printemps des sciences

Avec le soutien de la Ministre de l'Enseignement supérieur et de la Recherche scientifique

Terre à terres

10 - 16 mars 2008

ULB

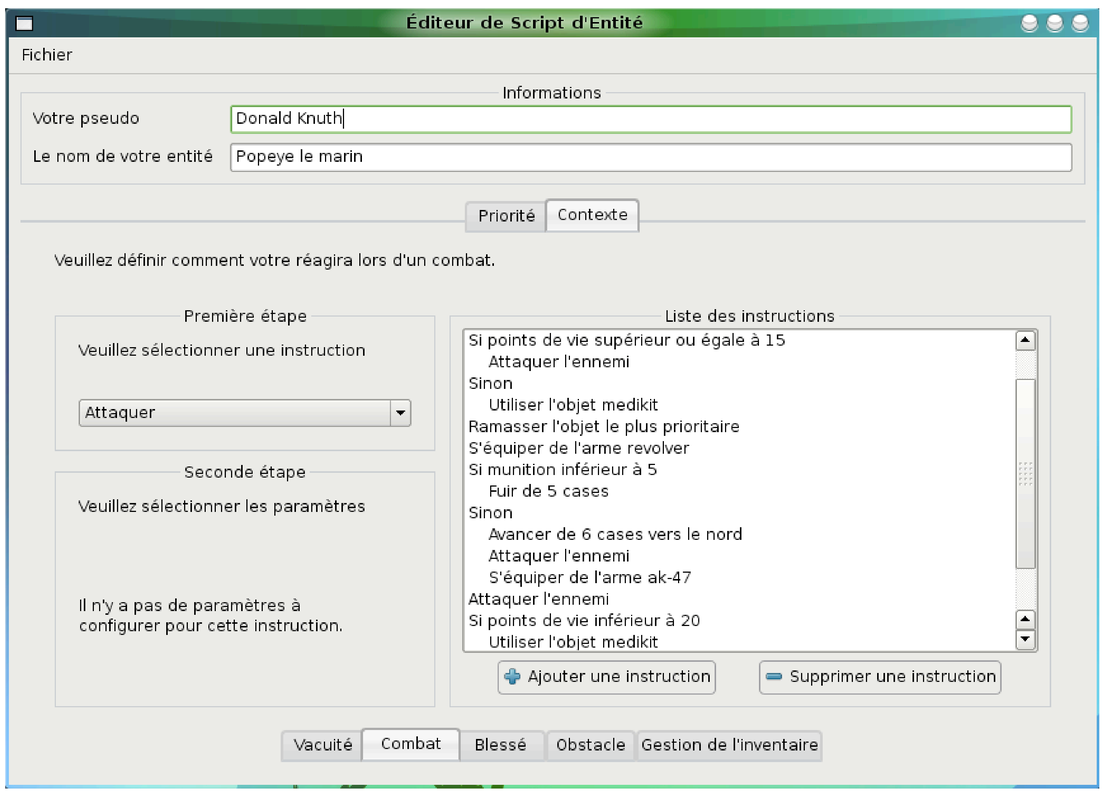
ENTITY PROJECT

DUCHENE Mathieu, PAOLILLO Antonio, SPUTAEL Olivier, SVOBODA Vladimir
Département d'informatique

Entity est le nom de code de notre projet.

Ce projet consiste en la création d'un jeu très spécial. En effet, les « avatars », ou « entités », des joueurs ne sont pas contrôlés par ceux-ci en temps réel. Au contraire, ils sont programmés à l'avance dans un mini-langage très simple qui est interprété par le jeu.

Ce langage comporte quelques instructions distinctes. Chacune d'elles permet aux entités de réaliser des actions basiques. En exploitant ces instructions selon les différents événements possibles du monde physique du jeu, les entités semblent réagir intelligemment, de façon totalement autonome.



Principe du jeu

Le projet est composé de 2 programmes : l'éditeur et le jeu

- **L'éditeur**

L'éditeur permet de définir le comportement de l'entité dans un script. Vous pouvez ajouter, modifier, supprimer des instructions qui définiront les actions à entreprendre par votre entité en fonction du contexte dans lequel elle se trouve (combat, attaqué, face à un obstacle...)
L'éditeur génère un fichier où les instructions sont traduites en un langage informatique (XML).

- **Le jeu**

Le jeu permet la connexion à un serveur où sont connectés d'autres joueurs. Chaque joueur peut envoyer plusieurs entités. Une fois la partie lancée, ils peuvent regarder comment leurs entités se comportent et combattent, ce qui leur permettra d'améliorer leur script pour les parties de jeu suivantes. Le joueur dont l'entité est la dernière présente sur la carte est déclaré vainqueur.

Le jeu se déroule au tour par tour: chaque entité effectue une action par tour.

printemps des sciences

Avec le soutien de la Ministre de l'Enseignement supérieur et de la Recherche scientifique

Terre à terres

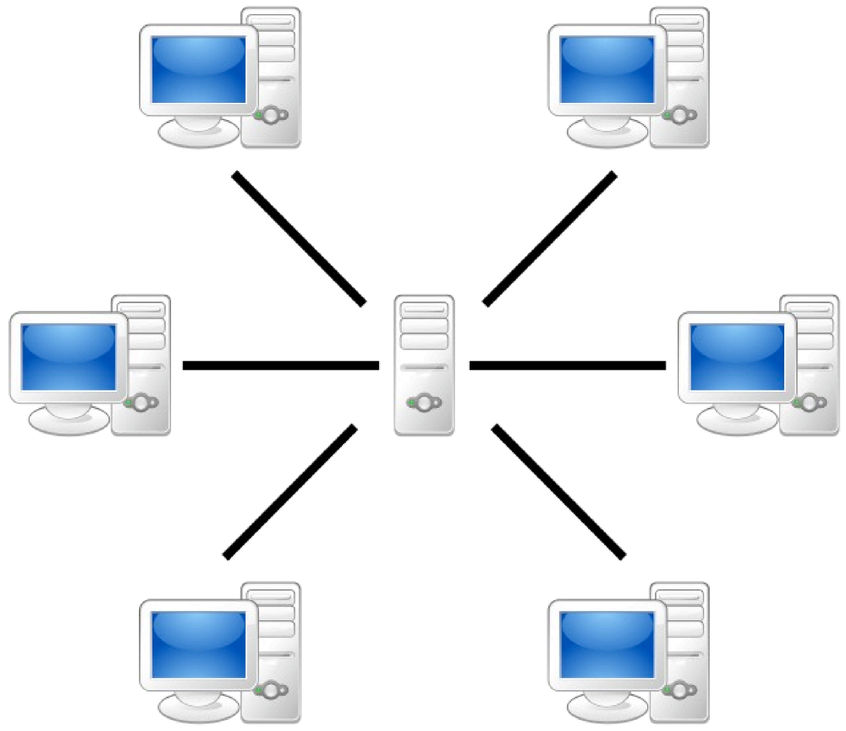
10 - 16 mars 2008

ENTITY PROJECT

DUCHENE Mathieu, PAOLILLO Antonio, SPUTAEL Olivier, SVOBODA Vladimir
Département d'informatique

Gestion réseau

Le serveur gère le jeu en calculant à chaque tour les actions des entités.
A chaque tour de jeu, le serveur demande à l'ordinateur du joueur qui a envoyé l'entité, de lui donner l'instruction qu'il désire effectuer en fonction de ce que l'entité perçoit du monde.
L'ordinateur du joueur (le client) ne fait que répondre aux demandes du serveur et afficher l'état du monde que lui envoie le serveur.
Un "chat" a aussi été implémenté, il permet d'envoyer des messages à tous les autres joueurs via le serveur.



Portabilité

Notre programme a pour contrainte de fonctionner sur 3 types de machines différentes : PC, Mac et Sun équipées respectivement des système d'exploitation suivant : Ubuntu Linux, MacOS X, et Solaris. Ces systèmes d'exploitation ont pour point commun d'être basés sur UNIX.

Langages et dépendances

Ce projet est écrit en C++
Pour l'interface graphique du jeu, nous avons utilisé SDL.
L'éditeur est réalisé à l'aide de Qt.
Les scripts sont codés en XML. L'analyseur de scripts et l'éditeur utilise la librairie TinyXml.

Equipe et répartition du travail

Nous avons séparé notre travail en modules pour optimiser l'avancement du projet :

Duchene Mathieu : responsable de la documentation relative au projet (Software Requirements Document) et tests unitaires.

Paolillo Antonio : responsable de la programmation système, réseau (UNIX) et du moteur du jeu.

Sputael Olivier : responsable de l'interface graphique du jeu (SDL).

Svoboda Vladimir : responsable de l'analyse et de la création de script (XML) et de l'éditeur (Qt).

Ensuite, nous avons dû réunir nos différents modules pour les faire s'imbriquer ensemble dans toute l'application.

ENTITY PROJECT

DUCHENE Mathieu, PAOLILLO Antonio, SPUTAEL Olivier, SVOBODA Vladimir
Département d'informatique

Programmation

C'est quoi?

La programmation permet d'écrire ce qu'on veut que notre programme fasse, dans un langage que la machine comprendra.

Voici un exemple de code écrit en méta-langage:

Si j'ai froid
Alors
    Si j'ai une écharpe ET des gants
    Alors
        Je mets mon écharpe et mes gants
    Sinon
        Je rentre chez moi chercher une écharpe OU des gants
Sinon
    Je profite de ne pas avoir froid

Dans notre cas, nous avons conçu un langage simple pour coder les instructions des entités. Nous avons aussi dû créer un analyseur de script pour que notre programme client puisse analyser ce script et traduire les instructions en actions.

Exemple d'instruction simple:

Avancer de 2 cases vers le nord

Exemple d'instruction conditionnelle:

Si mon capital de points de vie est inférieur ou égal à 15
    Fuir de 5 cases
Sinon
    Attaquer l'ennemi

Si le nombre points de vie est inférieur ou égal à 15, l'entité aura pour mission de Fuir de 5 cases; dans le cas contraire elle attaquera l'ennemi.