REMOTE MOUSE

COMPUTER NETWORKING

Time: 60 mins

Introduction

In this class, the student(s) will create a remote mouse application to control a remote computer by turning a mobile device into a remote mouse.

New Commands Introduced

mouse = Controller():	Used to control the mouse, such as moving the cursor or clicking.
mouse.press(Button.right):	Simulate the press of the mouse right press.
mouse.release(Button.right):	Simulate the release of the mouse right button.
monitor.width:	Get the width of a screen using monitor info.
get_monitors()	Retrieves information about the connected monitors.

Vocabulary

- **Remote access** is the ability to access a computer or device from another device, at any time, and from anywhere.
- **Kivy** is an open-source Python framework for developing multi-touch applications, particularly those with graphical user interfaces (GUIs).

Learning Objectives

Student(s) should be able to:

- ***Explain*** how to use the Kivy framework for developing multi-touch applications, particularly those with graphical user interfaces (GUIs).
- ***Explore*** ~~using~~ the Pydroid ~~a~~ code editor ~~app~~ on a mobile device and run the Python client file.
- ***Recall*** ~~to create~~ the client and server connections to make the remote mouse application functional.
- ***Demonstrate*** the creation of a remote mouse application to control the remote computer using the mobile device as a remote mouse.

Activities

1. **Class Narrative:** *(3 mins)*

- Brief the student(s) that they would create a remote mouse application to control the movement of the cursor/pointer of their computer screen using their mobile.

2. **Concept Introduction Activity:** *(4 mins)*

    - Explain the steps to student(s) that they would need to install necessary libraries, and run the server.py file on the computer, and run the remote mouse application.
        - Sreeninfo: Retrieves information about physical screens.
        - Pynput: Generates and controls mouse events in any window.
        - Autopy: Manages and simulates keyboard and mouse interactions on the screen.
            - *Note: If autopy is not installed properly, then try the following steps*:
                - pip install -U autopy
            - *If that fails, install rustup and then run.*
                - pip install -U setuptools -rust
                - pip install -U autopy
            - *If still this fails, then click here to download the Python 3.8.8 version. and then install autopy.*
                - choco install python --version 3.8.8
                - pip install -U autopy
    - Explain the steps to test the remote mouse application using the Pydroid code editor.
        - Download and install the Pydroid app on the mobile device.
        - Download the main.py file on the mobile device by typing https://bit.ly/main_py link in the mobile browser(Link Given on slide no: 9).
        - Click on the yellow triangle icon to run the main.py file.
        - Enter the IP address of the computer and click on “Connect with PC”.
    - Let the student(s) perform the explore-activity to control the computer using the mobile device by running the remote mouse application on the computer and the mobile device.
    - Using the slides, explain that the student(s) will learn to:
        - Install libraries for mouse control, screen information, and mouse simulation.
        - Use the Pydroid a code editor app on a mobile device to test the remote mouse app.
        - Use the trackpad on the mobile screen to control the remote computer.

3. **Activity 1: Create the mouse object** *(14 mins)*

    Teacher Activity: *(7 mins)*

    - Explore the steps to create the mouse object
    - Demonstrate how to import the modules and define the recvMessage() function to continuously receive messages from the client socket.

- Demonstrate how to create a mouse controller object and use it to check the data, and add instructions for **the left mouse click simulation.**

  Student Activity: *(7 mins)*

- Guide the student(s) to create a mouse object and **enable the mouse right click simulation**.

4. **Activity 2: Control the mouse position** *(12 mins)*

   Teacher Activity: *(6 mins)*

- Introduce the student(s) to the screen coordinates, which are used to position text boxes, images, and videos on the screen.
- Explain how to define a function that calculates the screen width and height of the computer's display to set* the Screen coordinates.
- Explain how to get the width of a screen and store its value in the variable screen_width using by extracting width from the monitor object.
- Demonstrate how to calculate the screen width and height to control the mouse pointer remotely on the computer screen.

  Student Activity: *(6 mins)*

- Guide the student(s) to calculate the screen height to control the mouse pointer remotely on the screen.

5. **Activity 3: Create a socket and connect to a server** *(12 mins)*

   Teacher Activity: *(6 mins)*

- Introduce the student(s) to the Kivy framework, which is an open-source Python framework for developing multi-touch applications, particularly those with graphical user interfaces (GUIs).
- Explain how to use the try block that attempts to create a socket and connect to the server to catch and handle exceptions if they occur.
- Demonstrate how to create a socket & client connection on the desktop and turn the Mobile into a remote mouse to control the computer.
- Note: To open the main.py file in a mobile device using the following ways:
  - Transfer the main.py file from the computer to mobile deceive using wifi, wireless file transfer apps (wifi), or using either micro-USB or USB-C as per the device.
  - Share the main.py file by uploading from a computer and downloading it from the mobile drive, like using, mail, messenger apps, or airdrop.
  - Use the main.py file downloaded earlier in the explore activity by typing the link https://bit.ly/main_py in the mobile browser.

  Student Activity: *(6 mins)*

- Guide the student(s) to create socket & client connections on their desktop and turn the mobile into a remote mouse to control the computer.

6. **Introduce the Post class project:** *(2 min)*
   - Create ~~the create~~ a remote keyboard application using a server and client connection, then make a mobile device as a remote keyboard.
7. **Test and Summarize the class learnings:** *(5 mins)*
   - Check for understanding through quizzes and summarize the learnings after the respective activities.
   - Summarize the overall class learning towards the end of the class.
8. **Additional activities:**
   - Encourage the student(s) to check why the right and left click are not working.
   - Encourage the student(s) to check why the mouse trackpad is not working correctly.

9. **State the Next Class Objective:** *(1 min)*
   - In the next class, student(s) will learn to set-up the SMTP server to send emails securely and handle errors and exceptions.

U.S. Standards:

CSTA: 2-AP-11, 2-AP-12, 2-AP-13, 2-AP-14, 2-AP-19

Links Table		
Activity	**Activity Name**	**Link**
Class Presentation	Remote Mouse	https://s3-whjr-curriculum-uploads.whjr.online/a3f9ba3a-ad37-4f78-b980-46cda9cf0797.html
Explore Activity	Remote Mouse	https://github.com/Tynker-Computer-Networks/TNK-M15-C117-SAS-BP
Teacher Activity 1	Create the Mouse object	https://github.com/Tynker-Computer-Networks/TNK-M15-C117-TAS-BP
Teacher Reference: Teacher Activity 1 Solution	Create the Mouse object	https://github.com/Tynker-Computer-Networks/TNK-M15-C117-TAS
Student Activity 1	Create the Mouse object	https://github.com/Tynker-Computer-Networks/TNK-M15-C117-SAS-BP
Teacher Reference: Student Activity 1 Solution	Create the Mouse object	https://github.com/Tynker-Computer-Networks/TNK-M15-C117-SAS
Teacher Activity 2	Control the Mouse Position	https://github.com/Tynker-Computer-Ne

		tworks/TNK-M15-C117-TAS-BP
Teacher Reference: Teacher Activity 2 Solution	Control the Mouse Position	https://github.com/Tynker-Computer-Networks/TNK-M15-C117-TAS
Student Activity 2	Control the Mouse Position	https://github.com/Tynker-Computer-Networks/TNK-M15-C117-SAS-BP
Teacher Reference: Student Activity 2 Solution	Control the Mouse Position	https://github.com/Tynker-Computer-Networks/TNK-M15-C117-SAS
Teacher Activity 3	Create a Socket and Connect to a Server	https://github.com/Tynker-Computer-Networks/TNK-M15-C117-TAS-BP
Teacher Reference: Teacher Activity 3 Solution	Create a Socket and Connect to a Server	https://github.com/Tynker-Computer-Networks/TNK-M15-C117-TAS
Student Activity 3	Create a Socket and Connect to a Server	https://github.com/Tynker-Computer-Networks/TNK-M15-C117-SAS-BP
Teacher Reference: Student Activity 3 Solution	Create a Socket and Connect to a Server	https://github.com/Tynker-Computer-Networks/TNK-M15-C117-SAS
Student's Additional Activity 1	Debug the Mouse Click!	https://github.com/Tynker-Computer-Networks/TNK-M15-C117-SAS-BP
Teacher Reference: Student's Additional Activity 1 Solution	Debug the Mouse Click!	https://github.com/Tynker-Computer-Networks/TNK-M15-C117-SAS
Student's Additional Activity 2	Debug the Trackpad!	https://github.com/Tynker-Computer-Networks/TNK-M15-C117-SAS-BP
Teacher Reference: Student's Additional Activity 2 Solution	Debug the Trackpad!	https://github.com/Tynker-Computer-Networks/TNK-M15-C117-SAS
Post Class Project	Remote Keyboard	https://github.com/Tynker-Computer-Networks/TNK-M15-C117-PCP-BP
Teacher Reference: Post Class Project Solution	Remote Keyboard	https://github.com/Tynker-Computer-Networks/TNK-M15-C117-PCP