

MULTIPLE KEYPAD SYSTEM V1.4
DOCUMENTATION

Contents

Introduction.....	2
FAQ	3
ONLINE DOCUMENTATION.....	3
Manual Setup – Initial #1	4
Patch Notes	7
Contact	9
Online Documentation	9

MULTIPLE KEYPAD SYSTEM V1.4 DOCUMENTATION

Introduction

DO NOT MOVE OR REPOSITION KEYPAD CANVAS ONCE PLACED IN YOUR SCENE!

Multiple Keypad System which is a complete template for implementing unlimited styles of UI keypads. The system has 3 included variants: Modern, Scifi and full keyboard - These allow you to input a varied string code (Number or symbols) to the length of your choice. If matched by the player will allow some type of interaction. I have included a PBR Door and controller for testing!



The asset includes:

- Full functioning number or symbol input keypad with display output at the top, along with a clear, enter and close buttons all directly editable and customisable within the appropriate canvas
- Includes 3 simple keypad models, walls and pivot door
- Easy to change valid code, by changing a single element in the inspector
- Includes a basic door opening script, and integration with the default character controller
- Includes some basic sounds for key pressing, and invalid code input.
- Easy setup

FAQ

Q). Is there an example of this asset working?

A). Yes, you can open the **Keypad_Demo_Scene** to see the keypad asset or use this scene as your initial base of your project.

Q). How can I manually setup this asset?

A). See the manual setup instructions on [This Page](#) Make sure to remember to import the standard assets into your scene before starting or you may have errors!

Q). Why can't I see the UI or why is the UI in the wrong place?

A). that might be because you may have dragged some UI objects into the scene rather than onto the hierarchy. Make sure the X, Y, Z positioning of all parent objects (Which contain the UI's) are set to 0, 0, 0 in the inspector.

Q). Can I use this asset with a different character controller or door interaction?

A). This asset is intended for use with the provided Unity Standard Assets character controller and my included door interaction script provided but can be replaced by doing a couple of simple modifications. See [Extending the keypad asset](#) or send me an email and I'll be happy to help you out!

Q). How can I add more than one keypad to my scene?

A). Check the [Extending the keypad system](#)

Q). Do you want to organise the keypad elements into an empty GameObject?

A). When you create an empty GameObject, please make sure that the X, Y, Z coordinates of the empty GameObject are all 0, 0, 0 before dragging the parts into the GameObject.

Q). Can I use this asset with a 3rd person character with this asset?

A). Yes, all you need is to use the trigger event and add the **KeypadTrigger** script and attach the specific keyboard object which has the script **KeypadItem**

Q). My managers don't stay between scenes properly?

A). Take make these work best, they shouldn't be in an empty game object, but loose within the hierarchy. This is just how the DontDestroyOnLoad works, I only put them with a managers parent gameobject to keep it looking neat and tidy!

MULTIPLE KEYPAD SYSTEM V1.4 DOCUMENTATION

Manual Setup – Initial #1

QUICK SETUP: You can drag the **Entire_Keypad_DemoScene** into your game to get a really quick start, it will have everything ready to go!

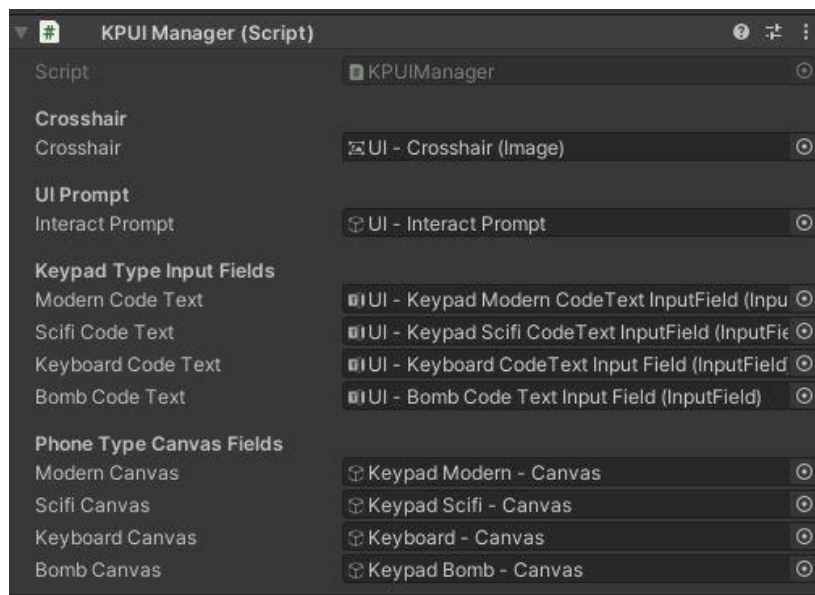
Tag Setup:

- Raycast: 3D keypad model should be tagged with: **Keypad**
- Trigger: 3D keypad model should be set to **Untagged**
- FPSController should be tagged **Player**

1). Place the **FPSController** prefab in your scene from the **Keypad System > Prefabs** folder

- On the **MainCamera** – Attach **KeypadInteractor** Script. (If not already).
- Set the **InteractDistance** on the raycast script
- You can set the **KeypadTag** here, by default it should be **Keypad**
- Set the **FPSController** object to the tag of **Player** (Can be changed in the **KeypadTrigger** script)

2). Add the **UIManager** object into the hierarchy and add the following references:



- Crosshair: Add the **UI - Crosshair** image from the child below
- InteractPrompt: Add the **UI - InteractPrompt** from the child below
- Keypad Input Fields: Add out of the **UI - Type - InputFields** from the children below
- Keypad Type Canvas: Add the canvas' to the correct slot

MULTIPLE KEYPAD SYSTEM V1.4
DOCUMENTATION

3). Add ONE **AudioManager** to the hierarchy:

- Add the ScriptableObject named **SO_AccessDenied**, **SO_ShortBeep** and so on from the SOs folder

4). Add ONE **Disable Manager** to the hierarchy:

- Make sure to add the **FPSController** object to this slot
- Add the KeypadInteractor script

5). Add ONE **InputManager** (Make sure to set the inputs you'd like)

6). You can choose to add one of the **Raycast_Prefabs** to your scene. (Please drag one of those out to the hierarchy! This should include: A phone controller, phone model, and various miscellaneous objects

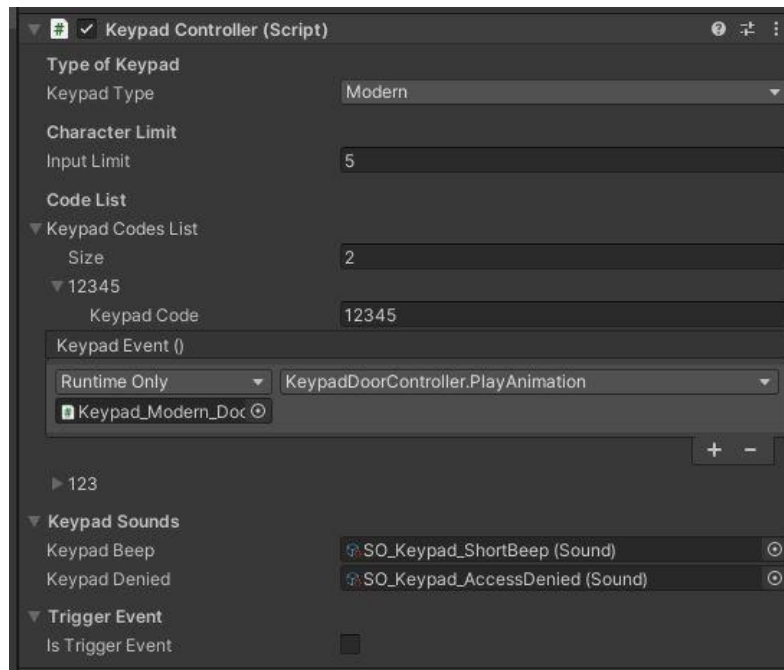
7). Open any of the Raycast or Trigger Phone collections and find the Keypad 3D model:

- Set tag: **Keypad** (If using for a FirstPerson setup)
- Has the **KeypadItem** script attached with the controller added to the slot
- Has a **Box Collider**



MULTIPLE KEYPAD SYSTEM V1.4 DOCUMENTATION

8). Take a look at the Controller script for that set of objects - See settings below:



Keypad Type - Choose: Modern, Scifi, Keyboard

Character Limit - Set character limit, based on the code you've just chosen

Code List

- Set the array size to add more code option
- Inside each option is a "Keypad Codes" to set the code you would like player to input
- Then an event to activate when this code is input into the keypad

Keypad Sounds - Add the ScriptableObject objects from the corresponding folder

Trigger Event - Set whether this is a trigger event version by selecting the check box, if true. You will get another option to add the trigger object

9). Unity Event (UNLOCK)

I have added a Unity event call for when you get the code correct

Click the little + icon in the bottom right, it will add a new event and you can place any gameobject which may have a script with a public method you want to call.

In this case I have added the door object, this has a script called **DoorController** which has a public method called **PlayAnimation**

NEED TO FIND MORE ANSWERS, I HAVE WRITTEN EXTENSION ONLINE DOCS HERE:

<https://speedtutoruk.gitbook.io/keypad-system-doc/>

Patch Notes

Version 1.4 - July 2023

- Added **11 new sounds effects** for various different interactions
- Added brand new Keypad Type: **C4 Explosive**
 - Added new bomb UI and integrated with current code
 - Added grid layout for ButtonContainer to keep it organised
- Removed custom inspector from **KeypadController**
- Refactored **KeypadController**
 - Added sound effect for initial interaction of the keypad
 - Added sound effects for correct code input
 - Added a split for opening and closing sounds
- **KPUIManager** to use every more efficiently
- **Keypad UI Canvas** Changes:
 - Renamed all UI elements for all canvases to match styles
 - Added new grid container groups for buttons for easier modification
- **DisableManager** fixes
 - Added null checks and log errors to the disable manager for better explanations
 - Disabled the **KeypadInteractor** script to avoid sound duplication issues
- Resized entire demo scene for 1.8m average human
- Renamed **KeypadRaycast** to **KeypadInteractor**
- Edited **AudioManager** to now use SO main object instead of names
- Made **Sound** script generic
- Fixed GUID issues by duplicating all scripts to avoid conflicts
- Applied root motion to all doors

Version 1.3 - July 2022

- Created new UIManager
 - Moved Crosshair UI from the raycast manager to the UIManager
 - Moved all UI's to UIManager and refined KeypadController
- Removed UI references from KeypadController
 - Custom Editor written for this script
 - Added new ScriptableObjects for audio
 - Added multiple interactions for codes. Unlimited events can be created
- Refined the DisableManager
 - Moved Crosshair UI from this and refined the system
- Created new ScriptableObjects for audio
- Renamed **KeypadItemController** to **KeypadItem**
- Updated **KeypadTrigger** script to remove UI elements and refinements
- Created a **RemoveTag** script so this can be added to an event, if you don't wish to interact with the keypad after a certain interaction

MULTIPLE KEYPAD SYSTEM V1.4 DOCUMENTATION

Version 1.2.3 – August 2021

- Documentation updates
- Change of **ValidCode** method to disable keypads when correct code is entered by disabling the script (Stops interaction)

Version 1.2.2 - June 2021

- Added a new Raycast so tags and layers aren't needed anymore
- Re-imported all keypad models to a 1, 1, 1 scale
- Added a way to close the keypad with a right click

Version 1.2.1 - March 2021

- o Removed Unity warnings from included scripts
- o Created a variable for door controller example - Meaning animation name can be easily changed in the inspector
- o Created a set of variables in the **KeypadController** so sounds can easily be changed via name based on the **AudioManager** strings
- o Refactored some basic elements of scripts to make them more optimised and efficient
- o Reduced asset file size by removing baked lighting data
- o Created new basic material for trigger event boxes, making them more visible
- o Updated display text material - Stops 3D being rendered over other 3D world objects
- o Updated package to Unity 2018.4 (As per Unity rules)

Version 1.2 – July 2020

- o Edited the canvas layout and the way it is display in Unity
- o Updated the Keypad Controller to incorporate **UnityEvents** to make successful code interaction multi-use, meaning you can produce multiple outcomes with ease
- o Added a new Audio Manager, so more sounds can be added and called within script
- o Added a new input manager, which allows easy control of all inputs from one place
- o Refactored the Raycast, KeypadController, ItemControllers and more!
- o Updated and edited sounds, textures and materials
- o Added new manager examples to the prefabs and demo scene
- o Update all prefabs and set them to 0, 0, 0 for future use
- o Changed tag type to only require “Keypad” on each item

Version 1.1 – June 2020

- o Added a trigger event (For Example: 3rd Person Controller Integration)
- o Created custom inputs for the raycast in the inspector for easy customisation
- o Refactored and renamed code to be more consistent and optimised throughout
- o Anchored all UI's for correct functionality

