

2024

**SHARK**  
STUDIO



# INPUT SYSTEM

## User Manual

<https://assetstore.unity.com/publishers/55210>

2024

**SHARK**  
STUDIO

How to get started with the asset, step by step

# SUMMARY

## 1. Installation & Setup

- Downloading from the Asset Store
- Configure the old InputManager

## 2. Workflow

- The NYX InputManager
- Core Properties

## 3. Code Implementation

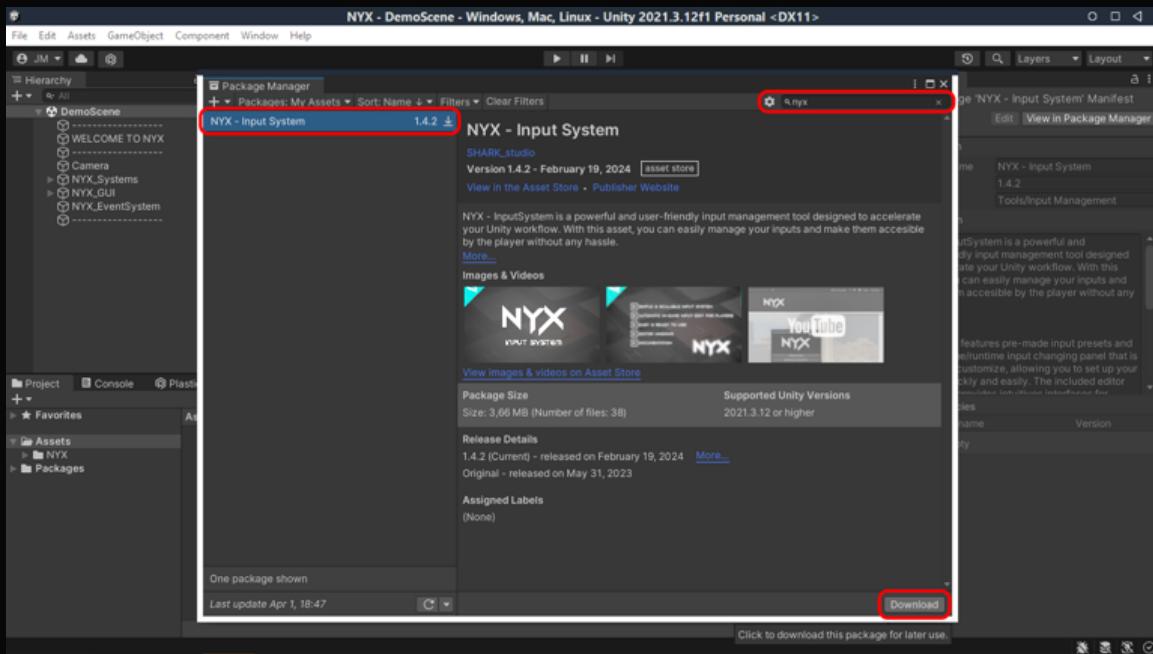
- Scripting requirements
- How to handling inputs

Downloading from the Asset Store

# ⬇️ INSTALLATION

You can find **NYX** on the Unity [AssetStore](#) at the link bellow :

[assetstore.unity.com/packages/tools/input-management/nyx-input-system](https://assetstore.unity.com/packages/tools/input-management/nyx-input-system)

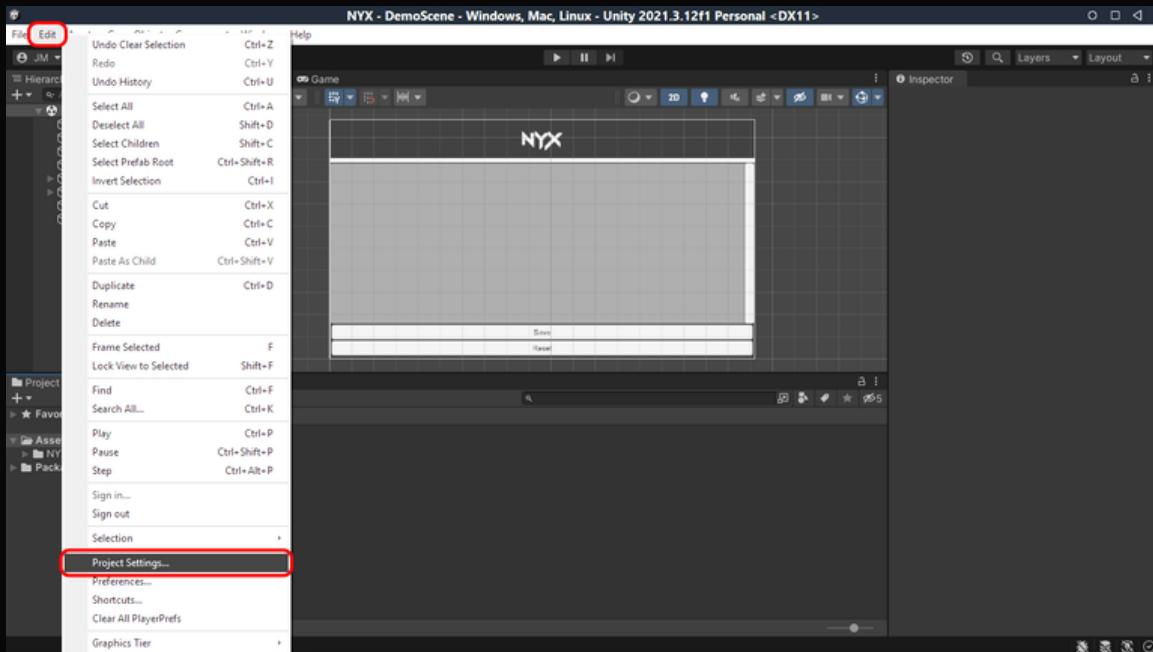


You can then import NYX like any other asset using the package manager.

Configure the old InputManager

# ⬇️ INSTALLATION

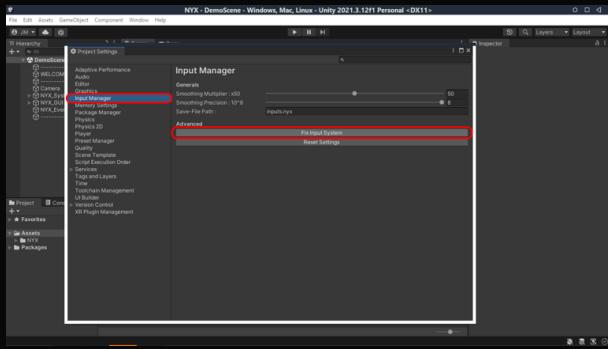
First, ensure that you are using the old [Unity Input Manager](#) in your project.



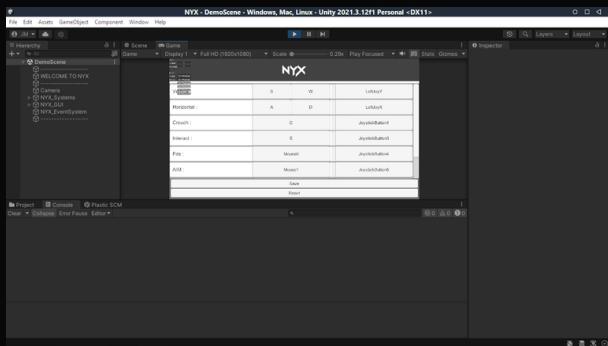
To continue the setup, navigate to [Edit >> Project Settings](#).

Configure the old InputManager

# INSTALLATION



Finally, within the project settings, navigate to **Input Manager** and then click the '[Fix Input System](#)' button.



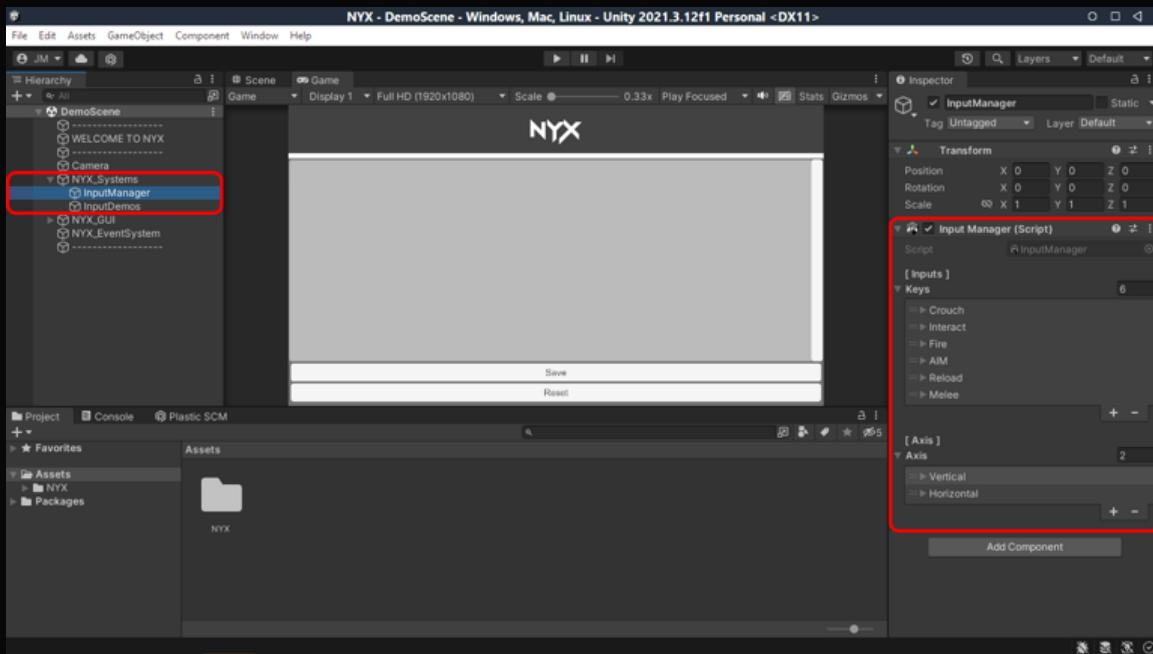
NYX is now installed and production-ready.  
It is recommended to run the demo scene to ensure everything is working and you don't encounter any errors.

If you run into any issues, please visit our [Discord](#) server.

The NYX InputManager

# WORKFLOW

In order to work with [NYX](#), we will need to use its main [Component](#), the `InputManager.cs`. It is the core of the asset that handles all input logic in your game.

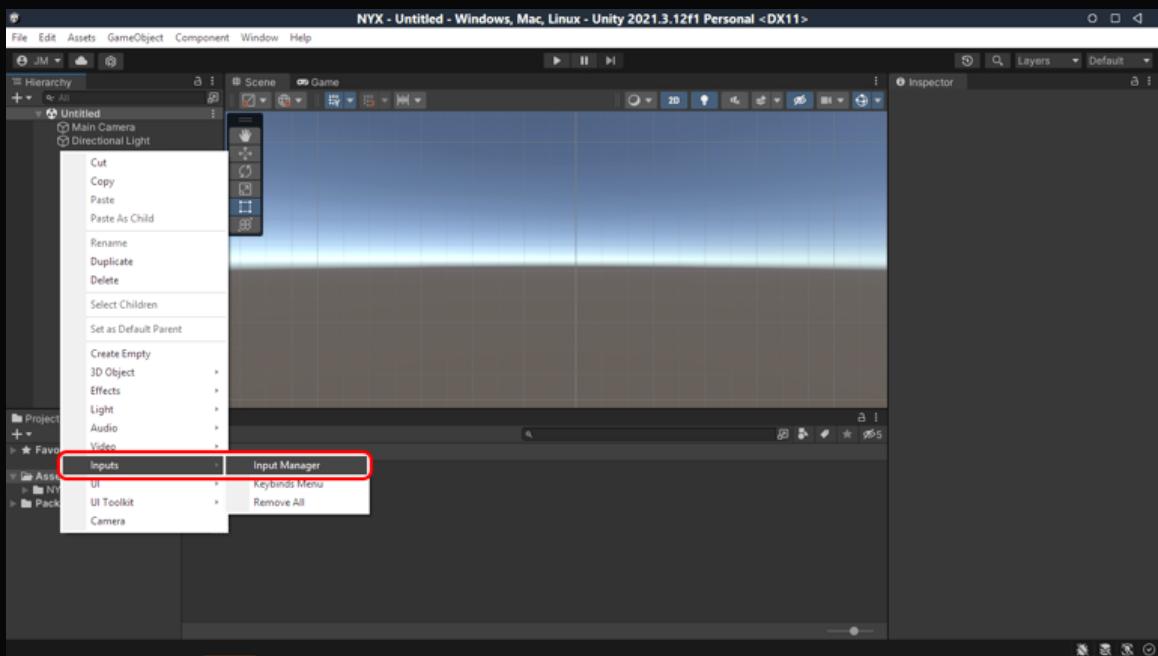


It is placed on a [GameObject](#) of the same name and is present by default when you create a new scene.

The NYX InputManager

# WORKFLOW

In order to work with [NYX](#), we will need to use its main [Component](#), the `InputManager.cs`. It is the core of the asset that handles all input logic in your game.

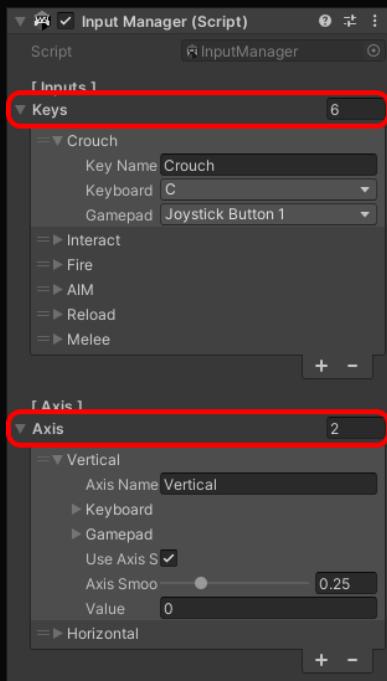


If you don't have an [InputManager](#) in your scene, you can add one by right-clicking on the [Hierarchy](#) and navigating to: [Inputs >> Input Manager](#).

## Core Properties

# WORKFLOW

The NYX `InputManager.cs` possesses **Keys** and **Axes** similar to Unity's [Input Manager](#).



Each element of both **Axes & Keys** possesses its own properties such as **Name**, **Keyboard Key(s)**, **Gamepad Key(s)**, etc.

For more infos on others features such as the in-game GUI, please watch the [youtube tutorials](#) or head to my [Discord](#) server.

Scripting requirements

# 🛠 IMPLEMENTATION

Before proceeding further, to work with NYX, you need to import it :

```
using NYX; // Importing NYX namespace
```

Because the NYX `InputManager` is scene-based, we need a reference to it. There are three ways of referencing the `InputManager`:

Via the provided `static class` instance (works with only 1 `InputManager`)

```
private InputManager inputs; // NYX InputManager

// Get reference to static instance provided by NYX
void Start(){
    inputs = InputManager.instance;
}
```

Manually, using a `public` variable

```
public InputManager inputs; // NYX InputManager
```

Or locating it at game start

```
private InputManager inputs; // NYX InputManager

// Search GameObject named 'InputManager' and get its component
void Start(){
    GameObject.Find("InputManager").
        TryGetComponent<InputManager>(out inputs);
}
```

How to handling inputs

# 🛠 IMPLEMENTATION

There are two things you can work with: [Axes & Keys](#)

To work with keys, you can use these [methods](#)

```
void Update(){
    // Return 'true' while holding key
    if(inputs.GetKey("MyKeyName"))
        { Debug.Log("Holding Key!"); }

    // Return 'true' at key press
    if (inputs.GetKeyDown("MyKeyName"))
        { Debug.Log("Pressing Key!"); }

    // Return 'true' at key release
    if (inputs.GetKeyUp("myKeyName"))
        { Debug.Log("Releasing Key!"); }
}
```

To work with axes, you can use this [method](#)

```
void Update(){
    float myAxisValue = inputs.GetAxis("myAxisName");
}
```

NYX

THANK YOU  
FOR DOWNLOADING

MY ASSET :)