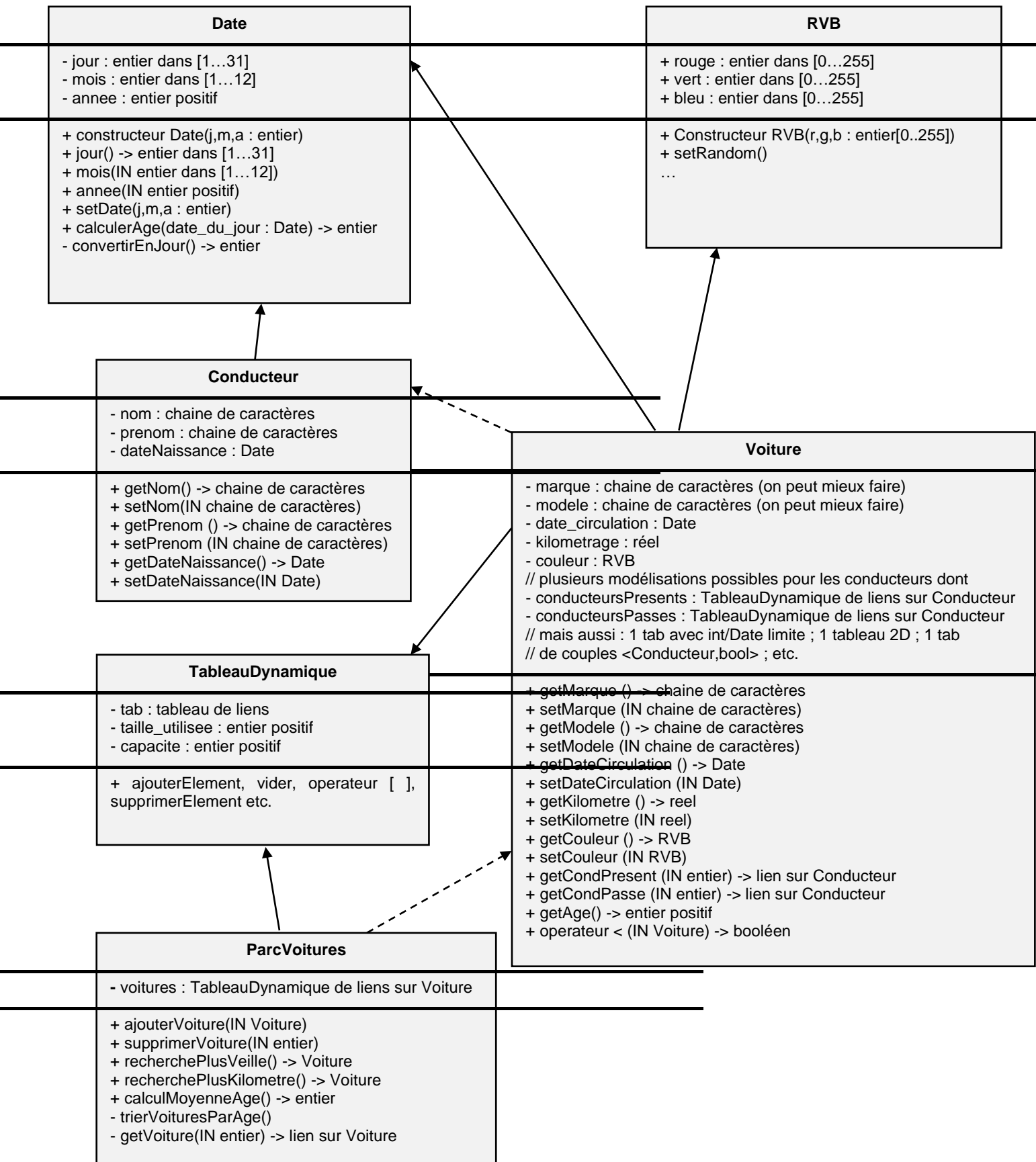


Corrigé – Parc de voitures

Diagramme des classes

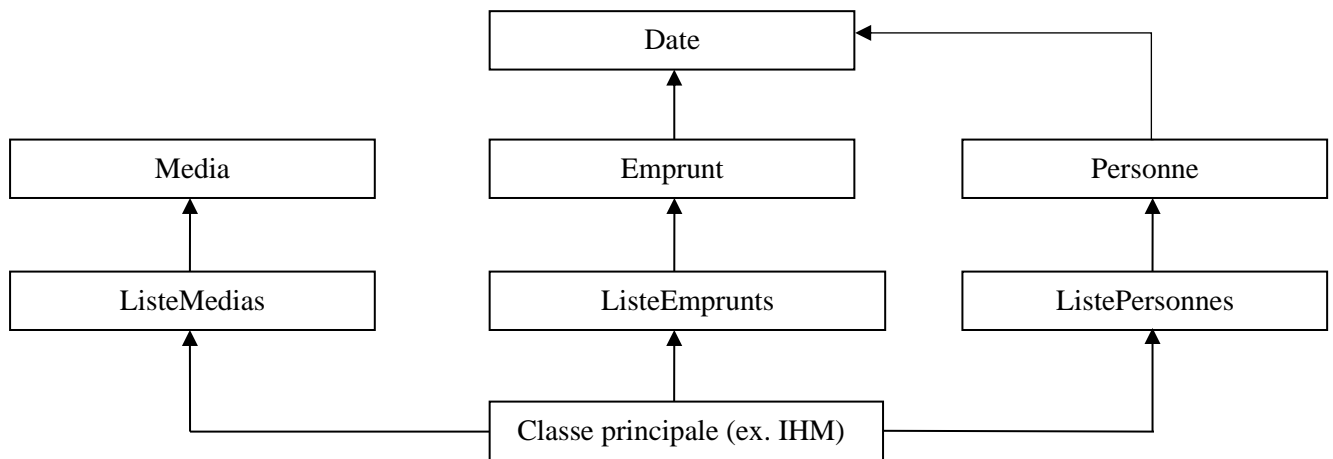


Corrigé – Médiathèque

Liste des classes

Date		Emprunt	
	<ul style="list-style-type: none"> - jour : entier dans [1...31] - mois : entier dans [1...12] - annee : entier 		<ul style="list-style-type: none"> - num_emprunt : entier - num_media : entier - num_personne : entier - date_debut : Date - date_fin : Date
	<ul style="list-style-type: none"> + getJour () -> entier + setJour (IN entier) + getMois () -> entier + setMois (IN entier) + getAnnee () -> entier + setAnnee (IN entier) + getChaine () -> chaine de caractères 		<ul style="list-style-type: none"> + getNumEmprunt () -> entier + setNumEmprunt (IN entier) + getNumMedia () -> entier + setNumMedia (IN entier) + getNumPersonne () -> entier + setNumPersonne (IN entier) + getDateDebut () -> Date + setDateDebut (IN Date) + getDateFin () -> Date + setDateFin (IN Date)
Media		ListeMedias	
	<ul style="list-style-type: none"> - num_media : entier - titre : chaine de caractères - auteur : chaine de caractères - type : entier dans {LIVRE, CD, DVD, VHS} 		<ul style="list-style-type: none"> - tab_Medias : tableau de Media - nb_Medias : entier
	<ul style="list-style-type: none"> + getNumMedia() -> entier + setNumMedia (IN entier) + getTitre () -> chaine de caractères + setTitre (IN chaine de caractères) + getAuteur () -> chaine de caractères + setAuteur (IN chaine de caractères) + getType () -> entier + setType (IN entier) 		<ul style="list-style-type: none"> + viderListeMedias() + getNbMedias() -> entier + getMedia (IN entier) -> Media + ajouterMedia (IN Media) + supprimerMedia(IN entier)
Personne		ListePersonnes	
	<ul style="list-style-type: none"> - num_personne : entier - nom : chaine de caractères - prenom : chaine de caractères - adresse : chaine de caractères - dateNaissance : Date 		<ul style="list-style-type: none"> - tab_Personnes : tableau de Personne - nb_Personnes : entier
	<ul style="list-style-type: none"> + getNumPersonne() -> entier + setNumPersonne (IN entier) + getNom () -> chaine de caractères + setNom (IN chaine de caractères) + getPrenom () -> chaine de caractères + setPrenom (IN chaine de caractères) + getAdresse () -> chaine de caractères + setAdresse (IN chaine de caractères) + getNaissance () -> Date + setNaissance (IN Date) 		<ul style="list-style-type: none"> + viderListePersonnes() + getNbPersonnes() -> entier + getPersonne(IN entier) -> Personne + ajouterPersonne(IN Personne) + supprimerPersonne(IN entier)
		ListeEmprunts	
			<ul style="list-style-type: none"> - tab_emprunts : tableau d'Emprunt - nb_emprunts : entier
			<ul style="list-style-type: none"> + viderListeEmprunts() + getNbEmprunts() -> entier + getEmprunt(IN entier) -> Emprunt + ajouterEmprunt(IN Emprunt) + supprimerEmprunt (IN entier)

Diagramme des dépendances



Corrigé - Pacman, éléments de réponse pour le cahier des charges

1. Présentation des acteurs

La société InfogamesProf est une Société Anonyme au capital de 1'000'000 d'euros. Son activité principale est l'édition de jeu vidéo sur PC. **blahblah...**

La société UnivGamesEtu est une start-up issue de l'université Claude Bernard Lyon 1. Elle est composée de 4 développeurs à plein temps. Son activité principale est le développement de produits informatiques pour des grands groupes. Son expertise est essentiellement centrée sur le C/C++ et le développement de listes chaînées et de tableau dynamique. **blahblah ...**

Dans ce contexte, la société InfogamesProf commande le produit « Pacman en folie » à la société UnivGamesEtu. Le présent cahier des charges est composé :

- d'une description détaillée du produit
- d'une prévision détaillée du déroulement du développement
- d'un diagramme de Gantt
- d'un diagramme des modules

L'ensemble de ces documents sera maintenu pendant toute la durée de développement du produit.

2. Description du produit

Principe et règles du jeu

Le jeu commence en cliquant sur l'écran de jeu. Le but du jeu est de collecter tous les points dans chaque niveau. Pendant ce temps on est suivi par plusieurs fantômes qui ne doivent pas nous toucher.

Pacman - son pion - est déplacé grâce aux touches 'flèche'. Partout où il passe, il mange les points jaunes. Lorsque tous les points ont été mangés, le niveau est terminé et on passe au suivant. Pacman se transforme en super-pacman en mangeant les pastilles blanches. Super-pacman peut manger les fantômes. Super-pacman redevient pacman après N secondes. Les fantômes ont un déplacement automatique en direction de pacman pour le manger ou alors en direction opposé à super-pacman pour ne pas se faire manger.

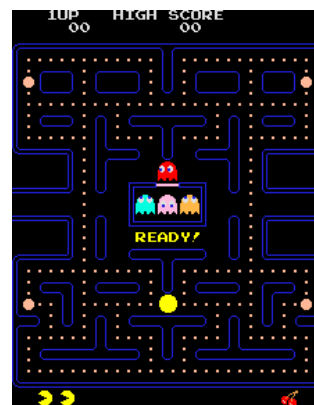
Le jeu se déroule avec une musique et des bruitages en fonction de qui mange qui.

Fin de la partie : le jeu se termine lorsque l'on a perdu ses trois vies.

Détail de l'interface et des niveaux du jeu

Ci-contre un exemple du niveau 1

Croquis des niveaux (au stylo par exemple) ...



Description de la gestion des scores

Descriptions des autres fonctionnalités ...

3. Contraintes

- Le jeu sera développé en C/C++ sous Linux
- Les bibliothèques utilisées seront SDL2 et une bibliothèque texte pour une version alpha
- Le code respectera le standard suivant : code indenté, variable ayant du sens, etc.
- Le code sera géré et archivé par Mercurial
- La documentation du code sera produite par Doxygen
- Un diagramme des modules permettra d'avoir une vision de haut niveau de l'implantation
- L'équipe de développement utilisera les outils de debug et de profiling : gdb et valgrind
- Le code sera fourni à la société InfogamesProf

4. Déroulement du projet

Tâche 0 : rédiger le cahier des charges

Membres impliqués : tous
Durée : 2 semaines

Tâche 1 : définir le diagramme des modules

Membres impliqués : tous
Durée : 1 semaine

Tâche 2 : développement d'un 1er prototype en mode texte

Durée : 3 semaines

Tâche 2.1 : écriture et test du module Pacman (Pacman.h/.cpp)

Membres impliqués : Bob Lecodeur
Un pacman est défini par une position, un état, un score, **blahblah**
Le pacman peut se déplacer vers la gauche, la droite etc. **blahblah**

Tâche 2.2 : écriture et test du module Terrain (Terrain.h/.cpp)

Membres impliqués : Pierre Lerouleur et Joe Lefort
Ce module pourra charger un terrain depuis un fichier. Le format d'un fichier de données respectera la syntaxe suivante : **blahblah**

Tâche 2.3 : écriture et test du module Fantôme (Fantome.h/.cpp)

Membres impliqués : Bob Lecodeur et Pierre Lerouleur
Définition des classes pour un Fantome: **blahblah**

Tâche 2.4 : écriture et test du module Jeu (Jeu.h/.cpp)

Membres impliqués : **blahblah**
Ce module propose toutes les fonctions pour gérer une étape du jeu :

- reçoit en entrée ce qu'un utilisateur a tapé au clavier
=> Une structure sera définie à cet effet
Déplace le pacman en utilisant le module Pacman et Terrain
- déplacement d'un fantôme en utilisant le module Fantôme et Terrain
- mise à jour du score
- test de fin de niveau

Attention : ce module ne fait aucun affichage. Il doit pouvoir servir à la version texte comme à la version graphique.

Tâche 2.5 : écriture du module JeuModeTexte (`JeuModeTexte.h/.cpp`)

Membres impliqués : **blahblah**

Récupère les données de Terrain, Pacman et Fantôme et les affiche au format texte.
blahblah

Tâche 2.6 : mise en commun des modules et tests. **blahblah**

Tâche 3 : développement d'un 2e prototype en mode graphique (SDL2)

Durée : 4 semaines

Tâche 3.1 : exploration et compréhension de SDL2

blahblah

Tâche 3.2 : Affichage de sprites et tests de performance

blahblah

Tâche 3.3 : Développement d'une boucle de jeu en SDL2 dans le module JeuModeGraphique (`JeuModeGraphique.h/.cpp`) et intégration au reste du jeu

blahblah

Tâche 3.4 : Test et debug

blahblah

Tâche 4 : développement d'un 3e prototype en mode graphique avec la musique et gestion des scores

blahblah

Tâche 5 : développement d'un 4e prototype avec ajout d'un mode multi-joueurs

blahblah

Tâche 6 : développement d'un 5e prototype multi-joueurs en réseau

blahblah

5. Diagramme de Gantt

	Semaine 1	Semaine 2	Semaine 3	Semaine 4	Semaine 5	Semaine 6	Semaine 7
Tâche 0	X	X					
Tâche 1		X					
Tâche 2.1		X	X				
Tâche 2.2		X	X				
Tâche 2.3		X	X				
Tâche 2.4			X	X			
Tâche 2.5			X	X			
Tâche 3.1		X	X	X			
etc.							

6. Diagramme des classes

