

SAPIENZA Università di Roma  
Facoltà di Ingegneria e Facoltà di Ingegneria dell'Informazione  
Corsi di Laurea in Ingegneria Informatica ed Automatica ed in Ingegneria dei Sistemi Informatici  
**Corso di Progettazione del Software**  
Esame del **23 Giugno 2011**  
*Tempo a disposizione: 3 ore*

**Requisiti.** L'applicazione da progettare riguarda un gioco di azione in cui ci si spara dei proiettili che lasciano macchie colorate sugli indumenti dei giocatori. Una partita è caratterizzata da un codice (una stringa) e da un insieme di giocatori ciascuno con un nome ed un personaggio animato (una stringa che rappresenta il path ad un filmato). Ogni giocatore, che è associato ad esattamente una partita, ha un'arma a ripetizione, caratterizzata da un nome, dal numero di colpi disponibili e dal numero di colpi sparati contemporaneamente in una raffica. Alcune armi sono speciali ed ad esse è associato un filmato. Ogni arma appartiene al più ad un giocatore. Per ciascun giocatore è di interesse ricordare quante volte è stato colpito durante l'esecuzione della partita. Ciascun giocatore può stringere alleanze con altri giocatori ed è di interesse memorizzare tali alleanze. (Si noti che se il giocatore  $\alpha$  è alleato con il giocatore  $\beta$  allora ovviamente anche  $\beta$  è alleato ad  $\alpha$ .)

Un giocatore è inizialmente "non in gioco". Quando riceve l'evento "inizio partita" passa allo stato "in gioco". Ciascun giocatore in gioco quando riceve un raffica rappresentato da un evento "raffica" con un parametro che ne caratterizza il numero di colpi fa la seguente azione: (i) incrementa le volte che è stato colpito con il numero di colpi nella raffica, e (ii) spara una raffica della propria arma ad un altro giocatore scelto arbitrariamente tra i giocatori che non sono alleati. Quando il giocatore spara, in realtà genera un nuovo evento "raffica" con numero di colpi pari a quello della raffica della sua arma, se i colpi dell'arma sono sufficienti, o pari al numero di colpi ancora disponibili, in caso contrario. Nel fare questo aggiorna i colpi disponibili rimanenti nell'arma. Si noti che la raffica conterrà 0 colpi se non ci sono colpi rimanenti. Quando un giocatore riceve l'evento di "fine partita" il giocatore torna nello stato "non in gioco". A parte i colpi subiti, il giocatore può essere modificato solo quando è nello stato "non in gioco".

Siamo interessati a progettare un'attività principale, che prende come parametro una partita, azzera il numero di colpi subiti da ogni giocatore e inizia l'esecuzione della partita stessa, attraverso l'invio in broadcasting a tutti i giocatori dell'evento "inizio partita" e dell'ulteriore invio dell'evento "raffica" con zero colpi a tutti i giocatori. Poi si mette in attesa (attraverso un'opportuna attività di input/output) del comando di fine esecuzione da parte dell'utente, che termina la partita riportando tutti i giocatori nello stato "non in gioco". A questo punto contemporaneamente: (i) identifica l'insieme dei giocatori che sono stati colpiti il numero minore di volte e li proclama vincitori stampandone su una apposita finestra i loro nomi; (ii) stampa su un'altra finestra le alleanze tra i giocatori. Al termine di entrambe le sottoattività, viene stampato un messaggio di saluto concludendo l'attività principale.

**Domanda 1.** Basandosi sui requisiti riportati sopra, effettuare l'analisi producendo lo schema concettuale in UML per l'applicazione, comprensivo del diagramma delle classi, diagramma stati e transizioni per la classe *Giocatore*, diagramma delle attività, specifica del diagramma stati e transizioni, specifica delle attività complesse, motivando, qualora ce ne fosse bisogno, le scelte effettuate.

**Domanda 2.** Effettuare il progetto, illustrando i prodotti rilevanti di tale fase e motivando, qualora ce ne fosse bisogno, le scelte effettuate.

È obbligatorio definire solo le responsabilità sulle associazioni del diagramma delle classi.

**Domanda 3.** Effettuare la realizzazione, producendo un programma JAVA e motivando, qualora ce ne fosse bisogno, le scelte effettuate.

È obbligatorio realizzare in JAVA solo i seguenti aspetti dello schema concettuale:

- La classe *Giocatore* (con le eventuale classe *fred* corrispondente) e le *associazioni* che la legano alla classe *Arma*.
- L'*attività principale* con eventuali sottoattività, escluse le attività atomiche.