

Contexte

Jeu type Shoot'em up. Le joueur pourra choisir son avion et le but sera de survivre a des vagues d'ennemi qui spawn à l'écran de façon aléatoire. Les ennemis pourront tirer sur le joueur. Le meilleur score sera enregistré à chaque fois. Le joueur pourra choisir un avion entre plusieurs sélection d'avion ayant chacun différent stat.

Patrons utilisés

Patron Médiateur (GameManager.class) : Dans notre cas, le patron médiateur a servis à établir une classe qui s'occupe des règles au sein de notre jeu et des différentes entités. Par exemple la gestion des collisions ou les déplacements du joueur.

Explication UML

Notre UML présente plusieurs package: launch, vue, model, stub et pers.

Le package launch contient la classe Launch qui est la méthode main du projet, qui est le point de départ.

Ensuite notre package vue contient tout les controller de nos vue fxml, de gérer différentes actions qui sont effectué au sein de notre vue. Dans ce package les deux classes les plus importantes sont Fenetre, FenetreMenuJeu et FenetreJeu. La Fenetre est une classe abstraite regroupant la propriété Joueur, et les méthodes permettant la navigation d'une fenêtre à une autre. La classe FenetreMenuJeu et en quelque sorte la classe qui peut mener à toutes les autres fenêtre. Ensuite la classe FenetreJeu est la classe qui contient la boucle de jeu.

La package pers est important pour la mise en place de la persistance au sein de l'application car il contient les deux classes permettant le chargement et la sauvegarde de nos modèles

Ensuite notre package stub contient les fabrique. Nous avons la fabrique d'avion permettant la génération d'avion par exemple, qui sont important pour la fenêtre shop pour afficher des avions disponibles de base à l'achat.

Notre package model contient toutes les classes métier du projet, c'est-à-dire les classes qui représente les éléments de notre application. Nous avons par exemple dans ce package des classes tel que GameManager qui permet la gestion des classe entre eux pour permettre la mise en place des règles du jeu. Nous avons également la classe abstraite Entité qui est très importante dans notre projet pour représenter une entité au sein du jeu. Il y a également les interfaces Sauvegarde et Chargement qui permette, lorsqu'il sont intégrer, de réaliser une classe prenant en compte la persistance du modèle.

Certains éléments du package modèle tels que les interfaces de persistance sont intégrer dans les éléments du package pers, XMLSauvegarde et XMLChargement pour permettre la réalisation de classe de persistance du modèle dans des fichier.xml

Ensuite les fabrique d'avion sont par exemple appelé au sein de FenetreShop pour permettre de générer des avions de bases disponible à l'achat par le joueur.



