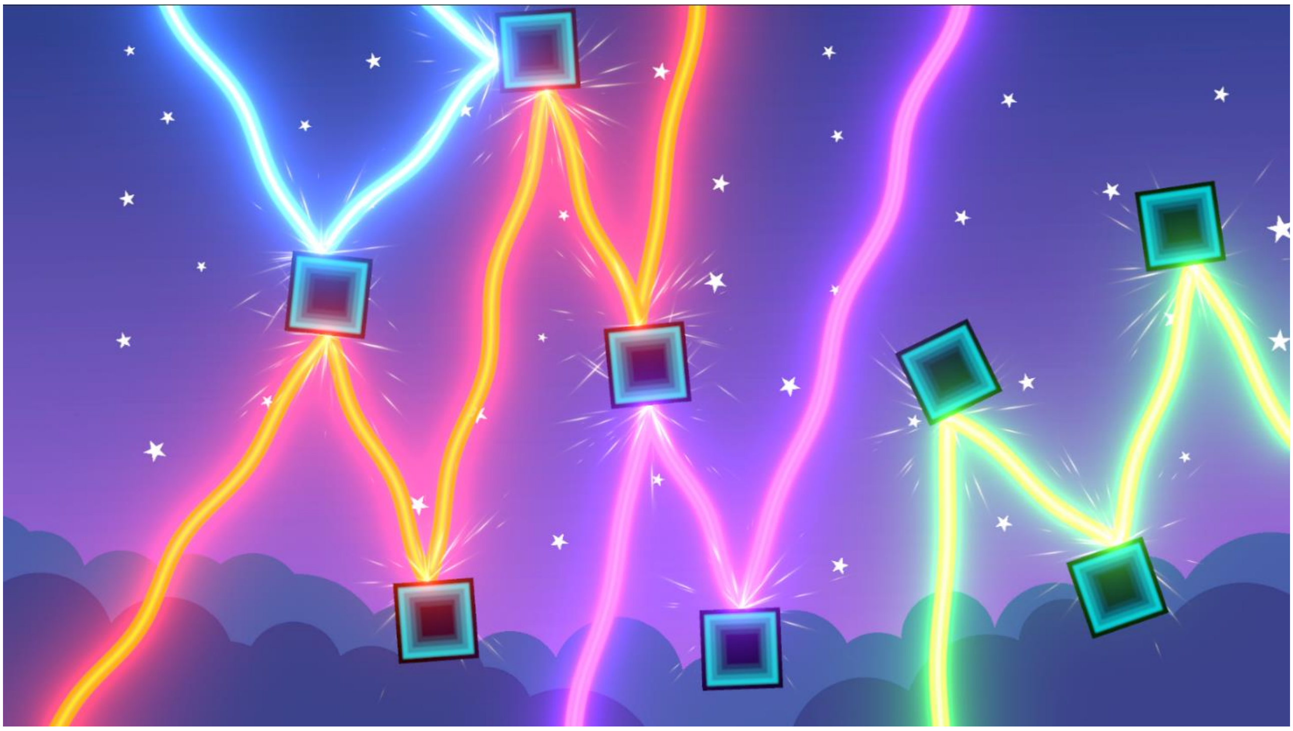
2D Laser System



Thank you for purchasing my asset. I would be grateful if you leave your feedback, it means a lot to me)

Usage. Video tutorial

1. Attach a **LaserManager** script to an empty **GameObject** (Once)
2. Attach a **Laser** script to an empty **GameObject**
3. Set laser-material to **MeshRenderer** (you can pick ready-to-use materials from the folder "2D Laser system/Material presets")
4. Here you go :)
5. (Optional) If you want glowing lasers, don’t forget to add Post-processing “Bloom” effect
6. (Optional) If you want your laser to be automatic (sort of obstacle) then attach an additional script **Automatic laser**

If you want to control your laser manually, simply use **Enable( ) / Disable( )** methods like this:

```
public class LaserGun : MonoBehaviour
{
    [SerializeField] private Transform _laserShootPoint;
    [SerializeField] private Laser _laser;

    private void Update()
    {
        if (Input.GetMouseButtonDown(0))
        {
            _laser = LaserManager.Instance.LaserPool.GetLaser(_laser);
            _laser.Enable(_laserShootPoint);
        }

        if (Input.GetMouseButtonUp(0))
        {
            _laser.Disable();
        }
    }
}
```

This line of code is optional, but it would provide a non-animating laser from the pool for a better user experience.

```
_laser = LaserManager.Instance.LaserPool.GetLaser(_laser);
```

Interaction

To add **reflections** to your objects just attach **LaserReflectingObject** component to your Collider2D.

To interact with lasers here it is 3 interfaces (similar to OnTrigger callbacks):

- ILaserEnter
- ILaserStay
- ILaserExit

Just implement some/all of them with your MonoBehaviour. **Make sure your component is attached to Collider2D.**

```
public class LaserHitObject : MonoBehaviour,
    ILaserEntered, ILaserStay, ILaserExited
{
    public void OnLaserEntered(Laser laser)
    {
        Debug.Log("Entered");
    }

    public void OnLaserStay(Laser laser)
    {
        Debug.Log("On stay");
    }

    public void OnLaserExited(Laser laser)
    {
        Debug.Log("Exited");
    }
}
```

Also, you can use **HitEvent**, for instance, to add pitch offset to your hit sound

```
public class HitSound : MonoBehaviour
{
    [SerializeField] private Laser _laser;
    [SerializeField] private AudioSource _hitAudioSource;

    private void OnEnable()
    {
        _laser.HitEvent.AddListener(PlayHitSound);
    }

    private void OnDisable()
    {
        _laser.HitEvent.RemoveListener(PlayHitSound);
    }

    private void PlayHitSound(LaserHit hit)
    {
        _hitAudioSource.pitch = // add offset;
        _hitAudioSource.Play()
    }
}
```

If you have any questions, please reach me at dunnospace@gmail.com. I would be glad to help