

# Making a physics-based puzzle shooting game

In this exercise, you will create a physics-based puzzle shooting game based on a popular mobile game series. We will use basic geometric primitives to build simple 3D objects as placeholders for more complex designs and add scripts to these objects so they can be used as game objects.

The game we will create will use 3D assets but will function on a 2D plane where we will launch birds at various structures inhabited by pigs, with the goal of knocking these structures down and destroying all of the pigs.

The following scripts will be added to the project during the lesson. You can refer to these if you need to check that you have entered the correct lines of code.

## 1) Floor Controller

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

public class FloorController : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

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

    }
}
```

## 2) Piggie Controller

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

public class PiggieController : MonoBehaviour
{
    public delegate void OnPiggieDestroyAction();
    public OnPiggieDestroyAction OnPiggieDestroyed;

    // Start is called before the first frame update
    void Start()
    {

    }

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

    }

    void OnCollisionEnter(Collision collision)
    {

        //Destroy piggie based on force applied
        if (this.GetComponent<Rigidbody>().velocity.magnitude > 0.8f)
        {
            GameObject.Destroy(this.gameObject);

            if (OnPiggieDestroyed != null)
            {
                OnPiggieDestroyed();
            }
        }
    }
}
```

### 3) Birdie Controller

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

public class BirdieController : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        if (this.transform.position.y < -6f)
        {
            GameObject.Destroy(this.gameObject);
        }
    }
}
```

#### 4) Launcher Controller

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

public class LauncherController : MonoBehaviour
{
    public Camera MainCamera;
    public GameObject LauncherBase;
    public GameObject BirdiePrefab;
    public float MaxBirdieForce = 1000f;
    public float LauncherForce = 100f;

    // Start is called before the first frame update
    void Start()
    {
    }

    // Update is called once per frame
    void Update()
    {
        Vector2 mousePosition = Input.mousePosition;
        Vector3 worldPosition = MainCamera.ScreenToWorldPoint(new Vector3(mousePosition.x,
mousePosition.y, transform.position.z - MainCamera.transform.position.z));

        if (worldPosition.x > LauncherBase.transform.position.x + 1) //Only when the mouse
is to the Right of the Launcher
        {
            LauncherBase.transform.localEulerAngles = new Vector3(
                LauncherBase.transform.localEulerAngles.x,
                LauncherBase.transform.localEulerAngles.y,
                Mathf.Atan2((worldPosition.y - LauncherBase.transform.position.y),
(worldPosition.x - LauncherBase.transform.position.x)) * Mathf.Rad2Deg
            );
        }

        if (Input.GetMouseButton(0))
        {
            LauncherForce += 2f;
            LauncherForce = Mathf.Clamp(LauncherForce, 100f, MaxBirdieForce);
        }

        if (Input.GetMouseButtonUp(0))
        {
            GameObject birdieInstance = Instantiate(BirdiePrefab);
            birdieInstance.transform.position = LauncherBase.transform.position +
LauncherBase.transform.right * 1.5f;
            birdieInstance.GetComponent<Rigidbody>().AddForce(LauncherBase.transform.right
* LauncherForce);
            LauncherForce = 100f;
        }
    }
}
```

## 5) GameSceneController

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

public class GameSceneController : MonoBehaviour
{
    public TextMeshProUGUI GameText;
    public TextMeshProUGUI PowerMeterText;
    public GameObject PiggieContainer;
    public LauncherController Launcher;
    PiggieController[] piggies;
    int piggiesToDestroy;

    // Start is called before the first frame update
    void Start()
    {
        piggies = PiggieContainer.GetComponentsInChildren<PiggieController>();
        piggiesToDestroy = piggies.Length;
        foreach (PiggieController piggie in piggies)
        {
            piggie.OnPiggieDestroyed += PiggieDestroyed;
        }
    }

    void PiggieDestroyed()
    {
        piggiesToDestroy--;

        if (piggiesToDestroy == 0)
        {
            GameText.text = "You did it!";
        }
    }

    // Update is called once per frame
    void Update()
    {
        PowerMeterText.text = "Cannon Power: " + Launcher.LauncherForce;

        if (Input.GetKeyDown("r"))
        {
            SceneManager.LoadScene (SceneManager.GetActiveScene().name);
            return; // Skip the fo
        }
    }
}
```

### Extensions

- Extend the level layout
- Design a second bird 3D model
- Create a prefab for a second type of bird and alternate which one spawns when fired