Custom Home and Power Button Functions

## Applicable Products

Goblin; Pico Neo; Goblin 2

## Introduction of System Buttons Customization

Key Values Description:

|  |  |  |
| --- | --- | --- |
| Buttons | Key Value | Function |
| Home button | Single tap, Double tap, Hold; Disable | Return to launcher; Start an application; Return; Calibrate positive direction; Clean up memory; Notice; |
| Power button | Single tap | Return; Invalid; Normal (default) |

## Manual Modification Scheme

* + 1. Get config file “SystemKeyConfig.prop”; (File path: <https://github.com/PicoSupport/PicoSupport/tree/master/Assets/HomeKeyConfig>
    2. Open config file “SystemKeyConfig.prop” with notepad
    3. Modify Button Customization

The key value of home button contains "Tap", "Double - Tap" and "Hold", as follows:

*action\_home\_single\_tap* Single tap

*action\_home\_double\_tap* Double - Tap

*action\_home\_long\_press* Hold

The key values above can be set to 0/1/2/3/4/5/6/7. The corresponding response events for each key value are as follows:

*0: Invalid;*

*1: Start application under specified package name and class name ;*

*2. Back to Launcher;*

*3. Start settings;*

*4. Return;*

*5. No function, only send key notice;*

*6.* Calibrate *positive direction of vision;*

*7. Clean up memory.*

The key value of Power button contains single tap:

action\_power\_press Single tap

The key value can be set to: 0/1;

0: Invalid;

1: Return;

Change key value at the end according to the requirement, assign each key value to different function respectively;

*For example: Change single tap of Home key to be return function: Modify the key vaule to be "action\_home\_single\_tap=4".*

Set the double - tap Interval; Hold time duration, in milliseconds, as follows:

*time\_home\_double\_tap Double-tap logical interval*

*time\_home\_long\_press Long press logical interval*

Set *application needed to start under specified package name and class name*, as follows:

*action\_home\_XXX \_package=YYY*

*action\_home\_XXX \_class=ZZZ*

*XXX refers to single tap, double tap or hold, which are respectively corresponding to single\_tap、double\_tap、long\_press*

*YYY refers to name of the application needed to start*

*ZZZ refers to name of the class name*

* + 1. Open adb, copy the file to data/local/tmp catalog. Instructions are as follows:

adb push SystemKeyConfig.prop /data/local/tmp/

* + 1. Reboot device.

*Note: The custom of Home button only refers to the Home button on the helmet device; The Home button on the gamepad is not yet customizable.*

## Code Dynamic Modification Scheme

For the manual modification scheme mentioned in 3.3, you need to reboot device to apply the new customization after every modification of the profile. Code dynamic modification can implement system key screening and return to default system key.

Main principle is to place two config files with different names (DefaultKeyConfig.prop and UserKeyConfig.prop), one is about system key valid configuration (UserKeyConfig.prop), the other is about contents screening (DefaultKeyConfig.prop). According to different scene program to call config file, implement dynamic modification of System button. The specific procedure is as follows.

1. Get related configuration data: Https://pan.baidu.com/s/1bGNouE
2. Modify key configuration as required: Open "config file/userkeyconfig.prop", see the third step in 3.2;
3. Create a new Unity project, copy PicovrWifiManager\_vxxx.jar in Plugins->Android of Demo to Unity project’s corresponding catalog. Put the modified config file under any system catalog. Set the file name and path as parameters into parameter interface.

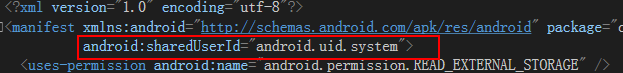
For example:

PicoUnityActivity.CallObjectMethod("androidSetUserKey", "/storage/emulated/0/download/UserKeyConfig.prop");

Note: Put UserKeyConfig.prop under system Download catalog.

1. Modify Androidminifest.xml.

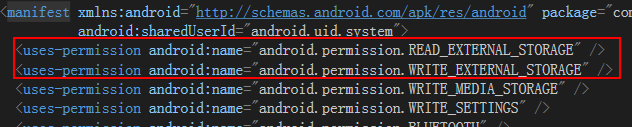
In manifest tags add: android:shareduserid= "Android.uid.system"



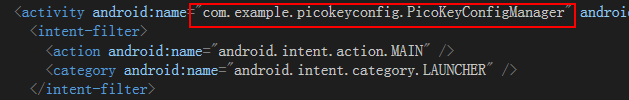
Add permissions (If needed):

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />



Modify main activity：android:name=" com.example.picokeyconfig.PicoKeyConfigManager"



1. Copy PicoUnityActivity.cs under Demo to any Unity project catalog.
2. Call Android interface,

For example: PicoUnityActivity.CallObjectMethod("androidSetUserKey", "path of the file");

Introduction of Interface

|  |  |  |  |
| --- | --- | --- | --- |
| Ordinal | **Interface** | **Description** | **Notes** |
| 1 | androidSetUserKey | Modify System Key Value | System Signature Required |