	Document Title	Application Guide
	Application Name	Macro-Maker
	Version	1.0
	Written By	Vlad Feldfix

1. RUN

1.1. This application will automatically complete a set of instructions written in a script file.

1.2. Once the RUN button is pressed, don't touch the computer until the program is finished, unless specified otherwise by the script file.

1.3. When the algorithm is done press ENTER.

2. EDIT LIST

2.1. A .csv file called "List.csv" serves as a database. It can be useful if you need to pull data during the automatic procedure.

2.2. Before every automation, you need to update this file, depending on what you need the program to do.

2.3. The .csv file will be open in your local .csv file editing software such as Microsoft Excel, Libra Office, Windows Notepad, or whatever other software else you are using.

2.4. This application does not provide you with a .csv file editing software.

2.5. This function is optional, it might not even be necessary, depending on the script you are running on it.

3. EDIT SCRIPT

3.1. This is the most important part of Macro Maker. The script file. "script.txt" is the place where you write exactly what you want your application to do. Here is a list of possible commands:

3.2. MOUSE_POS_SET(X, Y)

3.2.1. Sets the new position of the cursor.

3.2.2. **X** new x cursor position.

3.2.3. **Y** new y cursor position.

3.3. MOUSE_CLICK(BUTTON)

3.3.1. Click the mouse button.

3.3.2. **BUTTON** possible options for this argument are: left, right, and middle.

3.3.3. So for example, if you need to click a button on the screen you can write:

```
MOUSE_POS_SET(158,125)
MOUSE_CLICK(left)
```

3.4. MOUSE_DOWN(BUTTON)

3.4.1. The difference between MOUSE_CLICK and MOUSE_DOWN is that MOUSE_DOWN clicks without releasing, whilst MOUSE_CLICK clicks and releases immediately.

3.4.2. This command together with MOUSE_UP can be used to drag and drop.

3.4.3. **BUTTON** See argument BUTTON for the MOUSE_CLICK command.

3.5. MOUSE_UP(BUTTON)

3.5.1. This command releases the mouse button. See MOUSE_DOWN to learn more

3.5.2. **BUTTON** See argument BUTTON for the MOUSE_CLICK command.

3.6. KEYBOARD_CLICK(BUTTON)

3.6.1. This command clicks a given keyboard button.

3.6.2. **BUTTON** This argument can take many values such as alt, shift, enter, home, etc, up, down, etc. For more information see Python library: keyboard 0.13.5.

3.7. KEYBOARD_PRESS(BUTTON)

3.7.1. Similarly to the mouse functions, this command presses (without releasing) any key on the keyboard. This can be used together with the KEYBOARD_RELEASE command to click two keys at the same time. For example:

```
KEYBOARD_PRESS(ctrl)
KEYBOARD_PRESS(s)
KEYBOARD_RELEASE(s)
KEYBOARD_RELEASE(ctrl)
```

3.7.2. **BUTTON** see KEYBOARD_CLICK command.

3.8. KEYBOARD_RELEASE(BUTTON)

3.8.1. This command releases a pressed keyboard button.

3.8.2. See more information in the KEYBOARD_PRESS command.

3.8.3. **BUTTON** see KEYBOARD_CLICK command.

3.9. SLEEP(SECONDS)

3.9.1. Pauses for a given number of seconds. This can be useful if you need to let something load in the middle of the automation, for example, a web page.

3.9.2. **SECONDS** the number of seconds to wait.

3.10. SET_ALARM()

3.10.1. This command allows the user to set an alarm when they need the automation to start (or continue) working.

3.10.2. The alarm command will ask the user to provide a date and time for when the alarm should go off.

3.10.3. Make sure that the application is open until the time the alarm is set.

3.11. INPUT(PROMPT)

3.11.1. Asks the user to input data. It can also be useful to display a message/warning.

3.11.2. **PROMPT** This argument is the message that is displayed for the user on the console screen of the application.

3.11.3. Here is an example of how can this command be used:

3.11.4. INPUT(Hello user, please make everything ready and click ENTER to continue...)

3.11.5. INPUT(Insert your birth date as YYYY-MM-DD)

3.12. LOOP(CYCLES)

3.12.1. You can run automation in a loop X times

3.12.2. **CYCLES** is the number of times the loop will run until the command END_LOOP.

3.12.3. The CYCLES argument can take 3 possible values:

3.12.3.1. a number 1, 2, 3, 4, …

3.12.3.2. LIST If you set argument CYCLES to LIST, the loop will go for every item in the list file List.csv

3.12.3.3. INPUTDATA if you set argument CYCLES to INPUTDATA, the user will choose how many times the loop will cycle.

3.13. END_LOOP()

3.13.1. Every loop must be completed by the END_LOOP command.

3.13.2. Here is an example of a loop:

```
LOOP(15)
MOUSE_CLICK(left)
END_LOOP()
INPUT(Set the number of times to loop)
LOOP(INPUTDATA)
MOUSE_CLICK(right)
END_LOOP()
LOOP(LIST)
MOUSE_CLICK(middle)
END_LOOP()
```

3.13.3. In this simple example, there are 3 loops. The first one will make the mouse click left 15 times. The second loop will make the mouse click right as many times as the user inputs. the third loop will make the mouse click the middle for as many lines as there are in the list.

3.14. KEYBOARD_WRITE(TEXT)

3.14.1. This command will input a pre-written text. This text can be given in the code or taken from the list.

3.14.2. **TEXT** This argument is the text the automation will write. It can be a per-written text or a list value. To make it a list value set this argument to LSTVAL#X where X is the column of the list you want to take information from. Another possible value can be INPUTDATA where the keyboard will write the user input taken from the INPUT command (see INPUT command for more info).

3.14.3. Here is an example of automation with the KEYBOARD_WRITE function:

3.14.4. Consider the example when the List.csv file is filled as follows:

```
row11,row12,row13
row21,row22,row23
row31,row23,row33
```

3.14.5. Open Macro Maker.

3.14.6. Press 3 to activate the EDIT SCRIPT function.

3.14.7. Insert the following code:

```
INPUT(Now open notepad and press ENTER
to
continue)
SLEEP(5)
LOOP(4)
KEYBOARD_WRITE(Hello World)
KEYBOARD_CLICK(enter)
END_LOOP()
LOOP(LIST)
KEYBOARD_WRITE(From List:)
KEYBOARD_CLICK(enter)
KEYBOARD_WRITE(LSTVAL#0)
KEYBOARD_CLICK(enter)
KEYBOARD_WRITE(LSTVAL#1)
KEYBOARD_CLICK(enter)
KEYBOARD_WRITE(LSTVAL#2)
KEYBOARD_CLICK(enter)
END_LOOP()
```

3.14.8. Save and close the file.

3.14.9. Press ENTER to restart.

3.14.10. Press 1 to RUN

3.14.11. If you follow all the instructions correctly you are supposed to

3.14.12. have an open notepad with the following text:

```
Hello World
Hello World
Hello World
Hello World
From List:
row11
row12
row13
From List:
row21
```

```
row22
row23
From List:
row31
row23
row33
From List:
row31
row23
row33
```

4. SETTINGS

4.1. Fill the file settings.txt according to the provided example:

4.2. Script Location > script.txt

5. HELP

5.1. Open the help file

6. EXIT

6.1. Exist the application