

SAPIENZA Università di Roma
Facoltà di Ingegneria dell'Informazione, Informatica e Statistica
Corsi di Laurea in Ingegneria Informatica ed Automatica ed in Ingegneria dei Sistemi Informatici
Corso di Progettazione del Software
Esame del **9 Luglio 2014**
Tempo a disposizione: 3 ore

Requisiti. L'applicazione da progettare riguarda una parte di un videogioco di combattimento in una ambientazione fantascientifica. Un gioco è costituito da diversi giocatori, ciascuno dei quali appartiene esclusivamente al gioco stesso. I giocatori, che hanno un nome (una stringa), sono divisi in 3 categorie: soldati, replicanti e alieni. Dei soldati interessa, l'abilità in combattimento (un intero compreso tra 0 e 9), il numero di colpi subiti e la resistenza, cioè il numero massimo di colpi che può subire prima di uscire dal gioco. Dei replicanti interessa il soldato di cui sono replica. I replicanti hanno una particolare capacità: possono creare copie esatte di se stessi del tutto identiche all'originale, ma con in più l'informazione del replicante che li ha generati. Degli alieni interessa il livello di energia corrente ed il livello di energia minimo per rimanere in gioco.

In questo compito siamo interessati al comportamento dei soli replicanti. Un replicante è inizialmente a *riposo*. Quando è a *riposo*, se riceve il comando di *combattere* contro un dato alieno, passa in *combattimento*. Quando è in combattimento, può ricevere il comando *colpisci*: se il soldato di cui è replica ha un livello di abilità superiore a 5, allora riduce del 10% l'energia dell'alieno; altrimenti il soldato di cui è replica subisce un colpo, e (i) se con questo colpo, si raggiunge il numero massimo di colpi, allora il replicante torna a *riposo*; (ii) altrimenti il replicante crea una replica di se stesso e lo comanda di combattere lo stesso alieno e rimane lui stesso in combattimento.

Siamo interessati alla seguente attività principale. L'attività prende in input un gioco G , verifica che ci siano almeno 2 soldati, 1 replicante e 4 alieni. In caso negativo termina, altrimenti concorrentemente esegue le seguenti due sottoattività: (i) gioca, e (ii) analisi. La sottoattività di gioco (i) avvia il gioco attivando tutti i giocatori di G mandando opportuni eventi (i dettagli non interessano). Poi si mette in attesa del comando di fine-gioco da parte dell'utente che interrompe il gioco. La sottoattività di analisi (ii) calcola e stampa per tutti i soldati di G il numero massimo di colpi e per tutti gli alieni di G il livello minimo di energia. Una volta che tali sottoattività sono state completate, si calcolano soldati e alieni fuori gioco e si stampa chi ha vinto tra soldati e alieni

Domanda 1. Basandosi sui requisiti riportati sopra, effettuare l'analisi producendo lo schema concettuale in UML per l'applicazione, comprensivo del diagramma delle classi (inclusi vincoli non esprimibili in UML), diagramma stati e transizioni per la classe *Replicante*, diagramma delle attività, specifica del diagramma stati e transizioni, e specifica dell'attività principale e delle sottoattività NON atomiche (indicando in modo esplicito quali attività atomiche sono di I/O e quali sono Task), motivando, qualora ce ne fosse bisogno, le scelte di progetto.

Domanda 2. Effettuare il progetto, illustrando i prodotti rilevanti di tale fase e motivando, qualora ce ne fosse bisogno, le scelte di progetto. 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 di progetto. È obbligatorio realizzare in JAVA solo i seguenti aspetti dello schema concettuale:

- La classe *Replicante* con la classe *ReplicanteFired*, eventuali superclassi, e le classi JAVA per rappresentare le *associazioni* di cui queste classi hanno responsabilità.
- L'*attività principale*.