

Rapport Le jeu du serpent

L'objectif de ce TP est de programmer le très célèbre jeu du serpent. Le principe est de piloter un serpent (vu du dessus) pour attraper des pommes et grandir progressivement.

Exercice 1. (Conception du projet)

2. Le découpage a été exécuté de manière à avoir:

-1 module "snake" - Génération, ajout, recherche, suppression, de pomme - La gestion du poison (case occupée...) - Les cases occupées (par le serpent ou une pomme) - L'initialisation et tout ce qui est déplacements, avancement, changement direction, mort du serpent ainsi que le fait que ce dernier mange une pomme - L'initialisation du monde

-1 module "snake_graphique" - Gestion de l'affichage du quadrillage - Gestion de l'affichage de la pomme - Gestion de l'affichage du score - Gestion de l'affichage du serpent (tête, corps, queue) - Gestion de l'affichage du monde - Gestion de l'affichage début/fin de la partie et du "pause"

3. On avait commencé par plus découpé le module actuel "snake" mais on a fini par regrouper car selon nous c'est plus adapté de cette façon.

Exercice 2. (Déclarations des types)

1. Le type Case est composé de deux entiers ("int x", "int y") pour représenter la case du quadrillage.
2. Le type Direction est composé de "NORD", "SUD", "EST", "OUEST" pour représenter les différentes directions.
3. Le type Serpent est composé d'un tableau statique, d'une Direction et de deux entiers "taille" et "mort".
4. Le type Pomme est composé d'une case, et d'un type de pomme. Le type de pomme correspond à normal, super ou poison.
5. Le type Monde est composé d'un type serpent, d'un type pomme et de deux entiers "nb_pommes" et "score".

Exercice 3. (Mécaniques du jeu)

1. Pour la fonction “pomme_gen_alea(int m, int n)”, on utilisera notamment les assertions, la fonction randint, et la fonction generetype.
2. Pour la fonction “ajouter_pomme_monde”, on utilisera en particulier la fonction précédente(“pomme_gen_alea(int m, int n)”) ainsi que “case_occupee_serp” et “pomme_tout_poison”.
3. On écrira la fonction “Serpent init_serpent()” principalement à l’aide du type Serpent.
4. Pour la fonction “Monde init_monde(int nb_pommes)”, on a eu besoin d’utiliser la fonction “assert”, “init_serpent()” et “ajouter_pomme_monde(&m)”.
5. Pour la fonction “deplacer_serpent”, on utilisera les fonctions “assert”, “case_suivante”, “case_occupée_serp”, “case_occupee_pomme” et “avance_serpent”.

Exercice 4. (Graphisme)

1. Tout cet exercice est géré dans le module “snake_graphique”.