

Project Assignment 2

2018009234 한관희

UI Implementation

1. Camera

If "c" or "space bar" is pressed, the camera is changed

If the number (0, 1, 2, 3, 4) is pressed, the camera is changed to the camera corresponding the number.

2. Click

"Clicked" means the state of the mouse goes DOWN and then goes UP.

If the right mouse is clicked, you can see the position of the cursor.

If the left mouse is clicked, there is no control point and the cow is not clicked, nothing happens.

If the left mouse is clicked and the number of control points is less than 6, you can add a new control point at the location. Exactly, the control point is added when the mouse is UP.

If the left mouse is clicked and cow is moving following catmull-rom spline curve, you can go back to the initial mode.

3. Drag

If the mouse is DOWN and not UP yet, you can change the cow's vertical position by moving the cursor with holding down the left mouse.

If the cow is already clicked, you can change the cow's horizontal position by moving the cursor.

Variable I made

You can see this variables at line 109.

-first_click :

If the left mouse is clicked for the first time in initial mode, a control point is not added. "first_click" is the flag for that

-clicked_pos :

If current drag is V_DRAG(vertical drag), I have to save the current cow's position for vertical positioning.

-start_cow2wld :

Variable to save the first cow2wld to implement yaw and pitch rotation.

"start_cow2wld" is initialized in "inititalize", and is not changed forever.

-cow2wld_arr:

List to save cow2wld in control point.

The element of cow2wld_arr is added in "onMouseButton", cow2wld_arr is cleared

in "clean_up."

-coeff_arr :

List to save the coefficients resulting from the calculation of the catmull-rom spline.

-start_time :

Variables to set the criteria for starting animation.

"start_time" is set in "onMouseButton"

-animate_count = 0

Variables for three animations. "animate_count" is used in "display" and cleared in "clean_up."

-prev_coeff_idx = 0

Variable to check whether a track has been run. If "prev_coeff_idx" is 5 and current "coeff_idx" is 0, the cow moved the track once.

-curr_pos :

Position the cow should be in when the cow moves following the catmull-rom spline curve.

-hermite_matrix, catmullrom_matrix :

Matrices just saved to calculate the catmull-rom spline. They are just data.

Function I made

You can see this functions at line 133.

-cal_catmullrom :

This function takes 4 cow2wld as arguments. It extracts position from cow2wld and calculates the coefficients of the cubic equation by using hermite spline and catmull-rom spline.

-cal_cow_pos :

This function interpolates the cow's next position by using current time and the coefficients obtained from "cal_catmullrom"

-transformation :

This function takes the cow's current position and next position. It calculates Translate Matrix, Rotate Matrices and returns cow2wld to be drawn.

-clean_up :

This function cleans up the flags, the list and makes the direction the cow is looking the same as start_cow2wld.

How to Implement

1. Drag

When dragging, the cow should follow the mouse cursor. I modified "onMouseButton" and "onMouseDown."

Here are what I modified :

IsDrag is 0 only in initial mode.

If the cow is clicked in initial mode, isDrag is always V_DRAG or H_DRAG.

Therefore, I updated the cow's "pickInfo" in "onMouseDown."

For vertical positioning, the cow's position is saved when clicked and the cow is fixed by using the clicked position in "onMouseDown".

2. Catmull-rom spline

I calculated the catmull-rom spline curve using 6 control points.

The cow's position was interpolated by using the coefficients resulting from the calculation of the catmull-rom spline.

I set "time" to have a value of 0~1 for 6 times in "display." I interpolate the position by using "time" and the coefficients.

3. Transformation (Yaw and Pitch)

The most difficult thing to implement was the implementation of yaw and pitch rotation. At first, I applied continuously rotation matrix to cow2wld. So the cow's direction became strange. It felt like roll rotation had also applied.

To solve this problem I saved the first cow2wld to "start_cow2wld" in "initialize".

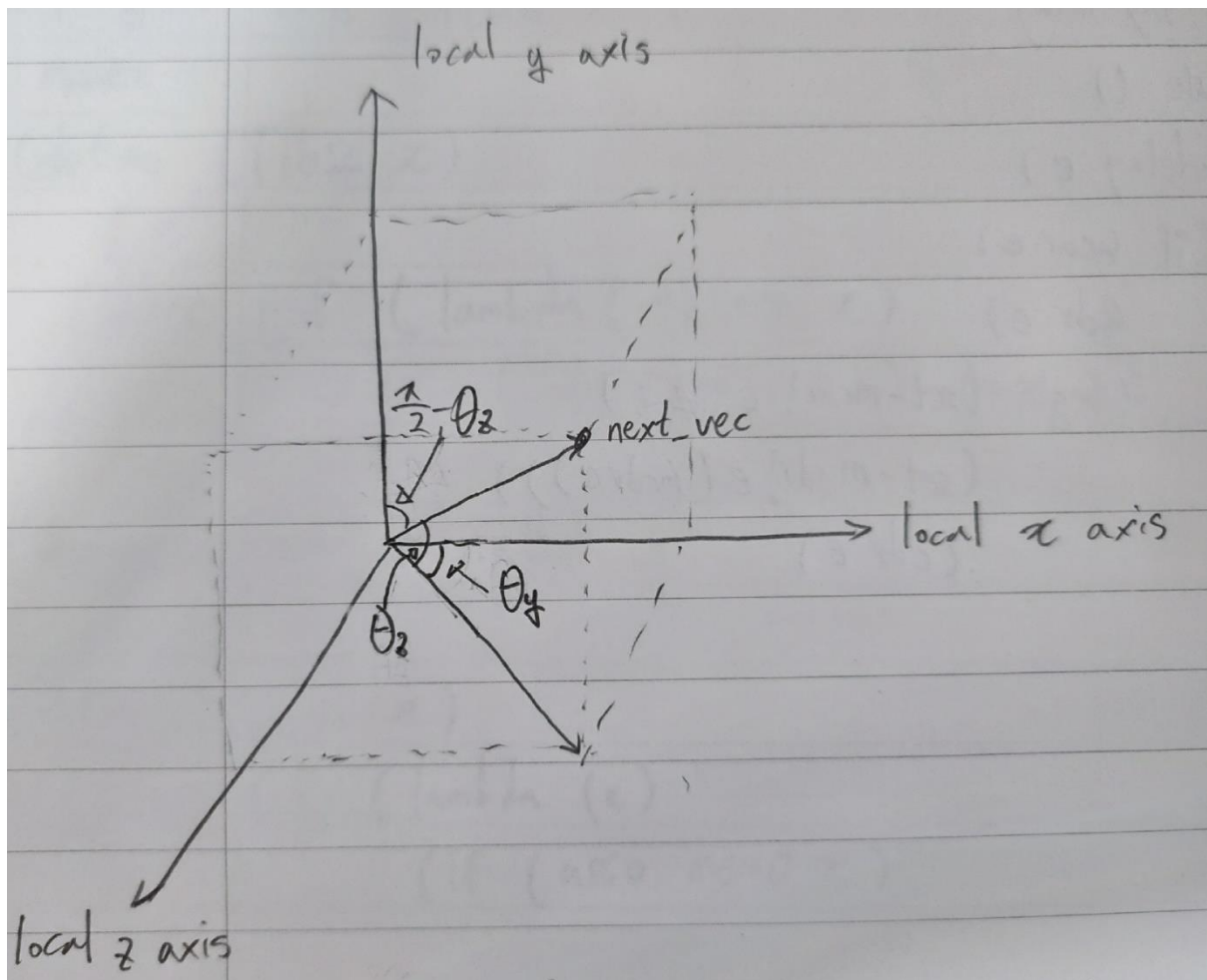
And I always applied the transformation matrix to start_cow2wld.

line 198: $\text{ret_cow2wld} = T @ \text{copy_cow2wld} @ R_y @ R_z$

I applied the translate matrix from the origin of "start_cow2wld" to the current position with global coordinate.

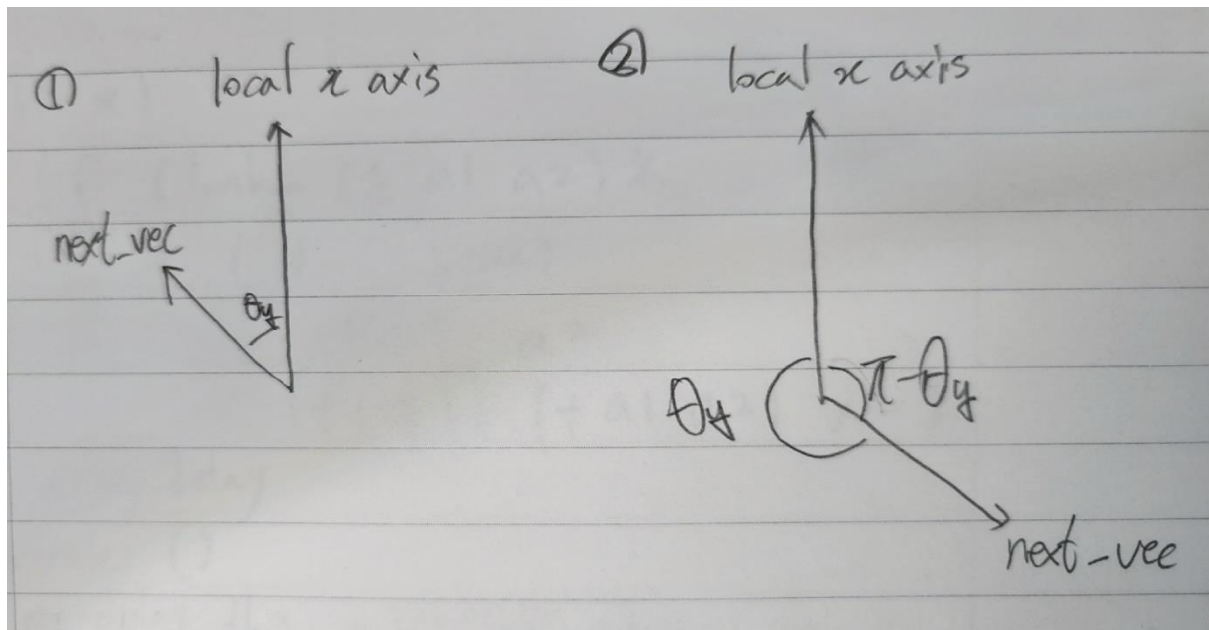
And I applied the rotation matrices with local coordinate.

To do that, I had to calculate the angle for yaw, pitch rotations.



First, I calculated the angle between the local y axis of "start_cow2wld" and the next direction vector(next_vec)

yaw angle("theta_z") should be $\pi/2$ - the angle.



And I calculated the angle between the local x axis of "start_cow2wld" and a vector with only x and z values of the next direction vector(next_vec)

In case 1, pitch angle("theta_y") is just the angle.

In case 2, pitch angle("theta_y") should be π - the angle.