



UNIVERSITÉ D'AVIGNON  
ET DES PAYS DE VAUCLUSE

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

>>>

Licence Informatique  
Parcours Ingénierie Logicielle  
UE Projet de programmation

## Rapport01

Gregoire Kevin, Honel Steve & Prénom3 Nom3

19 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 Introduction	3
2 Description des interactions	3
3 Structure de données	3

## 1 Introduction

Le moteur physique s'attaque à la résolution des différents problèmes de mécanique. Il gère principalement les collisions, les déplacements, ainsi que tout ce qui touche aux états des objets. Il est également responsable des entrées de commandes des différents joueurs.

## 2 Description des interactions

Le moteur physique interagit avec les joueurs par le biais des commandes et des déplacements. Il interagit également avec l'interface qui lui transmet les commandes des joueurs. Il envoie les mises à jour à l'interface.

## 3 Structure de données

Partagée : L'interface communique au moteur physique l'ID des différents joueurs. Le moteur physique renvoie les ID suivant le classement de la partie. Le moteur physique envoie les déplacements à l'interface

Class : Round : Attributs : int players[]; //Contient les différents id des joueurs de la partie  
Field field; //Contient les caractéristiques de la map Méthodes : void dead(int idPlayer); //Envoie à l'interface d'un décès

Field : Attributs : int L; //Longueur de la map int l; //Largeur de la map Case Map[L][l];  
//Tableau 2D de case Méthodes : Field() //Constructeur par défaut de taille + initialisation des  
toutes les cases... Field(int L, int l); // Constructeur de la map avec taille personnalisée + initialisation  
des toutes les cases...

Case : Attributs : int x; int y; Player owner; //si un joueur est dessus ou non Méthodes : bool  
isOccupied(); //Renvoie un bool suivant l'état de la case int isOccupiedBy(); //Renvoie -1 si libre  
sinon renvoie l'ID du joueur

Snake : Attributs : int idPlayer; int headX; //Position de la tête de serpent int headY; double  
direction; bool alive;