```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player : MonoBehaviour
{
    public float speed;
    public bool collide;
    public float jumpForce = 5;
    public float walkSpeed = 1;
    private Rigidbody2D rb;
    private SpriteRenderer sprite;
    private Vector2 moveVelocity;

    private int eggCount = 0;
    private int redbullCount = 0;

    public GameObject lvlManager;

    public GameObject laserPrefab;

    bool onCooldown = false;
    float cooldown = 3f;
    float deathBoundray = -10f;

    //on startup (first frame)
    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
        sprite = GetComponent<SpriteRenderer>();

    }

    //on frame update
    void Update()
    {
        if(GameObject.Find("LevelManager").GetComponent<LevelManager>().gameOver)
        {
            return;
        }
        Debug.Log(cooldown);
        cooldown -= Time.deltaTime;
        if (cooldown <= 0)
        {
            onCooldown = false;
        }
        else if (cooldown > 0)
        {
            onCooldown = true;
        }
        if (onCooldown == false)
        {
            FireLaser();
        }
    }

    // called every physics step (not attached to frames)
    void FixedUpdate()
    {
        if (GameObject.Find("LevelManager").GetComponent<LevelManager>().gameOver)
        {
            return;
        }
        if (transform.position.y < deathBoundray)
        {
            GameManager.control.ResetLevel();
        }
        RaycastHit2D hit;
        hit = Physics2D.Raycast(transform.position - sprite.bounds.extents, transform.TransformDirection(Vector3.down * 0.1f));
        Ray2D surfaceCheck = new Ray2D(transform.position - sprite.bounds.extents, Vector2.down * 0.1f);
        Debug.DrawRay(transform.position - new Vector3(0, .5f, 0), Vector2.down * 0.1f, Color.red);
        Vector2 moveDirection = rb.velocity;
        //enables jumping if space key is pressed
        if (Input.GetAxis("Vertical") > 0 && hit && hit.transform != transform && hit.distance < .2f)
        {
            Debug.Log(hit.transform.gameObject.name);
            moveDirection.y = jumpForce;
        }
        //2d Vector, enables horizontal movement input
        moveDirection.x = Input.GetAxis("Horizontal") * walkSpeed;
        rb.velocity = moveDirection;
    }

    public int GetComponentCount()
    {
        return redbullCount + eggCount;
    }

    private void FireLaser()
    {

        if (eggCount == 3 && redbullCount == 1 && Input.GetKeyDown("space"))
        {
            Debug.Log("LASER");
            Instantiate(laserPrefab, new Vector3(transform.position.x + 3f, transform.position.y, 0), laserPrefab.transform.rotation);
            cooldown = 3f;
            onCooldown = true;
        }
    }

    private void OnCollisionEnter2D(Collision2D collision)
    {
        if (collision.transform.CompareTag("egg"))
        {
            eggCount++;
            Destroy(collision.gameObject);
            GameObject.Find("LevelManager").GetComponent<LevelManager>().CreateComponent("egg");
        } else if (collision.transform.CompareTag("redbull"))
        {
            redbullCount++;
            Destroy(collision.gameObject);
            GameObject.Find("LevelManager").GetComponent<LevelManager>().CreateComponent("redbull");


        } else if (collision.transform.CompareTag("birb"))
        {
            GameManager.control.ResetLevel();
        }
    }


}

```