Projet Techniques de Programmation

Sujet : Simulateur à Evénements Discrets

Présentation

Il s'agit de produire une application (un exécutable testé et opérationnel) en situation de projet au sein d'une équipe. Le sujet est présenté dans la suite du document. Le code à produire doit être du C ANSI.

Dans ce projet vous devrez respecter strictement certaines contraintes. En revanche vous aurez à choisir et spécifier vous-même le thème.

Dans la suite du document le "client" est l'équipe des enseignants.

Objectifs du projet dans le cursus ITII

Approche « Programmation »

Il s'agit de valider les compétences de la seconde partie du cours de programmation par un programme de 1000 à 2 000 lignes de code en langage C, faisant un tout cohérent.

Le projet vous permet de faire du développement autrement qu'en temps limité et sur des petits exercices de validation.

L'application à produire vous demande un travail depuis les spécifications jusqu'aux tests et à la notion d'exécutable opérationnel.

Le but principal est donc de rendre un ensemble contenant l'exécutable testé et opérationnel, un document de conception, une documentation d'installation et d'utilisation.

L'attention doit être mise sur le code : présentation (indentation, nommage des variables, commentaires), l'architecture : les fonctions (dédiées à un usage, d'environ 20 lignes maximum avec variables locales et passage de paramètres).

Techniques de programmation à utiliser impérativement : fonctions, pointeurs, tableaux, structures, mémoire dynamique, fichiers, librairies standards.

Approche « Ingénieur »

Techniquement vous allez aborder les aspects spécification et architecture, vous préoccuper du cycle de développement complet.

Enfin il s'agit de « livrer » un produit fini, testé et opérationnel et d'intégrer complètement cette démarche.

Exigences Organisationnelles

Les étudiants travaillent en binôme ou trinôme, à titre exceptionnel (dérogation) seuls.

Rôle au sein de l'équipe

Rôle technique

Rôle concepteur (expression des besoins, définition des fonctionnalités)

Rôle architecte technique (architecture du programme, spécification des fonctions, recette du code C, assemblage des parties, tests d'intégration)

Rôle développeur (production du code en C, développement suivant spécifications, tests unitaires)

Rôles gestion de projet

Rôle communication (présentation : préparation, animation)

Rôle qualité (plan de tests, définition des objectifs, validation des objectifs, mise au point des jeux de tests et validation des tests)

Rôle suivi du projet (gestion de l'avancement du projet)

Evaluation – ce qu'il faut produire

Evaluation du projet

Chacun rend : une note sur son rôle dans l'équipe, son travail technique personnel, son travail en gestion de projet. Chaque étudiant peut être interrogé individuellement sur des aspects techniques (conception et développement).

La présentation du projet se déroule devant les autres élèves. Le temps imparti à la présentation vous sera précisé par la suite. Il s'agit de 10 à 15 minutes

La note attribuée est fonction de la qualité du code produit, de la démarche adoptée, des outils utilisés, mais aussi de la qualité des documents écrits (orthographe et lisibilité). La note tient aussi compte de la présentation du projet.

Le temps de travail est environ évalué à 30 heures dont 20 à 25 heures sur la conception et le développement en C. A noter que certains mettront plus de temps.

Critères d'évaluation

Un projet dont l'exécutable ne fonctionne pas correctement n'a pas la moyenne. Il est donc impératif de viser des objectifs atteignables et de le tester de manière approfondie. La note du projet vient en appréciation de la note du test de mise à niveau.

La note prend en compte les aspects suivants :

• Qualité technique de la réalisation.

• La documentation associée.

• Stabilité et robustesse de l'application (gestion d'erreurs, …).

• Qualité de la présentation et originalité de la mise en œuvre.

• Entretien individuel si nécessaire.

• Les techniques de programmation utilisées.

Documents à rendre (livrables)

• Un document écrit décrivant le thème adressé, les spécifications et la manière (outils, démarche, organisation de l'équipe dont vous avez abordé le projet).

• La note personnelle concernant son travail et son rôle dans l'équipe.

• La documentation d'installation et d'utilisation.

• Architecture Logicielle (modules et principales fonctions).

• La liste des exigences (que vous allez définir vous-même).

• Le cahier de recette avec les exigences réalisées (OK) et celles non validées (FAIL).

• L'ensemble du code

Calendrier et notation

Les documents devront être rendu le jour de la soutenance
La soutenance est prévue le dernier cours de projet de sur la base des documents envoyés.
La présentation finale est de 15 minutes.

Spécifications

Au niveau du simulateur :

1. Lancer une simulation.
2. Arrêter une simulation.
3. Modifier les paramètres de la simulation (accélérer / ralentir le temps simulé).
4. Sauvegarder une simulation à un temps t donné.
5. Charger une simulation depuis un fichier.

Au niveau des objets de la simulation :

1. Créer.
2. Supprimer.
3. Modifier.
4. Afficher.
5. Sauvegarder les objets et leur état (au format texte ou binaire).
6. Charger des objets et leur état (au format texte ou binaire).

Les spécifications fonctionnelles directement liées au thème que vous avez choisi seront à exprimer dans le dossier de spécifications. Elles sont donc laissées à votre appréciation et viennent s'ajouter aux précédentes que l'on peut considérer comme des spécifications fonctionnelles génériques.

Principes à prendre en compte

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 :

1. Les individus (dont les caractéristiques évoluent dans le temps).
2. 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 peuvent être caractérisés:

1. L'émetteur
2. Le destinataire
3. Le type d'action
4. La date future de l'action

Exemple :

- Un Client (émetteur) passe une commande (type d'action) sur un site internet (destinataire) et la livraison sera effectuer à un certain moment(date future action)
- Un personnage (émetteur) effectue une attaque (type) sur un ennemi(destinataire)
- Un personnage sims (émetteur) prend une douche(type, destinataire = lui-même) qui dure un certain temp (date).

Temps simulé

Sur certains simulateurs il est possible de régler la valeur du temps simulé, par exemple : une seconde simulée = une heure réelle, ou avec un system de tour par tour : un tour = 1heure.
Le temps courant de la simulation est géré par le simulateur qui interroge l'échéancier des évènements. Le temps courant 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.

Exemples de sujet

Simulation de caisses de supermarché

Les supermarchés sont modélisés avec des individus (les clients assimilés à leurs chariots) et des services les caisses et éventuellement les rayons boucherie,... Il est possible de simuler des périodes de type samedi après-midi avec un fort taux d'individus.

La simulation prend son intérêt dans l'optimisation des services associés qui peut conduire à un véritable jeu de simulation.

Mini jeu de type Sims

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 son compteur sommeil s'incrémente, son compteur santé aussi, si le mini-sims mange son compteur sommeil s'incrémente, son compteur santé aussi.

Le cinéma : le mini-sims y dépense de l'argent, incrémente son 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.

Jeu RPG Simulateur

Mini jeux RPG

Ce projet consiste à créer un jeu RPG en C où les joueurs contrôlent des personnages (héros) qui interagissent avec le monde et les NPCs (Non-Playable Characters) en temps simulé. Le jeu permet de gérer des quêtes, des combats et l'évolution des personnages.

Personnages (Héros et NPCs) :

- Créer un héros : Le joueur peut créer un personnage avec des attributs initiaux comme la santé, la force, la magie, fatigue et l'expérience.

- Gérer les attributs : Les attributs du héros évoluent en fonction des actions (gagner de l'expérience, prendre des dégâts, utiliser des objets).

- Interactions : Les héros peuvent interagir avec les NPCs pour recevoir des quêtes, acheter des objets ou obtenir des informations.

Monde du Jeu

- Exploration : Le héros peut se déplacer sur une carte du monde divisé en différentes zones (villages, forêts, donjons).

- Événements : Des événements se déclenchent lorsque le héros entre dans une nouvelle zone (rencontres ennemies, découvertes de trésors).

Système de Quêtes

- Attribution de Quêtes : Les NPCs peuvent attribuer des quêtes au héros. Une quête a des objectifs spécifiques (vaincre un monstre, trouver un objet).

- Suivi de Quêtes : Les quêtes en cours sont suivies et affichées pour le joueur. Une fois les objectifs atteints, la quête est complétée et des récompenses sont attribuées (expérience, objets).

Système de Combat

- Initiation de Combat : Les combats peuvent être déclenchés lors des rencontres avec des ennemis dans le monde.

- Gestion des Tours : Le combat se déroule au tour par tour, avec des options d'attaque, de défense et d'utilisation de magie ou d'objets.

- Résolution de Combat : Les résultats des combats modifient les attributs du héros (gain d'expérience, perte de santé).

Gestion du Temps

- Simulation du Temps : Le temps dans le jeu est simulé (par exemple une action = x minutes) et affecte les événements (le jour et la nuit influencent la visibilité, certains événements ne se produisent qu'à des moments spécifiques).

- Sauvegarde/Chargement : Le jeu peut être sauvegardé à tout moment et chargé plus tard, conservant l'état du monde et du personnage (les évènements prévu doivent aussi être sauvegardé).

Exemples d'Interactions

Exploration :

- Un héros (émetteur) entre dans une forêt (destinataire) à t=0, et une rencontre ennemie est déclenchée à t+5 (date future).

Quête :

- Un héros (émetteur) accepte une quête (type d’action) de récupérer un artefact de NPC (destinataire). L'objectif est de trouver l'artefact dans un donjon avant t+100 (date future).

Combat :

- Un héros (émetteur) attaque (type d’action) un ennemi (destinataire) pendant un combat. Le résultat (dégâts infligés, expérience gagnée) est résolu immédiatement.