

TP Android

Description : Nous allons réaliser une application mobile permettant d'afficher les informations obtenues du robot martien InSight.

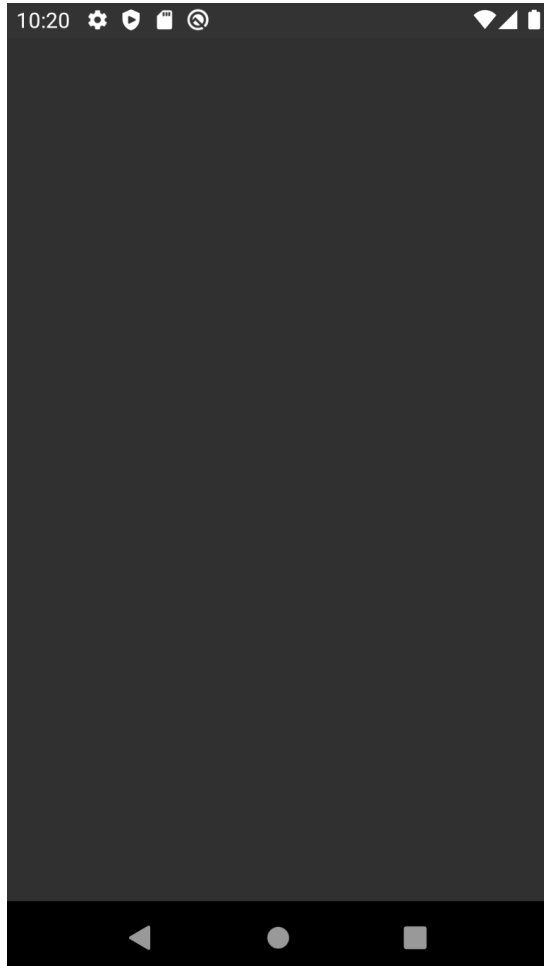
Exercice 1 :

L'application mobile ouvrira sur un premier écran d'accueil. L'écran devra être identique à ceci :



Exercice 2 :

Lorsque l'on clique sur le bouton « Commencer », nous devons rediriger l'utilisateur sur un deuxième écran : la « Home ».



Exercice 3 :

Avant d'afficher les données météo de Mars nous devons obtenir cette information sur une source fiable. Nous allons utiliser pour cela, l'API mise à disposition par la NASA !

Vous pouvez obtenir en un clic une API_KEY via l'url suivante :

<https://api.nasa.gov/index.html#apply-for-an-api-key>

Nous allons utiliser la methode suivant de l'API pour récupérer la liste de nos Asteroides :

https://api.nasa.gov/insight_weather/?api_key=DEMO_KEY&feedtype=json&ver=1.0

Remplacer DEMO_KEY par votre API_KEY. Récupérer la réponse et afficher un Toast indiquant que les données sont bien téléchargées.

Exercice 4 :

Extrayez les informations depuis la réponse de l'api de la NASA. Voici une description de la structure :

```
{
  "471": {↔},
  "472": {↔},
  "473": {↔},
  "474": {↔},
  "475": {↔},
  "476": {↔},
  "477": {↔},
  "sol_keys": [
    "471",
    "472",
    "473",
    "474",
    "475",
    "476",
    "477"
  ],
  "validity_checks": {↔}
}
```

Vous devriez avoir ce genre de réponse. Les clés : 471, 472, 473, sont ce qu'on appelle des « Sol ». Un Sol sur Mars, est l'équivalent d'une Journée sur Terre. Les clés « 471 », etc ... sont obtenues en parcourant le tableau « sol_keys ».

Si on ouvre un Sol on obtient un premier attribut :

```
"AT": {  
  "av": -58.442,  
  "ct": 308034,  
  "mn": -94.883,  
  "mx": -7.102  
},
```

Il s'agit des données de température. « av » pour « average » donc il s'agit de la moyenne pour le Sol. « mn » pour minimum et « mx » pour maximum.

Ensuite vous avez :

```
"PRE": {  
  "av": 640.798,  
  "ct": 153684,  
  "mn": 614.5621,  
  "mx": 669.9838  
},
```

Il s'agit cette fois des données de Pression pour le Sol

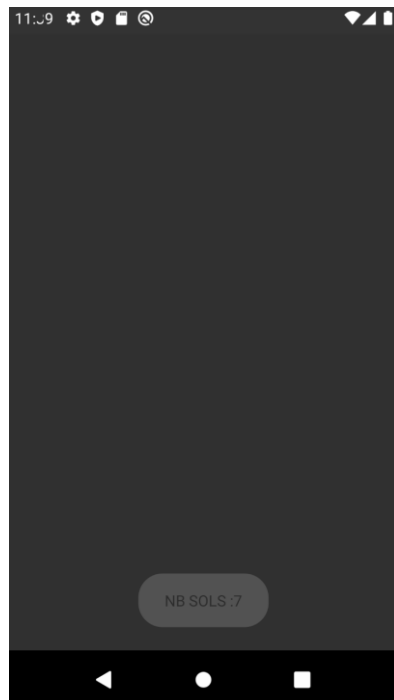
```

"WD": {
  "0": {
    "compass_degrees": 0.0,
    "compass_point": "N",
    "compass_right": 0.0,
    "compass_up": 1.0,
    "ct": 1466
  },
  "1": {
    "compass_degrees": 22.5,
    "compass_point": "NNE",
    "compass_right": 0.382683432365,
    "compass_up": 0.923879532511,
    "ct": 542
  },
  "2": {
    "compass_degrees": 45.0,
    "compass_point": "NE",
    "compass_right": 0.707106781187,
    "compass_up": 0.707106781187,
    "ct": 538
  },
  "3": {
    "compass_degrees": 67.5,
    "compass_point": "ENE",
    "compass_right": 0.923879532511,
    "compass_up": 0.382683432365,
    "ct": 2704
  },

```

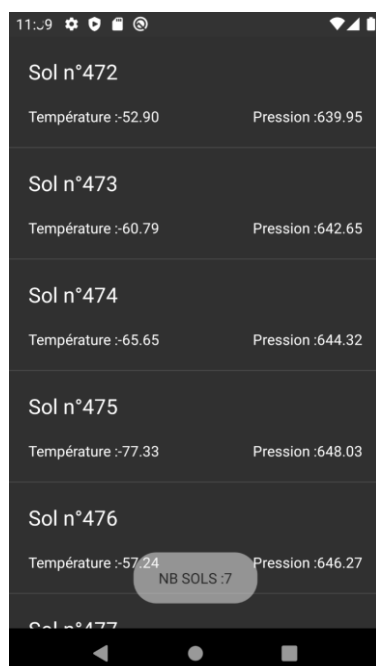
Cette fois il s'agit des données du « Vent », avec « WD » pour « Wind Direction ». Vous aurez cette fois 16 clés pour 16 directions différentes. Attention il se peut qu'il manque des « clés », testez avec la méthode « has » et renseignez alors 0 comme valeur car il n'y a pas eu de vent dans cette direction. Le `compass_degrees` permet de connaître la direction en degrés. La valeur « ct » correspond à la force du vent.

Créez une structure objet comme il vous plaira pour stocker ces informations.



Exercice 5 :

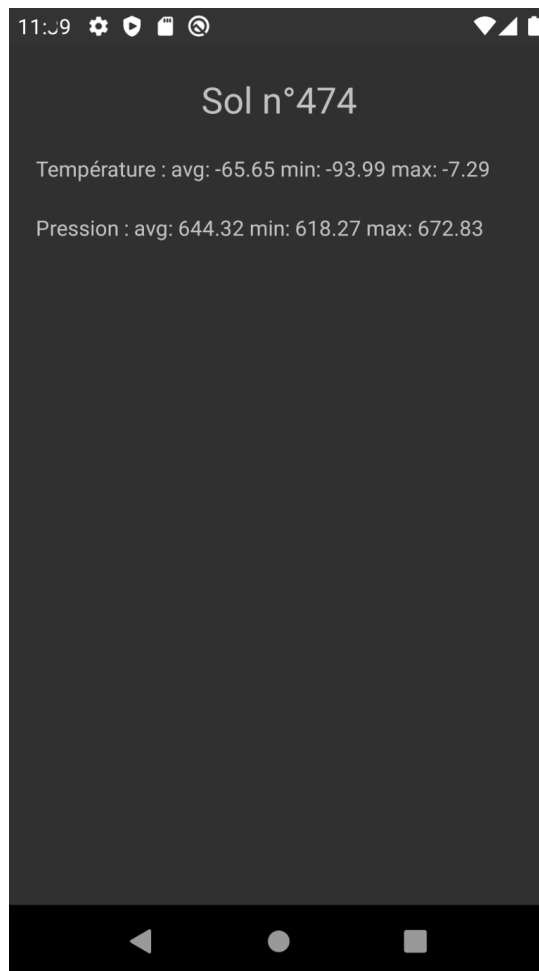
Ajoutons maintenant les sols ! Pour cela utiliser une ListView dans laquelle on affichera les infos suivantes :



Les valeurs renseignées ici sont les « moyennes ».

Exercice 6 :

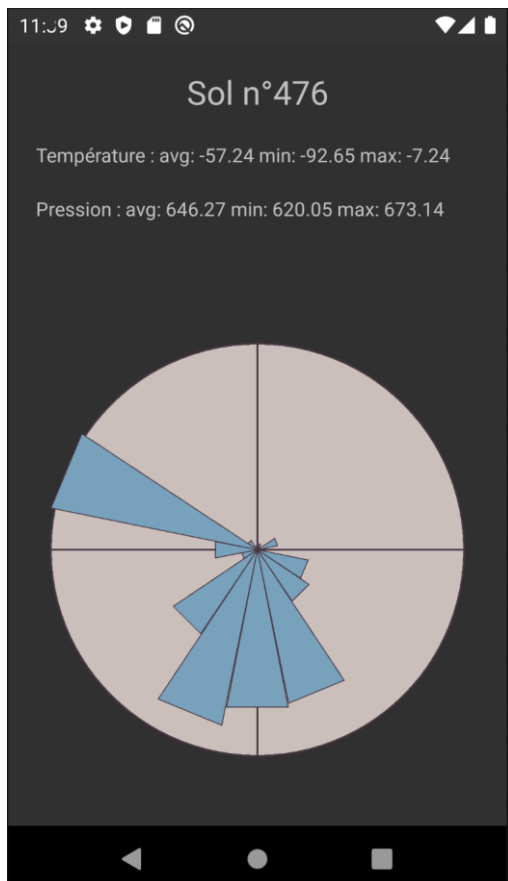
Une fois que l'on clique sur une ligne, nous voulons ouvrir une nouvelle page de detail qui affichera les éléments suivants.



Exercice 7 :

Illustrons maintenant les directions du vent avec un graphique qui va ressembler à https://api.nasa.gov/assets/insight/insight_mars_wind_rose.html

Un triangle représente une valeur de vent. Dessinez le cercle extérieur qui aura pour rayon $\text{width}/2$ et pour « valeur » la valeur maximale parmi nos 16 valeurs. Affichez ensuite 16 triangles pour les 16 valeurs tout au long du cercle. La clé 0 étant celle en direction du Nord.



Pour tracer un triangle, voici un exemple :

```
Path path = new Path();
path.lineTo(-tmpX, -radius);
path.lineTo(tmpX, -radius);
path.lineTo(0, 0);
path.close();
canvas.drawPath(path, innerPaint);
```