UNIVERSITÉ DE SHERBROOKE

Faculté de génie

Département de génie électrique et génie informatique

Rapport d’APP

Vérification logicielle

APP1

Présenté à :

Ruben Gonzalez-Rubio

Présenté par :

Vincent Bougie (bouv1906)

Ludovic Deprey (depl2401)

Abderrahman Meftahi (mefa2301)

Kelly Roussel (rouk3205)

Fatima Zahra Banani (banf3002)

Sherbrooke – 6 septembre 2019

# Table des matières

[Table des matières 1](#_Toc536147431)

[1. Introduction 2](#_Toc536147432)

[2. Récits utilisateurs 2](#_Toc536147433)

[3. Diagramme de classes 3](#_Toc536147434)

[4. Tests 3](#_Toc536147435)

[6. Conclusion 8](#_Toc536147437)

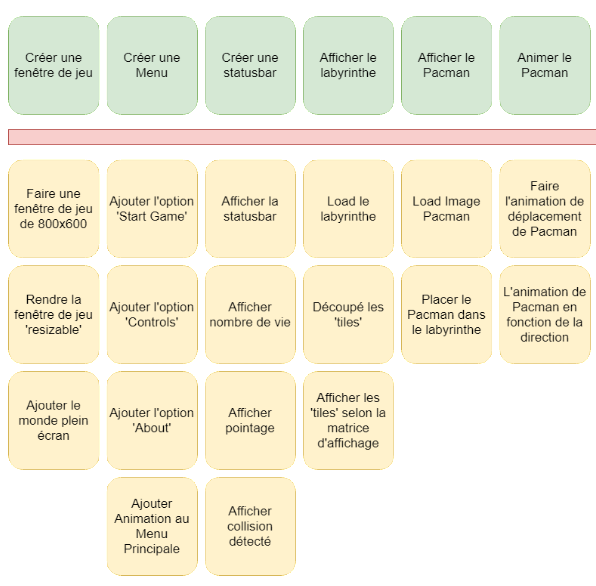
[7. Références 9](#_Toc536147438)

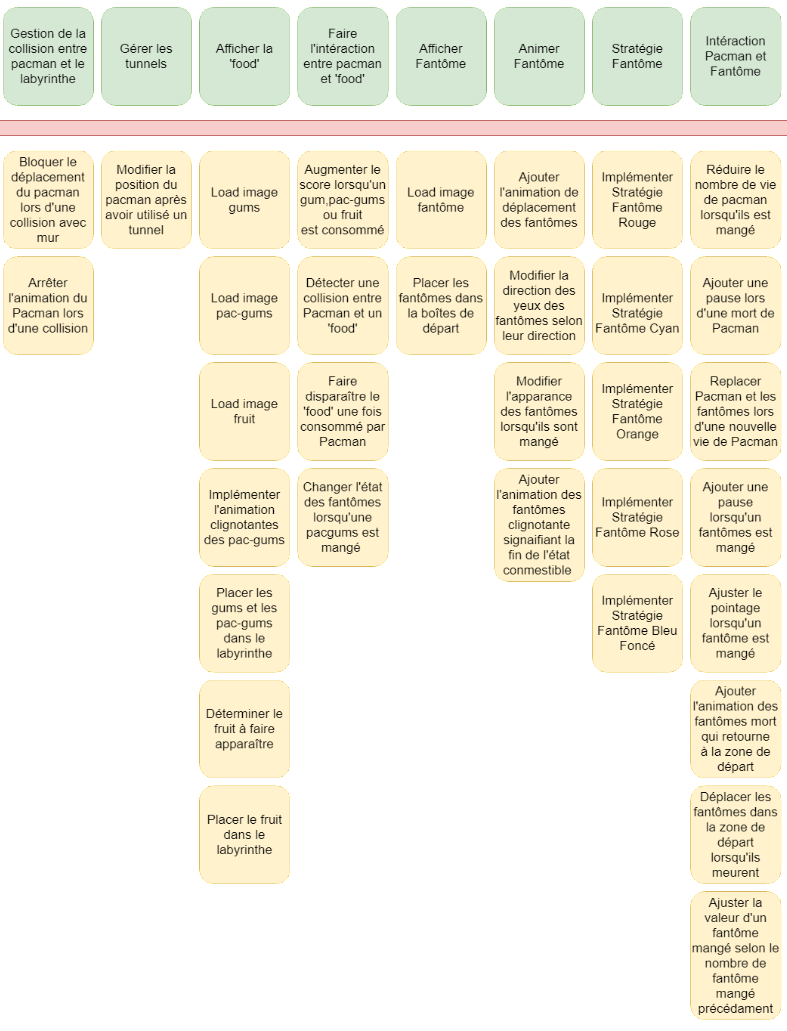
# Introduction

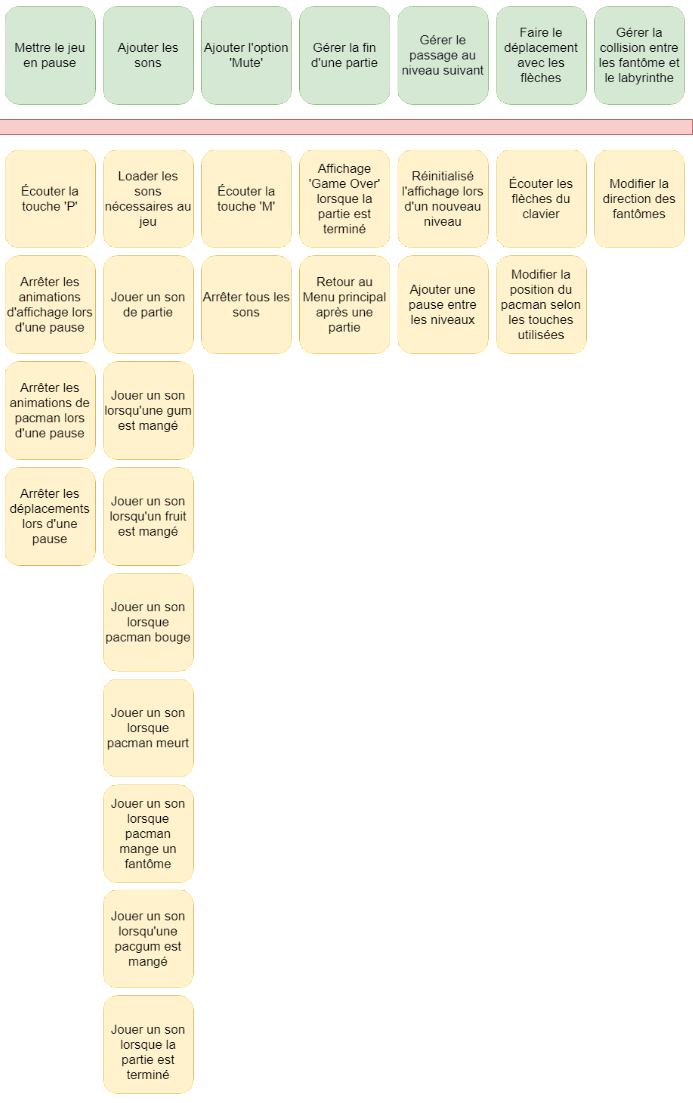
Le jeu PacMan, développé il y a 40 ans par  [Tōru Iwatani](https://fr.wikipedia.org/wiki/T%C5%8Dru_Iwatani" \o "Tōru Iwatani) pour l’entreprise japonaise [Namco](https://fr.wikipedia.org/wiki/Namco) au japon, est emblématique des jeux d’arcades et connu par de nombreuses générations. A l’occasion de son anniversaire, nous décidons d’en réaliser une version Java. Le présent rapport présente les débuts de notre projet. Il inclue la conception en amont selon la méthode Agile mais aussi le début du développement.

# Récits utilisateurs

Connaissant tous le jeu du PacMan, nous avons déjà une bonne vision globale du projet. Nous décidons d’entrer dans les détails en explicitant les différents récits utilisateurs. Nous partons de tâches « macros » en vert (épopées) que nous subdivisons en tâches unitaires en jaune (récits utilisateurs). Nous présentons ici la visualisation de ces récits utilisateurs.





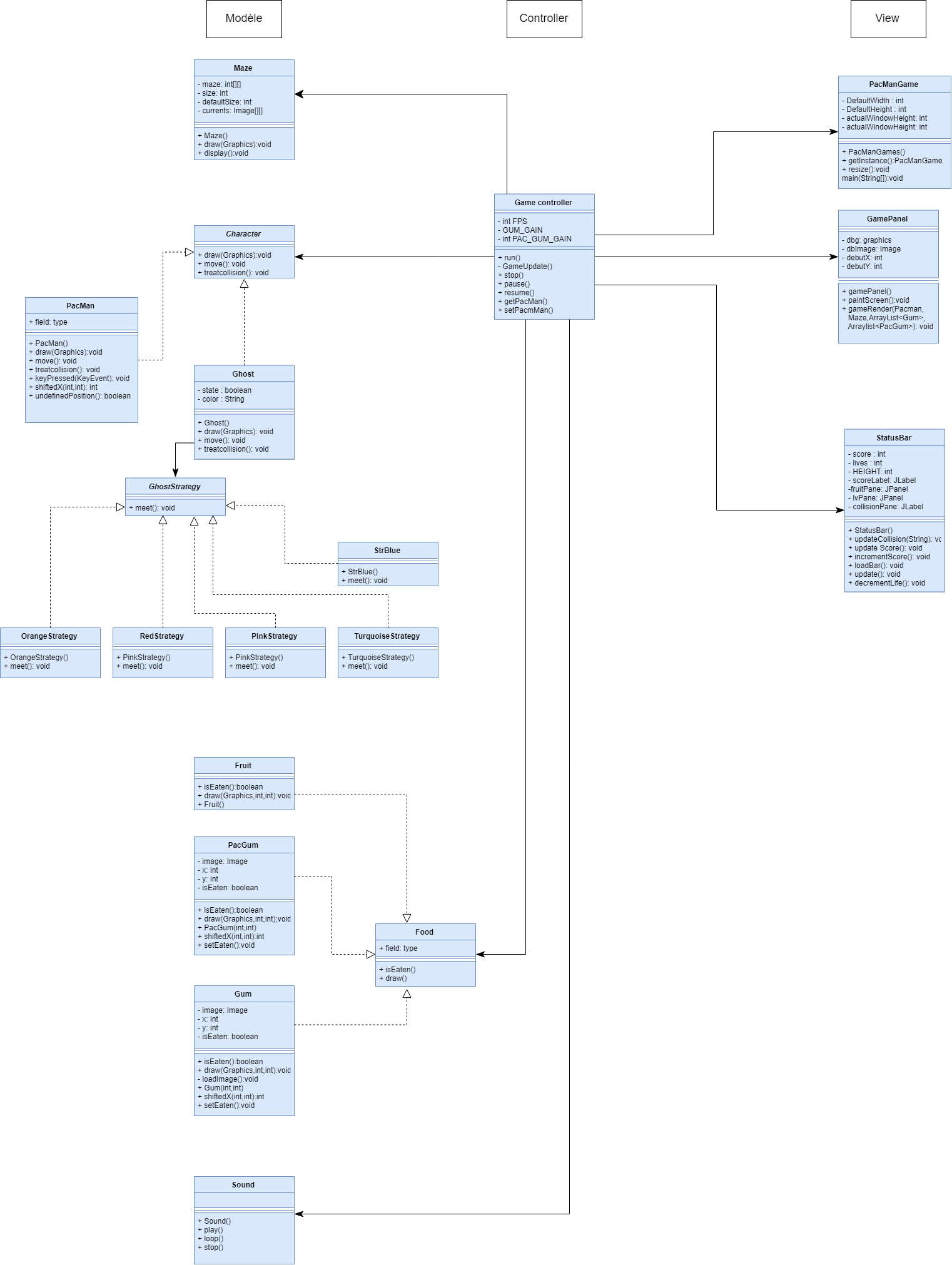


Une fois ce « product backlog » réalisé, nous organisons un sprint planning ainsi qu’un sprint backlog permettant de préciser les tâches à réaliser dans le sprint à venir ainsi que l’attribution des tâches. Nous ferons de même pour le deuxième sprint. Ces répartitions sont disponibles dans le tableau de bord.

# Diagramme de classe

Avant de commencer le code de notre jeu, il nous faut encore réaliser un diagramme de classes permettant d’anticiper la structure de notre logiciel.

Voilà le diagramme de classe réalisé.



Afin de pouvoir gérer l’évolution des données du jeu indépendamment de son affichage, nous utilisons le modèle de conception Model-View-Controller : le contrôleur « GameController » va manipuler les données présentes dans les modèles (Characters (dont PacMan, Ghosts…), Food(Gum, PacGum…), Maze) et demander la mise à jour de l’affichage aux Views qui ne gèrent que l’affichage (GamePanel pour le panel de jeu, StatusBar pour l’affichage des scores, PacManGame pour le cadre du jeu).

Nous utiliserons également le modèle de conception Strategy pour gérer les fantômes ayant des stratégies différentes en fonction de leur couleur. Dans ce cas, notre classe de contexte est la classe « Ghost », qui possède une stratégie de l’interface Stratégie et une méthode permettant de la changer, l’interface « Strategy » est l’interface implémentée par les différentes stratégies concrètes (différents algorithmes en fonction du fantôme), elle contient une fonciton « execute » redéfinit dans les différentes stratégies concrètes.

# Tests

# Conclusion

Finalement, après ces deux premiers sprints, nous avons défini une structure de code complète en essayant de respecter au maximum les principes SOLID et nous avons commencé l’implémentation de diverses fonctionnalités (voir tableau de bord). Globalement, nous avons une fenêtre de jeu contenant un labyrinthe à l’intérieur duquel peut se déplacer PacMan selon les instructions du joueur, tout en mangeant des Gums, PacGums et fruits afin d’augmenter son score. Il peut également passer au travers d’un tunnel des deux côtés de l’écran et ne peut pas traverser les murs du labyrinthe. Une musique de fond est lancée automatiquement au début du jeu mais peut être coupé à tout moment à l’aide de la touche m, de même que le jeu peut se mettre en pause à l’aide de la touche P. La status bar indique au joueur son score et nombre de vies restantes. Le joueur peut jouer en taille réduite ou en plein écran et changer de mode lorsqu’il le souhaite.