Topic	Remote Mouse	
Class Description	**Student will be able to access other PCs on their LAN by turning their PC into a remote mouse**	
Class	**C-213**	
Class time	**45 mins**	
Goal	• Understand about Remote Mouse • Building Socket & Client Connection	

Resources Required	• Teacher Resources:
	  ○ Laptop with internet connectivity
	  ○ Earphones with mic
	  ○ Notebook and pen
	  ○ Visual Studio Code
	• Student Resources:
	  ○ Laptop with internet connectivity
	  ○ Earphones with mic
	  ○ Notebook and pen
	  ○ Visual Studio Code	

Class structure	**Warm-Up** **Teacher - led Activity 1** **Student - led Activity 1** **Wrap-Up**	**10 mins** **10 mins** **20 mins** **5 mins**
WARM UP SESSION - 10mins		
Teacher Action		**Student Action**
Hey <student's name>. How are you? It's great to see you! Are you excited to learn something new today?		**ESR**: Hi, thanks, yes, I am excited about it!
Q&A Session		

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

Question	Answer
What do you mean by widgets? A. Act as a class. B. Act as a function C. Act as a events D. None of the above	**A**
What is the need of FTP? A. To Upload B. For download C. File transfer D. None of the above	**C**
TEACHER-LED ACTIVITY - 10mins	
Teacher Initiates Screen Share	
ACTIVITY • **Socket & Client Connection** • **Get screen size** • **Call the function**	
Teacher Action	**Student Action**
Okay, so you remember what we did in the last session Great! Any doubts from last session? *The teacher clarifies doubts (if any)* Any idea what we will be doing in today's session?	**ESR** Yes! **ESR:** Varied!

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

Mouse, have you heard about it! Do you know which mouse I'm referring to? Here, I am talking about a computer mouse! How does your computer's mouse work? The computer mouse transmits your commands to the computer by controlling the movement of the cursor/pointer on the screen. The pointer on the screen moves in the same direction as you move the mouse. What if I use a remote mouse? What will the remote mouse do? Remote Mouse uses your computer to act as a remote control for other computers who are connected on a LAN . It simulates the functions of a wireless mouse, left click, right click and touchpad	**ESR:** Computer Mouse or Actual Mouse ! **ESR** **Varied!** **ESR** Varied!
Open the Teacher Activity 1	*Student opens the Student Activity 1*
Isn't it awesome if we can make our own remote mouse and control PCs which are on same LAN Let's start making one! To make our remote mouse work, we need to install a few libraries in our system We will first learn what kind of libraries we are using and how we can install them • Using the Python ***pynput*** library, we can control and monitor input devices. Using ***pynput*** we are able to simulate mouse events into any window.	

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

- ***pynput.mouse*** contains classes for controlling and monitoring the mouse buttons & controller. The mouse is tracked and controlled by using the coordinate system of the screen.
- ***screeninfo*** library is used to fetch location and size of physical screens.
- ***autoPy*** is a GUI automation library for Python. It includes functions for controlling the keyboard and mouse

Note: Install the same libraries on the student's computer as well.

For windows/Mac:

Go to Command prompt

- pip install pynput
- pip install screeninfo
- pip install autopy

```
C:\Users\Tamanna>pip install pynput
Requirement already satisfied: pynput in c:\users\tamanna\appdata\local\programs\python\python38-32\lib\site-packages (1
.7.3)
Requirement already satisfied: six in c:\users\tamanna\appdata\local\programs\python\python38-32\lib\site-packages (from
 pynput) (1.15.0)

C:\Users\Tamanna>pip install screeninfo
Requirement already satisfied: screeninfo in c:\users\tamanna\appdata\local\programs\python\python38-32\lib\site-package
s (0.6.7)

C:\Users\Tamanna>pip install autopy
Requirement already satisfied: autopy in c:\users\tamanna\appdata\local\programs\python\python38-32\lib\site-packages (4
.0.0)
```

We know that a network-based application would not be possible if server and client sockets weren't available.

So our first task is to make server and client

For remote clients, apks will be downloaded directly

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

<table>
<tr><td>and for that we will use kivi framework

Let’s make a server file now.

Open visual code and make a server.py</td><td></td></tr>
<tr><td>Teacher download the boilerplate code from Teacher Activity 2

Note :- Code to create a client is already provided in the boilerplate code. Explain the code to the Student.</td><td>Student download the repository from Student Activity 2</td></tr>
<tr><td>Let’s make a server socket first

For creating sockets, Python has a very famous and widely used library called socket

As a result, we have to import socket into server.py.

For the server and client to communicate simultaneously, we will be using threading to execute these functions parallely.Thus, we'll import the Thread library too.</td><td></td></tr>
<tr><td colspan="2"><pre>import socket
from threading import Thread</pre></td></tr>
<tr><td>We have already installed the required remote mouse libraries, so now it is time to import them.

● import pynput.mouse import Button, Controller
pynput.mouse contains classes for controlling and monitoring the mouse buttons & controller. The mouse is tracked and controlled by using the coordinate system of the screen.

● from screeninfo import get_monitors
This library is used to fetch location and size of physical screens.</td><td></td></tr>
</table>

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

- ***import autopy***
  It includes functions for controlling the keyboard and mouse

Note: If autopy is not installed properly:

Then try following steps:

pip install -U autopy

If that fails, install rustup and then run

pip install -U setuptools -rust
pip install -U autopy

It's time to install libraries in our system as well

```
from pynput.mouse import Button, Controller
from screeninfo import get_monitors
import autopy
```

We need to declare some global variables so we can access them at any time

- Declare some global variables such as ***SERVER*** and set their values to None. None defines a null value, or no value at all. **None** is not the same as 0, False, or an empty string.
- Set the port to 8000, however this port can be any number. Just make sure that it is not anything lower than ***1,024***, since those are reserved ports
- Take the IP_Address input from the user. This is the user's LAN address
- Set ***screen_width*** and ***screen_height*** to None

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

```
SERVER = None
PORT = 8000
IP_ADDRESS = input("Enter your computer IP ADDR : ").strip()
screen_width = None
screen_height = None
```

A function called **setup()** was created to setup the server

- First, we print the name of our application "**Welcome to Remote Mouse"**
- Then we call all the global variables that we declared earlier, namely ***PORT*, *IP_ADDRESS*, and *SERVER*.**
- Then we define the ***IP_ADDRESS*** and the ***PORT*** we are using.
- Next we bind our server with the IP Address and the Port that we are using and then we are ready to listen for any incoming requests from the clients.
- As for the server, we're using ***socket.socket()*** function, and we're binding to it via ***bind()***, which takes a tuple containing the ***IP_ADDRESS*** and ***port number.***
- Our server has now successfully bound, so we can start listening on this server socket using the ***listen()*** function.
- Here, we specify that we only want 10 connections.
- Then we are printing a text which says that the "***SERVER IS WAITING FOR INCOMING CONNECTIONS***."

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

Then finally we'll call the ***setup()*** wherever we need it.

```
def setup():
    print("\n\t\t\t\t\t*** Welcome To Remote Mouse ***\n")


    global SERVER
    global PORT
    global IP_ADDRESS


    SERVER = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    SERVER.bind((IP_ADDRESS, PORT))

    SERVER.listen(10)

    print("\t\t\t\tSERVER IS WAITING FOR INCOMMING CONNECTIONS...\n")



setup()
```

Now the server is ready but we have to accept all incoming requests from the client side

For that we will make a function ***acceptConnections()*** and will call the same in our server ***setup()*** function

- Declared global variables **SERVER & Clients**.
- Since this indefinite process will make the while loop true, all incoming connections should be accepted using the ***accept()*** method, which waits for an incoming connection
- When a client connects, it returns a new socket object representing the connection and a tuple holding the address of the client.
- Now we will printing client and address

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

```
def acceptConnections():
    global SERVER

    while True:
        client_socket, addr = SERVER.accept()

        print(f"Connection established with {client_socket} : {addr}")
```

It's time to call the ***acceptConnections()*** function in the ***setup()*** function.

The boiler plate code ends here

```
def setup():
    print("\n\t\t\t\t\t*** Welcome To Remote Mouse ***\n")


    global SERVER
    global PORT
    global IP_ADDRESS


    SERVER = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    SERVER.bind((IP_ADDRESS, PORT))

    SERVER.listen(10)

    print("\t\t\t\tSERVER IS WAITING FOR INCOMMING CONNECTIONS...\n")


    acceptConnections()
setup()
```

Teacher starts writing the code from here
Use a variable mouse to store the information we get from the pynput mouse controller

```
mouse = Controller()
```

Now, we need to fetch the size of the screen for that we will make a function for that.

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

We'll call this function ***getDeviceSize()***

- Declare Global ***Screen_width & Screen_height***
- Screeninfo library helps in the get the screen_width & screen_height
- By using the for loop, we can retrieve the combined height and width of the monitor:
- ***get_monitors*** will retrieve x, y coordinates along with height and width
- We need to store only width and height in the "m" variable that we get from ***get_monitors***. The ***split()*** method constructs a list from the string, and the ***strip()*** method removes the unwanted characters.

```
def getDeviceSize():
    global screen_width
    global screen_height
    for m in get_monitors():
        screen_width = int(str(m).split(",")[2].strip().split('width=')[1])
        screen_height = int(str(m).split(",")[3].strip().split('height=')[1])
```

Now that we have screen_height & screen_width, we need to call ***getDeviceSize()*** in our main ***setup()*** function

While our ***setup()*** function is running, it will also get the screen_size

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

```
def setup():
    print("\n\t\t\t\t\t*** Welcome To Remote Mouse ***\n")


    global SERVER
    global PORT
    global IP_ADDRESS


    SERVER = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    SERVER.bind((IP_ADDRESS, PORT))

    SERVER.listen(10)

    print("\t\t\t\tSERVER IS WAITING FOR INCOMMING CONNECTIONS...\n")

    getDeviceSize()
    acceptConnections()

setup()
```

As we have screen information and mouse control information, we must transmit these from the server side to the client side. For that we will create a function ***recvMessage()***

Teacher Stops Screen Share

STUDENT-LED ACTIVITY - 20 mins

- **Ask the student to press the ESC key to come back to the panel.**
- **Guide the student to start Screen Share.**
- **The teacher gets into Full Screen.**

ACTIVITY

- **Get mouse value & positions**
- **Create thread**

Teacher Action	Student Action
Guide the student to get the boilerplate code from Student Activity 2	*Student clones the code*

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	from Student Activity2
Now ist time to evaluate whether the user is pressing a left click or a right click or whether the user is using a trackpad, and how our remote computer will respond accordingly based on that information The function is called ***recvMessage()*** and pass the argument ***client_socket()*** ● Declare the mouse as global variable ● As this will be an indefinite process, use a while loop ● Start writing the code with try and exception to handle selected exceptions ● Create variable message and store data using **recv()** and decode it ● Now it's time to evaluate the data we received from the mouse controller ● New_message is another variable that we should create to store the message data ● With if condition we will check whether the user has pressed left click or right click ● Here press means mouse is “clicked” and release means “no click” ● In else condition we will check our mouse trackpad conditions will get the position of our cursor and display the same position at remote compute too	

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

- Then the xpos variable will save the x coordinate position that we get from our new_message variable, which continuously evaluates the positions and then multiplies this value with our screen _width.Here we are using 100% width size.

- Now ypos variable will save the y coordinate position that we get from our new_message. On the vertical side of the screen, 20% is used for a label, where we have written **"Remote Mouse"** and 20% is used for **"Left Click"** and **"Right Click"**So we are using only 60% of the screen size. Subtracting this new_message data position from 1 and then subtracting 0.2, which equals 20%, we divide this by 60 percent, which equals 0.6 which will give the exact position for y-axis position.

- Any exception that occurs is passed to the error handler

```
def recvMessage(client_socket):
    global mouse

    while True:
        try:
            message = client_socket.recv(2048).decode()
            if(message):
                new_message  = eval(message)
                if(new_message["data"] == 'left_click'):
                    mouse.press(Button.left)
                    mouse.release(Button.left)
                    mouse.press(Button.right)
                    mouse.release(Button.right)
                else:
                    xpos =  new_message["data"][0] * screen_width
                    ypos = screen_height * (1 - (new_message["data"][1] - 0.2) / 0.6 )
                    mouse.position = (int(xpos), int(ypos))

        except Exception as error:
            pass
```

Now create a thread in ***acceptConnection()*** and this time

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

target will be our ***recvMessage()*** pass arguments **client_socket.**

And then start the thread using ***start()*** method

```
def acceptConnections():
    global SERVER

    while True:
        client_socket, addr = SERVER.accept()

        print(f"Connection established with {client_socket} : {addr}")

        thread1 = Thread(target = recvMessage, args=(client_socket,))
        thread1.start()
```

It's time to run the code. When you run the server file, it will ask you for your IP address
Type ipconfig at the command prompt
You can find out your IP address with the ipconfig command

```
Enter your computer IP ADDR :
```

```
Enter your computer IP ADDR : 192.168.0.100

                    *** Welcome To Remote Mouse ***

                SERVER IS WAITING FOR INCOMMING CONNECTIONS...
```

From student activity 2, download the **main.py** file

- Upload the file to your mobile device
- The Pydroid Py3 can be downloaded from the Play Store
- Open Pydroid
- Click on the rectangle type box
- Click "open"

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

- Upload the main.py file (see Phone download section)
- Click on the yellow triangle icon in the bottom right corner
- Below is a window that appears:
- Enter the IP address of your computer. This should be the same one we used in the server side
- Click on “Connect With PC”

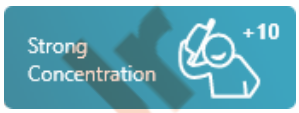
A window will appear after you click connect with PC:

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

© 2021 - WhiteHat
Note: This documer
Please don't share, download or copy this file without permission.

Teacher Guides Student to Stop Screen Share	
WRAP UP SESSION - 5 Mins	
Quiz time - Click on in-class quiz	
Question	**Answer**
What is the purpose of **strip()** function? A. Remove leading and trailing characters B. Show a list of characters C. Remove all the characters D. None of the above	**A**
What is the procedure to install pynput? A. pip3 install pynput B. pip install py input C. pip install pynput D. None of the above	**C**
Why do we need the init() function in class? A. Act as a constructor B. Automatically call the function C. Act as a intilizarer D. All of the above	**D**
End the quiz panel	
FEEDBACK • **Appreciate the students for their efforts in the class.** • **Ask the student to make notes for the reflection journal along with the code they wrote in today's class.**	
Teacher Action	**Student Action**
You get Hats off for your excellent work!	*Make sure you have given*

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

In the next class	*at least 2 Hats Off during the class for:*   
Project Discussion	
Teacher Clicks 	
ADDITIONAL ACTIVITIES	
Additional Activities *Encourage the student to write reflection notes in their reflection journal using markdown.* Use these as guiding questions: ● What happened today? ○ Describe what happened. ○ The code I wrote. ● How did I feel after the class? ● What have I learned about programming and developing games? ● What aspects of the class helped me? What did I find difficult?	*The student uses the markdown editor to write her/his reflections in the reflection journal.*

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

ACTIVITY LINKS		
Activity Name	**Description**	**Link**
Teacher Activity1	Computer Mouse	https://en.wikipedia.org/wiki/Computer_mouse
Teacher Activity 2	Boilerplate Code	https://github.com/pro-whitehatjr/C213-TeacherBoilerPLateCode
Teacher Activity 2	Reference Code	https://github.com/pro-whitehatjr/C213-ReferenceCode
Student Activity 1	Computer Mouse	https://en.wikipedia.org/wiki/Computer_mouse
Student Activity 2	Boilerplate Code	https://github.com/pro-whitehatjr/PRO-C213-StudentBoilerPlate

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.