Raycast Controller

This project provides a modular **Raycast Controller** system in Unity to simplify detecting and responding to raycast interactions with colliders. The system is intuitive, extensible, and integrates seamlessly with the Unity Editor.

Table of Contents

Overview

The **Raycast Controller** system is designed to detect and handle collider interactions using Unity's raycasting capabilities. It provides:

- Flexible raycasting configuration for distance, layers, and runtime control.
- Event-based callbacks (`OnEnter`, `OnStay`, `OnExit`) for interaction logic.
- Debugging tools to visualize raycasts and hit points in the Scene view.

Scripts

RaycastController.cs

The **RaycastController.cs** script is a self-contained system for raycasting in Unity. It supports real-time detection and event-driven behavior for collider interactions.

Key Features:

- **Customizable Raycasting:**
    - Configurable `distance` and `layerMask` for fine-tuned detection.
    - Option to enable or disable raycasting dynamically with `castOnUpdate`.

- **Event Handling:**
  - `OnEnter`: Triggered when a new collider is detected.
  - `OnStay`: Triggered when the ray continues to hit the same collider.
  - `OnExit`: Triggered when the ray stops hitting a collider.
- **Debugging Tools:**
  - Visualize the raycast path, hit points, and hit sphere in the Scene view.
  - Customizable debug colors for the ray, hit, and sphere.

Inspector Configuration:

- **Configuration:** Set the raycast `distance`, `layerMask`, and whether to cast on every `Update`.
- **Debug Options:** Adjust the ray and sphere colors or enable hit sphere visualization.
- **Events:** Hook up UnityEvents to trigger custom behavior during interactions.

Getting Started

1. Add the **RaycastController** script to any GameObject in your scene.
2. Configure the script in the Inspector:
   - Adjust the `distance` for ray length and specify a `layerMask` for target detection.
   - Choose whether the raycasting runs on every `Update` using `castOnUpdate`.
3. Connect custom logic to the `OnEnter`, `OnStay`, and `OnExit` events in the Inspector.

Usage

Adding to a GameObject

1. Attach the **RaycastController** component to any GameObject.
2. Configure its properties in the Inspector:
   - Set `distance` to define the maximum ray length.
   - Assign a `layerMask` to limit detection to specific layers.
3. Connect UnityEvents (`OnEnter`, `OnStay`, `OnExit`) to desired functions or behaviors.

Debugging

- Enable debugging in the Inspector to visualize:
  - The raycast path with `rayColor`.

    - Hit points using `rayHitColor` and a sphere at the hit location with `sphereColor`.
- Adjust `hitSphereSize` to scale the debug sphere at hit points.

License

This project is licensed under the MIT License. See the `LICENSE` file for details.