In [1]:
```
!pip install inputs
```

```
Requirement already satisfied: inputs in c:\users\dean4ta\anaconda3\lib\site-
packages
```

In [2]:
```
from inputs import devices
```

```
we found a gamepad.
```

In [3]:
```
devices
```

Out[3]:
```
<inputs.DeviceManager at 0x269d62d7588>
```

In [4]:
```
devices.gamepads
```

Out[4]:
```
[inputs.GamePad("/dev/input/by_id/usb-Microsoft_Corporation_Controller_0-even
t-joystick")]
```

In [1]:
```
from inputs import get_gamepad
```

```
we found a gamepad.
```

In [2]:
```
events = []
stop = 0
while(1):
    events = get_gamepad()
    for event in events:
        print(event.ev_type, event.code, event.state)
    if stop == 1:
        break
    if event.state == 0:
        if event.code == 'BTN_NORTH':
            stop = 1
```

```
Key BTN_EAST 1
Sync SYN_REPORT 0
Key BTN_EAST 0
Sync SYN_REPORT 0
Key BTN_NORTH 1
Sync SYN_REPORT 0
Key BTN_NORTH 0
Sync SYN_REPORT 0
```

In [40]:
```
type(event.state)
```

Out[40]:
```
int
```

In [43]:
```
if event.state == 0:
    if event.code == 'SYN_REPORT':
        print('yes')
```

```
yes
```

In [28]:
```
!pip install pyautogui
```

```
Requirement already satisfied: pyautogui in c:\users\dean4ta\anaconda3\lib\si
te-packages
Requirement already satisfied: pymsgbox in c:\users\dean4ta\anaconda3\lib\sit
e-packages (from pyautogui)
Requirement already satisfied: Pillow in c:\users\dean4ta\anaconda3\lib\site-
packages (from pyautogui)
Requirement already satisfied: PyTweening>=1.0.1 in c:\users\dean4ta\anaconda
3\lib\site-packages (from pyautogui)
Requirement already satisfied: pyscreeze in c:\users\dean4ta\anaconda3\lib\si
te-packages (from pyautogui)
Requirement already satisfied: olefile in c:\users\dean4ta\anaconda3\lib\site
-packages (from Pillow->pyautogui)
```

In [3]:
```
from pyautogui import typewrite, hotkey
```

In [9]:
```
hotkey('ctrl', 'd')
```

In [11]:
```
events = []
stop = 0
dPadXY = [0,0]
dPadXY2 = [0,0]
dPadStart = [0,0]
dPadEnd = [0,0]
while(1):
    events = get_gamepad()
    for event in events:
        pass
        #print(event.ev_type, event.code, event.state)


    ## The Keyboard input ##

    #Listen for D-Pad input (save whatever the last D-Pad input is)
    if event.code[0:8] == 'ABS_HAT0':
        if event.code[8] == 'X':
            dPadXY[0] = event.state
            #print(dPadX)
        if event.code[8] == 'Y':
            dPadXY[1] = event.state
            #print(dPadY)

    if event.code == 'BTN_SOUTH':
        if event.state == 1:
            print('enter listen mode')
            dPadStart = dPadXY
            #print starting D-Pad state
            if dPadXY[0] == 1:
                xStr = 'right'
            elif dPadXY[0] == -1:
                xStr = 'left'
            else:
                xStr = ''
            if dPadXY[1] == 1:
```

```
    yStr = 'down'
elif dPadXY[1] == -1:
    yStr = 'up'
else:
    yStr = ''
print('starting D-Pad State:', xStr, yStr)

while (1):
    events = get_gamepad()
    for event in events: #for loop used for printing event info
        pass
        #print(event.ev_type, event.code, event.state)
    #grab the ending D-Pad State
    if event.code[0:8] == 'ABS_HAT0':
        if event.code[8] == 'X':
            dPadXY2[0] = event.state
        if event.code[8] == 'Y':
            dPadXY2[1] = event.state

    if event.code == 'BTN_SOUTH':
        if event.state == 0:
            #print ending D-Pad State
            if dPadXY2[0] == 1:
                xStr = 'right'
            elif dPadXY2[0] == -1:
                xStr = 'left'
            else:
                xStr = ''
            if dPadXY2[1] == 1:
                yStr = 'down'
            elif dPadXY2[1] == -1:
                yStr = 'up'
            else:
                yStr = ''
            dPadEnd = dPadXY2
            print('ending D-Pad State:', xStr, yStr)
            #grab the last line
            events = get_gamepad()
            for event in events:
                pass
                #print(event.ev_type, event.code, event.state)
            print(dPadStart, dPadEnd)

            '''output the keyboard'''
            #a
            if dPadStart == [-1,-1] and dPadEnd == [-1,-1]:
                typewrite('a')
            #b
            if dPadStart == [0,0] and dPadEnd == [1,0]:
                typewrite('b')
            #c
            if dPadStart == [0,0] and dPadEnd == [-1,0]:
                typewrite('c')
            #d
            if dPadStart == [0,0] and dPadEnd == [0,1]:
                typewrite('d')
            #e
```

```
if dPadStart == [0,1] and dPadEnd == [0,1]:
    typewrite('e')
#f
if dPadStart == [1,1] and dPadEnd == [0,0]:
    typewrite('f')
#g
if dPadStart == [0,0] and dPadEnd == [-1,1]:
    typewrite('g')
#h
if dPadStart == [-1,0] and dPadEnd == [-1,0]:
    typewrite('h')
#i
if dPadStart == [1,-1] and dPadEnd == [1,-1]:
    typewrite('i')
#j
if dPadStart == [0,0] and dPadEnd == [1,1]:
    typewrite('j')
#k
if dPadStart == [-1,0] and dPadEnd == [0,0]:
    typewrite('k')
#l
if dPadStart == [0,-1] and dPadEnd == [0,0]:
    typewrite('l')
#m
if dPadStart == [1,0] and dPadEnd == [0,0]:
    typewrite('m')
#n
if dPadStart == [0,-1] and dPadEnd == [0,-1]:
    typewrite('n')
#o
if dPadStart == [0,0] and dPadEnd == [0,0]:
    typewrite('o')
#p
if dPadStart == [0,0] and dPadEnd == [1,-1]:
    typewrite('p')
#q
if dPadStart == [0,0] and dPadEnd == [-1,-1]:
    typewrite('q')
#r
if dPadStart == [1,0] and dPadEnd == [1,0]:
    typewrite('r')
#s
if dPadStart == [1,1] and dPadEnd == [1,1]:
    typewrite('s')
#t
if dPadStart == [-1,1] and dPadEnd == [-1,1]:
    typewrite('t')
#u
if dPadStart == [0,0] and dPadEnd == [0,-1]:
    typewrite('u')
#v
if dPadStart == [-1,-1] and dPadEnd == [0,0]:
    typewrite('v')
#w
if dPadStart == [0,1] and dPadEnd == [0,0]:
    typewrite('w')
#x
```

```
                        if dPadStart == [1,-1] and dPadEnd == [0,0]:
                            typewrite('x')
                        #y
                        if dPadStart == [-1,1] and dPadEnd == [0,0]:
                            typewrite('y')
                        #z
                        if dPadStart == [0,1] and dPadEnd == [1,0]:
                            typewrite('l')

                        dPadEnd = [0,0]


                        break



    #stop condition (this is placed after for loop so program can clear final
line from the queue)
    if stop == 1:
        break
    #check to trigger stop condition
    if event.state == 0:
        if event.code == 'BTN_NORTH':
            stop = 1
```

In [18]:
```
events = []
stop = 0
dPadXY = [0,0]
dPadStart = [0,0]
dPadEnd = [0,0]
while(1):
    events = get_gamepad()
    for event in events:
        pass
        #print(event.ev_type, event.code, event.state)


    ## The Keyboard input ##

    #listen for D-Pad input (save whatever the last D-Pad input is)
    if event.code[0:8] == 'ABS_HAT0':
        if event.code[8] == 'X':
            dPadXY[0] = event.state
        if event.code[8] == 'Y':
            dPadXY[1] = event.state
        print(dPadXY)

    if event.code == 'BTN_SOUTH':
        if event.state == 1:
            print('enter listen mode')
            dPadStart = dPadXY
            #print starting D-Pad state
            if dPadXY[0] == 1:
                xStr = 'right'
            elif dPadXY[0] == -1:
                xStr = 'left'
            else:
```

```
        xStr = ''
    if dPadXY[1] == 1:
        yStr = 'down'
    elif dPadXY[1] == -1:
        yStr = 'up'
    else:
        yStr = ''
    print('starting D-Pad State:', xStr, yStr)

    while (1):
        events = get_gamepad()
        for event in events: #for loop used for printing event info
            pass
            #print(event.ev_type, event.code, event.state)
        #grab the ending D-Pad State
        if event.code[0:8] == 'ABS_HAT0':
            if event.code[8] == 'X':
                dPadXY2[0] = event.state
            if event.code[8] == 'Y':
                dPadXY2[1] = event.state

        if event.code == 'BTN_SOUTH':
            if event.state == 0:
                #print ending D-Pad State
                if dPadXY2[0] == 1:
                    xStr = 'right'
                elif dPadXY2[0] == -1:
                    xStr = 'left'
                else:
                    xStr = ''
                if dPadXY2[1] == 1:
                    yStr = 'down'
                elif dPadXY2[1] == -1:
                    yStr = 'up'
                else:
                    yStr = ''
                dPadEnd = dPadXY2
                print('ending D-Pad State:', xStr, yStr)
                #grab the last line
                events = get_gamepad()
                for event in events:
                    pass
                    #print(event.ev_type, event.code, event.state)
                print(dPadStart, dPadEnd)

                '''output the keyboard'''
                #a
                if dPadStart == [-1,-1] and dPadEnd == [-1,-1]:
                    typewrite('a')
                #b
                if dPadStart == [0,0] and dPadEnd == [1,0]:
                    typewrite('b')
                #c
                if dPadStart == [0,0] and dPadEnd == [-1,0]:
                    typewrite('c')
                #d
                if dPadStart == [0,0] and dPadEnd == [0,1]:
```

```
        typewrite('d')
    #e
    if dPadStart == [0,1] and dPadEnd == [0,1]:
        typewrite('e')
    #f
    if dPadStart == [1,1] and dPadEnd == [0,0]:
        typewrite('f')
    #g
    if dPadStart == [0,0] and dPadEnd == [-1,1]:
        typewrite('g')
    #h
    if dPadStart == [-1,0] and dPadEnd == [-1,0]:
        typewrite('h')
    #i
    if dPadStart == [1,-1] and dPadEnd == [1,-1]:
        typewrite('i')
    #j
    if dPadStart == [0,0] and dPadEnd == [1,1]:
        typewrite('j')
    #k
    if dPadStart == [-1,0] and dPadEnd == [0,0]:
        typewrite('k')
    #l
    if dPadStart == [0,-1] and dPadEnd == [0,0]:
        typewrite('l')
    #m
    if dPadStart == [1,0] and dPadEnd == [0,0]:
        typewrite('m')
    #n
    if dPadStart == [0,-1] and dPadEnd == [0,-1]:
        typewrite('n')
    #o
    if dPadStart == [0,0] and dPadEnd == [0,0]:
        typewrite('o')
    #p
    if dPadStart == [0,0] and dPadEnd == [1,-1]:
        typewrite('p')
    #q
    if dPadStart == [0,0] and dPadEnd == [-1,-1]:
        typewrite('q')
    #r
    if dPadStart == [1,0] and dPadEnd == [1,0]:
        typewrite('r')
    #s
    if dPadStart == [1,1] and dPadEnd == [1,1]:
        typewrite('s')
    #t
    if dPadStart == [-1,1] and dPadEnd == [-1,1]:
        typewrite('t')
    #u
    if dPadStart == [0,0] and dPadEnd == [0,-1]:
        typewrite('u')
    #v
    if dPadStart == [-1,-1] and dPadEnd == [0,0]:
        typewrite('v')
    #w
    if dPadStart == [0,1] and dPadEnd == [0,0]:
```

```
                        typewrite('w')
                    #x
                    if dPadStart == [1,-1] and dPadEnd == [0,0]:
                        typewrite('x')
                    #y
                    if dPadStart == [-1,1] and dPadEnd == [0,0]:
                        typewrite('y')
                    #z
                    if dPadStart == [0,1] and dPadEnd == [1,0]:
                        typewrite('l')

                    dPadEnd = [0,0]
                    dPadStart = [0,0]
                    dPadXY = [0,0]

                    break



    #stop condition (this is placed after for loop so program can clear final
 line from the queue)
    if stop == 1:
        break
    #check to trigger stop condition
    if event.state == 0:
        if event.code == 'BTN_NORTH':
            stop = 1
```

```
enter listen mode
starting D-Pad State:
ending D-Pad State: right up
[0, 0] [1, -1]
[0, 0]
enter listen mode
starting D-Pad State:
ending D-Pad State: left up
[0, 0] [-1, -1]
[0, 0]
enter listen mode
starting D-Pad State:
ending D-Pad State: left up
[0, 0] [-1, -1]
[1, 0]
[1, -1]
[0, -1]
[0, 0]
[1, 0]
[0, 0]
[0, 1]
[-1, 1]
[0, 1]
[0, 0]
enter listen mode
starting D-Pad State:
ending D-Pad State: left up
[0, 0] [-1, -1]
```