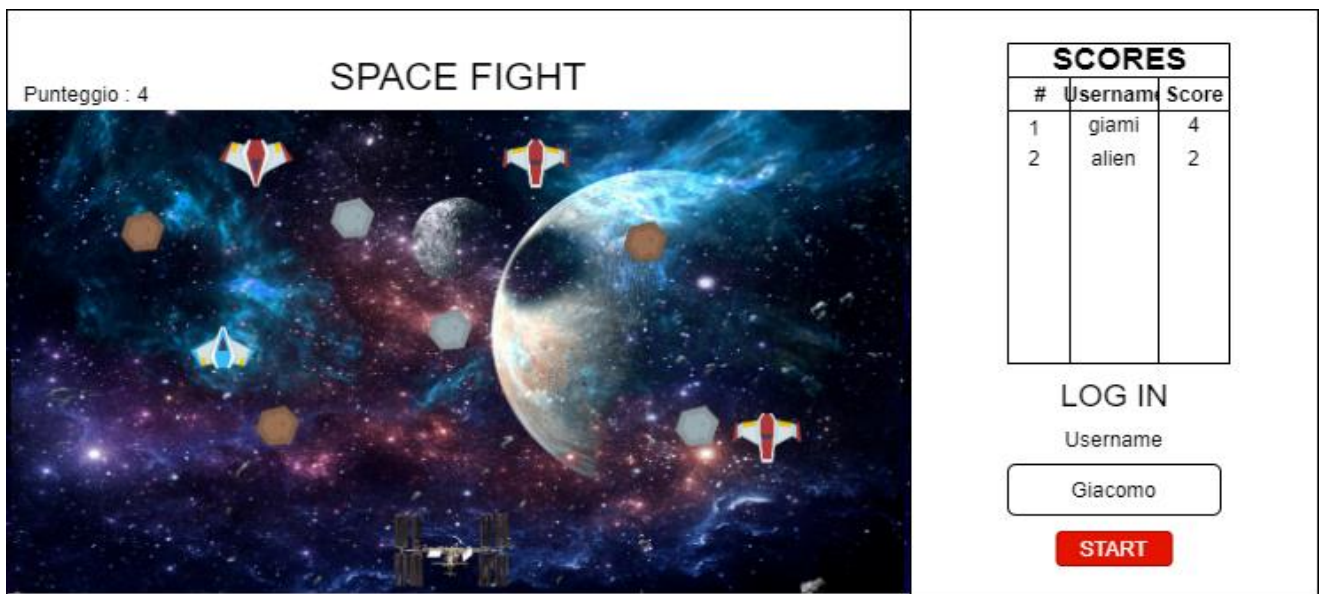


Space Fight – Documento di analisi

Piacentini Giacomo – Matricola: 627555

Artificial Intelligence and Data Engineering – Corso di programmazione avanzata 2020/2021

Vista Statica



Vista dinamica

Caso d'uso (inserimento dati giocatore e inizio nuovo gioco)

1. FOREACH Partita terminata e salvata
 - 1.1 Il sistema visualizza sotto al tasto SCORES, una tabella con il ranking dei vari giocatori: #, USERNAME, SCORE salvati in ordine decrescente
2. L'utente inserisce Username
3. IF l'utente preme PLAY
 - 3.1 Viene inizializzato il campo points a 0

- 3.2 Il sistema inizia la partita, mostrando a schermo la Navicella ship.
- 3.3 IF l'utente preme i tasti per il movimento (W,S,A,D), la navicella si sposta, se possibile, nella suddetta direzione
- 3.4 IF Il giocatore preme il tasto configurato per sparare (SPACE) e nessun altro proiettile proveniente dalla navicella è attivo
 - 3.4.1 Il Sistema fa sparare un laser dal ship nella direzione in cui esso è rivolto
- 3.5 IF il proiettile del giocatore colpisce un shipEnemy
 - 3.5.1. Il shipEnemy ed il laser del giocatore vengono eliminati dal gioco
 - 3.5.2. Il Sistema aumenta il Player Score di 1
- 3.6 IF un laser colpisce un rock
 - 3.6.1. Il laser viene eliminato dal gioco
- 3.7 IF il ship viene colpito dal shipEnemy o dal laserNemico
 - 3.7.1. Game Over
- 3.8 IF un shipEnemy colpisce con un laser la base o la tocca
 - 3.8.1. Game Over
- 3.9 IF Game Over
 - 3.9.1. Il Sistema aggiorna la tabella dei ranking con l'ultimo punteggio.
 - 3.9.2. Il Sistema blocca il gioco.
 - 3.9.3. Torna al Punto 1.

File di configurazione locale in XML

Il sistema all'avvio legge dal file di configurazione locale i seguenti dati:

- Il numero di campi da visualizzare nella tabella dei Ranking
- L'età massima (in giorni) della partita per poter essere visualizzata nella tabella dei Ranking
- L'indirizzo IP del client, l'indirizzo IP e la porta del server di log (8080)
- User e Password del Database

Cache locale degli input

1. Al termine il sistema salva su un file binario: Username, points, posizione X, posizione Y e direzione di ship e di shipEnemy, se giocoIniziato = True.

All'avvio il Sistema carica da file binario i suddetti dati.

Archivio

1. Il sistema archivia i seguenti dati:
 - Username, score, data della partita e identificatore partita relativa al punteggio per ogni partita.

File di LOG remoto in XML

Il sistema invia una riga di log al server di log per ogni evento specificato di seguito:

- Avvio dell'applicazione ("AVVIO")
- Pressione del pulsante "START"
- Termine dell'applicazione ("EXIT")

La riga di log contiene: nome dell'applicazione, indirizzo IP del client, data corrente, ora corrente, l'etichetta associata all'evento.

Space Fight – Documento di Progetto

Piacentini Giacomo – Matricola: 627555

Artificial Intelligence and Data Engineering – Corso di programmazione avanzata 2020/2021

Registration: Inizializza il campo della GUI , che gestisce l’inserimento dell’Username, gestisce inoltre l’avvio della partita tramite il pulsante START; controlla se prima all’avvio della partita non vi è alcun username inserito.

Rocks: Tutte le informazioni riguardanti i meteoriti all’interno del gioco sono memorizzate da questa classe; gestisce il posizionamento dell’elemento di gioco; gestisce la collisione tra meteoriti e navicelle amiche oppure laser.

StatoPassatoGioco: Memorizza su file binario (cache.bin) lo stato del gioco (posizioni e direzioni delle ship presenti al momento della chiusura del gioco), il campo username ed il punteggio corrente della partita; reimmette tali dati nelle relative locazioni all’avvio dell’applicativo.

DBScores: Gestisce la classifica delle partite precedenti.

EventGUI: (client) contiene le informazioni degli eventi di Log quali avvio e chiusura dell’applicativo, tasto play premuto; si serializza in XML; invia l’evento di LOG in XML alla classe LogGUI.

LogGUI: (server) riceve un evento di log XML; invoca la validazione dell’evento tramite XML schema, aggiunge la riga XML al file di log incrementalmente e la stampa in output.

ParametriConfig: legge il file di configurazione XML; invoca la validazione del file di configurazione XML; deserializza il contenuto XML come oggetto Java; mette a disposizione a ciascuna classe i parametri username e password di accesso al database, ipClient, ipServer e numero di porta del server (8080) per la comunicazione tra il server di log

e l'applicativo, i numeri di campi del ranking da visualizzare in tabella visuale.

Partita: classe bean che memorizza le informazioni di una partita; mette a disposizione alle classi DBScores e Scores tali informazioni.

Lasers: Gestisce tutte le informazioni necessarie a descrivere un laser all'interno del gioco, amico e nemico che sia. Gestisce inoltre il posizionamento di tali laser

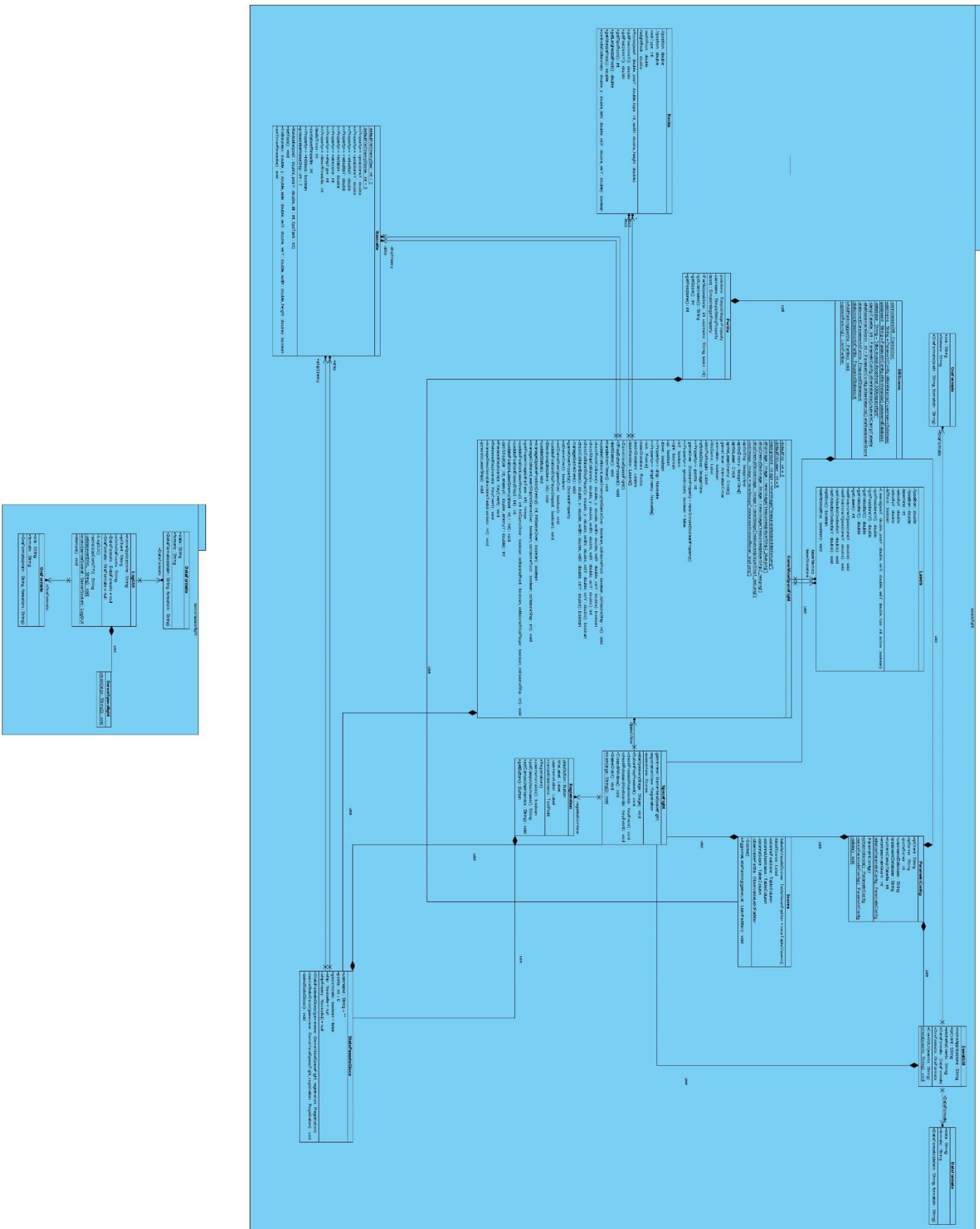
ServerLog: (server) riceve un evento di log XML; invoca la validazione dell'evento tramite XML schema, aggiunge la riga XML al file di log incrementalmente e la stampa in output.

Scores: inizializza la GUI relativa alla visuale dei punteggi nella tabella degli Scores.

SpaceFight: avvia la GUI, inizializzandola : in particolare inizializza le aree Scores, Registration, GameViewSpaceFight; effettua il demultiplexing dei vari eventi generabili dai vari elementi della GUI e dalla tastiera nell'area responsabile alla gestione; coordina memorizzazione e caricamento dei dati in StatoPassatoGioco; coordina invio eventi tramite EventoDiNavigazioneGUI, e coordina aggiornamento Scores tramite apposito evento di fine partita.

Navicelle: Memorizza ogni informazione che gestisce una navicella all'interno del gioco; gestisce posizionamento dell'elemento di gioco tramite specifici metodi setter e getter. gestisce la possibile collisione fra una navicella e un laser di gioco

GameViewSpaceFight: Area della GUI che inizializza l'area di gioco; gestisce le animazioni dei vari elementi di gioco e determina la loro presenza durante la partita gestendo; gestisce gli input della tastiera preta/rilasciata per gestire la navicella; determina eventi di gameOver.

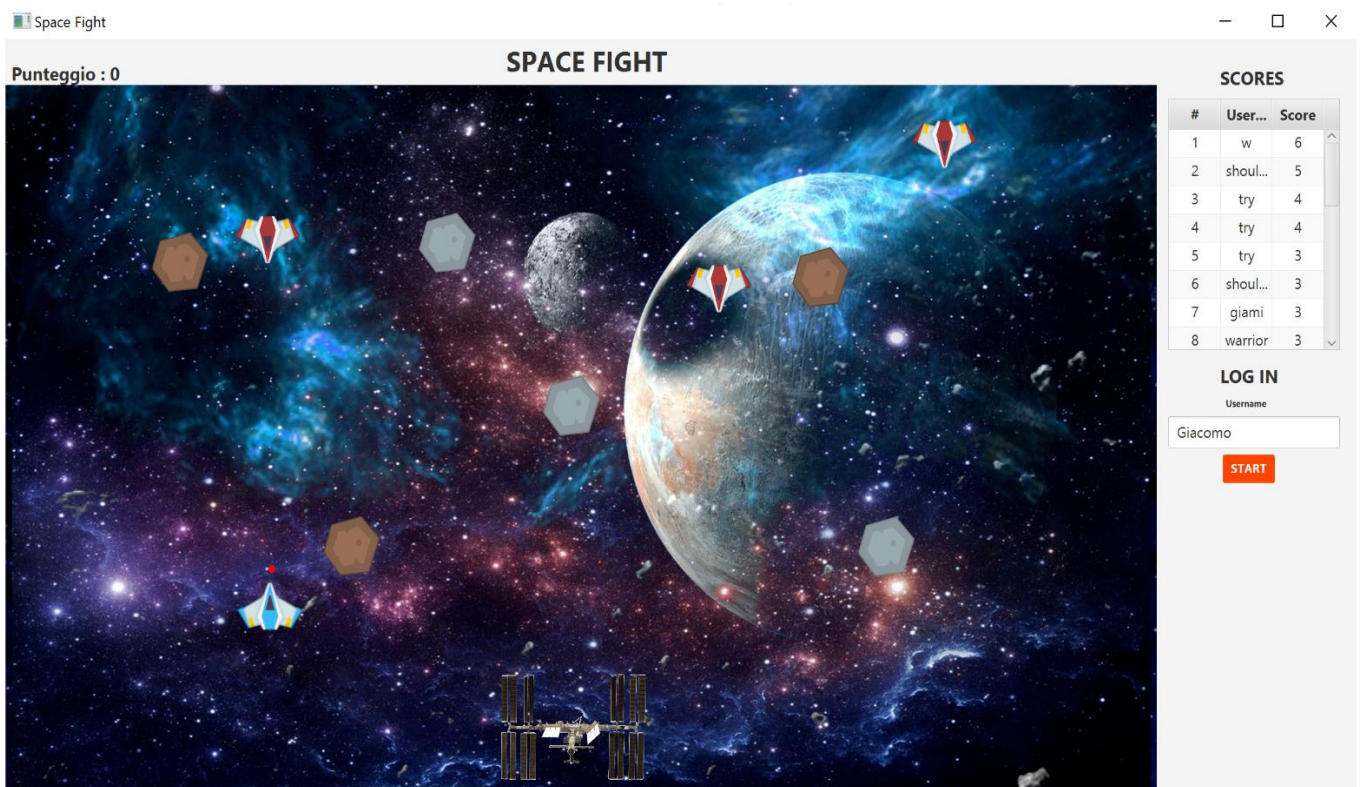


Space Fight – Manuale Utente

Piacentini Giacomo – Matricola: 627555

Artificial Intelligence and Data Engineering – Corso di programmazione avanzata 2020/2021

INTERFACCIA GRAFICA




L'interfaccia di gioco è un VBox, composto da tre differenti parti:

- 1) La prima parte (sulla sinistra) è la vera e propria schermata di gioco
- 2) La seconda parte (in alto a destra) è l'area degli scores, ovvero una tabella che tiene conto dei vari punteggi generati durante le diverse partite e organizza i punteggi in ordine decrescente. Vengono mostrati i primi 30 punteggi più alti.
Il # fa riferimento alla posizione in classifica del giocatore con l'username corrispondente.
Username e score sono rispettivamente il nome e il punteggio del giocatore.

- 3) La terza (in basso a destra) è l'area registrazione, dove l'utente deve inserire il proprio username.

AVVIO DI UNA NUOVA PARTITA

Per avviare una partita, bisogna premere il tasto play, dopo aver inserito il proprio nome username.

Dopodiché, compariranno sulla vista di gioco, la navicella del giocatore e cinque navicelle nemiche che scenderanno dall'alto verso il basso, con tempistiche e velocità differenti (attenzione, quelle con la forma più squadrata sono più veloci )

La navicella amica può muoversi tramite i comuni tasti da gioco W,A,S,D. W per andare in alto, A per andare verso sinistra, D per andare verso destra e S per scendere verso il basso.

Il player, inoltre, può sparare un laser tramite il tasto SPACE, che se colpisce una navicella nemica la distrugge e fa salire il punteggio di 1.

All'interno della mappa sono presenti diversi meteoriti fissi, che impediscono i movimenti alla navicella amica e se colpiti da laser, il laser viene distrutto.

Il gioco termina in questi casi:

- 1) La navicella del giocatore viene colpita da un laser nemico
- 2) La stazione spaziale viene colpita da un laser nemico
- 3) La stazione spaziale viene colpita da una navicella nemica
- 4) Collisione fra una navicella nemica e una amica

FILE CONFIGURAZIONE XML

Il file per la configurazione XML, contiene:

- ipClient: contiene l'indirizzo IP del client (127.0.0.1)
- ipServer: contiene l'indirizzo IP del server di log (127.0.0.1)
- portaServer: contiene il numero di porta (8080) del server di log
- username: username di accesso per il database (root)
- password: password di accesso per il database (non messa)
- numeroCampiTabella: numero dei campi da visualizzare nella tabella visuale (30)
- etaMassimaInGiorni: età massima (in giorni) di una partita (30)

SERVER LOG

Vi è un server di log che registra, tramite stampa su terminale e salvataggio su file XML chiamato eventoXML.xml i seguenti eventi:

avvio dell'applicativo

termine dell'applicativo pressione del bottone start.

```
ServerSpacefight (run) × SpaceFight (jfxsa-run) ×
run:
Nome Applicazione: SpaceFight
Ip Client: 127.0.0.1
Data (yyyy-MM-dd) : 2021-01-11
Ora (HH:mm:ss): 19:18:48
Evento: AVVIO
*****
Nome Applicazione: SpaceFight
Ip Client: 127.0.0.1
Data (yyyy-MM-dd) : 2021-01-11
Ora (HH:mm:ss): 19:18:52
Evento: PLAY
*****
Nome Applicazione: SpaceFight
Ip Client: 127.0.0.1
Data (yyyy-MM-dd) : 2021-01-11
Ora (HH:mm:ss): 19:18:57
Evento: TERMINE
*****
```