Introduction

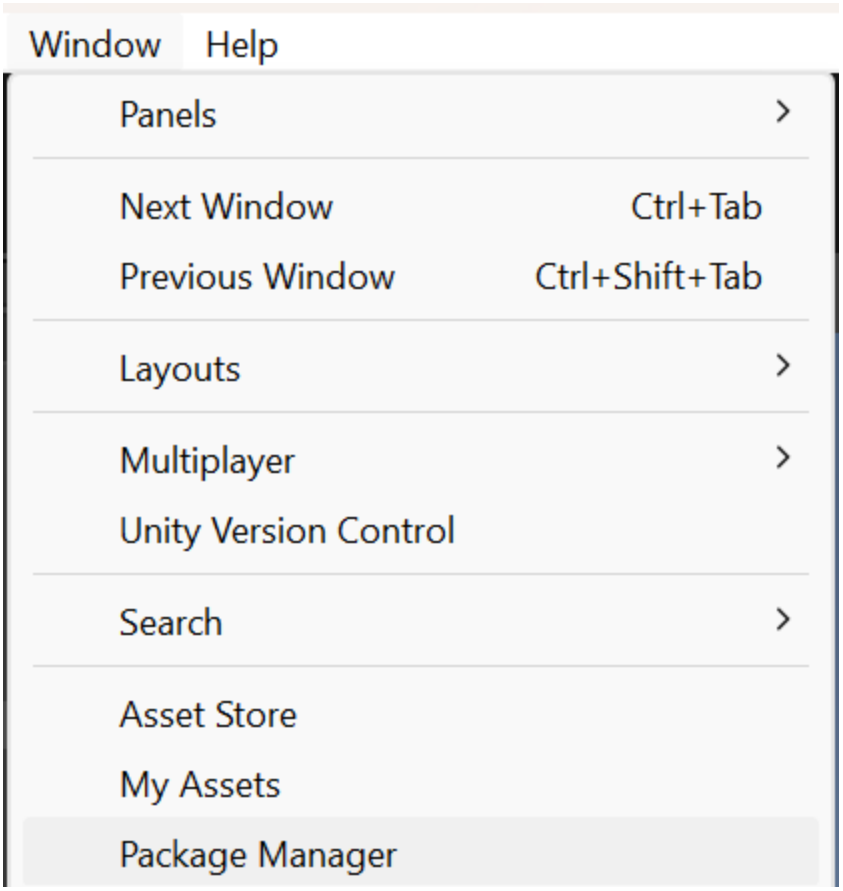
> Welcome and thank you for purchasing Last Virtual keyboard! This manual should lead you through the steps of setting up a virtual keyboard for your specific project. If you're looking for the API reference, it can be found Here

NOTE

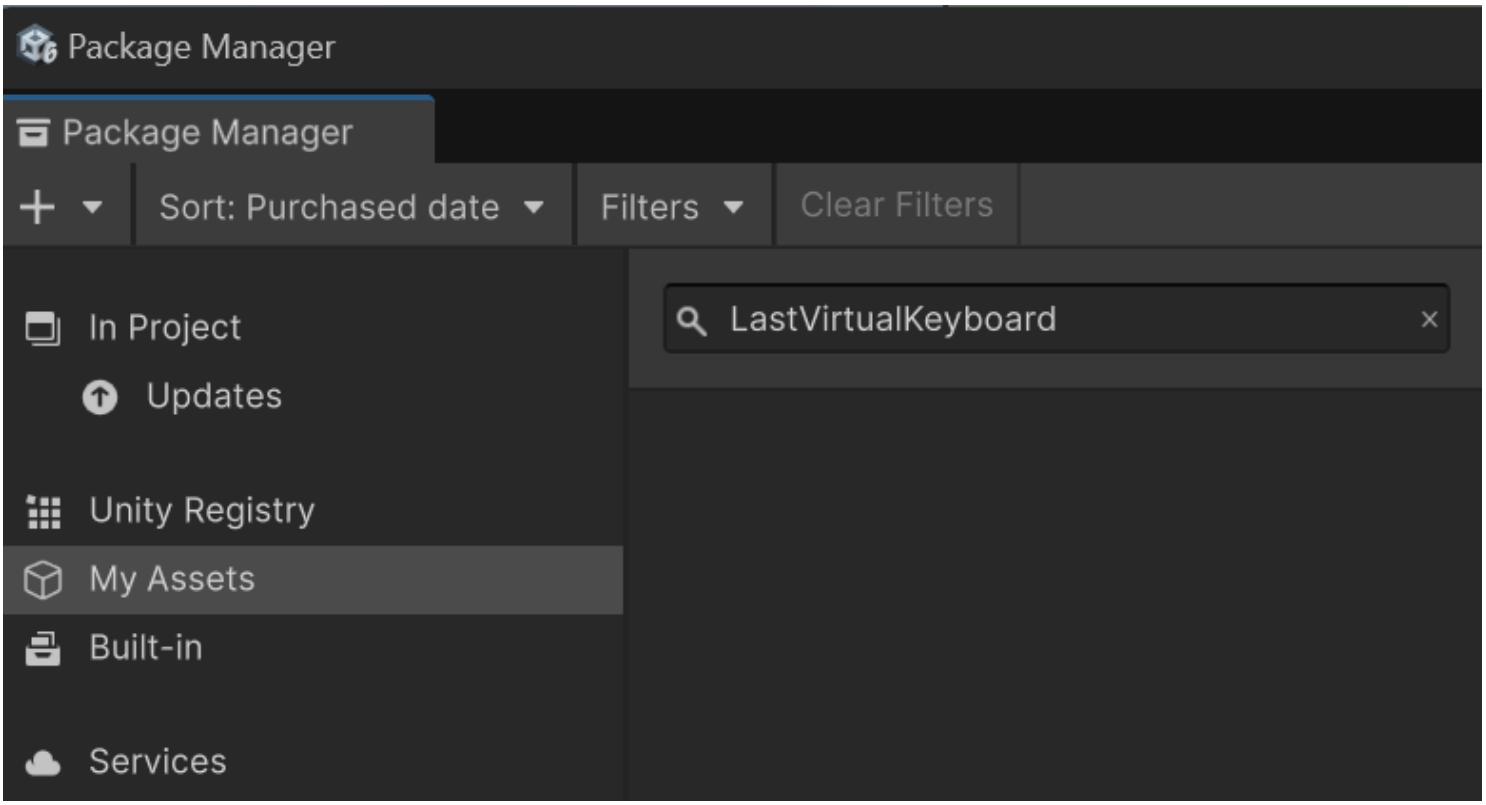
If you have any issues, please feel free to contact me at **lastassetscontact@gmail.com**

Importing into project

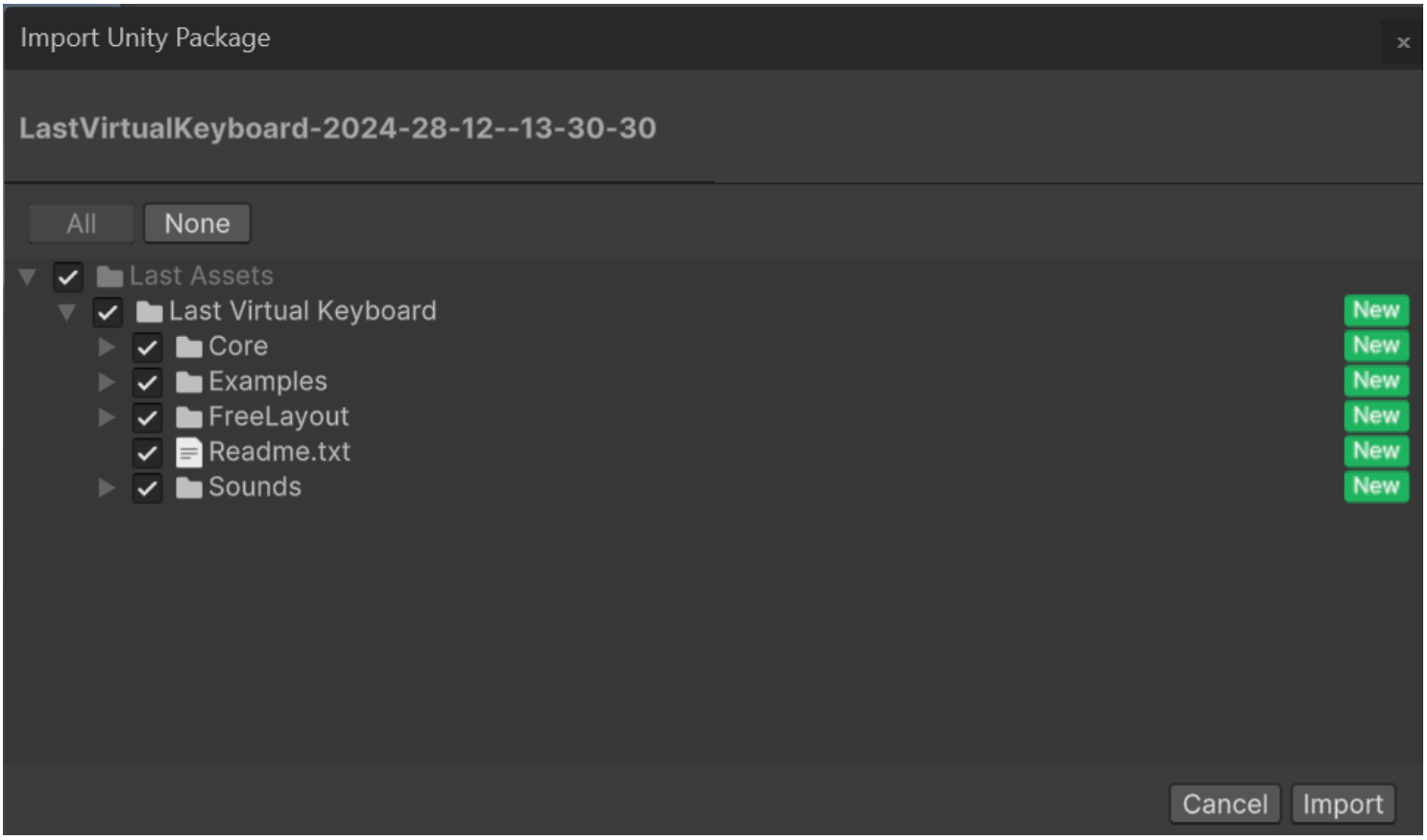
Once you've purchased the asset, open your project, and then open **Window/Packagemanager**.



Make sure youre logged in, and navigate to ***My Assets***. Search for ***Last Virtual Keyboard***.



You need to download and press import. You should be given options for what to import from the package. **Core** is necessary for functionality, and **Examples** have integrations that are worth looking into.



NOTE

One of the examples in the package relies on the new input system, so please follow the earlier procedure for downloading the **Input System** package from the **Unity Registry** under package manager.

Recommendations

> For the best usage example, I **HIGHLY** recommend you import the examples folder and study the provided demo. This provides an example script for connecting unity's new input system to the keyboard instance. This includes an example for custom key logic based on the string, as well as shortcuts to perform actions without interacting directly with a key.

Adding the components

> First step is to add the Keyboard component to a gameobject. This should automatically add the KeyboardLayout dependency, but if it doesn't, add this component to the object as well.

Customization

> The keyboard is fairly cusomizable, simply change the fields under *Layout* on the keyboardlayout component. This should give you options for colors, icons, etc. While you're here, populate the **Keys** array to start filling out your keyboard.

Errors

> You may see errors in console after adding the script to an object. This is due to dependencies that need to be assigned such as the audiosource and font. As you start filling out the references, these object reference errors should go away.

Using the keyboard

> Depending on which input system you're using, you can interact with the virtual keyboard by creating a reference to the VirtualKeyboard and utilizing these functions:
>
> Show() - This method enables the keyboard object for use on screen
>
> Hide() - This method disables/hides the keyboard object
>
> IsOpen() - Returns true or false if the keyboard is open or closed
>
> ReadKey() - "Presses" whatever key is highlighted under the virtual keyboard and returns the string

NOTE

For an implementation example, import the **Examples** folder, and open the **Demo_NewInputSystem** Scene. This should show you all the information you need to get your own keyboard going.