DM - Réalisation du jeu Snake

Ce DM (Devoir à la Maison) est un travail en binôme destiné à évaluer vos connaissances en programmation fonctionnelle avec OCaml. Il sera noté et comptera comme note de contrôle continu pour PRG2. Il doit être rendu à votre enseignant sur support papier et par email pour votre séance de TD 9 (semaine du 28 mars). Au cours des séances de TP 5 ou 6, vous passerez un bref oral individuel, où vous devrez répondre à des questions sur votre solution. Le travail personnel et la qualité de votre solution (correction et présentation) sont aussi importants que son degré d'avancement.

1 Problème

Il s'agit de programmer le jeu bien connu de *Snake* en utilisant la librairie graphique [1] de OCaml. Tous les détails du jeu sont disponibles sur Wikipédia [2].

La section "Cahier des charges" spécifie les contraintes que votre solution doit absolument vérifier. La section "Améliorations possibles" donne des pistes pour améliorer votre jeu. Celles-ci sont optionnelles, mais peuvent vous valoir des points supplémentaires.

2 Cahier des charges

Le programme doit permettre de jouer au jeu de Snake à un joueur selon les règles de base de ce jeu. Le plateau de jeu est constitué d'une bordure, de pastilles et d'un serpent (*snake*). Les pastilles sont disposées de façon aléatoire. Le serpent est initialement de longueur courte et au centre du plateau. Le serpent avance à vitesse constante et le joueur peut seulement contrôler la direction de sa tête avec des touches du clavier. À chaque fois que le serpent mange une pastille, le joueur marque un point et le serpent s'allonge. Le joueur perd dès que la tête du serpent touche le bord du plateau ou le corps du serpent. Lorsque toutes les pastilles ont été mangées par le serpent, soit la partie se termine (le joueur gagne), soit de nouvelles pastilles sont affichées et le jeu continue avec un serpent de plus en plus long. Quand la partie est terminée, le score atteint par le joueur est affiché.

Un squelette de programme commenté `snake.ml` vous est fourni et contient entre autre une boucle d'interaction `loop` minimale. Une alarme système permet d'appeler la fonction `make_step` à chaque pas de jeu (c-à-d., toutes les fractions de seconde) avec comme paramètre l'état courant du jeu et avec comme résultat le nouvel état du jeu. Il vous revient de définir la nature de cet état et les changements d'état. L'appui de la touche 'q' termine le programme.

Pour compiler le programme, exécuter dans un terminal la commande suivante.

1. `http://caml.inria.fr/pub/docs/manual-ocaml/libref/Graphics.html`
2. `http://fr.wikipedia.org/wiki/Snake_(jeu_vidéo)`

```
ocamlc -c snake unix.cma graphics.cma snake.ml
```

Pour lancer l'exécutable produit, il suffit d'exécuter la commande `./snake`.

3 Améliorations possibles

S'il vous reste du temps et que le sujet vous intéresse, il y a de multiples façons d'améliorer le programme.

- afficher le score courant sur le plateau de jeu
- ajouter des obstacles sur le plateau de jeu, autour desquels le serpent devra slalomer
- introduire des niveaux de jeu : quand le serpent a mangé toutes les pastilles, il passe à un niveau plus difficile (plus rapide, ou serpent qui s'allonge plus vite)
- améliorer le graphisme du plateau, du serpent et des pastilles
- faire une version pour deux joueurs

4 Livrables

Il y a 2 documents à rendre : un rapport et le fichier source. Le rapport devra contenir les éléments suivants :

- une introduction présentant votre compréhension du problème,
- votre analyse du problème, son découpage en sous-problèmes (fonctions intermédiaires),
- un bilan de ce que vous avez réalisé, si le cahier des charges est entièrement rempli, si vous avez réalisé des améliorations,
- la commande permettant d'exécuter votre programme,
- un rapport de test de votre programme, avec 2 ou 3 exemples illustratifs,
- une conclusion où vous indiquerez le temps que vous avez consacré à ce projet, les difficultés rencontrées, ce que vous avez appris.

Veillez à soigner la lisibilité du code (commentaires, nom des fonctions et des variables, indentation) autant que la présentation du rapport (orthographe, grammaire, style, mise en page).