

# Computer Graphics HW2

104062203 陳涵宇

## 1. Translation Matrix

根據上課投影片的第 39 頁將對應的值填入。

## 2. Scaling Matrix

根據上課投影片的第 40 頁將對應的值填入。

## 3. Rotate alone axis-X、axis-Y、axis-Z

根據上課投影片的第 41~43 頁將對應的值填入。

## 4. Set viewing matrix

首先根據上課投影片第 69 頁完成矩陣  $T$ ，再根據第 70 頁完成矩陣  $R$ ，最後利用第 72 頁的公式  $M_{view} = R \cdot T$  計算出 viewing matrix。

## 5. Set orthogonal projection matrix

上課投影片中有兩種版本，分別在第 89 及 125 頁。兩者的差異是前者的  $z_{far}$  及  $z_{near}$  與後者的  $far$ 、 $near$  正負號相反。由於 projection 中的  $farClip$  及  $nearClip$  皆為正值，因此選擇根據第 125 頁進行實作。

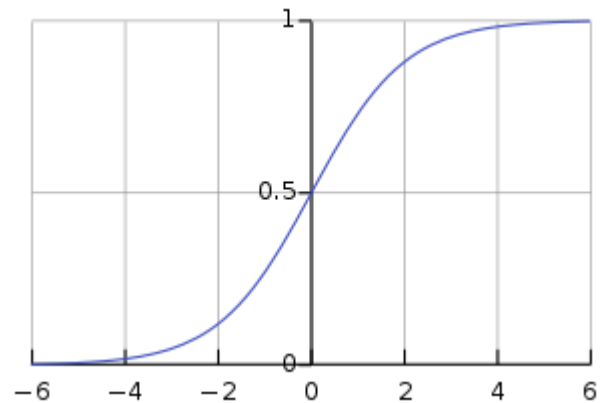
## 6. Set perspective projection matrix

上課投影片中有兩種版本，分別在第 95 及 130 頁。由於後者看起來較為簡潔，因此選擇根據後者進行實作。

## 7. Model movement

```
// new add variables
enum Movement {
    Stop = 0,
    Z = 1,
    X = 2,
};

GLfloat smooth_movement = 0;
GLfloat sample = -6;
GLfloat* x_positions;
Movement moving = Stop;
```



為了實作出 smooth transition，我新增了左上圖中的幾個 global 變數。這個 function 是將 sample 代入 logistic function( $\frac{1}{1+e^{-t}}$ ，其中 t 即為 sample)中計算出一個介於 0~1 的值。每次計算完會將 sample 加上 0.005 為下次計算做準備。關於 sample 初值的選擇，是因為我選擇的 logistic function 會呈現右上圖的圖形，從圖中可以看到當 t=-6 時已經極為接近 0，所以我選擇以 -6 作為 sample 的初值。

## 8. Draw model

在 draw model 前會根據前項提到的 moving 的狀態進行不同的計算，再搭配前項算出的 smooth\_movement 以及記錄各個 model 當前 x 座標的 x\_positions 算出正確的值並 assign 給 position vector 的 x。在預設狀態下，各 model 的 x 座標會是 index\*2。

### (1) 初始狀態

如果是剛執行程式的話，cur\_idx=0 且 moving=Stop，這時的 x 值便是預設值。

### (2) 按下 Z

當按下 Z 時會有兩種情況。一種是 cur\_idx 不為 0，所有 model 右移；另一種是 cur\_idx 為 0，這時所有 model 便需要左移直到最右邊 model 的預設 x 座標在 0。由於當各個 model 的 index 是 cur\_idx 有可能會做各種 transform，所以用 x\_positions 記錄各個 model 的當前 x 座標，再加上計算出的預設 offset 就會得到正確的 x 座標

### (3) 按下 x

與(2)相同，只是方向相反。

## 9. Assign MVP correct value

當 model 的 position 計算出來後便可以生成對應的 translate、rotation 以及 scaling matrix，再利用公式  $\text{model} = \text{translate} \cdot \text{rotate} \cdot \text{scaling}$  以及  $\text{MVP} = \text{projection} \cdot \text{viewing} \cdot \text{model}$  即可求出。

## 10. Draw the plane

同 draw model 中的 command，但是 plane 只由兩個三角形組成，因此 `glDrawArrays` 的第三個參數直接輸入 6 即可。

## 11. Self rotation on current model

讓 current model 的 rotation vector 中的 y 一直加上固定值即可。為避免程式執行過久時可能會 overflow 造成意外的錯誤，因此加完值會檢查值是否超過  $2\pi$ ，若超過則會減去  $2\pi$ 。

## 12. Switch case in onMouse

若調整的是 camera，那麼根據模式，camera 對應的 vector 的 z 值會改變，並且呼叫 `setViewingMatrix` 計算新的矩陣。

若調整 model 本身，那麼直接改變目前 model 對應的 vector 的 z 值即可。

## 13. Switch case in onMouseMotion

類似於 12，只是調整的是對應 vector 的 x 與 y 值，而且會根據 `diff_x`、`diff_y` 進行計算。其中 `GeoTranslation` 為了配合 8，因此 x 值會存在 `x_positions` 而不是直接存在 vector 的 x 值。另外 `GeoRotation` 也比較特別，因為當滑鼠沿著 x 軸移動時，model 實際上沿著 y 軸旋轉，反之亦然，因此 vector 的 x/y 值與 `diff_y/diff_x` 相關，是與其它模式較為不同的部分。

## 14. Switch case in onKeyboard

z/x: 切換 model，為了實現 smooth transition，因此每次按下都會將 sample 設為-6，並將 moving 切換到對應的值。

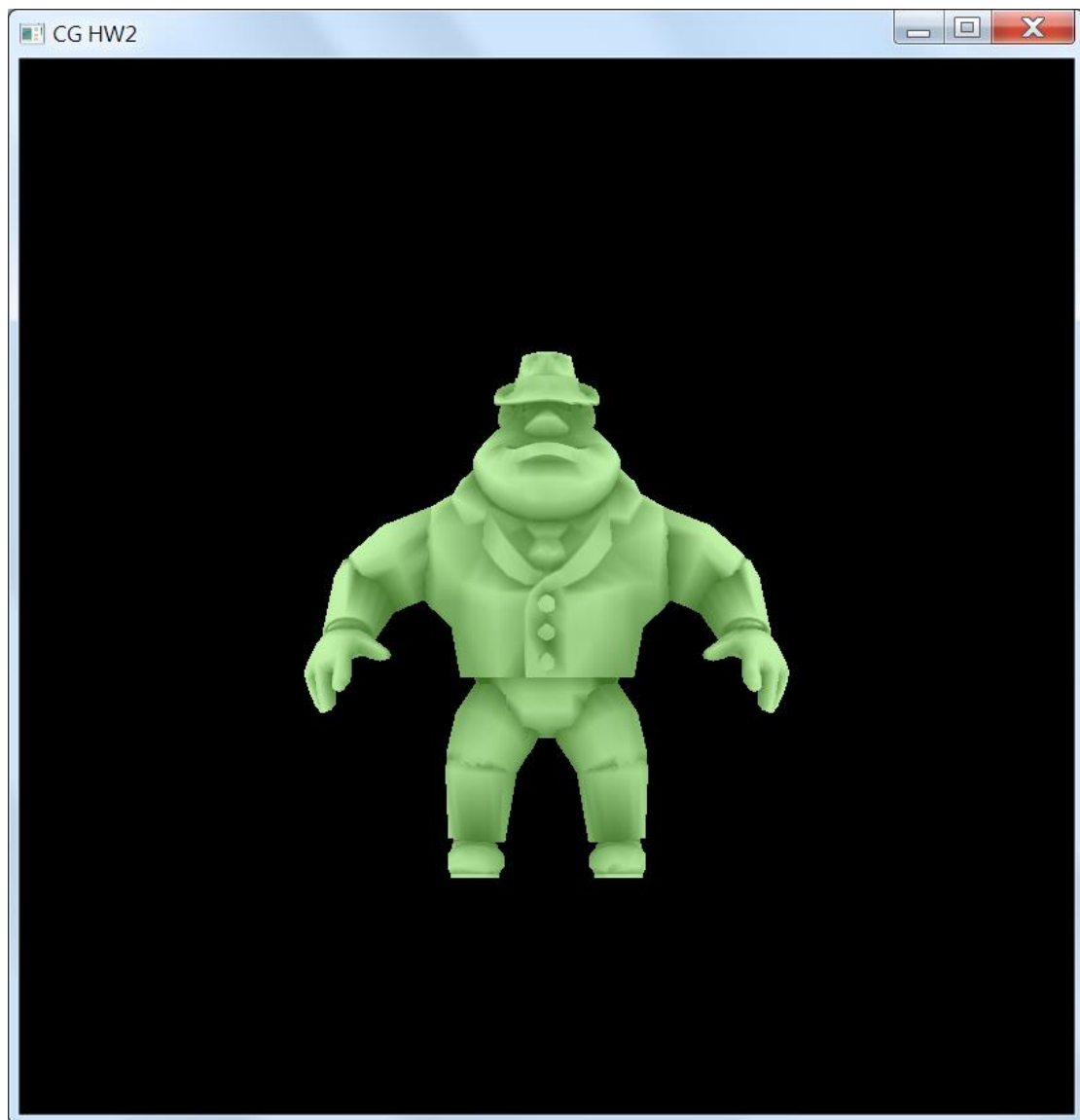
o: 直接呼叫 setOrthogonal。

p: 直接呼叫 setPerspective。

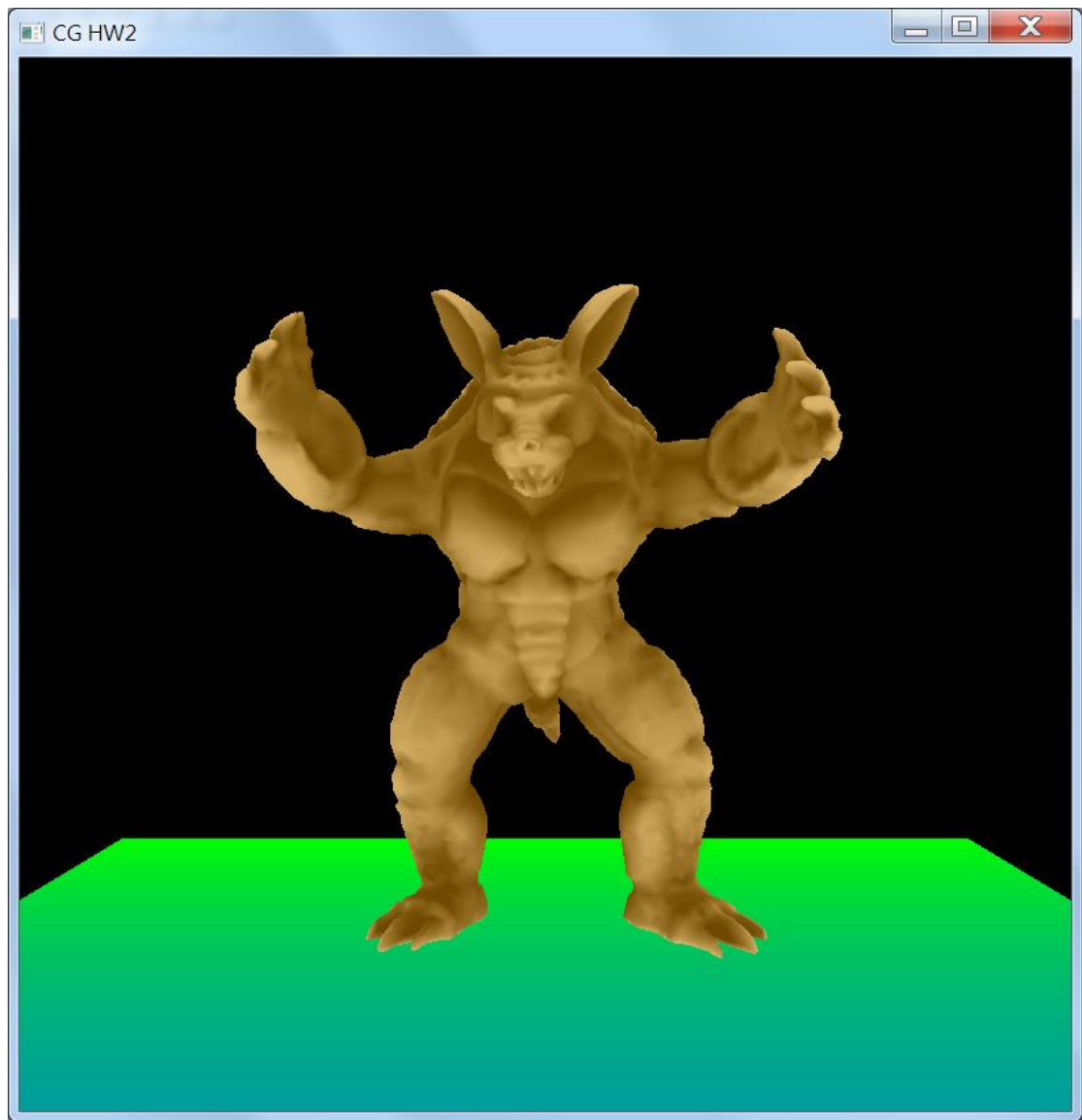
t/s/r/e/c/u: 將 cur\_trans\_mode 切換到對應的值。

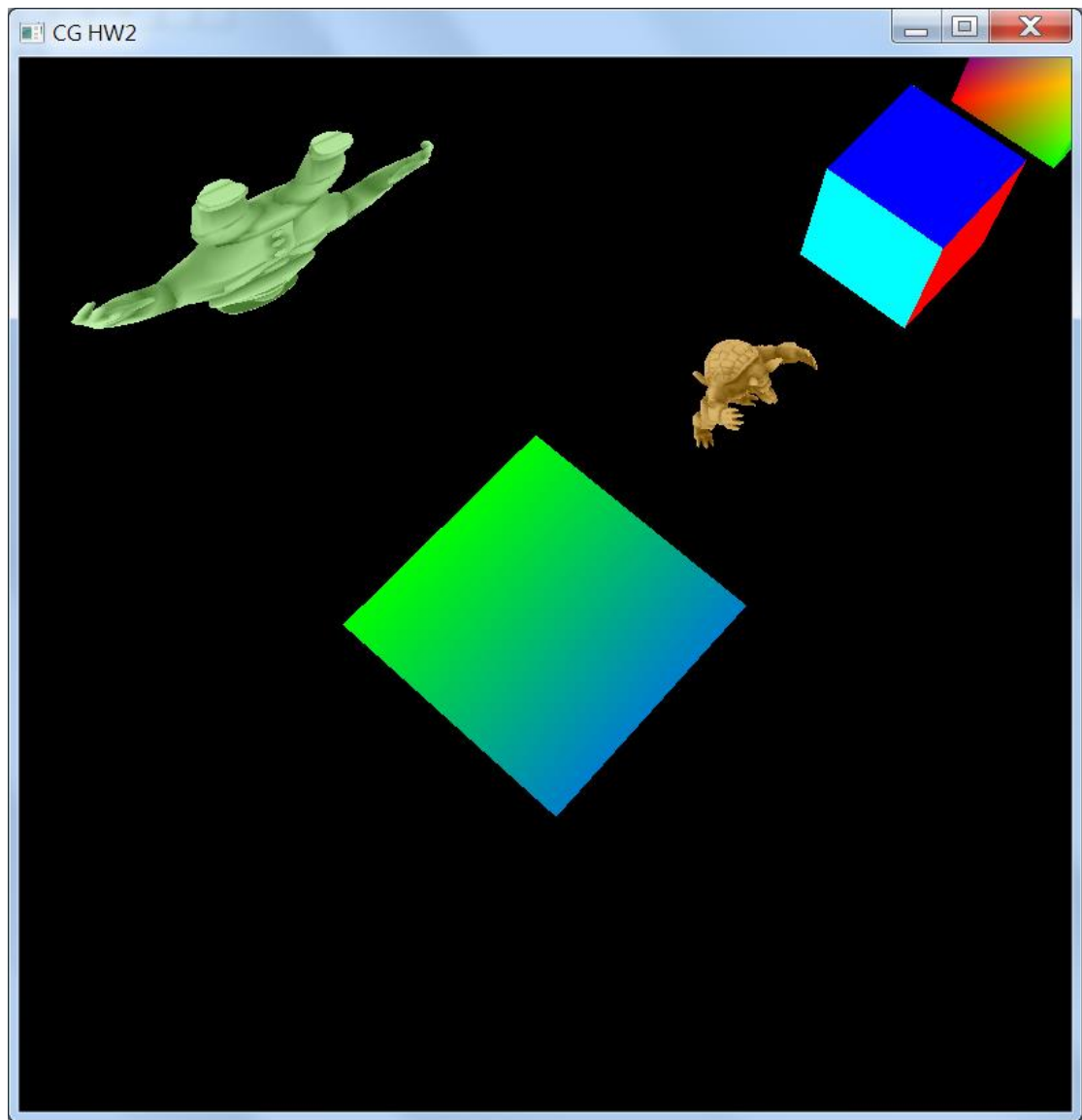
i: 印出當前 model、model 的三個 vector、camera 的三個 vector、目前的 projection mode 以及目前的 transformation mode。

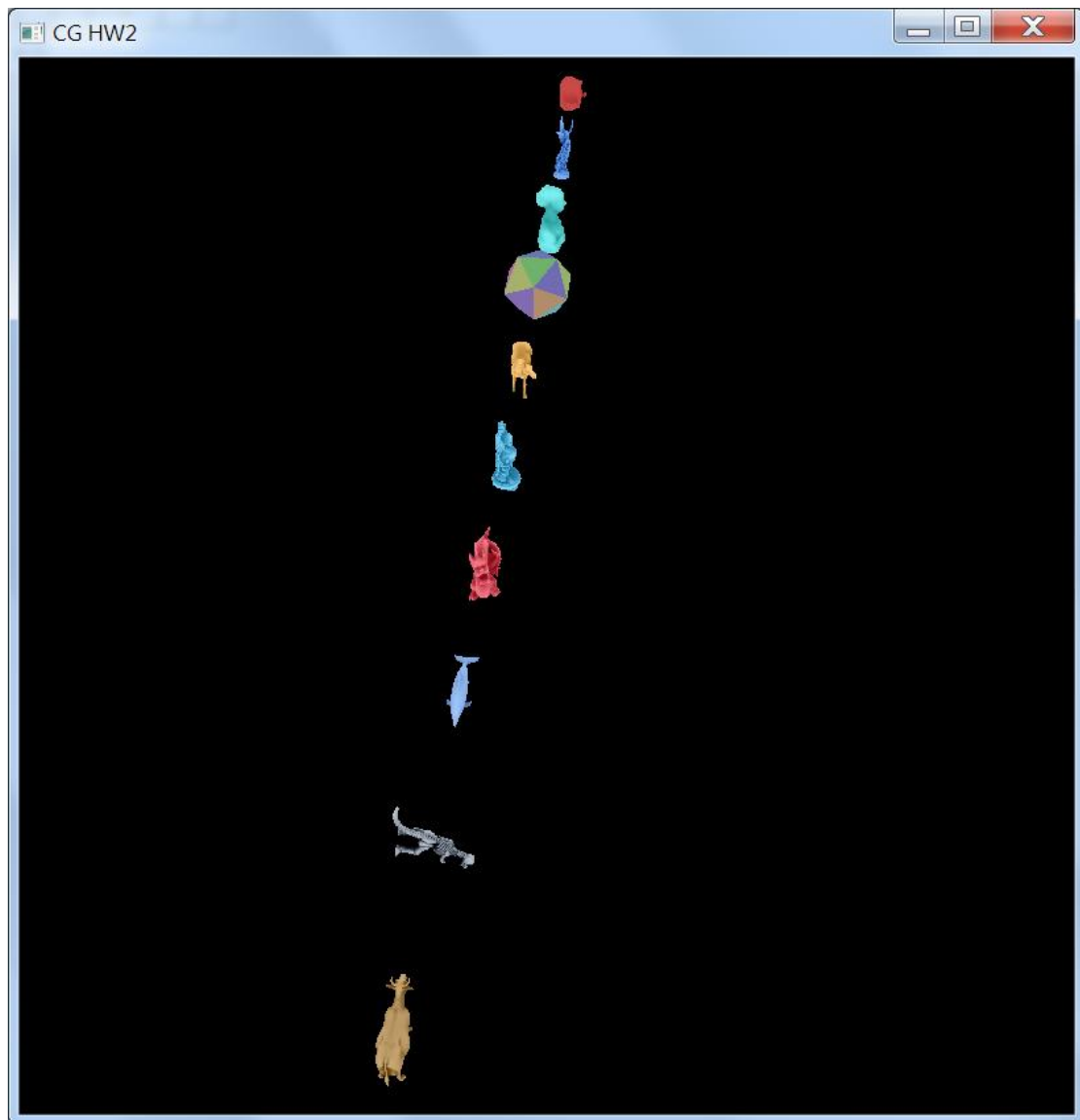
## Screenshots



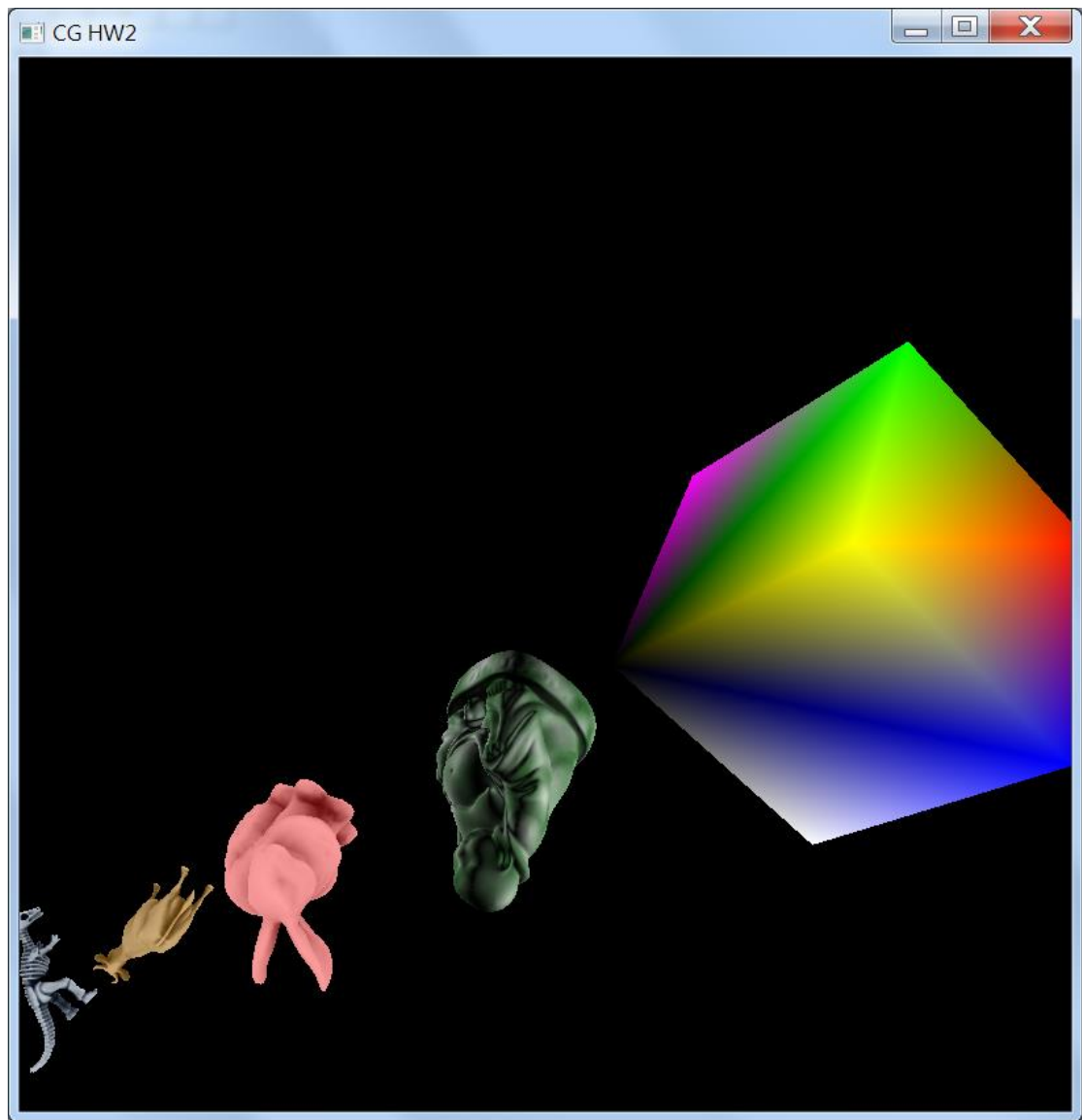


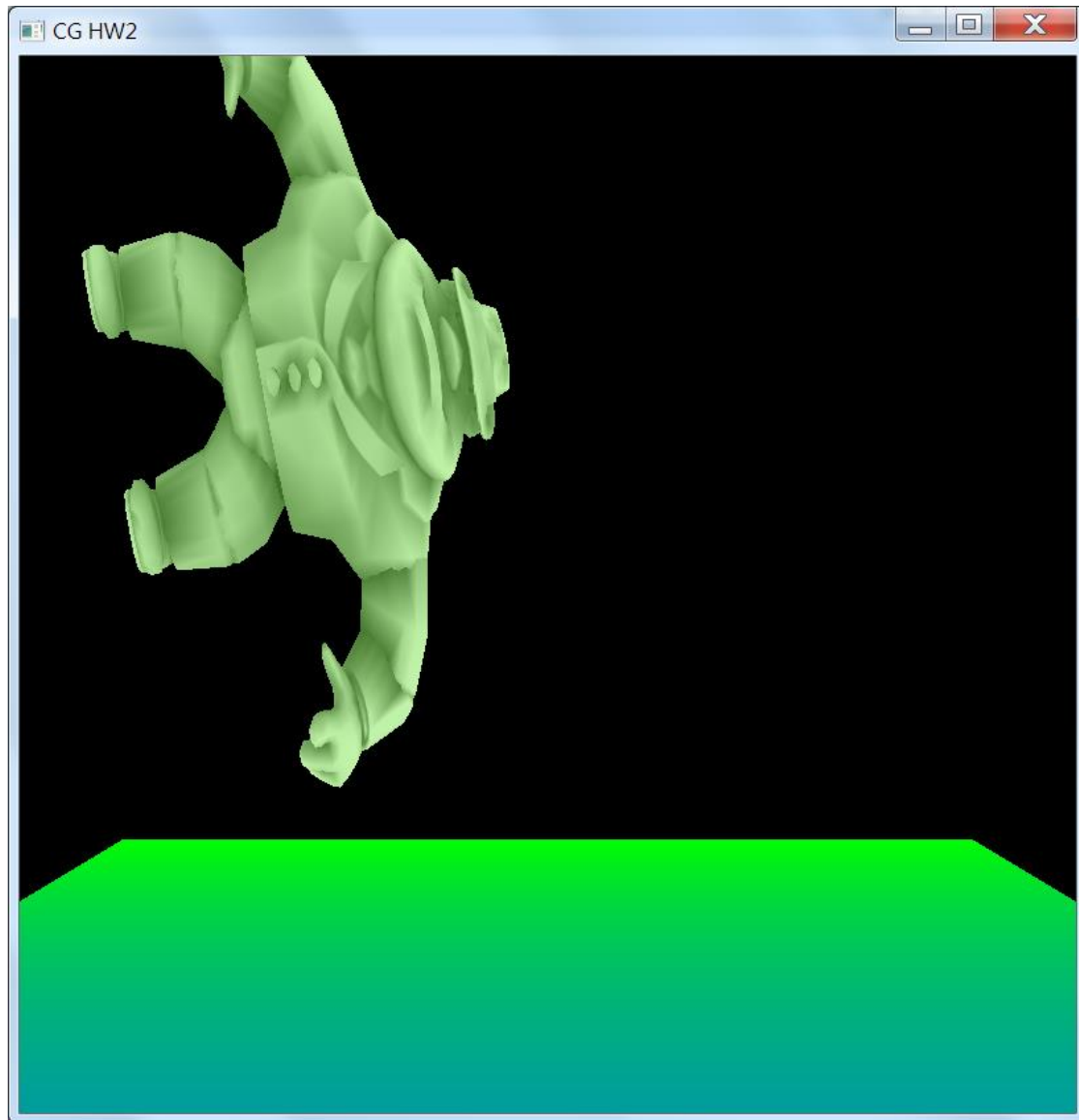












```
E:\Computer Graphics Assignment\HW2\104062203_HW_2\Assignment2\Debug\Assig...
onKeyboardSpecial(): (18, 186) key: 0x74
    onMouse(): (223, 184) start
    onMouseMotion(): (223, 186) mouse move
    onMouse(): (223, 186) end
    onKeyboard(): (230, 261) key: i(0x69)

=====INFO=====
current model: 0
    position: (-0.925000, 0.900000, -3.500000)
    rotation: (0.005236, 0.000000, 1.745329)
    scale: (-3.799998, -1.700000, 1.000000)

camera:
    position: (0.000000, 0.000000, 2.000000)
    center: (0.000000, 0.000000, 0.000000)
    up vector: (0.000000, 1.000000, 0.000000)

current projection mode: Perspective

now in geometrical transformation mode: rotation
=====

onKeyboardSpecial(): (5, 217) key: 0x74
```