

## Разработка макроса для управления мышью на python

*Андрюенко Иван Сергеевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### Аннотация

В данной статье описывается процесс разработки макроса для управления мышью. Для разработки программы использовался язык программирования python и модуль pynput. Результатом исследования является разработанный макрос, а также описание его работы.

**Ключевые слова:** Python, макрос, автокликер, pynput, мышь.

## Developing a macro for mouse control in python

*Andrienko Ivan Sergeevich*

*Sholom-Aleichem Priamursky State University*

*Student*

### Abstract

This article describes the process of developing a macro for mouse control. The python programming language and the pynput module were used to develop the program. The result of the study is a developed macro, as well as a description of its work.

**Keywords:** Python, macro, autoclicker, pynput, mouse.

## 1 Введение

### 1.1 Актуальность

В последнее время использование макроса набирает большую популярность. Макрос — это действие или набор действий, которые можно выполнить сколько угодно раз. Данная программа способна заменить человека для выполнения циклической работы мышью или клавиатурой. Зачастую, макрос используют для: ускорения часто выполняемых операций редактирования или форматирования, объединения нескольких команд, например, вставки таблицы с указанными размерами и границами, автоматизации обработки сложных последовательных действий в задачах и т.д.

### 1.2 Обзор исследований

В своей работе В.Д. Тромза описала создание макроса для программы Компас-3D, автоматизирующий процесс создания осевых линий между параллельными и равными прямыми [1]. В.Ю. Ильичев описал процесс и результаты разработки методики управления виртуальными моделями

роботов в симуляторе программы FreeCAD при помощи программ (скриптов, макросов), написанных на языке Python [2]. В своей работе А.А. Пермяков, Л.Ю. Забелин рассмотрели несколько примеров применения языка программирования Python (версия 3+) для оптимизации и расширения стандартных возможностей такой среды трехмерного моделирования как Cinema 4D с помощью макросов [3]. Д.В. Дудукало, М.С. Чепчуров в своей работе описали использование программы, написанной на языке Python, возможность гибкой настройки входных параметров, таких как обороты заготовки и инструмента, размерные параметры заготовки и инструмента, а также начальные углы касания [4].

### 1.3 Цель исследования

Цель исследования – создать макрос для управления мышью.

## 2 Материалы и методы

В Процессе создания макроса использовалась среда программирования PyCharm Community Edition 2022.1.3, язык программирования python и модуль pynput.

## 3 Результаты и обсуждения

Перед началом разработки макроса необходимо установить модуль pynput. Pynput – это модуль, используемый для управления движениями мыши. С помощью команды `pip install pynput` скачиваем модуль через терминал (рис. 1).

```
(venv) PS C:\Users\andrv\PycharmProjects\macros> pip install pynput
Collecting pynput
  Using cached pynput-1.7.6-py2.py3-none-any.whl (89 kB)
Collecting six
  Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: six, pynput
Successfully installed pynput-1.7.6 six-1.16.0
WARNING: You are using pip version 21.3.1; however, version 23.0 is available.
You should consider upgrading via the 'C:\Users\andrv\PycharmProjects\macros>
(venv) PS C:\Users\andrv\PycharmProjects\macros> |
```

Рисунок 1 – Установка модуля

После установки модуля импортируем его. Button и Controller необходим для управления мышью, а Listener и KeyCode для отслеживания клавиатуры, запуска и остановки макроса (рис. 2).

```
1 import time
2 import threading
3 from pynput.mouse import Button, Controller
4 from pynput.keyboard import Listener, KeyCode
```

Рисунок 2 – Импорт модулей

Создадим переменные. Переменная `delay` будет отвечать за задержку между выполнением задачи, прописанной в макросе, `button` – кнопка, которую будет нажимать программа, `start_stop_key` и `exit_key` будут содержать в себе клавиши необходимые для запуска и выключения макроса (рис. 3).

```
6 delay = 0.001
7 button = Button.left
8 start_stop_key = KeyCode(char='s')
9 exit_key = KeyCode(char='e')
```

Рисунок 3 – Создание переменных

Далее создадим класс для расширения потоков. Созданный поток позволит управлять щелчками мыши. Есть два параметра – задержка и кнопка. Есть также два значения о том, запущена программа или нет. «`__init__`» – это встроенная функция в Python, которая вызывается всякий раз, когда создается объект. «`__init__()`» инициализирует состояние объекта, то есть это место, где возможно определить начальное или первичное состояние нашего объекта. Вызываем «`super`» для наследования базовых классов. Также добавим методы для внешнего управления потоками (рис. 4).

```
12 class ClickMouse(threading.Thread):
13     def __init__(self, delay, button):
14         super(ClickMouse, self).__init__()
15         self.delay = delay
16         self.button = button
17         self.running = False
18         self.program_run = True
19
20     def start_clicking(self):
21         self.running = True
22
23     def stop_clicking(self):
24         self.running = False
25
26     def exit(self):
27         self.stop_clicking()
28         self.program_run = False
```

Рисунок 4 – Создание необходимого класса и методов

Создадим цикл. «program\_running» выполняется в цикле, пока значение не станет истинным, а также создает другой цикл внутри существующего цикла, где он проверяет, установлен ли для running значение true или false. В случае нахождения внутри обоих циклов, будет произведен переход в режим ожидания для заданной задержки.

```
30 def run(self):
31     while self.program_run:
32         while self.running:
33             mouse.click(self.button)
34             time.sleep(self.delay)
35             time.sleep(0.1)
36
```

Рисунок 5 – Создание цикла

Далее создадим экземпляры для контроллера мыши. Созданный поток «Mouse\_click» запустится, когда пользователь попадет в цикл в методе run (рис. 6).

```
37
38 mouse = Controller()
39 thread = ClickMouse(delay, button)
40 thread.start()
41
```

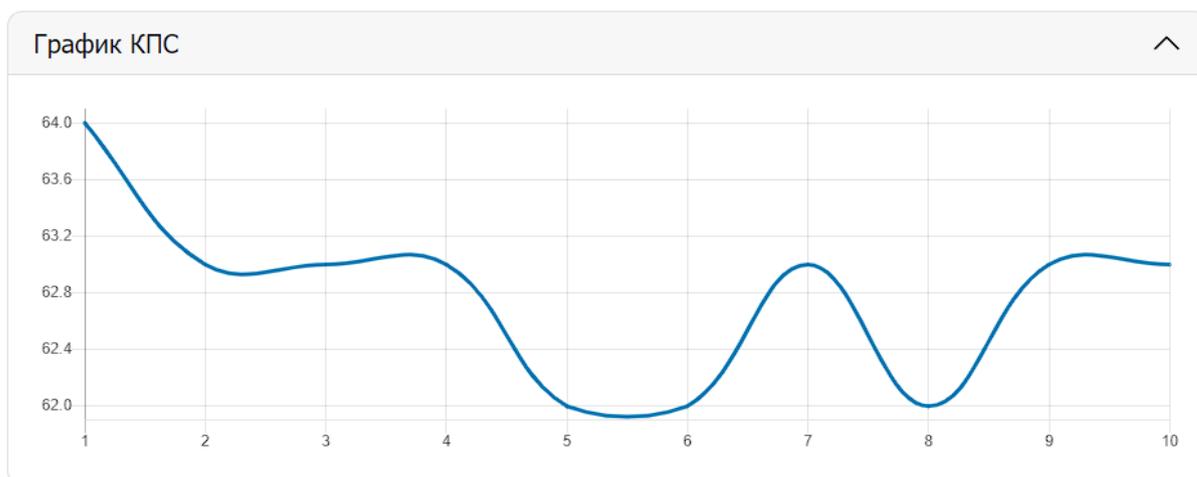
Рисунок 6 – Создание экземпляров для контроллера мыши

Создаем метод «on\_press», который принимает ключ в качестве аргумента. Клавиша start\_stop\_key совпадает с клавишей start при ее выполнении. Затем щелчок должен быть прекращен, когда значение running установлен в True в потоке. Метод Exit вызывается в методе, если выполняется клавиша exit и останавливает прослушиватель (рис. 7).

```
43 def on_press(key):
44     if key == start_stop_key:
45         if thread.running:
46             thread.stop_clicking()
47         else:
48             thread.start_clicking()
49     elif key == exit_key:
50         thread.exit()
51         listener.stop()
52
53
54 with Listener(on_press=on_press) as listener:
55     listener.join()
56
```

Рисунок 7 – Создание метода «on\_press»

Запускаем и проверяем макрос. При нажатии клавиши «s» макрос начинает работу и производит большое количество щелчков левой кнопкой мыши в секунду. Для отключения программы достаточно нажать клавишу «e». С помощью онлайн сервиса было выявлено, что макрос в среднем делает 63.5 нажатий в секунду (рис. 8).



## Ваши результаты

Ваш максимальный результат: 63.5 клик/с

Рисунок 8 – Проверка макроса

## Выводы

В данной работе был описан процесс разработки макроса. Для разработки программы использовался язык программирования python и модуль ruprut. В результате был получен макрос для управления мышью, делающий 63,5 клика в секунду.

## Библиографический список

1. Тромза В.Д. Макрос для программы компас-3д на языке python. // Инновационное развитие современной науки: проблемы, закономерности, перспективы. сборник статей III Международной научно-практической конференции. 2017. С. 18-20.
2. Ильичев В.Ю. Создание скриптов python для управления роботами в симуляторе программы freescad // Заметки ученого. 2021. № 11-1. С. 181-184.
3. Пермяков А.А., Забелин Л.Ю. Python в среде 3d-моделирования cinema 4d // Обработка информации и математическое моделирование. Материалы Российской научно-технической конференции. 2017. С. 248-251.
4. Дудукало Д.В., Чепчуров М.С. Моделирование получения профиля поверхностей цилиндрических деталей в программе freescad // Международная научно-техническая конференция молодых ученых. Белгород, 2020. С. 3584-3588.
5. Python Documentation contents URL: <https://docs.python.org/3/contents.html>