

Projet de Conception logicielle avancée (L2 Informatique Semestre 2)

CoreWar

Énoncé : Le but de ce projet est fournir un outil permettant de générer des programmes efficace au CoreWar. CoreWar est eu de programmation dans lequel deux programmes informatiques sont en concurrence pour le contrôle d'une machine virtuelle appelée MARS (Memory Array Redcode Simulator). Le but du jeu est de faire se terminer toutes les instances du (ou des) programme(s) adverse(s). Dans un premier temps, il faudra développer une plateforme de simulation de la machine virtuelle (en utilisant une version simple du langage de programmation appelé RedCode). Dans un second temps, il s'agira d'être capable d'exécuter les programmes écrits en RedCode et de déterminer le vainqueur. Dans la dernière étape, il faudra proposer une méthode (construction aléatoire, système à base de règles ou encore algorithme génétique) permettant d'obtenir un programme performant.

Liens utiles :

- [CoreWar \(Wikipedia\)](#)
- Sites de ressources : [CoreWar \(Ressources\)](#) & [CoreWar Info. \(Ressources\)](#)
- [Algorithmes génétiques \(Wikipedia\)](#)

Éditeur de livres dont vous êtes le héros

Énoncé : Les livres dont vous êtes le héros (ou LDVEH) sont des jeux de rôles en solitaire dont la narration est décomposées en paragraphes, dispersés dans le livre. Des liens, en fonction des choix du lecteurs, permettent d'aller d'un paragraphe à l'autre. Dans un premier temps, il s'agira de développer un éditeur de texte qui maintienne le graphe du jeu et qui puisse réorganiser aléatoirement le livre. L'affichage de la ou les solutions (paragraphe devant être traversés pour gagner) devront être calculées ainsi que la difficulté du jeu (proportions de solutions au regard des chemins possibles). Dans un second temps, il s'agira d'ajouter un système de rencontres et de combat, ainsi que la gestion d'objets ou d'indices qui peuvent être nécessaire pour gagner. Enfin, un mode lecture permettant de jouer au LDVEH pourra être implanté.

Liens utiles :

- [Livre-jeu \(Wikipedia\)](#)
- [ADVELH \(Logiciel\)](#)

Générateurs de flores vidéos-ludiques

Énoncé : Il n'est pas rare de rencontrer arbres, buissons ou plantes, plus ou moins réelles, dans des jeux vidéos ou des films d'animation. Les L-systèmes permettent de représenter ces modèles végétaux sous formes de système de réécriture. Le but de ce projet est donc de réaliser un simulateur de L-système végétal qui prend des règles de réécritures en entrée et produit une image 2D (ou une scène 3D dans un second temps) de l'objet obtenu par la simulation de ce système. Il faudra donc implémenter un parser de L-système, un moteur de réécriture, puis un moteur de rendu graphique pour visualiser ces plantes.

Liens utiles :

- [L-systèmes \(Wikipedia\)](#)

- [The Algorithmic Beauty of Plants de Prusinkiewicz et Lindenmayer \(Article scientifique\)](#)

Interpréteur de programmes chimiques

Énoncé : La programmation chimique (aussi appelée paradigme de programmation par réécriture de multi-ensembles) est un paradigme de programmation dans lequel les données en entrée sont réécrites par des règles du type remplacer A par B si C. Par exemple, un programme qui calcule le maximum d'un ensemble utilise la règle remplacer les entiers x et y par x si $x > y$ et un crible d'Ératosthène utilise remplacer les entiers x et y par x si y est multiple de x. Le but de ce projet est d'écrire un interpréteur de programme chimique qui, à partir d'un langage simple, permettent de spécifier des éléments et des règles de réaction deux à deux. Un premier jalon consistera à définir le langage, réaliser un parser et un interpréteur simple. Dans un second temps, il s'agira d'interpréter de manière parallèle des multi-ensembles (l'équivalent de sous-programmes). Enfin, il est demandé d'étendre le langage pour utiliser des règles de dissolution (détruisant un sous-programme) et d'injection (produisant une donnée sans un sous-programme).

Liens utiles :

- [Programmation chimique \(Article scientifique\)](#)
- [P-Systems \(Wikipedia\)](#)
- [Introduction au parallélisme en Java](#)

Le Castor Affairé

Énoncé : Le but du problème du castor affairé est d'écrire sur une sortie standard (ruban, écran ou variable) la plus grande valeur possible. Afin de rendre ce petit jeu de programmation intéressant il existe deux contraintes : (1) le programme doit pouvoir s'arrêter (ne pas posséder de boucle infinie) ; (2) la taille du programme est limitée (cette taille est une donnée du problème). Ce problème a été pensé initialement sur machine de Turing qui est une machine à états fonctionnant à l'aide de rubans infinis. Dans ce cas, la taille du programme est un nombre d'états. Toutefois, il est aussi possible d'utiliser une machine RAM qui est une machine utilisant des instructions proches de l'assembleur où, cette fois, la taille correspond à un nombre maximal d'instructions. Dans un premier temps il est nécessaire d'implémenter un émulateur de la machine que vous comptez utiliser. Puis dans un second d'implémenter une méthode permettant de générer des programmes et d'optimiser ceux-ci afin d'améliorer le score du castor affairé.

Liens utiles :

- [Castor affairé \(Wikipedia\)](#)
- [Machine de Turing \(Wikipedia\)](#)
- [Histoire et découverte des castors \(Article scientifique\)](#)

Rendu 3D par lancer de rayons

Énoncé : Le but de ce projet est de réaliser un moteur de rendu 3D en utilisant la technique de lancer de rayons. Cette technique permet d'obtenir des rendus 3D très réalistes en simulant les lois de physiques auxquelles sont soumis les rayons de lumière traversant, se réfléchissant ou se réfractant sur différentes surfaces. Il s'agira de réaliser un tel moteur de rendu qui prendra en entrée une description de scène 3D au format .pov et générera l'image correspondante. Il faudra donc implémenter un parser de fichier .pv et le moteur de rendu à proprement

parler. La réflexion et la réfraction seront implémentées dans un premier temps. Dans un second temps, un ombrage de Phong sera implémenté.

Liens utiles :

- [Introduction au lancer de rayons](#)
- [POV-Ray](#)
- [Ombrage de Phong \(Wikipedia\)](#)

Simulateur pour N corps

Énoncé : Le problème à N corps est un problème d'astronomie classique où plusieurs corps se déplacent dans l'espace en étant soumis à leur propre inertie et l'attraction des autres corps. L'équation différentielle qui modélise ce problème est en pratique inutilisable pour $N > 2$. Le but de ce projet est dans un premier de simuler un espace newtonien où N corps interagissent et visualiser cette simulation. Il s'agira ensuite d'améliorer ce simulateur avec diverses propositions parmi les suivantes : instancier des chorégraphies à N corps, accélérer l'optimisation avec un découpage spatial récursif, intégrer un jeu de pilotage d'un corps au clavier, développer une IA pour optimiser les déplacements avec une trajectoire faible en énergie comme les orbites de transfert.

Liens utiles :

- [Algorithmique pour le problème à N corps](#)
- [Chorégraphies à N corps](#)
- [Découpage spacial récursif \(Wikipedia\)](#)
- [Orbites de transfert \(Wikipedia\)](#)

Solveur de Ricochet Robots

Énoncé : Le principe du jeu de société Ricochet Robots est de trouver en moins d'une minute la séquence de mouvement qui permettra à un robot donné (parmi quatre) d'atteindre un objectif désigné sur une case du plateau de jeu. Cependant, les robots ne peuvent que se déplacer en ligne droite jusqu'à rencontrer un obstacle. Le but de ce projet est de développer un programme permettant de trouver une solution optimale pour toute situation du jeu. Dans un premier temps, il s'agira de développer le moteur du jeu (cf. règles) puis d'implanter un algorithme de résolution naïf, appelé A*. Cependant, le problème est trop complexe pour être résolu dans de bonnes conditions. Dans un second temps, il s'agira alors de proposer des méthodes d'optimisation de l'algorithme, par exemple en utilisant des tables de transposition. Enfin, s'il reste du temps, il pourra être intéressant de réaliser une interface graphique permettant à un utilisateur de sélectionner un objectif.

Liens utiles :

- [Ricochet Robots \(Règles du jeu\)](#)
- [Plateau de jeu \(Image\)](#)
- [Algorithme A* \(Wikipedia\)](#)
- [Tables de transposition \(Wikipedia\)](#)