BLInformatique — GameTools – User Guide

⚓ 💪TEAM VICTOR & BORIS

Unity 6000.4.7f1 • Generated 2026-06-09 15:27

Table of Contents

📦 Components (34)

- 📦 Components Overview
- 🧩 BLInformatique.GameTools (34)
  - AdditiveSceneLoader
  - CameraFollow
  - CloudLayerAnimator
  - DayNightAudioGeo
  - DoorTrigger
  - DynamicShadowManager
  - EnterCollider
  - EventGroupObjectSwitcher
  - FogZoneFader
  - FPSController
  - FPSFootstepPlayerGT
  - GeoSunCyclePro
  - GeoSunPaletteSO
  - GeoSunPreset
  - GeoSunSharedPalette
  - InteriorAudioZone
  - KeysEvent
  - MouseLockToggle
  - MouseLookCamera
  - NPCNavMeshGT
  - ObjectMoverRotator
  - PunchySkyVolumePreset
  - QuitGameManager
  - rotateObjects
  - RuntimeMinimapGenerator
  - SkyboxApplier
  - StartupSplash
  - SunClockHUDGeo
  - SurfaceDataGT
  - SurfaceTypeGT
  - TimerEventManager
  - UIGameObject
  - UITooltip
  - WaypointGT

📖 Overview

Overview

Welcome to GameTools V5.0

GameTools is a collection of ready-to-use systems designed to accelerate game development in Unity.

The package provides a centralized workflow through the GameTools Editor menu, allowing you to create complete gameplay systems without manually building complex hierarchies.

All major features can be created directly from:

Tools → BLInformatique → Game Tools

This menu gives access to:

🎮 Gameplay Systems

* FPS Player Controller
* Door Systems
* Trigger Systems
* Event Systems
* Surface Detection

⛅ Environment Systems

* GeoSun Day / Night Cycle
* Cloud Layer Animator

* Fog Zones
* Dynamic Shadows
* Skybox Tools

🎵 Audio Systems

* Interior Audio Zones
* Day / Night Ambience

🤖 AI Systems

* NPC Navigation
* Patrol Systems
* Follow Systems
* Waypoint Actions

🗺 User Interface

* Runtime Minimap
* Mission Markers
* Tooltips
* HUD Systems

⚙ Utilities

* Startup Splash
* Mouse Lock
* Timers
* Object Animators

* Scene Utilities

Most GameTools features can be created automatically from the Create menu.

Recommended workflow:

1. Install GameTools Tags And Layers
2. Create a Player FPS
3. Create your gameplay systems
4. Configure the generated objects
5. Press Play

GameTools supports:

✅ Unity 6000

✅ Universal Render Pipeline (URP)

✅ Built-in Render Pipeline

✅ Legacy Input System

✅ New Input System

⚓ 💪 TEAM VICTOR & BORIS

BLInformatique

https://www.blinformatique.fr

🎭 Optional Sections

📌 **INSTALLATION**

Installation

Requirements

GameTools has been designed for modern Unity projects and supports both beginners and advanced developers.

Supported versions:

* Unity 6000 or newer

Supported Render Pipelines:

* Universal Render Pipeline (URP)

* Built-in Render Pipeline

Supported Input Systems:

* Unity Legacy Input System

* Unity New Input System

Importing GameTools

Import the GameTools package into your Unity project.

Once imported, Unity will automatically compile the scripts and install the GameTools Editor menus.

First Launch

After importing the package, a new menu will appear in Unity:

Tools → BLInformatique → Game Tools

This menu is the main entry point for all GameTools features.

Recommended Setup

Before creating gameplay systems, run:

Tools → BLInformatique → Game Tools → Setup → Install Game Tools Tags And Layers

This automatically installs the recommended tags and layers used by GameTools.

This operation is completely safe and can be executed multiple times.

You are now ready to create GameTools systems inside your project.

⚓ 💪 TEAM VICTOR & BORIS

📌 GAMETOOLS MENU OVERVIEW

GameTools Menu Overview

GameTools uses a centralized workflow based on the Unity Editor menu.

Main menu:

Tools → BLInformatique → Game Tools

From this menu you can access all GameTools features.

Create

The Create section allows you to automatically generate complete systems inside your scene.

Examples:

* Create Player FPS
* Create Runtime Minimap
* Create GeoSun Cycle
* Create NPC Systems
* Create Audio Systems
* Create Environment Systems

Setup

The Setup section contains project configuration utilities.

Examples:

* Install Tags And Layers
* Project Configuration Tools
* Automatic Setup Wizards

Utilities

The Utilities section contains helper tools and productivity features.

Examples:

* Timers
* Event Systems
* Scene Helpers
* Debug Utilities

Workflow Philosophy

GameTools has been designed so that most systems can be created directly from the menu without manually building complex hierarchies.

This significantly reduces setup time and helps maintain a consistent project structure.

⚓ 💪 TEAM VICTOR & BORIS

📌 QUICK START

Quick Start

This guide will help you create your first GameTools scene in just a few minutes.

Step 1 — Install Tags And Layers

Open:

Tools → BLInformatique → Game Tools → Setup → Install Game Tools Tags And Layers

This installs the recommended project configuration used by several GameTools systems.

Step 2 — Create A Player

Open:

Tools → BLInformatique → Game Tools → Create → Create Player FPS

A complete First Person Controller will automatically be created.

Step 3 — Create Additional Systems

Examples:

Tools → BLInformatique → Game Tools → Create → Create Runtime Minimap

Tools → BLInformatique → Game Tools → Create → Create GeoSun Cycle

Tools → BLInformatique → Game Tools → Create → Create NPC

Step 4 — Configure

Select the generated GameObjects and adjust their settings in the Inspector.

Every public field is documented inside this manual.

Step 5 — Press Play

Your scene is now ready for testing.

Most GameTools systems are functional immediately after creation.

⚓💪 TEAM VICTOR & BORIS

📌 **CREATE PLAYER FPS**

Create Player FPS

GameTools can automatically generate a ready-to-use First Person Controller.

Open:

Tools → BLInformatique → Game Tools → Create → Create Player FPS

Generated Components

The generated player includes:

* Character Controller
* FPS Controller
* Mouse Look Camera
* Camera Pivot
* Jump System
* Run System
* Crouch System

Default Controls

Movement:

* W → Forward
* A → Left
* S → Backward

* D → Right

Actions:

* Mouse → Look Around
* Space → Jump
* Left Shift → Run
* Left Control → Crouch

Configuration

The FPS Controller can be customized directly from the Inspector.

Available settings include:

* Movement Speed
* Running Speed
* Mouse Sensitivity
* Jump Force
* Head Bob
* Input System Selection

The controller supports both Legacy Input and the New Input System.

⚓ 💪 TEAM VICTOR & BORIS

📌 Create Runtime Minimap

Create Runtime Minimap

GameTools can automatically create a complete minimap system for your scene.

Open:

Tools → BLInformatique → Game Tools → Create → Create Runtime Minimap

---

Generated Components

The generated minimap includes:

* Minimap Camera
* RuntimeMinimapGenerator
* Minimap UI
* Player Marker
* Runtime Tracking System

Features

The minimap can display:

* Player Position
* Tracked Objects
* Mission Targets
* Mission Distance

* Direction Arrows

* Custom Icons

Mission System Support

Mission targets can be assigned at runtime.

The minimap can automatically display:

* Target Marker

* Edge Arrow

* Distance To Target

This makes it ideal for adventure, survival and open-world games.

Configuration

Available settings include:

* Camera Height

* Zoom Level

* Tracking Layers

* Minimap Size

* Marker Prefabs

* Mission Display

⚓💪 TEAM VICTOR & BORIS

📌 **Create GeoSun Day / Night Cycle**

Create GeoSun Day / Night Cycle

GeoSunCyclePro provides a complete astronomical day and night cycle system.

Open:

Tools → BLInformatique → Game Tools → Create → Create GeoSun Cycle

---

Generated Components

The generated setup includes:

* GeoSunCyclePro
* Directional Sun Light
* Skybox Integration
* Day/Night Events

Features

GeoSun can automatically control:

* Sun Rotation

* Sun Color

* Sun Intensity

* Ambient Lighting

* Reflection Intensity

* Fog

* Skybox Rotation

Real World Locations

You can configure:

* Latitude

* Longitude

* Date

* Time Zone

This allows the sun to follow realistic astronomical calculations.

Events

GeoSun can trigger:

* Day Started

* Night Started

* Hour Changed

* Minute Changed

These events can be connected directly to gameplay systems.

⚓ 💪 TEAM VICTOR & BORIS

📌 **Create NPC Navigation System**

Create NPC Navigation System

GameTools provides a simple but powerful NPC navigation system.

Open:

Tools → BLInformatique → Game Tools → Create → Create NPC

Generated Components

The generated NPC includes:

* NavMeshAgent
* NPCNavMeshGT
* Waypoint Support
* Animator Integration

Behaviour Modes

NPCs can operate in several modes:

* Idle
* Patrol
* Follow
* Flee

Waypoint System

NPCs can navigate between waypoints and execute actions when reaching them.

Examples:

* Play Animations
* Wait
* Trigger Events
* Start Interactions

Animator Integration

NPCNavMeshGT can automatically control Animator parameters.

Supported actions include:

* Walking
* Running

* Idle

* Custom Waypoint Animations

Use Cases

* Civilian NPCs

* Guards

* Animals

* Followers

* Enemies

* Interactive Characters

⚓ 💪 TEAM VICTOR & BORIS

📌 Create Door System

Create Door System

GameTools includes a flexible door system suitable for many types of projects.

Open:

Tools → BLInformatique → Game Tools → Create → Create Door System

---

Supported Door Types

* Single Doors

* Double Doors

* Sliding Doors

* Automatic Doors

* Interactive Doors

Control Methods

Doors can be opened using:

* Triggers

* Keyboard Input

* Input Actions

* Unity Events

* External Scripts

Animation Support

Doors can be controlled through:

* Animator Parameters

* Manual Rotation

* Manual Sliding

Events

The system provides built-in events:

* Door Opened

* Door Closed

* Door Toggled

* Trigger Entered

* Trigger Exited

These events can be connected directly from the Inspector.

⚓ 💪 TEAM VICTOR & BORIS

📌 **Create Interior Audio Zone**

Create Interior Audio Zone

InteriorAudioZone allows realistic transitions between outdoor and indoor environments.

Open:

Tools → BLInformatique → Game Tools → Create → Create Interior Audio Zone

---

Features

The system can automatically modify:

* Outdoor Audio Volume

* Audio Mixer Parameters

* Low Pass Filters

* Ambient Sounds

Presets

Several presets are available:

* Small Room

* House

* Large Building

* Church

* Custom

Audio Mixer Support

InteriorAudioZone can drive exposed Audio Mixer parameters.

This allows advanced control over:

* Exterior Volume

* Interior Volume

* Low Pass Effects

Use Cases

* Houses
* Apartments
* Offices
* Warehouses
* Churches
* Underground Areas

⚓ 💪 TEAM VICTOR & BORIS

📌 **Create Startup Splash**

Create Startup Splash

StartupSplash allows you to display professional startup sequences before entering your game.

Open:

Tools → BLInformatique → Game Tools → Create → Create Startup Splash

---

Features

StartupSplash supports:

* Multiple Intro Messages
* Fade Effects

* Company Logos

* Intro Music

* Automatic Scene Loading

Typical Workflow

1. Create a Startup Splash

2. Add your logo

3. Configure intro messages

4. Assign music

5. Select the next scene

Visual Effects

The system can automatically perform:

* Fade In

* Fade Out

* Logo Transitions

* Music Fade Out

Use Cases

* Studio Logos

* Publisher Logos

* Intro Sequences

* Project Presentations

* Demo Scenes

⚓💪 TEAM VICTOR & BORIS

📌 **Frequently Asked Questions (FAQ)**

Frequently Asked Questions (FAQ)

Does GameTools require additional packages?

No.

GameTools works with standard Unity installations and does not require any third-party assets.

Which Unity versions are supported?

GameTools is designed for Unity 6000 and newer versions.

Is URP supported?

Yes.

GameTools supports:

* Universal Render Pipeline (URP)
* Built-in Render Pipeline

Does GameTools support the New Input System?

Yes.

Most GameTools components support both:

* Legacy Input System
* New Input System

Can I modify the generated objects?

Absolutely.

All generated GameObjects remain fully editable and can be customized to fit your project requirements.

Can I use only a few components?

Yes.

Each GameTools component is designed to work independently.

You can use a single component or combine multiple systems together.

Is GameTools suitable for commercial projects?

Yes.

GameTools can be used in both personal and commercial Unity projects.

Where can I find detailed information about a component?

The Component Reference section of this manual contains detailed documentation for every GameTools component.

📌 **Troubleshooting**

Troubleshooting

The GameTools menu does not appear

Try the following:

1. Wait for Unity compilation to finish.
2. Check the Console for compilation errors.

3. Reimport the GameTools package.

4. Restart Unity.

GameTools menus will not appear while compilation errors are present.

A component is not working

Verify:

* Required references are assigned.

* Required tags and layers are installed.

* The component is enabled.

* The GameObject is active.

Most configuration issues can be solved by reviewing the component Inspector settings.

Input actions are not detected

Verify:

* The correct Input System is selected.

* Input Actions are assigned.

* The Input Action Asset is enabled.

Most components also provide support for the Legacy Input System.

Minimap markers are missing

Verify:

* Target objects have the correct tags.
* Tracking tags are configured.
* The minimap camera culling mask includes the target layers.

NPCs are not moving

Verify:

* A NavMesh has been baked.
* A NavMeshAgent is present.
* Waypoints are assigned correctly.
* The target object exists.

Doors do not open

Verify:

* Trigger colliders are enabled.
* Required tags are configured.

* Input bindings are correct.

* Animator references are assigned.

Most DoorTrigger issues are related to missing references or trigger configuration.

📌 **Support & Credits**

Support & Credits

Support

If you need assistance with GameTools, please contact:

BLInformatique

Website:

https://www.blinformatique.fr

Email:

gametools@blinformatique.fr

Documentation

This manual has been generated using:

ManualBuilder

Developed by:

⚓ 💪 TEAM VICTOR & BORIS

Credits

GameTools is developed and maintained by BLInformatique.

Special thanks to all users who provide feedback and help improve the tools.

Updates

GameTools is actively maintained.

New features, improvements and bug fixes are regularly added through updates.

Please check the Asset Store page for the latest version information.

Thank You

Thank you for choosing GameTools.

We hope these tools help you create amazing projects and save valuable development time.

Happy developing!

⚓💪 TEAM VICTOR & BORIS

📌 **About GameTools**

About GameTools

Built By Developers, For Developers

GameTools was created with a simple goal:

Provide practical, production-ready tools that save development time while remaining easy to use and fully customizable.

After years of developing Unity projects and creating tools for our own games, we decided to gather our most useful systems into a single package.

The result is GameTools.

Why GameTools?

Many developers spend valuable time recreating the same systems for every project:

* Doors

* Day/Night Cycles

* NPC Behaviours

* Minimap Systems

* Audio Zones

* Event Managers

* Utility Components

GameTools provides these systems ready to use, allowing developers to focus on creating their game rather than rebuilding common features.

Designed For Real Projects

Every component included in GameTools was originally developed to solve real production needs.

The systems have been tested in various types of projects, including:

* First Person Games

* Adventure Games

* Survival Games

* Simulation Games

* Vehicle Games

* Open World Projects

This practical approach ensures that GameTools remains lightweight, flexible and easy to integrate.

Unity 6000 Ready

GameTools has been designed specifically for modern Unity workflows.

Supported technologies include:

* Unity 6000
* Universal Render Pipeline (URP)
* Built-in Render Pipeline
* Legacy Input System
* New Input System

Whenever possible, components are designed to support multiple workflows and project configurations.

Our Philosophy

We believe development tools should:

* Be simple to understand
* Be fast to configure
* Remain fully customizable
* Save time without limiting creativity

GameTools follows these principles throughout the entire package.

Most systems can be created directly from the GameTools menu and configured in only a few minutes.

Constantly Improving

GameTools is an evolving project.

New systems, improvements and user-requested features are regularly added through updates.

Feedback from the community plays an important role in helping us improve the package and prioritize future developments.

Our Commitment

Our objective is to provide professional-quality tools that remain accessible to both beginners and experienced developers.

Whether you are creating your first game or working on a large production, GameTools aims to simplify your workflow and accelerate development.

Welcome Aboard

Thank you for supporting our work and for being part of the GameTools community.

We hope GameTools becomes a valuable companion in your Unity projects.

Fair winds and happy developing.

⚓ 💪 TEAM VICTOR & BORIS

BLInformatique

www.blinformatique.fr (http://www.blinformatique.fr)

gametools@blinformatique.fr

CONTENTS

🔎 Filter components…

BLINFORMATIQUE.GAMETOOLS

CloudLayerAnimator

DayNightAudioGeo

DoorTrigger

DynamicShadowManager

EnterCollider

EventGroupObjectSwitcher

FogZoneFader

FPSController

FPSFootstepPlayerGT

GeoSunCyclePro

GeoSunPaletteSO

GeoSunPreset

GeoSunSharedPalette

InteriorAudioZone

KeysEvent

MouseLockToggle

MouseLookCamera

NPCNavMeshGT

ObjectMoverRotator

PunchySkyVolumePreset

QuitGameManager

rotateObjects

RuntimeMinimapGenerator

SkyboxApplier

StartupSplash

SunClockHUDGeo

SurfaceDataGT

SurfaceTypeGT

TimerEventManager

UIGameObject

UITooltip

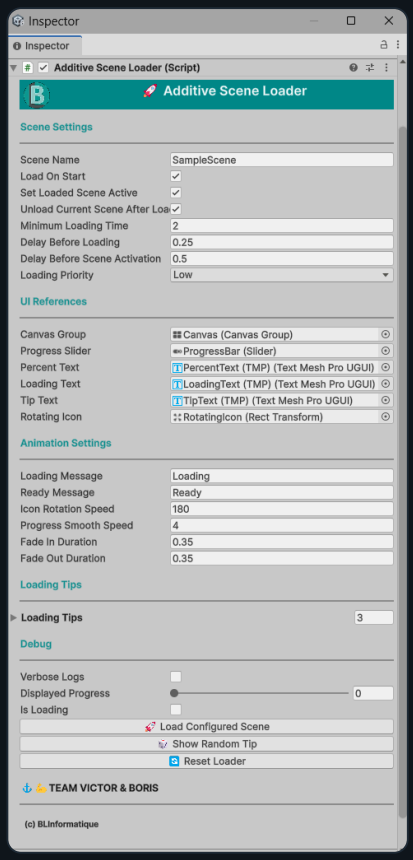
WaypointGT

📦 Components

Tooltips are in **English**. Missing tooltips can be completed automatically from field names. Tables list documented fields. Optional notes (Markdown) and images are shown below each component when present.

AdditiveSceneLoader MonoBehaviour

BLInformatique.GameTools



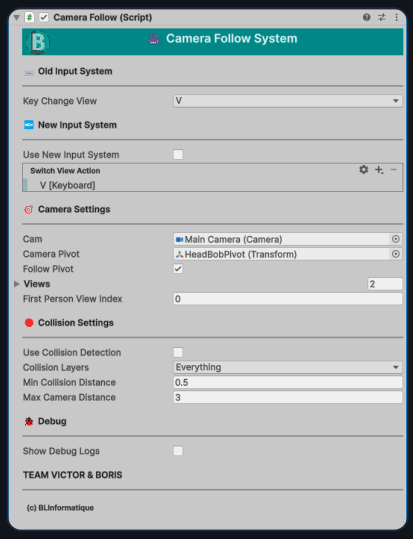
Field	Type	Tooltip	Hints
↓			
sceneName	`String`	Name of the scene to load additively. The scene must be added to Build Settings.	—
loadOnStart	`Boolean`	Start loading automatically when this component starts.	—
setLoadedSceneActive	`Boolean`	Set the loaded scene as the active scene after loading.	—
unloadCurrentSceneAfterLoad	`Boolean`	Unload the current loader scene after the target scene has been activated.	—
minimumLoadingTime	`Single`	Minimum loading screen duration in seconds.	—
delayBeforeLoading	`Single`	Delay before starting the async loading. Useful to let the loading UI appear first.	—
delayBeforeSceneActivation	`Single`	Delay before activating the loaded scene once loading is ready.	—
loadingPriority	`ThreadPriority`	Background loading priority used while loading the scene.	Enum
↓			
canvasGroup	`CanvasGroup`	CanvasGroup used to fade the loading screen.	Object Reference
progressSlider	`Slider`	Slider used to display loading progress.	Object Reference
percentText	`TMP_Text`	Text used to display loading percentage.	Object Reference
loadingText	`TMP_Text`	Text used to display the animated loading message.	Object Reference
tipText	`TMP_Text`	Text used to display random loading tips.	Object Reference
rotatingIcon	`RectTransform`	Optional rotating icon displayed during loading.	Object Reference
↓			
loadingMessage	`String`	Main loading message displayed before animated dots.	—
readyMessage	`String`	Message displayed when the target scene is ready.	—

Field	Type	Tooltip	Hints
iconRotationSpeed	`Single`	Rotation speed of the loading icon.	—
progressSmoothSpeed	`Single`	Speed used to smooth the displayed progress.	—
fadeInDuration	`Single`	Fade-in duration of the loading screen.	—
fadeOutDuration	`Single`	Fade-out duration after the loaded scene is ready.	—
↓			
loadingTips	`String[]`	Random tips displayed while the scene is loading. • TextArea	List / Array TextArea
↓			
verboseLogs	`Boolean`	Show debug logs in the console.	—
displayedProgress	`Single`	Current displayed loading progress. • Range [0..1]	Range
isLoading	`Boolean`	True while a scene is currently loading.	—
headerDummy1	`String`	Controls the manual header display.	—
headerDummy2	`String`	Controls the manual header display.	—
headerDummy3	`String`	Controls the manual header display.	—

↑ Back to top

CameraFollow MonoBehaviour

BLInformatique.GameTools

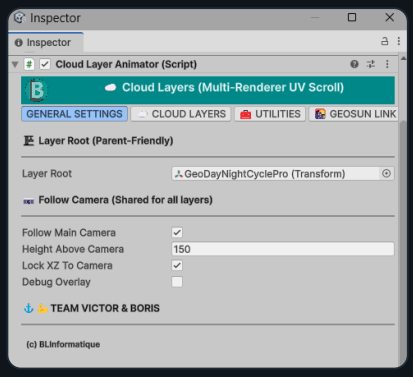


Field	Type	Tooltip	Hints
↓			
keyChangeView	`KeyCode`	Key used to switch camera view (Old Input System only).	Enum
↓			
useNewInputSystem	`Boolean`	Enable if you're using the new Input System.	—
switchViewAction	`InputAction`	Input Action used to switch views (New Input System).	—
↓			
cam	`Camera`	Main camera to follow the target.	Object Reference
cameraPivot	`Transform`	Transform that the camera should follow.	Object Reference
followPivot	`Boolean`	Enable to follow the pivot (position only).	—
views	`Transform[]`	List of view targets to cycle through.	List / Array
firstPersonViewIndex	`Int32`	Index of the first-person view in the views array.	—
↓			
useCollisionDetection	`Boolean`	Enable camera collision detection (useful in third-person).	—
collisionLayers	`LayerMask`	Layer mask used to detect camera obstacles.	—
minCollisionDistance	`Single`	Minimum distance allowed between the pivot and camera.	—
maxCameraDistance	`Single`	Maximum allowed camera distance from the pivot.	—
↓			
showDebugLogs	`Boolean`	Enable to log debug info to the console.	—

↑ Back to top

CloudLayerAnimator MonoBehaviour

BLInformatique.GameTools



Field	Type	Tooltip	Hints
↓			
layerRoot	`Transform`	Transform to move/rotate/scale for the whole cloud rig. Leave empty to use this GameObject.	Object Reference
↓			
followMainCamera	`Boolean`	Keep the whole cloud rig centered above the main camera (infinite horizon feel).	—
heightAboveCamera	`Single`	Height above camera for the cloud rig (world units).	—
lockXZToCamera	`Boolean`	Lock XZ to camera so the rig looks infinite.	—
↓			
layers	`List<Layer>`	List of cloud layers. Each entry controls one child renderer.	List / Array
↓			
orientLayersAsFlatDown	`Boolean`	When ON, per-layer snap will also reorient each layer to look downward (flat quads).	—
enforceFlatSizeIfTiny	`Single`	If the layer's X/Z scale is small, per-layer snap will enforce this target flat size (X/Z). 0 = disabled.	—
__btnSnapPerLayer	`Int32`		—
↓			
scanRootOverride	`Transform`	When scanning, only include Renderers found under this Transform. Leave empty to use layerRoot or this GameObject.	Object Reference
scanIncludeDisabled	`Boolean`	Include disabled Renderers when scanning children.	—
replaceListOnScan	`Boolean`	Replace the current list with the scan result (otherwise append).	—
__btnScan	`Int32`		—
__btnSnap	`Int32`		—

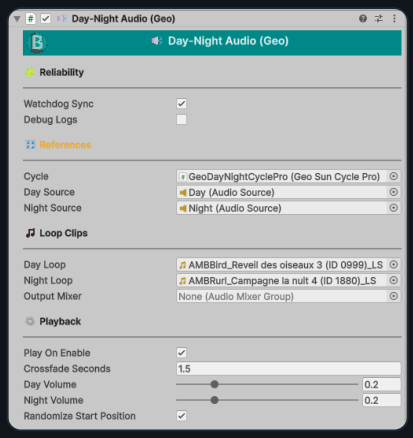
Field	Type	Tooltip	Hints
↓			
geoSun	`GeoSunCyclePro`	Optional reference to GeoSunCyclePro. When set, clouds can react to solar elevation (sunrise/sunset look).	Object Reference
enableGeoLink	`Boolean`	Enable linking to GeoSun: sample sun elevation and modulate clouds accordingly.	—
invertElevation	`Boolean`	If ON, uses (1 - elevation01) so that '0=noon, 1=night' — handy if your curve/gradient was authored inverted.	—
opacityByElevation	`AnimationCurve`	Opacity response versus elevation01 (0=night, 1=noon unless inverted). This scales each layer's Global Opacity.	—
tintByElevation	`Gradient`	Color tint versus elevation01 (0=night, 1=noon unless inverted).	—
tintGlobalInfluence	`Single`	How strongly the tint overrides the material color (0=no tint, 1=full tint). • Range [0..1]	Range
tintPerLayerInfluence	`Single`	Range [0..1]	Range
minOpacityClamp	`Single`	Optional min/max clamp after applying curve and layer opacity. • Range [0..1]	Range
maxOpacityClamp	`Single`	Range [0..1]	Range
opacitySmoothTime	`Single`	Smooth time (seconds) for opacity changes (0 = instant).	—
↓			
fogPreset	`PunchySkyVolumePreset`	Optional PunchySkyVolumePreset to drive fog color and density from the sun elevation.	Object Reference
driveFog	`Boolean`	If ON, this component will drive fog color/density each frame. Turn OFF to let weather/GeoSun manage fog.	—

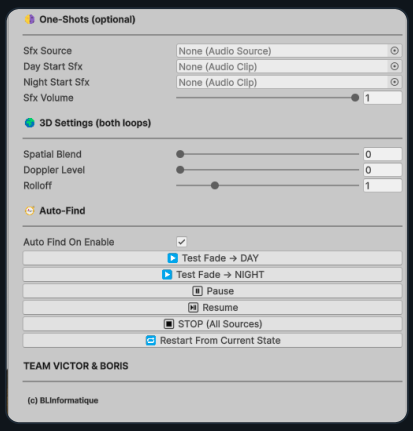
Field	Type	Tooltip	Hints
builtInFogFallback	`Boolean`	If ON, also writes to Built-in RenderSettings fog. If OFF, Drive Fog only targets the Volume preset (when assigned).	—
respectInitialFogState	`Boolean`	If ON, never enables built-in Fog if it was OFF at scene start.	—
forceEnableFogWhenDriving	`Boolean`	When ON and driveFog is enabled, ensures RenderSettings.fog is forced ON before writing color/density (Built-In only).	—
forceBuiltInFogMode	`FogMode`	Force a Built-In fog mode to keep density/color predictable. Exponential recommended.	Enum
debugFogWrites	`Boolean`	Print a log once per second with elevation, target fog color and current RenderSettings.fogColor (helps find who overwrites).	—
fogDensityByElevation	`AnimationCurve`	Fog density vs sun elevation (0 = night, 1 = noon). Built-in uses RenderSettings.fogDensity; URP uses Volume Fog density.	—
fogColorByElevation	`Gradient`	Fog color vs sun elevation (0 = night, 1 = noon).	—
fogSmoothTime	`Single`	Smooth time for fog transitions (seconds). 0 = instant.	—
debugOverlay	`Boolean`	Show a tiny debug overlay with elevation01 and tint info (Play only).	—

↑ Back to top

DayNightAudioGeo MonoBehaviour

BLInformatique.GameTools





Field	Type	Tooltip	Hints
_header	`String`	Controls the manual header display.	—
↓			
watchdogSync	`Boolean`	If ON, also polls GeoSunCyclePro each frame and crossfades if a day/night event was missed.	—
debugLogs	`Boolean`	Write simple logs when binding to a GeoSunCyclePro and on state changes.	—
↓			
cycle	`GeoSunCyclePro`	GeoSunCyclePro to listen for day/night events.	Object Reference
daySource	`AudioSource`	AudioSource used for DAY looping ambience. Created at runtime if missing.	Object Reference
nightSource	`AudioSource`	AudioSource used for NIGHT looping ambience. Created at runtime if missing.	Object Reference
↓			
dayLoop	`AudioClip`	Loop to play during the DAY state (will be looped).	Object Reference
nightLoop	`AudioClip`	Loop to play during the NIGHT state (will be looped).	Object Reference
outputMixer	`AudioMixerGroup`	Optional AudioMixerGroup for both sources (keeps routing tidy).	Object Reference
↓			
playOnEnable	`Boolean`	Start playing on Enable, matching the current day/night state.	—
crossfadeSeconds	`Single`	Crossfade duration in seconds between day and night loops.	—
dayVolume	`Single`	Target volume for DAY loop (post-fade). • Range [0..1]	Range
nightVolume	`Single`	Target volume for NIGHT loop (post-fade). • Range [0..1]	Range
randomizeStartPosition	`Boolean`	Randomize starting playback position when starting a loop (avoids obvious seam).	—
↓			

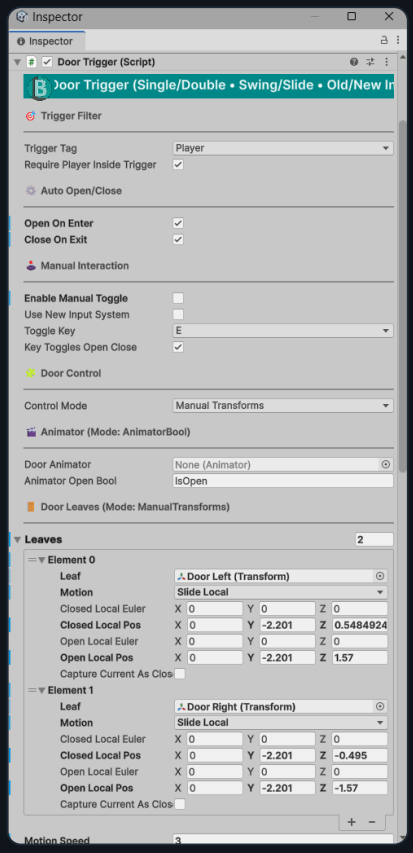
Field	Type	Tooltip	Hints
sfxSource	`AudioSource`	Optional AudioSource dedicated to one-shot SFX on transitions.	Object Reference
dayStartSfx	`AudioClip`	Played once when day begins (e.g., rooster, birds burst).	Object Reference
nightStartSfx	`AudioClip`	Played once when night begins (e.g., owl, cricket chirp).	Object Reference
sfxVolume	`Single`	Volume multiplier for one-shot SFX. • Range [0..1]	Range
↓			
spatialBlend	`Single`	Set >0 to make loops spatialized; keep 0 for UI/global ambience. • Range [0..1]	Range
dopplerLevel	`Single`	Doppler effect amount for spatialized loops. • Range [0..5]	Range
rolloff	`Single`	How fast volume drops with distance (3D only). • Range [0..5]	Range
↓			
autoFindOnEnable	`Boolean`	Auto-find cycle and/or create sources on Enable if missing.	—
btnTestDay	`Int32`		—
btnTestNight	`Int32`		—
btnPause	`Int32`		—
btnResume	`Int32`		—
btnStop	`Int32`		—
btnRestart	`Int32`		—

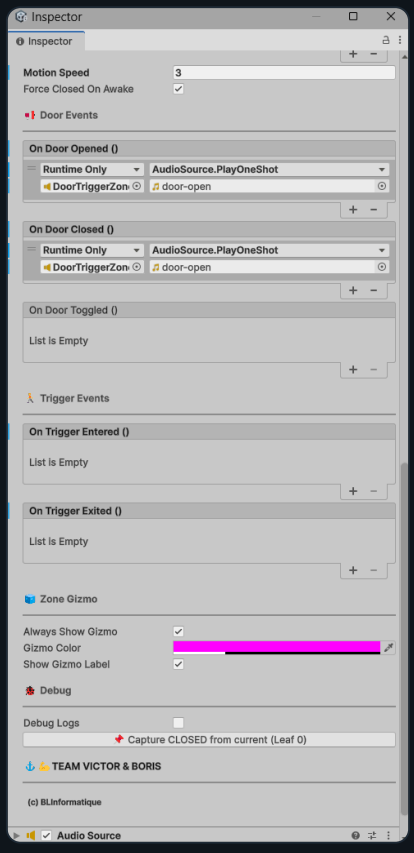
↑ Back to top

DoorTrigger MonoBehaviour

BLInformatique.GameTools

Requires: BoxCollider, AudioSource





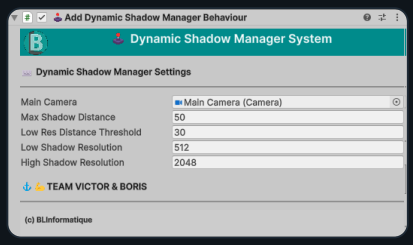
Field	Type	Tooltip	Hints
↓			
triggerTag	`String`	Tag required to interact with this door trigger.	—
requirePlayerInsideTrigger	`Boolean`	If true, interaction is allowed only while the player is inside the trigger.	—
↓			
openOnEnter	`Boolean`	If true, the door opens automatically when entering the trigger.	—
closeOnExit	`Boolean`	If true, the door closes automatically when exiting the trigger.	—
↓			
enableManualToggle	`Boolean`	If true, allows manual toggle with a key/action.	—
useNewInputSystem	`Boolean`	Use Unity New Input System (InputAction) instead of legacy Input.	—
toggleKey	`KeyCode`	Key used to toggle the door (Old Input System).	Enum
toggleAction	`InputAction`	InputAction used to toggle the door (New Input System). Suggested binding: <Keyboard>/e or gamepad button.	—
keyTogglesOpenClose	`Boolean`	If true, pressing the key will toggle Open/Close. If false, the key will only OPEN (and you close via exit/other events).	—
↓			
controlMode	`DoorControlMode`	How the door is controlled. - AnimatorBool: uses an Animator bool parameter. - ManualTransforms: moves leaves with rotation/slide.	Enum
↓			
doorAnimator	`Animator`	Animator controlling the door (can drive single or double doors in one animation).	Object Reference
animatorOpenBool	`String`	Animator bool parameter used to open/close the door.	—

Field	Type	Tooltip	Hints
↓			
leaves	`DoorLeaf[]`	List of door leaves/panels. For double doors, add 2 leaves. Each leaf can be RotateLocal or SlideLocal.	List / Array
motionSpeed	`Single`	Motion smoothing speed (higher = faster). Applies to both rotation and sliding.	—
forceClosedOnAwake	`Boolean`	If true, the CLOSED pose is applied on Awake (useful to reset scene state consistently).	—
↓			
onDoorOpened	`UnityEvent`	Event fired when the door opens.	UnityEvent
onDoorClosed	`UnityEvent`	Event fired when the door closes.	UnityEvent
onDoorToggled	`UnityEvent`	Event fired when the door toggles (always invoked after open/close).	UnityEvent
↓			
onTriggerEntered	`UnityEvent`	Event fired when a valid object enters the trigger (use this for UI prompt, sounds, etc.).	UnityEvent
onTriggerExited	`UnityEvent`	Event fired when a valid object exits the trigger (use this to hide UI prompt, etc.).	UnityEvent
↓			
alwaysShowGizmo	`Boolean`	If true, the trigger gizmo is always visible in the Scene view.	—
gizmoColor	`Color`	Color of the trigger gizmo.	—
showGizmoLabel	`Boolean`	If true, a label is displayed in the Scene view for this zone (Editor only).	—
↓			
debugLogs	`Boolean`	Enable console logs for debugging.	—
__btnCaptureLeaf0	`Int32`		—

↑ Back to top

DynamicShadowManager MonoBehaviour

BLInformatique.GameTools



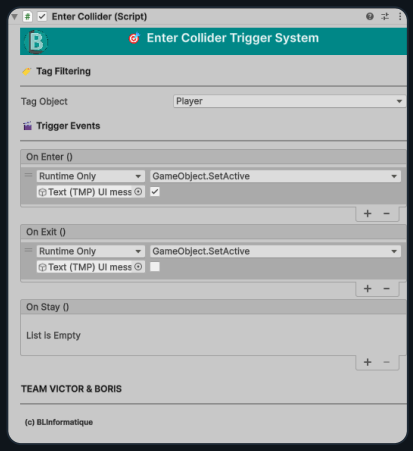
Field	Type	Tooltip	Hints
⬇			
mainCamera	`Camera`	Reference to the main camera used to calculate distances	Object Reference
maxShadowDistance	`Single`	Maximum distance at which lights cast shadows	—
lowResDistanceThreshold	`Single`	Distance threshold to reduce shadow resolution	—
lowShadowResolution	`Int32`	Custom shadow resolution for distant lights (lower value = lower quality)	—
highShadowResolution	`Int32`	Custom shadow resolution for close lights (higher value = better quality)	—

↑ Back to top

EnterCollider MonoBehaviour

BLInformatique.GameTools

Requires: BoxCollider

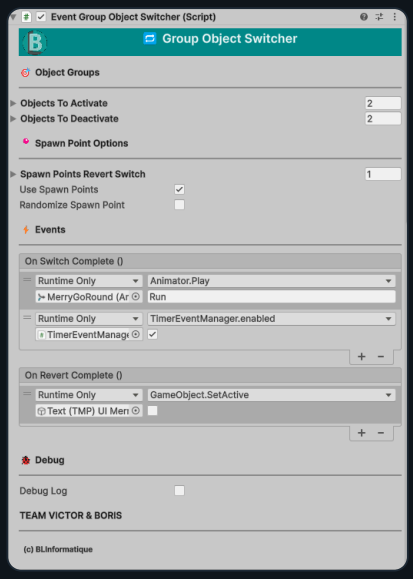


Field	Type	Tooltip	Hints
⬇			
tagObject	`String`	Tag of the object that will trigger the event.	—
⬇			
OnEnter	`UnityEvent`	Event triggered when an object with the correct tag enters the collider.	UnityEvent
OnExit	`UnityEvent`	Event triggered when an object with the correct tag exits the collider.	UnityEvent
OnStay	`UnityEvent`	Event triggered every frame while an object with the correct tag stays in the collider.	UnityEvent
⬇			
alwaysShowGizmo	`Boolean`	If true, the interior zone gizmo is always visible in the Scene view.	—
gizmoColor	`Color`	Color of the interior zone gizmo.	—
showGizmoLabel	`Boolean`	If true, a label is displayed in the Scene view for this zone (Editor only).	—

↑ Back to top

EventGroupObjectSwitcher MonoBehaviour

BLInformatique.GameTools



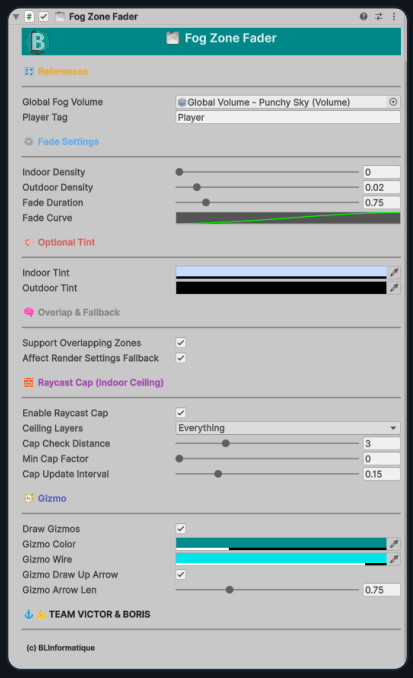
Field	Type	Tooltip	Hints
↓			
objectsToActivate	`List<GameObject>`	Objects to activate during switch.	List / Array
objectsToDeactivate	`List<GameObject>`	Objects to deactivate during switch.	List / Array
↓			
useSpawnPoints	`Boolean`	Use spawn point for revert operation.	—
randomizeSpawnPoint	`Boolean`	Choose a random spawn point when reverting.	—
spawnPointsRevertSwitch	`Transform[]`	Optional spawn points to reposition the reverted object.	List / Array
↓			
onSwitchComplete	`UnityEvent`	Event triggered after the switch is completed.	UnityEvent
onRevertComplete	`UnityEvent`	Event triggered after revert is completed.	UnityEvent
↓			
debugLog	`Boolean`	Enable to print debug logs in the console.	—

↑ Back to top

FogZoneFader MonoBehaviour

BLInformatique.GameTools

Requires: BoxCollider



Field	Type	Tooltip	Hints
header	`String`	Controls the manual header display.	—
↓			
globalFogVolume	`Object`	URP: assign a Global Volume that contains a Fog override. Leave empty in Built-in (RenderSettings fallback).	Object Reference
playerTag	`String`	Tag of the player collider that triggers the zone.	—
↓			
indoorDensity	`Single`	Fog density inside the zone (0 = clear). • Range [0..0,2]	Range
outdoorDensity	`Single`	Outdoor fog density (if 0 at start, we snapshot current RenderSettings.fogDensity). • Range [0..0,2]	Range
fadeDuration	`Single`	Fade duration in seconds. • Range [0..5]	Range
fadeCurve	`AnimationCurve`	Fade curve shape.	—
↓			
indoorTint	`Color`	Indoor fog color (alpha=0 to ignore color lerp).	—
outdoorTint	`Color`	Outdoor fog color (alpha=0 to ignore color lerp).	—
↓			
autoDisableGlobalFogDrivers	`Boolean`	If ON, pause global fog drivers while inside (GeoSun/Clouds/Punchy).	—
debugLogs	`Boolean`	Enable verbose debug logs.	—
__btnForceIndoorTest	`Int32`		—
__btnForceOutdoorTest1	`Int32`		—
↓			
alwaysShowGizmo	`Boolean`	If true, the interior zone gizmo is always visible in the Scene view.	—
gizmoColor	`Color`	Color of the interior zone gizmo.	—

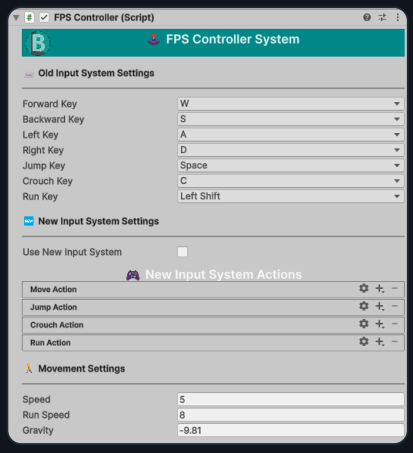
Field	Type	Tooltip	Hints
showGizmoLabel	`Boolean`	If true, a label is displayed in the Scene view for this zone (Editor only).	—

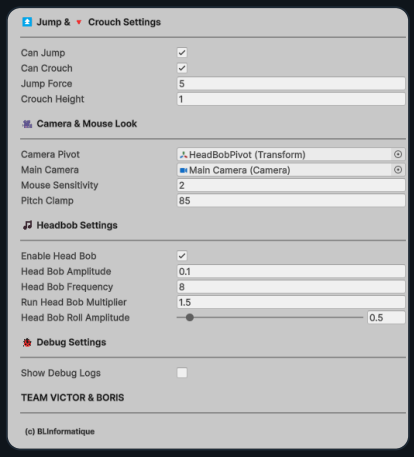
↑ Back to top

FPSController MonoBehaviour

BLInformatique.GameTools

Requires: CharacterController





Field	Type	Tooltip	Hints
⬇			
forwardKey	`KeyCode`	⬆️ Key for moving forward (Old Input System).	Enum
backwardKey	`KeyCode`	⬇️ Key for moving backward (Old Input System).	Enum
leftKey	`KeyCode`	⬅️ Key for moving left (Old Input System).	Enum
rightKey	`KeyCode`	➡️ Key for moving right (Old Input System).	Enum
jumpKey	`KeyCode`	⏫ Key for jumping (Old Input System).	Enum
crouchKey	`KeyCode`	🔻 Key for crouching (Old Input System).	Enum
runKey	`KeyCode`	🏃 Key for running (Old Input System).	Enum
⬇			
useNewInputSystem	`Boolean`	Enable this if using the new Input System.	—
⬇ 🎮 New Input System Actions			
moveAction	`InputAction`	🕹️ Input action for movement (New Input System).	—
jumpAction	`InputAction`	⏫ Input action for jumping (New Input System).	—
crouchAction	`InputAction`	🔻 Input action for crouching (New Input System).	—
runAction	`InputAction`	🏃 Input action for running (New Input System).	—
⬇			
speed	`Single`	Walking speed of the player.	—
runSpeed	`Single`	Running speed when holding the run key.	—
gravity	`Single`	Gravity applied to the character.	—
⬇			
canJump	`Boolean`	Enable or disable jumping.	—
jumpForce	`Single`	Force applied when jumping.	—
canCrouch	`Boolean`	Enable or disable crouching.	—
crouchHeight	`Single`	Height of the character when crouching.	—

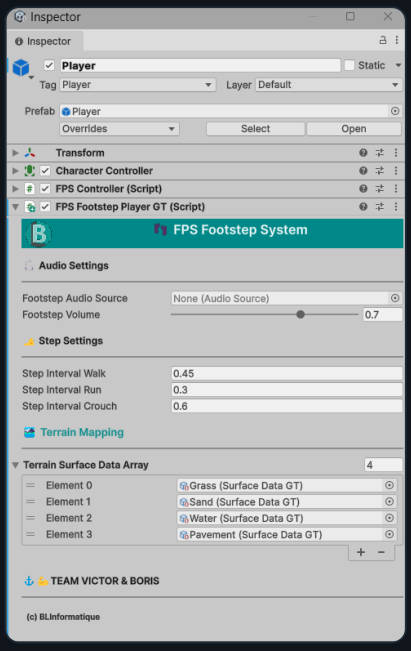
Field	Type	Tooltip	Hints
↓			
cameraPivot	`Transform`	Reference to the camera pivot (for pitch/headbob/roll).	Object Reference
mainCamera	`Camera`	Camera component (child of cameraPivot).	Object Reference
mouseSensitivity	`Single`	Mouse look sensitivity.	—
pitchClamp	`Single`	Clamp look up/down (pitch).	—
↓			
enableHeadBob	`Boolean`	Enable camera headbob while walking/running.	—
headBobAmplitude	`Single`	Amplitude of the headbob movement.	—
headBobFrequency	`Single`	Frequency of the headbob movement.	—
runHeadBobMultiplier	`Single`	Headbob multiplier when running.	—
headBobRollAmplitude	`Single`	Maximum roll Z (degrees). • Range [0..10]	Range
↓			
showDebugLogs	`Boolean`	Enable debug logs in the console.	—
isCrouching	`Boolean`		—
isRunning	`Boolean`		—
moveInput	`Vector2`		—

↑ Back to top

FPSFootstepPlayerGT MonoBehaviour

BLInformatique.GameTools

Requires: CharacterController, FPSController



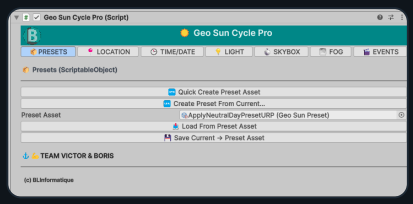
Field	Type	Tooltip	Hints
⬇			
footstepAudioSource	`AudioSource`	AudioSource to use for footsteps (optional, can use PlayClipAtPoint if null).	Object Reference
footstepVolume	`Single`	Volume for footsteps. • Range [0..1]	Range
⬇			
stepIntervalWalk	`Single`	Interval between steps when walking.	—
stepIntervalRun	`Single`	Interval between steps when running.	—
stepIntervalCrouch	`Single`	Interval between steps when crouched.	—
⬇			
terrainSurfaceDataArray	`SurfaceDataGT[]`	SurfaceData array matching your terrain textures order. One SurfaceData per terrain texture slot!	List / Array

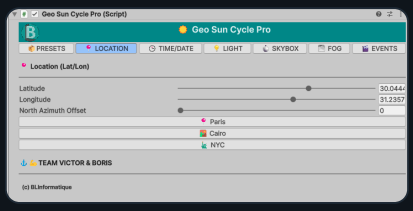
↑ Back to top

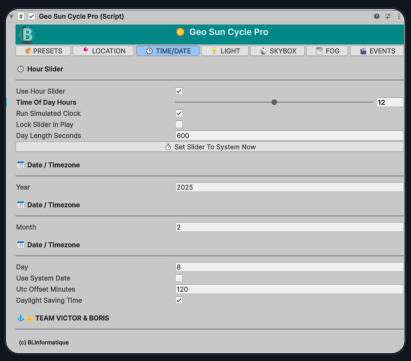
GeoSunCyclePro

MonoBehaviour

BLInformatique.GameTools







Field	Type	Tooltip	Hints
↓			
btnQuickCreatePreset	`Int32`	Editor only: quickly create a preset asset from current values.	—
btnCreatePreset	`Int32`	Editor only: create a new preset asset and save current settings into it.	—
presetAsset	`GeoSunPreset`	Optional ScriptableObject preset to load from or save to.	Object Reference
btnLoadPresetAsset	`Int32`	Editor only: load values from the assigned preset asset.	—
btnSavePresetAsset	`Int32`	Editor only: save current values into the assigned preset asset.	—
↓			
latitude	`Single`	Observer latitude in degrees (-90 to 90). • Range [-90..90]	Range
longitude	`Single`	Observer longitude in degrees (-180 to 180). • Range [-180..180]	Range
northAzimuthOffset	`Single`	Additional azimuth offset (degrees) to align 'north' with your scene. • Range [0..359,999]	Range
btnParis	`Int32`	Editor only: set coordinates to Paris.	—
btnCairo	`Int32`	Editor only: set coordinates to Cairo.	—
btnNYC	`Int32`	Editor only: set coordinates to New York City.	—
↓			
useHourSlider	`Boolean`	Enable a manual hour slider to control simulated time.	—
timeOfDayHours	`Single`	Current time of day in hours (0 - 24). Used by slider/simulation. • Range [0..24]	Range
runSimulatedClock	`Boolean`	When true, the clock runs automatically according to day length.	—
lockSliderInPlay	`Boolean`	When true, lock the hour slider while the scene is playing.	—
dayLengthSeconds	`Single`	Length of a simulated day in real seconds (min 0.01).	—

Field	Type	Tooltip	Hints
btnSetNow	`Int32`	Editor only: set the slider to the current system time.	—
↓			
year	`Int32`	Date used for solar calculations (year, month, day).	—
↓			
month	`Int32`	Date used for solar calculations (year, month, day).	—
↓			
day	`Int32`	Date used for solar calculations (year, month, day).	—
useSystemDate	`Boolean`	When true, use the system date instead of the year/month/day fields.	—
utcOffsetMinutes	`Int32`	Local UTC offset in minutes (positive east of UTC).	—
daylightSavingTime	`Boolean`	Apply daylight saving time (+60 minutes) when computing UTC.	—
↓			
sunLight	`Light`	Directional Light used as the sun source (optional).	Object Reference
sunPivotOverride	`Transform`	Optional transform to rotate instead of the light's transform.	Object Reference
btnApplyLightingEnv	`Int32`	Editor only: push this component's sun and skybox to Lighting > Environment.	—
↓			
intensityByElevation	`AnimationCurve`	Intensity curve evaluated by normalized solar elevation (0=midnight,0.5=horizon,1=noon).	—
maxSunIntensity	`Single`	Maximum multiplier applied to the evaluated sun intensity.	—
colorByElevation	`Gradient`	Sun color gradient by elevation (0=midnight,0.5=horizon,1=noon).	—

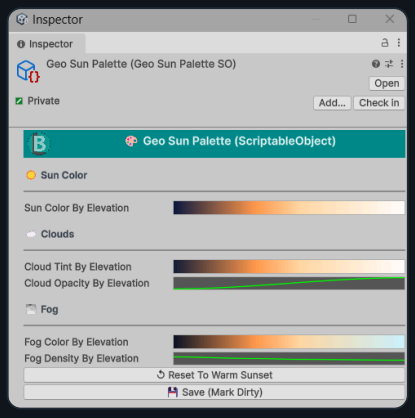
Field	Type	Tooltip	Hints
ambientByElevation	`AnimationCurve`	Ambient intensity curve evaluated by normalized solar elevation.	—
↓			
reflectionIntensity	`Single`	Reflection probe intensity multiplier applied to RenderSettings. • Range [0..2]	Range
↓			
invertLightResponse	`Boolean`	Invert evaluation of light/ambient/color curves (1→0 mapping).	—
btnResetLightCurves	`Int32`	Editor only: reset intensity, ambient and color curves to defaults.	—
↓			
skyboxMaterial	`Material`	Optional skybox material to modify yaw and/or assign to scene.	Object Reference
driveSkyboxYaw	`Boolean`	Write computed yaw to the skybox material when enabled.	—
skyboxYawProperty	`String`	Float property name on the skybox material to set for rotation (e.g. _Rotation).	—
skyboxYawOffset	`Single`	Extra yaw offset (degrees) added to computed skybox rotation.	—
↓			
manageFog	`Boolean`	When true, GeoSunCyclePro will control fog state in the scene.	—
fogState	`FogState`	Desired fog state when managing fog (Unchanged / ForceOff / ForceOn).	Enum
globalFogVolume	`Object`	Optional specific Volume (or component) to target when toggling fog.	Object Reference
btnKillAllFog	`Int32`	Editor only: disable all fog across scene Volumes.	—
↓			
daylightElevationThreshold	`Single`	Normalized elevation threshold (0-1) above which 'day' is considered	Range

Field	Type	Tooltip	Hints
		active. • Range [0..1]	
daylightHysteresis	`Single`	Hysteresis applied to the day/night threshold to avoid rapid toggles. • Range [0..0,5]	Range
minTransitionGapSeconds	`Single`	Minimum realtime seconds between day/night transitions (debounce).	—
emitHourEvents	`Boolean`	Emit an event when the visible hour changes.	—
emitMinuteEvents	`Boolean`	Emit events when the visible minute changes.	—
use24hClock	`Boolean`	When true, formatted time will use 24-hour format.	—
↓			
invertDayNightEvents	`Boolean`	Invert day/night events (swap day and night signals).	—
↓			
fireEventsOnStart	`Boolean`	Fire day/night and clock events when component is enabled.	—
onDayBegan	`UnityEvent`	Invoked when a day begins according to the configured threshold.	UnityEvent
onNightBegan	`UnityEvent`	Invoked when a night begins according to the configured threshold.	UnityEvent
onHourChanged	`UnityEvent<Int32>`	Invoked when the visible hour changes. Parameter: new hour (0-23).	UnityEvent
onMinuteChanged	`UnityEvent<Int32, Int32>`	Invoked when the visible minute changes. Parameters: hour, minute.	UnityEvent
onFormattedTime	`UnityEvent<String>`	Invoked with a formatted time string when the minute event fires.	UnityEvent

↑ Back to top

GeoSunPaletteSO ScriptableObject

BLInformatique.GameTools

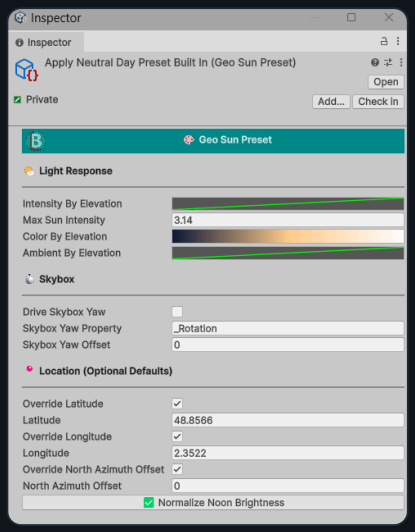


Field	Type	Tooltip	Hints
⬇			
sunColorByElevation	`Gradient`	Sun light color by normalized elevation (0 = night, 0.35 = dawn/dusk, 0.55 = morning, 1 = noon).	—
⬇			
cloudTintByElevation	`Gradient`	Cloud tint by normalized elevation.	—
cloudOpacityByElevation	`AnimationCurve`	Cloud opacity by normalized elevation.	—
⬇			
fogColorByElevation	`Gradient`	Fog color by normalized elevation.	—
fogDensityByElevation	`AnimationCurve`	Fog density by normalized elevation.	—
__btnResetWarm	`Int32`		—
__btnSave	`Int32`		—

↑ Back to top

GeoSunPreset ScriptableObject

BLInformatique.GameTools

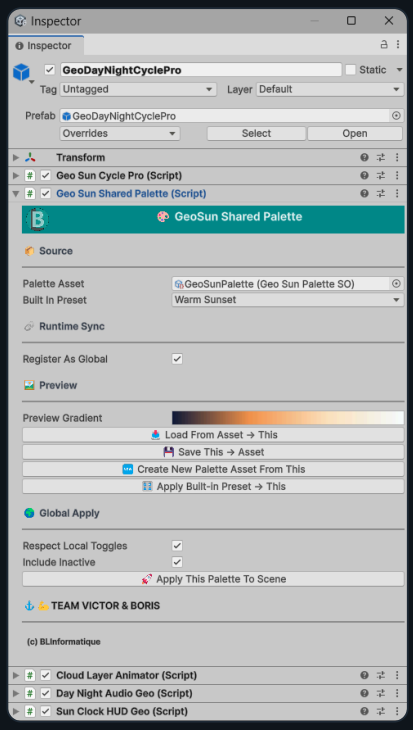


Field	Type	Tooltip	Hints
header	`String`	Controls the manual header display.	—
↓			
intensityByElevation	`AnimationCurve`	Intensity by elevation (0 = midnight, 0.5 = horizon, 1 = noon). Noon should be the brightest.	—
maxSunIntensity	`Single`	Maximum sun intensity multiplier.	—
colorByElevation	`Gradient`	Color by elevation (0 = midnight, 0.5 = horizon, 1 = noon).	—
ambientByElevation	`AnimationCurve`	Ambient intensity by elevation.	—
↓			
driveSkyboxYaw	`Boolean`	If ON, write a yaw to skybox material based on sun azimuth + offset.	—
skyboxYawProperty	`String`	Yaw property name on the skybox material (e.g., _Rotation or _SkyboxRotation).	—
skyboxYawOffset	`Single`	Extra yaw offset added to skybox rotation.	—
↓			
overrideLatitude	`Boolean`	Optional default latitude (leave empty to not override).	—
latitude	`Single`		—
overrideLongitude	`Boolean`	Optional default longitude (leave empty to not override).	—
longitude	`Single`		—
overrideNorthAzimuthOffset	`Boolean`	Optional default north azimuth offset (leave empty to not override).	—
northAzimuthOffset	`Single`		—
normalizeButton	`Int32`		—

↑ Back to top

GeoSunSharedPalette MonoBehaviour

BLInformatique.GameTools



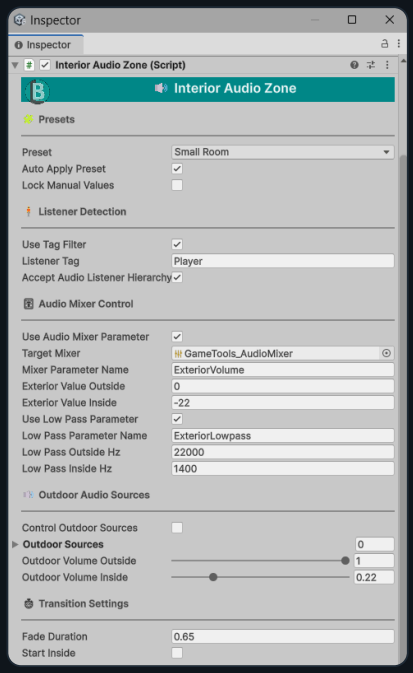
Field	Type	Tooltip	Hints
↓			
paletteAsset	`GeoSunPaletteSO`	If assigned, values come from this ScriptableObject palette.	Object Reference
builtInPreset	`BuiltInPreset`	If no asset is assigned, use this built-in preset.	Enum
↓			
registerAsGlobal	`Boolean`	If ON, this instance becomes the global source for static getters used by other systems.	—
↓			
previewGradient	`Gradient`	Composite gradient showing the current Sun/Cloud/Fog palette. (visual only)	—
__btnLoad	`Int32`		—
__btnSave	`Int32`		—
__btnCreate	`Int32`		—
__btnApplyPreset	`Int32`		—
↓			
respectLocalToggles	`Boolean`	If ON, only components with 'Use Shared Palette' toggles enabled will be updated.	—
includeInactive	`Boolean`	Also include inactive GameObjects in the search.	—
__btnApplyPaletteToScene	`Int32`		—

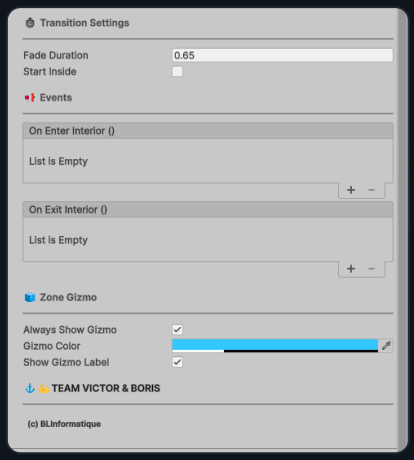
↑ Back to top

InteriorAudioZone MonoBehaviour

BLInformatique.GameTools

Requires: Collider





Field	Type	Tooltip	Hints
header	`String`	Header placeholder for BLInformatique styled box in the inspector.	—
↓			
preset	`InteriorPreset`	Select a preset that configures indoor feel (volume attenuation + optional low-pass + fade).	Enum
autoApplyPreset	`Boolean`	If true, changing the preset will automatically apply its values in the inspector (Editor only).	—
lockManualValues	`Boolean`	If true, keeps your manual values even if you select a preset (preset will not overwrite).	—
↓			
useTagFilter	`Boolean`	If true, only GameObjects with this tag will trigger the interior effect (usually the Player).	—
listenerTag	`String`	Tag used to detect the player/listener entering the interior zone.	—
acceptAudioListenerHierarchy	`Boolean`	If true, a Collider that has an AudioListener anywhere in its hierarchy will also be accepted as listener.	—
↓			
useAudioMixerParameter	`Boolean`	If true, this zone will drive an exposed parameter on an AudioMixer to lower outside sound.	—
targetMixer	`AudioMixer`	Target AudioMixer that contains the exposed parameter controlling outside sound volume.	Object Reference
mixerParameterName	`String`	Name of the exposed float parameter in the AudioMixer (in dB, typically from -80 to 0).	—
exteriorValueOutside	`Single`	Mixer parameter value (dB) when the listener is OUTSIDE all interior zones.	—
exteriorValueInside	`Single`	Mixer parameter value (dB) when the listener is INSIDE this interior zone.	—

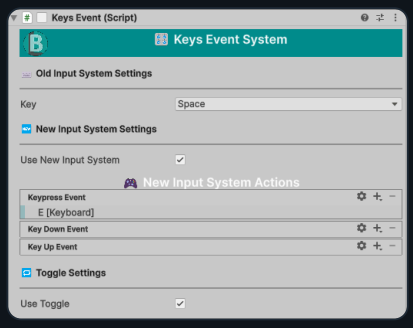
Field	Type	Tooltip	Hints
useLowPassParameter	`Boolean`	If true, this zone will also drive an exposed LowPass Cutoff parameter (more realistic 'walls muffling').	—
lowPassParameterName	`String`	Name of the exposed LowPass cutoff parameter in the AudioMixer (Hz). Example: ExteriorLowpass.	—
lowPassOutsideHz	`Single`	LowPass cutoff value (Hz) when OUTSIDE (clear sound). Typical: 22000.	—
lowPassInsideHz	`Single`	LowPass cutoff value (Hz) when INSIDE (muffled sound). Typical: 800-2500 depending on room).	—
↓			
controlOutdoorSources	`Boolean`	If true, this zone will also scale the volume of these AudioSources when inside / outside.	—
outdoorSources	`List<AudioSource>`	List of AudioSources that represent outdoor sounds (traffic, wind, etc.). They will be attenuated when inside.	List / Array
outdoorVolumeOutside	`Single`	Volume for outdoor sources when OUTSIDE all interior zones. • Range [0..1]	Range
outdoorVolumeInside	`Single`	Volume for outdoor sources when INSIDE this interior zone. • Range [0..1]	Range
↓			
fadeDuration	`Single`	Duration of fade between inside and outside values (seconds).	—
startInside	`Boolean`	If true, the listener is considered inside at Start (useful if player spawns in this interior).	—
↓			
onEnterInterior	`UnityEvent`	Event invoked when the listener enters this interior zone.	UnityEvent
onExitInterior	`UnityEvent`	Event invoked when the listener exits this interior zone.	UnityEvent
↓			

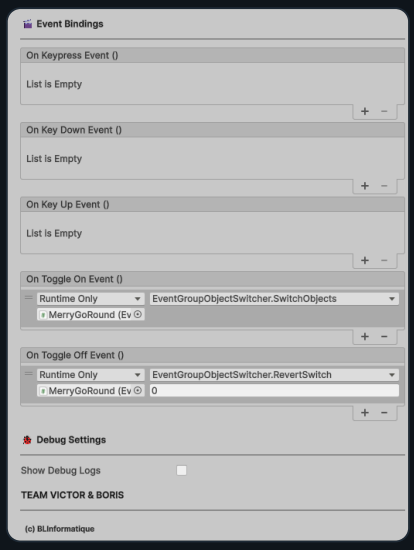
Field	Type	Tooltip	Hints
alwaysShowGizmo	`Boolean`	If true, the interior zone gizmo is always visible in the Scene view.	—
gizmoColor	`Color`	Color of the interior zone gizmo.	—
showGizmoLabel	`Boolean`	If true, a label is displayed in the Scene view for this zone (Editor only).	—

↑ Back to top

KeysEvent MonoBehaviour

BLInformatique.GameTools



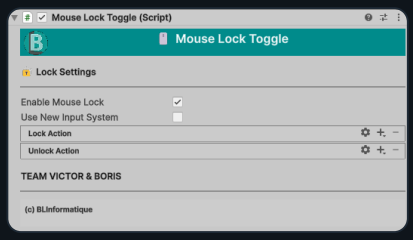


Field	Type	Tooltip	Hints
⬇			
key	`KeyCode`	🔘 Key used to trigger the event (Old Input System).	Enum
⬇			
useNewInputSystem	`Boolean`	Enable this if you're using the new Input System.	—
⬇ 🎮 New Input System Actions			
keypressEvent	`InputAction`	🔵 Input action for 'key held'.	—
keyDownEvent	`InputAction`	⬇️ Input action for 'key down'.	—
keyUpEvent	`InputAction`	⬆️ Input action for 'key up'.	—
⬇			
useToggle	`Boolean`	If true, pressing the key toggles between ON/OFF states.	—
⬇			
onKeypressEvent	`UnityEvent`	Event triggered while the key is held.	UnityEvent
onKeyDownEvent	`UnityEvent`	Event triggered when the key is pressed down.	UnityEvent
onKeyUpEvent	`UnityEvent`	Event triggered when the key is released.	UnityEvent
onToggleOnEvent	`UnityEvent`	Event triggered when toggle is turned ON.	UnityEvent
onToggleOffEvent	`UnityEvent`	Event triggered when toggle is turned OFF.	UnityEvent
⬇			
showDebugLogs	`Boolean`	Enable debug logs in the console.	—

↑ Back to top

MouseLockToggle MonoBehaviour

BLInformatique.GameTools

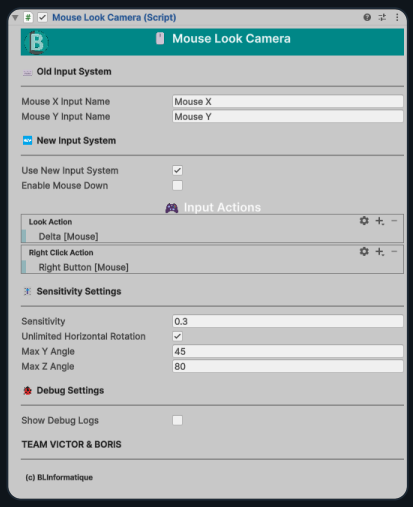


Field	Type	Tooltip	Hints
⬇			
enableMouseLock	`Boolean`	If true, enables mouse lock system.	—
⬇			
mouseLockAction	`MouseButton`	Key used to trigger the event (Old Input System).	Enum
keyUnLockAction	`KeyCode`	Key used to trigger the event (Old Input System).	Enum
⬇			
useNewInputSystem	`Boolean`	Enable the new Input System (InputAction).	—
lockAction	`InputAction`	Action to lock the cursor.	—
unlockAction	`InputAction`	Action to unlock the cursor.	—

↑ Back to top

MouseLookCamera MonoBehaviour

BLInformatique.GameTools



Field	Type	Tooltip	Hints
header	`String`	Header placeholder for BLInformatique styled box in the inspector.	—
↓			
mode	`CameraMode`	Camera behaviour mode. - FreeFly: free camera (like editor fly). - LocalFPSInsideVehicle: camera stays at its local position, only rotates locally (ideal when parented to a moving vehicle).	Enum
↓			
mouseSensitivityX	`Single`	Mouse sensitivity on X (horizontal yaw).	—
mouseSensitivityY	`Single`	Mouse sensitivity on Y (vertical pitch).	—
invertY	`Boolean`	If true, Y mouse movement is inverted.	—
clampVertical	`Boolean`	If true, vertical look angle is clamped between minVerticalAngle and maxVerticalAngle.	—
minVerticalAngle	`Single`	Minimum vertical angle (looking down).	—
maxVerticalAngle	`Single`	Maximum vertical angle (looking up).	—
unlimitedHorizontalRotation	`Boolean`	If true, yaw (horizontal rotation) can rotate freely without clamp.	—
maxHorizontalAngle	`Single`	If horizontal rotation is clamped, maximum yaw angle offset (in degrees) from initial orientation.	—
requireRightMouseButton	`Boolean`	If true, mouse look is only active while the right mouse button is held.	—
lockCursor	`Boolean`	If true, locks cursor on start and while looking around.	—
↓			
allowMovementInFreeFly	`Boolean`	If true in FreeFly mode, WASD/Space/Ctrl (or New Input actions) move the camera.	—
moveSpeed	`Single`	Base movement speed in FreeFly mode.	—
fastMoveMultiplier	`Single`	Multiplier applied when holding Left Shift in FreeFly mode.	—
slowMoveMultiplier	`Single`	Multiplier applied when holding Left Ctrl in FreeFly mode.	—
↓			
horizontalAxis	`String`	Horizontal axis name used for FreeFly movement (typically 'Horizontal').	—

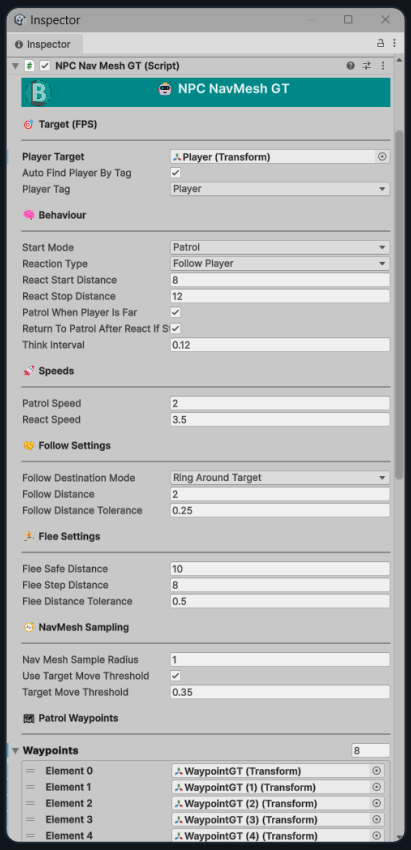
Field	Type	Tooltip	Hints
verticalAxis	`String`	Vertical axis name used for FreeFly movement (typically 'Vertical').	—
mouseXAxis	`String`	Mouse X axis name (legacy Input).	—
mouseYAxis	`String`	Mouse Y axis name (legacy Input).	—
↓			
useNewInputSystem	`Boolean`	If true, uses Unity's New Input System (InputAction) instead of legacy Input axes.	—
lookAction	`InputAction`	InputAction used for looking around. Expected type: Vector2 (X=horizontal, Y=vertical).	—
moveAction	`InputAction`	InputAction used for movement in FreeFly mode. Expected type: Vector2 (X=horizontal, Y=forward/backward).	—
verticalMoveAction	`InputAction`	Optional InputAction used for vertical movement in FreeFly mode. Expected type: float (-1..1), e.g. down/up.	—
↓			
allowEscapeToUnlock	`Boolean`	If true, pressing Escape will unlock the cursor and stop mouse look until clicked again.	—

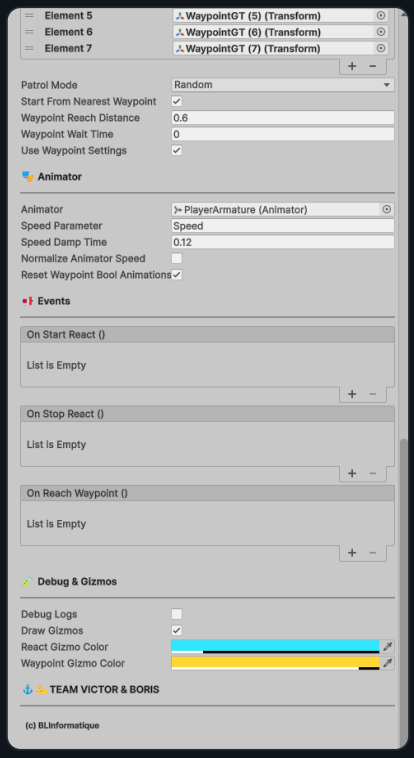
↑ Back to top

NPCNavMeshGT MonoBehaviour

BLInformatique.GameTools

Requires: NavMeshAgent





Field	Type	Tooltip	Hints
↓			
playerTarget	`Transform`	Player transform used as the threat or target.	Object Reference
autoFindPlayerByTag	`Boolean`	Auto-find player by tag at Start if Player Target is null.	—
playerTag	`String`	Tag used to auto-find the player.	—
↓			
startMode	`BehaviourMode`	Initial behaviour when the game starts.	Enum
reactionType	`ReactionType`	Reaction type when the player is close enough.	Enum
reactStartDistance	`Single`	Distance at which the NPC enters React mode.	—
reactStopDistance	`Single`	Distance at which the NPC exits React mode. Should be greater than or equal to React Start Distance.	—
patrolWhenPlayerIsFar	`Boolean`	If true, NPC patrols when the player is far, and reacts only when the player is close enough.	—
returnToPatrolAfterReactIfStartModeWasPatrol	`Boolean`	If true and Start Mode is Patrol, NPC will always return to Patrol after	—

Field	Type	Tooltip	Hints
		losing the player.	
thinkInterval	`Single`	How often in seconds the NPC evaluates distances and refreshes destinations.	—
↓			
patrolSpeed	`Single`	NavMeshAgent speed while patrolling.	—
reactSpeed	`Single`	NavMeshAgent speed while reacting.	—
↓			
followDestinationMode	`FollowDestinationMode`	How the follow destination is computed.	Enum
followDistance	`Single`	Desired distance to keep from the player while following.	—
followDistanceTolerance	`Single`	Extra tolerance in meters for follow distance checks.	—
↓			
fleeSafeDistance	`Single`	Desired safety distance from the player while fleeing.	—
fleeStepDistance	`Single`	How far the NPC tries to run away each update.	—
fleeDistanceTolerance	`Single`	Extra tolerance in meters for flee distance checks.	—

Field	Type	Tooltip	Hints
↓			
navMeshSampleRadius	`Single`	NavMesh sample radius used when computing ring or flee destinations.	—
useTargetMoveThreshold	`Boolean`	If true, destination updates only if player moved enough since last update.	—
targetMoveThreshold	`Single`	Minimum player movement in meters before refreshing destination.	—
↓			
waypoints	`Transform[]`	Waypoints used for patrolling.	List / Array
patrolMode	`PatrolPathMode`	Patrol path mode.	Enum
startFromNearestWaypoint	`Boolean`	If true, NPC starts patrol by going to the nearest waypoint.	—
waypointReachDistance	`Single`	How close the agent must be to a waypoint to consider it reached.	—
waypointWaitTime	`Single`	Default stop duration at each waypoint in seconds if the waypoint does not override it.	—

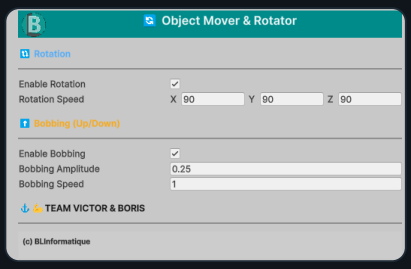
Field	Type	Tooltip	Hints
useWaypointSettings	`Boolean`	If true, the NPC will read per-waypoint settings from a WaypointGT component.	—
↓			
animator	`Animator`	Optional animator used to drive locomotion and waypoint actions.	Object Reference
speedParameter	`String`	Animator float parameter receiving current speed magnitude.	—
speedDampTime	`Single`	Damp time for the animator speed parameter.	—
normalizeAnimatorSpeed	`Boolean`	If true, speed sent to animator is normalized between 0 and 1.	—
resetWaypointBoolAnimationsOnExit	`Boolean`	If true, waypoint bool animation parameters are forced back to false when leaving a waypoint action.	—
↓			
onStartReact	`UnityEvent`	Called when NPC enters React mode.	UnityEvent
onStopReact	`UnityEvent`	Called when NPC exits React mode back to Patrol or Idle.	UnityEvent

Field	Type	Tooltip	Hints
onReachWaypoint	`UnityEvent`	Called when NPC reaches a waypoint.	UnityEvent
⬇			
debugLogs	`Boolean`	Enable debug logs.	—
drawGizmos	`Boolean`	Draw gizmos in Scene view.	—
reactGizmoColor	`Color`	Gizmo color for react ranges.	—
waypointGizmoColor	`Color`	Gizmo color for waypoint path.	—

↑ Back to top

ObjectMoverRotator MonoBehaviour

BLInformatique.GameTools

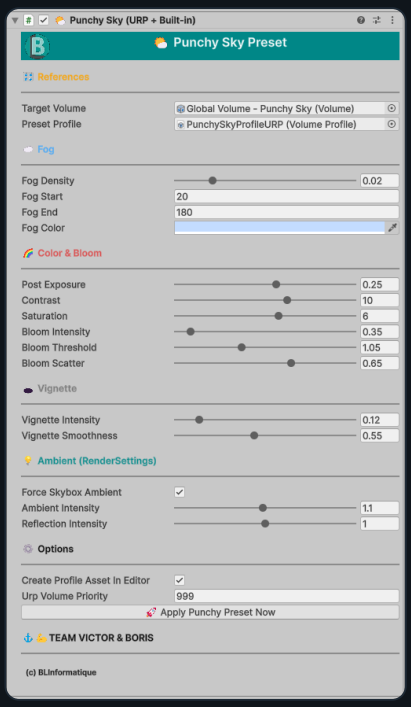


Field	Type	Tooltip	Hints
⬇			
enableRotation	`Boolean`	Enable rotation.	—
rotationSpeed	`Vector3`	Rotation speed in degrees per second.	—
⬇			
enableBobbing	`Boolean`	Enable bobbing (up/down movement).	—
bobbingAmplitude	`Single`	Amplitude of the bobbing (distance in units).	—
bobbingSpeed	`Single`	Speed of bobbing (oscillation per second).	—

↑ Back to top

PunchySkyVolumePreset MonoBehaviour

BLInformatique.GameTools



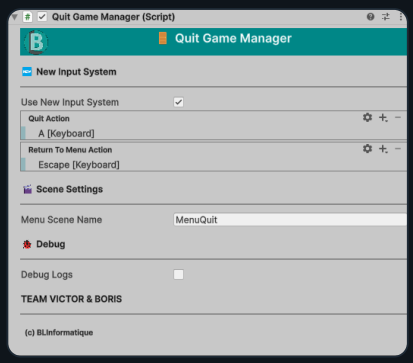
Field	Type	Tooltip	Hints
↓			
targetVolume	`Object`	URP only: assign a Global Volume (Component). In Built-in, leave empty.	Object Reference
presetProfile	`Object`	URP only: assign a VolumeProfile asset (created if missing when applying).	Object Reference
↓			
fogDensity	`Single`	URP Fog density; Built-in uses RenderSettings.fogDensity. • Range [0..0,1]	Range
fogStart	`Single`	URP Fog start distance.	—
fogEnd	`Single`	URP Fog end distance.	—
fogColor	`Color`	Fog color for URP and Built-in fallback.	—
↓			
postExposure	`Single`	Post Exposure (Color Adjustments). • Range [-2..2]	Range
contrast	`Single`	Contrast (Color Adjustments). • Range [-40..40]	Range
saturation	`Single`	Saturation (Color Adjustments). • Range [-40..40]	Range
bloomIntensity	`Single`	Bloom intensity. • Range [0..5]	Range
bloomThreshold	`Single`	Bloom threshold. • Range [0,5..2]	Range
bloomScatter	`Single`	Bloom scatter. • Range [0..1]	Range
↓			
vignetteIntensity	`Single`	Vignette intensity. • Range [0..1]	Range
vignetteSmoothness	`Single`	Vignette smoothness. • Range [0,2..1]	Range
↓			
forceSkyboxAmbient	`Boolean`	If ON, forces Ambient from Skybox and sets intensities (URP & Built-in).	—
ambientIntensity	`Single`	RenderSettings.ambientIntensity. • Range [0,25..2]	Range
reflectionIntensity	`Single`	RenderSettings.reflectionIntensity. • Range [0..2]	Range
↓			

Field	Type	Tooltip	Hints
createProfileAssetInEditor	`Boolean`	Editor only: save a new VolumeProfile asset if none is provided.	—
urpVolumePriority	`Single`	URP Volume priority when auto-created.	—
btnApply	`Int32`		—

↑ Back to top

QuitGameManager MonoBehaviour

BLInformatique.GameTools

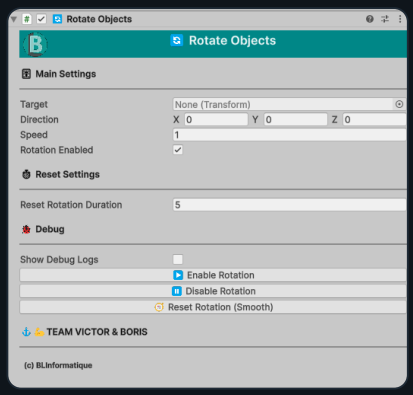


Field	Type	Tooltip	Hints
↓			
quitKey	`KeyCode`	Key to quit the application (only in build).	Enum
returnToMenuKey	`KeyCode`	Key to return to menu scene.	Enum
↓			
useNewInputSystem	`Boolean`	Use new Input System instead of legacy.	—
quitAction	`InputAction`	Input Action to quit the game.	—
returnToMenuAction	`InputAction`	Input Action to return to menu scene.	—
↓			
menuSceneName	`String`	Scene name to load when returning to menu.	—
↓			
debugLogs	`Boolean`	Enable logs for testing.	—

↑ Back to top

rotateObjects MonoBehaviour

BLInformatique.GameTools

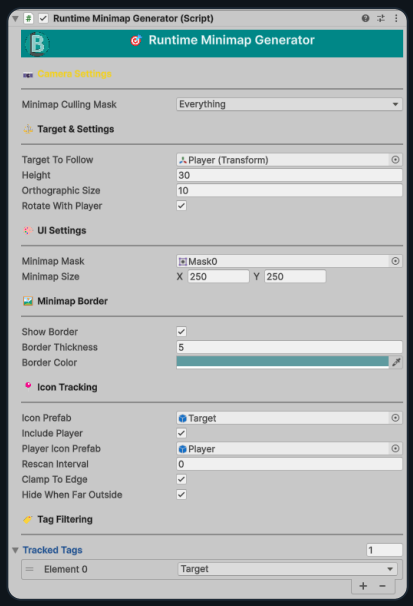


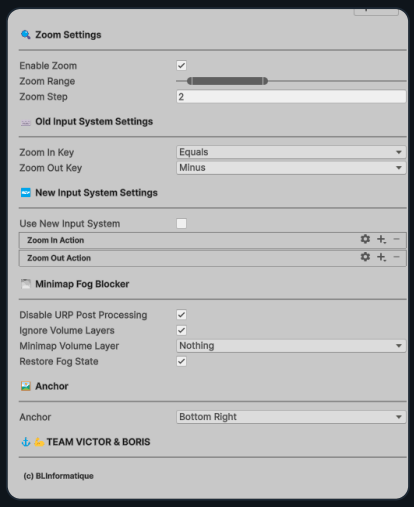
Field	Type	Tooltip	Hints
⬇			
target	`Transform`	Target transform to rotate. If empty, this component's Transform is used.	Object Reference
direction	`Vector3`	Local rotation direction multiplier per axis. Example: (0,1,0) to rotate around Y.	—
speed	`Single`	Angular speed multiplier (degrees per second). Effective rotation is direction * speed.	—
rotationEnabled	`Boolean`	Enable/Disable continuous rotation at runtime.	—
⬇			
resetRotationDuration	`Single`	Duration (seconds) used by the smooth reset to identity rotation.	—
⬇			
showDebugLogs	`Boolean`	Show runtime logs when using public API (toggle, reset, etc.).	—
__btn1	`Int32`		—
__btn2	`Int32`		—
__btn3	`Int32`		—

↑ Back to top

RuntimeMinimapGenerator MonoBehaviour

BLInformatique.GameTools





Field	Type	Tooltip	Hints
↓			
anchor	`MinimapAnchor`	Choose minimap screen corner (auto anchor and pivot).	Enum
↓			
minimapCullingMask	`LayerMask`	Layers rendered by the minimap camera.	—
↓			
targetToFollow	`Transform`	Transform followed by the minimap camera center, usually the player.	Object Reference
height	`Single`	Vertical height offset of the minimap camera above the follow target.	—
orthographicSize	`Single`	Orthographic size of the minimap camera.	—
rotateWithPlayer	`Boolean`	If true, the minimap rotates using the follow target yaw.	—
↓			
minimapMask	`Sprite`	Sprite used as the minimap mask shape.	Object Reference
minimapSize	`Vector2`	Outer size of the minimap on screen in pixels.	—
↓			
showBorder	`Boolean`	If true, displays a visible border around the minimap.	—
borderThickness	`Single`	Border thickness in pixels.	—
borderColor	`Color`	Color of the minimap border.	—
↓			
iconPrefab	`GameObject`	Default icon prefab used for tracked world objects. Must contain a RectTransform.	Object Reference
includePlayer	`Boolean`	If true, creates a dedicated player icon at the minimap center.	—

Field	Type	Tooltip	Hints
playerIconPrefab	`GameObject`	Optional player icon prefab. If empty, the default icon prefab is reused.	Object Reference
rescanInterval	`Single`	Automatic tag rescan interval in seconds. Set to 0 to disable auto rescan.	—
clampToEdge	`Boolean`	If true, tracked icons are clamped to the minimap edge.	—
hideWhenFarOutside	`Boolean`	If true, normal tracked icons are hidden when they are very far outside the minimap.	—
↓			
trackedTags	`List<String>`	Tags of scene objects to track on the minimap.	List / Array
↓			
missionTarget	`Transform`	Mission target transform displayed by the minimap. Can be changed at runtime.	Object Reference
showMissionTarget	`Boolean`	If true, enables the mission target display system.	—
missionTargetIconPrefab	`GameObject`	Prefab used for the mission target icon inside the minimap. If empty, the default icon prefab is reused.	Object Reference
showMissionTargetInMap	`Boolean`	If true, shows the mission target icon inside the minimap when the target is visible.	—
rotateMissionTargetIcon	`Boolean`	If true, rotates the mission target icon inside the minimap toward the target direction.	—
missionTargetIconRotationOffset	`Single`	Extra rotation offset applied to the mission target icon in degrees.	—
↓			
showMissionEdgeArrow	`Boolean`	If true, shows a directional arrow on the minimap border when the mission target is outside the visible minimap area.	—
missionEdgeArrowPrefab	`GameObject`	Prefab used for the mission edge arrow. If empty, the mission target prefab is reused.	Object Reference

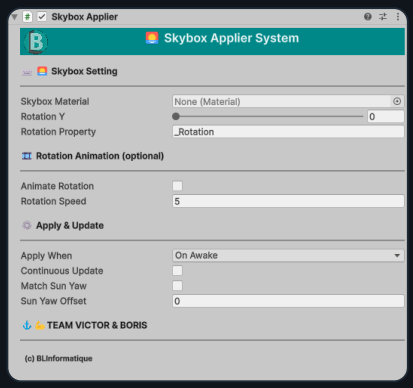
Field	Type	Tooltip	Hints
missionEdgeArrowPadding	`Single`	Padding in pixels between the edge arrow and the minimap border.	—
rotateMissionEdgeArrow	`Boolean`	If true, rotates the mission edge arrow to face the target direction.	—
missionEdgeArrowRotationOffset	`Single`	Extra rotation offset applied to the mission edge arrow in degrees.	—
↓			
showMissionDistance	`Boolean`	If true, generates and shows a TMP distance text in the runtime minimap UI.	—
missionDistancePrefix	`String`	Distance text prefix shown before the numeric value.	—
missionDistanceSuffix	`String`	Distance text suffix shown after the numeric value.	—
missionDistanceFontSize	`Int32`	Font size used for the generated mission distance TMP text.	—
missionDistanceColor	`Color`	Text color used for the generated mission distance TMP text.	—
missionDistanceOffset	`Vector2`	Anchored position of the generated mission distance TMP text relative to the minimap frame.	—
missionDistanceWidth	`Single`	Optional fixed width for the generated distance text area. Set 0 to use minimap width.	—
↓			
enableZoom	`Boolean`	Enable minimap zoom controls.	—
zoomRange	`Vector2`	Minimum and maximum orthographic size allowed for zoom.	—
zoomStep	`Single`	Zoom step applied by inputs.	—
↓			
zoomInKey	`KeyCode`	Key used to zoom in with the old Input System.	Enum
zoomOutKey	`KeyCode`	Key used to zoom out with the old Input System.	Enum
↓			

Field	Type	Tooltip	Hints
useNewInputSystem	`Boolean`	Use Unity New Input System for zoom controls.	—
zoomInAction	`InputAction`	Input Action used to zoom in.	—
zoomOutAction	`InputAction`	Input Action used to zoom out.	—
↓			
disableURPPostProcessing	`Boolean`	If true, disables URP post processing on the minimap camera.	—
ignoreVolumeLayers	`Boolean`	If true, ignores all Volume layers for the minimap camera.	—
minimapVolumeLayer	`LayerMask`	Volume layer mask used if Ignore Volume Layers is disabled.	—
restoreFogState	`Boolean`	If true, temporarily disables RenderSettings fog only while the minimap camera renders.	—
↓			
debugLogs	`Boolean`	Enable debug logs in the console.	—

↑ Back to top

SkyboxApplier MonoBehaviour

BLInformatique.GameTools

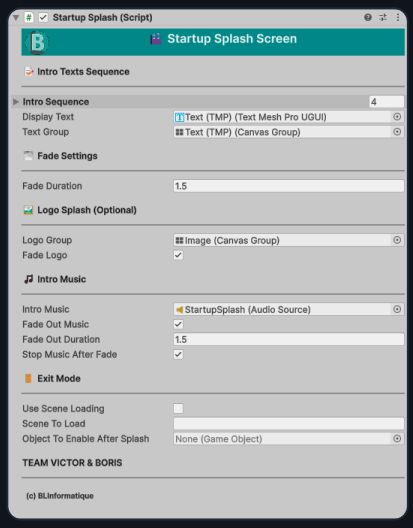


Field	Type	Tooltip	Hints
↓			
skyboxMaterial	`Material`	Assign your Skybox/Panoramic material here.	Object Reference
rotationY	`Single`	Optional Y rotation in degrees for skyboxes that support rotation. • Range [0..360]	Range
rotationProperty	`String`	Shader float property used to rotate the skybox (default: _Rotation).	—
↓			
animateRotation	`Boolean`	Enable continuous yaw animation at runtime (useful for demo scenes).	—
rotationSpeed	`Single`	Degrees per second for automatic yaw animation (if enabled).	—
↓			
applyWhen	`ApplyMode`	When to apply the skybox settings to RenderSettings.	Enum
continuousUpdate	`Boolean`	Continuously writes the rotation property every frame.	—
matchSunYaw	`Boolean`	Match skybox yaw to the Sun (RenderSettings.sun) Y rotation.	—
sunYawOffset	`Single`	Extra yaw offset added when matching the Sun (in degrees).	—

↑ Back to top

StartupSplash MonoBehaviour

BLInformatique.GameTools

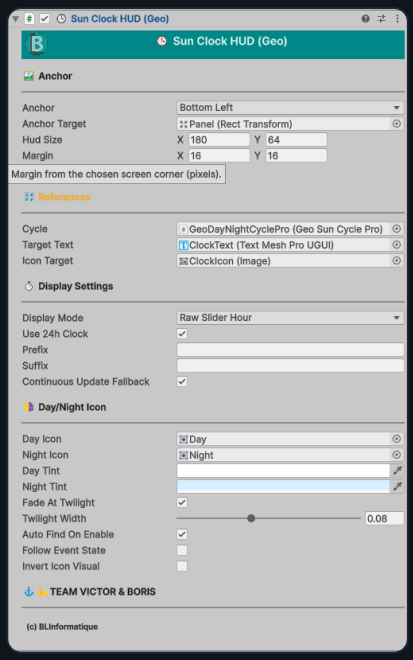


Field	Type	Tooltip	Hints
↓			
introSequence	`List<IntroText>`	List of messages to display with fade effect.	List / Array
displayText	`TextMeshProUGUI`	UI Text (TextMeshProUGUI) to update.	Object Reference
textGroup	`CanvasGroup`	CanvasGroup attached to the text element.	Object Reference
↓			
fadeDuration	`Single`	Duration of fade in/out transitions for text and logo.	—
↓			
logoGroup	`CanvasGroup`	Logo CanvasGroup to fade in/out.	Object Reference
fadeLogo	`Boolean`	Enable fade effect for logo.	—
↓			
introMusic	`AudioSource`	AudioSource for intro music.	Object Reference
fadeOutMusic	`Boolean`	Enable fade out of music after splash.	—
fadeOutDuration	`Single`	Duration of music fade out (in seconds).	—
stopMusicAfterFade	`Boolean`	Stop the music entirely after fade out.	—
↓			
useSceneLoading	`Boolean`	Load a scene automatically after splash sequence.	—
sceneToLoad	`String`	Name of the scene to load.	—
objectToEnableAfterSplash	`GameObject`	Object to enable at the end of the splash (if no scene).	Object Reference

↑ Back to top

SunClockHUDGeo MonoBehaviour

BLInformatique.GameTools



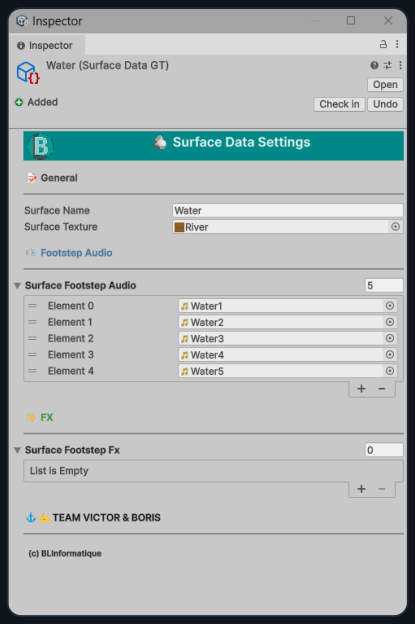
Field	Type	Tooltip	Hints
↓			
anchor	`MinimapAnchor`	Choose screen corner (auto anchor & pivot), same as the minimap.	Enum
anchorTarget	`RectTransform`	Which RectTransform to anchor (leave empty to use this GameObject).	Object Reference
hudSize	`Vector2`	HUD outer size in pixels (applied to anchorTarget.sizeDelta).	—
margin	`Vector2`	Margin from the chosen screen corner (pixels).	—
autoApplyAnchor	`Boolean`	If ON, applies anchor/pivot/size/offset on enable & validate.	—
↓			
cycle	`GeoSunCyclePro`	GeoSunCyclePro providing time/events.	Object Reference
targetText	`TextMeshProUGUI`	Target TextMeshProUGUI used to display the time.	Object Reference
iconTarget	`Image`	Target Image used to display day/night icon.	Object Reference
↓			
displayMode	`DisplayMode`	Choose what the HUD displays: visible time from cycle or raw slider hour.	Enum
use24hClock	`Boolean`	Use 24h time (13:05). If OFF, uses 12h (1:05 PM).	—
prefix	`String`	Optional prefix before the time text (e.g., 'LT: ').	—
suffix	`String`	Optional suffix after the time text (e.g., ' UTC').	—
continuousUpdateFallback	`Boolean`	Update every frame as a safe fallback if events aren't wired yet.	—
↓			
dayIcon	`Sprite`	Sprite shown during DAY.	Object Reference
nightIcon	`Sprite`	Sprite shown during NIGHT.	Object Reference
dayTint	`Color`	Tint color applied to the day icon.	—

Field	Type	Tooltip	Hints
nightTint	`Color`	Tint color applied to the night icon.	—
fadeAtTwilight	`Boolean`	If ON, fades icon when sun is very low (optional, cosmetic).	—
twilightWidth	`Single`	Margin around day threshold to start fading (e.g., 0.08 = 8%). • Range [0..0,2]	Range
autoFindOnEnable	`Boolean`	Try to auto-find the cycle and a TextMeshProUGUI/Image in scene if missing.	—
followEventState	`Boolean`	If ON, HUD follows GeoSunCyclePro events (affected by 'invertDayNightEvents'). If OFF, HUD follows physical sun elevation (ignores event inversion).	—
invertIconVisual	`Boolean`	Swap the visual mapping (Day→moon, Night→sun). Usually keep OFF.	—

↑ Back to top

SurfaceDataGT ScriptableObject

BLInformatique.GameTools



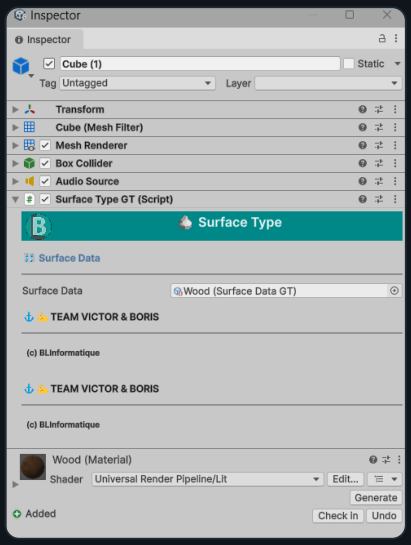
Field	Type	Tooltip	Hints
⬇			
surfaceName	`String`	Friendly display name for this surface (for UI, debug, etc).	—
surfaceTexture	`Texture2D`	Main texture that represents this surface (for material/preview/UI).	Object Reference
⬇			
surfaceFootstepAudio	`AudioClip[]`	Footstep sounds played when player walks/runs/crouches on this surface.	List / Array
⬇			
surfaceFootstepFx	`GameObject[]`	Optional particle/FX objects to spawn on footsteps (dust, splash, etc).	List / Array

↑ Back to top

SurfaceTypeGT MonoBehaviour

BLInformatique.GameTools

Requires: AudioSource

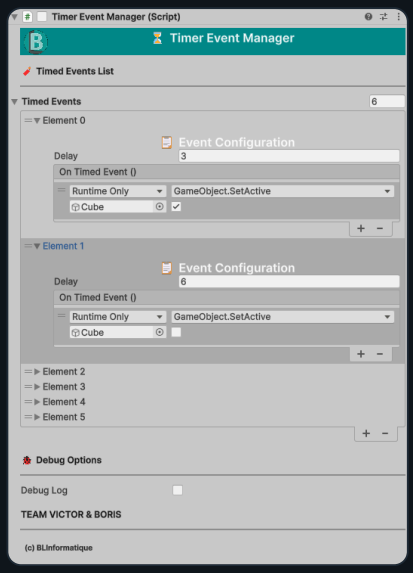


Field	Type	Tooltip	Hints
⬇			
surfaceData	`SurfaceDataGT`	Assign the SurfaceData (ScriptableObject) for this object. Defines all sounds, FX, and texture used by GameTools (steps, impacts, visuals, etc).	Object Reference

↑ Back to top

TimerEventManager MonoBehaviour

BLInformatique.GameTools

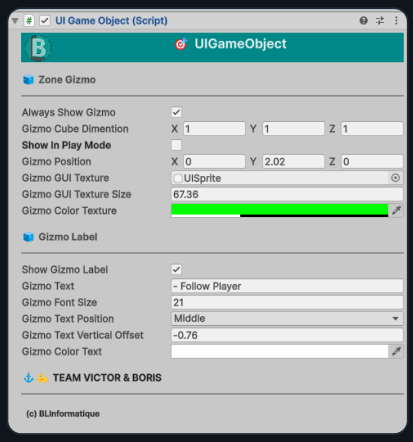


Field	Type	Tooltip	Hints
timedEvents	`List<TimedEvent>`	List of timed events with their respective delays.	List / Array
playOnStart	`Boolean`	If true, starts all timed events automatically on Start().	—
debugLog	`Boolean`	If true, logs triggered events in the Console.	—

↑ Back to top

UIGameObject MonoBehaviour

BLInformatique.GameTools

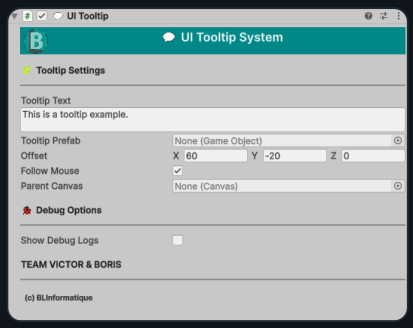


Field	Type	Tooltip	Hints
↓			
alwaysShowGizmo	`Boolean`	If true, the interior zone gizmo is always visible in the Scene view.	—
gizmoCubeDimention	`Vector3`		—
showInPlayMode	`Boolean`	If true, the gizmo and label will also be drawn in the Game view while in Play mode.	—
gizmoPosition	`Vector3`		—
gizmoGUITexture	`Texture2D`	Optional GUI texture to draw at the gizmo position in the Scene view.	Object Reference
gizmoGUITextureSize	`Single`	Size in pixels of the GUI texture drawn in the Scene view.	—
gizmoColorTexture	`Color`	Color of the interior zone gizmo.	—
↓			
showGizmoLabel	`Boolean`	If true, a label is displayed in the Scene view for this zone (Editor only).	—
gizmoText	`String`	If true, the interior zone gizmo is always visible in the Scene view.	—
gizmoFontSize	`Int32`	Font size for the gizmo label in the Scene view.	—
gizmoTextPosition	`TextPosition`	Horizontal position of the gizmo label in the Scene view.	Enum
gizmoTextVerticalOffset	`Single`		—
gizmoColorText	`Color`	Color of the text gizmo.	—
↓			
footerDummy	`String`		—

↑ Back to top

UITooltip MonoBehaviour

BLInformatique.GameTools

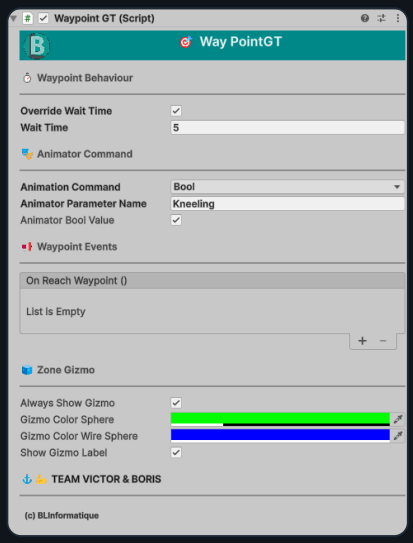


Field	Type	Tooltip	Hints
↓			
tooltipText	`String`	Text to display when hovering over this UI element. • TextArea	TextArea
tooltipPrefab	`GameObject`	Tooltip prefab to spawn. Must include a Text or TextMeshProUGUI component.	Object Reference
offset	`Vector3`	Offset in pixels from the mouse cursor when displayed.	—
followMouse	`Boolean`	If true, the tooltip will follow the mouse while hovering.	—
parentCanvas	`Canvas`	Optional parent canvas override (if not set, uses root canvas).	Object Reference
↓			
showDebugLogs	`Boolean`	Show log messages when tooltip spawns or is destroyed.	—

↑ Back to top

WaypointGT MonoBehaviour

BLInformatique.GameTools



Field	Type	Tooltip	Hints
↓			
overrideWaitTime	`Boolean`	If true, this waypoint overrides the NPC global wait time.	—
waitTime	`Single`	Wait time in seconds for this waypoint.	—
↓			
animationCommand	`AnimationCommand`	Animation command sent to the NPC Animator when this waypoint is reached.	Enum
animatorParameterName	`String`	Animator parameter name used on reach.	—
animatorBoolValue	`Boolean`	Bool value applied if Animation Command is set to Bool.	—
↓			
onReachWaypoint	`UnityEvent`	Event invoked when the NPC reaches this waypoint.	UnityEvent
↓			
alwaysShowGizmo	`Boolean`	If true, the waypoint gizmo is always visible in the Scene view.	—
gizmoColorSphere	`Color`	Fill color of the waypoint gizmo sphere.	—
gizmoColorWireSphere	`Color`	Wire color of the waypoint gizmo sphere.	—
showGizmoLabel	`Boolean`	If true, a label is displayed in the Scene view.	—
↓			
footerDummy	`String`		—

↑ Back to top