



v3.0

Documentation

[download](#) | [install](#) | [learn](#)

Overview

Core classes

These are the most useful classes of the plugin, like gamepads and buttons.

- Gamepads
 - **Hinput** (static): The main class of the plugin, used to access gamepads.
 - **Gamepad**
 - **VibrationPreset** (enum): A list of fine-tuned vibration presets.
- Buttons
 - **Pressable** (abstract): Anything that can be pressed and released.
 - **Button** (inherits **Pressable**)
 - **Trigger** (inherits **Pressable**)
 - **StickDirection** (inherits **Pressable**): A virtual button considered pressed if its stick is pushed in the right direction.
 - **StickPressedZone** (inherits **Pressable**): A virtual button considered pressed if its stick is pushed in any direction.
 - **Press**: A specific way of pressing a **Pressable**, like a simple press, a double press or a long press.
- Sticks
 - **Stick**
- Utility
 - **Settings** (inherits **MonoBehaviour**): Handles the user parameters of Hinput.

***Hinput** can be called from anywhere. However, the rest of these classes are part of a namespace called **HinputClasses**, and need a “using HinputClasses;” directive in order to be used.*

Internal classes

These are the internal classes of the plugin. You do NOT need to interact with them, and they are not described any further in this document.

Greyed out classes are not internal, but they are shown to illustrate inheritance.

- Gamepads
 - **Gamepad**
 - **AnyGamepad** (inherits **Gamepad**): Every gamepad at once.
 - **Vibration**: Handles the vibration of a gamepad.
- Buttons
 - **Pressable** (abstract)
 - **GamepadPressable** (abstract, inherits **Pressable**): A **Pressable** on a **Gamepad**.
 - **Button** (inherits **GamepadPressable**)
 - **AnyGamepadButton** (inherits **Button**): A given **Button** on every gamepad at once.
 - **Trigger** (inherits **GamepadPressable**)
 - **AnyInput** (inherits **Pressable**): Every button of a gamepad at once.
 - **StickPressable** (abstract, inherits **Pressable**): A pressable on a **Stick**.
 - **StickDirection** (inherits **StickPressable**)
 - **StickPressedZone** (inherits **StickPressable**)
- Sticks
 - **Stick**
 - **AnyGamepadStick** (inherits **Stick**): A given **Stick** on every gamepad at once.
 - **Axis**: Used to calculate the position of a **Stick**.
- Utility
 - **Setup** (static): Handles the setup and uninstall sequences of Hinput.
 - **Updater** (inherits **MonoBehaviour**): Handles gamepad refresh.
 - **Utils** (static): Contains useful methods and properties needed by the plugin.
- Test scenes
 - **Debugger**: Handles the Debug Scene.
 - **ExampleManager**: Handles the Example Scene.
 - **ExampleGamepad**: Handles one gamepad of the Example Scene.

*These classes are part of the **HinputClasses.Internal** namespace.*

Hinput (static)

This class is not part of a namespace - It can be accessed from anywhere without a “using” directive.

The main class of the plugin, used to access gamepads.

Static properties

- **gamepad (List<Gamepad>)**
 - A list of 8 gamepads, labelled 0 to 7.
- **anyGamepad (Gamepad)**
 - A virtual gamepad that returns the inputs of every gamepad at once.
 - A button is pressed on *AnyGamepad* if there is at least one gamepad on which it is pressed. It is released on *AnyGamepad* if it is released on all gamepads.
 - The position of a stick on *AnyGamepad* is the average position of pushed sticks of that type on all gamepads.
 - Vibrating *AnyGamepad* vibrates all gamepads.
 - Examples:
 - If player 1 pushed their A button and player 2 pushed their B button, both the A and the B button of *AnyGamepad* will be *pressed*.
 - If both player 1 and player 2 hold their A button, and player 1 releases it, the A button of *AnyGamepad* will still be considered pressed, and will not trigger a *justReleased* event.
 - If player 1 pushed their left trigger by 0.2 and player 2 pushed theirs by 0.6, the left trigger of *AnyGamepad* will have a *position* of 0.6.
 - If player 1 positioned their right stick at (-0.2, 0.9) and player 2 has theirs at (0, 0), the position of the right stick of *AnyGamepad* will be (-0.2, 0.9).
 - If player 1 positioned their right stick at (-0.2, 0.9) and player 2 has theirs at (0.6, 0.3), the position of the right stick of *AnyGamepad* will be the average of both positions, (0.2, 0.6).
- **anyInput (Pressable)**
 - A virtual button that returns every input of every gamepad at once.
 - *AnyInput* is considered pressed if at least one input is pressed.
 - *AnyInput* is considered released if every input is released.

Settings : MonoBehaviour

This class is part of the `HinputClasses` namespace.

Hinput class responsible for handling settings.

Can be attached to a gameobject to expose settings. Otherwise it will automatically be instantiated at runtime the first time **Hinput** is called, with default settings.

Settings calls *DontDestroyOnLoad* when created, which means it is persistent across all scenes.

Static properties (serialized in the editor)

Implicit Conversion

- *defaultPressType* (**DefaultPressTypes**, default: Simple Press)
 - The default conversion of **Pressable** to **Press** values.
 - Determines how Hinput interprets buttons, triggers and stick directions when the type of Press to use is not specified.
 - Possible values: Simple Press, Double Press, Long Press.
- *defaultPressFeature* (**DefaultPressFeatures**, default: Pressed)
 - The default conversion of **Press** and **Pressable** to **bool** values.
 - Determines how Hinput interprets buttons, triggers and stick directions when the feature to use is not specified.
 - Possible values: Pressed, Just Pressed, Released, Just Released.

Presses

- *doublePressDuration* (**float**, range (0,2), default : 0.3)
 - The maximum duration between the start of two presses for them to be considered a double press.
- *longPressDuration* (**float**, range (0,2), default : 0.3)
 - The minimum duration of a press for it to be considered a long press.

Sticks

- *stickType* (**StickTypes**, default : Eight Directions)
 - The type of stick to use.
 - Set it to Four Directions for 4-directional sticks, with virtual buttons that span $\frac{1}{4}$ of a circle (90 degrees). Use diagonals with caution in this case.
 - Set it to Eight Directions for 8-directional sticks, with virtual buttons that span $\frac{1}{8}$ of a circle (45 degrees).
 - Possible values: Four Directions, Eight Directions.



- *stickDeadZone* (**float**, range (0,1), default : 0.2)
 - The distance from the origin beyond which stick inputs start being registered.
- *stickPressedZone* (**float**, range (0,1), default : 0.5)
 - The distance from the end of the dead zone beyond which stick inputs are considered pushed.
- *worldCamera* (**Camera**, default : null)
 - The **Camera** on which the *worldPositionCamera* property of **Stick** should be calculated.
 - If no **Camera** is set, Hinput will try to get the gameobject tagged "MainCamera". If there isn't one, Hinput will get the first gameobject on the scene that has a **Camera** component.
 - If there is no **Camera** on the scene, Hinput will return an error whenever *worldPositionCamera* is called.

Triggers

- *triggerDeadZone* (**float**, range (0,1), default : 0.1)
 - The distance from the origin beyond which trigger inputs start being registered
- *triggerPressedZone* (**float**, range (0,1), default : 0.5)
 - The distance from the end of the dead zone beyond which trigger inputs are considered pushed.

Vibration Defaults

- *vibrationDefaultLeftIntensity* (**float**, range (0,1), default : 0.2)
 - The default intensity of the left (low-frequency) motor when controllers vibrate.
 - The left motor's vibration feels like a low rumble.
- *vibrationDefaultRightIntensity* (**float**, range (0,1), default : 0.8)
 - The default intensity of the right (high-frequency) motor when controllers vibrate.
 - The right motor's vibration feels like a sharp buzz.
- *vibrationDefaultDuration* (**float**, range (0,2), default : 0.2)
 - The default duration of gamepad vibrations.

Performance

If your game needs less than 8 gamepads, or if there are inputs that you will never use, you can disable some features of the plugin to improve performance.

By default, Hinput updates every button of each gamepad from the moment it is first connected. However, disabled gamepads and features have *isEnabled* set to false by default, and are not updated by Hinput (which slightly reduces update time each frame).

These settings need to be set before entering play mode, and the only way to reactivate a feature afterwards is to call its *Enable* method.

- *amountOfGamepads* (**int**, range (0.8), default : 8)
 - How many gamepads should be automatically enabled when they are connected for the first time.
- *disableAnyGamepad* (**bool**, default : false)
 - Cause the *AnyGamepad* feature to not be tracked by Hinput.
- *disableAnyInput* (**bool**, default : false)
 - Cause the *AnyInput* feature of gamepads to not be tracked by Hinput.
- *disableA, disableB, disableX, disableY, disableLeftBumper, disableRightBumper, disableLeftTrigger, disableRightTrigger, disableBack, disableStart, disableLeftStickClick, disableRightStickClick, disableLeftStick, disableRightStick, disableDPad* (**bool**, default : false)
 - Cause the corresponding input to not be tracked by Hinput.



Gamepad

This class is part of the `HinputClasses` namespace.

Hinput class representing a gamepad.

Properties

- ***isConnected* (bool)**
 - Returns true if a gamepad is currently connected. Returns false otherwise.
 - On *AnyGamepad*, returns true if at least one gamepad is connected. Returns false otherwise.
- ***isEnabled* (bool)**
 - Returns true if a gamepad is being tracked by Hinput. Returns false otherwise.
 - On *AnyGamepad*, returns true if *AnyGamepad* is enabled. This does NOT give any information on regular gamepads. Returns false otherwise.
- ***name* (string)**
 - The name of a gamepad, like "Gamepad0" or "AnyGamepad".
- ***index* (int)**
 - The index of a gamepad in the *gamepad* list of Hinput. *AnyGamepad* returns -1.
- ***type* (int)**
 - The type of a gamepad, like "Controller (Xbox One For Windows)", "Wireless Controller" or "AnyGamepad".
- ***A, B, X, Y, leftBumper, rightBumper, back, start, leftStickClick, rightStickClick* (Button)**
 - The buttons of a gamepad.
- ***leftTrigger, rightTrigger* (Trigger)**
 - The triggers of a gamepad.
- ***leftStick, rightStick, dPad* (Stick)**
 - The sticks of a gamepad.
- ***anyInput* (Pressable)**
 - The virtual button that returns every input of a gamepad at once.
 - *AnyInput* is considered pressed if at least one input on that gamepad is pressed.
 - *AnyInput* is considered released if every input on that gamepad is released.

- *leftVibration, rightVibration (float)*
 - The intensity at which the left and right motors of a gamepad are currently vibrating.
 - On *AnyGamepad*, returns -1.
- *sticks (List<Stick>)*
 - The list containing a gamepad's sticks, in the following order : { *leftStick, rightStick, dPad* }
- *buttons (List<Pressable>)*
 - The list containing a gamepad's buttons, in the following order : { *A, B, X, Y, leftBumper, rightBumper, leftTrigger, rightTrigger, back, start, leftStickClick, rightStickClick* }

Methods

- *Enable* (no arguments)
 - Enable a gamepad so that Hinput starts tracking it.
 - This method is called automatically on a gamepad the first time it is connected, except if it was disabled in **Settings**.
 - Calling this method on *AnyGamepad* only enables *AnyGamepad*.
- *Disable* (no arguments)
 - Reset and disable a gamepad so that Hinput stops tracking it.
 - This may improve performance.
 - Calling this method on *AnyGamepad* only disables *AnyGamepad*.
- *Vibrate* (no arguments)
 - Vibrate a gamepad with a constant intensity.
 - The duration and intensity are fetched from **Settings**, with a default value of (0.2f, 0.8f, 0.2f) or (**VibrationPreset.Impact**).
 - Calling this method on *AnyGamepad* vibrates all gamepads.
 - Overloads : the intensity or duration of the vibration can be modified.
 - (leftIntensity (**float**), rightIntensity (**float**))
 - (duration (**float**))
 - (leftIntensity (**float**), rightIntensity (**float**), duration (**float**))
- *Vibrate* (argument : vibrationPreset (**VibrationPreset**))
 - Vibrate a gamepad with an intensity based on a **VibrationPreset**.
 - Calling this method on *AnyGamepad* vibrates all gamepads.
 - Overloads : duration or the intensity of the **VibrationPreset** can be modified with multipliers.
 - (vibrationPreset (**VibrationPreset**), leftIntensity (**float**), rightIntensity (**float**))
 - (vibrationPreset (**VibrationPreset**), duration (**float**))

- (vibrationPreset (**VibrationPreset**), leftIntensity (**float**), rightIntensity (**float**), duration (**float**))
- *Vibrate* (argument : curve (**AnimationCurve**))
 - Vibrate a gamepad with an intensity based on an **AnimationCurve**.
 - Calling this method on *AnyGamepad* vibrates all gamepads.
 - Overloads : two animation curves can be used instead, one for the left side and one for the right side.
 - (leftCurve (**AnimationCurve**), rightCurve (**AnimationCurve**))
- *VibrateAdvanced* (arguments : leftIntensity (**float**), rightIntensity (**float**))
 - Vibrate the left motor of a gamepad with an intensity of *leftIntensity*, and the right motor with an intensity of *rightIntensity*, FOREVER.
 - Don't forget to call *StopVibration* !
 - Calling this method on *AnyGamepad* vibrates all gamepads.
- *StopVibration* (no arguments)
 - Stop all vibrations on a gamepad immediately.
 - Calling this method on *AnyGamepad* stops all gamepads.
 - Overloads : vibration can be stopped progressively over a duration.
 - (duration (**float**))



VibrationPreset (enum)

This class is part of the HinputClasses namespace.

Hinput enum listing some vibration patterns that can be played on a gamepad.

Values

- *ButtonPress*
 - A short vibration, suitable for feedback after the player pressed a button.
 - Similar to *Vibrate(0.5f, 0.5f, 0.1f)*.
- *ImpactLight*
 - A short and intense vibration, suitable for a light impact.
 - Similar to *Vibrate(0f, 0.5f, 0.2f)*.
- *Impact*
 - A short and intense vibration, suitable for an impact.
 - Similar to *Vibrate(0.2f, 0.8f, 0.2f)*.
- *ImpactHeavy*
 - A short and intense vibration, suitable for a heavy impact.
 - Similar to *Vibrate(0.5f, 1f, 0.2f)*.
- *ExplosionShort*
 - A low and powerful vibration, suitable for a short or distant explosion.
 - Similar to *Vibrate(0.5f, 0.25f, 0.2f)*.
- *Explosion*
 - A low and powerful vibration, suitable for an explosion.
 - Similar to *Vibrate(0.8f, 0.4f, 0.5f)*.
- *ExplosionLong*
 - A low and powerful vibration, suitable for a long or nearby explosion.
 - Similar to *Vibrate(1f, 0.5f, 1f)*.
- *AmbientSubtle*
 - A 10-second constant, low and subtle vibration, suitable for an ongoing event.
 - Similar to *Vibrate(0.1f, 0f, 10f)*.
- *Ambient*
 - A 10-second constant, low vibration, suitable for an ongoing event.
 - Similar to *Vibrate(0.3f, 0.1f, 10f)*.
- *AmbientStrong*



- A 10-second constant, low and strong vibration, suitable for an ongoing event.
- Similar to *Vibrate(0.6f, 0.3f, 10f)*.



Pressable (abstract)

This class is part of the `HinputClasses` namespace.

Hinput abstract class representing anything that can be pressed and released.

The classes **Button**, **Trigger**, **StickDirection**, and **StickPressedZone** are derived from **Pressable**.

Implicit Conversions

By default, if no property of a **Pressable** is used, it is automatically cast to one of these:

- A **bool** with the value `simplePress.pressed`
- A **Press** with the value `simplePress`

//What you write	//How Hinput interprets it
<code>gamepad.A</code>	<code>gamepad.A(.simplePress)(.pressed)</code>
<code>gamepad.A.justPressed</code>	<code>gamepad.A(.simplePress).justPressed</code>
<code>gamepad.A.released</code>	<code>gamepad.A(.simplePress).released</code>
<code>gamepad.A.justReleased</code>	<code>gamepad.A(.simplePress).justReleased</code>
<code>gamepad.A.doublePress</code>	<code>gamepad.A.doublePress(.pressed)</code>
<code>gamepad.A.longPress</code>	<code>gamepad.A.longPress(.pressed)</code>
<code>gamepad.A.simplePress.pressed</code>	<code>gamepad.A.simplePress.pressed</code>
<code>gamepad.A.doublePress.justPressed</code>	<code>gamepad.A.doublePress.justPressed</code>
<code>gamepad.A.longPress.justReleased</code>	<code>gamepad.A.longPress.justReleased</code>

Properties

- **name (string)**
 - The name of an input, like “Gamepad0_A”, “AnyGamepad_RightStick_Up” or “Gamepad2_AnyInput”.
- **gamepad (Gamepad)**
 - The gamepad an input is attached to.



- **isEnabled (bool)**
 - Returns true if an input is being tracked by Hinput. Returns false otherwise.
 - On *AnyInput*, returns true if *AnyInput* is enabled (this does NOT give any information on regular inputs). Returns false otherwise.
- **pressed (bool)**
 - Returns true if an input is pressed. Returns false otherwise.
- **released (bool)**
 - Returns true if an input is released. Returns false otherwise.
- **justPressed (bool)**
 - Returns true if an input has been pressed this frame. Returns false otherwise.
- **justReleased (bool)**
 - Returns true if an input has been released this frame. Returns false otherwise.
- **pressDuration (float)**
 - How long an input has been pressed (0 if it is released).
- **releaseDuration (float)**
 - How long an input has been released (0 if it is pressed).
- **simplePress (Press)**
 - Considered pressed whenever an input is pressed.
- **doublePress (Press)**
 - Considered pressed when an input has been pressed twice in a row.
- **longPress (Press)**
 - Considered pressed when an input has been pressed for a long time.

Methods

- **Enable (no arguments)**
 - Enable an input so that Hinput starts tracking it.
 - Calling this method on *AnyInput* only enables *AnyInput*.
- **Disable (no arguments)**
 - Disable an input so that Hinput stops tracking it.
 - This may improve performance.
 - Calling this on *AnyInput* only disables *AnyInput*.

Derived classes

The classes **Button**, **Trigger**, **StickDirection**, and **StickPressedZone** are derived from **Pressable**.

Button

This class is part of the `HinputClasses` namespace.

Hinput class representing a physical button of the controller, such as the A button, a bumper or a stick click.

- *index* (**int**) : The index of a button on its gamepad.

Trigger

This class is part of the `HinputClasses` namespace.

Hinput class representing the left or right trigger of a controller.

- *index* (**int**) : The index of a trigger on its gamepad.
- *position* (**float**) : The position of a trigger, between 0 and 1.

StickDirection

This class is part of the `HinputClasses` namespace.

Hinput class representing a given direction of a stick or D-pad, such as the up or down-left directions.

- *stick* (**Stick**) : The stick a direction is attached to.

StickPressedZone

This class is part of the `HinputClasses` namespace.

Hinput class representing a stick or D-pad as a button. It is considered pressed if the stick is pushed in any direction.

- *stick* (**Stick**) : The stick this button is attached to.



Press

This class is part of the `HinputClasses` namespace.

Hinput class representing a specific way of pressing a button, like a regular press, a double press or a long press.

Properties

- *button* (**Pressable**)
 - The button a press refers to.
- *gamepad* (**Gamepad**)
 - The gamepad of the button a press refers to.
- *pressed* (**bool**)
 - Returns true if a press is pressed. Returns false otherwise.
- *released* (**bool**)
 - Returns true if a press is released. Returns false otherwise.
- *justPressed* (**bool**)
 - Returns true if a press has been pressed this frame. Returns false otherwise.
- *justReleased* (**bool**)
 - Returns true if a press has been released this frame. Returns false otherwise.
- *pressDuration* (**float**)
 - How long a press has been pressed (0 if it is released).
- *releaseDuration* (**float**)
 - How long a press has been released (0 if it is pressed).



Stick

This class is part of the `HinputClasses` namespace.

Hinput class representing a gamepad stick, such as a left stick, a right stick, or a D-pad.

Implicit Conversion

By default, if no property of a **Stick** is used, it is automatically cast to one of these:

- A **Vector2** with the value *position*
- A **bool** with the value *inPressedZone.simplePress.pressed*

//What you write	//How Hinput interprets it
// As a Vector2 gamepad.leftStick	gamepad.leftStick.position
// As a bool gamepad.leftStick	gamepad.leftStick.inPressedZone.simplePress.pressed

Properties

- *index* (**int**)
 - The index of a stick on its gamepad (0 for a left stick, 1 for a right stick, 2 for a D-pad).
- *name* (**string**)
 - The name of a stick, like “Gamepad0_LeftStick” or “AnyGamepad_DPad”.
- *gamepad* (**Gamepad**)
 - The gamepad a stick is attached to.
- *isEnabled* (**bool**)
 - Returns true if a stick is being tracked by Hinput. Returns false otherwise.
- *position* (**Vector2**)
 - The coordinates of a stick.
- *horizontal* (**float**)
 - The position of a stick along the horizontal axis (between -1 and 1).
- *vertical* (**float**)
 - The position of a stick along the vertical axis (between -1 and 1).

- *distance* (**float**)
 - The distance from the current position of a stick to its origin (between 0 and 1).
- *angle* (**float**)
 - The angle between the current position of a stick and the horizontal axis (In degrees : left=180, up=90, right=0, down=-90).
- *worldPositionCamera* (**Vector3**)
 - The coordinates of a stick as a Vector3 facing the camera.
 - The horizontal and vertical axes of the stick are interpreted as the right and up directions of the camera.
- *worldPositionFlat* (**Vector3**)
 - The coordinates of a stick as a horizontal Vector3, with a y value of 0.
 - The horizontal and vertical axes of the stick are interpreted as the absolute right and forward directions.
- *inPressedZone* (**StickPressedZone**)
 - The virtual button considered pressed if a stick is pushed in any direction.
- *up, down, left, right, upLeft, downLeft, upRight, downRight* (**StickDirection**)
 - Virtual buttons considered pressed if a stick is pushed in the right direction.

