Projet NSI - Lemmings

1 Objectif

L'objectif de ce projet est de programmer une version très simplifiée du jeu des *Lemmings* (édité en 1991). Sa programmation vous permettra de mettre en œuvre la programmation objet. Ce sera, bien sûr, également l'occasion d'appliquer vos connaissances en programmation (boucles, condition, fonction, structures telles que tableaux et tuples)
En outre, comme pour tout projet, diverses capacités sont attendues :

- Travailler en groupe
- Gestion de projet
- Écriture / respect d'un cahier des charges
- Rédaction d'un rapport

2 Le jeu

Dans ce jeu, les lemmings marchent dans une grotte représentée par un tableau à 2 dimensions dont chaque case est soit un mur soit un espace vide. Un espace vide peut contenir un lemming au plus à un instant donné. Les lemmings apparaissent l'un après l'autre à une case de départ et disparaissent à une case de sortie. le but du jeu est de faire sortir le maximum de lemmings, le plus rapidement possible. Chaque lemming a une coordonnée horizontale et une coordonnée verticale indiquant la case qu'il occupe, ainsi qu'une direction (gauche ou droite). Les lemmings se déplacent à tour de rôle, toujours dans l'ordre de leur apparition dans le jeu, en respectant les règles suivantes :

- si l'espace immédiatement en dessous est libre, le lemming tombe d'une case ;
- sinon, si l'espace immédiatement devant lui est libre, le lemming avance d'une case ;
- sinon le lemming se retourne.

3 Consignes

Le travail doit être fait en groupe de 2 à 3 personnes.

3.1 Dates de rendu de projet

- Une première version devra être présentée au retour des vacances de la Toussaint.
- Le projet doit être rendu au plus tard le dimanche 9 novembre à 21h59 par la messagerie de l'ENT.

Tout projet rendu en retard se verra automatiquement attribuer la note de 0.

3.2 Cahier des charges

Le programme devra comporter une classe `Lemming` pour les lemmings, une classe `Case` pour les case de la grotte, et une classe principale `Jeu` pour les données globales.

- La classe principale `Jeu` contient un attribut `grotte` contenant un tableau à 2 dimensions de cases et un attribut `lemmings` contenant un tableau des lemmings actuellement en jeu. Son constructeur initialise la grotte à partir d'une carte donnée par un fichier texte, où `#` représente un mur, où les les lemmings apparaissent au niveau d'une case vide de la première ligne et `O` représente la sortie.
  Cette classe fournira les méthodes suivantes :
  - `affiche(self)` affiche la carte avec les positions et directions de tous les lemmings en jeu ;
  - `tour(self)` fait agir chaque lemming une fois et affiche le nouvel état du jeu ;
  - `demarre(self)` lance une boucle infinie attendant des commandes de l'utilisateur. Par exemple, `'l'` pour ajouter un lemming, `q` pour quitter et `Entrée` pour jouer un tour.
- La classe `Lemming` contient des attributs entiers positifs `l` et `c` qui indiquent la ligne et la colonne de la case occupée par le lemming et un attribut `d` valant 1 si le lemming se dirige vers la droite et -1 s'il se dirige vers la gauche. On ajoutera un attribut `j` indiquant l'objet de la classe `jeu` pour laquelle le lemming a été créé, ce qui permettra d'accéder au terrain et à la liste des lemmings.
  Par ailleurs, cette classe fournira les méthodes suivantes :
  - `__str__(self)` renvoie `'>'` ou `'<'` suivant la direction du lemming ;
  - `action(self)` deplace ou retourne le lemming ;
  - `sort(self)` fait sortir le lemming du jeu.
- La classe `Case` contient un attribut `terrain` contenant le caractère qui représente la caractéristique de la case (mur, vide, sortie) et un attribut `lemming` contenant l'éventuel lemming présent dans cette case et `None` si la case est libre.
  Par ailleurs, cette classe fournira les méthodes suivantes :
  - `__str__(self)` renvoie le caractère à afficher pour représenter cette case ou son éventuel occupant ;
  - `libre(self)` renvoie `True` si la case peut recevoir un lemming (ni un mur ni occupée) ;
  - `depart(self)` retire le lemming présent ;
  - `arrivee(self, lem)` place le lemming `lem` sur la case ou le fait sortir du jeu si la case était la sortie.

3.3 Rendu de projet

Le projet doit comprendre :

1. Le code du projet
2. Un rapport de 4 pages **maximum**. Ce rapport contiendra :
   - le cahier des charges,
   - la répartition des tâches au sein du groupe et l'organisation (moments d'échange et éventuels outils utilisés),
   - la présentation d'un sous problème et de sa solution algorithmique (une par membre du groupe),
   - les méthodes de débogage, tests,
   - les sources utilisées,
   - les évolutions éventuelles.

4 Évaluation

Le projet sera évalué sur les points suivants (barème indicatif) :

1. Code (13 points) :
   (a) Efficacité
      i. Code fonctionnel
      ii. Niveau de difficulté
      iii. Interface graphique
   (b) Qualité
      i. Nommage des variables/fonctions
      ii. Commentaires
      iii. Documentation
2. Rapport (7 points) :
   (a) Cahier des charges
   (b) Bilan, sources
   (c) Qualité de l'expression écrite
   (d) Organisation

Plagiat
Le plagiat sera très sévèrement sanctionné. Le code que vous fournissez doit être le vôtre. Si vous êtes amenés à devoir utiliser un bout de code de quelqu'un d'autre, la source doit être citée en commentaire dans le code et vous devez être capable d'expliquer ce qu'il fait. Rappel : Le but du projet est de vous améliorer en programmation, ce n'est qu'en codant qu'on devient un programmeur.

5 Source

Source indiquée après le rendu !