

Università degli studi di Bari facoltà di
scienze MM.FF.NN

Progetto Metodi Avanzati di Programmazione

Phosphorus textual-adventure

by

Vito Proscia mat. 735975

Email: v.proscia3@studenti.uniba.it



UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

Repository github: [Phosphorus](#)

Anno accademico 2022-2023

Contents

1	Introduzione	3
1.1	Definizione obiettivo principale	3
1.2	Trama	3
1.3	Soluzione gioco	3
1.4	Mappa di gioco	4
2	Dettagli implementativi	5
2.1	Progettazione Object-Oriented	5
2.2	File	5
2.3	Database	7
2.4	Threads	8
2.5	Api REST	8
2.6	Java Swing	8
3	Conclusioni	8
4	Considerazioni finali	8
I	Appendice Musica	8
I.1	Composizione	8
I.2	Produzione	8

1 Introduzione

1.1 Definizione obiettivo principale

L'obiettivo principale del progetto è la realizzazione di un'avventura testuale in Java, inglobando, in essa, le tecniche e gli argomenti trattati durante il corso di Metodi Avanzati di Programmazione.

1.2 Trama

Il protagonista, l'agente f24, si trova su una navicella spaziale in ritorno alla Terra da una missione che ha consistito nel catturare alieni per produrre il fosforo necessario alla sopravvivenza della Terra, infatti, sulla quest'ultima, il fosforo, che riveste un ruolo fondamentale per la sopravvivenza dei vegetali e quindi per il sostentamento dell'uomo è cominciato a diminuire drasticamente, per questo si organizzano spedizioni per catturare alieni in grado di produrlo.

Inizialmente, f24 si sveglierà dal sonno criogenico nel dormitorio con un ordine, impartito dal comandante, di indagare sulla misteriosa scomparsa di due alieni prigionieri. Il protagonista cercherà i due fuggitivi, districandosi tra le stanze dell'astronave ed interrogando i membri dell'equipaggio, fino a scoprire cosa viene fatto agli alieni prigionieri. Sarà solo a lui decidere se mantenere lo status quo o cambiare la situazione.

1.3 Soluzione gioco

La stanza iniziale è il dormitorio, la prima cosa che si deve fare è prendere la bombola d'ossigeno che sarà utile più avanti, una volta parlato con l'agente a13 bisogna recarsi alla sala meeting, a nord del dormitorio, dopo aver ascoltato il comandante si dovrà prendere la pistola e ci si recherà ad ovest per la mensa, lì ci saranno due scienziati s56 e s99, per proseguire ci si dovrà spostare ad est verso la sala macchine, una volta dentro ci si imbatte nel primo nemico, l'alieno Orionix, una volta sconfitto bisognerà prendere la chiave dello sgabuzzino, dopo di che ci si dovrà entrare andando a sud, una volta all'interno si dovrà prendere il bigliettino che conterrà il codice per accedere al laboratorio (4815) che si troverà appena a nord della sala meeting, una volta entrati troveremo l'ultimo nemico Nebulor, che sarà protetto da un altro scienziato, s47, una volta parlato con quest'ultimo starà al giocatore decidere se eliminare l'ultimo nemico o prendere la modifica della pistola per cercare di cambiare la situazione.

1.4 Mappa di gioco

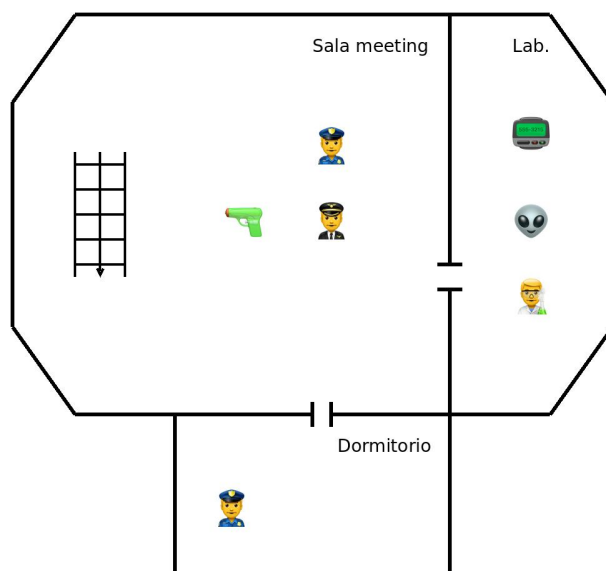


Figure 1: Mappa primo piano

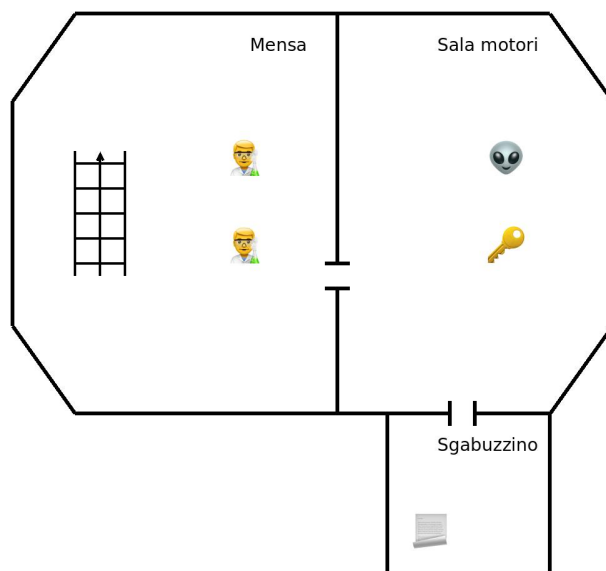


Figure 2: Mappa secondo piano

2 Dettagli implementativi

In questa sezione verranno analizzate tutte le tecniche e gli argomenti del corso sfruttati per la realizzazione dell'avventura testuale.

2.1 Progettazione Object-Oriented

La progettazione orientata agli oggetti (OOD) è un paradigma di programmazione che utilizza "oggetti" come unità fondamentali per progettare, organizzare e implementare software. Gli oggetti sono istanze di classi, che sono astrazioni che rappresentano concetti, dati e operazioni associati al dominio del problema che si sta affrontando, questa metodologia di progettazione garantisce riutilizzabilità del codice, facilità di manutenzione, chiarezza nel flusso di controllo e scalabilità.

Nello specifico si sono andate a progettare e realizzare una serie di classi garantendo l'incapsulamento di dati e operazioni, di modo che l'accesso a quest'ultimi sia limitato ai metodi della stessa classe, il che promuove la sicurezza e la modularità del codice.

Al fine di chiarire la struttura delle classi e delle loro relazioni, abbiamo utilizzato il linguaggio di modellazione UML (Unified Modeling Language) per appresentare in modo visuale la progettazione del software in modo chiaro e conciso.

Il diagramm è presente al seguente link: [diagramma_uml](#) ??????????????????????????????

2.2 File

Nel progetto i file sono stati utilizzati per modellare gli elementi narrativi e non dell'avventura, in particolare si hanno a disposizione tre file JSON (JavaScript Object Notation) che vanno a descrivere le stanze, i personaggi e gli oggetti di gioco in modo che le informazioni legate a questi siano facilmente modificabili ed estendibili per migliorare l'avventura, magari aggiungendo più stanze, modificando i dialoghi dei personaggi e perfino stravolgere la trama di gioco.

All'avvio del gioco avviene la lettura di questi file per inizializzare tutti i vari oggetti che costituiscono l'avventura, questo avviene utilizzando la libreria [jackson](#), che permette di leggere e utilizzare facilmente i file di tipo JSON, convertendoli in una mappa chiave-valore.

```

1  "0": {
2      "roomId": 0,
3      "roomName": "Dormitorio",
4      "roomDescription": "Dormitorio non molto grande ...",
5      "lookDescription": "Sei nel dormitorio, vicino a te c'è ...",
6      "floorNumber": 0,
7      "locked": false,
8      "south": null,
9      "north": 1,
10     "west": null,
11     "est": null,
12     "items": [
13         4
14     ],
15     "characters": [
16         0
17     ],
18     "oxygen": true,
19     "passwordRequired": false
20 },

```

(a) File delle stanze

```

1  "0": {
2      "characterId": 0,
3      "characterName": "a13",
4      "mainDialog": "Buongiorno f24, ho saputo che sei nuovo, ...",
5      "defaultDialog": "Hai sentito il comandante, dobbiamo andare ...",
6      "type": "friend"
7  },

```

(b) File dei personaggi

```

1  "0": {
2      "itemID": 0,
3      "itemType": "Weapon",
4      "itemName": "pistola",
5      "itemDescription": "Particolare pistola, ...",
6      "itemLocation": 1,
7      "itemAction": "spara",
8      "itemAlias": [
9          "pistola",
10         "arma"
11     ]
12 },

```

(c) File degli oggetti

2.3 Database

I salvataggi di gioco sono stati gestiti mediante l'uso della basi di dati, in particolare sono state progettate tre relazioni:

- game: memorizza i dati relativi al giocatore, come l'id, la stanza in cui si trova, il numero di nemici eliminati e il timestamp del salvataggio;
- inventory: raccoglie tutti gli oggetti recuperati durante l'avventura;
- killedCharacter: conserva i dati relativi ai personaggi eliminati nel corso dell'avventura.

Ogni qual volta che viene salvata la partita, se si esegue il primo salvataggio viene creato il database con le sue relazioni per poi inserirne i relativi dati, altrimenti con il nuovo salvataggio vengono sovrascritti i dati del vecchio.

Si è pensato di sovrascrivere sempre il salvataggio per motivi di semplicità, ma con la corrente progettazione è possibile modificare quest'aspetto dando la possibilità al giocatore di salvare/caricare più sessioni.

Esempio:

GAMEID	CURRENTROOMID	ENEMYCOUNT	GAMETIME	SAVETIMESTAMP
0	4	1	34	2024-01-30 11:08:48.017

Figure 3: Relazione game

ITEMID	GAMEID
0	0
4	0

Figure 4: Relazione inventory

CHARACTERID	GAMEID
5	0

Figure 5: Relazione killedCharacter

2.4 Threads

2.5 Api REST

2.6 Java Swing

3 Conclusioni

4 Considerazioni finali

I Appendice Musica

I.1 Composizione

Per arricchire l'atmosfera ho incluso nel progetto una mia improvvisazione musicale, cercando di trasmettere un po' quel senso di mistero che avvolge la vicenda di gioco, al fine di far immergere il giocatore ancora meglio nell'avventura.

I.2 Produzione

La parte relativa alla produzione musicale è stata una delle più ardue e al contempo una delle più divertenti, personalmente non mi ero mai cimentato, fino a questo momento, nella produzione di un brano musicale.

Nello specifico, con la funzione record del pianoforte digitale, ho registrato il brano per poi salvarlo in formato MIDI (Musical Instrument Digital Interface), successivamente ho esportato il file nell'applicazione per Ipad Garagaband, che permette amatorialmente di creare musica, per poi modificare lo strumento di esecuzione, aggiungendo le degli effetti sonori come il wobble ed il riverbero e andando ad esitare la traccia audio, tagliando o registrando nuovamente delle parti, infine ho salvato il tutto sotto forma di WAV (Waveform) per poi imporre il brano nel progetto facendo sì che sia riprodotta in loop per tutta la durata del gioco.

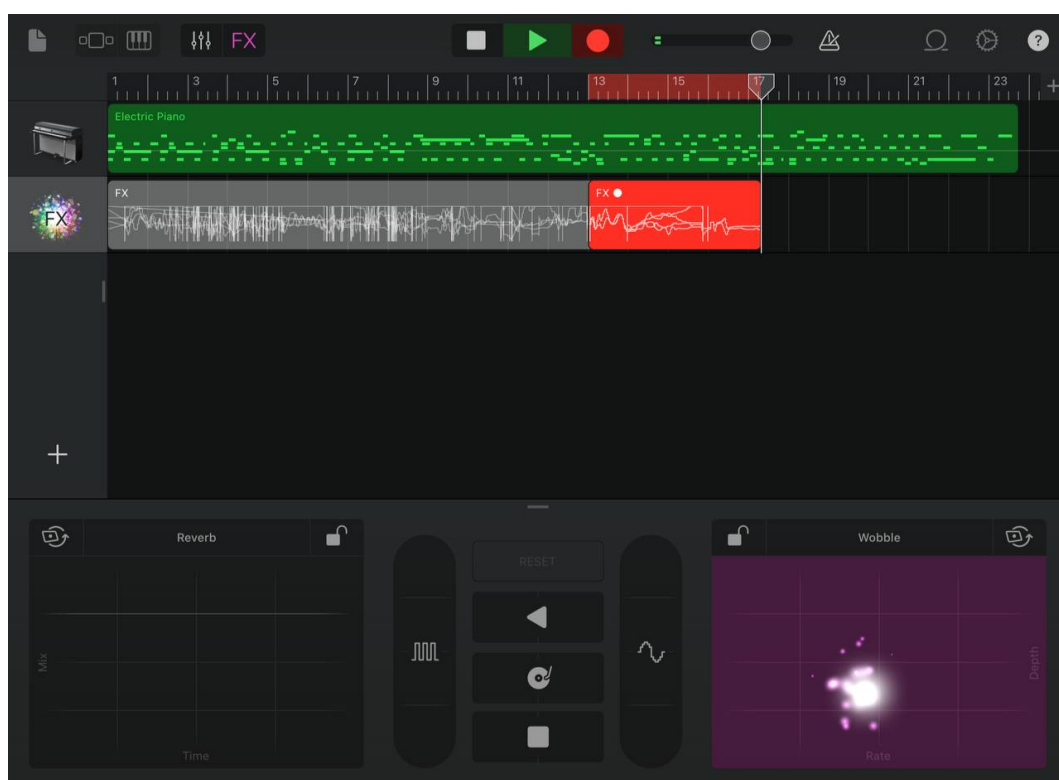


Figure 6: Screenshot dell'applicazione Garageband