Computer Graphics 2024-1 PA1

2023-15725 곽민서

**1. File Description**

I modified two .py files from skeleton, main.py and render.py. For render.py, I modified update() method that implements animation. And modified \_\_init\_\_() to make it easier to initialize. Then, I modified main.py to add objects.

**2. Scenario**

There are SIX big scene in my animation. S#1, a man appears and walks into the stage. He starts at (-60, 0, 0) and walks along x axis. As the viewpoint is (20, 15, 20), he came from diagonal. Then he stops. S#2, he places the box stuff around him. He place the boxes in West, East, South with his MAGIC wand. There are two small boxes in East. Then he soon realized he misplaced the box above. He tries to break it. S#3, soon, the ball-shaped-TNT falls from the sky and land on right side of him. He wonders what it is. Soon, the TNT ignites! He astonished at it! S#4, he started to run away…. But he comes back.... As he touched TNT, it becomes bigger! S#5, he tries to escape with his magic wand. He uses his magic wand as helicopter. S#6, But it was not enough for it to carry him, he falls. He get hurts. The End.

**3. Implementation**

There are 3 hierarchy levels and 6 links, 5 joints. First level, body, and second level, head, arms and legs, and third level, the MAGIC wand attached to his right hand. Joints are neck, connecting body and head, shoulder, connecting body and arm, and hand connecting wand and arm. All actions are implemented with CustomGroup.transform\_mat, the matrix that is totally multiplied. It is easy for dealing with translation. As the Mat4.from\_translation, which makes 3D affine translation, so what I needed was just using function to make 4x4 Matrix and multiply to transform\_mat. But for rotation, we need to change axis and origin. To do so, I needed to figure where the object is. How I implemented is to get x, y, z from (transform\_mat) @ (start Vec4). Then, trns1, that translate the object to origin, rt, that rotate object, last, translate the object back to the original place. Same for changing size.

Next, I needed to make hierachy. How I implemented is using heap() function from CustomGroup class. I changed \_\_init\_\_() function to give own custom value of order. I placed high level(body) to lowest order, then placed low level(wand) to high order. For exception I placed not-hierachial objects such as boxes, TNT to even lower order, -1, -5…. Then, I made range of heap, such as heap >=0, for total movement, heap >= 7, for wand movement only. By multiplying multiple matrix multiple times, we can implement hierachy.

Finally, I need to make it time-based. As update() method calls at 1frame interval, I made every instruction in update() method, and distinguish them with my custom variable, self.counter. It represents tick in my update() method. It increases by 1 after 1 update method is called. By setting range of self.counter to distinguish instructions, Many instructions in one method can be called orderly.

**4. Flaws**

There are some flaws in my code. First, it was not well-organized. I didn’t plan up all implementation before start programming, I can’t make all variable at once. So variables are badly organized, repeated, useless. Also, I couldn’t used textures at all. And also there are not many hierachy levels too. Most importantly, the animations is played totally different due to the computer environment. The code that I implemented can be only run at specific RAM and VRAM usage percentage, or it can move different. The screenshot attached is what I intented to implement.

**5. Screenshots** : I cannot captured my animation because the movement changes during recording.





