```
0 references
void Update()
{
    ProcessTranslation();
    ProcessRotation();
}

1 reference
void ProcessRotation()
{
    float pitchDueToPosition = transform.localPosition.y * positionPitchFactor;
    float pitchDueToControl = yThrow * controlThrowFactor;
    float pitch = pitchDueToPosition + pitchDueToControl;
    float yaw = transform.localPosition.x * positionYawFactor;
    float roll = xThrow * controlRollFactor;
    transform.localRotation = Quaternion.Euler(pitch, yaw, roll);        // Setting the pitch, yaw and roll

}
```

Quaternion.Euler(pitch, yaw, roll); (takes numeric value in the parameter of Euler pre-defined method)

Note: *transform.localposition* or *transform.localRotation* stores the current value of the game object where as **new Vector3(x-axis , y-axis , z-axis)** and **Quaternion.Euler(pitch, yaw, roll)** let us set the new value of position and orientation.