

Manager Script Documentation

Overview

The `Manager` class is a Unity MonoBehaviour script designed to manage waypoint indicators on a UI canvas. It tracks multiple `IndicatorGroup` objects, handling their visibility and positioning based on the player's location and screen boundaries.

Public Fields

General Settings

- `public List<IndicatorGroup> groups`
A list containing different groups of indicators.

References

- `public Transform player`
The player's transform used to calculate distances.
- `public Camera mainCamera`
The main camera to determine screen positions.
- `public Canvas mainCanvas`
The canvas where indicators will be displayed.
- `private RectTransform canvasRect`
The rect transform of the main canvas.

Configuration

- `public float offset`
Vertical offset to adjust indicator positioning.
- `public float radiusFromCenter`
Radius for centered clamping.
- `public float screenOffset`
Offset from the screen boundaries.
- `public float SmoothClamp`
Smoothing factor for position transitions.

Editor Values (for debugging and UI adjustments)

- `public bool unfoldSettings, isAddingLayer, isAddingIndicator, isEditingLayer, addMultiple`
Various boolean flags for UI interactions.
- `public IndicatorGroup editingIndicatorGroup`
Stores the currently edited indicator group.
- `public RectTransform markerAdd`

- `public List<RectTransform> markers, targets`
Temporary lists for marker and target assignments.
-

Methods

Unity Callbacks

- `void Awake()`
Initializes `canvasRect` by retrieving the `RectTransform` component from `mainCanvas`.
- `void FixedUpdate()`
 - Monitors screen size changes and updates parameters if needed.
 - Iterates through all indicator groups and manages visibility and positioning.

Screen Management

- `private void ScreenManager()`
 - Calculates screen width, height, and boundary values.
- `private void ClampManager(IndicatorGroup indicatorGroup, Indicator indicator)`
 - Clamps indicators based on the defined `clampstyle` (Rectangle, Ellipse, Centered, None).

Indicator Positioning

- `private void GetScreenPosition(IndicatorGroup indicatorGroup, Indicator indicator)`
 - Calculates the screen position of an indicator.
- `private Vector3 OnScreenClamp(Vector3 indicatorPosition)`
 - Clamps an indicator to stay within screen boundaries.
- `private Vector2 MarkSize(RectTransform parentRect)`
 - Determines the size of a marker relative to the canvas.

Indicator Management

- `public void UpdateConfigurations()`
 - Updates indicator visibility based on group settings.
- `public void AddIndicatorList()`
 - Adds multiple indicators at once.
- `public void AddIndicator(IndicatorGroup indicatorGroup, RectTransform marker, Transform target)`
 - Adds a single indicator to a specified group.
- `public void RemoveIndicator(Transform target)`
 - Removes an indicator associated with a given target.

IndicatorGroup Class

Fields

- `public enum ClampStyle { Rectangle, Ellipse, Centered, None }`
Defines how indicators are constrained on the screen.
 - `public bool enabled`
Determines if the group is active.
 - `public float minDistance, maxDistance`
Minimum and maximum distance range for indicators.
 - `public List<Indicator> indicators`
List of individual indicators within the group.
-

Indicator Class

Fields

- `public Transform target`
The object being tracked.
 - `public RectTransform marker`
The UI marker representing the target.
 - `public Vector3 screenPosition`
The calculated screen position of the marker.
 - `public bool isOnEdge`
Whether the indicator is clamped to the screen boundary.
-

Notes

- The script dynamically adjusts indicators to ensure visibility within defined screen boundaries.
- `ClampStyle` affects how indicators are positioned when off-screen.
- The script is optimized to update indicators efficiently in `FixedUpdate()`.