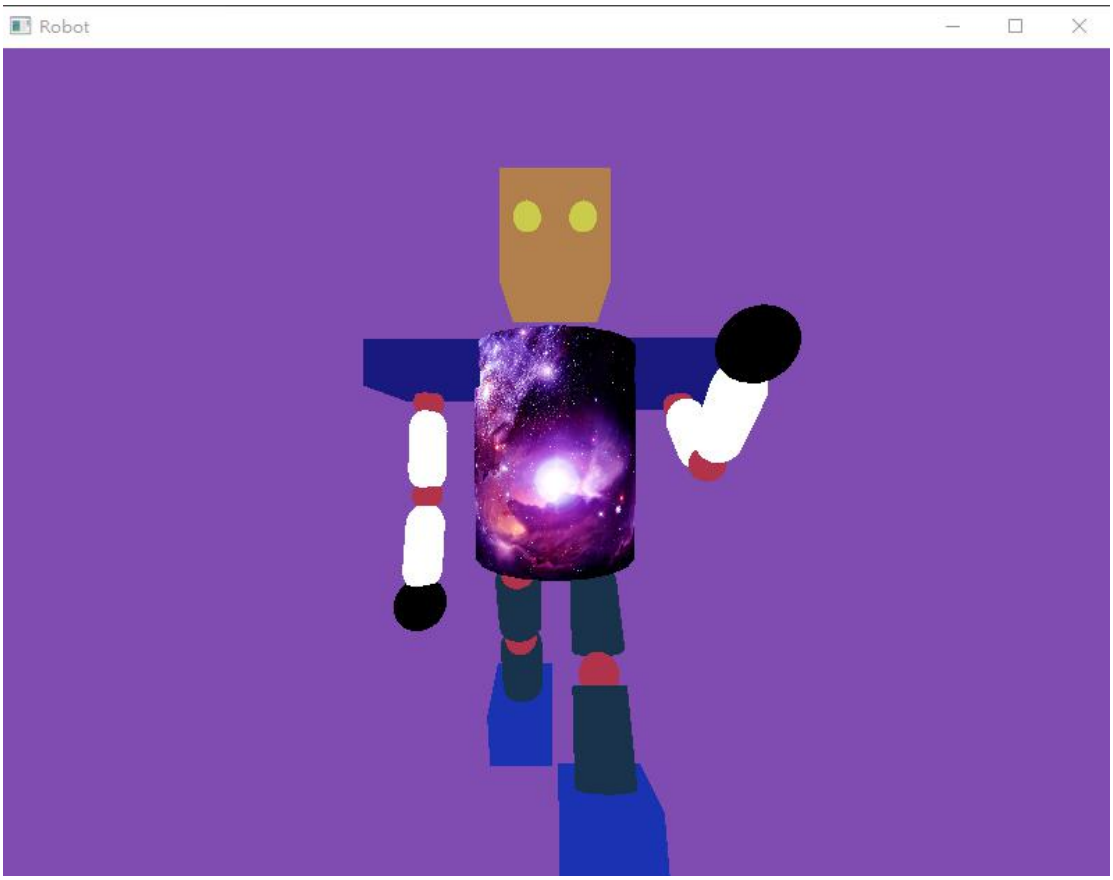
Assignment1 Report

1. Screenshot of Robot



2. The relationship/transformation stack of robot body part

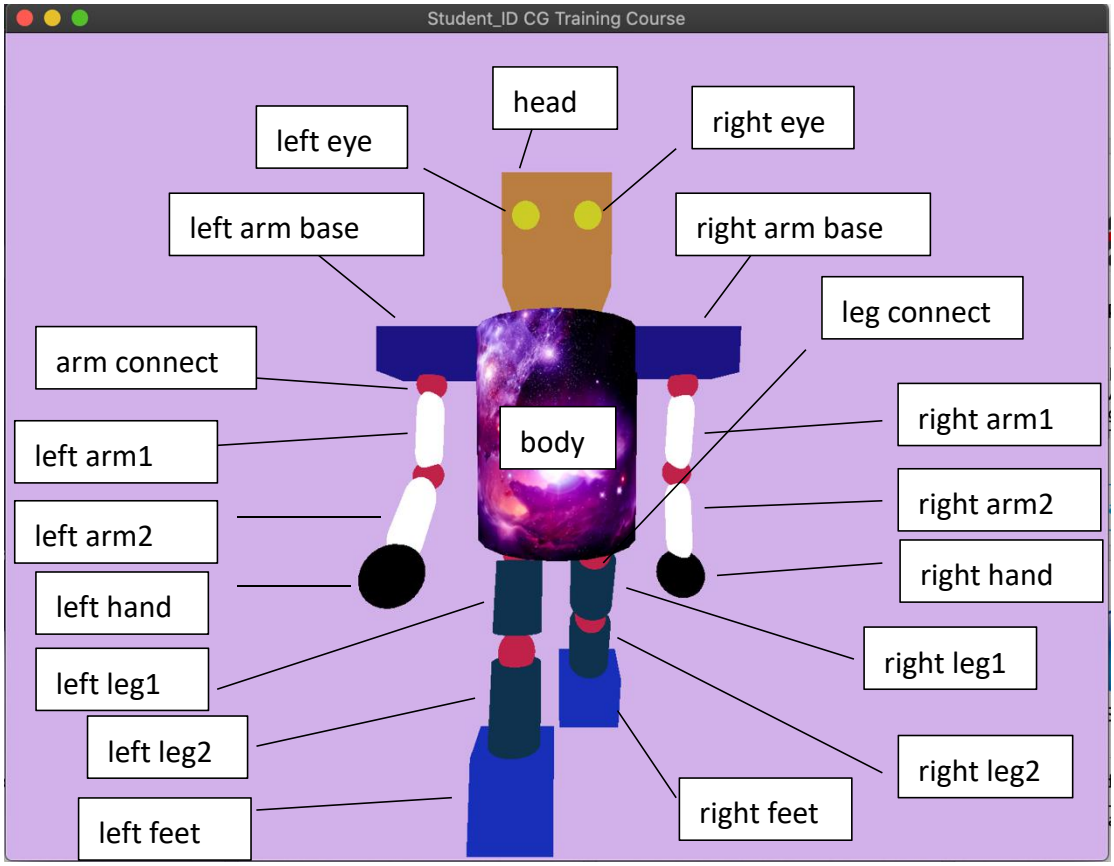
機器人的各部位會有層級的繼承關係，以手部來說，手就是由下臂，上臂和身體繼承而來，以腳部來說，腳就是由小腿，大腿和身體繼承的，也因此在跑動作的時候會有大臂帶小臂的連動感，以右手來舉例：

```
    // // /// /// /// // // // // / /// // // //// // // // // / / right hand

    mat4 T_hand2, R_hand2, S_hand2;
    //models[4].position = vec3(0, -0.9, 0);
    T_hand2 = translate(mat4(1.0), vec3(0, -0.9, 0));
    S_hand2 = scale(mat4(1.0), vec3(0.8, 0.8, 0.8));
    //Build rotation matrix
//  degree = glfwGetTime();
    rotate_axis = vec3(0.0, 1.0, 0.0);
    R_hand2 = rotate(mat4(1.0), degree, rotate_axis);

    mat4 model_tmp_hand2 = R_hand2 * T_hand2;
    mat4 model_matrix_hand2 = model_tmp_body * model_tmp_arm_base2 * model_tmp_arm_connect3 * model_tmp_arm3 * model_tmp_arm_connect4 * model_tmp_arm4 * model_tmp_hand2 * S_hand2;
```

右手的 model matrix 就是由 body, arm 等部位的 model matrix 相乘後再乘上自己的 R(旋轉), T(位移), S(縮放)，如此一來就成功完成部位間的繼承，其他部位也是以此類推



3. Functions in program/how to use

(a) 滑鼠事件:

用滑鼠左鍵點畫面左方看將機器人向左移動，點選左上可以向左上移，點左下可向左下移，點選右方可將機器人往右移，點右下往右下移，點右上網右上移，左右我是以畫面正中央區分，上中下是將畫面的高分成三塊來區分

(b) 鍵盤事件:

Key Z:

點選 z 可以使機器任上下頭腳顛倒，實作方法是多乘一個鏡像矩陣

Key X:

點選 x 可以讓機器人沿 z 軸左轉 90 度，再點一次可以右轉 90 度回來，實作方法是多成一個旋轉矩陣

Key C:

點選 c 可以使機器人動畫暫停，並且不能執行任何其他動作，再點一次即可啟動，實作方法是用個 flag 判斷要不要 render

Key V:

點選 v 可以讓機器人開始逆時針旋轉，再按一次可暫停旋轉，實作方法是用個 flag 判斷四肢所繼承的身體要不要旋轉，身體不轉四肢也就不會轉

Menu:

在視窗上點右鍵可以開啟 menu，menu 有兩個功能，start 或 stop 機器人和 exit，其中 start 和 stop 和 key C 的功用可以互通，exit 就是直接結束顯示的視窗

4. 問題與困難

(a) 剛開始以為每多一個顏色就要多一個 shader 出來，後來發現可以藉由傳 uniform 來指定顏色，不需額外多增加 shader

(b)一開始旋轉的時候，只要經過暫停，下一次開始旋轉的位置不會是當前暫停的位置，原因是 timer_cnt 會繼續跑，解決方法是將暫停的總時長記錄下來，在開始時再由 timer_cnt 扣掉暫停時長，就可以從上次暫停的位置開始繼續旋轉

(c)在做旋轉 90 度時，我一開始是旋轉 camera 來達到這個效果，但後來發現這樣會讓機器人位置跑掉，因為轉了 camera 機器人的相對位置就會跟著 camera 跑掉，想了一陣子後，最後改為乘以一個旋轉矩陣取代旋轉 camera，如此一來因為 camera 沒有動，所以機器人旋轉後也能停在相同位置，不會偏離

IDE:
Visual Studio 2017