

Rapport Developpement mobiles : Memories

Allan BOOZ, Yanis BOUDEMIA ETTOUIL

23 novembre 2019

Résumé

Il s'agit pour ce projet de concevoir une application Android iOS faisant appel à différentes notions.

1 Introduction

L'application que nous avons développée se nomme "Memories". C'est une application qui permet à son utilisateur de prendre une photo et de plus tard, regarder sur une carte Google Maps l'endroit où cette photo a été prise. Ainsi pour permettre le bon fonctionnement de ce projet, nous devons respecter quelques conditions :

- Changements de configuration (Android) : l'application doit gérer ces changements. (changement de langue, passage de portrait à paysage...)
- Internationalisation : l'application doit être disponible en français et en anglais.
- Géolocalisation : l'application doit au moins récupérer et utiliser les positions de l'utilisateur.
- Capteurs : l'application doit utiliser au moins un capteur. (accéléromètre...)
- Lecture audio.
- Gestes courants : l'application doit faire appel à au moins un geste non-trivial. (double tap, balayage...)
- Appareil photo.

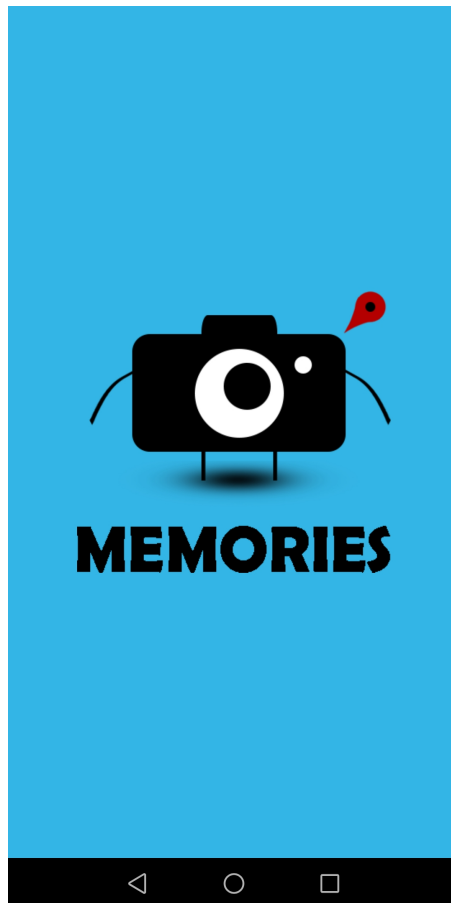


FIGURE 1 – Splashscreen du projet Memories

2 Le corps du projet

2.1 Camera

Dans un premier temps, nous aurons besoin d'une activité "Camera" qui permettra à l'utilisateur de prendre ou non ça photo. Au premier lancement de l'application, un pop-up se lance afin de demander l'autorisation à l'utilisateur d'accéder aux photos, média et autres, soit l'écriture en stockage interne du telephone.

```
@Override
protected void onResume() {
    super.onResume();
    if (ContextCompat.checkSelfPermission( context: this,Manifest.permission.WRITE_EXTERNAL_STORAGE) != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, MY_CAMERA_REQUEST_CODE);
    }
    mSensorManager.registerListener(mShakeDetector, mAccelerometer, SensorManager.SENSOR_DELAY_UI);
}
```

FIGURE 2 – Permission

Puis, arriver sur l'activité Camera, une image explique à l'utilisateur comment effacer une image qu'il a prise en appuyant sur le bouton "ouvrir camera", c'est-à-dire en secouant son telephone.

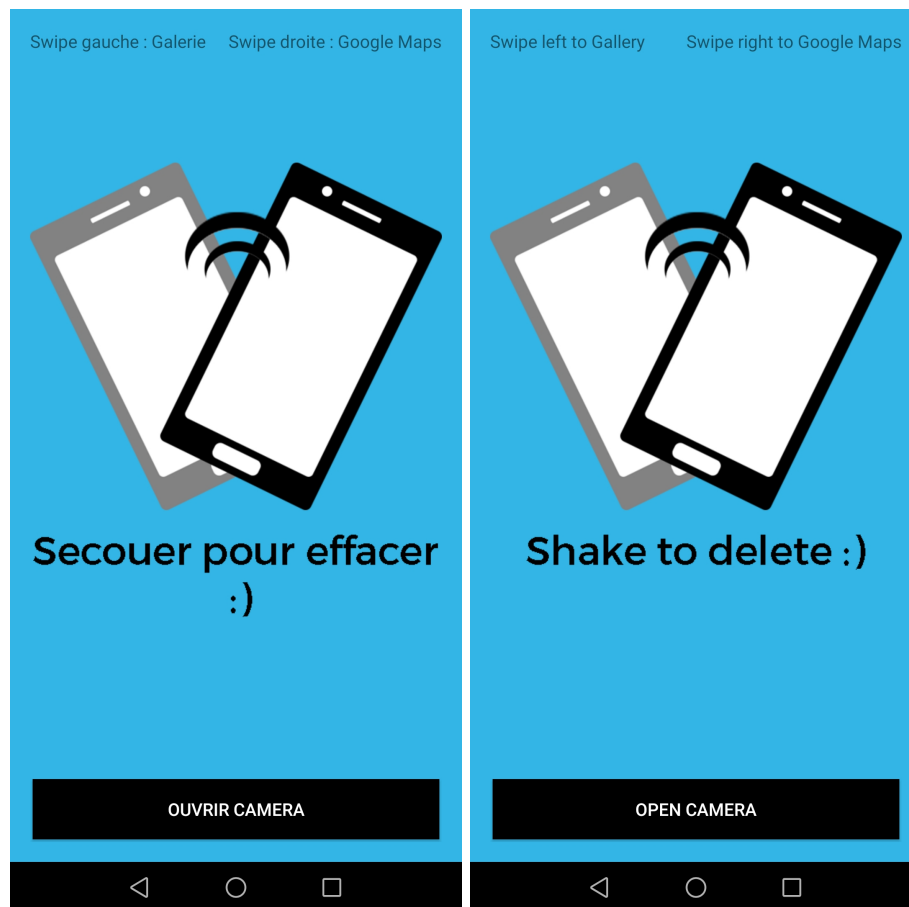


FIGURE 3 – Comment effacer une photo ?

Enfin lorsque l'utilisateur a appuyé sur le bouton "ouvrir camera", son application "photo" se lance et il ne lui reste plus qu'à prendre la photo et de la valider. Après cela, il revient à l'activité "Camera" et voit la photo prise. A ce moment-là, un bouton "enregistrer" apparaît et lui laisse le choix de le faire ou de secouer son telephone pour la supprimer.

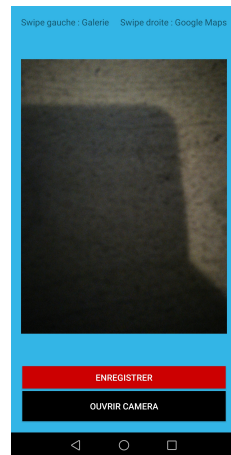


FIGURE 4 – Enregistrement d'une photo mode portrait

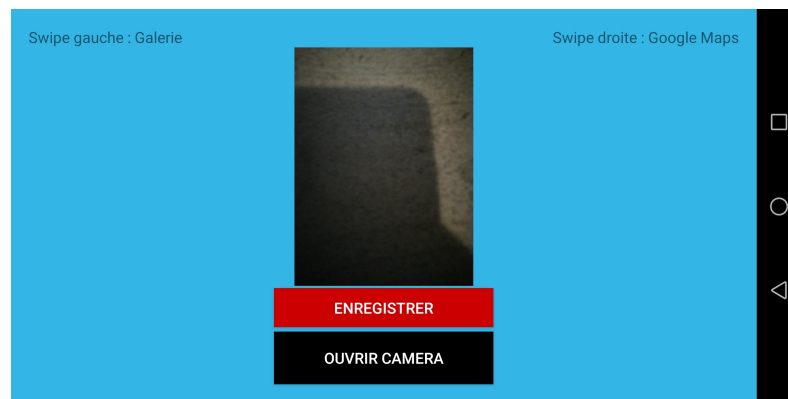


FIGURE 5 – Enregistrement d'une photo mode paysage

L'action de secouer est gérée par une méthode d'Android nommée : onShakeListener. A l'intérieur il s'agira de faire en sorte que la photo prise soit libérée de notre variable "image" et de générer un son pour confirmer la suppression de l'image.

```
@Override
public void onShake(int count) {

    if(test_soni == null && photoUri!=null){

        if(Locale.getDefault().getDisplayLanguage().equals("English")){
            test_soni = MediaPlayer.create(getApplicationContext(),R.raw.delete_voice);
            test_soni.start();

            test_soni = null;
            image = null;
            photoUri = null;
            imageView.setImageBitmap(image);

            Toast.makeText(getApplicationContext(), "Current picture has been deleted !", Toast.LENGTH_LONG).show();
        }else{
            MediaPlayer.create(getApplicationContext(),R.raw.delete_voice);
        }
    }
}
```

FIGURE 6 – Le shake

Enfin avant de pouvoir enregistrer notre Photo, il s'agira de lancer l'application appareil photo, de sauvegarder dans un fichier temporaire, et de l'envoyer à l'intent pour ensuite la récupérer et la sauvegarder.

```
// OUVERTURE DE L'APPLICATION APPAREIL PHOTO
btn_camera.setOnClickListener((v) -> {
    //Ouverture d'une fenetre pour prendre la photo
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);

    // - intent.resolveActivity retourne le composant qui doit être utilisé pour traiter l'intent.
    // - Si getPackage() est non-NULL, seuls les composants de cette activité seront pris en compte.
    if(intent.resolveActivity(getPackageManager()) != null){
        String name = new SimpleDateFormat( pattern "yyyyMMdd HHmmss").format(new Date()); // créer un nom de fichier
        File repertoirePhoto = getExternalFilesDir(Environment.DIRECTORY_PICTURES); // définit le répertoire

        try {
            // créer un fichier représentant la photo
            photoFile = File.createTempFile( prefix "photo"+name, suffix ".jpg", repertoirePhoto);
            photoPath = photoFile.getAbsolutePath(); // Initialise le chemin complet de la photo

            //Accès au fichier
            photoUri = FileProvider.getUriForFile( context: Camera.this,
                authority: Camera.this.getApplicationContext().getPackageName()+".provider", photoFile);

            //Enregistre la photo dans le fichier temporaire
            intent.putExtra(MediaStore.EXTRA_OUTPUT, photoUri);

            //Ouvrir l'activité par rapport à l'intent
            startActivityForResult(intent, PICTURE_RESULT);

        } catch (IOException e) {e.printStackTrace();}
    }
});
```

FIGURE 7 – sauvegarde + prendre la photo

```
// ENREGISTREMENT DE LA PHOTO + ECOUTE SUR LE BOUTON ENREGISTRER
btn_save.setOnClickListener((v) -> {
    //MediaStore.Images.Media.insertImage(getContentResolver(), image, "nom image", "description");

    // name est un nom quelconque : on a choisit comme nom le format date.
    String name = new SimpleDateFormat( pattern "yyyyMMdd_HHmmss").format(new Date());
    savePicture(image, imgName: name+".jpg");

    setImgLatLng(name+".jpg");
    image = null;
    imgview.setImageBitmap(image);

    Toast.makeText(getApplicationContext(), text: "Picture saved !", Toast.LENGTH_LONG).show();

    btn_save.setEnabled(false);
    btn_save.setVisibility(View.INVISIBLE);

});
```

FIGURE 8 – sauvegarde + prendre la photo

2.2 Galerie

En partant de l'activité "Camera", si l'utilisateur effectue un "swipe" vers la gauche, il change d'activité et se retrouve dans l'activité "Galerie" où toutes ses images prises avec l'application sont présentes. Il s'agit alors de créer un dossier spécifique pour cette application. Les photos sont stockés dans une GridView et il est possible à l'utilisateur de cliquer sur la photo afin de la voir en "un peu plus grand" dans une autre activité (il pourra faire retour à tout moment).

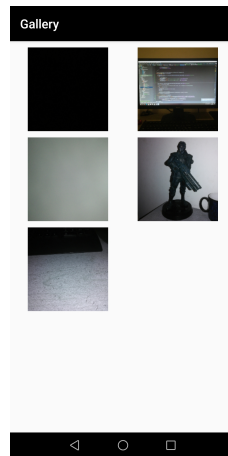


FIGURE 9 – Les photos de la galerie

2.3 Google Maps

Enfin partant de l'activité "Camera" ou de "Galerie" en faisant un swipe vers la droite, nous observons une map Google Maps (avec un demande d'autorisation d'utiliser la localisation la première fois) où l'utilisateur peut voir toutes ses photos prises à différents endroits dans le monde.

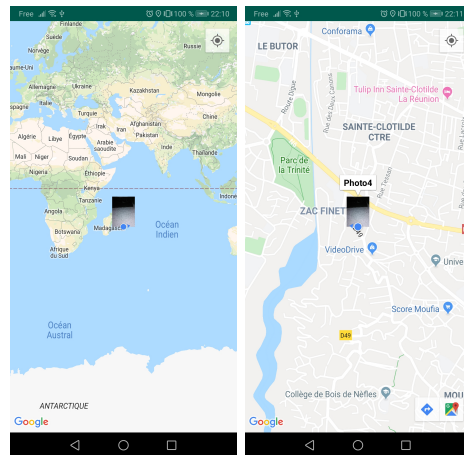


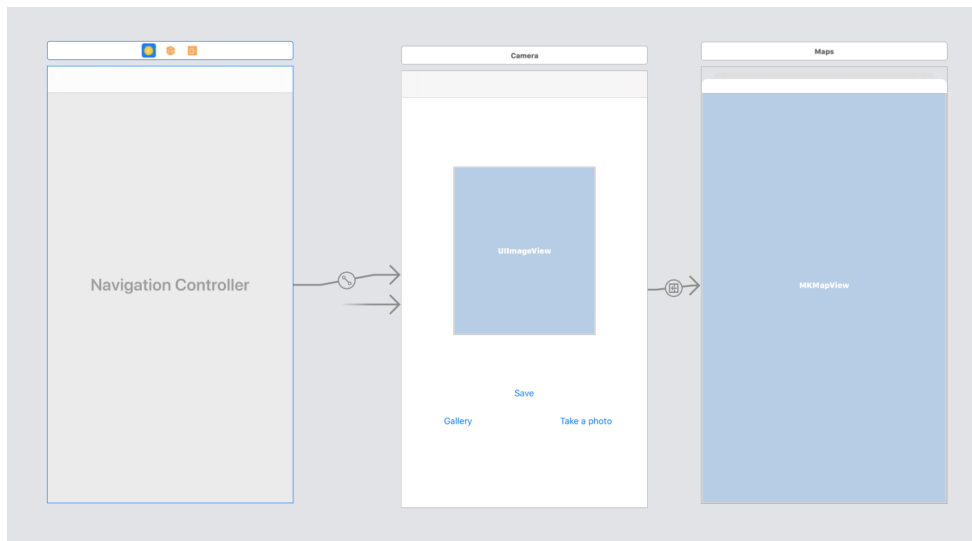
FIGURE 10 – Google Maps

2.4 iOS

2.4.1 Architecture de l'application

Pour la version iOS nous avons renoncé à la création d'un répertoire propre à l'application et de prendre comme base la galerie du téléphone directement. Il est donc possible de consulter toutes les photos présentes sur le téléphone et de rajouter à celles-ci les photos prises par *Memories*. Notre application se présente donc de la manière suivante :

- Camera = Menu de l'application mais aussi le controller permettant la prise ou consultations de photos
- Maps = La carte du monde



2.4.2 L'appareil photo

Dans *Memories* tout repose sur les photos de l'utilisateur. C'est pourquoi nous avons besoin de l'appareil photo du téléphone.

Ici nous testons sur l'action du bouton *Take a photo* si la camera est accessible sur l'appareil concerné. Si elle est accessible, alors nous nous occupons d'ouvrir l'appareil photo et laissons donc l'utilisateur prendre la photo de son choix. Dans le cas où la camera de l'appareil n'est pas disponible (sur un émulateur par exemple) nous redirigeons vers la galerie de l'appareil et permettons à l'utilisateur de choisir la photo qu'il désire.

```
@IBAction func takePhoto(_ sender: UIButton) {
    guard UIImagePickerController.isSourceTypeAvailable(.camera) else {
        selectImageFrom(.photoLibrary)
        return
    }
    selectImageFrom(.camera)
}
```



```

func selectImageFrom(_ source: ImageSource){
    imagePicker = UIImagePickerController()
    imagePicker.delegate = self as UIImagePickerControllerDelegate & UINavigationControllerDelegate
    switch source {
    case .camera:
        imagePicker.sourceType = .camera
    case .photoLibrary:
        imagePicker.sourceType = .photoLibrary
    }
    present(imagePicker, animated: true, completion: nil)
    saveButton.isEnabled = true
}

```

Une fois la photo prise on peut voir apparaître sur l'*ImageView* de *Camera* notre photo. Nous à cet instant 2 choix qui se présentent :

- Enregistrer l'image dans la galerie du téléphone avec le bouton *Save*
- Ne pas vouloir garder la photo actuellement sur l'écran

Dans le deuxième cas il suffit de secouer le téléphone pour ne pas enregistrer la photo dans le téléphone.

```

override func motionEnded(_ motion: UIEvent.EventSubtype, with event: UIEvent?) {
    if motion == .motionShake {
        //Suppression de la photo prise
        imageTake.image = nil
        saveButton.isEnabled = false
    }
}

```

Enfin il reste à mettre en place une carte du monde sur laquelle on pourra dans de futures versions de l'application afficher les différents lieux où l'utilisateur a pris ses photos. Nous allons ici utiliser MapKit pour créer notre carte

```

if CLLocationManager.locationServicesEnabled() {
    locationManager.delegate = self
    locationManager.desiredAccuracy = kCLLocationAccuracyHundredMeters

    locationManager.requestWhenInUseAuthorization()

    mapView.showsUserLocation = true
}

```

On va ainsi récupérer la position de l'utilisateur afin d'afficher une carte centrée sur ses coordonnées.

3 Conclusion

Notre projet de création d'application mobile pour Android et iOS a donc abouti au développement de *Memories*, notre version d'une galerie photo, que nous continueront de mettre à jour pour que sa version iOS se rapproche un maximum de celle pour Android.

Références

- [1] <https://stackoverflow.com/questions/23024831/android-shared-preferences-example>
- [2] <https://developer.android.com>
- [3] <https://cloud.google.com/maps-platform/?hl=fr>
- [4] <https://stackoverflow.com>