Documentazione SPACEWAR

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

2 Analisi 4

2.1 Analisi del dominio 4

2.2 Analisi e specifica dei requisiti 4

2.2.1 Spiegazione elementi tabella dei requisiti: 5

2.3 Use case 5

2.4 Pianificazione 5

2.5 Analisi dei mezzi 5

2.5.1 Software 6

2.5.2 Hardware 6

3 Progettazione 6

3.1 Design dell’architettura del sistema 6

3.2 Design dei dati e database 6

3.3 Design delle interfacce 6

3.4 Design procedurale 6

4 Implementazione 7

5 Test 7

5.1 Protocollo di test 7

5.2 Risultati test 8

5.3 Mancanze/limitazioni conosciute 8

6 Consuntivo 8

7 Conclusioni 8

7.1 Sviluppi futuri 8

7.2 Considerazioni personali 8

8 Glossario 8

9 Bibliografia 9

9.1 Bibliografia per articoli di riviste: 9

9.2 Bibliografia per libri 9

9.3 Sitografia 9

10 Allegati 9

# Introduzione

## Informazioni sul progetto

Allievo: Alessandro Castelli

Docente coinvolto: Geo Petrini

Data inizio: 09.09.2022

Data fine 23.12.2022

## Abstract

In questo documento sono presenti le diverse fasi della realizzazione del progetto di questo semestre (Analisi, Progettazione, Implementazione e Test). Il progetto in questione è SpaceWar, proposto dal docente Geo Petrini.

## Scopo

Lo scopo del progetto è quello di ricreare un vecchio gioco chiamato appunto SpaceWar; quindi, creare un gioco ambientato nello spazio dove due navicelle si devono sparare tra di loro. Il gioco finisce quando una delle due navicelle esaurisce le vite disponibili. In questo gioco sono presenti la modalità con due giocatori, questi due giocatori muovono la navicella e sparano dalla stessa tastiera per rispettare il più possibile la versione originale del gioco, la modalità in singleplayer, la modalità con il pianeta al centro della schermata e l’attrazione gravitazionale verso il centro dello schermo.

# Analisi

## Analisi del dominio

In questo progetto dovrò ricreare il gioco SpaceWar e per farlo ho deciso di utilizzare python come linguaggio di programmazione con l’ausilio di pygame.

## Analisi e specifica dei requisiti

Questi sono i requisiti che sono riuscito a raccogliere

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Muovere la navicella con la tastiera |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | “WAD” per il player 1 e “IJL” per il player 2 |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Sparare con la tastiera |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Player 1: laser “q” e missili “e”  Player 2: laser “u” e missili “o” |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Possibilità di giocare da solo o con un altro giocatore dalla stessa macchina |
| **Priorità** | 1 |
| **Versione** | 1.0 |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Possibilità di scegliere se avere la gravità oppure no |
| **Priorità** | 2 |
| **Versione** | 1.0 |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | La navicella deve poter sparare sia missili che laser |
| **Priorità** | 1 |
| **Versione** | 1.0 |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Laser con una determinata dimensione |
| **Priorità** | 2 |
| **Versione** | 1.0 |

|  |  |
| --- | --- |
| **ID: REQ-007** | |
| **Nome** | La navicella deve avere l’energia che si scarica sparando |
| **Priorità** | 3 |
| **Versione** | 1.0 |

|  |  |
| --- | --- |
| **ID: REQ-008** | |
| **Nome** | La navicella deve essere dotata di scudi (vite) |
| **Priorità** | 2 |
| **Versione** | 1.0 |

|  |  |
| --- | --- |
| **ID: REQ-009** | |
| **Nome** | Possibilità di scegliere se avere il pianeta oppure no |
| **Priorità** | 2 |
| **Versione** | 1.0 |

## Use case

I casi d’uso rappresentano l’interazione tra i vari attori e le funzionalità del prodotto.

## Pianificazione

Prima di stabilire una pianificazione bisogna avere almeno una vaga idea del modello di sviluppo che si intende adottare. In questa sezione bisognerà inserire il modello concettuale di sviluppo che si seguirà durante il progetto. Gli elementi di riferimento per una buona pianificazione derivano da una scomposizione top-down della problematica del progetto.

La pianificazione può essere rappresentata mediante un diagramma di Gantt:

|  |
| --- |
| Figura : Esempio di diagramma di Gantt. |

Se si usano altri metodi di pianificazione (p.es. scrum), dovranno apparire in questo capitolo.

## Analisi dei mezzi

### Software

* Python
* Pygame

### Hardware

* PC scolastico
* Hard Disk personale

# Progettazione

## Design delle interfacce

Ho iniziato creando per prima cosa con i design delle interfacce che voglio realizzare utilizzando un sito che mi ha permesso di disegnarle (…….), queste sono le interfacce:

Schermata home:



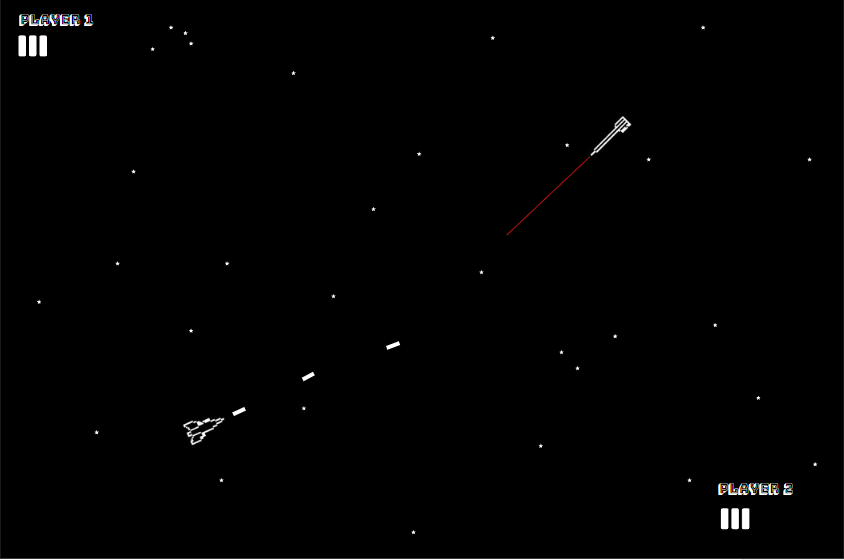
Ho deciso di mettere le modalità di gioco cambiabili tramite i tasti della tastiera; quindi, in questa schermata appariranno le scritte con definito quale tasto bisogna premere per cambiare una determinata impostazione del gioco.

Schermata informazioni:



In questa schermata durante il design avevo pensato di mettere l’immagine delle due navicelle presenti in gioco ma durante l’implementazione ho deciso di scrivere solo i tasti che l’utente dovrà utilizzare per giocare.

Schermata di gioco:



Questa è la schermata di gioco con presenti le vite dei giocatori. In questo design il laser sparato dalla navicella è un semplice laser dritto, nell’implementazione però ho optato per una sfera laser, più veloce e molto più grande dei proiettili che ha bisogno di arrivare all’energia massima per essere sparato.

Schermata finale:



In questa schermata viene semplicemente rappresentato quale giocatore ha vinto e il tasto da schiacciare per tornare alla schermata home.

## Design procedurale

Descrive i concetti dettagliati dell’architettura/sviluppo utilizzando ad esempio:

* Diagrammi di flusso e Nassi.
* Tabelle.
* Classi e metodi.
* Tabelle di routing
* Diritti di accesso a condivisioni …

Questi documenti permetteranno di rappresentare i dettagli procedurali per la realizzazione del prodotto.

# Implementazione

In questo capitolo dovrà essere mostrato come è stato realizzato il lavoro. Questa parte può differenziarsi dalla progettazione in quanto il risultato ottenuto non per forza può essere come era stato progettato.

Sulla base di queste informazioni il lavoro svolto dovrà essere riproducibile.

In questa parte è richiesto l’inserimento di codice sorgente - Print Screen - di maschere solamente per quei passaggi particolarmente significativi e/o critici.

Inoltre, dovranno essere descritte eventuali varianti di soluzione o scelte di prodotti con motivazione delle scelte.

Non deve apparire nessuna forma di guida d’uso di librerie o di componenti utilizzati. Eventualmente questa va allegata.

Per eventuali dettagli si possono inserire riferimenti ai diari.

# Test

## Protocollo di test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-001  REQ-002 | **Nome:** | Test tastiera |
| **Descrizione:** | I 2 giocatori devono poter muovere e sparare tramite la tastiera | | |
| **Procedura:** | Avviare il gioco e controllare che i 2 giocatori possono muoversi e sparare con I tasti definiti nei Requisiti 001 e 002 | | |
| **Risultati attesi:** | I tasti permettono di muovere la navicella e sparare correttamente | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-002  REQ-003 | **Nome:** | Modalità singleplayer o 2 player |
| **Descrizione:** | Gioco funzionante sia in single player che con 2 giocatori | | |
| **Procedura:** | Avviare il gioco in entrambe le modalità e testare se il secondo player funziona autonomamente in single player e se con 2 giocatori la seconda navicella funziona con i comandi. | | |
| **Risultati attesi:** | In sigleplayer la seconda navicella si muove e spara da sola, mentre con 2 giocatori la seconda navicella si muove e spara con l’utilizzo della tastiera | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-003  REQ-005 | **Nome:** | Sparare missile e laser |
| **Descrizione:** | La navicella deve poter sparare missili o laser | | |
| **Procedura:** | Avviare il gioco e testare che con un tasto la navicella spara dei missili e con l’altro invece un laser | | |
| **Risultati attesi:** | La navicella può sparare si laser che missili | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-004  REQ-004 | **Nome:** | Test gravità |
| **Descrizione:** | Il gioco deve avere la modalità con gravità e quella senza | | |
| **Procedura:** | Avviare il gioco con la gravità attivata e vedere se la navicella e I missile vengono attratti dal pianeta al centro della schermata di gioco | | |
| **Risultati attesi:** | La naviccella e I missile sono soggetti alla gravità del pianeta | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-005  REQ-006 | **Nome:** | Laser con determinata dimensione |
| **Descrizione:** | Il laser sparato dalla navicella deve avere una determinate dimensione | | |
| **Procedura:** | Provare a sparare il laser con la navicella e controllare che il laser abbia una dimensione ne troppo grande, ne troppo piccola | | |
| **Risultati attesi:** | Il laser ha una dimensione accettabile (ne troppo grande, ne toppo piccolo) | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-006  REQ-007 | **Nome:** | Energia navicella |
| **Descrizione:** | La navicella ha un limite di energia che si scarica sparando | | |
| **Procedura:** | Provare a sparare con la navicella e controllare che non possa sparare troppi colpi di fila e aspettare che l’energia si ricarichi per poter sparare di nuovo | | |
| **Risultati attesi:** | La navicella non può sparare troppi colpi di fila e l’energia si ricarica quando non spari | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-007  REQ-008 | **Nome:** | Scudi navicella |
| **Descrizione:** | La navicella non esplode quando viene colpita ma ha degli scudi | | |
| **Procedura:** | Sparare ad una navicella e vedere se quando viene colpita scende lo scudo | | |
| **Risultati attesi:** | Quando viene colpita la navicella scende lo scudo | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-008  REQ-009 | **Nome:** | Possibilità di avere il pianeta |
| **Descrizione:** | Al centro della schermata di gioco deve esserci la possibilità di avere un pianeta che distruggerà le navicelle che ci vanno contro | | |
| **Procedura:** | Avviare il gioco con il pianeta attivo e controllare che la navicella si distrugga quando ci va contro | | |
| **Risultati attesi:** | Il pianeta è presente al centro della schermata | | |

## Risultati test

|  |  |
| --- | --- |
| Test | Risultato |
| TC-001 | Passato |
| TC-002 | Fallito |
| TC-003 | Passato |
| TC-004 | Fallito |
| TC-005 | Passato |
| TC-006 | Passato |
| TC-007 | Passato |
| TC-008 | Passato |

## Mancanze/limitazioni conosciute

Purtroppo, non sono riuscito ad implementare la modalità in singleplayer e la gravità.

Avendo perso 12 ore di lezione e avendo riscontrato qualche problema nel realizzare altre cose non sono riuscito ad implementare queste due funzionalità.

Per la gravità mi ero preso del tempo per fare ricerche su come farla ma non ho trovato nulla di veramente utile per il mio caso e essendomi rimasto troppo poco alla fine del progetto non sono riuscito ad implementarla.

Anche per la modalità in singleplayer per questioni di tempo non sono riuscito ad implementarla.

# Consuntivo

Consuntivo del tempo di lavoro effettivo e considerazioni riguardo le differenze rispetto alla pianificazione (cap. 1.7) (ad esempio Gantt consuntivo).

# Conclusioni

Quali sono le implicazioni della mia soluzione? Che impatto avrà? Cambierà il mondo? È un successo importante? È solo un’aggiunta marginale o è semplicemente servita per scoprire che questo percorso è stato una perdita di tempo? I risultati ottenuti sono generali, facilmente generalizzabili o sono specifici di un caso particolare? ecc.

## Sviluppi futuri

Sicuramente bisognerebbe implementare il singleplayer

## Considerazioni personali

Grazie a questo progetto ho migliorato le mie capacità nella programmazione ad oggetti e all’utilizzo del linguaggio di programmazione python.

# Bibliografia

## Sitografia

1. URL del sito (se troppo lungo solo dominio, evt completo nel diario),
2. Eventuale titolo della pagina (in italico),
3. Data di consultazione (GG-MM-AAAA).

# Allegati

Elenco degli allegati, esempio:

* Diari di lavoro
* Codici sorgente/documentazione macchine virtuali
* Istruzioni di installazione del prodotto (con credenziali di accesso) e/o di eventuali prodotti terzi
* Documentazione di prodotti di terzi
* Eventuali guide utente / Manuali di utilizzo
* Mandato e/o QdC
* Prodotto
* …