

# Progetto per il corso di Sistemi Operativi Real-Time

Nicola Rialti      Alessandro Vannuccini      Matteo Corsini

3 Luglio 2004

## 1 Descrizione del Progetto

Il programma simula un gioco (in perfetto stile anni 80), in cui una serie di invasori alieni cercano di invadere il nostro pianeta calando dall'alto e sparando temibilissimi missili. L'utente dovrà spostare la propria navicella nella parte bassa dello schermo per evitare i missili nemici, e al contempo colpire gli alieni che avanzeranno inesorabilmente. Ogni volta che un invasore è stato ucciso, ne verrà rigenerato un altro in modo che siano sempre presenti un numero di alieni pari a `MAX_ENEMIES`. Il gioco termina quando gli alieni sono riusciti a conquistare il pianeta oppure quando l'astronave ha terminato le proprie vite.

L'applicazione è formata da 5 thread concorrenti, ognuno dei quali si preoccupa di gestire un 'protagonista':

- l'astronave comandata dal giocatore
- i missili sparati dall'astronave
- il generatore di astronavi nemiche (la morte nera III)
- le astronavi nemiche
- i missili sparati dagli invasori

## 2 Le strutture condivise

I thread scrivono e leggono in 2 strutture:

1. *navicella\_table* dove sono dichiarati i semafori e i parametri di gioco
  - punteggi
  - astronavi rimaste
  - nemici attivi
  - missili sparati

- posizione dell'astronave

Per proteggere le strutture condivise viene utilizzato un semaforo per la struttura `navicella_table`, e un semaforo di mutua esclusione per ogni nemico (quindi la struttura `enemy_table[i]` sar prote to dal semaforo `mx_enemy_table[i]`).

2. *enemy\_table* In questo secondo tipo di struttura sono registrati i parametri relativi ad ogni alieno:

- posizione
- stato
- flag per segnalare l'uccisione

  formata da un array di strutture di dimensione uguale al valore del massimo numero di nemici.

### 3 I Task

All'avvio dell'applicazione viene inizializzata la struttura generale, quindi   creato il task della navicella comandata dall'utente, e il task che si preoccupa di generare gli alieni nemici.

#### 3.1 Task 'Navicella'

Il task della navicella   stato scelto hard, con `wcet` pari a  $500 \mu\text{sec}$ . (sperimentalmente consuma circa  $30 \mu\text{sec}$ ).

All'avvio del task viene inizializzato il 'listener' degli eventi della tastiera sui tasti 'z', 'x' e 'INVIO'. Al verificarsi di uno di questi tre eventi vengono richiamate le funzioni che spostano l'astronave a sinistra ('z'), a destra ('x'), oppure, alla pressione del tasto 'INVIO' viene generato un task 'Missile'.

Quindi ad ogni attivazione del task, controlla che la navicella non sia stata colpita da un missile nemico, e che non debba terminare il programma in seguito alla fine delle vite a disposizione.

#### 3.2 Task 'Generatore Nemici'

Il task generatore di nemici controlla ad ogni attivazione se il numero degli invasori   inferiore al numero massimo previsto. In caso affermativo, controlla nell'array delle strutture istanziate per ogni nemico la prima posizione libera, e l  va a scrivere le ipostazioni del nemico creato. Viene creato quindi un task per ogni nemico attivo.

### 3.3 Task 'Nemici'

Il task dei nemici si occupa di disegnare l'invasore e di spostarlo nello schermo. come prima operazione ad ogni attivazione, controlla di non essere stato colpito, quindi procede. Nel caso in cui risulta colpito da un missile sparato dalla navicella, si preoccuperà di liberare la risorsa occupata con i propri dati, e quindi di terminare il task, dopo avere comunicato il proprio 'suicidio'. In maniera casuale ogni nemico sparerà un missile contro la nostra navicella. Infine andrà a scrivere la nuova posizione nella propria struttura.

### 3.4 Task 'Missili'

Il task missile e il task missile nemico, sono sostanzialmente simili. Entrambi ad ogni ciclo si spostano di una posizione lungo l'asse delle ordinate, controllando se hanno colpito rispettivamente un nemico o la navicella. In tal caso andranno a comunicare (al nemico corrispondente o alla navicella) che al prossimo ciclo dovranno provvedere a morire.

### 3.5 Tempi di esecuzione dei Task

Questi i tempi medi dei task rilevati sperimentalmente:

- astronave 31  $\mu\text{sec}$ .
- invasore 230  $\mu\text{sec}$ .
- missile 20  $\mu\text{sec}$ .
- generatori alieni 70  $\mu\text{sec}$ .