

# Rapport Flappy Bird pour le cours de Programmation avancée

par Lucas Lacoste et Julie Chartier en L1 SdN

## I/ Principe général

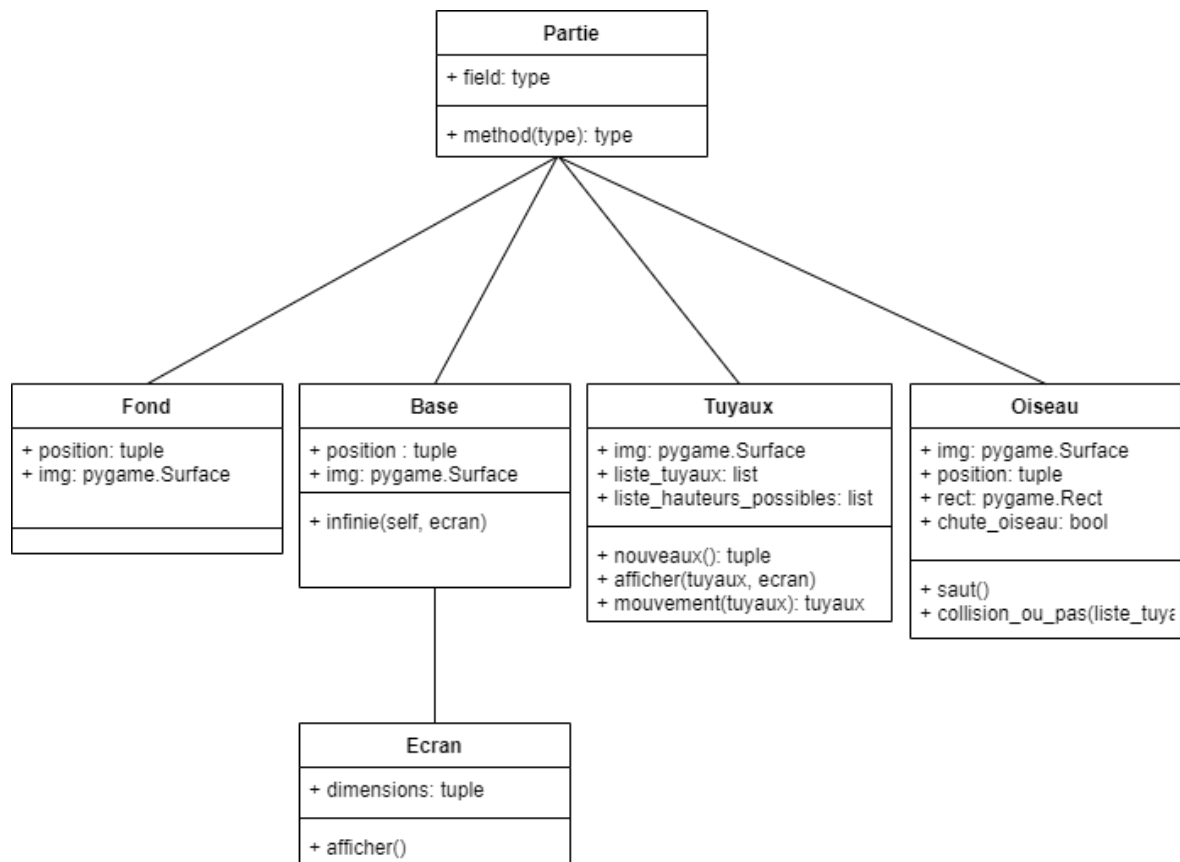
Notre Flappy Bird fonctionne de la manière suivante.

Lorsque nous lançons le programme, il y a l'écran de bienvenue qui s'affiche, il suffit d'appuyer sur la flèche du haut du clavier pour lancer la partie. L'oiseau chute de manière uniforme et accélérée sauf si l'on appuie sur la flèche du haut du clavier pour lui faire faire un saut. Quand l'oiseau saute, la vitesse revient à 0. Lorsque la coordonnée y de la position de l'oiseau n'est pas comprise entre 0 et 400 (la zone dans laquelle il peut voler), le joueur perd. Il perd aussi lorsque l'oiseau entre en collision avec un obstacle. Lorsque le joueur perd, le jeu affiche game over et le score auquel il s'est arrêté. Il suffit de ré-appuyer sur la flèche du haut pour relancer une partie.

## II/ Diagramme de classes

(aussi présent en .drawio dans le dossier rendu)

Notre programme n'est pas vraiment abouti en ce qui concerne les classes, voici un diagramme représentant notre travail :



### **III/ Retour sur expérience**

Premièrement, la difficulté sur laquelle nous avons le plus bloqué était celle de faire fonctionner le Flappy Bird avec des classes, chose qui nous paraissait plus facile à faire sans les objets.

Deuxièmement le saut de l'oiseau n'est pas parfait puisqu'il se téléporte et qu'il ne bouge plus si on reste appuyé sur la flèche du haut (ce qui nous paraît étrange car il nous semblait l'avoir résolu durant le dernier cours), cela n'a pas l'air très naturel mais nous l'aurions corrigé si nous avions disposé de plus de temps. Nous avons préféré nous concentrer sur notre problème concernant les classes et sur nos autres projets à rendre cette même semaine.

Ensuite, concernant le score, il n'est pas généré de la bonne manière, il s'incrémente en fonction du temps et non pas en fonction de collisions. De plus, si on lance plusieurs parties à la suite, il se décale et ne s'incrémente plus au bon moment.

Il était intéressant de travailler en binôme sur ce projet car ce n'est pas dans nos habitudes. Nous avons pu apprendre à séparer les tâches du projet : les bases du code, mettre les images, faire défiler la base à l'infini, chute uniformément accélérée de l'oiseau et saut, génération aléatoire et mouvement des obstacles, collisions, score.

Cependant, nous aurions dû commencer par essayer de réaliser le diagramme UML au début car c'est plus difficile de transformer le programme sans les classes une fois qu'il est fonctionnel. De ce fait, nous avons manqué de temps pour finir le travail et produire quelque chose de plus propre.

### **IV/ Présentation d'éléments supplémentaires**

Nous n'avons pas implémenté d'éléments supplémentaires à par une page de démarrage "bienvenue" et une image "game over" quand il y a une collision.