

## Projet Android Adrien BLAY A3, Hector PITEAU B2

### Contexte de l'application

Le but de notre projet est de créer un jeu. Dans ce jeu, nous incarnons un requin et ce requin doit survivre à la vague qui est derrière lui. Pour cela, il doit éviter les cailloux qui arrivent vers lui et prendre des bonus lorsque ceux-ci apparaissent. Pour éviter les cailloux, rien de plus simple, nous devons pencher le téléphone vers la droite ou la gauche pour déplacer le requin. Lorsque celui-ci touche un rocher, tout n'est pas fini, il sera juste ralenti par certains tandis qu'il se fera bloquer par d'autres. Les obstacles apparaîtront aléatoirement. Le but de ce jeu est de faire le plus de distance possible et donc le plus haut score possible.

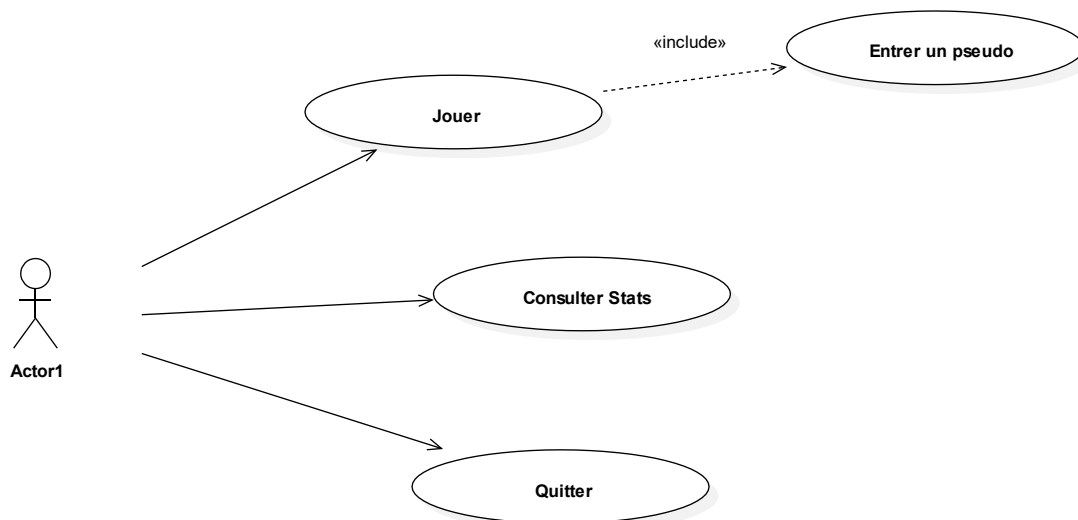
Lorsque nous avons perdu, un écran de Game Over s'affiche, sur celui-ci, on retrouve notre score de la partie, notre score total, ainsi qu'un bouton pour recommencer et un bouton pour aller au menu principal.

Dans le menu, nous retrouvons :

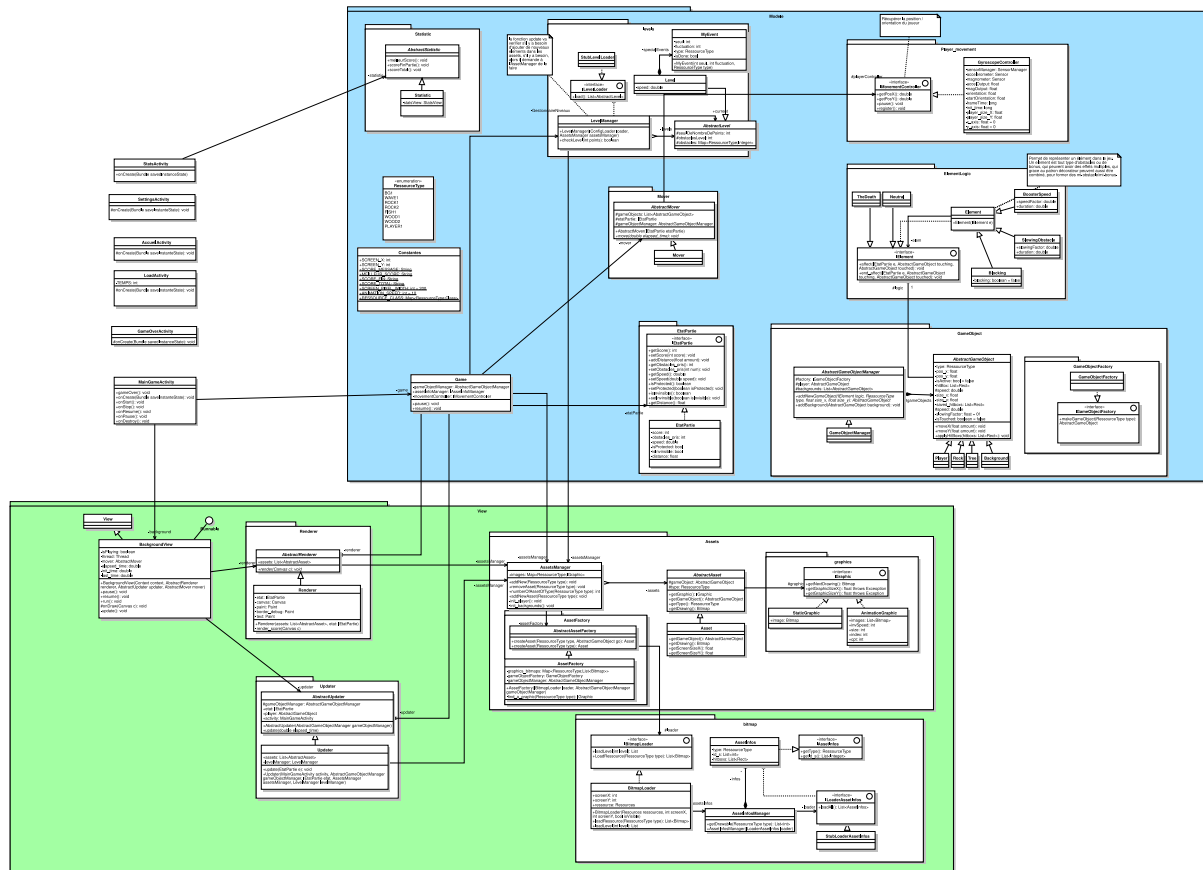
- Un bouton Jouer où l'on devra cliquer pour lancer le jeu après avoir rentré son pseudo.
- Une page de statistique qui affiche le meilleur score, le score de la dernière partie ainsi que les scores additionnés depuis que vous jouez.
- Une page paramètre où nous pouvons enlever la musique ou en mettre lorsque celle-ci est enlevée.
- Et enfin un bouton pour quitter l'application.

### Diagramme de cas d'utilisation

Dans le diagramme de cas d'utilisation, nous avons un utilisateur. Cet utilisateur va avoir trois possibilités. Il va pouvoir consulter les statistiques en cliquant sur le bouton Stats, ensuite il pourra quitter l'application en cliquant sur le bouton Quitter. Pour Jouer, il devra obligatoirement rentrer un pseudo non vide puis cliquer sur le bouton Jouer.



## Diagramme de cas d'utilisation



Dans le diagramme de classes, nous avons plusieurs pôles importants : le modèle avec les GameObjects, les Levels, le Mover, et la Logic, et la Vue, avec les Assets, le Render et l'updater.

**Les GameObjects :** Ils permettent de représenter un objet en jeu, il contient seulement les informations métier de l'objet. On peut changer sa position, sa hitbox. Une hitbox est un ensemble de Rect. Cet ensemble représente la zone qu'il recouvre. On peut ainsi calculer si deux GameObjects sont l'un sur l'autre ou non.

**Les Levels :** Ils permettent de changer la configuration du jeu en fonction de l'avancé du joueur dans celui-ci. Grâce au stub, nous pouvons déterminer quels seront les obstacles et bonus en jeu, et ce à tous moments de la partie.

**Le Mover :** Il permet de déplacer les objets dans l'espace. Il ne dépend pas d'une vue, il permet simplement de faire avancer les positions de chaque GameObject dans le jeu.

**La logic :** Elle permet d'assigner à un GameObject un comportement lorsqu'il rentre en contact avec un autre GameObject. Le plus courant est l'interaction entre le joueur et un Obstacle. Il peut par exemple subir un ralentissement, voir même un blocage.

Les Assets : Un Asset représente un élément affichable dans le jeu. Il possède donc à la fois un GameObject, qui représente le fonctionnement de sa logique interne, et un IGraphic qui représente sa façon de s'afficher à l'écran.

Le Renderer : Il permet de dessiner à l'écran tous les Assets visibles. Il utilise un canvas dans lequel il dessine simplement chaque asset étiqueté comme « actif ».

L'Updater : Il permet de mettre à jour les status des Assets dans la scène en fonction de leur position. Il va passer tous les éléments invisibles à l'écran en « inactif ». Il va aussi vérifier les collisions. Entre le joueur et les Autres Assets. Il va aussi augmenter le score en fonction de l'avancée du joueur.