

**ALMA MATER STUDIORUM – UNIVERSITA' DI
BOLOGNA**

SECONDA FACOLTA' DI INGEGNERIA

Corso di Laurea Specialistica in ingegneria dei Sistemi e
delle Tecnologie dell'Informazione

QUAKE AGENT

di

Rosaldi Marco

Manuale di installazione

Sommario

1-Introduzione	Pag.2
2-Obiettivi	Pag.2
3-Analisi	Pag.2
4-Progetto	Pag.3
5-Simulazioni	Pag.8
6-Conclusioni	Pag.8

Sistemi Intelligenti Distribuiti LS
A.A. 2004/05

1. Introduzione

Questo progetto rappresenta la conclusione del percorso didattico effettuato durante il corso di Sistemi intelligenti distribuiti LS.

Il sistema proposto è stato denominato Quake Agent, ed è liberamente ispirato al famoso gioco per PC.

2. Obiettivi

L'obiettivo di questa esercitazione è quello di creare un sistema ad Agenti capace di simulare una "battaglia virtuale" tra alcuni personaggi introdotti all'interno di un labirinto, denominato Dungeon.

Gli agenti/personaggi avranno caratteristiche "fisiche" e "intellettive" differenti tra di loro, che permetteranno ad essi di sopravvivere più o meno facilmente all'interno del labirinto.

Alla fine della simulazione solo uno tra gli agenti dovrà sopravvivere al combattimento.

3. Analisi

Quake è un gioco tanto famoso quanto semplice nella sua idea di base: alcuni personaggi giocanti si muovono all'interno di un labirinto e il loro scopo è quello di sopravvivere alle insidie del Dungeon portate sia dagli altri Players che da eventuali trappole situate lungo la mappa.

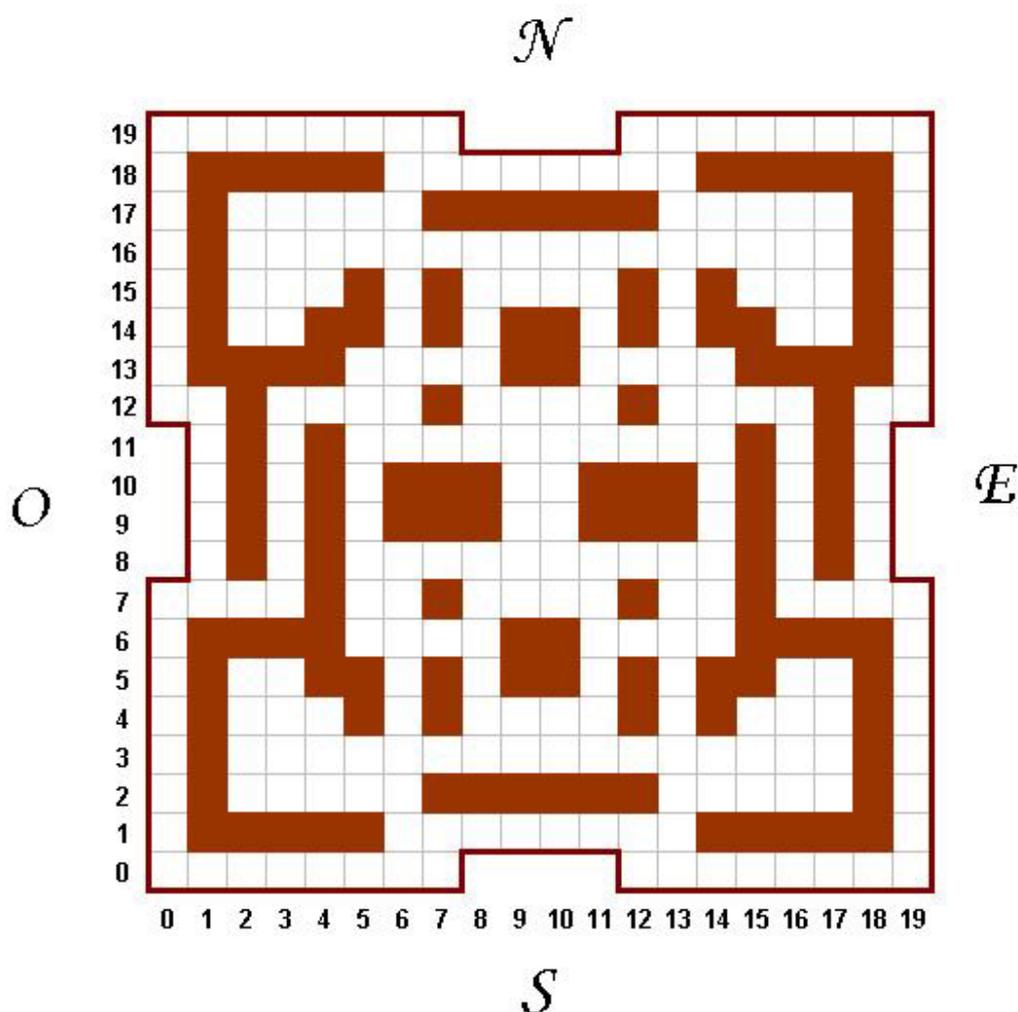
In prima analisi emergono subito le problematiche principali riguardanti la possibile realizzazione del sistema, ovvero quelle relative all'astrazione della mappa di gioco (un labirinto bidimensionale all'interno del quale individuare personaggi, muri e tutto il resto tramite coordinate cartesiane) e alla coordinazione/simultaneità degli agenti presenti nel sistema.

Tutte le considerazioni effettuate durante la fase di progetto dovranno prendere in esame un importante requisito: il sistema va realizzato utilizzando degli Agenti che sappiano coordinarsi tra di loro: per questo risulterà fondamentale l'utilizzo di Tucson come infrastruttura alla base del progetto e di tutto ciò che ne consegue (tuple, spazi di tuple, Respect Language, ecc..).

4. Progetto

4.1 Il Dungeon

Come prima fase della progettazione è stata presa in considerazione la problematica della mappa (o Dungeon), sulla quale si dovranno muovere tutti i personaggi/agenti che prendono parte al gioco. Il risultato finale di tale sviluppo, per il quale si è tenuto conto di molte variabili quali la facilità di astrazione e il movimento degli agenti all'interno della mappa stessa, è risultato il seguente:



Il labirinto si presenta come un quadrato di dimensioni 20x20, con 4 pseudo-stanze e numerosi corridoi; ogni posizione della mappa è individuabile attraverso delle coordinate cartesiane del

tipo (X, Y) . In figura sono indicati anche i punti cardinali, che risulteranno poi utili per segnalare le direzioni prese dai personaggi all'interno del Dungeon.

Si è quindi deciso di rappresentare i muri (per la precisione ciascun quadretto di posizione (x, y) di colore marrone in figura) come delle tuple del tipo:

- *'Wall'* (X, Y) ;

che risulteranno in totale 144 più una indicante le dimensioni della mappa, del tipo:

- *'Dungeon'* (X, Y) ;

4.2 Gli Agenti

Come prima cosa è indispensabile inserire tutte le tuple all'interno dello spazio di tuple relativo alle informazioni sul Dungeon, che è stato denominato *hell*; tale compito sarà svolto dall'unico agente non giocante che si occuperà in pratica solo di preparare il "campo di gioco".

I personaggi/agenti del nostro sistema dovranno avere caratteristiche di vario tipo, per diversificare il loro comportamento all'interno di esso. A questo proposito si è pensato a 4 giocatori di "intelligenza" e "capacità fisiche" di questo tipo:



IL GOBLIN

È l'elemento più vulnerabile di tutti. Possiede 100 Life-points (Punti-Vita) che stanno ad indicare la forza fisica di ciascun personaggio. Inoltre non possiede l'intelligenza necessarie per rilevare le trappole all'interno del Dungeon, che gli infliggeranno quindi, se innescate, numerosi danni fisici.



IL TROLL

In quanto a capacità fisiche è molto più forte del Goblin e possiede infatti 200 Punti-Vita, ma anche lui non ha l'intelligenza per disinnescare le trappole.



IL RANGER

La sua prestanza fisica sta a metà tra il Troll e il Goblin, avendo a disposizione 150 Punti-Vita. Oltre a questo però ha la capacità di rilevare e disinnescare il 50% delle trappole all'interno del labirinto.



IL PALADINO

E' il personaggio più potente e intelligente di tutto il gioco. I suoi Life-Points sono pari a quelli del Troll, quindi 200, ma in più ha la capacità di disinnescare qualsiasi trappola presente all'interno del Dungeon, non subendone così alcun effetto.

Ciascun personaggio comincia la sua battaglia partendo da una delle stanze del labirinto e da lì comincia a muoversi all'interno di esso. Tutte le posizioni successive saranno indicate all'interno dello spazio di tuple relativo, denominato *arena*, in questo modo:

- '*NomePersonaggio*'(*X*, *Y*, *LP*)

dove X e Y stanno ad indicare le coordinate attuali e LP il numero di Punti-Vita rimanenti.

4.3 Le trappole e i punti di partenza

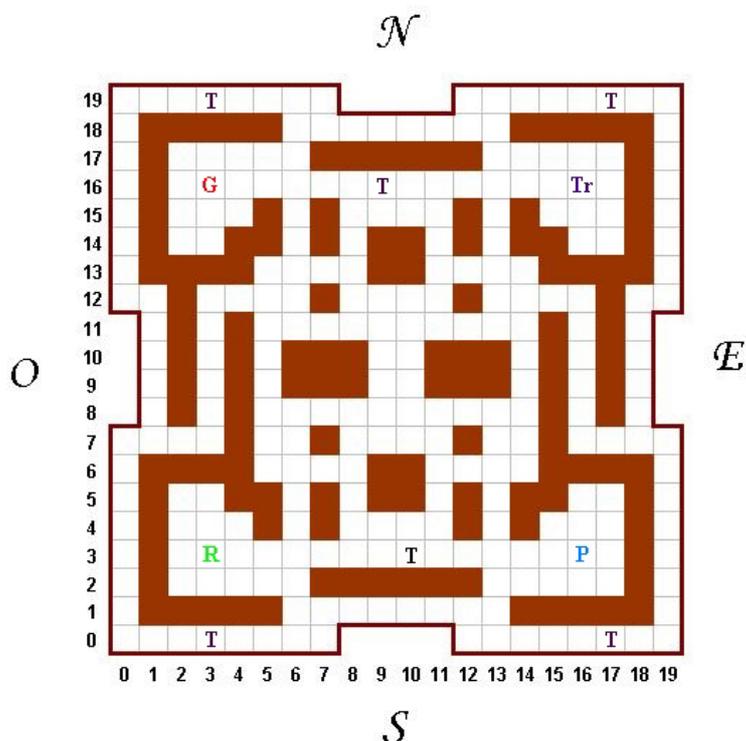
Le trappole sono posizionate in punti specifici della mappa e in totale sono 6: scattano ogni qual volta un Player vi passa sopra (sempre che esso non sia in grado di disinnescarle). Ognuna di esse può infliggere al malcapitato di turno da 4 fino a 10 Lifepoints.

Se un giocatore cade in trappola, l'evento verrà segnalato tramite una tupla all'interno dello spazio di tuple arena, in questo modo:

- *'NomePersonaggio' on trap'(X, Y, 'Damage: 4')*

dove oltre alla solita informazione sulla posizione è presente anche il danno inflitto dalla trappola.

Nella seguente figura possiamo vedere i punti di partenza dei Personaggi/Agenti (indicati con l'iniziale del nome) e le trappole (segnalate con la lettera T):



4.4 I combattimenti

Due personaggi entreranno in combattimento ogni qualvolta si ritroveranno insieme all'interno della stessa casella. L'esito dello scontro sarà deciso in maniera molto semplice dai Punti-Vita residui di ciascuno di essi: chi ha il maggior numero di Life-Points uccide il nemico, ma a sua volta subisce un danno pari ai LP dell'avversario meno uno. Nel caso si incontrino due giocatori con un numero identico di LP, vince lo scontro chi ha attaccato per primo, ovvero colui che ha fatto l'ultimo passo prima che i due fossero all'interno di una stessa casella.

I combattimenti sono come sempre segnalati all'interno dello spazio di tuple *arena* tramite la seguente tupla:

- '*NomePersonaggioA attacks NomePersonaggioB*'

Inoltre, l'esito dello scontro si potrà facilmente intuire da un'ulteriore tupla rappresentante il Player uscito sconfitto, in questo modo:

- '*NomePersonaggio is dead*'(*X, Y, LP*)

4.5 Esito finale della battaglia

La partita non terminerà fino a quando non rimarrà un solo giocatore all'interno del Dungeon; a quel punto il vincitore verrà segnalato come sempre in *arena* tramite la tupla:

- '*NomePersonaggio wins*'

In questo modo il vincitore verrà può terminare i suoi movimenti, in quanto non ci sono più avversari da sconfiggere. Questa segnalazione è effettuata tramite la tupla:

- '*All_dead*'(*1*)

4.6 Il Linguaggio Respect

Sia le trappole che i combattimenti sono stati simulati tramite il linguaggio Respect, con il quale è stato possibile programmare lo spazio di tuple *arena* in modo appropriato per implementare

la coordinazione degli agenti. Anche lo spazio di tuple *hell* reagisce in modo predefinito ad alcuni eventi specifici, quali l'inserimento di muri al di fuori del Dungeon o di dimensioni negative dello stesso, al fine di controllare che tutte le tuple siano state inserite in maniera corretta.

5. Simulazioni

Per rendere comprensibile lo svolgimento delle simulazioni, è stato inserito un cosiddetto "ritardo di passo" (personalizzabile) ad ogni Agente/personaggio: in questo modo, in pratica, tra un movimento e l'altro di ogni giocatore trascorrono circa due secondi.

Sono state effettuate numerose simulazioni di battaglie virtuali (con ritardo di passo molto inferiore ai 2 secondi, per ovvi motivi di tempo, al solo scopo di verificarne l'esito) e come facilmente pronosticabile si è notato come il Paladino ha la meglio sugli avversari in almeno il 75% delle battaglie.

5. Conclusioni e future implementazioni

In definitiva si può affermare che è stato raggiunto a pieno lo scopo di questa esercitazione, ovvero quello di realizzare un sistema multi agente e di prendere confidenza con alcune tecnologie, quali Tucson. E' anche vero però che diversi aspetti, principalmente per problemi di tempo, sono stati trascurati, come per esempio una visualizzazione grafica della simulazione che può essere presa in considerazione, assieme ad altri miglioramenti, in un eventuale sviluppo futuro del progetto.