

Namespace TS.GazeInteraction

Classes

[GazeInteractable](#)

Represents an interactable object that responds to gaze-based interactions.

[GazeInteractor](#)

Manages gaze-based interactions with interactables.

[GazeReticle](#)

Visual representation of the point of interaction.

[ResourcesManager](#)

Loads assets from the Resources directory.

Enums

[ReticleType](#)

Types of reticles (visual feedback) for gaze interactions.

Delegates

[GazeInteractable.OnActivated](#)

Delegate for handling the activation event.

[GazeInteractable.OnEnter](#)

Delegate for handling the gaze enter event.

[GazeInteractable.OnExit](#)

Delegate for handling the gaze exit event.

[GazeInteractable.OnStay](#)

Delegate for handling the gaze stay event.

Class GazeInteractable

Namespace: [TS.GazeInteraction](#)

Represents an interactable object that responds to gaze-based interactions.

```
public class GazeInteractable : MonoBehaviour
```

Inheritance

System.Object ← GazeInteractable

Fields

OnGazeActivated

```
public UnityEvent OnGazeActivated
```

Field Value

UnityEvent

OnGazeEnter

```
public UnityEvent OnGazeEnter
```

Field Value

UnityEvent

OnGazeExit

```
public UnityEvent OnGazeExit
```

Field Value

UnityEvent

OnGazeStay

```
public UnityEvent OnGazeStay
```

Field Value

UnityEvent

OnGazeToggle

```
public UnityEvent<bool> OnGazeToggle
```

Field Value

UnityEvent<[bool](#)>

Properties

IsActivable

Indicates whether the interactable is activable.

```
public bool IsActivable { get; }
```

Property Value

[bool](#)

IsActivated

Indicates whether the interactable is currently activated.

```
public bool IsActivated { get; }
```

Property Value

[bool](#)

IsEnabled

Indicates whether the interactable is enabled.

```
public bool IsEnabled { get; set; }
```

Property Value

[bool](#)

Methods

Activate()

Activates the interactable and invokes the activated event.

```
public void Activate()
```

Enable(bool)

Enables or disables the interactable game object.

```
public void Enable(bool enable)
```

Parameters

enable [bool](#)

True to enable, false to disable.

GazeEnter(GazeInteractor, Vector3)

Invokes the gaze enter event.

```
public void GazeEnter(GazeInteractor interactor, Vector3 point)
```

Parameters

interactor [GazeInteractor](#)

point Vector3

GazeExit(GazeInteractor)

Invokes the gaze exit event after the _exitDelay duration.

```
public void GazeExit(GazeInteractor interactor)
```

Parameters

interactor [GazeInteractor](#)

GazeStay(GazeInteractor, Vector3)

Invokes the gaze stay event.

```
public void GazeStay(GazeInteractor interactor, Vector3 point)
```

Parameters

interactor [GazeInteractor](#)

point Vector3

Events

Activated

Event triggered when the interactable is activated.

```
public event GazeInteractable.OnActivated Activated
```

Event Type

[GazeInteractable.OnActivated](#)

Enter

Event triggered when the gaze enters the interactable.

```
public event GazeInteractable.OnEnter Enter
```

Event Type

[GazeInteractable.OnEnter](#)

Exit

Event triggered when the gaze exits the interactable.

```
public event GazeInteractable.OnExit Exit
```

Event Type

[GazeInteractable.OnExit](#)

Stay

Event triggered while the gaze remains on the interactable.

```
public event GazeInteractable.OnStay Stay
```

Event Type

[GazeInteractable.OnStay](#)

Delegate GazeInteractable.OnActivated

Namespace: [TS.GazeInteraction](#)

Delegate for handling the activation event.

```
public delegate void GazeInteractable.OnActivated(GazeInteractable interactable)
```

Parameters

interactable [GazeInteractable](#)

The interactable object.

Delegate GazeInteractable.OnEnter

Namespace: [TS.GazeInteraction](#)

Delegate for handling the gaze enter event.

```
public delegate void GazeInteractable.OnEnter(GazeInteractable interactable, GazeInteractor  
interactor, Vector3 point)
```

Parameters

interactable [GazeInteractable](#)

The interactable object.

interactor [GazeInteractor](#)

The gaze interactor.

point Vector3

The point where the gaze entered.

Delegate GazeInteractable.OnExit

Namespace: [TS.GazeInteraction](#)

Delegate for handling the gaze exit event.

```
public delegate void GazeInteractable.OnExit(GazeInteractable interactable,  
GazeInteractor interactor)
```

Parameters

interactable [GazeInteractable](#)

The interactable object.

interactor [GazeInteractor](#)

The gaze interactor.

Delegate GazeInteractable.OnStay

Namespace: [TS.GazeInteraction](#)

Delegate for handling the gaze stay event.

```
public delegate void GazeInteractable.OnStay(GazeInteractable interactable, GazeInteractor  
interactor, Vector3 point)
```

Parameters

interactable [GazeInteractable](#)

The interactable object.

interactor [GazeInteractor](#)

The gaze interactor.

point Vector3

The current gaze point.

Class GazeInteractor

Namespace: [TS.GazeInteraction](#)

Manages gaze-based interactions with interactables.

```
public class GazeInteractor : MonoBehaviour
```

Inheritance

System.Object ← GazeInteractor

Class GazeReticle

Namespace: [TS.GazeInteraction](#)

Visual representation of the point of interaction.

```
public class GazeReticle : MonoBehaviour
```

Inheritance

System.Object ← GazeReticle

Methods

Enable(bool)

Toggles the visibility of the reticle GameObject.

```
public void Enable(bool enable)
```

Parameters

enable [bool](#)

Whether to enable or disable the reticle.

SetInteractor(GazeInteractor)

Assigns the GazeInteractor using this reticle.

```
public void SetInteractor(GazeInteractor interactor)
```

Parameters

interactor [GazeInteractor](#)

The GazeInteractor to associate with this reticle.

SetProgress(float)

Updates the progress visual indicator.

```
public void SetProgress(float progress)
```

Parameters

progress [float](#)

The progress value (0 to 1) to display.

SetTarget(RaycastHit)

Assigns the current hit point to adjust the reticle position and rotation.

```
public void SetTarget(RaycastHit hit)
```

Parameters

hit RaycastHit

The RaycastHit containing information about the hit point.

SetType(ReticleType)

Sets the type of reticle (visible or invisible).

```
public void SetType(ReticleType type)
```

Parameters

type [ReticleType](#)

The desired reticle type.

Class ResourceManager

Namespace: [TS.GazelInteraction](#)

Loads assets from the Resources directory.

```
public class ResourceManager : MonoBehaviour
```

Inheritance

System.Object ← ResourceManager

Fields

FILE_PREFAB_RETICLE

```
public const string FILE_PREFAB_RETICLE = "gaze_reticle"
```

Field Value

[string](#) 

Methods

GetPrefab(string)

Loads and returns the asset specified in file from the Resources directory.

```
public static GameObject GetPrefab(string file)
```

Parameters

file [string](#) 

Name of the asset to load. Use one of the constants.

Returns

GameObject

Returns the loaded GameObject.

Enum ReticleType

Namespace: [TS.GazeInteraction](#)

Types of reticles (visual feedback) for gaze interactions.

```
public enum ReticleType
```

Fields

Default = 0

Visible only on detection.

Invisible = 2

Always invisible.

Visible = 1

Always visible.