Projet Techniques de Programmation
ITII P20
2011

Sujet : Réalisation d'un projet de simulation intégrant un simulateur à Evénements Discrets

A rendre le 3 septembre 2011

A Stéphane VERA & Christian PAUL
Informations supplémentaires (paul.christian.emse@gmail.com)

Tout les documents rapports, documentation et CD devront être remis dans une seule pochette

Spécifications

Le programme doit permettre de réaliser les actions suivantes :

Au niveau de la simulation :
- lancer une simulation,
- arrêter une simulation,
- modifier les paramètres de la simulation (accélérer / ralentir le temps simulé),
- Sauvegarder une simulation à un temps t donné (t > tcourant),
- Charger une simulation depuis un fichier,

Au niveau des objets :
- Créer,
- Supprimer,
- Modifier,
- Afficher,
- Sauvegarder les objets et leur état (au format texte),
- Charger des objets et leur état (au format texte),

Au niveau de l'échéancier et des évènements :
- Créer un évènement,
- Supprimer un événement,
- Modifier un évènement,
- Afficher la file des évènements,
- Sauvegarder dans un fichier la file des évènements en cours,
- Charger depuis un fichier une file des évènements,

Contraintes techniques

Parties du projet à écrire impérativement en C ANSI :
- La gestion des fichiers,
- L'analyse des fichiers,
- Les files d'objets individus et services,
- La gestion de l'échéancier,
- Le moteur de simulation,

Partie du projet laissée à l'appréciation des concepteurs :
- Les interfaces d'écrans pour l'application générique,
- La formalisation du système de simulation,
- Les interfaces applicatifs, si il y a une application spécifique utilisant le simulateur.

Techniques de programmation à utiliser impérativement :
- Modularité,
- Listes chaînées,
- Récursivité,

Environnement(s) cible(s) :
- Linux impératif,
- Windows autorisé,

Documents à fournir

Ces documents sont à fournir à la première évaluation (voir plus loin)

1) Le rapport sur le projet contenant :
- L'analyse des besoins (retranscrivant le travail préalable au codage),
- Les spécifications de l'application (décrivant les choix techniques arrêtés, l'architecture du programme),
- La procédure de test mise en œuvre pour valider l'adéquation du programme au "cahier des charges"
- L'organisation du binôme sur les tâches de conception et de réalisation,
- Le calendrier prévisionnel et le calendrier réalisé,

2) Les documentations d'utilisation de l'application :
- Guide d'installation (et de désinstallation !),
- Guide d'utilisation.

3) Les sources

4) Le ou les exécutables

5) Le document de présentation du projet (voir plus loin).

Ces documents seront à fournir sur sur support numérique pour (1), (2), (3) et (4).

Notation (N)

10<=N<12 : Le projet est mené à terme dans le respect du cahier des charges et des spécifications, la documentation est correcte. La présentation en binôme et la présentation individuelle montre que le sujet est maîtrisé.

12<=N<15 : **Le cahier des charges est respecté.** La réalisation est soignée et la présentation est bonne. Des améliorations, suggérées ou non, ont été produites. La présentation en binôme et la présentation individuelle montrent une bonne maîtrise du sujet. Le sujet choisi est plus ou moins complexe, bien analysé et bien spécifié.

15<=N<20 : **Le cahier des charges est respecté.** La réalisation est particulièrement soignée, la présentation excellente. Des améliorations, suggérées ou non, ont été produites. La présentation en binôme et la présentation individuelle montre une très bonne maîtrise du sujet. Le sujet choisi est plus ou moins complexe, bien analysé et bien spécifié.

0<=N<10 : **Le cahier des charges n'est pas respecté.** En particulier la présentation en binôme montre que les objectifs ne sont pas atteints ou les tests du correcteur montrent des erreurs de conception ou la présentation individuelle montre que l'élève ne s'est pas investi dans le projet.

Dates :

- Le 3 septembre 2011 : Première évaluation et fourniture des « délivrables » du projet

- Le 1 octobre 2011 pour l'entretien individuel

Introduction à la simulation

La simulation à événement discrets (SED) permet de représenter et d'étudier le comportement des systèmes physiques dont la caractéristique principale est d'évoluer dans le temps par des changements d'états.

Vous devez mettre en œuvre un simulateur à événement discrets sur un sujet de votre choix. Les exemples classiques sont :

- L'atelier de lavage de voiture (voir le sujet de l'année dernière)
- La modélisation des systèmes de gestion de production (atelier de fabrication de pièce)

Vous pouvez aussi appliquer la simulation à évènements discrets sur des exemples que vous vous approprierez :

- Atelier de lavage de voiture,
- Simulation et prévisions de bouchons aux péages autoroutiers,
- Simulation de propagation d'incendie,
- Mini jeu de type Sims(Will Wright)

Principes à prendre en compte pour le projet

Les principes suivants de la simulation à événement discrets devront adoptés dans le projet.

Objets manipulés

Vous utiliserez principalement deux types d'objets gérés sous la forme de files :

- Les individus (dont les caractéristiques évoluent dans le temps),
- Les services qui traitent les individus,

Evènements générés

Ces objets génèrent des évènements qui à leur tour sont susceptibles de modifier l'état courant des objets. Les événements seront caractérisés par :

- L'émetteur (`id_emet`),
- Le destinataire (`id_dest`),
- Le type d'action, (`type_act`),
- La date future de l'action (`t_evt`),

L'échéancier des évènements (structure de type ECH) est une liste d'évènements (EVT) maintenue sous forme d'une file d'attente à priorité. C'est la date de l'action (`t_evt`) qui permet l'ordonnancement dans la liste.

Evidemment tout événement généré à un temps courant t_cour se déroulera dans le futur (t_evt > t_cour).

Evènement	Echéancier

typedef struct evt
{
  int id_emet;
  int id_dest;
  int type_act;
  int t_evt;
  ... /* autres info suivant système */
  struct evt * suiv;
} EVT;
	typedef struct
	{
	  int t_cour;
	  int nb_evt;
	  EVT * debut;
	  ... /* autres informations suivant système*/
	} ECH;

Temps simulé

Une simulation démarre à un temps (t_ini) et se termine à un temps (t_fin). Le temps courant (t_cour) de la simulation permet est généralement accéléré par rapport au temps réel. Sur certains simulateurs il est possible de régler la valeur du temps simulé, par exemple : une seconde simulée = une heure réelle.

Le temps courant de la simulation (t_cour) est géré par le simulateur qui interroge l'échéancier des évènements. Le temps courant t_cour de la simulation peut être donné par:

1) un compteur qui s'incrémente. A chaque tic de l'horloge il faut vérifier et éventuellement traiter les évènements de l'échéancier.
2) le temps de l'élément en cours de traitement. On prend dans l'ordre de priorité le premier événement de l'échéancier. On temporise si nécessaire pour garder un écoulement constant du temps.

Le moteur du simulateur est résumé ci-dessous. La simulation continue tant qu'il y a des évènements dans l'échéancier. A chaque traitement du premier évènement de l'échéancier, le simulateur génère et intègre dans l'échéancier de nouveaux évènements.

```
tant que (toujours)
faire
      toujours = simul(ECHEANCIER);
fin faire
/* D'autres paramètres que l'échéancier seront utilisés en fonction du système spécifié */
int simul(ECH *A,. . . )
{
      int temps_traite;
      int continue_simul = 1;
      EVT evt_a_traiter;
      EVT * liste_evt_crees = NULL;
      if (file_vide(A))           // fin de la simulation
            continue_simul = 0;
      else
      {
            retirer(A, &evt_a_traiter);
            traiter_evt(evt_a_traiter, A);
      }
  return continue_simul;
}
```

Les fonctions **`file_vide(ECH A)`** et **`retirer(ECH* A, EVT* evt_a_traiter)`** sont des fonctions génériques de traitement des FIFO.

La fonction **`traiter_evt(EVT evt_a_traiter, ECH* A)`** permet de générer les nouveaux événements en fonction de l'événement **`evt_a_traiter`** en cours de traitement. Elle est propre au système spécifié.

Partie spécifique

En fonction du système spécifié, les individus et les services seront différents. Il vous est demandé de spécifier cette partie spécifique en fonction du thème de simulation que vous aurez choisi. Vous spécifierez les objets : les caractéristiques des individus et des services. Vous spécifierez les évènements : ce que les individus et les services génèrent en fonction des évènements dont ils sont destinataires.

Par exemple dans les cas cités plus haut les objets et les événements seraient les suivants.

Atelier de lavage de voiture
Les individus sont les voitures, les services sont les postes de lavage. Un générateur aléatoire produit les évènements initiaux (arrivée des voitures).
Si à l'arrivée d'une voiture un poste de lavage est disponible, l'événement début de lavage pour la première voiture est généré.
Si l'événement début de lavage est traité, il va générer un événement fin de lavage à t+tps_lavage. On peut aussi rajouter un nouveau type d'individu : le laveur.

Simulation et prévisions de bouchons aux péages autoroutiers
Les individus sont des voitures, les services sont les péages. Le réseau autoroutier est matérialisé par des points d'entrée et des points de sortie. Un générateur aléatoire d'évènements produit les arrivées de voitures sur les points d'entrées.
Les services sont les péages, ils ont une ou plusieurs files d'attente, et une capacité de traitement en parallèle.

Simulation de propagation d'incendie
Chaque carré de la carte est un individu. Les individus sont typés en fonction de ce qu'ils contiennent : forêt, mer, zone aride, zone contenant une maison individuelle, zone urbaine. Chaque fois que le feu atteint un carré de la carte, il passe aux voisins (s'ils ne sont pas déjà en train de bruler) au bout d'un temps variable en fonction des conditions météos initiales et de la typologie du carré. Tous les individus connaissent leurs voisins. Les événements générés par les individus sont uniquement à l'attention de leurs voisins.
Dans ce cas là il n'y a pas de services. Seuls les individus interagissent avec leurs voisins.

Mini jeu de type Sims(Will Wright)
On modélise un groupe d'individus ayant les activités suivantes : rester à la maison pour dormir ou manger, aller travailler pour gagner de l'argent, aller au magasin pour acheter de la nourriture, aller au cinéma pour les loisirs. Toutes les actions sont modélisées par des évènements, exemple si un mini-sims s'endort à t, il doit se réveiller à t+8, si il mange à t il termine de manger à t+1.

Les individus ont une position, un compteur travail, un compteur argent, un compteur loisir, un compteur santé, un compteur sommeil, un compteur nourriture.

Les services sont les suivants :
le bureau : il incrémente le compteur travail et le compteur argent des individus, il décrémente le compteur santé (par exemple)

la maison :

une maison correspond à un ou plusieurs individus,
si le min-sims dort sont compteur sommeil s'incrémente, sont compteur santé aussi,
si le mini-sims mange sont compteur sommeil s'incrémente, sont compteur santé aussi,

le cinéma :
le mini-sims y dépense de l'argent, incrémente sont compteur loisir,
le cinéma a une capacité de traitement des individus en parallèle, une ou plusieurs files d'attente.

Les individus

Ils peuvent avoir des coordonnées spatiales et peuvent se déplacer (changement de position).
Ils peuvent avoir des compteurs figurant des caractéristiques spécifiques. Ces compteurs sont mis à jour en fonction des services.
Ils peuvent interagir avec d'autres individus via des évènements.

Les services

Ils sont en général fixes (pas de changement de position). Ils effectuent des traitements sur les individus par le biais d'événements et de changements d'états.

Exemples de caractéristiques :
Capacités de traitement en parallèle : un cinéma, un atelier de lavage de voiture peut "traiter" un nombre N d'individus en parallèle.
Files d'attentes avec taille maximale : un cinéma, un atelier de lavage de voiture peuvent être saturer et gérer des files d'attente d'individus (avec priorité ou non).

Note préliminaire sur le rendu graphique : *En général la spécification d'un SED est couplée avec une représentation graphique du système physique simulé. Ceci implique que les objets manipulés soient positionnés dans un espace donné (2D en général). La représentation graphique du système n'est pas demandée. Vous pouvez cependant utiliser une matrice de caractères pour une représentation minimale (comme dans le jeu de la vie).*

Techniques de programmation utilisées

Vous mettrez en œuvre les techniques de programmation que vous avez acquises dans la deuxième partie du cours, à savoir :

- Piles et Files à l'aide de listes chainées,
- Récursivité,
- Modularité,

Géographie et spatialité

Il n'y a pas de spécification projet sur ces aspects. Vous êtes libre de représenter ou non graphiquement vos données. Certaines simulations n'en ont pas besoin.

Spécification du système étudié

Vous décrirez les objectifs de la simulation : capacité de prévision, étude du système en charge, jeux de rôle avec possibilité de prendre la main sur certains acteurs au cour de la simulation.
Suivant le système vous serez amené à utiliser plusieurs types de structures pour les individus et plusieurs types de structures pour les objets. Il vous est demandé de spécifier le comportement de chaque type d'individu et de chaque type de service.

La spécification intègre a minima :

- Une description globale du système simulé avec ses objectifs,
- La description de la (des) structure(s) associée(s) aux individus,
- La description de la (des) structure(s) associée(s) aux services,
- La description des évènements : condition de déclenchement, condition d'exécution.