

## Projet Conception d'Application Mobile

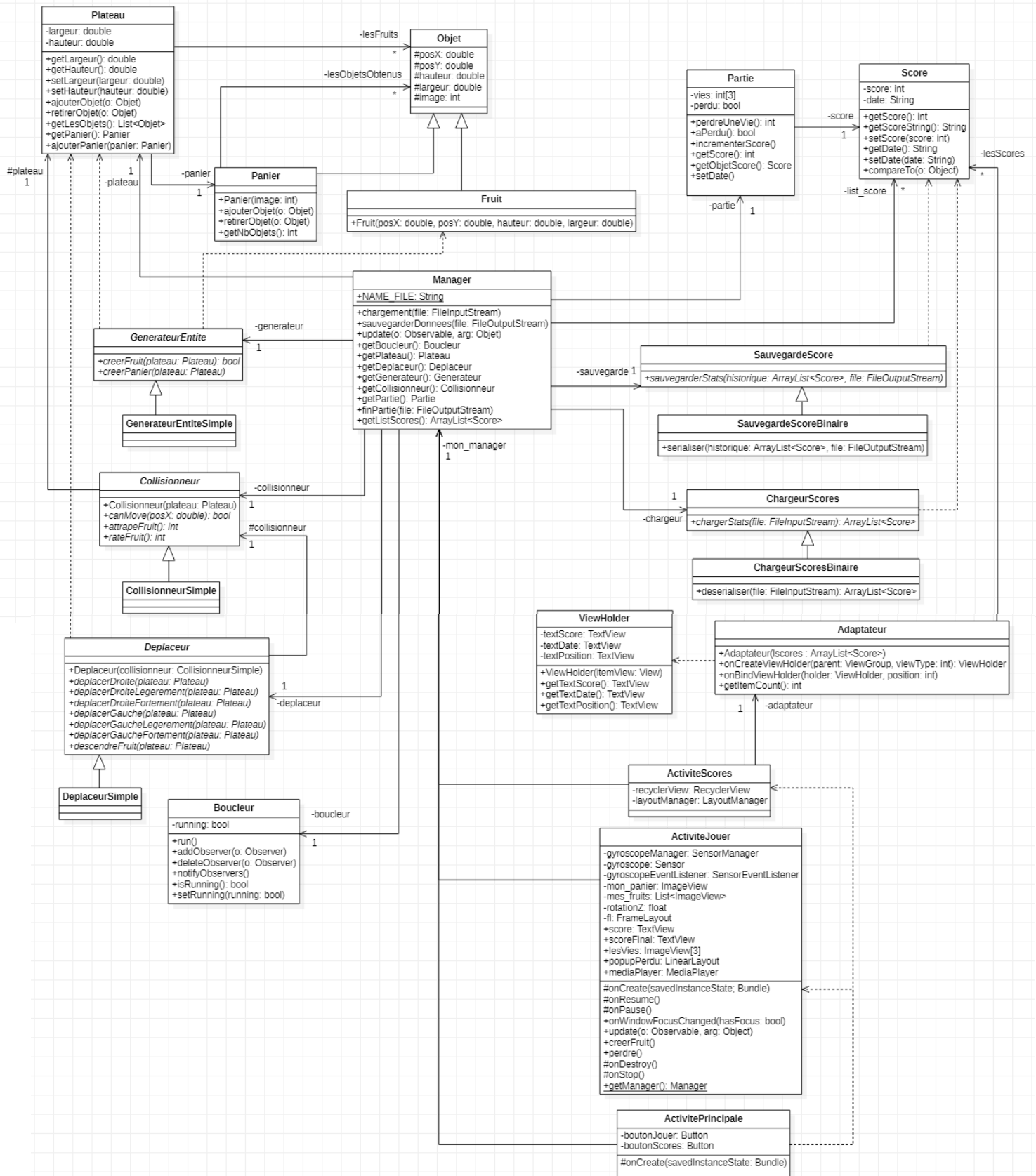
### DOCUMENTATION

#### CONTEXTE

Les vacances avaient si bien commencé. Île paradisiaque, eau turquoise et sable blanc, vous passiez vos journées à siroter des piña coladas sur un transat en bord de mer. Vous aviez même pensé à faire le plein de crème solaire avant de partir pour ne pas laisser de vilains coups de soleil gâcher ce séjour de rêve ! Tout était parfait. Enfin ça, c'était avant qu'une énorme vague de nous emporte et vous dépose sur un petit îlot abandonné, loin de toute civilisation.

Vous voilà donc livré à vous-même, sans nourriture ni piña colada. Il va vous falloir trouver de quoi manger, et vite. Heureusement, vous avez repéré un arbre qui semble bien prospère. Seul point négatif : il se trouve en bord de falaise et tous les fruits qu'il produit tombent directement dans l'eau. Avec quelques lianes et des feuilles de palmes, vous parvenez à fabriquer un panier afin de récolter les fruits. Mais attention, si vous n'en ramassez pas assez, votre survie n'est pas assurée... Parviendrez-vous à tenir jusqu'à l'arrivée des secours ?

Notre jeu **Tropic Panic** consiste à ramasser des fruits qui défilent sur l'écran à l'aide d'un panier. Le panier peut être déplacé en penchant le téléphone grâce au gyroscope. Chaque fruit attrapé représente un point et chaque fruit loupé fait perdre une vie. Le joueur dispose de trois vies, la partie se termine donc lorsque trois fruits ont été manqués. Les scores associés à leur date sont sauvegardés et sont consultables sur la page « meilleurs scores » accessibles depuis l'écran d'accueil. Ils sont classés par ordre décroissant, c'est-à-dire les meilleurs scores en premier.



Ce diagramme de classe montre bien que nous avons créé de nombreux **points d'extensions** puisque toutes les classe « utilitaires » sont des classes abstraites spécifiées par la suite. Cela permettrait donc de pouvoir choisir entre plusieurs manières de générer un fruit ou de sauvegarder les données.

Nous avons également porté attention à la **séparation du modèle et des vues** puisque les vues n'ont accès qu'au manager qui a accès au reste du modèle. Il joue donc le rôle de porte d'entrée sur le reste du modèle.

La classe **Objet** : classe abstraite qui représente chaque objet du jeu qui sera visible à l'écran. Dans notre cas, il s'agit du panier et des fruits. L'abstraction de cette classe permettrait de facilement rajouter de nouveaux éléments. Ainsi, le plateau par exemple n'a pas une liste de fruits mais une liste d'objets. Cette liste pourrait contenir d'autres objets sans empêcher le fonctionnement du jeu.

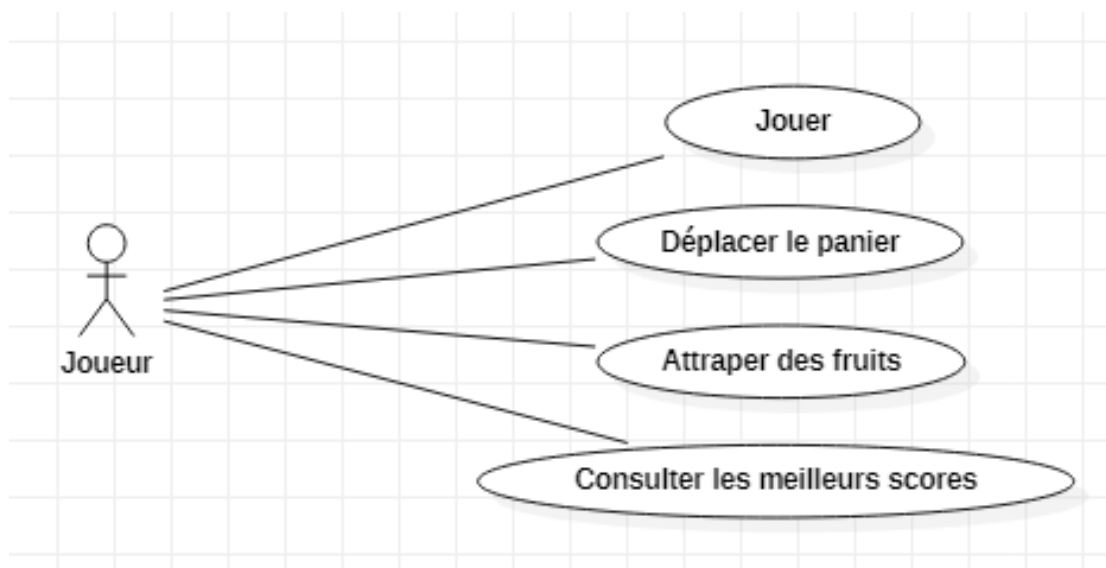
La classe **Plateau** : représente l'écran de jeu, notamment en donnant accès au panier et aux fruits et ainsi pouvoir changer leur position par exemple.

La classe **Partie** : contient un objet Score ainsi qu'un booléen indiquant si la partie est terminée ou non. Permet de modifier l'objet Score.

La classe **Score** : est composée d'un score (int) et d'une date (string). C'est cet objet qui sera sauvegardé en persistance profonde afin de garder en mémoire tous les scores qui ont été réalisés ainsi que la date de leur réalisation. C'est aussi l'objet qui sera affiché dans la RecyclerView de l'ActiviteScores.

Les classes **Adaptateur** et **ViewHolder** : classes nécessaires à la création de la RecyclerView. Le ViewHolder contient les différentes TextView qui afficheront les données (définies dans score\_item.xml) tandis que l'Adaptateur va injecter dans les TextView du ViewHolder les différentes données du Score. Chaque score s'affichera donc dans la RecyclerView comme défini dans le fichier score\_item.xml.

## DIAGRAMME DE CAS D'UTILISATION



Cas n°1 : Jouer

- Acteur : Joueur
- Condition d'utilisation : aucune
- Description : Un joueur peut jouer à notre jeu
- Scénario de succès :
  - Le joueur ouvre le jeu.
  - Le joueur clique sur « JOUER ».
  - Le système affiche l'écran de jeu.
  - Le joueur peut jouer.

Cas n°2 : Déplacer le panier

- Acteur : Joueur
- Condition d'utilisation : le joueur doit être en train de jouer.
- Description : Un joueur peut déplacer le panier en penchant son panier.
- Scénario de succès :
  - Le joueur joue au jeu.
  - Le joueur penche son téléphone.
  - Le système détecte le mouvement du téléphone et déplace le panier.

Cas n°3 : Attraper des fruits

- Acteur : Joueur
- Condition d'utilisation : le joueur doit être en train de jouer.
- Description : Un joueur peut attraper des objets grâce au panier.
- Scénario de succès :
  - Le joueur joue au jeu.
  - Le joueur déplace le panier.
  - Le système détecte qu'un fruit touche le panier.
  - Le système comptabilise un point et l'ajoute au score.
- Scénario d'échec :
  - Le joueur joue au jeu.
  - Le système détecte que le fruit touche le bas de l'écran sans avoir touché le panier.
  - Le joueur a loupé un fruit et perd une vie.

Cas n°4 : Consulter les scores

- Acteur : Joueur
- Condition d'utilisation : aucune
- Description : Un joueur peut consulter les scores.
- Scénario de succès :
  - Le joueur clique sur le bouton « MEILLEURS SCORES ».
  - Le système ouvre la vue des scores et affiche les scores par ordre décroissant.
  - Le joueur peut consulter tous les scores sauvegardés.