TD10

Matière : DEVELOPPEMENT MOBILE AVANCE | **Classes : SEM31**

Exercice

On désire écrire une application Client/Serveur qui utilise les sockets pour permettre à une application android de contrôler une caméra de surveillance ip connectée à un pc. Le contrôle consiste à tourner la caméra ip d'une façon panoramique (de 0° à 255°) ou verticalement (de 0° à 90°) et de faire un zoom (de 0x à 4x).

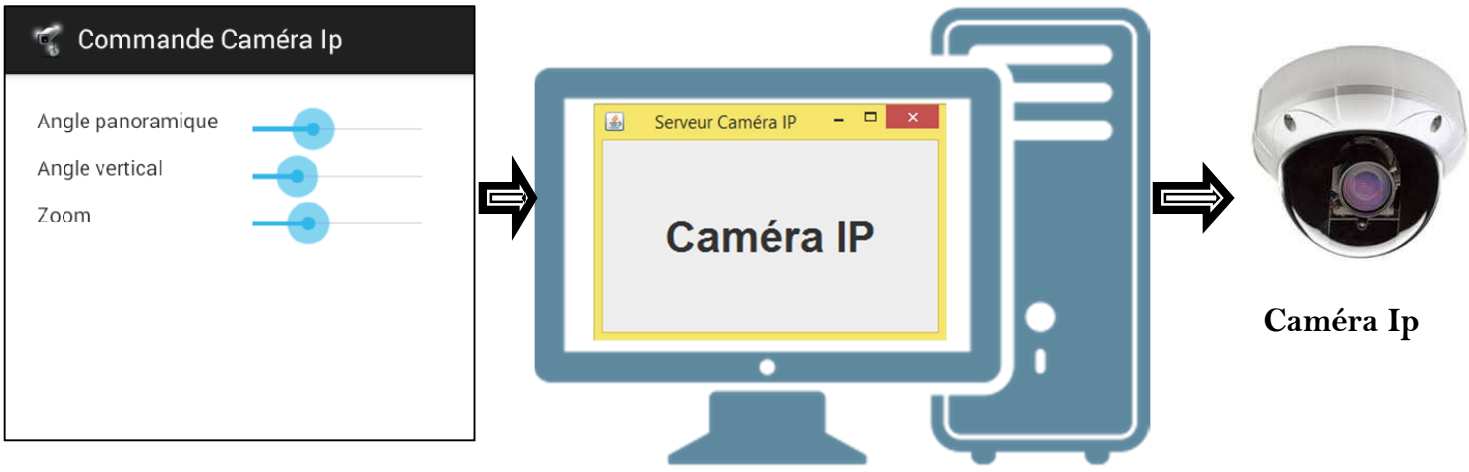
L'application serveur s'exécute sur un pc d'adresse ip "10.30.100.25" et fait l'écoute sur le port 6025. le code du serveur est dans la classe "**JFCameraIp**" qui possède une méthode `tournerPanoram(int valeur)` pour tourner la caméra ip d'une façon panoramique, une méthode `tournerVertical(int valeur)` pour tourner la caméra ip verticalement et une méthode `zoom(int valeur)` pour faire un zoom avec la caméra.

L'application client s'exécute sur le smartphone. Le code du client est dans la classe "**MainActivity**". "seekPanoram" permet d'indiquer l'angle de tour panoramique. "seekVertical" permet d'indiquer l'angle de tour vertical. "seekZoom" permet d'indiquer le coefficient du zoom.

Les messages envoyés du client au serveur sont de la forme suivante : "**Type:Valeur**" :

- Type : 0 pour tourner d'une façon panoramique, 1 pour tourner verticalement et 2 pour faire un zoom.
- Valeur : un entier qui représente l'angle de tour panoramique ou vertical ou le coefficient du zoom.

Le fonctionnement de l'application est comme suit :



1- Donner le code des méthodes « lancerThreadServeur()», « demarrerServeur()» et « executerCommande() » de la classe "**JFCameraIp**" (coté serveur).

2- Donner le code des méthodes « lancerThreadClient()», « demarrerClient()», « tournerPanoram() », « tournerVertical() » et « zoom() » de la classe "**MainActivity**" (coté client).

```
package com.cam;
//imports
public class JFCameraIp extends JFrame {
  private static final long serialVersionUID = 1L;
  private boolean actif;
  private static final int port = 6025;
  private ServerSocket ss;
  private Socket s;
  private BufferedReader br;
  public static void main(String[] args) {
    JFCameraIp f = new JFCameraIp();
    f.setVisible(true);
  }
  public JFCameraIp() {
    actif = true;
    setSize(300, 200);
    setLocation(500, 200);
    setTitle("Serveur Caméra IP");
    addWindowListener(new java.awt.event.WindowAdapter() {
      @Override
      public void windowClosing(java.awt.event.WindowEvent windowEvent) {
        actif = false;
        System.exit(0);
      }
    });
    ajouterLabel();
    lancerServeur();
  }
  private void ajouterLabel() {
    JLabel l = new JLabel("Caméra IP");
    l.setFont(new Font("dialog", Font.BOLD, 36));
    l.setHorizontalAlignment(JLabel.CENTER);
    l.setVerticalAlignment(JLabel.CENTER);
    getContentPane().add(l, BorderLayout.CENTER);
  }
  private void tournerPanoram(int valeur) {
    System.out.println("Tournée panoramique de: " + valeur + "°");
  }
  private void tournerVertical(int valeur) {
    System.out.println("Tournée verticale de: " + valeur + "°");
  }
  private void zoom(int valeur) {
    System.out.println("Zoom de: " + valeur);
  }
  private void lancerThreadServeur() {
  }
  private void demarrerServeur() {
  }
  private void executerCommande(String commande) {
  }

}
package com.cam;
//imports
public class MainActivity extends Activity {
  private SeekBar seekPanoram;
  private SeekBar seekVertical;
  private SeekBar seekZoom;
```

```
  private static final String ADR_SERVEUR = "10.30.100.25";
  private static final int PORT = 6025;
  private Socket s;
  private PrintWriter pw;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    init();
  }
  private void init() {
    seekPanoram = (SeekBar) findViewById(R.id.seekPanoram);
    seekVertical = (SeekBar) findViewById(R.id.seekVertical);
    seekZoom = (SeekBar) findViewById(R.id.seekZoom);
    seekPanoram.setMax(355);
    seekVertical.setMax(90);
    seekZoom.setMax(3);
    ajouterEcouteur();
    lancerClient();
  }
  private void ajouterEcouteur() {
    seekPanoram.setOnSeekBarChangeListener(new OnSeekBarChangeListener() {
      @Override      public void onStopTrackingTouch(SeekBar seekBar) {    }
      @Override      public void onStartTrackingTouch(SeekBar seekBar) {   }
      @Override
      public void onProgressChanged(SeekBar seekBar, int progress, boolean
fromUser) {         tournerPanoram();
       }
    });
    seekVertical.setOnSeekBarChangeListener(new OnSeekBarChangeListener() {
      @Override      public void onStopTrackingTouch(SeekBar seekBar) {     }
      @Override
      public void onStartTrackingTouch(SeekBar seekBar) {    }
      @Override
      public void onProgressChanged(SeekBar seekBar, int progress, boolean
fromUser) {          tournerVertical();
       }
    });
    seekZoom.setOnSeekBarChangeListener(new OnSeekBarChangeListener() {
      @Override      public void onStopTrackingTouch(SeekBar seekBar) {     }
      @Override      public void onStartTrackingTouch(SeekBar seekBar) {     }
      @Override
      public void onProgressChanged(SeekBar seekBar, int progress, boolean
fromUser) {         zoom();
       }
    });
  }
```

```
  private void lancerThreadClient() {
  }
  protected void demarrerClient() {
  }
  protected void tournerPanoram() {
  }
  protected void tournerVertical() {
  }
  protected void zoom() {
  }
}
```