

# Progetto di Lab. di Programmazione di Rete

## Laurea Triennale in Informatica Applicata

### Anno Accademico 2005/2006

## 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 *MultiPacMan*, un semplice gioco multiplayer. *PacMan* costituisce un classico nel campo dei videogiochi. Dopo la sua pubblicazione avvenuta nel 1980 ne sono state rilasciate diverse versioni per la quasi totalità delle console e dei computers. Una tipica schermata di gioco è mostrata nella seguente figura:



Il giocatore deve guidare una creatura sferica di colore giallo, il *PacMan*, con lo scopo di mangiare tutte le numerose palline gialle disseminate all'interno del labirinto e, nel far questo, deve evitare di farsi catturare dai quattro fantasmi

pena la perdita immediata di una delle vite a disposizione. Per facilitare il compito al giocatore, sono presenti nel labirinto quattro pillole speciali *power pills* che rovesciano la situazione rendendo vulnerabili i fantasmi, che diverranno subito blu e faranno dietro-front, per un breve intervallo di tempo in cui, per guadagnare punti, *PacMan* può andare alla loro caccia, per mangiarli. Ma una volta fagocitati, i fantasmi tornano alla base (il rettangolo al centro dello schermo) sotto forma di un paio di occhi, per rigenerarsi ed attaccare di nuovo *PacMan*. Durante il gioco, ogni oggetto ingerito da *PacMan*, gli fornisce punti che gli permettono di ottenere una vita supplementare oppure di rimanere nella storia come miglior giocatore di sempre. Le palline gialle, le *power pills* ed i fantasmi hanno un valore associato che indica il numero di punti ottenuti da *PacMan*, nel caso ingerisca quell'oggetto. Durante il gioco può apparire al centro del labirinto una icona, che in generale rappresenta un frutto e che fornisce ulteriori punti.

Si consideri una versione semplificata di *PacMan* in cui i giocatori si possono muovere liberamente in un ambiente di gioco privo di ostacoli (non occorre rappresentare il labirinto). Un fantasma (rs. *PacMan*) cattura *PacMan*(rs. un fantasma) quando la sua distanza dalla preda risulta  $\leq \delta$ , dove  $\delta$  è un parametro del gioco. I fantasmi catturati non rientrano in gioco. I punti accumulati da *PacMan* non gli forniscono vite supplementari.

Si vuole realizzare una versione multiplayer, *MultiPacMan* che faccia riferimento alla versione semplificata di *PacMan* single-player. In *MultiPacMan* un giocatore impersona *PacMan*, gli altri giocatori i fantasmi. Il gioco avviene in più rounds. Il vincitore di *MultiPacMan* è il giocatore che raggiunge un certo numero prestabilito  $P$  di punti. Il giocatore che impersona *PacMan* ottiene punti ingerendo gli oggetti, i giocatori che interpretano i fantasmi cercano di catturare *PacMan*, perchè questo fornisce loro il diritto di impersonare *PacMan* nel prossimo round di gioco e quindi di ottenere punti. Un round del gioco termina quando *PacMan* viene catturato, oppure quando esso ha catturato tutti i fantasmi. In quest'ultimo caso il giocatore potrà interpretare di nuovo *PacMan*, nel prossimo round di gioco. Il gioco termina quando uno dei giocatori totalizza  $P$  punti.

Si strutturi l'ambiente di gioco mediante un insieme di *stanze*. In generale, ogni giocatore visualizza solo gli eventi che avvengono nella stanza in cui esso si trova (movimenti degli altri giocatori, raccolta di oggetti,...). Esiste però la possibilità di orientare le proprie scelte di gioco, visualizzando temporaneamente la situazione dei giocatori presenti nelle altre stanze. Ogni giocatore può infatti raccogliere dal terreno di gioco dei *binocoli* che permettono di avere, *per un tempo limitato*, visibilità della posizione di tutti i giocatori (anche quelli nelle altre stanze).

## 2 Architettura del Sistema

Si deve realizzare una applicazione distribuita che implementi *MultiPacMan*, secondo una architettura *Peer to Peer* dove ogni giocatore utilizza una applicazione *Peer MultiPacMan* 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 (esempio: spostamento di un giocatore, raccolta di un oggetto,...).

E' richiesta anche la presenza di un unico server centralizzato, *Server MultiPacMan*, 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, decide quale giocatore deve interpretare *PacMan*, quindi notifica ad ogni giocatore che può iniziare il gioco
- riceve la notifica della fine di un round e registra l'informazione necessaria per la prosecuzione del gioco (identificatori dei giocatori, numero di punti accumulati dal pacman, guardia che ha eventualmente catturato pacman)

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

- *consistenza*: lo stato del gioco sui diversi giocatori deve rimanere consistente con un ragionevole livello di approssimazione. Ad esempio, occorre evitare che *PacMan* raccolga una *Power Pill* e catturi un fantasma, e contemporaneamente il fantasma catturi *PacMan*.
- *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. Ovviamente, questa condizione può solamente essere approssimata in una situazione reale.
- *ottimizzazione delle comunicazioni*: è possibile *ottimizzare* il numero di comunicazione tra i *Peer MultiPacMan*. 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 deve essere invece notificata a tutte le entità.

I problemi precedenti possono essere affrontati utilizzando tecniche diverse, di complessità crescente, alcune delle quali sono state discusse a lezione. Nella valutazione finale del progetto verrà ovviamente tenuta in considerazione la complessità della soluzione utilizzata.

### 3 Implementazione dal Sistema

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 *PeerMultiPacMan* che *ServerMultiPacMan* mediante più threads di controllo e gestire accuratamente le eccezioni.

Per semplificare l'implementazione si può supporre che ad ogni round del gioco corrisponda una sessione di gioco diversa.

### 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 *PeerMultiPacMan* e da *ServerMultiPacMan* 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 prevede la semplificazione di diverse caratteristiche rispetto alla versione originale di *PacMan*. L'implementazione di alcune di queste caratteristiche verrà ovviamente valutata positivamente.

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.* La prova del corretto funzionamento del programma verrà effettuata presso il laboratorio del Polo Didattico della Spezia.