

# Relazione per il progetto “The Visitors”

Cofini Riccardo

29 Febbraio 2016

# CAPITOLO 1

## ANALISI

Il software proposto ha come scopo la creazione di un arcade game rifacendosi al modello del classico Space Invaders, il cui scopo è quello di distruggere le astronavi aliene controllando a propria volta una navicella spaziale.

### Requisiti

- Il videogioco permetterà al giocatore di controllare un carro armato su un piano bidimensionale e sparare agli alieni
- Verrà gestito il movimento degli alieni verso la Terra
- Verrà gestito il conteggio del punteggio effettuato ad ogni partita
- Verrà memorizzato il record massimo ottenuto
- Il gioco terminerà quando tutti gli alieni sono stati distrutti oppure quando un alieno riuscirà a raggiungere la Terra

Il gioco parte da un menù iniziale, dove si ha la possibilità di iniziare una nuova partita o semplice uscire dall'applicazione.

All'inizio di ogni partita verrà preparato un campo di gioco dove sulla parte superiore saranno presenti varie file di alieni, mentre al centro della parte inferiore verrà collocato un carro armato.

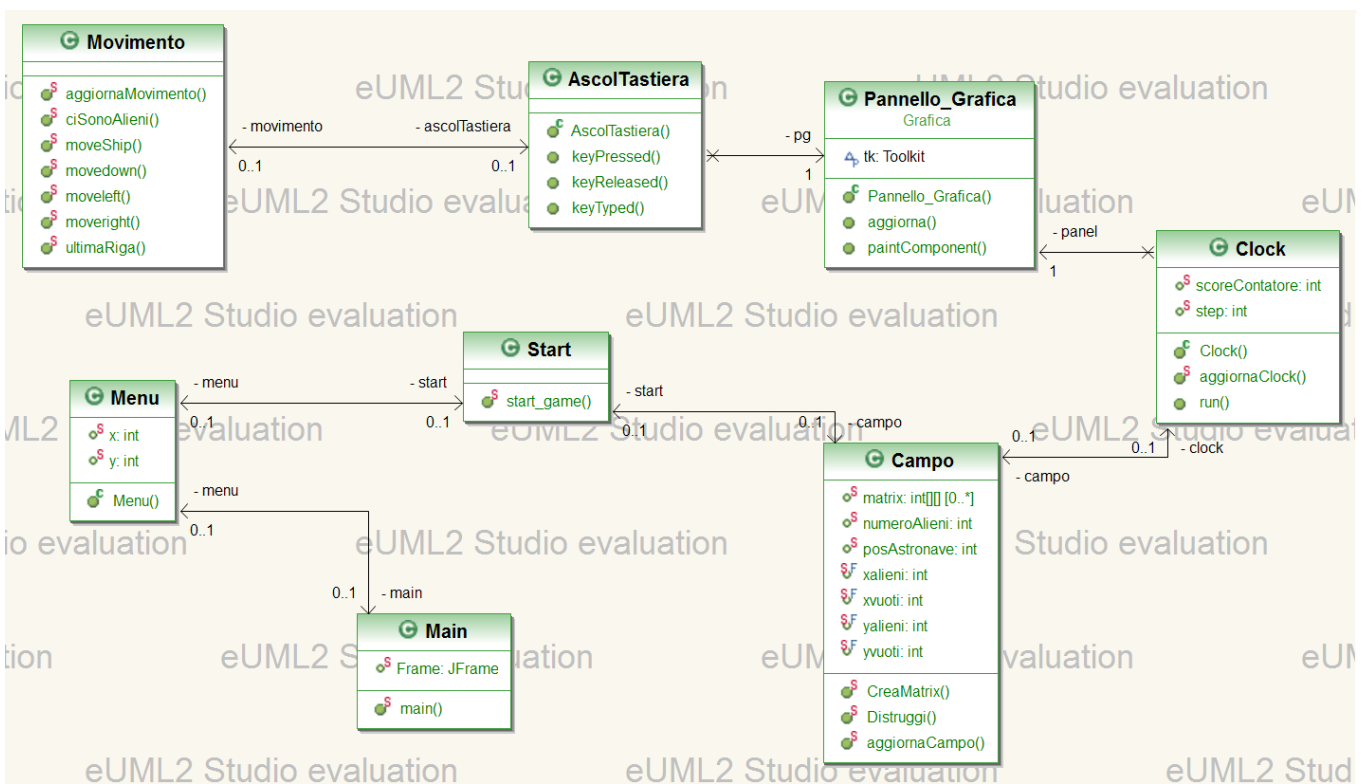
Gli alieni scenderanno verso la Terra muovendosi da destra a sinistra e da sinistra a destra, aumentato periodicamente la loro velocità.

Il giocatore ha il ruolo di controllare il carro armato, muovendolo a destra e sinistra, e sparare agli alieni per eliminarli prima che riescano a completare la loro discesa.

Il giocatore guadagnerà 5 punti per ogni alieno distrutto, ma perderà un punto per ogni proiettile sparato.

Quando il giocatore eliminerà tutti gli alieni, o quando un alieno riuscirà a raggiungere la Terra, il gioco si ferma e verrà visualizzata una schermata che, oltre a mostrare il punteggio ottenuto e il record massimo, permetterà di scegliere se si vuole iniziare una nuova partita o chiudere il gioco.

Nella figura a seguire sono riportate le relazioni tra le entità principali del problema.



# CAPITOLO 2

## ARCHITETTURA

Ho cercato di sviluppare il progetto secondo la logica MVC (Model – View - Controller), ossia tenere separata la logica del gioco racchiusa nelle classi relative ai vari elementi del progetto.

Il progetto è suddiviso in 3 package:

- Gioco
- Grafica
- Dati

Nel package **Gioco** sono contenute tutte le classi che si occupano della gestione dei vari elementi del progetto. Vengono definite le varie entità, le relazioni che devono avere tra di loro, le regole e le azioni possibili.

Nel package **Grafica** sono inserite tutte le classi che si occupano della visualizzazione grafica dei vari componenti. La GUI dovrà visualizzare lo sfondo del menù e del gioco, gli alieni e il carro armato.

Nel package **Dati** è contenuta la classe che si occupa di salvare su disco il record massimo ottenuto dal giocatore, in modo da poter essere sempre visualizzato a fine partita e essere eventualmente aggiornato.

## ARCHITETTURA – DETTAGLI

### Package **Gioco**

**Main:** classe contenente il metodo “main”. All’avvio richiama i metodi per la creazione del menù iniziale.

**Menù:** classe che crea il frame del menù principale con posizione e dimensione fissa.

**Campo:** classe che crea la griglia di gioco e dispone gli alieni e il carro armato all’avvio della partita tramite la funzione ‘CreaMatrix’. Questa funzione inizialmente crea una matrice di zeri di dimensioni fisse. Successivamente si occupa di sostituire con 1 gli zeri della matrice in corrispondenza della posizione iniziale degli alieni. Inoltre calcola la posizione iniziale del carro armato, che verrà sempre collocato al centro dell’ultima riga della matrice.

La classe contiene anche la funzione ‘Distruggi’, che si occupa di eliminare gli alieni colpiti e incrementare il contatore del punteggio.

Infine è presente la funzione ‘aggiornaCampo’, che ha il compito di resettare la matrice di gioco allo stato iniziale

**Movimento:** classe contenente le funzioni per il movimento degli alieni ‘moveleft’, ‘movedown’, ‘moveright’ e quella per il movimento del carro armato ‘moveShip’. Anche qui è presente la funzione che permette di riportare le variabili utilizzate alle condizioni iniziali, necessaria per poter far ripartire il gioco ‘aggiornaMovimento’.

**Clock:** classe che gestisce le dinamiche di gioco. Contiene il metodo run(), che viene invocato subito dopo la creazione del frame di gioco. Gestisce il movimento degli alieni per tutta la durata della partita.

**AscolTastiera:** questa classe gestisce gli ascoltatori della tastiera relativi movimento del carro armato. Contiene la funzione ‘keyReleased’, che ridefinisce il metodo della classe keyListener, associando i vari keyCode alle diverse azioni da eseguire.

**Start:** questa classe contiene il metodo per far iniziare la partita. Crea il frame di gioco e fa partire il thread di movimento degli alieni.

## Package **Grafica**

**Menu\_Grafica:** questa classe contiene i metodi per la creazione del pannello del frame di menù. Il pannello ha una immagine come sfondo e due bottoni. Il bottone ‘nuova partita’ chiama il metodo ‘start\_game()’ della classe Start e avvia il gioco, mentre il bottone ‘esci’ chiude l’applicazione.

**Campo\_Grafica:** classe contenente il metodo per la creazione del frame di gioco con posizione e dimensione fissa

**Pannello\_Grafica:** questa classe crea il pannello del frame di gioco e tramite il metodo ‘paintComponent’ permette di disegnare le immagini di sfondo, degli alieni e del carro su di esso.

**AboutDialog:** questa classe crea la JDialog di fine partita. Permette di visualizzare il punteggio ottenuto ed il record massimo. Presenta il bottone ‘esci’ che permette di chiudere l’applicazione, e il bottone ‘rigioca’, che invoca i vari metodi per ripristinare il gioco ai valori iniziali e fa partire una nuova sfida.

## Package **Dati**

**Salva\_Carica:** questa classe gestisce la sezione relativa al salvataggio del punteggio ottenuto e quella relativa al caricamento del record massimo salvato nel file “High-score.txt”. Il metodo ‘salva()’ confronta il punteggio ottenuto a fine partita con quello memorizzato nel file, che verrà sostituito nel caso si abbia fatto un nuovo record.

## CAPITOLO 3

### SVILUPPO

Ho iniziato analizzando il progetto nella sua globalità e valutando cosa andava fatto e di cosa avevo bisogno.

Il primo step è stato la creazione della matrice del campo di gioco, l'inserimento degli alieni e pensare al loro movimento. Inizialmente avevo creato un'unica funzione che facesse muovere gli alieni lungo il loro percorso. Ovviamente in un secondo momento ho dovuto rivalutare la cosa e creare le 3 funzioni per i 3 movimenti, poiché altrimenti non sarei riuscito ad aggiornare la GUI ad ogni passo.

Sistemata la questione alieni ho iniziato a creare i vari Frame per il menù e il campo di gioco, con i relativi pulsanti e ascoltatori. Non ci sono stati particolari intoppi per questa fase.

Arrivato il momento di implementare la GUI ho dovuto creare delle immagini con dimensioni ad hoc, in modo tale che si inserissero bene nella griglia di gioco che avevo preparato.

Una volta soddisfatto del risultato grafico ho creato gli ascoltatori per permettere al giocatore di muovere il carro armato e sparare agli alieni tramite la tastiera.

Infine ho implementato la classe che permette di salvare e caricare i dati dal file che viene creato sul pc dell'utente

### TESTING

Non c'è stata una vera e propria fase di testing, provavo di volta in volta le varie funzioni che andavo a creare e valutavo come si comportavano con le altre, andando a sistemare e perfezionare eventuali problemi che si presentavano lungo il percorso.

Interessante e degno di nota è il Bug che ho rinominato "Bug dell'Impiccato". In pratica in un primo momento avevo utilizzato una immagine in formato '.gif' per gli alieni, davano parecchio movimento al gioco poiché si animavano ad ogni passo. Purtroppo durante le prove ho notato che in particolari circostanze alcuni alieni dell'ultima colonna, arrivati al bordo destro del campo di gioco, invece che scendere di una riga e ripartire verso sinistra, rimanevano lì bloccati e continuavano a muovere i tentacoli rimanendo fermi sul posto, come se appunto fossero "impiccati". Stranamente anche dopo parecchio tempo passato in debug non sono riuscito a trovare una motivazione al problema, che si è risolto dopo una pulizia del codice e avendo sostituito l'immagine degli alieni con quella statica attuale.

# CAPITOLO 4

## COMMENTI FINALI

Nel complesso sono abbastanza soddisfatto del risultato finale, valutando il fatto di lavorare da solo e il tempo a disposizione.

Sicuramente ci sono parecchie cose che si potrebbero migliorare e alcune funzionalità che ho dovuto lasciare indietro causa mancanza di tempo (vedi l'animazione mancante del proiettile del carro armato), ma sicuramente presenta una buona base per una possibile futura implementazione.

Purtroppo mi sono reso conto solo durante la stesura della relazione che ho fatto un po' di confusione della suddivisione delle classi all'interno dei package e che avrei potuto sicuramente organizzare meglio il codice.

Le difficoltà più grosse sono state il capire bene come il progetto andava sviluppato e le varie modalità di consegna. Non avendo avuto la possibilità di seguire le lezioni, causa impegni lavorativi, non sono molto sicuro di aver rispettato a pieno le linee guida assegnate.

Nonostante ciò è stata una bella sfida e mi sono appassionato allo sviluppo di questo progetto, non nego che magari a tempo perso potrei un futuro rimetterci le mani e implementare\migliorare tutte quelle cose che per mancanza di tempo e conoscenze non del tutto approfondite ho dovuto lasciare indietro.