Keyboard and Mouse Recorder and Player 4.0

User manual

Keyboard and Mouse Input Recorder and Player (kmRecorderAndPlayer) is a Linux desktop application for recording sequences of keyboard and mouse input that can be saved to a file and be played at any time to simulate user input.

Dependencies

In order to run kmRecorderAndPlayer, your system needs to meet the following dependencies:

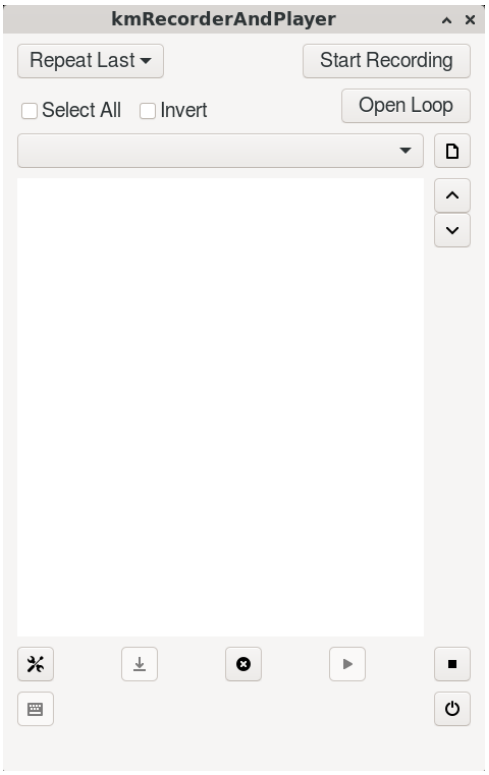
- imagemagick
- x11-utils

What's new?

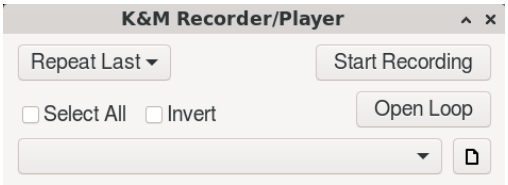
- Enable the use of native UTF8 charset from keyboar input.

GUI

kmRecorderAndPlayer has a minimalistic GUI:



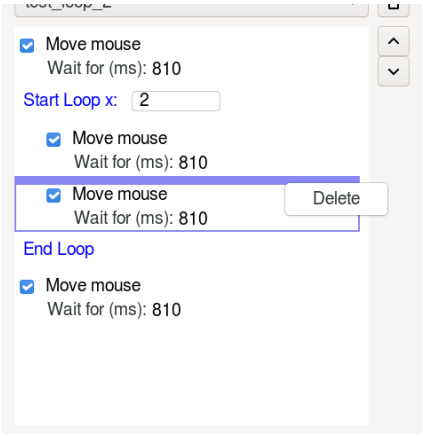
We can see three section:
Top section



Elements in this section:

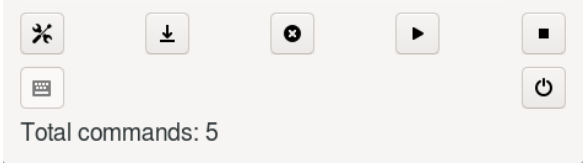
- drop-down button to select recording mode
- Start Recording button: after selecting the recording mode and start recording
- two checkbox to enable or disable input commands from the command list (middle section)
- Open Loop/Close Loop button: open and close a block in the command list. Commands inside this block will be repeated in a loop.
- Drop-down button to select command file previously save
- file manager button. Clicking it will open the list of saved files. One can rename or delete files.

Middle section



- The main area is the list of commands, this can be loaded from a command file or been added as they are recorded. New commands can be added at any time.
- Up/Down buttons to reposition commands in the list. This is useful in case one forgets a command and after added it at the end of the list, it can be moved to the right position in the list. To select a particular command in the list, hover over it and do left click on the blue top edge, once it has been selected, it can be moved up/down the list using the arrow buttons on the left. To unselect a previously selected command, just do left-click on the blue area of it.
- Right click on the blue area of a command listed, will select the command and display a context menu.

Bottom section

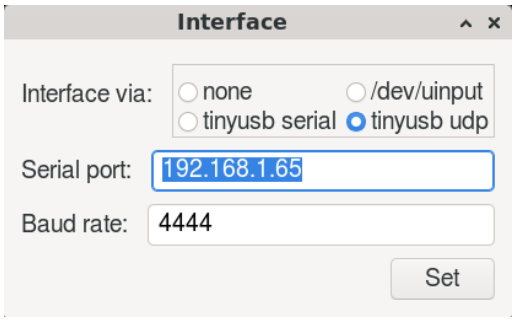


- Settings button
- Save to file button
- Clear command list
- Play commands in list
- Stop button
- Aux keyboard
- Shutdown button.
- Status bar

Interface Setting up

Before starting to record any input command for first time, you need to set up the interface for kmRecorderAndPlayer.

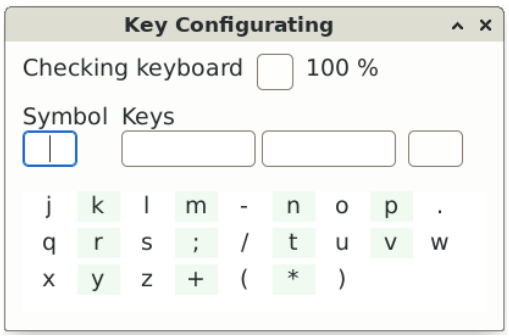
- Open settings
- Click Set Interface button this will display the following pop up



Please read Interface Methods for explanation of specifics.

Disclaimer: when using /dev/uinput you should be aware of the File Descriptor Hijack vulnerability (CVE-2023-34059). Notice that (CVE-2023-34059) Does NOT say that opening file descriptors to /dev/uinput will exposes your Linux OS, what it actually says is that if your system has already been compromised with a malicious actor that is able to exploit the exposure of a file descriptor to /dev/uinput, this could hijack such file descriptor and simulate user input.

After setting up the interface, the Key Configuration window will pop up:



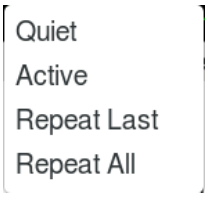
Please read the respective **Key Configuration** section for details.

How it works

To record an input command kmRecorderAndPlayer uses a translucence pop up that covers the full screen. On this pop up the user can call a context menu to select the kind of input command to add to the command list. In this way a command is recorded.
As mentioned above, this process is done over a translucence pop up. However, in some cases the translucence background of the pop up is replaced by a screenshot of the full screen after the last input command has been applied (this is done automatically) in this way the state of the screen, after the command is applied, is captured.
Bear in mind that despite our pop up will tried stay at the top of any other window, some elements will put themselves on top of this pop up anyway, in this case we need to bring our pop up to the top again, and we can do this by clicking anywhere in the visible part of the image of our pop up.

The reason for using the image of the full screen as background for our pop up is because some elements will go away as soon as they lose focus, therefore we take a screenshot of the screen and work on it as if it were the actual screen. For example if we want to record the sequence of inputs for selecting an item on a right-click menu. This will be clarified in the next section.

Recording Modes



- *Quiet:* will save the input commands as they are set.
- *Active:* will apply the command after it is have been added to the command list
- *Repeat Last:* apply the last command and take a screenshot of the screen after the command was execute. For example, if you do a right click that display a context menu, it would be impossible to record a command that select any item in the menu because the menu will disappear as soon as the context menu lost focus. For this reason kmRecorderAndPlayer will take a screenshot with the context menu and in this way the user can set any command on the image that contains the context menu.
- *Repeat All:* in this recording mode, kmRecorderAndPlayer will repeat all the commands on the list **from the point it was set**, and take a screenshot of the state of the screen after the last command was execute, so the user can set the new command on the image. Repeating all the commands is done every time a new command is added to the list.

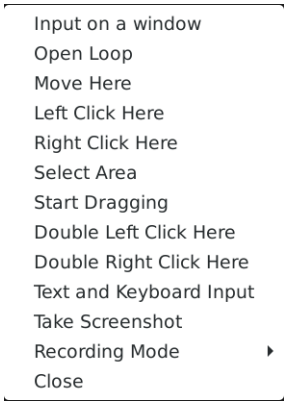
You can change recording mode at any time accordingly to the particular situation.

Notice: keep in mind that when using any of the interactive modes (Active, Repeat Last, and Repeat All) any window that might pop-up as a result of a command being executed, might appear on top of the *Input Blocker* window, in that case, just left-click on it, that will bring it back on top of all the windows.

Recording Input Commands

After pressing the *Start Recording* button this will happen:

1. the main GUI will minimise
2. a translucence pop up will cover the full screen. Right click will display the *recording context menu*

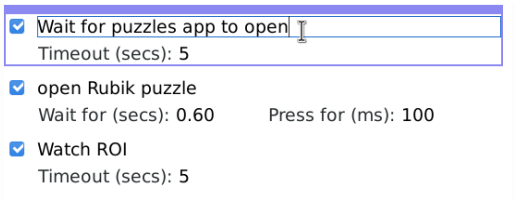


Elements in the context menu

- *Input on a window:* see full details in the Input in a Window section.
- *Open Loop/Close Loop:* same functionality than the button with the same name in the top section of the main GUI.
- *Move Here:* it will set a mouse command to move the mouse to the point where the right button was clicked (when Recording Context Menu was open)
- *Left Click Here/Right Click Here:* it will set a mouse input of left/right click mouse button at the same position of the cursor when open the context menu.
- *Select Area:* this item will let you to select a rectangle/area. After clicking on it, the menu is close and left click and drag (with button pressed) to select the desired area/rectangle. This will pop up a selection from which you can set if the selected area is: Select or Watch ROI (Control Command). If you selected Select, you will need to input the respective input commands to manipulate that area, like clicking on top and do a copy/paste/dragging.
- *Start Dragging:* after selecting this item, the context menu is closed, then you can do a left-click and drag the mouse and release it to the desire final position.
- *Double Left Click Here/Double Right Click Here:* Set a mouse double left/right button click at the position of the cursor when the context menu was called.
- *Text and Keyboard Input:* this option will display the *Aux Keyboard* which is used for all the keyboard input. See the respective section.
- *Take Screenshot*: will set a control command with the image of a particular window or full screen. More on Control Command section
- *Recording Mode:* Same functionality of the drop-down in the top selection of the main GUI.
- *Close:* close the context menu and return the control to the main GUI.

Command Edition

On the command list, each item has a checkbox and two text line. The checkbox enable or disable the respective command to be executed. The first line is the description of the command. You can open it for editing doing a right click on it following by a left click, a blue border will be display on the input field, to save the change just do *Enter* or a right click on it again and the blue border will disappear.



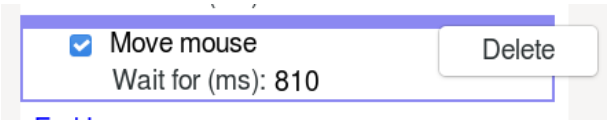
Each Input Command has a *Waiting for (secs)* section. It can be edited in the same way the description is edited. The values are in seconds and it means for how long to **wait after** the command has been executed before the next command can be executed.

Left/Right mouse button click has an additional customizible field ***Press for***, which means for how long the mouse button should be pressed before release it.

In the case of *Control Commands*, there is not *Waiting for,* instead they have *Timeout* which is the time the commands will check for their conditions to be satisfied before giving up. During this time, a Control Command will be checking for any changes in the area relevant for the command.

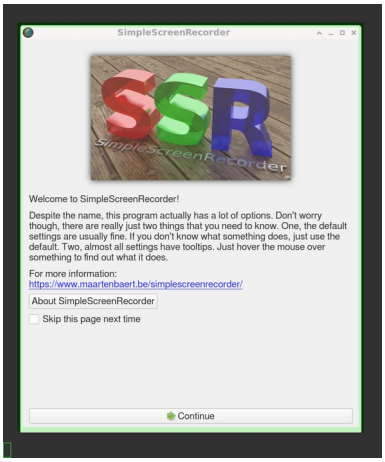
To edit a *Control Command*, open its context menu (right-click on the top blue area when hover on it).

To delete a command, select the specific command and do a right click with the pointer on top of the blue area, this will display a context menu with the “Delete” option.



Input in a Window

Input into a Window menu item will pop up a text input where the user will fill in with the name of any window currently open. On success it will display a small image of that window for the user to confirm. On failure it will display a warning. After confirmation the translucence pop up will draw a rectangle around the selected window. In the following image we had selected “SimpleScreenRecorder” as the target window:

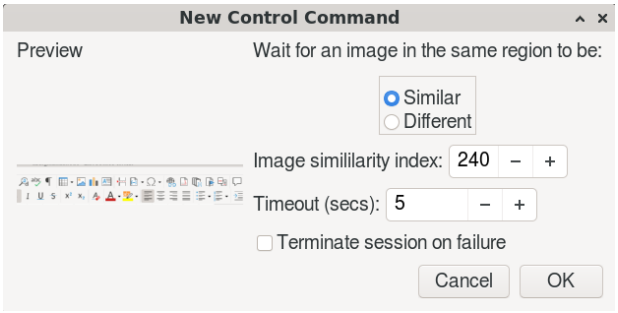


Now all the input commands will be on the context of the specific window selected. And those commands are independent of the position of the window. The *Recording Context Menu* will be only available over the specific window. Most of the functionality of the main *Recording Context Menu* are available on the local context of a specific window.

Control Command

The most important feature of *kmRecorderAndPlayer* is *Control Command.* In essence, a *Control Command* compares a region of the screen, (a region, full screen or specific opened window) with one image of the same region previously taken. The comparison can be similar or different. By default if the comparison fails, *kmRecorderAndPlayer* will stop and no more commands are executed. In this way we can control the way to apply input commands. For example to press a button on a pop-up, or when some process has been completed.

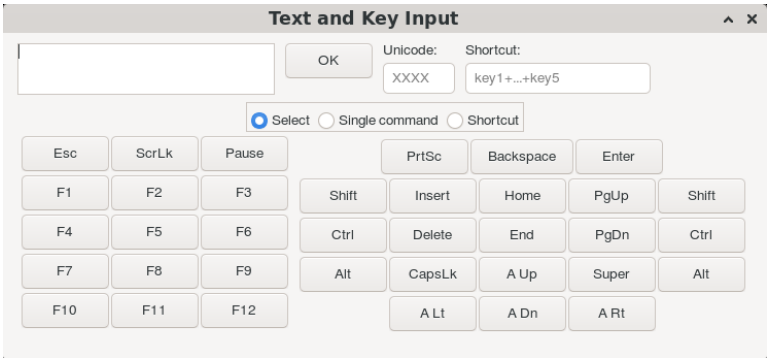
To set a *Control Command* from the Recording Context Menu we select Select Rectangle, after selecting an area, we select Watch ROI then we are presented with a dialogue to set some properties



- *Preview:* the preview of the saved image.
- *Wait for an image…:* select Similar if the command should wait for the selected area to be similar to the saved image. Select Different otherwise.
- *Image similarity index:* one should think of it as how different or similarity the selected region should match the saved image. The range goes from 0 (for identical images pixel-wise) to 441(matching any image). However, the size of the selected area have an impact on it. So this value should take into account the size of the image. Thus, in the case of similar images, the smaller the image, the smaller the Image Similarity Index. For example similarity of areas like 20x20 will need something like in the order of 80. The default is 240 which seen fair for similar image. This would be a good value to start.
- *Timeout:* for how log the command should wait before giving up.
- *Terminate session on failure:* check this box if you want to stop *kmRecorderAndPlayer* playing any command after it in case the command fails.

Text and Key Input Pop up

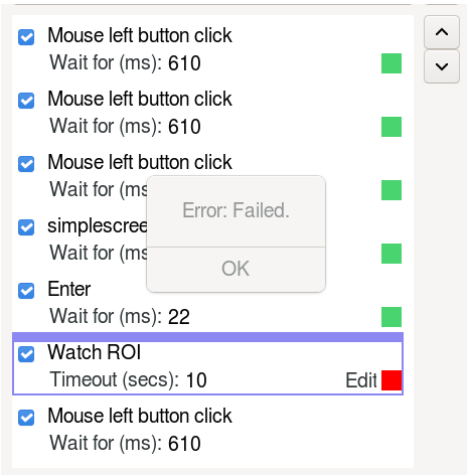
To input text commands, shortcuts, unicode we need to use the Aux Keyboard pop up



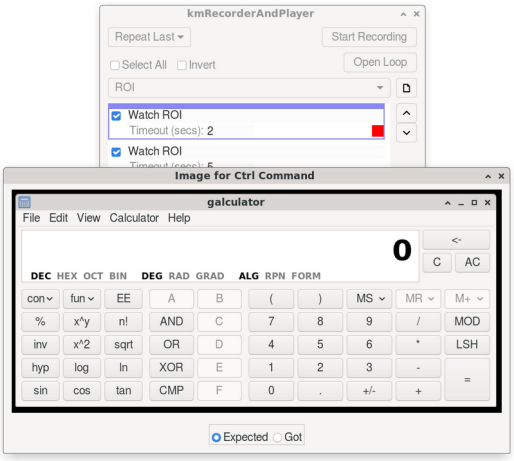
- top left corner: text input area, to set the command click *OK* button
- Unicode field: in *code point* format WITHOUT the leading *U*. For example the unicode for ⁋ (reversed pilcrow sign) is U+204B, we ONLY input 204B and press ENTER in the physical keyboard to submit the command.
- Shortcut field: select *Shortcut* from the options below and use the buttons in the pop up for any of the respective physical keys and physical keyboard keys for any other alpha-numeric character, anything not matching this is ignore.
- Single key commands: select *Single command* from the options and then press the corresponding button in the Pop up.
- Note about Tab: for the moment Tab can be set via the text field.

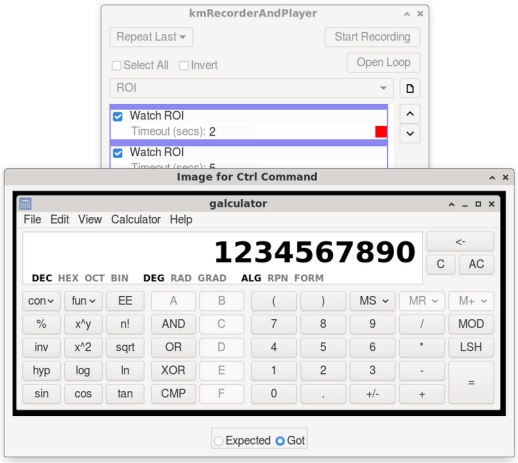
Reporting

After playing the commands, we can see which commands were applied successfully and those that failed.



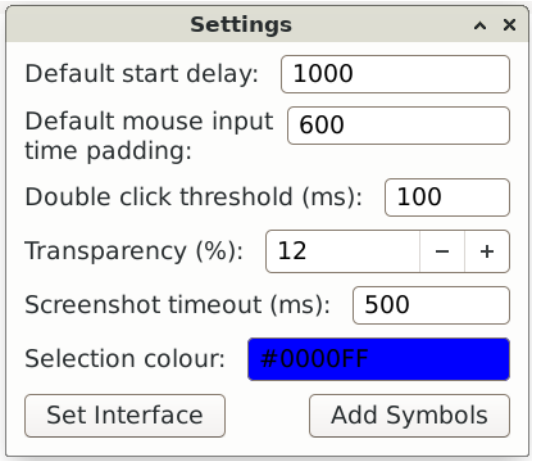
On failure a command will display a red squared on the right. Doing left-click on it will show an error message. In the case of *Control Commands*, left-click on the red square will bring a pop-up with the image that was expected and the image it got:





This can help us to calibrate the settings for that command.

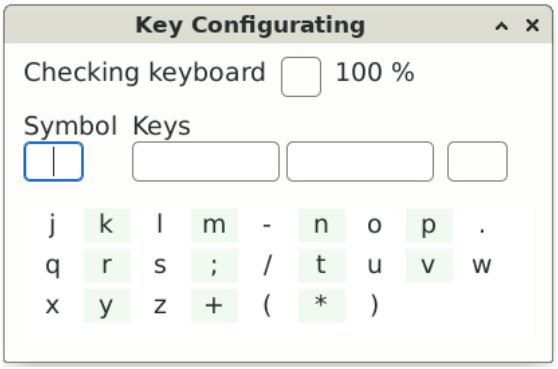
Settings



- *Default start delay:* time delay for starting the first command in the list, in milliseconds.
- *Default mouse input time padding:* default milliseconds set as a waiting time for mouse input commands.

- *Double click threshold:* the maximum time (in milliseconds) between two consecutive mouse button (left or right) clicks to be consider as double mouse button click (left or right). This time should be the same or less of your system settings.
- *Transparency:* transparency level for the input blocker pop up.
- *Screenshot timeout:* timeout for taking screenshot to be use by the input blocker pop up.
- *Selection colour:* colour of the brush use for the selection in html format #rrggbb.
- *Set Interface: Interface Methods.*
- *Add Symbol:* clicking this button, will display the Key Configuration pop up window

Key Configuration



After the initial configuration you will be able to add any symbol that can be displayed using combination of your keyboard's keys. Note that this depends on system fonts, language and locale set in your system. For instance the characters: á, ñ (Spanish) or ß, š (Slovakian) can be typed using a key combination.

Let's use few examples to explain how to add a new symbol (character).

1. For British keyboard. To add the character £ we do the following:
   - Type £ in Symbol field
   - In the first Keys fields we press, ONE BY ONE, the keys we pressed to display £, in this case, we press and release the keys <shift> and 3. Those keys will be displayed as **
   - Press Enter.
2. If using Spanish keyboard, to add the symbol Á:
   - Type Á in the Symbol input field
   - In the first Keys fields we press, ONE BY ONE, the keys we pressed to display Á: <apostrophe> <shift> a. The Keys field will display: *** (one * for each pressed key)
   - Enter
3. Some characters need at least two series of key combinations. For instance in a Slovakia keyboard to display the character Š we do the following:
   - In the Symbol input field we type Š as you normaly would do it. In the case my keyboard I press: <alt><shift><apostrophe> (and then) <shift> f
   - In the first Keys input field we type (one by one, pressing and releasing) <alt><shift><apostrophe>
   - In the second Keys input field we type (again one by one, pressing and releasing): <shift> f
   - Enter

Commandline

We can pass an argument to *kmRecorderAndPlayer* when called as command line. The argument should be the name of a input command file previously recorded:

./kmRecorderAndPlayer Test_app_gui

will run the file *Test_app_gui*