin_7_y=0
, fin_10_y = 0, fin_13_y = 0;


void myInit(char *progname)
{
   glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA);
   glutInitWindowSize(1600, 900);
   glutInitWindowPosition(0, 0);
   glutCreateWindow(progname);
   glClearColor(0.0, 0.0, 0.0, 0.0);
}
```

```
        glutWireCube(1.0);
            glPushMatrix();/* 6th link *人指し指爪*/
            glTranslated(0.5, 0.0, 1.0);
            glRotated((double)fin_6, 0.0, 0.0, 1.0);
            glTranslated(0.5, 0.0, -1.0);
            glutWireCube(1.0);
            glPopMatrix();
        glPopMatrix();
    glPopMatrix();
/* 7th link *中指第2関節まで*/
    glPushMatrix();
    glTranslated(0.5, 0.0, -0.5);
    glRotated((double)fin_7, 0.0, 0.0, 1.0);
    glRotated((double)fin_7_y, 0.0, 1.0, 0.0);
    glTranslated(0.25, 0.0, 0.3);
    glScaled(0.45, 0.3, 0.4);
    glutWireCube(1.0);
        glPushMatrix();/* 8th link *中指第1関節まで*/
        glTranslated(0.5, 0.0, 0.5);
        glRotated((double)fin_8, 0.0, 0.0, 1.0);
        glTranslated(0.5, 0.0, -0.45);
        glutWireCube(1.0);
            glPushMatrix();/* 9th link *中指爪*/
            glTranslated(0.5, 0.0, 1.0);
            glRotated((double)fin_9, 0.0, 0.0, 1.0);
            glTranslated(0.5, 0.0, -1.0);
            glutWireCube(1.0);
            glPopMatrix();
        glPopMatrix();
    glPopMatrix();
/* 10th link *薬指第2関節まで*/
    glPushMatrix();
    glTranslated(0.5, 0.0, -0.5);
    glRotated((double)fin_10, 0.0, 0.0, 1.0);
    glRotated((double)fin_10_y, 0.0, 1.0, 0.0);
    glTranslated(0.25, 0.0, 0.7);
    glScaled(0.45, 0.3, 0.4);
    glutWireCube(1.0);
        glPushMatrix();/* 11th link *薬指第1関節まで*/
        glTranslated(0.5, 0.0, 0.5);
        glRotated((double)fin_11, 0.0, 0.0, 1.0);
```

```
        glTranslated(0.5, 0.0, -0.45);
        glutWireCube(1.0);
            glPushMatrix();/* 12th link *薬指爪*/
            glTranslated(0.5, 0.0, 1.0);
            glRotated((double)fin_12, 0.0, 0.0, 1.0);
            glTranslated(0.5, 0.0, -1.0);
            glutWireCube(1.0);
            glPopMatrix();
        glPopMatrix();
    glPopMatrix();
/* 13th link *小指第2関節まで*/
    glPushMatrix();
    glTranslated(0.5, 0.0, -0.5);
    glRotated((double)fin_13, 0.0, 0.0, 1.0);
    glRotated((double)fin_13_y, 0.0, 1.0, 0.0);
    glTranslated(0.25, 0.0, 1.1);
    glScaled(0.45, 0.3, 0.4);
    glutWireCube(1.0);
        glPushMatrix();/* 14th link *小指第1関節まで*/
        glTranslated(0.5, 0.0, 0.5);
        glRotated((double)fin_14, 0.0, 0.0, 1.0);
        glTranslated(0.5, 0.0, -0.45);
        glutWireCube(1.0);
            glPushMatrix();/* 15th link *小指爪*/
            glTranslated(0.5, 0.0, 1.0);
            glRotated((double)fin_15, 0.0, 0.0, 1.0);
            glTranslated(0.5, 0.0, -1.0);
            glutWireCube(1.0);
            glPopMatrix();
        glPopMatrix();
    glPopMatrix();
/* 16th link *親指第一関節まで*/
    glTranslated(0.3, 0.0, 0.1);
    glPushMatrix();
    glTranslated(0.5, 0.0, -0.5);
    glRotated(20.0, 0.0, 1.0, 0.0);
    glRotated((double)fin_16, 1.0, 0.0, 0.0);
    glTranslated(-0.6, 0.0, -0.5);
```

```
    glScaled(0.45, 0.3, 0.4);
    glutWireCube(1.0);
        glPushMatrix();/* 17th link *親指爪*/
        glTranslated(0.5, 0.0, 0.5);
        glRotated((double)fin_17, 0.0, 0.0, 1.0);
        glTranslated(0.5, 0.0, -0.45);
        glutWireCube(1.0);
        glPopMatrix();
    glPopMatrix();
    glPopMatrix();
    glPopMatrix();
    glutSwapBuffers();
}

void myReshape (int width, int height)
{
    glViewport(0, 0, width, height);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(60.0, (double)width/(double)height, 0.1, 20.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    //glTranslated( -2.0, 3.0, -5.0);
    gluLookAt(0.0, 4.0, 7.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);  // move to enable viewing
}

void myTimer(int value)
{

    if (value == 1)
    {
        if (rps_type == 0) {//グー
            timenow = time(NULL) - start_time;
            glutTimerFunc(samplingTime, myTimer, 1);
            if (timenow <= 2) {
                shoulder = (shoulder + 2) % 360;//肩を振り上げる
                elbow = (elbow + 2) % 360;//肘を振り上げる
                glutPostRedisplay();
```

```
            }
            if (timenow > 3 && timenow <= 5) {
                shoulder = (shoulder - 2) % 360;//肩を振り下げる
                elbow = (elbow - 2) % 360;//肘を振り下げる
                glutPostRedisplay();
            }
            if (timenow <= 5) {
                hand = (hand + 2) % 360;//手を上げる
                fin_4 = (fin_4 - 3) % 360;//人差し指曲げる
                fin_5 = (fin_5 - 2) % 360;
                fin_6 = (fin_6 -1) % 360;
                fin_7 = (fin_7 - 3) % 360;//中指曲げる
                fin_8 = (fin_8 - 2) % 360;
                fin_9 = (fin_9 - 2) % 360;
                fin_10 = (fin_10 - 3) % 360;//薬指曲げる
                fin_11 = (fin_11 - 2) % 360;
                fin_12 = (fin_12 - 2) % 360;
                fin_13 = (fin_13 - 3) % 360;//小指曲げる
                fin_14 = (fin_14 - 2) % 360;
                fin_15 = (fin_15 - 2) % 360;
                fin_16 = (fin_16 - 1) % 360;//親指曲げる
                fin_17 = (fin_17 - 2) % 360;
                glutPostRedisplay();
            }

        }
        if (rps_type == 1) {//チョキ

            timenow = time(NULL) - start_time;
            glutTimerFunc(samplingTime, myTimer, 1);
            if (timenow <= 2) {
                shoulder = (shoulder + 2) % 360;//肩を振り上げる
                elbow = (elbow + 2) % 360;//肘を振り上げる
                glutPostRedisplay();
            }
            if (timenow > 3 && timenow <= 5) {
                shoulder = (shoulder - 2) % 360;//肩を振り下げる
                elbow = (elbow - 2) % 360;//肘を振り下げる
```

```
223                 fin_4_y = (fin_4_y + 1) % 360;//人差し指の第2関節までを開く
224                 fin_7_y = (fin_7_y - 1) % 360;//中指の第2関節までを開く
225                 glutPostRedisplay();
226             }
227
228             if (timenow <= 5) {//kusu
229                 hand = (hand + 2) % 360;//手を上げる
230                 fin_10 = (fin_10 - 3) % 360;//薬指曲げる
231                 fin_11 = (fin_11 - 3) % 360;
232                 fin_12 = (fin_12 - 3) % 360;
233                 fin_13 = (fin_13 - 3) % 360;//小指曲げる
234                 fin_14 = (fin_14 - 3) % 360;
235                 fin_15 = (fin_15 - 3) % 360;
236                 fin_16 = (fin_16 - 1) % 360;//親指曲げる
237                 fin_17 = (fin_17 - 2) % 360;
238                 glutPostRedisplay();
239             }
240         }
241         if (rps_type == 2) {//パー
242             timenow = time(NULL) - start_time;
243             glutTimerFunc(samplingTime, myTimer, 1);
244             if (timenow <= 2) {
245                 shoulder = (shoulder + 2) % 360;//肩を振り上げる
246                 elbow = (elbow + 2) % 360;//肘を振り上げる
247                 glutPostRedisplay();
248             }
249             if (timenow <= 3) {
250                 fin_10_y = (fin_10_y - 1) % 360;//薬指を開く
251                 glutPostRedisplay();
252             }
253             if (timenow > 3 && timenow <= 5) {
254                 shoulder = (shoulder - 2) % 360;//肩を振り下げる
255                 elbow = (elbow - 2) % 360;//肘を振り下げる
256                 glutPostRedisplay();
257             }
258             if (timenow <= 4) {
259                 fin_13_y = (fin_13_y + -1) % 360;//小指を開く
260                 glutPostRedisplay();
```

```
261             }
262             if (timenow <= 5) {
263                 hand = (hand + 2) % 360;//手を上げる
264                 fin_4_y = (fin_4_y + 1) % 360;//人差し指を開く
265                 glutPostRedisplay();
266             }
267
268         }
269         glutPostRedisplay();
270     }
271
272 }
273
274
275 int main(int argc, char** argv){
276     start_time = time(NULL);
277     rps_type = 2;//0の場合グー、１の場合チョキ、２の場合パー
278     glutInit(&argc, argv);
279     myInit(argv[0]);
280     glutTimerFunc(samplingTime, myTimer, 1);
281     glutReshapeFunc(myReshape);
282     glutDisplayFunc(myDisplay);
283     glutMainLoop();
284     return 0;
285 }
286
```