```
using UnityEngine;
using System.Collections;

public class TurretGun : MonoBehaviour
{
    private int m_damage;

    private float m_attackSpeed;
    private float s_attackSpeed;
    private float m_aimSpeed;
    private float m_step;

    private Vector3 m_forward;
    private Vector3 m_targetDotPos;
    private Vector3 m_lookDirection;

    private Quaternion m_lookRotation;

    private bool m_fired = false;
    private bool m_inLOS = false;

    private GameObject target;

    private TurretBody Turret;

    // Use this for initialization
    void Start()
    {
        InitiateStats();
    }

    // Update is called once per frame
    void Update()
    {
        GetTarget();

        CheckTargetState();
        CheckAttackStatus();
    }

    void InitiateStats()
    {
        Turret = GameObject.FindWithTag("Turret").GetComponent<TurretBody>();

        m_damage = Turret.damage;
        m_attackSpeed = Turret.attackSpeed;
        s_attackSpeed = m_attackSpeed;
        m_aimSpeed = Turret.aimSpeed;
    }

    void GetTarget()
    {
        target = Turret.target;
    }

    void LookAtTarget()
    {
        m_lookDirection = (target.transform.position - transform.position).normalized;
        m_lookRotation = Quaternion.LookRotation(m_lookDirection);

        m_step = m_aimSpeed * Time.deltaTime;

        transform.rotation = Quaternion.Slerp(transform.rotation, m_lookRotation, m_step);
    }

    void CheckTargetState()
    {
        if(target)
        {
            LookAtTarget();
            TargetLockCheck();
        }
        else
        {
            m_inLOS = false;
            m_fired = false;
        }
    }

    void TargetLockCheck()
    {
        m_forward = transform.TransformDirection(Vector3.forward);
        m_targetDotPos = (target.transform.position - transform.position).normalized;

        if(Vector3.Dot(m_forward, m_targetDotPos) >= 0.9f)
        {
            m_inLOS = true;
        }
        else
        {
            m_inLOS = false;
        }
    }

    void CheckAttackStatus()
    {
        if(target)
        {
            if(m_inLOS && !m_fired)
            {
                m_fired = true;
                Attack();
                print("boom");
            }

            if(m_fired)
            {
                AttackCooldown();
            }
        }
    }

    void AttackCooldown()
    {
        m_attackSpeed -= Time.deltaTime;

        if(m_attackSpeed <= 0)
        {
            m_fired = false;
            m_attackSpeed = s_attackSpeed;
        }
    }

    void Attack()
    {
        if(target)
        {
            target.GetComponent<EnemyStats>().health -= m_damage;
        }
    }
}
```