

**LPR - Corso A**  
**Corso di Laurea Triennale in Informatica**  
**Anno Accademico 2005/2006**  
**Progetto**

## **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 l'un l'altro come nei giochi *FPS*, *first person shooters*) oppure indiretto, (ad esempio manipolando oggetti passivi del mondo condiviso come armi, pozioni, come nei MMRG, *massive multiplayer role games*).

Scopo del progetto è quello di realizzare *Cops And Robbers (Guardie e Ladri)* un semplice gioco multiplayer. In *Cops And Robbers* il mondo virtuale è popolato da un certo numero *Ladri* e da un insieme di *Guardie*. Ogni giocatore controlla un singolo ladro, e lo può muovere nelle quattro direzioni destra, sinistra, alto, basso, oppure fermarlo per un intervallo di tempo arbitrario. Nel mondo virtuale si muovono inoltre alcune *Guardie*, dotate di *intelligenza artificiale* il cui movimento (anche esso nelle quattro direzioni) è determinato in base ad un semplice algoritmo. Ad ogni guardia ed ad ogni giocatore è associato un *avatar* diverso, che lo identifica univocamente nel gioco. La mappa di gioco è divisa in un certo numero di stanze, divise da porte. I ladri e le guardie possono spostarsi da una stanza ad una adiacente mediante la porta di accesso. Quando un giocatore passa da in un'altra stanza, si ferma per un breve intervallo di tempo davanti alla porta di ingresso per cercare la chiave ed aprirla. In ogni stanza sono presenti un insieme di *cibi* ed un insieme di *gioielli*, che i giocatori possono raccogliere passando sulla posizione occupata sulla mappa dall'oggetto. Ogni ladro è caratterizzato da un *livello di energia*. Ogni cibo è caratterizzato da un valore intero, che indica l'energia che può fornire al giocatore che lo raccoglie. Il livello di energia di ogni ladro viene *incrementato* o *decrementato* durante il gioco, secondo le seguenti regole:

- ad ogni ladro viene attribuito un livello *IE* di energia, all'inizio del gioco, uguale per tutti i ladri.
- ogni ladro può *incrementare* il suo livello di energia raccogliendo cibo.
- il livello di energia di un ladro viene *decrementato* se e quando si muove.

Lo spostamento di un ladro è determinato dal suo livello di energia. Se un ladro raggiunge il livello 0 di energia muore e viene eliminato dal gioco.

Il comportamento di ogni guardia può essere determinato mediante il seguente algoritmo. Ogni guardia controlla se vi sono ladri nelle vicinanze. Se esiste un

ladro a distanza  $< \delta$ , lo cattura e lo elimina dal gioco. Altrimenti se esiste un ladro a distanza  $< \gamma$ , con  $\gamma > \delta$ , ( $\delta$  e  $\gamma$  sono parametri del gioco) insegue il ladro. Se, in uno dei due casi precedenti, vi sono più ladri che soddisfano una delle condizioni, viene scelto casualmente un ladro ed eseguito il movimento associato. Altrimenti, la guardia effettua un movimento generato casualmente. Anche le guardie sono caratterizzate da un livello di energia, che viene loro attribuito all'inizio del gioco. L'energia di un guardia si incrementa ogni volta che essa cattura un ladro e si decrementa quando essa si muove.

E' possibile iniziare una partita di *Cops And Robbers* solo quando sono presenti esattamente  $k$  giocatori, dove  $k$  è un parametro del gioco. *Cops And Robbers* è un gioco di squadra. Vince la squadra delle guardie se esse riescono a catturare tutti i ladri prima che questi ultimi raccolgano tutti i gioielli. In caso contrario vince la squadra dei ladri.

## 2 Architettura del Sistema

Si deve realizzare una applicazione distribuita che implementi *Cops And Robbers*, secondo una architettura *Peer to Peer* dove ogni giocatore utilizza una applicazione *Peer Cops And Robbers* per partecipare ad una partita. Lo stato del gioco deve essere gestito in modo *completamente distribuito*. Questo implica che ogni peer (host) possieda una *copia replicata* di una *parte del mondo virtuale*, copia che aggiorna a seconda degli eventi generati dal giocatore locale e dei messaggi che notificano gli eventi generati dagli altri giocatori. La simulazione del comportamento delle guardie può essere assegnata ai diversi *Peer Cops And Robbers* all'inizio del gioco, oppure dinamicamente, durante la sua evoluzione. *Peer Cops And Robbers* esegue quindi un *ciclo infinito* in cui gestisce gli eventi generati dal giocatore locale (ladro), simula il comportamento delle guardie che gli sono state assegnate, spedisce agli altri giocatori le modifiche dello stato del gioco, 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, *Server Cops And Robber*, 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 (eventualmente notifica ad ogni giocatore quale guardie gli sono state assegnate). In seguito, 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
- notifica la vincita/perdita ai giocatori (ladri)

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 perchè 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.
- *ottimizzazione delle comunicazioni*: è possibile *ottimizzare* il numero di comunicazione tra i *Peer Cops And Robbers*. Ad esempio, ogni entità ha visibilità solo delle altre entità presenti nella stessa stanza e non è quindi interessata a ricevere le posizioni delle entità presenti nelle altre stanze. Ogni modifica agli oggetti passivi del mondo (oggetti e gioielli) deve essere invece notificata a tutte le entità.

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 utilizzata.

### 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 almeno una volta *sia il protocollo TCP/IP che RMI* per le comunicazioni tra giocatori e server.
- strutturare sia *PeerCopsAndRobbers* che *ServerCopsAndRobbers* mediante più threads di controllo

- tutte le eccezioni che si possono verificare devono essere esplicitamente gestite. In ogni caso, occorre intercettare l'eccezione più specifica e gestirla nel modo più appropriato.

## 4 Modalità di svolgimento del Progetto

Il progetto può essere svolto in gruppo. Ogni gruppo deve essere composto al massimo da *due 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 *in formato pdf* che descriva tutte le scelte effettuate. La relazione deve contenere
  - una descrizione generale delle scelte di progetto effettuate
  - uno schema generale dei threads attivati da *PeerCopsAndRobbers* e da *ServerCopsAndRobbers* e delle strutture dati utilizzate.
  - una descrizione delle classi definite

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 dell'orale. L'orale verterà sia sulla discussione del progetto che sul programma svolto durante il corso. *Al momento della prova orale sarà richiesto di effettuare una piccola modifica al progetto e di eseguire il progetto modificato in laboratorio.*