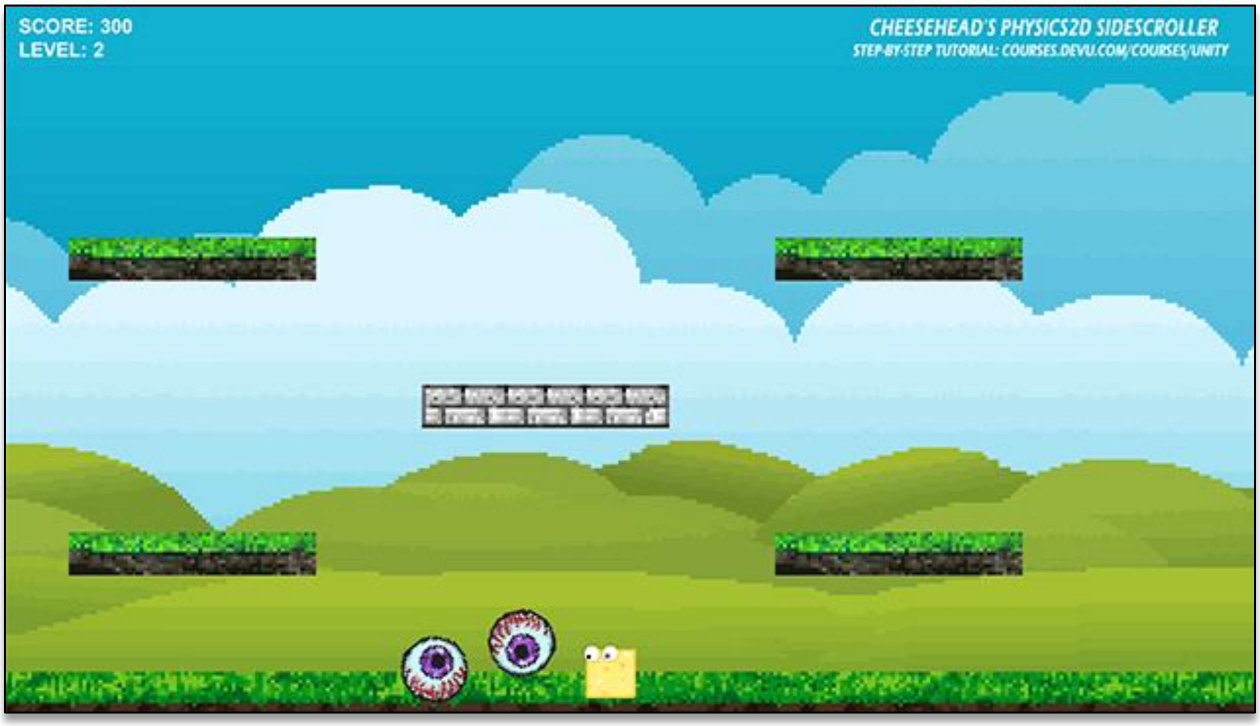
DEVU
DEVELOPER UNIVERSITY

Presents

Unity2D Physics SideScroller Demo Project
- FREE Asset Store Package -

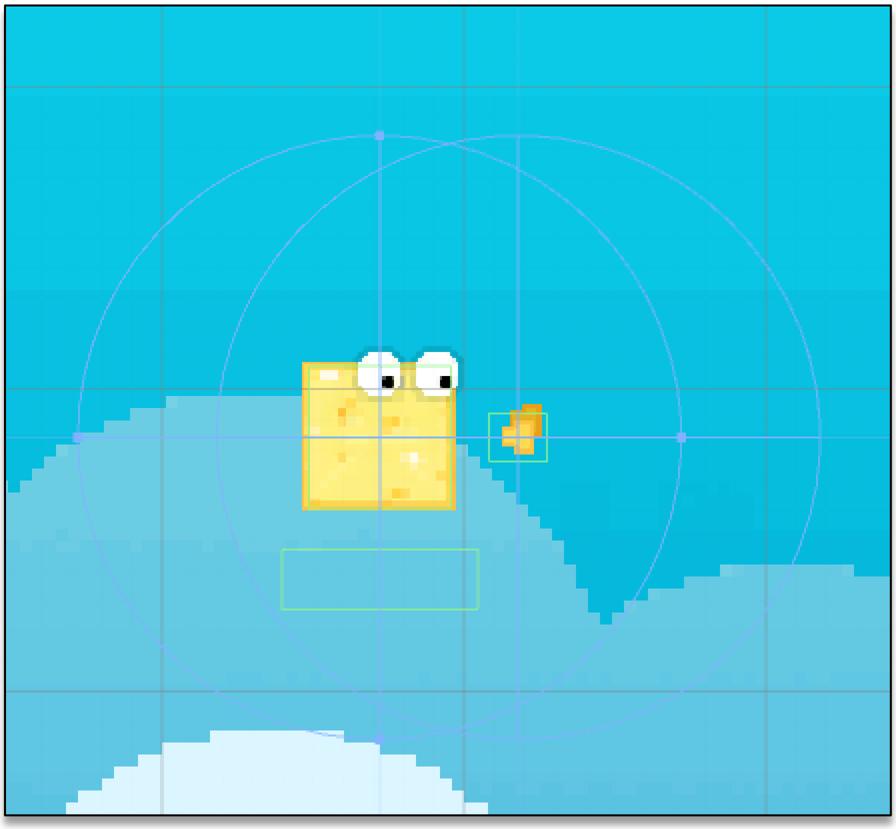
For More Info: http://www.DevU.com/Unity

See How to Make a Physics-Based 2D SideScrolling Platformer in Unity with C#!

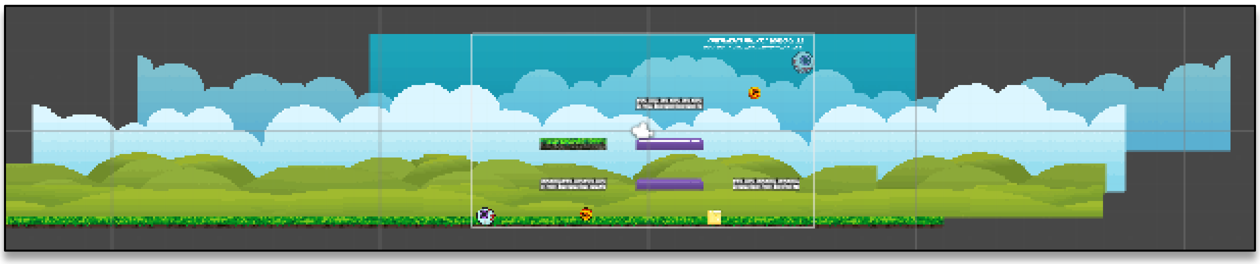

SCORE: 300
LEVEL: 2
CHEESEHEAD'S PHYSICS2D SIDESCROLLER
STEP-BY-STEP TUTORIAL: COURSES.DEVU.COM/COURSES/UNITY

This Project Demonstrates How To…

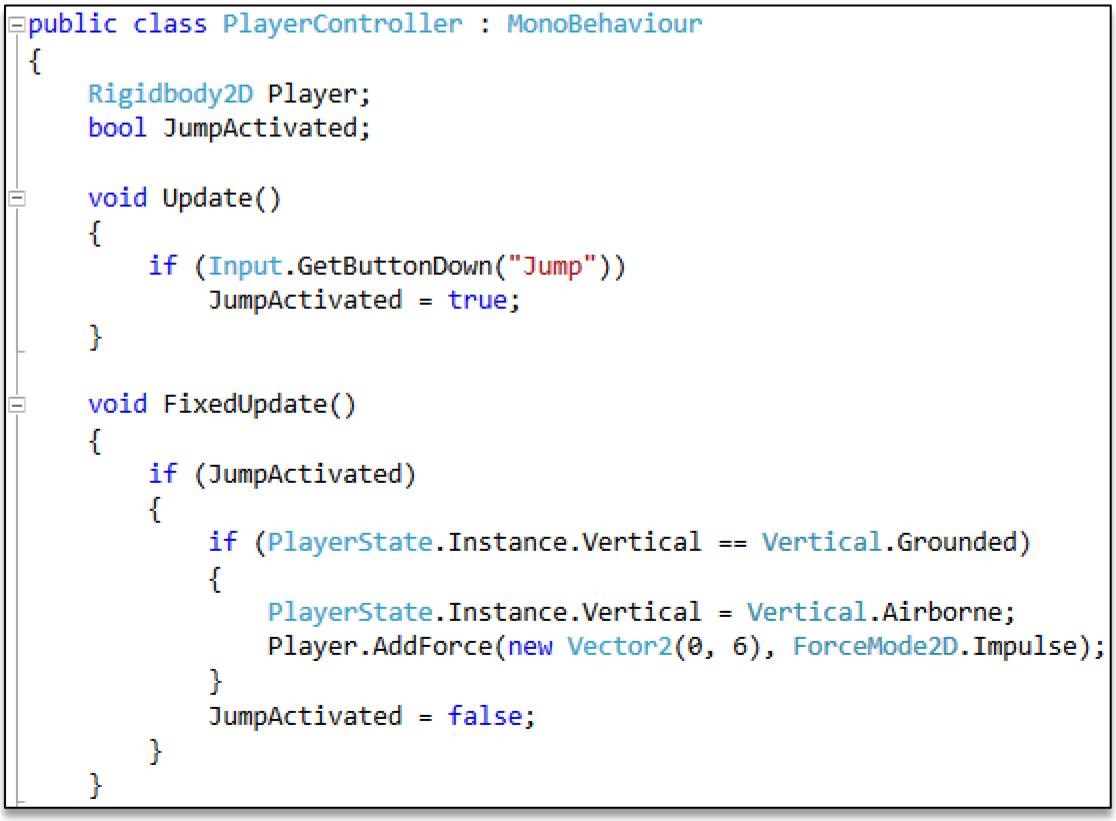
- **Work with Physics Colliders:**



- **Create Infinitely Tiling Backgrounds Along with Parallax Scrolling:**

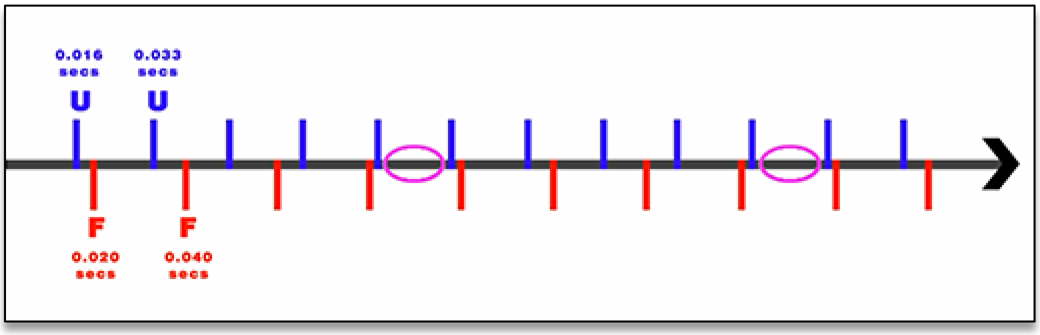


- **Access the Physics Engine in Code:**

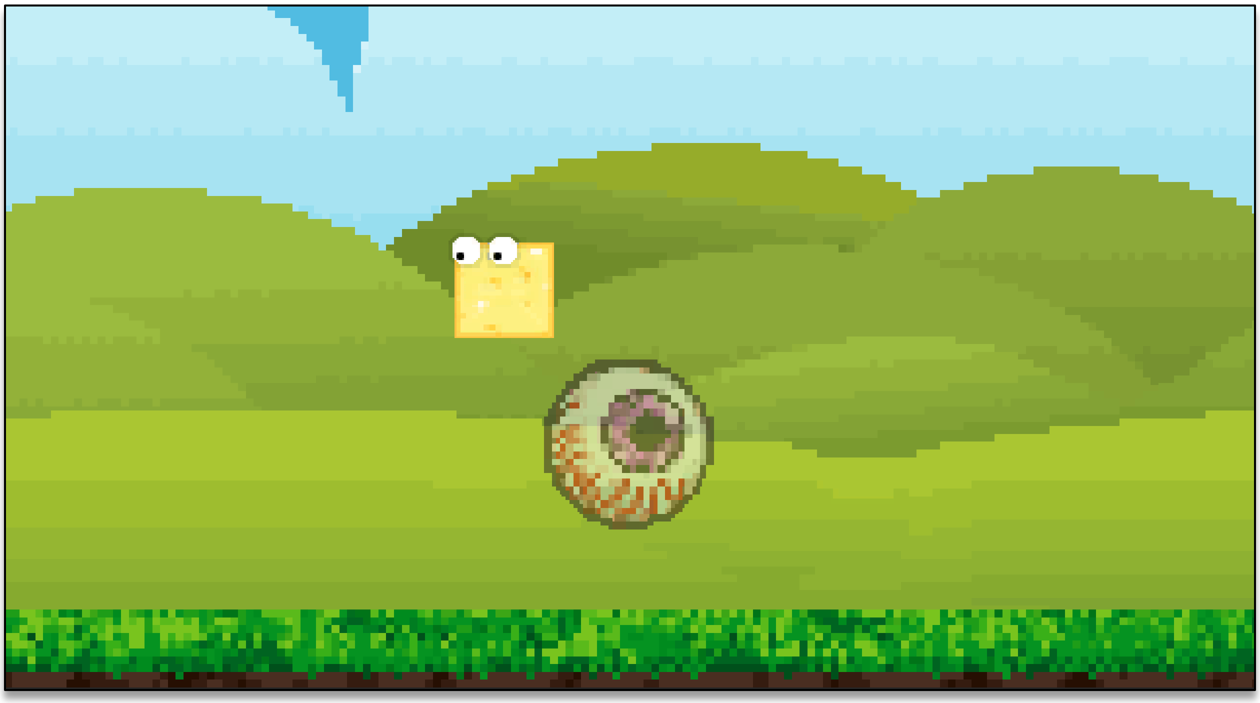


```
public class PlayerController : MonoBehaviour
{
    Rigidbody2D Player;
    bool JumpActivated;

    void Update()
    {
        if (Input.GetButtonDown("Jump"))
            JumpActivated = true;
    }

    void FixedUpdate()
    {
        if (JumpActivated)
        {
            if (PlayerState.Instance.Vertical == Vertical.Grounded)
            {
                PlayerState.Instance.Vertical = Vertical.Airborne;
                Player.AddForce(new Vector2(0, 6), ForceMode2D.Impulse);
            }
            JumpActivated = false;
        }
    }
```

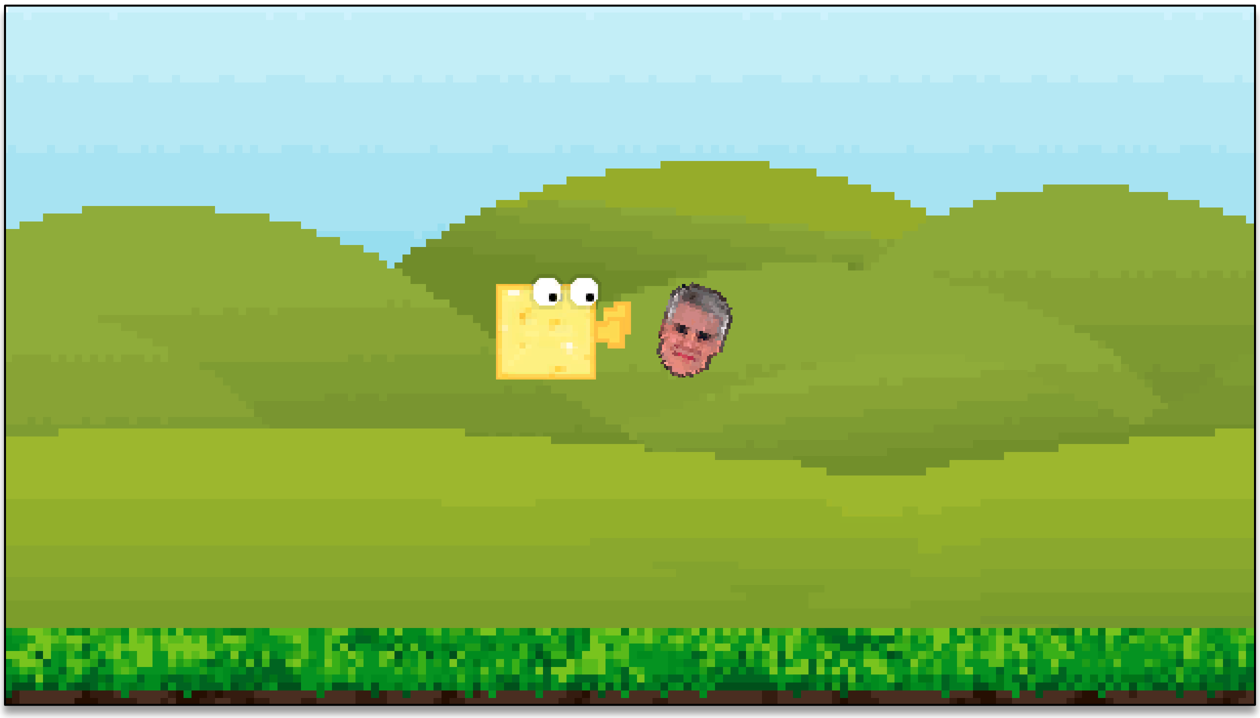
- **Use FixedUpdate() and Update() in Tandem:**



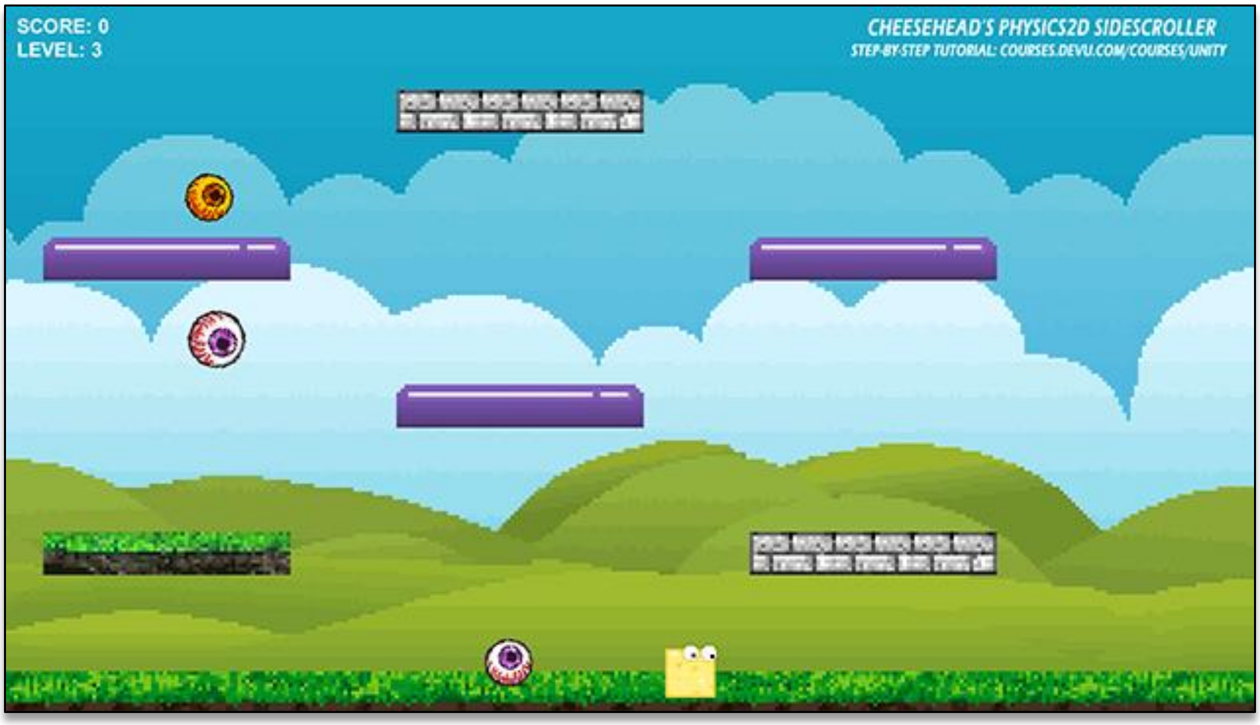
- **Use RigidBody Components that Respond to Forces and Gravity:**



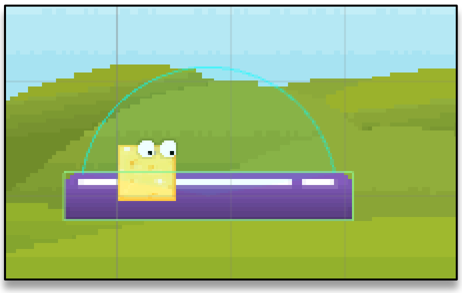
- **Create Attacks like Punching, Stomping and Throwing Projectiles:**



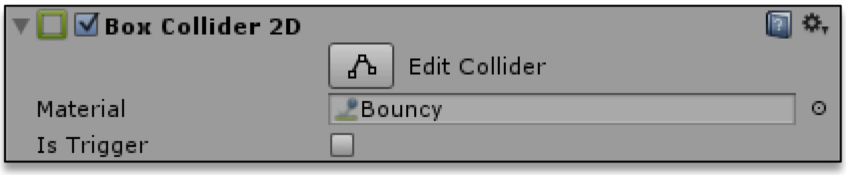
- **Procedurally Generate Enemies and Platforms:**


SCORE: 0
LEVEL: 3
CHEESEHEAD'S PHYSICS2D SIDESCROLLER
STEP-BY-STEP TUTORIAL: COURSES.DEVU.COM/COURSES/UNITY

- **Create Platforms with Physics Properties using Effectors:**



- **Work with Physics Materials:**

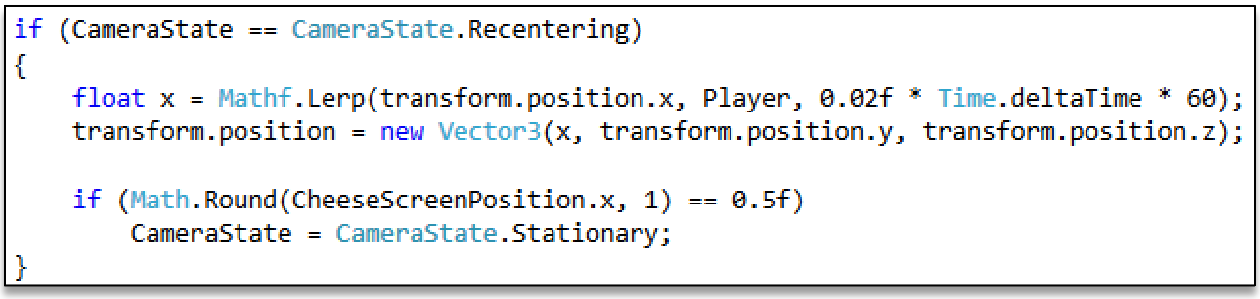


Box Collider 2D

Edit Collider

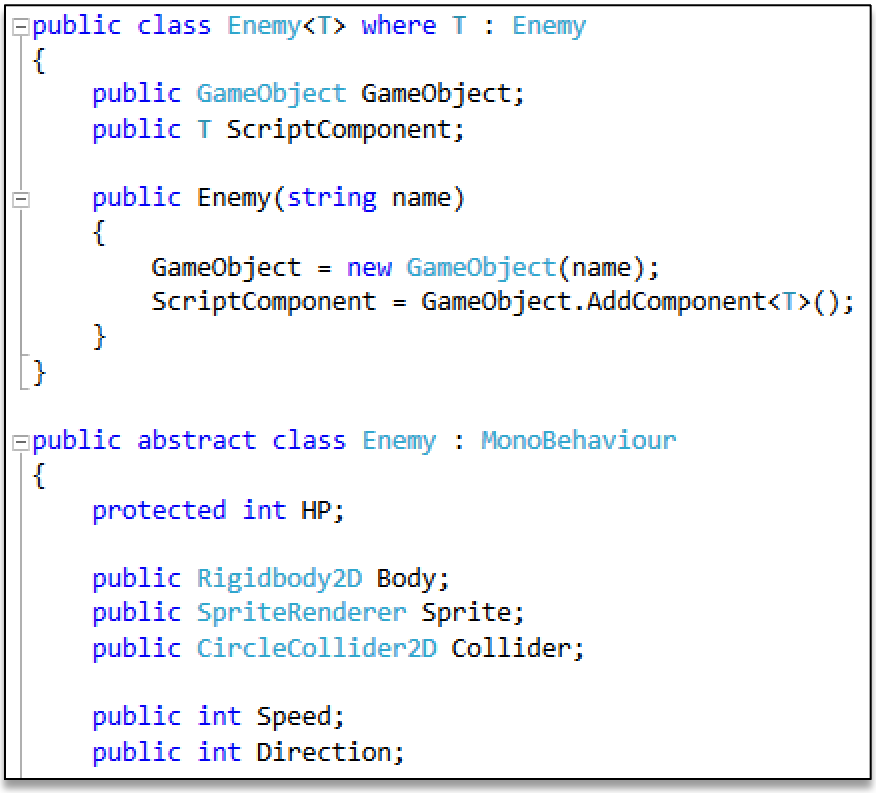
Material	Bouncy
Is Trigger	☐

- **Create Dynamic Camera Movement:**



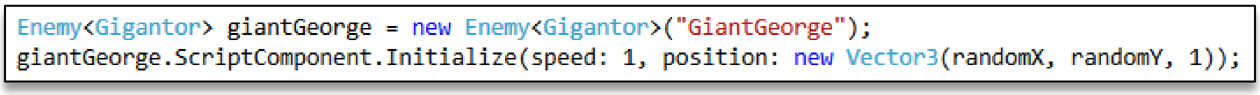
```
if (CameraState == CameraState.Recentering)
{
    float x = Mathf.Lerp(transform.position.x, Player, 0.02f * Time.deltaTime * 60);
    transform.position = new Vector3(x, transform.position.y, transform.position.z);

    if (Math.Round(CheeseScreenPosition.x, 1) == 0.5f)
        CameraState = CameraState.Stationary;
}
```

- **Create OOP Enemy Classes:**



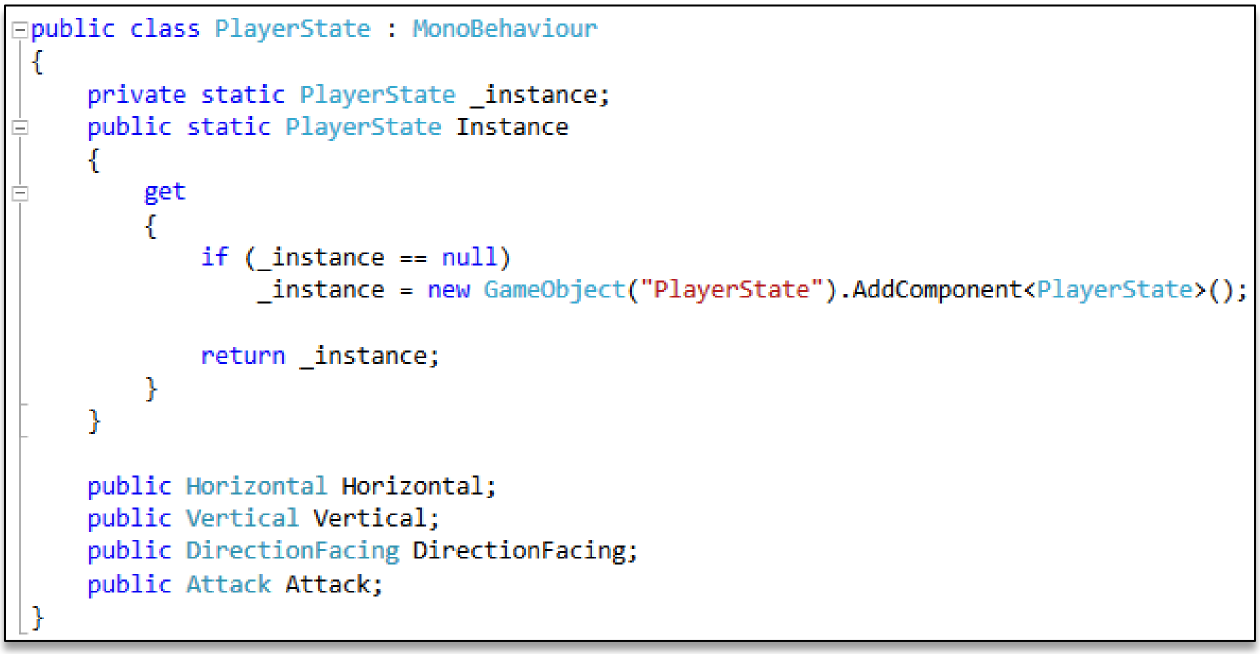
```
public class Enemy<T> where T : Enemy
{
    public GameObject GameObject;
    public T ScriptComponent;

    public Enemy(string name)
    {
        GameObject = new GameObject(name);
        ScriptComponent = GameObject.AddComponent<T>();
    }
}

public abstract class Enemy : MonoBehaviour
{
    protected int HP;

    public Rigidbody2D Body;
    public SpriteRenderer Sprite;
    public CircleCollider2D Collider;

    public int Speed;
    public int Direction;
```

- **Instantiate Enemies and Set Their Properties all Within Code:**



```
Enemy<Gigantor> giantGeorge = new Enemy<Gigantor>("GiantGeorge");
giantGeorge.ScriptComponent.Initialize(speed: 1, position: new Vector3(randomX, randomY, 1));
```

- **Use Singletons and Enums to Create and Manage States:**



```
public class PlayerState : MonoBehaviour
{
    private static PlayerState _instance;
    public static PlayerState Instance
    {
        get
        {
            if (_instance == null)
                _instance = new GameObject("PlayerState").AddComponent<PlayerState>();

            return _instance;
        }
    }

    public Horizontal Horizontal;
    public Vertical Vertical;
    public DirectionFacing DirectionFacing;
    public Attack Attack;
}
```

Also Learn How To...

- **Display Text,**
- **Utilize Collision and Trigger Events,**
- **Create Scoring Mechanics,**
- **Reference Outside GameObjects in Code,**
- **Load Assets Dynamically at Runtime,**
- **Change Framerates for Testing Purposes,**
- **Create Animations Entirely with Code,**
- **Implement a Variety of Game Design Basics using Unity and C#!**

For More Information…

See the "**Script Synopsis**" at the top of every script to better understand the purpose of each script used in this project.

For detailed **step-by-step video lesson tutorials** showing how this game was constructed - along with additional beginner-focused C# and Unity instruction - please see the course available at:

http://courses.devu.com/courses/unity

For more information on this course check out:

http://www.devu.com/unity

See Below for a List of all Lessons Included in the Course:

All Lessons for Introduction to Unity with C#