

Contexte :

Notre application est composée d'une page d'identification qui permet d'écrire son pseudo de jeu. Puis nous arrivons à la page d'accueil qui permet de dire « Bienvenue au joueur » en reprenant son pseudo. Elle permet ainsi d'aller au menu principal du jeu.

Nous arrivons ensuite sur le menu principal du jeu.

Ce menu est composé d'un bouton pour aller sur la page des listes des animaux. Il permet de sélectionner un animal qui fera office de personnage dans le jeu. De plus qu'un bouton qui permet de nourrir son animal. Celle-ci déclenche une augmentation de vie selon la nourriture sélectionnée.

Puis un bouton jeu qui permet d'arriver sur la page de tableaux des scores puis de lancer le jeu.

L'application démarre avec une fenêtre d'identification, avec deux labels qui annonce le titre (identification) et (le pseudo), puis un textfield qui permet d'entrer son pseudo de jeu. Si aucun pseudo n'est rentré un message d'erreur est déclenché.

Ensuite nous avons une page d'accueil avec le nom du jeu (ZOO Fx) et le nom du joueur (Pseudo). Puis un bouton qui permet d'aller au menu principal du jeu.

Le menu du jeu se compose de trois images de cœurs, en effet le jeu attaque avec un minimum de 3 vies . Ensuite sur la page il y a un label avec le nombre de 3 qui correspond aux trois vies. Au milieu de la page nous avons un panel de label qui indique les étapes à suivre pour commencer le jeu. En bas de la page nous avons aussi un panel de bouton. Ils permettent de sélectionner un personnage, sélectionné une nourriture, jouer et quitter le jeu.

Notre master-détail d'animaux a une liste des animaux sur la partie gauche de la fenêtre. Puis par un bind, quand on clic sur un animal le détail s'affiche sur la droite. Si on modifie un élément de description de droite, les modifications s'affichent automatiquement dans la liste et reste en sauvegarde jusqu'à la fermeture de l'application.

Le détail se comporte d'un label (Nom d'animaux), puis d'un textfield qui comporte le nom de l'animal, un label (Nourriture), qui correspond à la nourriture associée à l'animal, une image qui correspond à l'animal et un bouton image qui correspond au bruit de l'animal. En bas de la fenêtre, nous avons trois boutons, sélectionner son personnage pour le jeu, ajouter un personnage qui permet d'ajouter un personnage mais qui ne sert pas dans cette application car les détails ne sont pas conçus pour être rentré. Puis un bouton supprimé personnage, il permet de supprimer des personnages de la liste.

Notre master-détail de nourriture a une liste des nourritures sur la partie gauche de la fenêtre. Puis par un bind, quand on clic sur une nourriture, cela déclenche un détail sur la droite. Tout d'abord nous avons les points de vie selon l'ajout de la nourriture. Puis un label (animal associé), un label avec le nom de l'animal. Un label (nourriture), qui correspond à la nourriture associée à l'animal du dessus, puis un label qui correspond au nombre du point de vie que procure la nourriture. En bas de la fenêtre, il y a un bouton sélectionner nourriture qui permet de sélectionner le nombre de points de vie dans le jeu.

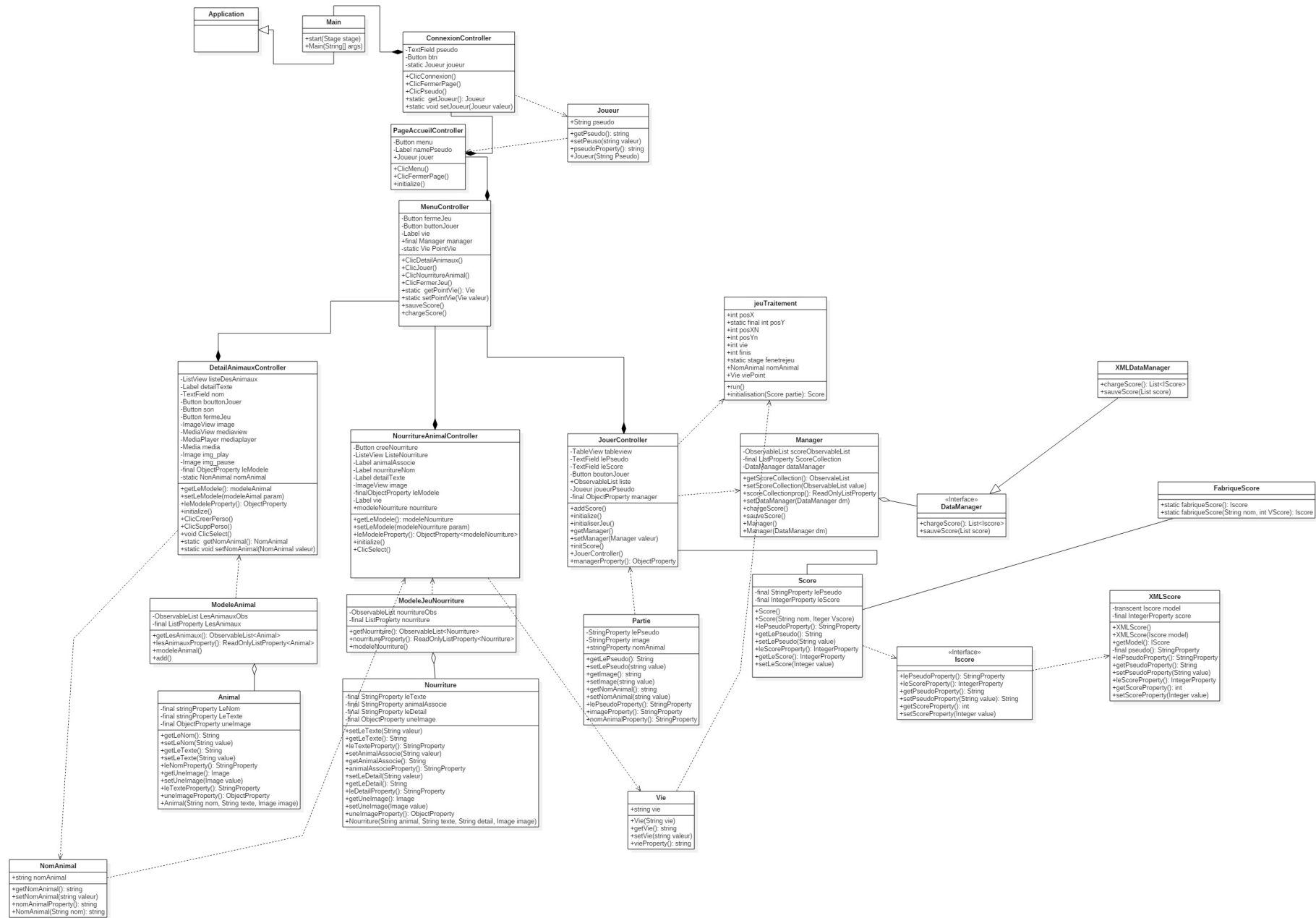
Puis nous avons le jeu, en cliquant sur le bouton nous arrivons sur une page des scores. En effet, cela correspond à un tableau de liste de score. Il est inscrit le pseudo du joueur et son score du jeu. En bas de la page il y a un bouton

jeu. Il déclenche le jeu.

But du jeu, le jeu est initialisé à 3 vies au départ (celui-ci peut changer en fonction de la nourriture donnée à l'animal). Nous utilisons le personnage sélectionné auparavant. Le personnage peut se déplacer avec les flèches du clavier de droite à gauche. Par la suite, un panel de fruit tombe de la page de haut en bas. Le but étant de les rattraper en se déplaçant. Si un fruit n'est pas attrapé le joueur perd une vie. Lorsque le joueur a plus de vie, le jeu se termine et son score est ajouté à la liste des scores. Quand le jeu est terminé un message est affiché, il donne le score du joueur et le fait qu'il a perdu. La page de jeu se ferme en appuyant sur ok. Sur la page du tableau de score nous avons un bouton rejouer, celui-ci permet de relancer une partie.

Pour finir, un bouton quitter jeu, qui permet de fermer l'application.

Diagramme de classe de notre application : si la qualité n'est pas assez bonne nous avons laissé la photo du diagramme et le dossier starUml dans le dossier javafxdoc



Cas d'utilisation de notre application : si la qualité n'est pas assez bonne nous avons laissé la photo du diagramme et le dossier starUml dans le dossier javafxdoc

