108(上)電腦圖學 作業二 機器人大變身

資工三乙 406262216 劉品萱

程式架構

機器人的身體結構主要為帽子、頭、身體、四肢，而搭配動作 Boxing 及 Lifting dumbbells 還有另外繪製拳擊手套和啞鈴。而機器人的動作包含原有可以控制各部位細節角度的功能，還額外添加了四個動作，分別為：坐下翹腳、拳擊、舉啞鈴、開合跳。

整個程式的函式架構為以下：

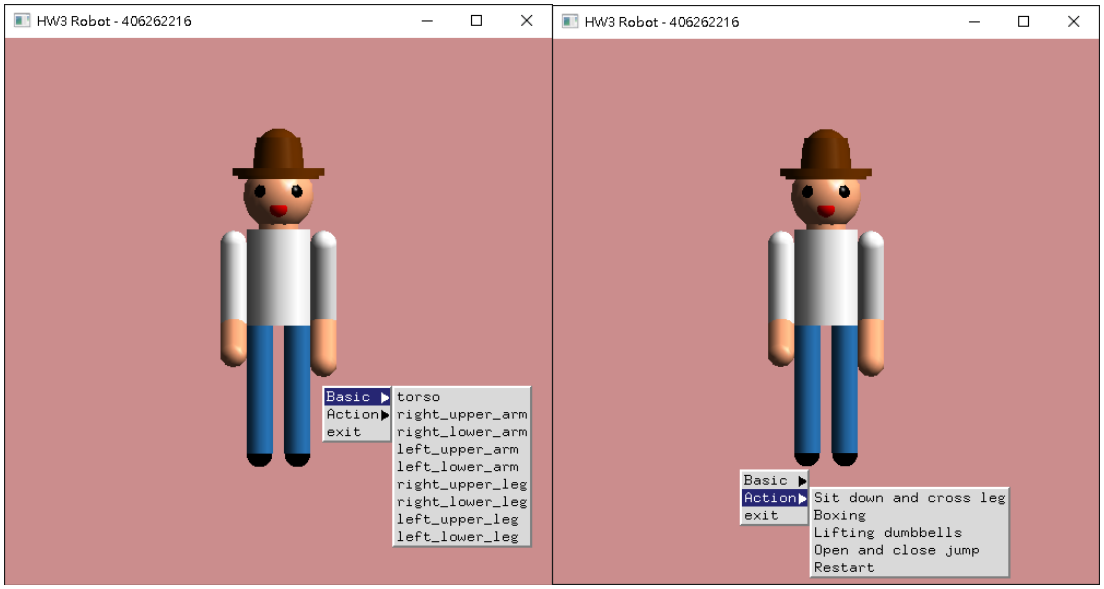
1. 用來繪畫機器人身體結構的 10 個函式
   void torso()、void head()、void left_upper_arm()、void left_lower_arm()、void right_upper_arm()、void right_lower_arm()、void left_upper_leg()、void left_lower_leg()、void right_upper_leg()、void right_lower_leg()
2. 處理當 Action 子選單點擊來更換動作
   void ActionChange(int value)
3. 處來 glut 一些設定的函式
   void myReshape(int w, int h)、void mouse(int btn, int state, int x, int y)、void display(void)
4. 處理選單選項的函數
   void MainMenu(int id)、void BasicMenu(int id)、void ActionMenu(int id)
5. 光源、材質、背景顏色等的初始化函式
   void myinit()
6. 主程式
   int main(int argc, char **argv)

討論

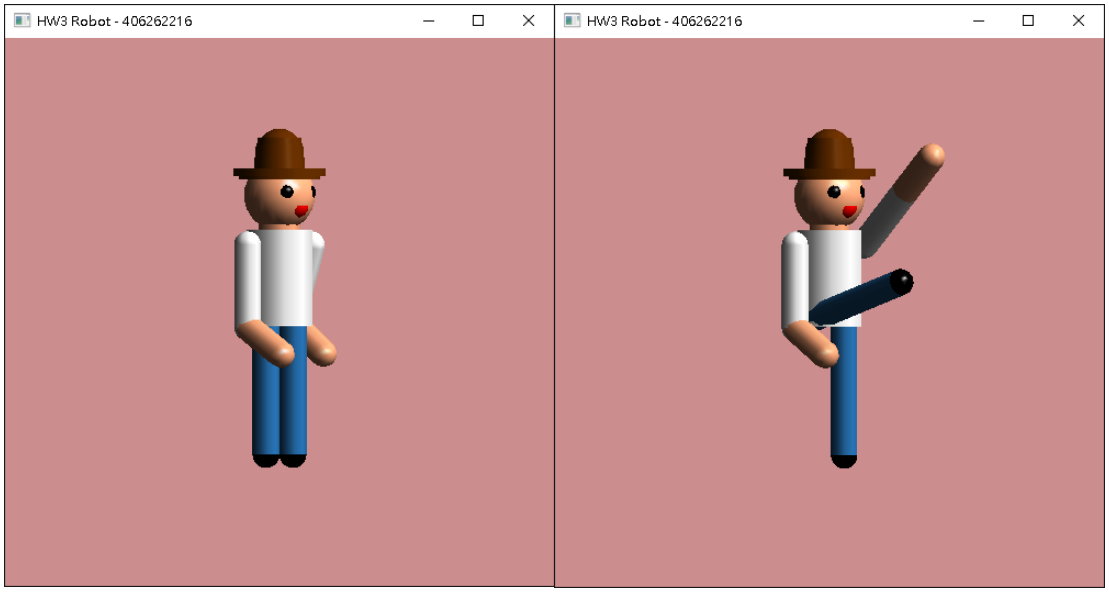
此次作業整體相較於第一次的程式作業 - 參數化線性軸曲面設計容易許多，花費在搞懂 Hierarchical Objects 3D 繪圖的時間較多，但還好上課有認真聽所以不至於看不懂。在實作上沒有太大的問題，將機器人的身體架構建構出來其實不難，主要是機器人的動作及旋轉角度前前後後調整了好幾次，且花了一點時間研究 openGL 的計時器功能，後來翻了參考資料並上網看了一些教學文才找到並理解 glutTimerFunc 的用法。

執行畫面

可以透過按下滑鼠中鍵來開啟選單，選單分為 Basic、Action、exit。

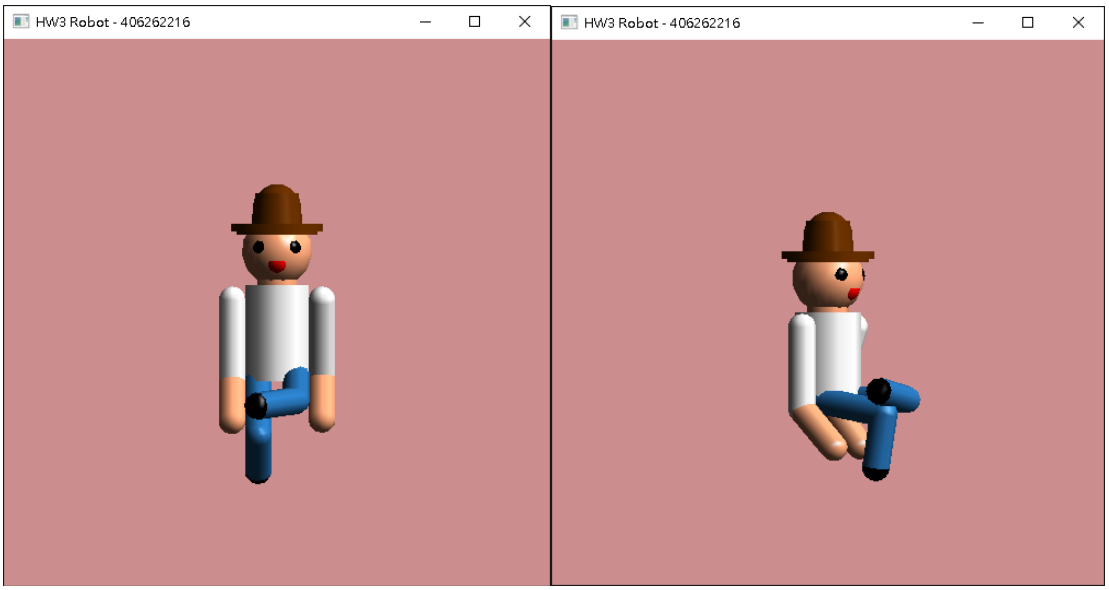


子選單 Basic 下的選項能夠利用滑鼠的左右鍵來針對機器人的各個部位做細部微調。例如 torso 可以轉動機器人的全身。Right_upper_arm 可以針對右手根部轉動、Left_lower_leg 可以針對左腳的小腿轉動等。

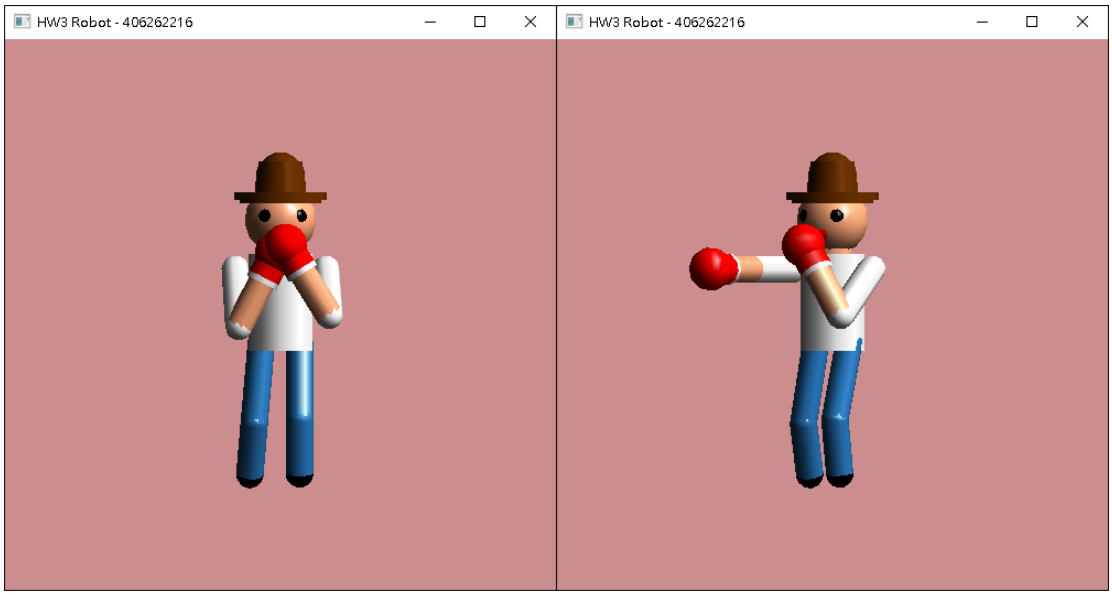


而子選單下的選項則是可以選擇機器人的動作。

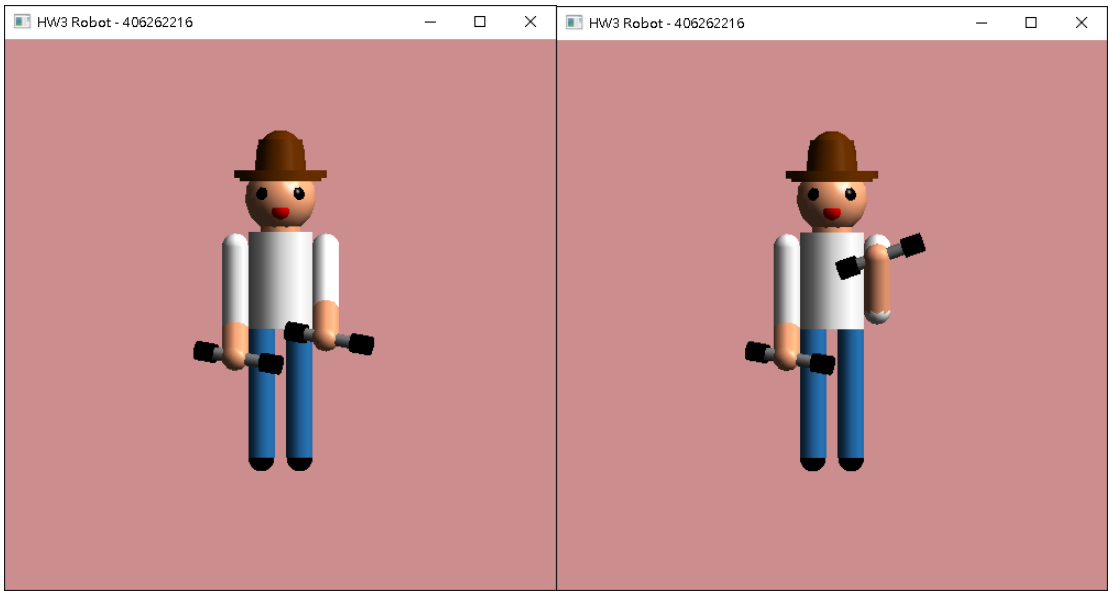
動作 1：坐下並翹腳（Sit down and cross leg）



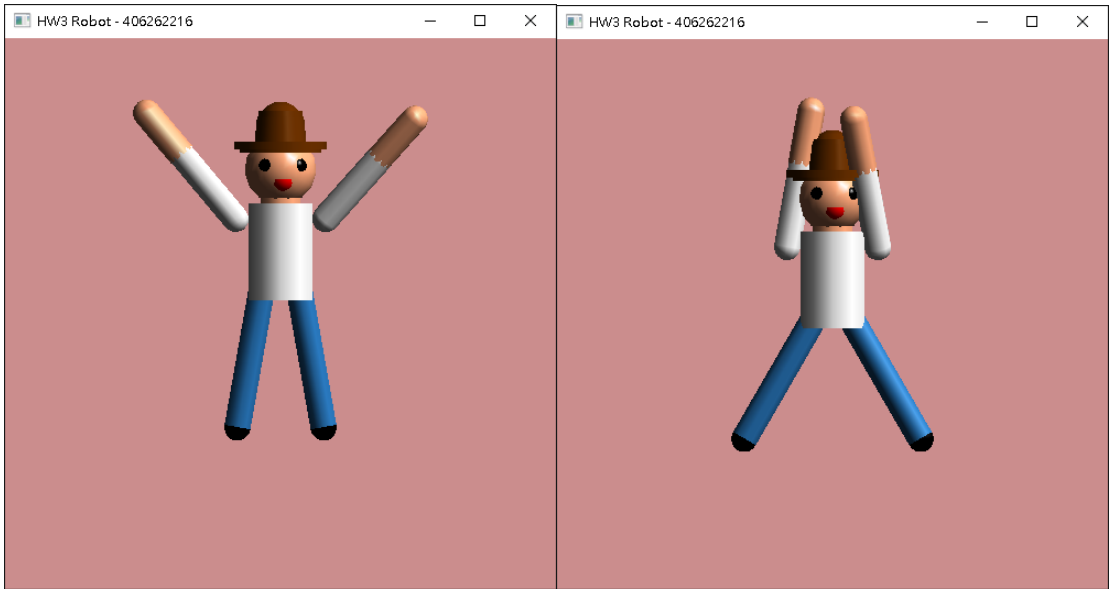
動作二：拳擊（Boxing）



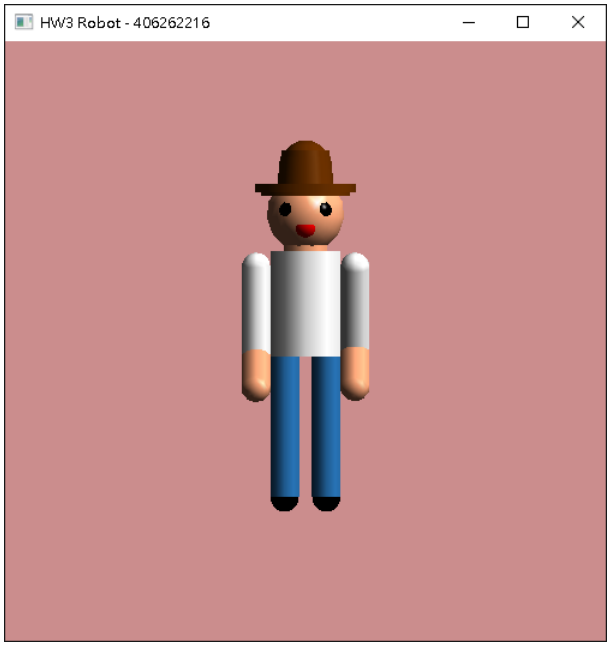
動作三：舉啞鈴（Lifting dumbbells）



動作四：開合跳（Open and close jump）



動作五：回復原始動作（Restart）



程式碼

```
1.  //
2.  //  main.cpp
3.  //  HW2
4.  //
5.  //  Created by 劉品萱 on 2019/12/14.
6.  //  Copyright © 2019 劉品萱. All rights reserved.
7.  //
8.
9.  #include <cstdlib>
10. #include <cmath>
11. #include <cstdio>
12. #include <GL/glut.h>
13. #include <GL/glui.h>
14. #include <ctime>
15.
16. int SelectActionNum;
17. int timer = 0;
18.
19. #define TORSO_HEIGHT 3.5
20. #define UPPER_ARM_HEIGHT 3.0
21. #define LOWER_ARM_HEIGHT 2.0
22. #define UPPER_LEG_RADIUS  0.5
```

```
23. #define LOWER_LEG_RADIUS    0.5
24. #define LOWER_LEG_HEIGHT 2.0
25. #define UPPER_LEG_HEIGHT 3.0
26. #define UPPER_LEG_RADIUS    0.5
27. #define TORSO_RADIUS 0.75
28. #define UPPER_ARM_RADIUS    0.5
29. #define LOWER_ARM_RADIUS    0.5
30. #define HEAD_HEIGHT 1.5
31. #define HEAD_RADIUS 1.0
32. #define TShirt_RADIUS 1.2
33.
34. GLfloat Squat = 0.0;
35.
36. typedef float point[3];
37.
38. static GLfloat theta[11] = {0.0, 0.0, 0.0, 20.0,-60.0, 20.0, -60.0, 180.0, 0.0, 180.0,
    0.0};
39. GLfloat ActionTheta[11] = {0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0};
40. GLfloat ActionPosi[24] = { 0.0,        //右手 軸肢 X 軸 ActionPosi[0]
41.                           0.0,        //右手 軸肢 Y 軸 ActionPosi[1]
42.                           0.0,        //右手 軸肢 Z 軸 ActionPosi[2]
43.                           0.0,        //右手 前肢 X 軸 ActionPosi[3]
44.                           0.0,        //右手 前肢 Y 軸 ActionPosi[4]
45.                           0.0,        //右手 前肢 Z 軸 ActionPosi[5]
46.                           180.0,      //左手 軸肢 X 軸 ActionPosi[6]
47.                           0.0,        //左手 軸肢 Y 軸 ActionPosi[7]
48.                           0.0,        //左手 軸肢 Z 軸 ActionPosi[8]
49.                           0.0,        //左手 前肢 X 軸 ActionPosi[9]
50.                           0.0,        //左手 前肢 Y 軸 ActionPosi[10]
51.                           0.0,        //左手 前肢 Z 軸 ActionPosi[11]
52.                           0.0,        //右腳 軸肢 X 軸 ActionPosi[12]
53.                           0.0,        //右腳 軸肢 Y 軸 ActionPosi[13]
54.                           0.0,        //右腳 軸肢 Z 軸 ActionPosi[14]
55.                           0.0,        //右腳 前肢 X 軸 ActionPosi[15]
56.                           0.0,        //右腳 前肢 Y 軸 ActionPosi[16]
57.                           0.0,        //右腳 前肢 Z 軸 ActionPosi[17]
58.                           0.0,        //左腳 軸肢 X 軸 ActionPosi[18]
59.                           0.0,        //左腳 軸肢 Y 軸 ActionPosi[19]
60.                           0.0,        //左腳 軸肢 Z 軸 ActionPosi[20]
61.                           0.0,        //左腳 前肢 X 軸 ActionPosi[21]
```

```
62.                                 0.0,        //左腳 前肢 Y 軸 ActionPosi[22]
63.                                 0.0         //左腳 前肢 Z 軸 ActionPosi[23]
64.                             };
65. static GLint angle = 2;
66.
67. GLUquadricObj *t, *h, *lua, *lla, *rua, *rla, *lll, *rll, *rul, *lul, *quadratic;
68.
69. double size=1.0;
70.
71. void torso()
72. {
73.     glPushMatrix();
74.     glRotatef(-90.0, 1.0, 0.0, 0.0);
75.     glColor3f(0.996, 0.655, 0.478);
76.     gluCylinder(t, TORSO_RADIUS, TORSO_RADIUS, TORSO_HEIGHT, 10, 10);
77.     glColor3f(1.0, 1.0, 1.0);
78.     glTranslatef(0.0, 0.0,-0.5);
79.     gluCylinder(t, TShirt_RADIUS, TShirt_RADIUS, TORSO_HEIGHT, 10, 10);
80.     glPopMatrix();
81. }
82.
83. void head()
84. {
85.     // 畫頭
86.     glPushMatrix();
87.       glTranslatef(0.0, 0.5 * HEAD_HEIGHT, 0.0);
88.       glColor3f(0.996, 0.655, 0.478);
89.       gluSphere(h, 1.3, 15, 15);
90.                                   /*void gluSphere( GLUquadricObj *qobj, //
    繪圖對象
91.                                                   GLdouble radius,       //
    球體半徑
92.                                                   GLint slices,          // Z 軸
    方向片數，經度方向
93.                                                   GLint stacks)          // Z
    軸方向片數，維度方向*/
94.       glPopMatrix();
95.
96.       // 眼睛
97.       glPushMatrix();
```

```
98.       glTranslatef(0.5, 0.85 * HEAD_RADIUS, HEAD_RADIUS * 0.8);
99.       glColor3f(0.0, 0.0, 0.0);
100.    // 右眼
101.      gluSphere(h, 0.4, 20, 20);
102.      glTranslatef(-1.0, 0.0, 0.0);
103.    // 左眼
104.      gluSphere(h, 0.4, 20, 20);
105.      glPopMatrix();
106.
107.    GLdouble CutUp[4] = { 0.0, -1.0, 0.0, 0.0 };
108.    glPushMatrix();
109.    glTranslatef(0.0, -0.3*HEAD_RADIUS+0.7, 0.5*HEAD_RADIUS+0.3);
110.    glColor3f(1.0, 0.0, 0.0);
111.    glClipPlane(GL_CLIP_PLANE0, CutUp);
112.    glEnable(GL_CLIP_PLANE0);
113.    gluSphere(h, 0.5, 20, 20);
114.    glDisable(GL_CLIP_PLANE0);
115.    glPopMatrix();
116.
117.
118.      // 帽子
119.      glPushMatrix();
120.      glTranslatef(0.0, 1.15 * HEAD_RADIUS, 0.0);
121.      glRotatef(-90.0f,1.0f,0.0f,0.0f);
122.      glColor3f(0.396, 0.18, 0);
123.      gluCylinder(quadratic, 1.0f ,0.95f ,0.5f ,30.0f ,30.0f);
124.
125.      glTranslatef(0.0, 0.0,0.5);
126.      glColor3f(0.396, 0.18, 0);
127.      gluCylinder(quadratic, 0.95f ,0.9f ,0.5f ,30.0f ,30.0f);
128.
129.      glTranslatef(0.0, 0.0,0.5);
130.      glColor3f(0.396, 0.18, 0);
131.      gluCylinder(quadratic, 0.9f ,0.85f ,0.5f ,30.0f ,30.0f);
132.
133.      glTranslatef(0.0, 0.0,0.2);
134.      glColor3f(0.396, 0.18, 0);
135.    gluSphere(quadratic,0.85f,10,10);
136.      gluCylinder(quadratic, 0.85f ,0.8f ,0.5f ,30.0f ,30.0f);
137.
```

```
138.         glTranslatef(0.0, 0.0,-1);
139.         gluCylinder(quadratic, 1.5f ,1.5f ,0.3f ,30.0f ,30.0f);
140.     glTranslatef(0.0, 0.0, 0.1);
141.         gluCylinder(quadratic, 1.7f ,1.7f ,0.3f ,30.0f ,30.0f);
142.         glPopMatrix();
143. }
144.
145. // 我視角的左手 軸肢
146. void left_upper_arm()
147. {
148.     glPushMatrix();
149.         glColor3f(1, 1, 1);
150.         glRotatef(-90.0, 1.0, 0.0, 0.0);
151.         gluSphere(lua, UPPER_ARM_RADIUS, 10, 10);
152.         gluCylinder(lua, UPPER_ARM_RADIUS, UPPER_ARM_RADIUS,
    UPPER_ARM_HEIGHT, 10, 10);
153.         glTranslatef(0.0, 0.0, UPPER_ARM_HEIGHT);
154.         gluSphere(lua,UPPER_ARM_RADIUS,10,10);
155.         glPopMatrix();
156. }
157.
158. // 我視角的左手 前肢
159. void left_lower_arm()
160. {
161.     glPushMatrix();
162.     glColor3f(0.996, 0.655, 0.478);
163.     glRotatef(-90.0, 1.0, 0.0, 0.0);
164.     gluCylinder(lla,LOWER_ARM_RADIUS, LOWER_ARM_RADIUS,
    LOWER_ARM_HEIGHT,10,10);
165.     glTranslatef(0.0, 0.0, LOWER_ARM_HEIGHT);
166.     gluSphere(lla,LOWER_ARM_RADIUS,10,10);
167.
168.     if(SelectActionNum == 2)
169.     {
170.         // 拳擊手套
171.         glColor3f(1, 0, 0);
172.         gluCylinder(rla, 0.6, 0.6, 0.7, 10, 10);
173.         glTranslatef(0.0, 0.0, 1);
174.         gluSphere(lua,0.8,10,10);
175.         glTranslatef(0.0, 0.0, -1.2);
```

```
176.         glColor3f(1, 1, 1);
177.         gluCylinder(rla, 0.6, 0.6, 0.2, 10, 10);
178.     }
179.     if(SelectActionNum == 3)
180.     {
181.         // 手舉啞鈴
182.         glRotatef(70.0, 0.0, 1.0, 0.0);
183.         glTranslatef(0.0, 0.0, -1.5);
184.         glColor3f(0.514, 0.514, 0.514);
185.         gluCylinder(lla, 0.23, 0.23, 3, 10, 10);
186.         glColor3f(0, 0, 0);
187.         gluCylinder(lla, 0.35, 0.35, 0.75, 10, 10);
188.         glTranslatef(0.0, 0.0, 2.5);
189.         glColor3f(0, 0, 0);
190.         gluCylinder(lua, 0.35, 0.35, 0.75, 10, 10);
191.     }
192.     glPopMatrix();
193. }
194.
195. // 我的右手 軸肢
196. void right_upper_arm()
197. {
198.     glPushMatrix();
199.     glColor3f(1, 1, 1);
200.     glRotatef(-90.0, 1.0, 0.0, 0.0);
201.     gluSphere(lua, UPPER_ARM_RADIUS, 10, 10);
202.     gluCylinder(rua,UPPER_ARM_RADIUS, UPPER_ARM_RADIUS,
    UPPER_ARM_HEIGHT,10,10);
203.     glTranslatef(0.0, 0.0, UPPER_ARM_HEIGHT);
204.       gluSphere(lua,UPPER_ARM_RADIUS,10,10);
205.     glPopMatrix();
206. }
207.
208. // 我的右手 前肢
209. void right_lower_arm()
210. {
211.     glPushMatrix();
212.     glColor3f(0.996, 0.655, 0.478);
213.     glRotatef(-90.0, 1.0, 0.0, 0.0);
214.     gluCylinder(rla,LOWER_ARM_RADIUS, LOWER_ARM_RADIUS,
```

```
    LOWER_ARM_HEIGHT,10,10);
215.    // 讓手前肢末看起來不會是空心
216.    glTranslatef(0.0, 0.0, LOWER_ARM_HEIGHT);
217.      gluSphere(rla,LOWER_ARM_RADIUS,10,10);
218.    if(SelectActionNum == 2)
219.    {
220.        // 拳擊手套
221.        glColor3f(1, 0, 0);
222.        gluCylinder(rla, 0.6, 0.6, 0.7, 10, 10);
223.        glTranslatef(0.0, 0.0, 1);
224.        gluSphere(lua,0.8,10,10);
225.        glTranslatef(0.0, 0.0, -1.2);
226.        glColor3f(1, 1, 1);
227.        gluCylinder(rla, 0.6, 0.6, 0.2, 10, 10);
228.    }
229.    if(SelectActionNum == 3)
230.    {
231.        // 手舉啞鈴
232.        glRotatef(70.0, 0.0, 1.0, 0.0);
233.        glTranslatef(0.0, 0.0, -1.5);
234.        glColor3f(0.514, 0.514, 0.514);
235.        gluCylinder(rla, 0.23, 0.23, 3, 10, 10);
236.        glColor3f(0, 0, 0);
237.        gluCylinder(rla, 0.35, 0.35, 0.75, 10, 10);
238.        glTranslatef(0.0, 0.0, 2.5);
239.        glColor3f(0, 0, 0);
240.        gluCylinder(rla, 0.35, 0.35, 0.75, 10, 10);
241.    }
242.    glPopMatrix();
243. }
244. // 我的左腳 軸肢
245. void left_upper_leg()
246. {
247.    glPushMatrix();
248.    glColor3f(0.157, 0.451, 0.71);
249.    glRotatef(-90.0, 1.0, 0.0, 0.0);
250.    gluSphere(rla,LOWER_LEG_RADIUS,10,10);
251.    gluCylinder(lul,UPPER_LEG_RADIUS, UPPER_LEG_RADIUS,
    UPPER_LEG_HEIGHT,10,10);
252.    glPopMatrix();
```

```
253. }
254. // 我的左腳 前肢
255. void left_lower_leg()
256. {
257.     glPushMatrix();
258.     glColor3f(0.157, 0.451, 0.71);
259.     glRotatef(-90.0, 1.0, 0.0, 0.0);
260.     gluSphere(rla,LOWER_LEG_RADIUS,10,10);
261.     gluCylinder(lll,LOWER_LEG_RADIUS, LOWER_LEG_RADIUS,
    LOWER_LEG_HEIGHT,10,10);
262.     glTranslatef(0.0, 0.0, LOWER_LEG_HEIGHT);
263.     glColor3f(0, 0, 0);
264.       gluSphere(rla,LOWER_LEG_RADIUS,10,10);
265.     glPopMatrix();
266. }
267. // 我的右腳 軸肢
268. void right_upper_leg()
269. {
270.     glPushMatrix();
271.     glColor3f(0.157, 0.451, 0.71);
272.     glRotatef(-90.0, 1.0, 0.0, 0.0);
273.     gluSphere(rla,LOWER_LEG_RADIUS,10,10);
274.     gluCylinder(rul,UPPER_LEG_RADIUS, UPPER_LEG_RADIUS,
    UPPER_LEG_HEIGHT,10,10);
275.     glPopMatrix();
276. }
277. // 我的右腳 前肢
278. void right_lower_leg()
279. {
280.     glPushMatrix();
281.     glColor3f(0.157, 0.451, 0.71);
282.     glRotatef(-90.0, 1.0, 0.0, 0.0);
283.     gluSphere(rla,LOWER_LEG_RADIUS,10,10);
284.     gluCylinder(rll,LOWER_LEG_RADIUS, LOWER_LEG_RADIUS,
    LOWER_LEG_HEIGHT,10,10);
285.     glTranslatef(0.0, 0.0, LOWER_LEG_HEIGHT);
286.     glColor3f(0, 0, 0);
287.       gluSphere(rla,LOWER_LEG_RADIUS,10,10);
288.     glPopMatrix();
289. }
```

```
290.
291. void ActionChange(int value)
292. {
293.     if(SelectActionNum == 1) // Sit down and cross leg 坐下翹腳
294.     {
295.         Squat = 0;
296.         for(int i = 0; i < 11 ; i++)
297.             ActionTheta[i] = 0.0;
298.         for (int i = 0; i < 24; i++)
299.         {
300.             ActionPosi[i] = 0.0;
301.             if(i == 6)
302.                 ActionPosi[i] = 180.0;
303.         }
304.         timer = timer + 1;
305.         if(timer % 25 == 0 || timer % 25 == 1 || timer % 25 == 2 ||
306.             timer % 25 == 3 || timer % 25 == 4 || timer % 25 == 5)
307.         {
308.             Squat = -1.0;
309.             ActionPosi[12] = 0;
310.             ActionPosi[15] = 0.0;
311.             ActionPosi[18] = -20.0;
312.             ActionPosi[19] = -10.0;
313.             ActionPosi[20] = 20.0;
314.             ActionPosi[21] = 20.0;
315.             ActionPosi[22] = -30.0;
316.             ActionPosi[23] = 10.0;
317.         }
318.         if(timer % 25 == 6 || timer % 25 == 7 || timer % 25 == 8 ||
319.             timer % 25 == 9 || timer % 25 == 10 || timer % 25 == 11)
320.         {
321.             Squat = -2.0;
322.             ActionPosi[12] = -40.0;
323.             ActionPosi[15] = 90.0;
324.             ActionPosi[18] = -50.0;
325.             ActionPosi[19] = -90.0;
326.             ActionPosi[20] = 20.0;
327.             ActionPosi[21] = 50.0;
328.             ActionPosi[22] = -30.0;
329.             ActionPosi[23] = 10.0;
```

```
330.        }
331.
332.        if(timer % 25 == 12 || timer % 25 == 13 || timer % 25 == 14 ||
333.            timer % 25 == 15 || timer % 25 == 16 || timer % 25 == 17 ||
334.            timer % 25 == 18 || timer % 25 == 19 || timer % 25 == 20 ||
335.            timer % 25 == 21 || timer % 25 == 22 || timer % 25 == 23 ||
336.            timer % 25 == 24)
337.        {
338.            Squat = -3.0;
339.            ActionPosi[12] = -80.0;
340.            ActionPosi[15] = 90.0;
341.            ActionPosi[18] = -70.0;
342.            ActionPosi[19] = -90.0;
343.            ActionPosi[20] = 20.0;
344.            ActionPosi[21] = 90.0;
345.            ActionPosi[22] = -30.0;
346.            ActionPosi[23] = 10.0;
347.        }
348.    }
349.    if(SelectActionNum == 2)    // Boxing 拳擊
350.    {
351.        Squat = -0.8;
352.        for(int i = 0; i < 11 ; i++)
353.            ActionTheta[i] = 0.0;
354.        for (int i = 0; i < 24; i++)
355.        {
356.            ActionPosi[i] = 0.0;
357.            if(i == 6)
358.                ActionPosi[i] = 180.0;
359.        }
360.        timer = timer + 1;
361.        if(timer % 25 == 0 || timer % 25 == 1 || timer % 25 == 2 ||
362.            timer % 25 == 3 || timer % 25 == 4 || timer % 25 == 5 )
363.        {
364.            ActionPosi[0] = -50.0;
365.            ActionPosi[1] = -60.0;
366.            ActionPosi[2] = -5.0;
367.            ActionPosi[3] = -60.0;
368.            ActionPosi[4] = -30.0;
369.            ActionPosi[5] = 20.0;
```

```
370.            ActionPosi[6] = 100.0;
371.            ActionPosi[7] = -60.0;
372.            ActionPosi[8] = -5.0;
373.            ActionPosi[9] = -60.0;
374.            ActionPosi[10] = -30.0;
375.            ActionPosi[11] = 80.0;
376.            ActionPosi[12] = 0.0;
377.            ActionPosi[13] = -60.0;
378.            ActionPosi[14] = 10.0;
379.            ActionPosi[15] = 0.0;
380.            ActionPosi[16] = -30.0;
381.            ActionPosi[17] = -30.0;
382.            ActionPosi[18] = -20.0;
383.            ActionPosi[19] = -60.0;
384.            ActionPosi[20] = 0.0;
385.            ActionPosi[21] = 0.0;
386.            ActionPosi[22] = -30.0;
387.            ActionPosi[23] = -30.0;
388.        }
389.        if(timer % 25 == 6 || timer % 25 == 7 || timer % 25 == 8 ||
390.            timer % 25 == 9 || timer % 25 == 10 || timer % 25 == 11 )
391.        {
392.            ActionPosi[0] = -70.0;
393.            ActionPosi[1] = -60.0;
394.            ActionPosi[2] = 10.0;
395.            ActionPosi[3] = -10.0;
396.            ActionPosi[4] = 60.0;
397.            ActionPosi[5] = 0.0;
398.            ActionPosi[6] = 100.0;
399.            ActionPosi[7] = -60.0;
400.            ActionPosi[8] = -5.0;
401.            ActionPosi[9] = -60.0;
402.            ActionPosi[10] = -30.0;
403.            ActionPosi[11] = 80.0;
404.            ActionPosi[12] = 0.0;
405.            ActionPosi[13] = -60.0;
406.            ActionPosi[14] = 10.0;
407.            ActionPosi[15] = 0.0;
408.            ActionPosi[16] = -30.0;
409.            ActionPosi[17] = -30.0;
```

```
410.            ActionPosi[18] = -20.0;
411.            ActionPosi[19] = -60.0;
412.            ActionPosi[20] = 0.0;
413.            ActionPosi[21] = 0.0;
414.            ActionPosi[22] = -30.0;
415.            ActionPosi[23] = -30.0;
416.        }
417.        if(timer % 25 == 12 || timer % 25 == 13 || timer % 25 == 14 ||
418.            timer % 25 == 15 || timer % 25 == 16 || timer % 25 == 17 ||
419.            timer % 25 == 18 || timer % 25 == 19 || timer % 25 == 20 ||
420.            timer % 25 == 21 || timer % 25 == 22 || timer % 25 == 23 ||
421.            timer % 25 == 24)
422.        {
423.            ActionPosi[0] = -90.0;
424.            ActionPosi[1] = -60.0;
425.            ActionPosi[2] = 0.0;
426.            ActionPosi[3] = 60.0;
427.            ActionPosi[4] = 0.0;
428.            ActionPosi[5] = 0.0;
429.            ActionPosi[6] = 100.0;
430.            ActionPosi[7] = -60.0;
431.            ActionPosi[8] = -5.0;
432.            ActionPosi[9] = -60.0;
433.            ActionPosi[10] = -30.0;
434.            ActionPosi[11] = 80.0;
435.            ActionPosi[12] = 0.0;
436.            ActionPosi[13] = -60.0;
437.            ActionPosi[14] = 10.0;
438.            ActionPosi[15] = 0.0;
439.            ActionPosi[16] = -30.0;
440.            ActionPosi[17] = -30.0;
441.            ActionPosi[18] = -20.0;
442.            ActionPosi[19] = -60.0;
443.            ActionPosi[20] = 0.0;
444.            ActionPosi[21] = 0.0;
445.            ActionPosi[22] = -30.0;
446.            ActionPosi[23] = -30.0;
447.        }
448.    }
449.    if(SelectActionNum == 3) // Lifting dumbbells 舉重
```

```
450.    {
451.        Squat = 0;
452.        for(int i = 0; i < 11 ; i++)
453.            ActionTheta[i] = 0.0;
454.        for (int i = 0; i < 24; i++)
455.        {
456.            ActionPosi[i] = 0.0;
457.            if(i == 6)
458.                ActionPosi[i] = 180.0;
459.        }
460.        ActionPosi[0] = 0.0;
461.        ActionPosi[6] = 120.0;
462.        ActionPosi[12] = 0.0;
463.        ActionPosi[13] = 0.0;
464.        ActionPosi[14] = 0.0;
465.        ActionPosi[18] = 0.0;
466.        ActionPosi[19] = 0.0;
467.        ActionPosi[20] = 0.0;
468.        ActionPosi[21] = 0.0;
469.        ActionPosi[22] = 0.0;
470.        timer = timer + 1;
471.        if(timer % 24 == 0)
472.            ActionTheta[6] = -60.0;
473.        if(timer % 24 == 1)
474.            ActionTheta[6] = -50.0;
475.        if(timer % 24 == 2)
476.            ActionTheta[6] = -40.0;
477.        if(timer % 24 == 3)
478.            ActionTheta[6] = -30.0;
479.        if(timer % 24 == 4)
480.            ActionTheta[6] = -20.0;
481.        if(timer % 24 == 5)
482.            ActionTheta[6] = -10.0;
483.        if(timer % 24 == 6)
484.            ActionTheta[6] = 0.0;
485.        if(timer % 24 == 7)
486.            ActionTheta[6] = 10.0;
487.        if(timer % 24 == 8)
488.            ActionTheta[6] = 20.0;
489.        if(timer % 24 == 9)
```

```
490.            ActionTheta[6] = 30.0;
491.        if(timer % 24 == 10)
492.            ActionTheta[6] = 40.0;
493.        if(timer % 24 == 11)
494.            ActionTheta[6] = 50.0;
495.        if(timer % 24 == 12)
496.            ActionTheta[6] = 60.0;
497.        if(timer % 24 == 13)
498.            ActionTheta[6] = 50.0;
499.        if(timer % 24 == 14)
500.            ActionTheta[6] = 40.0;
501.        if(timer % 24 == 15)
502.            ActionTheta[6] = 30.0;
503.        if(timer % 24 == 16)
504.            ActionTheta[6] = 20.0;
505.        if(timer % 24 == 17)
506.            ActionTheta[6] = 10.0;
507.        if(timer % 24 == 18)
508.            ActionTheta[6] = 0.0;
509.        if(timer % 24 == 19)
510.            ActionTheta[6] = -10.0;
511.        if(timer % 24 == 20)
512.            ActionTheta[6] = -20.0;
513.        if(timer % 24 == 21)
514.            ActionTheta[6] = -30.0;
515.        if(timer % 24 == 22)
516.            ActionTheta[6] = -40.0;
517.        if(timer % 24 == 23)
518.            ActionTheta[6] = -50.0;
519.    }
520.    if(SelectActionNum == 4)    // Open and close jump 開合跳
521.    {
522.        for(int i = 0; i < 11 ; i++)
523.            ActionTheta[i] = 0.0;
524.        for (int i = 0; i < 24; i++)
525.        {
526.            ActionPosi[i] = 0.0;
527.            if(i == 6)
528.                ActionPosi[i] = 180.0;
529.        }
```

```
530.        timer = timer + 1;
531.        if(timer % 25 == 0 || timer % 25 == 1 || timer % 25 == 2 ||
532.            timer % 25 == 3 || timer % 25 == 4 || timer % 25 == 5 )
533.        {
534.            Squat = 1;
535.            ActionPosi[0] = 0.0;
536.            ActionPosi[1] = 0.0;
537.            ActionPosi[2] = 140.0;
538.            ActionPosi[3] = -30.0;
539.            ActionPosi[4] = 90.0;
540.            ActionPosi[5] = 90.0;
541.            ActionPosi[6] = -180.0;
542.            ActionPosi[7] = -180.0;
543.            ActionPosi[8] = 140.0;
544.            ActionPosi[9] = -30.0;
545.            ActionPosi[10] = 90.0;
546.            ActionPosi[11] = 90.0;
547.            ActionPosi[12] = 0.0;
548.            ActionPosi[13] = 0;
549.            ActionPosi[14] = 10;
550.            ActionPosi[15] = 0.0;
551.            ActionPosi[16] = 0;
552.            ActionPosi[17] = 0;
553.            ActionPosi[18] = 0;
554.            ActionPosi[19] = 0;
555.            ActionPosi[20] = -10.0;
556.            ActionPosi[21] = 0.0;
557.            ActionPosi[22] = 0;
558.            ActionPosi[23] = 0;
559.        }
560.        if(timer % 25 == 6 || timer % 25 == 7 || timer % 25 == 8 ||
561.            timer % 25 == 9 || timer % 25 == 10 || timer % 25 == 11 )
562.        {
563.            Squat = 0;
564.            ActionPosi[0] = 0.0;
565.            ActionPosi[1] = 0.0;
566.            ActionPosi[2] = 190.0;
567.            ActionPosi[3] = -30.0;
568.            ActionPosi[4] = 90.0;
569.            ActionPosi[5] = 90.0;
```

```
570.            ActionPosi[6] = -180.0;
571.            ActionPosi[7] = -180.0;
572.            ActionPosi[8] = 190.0;
573.            ActionPosi[9] = -30.0;
574.            ActionPosi[10] = 90.0;
575.            ActionPosi[11] = 90.0;
576.            ActionPosi[12] = 0.0;
577.            ActionPosi[13] = 0;
578.            ActionPosi[14] = 30;
579.            ActionPosi[15] = 0.0;
580.            ActionPosi[16] = 0;
581.            ActionPosi[17] = 0;
582.            ActionPosi[18] = 0;
583.            ActionPosi[19] = 0;
584.            ActionPosi[20] = -30.0;
585.            ActionPosi[21] = 0.0;
586.            ActionPosi[22] = 0;
587.            ActionPosi[23] = 0;
588.        }
589.        if(timer % 25 == 12 || timer % 25 == 13 || timer % 25 == 14 ||
590.            timer % 25 == 15 || timer % 25 == 16 || timer % 25 == 17 )
591.        {
592.            Squat = 1;
593.            ActionPosi[0] = 0.0;
594.            ActionPosi[1] = 0.0;
595.            ActionPosi[2] = 140.0;
596.            ActionPosi[3] = -30.0;
597.            ActionPosi[4] = 90.0;
598.            ActionPosi[5] = 90.0;
599.
600.            ActionPosi[6] = -180.0;
601.            ActionPosi[7] = -180.0;
602.            ActionPosi[8] = 140.0;
603.
604.            ActionPosi[9] = -30.0;
605.            ActionPosi[10] = 90.0;
606.            ActionPosi[11] = 90.0;
607.
608.            ActionPosi[12] = 0.0;
609.            ActionPosi[13] = 0;
```

```
610.            ActionPosi[14] = 10;
611.
612.            ActionPosi[15] = 0.0;
613.            ActionPosi[16] = 0;
614.            ActionPosi[17] = 0;
615.
616.            ActionPosi[18] = 0;
617.            ActionPosi[19] = 0;
618.            ActionPosi[20] = -10.0;
619.
620.            ActionPosi[21] = 0.0;
621.            ActionPosi[22] = 0;
622.            ActionPosi[23] = 0;
623.        }
624.        if(timer % 25 == 18 || timer % 25 == 19 || timer % 25 == 20 ||
625.            timer % 25 == 21 || timer % 25 == 22 || timer % 25 == 23 || timer % 25
    == 24)
626.        {
627.            Squat = 0;
628.            ActionPosi[0] = 0.0;      // 我視角的右手 軸肢
629.            ActionPosi[1] = 0.0;
630.            ActionPosi[2] = 0.0;
631.            ActionPosi[3] = -30.0;   // 我視角的右手 前肢
632.            ActionPosi[4] = 90.0;
633.            ActionPosi[5] = 90.0;
634.            ActionPosi[6] = -180.0; // 我視角的左手 軸肢
635.            ActionPosi[7] = -180.0;
636.            ActionPosi[8] = 0.0;
637.            ActionPosi[9] = -30.0;   // 我視角的左手 前肢
638.            ActionPosi[10] = 90.0;
639.            ActionPosi[11] = 90.0;
640.            ActionPosi[12] = 0.0;    // 我視角的右腳 軸肢
641.            ActionPosi[13] = 0;
642.            ActionPosi[14] = 0;
643.            ActionPosi[15] = 0.0;    // 我視角的右腳 前肢
644.            ActionPosi[16] = 0;
645.            ActionPosi[17] = 0;
646.            ActionPosi[18] = 0;       // 我視角的左腳 軸肢
647.            ActionPosi[19] = 0;
648.            ActionPosi[20] = 0.0;
```

```
649.            ActionPosi[21] = 0.0;    // 我視角的左腳 前肢
650.            ActionPosi[22] = 0;
651.            ActionPosi[23] = 0;
652.        }
653.    }
654.    if(SelectActionNum == 5)    // Restart 初始動作
655.    {
656.        Squat = 0;
657.        for(int i = 0; i < 11 ; i++)
658.            ActionTheta[i] = 0.0;
659.        for (int i = 0; i < 24; i++)
660.        {
661.            ActionPosi[i] = 0.0;
662.            if(i == 6)
663.                ActionPosi[i] = 180.0;
664.        }
665.    }
666.    glutPostRedisplay();
667.      glutTimerFunc(60,ActionChange,1);
668. }
669.
670. void display(void)
671. {
672.    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
673.    glLoadIdentity();
674.    glColor3f(1.0, 0.0, 0.0);
675.
676.    /* ********************************  身體
    **********************************/
677.    glRotatef(theta[0], 0.0, 1.0, 0.0);
678.    glTranslatef(0.0, Squat, 0.0);
679.    torso();
680.
681.    /* ********************************  頭  ***********************************/
682.    glPushMatrix();
683.    glTranslatef(0.0, TORSO_HEIGHT+0.5*HEAD_HEIGHT, 0.0);
684.    glRotatef(theta[1], 1.0, 0.0, 0.0);
685.    glRotatef(theta[2], 0.0, 1.0, 0.0);
686.    glTranslatef(0.0, -0.5*HEAD_HEIGHT, 0.0);
687.    head();
```

```
688.    glPopMatrix();
689.
690.    /* ****************************  我視角的左手
    *******************************/
691.    glPushMatrix();
692.    glTranslatef(TORSO_HEIGHT * -0.47, 0.7 * TORSO_HEIGHT, 0.0);
693.    glRotatef(180, 1.0, 0.0, 0.0);
694.    glRotatef(ActionPosi[0], 1.0, 0.0, 0.0);
695.    glRotatef(ActionPosi[1], 0.0, 1.0, 0.0);
696.    glRotatef(ActionPosi[2], 0.0, 0.0, 1.0);
697.    // 軸肢
698.    left_upper_arm();
699.    glTranslatef(0.0, UPPER_ARM_HEIGHT, 0.0);
700.    glRotatef(theta[4], 1.0, 0.0, 0.0);
701.    glRotatef(ActionTheta[5], 1.0, 0.0, 0.0);
702.    // 前肢
703.    glRotatef(ActionPosi[3], 1.0, 0.0, 0.0);
704.    glRotatef(ActionPosi[4], 0.0, 1.0, 0.0);
705.    glRotatef(ActionPosi[5], 0.0, 0.0, 1.0);
706.    left_lower_arm();
707.    glPopMatrix();
708.
709.    /* ****************************  我視角的右手
    *******************************/
710.    glPushMatrix();
711.    glTranslatef(TORSO_HEIGHT * 0.47, 0.7 * TORSO_HEIGHT, 0.0);
712.    glRotatef(theta[5], 1.0, 0.0, 0.0);
713.    glRotatef(ActionPosi[6], 1.0, 0.0, 0.0);
714.    glRotatef(ActionPosi[7], 0.0, 1.0, 0.0);
715.    glRotatef(ActionPosi[8], 0.0, 0.0, 1.0);
716.    // 軸肢
717.    right_upper_arm();
718.    glTranslatef(0.0, UPPER_ARM_HEIGHT, 0.0);
719.    glRotatef(theta[6], 1.0, 0.0, 0.0);
720.    glRotatef(ActionTheta[6], 1.0, 0.0, 0.0);
721.    // 前肢
722.    glRotatef(ActionPosi[9], 1.0, 0.0, 0.0);
723.    glRotatef(ActionPosi[10], 0.0, 1.0, 0.0);
724.    glRotatef(ActionPosi[11], 0.0, 0.0, 1.0);
725.    right_lower_arm();
```

```
726.    glPopMatrix();
727.
728.    /* ****************************  我視角的左腳
    ******************************/
729.    glPushMatrix();
730.    glTranslatef(-(TORSO_RADIUS+UPPER_LEG_RADIUS),
    0.1*UPPER_LEG_HEIGHT, 0.0);
731.    glTranslatef(0.5555, -0.5, 0.0);
732.    glRotatef(theta[7], 1.0, 0.0, 0.0);
733.    // 軸肢
734.    glRotatef(ActionPosi[12], 1.0, 0.0, 0.0);
735.    glRotatef(ActionPosi[13], 0.0, 1.0, 0.0);
736.    glRotatef(ActionPosi[14], 0.0, 0.0, 1.0);
737.    left_upper_leg();
738.    glTranslatef(0.0, UPPER_LEG_HEIGHT, 0.0);
739.    glRotatef(theta[8], 1.0, 0.0, 0.0);
740.    // 前肢
741.    glRotatef(ActionPosi[15], 1.0, 0.0, 0.0);
742.    glRotatef(ActionPosi[16], 0.0, 1.0, 0.0);
743.    glRotatef(ActionPosi[17], 0.0, 0.0, 1.0);
744.    left_lower_leg();
745.    glPopMatrix();
746.
747.    /* ****************************  我視角的右腳
    ******************************/
748.    glPushMatrix();
749.    glTranslatef(TORSO_RADIUS+UPPER_LEG_RADIUS, 0.1*UPPER_LEG_HEIGHT,
    0.0);
750.    glTranslatef(-0.5555, -0.5, 0.0);
751.    glRotatef(theta[9], 1.0, 0.0, 0.0);
752.    // 軸肢
753.    glRotatef(ActionPosi[18], 1.0, 0.0, 0.0);
754.    glRotatef(ActionPosi[19], 0.0, 1.0, 0.0);
755.    glRotatef(ActionPosi[20], 0.0, 0.0, 1.0);
756.    right_upper_leg();
757.    glTranslatef(0.0, UPPER_LEG_HEIGHT, 0.0);
758.    glRotatef(theta[10], 1.0, 0.0, 0.0);
759.    // 前肢
760.    glRotatef(ActionPosi[21], 1.0, 0.0, 0.0);
761.    glRotatef(ActionPosi[22], 0.0, 1.0, 0.0);
```

```
762.     glRotatef(ActionPosi[23], 0.0, 0.0, 1.0);
763.     right_lower_leg();
764.     glPopMatrix();
765.
766.     glFlush();
767.     glutSwapBuffers();
768. }
769.
770. void mouse(int btn, int state, int x, int y)
771. {
772.       if(btn==GLUT_LEFT_BUTTON && state == GLUT_DOWN)
773.     {
774.         theta[angle] += 5.0;
775.         if( theta[angle] > 360.0 ) theta[angle] -= 360.0;
776.     }
777.       if(btn==GLUT_RIGHT_BUTTON && state == GLUT_DOWN)
778.     {
779.         theta[angle] -= 5.0;
780.         if( theta[angle] < 360.0 ) theta[angle] += 360.0;
781.     }
782.     display();
783. }
784.
785. void myReshape(int w, int h)
786. {
787.       glViewport(0, 0, w, h);
788.       glMatrixMode(GL_PROJECTION);
789.       glLoadIdentity();
790.       if (w <= h)
791.             glOrtho(-10.0, 10.0, -10.0 * (GLfloat) h / (GLfloat) w, 10.0 * (GLfloat) h /
    (GLfloat) w, -10.0, 10.0);
792.       else
793.             glOrtho(-10.0 * (GLfloat) w / (GLfloat) h, 10.0 * (GLfloat) w / (GLfloat) h,
    0.0, 10.0, -10.0, 10.0);
794.       glMatrixMode(GL_MODELVIEW);
795.       glLoadIdentity();
796. }
797.
798. void myinit()
799. {
```

```
800.    GLfloat mat_specular[]={1.0, 1.0, 1.0, 1.0};
801.      GLfloat mat_diffuse[]={1.0, 1.0, 1.0, 1.0};
802.      GLfloat mat_ambient[]={1.0, 1.0, 1.0, 1.0};
803.      GLfloat mat_shininess={100.0};
804.      GLfloat light_ambient[]={0.0, 0.0, 0.0, 1.0};
805.      GLfloat light_diffuse[]={1.0, 1.0, 1.0, 1.0};
806.      GLfloat light_specular[]={1.0, 1.0, 1.0, 1.0};
807.      GLfloat light_position[]={10.0, 10.0, 10.0, 0.0};
808.
809.    // 設置光源 glLightfv(光源編號, 光源特性, 參數數據)
810.      glLightfv(GL_LIGHT0, GL_POSITION, light_position);
811.      glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
812.      glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
813.      glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular);
814.
815.    // 設置材質 決定打光出來的效果
816.      glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
817.      glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient);
818.      glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse);
819.      glMaterialf(GL_FRONT, GL_SHININESS, mat_shininess);
820.
821.    // 控制繪製指定兩點間其他點顏色的過度模式
822.      glShadeModel(GL_SMOOTH);
823.    glDepthFunc(GL_LEQUAL);
824.      glEnable(GL_LIGHTING);
825.      glEnable(GL_LIGHT0);
826.      glEnable(GL_DEPTH_TEST);
827.      glEnable(GL_COLOR_MATERIAL);
828.    glEnable(GL_COLOR_MATERIAL);
829.
830.    // 設置視窗背景顏色
831.    glClearColor(0.796, 0.553, 0.553, 1.0);
832.
833.    h=gluNewQuadric();
834.    gluQuadricDrawStyle(h, GLU_FILL);
835.    t=gluNewQuadric();
836.    gluQuadricDrawStyle(t, GLU_FILL);
837.    lua=gluNewQuadric();
838.    gluQuadricDrawStyle(lua, GLU_FILL);
839.    lla=gluNewQuadric();
```

```
840.     gluQuadricDrawStyle(lla, GLU_FILL);
841.     rua=gluNewQuadric();
842.     gluQuadricDrawStyle(rua, GLU_FILL);
843.     rla=gluNewQuadric();
844.     gluQuadricDrawStyle(rla, GLU_FILL);
845.     lul=gluNewQuadric();
846.     gluQuadricDrawStyle(lul, GLU_FILL);
847.     lll=gluNewQuadric();
848.     gluQuadricDrawStyle(lll, GLU_FILL);
849.     rul=gluNewQuadric();
850.     gluQuadricDrawStyle(rul, GLU_FILL);
851.     rll=gluNewQuadric();
852.     gluQuadricDrawStyle(rll, GLU_FILL);
853.     quadratic = gluNewQuadric();
854.     gluQuadricDrawStyle(quadratic, GLU_FILL);
855. }
856.
857. void BasicMenu(int id)
858. {
859.     if(id < 11)
860.         angle = id;
861.     glutPostRedisplay();
862. }
863.
864. void ActionMenu(int id)
865. {
866.     if(id == 11) // Sit down and cross leg 坐下翹腳
867.         SelectActionNum = 1;
868.     if(id == 12) // Boxing 拳擊
869.         SelectActionNum = 2;
870.     if(id == 13) // Lifting dumbbells 舉重
871.         SelectActionNum = 3;
872.     if(id == 14)
873.         SelectActionNum = 4;
874.     if(id == 15) // Restart
875.         SelectActionNum = 5;
876.     glutPostRedisplay();
877. }
878.
879. void MainMenu(int id)
```

```
880. {
881.     if(id == 16)
882.         exit(0);
883. }
884.
885. int main(int argc, char **argv)
886. {
887.     glutInit(&argc, argv);
888.     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
889.     glutInitWindowSize(500, 500);
890.     glutCreateWindow("HW3 Robot - 406262216");
891.     myinit();
892.     glutReshapeFunc(myReshape);
893.     glutDisplayFunc(display);
894.     glutMouseFunc(mouse);
895.     // 計時器
896.     glutTimerFunc(33,ActionChange,1);
897.
898.     // 創建基本的動作選單
899.     int Basic_Menu = glutCreateMenu(BasicMenu);
900.     glutAddMenuEntry("torso", 0);
901.     glutAddMenuEntry("right_upper_arm", 3);
902.     glutAddMenuEntry("right_lower_arm", 4);
903.     glutAddMenuEntry("left_upper_arm", 5);
904.     glutAddMenuEntry("left_lower_arm", 6);
905.     glutAddMenuEntry("right_upper_leg", 7);
906.     glutAddMenuEntry("right_lower_leg", 8);
907.     glutAddMenuEntry("left_upper_leg", 9);
908.     glutAddMenuEntry("left_lower_leg", 10);
909.
910.     // 創建設計的動作選單
911.     int Action_Menu = glutCreateMenu(ActionMenu);
912.     glutAddMenuEntry("Sit down and cross leg", 11);
913.     glutAddMenuEntry("Boxing", 12);
914.     glutAddMenuEntry("Lifting dumbbells", 13);
915.     glutAddMenuEntry("Open and close jump", 14);
916.     glutAddMenuEntry("Restart", 15);
917.
918.     // 創建主選單，並將 Basic 和 Action Menu 將上主選單
919.     glutCreateMenu(MainMenu);
```

```
920.     glutAddSubMenu("Basic", Basic_Menu);
921.     glutAddSubMenu("Action", Action_Menu);
922.     glutAddMenuEntry("exit", 16);
923.     glutAttachMenu(GLUT_MIDDLE_BUTTON);
924.
925.     glutMainLoop();
926. }
```