GameDev.tv Game Jam Starter Kit

Obstacle Course Project

About

Thank you for downloading the GameDev.tv Game Jam 'Obstacle Course Project' Starter Kit!

This kit provides you the basic components you'll need to create simple moving objects. Perfect for an obstacle course but also useful for various projects.

The kit including:

- Waypoint Movement System
- Rotating Platforms
- Hinged Door

This document will guide you through the setup process and explain everything you need to get started!

Contents

About 1

Requirements 2

Asset Overview 2

Waypoint System Explained 2

Code Overview 3

- PlayerController.cs 3
- Rotator.cs 3
- WaypointFollow.cs 3

Need Help? 3

Requirements

- Unity 2021.2 or later
- Input System 1.3 or later

Asset Overview

Once imported, you will find everything organized into several folders:

- Asset Packs – Includes several models from the Polygon Prototype pack, generously provided by Synty Studios
- Input – Contains the input settings and actions for the Unity Input System
- Prefab – Includes the player and various obstacle prefabs
- Scenes – Includes the; "Sandbox" demo level
- Scripts – Contains a player controller and everything you need to make your obstacles move

Waypoint System Explained

This project contains a simple waypoint movement system that can be used to move objects along a fixed path.

In the 'Sandbox' level, the "Pushers" and the "Drillers" have a `WaypointFollow` script attached to them.

This script has several parameters to be aware of:

1. Path Container – A GameObject that lives in the hierarchy (found in the "Waypoint Paths" folder parent GameObject). This GameObject is made up of several children that map out where the path should go.
2. IsPathCircular – if true, the first and last waypoint in the path will be connected. Otherwise, the path will reverse when the follower reaches the last waypoint.
3. Path (**for testing only**) – This is a Serialized field to help you debug the path that has been created. You should remove the [SerializeField] property once you're comfortable with how the system works!

Code Overview

All of the code is organized to be as readable as possible for beginners and additional comments have been added where necessary.

However, here's a brief overview of the included classes.

PlayerController.cs

This script is attached to the "Player" prefab and is responsible for handling the movement of the player.

Input is handled using the new Unity Input system and uses 'Send Message' rather than the more complex event driven system.

Rotator.cs

This script is attached to any object that you want to rotate.
Once added, just specify the rotation speed in degrees per second.

WaypointFollow.cs

This script is attached to any object that you want to move.
See the "Waypoint System Explained" section above for more details.

Need Help?

If you're not already part of our amazing community, why not come and say hi!

- Community Forum
- Discord
- Facebook
- Twitter
- Instagram
- YouTube

We also have a ton of great courses to help you develop your skills in; Unity, Blender, Pixel Art, and more.

Swing by GameDev.tv and supercharge your game development skills today!