Unity Projectile Guide

Creating a player

- First, create a new cube under the game hierarchy.
- Then, under the player's inspector, click "add component" and create a new script called `LaunchProjectile`.
- Open `LaunchProjectile` in your editor/IDE of choice.

Creating a `Projectile` prefab

- Create a `Projectile` prefab by inserting a new object of your choice, such as a sphere, under the game hierarchy.
- Add a rigid body to the `Projectile` by adding a component under the `Projectile` inspector, then searching for a rigid body.
- Drag the object from the hierarchy into the `assets` folder under the project space.
- Double click on the new prefab to edit it, then adjust it to your liking, then exit and save.

Launching `Projectile`s

- Add the following code to the `LaunchProjectile` script:

```
using System.Diagnostics;
using UnityEngine;

public class Movement : MonoBehaviour
{
  private const float launchVelocity = .5f;
  public GameObject projectile;

  private bool firstShot;

  private bool jumpDisabled;
  private bool lastShoot;

  private bool released;

  private Stopwatch shootStopwatch;
  private Stopwatch shootPower;
  void Start()
  {
      firstShot = true;

      released = true;

      shootStopwatch = new Stopwatch();
      shootStopwatch.Start();

      shootPower = new Stopwatch();
  }

  void Update()
  {
    if (Input.GetKey(KeyCode.F) && released && (firstShot ||
shootStopwatch.ElapsedMilliseconds > 500) &&
shootPower.ElapsedMilliseconds < 1000)
    {
        if (!lastShoot)
        {
            shootPower.Start();
            lastShoot = true;
        }
    }
    else if (lastShoot && shootPower.ElapsedMilliseconds > 500)
    {
        GameObject projectileObject = Instantiate(projectile,
            new Vector3(transform.position.x, transform.position.y + .5f,
transform.position.z),
            Quaternion.Euler(0 , 0, 0)
        );

        projectileObject.GetComponent<Rigidbody>
().AddRelativeForce(transform.forward * launchVelocity *
shootPower.ElapsedMilliseconds);

        shootStopwatch.Restart();
        shootPower.Reset();

        if (firstShot)
        {
            firstShot = false;
        }

        lastShoot = false;
        released = false;
    }

    if (!Input.GetKey(KeyCode.F))
    {
        released = true;
    }
  }
}
```

  - The above code holds a public `GameObject` which holds the projectile prefab. On each frame render, if the `f` button is being pressed, a stopwatch is started that determines how long the button is being held. When the button is released or has been pressed for the maximum amount of time, a `Projectile` is instantiated and then given a relative force, which determines the direction and velocity at which the object will travel. The stopwatch elapsed time is used with a default velocity to calculate the final release velocity. An additional timer is included to make sure the launch button is released before the next shot and that the time between shots is at least half a second.

- Finally, to finish the projectile launching, go to the player's inspector and under the component containing the script, then drag the `Projectile` prefab from the project `assets` directory into the variable slot for the public `Projectile` . This will set the variable, thus allowing the player to properly instantiate a `Projectile` object.