

# CS7GV5 Real-Time Animation Assignment 1

Van Allen Bruns Jr - 19329560

## Simple Plane

A simple representation of a plane with yaw, pitch, and roll rotations, using Euler angles, was implemented. "a" and "d" controlled roll, while the mouse controlled pitch and yaw. Rotations were performed in the following order: roll, then pitch, then yaw. Gimbal-lock was observed while in a nose-dive, as changes to both yaw and roll appeared to perform the same function.



Here, we can see the mouse and keyboard capture specifically for roll, pitch, and yaw, along with how the plane model matrix is updated from these.

```
void processPassiveMouse(int xpos, int ypos) {
    float sensitivity = 0.3f; // change this value to your liking
    float xoffset = xpos - lastX;
    float yoffset = ypos - lastY;

    lastX = xpos;
    lastY = ypos;

    xoffset *= sensitivity;
    yoffset *= sensitivity;

    yaw += xoffset;
    pitch += yoffset;
}
```

```
void processNormalKeys(unsigned char key, int x, int y) {
    float rollSpeed = 5.0f;

    switch (key) {
        case 'a':
            roll -= rollSpeed;
            break;
        case 'd':
            roll += rollSpeed;
            break;
    }

    void updatePlaneModel() {
        planeModel = scale(identity_mat4(), vec3(0.3f, 0.3f, 0.3f));
        planeModel = translateFirst(rotate_y_deg(rotate_x_deg(rotate_z_deg(planeModel,
            roll), -pitch), yaw), planeLocation);
    }
}
```

## Extras

Extra features were the following:

1. Hierarchical moving elements: propeller and wheels spun individually, while adjusting to plane movements.
2. Partial implementation of first-person view (third-person original implementation).
3. Good visual appearance using loaded models (as seen above).
4. Multiple mini-viewports for seeing different angles of the plane.



## References

- DHHH. (2020, February 4). *Spowith f1 camel*. Retrieved from TURBOSQUID: <https://www.turbosquid.com/3d-models/free-spowith-f1-camel-3d-model/516387>
- DISQUS. (2020, February 4). *Camera*. Retrieved from Learn OpenGL: <https://learnopengl.com/Getting-started/Camera>
- Lantz, K. (2020, February 4). *A preliminary Wavefront OBJ loader in C++*. Retrieved from [www.keithlantz.net](http://www.keithlantz.net): <https://www.keithlantz.net/2011/10/a-preliminary-wavefront-obj-loader-in-c/>
- OpenGL Programming/Modern OpenGL Tutorial Load OBJ*. (2020, February 4). Retrieved from WIKIBOOKS: [https://en.wikibooks.org/wiki/OpenGL\\_Programming/Modern\\_OpenGL\\_Tutorial\\_Load\\_OBJ](https://en.wikibooks.org/wiki/OpenGL_Programming/Modern_OpenGL_Tutorial_Load_OBJ)