

M1 - Rapport Développement Mobile

LARUE Sebastien et CLAIN Jonathan, M1 Informatique

05/11/2018

Abstract

Dans le cadre de l'UE obligatoire développement mobiles, il est demandé de concevoir un jeu de notre choix sur la plateforme Android et iOS. Dans ce cadre, nous avons travaillé en binôme pour réussir à coder au mieux le projet.

A cet effet, nous avons combiné notre temps, volonté et motivation, ainsi que nos capacités en informatique pour pouvoir réaliser au mieux ce projet qui nous tenait à cœur.

Nous avons dû surmonter et pallier avec nos erreurs et faiblesses pour concevoir et répondre aux cahier des charges ainsi que pour offrir aux utilisateurs le jeu le plus attractifs et distrayants possible.

A terme de quelques semaines de travail, nous avons donc réussi à mettre en place ce projet, qui répond au mieux aux demandes du cahier des charges.

Introduction

Le jeu Miaou Aventures, en accord avec les cahiers des charges proposés par le professeur de développement mobiles, a donc été réalisé sur Android et sur iOS. Nous avons fait en sorte que nos jeux soient semblables en tout point de vue.

Nous nous attellerons donc à vous présenter les différentes ficelles de sa conceptions que ce soit, d'abord, à travers la version Android puis, dans un second temps, celle d'iOS.

Afin de réaliser ce projet, différents sites et forums nous ont été utiles tels que: [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#) et [\[5\]](#)

1 Description

Miaou Adventures s'inspire du principe et des modalités traditionnels des 'runners', à savoir encourager l'utilisateur à toujours dépasser son meilleur score. L'objectif est simple: guider le chat Miaou et son vaisseau intersidérale vers des territoires inconnus et à travers une tempête d'astéroïdes mortelles. Il s'agit d'un jeu amusant et addictif, et surtout accessible à tous les âges. Le jeu s'arrête seulement lorsque le vaisseau de Miaou est détruit .

Les deux versions encouragent le joueur à poursuivre sa quête et ses aventures à travers l'hyperespace grâce à un thème futuriste et esthétique. Les nombreuses autres options comme la localisation et l'appareil photo rajoutent un plus non négligeable fondamental dans l'attractivité du jeu. Il s'agit de fonctionnalités novatrices dans les jeux de 'runners' et qui nous permettraient de nous démarquer.

2 Architecture générale

2.1 Android

Cette partie du rapport sera consacrée à présenter l'architecture générale de la version Android de notre projet.

La version Android a été réalisée sur Android Studio en langage Java par Jonathan Clain.

2.1.1 MainActivity.java

C'est dans cette classe que l'on met en place le menu de notre jeu. En effet, le menu se compose de plusieurs boutons dont le bouton 'Play', 'Score', 'Son' et 'Réinitialiser Score'. De plus cette classe coordonne les événements sur les boutons. L'utilisateur peut ainsi lancer le jeu, visualiser le score, activer ou désactiver le son et réinitialiser le score.

2.1.2 GameActivity.java

C'est l'une des classes les plus importantes. C'est là que l'on gère les activités de notre jeu tel que le lancement, la mise en pause et enfin l'arrêt du jeu. Premièrement, nous avons la méthode `onResume()` qui va tout simplement lancer le jeu. Ensuite nous avons la méthode `onPause()`, elle permet de mettre le jeu en pause. Aussi c'est dans cette classe que l'on initialise la vue du jeu "Gameview" que l'on abordera par la suite. Une bannière en haut de l'écran est aussi intégrée. Elle est composée de l'affichage du score, d'un bouton retour, pause et reset. L'utilisateur peut donc effectuer ces différentes actions lorsque le jeu est en exécution.

2.1.3 GameView.java

La classe `GameView` hérite donc de `Surfaceview` et intègre l'interface `SurfaceHolder.Callback`. En effet, nous avons besoin de créer une surface dédiée afin de dessiner. On a également besoin de plusieurs éléments pour dessiner: un `Canvas`, un `Bitmap` et un `Paint`. C'est dans cette classe que l'on va instancier les éléments de notre jeu, puis les afficher en utilisant les fonctions de dessin.

Afin d'afficher le gameView dans notre "scene.xml" il faut donc lier ce dernier en tant que vue.

```
<com.example.clain.miaou.GameView
    android:id="@+id/SurfaceView01"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
}
```

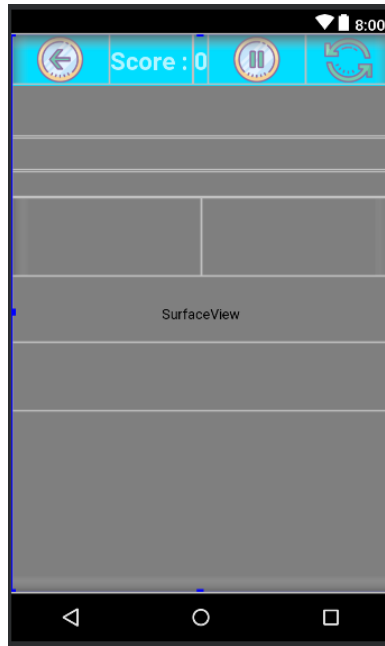


Figure 1 – GameView

2.1.4 Items.java

Dans cette classe, nous avons ajouté les différents sprites du jeu tels que la météorite, la pièce et le héros. Lors de l'instanciation, la position de ces derniers sont initialisées à l'aide du constructeur qui récupère en paramètres le contexte de l'application et les dimensions de l'écran. Une méthode Update permet le déplacement de ces sprites lorsque le jeu est en exécution. Enfin, des 'getters' permettent de récupérer les positions et les bitmaps et enfin des 'setters' afin de les modifier.

2.1.5 Hero.java

La classe hero permet d'ajouter le personnage principal dans le jeu. Tout d'abord, nous avons un constructeur. On récupère ainsi l'image du héros et on initialise sa position initiale. Une méthode moveHero permet de mettre à jour la position du héros dans le gameView lorsque le jeu est en exécution.

De plus, le déplacement du héros se fait en utilisant les capteurs du smartphone, l'accéléromètre. Ainsi, la classe implémente un `SensorEventListener`.

```
manager = (SensorManager) context.getSystemService
(Service.SENSOR_SERVICE);
    if (manager != null) {
        mAccelerometer = manager.getDefaultSensor
            (Sensor.TYPE_ACCELEROMETER);
    }
```

Deux méthodes permettent d'activer ou désactive le déplacement:

```
public void activerDeplacement(){
    accelSupported = manager.registerListener
        (this, mAccelerometer, SensorManager.SENSOR_DELAY_GAME);
}

public void desactiverDeplacement(){
    if(accelSupported){
        manager.unregisterListener(this,mAccelerometer);
    }
}
```

2.1.6 ScoreView.java

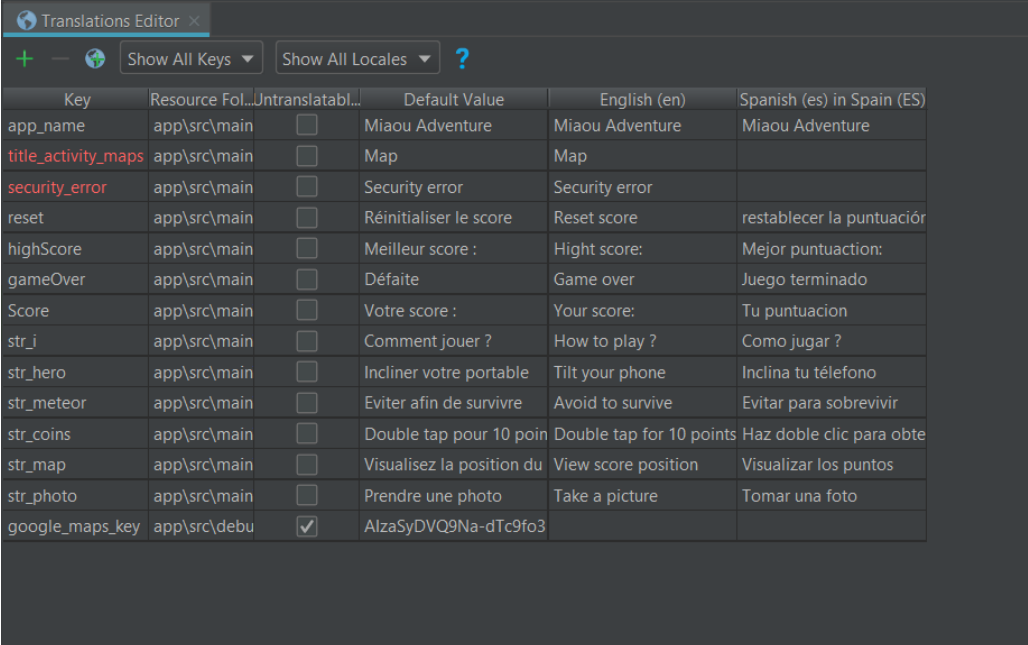
En démarrant le jeu pour la première fois, le score est à 0. A la fin de chaque de partie le meilleur score est sauvegardé. La sauvegarde s'effectue à travers le 'SharedPreferences'. Les SharedPreferences sont des espaces de stockage propres à chaque application Android.

2.1.7 MapsActivity.java

Afin de répondre aux demandes du cahier des charges, l'intégration de la géo-localisation était nécessaire. Le MapsActivity permet à l'application d'accéder à Google Maps et ainsi de déterminer la position actuelle du joueur.

2.1.8 Internationalisation

L'internationalisation sur Android est relativement simple:



The screenshot shows the 'Translations Editor' window in Android Studio. It displays a table of localized strings for the 'Miaou Adventure' app. The table has columns for 'Key', 'Resource Folder', 'Untranslatable', 'Default Value', 'English (en)', and 'Spanish (es) in Spain (ES)'. The 'google_maps_key' row is highlighted with a checkmark in the 'Untranslatable' column.

Key	Resource Fol...	Untranslatable...	Default Value	English (en)	Spanish (es) in Spain (ES)
app_name	app\src\main	<input type="checkbox"/>	Miaou Adventure	Miaou Adventure	Miaou Adventure
title_activity_maps	app\src\main	<input type="checkbox"/>	Map	Map	
security_error	app\src\main	<input type="checkbox"/>	Security error	Security error	
reset	app\src\main	<input type="checkbox"/>	Réinitialiser le score	Reset score	restablecer la puntuación
highScore	app\src\main	<input type="checkbox"/>	Meilleur score :	Hight score:	Mejor puntuación:
gameOver	app\src\main	<input type="checkbox"/>	Défaite	Game over	Juego terminado
Score	app\src\main	<input type="checkbox"/>	Votre score :	Your score:	Tu puntuación
str_i	app\src\main	<input type="checkbox"/>	Comment jouer ?	How to play ?	Como jugar ?
str_hero	app\src\main	<input type="checkbox"/>	Incliner votre portable	Tilt your phone	Inclina tu teléfono
str_meteor	app\src\main	<input type="checkbox"/>	Eviter afin de survivre	Avoid to survive	Evitar para sobrevivir
str_coins	app\src\main	<input type="checkbox"/>	Double tap pour 10 points	Double tap for 10 points	Haz doble clic para obtener
str_map	app\src\main	<input type="checkbox"/>	Visualisez la position du	View score position	Visualizar los puntos
str_photo	app\src\main	<input type="checkbox"/>	Prendre une photo	Take a picture	Tomar una foto
google_maps_key	app\src\main	<input checked="" type="checkbox"/>	AlzaSyDVQ9Na-dTc9fo3		

Figure 2 – Internationalisation

2.2 iOS

Cette partie-ci est destinée à la présentation de l'architecture générale de la version iOS de notre projet.

La version iOS a été réalisé sur Xcode en langage Swift avec SpriteKit par Sebastien Larue.

2.2.1 MainMenuScene.swift

C'est dans ce fichier que nous mettons en place le menu de notre jeu. Ce dernier se compose de divers boutons dont le bouton 'Play', 'Score' et 'Son'. C'est donc ce fichier qui coordonne les évènements sur les boutons, qui vont nous permettre de lancer le jeu, de visualiser le score et d'activer ou désactiver le son.

2.2.2 GameElements.swift

Ce fichier regroupe tous les éléments du jeu créés sous forme de fonctions. Ces fonctions sont ensuite appelées dans dans le GameplayScene.swift. Le code ci-dessous est une fonction qui permet de créer un bouton retour et de le positionner sur l'interface:

```
func createBackButton(){
    backButton = SKSpriteNode(imageNamed: "back")
    backButton.position = CGPoint(x: self.frame.width/12,
    y: self.frame.height * 0.95)
    backButton.zPosition = 4
    self.addChild(backButton)
}
```

2.2.3 GameplayScene.swift

C'est le fichier principale de notre projet. Il coordonne l'ensemble des actions de notre jeu. On retrouve dans un premier la déclaration et l'instanciation des variables du jeu. Le code fait appel à toutes nos fonctions du fichier GameElements.swift. On retrouve aussi d'autres codes qui gèrent chaque évènement sur les boutons, la collision entre les sprites et le geste courant.

Ci-dessous on retrouve le code qui permet d'ajouter un geste courant sur le 'view' du jeu, le double tap:

```
let tap = UITapGestureRecognizer(target: self,
    action: #selector(doubleTapped))
    tap.numberOfTapsRequired = 2
    tap.numberOfTouchesRequired = 1
    view.addGestureRecognizer(tap)
```

2.2.4 ScoreScene.swift

En démarrant le jeu pour la première fois, le score est à 0. A la fin de chaque de partie, le meilleur score est sauvegardé. La sauvegarde s'effectue à travers le 'UserDefaults'. C'est une classe qui permet de stocker des données de petite taille.

Dans un premier temps, on sauvegarde le score à la fin de la partie si il est supérieur à 0 ou du meilleur score précédent:

```
let defaults = UserDefaults.standard
let hscore = defaults.integer(forKey: "highestScore")
if hscore < score{
    defaults.set(score, forKey: "highestScore")
}
```

Dans un second temps, on récupère le score pour l'afficher dans l'interface de score:

```
let defaults = UserDefaults.standard
let highestScore = defaults.object(forKey: "highestScore")
```

2.2.5 MapView.swift

Afin de répondre aux demandes du cahier des charges, l'intégration de la géo-localisation était nécessaire. Le MapView.swift permet à l'application d'accéder à l'Apple Maps. Dans un premier temps, il fallait d'abord faire appel à la classe MKMapView. Cette classe nous permet d'afficher la map sur l'écran.

Dans un second temps, il fallait déterminer la position actuelle du joueur. Pour cela, il fallait faire appel à la classe CLLocationManager.

On a ajouté une épingle sur la position actuelle de l'utilisateur avec en légende les coordonnées géographiques.

2.2.6 ViewController.swift

Le ViewController.swift va nous permettre d'accéder à l'appareil photo, de prendre une photo et de la sauvegarder. Pour cela, on utilise la classe UIImagePickerControllerController.

Voici le code qui va nous permettre de sauvegarder une photo dans la librairie:

```
@IBAction func onSaveButton(_ sender: Any) {
    guard let selectedImage = imageView.image else {
        print("Image not found!")
        return
    }
    UIImageWriteToSavedPhotosAlbum(selectedImage, self, #selector
    (image(_:didFinishSavingWithError:contextInfo:)), nil)
}
```

2.2.7 Localizable.strings

Afin de rendre le jeu disponible en français et en anglais, nous avons fait appel au fichier Localizable.strings. A ce dernier, on ajoute les données de traduction sous formes de 'Key values'.

Vous pouvez voir ci-dessous un exemple de l'internationalisation dans notre application.

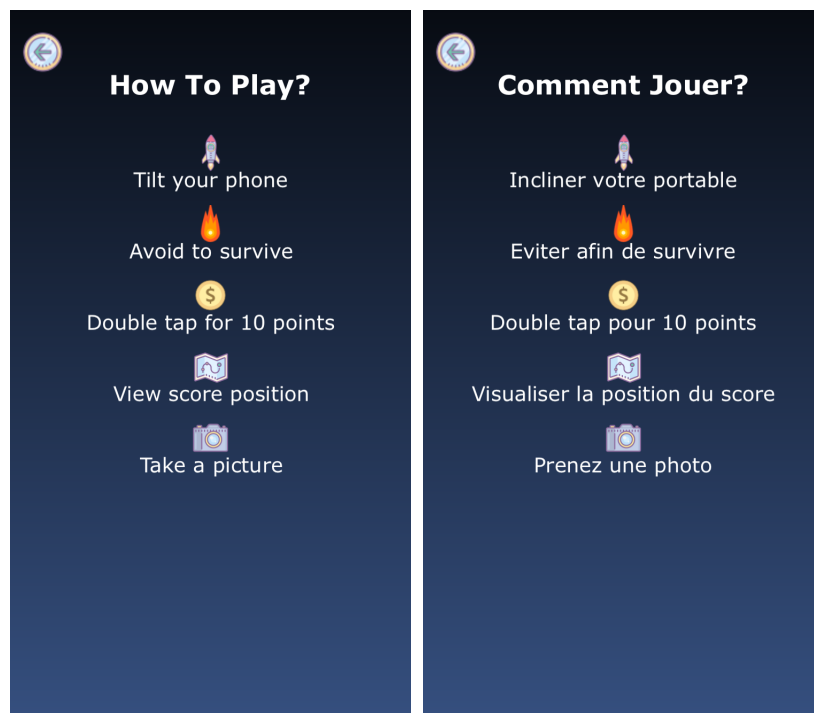


Figure 3 – Gauche: English et Droite: Français

3 Interfaces: Android et iOS

Miaou Adventures se compose des interfaces suivantes:

- Un menu de jeu
- L'interface de score
- L'interface de jeu
- L'interface de géo-localisation
- L'appareil photo
- L'interface de jeu

3.0.1 Le menu de jeu

Le menu nous offre un aperçu du thème du jeu. Il vous propose différentes fonctionnalités: démarrer une nouvelle partie, de nous montrer le meilleur score, de le réinitialiser, nous donner des indications sur le fonctionnement du jeu ou le mettre en mode sourdine. On peut également constater une bannière publicitaire.

3.0.2 L'interface de score

A la fin de chaque partie, il existe deux possibilités: soit le meilleur score qui n'a pas été dépassé au cours de la dernière partie s'affiche, soit c'est le nouveau record atteint qui s'affiche.

Voir ci-dessous les images correspondantes à l'interface menu et score du jeu:

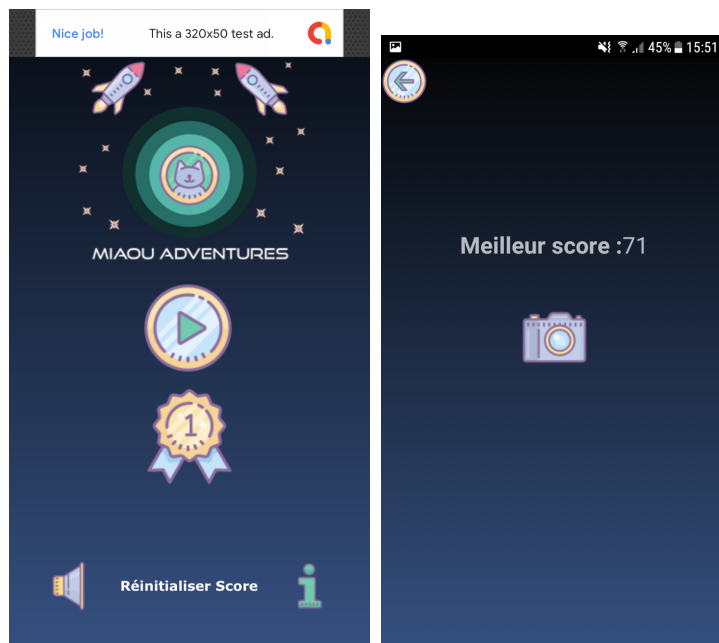


Figure 4 – Gauche: Menu et Droite: Score

3.0.3 L'interface de géo-localisation

A la fin de chaque partie, le joueur aura la possibilité de visualiser sa position géographique.

3.0.4 L'appareil photo

Si le joueur le souhaite, nous avons intégré l'appareil photo pour qu'il puisse prendre un petit selfie pour se rappeler de sa victoire.

Voir ci-dessous les images correspondantes:

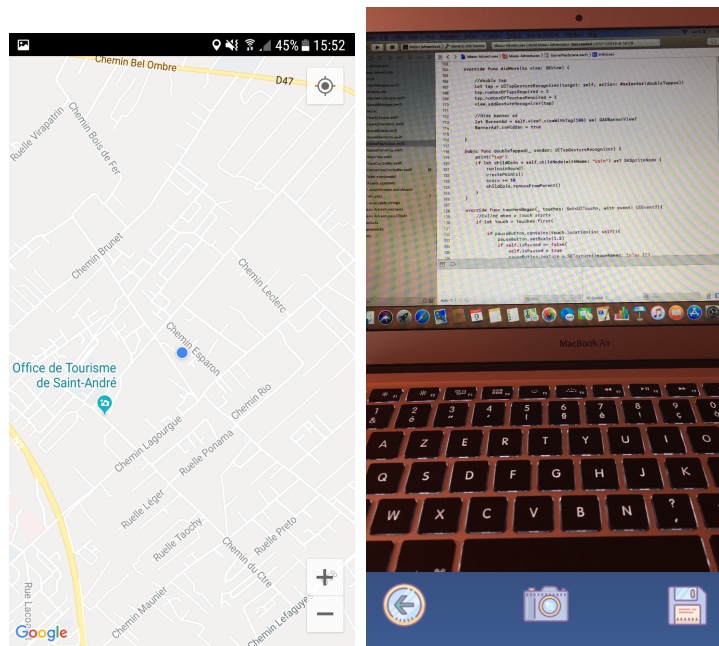


Figure 5 – Gauche: Géo-localisation et Droite: Appareil photo

3.0.5 L'interface de jeu

Sur l'interface de jeu, on retrouve tous les éléments de jeu. C'est la scène principale où le joueur prend contrôle de son vaisseau. Le but est simple, aller le plus loin possible en évitant les obstacles et récupérer les points bonus. L'utilisateur a la possibilité de quitter la partie en cours, de faire pause ou de redémarrer une partie.

3.0.6 Défaite du joueur

Lorsque le joueur perd, le Game Over apparaît. Sur cette partie, le jeu affiche votre score. On retrouve également les boutons pour quitter le jeu, redémarrer une partie ou afficher sa position actuelle.

Voir ci-dessous les images correspondantes:

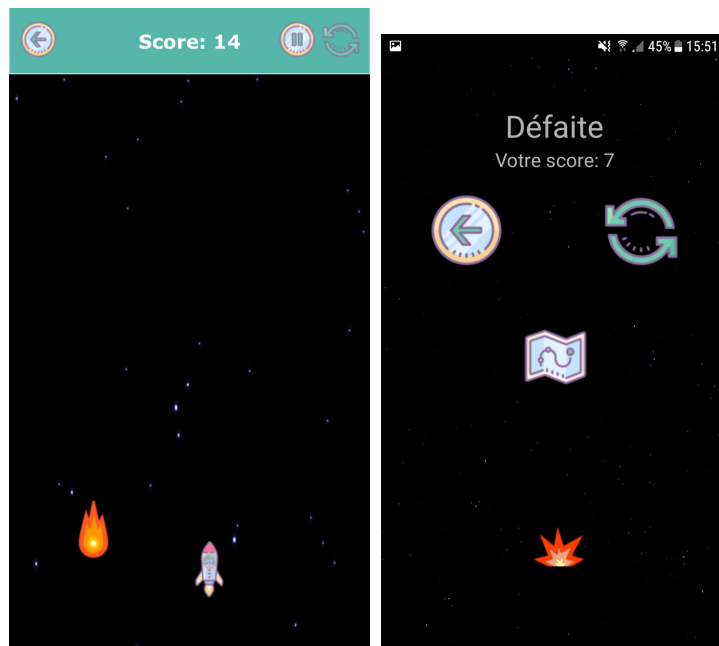


Figure 6 – Gauche: Jeu et Droite: Fin de partie

Conclusion

En conclusion, ce projet fut une belle épreuve qui nous a permis d'acquérir d'avantage d'expérience dans le développement d'applications pour mobiles.

L'année de L3 nous a permis d'emmagasiner de nombreuses connaissances. Cette année de Master, par contre, nous a permis d'améliorer nos compétences à travers la pratique. Notre savoir a été mis en application en situation réelle. Sur le plan personnel, il a été très fructueux de travailler en équipe. Nous avons pu créer une véritable cohésion et mettre en place un dialogue. Ce projet nous a permis de confronter nos idées, développer notre esprit d'équipe, d'améliorer nos capacités rédactionnelles et de mettre de côté nos différents et idées antagonistes afin de pouvoir créer le meilleur jeu possible.

Petit remerciement aux sites: Icons8 [\[6\]](#) et Zapsplat [\[7\]](#) pour nous avoir fourni les sprites et les sons.

References

- [1] Androidstudio. <https://developer.android.com/docs/>.
- [2] Apple. <https://developer.apple.com/documentation/>.
- [3] Openclassroom. <https://openclassrooms.com>.
- [4] Raywenderlich. <https://www.raywenderlich.com>.
- [5] Youtube. <https://www.youtube.com>.
- [6] Icons8. <https://icons8.com>.
- [7] Zapsplat. <https://www.zapsplat.com>.