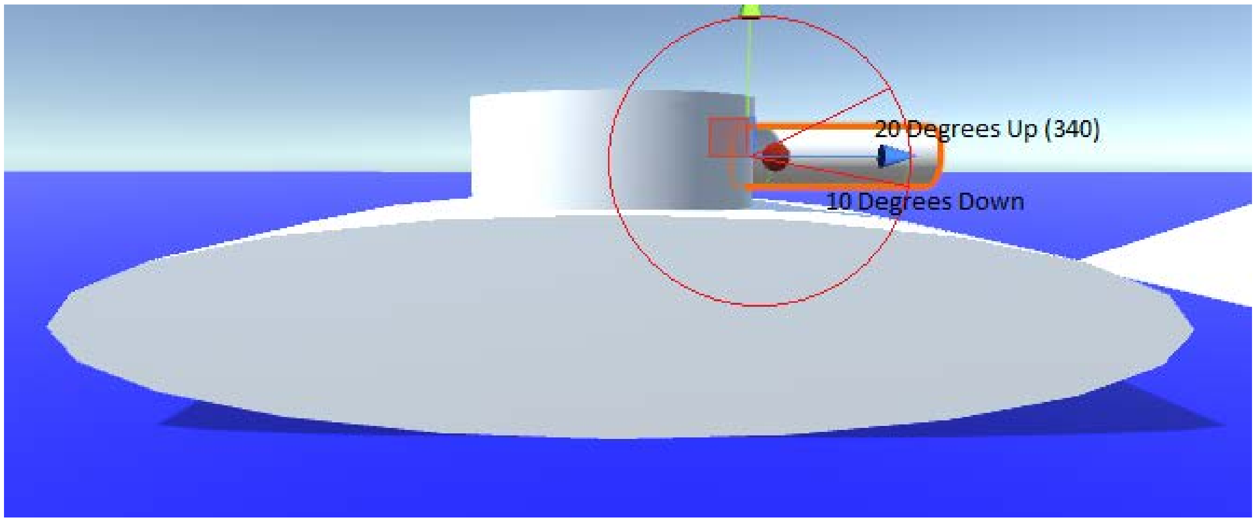
Clamping Movement

Next we will be looking at how to clamp the movement of certain objects. If you have the barrel of your tank moving up and down then you may notice that it can rotate 360 degrees right through the tank body. We will need to check the angles of the barrel anytime that it moves to make sure that we don't go too far. First you need to understand that when the barrel is pointing forward the angle on the X axis is 0. If you rotate down the barrel may start to go through the body at about 10 degrees. Also, we should probably stop the barrel from rotating up at about 340 degrees.



Now all that we need to do is look at the code that will accomplish this:

```
public class BarrelMovement : MonoBehaviour
{
    float rInput, rotationSpeed;

    Vector3 angles;

    // Use this for initialization
    void Start()
    {
        rotationSpeed = 100.0f;
    }

    // Update is called once per frame
    void Update()
    {
        rInput = Input.GetAxis("Barrel");

        transform.Rotate(Vector3.right, Time.deltaTime * rotationSpeed * rInput);

        angles = transform.rotation.eulerAngles;

        if (angles.x > 10 && angles.x < 90)
        {
            transform.rotation = Quaternion.Euler(10.0f, angles.y, angles.z);
        }
        if (angles.x < 340 && angles.x > 270)
        {
            transform.rotation = Quaternion.Euler(340.0f, angles.y, angles.z);
        }
    }
}
```

We need to get the angles in degrees by using eulerAngles. Once we have done this it is easier to detect when the rotation is past where you want to clamp the angle. If the rotation does go past your range then you can set that specific angle back while keeping the remaining angles where they are. Here is another example of how to do this from the Unity Standard assets MouseLook.cs:

```
transform.rotation = ClampRotationAroundXAxis(transform.rotation);

Quaternion ClampRotationAroundXAxis(Quaternion q)
{
    q.x /= q.w;
    q.y /= q.w;
    q.z /= q.w;
    q.w = 1.0f;

    float angleX = 2.0f * Mathf.Rad2Deg * Mathf.Atan(q.x);

    angleX = Mathf.Clamp(angleX, -20, 10);

    q.x = Mathf.Tan(0.5f * Mathf.Deg2Rad * angleX);

    return q;
}
```