

Projet Long Technologie Objet

Groupe KL-5

Membres :

- M'HAND OUAMMI Mohamed
- NEGREL--JERZY Philippe (Animateur)
- OUJID Selma
- OUKHNINI Hamid
- PARAIN Félix
- PONT Sébastien

Sommaire des idées

1. application pour apprendre une langue en devinant des mots et avec un système de "score" pour chaque mot.
2. système de stockage de données cartographiques avec pathfinding
3. gestion du stockage et de la distribution de produits
4. une smart city, Genre on fait des modules pour le business, Module pour le tourism
5. une simulation physique (à approfondir)
6. Une application qui réalise l'exécution et la compilation en java (IDE)

Système de stockage de données cartographiques avec pathfinding

Ce projet consiste à réaliser une application capable de stocker et d'afficher des données cartographiques de tout type (i.e. relief, axes de communication, cours d'eau, océans, bâtiments, etc...)

L'interface utilisateur sera assez intuitive, idéalement en 3d, mais une représentation 2d est déjà amplement suffisante, avec la carte représentée sous forme d'un plan que l'on peut déplacer à l'aide de la souris. Il sera possible de zoomer/dézoomer afin d'avoir une vision plus globale. Une barre de recherche permettant de localiser un élément géographique à partir de son nom ou de ses coordonnées sera également disponible.

Il sera également possible de rechercher le chemin le plus court entre deux points de la carte, en empruntant des moyens de transports au préalable sélectionnés par l'utilisateur.

L'enregistrement des données en RAM se fera selon un système communément utilisé en 3d : l'ensemble des points - ou nœuds - affichés sur la carte sera stocké individuellement dans des tableaux prévus à cet effet, avec leur rôle respectif. Séparément on stockera les liens entre les points (les "lignes" qui les relient), ainsi que le type de lien (courbe de niveau, axe routier, berge, sentier, etc...) . Chaque point permettra de connaître les liens qui l'utilisent, et chaque lien fera également référence aux points qu'il lie. Ainsi, la recherche d'un trajet d'un point A à un point B se fera en 3 étapes :

- recherche du point d'un axe routier le plus proche de A et création d'un nœud temporaire "A" à ces coordonnées
- recherche du point d'un axe routier le plus proche de B et création d'un nœud temporaire "B" à ces coordonnées
- calcul du chemin le plus court de nœud en nœud en appliquant l'algorithme de pathfinding A* ou Dijkstra (on fera des tests d'efficacité) entre les nœuds "A" et "B"

Il est important de noter que chaque lien caractérisant un axe de communication se verra associer des données telles que le trafic en temps réel, le taux de fréquentation (sentier - autoroute), les véhicules autorisés, ainsi que les éventuelles restrictions sur la vitesse, le sens, etc...

Ces données permettront d'optimiser la vitesse des trajets de façon plus intelligente, afin de proposer des résultats toujours plus pertinents à l'utilisateur.

Enfin, il sera possible sur le long terme que l'application établisse un profil de valeurs (telles que la vitesse moyenne, le type de véhicule préféré, etc...) propre à chaque utilisateur qui ouvrira la porte à une prédiction des temps de déplacement plus personnalisée.

Gestion du stockage et de la distribution de produits

Cette application vise à pourvoir aux besoins d'entreprises qui commercialisent des produits dans une chaîne de distribution. Notre application permettra aux entreprises clientes de gérer, visualiser, moduler des stocks de différents produits dans une multitude de points de vente.

Dans un premier temps, nous mettrons en place une structure qui régit la forme des informations, stockées ensuite dans une base de données. Nous définirons ainsi le type, taille, référence et toutes autres informations concernant les produits vendus par l'entreprise. De même, nous aurons une base de données pour les différents points de vente (localisation, capacités, produits demandés, stocks, vendeurs etc). Enfin, nous pourrions créer une base de données pour les entrepôts (localisation, capacité, stocks etc).

Nous aurions pour cette application, une première interface destinée à la direction de l'entreprise. Celle-ci permet d'avoir une vue d'ensemble sur tous les points de ventes, notifiant en cas de rupture de stocks dans l'un d'entre eux. Il y aura également la possibilité d'avoir une vue spécifique pour chaque point de vente, avec une vision plus détaillée de l'état du magasin. Nous pourrions également envisager un affichage comprenant une carte qui puisse afficher les points de ventes par pays, région, ou secteur prédéfini.

Une seconde interface serait plus destinée aux vendeurs dispatchés à chaque point de vente. Il pourrait servir d'interface d'édition pour mettre à jour les stocks en temps réel et notifier en cas de problème quelconque (rupture de stock, défauts dans les produits, erreur de distribution etc.). La même interface pourrait être utilisée pour gérer les stocks dans les entrepôts, à quelques modifications près.

Enfin, nous pourrions utiliser les informations récoltées pour faire un planning de distribution des ressources, afin d'optimiser les trajets entre les entrepôts et points de ventes tout en veillant à garantir la qualité du service.

Application pour apprendre une langue

Ce projet consiste à développer une application pour apprendre les mots d'une langue. L'utilisateur voit un mot à l'écran et doit écrire sa traduction, puis il fait de même avec les mots suivants. Il peut télécharger des listes de mots trouvées sur Internet ou entrer de nouveaux mots lui-même dans la base de données. Il peut également modifier ou supprimer des mots.

À chaque mot est associé une nature (verbe, nom, pronom...), la traduction en français et dans la langue étrangère et éventuellement un ou plusieurs thème(s) (ex : nourriture, corps humain, vêtements,). L'utilisateur peut alors choisir s'il souhaite apprendre des mots d'un thème ou d'une nature en particulier. Par exemple, il peut choisir d'apprendre les noms sur le thème de la guerre.

On associe également à chaque mot un état d'apprentissage (ex: retenu parfaitement, partiellement retenu, en cours d'apprentissage, pas encore appris...). L'état d'un mot évolue en fonction des dernières réponses de l'utilisateur sur ce mot.

L'application propose à l'utilisateur des mots en faisant en sorte qu'il n'y en ait pas trop dans l'état « en cours d'apprentissage » en même temps, et propose de temps en temps d'anciens mots appris afin de les « réactiver » pour que l'utilisateur ne les oublie pas.

Avec ce projet, il est assez rapide d'obtenir quelque chose de fonctionnel. Mais il est possible de rajouter beaucoup de fonctionnalités pour le perfectionner.

Si le projet avance bien, on envisage également d'implanter des exercices de grammaire.

Simulation physique

Sommaire

- Description courte
- Résumé
- Détails
 - Import d'objets
 - Paramétrage
 - Modèles scientifique
- Évolutions

Description courte

Ce projet consiste à réaliser un logiciel capable de calculer et afficher graphiquement des problèmes physiques.

Résumé

L'utilisateur peut importer des objets 3D dans la scène (format *.obj par exemple). Il les détaille ensuite à l'aide de paramètres physiques (matière, densité, ...), paramètre la scène (forces, température, ...) et les fait interagir entre eux. Le logiciel calcule l'état de chaque point du maillage à chaque intervalle de temps, puis l'affiche graphiquement.

Détails

Import d'objet

Les objets 3D sur lesquels les modèles mathématiques sont appliqués ne seront pas à modéliser directement dans ce logiciel (bien trop complexe à développer), mais dans un autre, comme Blender qui est gratuit est très complet. Il faudra alors les exporter au format *.obj qui est relativement facile à lire (c.f. cet [exemple](#)). Il faudra alors les importer dans le logiciel à l'aide de l'interface graphique.

Paramétrage

Il faut maintenant paramétrer les objets et la scène pour que les calculs physiques puissent être réalisés. Chaque objet peut recevoir certaines propriétés (C.F. [liste des propriétés des matériaux](#)). Seuls les paramètres nécessaires aux calculs sont nécessaires. Si des paramètres ne sont pas connus, ils pourront être trouvés dans une base de données (comme dans le logiciel CES Selector). La scène pourra de même être paramétrée si l'environnement doit être pris en compte pour les calculs.

Modèles scientifiques

Les modèles scientifiques les plus connus seront pré rentrés comme :

- mécanique classique
- la déformation des matériaux
- la mécanique des fluides

Cependant, il est possible de rentrer manuellement les équations reliant ces paramètres entre eux à l'aide d'opérateurs mathématiques.

Les calculs peuvent se faire par méthode des éléments finis.

Évolutions

Avec ce projet, il est possible de "*livrer tôt la valeur métier*" :

- Le rendu graphique 3D n'est pas obligatoire dans un premier temps, le projet contient déjà une interface graphique pour le paramétrage.
- Les paramètres physiques à implémenter peuvent être très succinct au début (en implémenter seulement quelque uns)
- La base de donnée des matériaux n'est pas obligatoire

Une application qui réalise l'exécution et la compilation en java (IDE)

objectif:

l'objectif de ce projet est de créer un éditeur java capable d'exécuter des programmes java en s'inspirant des fonctionnalités disponible sur eclipse

résumé :

l'éditeur offre les fonctionnalités suivantes:

- un éditeur de texte qui comportent les fonctionnalités suivantes :
 - Copier-coller ;
 - Ouverture simultanée de plusieurs fichiers ;
 - Fonctionnalités avancées de recherche et remplacement de texte;
 - Interaction avec des programmes externes sur les fichiers;
 - coloration syntaxique;
- une fenêtre UML pour créer des diagrammes de d'analyse en utilisant des outils nécessaires (tableau, flèches,).
- exécuter un programme java
- une rubrique d'affichage des erreurs st des messages (console)
- des outils pour créer un nouveau projet, une nouvelle classe
- Un outil de création d'interface graphique

conclusion:

Le projet semble un peu difficile mais on va essayer de créer et de développer le maximum possibles des fonctionnalités que propose Eclipse.