

# PROJET ANDROID

## Space Dragons

Eloi LE RENARD, Christophe NOEL

Novembre 2017

### Abstract

Cette article décrit la mise en oeuvre ainsi que le fonctionnement d'une application Android développer à l'aide d'Android Studio. Le but du jeu est de faire parcourir le plus longtemps dans l'espace un petit dragon nommé Jean-Marc. Le score obtenu peut être sauvegarder et afficher sur une carte Google Maps.

Mots clés : Développement pour mobile, Android Studio, Programmation orientée objet, Jeu video, Google Maps

## 1 Introduction

Pour l'unité d'enseignement en développement mobile, nous devons réaliser une application vidéo-ludique pour Android soumise à quelques contraintes techniques. Le jeu étant à définir selon nos propre critère, il fallait récupérer le score obtenu afin de pouvoir affiché une carte Google Maps, centrée sur la position géographique où celui-ci a été obtenu. De plus, un marqueur positionné sur ce dernier informe l'utilisateur de la personne ayant réaliser ce score.

## 2 Préliminaire

Cette partie regroupe toutes les images qui ont été nécessaires à la réalisation de l'application mobile. Seul 5 images sur les 10 concernent directement le jeu, ce sont les images dites de type "Sprite". Les 5 autres images ne servent qu'à l'amélioration esthétique du jeu.



Figure 1: Icône



Figure 2: Background



Figure 3: Bouton Start

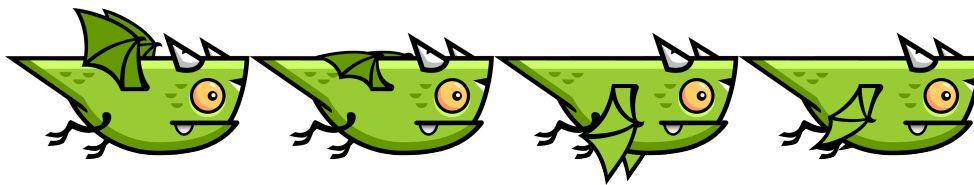


Figure 4: Sprite Jean-Marc



Figure 5: Sprite vie



Figure 6: Sprite ennemi lent



Figure 7: Sprite ennemi rapide



Figure 8: Sprite ennemi troll

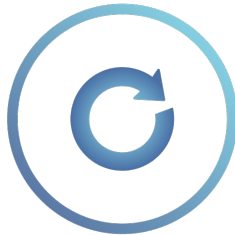


Figure 9: Bouton Replay



Figure 10: Bouton Google Maps

### 3 Architecture de l'application & Screenshots

Dans cette partie, nous allons voir en détails, les différents écrans de l'application.

#### 3.1 L'écran principal



Figure 11: Menu

Lors du lancement de l'application, l'utilisateur rencontre l'écran suivant. Il peut alors cliquer sur "START" pour commencer a jouer.

#### 3.2 Le jeu

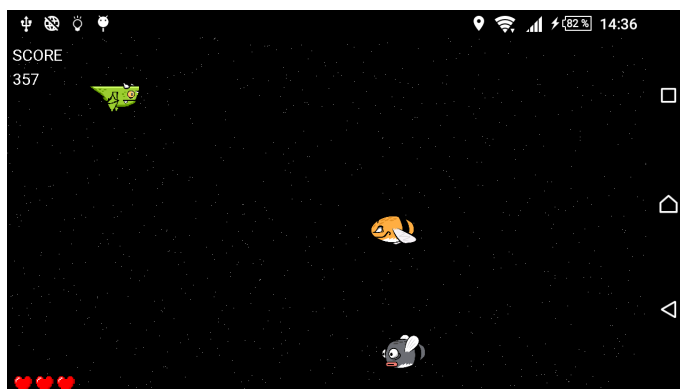


Figure 12: Jeu debut

Le jeu consiste a éviter des ennemis venant de la droite en orientant son téléphone afin de déplacer notre héros : Jean-Marc. Le joueur dispose de 3 vies et plus l'utilisateur survivra longtemps, plus son score augmentera.

A la mort, deux boutons sont affichés donnant le choix entre:

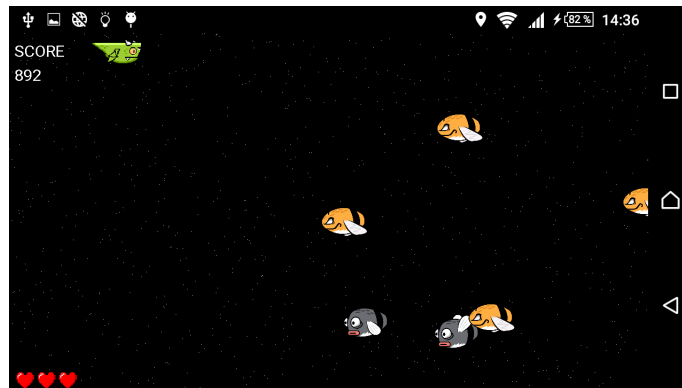


Figure 13: Jeu plus tard

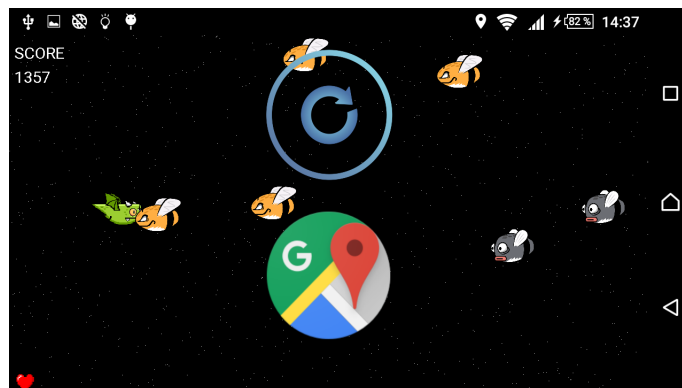


Figure 14: Game over

- Recommencer la partie (1er bouton)
- Sauvegarder son score (2ème bouton)

### 3.3 Entrer son score

En cliquant sur le bouton Google Maps, l'utilisateur est redirigé vers une page où sont affichés le score réalisé, la position actuel ainsi qu'une zone de saisie permettant de rentrer son nom. 2 choix s'offre ensuite:

- Le bouton "DELETE ALL SCORES" supprime tous les scores enregistrés auparavant et laisse toujours la possibilité de rentrer le sien
- Le bouton "SAVE YOUR SCORE" enregistre le score effectué (seulement si un nom est rentré) et redirige vers une carte Google Maps.

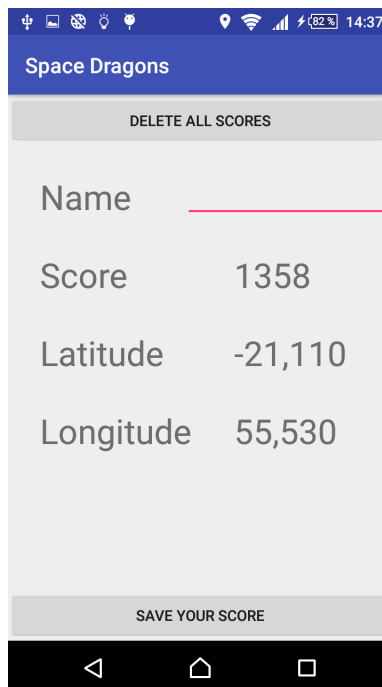
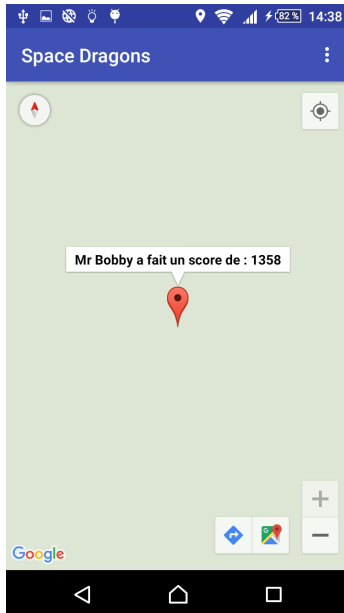


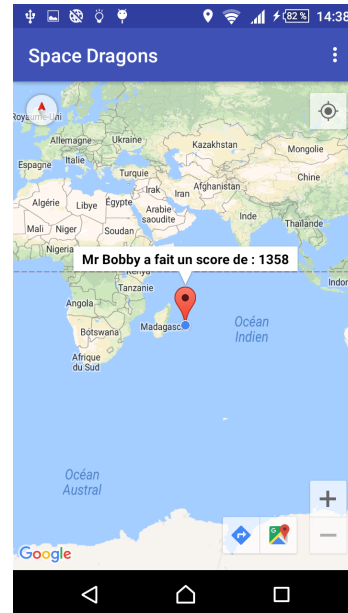
Figure 15: Entrer son score

### 3.4 Visualisation des score

Enregistrer son score redirige vers une activité Google Maps où l'on peut visualiser tous les scores sauvegardés par des marqueurs rouges. Enfin, le joueur peut aussi cliquer sur les 3 petits points en haut à droite pour visualiser les "High Scores".



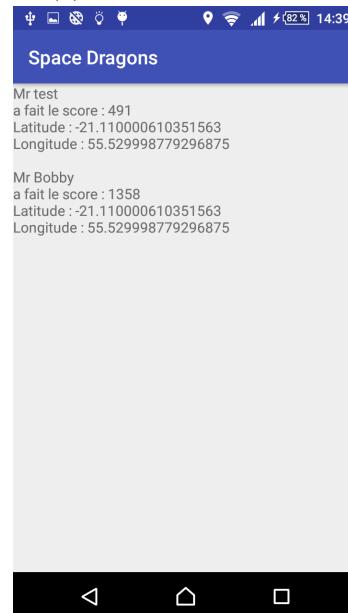
(a) Localisation zoomé



(b) Localisation dézoomé



(c) Localisation high scores



(d) High Scores

## 4 Points délicats - intéressants

### 4.1 Difficulté du jeu

La difficulté du jeu à du être adapter pour le passage oral , un compromis a donc était fait pour que la démonstration du jeu ne prenne pas trop de temps le jour venu (3 minutes max.). Notamment la présence de certains ennemis a été augmentée afin de pouvoir noter leur présence lors du test.

### 4.2 Curseur Google Maps

Un problème particulier à fait son apparition vers la fin de la conception de l'application. Les marqueurs de Google Maps peuvent se chevaucher et donc cacher certains autre scores. Pour palier à ce problème, nous avons mis en place un certain décalage entre les marqueurs (concrètement, chaque marqueur se décale légèrement vers le Nord). Cela rend donc visible les scores effectués au même endroit.

## 5 Code

Code s'occupant de la création des ennemis

```
private void createEnemies() {  
    enemyTroll = new EnemyTroll(getContext());  
    enemiesSlow = new ArrayList<>();  
    enemiesFast = new ArrayList<>();  
  
    for (int i = 0; i <= 10; i++) {  
        enemiesSlow.add(new EnemySlow(getContext()));  
        enemiesFast.add(new EnemyFast(getContext()));  
    }  
}
```

Figure 17: Méthode createEnemies()

Code gérant la mise à jour des ennemis "slow"



```

private void spawnSlow() {
    for (int i = 0; i < indexSlow; i++) {
        enemiesSlow.get(i).getCurrentFrame();
        canvas.drawBitmap(enemiesSlow.get(i).getBitmap(), enemiesSlow.get(i).getFrameSlow(), enemiesSlow.get(i).getPositionSlow(), paint);
    }

    int genSlow = new Random().nextInt( 300);
    if (genSlow == 3) {
        if (indexSlow < 10) {
            indexSlow++;
        }
    }
}
}

```

Figure 18: Méthode spawnSlow()

## 6 Conclusion

### 6.1 Amélioration à apporter

Voici une liste non exhaustive des améliorations pouvant être rajoutées:

- Une plus grande diversité d'ennemis
- Trois niveau de jeu : Easy - Normal - Hard
- Des items bonus : caisse de vie, invincibilité...
- Jean-Marc ayant la faculté de lancer des boules de feu

### 6.2 Le mot de la fin

Ce jeu met en pratique le déplacement de sprites animés dans une scène de jeu prédéfini. Il utilise l'accéléromètre, mais également l'outil de localisation du téléphone. Un système de persistance des données a été mis en place au travers d'une sauvegarde via l'utilisation d'un fichier XML. C'est un jeu basique mais techniquement riche au vue du nombre d'heures qu'a demandé son implémentation. Par conséquent, ce projet enrichissant, nous a concrètement permis de faire une première bonne expérience dans le monde du "Gaming".