

On-Screen Keyboard for Unity

Overview

This asset offers a complete on-screen keyboard solution for Unity, ideal for use with TextMesh Pro's TMP_InputField. Users can type and interact with text fields through an on-screen keyboard, compatible with gamepad, VR, and touch inputs. Designed for platforms where a physical keyboard isn't feasible, such as mobile or VR.

Features

- **Gamepad Support:** Type and navigate the keyboard with a gamepad.
- **Shift & Caps Lock:** Switch between upper and lowercase letters.
- **Primary & Secondary Keys:** Keys support both primary (e.g., lowercase) and secondary (uppercase/symbol) characters.
- **Simple Setup:** Drag-and-drop prefabs with easy-to-use components.
- **Customizable Colors:** Adjust colors for selected keys and input fields.
- **Responsive Input Fields:** Automatically displays the keyboard when an input field is selected.
- **Direct Integration:** Designed specifically for TMP_InputField.
- **Prefabs Included:** Ready-to-use keyboard, input field containers, and demo scenes.
- **New Input System Compatible:** Utilizes Unity's Input System with custom actions for select and deselect.

Asset Components

Scripts

- **OnScreenKeyboard.cs:** Manages key presses, shift toggling, and keyboard visibility.
- **ManagedInputField.cs:** Controls interaction between TMP_InputField and the keyboard.
- **Key.cs:** Defines individual keys with primary and secondary character support.

Prefabs

- **OnScreenKeyboard Prefab:** Keyboard UI to place within the Canvas, typically at the bottom.
- **InputFieldContainer Prefab:** Test container for TMP_InputField interactions.
- **TestPage Prefab:** A demo scene with the keyboard and input fields pre-configured for testing.

Demo Folder

- Contains the TestPage scene, set up for quick testing of keyboard functionality.

Setup Instructions

1. Scene Setup

- **Add OnScreenKeyboard Prefab:**
Drag into the Canvas and position at the bottom. Ensure it overlays other UI elements.
- **Add Input Fields:**
Attach ManagedInputField to each TMP_InputField.
- **Use InputFieldContainer (optional):**
Add this prefab to test keyboard and input field interactions.
- **TestPage Scene:**
Open for a ready-to-use example setup.

2. Script Configuration

- **OnScreenKeyboard.cs:**
Configure key array and assign optional Shift/Caps Lock buttons.
- **ManagedInputField.cs:**
Handles keyboard visibility and input field interactions.
- **Key.cs:**
Set primaryValue and secondaryValue for each key to enable dynamic character swapping.

3. Input System Configuration

Unity's Input System is required. Install and configure if not already included.

Option 1: Drag and Drop

- Use the provided InputActions asset, located in the scripts folder, and assign it to the "Actions Asset" in the EventSystem GameObject.

Option 2: Adding Custom Input Actions

- **Submit Action:**
Binds gamepad Button South (A button).
- **Select Action:**
Binds gamepad Button South (A button).
- **Deselect Action:**
Binds gamepad Button East (B button).

Additional Customization Options

- **Button Customization:** Modify key values (primaryValue and secondaryValue) in Key.cs.
- **Keyboard Position & Size:** Adjust the RectTransform of the OnScreenKeyboard prefab as needed.

Example Scene Setup

1. Import Input System & TextMeshPro:

- Install via Package Manager.

2. Create Canvas & Add EventSystem:

- Ensure an EventSystem GameObject is present in the scene.

3. Position Keyboard:

- Place OnScreenKeyboard prefab in the Canvas at the bottom of the UI hierarchy.

4. Add TMP_InputField:

- Attach ManagedInputField to TMP_InputField.

5. Test in Scene:

- Enter Play Mode and test with gamepad input.

License

This asset is available on the Unity Asset Store. It's free for use in personal and commercial projects. Modify freely to fit your project's needs.

Get Started Today!

Quickly add a gamepad-friendly, customizable keyboard to your Unity project. Perfect for VR, mobile, and other keyboard-less platforms.