Direction Générale des Etudes Technologiques

Institut Supérieur des Etudes Technologiques de Sfax

République Tunisienne

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Nom et Prénom : ---------- Groupe : ----------	Numéro de compostage	Note /20

AU 2022/2023 SEMESTRE I

Devoir Final en *Développement Mobile Avancé*

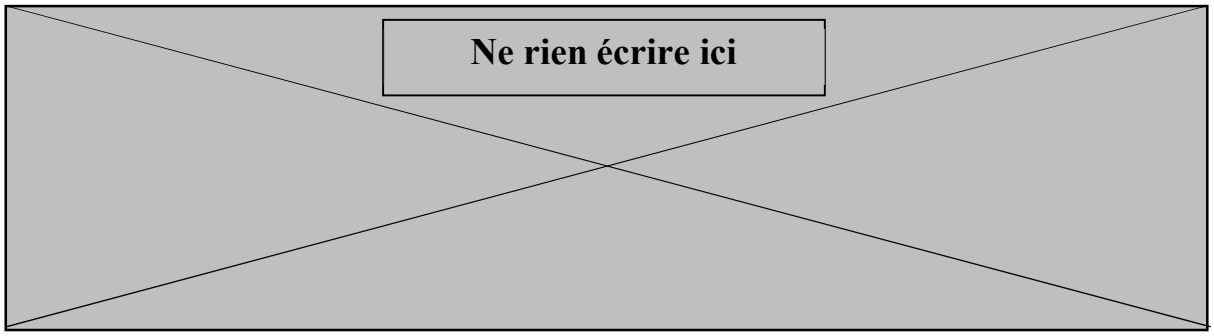
Groupes : SEM31 **Durée : 1h30**

Enseignant : Souissi Hafedh **Documents non autorisés**

Numéro de compostage	Note /20

Exercice1

```
private void analyser() {
  Bitmap bitmap          = ((BitmapDrawable) imgVisages.getDrawable()).getBitmap();
  FaceDetector.Builder b = new FaceDetector.Builder(this)                          ;
  b.setTrackingEnabled   (true                                                    );
  b.setLandmarkType      (FaceDetector.ALL_LANDMARKS                              );
  b.setClassificationType(FaceDetector.ALL_CLASSIFICATIONS                        );
  FaceDetector detector   = b.build()                                              ;
  Frame frame             = new Frame.Builder().setBitmap(bitmap).build()          ;
  SparseArray<Face> faces = detector.detect(frame)                                 ;
  int nbV = faces.size()                                                           ;
  int nbVS= 0                                                                      ;
  int nbYF= 0                                                                      ;
  int nbBF= 0                                                                      ;
  for (int i = 0; i < faces.size(); i++                                           ){
      Face face = faces.valuAt(i)                                                  ;
      if(face.getIsSmilingProbability()>0.6                                        )
            nbVS++                                                                 ;
      if(face.getIsRightEyeOpenProbability()<0.5                                   )
            nbYF++                                                                 ;
      if(face.getIsLeftEyeOpenProbability()<0.5                                    )
           nbYF++                                                                  ;
      double xNB = 0, yNB = 0, xBM = 0, yBM = 0                                    ;
      for (Landmark l : face.getLandmarks()                                       ){
          if (l.getType() == Landmark.NOSE_BASE                                    ){
              xNB = l.getPosition().x                                              ;
              yNB = l.getPosition().y                                              ;
          }
          if (l.getType() == Landmark.BOTTOM_MOUTH                                ){
              xBM = l.getPosition().x                                              ;
              yBM = l.getPosition().y                                              ;
          }
      }
```




```
        double hauteur= face.getHeight()                                    ;
        double dNBMB  = Math.sqrt(Math.pow(xNB - xBM, 2) + Math.pow(yNB - yBM, 2  ;
        double cNBMBH = dNBMB / hauteur                                     ;
        if(cNBMBH<0.27                                                      )
            nbBF++                                                          ;
    }
    tvNbV.setText (nbV+"")                                                  ;
    tvNbVS.setText(nbVS+"")                                                 ;
    tvNbYF.setText(nbYF+"")                                                 ;
    tvNbBF.setText(nbBF+"")                                                 ;


}
```

Exercice2

//MainActivity

```
private void ajouterEcouteur() {
    br = new BroadcastReceiver()                                            {
        @Override
        public void onReceive(Context context, Intent intent)              {
            actualiser(intent)                                              ;
        }
    };
}
private void demarrerService() {
    Intent i = new Intent(this, EtatMerService.class)                       ;
    startService(i)                                                         ;
}
private void arreterService() {
    Intent i = new Intent(this, EtatMerService.class)                       ;
    stopService(i)
}
@Override
protected void onResume() {
    demarrerService()                                                       ;
    registerReceiver(br, new IntentFilter(EtatMerService.ACTION_ETAT_MER))  ;
    super.onResume();
}
```

```
@Override
protected void onPause() {
  arreterService()                                                    ;
  unregisterReceiver(br)                                              ;
  super.onPause();
}
protected void actualiser(Intent intent) {
  double normeAcc = intent.getDoubleExtra("normeAcc", 0)              ;
  String etatMer  = intent.getStringExtra("etatMer")                  ;
  tvEtat.setText(normeAcc + " => " + etatMer)                         ;
  if (normeAcc < 13                                                   )
      imgEtat.setImageResource(IMG_MER_CALME)                         ;
  else if (normeAcc < 18                                              )
      imgEtat.setImageResource(IMG_MER_AGITEE)                        ;
  else
      imgEtat.setImageResource(IMG_MER_TRES_AGITEE)                   ;

}
```

//EtatMerService

```
private void init() {
  smg = (SensorManager) getSystemService(SENSOR_SERVICE)             ;
  acc = smg.getDefaultSensor(Sensor.TYPE_ACCELEROMETER)               ;
  smg.registerListener(this, acc, SensorManager.SENSOR_DELAY_UI)
}
@Override
public void onDestroy() {
  smg.unregisterListener(this, acc)                                   ;
  super.onDestroy();
}
@Override
public void onSensorChanged(SensorEvent event) {
  if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER            ){
      double x = event.values[0]                                      ;
      double y = event.values[1]                                      ;
      double z = event.values[2]                                      ;
      double normeAcc=Math.sqrt(Math.pow(x,2)+Math.pow(y,2)+Math.pow(z,2))  ;
      String etatMer="";
      if(normeAcc<13                                                  )
        etatMer="Mer Calme"                                           ;
      else if(normeAcc<17)
         etatMer="Mer Agitée"                                         ;
      else
        etatMer="Mer Très Agitée"                                     ;
      Intent i = new Intent()                                         ;
      i.setAction(ACTION_ETAT_MER)                                    ;
      i.putExtra("normeAcc", normeAcc)                                ;
      i.putExtra("etatMer", etatMer)                                  ;
      sendBroadcast(i)                                                ;


}
```

Exercice3

//ServPresentation

```
private void lancerServeur() {
  Runnable r = new Runnable(){
    @Override
    public void run() {
         demmarerServeur();
    }
  };
  Thread th = new Thread(r);
  th.start();
}
private void demarrerServeur() {
   try {
          ss = new ServerSocket(PORT)                                          ;
          s = ss.accept()                                                      ;
          br = new BufferedReader(new InputStreamReader(s.getInputStream()))   ;
          while (true)                                                         {
            String cmd;
            cmd = br.readLine()                                                ;
            executerCommande(cmd)                                              ;
          }
    } catch (IOException e) {
         e.printStackTrace();
    }
}
private void executerCommande(String commande) {
  String[] t = commande.split(":")                                             ;
  if (t.length > 1){
    int cont = Integer.parseInt(t[0])                                          ;
    int oper = Integer.parseInt(t[1])                                          ;
    switch (cont                                                             ) {
      case 0:
        if (oper == 0)
          precedent();
        else
          suivant();
        break;
      case 1:
        if (oper == 0)
          premier();
        else
          dernier(param);
        break;
      case 2:
         if(t.length>2){
            int param = Integer.parseInt(t[2]);
            if(oper==0)
              atteindre(param);
        }
        break;

}
```

Nom et Prénom : ---------- Groupe : ----------	Numéro de compostage	Note /20

Numéro de compostage

//MainActivity

```
private void lancerThreadClient() {
   Runnable r = new Runnable() {
     @Override
     public void run() {
           demmarerClient();
     }
   };
   Thread th = new Thread(r);
   th.start();
 }
protected void demarrerClient() {
  try {
      Inet4Address i = (Inet4Address)
                        Inet4Address.getByName(edAdresse.getText().toString());
      s = new Socket(i, Integer.parseInt(edPort.getText().toString()));
      pw = new PrintWriter(new BufferedWriter(new OutputStreamWriter
                           (s.getOutputStream())), true);
  } catch (UnknownHostException e) {
            e.printStackTrace();
   } catch (IOException e) {
            e.printStackTrace();
   }
}
private void envoyer(final String cmd) {
   Runnable r = new Runnable() {
      @Override
      public void run() {
        if (pw != null) {
            pw.println(cmd);
         }
      }
    };
    Thread th = new Thread(r);
    th.start();
}
private void precedent()   {          envoyer("0:0")                    ;}
protected void suivant()   {          envoyer("0:1")                    ;}
protected void premier()   {          envoyer("1:0")                    ;}
protected void dernier()   {          envoyer("1:1")                    ;}
protected void atteindre() {
     envoyer("2:0:" + edNum.getText().toString());


}
```