



UNIVERSITÉ D'AVIGNON
ET DES PAYS DE VAUCLUSE

C E N T R E
D'ENSEIGNEMENT
ET DE RECHERCHE
EN INFORMATIQUE



Licence Informatique
Ingénierie du Logiciel
UE Projet Curve Fever

Rapport 1

Charlie Brugvin & Sabri Boussetha

13 octobre 2015

CERI - LIA
339 chemin des Meinajariès
BP 1228
84911 AVIGNON Cedex 9
France

Tél. +33 (0)4 90 84 35 00
Fax +33 (0)4 90 84 35 01
<http://ceri.univ-avignon.fr>

Table des matières

Titre	1
Table des matières	2
1 Présentation de notre composante	3
2 Description des interactions avec les autres composantes	3
3 Structure de données partagées	3
4 Formalisation de la structure de classe	3

1 Présentation de notre composante

Notre composante est « moteur physique ». Notre binôme devra donc implémenter un ensemble de classe et méthodes qui permettent de gérer tout ce qui est déplacement, collision et classement LOCALE à une manche.

2 Description des interactions avec les autres composantes

Nous allons décrire ici les informations reçues et envoyées entre notre composante et les trois autres :

Composante graphique : On ne fait qu'envoyer des informations et on reçoit rien. On envoie en temps réels la position, la taille, l'état de chaque élément c'est-à-dire le snake et les items (bonus).

Composante Interface utilisateur : Nous envoyons à cette interface l'issue des manches, pour qu'ils puissent gérer la partie (qui est composée de plusieurs manches). Nous recevons de cette composante l'ID d'un joueur pour pouvoir différencier "les snakes" et nous récupérons aussi les actions des joueurs (l'appui sur une touche de clavier pour contrôler son snake).

Composante réseau : Selon nous, nous ne communiquons pas avec cette composante car l'échange des informations sur le réseau est géré par la composante interface. Nous communiquerons donc avec le réseau par le biais de l'interface.

3 Structure de données partagées

On doit partager avec les autres composantes (Composante graphique et Composante Interface Utilisateur) les classes Snake, Item et Board.

4 Formalisation de la structure de classe

Round

- board type Board
- Rank type Array
- endOfRound type bool
- Function : endRound(), upDate();
- getters and setters;

Snake

- head type Pos
- alive type bool
- speed type integer
- body type Array of BodyPoint
- Function : collisionWithSnake(); collisionWithItem();
- turn(); getters and setters;

Pos

- x and y type Integer
- getters and setters;

BodyPoint

- position type Pos
- full type bool
- getters and setters;

Board

- side type Integer
- snakeListe type Array

- itemListe type Array
- getters and setters;

Item :

étant donné le grand nombre d'items et leurs ressemblance,nous avons décidé de les mettre en relations d'héritage. Avec une classe mère item,qui contient les attributs fondamentaux tel que la position.