JAutoClicker

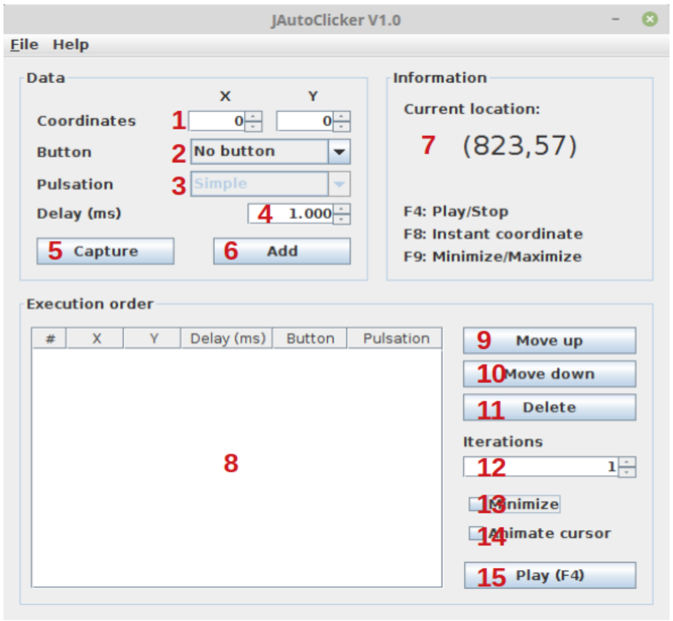
v1.0

It is a program capable of performing tasks using the cursor automatically. It is customizable and compatible with various operating systems such as Windows, GNU / Linux and Mac.

Featured functions.

- Perform tasks using the system cursor (movements and clicks).
- Allows capture of coordinates and buttons by pressing.
- It has keyboard shortcuts for fundamental functions: Play/Stop actions, capture instant coordinates and Minimize/Maximize the program window
- Save and recover programmed actions through the use of files.

Description of the interface



JAutoClicker V1.0

Data

	X	Y
Coordinates 1	0	0

Button 2 No button

Pulsation 3 Simple

Delay (ms) 4 1.000

Information

Current location: 7 (823,57)

F4: Play/Stop
F8: Instant coordinate
F9: Minimize/Maximize

Execution order

#	X	Y	Delay (ms)	Button	Pulsation

Iterations 12 1

Developed by Roger Lovera
E-Mail: rloverab@yahoo.es

1. **Coordinates:** Indicates the position in which the cursor must move.
2. **Button:** Indicates the button that must be pressed in a certain position. The available options are:
   - *Left:* Left mouse button.
   - *Center:* Middle mouse button.
   - *Right:* Right mouse button.
   - *None:* Does not make any keystrokes. It is useful for situations in which only one cursor is required to move from one position to another.
3. **Pulsation:** Indicates the type of pulsation that should be applied. The available options are:
   - *Simple:* Make a simple press of a button (click).
   - *Double:* Perform a double and rapid press of a button (doubleclic).
   - *Pressed:* Perform a long press of a button without releasing it (useful for drag and drop actions).
   - *Release:* Release a button that has an active long press (useful for drag and drop actions).
4. **Delay (ms):** Indicates the time that must elapse before executing a certain action. This value must be between 1 and 60,000 (both inclusive). The time is represented in milliseconds.
5. **Capture:** Capture the coordinates of the cursor with the press of a button and also capture the button pressed.
6. **Add:** Take the supplied data of the cursor action (1, 2, 3 and 4) and add it to the list of Execution Orders (8).
7. **Current location:** Displays in real time the current position of the cursor.
8. **Execution order:** It is a list of the actions that the cursor must perform. The execution of said actions will be carried out in the same order in which they are shown in it.
9. **Move up:** Moves some positions up one or several selected actions.
10. **Move down:** Move some positions down one or several selected actions.
11. **Delete:** Remove one or more selected actions from the order list.
12. **Iterations:** Indicates the number of times that the Execution Order list should be played.
13. **Minimize:** When this option is checked, the program window will be minimized when the actions begin to play. Otherwise, the window will remain maximized.

Developed by Roger Lovera
E-Mail: rloverab@yahoo.es

14. **Animate cursor:** When this option is marked the cursor will move pixel by pixel from one position to another, the duration of the animation is determined by the delay established in the actions (4) and the control of the cursor is lost during the execution of actions. Otherwise the cursor will move from one position to another instantaneously and the cursor control is not completely lost.
15. **Play (F4):** Starts and stops the execution of certain actions. The F4 key has the same function as the mentioned button, very useful if the options Minimize (14) and Animate cursor (15) are activated.

First steps.

First the actions are defined one by one by setting their respective parameters (1, 2, 3 and 4) in the data space, each programmed action must be added to the Execution order list (8) using the Add button (6).

After defining the actions to be executed, they are launched using the button Play (F4) (15).

Developed by Roger Lovera
E-Mail: rloverab@yahoo.es