

Namespace NWH.Common

Classes

MathUtility (NWH.Common.MathUtility.html)

Mathematical utility functions for common calculations.

Class MathUtility

Mathematical utility functions for common calculations.

Inheritance

↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)

↳ MathUtility

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).

Assembly: NWH.Common.dll

Syntax

```
public static class MathUtility
```

Methods

ClampWithRemainder(**ref float, in float, out float**)

Clamps a value to a range and outputs how much it exceeded the range. Useful for clamping values while preserving overflow information.

Declaration

```
public static void ClampWithRemainder(ref float x, in float range, out float remainder)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	x	Value to clamp (will be modified).
float (https://learn.microsoft.com/dotnet/api/system.single).	range	Range limit (value will be clamped to [-range, +range]).
float (https://learn.microsoft.com/dotnet/api/system.single).	remainder	Amount by which x exceeded the range (output).

Namespace NWH.Common.AssetInfo

Classes

[AssetInfo \(NWH.Common.AssetInfo.AssetInfo.html\)](#)

ScriptableObject containing metadata and URLs for an NWH asset. Used by the welcome window and asset information systems.

Class AssetInfo

ScriptableObject containing metadata and URLs for an NWH asset. Used by the welcome window and asset information systems.

Inheritance

```
↳ object (https://learn.microsoft.com/dotnet/api/system.object)
↳ Object (https://docs.unity3d.com/ScriptReference/Object.html)
↳ ScriptableObject (https://docs.unity3d.com/ScriptReference/ScriptableObject.html)
↳ AssetInfo
```

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[AssetInfo \(NWH.Common.AssetInfo.html\)](#).

Assembly: NWH.Common.dll

Syntax

```
[CreateAssetMenu(fileName = "AssetInfo", menuName = "NWH/AssetInfo", order = 0)]
public class AssetInfo : ScriptableObject
```

Fields

assetName

Display name of the asset.

Declaration

```
public string assetName
```

Field Value

Type	Description
string (https://learn.microsoft.com/dotnet/api/system.string).	

assetURL

Unity Asset Store URL for this asset.

Declaration

```
public string assetURL
```

Field Value

Type	Description
string .	

changelogURL

URL to the changelog documentation page.

Declaration

```
public string changelogURL
```

Field Value

Type	Description
string .	

discordURL

Discord server invite link for support and community.

Declaration

```
public string discordURL
```

Field Value

Type	Description
string .	

documentationURL

URL to the main documentation page.

Declaration

```
public string documentationURL
```

Field Value

Type	Description
string .	

emailURL

Support email contact link.

Declaration

```
public string emailURL
```

Field Value

Type	Description
string (https://learn.microsoft.com/dotnet/api/system.string).	

forumURL

Unity Forum thread URL for this asset.

Declaration

```
public string forumURL
```

Field Value

Type	Description
string (https://learn.microsoft.com/dotnet/api/system.string).	

publisherURL

NWH publisher page URL on Unity Asset Store.

Declaration

```
public string publisherURL
```

Field Value

Type	Description
string (https://learn.microsoft.com/dotnet/api/system.string).	

quickStartURL

URL to quick start guide documentation.

Declaration

```
public string quickStartURL
```

Field Value

Type	Description
<code>string_(https://learn.microsoft.com/dotnet/api/system.string)</code> .	

recentUpdates

Recent updates/changes in the current version (3-5 bullet points).

Declaration

```
[TextArea(3, 10)]  
public string[] recentUpdates
```

Field Value

Type	Description
<code>string_(https://learn.microsoft.com/dotnet/api/system.string).[]</code>	

upgradeNotesURL

URL to upgrade notes between versions.

Declaration

```
public string upgradeNotesURL
```

Field Value

Type	Description
<code>string_(https://learn.microsoft.com/dotnet/api/system.string)</code> .	

version

Current version string of the asset.

Declaration

```
public string version
```

Field Value

Type	Description
<code>string_(https://learn.microsoft.com/dotnet/api/system.string)</code> .	

Namespace NWH.Common.Cameras

Classes

CameraChanger (NWH.Common.Cameras.CameraChanger.html)

Switches between the camera objects that are children to this object and contain camera tag, in order they appear in the hierarchy or in order they are added to the vehicle cameras list.

CameraInsideVehicle (NWH.Common.Cameras.CameraInsideVehicle.html)

Empty component that should be attached to the cameras that are inside the vehicle if interior sound change is to be used.

CameraMouseDrag (NWH.Common.Cameras.CameraMouseDrag.html)

Camera that can be dragged with the mouse.

VehicleCamera (NWH.Common.Cameras.VehicleCamera.html)

Base class for vehicle camera implementations with automatic target detection.

Enums

CameraMouseDrag.POVType (NWH.Common.Cameras.CameraMouseDrag.POVType.html)

Class CameraChanger

Switches between the camera objects that are children to this object and contain camera tag, in order they appear in the hierarchy or in order they are added to the vehicle cameras list.

Inheritance

- ↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)
- ↳ [Object](https://docs.unity3d.com/ScriptReference/Object.html) (<https://docs.unity3d.com/ScriptReference/Object.html>)
 - ↳ [Component](https://docs.unity3d.com/ScriptReference/Component.html) (<https://docs.unity3d.com/ScriptReference/Component.html>)
 - ↳ [Behaviour](https://docs.unity3d.com/ScriptReference/Behaviour.html) (<https://docs.unity3d.com/ScriptReference/Behaviour.html>)
 - ↳ [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html) (<https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>)
 - ↳ CameraChanger

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Cameras \(NWH.Common.Cameras.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
[DefaultExecutionOrder(20)]  
public class CameraChanger : MonoBehaviour
```

Fields

autoFindCameras

If true vehicleCameras list will be filled through cameraTag.

Declaration

```
[Tooltip("    If true vehicleCameras list will be filled through cameraTag.")]  
public bool autoFindCameras
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	

cameras

List of cameras that the changer will cycle through. Leave empty if you want cameras to be automatically detected. To be detected cameras need to have camera tag and be children of the object this script is attached to.

Declaration

```
[FormerlySerializedAs("vehicleCameras")]
[Tooltip("List of cameras that the changer will cycle through. Leave empty if you want cameras to be automatically detected. To be detected cameras need to have camera tag and be children of the object this script is attached to.")]
public List<GameObject> cameras
```

Field Value

Type	Description
List< https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1 >< GameObject (https://docs.unity3d.com/ScriptReference/GameObject.html)>	

currentCameraIndex

Index of the camera from vehicle cameras list that will be active first.

Declaration

```
[Tooltip("Index of the camera from vehicle cameras list that will be active first.")]
public int currentCameraIndex
```

Field Value

Type	Description
int (https://learn.microsoft.com/dotnet/api/system.int32),	

Methods

NextCamera()

Activates the next camera in the list, cycling back to the first camera when reaching the end. Automatically disables all other cameras and their AudioListeners.

Declaration

```
public void NextCamera()
```

PreviousCamera()

Activates the previous camera in the list, cycling back to the last camera when reaching the beginning. Automatically disables all other cameras and their AudioListeners.

Declaration

```
public void PreviousCamera()
```

Class CameraInsideVehicle

Empty component that should be attached to the cameras that are inside the vehicle if interior sound change is to be used.

Inheritance

- ↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)
- ↳ [Object](https://docs.unity3d.com/ScriptReference/Object.html) (<https://docs.unity3d.com/ScriptReference/Object.html>)
- ↳ [Component](https://docs.unity3d.com/ScriptReference/Component.html) (<https://docs.unity3d.com/ScriptReference/Component.html>)
- ↳ [Behaviour](https://docs.unity3d.com/ScriptReference/Behaviour.html) (<https://docs.unity3d.com/ScriptReference/Behaviour.html>)
- ↳ [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html) (<https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>)
- ↳ CameraInsideVehicle

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Cameras \(NWH.Common.Cameras.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public class CameraInsideVehicle : MonoBehaviour
```

Fields

isInsideVehicle

Is the camera inside vehicle?

Declaration

```
[Tooltip("    Is the camera inside vehicle?")]
public bool isInsideVehicle
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

Class CameraMouseDrag

Camera that can be dragged with the mouse.

Inheritance

```
↳ object (https://learn.microsoft.com/dotnet/api/system.object)
  ↳ Object (https://docs.unity3d.com/ScriptReference/Object.html)
    ↳ Component (https://docs.unity3d.com/ScriptReference/Component.html)
      ↳ Behaviour (https://docs.unity3d.com/ScriptReference/Behaviour.html)
        ↳ MonoBehaviour (https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)
          ↳ VehicleCamera (NWH.Common.Cameras.VehicleCamera.html)
            ↳ CameraMouseDrag
```

Inherited Members

[VehicleCamera.target](#)

(NWH.Common.Cameras.VehicleCamera.html#[NWH Common Cameras VehicleCamera target](#)).

[VehicleCamera.Awake\(\)](#)

(NWH.Common.Cameras.VehicleCamera.html#[NWH Common Cameras VehicleCamera Awake](#)).

Namespace: [NWH \(NWH.html\)](#), [Common \(NWH.Common.html\)](#), [Cameras \(NWH.Common.Cameras.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public class CameraMouseDrag : VehicleCamera
```

Fields

allowPanning

Can the camera be panned by the user?

Declaration

```
[Tooltip("    Can the camera be panned by the user?")]
public bool allowPanning
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	

allowRotation

Can the camera be rotated by the user?

Declaration

```
[Tooltip("    Can the camera be rotated by the user?")]
public bool allowRotation
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	

distance

Distance from target at which camera will be positioned. Might vary depending on smoothing.

Declaration

```
[Range(0, 100)]
[Tooltip("    Distance from target at which camera will be positioned. Might vary depending
on smoothing.")]
public float distance
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

followTargetPitchAndYaw

If true the camera will rotate with the vehicle along the X and Y axis.

Declaration

```
[FormerlySerializedAs("followTargetsRotation")]
[Tooltip("    If true the camera will rotate with the vehicle along the X and Y axis.")]
public bool followTargetPitchAndYaw
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	

followTargetRoll

If true the camera will rotate with the vehicle along the Z axis.

Declaration

```
[Tooltip("    If true the camera will rotate with the vehicle along the Z axis.")]  
public bool followTargetRoll
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

initXRotation

Initial rotation around the X axis (up/down)

Declaration

```
[Tooltip("    Initial rotation around the X axis (up/down)")]  
public float initXRotation
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single)	

initYRotation

Initial rotation around the Y axis (left/right)

Declaration

```
[Tooltip("    Initial rotation around the Y axis (left/right)")]  
public float initYRotation
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single)	

maxDistance

Maximum distance that will be reached when zooming out.

Declaration

```
[Range(0, 100)]  
[Tooltip("    Maximum distance that will be reached when zooming out.")]  
public float maxDistance
```

Field Value

Type	Description
float	

minDistance

Minimum distance that will be reached when zooming in.

Declaration

```
[Range(0, 100)]  
[Tooltip("    Minimum distance that will be reached when zooming in.")]  
public float minDistance
```

Field Value

Type	Description
float	

panningSensitivity

Sensitivity of panning input.

Declaration

```
[Tooltip("    Sensitivity of panning input.")]  
public Vector2 panningSensitivity
```

Field Value

Type	Description
Vector2	

povType

Camera POV type. First person camera will invert controls. Zoom is not available in 1st person.

Declaration

```
[Tooltip("Camera POV type. First person camera will invert controls.\r\nZoom is not available in 1st person.")]
public CameraMouseDrag.POVType povType
```

Field Value

Type	Description
CameraMouseDrag (NWH.Common.Cameras.CameraMouseDrag.html) . POVType (NWH.Common.Cameras.CameraMouseDrag.POVType.html) .	

rotationSensitivity

Sensitivity of rotation input.

Declaration

```
[Tooltip("    Sensitivity of rotation input.")]
public Vector2 rotationSensitivity
```

Field Value

Type	Description
Vector2 (https://docs.unity3d.com/ScriptReference/Vector2.html) .	

rotationSmoothing

Smoothing of the camera rotation.

Declaration

```
[Range(0, 1)]
[Tooltip("    Smoothing of the camera rotation.")]
public float rotationSmoothing
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single) .	

shakeAxisIntensity

Movement intensity per axis. Set to 0 to disable movement on that axis or negative to reverse it.

Declaration

```
[Tooltip("    Movement intensity per axis. Set to 0 to disable movement on that axis or negative to reverse it.")]
public Vector3 shakeAxisIntensity
```

Field Value

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html)	

shakeIntensity

How much will the head move around for the given g-force.

Declaration

```
[Range(0, 1)]
[Tooltip("    How much will the head move around for the given g-force.")]
public float shakeIntensity
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single)	

shakeMaxOffset

Maximum head movement from the initial position.

Declaration

```
[Range(0, 1)]
[Tooltip("    Maximum head movement from the initial position.")]
public float shakeMaxOffset
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single)	

shakeSmoothing

Smoothing of the head movement.

Declaration

```
[Range(0, 1)]  
[Tooltip("    Smoothing of the head movement.")]  
public float shakeSmoothing
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

targetPositionOffset

Look position offset from the target center.

Declaration

```
[Tooltip("    Look position offset from the target center.")]  
public Vector3 targetPositionOffset
```

Field Value

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	

useShake

Should camera movement on acceleration be used?

Declaration

```
[Tooltip("Should camera movement on acceleration be used?")]  
public bool useShake
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	

verticalMaxAngle

Maximum vertical angle the camera can achieve.

Declaration

```
[Range(-90, 90)]  
[Tooltip("    Maximum vertical angle the camera can achieve.")]  
public float verticalMaxAngle
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

verticalMinAngle

Minimum vertical angle the camera can achieve.

Declaration

```
[Range(-90, 90)]  
[Tooltip("    Minimum vertical angle the camera can achieve.")]  
public float verticalMinAngle
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

zoomSensitivity

Sensitivity of the middle mouse button / wheel.

Declaration

```
[Range(0, 15)]  
[Tooltip("    Sensitivity of the middle mouse button / wheel.")]  
public float zoomSensitivity
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

Methods

ClampAngle(float, float, float)

Declaration

```
public float ClampAngle(float angle, float min, float max)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single)	<i>angle</i>	
float (https://learn.microsoft.com/dotnet/api/system.single)	<i>min</i>	
float (https://learn.microsoft.com/dotnet/api/system.single)	<i>max</i>	

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single)	

OnDrawGizmosSelected()

Declaration

```
public void OnDrawGizmosSelected()
```

Enum CameraMouseDrag.POVTType

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Cameras \(NWH.Common.Cameras.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public enum CameraMouseDrag.POVTType
```

Fields

Name	Description
FirstPerson	First-person camera view from inside the vehicle.
ThirdPerson	Third-person camera view from outside the vehicle.

Class VehicleCamera

Base class for vehicle camera implementations with automatic target detection.

Inheritance

```
↳ object (https://learn.microsoft.com/dotnet/api/system.object)
  ↳ Object (https://docs.unity3d.com/ScriptReference/Object.html)
    ↳ Component (https://docs.unity3d.com/ScriptReference/Component.html)
      ↳ Behaviour (https://docs.unity3d.com/ScriptReference/Behaviour.html)
        ↳ MonoBehaviour (https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)
          ↳ VehicleCamera
            ↳ CameraMouseDrag (NWH.Common.Cameras.CameraMouseDrag.html).
```

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Cameras \(NWH.Common.Cameras.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public class VehicleCamera : MonoBehaviour
```

Fields

target

Transform to track. Auto-detects parent Rigidbody if not assigned.

Declaration

```
[Tooltip("Transform that this script is targeting. Can be left empty if head movement is not being used.")]
public Transform target
```

Field Value

Type	Description
Transform (https://docs.unity3d.com/ScriptReference/Transform.html)	

Methods

Awake()

Declaration

```
public virtual void Awake()
```

See Also

[CameraChanger \(NWH.Common.Cameras.CameraChanger.html\)](#).

[CameraInsideVehicle \(NWH.Common.Cameras.CameraInsideVehicle.html\)](#).

[CameraMouseDrag \(NWH.Common.Cameras.CameraMouseDrag.html\)](#).

Namespace NWH.Common.CoM

Classes

[MassAffector \(NWH.Common.CoM.MassAffector.html\)](#)

Simple mass affector implementation that contributes a fixed mass at its transform position to the vehicle's center of mass calculations.

[VariableCenterOfMass \(NWH.Common.CoM.VariableCenterOfMass.html\)](#)

Dynamic center of mass and inertia calculation system that updates Rigidbody properties based on attached mass affectors like fuel tanks, cargo loads, and passengers.

Interfaces

[IMassAffector \(NWH.Common.CoM.ICollection<IMassAffector>.html\)](#)

Interface for objects that contribute mass and affect vehicle center of mass calculations. Implemented by fuel tanks, cargo systems, and other variable mass components.

Interface IMassAffecter

Interface for objects that contribute mass and affect vehicle center of mass calculations. Implemented by fuel tanks, cargo systems, and other variable mass components.

Namespace: [NWH \(NWH.html\)](#), [Common \(NWH.Common.html\)](#), [CoM \(NWH.Common.CoM.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public interface IMassAffecter
```

Remarks

Mass affectors allow dynamic vehicle physics by contributing their mass and position to the overall center of mass calculation. As fuel depletes or cargo loads change, the vehicle's handling characteristics update automatically.

Methods

GetMass()

Current mass of this affector in kilograms. Should return variable values for fuel tanks, cargo, etc.

Declaration

```
float GetMass()
```

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single) .	Mass in kg

GetTransform()

Returns transform of the mass affector.

Declaration

```
Transform GetTransform()
```

Returns

Type	Description
Transform (https://docs.unity3d.com/ScriptReference/Transform.html) .	

GetWorldCenterOfMass()

World position of this affector's center of mass. Used for weighted center of mass calculations.

Declaration

```
Vector3 GetWorldCenterOfMass()
```

Returns

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	

Class MassAffector

Simple mass affector implementation that contributes a fixed mass at its transform position to the vehicle's center of mass calculations.

Inheritance

- ↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)
- ↳ [Object](https://docs.unity3d.com/ScriptReference/Object.html) (<https://docs.unity3d.com/ScriptReference/Object.html>)
- ↳ [Component](https://docs.unity3d.com/ScriptReference/Component.html) (<https://docs.unity3d.com/ScriptReference/Component.html>)
- ↳ [Behaviour](https://docs.unity3d.com/ScriptReference/Behaviour.html) (<https://docs.unity3d.com/ScriptReference/Behaviour.html>)
- ↳ [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html) (<https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>)
- ↳ MassAffector

Implements

[IMassAffector](#) ([NWH.Common.CoM.IMassAffector.html](#))

Namespace: [NWH](#) ([NWH.html](#)), [Common](#) ([NWH.Common.html](#)), [CoM](#) ([NWH.Common.CoM.html](#))

Assembly: NWH.Common.dll

Syntax

```
public class MassAffector : MonoBehaviour, IMassAffector
```

Remarks

Use this component for static mass contributions like passengers, cargo, or equipment. For dynamic masses like fuel tanks, create a custom IMassAffector implementation that returns varying mass values.

Fields

mass

Mass contribution of this affector in kilograms.

Declaration

```
public float mass
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

Methods

GetMass()

Returns the mass of this affector.

Declaration

```
public float GetMass()
```

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Mass in kilograms.

GetTransform()

Returns the transform of this mass affector.

Declaration

```
public Transform GetTransform()
```

Returns

Type	Description
Transform (https://docs.unity3d.com/ScriptReference/Transform.html).	

GetWorldCenterOfMass()

Returns the world position of this mass affector's center of mass.

Declaration

```
public Vector3 GetWorldCenterOfMass()
```

Returns

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	

Implements

[IMassAffector](https://NWH.Common.CoM.IMassAffector.html) (NWH.Common.CoM.IMassAffector.html).

Class VariableCenterOfMass

Dynamic center of mass and inertia calculation system that updates Rigidbody properties based on attached mass affectors like fuel tanks, cargo loads, and passengers.

Inheritance

```
↳ object (https://learn.microsoft.com/dotnet/api/system.object)
↳ Object (https://docs.unity3d.com/ScriptReference/Object.html)
    ↳ Component (https://docs.unity3d.com/ScriptReference/Component.html)
        ↳ Behaviour (https://docs.unity3d.com/ScriptReference/Behaviour.html)
            ↳ MonoBehaviour (https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)
                ↳ VariableCenterOfMass
```

Namespace: [NWH \(NWH.html\)](#), [Common \(NWH.Common.html\)](#), [CoM \(NWH.Common.CoM.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
[DisallowMultipleComponent]
[DefaultExecutionOrder(-1000)]
[RequireComponent(typeof(Rigidbody))]
public class VariableCenterOfMass : MonoBehaviour
```

Remarks

VariableCenterOfMass enables realistic vehicle physics behavior by automatically adjusting center of mass and inertia tensor as vehicle loading changes. This affects handling characteristics, stability, and acceleration response without requiring complex rigidbody hierarchies.

The system calculates total mass, weighted center of mass position, and inertia contributions from all IMassAffecter components. Changes in fuel level, cargo loading, or passenger weight immediately affect vehicle dynamics, creating realistic weight distribution effects.

Critical for vehicle realism: Front-heavy vehicles understeer more, rear-heavy vehicles may oversteer, and high center of mass increases rollover tendency. The system updates these characteristics dynamically based on actual mass distribution.

Fields

affectors

Objects attached or part of the vehicle affecting its center of mass and inertia.

Declaration

```
[NonSerialized]
public IMassAffecter[] affectors
```

Field Value

Type	Description
IMassAffector (<i>NWH.Common.CoM.IMassAffector.html</i>)[]	

baseMass

Base mass of the object, without IMassAffectors.

Declaration

```
[Tooltip("Base mass of the object, without IMassAffectors.")]
public float baseMass
```

Field Value

Type	Description
float (<i>https://learn.microsoft.com/dotnet/api/system.single</i>) .	

centerOfMass

Center of mass of the object. Auto calculated. To adjust center of mass use centerOfMassOffset.

Declaration

```
[Tooltip("Center of mass of the rigidbody. Needs to be readjusted when new colliders are added.")]
public Vector3 centerOfMass
```

Field Value

Type	Description
Vector3 (<i>https://docs.unity3d.com/ScriptReference/Vector3.html</i>) .	

combinedCenterOfMass

Combined center of mass, including the Rigidbody and any IMassAffectors.

Declaration

```
public Vector3 combinedCenterOfMass
```

Field Value

Type	Description
Vector3 (<i>https://docs.unity3d.com/ScriptReference/Vector3.html</i>) .	

combinedInertiaTensor

Total inertia tensor. Includes Rigidbody and IMassAffectors.

Declaration

```
public Vector3 combinedInertiaTensor
```

Field Value

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html) .	

combinedMass

Total mass of the object with masses of IMassAffectors counted in.

Declaration

```
[Tooltip("Total mass of the object with masses of IMassAffectors counted in.")]  
public float combinedMass
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single) .	

dimensions

Object dimensions in [m]. X - width, Y - height, Z - length. It is important to set the correct dimensions or otherwise inertia might be calculated incorrectly.

Declaration

```
[Tooltip("Object dimensions in [m]. X - width, Y - height, Z - length.\r\nIt is important to  
set the correct dimensions or otherwise inertia might be calculated incorrectly.")]  
public Vector3 dimensions
```

Field Value

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html) .	

inertiaTensor

Vector by which the inertia tensor of the rigidbody will be scaled on Start(). Due to the uniform density of the rigidbodies, versus the very non-uniform density of a vehicle, inertia can feel off. Use this to adjust inertia tensor values.

Declaration

```
[Tooltip("    Vector by which the inertia tensor of the rigidbody will be scaled on Start  
().\r\n    Due to the uniform density of the rigidbodies, versus the very non-uniform density  
of a vehicle, inertia can feel\r\n    off.\r\n    Use this to adjust inertia tensor value  
s.")]  
public Vector3 inertiaTensor
```

Field Value

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html) .	

isDirty

When true, properties will be recalculated in the next FixedUpdate. Call `MarkDirty()` when mass affectors change to trigger update.

Declaration

```
[Tooltip("When true, properties will be recalculated in the next FixedUpdate. Automatically  
managed.")]  
public bool isDirty
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean) .	

useDefaultCenterOfMass

When enabled the Unity-calculated center of mass will be used.

Declaration

```
[Tooltip("When enabled the Unity-calculated center of mass will be used.")]  
public bool useDefaultCenterOfMass
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean) .	

useDefaultInertia

When true inertia settings will be ignored and default Rigidbody inertia tensor will be used.

Declaration

```
[Tooltip("When true inertia settings will be ignored and default Rigidbody inertia tensor will be used.")]  
public bool useDefaultInertia
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

useDefaultMass

Should the default Rigidbody mass be used?

Declaration

```
public bool useDefaultMass
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

useMassAffectors

If true, the script will search for any IMassAffectors attached as a child (recursively) of this script and use them when calculating mass, center of mass and inertia tensor.

Declaration

```
public bool useMassAffectors
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

Methods

CalculateInertia(Vector3, float)

Calculates inertia tensor for a cuboid with given dimensions and mass. Uses parallel axis theorem for rectangular prism approximation.

Declaration

```
public static Vector3 CalculateInertia(Vector3 dimensions, float mass)
```

Parameters

Type	Name	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	<i>dimensions</i>	Object dimensions in meters (width, height, length)
float (https://learn.microsoft.com/dotnet/api/system.single).	<i>mass</i>	Total mass in kilograms

Returns

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	Inertia tensor components (Ix, Iy, Iz) in kg·m ²

CalculateInertiaTensorOffset(Vector3)

Calculates the inertia tensor of the Rigidbody and attached mass affectors.

Declaration

```
public Vector3 CalculateInertiaTensorOffset(Vector3 dimensions)
```

Parameters

Type	Name	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	<i>dimensions</i>	

Returns

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	

CalculateMass()

Calculates the mass of the Rigidbody and attached mass affectors.

Declaration

```
public float CalculateMass()
```

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

CalculateRelativeCenterOfMassOffset()

Calculates the center of mass of the Rigidbody and attached mass affectors.

Declaration

```
public Vector3 CalculateRelativeCenterOfMassOffset()
```

Returns

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	

GetMassAffectors()

Updates list of IMassAffectors attached to this object. Call after IMassAffector has been added or removed from the object.

Declaration

```
public IMassAffector[] GetMassAffectors()
```

Returns

Type	Description
IMassAffector (NWH.Common.CoM.IMassAffector.html)[]	

GetWorldCenterOfMass()

Gets the combined center of mass position in world space coordinates.

Declaration

```
public Vector3 GetWorldCenterOfMass()
```

Returns

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	World space position of the center of mass

MarkDirty()

Mark properties as needing recalculation. Call this when mass affectors change (fuel consumption, cargo loading, etc.).

Declaration

```
public void MarkDirty()
```

UpdateAllProperties()

Recalculates all Rigidbody properties (mass, center of mass, and inertia) based on current settings and affectors. Called automatically when isDirty flag is set.

Declaration

```
public void UpdateAllProperties()
```

UpdateCoM()

Calculates and applies the CoM to the Rigidbody.

Declaration

```
public void UpdateCoM()
```

UpdateInertia(bool)

Calculates and applies the inertia tensor to the Rigidbody.

Declaration

```
public void UpdateInertia(bool applyUnchanged = false)
```

Parameters

Type	Name	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	<i>applyUnchanged</i>	

UpdateMass()

Calculates and applies the total mass to the Rigidbody. Includes mass from affectors if useMassAffectors is enabled.

Declaration

```
public void UpdateMass()
```

See Also

[IMassAffector \(NWH.Common.CoM.IMassAffector.html\)](#)

[MassAffector \(NWH.Common.CoM.MassAffector.html\)](#)

[Vehicle \(NWH.Common.Vehicles.Vehicle.html\)](#)

Namespace NWH.Common.Demo

Classes

DemoCameraNameDisplay (NWH.Common.Demo.DemoCameraNameDisplay.html)

Displays the name of the currently active main camera in a Text component. Updates every 0.1 seconds.

DemoDtSetter (NWH.Common.Demo.DemoDtSetter.html)

Sets Time.fixedDeltaTime to a specific value for demo scenes. Default is 0.008333s (120Hz) for optimal physics performance.

DemoOscillator (NWH.Common.Demo.DemoOscillator.html)

Moves a Rigidbody in a sinusoidal oscillation pattern. Useful for creating moving platforms or obstacles in demo scenes.

DemoRotator (NWH.Common.Demo.DemoRotator.html)

Continuously rotates a Rigidbody at a constant rate. Useful for rotating platforms or visual elements in demo scenes.

DemoVehicleNameDisplay (NWH.Common.Demo.DemoVehicleNameDisplay.html)

Displays the name and type of the currently active vehicle in a Text component. Updates every 0.1 seconds.

DemoWelcomeMessage (NWH.Common.Demo.DemoWelcomeMessage.html)

Controls the display of a welcome message panel in demo scenes. Shows the message when running outside the editor.

DragObject (NWH.Common.Demo.DragObject.html)

Simple script that drags Rigidbody behind the mouse cursor when MMB is held down.

FpsToText (NWH.Common.Demo.FpsToText.html)

Displays the current framerate in a Text component with optional color coding. Supports both instantaneous and averaged FPS measurements.

RigidbodyFPSController (NWH.Common.Demo.RigidbodyFPSController.html)

Simple first-person controller using physics-based movement. Useful for testing and navigating demo scenes on foot.

Class DemoCameraNameDisplay

Displays the name of the currently active main camera in a Text component. Updates every 0.1 seconds.

Inheritance

- ↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)
- ↳ [Object](https://docs.unity3d.com/ScriptReference/Object.html) (<https://docs.unity3d.com/ScriptReference/Object.html>)
- ↳ [Component](https://docs.unity3d.com/ScriptReference/Component.html) (<https://docs.unity3d.com/ScriptReference/Component.html>)
- ↳ [Behaviour](https://docs.unity3d.com/ScriptReference/Behaviour.html) (<https://docs.unity3d.com/ScriptReference/Behaviour.html>)
- ↳ [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html) (<https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>)
- ↳ DemoCameraNameDisplay

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Demo \(NWH.Common.Demo.html\)](#).

Assembly: NWH.Common.dll

Syntax

```
[RequireComponent(typeof(Text))]  
public class DemoCameraNameDisplay : MonoBehaviour
```

Class DemoDtSetter

Sets Time.fixedDeltaTime to a specific value for demo scenes. Default is 0.008333s (120Hz) for optimal physics performance.

Inheritance

- ↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)
- ↳ [Object](https://docs.unity3d.com/ScriptReference/Object.html) (<https://docs.unity3d.com/ScriptReference/Object.html>)
 - ↳ [Component](https://docs.unity3d.com/ScriptReference/Component.html) (<https://docs.unity3d.com/ScriptReference/Component.html>)
 - ↳ [Behaviour](https://docs.unity3d.com/ScriptReference/Behaviour.html) (<https://docs.unity3d.com/ScriptReference/Behaviour.html>)
 - ↳ [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html) (<https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>)
 - ↳ DemoDtSetter

Namespace: [NWH \(NWH.html\)](#), [Common \(NWH.Common.html\)](#), [Demo \(NWH.Common.Demo.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
[DefaultExecutionOrder(-500)]  
public class DemoDtSetter : MonoBehaviour
```

Fields

fixedDeltaTime

Target physics update rate in seconds. Default 0.008333s equals 120Hz.

Declaration

```
public float fixedDeltaTime
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

Class DemoOscillator

Moves a Rigidbody in a sinusoidal oscillation pattern. Useful for creating moving platforms or obstacles in demo scenes.

Inheritance

- ↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)
- ↳ [Object](https://docs.unity3d.com/ScriptReference/Object.html) (<https://docs.unity3d.com/ScriptReference/Object.html>)
 - ↳ [Component](https://docs.unity3d.com/ScriptReference/Component.html) (<https://docs.unity3d.com/ScriptReference/Component.html>)
 - ↳ [Behaviour](https://docs.unity3d.com/ScriptReference/Behaviour.html) (<https://docs.unity3d.com/ScriptReference/Behaviour.html>)
 - ↳ [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html) (<https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>)
 - ↳ DemoOscillator

Namespace: [NWH \(NWH.html\)](#), [Common \(NWH.Common.html\)](#), [Demo \(NWH.Common.Demo.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public class DemoOscillator : MonoBehaviour
```

Fields

speed

Speed of the oscillation in Hz. Higher values result in faster movement.

Declaration

```
public float speed
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

travel

Maximum displacement from the starting position in each axis.

Declaration

```
public Vector3 travel
```

Field Value

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	

Class DemoRotator

Continuously rotates a Rigidbody at a constant rate. Useful for rotating platforms or visual elements in demo scenes.

Inheritance

- ↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)
- ↳ [Object](https://docs.unity3d.com/ScriptReference/Object.html) (<https://docs.unity3d.com/ScriptReference/Object.html>)
- ↳ [Component](https://docs.unity3d.com/ScriptReference/Component.html) (<https://docs.unity3d.com/ScriptReference/Component.html>)
- ↳ [Behaviour](https://docs.unity3d.com/ScriptReference/Behaviour.html) (<https://docs.unity3d.com/ScriptReference/Behaviour.html>)
- ↳ [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html) (<https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>)
- ↳ DemoRotator

Namespace: [NWH \(NWH.html\)](#), [Common \(NWH.Common.html\)](#), [Demo \(NWH.Common.Demo.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public class DemoRotator : MonoBehaviour
```

Fields

rotation

Rotation speed in degrees per second for each axis.

Declaration

```
public Vector3 rotation
```

Field Value

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	

Class DemoVehicleNameDisplay

Displays the name and type of the currently active vehicle in a Text component. Updates every 0.1 seconds.

Inheritance

- ↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)
 - ↳ [Object](https://docs.unity3d.com/ScriptReference/Object.html) (<https://docs.unity3d.com/ScriptReference/Object.html>)
 - ↳ [Component](https://docs.unity3d.com/ScriptReference/Component.html) (<https://docs.unity3d.com/ScriptReference/Component.html>)
 - ↳ [Behaviour](https://docs.unity3d.com/ScriptReference/Behaviour.html) (<https://docs.unity3d.com/ScriptReference/Behaviour.html>)
 - ↳ [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html) (<https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>)
 - ↳ DemoVehicleNameDisplay

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Demo \(NWH.Common.Demo.html\)](#).

Assembly: NWH.Common.dll

Syntax

```
[RequireComponent(typeof(Text))]  
public class DemoVehicleNameDisplay : MonoBehaviour
```

Class DemoWelcomeMessage

Controls the display of a welcome message panel in demo scenes. Shows the message when running outside the editor.

Inheritance

- ↳ [object](#)
- ↳ [Object](#)
- ↳ [Component](#)
- ↳ [Behaviour](#)
- ↳ [MonoBehaviour](#)
- ↳ DemoWelcomeMessage

Namespace: [NWH \(NWH.html\)](#), [Common \(NWH.Common.html\)](#), [Demo \(NWH.Common.Demo.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public class DemoWelcomeMessage : MonoBehaviour
```

Fields

closeButton

Button used to close the welcome message panel.

Declaration

```
public Button closeButton
```

Field Value

Type	Description
Button	

welcomeMessageGO

GameObject containing the welcome message UI.

Declaration

```
public GameObject welcomeMessageGO
```

Field Value

Type	Description
GameObject (https://docs.unity3d.com/ScriptReference/GameObject.html)	

Class DragObject

Simple script that drags Rigidbody behind the mouse cursor when MMB is held down.

Inheritance

- ↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)
- ↳ [Object](https://docs.unity3d.com/ScriptReference/Object.html) (<https://docs.unity3d.com/ScriptReference/Object.html>)
- ↳ [Component](https://docs.unity3d.com/ScriptReference/Component.html) (<https://docs.unity3d.com/ScriptReference/Component.html>)
- ↳ [Behaviour](https://docs.unity3d.com/ScriptReference/Behaviour.html) (<https://docs.unity3d.com/ScriptReference/Behaviour.html>)
- ↳ [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html) (<https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>)
- ↳ DragObject

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Demo \(NWH.Common.Demo.html\)](#).

Assembly: NWH.Common.dll

Syntax

```
public class DragObject : MonoBehaviour
```

Class FpsToText

Displays the current framerate in a Text component with optional color coding. Supports both instantaneous and averaged FPS measurements.

Inheritance

- ↳ [object](#)
- ↳ [Object](#)
- ↳ [Component](#)
- ↳ [Behaviour](#)
- ↳ [MonoBehaviour](#)
- ↳ FpsToText

Namespace: [NWH \(NWH.html\)](#), [Common \(NWH.Common.html\)](#), [Demo \(NWH.Common.Demo.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
[RequireComponent(typeof(Text))]  
public class FpsToText : MonoBehaviour
```

Fields

bad

Text color when framerate is below badBelow threshold.

Declaration

```
public Color bad
```

Field Value

Type	Description
Color	

badBelow

FPS threshold below which the color changes to bad (red).

Declaration

```
public int badBelow
```

Field Value

Type	Description
int (https://learn.microsoft.com/dotnet/api/system.int32)	

forceIntResult

Round FPS to nearest integer for cleaner display.

Declaration

```
public bool forceIntResult
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

good

Text color when framerate is above okayBelow threshold.

Declaration

```
public Color good
```

Field Value

Type	Description
Color (https://docs.unity3d.com/ScriptReference/Color.html)	

groupSampling

Use averaging over multiple samples instead of single frame measurement.

Declaration

```
public bool groupSampling
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

okay

Text color when framerate is between badBelow and okayBelow.

Declaration

```
public Color okay
```

Field Value

Type	Description
Color (https://docs.unity3d.com/ScriptReference/Color.html).	

okayBelow

FPS threshold below which the color changes to okay (yellow).

Declaration

```
public int okayBelow
```

Field Value

Type	Description
int (https://learn.microsoft.com/dotnet/api/system.int32).	

sampleSize

Number of samples to average when groupSampling is enabled.

Declaration

```
public int sampleSize
```

Field Value

Type	Description
int (https://learn.microsoft.com/dotnet/api/system.int32).	

smoothed

Use Time.smoothDeltaTime instead of Time.deltaTime for calculations.

Declaration

```
public bool smoothed
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

updateTextEvery

Update text display every N frames. 1 = every frame.

Declaration

```
public int updateTextEvery
```

Field Value

Type	Description
int (https://learn.microsoft.com/dotnet/api/system.int32)	

useColors

Enable color coding based on framerate thresholds.

Declaration

```
public bool useColors
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

useSystemTick

Use Environment.TickCount instead of Time.deltaTime for calculations.

Declaration

```
public bool useSystemTick
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

Methods

GetSystemFramerate()

Declaration

```
protected virtual int GetSystemFramerate()
```

Returns

Type	Description
int (https://learn.microsoft.com/dotnet/api/system.int32)	

Group()

Declaration

```
protected virtual void Group()
```

Reset()

Declaration

```
protected virtual void Reset()
```

SingleFrame()

Declaration

```
protected virtual void SingleFrame()
```

Start()

Declaration

```
protected virtual void Start()
```

Update()

Declaration

```
protected virtual void Update()
```


Class RigidbodyFPSController

Simple first-person controller using physics-based movement. Useful for testing and navigating demo scenes on foot.

Inheritance

- ↳ [object](#)
- ↳ [Object](#)
- ↳ [Component](#)
- ↳ [Behaviour](#)
- ↳ [MonoBehaviour](#)
- ↳ [RigidbodyFPSController](#)

Namespace: [NWH \(NWH.html\)](#), [Common \(NWH.Common.html\)](#), [Demo \(NWH.Common.Demo.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
[RequireComponent(typeof(Rigidbody))]  
[RequireComponent(typeof(CapsuleCollider))]  
public class RigidbodyFPSController : MonoBehaviour
```

Remarks

Based on Unity Community Wiki example. Uses Rigidbody for physics-accurate movement with mouse-look camera control.

Fields

gravity

Downward acceleration force in m/s².

Declaration

```
public float gravity
```

Field Value

Type	Description
float	

jumpHeight

Maximum height of jumps in meters.

Declaration

```
public float jumpHeight
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single)	

maxVelocityChange

Maximum velocity change per fixed update, controls acceleration responsiveness.

Declaration

```
public float maxVelocityChange
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single)	

maximumY

Maximum upward look angle in degrees.

Declaration

```
public float maximumY
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single)	

minimumY

Maximum downward look angle in degrees.

Declaration

```
public float minimumY
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

sensitivityX

Horizontal mouse look sensitivity.

Declaration

```
public float sensitivityX
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

sensitivityY

Vertical mouse look sensitivity.

Declaration

```
public float sensitivityY
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

speed

Movement speed in meters per second.

Declaration

```
public float speed
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

Namespace NWH.Common.Input

Classes

[InputManagerSceneInputProvider](#)

[\(NWH.Common.Input.InputManagerSceneInputProvider.html\)](#)

Scene input provider using Unity's legacy Input Manager system. Requires input axes and buttons to be configured in Project Settings > Input Manager.

[InputProvider \(NWH.Common.Input.InputProvider.html\)](#)

Base class from which all input providers inherit.

[InputSystemSceneInputProvider](#)

[\(NWH.Common.Input.InputSystemSceneInputProvider.html\)](#)

Unity Input System implementation of scene input provider. Handles camera controls and scene navigation using the new Input System.

[InputUtils \(NWH.Common.Input.InputUtils.html\)](#)

Utility methods for safe input retrieval with automatic fallback to default keys. Prevents errors when Input Manager bindings are missing.

[MobileInputButton \(NWH.Common.Input.MobileInputButton.html\)](#)

Extended Unity UI Button with state tracking for mobile input handling. Provides hasBeenClicked and isPressed flags for easier input polling.

[MobileSceneInputProvider \(NWH.Common.Input.MobileSceneInputProvider.html\)](#)

Scene input provider for mobile platforms using on-screen UI buttons. Requires MobileInputButton components assigned to changeCameraButton and changeVehicleButton fields.

[SceneInputActions \(NWH.Common.Input.SceneInputActions.html\)](#)

Provides programmatic access to UnityEngine.InputSystem.InputActionAsset, UnityEngine.InputSystem.InputActionMap, UnityEngine.InputSystem.InputAction and UnityEngine.InputSystem.InputControlScheme instances defined in asset "Packages/com.nwh.common/Runtime/Input/InputSystem/SceneInputActions.inputactions".

[SceneInputProviderBase \(NWH.Common.Input.SceneInputProviderBase.html\)](#)

InputProvider for scene and camera related behavior.

Structs

SceneInputActions.CameraControlsActions **(NWH.Common.Input.SceneInputActions.CameraControlsActions.html)**

Provides access to input actions defined in input action map "CameraControls".

SceneInputActions.SceneControlsActions **(NWH.Common.Input.SceneInputActions.SceneControlsActions.html)**

Provides access to input actions defined in input action map "SceneControls".

Interfaces

SceneInputActions.ICameraControlsActions **(NWH.Common.Input.SceneInputActions.ICameraControlsActions.html)**

Interface to implement callback methods for all input action callbacks associated with input actions defined by "CameraControls" which allows adding and removing callbacks.

SceneInputActions.ISceneControlsActions **(NWH.Common.Input.SceneInputActions.ISceneControlsActions.html)**

Interface to implement callback methods for all input action callbacks associated with input actions defined by "SceneControls" which allows adding and removing callbacks.

Class InputManagerSceneInputProvider

Scene input provider using Unity's legacy Input Manager system. Requires input axes and buttons to be configured in Project Settings > Input Manager.

Inheritance

```
↳ object
↳ Object
↳ Component
↳ Behaviour
↳ MonoBehaviour
↳ InputProvider (NWH.Common.Input.InputProvider.html)
↳ ScenelInputProviderBase (NWH.Common.Input.ScenelInputProviderBase.html)
↳ InputManagerSceneInputProvider
```

Inherited Members

[ScenelInputProviderBase.requireCameraPanningModifier](#)
([NWH.Common.Input.ScenelInputProviderBase.html](#)#[NWH Common Input ScenelInputProviderBase requireCamer aPanningModifier](#))

[ScenelInputProviderBase.requireCameraRotationModifier](#)
([NWH.Common.Input.ScenelInputProviderBase.html](#)#[NWH Common Input ScenelInputProviderBase requireCamer aRotationModifier](#))

[InputProvider.Instances](#) ([NWH.Common.Input.InputProvider.html](#)#[NWH Common Input InputProvider Instances](#))

[InputProvider.Awake\(\)](#) ([NWH.Common.Input.InputProvider.html](#)#[NWH Common Input InputProvider Awake](#))

[InputProvider.OnDestroy\(\)](#)
([NWH.Common.Input.InputProvider.html](#)#[NWH Common Input InputProvider OnDestroy](#))

[InputProvider.CombinedInput<T>\(Func<T, int>\)](#)
([NWH.Common.Input.InputProvider.html](#)#[NWH Common Input InputProvider CombinedInput 1 System Func 0 System Int32 \)](#)

[InputProvider.CombinedInput<T>\(Func<T, float>\)](#)
([NWH.Common.Input.InputProvider.html](#)#[NWH Common Input InputProvider CombinedInput 1 System Func 0 System Single \)](#)

[InputProvider.CombinedInput<T>\(Func<T, bool>\)](#)
([NWH.Common.Input.InputProvider.html](#)#[NWH Common Input InputProvider CombinedInput 1 System Func 0 System Boolean \)](#)

[InputProvider.CombinedInput<T>\(Func<T, Vector2>\)](#)
([NWH.Common.Input.InputProvider.html](#)#[NWH Common Input InputProvider CombinedInput 1 System Func 0 UnityEngine Vector2 \)](#)

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Input \(NWH.Common.Input.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public class InputManagerSceneInputProvider : ScenelInputProviderBase
```

Methods

CameraPanning()

Returns camera panning input as a Vector2 (x = horizontal, y = vertical).

Declaration

```
public override Vector2 CameraPanning()
```

Returns

Type	Description
Vector2 (https://docs.unity3d.com/ScriptReference/Vector2.html).	

Overrides

[ScenelInputProviderBase.CameraPanning\(\)](#).

([NWH.Common.Input.ScenelInputProviderBase.html#NWH_Common_Input_ScenelInputProviderBase_CameraPanning](#)).

CameraPanningModifier()

Returns true when the camera panning modifier button is held. If requireCameraPanningModifier is false, always returns true.

Declaration

```
public override bool CameraPanningModifier()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	

Overrides

[ScenelInputProviderBase.CameraPanningModifier\(\)](#).

([NWH.Common.Input.ScenelInputProviderBase.html#NWH_Common_Input_ScenelInputProviderBase_CameraPanningModifier](#)).

CameraRotation()

Returns camera rotation input as a Vector2 (x = horizontal, y = vertical).

Declaration

```
public override Vector2 CameraRotation()
```

Returns

Type	Description
Vector2 (https://docs.unity3d.com/ScriptReference/Vector2.html).	

Overrides

[ScenelInputProviderBase.CameraRotation\(\)](#)

(NWH.Common.Input.ScenelInputProviderBase.html#NWH Common Input ScenelInputProviderBase CameraRotation)

CameraRotationModifier()

Returns true when the camera rotation modifier button is held. If requireCameraRotationModifier is false, always returns true.

Declaration

```
public override bool CameraRotationModifier()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

Overrides

[ScenelInputProviderBase.CameraRotationModifier\(\)](#)

(NWH.Common.Input.ScenelInputProviderBase.html#NWH Common Input ScenelInputProviderBase CameraRotationModifier)

CameraZoom()

Returns camera zoom input value. Positive = zoom in, negative = zoom out.

Declaration

```
public override float CameraZoom()
```

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single)	

Overrides

[ScenelInputProviderBase.CameraZoom\(\)](#)

(NWH.Common.Input.ScenelInputProviderBase.html#NWH Common Input ScenelInputProviderBase CameraZoom)

ChangeCamera()

Returns true when the change camera button is pressed.

Declaration

```
public override bool ChangeCamera()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

Overrides

[ScenelInputProviderBase.ChangeCamera\(\)](#)
([NWH.Common.Input.ScenelInputProviderBase.html#NWH Common Input ScenelInputProviderBase ChangeCamera](#)).

ChangeVehicle()

Returns true when the change vehicle button is pressed.

Declaration

```
public override bool ChangeVehicle()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

Overrides

[ScenelInputProviderBase.ChangeVehicle\(\)](#)
([NWH.Common.Input.ScenelInputProviderBase.html#NWH Common Input ScenelInputProviderBase ChangeVehicle](#)).

CharacterMovement()

Returns character movement input as a Vector2 (x = horizontal, y = forward/back).

Declaration

```
public override Vector2 CharacterMovement()
```

Returns

Type	Description
Vector2 (https://docs.unity3d.com/ScriptReference/Vector2.html)	

Overrides

[ScenelInputProviderBase.CharacterMovement\(\)](#)
([NWH.Common.Input.ScenelInputProviderBase.html#NWH Common Input ScenelInputProviderBase CharacterMovement](#)).

ToggleGUI()

Returns true when the toggle GUI button is pressed.

Declaration

```
public override bool ToggleGUI()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

Overrides

[SceneInputProviderBase.ToggleGUI\(\)](#).

([NWH.Common.Input.SceneInputProviderBase.html#NWH_Common_Input_SceneInputProviderBase_ToggleGUI](#)).

Class InputProvider

Base class from which all input providers inherit.

Inheritance

```
↳ object (https://learn.microsoft.com/dotnet/api/system.object)
  ↳ Object (https://docs.unity3d.com/ScriptReference/Object.html)
  ↳ Component (https://docs.unity3d.com/ScriptReference/Component.html)
    ↳ Behaviour (https://docs.unity3d.com/ScriptReference/Behaviour.html)
      ↳ MonoBehaviour (https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)
        ↳ InputProvider
          ↳ SceneInputProviderBase (NWH.Common.Input.SceneInputProviderBase.html).
```

Namespace: [NWH](#) ([NWH.html](#)), [Common](#) ([NWH.Common.html](#)), [Input](#) ([NWH.Common.Input.html](#))

Assembly: NWH.Common.dll

Syntax

```
public abstract class InputProvider : MonoBehaviour
```

Fields

Instances

List of all InputProviders in the scene.

Declaration

```
public static List<InputProvider> Instances
```

Field Value

Type	Description
List (https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1).< InputProvider (NWH.Common.Input.InputProvider.html).>	

Methods

Awake()

Declaration

```
public virtual void Awake()
```

CombinedInput<T>(Func<T, bool>)

Returns combined input of all InputProviders present in the scene. Result will be positive if any InputProvider has the selected input set to true. T is a type of InputProvider that the input will be retrieved from.

Declaration

```
public static bool CombinedInput<T>(Func<T, bool> selector) where T : InputProvider
```

Parameters

Type	Name	Description
Func (https://learn.microsoft.com/dotnet/api/system.func-2).<T, bool (https://learn.microsoft.com/dotnet/api/system.boolean).>	selector	

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

Type Parameters

Name	Description
T	

CombinedInput<T>(Func<T, int>)

Returns combined input of all InputProviders present in the scene. Result will be a sum of all inputs of the selected type. T is a type of InputProvider that the input will be retrieved from.

Declaration

```
public static int CombinedInput<T>(Func<T, int> selector) where T : InputProvider
```

Parameters

Type	Name	Description
Func (https://learn.microsoft.com/dotnet/api/system.func-2).<T, int (https://learn.microsoft.com/dotnet/api/system.int32).>	selector	

Returns

Type	Description
int (https://learn.microsoft.com/dotnet/api/system.int32)	

Type Parameters

Name	Description
T	

CombinedInput<T>(Func<T, float>)

Returns combined input of all InputProviders present in the scene. Result will be a sum of all inputs of the selected type. T is a type of InputProvider that the input will be retrieved from.

Declaration

```
public static float CombinedInput<T>(Func<T, float> selector) where T : InputProvider
```

Parameters

Type	Name	Description
Func (https://learn.microsoft.com/dotnet/api/system.func-2).<T, float (https://learn.microsoft.com/dotnet/api/system.single).>	selector	

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

Type Parameters

Name	Description
T	

CombinedInput<T>(Func<T, Vector2>)

Returns combined input of all InputProviders present in the scene. Result will be a sum of all inputs of the selected type. T is a type of InputProvider that the input will be retrieved from.

Declaration

```
public static Vector2 CombinedInput<T>(Func<T, Vector2> selector) where T : InputProvider
```

Parameters

Type	Name	Description
Func (https://learn.microsoft.com/dotnet/api/system.func-2).<T, Vector2 (https://docs.unity3d.com/ScriptReference/Vector2.html).>	selector	

Returns

Type	Description
Vector2 (https://docs.unity3d.com/ScriptReference/Vector2.html).	

Type Parameters

Name	Description
T	

OnDestroy()

Declaration

```
public virtual void OnDestroy()
```

Class InputSystemSceneInputProvider

Unity Input System implementation of scene input provider. Handles camera controls and scene navigation using the new Input System.

Inheritance

```
↳ object (https://learn.microsoft.com/dotnet/api/system.object)
↳ Object (https://docs.unity3d.com/ScriptReference/Object.html)
↳ Component (https://docs.unity3d.com/ScriptReference/Component.html)
↳ Behaviour (https://docs.unity3d.com/ScriptReference/Behaviour.html)
↳ MonoBehaviour (https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)
↳ InputProvider (NWH.Common.Input.InputProvider.html)
↳ ScenelInputProviderBase (NWH.Common.Input.ScenelInputProviderBase.html)
↳ InputSystemSceneInputProvider
```

Inherited Members

[ScenelInputProviderBase.requireCameraPanningModifier](#)
([NWH.Common.Input.ScenelInputProviderBase.html](https://docs.unity3d.com/ScriptReference/ScenelInputProviderBase.html)#NWH Common Input ScenelInputProviderBase requireCamer aPanningModifier).

[ScenelInputProviderBase.requireCameraRotationModifier](#)
([NWH.Common.Input.ScenelInputProviderBase.html](https://docs.unity3d.com/ScriptReference/ScenelInputProviderBase.html)#NWH Common Input ScenelInputProviderBase requireCamer aRotationModifier).

[InputProvider.Instances](#) ([NWH.Common.Input.InputProvider.html](https://docs.unity3d.com/ScriptReference/InputProvider.html)#NWH Common Input InputProvider Instances).

[InputProvider.OnDestroy\(\)](#)
([NWH.Common.Input.InputProvider.html](https://docs.unity3d.com/ScriptReference/InputProvider.html)#NWH Common Input InputProvider OnDestroy).

[InputProvider.CombinedInput<T>\(Func<T, int>\)](#)
([NWH.Common.Input.InputProvider.html](https://docs.unity3d.com/ScriptReference/InputProvider.html)#NWH Common Input InputProvider CombinedInput_1 System Func_0 System Int32).

[InputProvider.CombinedInput<T>\(Func<T, float>\)](#)
([NWH.Common.Input.InputProvider.html](https://docs.unity3d.com/ScriptReference/InputProvider.html)#NWH Common Input InputProvider CombinedInput_1 System Func_0 System Single).

[InputProvider.CombinedInput<T>\(Func<T, bool>\)](#)
([NWH.Common.Input.InputProvider.html](https://docs.unity3d.com/ScriptReference/InputProvider.html)#NWH Common Input InputProvider CombinedInput_1 System Func_0 System Boolean).

[InputProvider.CombinedInput<T>\(Func<T, Vector2>\)](#)
([NWH.Common.Input.InputProvider.html](https://docs.unity3d.com/ScriptReference/InputProvider.html)#NWH Common Input InputProvider CombinedInput_1 System Func_0 UnityEngine Vector2).

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Input \(NWH.Common.Input.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public class InputSystemSceneInputProvider : SceneInputProviderBase
```

Fields

sceneInputActions

Declaration

```
public SceneInputActions sceneInputActions
```

Field Value

Type	Description
SceneInputActions (NWH.Common.Input.SceneInputActions.html) .	

Methods

Awake()

Declaration

```
public override void Awake()
```

Overrides

[InputProvider.Awake\(\) \(NWH.Common.Input.InputProvider.html#NWH_Common_Input_InputProvider_Awake\)](#)

CameraPanning()

Returns camera panning input as a Vector2 (x = horizontal, y = vertical).

Declaration

```
public override Vector2 CameraPanning()
```

Returns

Type	Description
Vector2 (https://docs.unity3d.com/ScriptReference/Vector2.html) .	

Overrides

[SceneInputProviderBase.CameraPanning\(\)](#).

[\(NWH.Common.Input.SceneInputProviderBase.html#NWH_Common_Input_SceneInputProviderBase_CameraPanning\)](#).

CameraPanningModifier()

Returns true when the camera panning modifier button is held. If requireCameraPanningModifier is false, always returns true.

Declaration

```
public override bool CameraPanningModifier()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

Overrides

[ScenelInputProviderBase.CameraPanningModifier\(\)](#)

([NWH.Common.Input.ScenelInputProviderBase.html#NWH_Common_Input_ScenelInputProviderBase_CameraPanningModifier](#))

CameraRotation()

Returns camera rotation input as a Vector2 (x = horizontal, y = vertical).

Declaration

```
public override Vector2 CameraRotation()
```

Returns

Type	Description
Vector2 (https://docs.unity3d.com/ScriptReference/Vector2.html)	

Overrides

[ScenelInputProviderBase.CameraRotation\(\)](#)

([NWH.Common.Input.ScenelInputProviderBase.html#NWH_Common_Input_ScenelInputProviderBase_CameraRotation](#))

CameraRotationModifier()

Returns true when the camera rotation modifier button is held. If requireCameraRotationModifier is false, always returns true.

Declaration

```
public override bool CameraRotationModifier()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

Overrides

[ScenelInputProviderBase.CameraRotationModifier\(\)](#)

([NWH.Common.Input.ScenelInputProviderBase.html#NWH_Common_Input_ScenelInputProviderBase_CameraRotation](#))

onModifier)

CameraZoom()

Returns camera zoom input value. Positive = zoom in, negative = zoom out.

Declaration

```
public override float CameraZoom()
```

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

Overrides

SceneInputProviderBase.CameraZoom()

([NWH.Common.Input.SceneInputProviderBase.html#NWH Common Input SceneInputProviderBase CameraZoom](https://learn.microsoft.com/dotnet/api/NWH.Common.Input.SceneInputProviderBase.html#NWH_Common_Input_SceneInputProviderBase_CameraZoom)).

ChangeCamera()

Returns true when the change camera button is pressed.

Declaration

```
public override bool ChangeCamera()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	

Overrides

SceneInputProviderBase.ChangeCamera()

([NWH.Common.Input.SceneInputProviderBase.html#NWH Common Input SceneInputProviderBase ChangeCamera](https://learn.microsoft.com/dotnet/api/NWH.Common.Input.SceneInputProviderBase.html#NWH_Common_Input_SceneInputProviderBase_ChangeCamera)).

ChangeVehicle()

Returns true when the change vehicle button is pressed.

Declaration

```
public override bool ChangeVehicle()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	

Overrides

[SceneInputProviderBase.ChangeVehicle\(\)](#)

([NWH.Common.Input.SceneInputProviderBase.html#NWH Common Input SceneInputProviderBase ChangeVehicle](#))

CharacterMovement()

Returns character movement input as a Vector2 (x = horizontal, y = forward/back).

Declaration

```
public override Vector2 CharacterMovement()
```

Returns

Type	Description
Vector2 (https://docs.unity3d.com/ScriptReference/Vector2.html)	

Overrides

[SceneInputProviderBase.CharacterMovement\(\)](#)

([NWH.Common.Input.SceneInputProviderBase.html#NWH Common Input SceneInputProviderBase CharacterMovement](#))

ToggleGUI()

Returns true when the toggle GUI button is pressed.

Declaration

```
public override bool ToggleGUI()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

Overrides

[SceneInputProviderBase.ToggleGUI\(\)](#)

([NWH.Common.Input.SceneInputProviderBase.html#NWH Common Input SceneInputProviderBase ToggleGUI](#))

Class InputUtils

Utility methods for safe input retrieval with automatic fallback to default keys. Prevents errors when Input Manager bindings are missing.

Inheritance

↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)
↳ InputUtils

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Input \(NWH.Common.Input.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public class InputUtils
```

Methods

TryGetAxis(string, bool)

Attempts to retrieve axis value from Input Manager, returns 0 if binding is missing.

Declaration

```
public static float TryGetAxis(string axisName, bool showWarning = true)
```

Parameters

Type	Name	Description
string (https://learn.microsoft.com/dotnet/api/system.string).	<i>axisName</i>	Input Manager axis name to query.
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	<i>showWarning</i>	Display warning message when axis is missing.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Axis value between -1 and 1, or 0 if binding is missing.

TryGetAxisRaw(string, bool)

Attempts to retrieve raw axis value from Input Manager, returns 0 if binding is missing. Raw axes return only -1, 0, or 1 without smoothing.

Declaration

```
public static float TryGetAxisRaw(string axisName, bool showWarning = true)
```

Parameters

Type	Name	Description
string (https://learn.microsoft.com/dotnet/api/system.string).	<i>axisName</i>	Input Manager axis name to query.
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	<i>showWarning</i>	Display warning message when axis is missing.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Raw axis value (-1, 0, or 1), or 0 if binding is missing.

TryGetButton(string, KeyCode, bool)

Attempts to retrieve button state from Input Manager, falls back to KeyCode if binding is missing.

Declaration

```
public static bool TryGetButton(string buttonName, KeyCode altKey, bool showWarning = true)
```

Parameters

Type	Name	Description
string (https://learn.microsoft.com/dotnet/api/system.string).	<i>buttonName</i>	Input Manager button name to query.
KeyCode	<i>altKey</i>	Fallback KeyCode to use if binding is missing.
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	<i>showWarning</i>	Display warning message when falling back to default key.

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	True if button is currently held down.

TryGetButtonDown(string, KeyCode, bool)

Attempts to retrieve button press from Input Manager, falls back to KeyCode if binding is missing.

Declaration

```
public static bool TryGetButtonDown(string buttonName, KeyCode altKey, bool showWarning = true)
```

Parameters

Type	Name	Description
<code>string</code> (https://learn.microsoft.com/dotnet/api/system.string)	<i>buttonName</i>	Input Manager button name to query.
<code>KeyCode</code>	<i>altKey</i>	Fallback KeyCode to use if binding is missing.
<code>bool</code> (https://learn.microsoft.com/dotnet/api/system.boolean)	<i>showWarning</i>	Display warning message when falling back to default key.

Returns

Type	Description
<code>bool</code> (https://learn.microsoft.com/dotnet/api/system.boolean)	True on the frame the button was pressed.

Class MobileInputButton

Extended Unity UI Button with state tracking for mobile input handling. Provides hasBeenClicked and isPressed flags for easier input polling.

Inheritance

- ↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)
- ↳ [Object](https://docs.unity3d.com/ScriptReference/Object.html) (<https://docs.unity3d.com/ScriptReference/Object.html>)
- ↳ [Component](https://docs.unity3d.com/ScriptReference/Component.html) (<https://docs.unity3d.com/ScriptReference/Component.html>)
- ↳ [Behaviour](https://docs.unity3d.com/ScriptReference/Behaviour.html) (<https://docs.unity3d.com/ScriptReference/Behaviour.html>)
- ↳ [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html) (<https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>)
 - ↳ [UIBehaviour](#)
 - ↳ [Selectable](#)
 - ↳ [Button](#)
 - ↳ [MobileInputButton](#)

Implements

- [IMoveHandler](#)
- [IPointerDownHandler](#)
- [IPointerUpHandler](#)
- [IPointerEnterHandler](#)
- [IPointerExitHandler](#)
- [ISelectHandler](#)
- [IDeselectHandler](#)
- [IPointerClickHandler](#)
- [ISubmitHandler](#)
- [IEventSystemHandler](#)

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Input \(NWH.Common.Input.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
[DefaultExecutionOrder(1000)]  
public class MobileInputButton : Button, IMoveHandler, IPointerDownHandler, IPointerUpHandle  
r, IPointerEnterHandler, IPointerExitHandler, ISelectHandler, IDeselectHandler, IPointerClic  
kHandler, ISubmitHandler, IEvetnSystemHandler
```

Fields

hasBeenClicked

True for one frame after the button is clicked. Automatically resets to false.

Declaration

```
public bool hasBeenClicked
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

isPressed

True while the button is being held down. Updates every frame.

Declaration

```
public bool isPressed
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

Methods

OnPointerDown(PointerEventData)

Declaration

```
public override void OnPointerDown(PointerEventData eventData)
```

Parameters

Type	Name	Description
PointerEventData	<i>eventData</i>	

Overrides

UnityEngine.UI.Selectable.OnPointerDown(UnityEngine.EventSystems.PointerEventData)

Implements

UnityEngine.EventSystems.IMoveHandler
UnityEngine.EventSystems.IPointerDownHandler
UnityEngine.EventSystems.IPointerUpHandler
UnityEngine.EventSystems.IPointerEnterHandler
UnityEngine.EventSystems.IPointerExitHandler
UnityEngine.EventSystems.ISelectHandler
UnityEngine.EventSystems.IDeselectHandler
UnityEngine.EventSystems.IPointerClickHandler
UnityEngine.EventSystems.ISubmitHandler
UnityEngine.EventSystems.IEventSystemHandler

Class MobileSceneInputProvider

Scene input provider for mobile platforms using on-screen UI buttons. Requires MobileInputButton components assigned to changeCameraButton and changeVehicleButton fields.

Inheritance

```
↳ object (https://learn.microsoft.com/dotnet/api/system.object)
↳ Object (https://docs.unity3d.com/ScriptReference/Object.html)
↳ Component (https://docs.unity3d.com/ScriptReference/Component.html)
↳ Behaviour (https://docs.unity3d.com/ScriptReference/Behaviour.html)
↳ MonoBehaviour (https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)
↳ InputProvider (NWH.Common.Input.InputProvider.html)
↳ ScenelInputProviderBase (NWH.Common.Input.ScenelInputProviderBase.html)
↳ MobileSceneInputProvider
```

Inherited Members

[ScenelInputProviderBase.requireCameraPanningModifier](#)
([NWH.Common.Input.ScenelInputProviderBase.html](https://docs.unity3d.com/ScriptReference/ScenelInputProviderBase.html)#NWH Common Input ScenelInputProviderBase requireCamer aPanningModifier).

[ScenelInputProviderBase.requireCameraRotationModifier](#)
([NWH.Common.Input.ScenelInputProviderBase.html](https://docs.unity3d.com/ScriptReference/ScenelInputProviderBase.html)#NWH Common Input ScenelInputProviderBase requireCamer aRotationModifier).

[InputProvider.Instances](#) ([NWH.Common.Input.InputProvider.html](https://docs.unity3d.com/ScriptReference/InputProvider.html)#NWH Common Input InputProvider Instances).

[InputProvider.Awake\(\)](#) ([NWH.Common.Input.InputProvider.html](https://docs.unity3d.com/ScriptReference/InputProvider.html)#NWH Common Input InputProvider Awake)

[InputProvider.OnDestroy\(\)](#)
([NWH.Common.Input.InputProvider.html](https://docs.unity3d.com/ScriptReference/InputProvider.html)#NWH Common Input InputProvider OnDestroy).

[InputProvider.CombinedInput<T>\(Func<T, int>\)](#)
([NWH.Common.Input.InputProvider.html](https://docs.unity3d.com/ScriptReference/InputProvider.html)#NWH Common Input InputProvider CombinedInput 1 System Func 0 System Int32).

[InputProvider.CombinedInput<T>\(Func<T, float>\)](#)
([NWH.Common.Input.InputProvider.html](https://docs.unity3d.com/ScriptReference/InputProvider.html)#NWH Common Input InputProvider CombinedInput 1 System Func 0 System Single).

[InputProvider.CombinedInput<T>\(Func<T, bool>\)](#)
([NWH.Common.Input.InputProvider.html](https://docs.unity3d.com/ScriptReference/InputProvider.html)#NWH Common Input InputProvider CombinedInput 1 System Func 0 System Boolean).

[InputProvider.CombinedInput<T>\(Func<T, Vector2>\)](#)
([NWH.Common.Input.InputProvider.html](https://docs.unity3d.com/ScriptReference/InputProvider.html)#NWH Common Input InputProvider CombinedInput 1 System Func 0 UnityEngine Vector2).

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Input \(NWH.Common.Input.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public class MobileSceneInputProvider : ScenelInputProviderBase
```

Fields

changeCameraButton

UI button for changing camera. Should reference a MobileInputButton in the scene.

Declaration

```
public MobileInputButton changeCameraButton
```

Field Value

Type	Description
MobileInputButton (NWH.Common.Input.MobileInputButton.html)	

changeVehicleButton

UI button for changing vehicle. Should reference a MobileInputButton in the scene.

Declaration

```
public MobileInputButton changeVehicleButton
```

Field Value

Type	Description
MobileInputButton (NWH.Common.Input.MobileInputButton.html)	

Methods

CameraPanning()

Returns camera panning input as a Vector2 (x = horizontal, y = vertical).

Declaration

```
public override Vector2 CameraPanning()
```

Returns

Type	Description
Vector2 (https://docs.unity3d.com/ScriptReference/Vector2.html)	

Overrides

[SceneInputProviderBase.CameraPanning\(\)](#)

([NWH.Common.Input.SceneInputProviderBase.html#NWH_Common_Input_SceneInputProviderBase_CameraPanning](#))

198

g).

CameraPanningModifier()

Returns true when the camera panning modifier button is held. If requireCameraPanningModifier is false, always returns true.

Declaration

```
public override bool CameraPanningModifier()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	

Overrides

[ScenelInputProviderBase.CameraPanningModifier\(\)](#).

([NWH.Common.Input.ScenelInputProviderBase.html#NWH_Common_Input_ScenelInputProviderBase_CameraPanningModifier](#)).

CameraRotation()

Returns camera rotation input as a Vector2 (x = horizontal, y = vertical).

Declaration

```
public override Vector2 CameraRotation()
```

Returns

Type	Description
Vector2 (https://docs.unity3d.com/ScriptReference/Vector2.html).	

Overrides

[ScenelInputProviderBase.CameraRotation\(\)](#).

([NWH.Common.Input.ScenelInputProviderBase.html#NWH_Common_Input_ScenelInputProviderBase_CameraRotation](#)).

CameraRotationModifier()

Returns true when the camera rotation modifier button is held. If requireCameraRotationModifier is false, always returns true.

Declaration

```
public override bool CameraRotationModifier()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	

Overrides

[ScenelInputProviderBase.CameraRotationModifier\(\)](#)
([NWH.Common.Input.ScenelInputProviderBase.html#NWH Common Input ScenelInputProviderBase CameraRotationModifier](#)).

CameraZoom()

Returns camera zoom input value. Positive = zoom in, negative = zoom out.

Declaration

```
public override float CameraZoom()
```

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

Overrides

[ScenelInputProviderBase.CameraZoom\(\)](#)
([NWH.Common.Input.ScenelInputProviderBase.html#NWH Common Input ScenelInputProviderBase CameraZoom](#)).

ChangeCamera()

Returns true when the change camera button is pressed.

Declaration

```
public override bool ChangeCamera()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	

Overrides

[ScenelInputProviderBase.ChangeCamera\(\)](#)
([NWH.Common.Input.ScenelInputProviderBase.html#NWH Common Input ScenelInputProviderBase ChangeCamera](#)).

ChangeVehicle()

Returns true when the change vehicle button is pressed.

Declaration

```
public override bool ChangeVehicle()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

Overrides

[ScenelInputProviderBase.ChangeVehicle\(\)](#)
([NWH.Common.Input.ScenelInputProviderBase.html#NWH Common Input ScenelInputProviderBase ChangeVehicle](https://NWH.Common.Input.ScenelInputProviderBase.html#NWH_Common_Input_ScenelInputProviderBase_ChangeVehicle)).
)

CharacterMovement()

Returns character movement input as a Vector2 (x = horizontal, y = forward/back).

Declaration

```
public override Vector2 CharacterMovement()
```

Returns

Type	Description
Vector2 (https://docs.unity3d.com/ScriptReference/Vector2.html)	

Overrides

[ScenelInputProviderBase.CharacterMovement\(\)](#)
([NWH.Common.Input.ScenelInputProviderBase.html#NWH Common Input ScenelInputProviderBase CharacterMovement](https://NWH.Common.Input.ScenelInputProviderBase.html#NWH_Common_Input_ScenelInputProviderBase_CharacterMovement)).
)

ToggleGUI()

Returns true when the toggle GUI button is pressed.

Declaration

```
public override bool ToggleGUI()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

Overrides

[ScenelInputProviderBase.ToggleGUI\(\)](#)
([NWH.Common.Input.ScenelInputProviderBase.html#NWH Common Input ScenelInputProviderBase ToggleGUI](https://NWH.Common.Input.ScenelInputProviderBase.html#NWH_Common_Input_ScenelInputProviderBase_ToggleGUI)).
)

Class SceneInputActions

Provides programmatic access to UnityEngine.InputSystem.InputActionAsset, UnityEngine.InputSystem.InputActionMap, UnityEngine.InputSystem.InputAction and UnityEngine.InputSystem.InputControlScheme instances defined in asset "Packages/com.nwh.common/Runtime/Input/InputSystem/SceneInputActions.inputactions".

Inheritance

↳ [object](#) (<https://learn.microsoft.com/dotnet/api/system.object>).
↳ SceneInputActions

Implements

IInputActionCollection2
IInputActionCollection
IEnumerable (<https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1>).<InputAction>
IEnumerable (<https://learn.microsoft.com/dotnet/api/system.collections.ienumerable>).
IDisposable (<https://learn.microsoft.com/dotnet/api/system.idisposable>).

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Input \(NWH.Common.Input.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public class SceneInputActions : IInputActionCollection2, IInputActionCollection, IEnumerable<InputAction>, IEnumerable, IDisposable
```

Remarks

This class is source generated and any manual edits will be discarded if the associated asset is reimported or modified.

Examples

```
using namespace UnityEngine;
using UnityEngine.InputSystem;

// Example of using an InputActionMap named "Player" from a UnityEngine.MonoBehaviour implementing callback interface.
public class Example : MonoBehaviour, MyActions.IPlayerActions
{
    private MyActions_Actions m_Actions; // Source code representation of a asset.
    private MyActions_Actions.PlayerActions m_Player; // Source code representation of a action map.

    void Awake()
    {
        m_Actions = new MyActions_Actions(); // Create asset object.
        m_Player = m_Actions.Player; // Extract action map object.
        m_Player.AddCallbacks(this); // Register callback interface IPlayerActions.
    }

    void OnDestroy()
    {
        m_Actions.Dispose(); // Destroy asset object.
    }

    void OnEnable()
    {
        m_Player.Enable(); // Enable all actions within map.
    }

    void OnDisable()
    {
        m_Player.Disable(); // Disable all actions within map.
    }

#region Interface implementation of MyActions.IPlayerActions

    // Invoked when "Move" action is either started, performed or canceled.
    public void OnMove(InputAction.CallbackContext context)
    {
        Debug.Log($"OnMove: {context.ReadValue<Vector2>()}");
    }

    // Invoked when "Attack" action is either started, performed or canceled.
    public void OnAttack(InputAction.CallbackContext context)
    {
        Debug.Log($"OnAttack: {context.ReadValue<float>()}");
    }

#endregion
}
```

Constructors

SceneInputActions()

Constructs a new instance.

Declaration

```
public SceneInputActions()
```

Properties

CameraControls

Provides a new [SceneInputActions.CameraControlsActions](#)

([NWH.Common.Input.SceneInputActions.CameraControlsActions.html](#)) instance referencing this action map.

Declaration

```
public SceneInputActions.CameraControlsActions CameraControls { get; }
```

Property Value

Type	Description
SceneInputActions (NWH.Common.Input.SceneInputActions.html) . CameraControlsActions (NWH.Common.Input.SceneInputActions.CameraControlsActions.html) .	

SceneControls

Provides a new [SceneInputActions.SceneControlsActions](#)

([NWH.Common.Input.SceneInputActions.SceneControlsActions.html](#)) instance referencing this action map.

Declaration

```
public SceneInputActions.SceneControlsActions SceneControls { get; }
```

Property Value

Type	Description
SceneInputActions (NWH.Common.Input.SceneInputActions.html) . SceneControlsActions (NWH.Common.Input.SceneInputActions.SceneControlsActions.html) .	

asset

Provides access to the underlying asset instance.

Declaration

```
public InputActionAsset asset { get; }
```

Property Value

Type	Description
InputActionAsset	

bindingMask

Declaration

```
public InputBinding? bindingMask { get; set; }
```

Property Value

Type	Description
InputBinding?	

bindings

Declaration

```
public IEnumerable<InputBinding> bindings { get; }
```

Property Value

Type	Description
IEnumerable (https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1).<InputBinding>	

controlSchemes

Declaration

```
public ReadOnlyArray<InputControlScheme> controlSchemes { get; }
```

Property Value

Type	Description
ReadOnlyArray<InputControlScheme>	

devices

Declaration

```
public ReadOnlyArray<InputDevice>? devices { get; set; }
```

Property Value

Type	Description
ReadOnlyArray<InputDevice>?	

Methods

Contains(InputAction)

Declaration

```
public bool Contains(InputAction action)
```

Parameters

Type	Name	Description
InputAction	<i>action</i>	

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	

Disable()

Declaration

```
public void Disable()
```

Dispose()

Destroys this asset and all associated UnityEngine.InputSystem.InputAction instances.

Declaration

```
public void Dispose()
```

Enable()

Declaration

```
public void Enable()
```

~SceneInputActions()

Declaration

```
protected ~SceneInputActions()
```

FindAction(string, bool)

Declaration

```
public InputAction FindAction(string actionNameOrId, bool throwIfNotFound = false)
```

Parameters

Type	Name	Description
string (https://learn.microsoft.com/dotnet/api/system.string).	<i>actionNameOrId</i>	
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	<i>throwIfNotFound</i>	

Returns

Type	Description
InputAction	

FindBinding(InputBinding, out InputAction)

Declaration

```
public int FindBinding(InputBinding bindingMask, out InputAction action)
```

Parameters

Type	Name	Description
InputBinding	<i>bindingMask</i>	
InputAction	<i>action</i>	

Returns

Type	Description
int (https://learn.microsoft.com/dotnet/api/system.int32)	

GetEnumerator()

Returns an enumerator that iterates through the collection.

Declaration

```
public IEnumarator<InputAction> GetEnumerator()
```

Returns

Type	Description
IEnumarator (https://learn.microsoft.com/dotnet/api/system.collections.generic.enumerator-1). <InputAction>	An enumerator that can be used to iterate through the collection.

Implements

[UnityEngine.InputSystem.IInputActionCollection2](#)

[UnityEngine.InputSystem.IInputActionCollection](#)

[IEnumarable<T> \(<https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1>\).](#)

[IEnumarable \(<https://learn.microsoft.com/dotnet/api/system.collections.ienumerable>\).](#)

[IDisposable \(<https://learn.microsoft.com/dotnet/api/system.idisposable>\).](#)

Struct SceneInputActions.CameraControlsActions

Provides access to input actions defined in input action map "CameraControls".

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Input \(NWH.Common.Input.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public struct SceneInputActions.CameraControlsActions
```

Constructors

CameraControlsActions(SceneInputActions)

Construct a new instance of the input action map wrapper class.

Declaration

```
public CameraControlsActions(SceneInputActions wrapper)
```

Parameters

Type	Name	Description
SceneInputActions (NWH.Common.Input.SceneInputActions.html) .	<i>wrapper</i>	

Properties

CameraPanning

Provides access to the underlying input action "CameraControls/CameraPanning".

Declaration

```
public InputAction CameraPanning { get; }
```

Property Value

Type	Description
InputAction	

CameraPanningModifier

Provides access to the underlying input action "CameraControls/CameraPanningModifier".

Declaration

```
public InputAction CameraPanningModifier { get; }
```

Property Value

Type	Description
InputAction	

CameraRotation

Provides access to the underlying input action "CameraControls/CameraRotation".

Declaration

```
public InputAction CameraRotation { get; }
```

Property Value

Type	Description
InputAction	

CameraRotationModifier

Provides access to the underlying input action "CameraControls/CameraRotationModifier".

Declaration

```
public InputAction CameraRotationModifier { get; }
```

Property Value

Type	Description
InputAction	

CameraZoom

Provides access to the underlying input action "CameraControls/CameraZoom".

Declaration

```
public InputAction CameraZoom { get; }
```

Property Value

Type	Description
InputAction	

ChangeCamera

Provides access to the underlying input action "CameraControls/ChangeCamera".

Declaration

```
public InputAction ChangeCamera { get; }
```

Property Value

Type	Description
InputAction	

enabled

Declaration

```
public bool enabled { get; }
```

Property Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

Methods

AddCallbacks(ICameraControlsActions)

Adds UnityEngine.InputSystem.InputAction.started, UnityEngine.InputSystem.InputAction.performed and UnityEngine.InputSystem.InputAction.canceled callbacks provided via on all input actions contained in this map.

Declaration

```
public void AddCallbacks(SceneInputActions.ICameraControlsActions instance)
```

Parameters

Type	Name	Description
SceneInputActions (NWH.Common.Input.SceneInputActions.html) . ICameraControlsActions (NWH.Common.Input.SceneInputActions.ICameraControlsActions.html).	<i>instance</i>	Callback instance.

Remarks

If `instance` is `null` or `instance` have already been added this method does nothing.

See Also

[SceneInputActions \(NWH.Common.Input.SceneInputActions.html\)](#).
[CameraControlsActions](#)
(NWH.Common.Input.SceneInputActions.CameraControlsActions.html).

Disable()

Declaration

```
public void Disable()
```

Enable()

Declaration

```
public void Enable()
```

Get()

Provides access to the underlying input action map instance.

Declaration

```
public InputActionMap Get()
```

Returns

Type	Description
InputActionMap	

RemoveCallbacks(ICameraControlsActions)

Unregisters and unregisters all input action callbacks via [UnregisterCallbacks\(ICameraControlsActions\)](#)
(NWH.Common.Input.SceneInputActions.ICameraControlsActions.html).

Declaration

```
public void RemoveCallbacks(SceneInputActions.ICameraControlsActions instance)
```

Parameters

Type	Name	Description
SceneInputActions (NWH.Common.Input.SceneInputActions.html) . ICameraControlsActions (NWH.Common.Input.SceneInputActions.ICameraControlsActions.html).	<i>instance</i>	

See Also

[UnregisterCallbacks \(NWH.Common.Input.SceneInputActions.ICameraControlsActions.html\)\(ICameraControlsActions \(NWH.Common.Input.SceneInputActions.ICameraControlsActions.html\)\)](#)

SetCallbacks(ICameraControlsActions)

Replaces all existing callback instances and previously registered input action callbacks associated with them with callbacks provided via .

Declaration

```
public void SetCallbacks(SceneInputActions.ICameraControlsActions instance)
```

Parameters

Type	Name	Description
SceneInputActions (NWH.Common.Input.SceneInputActions.html) . ICameraControlsActions (NWH.Common.Input.SceneInputActions.ICameraControlsActions.html).	<i>instance</i>	

Remarks

If `instance` is `null`, calling this method will only unregister all existing callbacks but not register any new callbacks.

See Also

[AddCallbacks\(ICameraControlsActions\)](#)

([NWH.Common.Input.SceneInputActions.CameraControlsActions.html#NWH Common Input SceneInputActions CameraControlsActions AddCallbacks NWH Common Input SceneInputActions ICameraControlsActions](#)).

[RemoveCallbacks\(ICameraControlsActions\)](#)

([NWH.Common.Input.SceneInputActions.CameraControlsActions.html#NWH Common Input SceneInputActions CameraControlsActions RemoveCallbacks NWH Common Input SceneInputActions ICameraControlsActions](#)).

[UnregisterCallbacks \(NWH.Common.Input.SceneInputActions.ICameraControlsActions.html\)\(ICameraControlsActions \(NWH.Common.Input.SceneInputActions.ICameraControlsActions.html\)\)](#)

Operators

implicit operator InputActionMap(CameraControlsActions)

Implicitly converts an to an instance.

Declaration

```
public static implicit operator InputActionMap(SceneInputActions.CameraControlsActions set)
```

Parameters

Type	Name	Description
SceneInputActions (NWH.Common.Input.SceneInputActions.html) . CameraControlsActions (NWH.Common.Input.SceneInputActions.CameraControlsActions.html)	set	

Returns

Type	Description
InputActionMap	

Interface SceneInputActions.ICameraControlsActions

Interface to implement callback methods for all input action callbacks associated with input actions defined by "CameraControls" which allows adding and removing callbacks.

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Input \(NWH.Common.Input.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public interface SceneInputActions.ICameraControlsActions
```

Methods

OnCameraPanning(CallbackContext)

Method invoked when associated input action "CameraPanning" is either UnityEngine.InputSystem.InputAction.started, UnityEngine.InputSystem.InputAction.performed or UnityEngine.InputSystem.InputAction.canceled.

Declaration

```
void OnCameraPanning(InputAction.CallbackContext context)
```

Parameters

Type	Name	Description
InputAction.CallbackContext	<i>context</i>	

See Also

started
performed
canceled

OnCameraPanningModifier(CallbackContext)

Method invoked when associated input action "CameraPanningModifier" is either UnityEngine.InputAction.started, UnityEngine.InputAction.performed or UnityEngine.InputAction.canceled.

Declaration

```
void OnCameraPanningModifier(InputAction.CallbackContext context)
```

Parameters

Type	Name	Description
InputAction.CallbackContext	<i>context</i>	

See Also

started
performed
canceled

OnCameraRotation(CallbackContext)

Method invoked when associated input action "CameraRotation" is either UnityEngine.InputSystem.InputAction.started, UnityEngine.InputSystem.InputAction.performed or UnityEngine.InputSystem.InputAction.canceled.

Declaration

```
void OnCameraRotation(InputAction.CallbackContext context)
```

Parameters

Type	Name	Description
InputAction.CallbackContext	<i>context</i>	

See Also

started
performed
canceled

OnCameraRotationModifier(CallbackContext)

Method invoked when associated input action "CameraRotationModifier" is either UnityEngine.InputSystem.InputAction.started, UnityEngine.InputSystem.InputAction.performed or UnityEngine.InputSystem.InputAction.canceled.

Declaration

```
void OnCameraRotationModifier(InputAction.CallbackContext context)
```

Parameters

Type	Name	Description
InputAction.CallbackContext	<i>context</i>	

See Also

started
performed
canceled

OnCameraZoom(CallbackContext)

Method invoked when associated input action "CameraZoom" is either UnityEngine.InputSystem.InputAction.started, UnityEngine.InputSystem.InputAction.performed or UnityEngine.InputSystem.InputAction.canceled.

Declaration

```
void OnCameraZoom(InputAction.CallbackContext context)
```

Parameters

Type	Name	Description
InputAction.CallbackContext	context	

See Also

started
performed
canceled

OnChangeCamera(CallbackContext)

Method invoked when associated input action "ChangeCamera" is either UnityEngine.InputSystem.InputAction.started, UnityEngine.InputSystem.InputAction.performed or UnityEngine.InputSystem.InputAction.canceled.

Declaration

```
void OnChangeCamera(InputAction.CallbackContext context)
```

Parameters

Type	Name	Description
InputAction.CallbackContext	context	

See Also

started
performed
canceled

See Also

AddCallbacks

(NWH.Common.Input.SceneInputActions.CameraControlsActions.html#NWH_Common_Input_SceneInputActions_CameraControlsActions_AddCallbacks_NWH_Common_Input_SceneInputActions_ICameraControlsActions_)(ICameraControlsActions (NWH.Common.Input.SceneInputActions.ICameraControlsActions.html))

RemoveCallbacks

(NWH.Common.Input.SceneInputActions.CameraControlsActions.html#NWH_Common_Input_SceneInputActions_CameraControlsActions_RemoveCallbacks_NWH_Common_Input_SceneInputActions_ICameraControlsActions_)(ICameraControlsActions (NWH.Common.Input.SceneInputActions.ICameraControlsActions.html))

Interface SceneInputActions.ISceneControlsActions

Interface to implement callback methods for all input action callbacks associated with input actions defined by "SceneControls" which allows adding and removing callbacks.

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Input \(NWH.Common.Input.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public interface SceneInputActions.ISceneControlsActions
```

Methods

OnChangeVehicle(CallbackContext)

Method invoked when associated input action "ChangeVehicle" is either UnityEngine.InputSystem.InputAction.started, UnityEngine.InputSystem.InputAction.performed or UnityEngine.InputSystem.InputAction.canceled.

Declaration

```
void OnChangeVehicle(InputAction.CallbackContext context)
```

Parameters

Type	Name	Description
InputAction.CallbackContext	<i>context</i>	

See Also

started
performed
canceled

OnDragObjectModifier(CallbackContext)

Method invoked when associated input action "DragObjectModifier" is either UnityEngine.InputAction.started, UnityEngine.InputAction.performed or UnityEngine.InputAction.canceled.

Declaration

```
void OnDragObjectModifier(InputAction.CallbackContext context)
```

Parameters

Type	Name	Description
InputAction.CallbackContext	<i>context</i>	

See Also

started
performed
canceled

OnFPSMovement(CallbackContext)

Method invoked when associated input action "FPSMovement" is either UnityEngine.InputSystem.InputAction.started, UnityEngine.InputSystem.InputAction.performed or UnityEngine.InputSystem.InputAction.canceled.

Declaration

```
void OnFPSMovement(InputAction.CallbackContext context)
```

Parameters

Type	Name	Description
InputAction.CallbackContext	<i>context</i>	

See Also

started
performed
canceled

OnShowCursor(CallbackContext)

Method invoked when associated input action "ShowCursor" is either UnityEngine.InputSystem.InputAction.started, UnityEngine.InputSystem.InputAction.performed or UnityEngine.InputSystem.InputAction.canceled.

Declaration

```
void OnShowCursor(InputAction.CallbackContext context)
```

Parameters

Type	Name	Description
InputAction.CallbackContext	<i>context</i>	

See Also

started
performed
canceled

OnToggleGUI(CallbackContext)

Method invoked when associated input action "ToggleGUI" is either UnityEngine.InputSystem.InputAction.started, UnityEngine.InputSystem.InputAction.performed or UnityEngine.InputSystem.InputAction.canceled.

Declaration

```
void OnToggleGUI(InputAction.CallbackContext context)
```

Parameters

Type	Name	Description
InputAction.CallbackContext	<i>context</i>	

See Also

started

performed

canceled

See Also

[AddCallbacks](#)

([NWH.Common.Input.SceneInputActions.SceneControlsActions.html#NWH Common Input SceneInputActions SceneControlsActions AddCallbacks NWH Common Input SceneInputActions ISceneControlsActions](#))([ISceneControlsActions \(NWH.Common.Input.SceneInputActions.ISceneControlsActions.html\)](#))

[RemoveCallbacks](#)

([NWH.Common.Input.SceneInputActions.SceneControlsActions.html#NWH Common Input SceneInputActions SceneControlsActions RemoveCallbacks NWH Common Input SceneInputActions ISceneControlsActions](#))([ISceneControlsActions \(NWH.Common.Input.SceneInputActions.ISceneControlsActions.html\)](#))

Struct SceneInputActions.SceneControlsActions

Provides access to input actions defined in input action map "SceneControls".

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Input \(NWH.Common.Input.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public struct SceneInputActions.SceneControlsActions
```

Constructors

SceneControlsActions(SceneInputActions)

Construct a new instance of the input action map wrapper class.

Declaration

```
public SceneControlsActions(SceneInputActions wrapper)
```

Parameters

Type	Name	Description
SceneInputActions (NWH.Common.Input.SceneInputActions.html) .	<i>wrapper</i>	

Properties

ChangeVehicle

Provides access to the underlying input action "SceneControls/ChangeVehicle".

Declaration

```
public InputAction ChangeVehicle { get; }
```

Property Value

Type	Description
InputAction	

DragObjectModifier

Provides access to the underlying input action "SceneControls/DragObjectModifier".

Declaration

```
public InputAction DragObjectModifier { get; }
```

Property Value

Type	Description
InputAction	

FPSMovement

Provides access to the underlying input action "SceneControls/FPSMovement".

Declaration

```
public InputAction FPSMovement { get; }
```

Property Value

Type	Description
InputAction	

ShowCursor

Provides access to the underlying input action "SceneControls>ShowCursor".

Declaration

```
public InputAction ShowCursor { get; }
```

Property Value

Type	Description
InputAction	

ToggleGUI

Provides access to the underlying input action "SceneControls/ToggleGUI".

Declaration

```
public InputAction ToggleGUI { get; }
```

Property Value

Type	Description
InputAction	

enabled

Declaration

```
public bool enabled { get; }
```

Property Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

Methods

AddCallbacks(ISceneControlsActions)

Adds UnityEngine.InputSystem.InputAction.started, UnityEngine.InputSystem.InputAction.performed and UnityEngine.InputSystem.InputAction.canceled callbacks provided via on all input actions contained in this map.

Declaration

```
public void AddCallbacks(SceneInputActions.ISceneControlsActions instance)
```

Parameters

Type	Name	Description
SceneInputActions (NWH.Common.Input.SceneInputActions.html) . ISceneControlsActions (NWH.Common.Input.SceneInputActions.ISceneControlsActions.html).	<i>instance</i>	Callback instance.

Remarks

If `instance` is `null` or `instance` have already been added this method does nothing.

See Also

[SceneInputActions \(NWH.Common.Input.SceneInputActions.html\)](#).
[SceneControlsActions](#)
([NWH.Common.Input.SceneInputActions.SceneControlsActions.html](#)).

Disable()

Declaration

```
public void Disable()
```

Enable()

Declaration

```
public void Enable()
```

Get()

Provides access to the underlying input action map instance.

Declaration

```
public InputActionMap Get()
```

Returns

Type	Description
InputActionMap	

RemoveCallbacks(ISceneControlsActions)

Unregisters and unregisters all input action callbacks via [UnregisterCallbacks\(ISceneControlsActions\)](#).
[\(NWH.Common.Input.SceneInputActions.ISceneControlsActions.html\)](#).

Declaration

```
public void RemoveCallbacks(SceneInputActions.ISceneControlsActions instance)
```

Parameters

Type	Name	Description
SceneInputActions (NWH.Common.Input.SceneInputActions.html). ISceneControlsActions (NWH.Common.Input.SceneInputActions.ISceneControlsActions.html).	<i>instance</i>	

See Also

[UnregisterCallbacks](#) ([NWH.Common.Input.SceneInputActions.ISceneControlsActions.html](#)).
[\(NWH.Common.Input.SceneInputActions.ISceneControlsActions.html\)](#)

SetCallbacks(ISceneControlsActions)

Replaces all existing callback instances and previously registered input action callbacks associated with them with callbacks provided via .

Declaration

```
public void SetCallbacks(SceneInputActions.ISceneControlsActions instance)
```

Parameters

Type	Name	Description
SceneInputActions (NWH.Common.Input.SceneInputActions.html) . ISceneControlsActions (NWH.Common.Input.SceneInputActions.ISceneControlsActions.html).	<i>instance</i>	

Remarks

If *instance* is null , calling this method will only unregister all existing callbacks but not register any new callbacks.

See Also

[AddCallbacks\(ISceneControlsActions\)](#)

([NWH.Common.Input.SceneInputActions.SceneControlsActions.html#NWH_Common_Input_SceneInputActions_SceneControlsActions_AddCallbacks_NWH_Common_Input_SceneInputActions_ISceneControlsActions_](#))

[RemoveCallbacks\(ISceneControlsActions\)](#)

([NWH.Common.Input.SceneInputActions.SceneControlsActions.html#NWH_Common_Input_SceneInputActions_SceneControlsActions_RemoveCallbacks_NWH_Common_Input_SceneInputActions_ISceneControlsActions_](#))

[UnregisterCallbacks \(NWH.Common.Input.SceneInputActions.ISceneControlsActions.html\)\(ISceneControlsActions](#)

([NWH.Common.Input.SceneInputActions.ISceneControlsActions.html](#))

Operators

implicit operator InputActionMap(SceneControlsActions)

Implicitly converts an to an instance.

Declaration

```
public static implicit operator InputActionMap(SceneInputActions.SceneControlsActions set)
```

Parameters

Type	Name	Description
SceneInputActions (NWH.Common.Input.SceneInputActions.html) . SceneControlsActions (NWH.Common.Input.SceneInputActions.SceneControlsActions.html) .	<i>set</i>	

Returns

Type	Description
InputActionMap	

Class SceneInputProviderBase

InputProvider for scene and camera related behavior.

Inheritance

```
↳ object (https://learn.microsoft.com/dotnet/api/system.object)
  ↳ Object (https://docs.unity3d.com/ScriptReference/Object.html)
    ↳ Component (https://docs.unity3d.com/ScriptReference/Component.html)
      ↳ Behaviour (https://docs.unity3d.com/ScriptReference/Behaviour.html)
        ↳ MonoBehaviour (https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)
          ↳ InputProvider (NWH.Common.Input.InputProvider.html)
            ↳ SceneInputProviderBase
              ↳ InputManagerSceneInputProvider (NWH.Common.Input.InputManagerSceneInputProvider.html)
              ↳ InputSystemSceneInputProvider (NWH.Common.Input.InputSystemSceneInputProvider.html)
              ↳ MobileSceneInputProvider (NWH.Common.Input.MobileSceneInputProvider.html)
```

Inherited Members

[InputProvider.Instances](https://NWH.Common.Input.InputProvider.html#NWH_Common_Input_InputProvider_Instances) (NWH.Common.Input.InputProvider.html#[NWH Common Input InputProvider Instances](https://NWH.Common.Input.InputProvider.html#NWH_Common_Input_InputProvider_Instances)).

[InputProvider.Awake\(\)](https://NWH.Common.Input.InputProvider.html#NWH_Common_Input_InputProvider_Awake) (NWH.Common.Input.InputProvider.html#[NWH Common Input InputProvider Awake](https://NWH.Common.Input.InputProvider.html#NWH_Common_Input_InputProvider_Awake)).

[InputProvider.OnDestroy\(\)](https://NWH.Common.Input.InputProvider.html#NWH_Common_Input_InputProvider_OnDestroy).

(NWH.Common.Input.InputProvider.html#[NWH Common Input InputProvider OnDestroy](https://NWH.Common.Input.InputProvider.html#NWH_Common_Input_InputProvider_OnDestroy)).

[InputProvider.CombinedInput<T>](https://NWH.Common.Input.InputProvider.html#NWH_Common_Input_InputProvider_CombinedInput<T>) ([Func<T, int>](https://NWH.Common.Input.InputProvider.html)).

(NWH.Common.Input.InputProvider.html#[NWH Common Input InputProvider CombinedInput 1 System Func 0 System Int32](https://NWH.Common.Input.InputProvider.html#NWH_Common_Input_InputProvider_CombinedInput_1_System_Func_0_System_Int32)).

[InputProvider.CombinedInput<T>](https://NWH.Common.Input.InputProvider.html#NWH_Common_Input_InputProvider_CombinedInput<T>) ([Func<T, float>](https://NWH.Common.Input.InputProvider.html)).

(NWH.Common.Input.InputProvider.html#[NWH Common Input InputProvider CombinedInput 1 System Func 0 System Single](https://NWH.Common.Input.InputProvider.html#NWH_Common_Input_InputProvider_CombinedInput_1_System_Func_0_System_Single)).

[InputProvider.CombinedInput<T>](https://NWH.Common.Input.InputProvider.html#NWH_Common_Input_InputProvider_CombinedInput<T>) ([Func<T, bool>](https://NWH.Common.Input.InputProvider.html)).

(NWH.Common.Input.InputProvider.html#[NWH Common Input InputProvider CombinedInput 1 System Func 0 System Boolean](https://NWH.Common.Input.InputProvider.html#NWH_Common_Input_InputProvider_CombinedInput_1_System_Func_0_System_Boolean)).

[InputProvider.CombinedInput<T>](https://NWH.Common.Input.InputProvider.html#NWH_Common_Input_InputProvider_CombinedInput<T>) ([Func<T, Vector2>](https://NWH.Common.Input.InputProvider.html)).

(NWH.Common.Input.InputProvider.html#[NWH Common Input InputProvider CombinedInput 1 System Func 0 UnityEngine Vector2](https://NWH.Common.Input.InputProvider.html#NWH_Common_Input_InputProvider_CombinedInput_1_System_Func_0_UnityEngine_Vector2)).

Namespace: [NWH \(NWH.html\)](https://NWH.html).[Common \(NWH.Common.html\)](https://NWH.Common.html).[Input \(NWH.Common.Input.html\)](https://NWH.Common.Input.html)

Assembly: NWH.Common.dll

Syntax

```
public abstract class SceneInputProviderBase : InputProvider
```

Fields

requireCameraPanningModifier

If true a button press will be required to unlock camera panning.

Declaration

```
[Tooltip("    If true a button press will be required to unlock camera panning.")]
public bool requireCameraPanningModifier
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

requireCameraRotationModifier

If true a button press will be required to unlock camera rotation.

Declaration

```
[Tooltip("    If true a button press will be required to unlock camera rotation."")]
public bool requireCameraRotationModifier
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

Methods

CameraPanning()

Returns camera panning input as a Vector2 (x = horizontal, y = vertical).

Declaration

```
public virtual Vector2 CameraPanning()
```

Returns

Type	Description
Vector2 (https://docs.unity3d.com/ScriptReference/Vector2.html)	

CameraPanningModifier()

Returns true when the camera panning modifier button is held. If requireCameraPanningModifier is false, always returns true.

Declaration

```
public virtual bool CameraPanningModifier()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

CameraRotation()

Returns camera rotation input as a Vector2 (x = horizontal, y = vertical).

Declaration

```
public virtual Vector2 CameraRotation()
```

Returns

Type	Description
Vector2 (https://docs.unity3d.com/ScriptReference/Vector2.html)	

CameraRotationModifier()

Returns true when the camera rotation modifier button is held. If requireCameraRotationModifier is false, always returns true.

Declaration

```
public virtual bool CameraRotationModifier()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

CameraZoom()

Returns camera zoom input value. Positive = zoom in, negative = zoom out.

Declaration

```
public virtual float CameraZoom()
```

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

ChangeCamera()

Returns true when the change camera button is pressed.

Declaration

```
public virtual bool ChangeCamera()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	

ChangeVehicle()

Returns true when the change vehicle button is pressed.

Declaration

```
public virtual bool ChangeVehicle()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	

CharacterMovement()

Returns character movement input as a Vector2 (x = horizontal, y = forward/back).

Declaration

```
public virtual Vector2 CharacterMovement()
```

Returns

Type	Description
Vector2 (https://docs.unity3d.com/ScriptReference/Vector2.html).	

ToggleGUI()

Returns true when the toggle GUI button is pressed.

Declaration

```
public virtual bool ToggleGUI()
```

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

Namespace NWH.Common.Scene Management

Classes

[VehicleChanger \(NWH.Common.SceneManagement.VehicleChanger.html\)](#)

Manages switching between multiple vehicles in a scene with support for both instant and character-based (enter/exit) modes.

Enums

[VehicleChanger.CharacterLocation](#)

[\(NWH.Common.SceneManagement.VehicleChanger.CharacterLocation.html\)](#)

Represents the player's spatial relationship to vehicles.

Class VehicleChanger

Manages switching between multiple vehicles in a scene with support for both instant and character-based (enter/exit) modes.

Inheritance

- ↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)
- ↳ [Object](https://docs.unity3d.com/ScriptReference/Object.html) (<https://docs.unity3d.com/ScriptReference/Object.html>)
- ↳ [Component](https://docs.unity3d.com/ScriptReference/Component.html) (<https://docs.unity3d.com/ScriptReference/Component.html>)
- ↳ [Behaviour](https://docs.unity3d.com/ScriptReference/Behaviour.html) (<https://docs.unity3d.com/ScriptReference/Behaviour.html>)
- ↳ [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html) (<https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>)
- ↳ VehicleChanger

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[SceneManagement \(NWH.Common.SceneManagement.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
[DefaultExecutionOrder(500)]  
public class VehicleChanger : MonoBehaviour
```

Remarks

VehicleChanger supports two modes:
- Instant switching: Press a button to cycle through vehicles immediately
- Character-based: Player must walk to a vehicle and enter/exit at designated points

In character-based mode, the player can only enter vehicles when near an EnterExitPoint and the vehicle is moving slowly enough. This creates more realistic vehicle switching similar to GTA-style games.

Inactive vehicles can optionally be put to sleep to improve performance when managing many vehicles in a scene.

Fields

activeVehicleIndex

Index of the current vehicle in vehicles list.

Declaration

```
[Tooltip("    Index of the current vehicle in vehicles list.")]  
public int activeVehicleIndex
```

Field Value

Type	Description
int (https://learn.microsoft.com/dotnet/api/system.int32)	

characterBased

Is vehicle changing character based? When true changing vehicles will require getting close to them to be able to enter, opposed to pressing a button to switch between vehicles.

Declaration

```
[Tooltip("Is vehicle changing character based? When true changing vehicles will require getting close to them\r\n\tto be able to enter, opposed to pressing a button to switch between vehicles.")]  
public bool characterBased
```

Field Value

Type	Description
<code>bool (https://learn.microsoft.com/dotnet/api/system.boolean).</code>	

characterObject

Game object representing a character. Can also be another vehicle.

Declaration

```
[Tooltip("      Game object representing a character. Can also be another vehicle.")]  
public GameObject characterObject
```

Field Value

Type	Description
GameObject (https://docs.unity3d.com/ScriptReference/GameObject.html).	

enterDistance

Maximum distance at which the character will be able to enter the vehicle.

Declaration

```
[Range(0.2, 3)]  
[Tooltip("    Maximum distance at which the character will be able to enter the vehicle.")]  
public float enterDistance
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

enterExitTag

Tag of the object representing the point from which the enter distance will be measured. Useful if you want to enable your character to enter only when near the door.

Declaration

```
[Tooltip("Tag of the object representing the point from which the enter distance will be measured. Useful if you want to enable your character to enter only when near the door.")]
public string enterExitTag
```

Field Value

Type	Description
string (https://learn.microsoft.com/dotnet/api/system.string).	

location

When the location is Near, the player can enter the vehicle.

Declaration

```
[Tooltip("When the location is Near, the player can enter the vehicle.")]
public VehicleChanger.CharacterLocation location
```

Field Value

Type	Description
VehicleChanger (NWH.Common.SceneManagement.VehicleChanger.html). CharacterLocation (NWH.Common.SceneManagement.VehicleChanger.CharacterLocation.html).	

maxEnterExitVehicleSpeed

Maximum speed at which the character will be able to enter / exit the vehicle.

Declaration

```
[Tooltip("    Maximum speed at which the character will be able to enter / exit the vehicle")]
public float maxEnterExitVehicleSpeed
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

onDeactivateAll

Event invoked when all vehicles are deactivated (e.g., when exiting in character-based mode).

Declaration

```
public UnityEvent onDeactivateAll
```

Field Value

Type	Description
UnityEvent (https://docs.unity3d.com/ScriptReference/Events.UnityEvent.html).	

onVehicleChanged

Event invoked whenever the active vehicle changes.

Declaration

```
public UnityEvent onVehicleChanged
```

Field Value

Type	Description
UnityEvent (https://docs.unity3d.com/ScriptReference/Events.UnityEvent.html).	

putOtherVehiclesToSleep

Should the vehicles that the player is currently not using be put to sleep to improve performance?

Declaration

```
[Tooltip("    Should the vehicles that the player is currently not using be put to sleep to
improve performance?")]
public bool putOtherVehiclesToSleep
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	

startInVehicle

Should the player start inside the vehicle?

Declaration

```
[Tooltip("Should the player start inside the vehicle?")]
public bool startInVehicle
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

vehicles

List of all of the vehicles that can be selected and driven in the scene.

Declaration

```
[Tooltip("List of all of the vehicles that can be selected and driven in the scene.")]
public List<Vehicle> vehicles
```

Field Value

Type	Description
List (https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1).< Vehicle (NWH.Common.Vehicles.Vehicle.html)>	

Properties

Instance

Declaration

```
public static VehicleChanger Instance { get; }
```

Property Value

Type	Description
VehicleChanger (NWH.Common.SceneManagement.VehicleChanger.html)	

Methods

ChangeVehicle(Vehicle)

Switches to the specified vehicle if it exists in the vehicles list.

Declaration

```
public void ChangeVehicle(Vehicle ac)
```

Parameters

Type	Name	Description
Vehicle (<i>NWH.Common.Vehicles.Vehicle.html</i>)	ac	Vehicle reference to switch to.

ChangeVehicle(int)

Changes vehicle to requested vehicle.

Declaration

```
public void ChangeVehicle(int index)
```

Parameters

Type	Name	Description
int (<i>https://learn.microsoft.com/dotnet/api/system.int32</i>)	index	Index of a vehicle in Vehicles list.

DeactivateAllExceptActive()

Enables the current active vehicle and optionally disables all others based on putOtherVehiclesToSleep setting.

Declaration

```
public void DeactivateAllExceptActive()
```

DeactivateAllIncludingActive()

Disables all managed vehicles including the currently active one. Used when exiting vehicles in character-based mode.

Declaration

```
public void DeactivateAllIncludingActive()
```

DeregisterVehicle(Vehicle)

Removes a vehicle from the managed vehicles list. If the vehicle was active, automatically switches to the next vehicle.

Declaration

```
public void DeregisterVehicle(Vehicle v)
```

Parameters

Type	Name	Description
Vehicle (NWH.Common.Vehicles.Vehicle.html)	v	Vehicle to deregister.

EnterVehicle(Vehicle)

Puts the player inside the specified vehicle and activates it. In character-based mode, stores the entry position for later exit.

Declaration

```
public void EnterVehicle(Vehicle v)
```

Parameters

Type	Name	Description
Vehicle (NWH.Common.Vehicles.Vehicle.html)	v	Vehicle to enter.

ExitVehicle(Vehicle)

Removes the player from the vehicle and spawns the character object nearby. Character is positioned at the stored entry location.

Declaration

```
public void ExitVehicle(Vehicle v)
```

Parameters

Type	Name	Description
Vehicle (NWH.Common.Vehicles.Vehicle.html)	v	Vehicle to exit.

NextVehicle()

Switches to the next vehicle in the list, wrapping to the first vehicle when reaching the end.

Declaration

```
public void NextVehicle()
```

PreviousVehicle()

Switches to the previous vehicle in the list, wrapping to the last vehicle when at the beginning.

Declaration

```
public void PreviousVehicle()
```

RegisterVehicle(Vehicle)

Adds a vehicle to the managed vehicles list if not already present. Newly registered vehicles are automatically disabled unless they are the active vehicle.

Declaration

```
public void RegisterVehicle(Vehicle v)
```

Parameters

Type	Name	Description
Vehicle (NWH.Common.Vehicles.Vehicle.html)	v	Vehicle to register.

Enum VehicleChanger.CharacterLocation

Represents the player's spatial relationship to vehicles.

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[SceneManagement \(NWH.Common.SceneManagement.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public enum VehicleChanger.CharacterLocation
```

Fields

Name	Description
Inside	Player is currently inside a vehicle.
Near	Player is close enough to enter a vehicle.
OutOfRange	Player is too far from any vehicle to interact.

Namespace NWH.Common.ShiftingOrigin

Classes

[ShiftingOrigin \(NWH.Common.ShiftingOrigin.ShiftingOrigin.html\)](#)

Prevents floating point precision errors by shifting all scene objects back toward world origin when the main camera exceeds the distance threshold.

Class ShiftingOrigin

Prevents floating point precision errors by shifting all scene objects back toward world origin when the main camera exceeds the distance threshold.

Inheritance

- ↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)
- ↳ [Object](https://docs.unity3d.com/ScriptReference/Object.html) (<https://docs.unity3d.com/ScriptReference/Object.html>)
 - ↳ [Component](https://docs.unity3d.com/ScriptReference/Component.html) (<https://docs.unity3d.com/ScriptReference/Component.html>)
 - ↳ [Behaviour](https://docs.unity3d.com/ScriptReference/Behaviour.html) (<https://docs.unity3d.com/ScriptReference/Behaviour.html>)
 - ↳ [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html) (<https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>)
 - ↳ ShiftingOrigin

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[ShiftingOrigin \(NWH.Common.ShiftingOrigin.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public class ShiftingOrigin : MonoBehaviour
```

Remarks

As objects move far from world origin [0,0,0], floating point precision degrades causing physics jitter and rendering artifacts. ShiftingOrigin solves this by periodically moving all scene content back toward origin, keeping the player near [0,0,0] at all times.

The shift is transparent to gameplay - relative positions remain identical. Useful for open world games, flight simulators, or any scenario with large travel distances.

Only affects the current scene. For multi-scene setups, ensure one ShiftingOrigin instance per loaded scene set.

Fields

Instance

Declaration

```
public static ShiftingOrigin Instance
```

Field Value

Type	Description
ShiftingOrigin (NWH.Common.ShiftingOrigin.ShiftingOrigin.html) .	

distanceThreshold

Distance from world origin in meters that triggers an origin shift. Default 500m works well for most scenarios.

Declaration

```
public float distanceThreshold
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single)	

onAfterJump

Event invoked after the origin shift completes and physics is re-synced.

Declaration

```
public UnityEvent onAfterJump
```

Field Value

Type	Description
UnityEvent (https://docs.unity3d.com/ScriptReference/Events.UnityEvent.html)	

onBeforeJump

Event invoked before the origin shift begins. Rigidbody sleep thresholds are temporarily disabled.

Declaration

```
public UnityEvent onBeforeJump
```

Field Value

Type	Description
UnityEvent (https://docs.unity3d.com/ScriptReference/Events.UnityEvent.html)	

Properties

TotalOffset

Cumulative offset applied to all objects since scene start. Useful for tracking absolute world position despite origin shifts.

Declaration

```
public Vector3 TotalOffset { get; }
```

Property Value

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html)	

Methods

RefreshCaches()

Refreshes cached references to Rigidbodies and ParticleSystems. Call this if new physics objects or particle systems are added to the scene at runtime.

Declaration

```
public void RefreshCaches()
```

Namespace NWH.Common.Utility

Classes

AnimationCurveExtensions (NWH.Common.Utility.AnimationCurveExtensions.html)

Extension methods for AnimationCurve manipulation and processing.

ArrayExtensions (NWH.Common.Utility.ArrayExtensions.html)

Extension methods for array manipulation.

GameObjectExtensions (NWH.Common.Utility.GameObjectExtensions.html)

Extension methods for GameObject and Transform operations.

GeomUtility (NWH.Common.Utility.GeoUtility.html)

Collection of geometric utility functions for 3D math operations. Includes vector math, mesh calculations, triangle operations, and spatial queries.

PIDController (NWH.Common.Utility.PIDController.html)

Proportional-Integral-Derivative controller for smooth value regulation. Used for automated control systems like cruise control, stability systems, and steering assistance.

QuaternionExtensions (NWH.Common.Utility.QuaternionExtensions.html)

Extension methods for advanced Quaternion operations. Provides interpolation methods with control over rotation direction.

UnitConverter (NWH.Common.Utility.UnitConverter.html)

Static utility class for converting between various units of measurement. Includes conversions for distance, speed, fuel efficiency, and angular velocity.

Class AnimationCurveExtensions

Extension methods for AnimationCurve manipulation and processing.

Inheritance

↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)

↳ AnimationCurveExtensions

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Utility \(NWH.Common.Utility.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public static class AnimationCurveExtensions
```

Methods

GenerateCurveArray(AnimationCurve, int)

Samples an AnimationCurve at regular intervals and returns the values as an array. Useful for pre-calculating curve values for performance-critical code.

Declaration

```
public static float[] GenerateCurveArray(this AnimationCurve self, int resolution = 256)
```

Parameters

Type	Name	Description
AnimationCurve (https://docs.unity3d.com/ScriptReference/AnimationCurve.html).	<i>self</i>	The curve to sample.
int (https://learn.microsoft.com/dotnet/api/system.int32).	<i>resolution</i>	Number of samples to take. Higher values provide more precision.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).[]	Array of sampled values from 0 to 1.

MakeSmooth(AnimationCurve)

Smooths out a scripting-generated AnimationCurve by calculating appropriate tangents. Creates smooth transitions between keyframes.

Declaration

```
public static AnimationCurve MakeSmooth(this AnimationCurve inCurve)
```

Parameters

Type	Name	Description
AnimationCurve (https://docs.unity3d.com/ScriptReference/AnimationCurve.html).	<i>inCurve</i>	The curve to smooth.

Returns

Type	Description
AnimationCurve (https://docs.unity3d.com/ScriptReference/AnimationCurve.html).	A new smoothed AnimationCurve.

Class ArrayExtensions

Extension methods for array manipulation.

Inheritance

↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)

↳ ArrayExtensions

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Utility \(NWH.Common.Utility.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public static class ArrayExtensions
```

Methods

Fill<T>(T[], params T[])

Efficiently fills an array by repeating a pattern of values. Uses doubling strategy for performance.

Declaration

```
public static void Fill<T>(this T[] destinationArray, params T[] value)
```

Parameters

Type	Name	Description
T[]	<i>destinationArray</i>	Array to fill.
T[]	<i>value</i>	Pattern of values to repeat throughout the array.

Type Parameters

Name	Description
<i>T</i>	Type of array elements.

Class GameObjectExtensions

Extension methods for GameObject and Transform operations.

Inheritance

↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)

↳ GameObjectExtensions

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Utility \(NWH.Common.Utility.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public static class GameObjectExtensions
```

Methods

FindBoundsIncludeChildren(GameObject)

Calculates the combined bounds of all MeshRenderers in a GameObject and its children.

Declaration

```
public static Bounds FindBoundsIncludeChildren(this GameObject gameObject)
```

Parameters

Type	Name	Description
GameObject (https://docs.unity3d.com/ScriptReference/GameObject.html).	<i>gameObject</i>	GameObject to calculate bounds for.

Returns

Type	Description
Bounds (https://docs.unity3d.com/ScriptReference/Bounds.html).	Combined bounds encapsulating all child renderers.

GetComponentInParent<T>(Transform, bool)

Searches for a component in parent GameObjects, with option to include inactive objects. More flexible than Unity's built-in GetComponentInParent.

Declaration

```
public static T GetComponentInParent<T>(this Transform transform, bool includeInactive = true) where T : Component
```

Parameters

Type	Name	Description
<u>Transform</u> (https://docs.unity3d.com/ScriptReference/Transform.html).	<i>transform</i>	Starting transform.
<u>bool</u> (https://learn.microsoft.com/dotnet/api/system.boolean).	<i>includeInactive</i>	Include inactive GameObjects in search.

Returns

Type	Description
T	First component of type T found in parents, or null if none found.

Type Parameters

Name	Description
T	Type of component to find.

GetComponentInParentsOrChildren<T>(Transform, bool)

Searches for a component in parents first, then children if not found. Combines functionality of GetComponentInParent and GetComponentInChildren.

Declaration

```
public static T GetComponentInParentsOrChildren<T>(this Transform transform, bool includeInactive = true) where T : Component
```

Parameters

Type	Name	Description
<u>Transform</u> (https://docs.unity3d.com/ScriptReference/Transform.html).	<i>transform</i>	Starting transform.
<u>bool</u> (https://learn.microsoft.com/dotnet/api/system.boolean).	<i>includeInactive</i>	Include inactive GameObjects in search.

Returns

Type	Description
T	First component of type T found, or null if none found.

Type Parameters

Name	Description
T	Type of component to find.

Class GeomUtility

Collection of geometric utility functions for 3D math operations. Includes vector math, mesh calculations, triangle operations, and spatial queries.

Inheritance

↳ [object](#)
↳ GeomUtility

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Utility \(NWH.Common.Utility.html\)](#).

Assembly: NWH.Common.dll

Syntax

```
public static class GeomUtility
```

Methods

AreaFromFourPoints(Vector3, Vector3, Vector3, Vector3)

Calculates the area of a quadrilateral from four points.

Declaration

```
public static float AreaFromFourPoints(Vector3 p1, Vector3 p2, Vector3 p3, Vector3 p4)
```

Parameters

Type	Name	Description
Vector3	<i>p1</i>	First point.
Vector3	<i>p2</i>	Second point.
Vector3	<i>p3</i>	Third point.
Vector3	<i>p4</i>	Fourth point.

Returns

Type	Description
float	Area of the quadrilateral.

AreaFromThreePoints(Vector3, Vector3, Vector3)

Calculates the area of a triangle from three points.

Declaration

```
public static float AreaFromThreePoints(Vector3 p1, Vector3 p2, Vector3 p3)
```

Parameters

Type	Name	Description
Vector3 .	<i>p1</i>	First point.
Vector3 .	<i>p2</i>	Second point.
Vector3 .	<i>p3</i>	Third point.

Returns

Type	Description
float .	Area of the triangle.

ChangeLayersRecursively(Transform, string)

Changes the layer of a transform and all its children recursively.

Declaration

```
public static void ChangeLayersRecursively(this Transform trans, string name)
```

Parameters

Type	Name	Description
Transform .	<i>trans</i>	Root transform.
string .	<i>name</i>	Layer name.

ChangeObjectAlpha(GameObject, float)

Changes the alpha value of a GameObject's material color.

Declaration

```
public static void ChangeObjectAlpha(GameObject gameObject, float alpha)
```

Parameters

Type	Name	Description
GameObject .	<i>gameObject</i>	GameObject to modify.
float .	<i>alpha</i>	New alpha value (0-1).

ChangeObjectColor(GameObject, Color)

Changes the color of a GameObject's material.

Declaration

```
public static void ChangeObjectColor(GameObject gameobject, Color color)
```

Parameters

Type	Name	Description
GameObject	<i>gameObject</i>	GameObject to modify.
Color	<i>color</i>	New color.

ClampMagnitude(Vector3, float, float)

Clamps the magnitude of a vector between minimum and maximum values.

Declaration

```
public static Vector3 ClampMagnitude(this Vector3 v, float min, float max)
```

Parameters

Type	Name	Description
Vector3	<i>v</i>	Vector to clamp.
float	<i>min</i>	Minimum magnitude.
float	<i>max</i>	Maximum magnitude.

Returns

Type	Description
Vector3	Vector with clamped magnitude.

CopySign(float, float)

Copies the sign from one float to the magnitude of another.

Declaration

```
public static float CopySign(float mag, float sgn)
```

Parameters

Type	Name	Description
float	<i>mag</i>	Magnitude value.
float	<i>sgn</i>	Sign donor value.

Returns

Type	Description
float	Magnitude with the sign of <i>sgn</i> .

DistanceAlongNormal(Vector3, Vector3, Vector3)

Calculates the distance between two points projected along a normal vector.

Declaration

```
public static float DistanceAlongNormal(Vector3 a, Vector3 b, Vector3 normal)
```

Parameters

Type	Name	Description
Vector3	<i>a</i>	First point.
Vector3	<i>b</i>	Second point.
Vector3	<i>normal</i>	Normal vector to project along.

Returns

Type	Description
float	Distance along normal.

Equal(Quaternion, Quaternion)

Checks if two Quaternion values are approximately equal.

Declaration

```
public static bool Equal(this Quaternion a, Quaternion b)
```

Parameters

Type	Name	Description
Quaternion	<i>a</i>	First quaternion.
Quaternion	<i>b</i>	Second quaternion.

Returns

Type	Description
<code>bool</code> (https://learn.microsoft.com/dotnet/api/system.boolean)	True if angle between quaternions is less than 0.1 degrees.

FindArea(Vector3, Vector3, Vector3, Vector3)

Calculates the area of a quadrilateral defined by four points.

Declaration

```
public static float FindArea(Vector3 A, Vector3 B, Vector3 C, Vector3 D)
```

Parameters

Type	Name	Description
<code>Vector3</code> (https://docs.unity3d.com/ScriptReference/Vector3.html).	A	First corner.
<code>Vector3</code> (https://docs.unity3d.com/ScriptReference/Vector3.html).	B	Second corner.
<code>Vector3</code> (https://docs.unity3d.com/ScriptReference/Vector3.html).	C	Third corner.
<code>Vector3</code> (https://docs.unity3d.com/ScriptReference/Vector3.html).	D	Fourth corner.

Returns

Type	Description
<code>float</code> (https://learn.microsoft.com/dotnet/api/system.single).	Area of the quad.

FindCenter(Vector3, Vector3, Vector3, Vector3)

Finds the center point of a quad or triangle defined by 3 or 4 points.

Declaration

```
public static Vector3 FindCenter(Vector3 a, Vector3 b, Vector3 c, Vector3 d)
```

Parameters

Type	Name	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	<i>a</i>	First corner.
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	<i>b</i>	Second corner.
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	<i>c</i>	Third corner.
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	<i>d</i>	Fourth corner (can equal first corner for triangle).

Returns

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	Center point.

FindChordLine(Vector3, Vector3, Vector3, Vector3, float)

Finds a point along the chord line of a quad at the specified percentage.

Declaration

```
public static Vector3 FindChordLine(Vector3 a, Vector3 b, Vector3 c, Vector3 d, float chordPercent)
```

Parameters

Type	Name	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	<i>a</i>	First corner.
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	<i>b</i>	Second corner.
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	<i>c</i>	Third corner.
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	<i>d</i>	Fourth corner.
float (https://learn.microsoft.com/dotnet/api/system.single).	<i>chordPercent</i>	Position along chord (0-1).

Returns

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	Point on chord line.

FindDistanceToSegment(Vector3, Vector3, Vector3)

Calculates the distance between a point and a line segment.

Declaration

```
public static float FindDistanceToSegment(Vector3 pt, Vector3 p1, Vector3 p2)
```

Parameters

Type	Name	Description
Vector3	<i>pt</i>	Point to measure from.
Vector3	<i>p1</i>	First endpoint of segment.
Vector3	<i>p2</i>	Second endpoint of segment.

Returns

Type	Description
float	Distance to the segment.

FindMeshCenter(Mesh)

Find mesh center by averaging. Returns local center.

Declaration

```
public static Vector3 FindMeshCenter(Mesh mesh)
```

Parameters

Type	Name	Description
Mesh	<i>mesh</i>	

Returns

Type	Description
Vector3	

FindSpanLine(Vector3, Vector3, Vector3, Vector3, float)

Finds a point along the span line of a quad at the specified percentage.

Declaration

```
public static Vector3 FindSpanLine(Vector3 a, Vector3 b, Vector3 c, Vector3 d, float spanPercent)
```

Parameters

Type	Name	Description
Vector3	<i>a</i>	First corner.
Vector3	<i>b</i>	Second corner.
Vector3	<i>c</i>	Third corner.
Vector3	<i>d</i>	Fourth corner.
float	<i>spanPercent</i>	Position along span (0-1).

Returns

Type	Description
Vector3	Point on span line.

InverseTransformPointUnscaled(Transform, Vector3)

Transforms a point from world to local space without applying scale.

Declaration

```
public static Vector3 InverseTransformPointUnscaled(this Transform transform, Vector3 position)
```

Parameters

Type	Name	Description
Transform	<i>transform</i>	Transform to use.
Vector3	<i>position</i>	World position.

Returns

Type	Description
Vector3	Local position without scale.

LinePlaneIntersection(Vector3, Vector3, Vector3, Vector3)

Finds the intersection point between a line and a plane.

Declaration

```
public static Vector3 LinePlaneIntersection(Vector3 planePoint, Vector3 planeNormal, Vector3 linePoint, Vector3 lineDirection)
```

Parameters

Type	Name	Description
Vector3	<i>planePoint</i>	Point on the plane.
Vector3	<i>planeNormal</i>	Normal vector of the plane.
Vector3	<i>linePoint</i>	Point on the line.
Vector3	<i>lineDirection</i>	Direction of the line.

Returns

Type	Description
Vector3	Intersection point, or Vector3.zero if parallel.

MeshArea(Mesh)

Calculates area of a complete mesh.

Declaration

```
public static float MeshArea(Mesh mesh)
```

Parameters

Type	Name	Description
Mesh	<i>mesh</i>	

Returns

Type	Description
float	

NearEqual(Vector3, Vector3, float)

Checks if two Vector3 values are approximately equal within a threshold. Uses squared magnitude for performance.

Declaration

```
public static bool NearEqual(this Vector3 a, Vector3 b, float threshold = 0.01)
```

Parameters

Type	Name	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html)	<i>a</i>	First vector.
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html)	<i>b</i>	Second vector.
float (https://learn.microsoft.com/dotnet/api/system.single)	<i>threshold</i>	Maximum squared distance to consider equal.

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	True if vectors are within threshold distance.

NearestPointOnLine(Vector3, Vector3, Vector3)

Finds the nearest point on an infinite line to a given point.

Declaration

```
public static Vector3 NearestPointOnLine(Vector3 linePnt, Vector3 lineDir, Vector3 pnt)
```

Parameters

Type	Name	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html)	<i>linePnt</i>	Point on the line.
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html)	<i>lineDir</i>	Direction of the line.
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html)	<i>pnt</i>	Point to find nearest point from.

Returns

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html)	Nearest point on the line.

NearlyEqual(float, float, double)

Checks if two float values are nearly equal within an epsilon threshold.

Declaration

```
public static bool NearlyEqual(this float a, float b, double epsilon)
```

Parameters

Type	Name	Description
float	<i>a</i>	First value.
float	<i>b</i>	Second value.
double	<i>epsilon</i>	Maximum difference to consider equal.

Returns

Type	Description
bool	True if values are within epsilon.

Perpendicular(Vector3)

Calculates a perpendicular vector to the given vector.

Declaration

```
public static Vector3 Perpendicular(this Vector3 v)
```

Parameters

Type	Name	Description
Vector3	<i>v</i>	Input vector.

Returns

Type	Description
Vector3	Perpendicular vector.

PointInTriangle(Vector3, Vector3, Vector3, Vector3, float)

Checks if a point lies inside a triangle.

Declaration

```
public static bool PointInTriangle(Vector3 A, Vector3 B, Vector3 C, Vector3 P, float dotThreshold = 0.001)
```

Parameters

Type	Name	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	A	First triangle vertex.
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	B	Second triangle vertex.
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	C	Third triangle vertex.
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	P	Point to test.
float (https://learn.microsoft.com/dotnet/api/system.single).	dotThreshold	Tolerance for point-on-plane test.

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	True if point is inside triangle.

PointIsInsideRect(Vector2)

Checks if a 2D point is inside the screen rectangle.

Declaration

```
public static bool PointIsInsideRect(Vector2 point)
```

Parameters

Type	Name	Description
Vector2 (https://docs.unity3d.com/ScriptReference/Vector2.html).	point	Point to check.

Returns

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean).	True if point is inside screen bounds.

ProjectedMeshArea(Mesh, Vector3)

Calculates area of a mesh as viewed from the direction vector.

Declaration

```
public static float ProjectedMeshArea(Mesh mesh, Vector3 direction)
```

Parameters

Type	Name	Description
Mesh	<i>mesh</i>	
Vector3	<i>direction</i>	

Returns

Type	Description
float	

QuadLerp(Vector3, Vector3, Vector3, Vector3, float, float)

Performs bilinear interpolation on a quad defined by four points.

Declaration

```
public static Vector3 QuadLerp(Vector3 a, Vector3 b, Vector3 c, Vector3 d, float u, float v)
```

Parameters

Type	Name	Description
Vector3	<i>a</i>	First corner.
Vector3	<i>b</i>	Second corner.
Vector3	<i>c</i>	Third corner.
Vector3	<i>d</i>	Fourth corner.
float	<i>u</i>	U parameter (0-1).
float	<i>v</i>	V parameter (0-1).

Returns

Type	Description
Vector3	Interpolated point.

QuaternionMagnitude(Quaternion)

Calculates the magnitude of a quaternion.

Declaration

```
public static float QuaternionMagnitude(Quaternion q)
```

Parameters

Type	Name	Description
Quaternion (https://docs.unity3d.com/ScriptReference/Quaternion.html)	q	Quaternion.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Magnitude.

RectArea(Vector3, Vector3, Vector3, Vector3)

Calculates the area of a rectangle from four corner points.

Declaration

```
public static float RectArea(Vector3 p1, Vector3 p2, Vector3 p3, Vector3 p4)
```

Parameters

Type	Name	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	$p1$	First corner.
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	$p2$	Second corner.
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	$p3$	Third corner.
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	$p4$	Fourth corner.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Area of the rectangle.

RotatePointAroundPivot(Vector3, Vector3, Vector3)

Rotates a point around a pivot by the specified angles.

Declaration

```
public static Vector3 RotatePointAroundPivot(Vector3 point, Vector3 pivot, Vector3 angles)
```

Parameters

Type	Name	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	$point$	Point to rotate.
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	$pivot$	Pivot point.
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	$angles$	Euler angles for rotation.

Returns

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	Rotated point.

RoundToStep(int, int)

Rounds a value to the nearest multiple of step.

Declaration

```
public static float RoundToStep(int value, int step)
```

Parameters

Type	Name	Description
int (https://learn.microsoft.com/dotnet/api/system.int32).	<i>value</i>	Value to round.
int (https://learn.microsoft.com/dotnet/api/system.int32).	<i>step</i>	Step size.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Rounded value.

RoundToStep(float, float)

Rounds a value to the nearest multiple of step.

Declaration

```
public static float RoundToStep(float value, float step)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	<i>value</i>	Value to round.
float (https://learn.microsoft.com/dotnet/api/system.single).	<i>step</i>	Step size.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Rounded value.

RoundedMax(Vector3)

Returns a vector with only the largest component preserved (rounded to 1 or -1), others set to 0.

Declaration

```
public static Vector3 RoundedMax(this Vector3 v)
```

Parameters

Type	Name	Description
Vector3	v	Input vector.

Returns

Type	Description
Vector3	Vector with dominant axis isolated.

SignedVolumeOfTriangle(Vector3, Vector3, Vector3)

Calculates the signed volume contribution of a triangle relative to the origin.

Declaration

```
public static float SignedVolumeOfTriangle(Vector3 p1, Vector3 p2, Vector3 p3)
```

Parameters

Type	Name	Description
Vector3	p1	First vertex.
Vector3	p2	Second vertex.
Vector3	p3	Third vertex.

Returns

Type	Description
float	Signed volume.

SquareDistance(Vector3, Vector3)

Calculates squared distance between two points. Faster than regular distance.

Declaration

```
public static float SquareDistance(Vector3 a, Vector3 b)
```

Parameters

Type	Name	Description
Vector3	<i>a</i>	First point.
Vector3	<i>b</i>	Second point.

Returns

Type	Description
float	Squared distance.

TransformPointUnscaled(Transform, Vector3)

Transforms a point from local to world space without applying scale.

Declaration

```
public static Vector3 TransformPointUnscaled(this Transform transform, Vector3 position)
```

Parameters

Type	Name	Description
Transform	<i>transform</i>	Transform to use.
Vector3	<i>position</i>	Local position.

Returns

Type	Description
Vector3	World position without scale.

TriArea(Vector3, Vector3, Vector3)

Calculates area of a single triangle from it's three points.

Declaration

```
public static float TriArea(Vector3 p1, Vector3 p2, Vector3 p3)
```

Parameters

Type	Name	Description
Vector3	<i>p1</i>	
Vector3	<i>p2</i>	
Vector3	<i>p3</i>	

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

TriArea(Vector3, Vector3, Vector3, Vector3)

Declaration

```
public static float TriArea(Vector3 p1, Vector3 p2, Vector3 p3, Vector3 view)
```

Parameters

Type	Name	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	<i>p1</i>	
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	<i>p2</i>	
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	<i>p3</i>	
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	<i>view</i>	

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

Vector3Abs(Vector3)

Returns a vector with absolute values of all components.

Declaration

```
public static Vector3 Vector3Abs(Vector3 v)
```

Parameters

Type	Name	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	<i>v</i>	Input vector.

Returns

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	Vector with absolute values.

Vector3Lerp(Vector3, Vector3, float)

Linear interpolation between two vectors with value clamping.

Declaration

```
public static Vector3 Vector3Lerp(Vector3 v1, Vector3 v2, float value)
```

Parameters

Type	Name	Description
Vector3	v1	Start vector.
Vector3	v2	End vector.
float	value	Interpolation value (0-1).

Returns

Type	Description
Vector3	Interpolated vector.

Vector3OneOver(Vector3)

Returns a vector with reciprocal values (1/x, 1/y, 1/z).

Declaration

```
public static Vector3 Vector3OneOver(Vector3 v)
```

Parameters

Type	Name	Description
Vector3	v	Input vector.

Returns

Type	Description
Vector3	Vector with reciprocal values.

Vector3RoundToInt(Vector3)

Rounds all components of a vector to nearest integer.

Declaration

```
public static Vector3 Vector3RoundToInt(Vector3 v)
```

Parameters

Type	Name	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	v	Input vector.

Returns

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	Rounded vector.

VolumeOfMesh(Mesh)

Calculates the volume enclosed by a mesh.

Declaration

```
public static float VolumeOfMesh(Mesh mesh)
```

Parameters

Type	Name	Description
Mesh (https://docs.unity3d.com/ScriptReference/Mesh.html).	mesh	Mesh to calculate volume for.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Volume of the mesh.

Class PIDController

Proportional-Integral-Derivative controller for smooth value regulation. Used for automated control systems like cruise control, stability systems, and steering assistance.

Inheritance

↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)
↳ PIDController

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Utility \(NWH.Common.Utility.html\)](#).

Assembly: NWH.Common.dll

Syntax

```
public class PIDController
```

Remarks

PID controllers combine three control strategies:

- Proportional: Reacts to current error
- Integral: Eliminates accumulated error over time
- Derivative: Anticipates future error based on rate of change Tune the three gain values to achieve desired response characteristics.

Constructors

PIDController(float, float, float, float, float)

Creates a new PID controller with specified gains and output limits.

Declaration

```
public PIDController(float gainProportional, float gainIntegral, float gainDerivative, float  
outputMin, float outputMax)
```

Parameters

Type	Name	Description
<code>float</code> (https://learn.microsoft.com/dotnet/api/system.single).	<code>gainProportional</code>	Proportional gain (Kp). Higher values increase response to current error.
<code>float</code> (https://learn.microsoft.com/dotnet/api/system.single).	<code>gainIntegral</code>	Integral gain (Ki). Higher values eliminate steady-state error faster.
<code>float</code> (https://learn.microsoft.com/dotnet/api/system.single).	<code>gainDerivative</code>	Derivative gain (Kd). Higher values dampen oscillations.
<code>float</code> (https://learn.microsoft.com/dotnet/api/system.single).	<code>outputMin</code>	Minimum output value.
<code>float</code> (https://learn.microsoft.com/dotnet/api/system.single).	<code>outputMax</code>	Maximum output value.

Fields

maxValue

Maximum output value. Output will be clamped to this value.

Declaration

```
public float maxValue
```

Field Value

Type	Description
<code>float</code> (https://learn.microsoft.com/dotnet/api/system.single).	

minValue

Minimum output value. Output will be clamped to this value.

Declaration

```
public float minValue
```

Field Value

Type	Description
float	

Properties

GainDerivative

The derivative term is proportional to the rate of change of the error

Declaration

```
public float GainDerivative { get; set; }
```

Property Value

Type	Description
float	

GainIntegral

The integral term is proportional to both the magnitude of the error and the duration of the error

Declaration

```
public float GainIntegral { get; set; }
```

Property Value

Type	Description
float	

GainProportional

The proportional term produces an output value that is proportional to the current error value

Declaration

```
public float GainProportional { get; set; }
```

Property Value

Type	Description
float	

Remarks

Tuning theory and industrial practice indicate that the proportional term should contribute the bulk of the output change.

IntegralTerm

Adjustment made by considering the accumulated error over time

Declaration

```
public float IntegralTerm { get; }
```

Property Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single) .	

Remarks

An alternative formulation of the integral action, is the proportional-summation-difference used in discrete-time systems

ProcessVariable

The current value

Declaration

```
public float ProcessVariable { get; set; }
```

Property Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single) .	

ProcessVariableLast

The last reported value (used to calculate the rate of change)

Declaration

```
public float ProcessVariableLast { get; }
```

Property Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single) .	

SetPoint

The desired value

Declaration

```
public float SetPoint { get; set; }
```

Property Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single)	

Methods

ControlVariable(float)

The controller output

Declaration

```
public float ControlVariable(float timeSinceLastUpdate)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single)	<i>timeSinceLastUpdate</i>	timespan of the elapsed time since the previous time that ControlVariable was called

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single)	Value of the variable that needs to be controlled

Class QuaternionExtensions

Extension methods for advanced Quaternion operations. Provides interpolation methods with control over rotation direction.

Inheritance

↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)
↳ QuaternionExtensions

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Utility \(NWH.Common.Utility.html\)](#).

Assembly: NWH.Common.dll

Syntax

```
public static class QuaternionExtensions
```

Methods

Add(Quaternion, Quaternion)

Adds two quaternions component-wise.

Declaration

```
public static Quaternion Add(Quaternion p, Quaternion q)
```

Parameters

Type	Name	Description
Quaternion (https://docs.unity3d.com/ScriptReference/Quaternion.html)	<i>p</i>	First quaternion.
Quaternion (https://docs.unity3d.com/ScriptReference/Quaternion.html)	<i>q</i>	Second quaternion.

Returns

Type	Description
Quaternion (https://docs.unity3d.com/ScriptReference/Quaternion.html)	Component-wise sum.

Lerp(Quaternion, Quaternion, float, bool)

Linear interpolation between two quaternions with optional short/long path control. Unlike Unity's Quaternion.Lerp, this allows choosing rotation direction.

Declaration

```
public static Quaternion Lerp(Quaternion p, Quaternion q, float t, bool shortWay)
```

Parameters

Type	Name	Description
Quaternion (https://docs.unity3d.com/ScriptReference/Quaternion.htm)	<i>p</i>	Starting rotation.
Quaternion (https://docs.unity3d.com/ScriptReference/Quaternion.htm)	<i>q</i>	Target rotation.
float (https://learn.microsoft.com/dotnet/api/system.single)	<i>t</i>	Interpolation factor (0 to 1).
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	<i>shortWay</i>	True for shortest rotation path, false for longest.

Returns

Type	Description
Quaternion (https://docs.unity3d.com/ScriptReference/Quaternion.html)	Interpolated quaternion.

ScalarMultiply(Quaternion, float)

Multiplies all components of a quaternion by a scalar value.

Declaration

```
public static Quaternion ScalarMultiply(Quaternion input, float scalar)
```

Parameters

Type	Name	Description
Quaternion (https://docs.unity3d.com/ScriptReference/Quaternion.html)	<i>input</i>	Input quaternion.
float (https://learn.microsoft.com/dotnet/api/system.single)	<i>scalar</i>	Scalar multiplier.

Returns

Type	Description
Quaternion (https://docs.unity3d.com/ScriptReference/Quaternion.html)	Scaled quaternion.

Slerp(Quaternion, Quaternion, float, bool)

Spherical linear interpolation between two quaternions with optional short/long path control. Provides smooth rotation interpolation with control over rotation direction.

Declaration

```
public static Quaternion Slerp(Quaternion p, Quaternion q, float t, bool shortWay)
```

Parameters

Type	Name	Description
Quaternion (https://docs.unity3d.com/ScriptReference/Quaternion.html)	<i>p</i>	Starting rotation.
Quaternion (https://docs.unity3d.com/ScriptReference/Quaternion.html)	<i>q</i>	Target rotation.
float (https://learn.microsoft.com/dotnet/api/system.single)	<i>t</i>	Interpolation factor (0 to 1).
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	<i>shortWay</i>	True for shortest rotation path, false for longest.

Returns

Type	Description
Quaternion (https://docs.unity3d.com/ScriptReference/Quaternion.html)	Interpolated quaternion.

Class UnitConverter

Static utility class for converting between various units of measurement. Includes conversions for distance, speed, fuel efficiency, and angular velocity.

Inheritance

↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)
↳ UnitConverter

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Utility \(NWH.Common.Utility.html\)](#).

Assembly: NWH.Common.dll

Syntax

```
public static class UnitConverter
```

Methods

AngularVelocityToRPM(float)

Converts angular velocity (rad/s) to rotations per minute.

Declaration

```
public static float AngularVelocityToRPM(float angularVelocity)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	<i>angularVelocity</i>	Angular velocity in rad/s.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Rotations per minute (RPM).

Inch_To_Meter(float)

Converts inches to meters.

Declaration

```
public static float Inch_To_Meter(float inch)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	inch	Value in inches.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Value in meters.

KmlToL100km(float)

km/l to l/100km

Declaration

```
public static float KmlToL100km(float kml)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	kml	Fuel efficiency in km/l.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Fuel efficiency in l/100km.

KmlToMpg(float)

km/l to mpg

Declaration

```
public static float KmlToMpg(float kml)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	kml	Fuel efficiency in km/l.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Fuel efficiency in mpg.

L100kmToKml(float)

l/100km to km/l

Declaration

```
public static float L100kmToKml(float l100km)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	<i>l100km</i>	Fuel efficiency in l/100km.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Fuel efficiency in km/l.

L100kmToMpg(float)

l/100km to mpg

Declaration

```
public static float L100kmToMpg(float l100km)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	<i>l100km</i>	Fuel efficiency in l/100km.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Fuel efficiency in mpg.

Meter_To_Inch(float)

Converts meters to inches.

Declaration

```
public static float Meter_To_Inch(float meters)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	<i>meters</i>	Value in meters.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Value in inches.

MpgToKml(float)

mpg to km/l

Declaration

```
public static float MpgToKml(float mpg)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	<i>mpg</i>	Fuel efficiency in mpg.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Fuel efficiency in km/l.

MpgToL100km(float)

mpg to l/100km

Declaration

```
public static float MpgToL100km(float mpg)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	<i>mpg</i>	Fuel efficiency in mpg.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Fuel efficiency in l/100km.

MphToKph(float)

miles/h to km/h

Declaration

```
public static float MphToKph(float value)
```

Parameters

Type	Name	Description
float	<i>value</i>	Speed in mph.

Returns

Type	Description
float	Speed in km/h.

MpsToKph(float)

m/s to km/h

Declaration

```
public static float MpsToKph(float value)
```

Parameters

Type	Name	Description
float	<i>value</i>	Speed in m/s.

Returns

Type	Description
float	Speed in km/h.

MpsToMph(float)

m/s to miles/h

Declaration

```
public static float MpsToMph(float value)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	value	Speed in m/s.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Speed in mph.

RPMToAngularVelocity(float)

Converts rotations per minute to angular velocity (rad/s).

Declaration

```
public static float RPMToAngularVelocity(float RPM)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	RPM	Rotations per minute.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Angular velocity in rad/s.

Speed_kmhToMph(float)

Converts km/h to mph.

Declaration

```
public static float Speed_kmhToMph(float kmh)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	kmh	Speed in km/h.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Speed in mph.

Speed_kmhToMs(float)

Converts km/h to m/s.

Declaration

```
public static float Speed_kmhToMs(float kmh)
```

Parameters

Type	Name	Description
float	<i>kmh</i>	Speed in km/h.

Returns

Type	Description
float	Speed in m/s.

Speed_mphToKmh(float)

Converts mph to km/h.

Declaration

```
public static float Speed_mphToKmh(float mph)
```

Parameters

Type	Name	Description
float	<i>mph</i>	Speed in mph.

Returns

Type	Description
float	Speed in km/h.

Speed_mphToMs(float)

Converts mph to m/s.

Declaration

```
public static float Speed_mphToMs(float mph)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	mph	Speed in mph.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Speed in m/s.

Speed_msToKph(float)

Converts m/s to km/h.

Declaration

```
public static float Speed_msToKph(float ms)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	ms	Speed in m/s.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Speed in km/h.

Speed_msToMph(float)

Converts m/s to mph.

Declaration

```
public static float Speed_msToMph(float ms)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	ms	Speed in m/s.

Returns

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	Speed in mph.

Namespace NWH.Common.Vehicles

Classes

[FollowVehicleState \(NWH.Common.Vehicles.FollowVehicleState.html\)](#)

Automatically enables/disables a GameObject based on parent vehicle's active state. Useful for optimizing performance by disabling effects, audio, or visuals when a vehicle is inactive.

[ShowInSettings \(NWH.Common.Vehicles.ShowInSettings.html\)](#)

Attribute that marks a field to be displayed in runtime settings UI. Allows players to adjust vehicle parameters during gameplay.

[ShowInTelemetry \(NWH.Common.Vehicles.ShowInTelemetry.html\)](#)

Attribute that marks a field or property to be displayed in the runtime telemetry UI. Used for monitoring vehicle parameters during gameplay and debugging.

[Vehicle \(NWH.Common.Vehicles.Vehicle.html\)](#)

Base class for all NWH vehicles including VehiclePhysics2.VehicleController and DWP2.AdvancedShipController.

[VehicleReflectionProbe \(NWH.Common.Vehicles.VehicleReflectionProbe.html\)](#)

Manages vehicle reflection probe settings, switching between baked and realtime modes based on vehicle activity to optimize performance.

Enums

[VehicleReflectionProbe.ProbeType](#)

[\(NWH.Common.Vehicles.VehicleReflectionProbe.ProbeType.html\)](#)

Type of reflection probe to use.

Class FollowVehicleState

Automatically enables/disables a GameObject based on parent vehicle's active state. Useful for optimizing performance by disabling effects, audio, or visuals when a vehicle is inactive.

Inheritance

- ↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)
- ↳ [Object](https://docs.unity3d.com/ScriptReference/Object.html) (<https://docs.unity3d.com/ScriptReference/Object.html>)
- ↳ [Component](https://docs.unity3d.com/ScriptReference/Component.html) (<https://docs.unity3d.com/ScriptReference/Component.html>)
- ↳ [Behaviour](https://docs.unity3d.com/ScriptReference/Behaviour.html) (<https://docs.unity3d.com/ScriptReference/Behaviour.html>)
- ↳ [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html) (<https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>)
- ↳ FollowVehicleState

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Vehicles \(NWH.Common.Vehicles.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
[DefaultExecutionOrder(21)]  
public class FollowVehicleState : MonoBehaviour
```

Remarks

Attach this component to child objects that should only be active when the vehicle is being simulated. When the vehicle is put to sleep (disabled), attached objects are also disabled, saving processing time for effects that wouldn't be visible anyway.

Class ShowInSettings

Attribute that marks a field to be displayed in runtime settings UI. Allows players to adjust vehicle parameters during gameplay.

Inheritance

```
↳ object (https://learn.microsoft.com/dotnet/api/system.object)
  ↳ Attribute (https://learn.microsoft.com/dotnet/api/system.attribute)
    ↳ ShowInSettings
```

Namespace: [NWH \(NWH.html\)](#), [Common \(NWH.Common.html\)](#), [Vehicles \(NWH.Common.Vehicles.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
[AttributeUsage(AttributeTargets.Field)]
public class ShowInSettings : Attribute
```

Constructors

ShowInSettings()

Declaration

```
public ShowInSettings()
```

ShowInSettings(float, float, float)

Creates a settings attribute with specified min, max, and step values.

Declaration

```
public ShowInSettings(float min, float max, float step = 0.1)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	<i>min</i>	
float (https://learn.microsoft.com/dotnet/api/system.single).	<i>max</i>	
float (https://learn.microsoft.com/dotnet/api/system.single).	<i>step</i>	

ShowInSettings(string)

Creates a settings attribute with a custom display name.

Declaration

```
public ShowInSettings(string name)
```

Parameters

Type	Name	Description
string .	<i>name</i>	

ShowInSettings(string, float, float, float)

Creates a settings attribute with custom name and value constraints.

Declaration

```
public ShowInSettings(string name, float min, float max, float step = 0.1)
```

Parameters

Type	Name	Description
string .	<i>name</i>	
float .	<i>min</i>	
float .	<i>max</i>	
float .	<i>step</i>	

Fields

max

Maximum value for the setting slider.

Declaration

```
public float max
```

Field Value

Type	Description
float .	

min

Minimum value for the setting slider.

Declaration

```
public float min
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

name

Display name for the setting in the UI.

Declaration

```
public string name
```

Field Value

Type	Description
string (https://learn.microsoft.com/dotnet/api/system.string).	

step

Increment step for the slider. Smaller values allow finer adjustment.

Declaration

```
public float step
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

Class ShowInTelemetry

Attribute that marks a field or property to be displayed in the runtime telemetry UI. Used for monitoring vehicle parameters during gameplay and debugging.

Inheritance

```
↳ object (https://learn.microsoft.com/dotnet/api/system.object)
  ↳ Attribute (https://learn.microsoft.com/dotnet/api/system.attribute).
    ↳ ShowInTelemetry
```

Namespace: [NWH \(NWH.html\)](#), [Common \(NWH.Common.html\)](#), [Vehicles \(NWH.Common.Vehicles.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
[AttributeUsage(AttributeTargets.All)]
public class ShowInTelemetry : Attribute
```

Constructors

ShowInTelemetry(float, float, string, string, int)

Creates a ShowInTelemetry attribute with optional parameters.

Declaration

```
public ShowInTelemetry(float min = NaN, float max = NaN, string format = null, string unit =
null, int priority = 1)
```

Parameters

Type	Name	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	<i>min</i>	
float (https://learn.microsoft.com/dotnet/api/system.single).	<i>max</i>	
string (https://learn.microsoft.com/dotnet/api/system.string).	<i>format</i>	
string (https://learn.microsoft.com/dotnet/api/system.string).	<i>unit</i>	
int (https://learn.microsoft.com/dotnet/api/system.int32).	<i>priority</i>	

Properties

Format

Format string for displaying the value (e.g., "0.00", "0.0").

Declaration

```
public string Format { get; set; }
```

Property Value

Type	Description
string (https://learn.microsoft.com/dotnet/api/system.string).	

Max

Maximum value for the field (used for progress bar visualization).

Declaration

```
public float Max { get; set; }
```

Property Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

Min

Minimum value for the field (used for progress bar visualization).

Declaration

```
public float Min { get; set; }
```

Property Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

Priority

Display priority. 0 = highest (always visible), 3 = lowest (detailed info).

Declaration

```
public int Priority { get; set; }
```

Property Value

Type	Description
int (https://learn.microsoft.com/dotnet/api/system.int32)	

Unit

Unit of measurement (e.g., "km/h", "RPM", "N", "°").

Declaration

```
public string Unit { get; set; }
```

Property Value

Type	Description
string (https://learn.microsoft.com/dotnet/api/system.string)	

Class Vehicle

Base class for all NWH vehicles including VehiclePhysics2.VehicleController and DWP2.AdvancedShipController.

Inheritance

```
↳ object (https://learn.microsoft.com/dotnet/api/system.object)
  ↳ Object (https://docs.unity3d.com/ScriptReference/Object.html)
    ↳ Component (https://docs.unity3d.com/ScriptReference/Component.html)
      ↳ Behaviour (https://docs.unity3d.com/ScriptReference/Behaviour.html)
        ↳ MonoBehaviour (https://docs.unity3d.com/ScriptReference/MonoBehaviour.html)
          ↳ Vehicle
```

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Vehicles \(NWH.Common.Vehicles.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
[DisallowMultipleComponent]
[RequireComponent(typeof(Rigidbody))]
public abstract class Vehicle : MonoBehaviour
```

Fields

ActiveVehicles

Declaration

```
public static List<Vehicle> ActiveVehicles
```

Field Value

Type	Description
List (https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1)< Vehicle (NWH.Common.Vehicles.Vehicle.html) >	

INPUT_DEADZONE

Any input below this value will register as no input.

Declaration

```
public const float INPUT_DEADZONE = 0.02
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

KINDA_SMALL_NUMBER

Like SMALL_NUMBER but a bit bigger.

Declaration

```
public const float KINDA_SMALL_NUMBER = 0.01
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

SMALL_NUMBER

Anything below this can be considered 0.

Declaration

```
public const float SMALL_NUMBER = 1E-05
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

SPEED_DEADZONE

Any speed below this value will register as no speed.

Declaration

```
public const float SPEED_DEADZONE = 0.2
```

Field Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

isPlayerControllable

True if the vehicle can be driven by the player. False if the vehicle is passive (such as a trailer). A vehicle that has isPlayerControllable can be the ActiveVehicle.

Declaration

```
public bool isPlayerControllable
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

onActiveVehicleChanged

Called when active vehicle is changed. First parameter is the previously active vehicle and the second parameter is the currently active vehicle (at the time of the callback). Params can be null.

Declaration

```
public static UnityEvent<Vehicle, Vehicle> onActiveVehicleChanged
```

Field Value

Type	Description
UnityEvent< Vehicle (NWH.Common.Vehicles.Vehicle.html) , Vehicle (NWH.Common.Vehicles.Vehicle.html) >	

onCameraEnterVehicle

Called when the camera enters the vehicle.

Declaration

```
public UnityEvent onCameraEnterVehicle
```

Field Value

Type	Description
UnityEvent (https://docs.unity3d.com/ScriptReference/Events.UnityEvent.html)	

onCameraExitVehicle

Called when the camera exits the vehicle.

Declaration

```
public UnityEvent onCameraExitVehicle
```

Field Value

Type	Description
UnityEvent (https://docs.unity3d.com/ScriptReference/Events.UnityEvent.html).	

onDisable

Called when vehicle is put to sleep.

Declaration

```
[Tooltip("    Called when vehicle is put to sleep.")]
[NonSerialized]
public UnityEvent onDisable
```

Field Value

Type	Description
UnityEvent (https://docs.unity3d.com/ScriptReference/Events.UnityEvent.html).	

onEnable

Called when vehicle is woken up.

Declaration

```
[Tooltip("    Called when vehicle is woken up.")]
[NonSerialized]
public UnityEvent onEnable
```

Field Value

Type	Description
UnityEvent (https://docs.unity3d.com/ScriptReference/Events.UnityEvent.html).	

onMultiplayerStatusChanged

Invoked when MultiplayerIsRemote value gets changed. Is true if remote.

Declaration

```
public UnityEvent<bool> onMultiplayerStatusChanged
```

Field Value

Type	Description
UnityEvent<bool (https://learn.microsoft.com/dotnet/api/system.boolean)>	

vehicleRigidbody

Cached value of vehicle rigidbody.

Declaration

```
[Tooltip("    Cached value of vehicle rigidbody.")]
[NonSerialized]
public Rigidbody vehicleRigidbody
```

Field Value

Type	Description
Rigidbody (https://docs.unity3d.com/ScriptReference/Rigidbody.html).	

vehicleTransform

Cached value of vehicle transform.

Declaration

```
[Tooltip("    Cached value of vehicle transform.")]
[NonSerialized]
public Transform vehicleTransform
```

Field Value

Type	Description
Transform (https://docs.unity3d.com/ScriptReference/Transform.html).	

Properties

ActiveVehicle

Declaration

```
public static Vehicle ActiveVehicle { get; }
```

Property Value

Type	Description
Vehicle (NWH.Common.Vehicles.Vehicle.html)	

AngularVelocity

Cached angular velocity of the vehicle.

Declaration

```
public Vector3 AngularVelocity { get; protected set; }
```

Property Value

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html)	

AngularVelocityMagnitude

Cached angular velocity magnitude of the vehicle.

Declaration

```
public float AngularVelocityMagnitude { get; protected set; }
```

Property Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single)	

CameraInsideVehicle

True when camera is inside vehicle (cockpit, cabin, etc.). Set by the 'CameraInsideVehicle' component. Used for audio effects.

Declaration

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

Property Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

LocalAcceleration

Cached acceleration in local coordinates (z-forward)

Declaration

```
public Vector3 LocalAcceleration { get; }
```

Property Value

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html).	

LocalForwardAcceleration

Cached acceleration in forward direction in local coordinates (z-forward).

Declaration

```
public float LocalForwardAcceleration { get; }
```

Property Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

LocalForwardVelocity

Cached velocity in forward direction in local coordinates (z-forward).

Declaration

```
public float LocalForwardVelocity { get; }
```

Property Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single).	

LocalVelocity

Cached velocity in m/s in local coordinates.

Declaration

```
[ShowInTelemetry(NaN, NaN, null, null, 1)]
public Vector3 LocalVelocity { get; }
```

Property Value

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html)	

MultiplayerIsRemote

True if the vehicle is a client (remote) and not simulated. If true the input is expected to be synced through the network.

Declaration

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

Property Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	

Speed

Cached speed of the vehicle in the forward direction. ALWAYS POSITIVE. For positive/negative version use SpeedSigned.

Declaration

```
[ShowInTelemetry(NaN, NaN, "0.0", "m/s", 1)]
public float Speed { get; }
```

Property Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single)	

SpeedSigned

Cached speed of the vehicle in the forward direction. Can be positive (forward) or negative (reverse). Equal to LocalForwardVelocity.

Declaration

```
[ShowInTelemetry(NaN, NaN, "0.0", "m/s", 1)]  
public float SpeedSigned { get; }
```

Property Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single)	

Velocity

Cached velocity of the vehicle in world coordinates.

Declaration

```
public Vector3 Velocity { get; protected set; }
```

Property Value

Type	Description
Vector3 (https://docs.unity3d.com/ScriptReference/Vector3.html)	

VelocityMagnitude

Cached velocity magnitude of the vehicle in world coordinates.

Declaration

```
public float VelocityMagnitude { get; protected set; }
```

Property Value

Type	Description
float (https://learn.microsoft.com/dotnet/api/system.single)	

Methods

Awake()

Declaration

```
public virtual void Awake()
```

FixedUpdate()

Declaration

```
public virtual void FixedUpdate()
```

OnDisable()

Declaration

```
public virtual void OnDisable()
```

OnEnable()

Declaration

```
public virtual void OnEnable()
```

Class VehicleReflectionProbe

Manages vehicle reflection probe settings, switching between baked and realtime modes based on vehicle activity to optimize performance.

Inheritance

- ↳ [object](https://learn.microsoft.com/dotnet/api/system.object) (<https://learn.microsoft.com/dotnet/api/system.object>)
- ↳ [Object](https://docs.unity3d.com/ScriptReference/Object.html) (<https://docs.unity3d.com/ScriptReference/Object.html>)
- ↳ [Component](https://docs.unity3d.com/ScriptReference/Component.html) (<https://docs.unity3d.com/ScriptReference/Component.html>)
- ↳ [Behaviour](https://docs.unity3d.com/ScriptReference/Behaviour.html) (<https://docs.unity3d.com/ScriptReference/Behaviour.html>)
- ↳ [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html) (<https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>)
- ↳ VehicleReflectionProbe

Namespace: [NWH \(NWH.html\)](#), [Common \(NWH.Common.html\)](#), [Vehicles \(NWH.Common.Vehicles.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
[RequireComponent(typeof(ReflectionProbe))]  
[DefaultExecutionOrder(19)]  
public class VehicleReflectionProbe : MonoBehaviour
```

Remarks

Realtime reflection probes are expensive. This component switches to cheaper baked probes when the vehicle is inactive, maintaining visual quality while improving performance. The probe is automatically re-baked when the vehicle becomes inactive to capture the current environment state.

Fields

asleepProbeType

Probe type to use when vehicle is inactive/sleeping. Default is Baked for performance.

Declaration

```
public VehicleReflectionProbe.ProbeType asleepProbeType
```

Field Value

Type	Description
VehicleReflectionProbe (NWH.Common.Vehicles.VehicleReflectionProbe.html) , ProbeType (NWH.Common.Vehicles.VehicleReflectionProbe.ProbeType.html)	

awakeProbeType

Probe type to use when vehicle is active. Default is Realtime for accurate reflections.

Declaration

```
public VehicleReflectionProbe.ProbeType awakeProbeType
```

Field Value

Type	Description
VehicleReflectionProbe (NWH.Common.Vehicles.VehicleReflectionProbe.html) .	
ProbeType (NWH.Common.Vehicles.VehicleReflectionProbe.ProbeType.html) .	

bakeOnSleep

Automatically bake the probe when vehicle becomes inactive to capture environment state.

Declaration

```
public bool bakeOnSleep
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean) .	

bakeOnStart

Bake the probe once on start for initial environment capture.

Declaration

```
public bool bakeOnStart
```

Field Value

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean) .	

Enum VehicleReflectionProbe.ProbeType

Type of reflection probe to use.

Namespace: [NWH \(NWH.html\)](#).[Common \(NWH.Common.html\)](#).[Vehicles \(NWH.Common.Vehicles.html\)](#)

Assembly: NWH.Common.dll

Syntax

```
public enum VehicleReflectionProbe.ProbeType
```

Fields

Name	Description
Baked	Pre-rendered cubemap, low cost but static.
Realtime	Updates every frame, high cost but accurate.