

Inicio (<https://recursospython.com/>)

Códigos de fuente (<https://www.recursospython.com/category/codigos-de-fuente/>)

Guías y manuales (<https://www.recursospython.com/category/guias-y-manuales/>)


Foro (<https://foro.recursospython.com/>) Micro (<https://micro.recursospython.com/>)

Tutorial (<https://tutorial.recursospython.com/>) Newsletter (<https://recursospython.com/newsletter/>)

Consultoría (<https://recursospython.com/consultoria/>)

Contacto (<https://recursospython.com/contacto/>) Donar ❤️ (<https://recursospython.com/donar/>)

Grabador y reproductor de teclado y mouse

octubre 6, 2014 (<https://recursospython.com/codigos-de-fuente/grabador-teclado-mouse/>) by Recursos Python (<https://recursospython.com/author/admin/>)  (<https://recursospython.com/codigos-de-fuente/grabador-teclado-mouse/#comments>) 1 comentario (<https://recursospython.com/codigos-de-fuente/grabador-teclado-mouse/#comments>)

Un pequeño programa desarrollado con Python 2.7, PyQt 4 (<http://www.riverbankcomputing.co.uk/software/pyqt/intro>) y AutoPy (<https://www.recursospython.com/guias-y-manuales/autopy-toolkit/>) para grabar la entrada del usuario (mouse y teclado), y luego simularla. Utiliza Qt para la interfaz gráfica, como también para capturar los eventos de presión de teclas, y AutoPy para reproducirlas. Gracias a la portabilidad de estos módulos, el código corre en las distintas plataformas. Puedes descargar el código de fuente como archivo ZIP desde [este enlace](https://www.recursospython.com/wp-content/uploads/2014/10/inputrecorder.zip) (<https://www.recursospython.com/wp-content/uploads/2014/10/inputrecorder.zip>).

Vista previa



Últimas entradas

[Reproducir inyección](#)

[SQL en sqlite3 y](#)

[PyMySQL](#)

(<https://recursospython.com/guias-y-manuales/reproducir-inyeccion-sql-en-sqlite3-y-pymysql/>)

[Bloc de notas simple con](#)

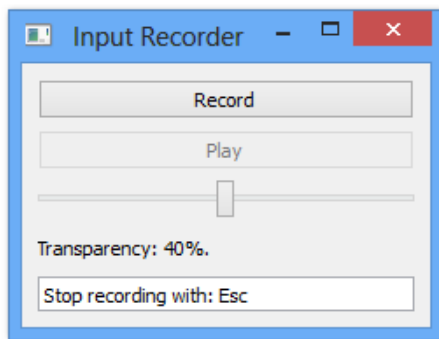
[Tk \(tkinter\)](#)

(<https://recursospython.com/codigos-de-fuente/bloc-de-notas-simple-con-tkinter/>)

[Examinar archivo o](#)

[carpeta en Tk \(tkinter\)](#)

(<https://recursospython.com/guias-y-manuales/examinar-archivo-o-carpeta-en-tk-tkinter/>)



(<https://www.recursospython.com/wp-content/uploads/2014/10/preview.png>)

Puedes ver el video del programa en funcionamiento en nuestra [página de Google+](https://plus.google.com/u/0/b/109760303901864157174/+Recursospyt) (<https://plus.google.com/u/0/b/109760303901864157174/+Recursospyt>)

Utilización

Sencillamente se presiona el botón «Record» para comenzar con la grabación. El programa torna a pantalla completa para capturar la totalidad de los eventos. Una vez finalizado, se presiona la tecla asignada para detener la grabación (Esc por defecto). El botón «Play» estará ahora habilitado para reproducir los eventos.

Limitaciones

Las teclas Alt, Control, Shift y Meta son soportadas como pulsaciones convencionales, no así como modificadores. Algunos programas pueden obtener la entrada del usuario por otros métodos, especialmente los juegos, por lo que las pulsaciones enviadas serán ignoradas.

Código de fuente

```
1.  #!/usr/bin/env python
2.  # -*- coding: utf-8 -*-
3.
4.  import sys
5.  from threading import Thread
6.  from time import sleep, time
7.
8.  import autopsy
9.  from PyQt4.QtCore import Qt, QEvent
10. from PyQt4.QtGui import (QApplication, QMainWindow, QLabel,
11.                           QPainter, QPushButton, QKeyEvent,
12.                           QKeySequence,
13.                           QSlider, QVBoxLayout, QWidget)
14.
15. additional_key_names = {
16.     Qt.Key_Control: "Ctrl",
17.     Qt.Key_Alt: "Alt",
18.     Qt.Key_AltGr: "Alt Gr",
```

[Múltiples configuraciones \(desarrollo/producción\) en Django](#)
(<https://recursospython.com/gui-y-manuales/multiples-configuraciones-desarrollo-produccion-en-django/>)

[Buscar el archivo de mayor tamaño en una ruta](#)
(<https://recursospython.com/codigo-de-fuente/buscar-el-archivo-de-mayor-tamano-en-una-ruta/>)

Comentarios recientes

Recursos Python en [Generar código QR](#)
(<https://recursospython.com/gui-y-manuales/generar-codigo-qr/#comment-2586>)

Joaquín en [Generar código QR](#)
(<https://recursospython.com/gui-y-manuales/generar-codigo-qr/#comment-2584>)

Recursos Python en [pickle – Serialización de objetos](#)
(<https://recursospython.com/gui-y-manuales/pickle-serializacion-de-objetos/#comment-2435>)

Recursos Python en [Lista desplegable \(Combobox\)](#)

```

19.     Qt.Key_Shift: "Shift",
20.     Qt.Key_Meta: "Meta"
21. }
22.
23. # Nombres de teclas de Qt a AutoPy
24. autoppy_keys = {
25.     "F1": autoppy.key.K_F1,
26.     "F2": autoppy.key.K_F2,
27.     "F3": autoppy.key.K_F3,
28.     "F4": autoppy.key.K_F4,
29.     "F5": autoppy.key.K_F5,
30.     "F6": autoppy.key.K_F6,
31.     "F7": autoppy.key.K_F7,
32.     "F8": autoppy.key.K_F8,
33.     "F9": autoppy.key.K_F9,
34.     "F10": autoppy.key.K_F10,
35.     "F11": autoppy.key.K_F11,
36.     "F12": autoppy.key.K_F12,
37.     "Meta": autoppy.key.K_META,
38.     "Shift": autoppy.key.K_SHIFT,
39.     "CapsLock": autoppy.key.K_CAPSLOCK,
40.     "Return": autoppy.key.K_RETURN,
41.     "Esc": autoppy.key.K_ESCAPE,
42.     "Space": " ",
43.     "PgUp": autoppy.key.K_PAGEUP,
44.     "PgDown": autoppy.key.K_PAGEDOWN,
45.     "Del": autoppy.key.K_DELETE,
46.     "End": autoppy.key.K_END,
47.     "Alt": autoppy.key.K_ALT,
48.     "Ctrl": autoppy.key.K_CONTROL,
49.     "Backspace": autoppy.key.K_BACKSPACE,
50. }
51.
52. EVENT_KEYBOARD = 0
53. EVENT_MOUSE = 1
54.
55.
56. class Ellipse(QWidget):
57.     """
58.     Simple círculo.
59.     """
60.
61.     def paintEvent(self, event):
62.         painter = QPainter(self)
63.         painter.drawEllipse(0, 0, 50, 50)
64.
65.
66. class KeySelector(QLineEdit):
67.     """
68.     Caja de texto para seleccionar una tecla.
69.     """
70.
71.     def __init__(self, parent, prompt, default_key):
72.         self.prompt = prompt
73.         self.key = default_key
74.         QLineEdit.__init__(self, parent)
75.         self.keyPressEvent(
76.             QKeyEvent(QEvent.KeyRelease, default_key,
77.                      Qt.KeyboardModifiers())
78.         )
79.
80.     def keyPressEvent(self, event):
81.         pass
82.
83.     def keyReleaseEvent(self, event):
84.         key_name = (additional_key_names[event.key()] if

```

[en Tcl/Tk \(tkinter\).
\(https://recursospython.com/gui-y-manuales/lista-desplegable-combobox-en-tkinter/#comment-2434\)](https://recursospython.com/gui-y-manuales/lista-desplegable-combobox-en-tkinter/#comment-2434)
 Herná en [Lista desplegable \(Combobox\) en Tcl/Tk \(tkinter\).
\(https://recursospython.com/gui-y-manuales/lista-desplegable-combobox-en-tkinter/#comment-2424\)](https://recursospython.com/gui-y-manuales/lista-desplegable-combobox-en-tkinter/#comment-2424)

```

85.         event.key() in additional_key_names
86.     else
87.         QKeySequence(event.key()).toString()
88.     self.key = event.key()
89.     self.setText(self.prompt.format(key_name))
90.
91.
92. class Window(QMainWindow):
93.     def __init__(self):
94.         QMainWindow.__init__(self)
95.         self.setWindowTitle("Input Recorder")
96.
97.         self.recording = False
98.         self.recorded_events = []
99.
100.        self.record_button = QPushButton("Record", self)
101.
102.        self.record_button.clicked.connect(self.start_record)
103.
104.        self.play_button = QPushButton("Play", self)
105.        self.play_button.clicked.connect(self.play)
106.        self.play_button.setEnabled(False)
107.
108.        self.key_label = QLabel(self)
109.        self.key_label.setGeometry(10, 10, 250, 50)
110.        self.key_label.setStyleSheet("font-size: 30px;")
111.
112.        self.transparency_slider = QSlider(Qt.Horizontal,
self)
113.        self.transparency_slider.setMaximum(80)
114.        self.transparency_slider.setSliderPosition(40)
115.        self.transparency_slider.valueChanged.connect(
self.set_transparency_value)
116.
117.        self.transparency_label = QLabel("Transparency:
40%", self)
118.
119.        self.key_selector = KeySelector(self, "Stop
recording with: {0}",
Qt.Key_Escape)
120.
121.
122.        self.ellipse = Ellipse(self)
123.        self.ellipse.resize(60, 60)
124.        self.ellipse.hide()
125.
126.        self.layout = QVBoxLayout()
127.        self.layout.addWidget(self.record_button)
128.        self.layout.addWidget(self.play_button)
129.        self.layout.addWidget(self.transparency_slider)
130.        self.layout.addWidget(self.transparency_label)
131.        self.layout.addWidget(self.key_selector)
132.
133.        self.central_widget = QWidget(self)
134.        self.central_widget.setLayout(self.layout)
135.        self.setCentralWidget(self.central_widget)
136.
137.        self.resize(500, 400)
138.
139.    def add_event(self, event, *params):
140.        sleep_time = time() - self.last_time
141.        self.last_time = time()
142.        self.recorded_events.append((sleep_time, event,
params))
143.
144.    def events_player_thread(self):
145.        self.showMinimized()
146.        for event in self.recorded_events:

```

```

147.         sleep_time, event_type, params = event
148.         sleep(sleep_time)
149.
150.         if event_type == EVENT_KEYBOARD:
151.             qt_key = str(params[0])
152.             # Enviar pulsación de tecla.
153.             autopy.key.tap(
154.                 autopy_keys[qt_key] if qt_key in
autopy_keys else
155.                 qt_key.lower()
156.             )
157.         elif event_type == EVENT_MOUSE:
158.             x, y = params
159.             # Enviar pulsación del ratón.
160.             autopy.mouse.move(x + 10, y)
161.             autopy.mouse.click()
162.             # Restaurar ventana.
163.             self.showNormal()
164.
165.         def mousePressEvent(self, event):
166.             if self.recording:
167.                 # Dibujar un círculo donde se haya presionado.
168.                 self.ellipse.move(event.x() - 25, event.y() -
25)
169.                 self.ellipse.show()
170.                 self.add_event(EVENT_MOUSE, event.x(),
event.y())
171.                 return QMainWindow.mousePressEvent(self, event)
172.
173.         def mouseReleaseEvent(self, event):
174.             if self.recording:
175.                 self.ellipse.hide()
176.                 return QMainWindow.mouseReleaseEvent(self, event)
177.
178.         def keyPressEvent(self, event):
179.             if self.recording:
180.                 if event.key() == self.key_selector.key:
181.                     self.stop_record()
182.                 else:
183.                     # Buscar un nombre predefinido disponible.
184.                     # De no encontrarlo, utilizar el nombre de
la tecla de Qt.
185.                     key_name =
(additional_key_names[event.key()] if
186.                         event.key() in
additional_key_names else
187.                         QKeySequence(event.key()).toString())
188.                     self.key_label.setText(key_name)
189.                     self.add_event(EVENT_KEYBOARD, key_name)
190.                     return QMainWindow.keyPressEvent(self, event)
191.
192.         def play(self):
193.             # Evitar bloquear el bucle principal.
194.             Thread(target=self.events_player_thread).start()
195.
196.         def set_transparency_value(self, value):
197.             self.transparency_label.setText("Transparency:
%d%%." % value)
198.
199.         def start_record(self):
200.             self.recording = True
201.             self.recorded_events = []
202.             self.last_time = time()
203.             # Ocultar controles.
204.             self.central_widget.hide()
205.             # Establecer transparencia.

```

```

206.         self.setWindowOpacity(
207.             (100. - self.transparency_slider.value()) /
208.             100.
209.         )
210.         # Pantalla completa.
211.         self.showFullScreen()
212.
213.     def stop_record(self):
214.         # Restaurar la configuración inicial.
215.         self.recording = False
216.         self.central_widget.show()
217.         self.setWindowOpacity(1.0)
218.         self.showNormal()
219.         self.key_label.setText("")
220.         if self.recorded_events:
221.             self.play_button.setEnabled(True)
222.
223. if __name__ == "__main__":
224.     app = QApplication([])
225.     window = Window()
226.     window.show()
227.     sys.exit(app.exec_())

```

Artículos relacionados

- [PyQt – Ícono en el área de notificaciones \(System tray\)](https://recursospython.com/guias-y-manuales/pyqt-icone-area-de-notificaciones-system-tray/)
(<https://recursospython.com/guias-y-manuales/pyqt-icone-area-de-notificaciones-system-tray/>)
- [Generador de contraseñas \(PyQt\)](https://recursospython.com/codigos-de-fuente/generador-de-contrasenas-pyqt/)
(<https://recursospython.com/codigos-de-fuente/generador-de-contrasenas-pyqt/>)
- [Reproductor de video simple con PyQt 4 y 5](https://recursospython.com/codigos-de-fuente/reproductor-de-video-pyqt-4/)
(<https://recursospython.com/codigos-de-fuente/reproductor-de-video-pyqt-4/>)
- [Reproductor de audio de YouTube con PyQt 4](https://recursospython.com/codigos-de-fuente/reproductor-youtube-pyqt4/)
(<https://recursospython.com/codigos-de-fuente/reproductor-youtube-pyqt4/>)
- [Explorador de archivos y carpetas](https://recursospython.com/codigos-de-fuente/explorador-archivos-carpetas/)
(<https://recursospython.com/codigos-de-fuente/explorador-archivos-carpetas/>)

Donar

¿Te gusta nuestro contenido? ¡Ayúdanos a [seguir creciendo con una donación \(/donar\)](#)!

Entrada publicada en
Códigos de fuente (<https://recursospython.com/category/codigos-de-fuente/>) con las
etiquetas [autopy](https://recursospython.com/tag/autopy/) (<https://recursospython.com/tag/autopy/>)
[PyQt](https://recursospython.com/tag/pyqt/) (<https://recursospython.com/tag/pyqt/>)



◀ [Cookies y autenticación de usuarios en CGI](https://recursospython.com/codigos-de-fuente/cookies-autenticacion-cgi/)
(<https://recursospython.com/codigos-de-fuente/cookies-autenticacion-cgi/>)

[Operaciones con archivos y carpetas](https://recursospython.com/guias-y-manuales/os-shutil-archivos-carpetas/) ▶
(<https://recursospython.com/guias-y-manuales/os-shutil-archivos-carpetas/>)

1 comentario.



Luis Pallares says:

[julio 7, 2018 at 6:03 pm](#)

(<https://recursospython.com/codigos-de-fuente/grabador-teclado-mouse/#comment-428>)

podrias convertirlo en ejecutable? no tengo python 2.7 ni pyqt4 , tengo 3.6 y pyqt5

[Responder](#)

Deja una respuesta

Comentario *

Nombre *

Email *

Publicar el comentario



© 2013 - 2023

¡Suscríbete a nuestra newsletter! (<https://www.rekursospython.com/newsletter/>)

[Políticas de Uso y Privacidad \(https://www.rekursospython.com/politicas-de-uso-y-privacidad/\)](https://www.rekursospython.com/politicas-de-uso-y-privacidad/)

En inglés: Python Assets (<https://pythonassets.com/>)



(<https://creativecommons.org/licenses/by-nc/3.0/deed.es>)