

TP Informatique – React Native

MapView + Géoloc

Etape 0 : mise en place

- ✓ Créer la structure d'une nouvelle application nommée « mapview »
- ✓ Modifier le composant principal « app.js » avec ce rendu :

```
import MapView, {Marker} from 'react-native-maps'
```

```
<View style={styles.container}>  
  <MapView  
    style={styles.mapStyle}  
    region={this.state.region}>  
  </MapView>  
</View>
```

Avec le style pour la carte à ajouter la variable styles :

```
mapStyle: {  
  width: Dimensions.get('window').width,  
  height: Dimensions.get('window').height,  
  flex: 1  
},
```

Et avec un state contenant les coordonnées et la précision de la carte à afficher :

```
state = {  
  region : {  
    latitude: 43.621350,  
    longitude: 2.261258,  
    latitudeDelta: 0.001,  
    longitudeDelta: 0.001,  
  },  
}
```

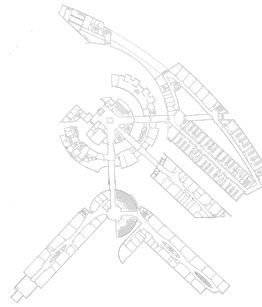
- ✓ Tester

Etape 1 : ajouter un marqueur

- ✓ Ajouter dans le state un objet « position » avec les mêmes valeurs que la region pour latitude et longitude.
- ✓ Dans le composant MapView, ajouter composant « Marker » avec comme props « coordinate » (ayant pour valeur la position stockée dans le state) et « title » (ayant pour valeur « ici »)

Etape 2 : ajouter un overlay

- ✓ L'objectif d'ajouter sur la carte un plan du bâtiment de l'IUT :



- ✓ Pour cela, dans le composant « MapView », il faut utiliser le composant « Overlay » qui a besoin de 2 props :

- image = chemin d'accès au fichier (fichier sur moodle à mettre dans le dossier assets)

```
image={require('./assets/IUTCastresGrisClair.png')}
```

- bounds = tableau des coordonnées du coin haut-gauche et du coin bas-droit pour positionner l'overlay

```
const boundsIUT = [ [ 43.622050,2.260360 ], [43.620220,2.262510] ]
```

marche mieux avec ces coordonnées :

```
const boundsIUT = [[43.620220,2.260360],[ 43.622050,2.262510] ]
```

Étape 3 : prise en compte de la position de l'utilisateur (soumis à son autorisation)

Expo propose le module « expo-location » permettant de récupérer les informations de position du smartphone. Mais son utilisation est soumise à l'acceptation de l'utilisateur, il faut donc lui demander sa permission.

```
_getLocationAsync() {
  Location.requestForegroundPermissionsAsync()
    .then (response => {
      if (response.status !== 'granted') {
        this.setState({ errorMessage: 'Permission to access location was denied' });
      }
      else {
        Location.watchPositionAsync( {accuracy : Location.Accuracy.High }, (location) => {
          console.log(location)
        })
      }
    })
}
```

Intégrer ce code dans l'application et l'adapter pour que le marqueur (et la carte) suive la position de l'utilisateur.

Étape 4 : ajout de marqueurs

Il s'agit ici de positionner des marqueurs sur les divers lieux et salles du dept MMI.

Intégrer tout d'abord les ressources ci-dessous ;

Les icônes à utiliser sont disponibles sur moodle ; à stocker dans les assets.

```
import WifiIcon from './assets/wifi.png'
import WCIcon from './assets/toilets.png'
import SnackIcon from './assets/snack_bar.png'
import BlueIcon from './assets/blue-pushpin.png'

const marqueurs = [
  { // borne WIFI MMI TD rouge
    position: { latitude:43.620680, longitude:2.261520 },
    icon: WifiIcon,
    label: "Borne WIFI"
  },
  { // bornes WIFI MMI central
    position: { latitude:43.620460, longitude:2.261850},
    icon: WifiIcon,
    label: "Borne WIFI"
  },
  { // toilettes MMI TD rouge
    position: { latitude:43.620780, longitude:2.261480},
    icon: WCIcon,
  },
  { // toilettes MMI central
    position: { latitude:43.620580, longitude:2.261680},
    icon: WCIcon,
  },
  { // toilettes MMI fond couloir
    position: { latitude:43.620400, longitude:2.261940},
    icon: WCIcon,
  },
  { // salle etudiants MMI
    position: { latitude:43.620360, longitude:2.262100},
    icon: SnackIcon,
  },
  { // grand amphi
    position: { latitude:43.620840, longitude:2.261310},
    label: "Grand amphi",
    icon: BlueIcon,
  },
  { // petit amphi
    position: { latitude:43.620740, longitude:2.261310},
    label: "Petit amphi",
    icon: BlueIcon,
  },
]
```

Compléter votre code pour que tous les marqueurs s'affichent sur la carte.

Pour que les étiquettes des marqueurs soient toujours visibles (par exemple pour les amphis), utiliser un composant « Text ».

Idem, pour pouvoir appliquer un style aux icônes des marqueurs, les afficher avec un composant « Image »

```
<Marker
  key = {index}
  coordinate = {m.position}
  title = {m.label}

  ><Text>{m.label}</Text>
  <Image source={m.icon} style = { styles.marker }></Image>
</Marker>
```

Avec par exemple, ces styles pour les icônes :

```
marker: {
  width: 40,
  height: 40
}
```