

Progetto di Laboratorio di Programmazione di Rete Corso di Laurea in Informatica Applicata Anno Accademico 2004/2005

1 Scopo del Progetto

Un gioco multiplayer è un'applicazione distribuita in cui un insieme di giocatori, collegati ad una rete *WAN* o *LAN* mediante hosts remoti, condividono un *mondo virtuale*, interagendo tra di loro in modo diretto (ad esempio sparandosi a vicenda come nei giochi *FPS*, *first person shooters*) oppure indiretto, ad esempio manipolando oggetti passivi del mondo condiviso (ad esempio armi, pozioni, come nei MMRG, *massive multiplayer role games*).

Scopo del progetto è quello di realizzare *Survive*, un semplice gioco multiplayer. In *Survive* il mondo virtuale è rappresentato da una griglia bidimensionale in cui i giocatori, rappresentati mediante *avatars*, possono muoversi nelle quattro direzioni destra, sinistra, alto, basso, oppure fermarsi e rimanere fermi per un intervallo di tempo arbitrario. Il mondo virtuale condiviso è popolato da un certo numero di *oggetti*, che i giocatori possono raccogliere spostandosi nella stessa posizione occupata sulla griglia dall'oggetto. Ogni giocatore è caratterizzato da un certo *livello di energia*. Ogni oggetto è caratterizzato da un valore intero, che indica l'energia che può fornire al giocatore che lo raccoglie. Il livello di energia di ogni giocatore viene *incrementato* o *decrementato* durante il gioco, secondo le seguenti regole:

- ad ogni giocatore viene attribuito un livello *IE* di energia, all'inizio del gioco, uguale per tutti i giocatori.
- ogni giocatore può *incrementare* il suo livello di energia raccogliendo gli oggetti del mondo.
- il livello di energia di un giocatore viene *decrementato* col passare del tempo, ad esempio ogni t millisecondi, dove t è un parametro del gioco.

Quando un giocatore raggiunge il livello 0 di energia muore e viene eliminato dal gioco. È possibile iniziare una partita di *Survive* solo quando sono presenti esattamente k giocatori, dove k è un parametro del gioco. Il vincitore risulta l'ultimo giocatore rimasto in gioco.

È previsto che la mappa del gioco venga suddivisa in un certo numero di *zone*. Supporre che ogni giocatore abbia visibilità solo dei giocatori presenti nella zona in cui si trova (ad esempio questo può corrispondere a zone separate da muri). Per semplicità supporre che un giocatore possa passare da una zona ad una zona adiacente in un qualsiasi punto della linea di confine.

2 Architettura del Sistema

Si deve realizzare una applicazione distribuita che implementi *Survive*, secondo una architettura *Peer to Peer*. Lo stato del gioco deve essere gestito in modo *completamente distribuito*. Questo implica che ogni peer(host) possiede una *copia replicata* di una *parte del mondo virtuale*, copia che aggiorna a seconda degli eventi eseguiti dal giocatore locale e dei messaggi che gli notificano gli eventi generati dagli altri giocatori.

E' richiesta la realizzazione di un'applicazione *PeerSurvive* che possa essere utilizzata da ogni giocatore per partecipare ad una partita di *Survive*. *PeerSurvive* deve quindi eseguire un *ciclo infinito* in cui gestisce gli eventuali eventi generati dal giocatore locale, li spedisce agli altri giocatori, riceve gli eventi generati dagli altri giocatori e visualizza il nuovo stato del gioco, in base alle informazioni ricavate nei passi precedenti.

E' richiesta anche la presenza di un unico server centralizzato, *ServerSurvive*, a cui sono attribuite *solamente* compiti di *Amministrazione* del gioco, come i seguenti:

- riceve le richieste di connessione al gioco da parte dei giocatori
- quando ha ricevuto esattamente k richieste, invia ai giocatori la posizione degli oggetti all'interno della mappa, quindi un messaggio con cui notifica ad ogni giocatore che può iniziare il gioco
- riceve periodicamente il livello di energia dei giocatori
- invia ai giocatori che ne fanno richieste la lista dei giocatori che partecipano al gioco, con la loro energia
- se un unico giocatore è rimasto in gioco, gli notifica che ha vinto

Si tenga presente che la realizzazione di questo tipo di applicazioni comporta la risoluzione di alcuni problemi, tipici dei sistemi distribuiti:

- *consistenza*: occorre garantire che tutti i giocatori posseggano uno stato consistente del gioco. Ad esempio occorre evitare che un cibo sia raccolto contemporaneamente da due giocatori. Questo porterebbe ad uno stato inconsistente del gioco, poichè entrambi vedrebbero aumentata la propria energia.
- *simultaneità*: tutti i giocatori dovrebbero percepire lo stesso stato del gioco *nello stesso istante*. Ad esempio se un giocatore *G1* percepisce le mosse di *G2* con molto ritardo, mentre *G2* percepisce le mosse di *G1* tempestivamente, *G1* può risultare svantaggiato, ad esempio potrebbe decidere di dirigersi verso un oggetto che è stato già raccolto da *G2*. Ovviamente questa condizione può solamente essere approssimata in una situazione reale.

I problemi precedenti possono essere affrontati utilizzando tecniche diverse, di complessità crescente. Ad esempio la tecnica del *local lag*, vista a lezione viene utilizzata per garantire la simultaneità. Nella valutazione finale del progetto verrà ovviamente tenuta in considerazione la complessità della soluzione proposta.

3 Implementazione dal Sistema: Vincoli e Scelte di Progetto

La soluzione implementata deve soddisfare inoltre i seguenti vincoli:

- utilizzare comunicazioni di tipo *multicast* tra i giocatori.
- utilizzare il protocollo TCP/IP e/o RMI per le comunicazioni tra giocatori e server.
- strutturare sia *ServerSurvive* che *PeerSurvive* mediante più threads di controllo
- tutte le eccezioni che si possono verificare a tempo di esecuzione vanno esplicitamente gestite. In ogni caso occorre intercettare l'eccezione più specifica e gestirla nel modo più idoneo.

4 Modalità di svolgimento del Progetto

Il progetto può essere svolto in gruppo. Ogni gruppo deve essere composto al massimo da *tre studenti*. Il materiale consegnato deve comprendere:

- Una stampa di tutto il codice dello strumento e di eventuale programmi utilizzati per il test delle funzionalità dello strumento
- Una relazione che descriva tutte le scelte effettuate. L'organizzazione e la chiarezza dell'esposizione della relazione influiranno sul voto finale dell'esame

Il progetto deve essere consegnato via e-mail una settimana prima della data in cui si intende sostenere l'orale. (Gli appelli sono fissati per appuntamento, via e-mail).