

AllInputHandler Documentation

Overview

The `AllInputHandler` class extends the `BaseInputHandler` and implements the `IInputHandler` interface. It is part of the **EasyFlyingSystem** under the namespace `RageRunGames`. The class controls AI movement behaviors including **follow**, **wander**, and **waypoint navigation**, while also handling obstacle avoidance and PID (Proportional-Integral-Derivative) control for smooth movements. Check out the demo scene under **Demo > AI**

Features

- **AI Behavior Types:** Choose from Follow, Wander, or Waypoint-based AI types.
- **Obstacle Avoidance:** Raycasting system detects obstacles and applies steering to avoid them.
- **PID Controllers:** Fine-tunes movement (roll, pitch, yaw, lift) for smooth AI control.
- **Waypoint Navigation:** Supports random and ping-pong waypoint traversal.
- **Gizmos Support:** Visualizes waypoints in the Unity Editor.

Public Enums

AIType

Defines the different AI behaviors:

- `Follow`: AI follows a target.
- `Wander`: AI randomly moves within a specified radius.
- `Waypoint`: AI moves through waypoints in sequence.
- `WaypointRandom`: AI moves to random waypoints.
- `WaypointPingPong`: AI moves through waypoints in a back-and-forth (ping-pong) manner.

Serialized Fields

AI Settings

- `aiType`: Defines the AI behavior (default is `Follow`).
- `targetTransform`: The transform of the target for the AI to follow.
- `lerpSpeed`: Speed at which the AI rotates towards the target (default 5).
- `releaseLerpSpeed`: Speed for releasing controls, allowing the AI to stabilize (default 10).
- `stopDistance`: Distance at which the AI will stop when near the target.

Wander Settings

- **wanderRadius**: The maximum radius for wandering (default 5).
- **wanderTimer**: Interval in seconds between choosing new random points for wandering (default 3).

Patrol Settings

- **wayPointHolder**: A transform containing the waypoints for patrol behavior.

Obstacle Avoidance Settings

- **obstacleLayer**: The layer mask for detecting obstacles.
- **obstacleDetectionDistance**: Maximum distance for detecting obstacles.
- **numberOfRays**: Number of rays cast for obstacle detection.
- **raySpreadAngle**: Spread angle of the rays for obstacle detection.
- **useLocalForwardRay**: Enables local forward direction for rays.

PID Controller

- **usePIDController**: Toggle for using PID control for smooth AI movement.

Private Fields

- **currentWaypointIndex**: Index of the current waypoint in patrol mode.
- **pingPongDirection**: Boolean to switch directions in **WaypointPingPong** mode.
- **wanderTarget**: Target position for wandering behavior.
- **rollPID**, **pitchPID**, **yawPID**, **liftPID**: Instances of **PIDController** for smooth AI rotation and movement.

Methods

Start()

- Initializes the AI settings, including the wander timer and waypoint availability check.

HandleInputs()

- Main method to handle AI inputs based on the selected **AIType**.
- Applies obstacle avoidance and calls specific AI behavior methods (**HandleFollow**, **HandleWander**, **HandlePatrol**, etc.).

DetectObstacles()

- Uses raycasting to detect obstacles in the AI's path.
- Adjusts the AI's direction based on obstacle proximity using steering calculations.

AI Behavior Handlers

- **HandleFollow(Vector3 avoidanceSteering)**: Follows the target while considering obstacle avoidance.
- **HandleWander(Vector3 avoidanceSteering)**: Randomly moves within the wander radius.
- **HandlePatrol(Vector3 avoidanceSteering)**: Moves through waypoints sequentially.
- **HandleWaypointRandom(Vector3 avoidanceSteering)**: Moves to random waypoints.
- **HandleWaypointPingPong(Vector3 avoidanceSteering)**: Moves through waypoints in ping-pong manner.

MoveTowardsTarget(Vector3 targetPosition, Vector3 avoidanceSteering)

- Moves the AI towards the target position, accounting for obstacle avoidance and applying PID control for smooth movement adjustments.

OnDrawGizmos()

- Visualizes waypoints and the path between them in the Unity Editor.

PIDController Class

Handles PID control for smooth movement transitions.

Fields:

- **pCoeff, iCoeff, dCoeff**: Coefficients for proportional, integral, and derivative control.
- **minimum, maximum**: Min and max limits for PID output.

Seek(float seekValue, float currentValue)

- Calculates and returns the smooth output value using PID logic.

Usage Example

1. Assign the **AllInputHandler** script to your AI-controlled GameObject.
2. Set the **AIType** (e.g., **Follow**, **Wander**, etc.) and configure relevant settings (e.g., target, waypoints).
3. Enable obstacle avoidance by setting the obstacle layer and parameters like detection distance and ray spread.

The AI will follow, wander, or patrol through the scene, avoiding obstacles smoothly using the PID controller system.

Remarks: I am currently working on improving the asset further and also more newer models will be added in the pack from time to time. If you have any issues, please reach me out on my email: **ragerungames@gmail.com**