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

Sono Alessandro Perri, allievo informatico presso la scuola Arti e Mestieri di Trevano e sono il produttore di questo progetto che è stato supervisionato dal professor Geo Petrini.

Il progetto è stato assegnato il 09.09.2022 con data di consegna entro il 23.12.2022.

## Abstract

## Situazione iniziale

Inizialmente in questo progetto mi è stato chiesto di realizzare un gioco già esistente, ma migliorato.  
Questo gioco permette di giocare in due utilizzando una sola tastiera, oppure giocare da solo contro un bot.

## Approccio

Per questo progetto, saranno necessarie buone conoscenze nella programmazione web, poiché per la creazione del gioco si utilizzerà il framework phaser che utilizza HTML5 Canvas per la visualizzazione e JavaScript per l’esecuzione. Grazie a questo framework sono riuscito a realizzare il gioco.

## Risultati

Per questo progetto ho utilizzato il framework phaser che non avevo mai utilizzato prima, dunque ho avuto la possibilità di ampliare il mio bagaglio delle conoscenze. Uno dei punti fondamentali tenuti in considerazione durante il progetto riguarda lo studio per raggiungere la facilità di utilizzo per un utente medio. In conclusione, il progetto che ho realizzato è ora l’unico gioco moderno che simula il vecchio gioco Spacewar.

## Scopo

Scopi

* Didattici:
  + Saper creare e rispettare una progettazione.
  + Saper documentare il lavoro.
  + Saper creare i diari.
  + Saper creare Use-case.
* Operativi:
  + Saper creare una Gui.
  + Saper utilizzare phaser.

# Analisi

## Analisi del dominio

Per questo progetto mi è stato chiesto di creare un gioco dove si possa giocare sia da solo che con un amico e che abbia varie opzioni di scelta per poterlo rendere più bello e divertente. Questo gioco potrà essere utilizzato quando si avrà bisogno di un momento di svago e di divertimento.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-01** | |
| **Nome** | Navicelle che sparano laser e missili |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Sotto requisiti** | |
| **001** | Delay tra i colpi |
| **002** | Lunghezza massima dei laser |
| **003** | Sparare diminuisce l’energia |
| **004** | Essere colpiti diminuisce lo scudo |
| **ID: REQ-01** | |

|  |  |
| --- | --- |
| **ID: REQ-02** | |
| **Nome** | Modalità di gioco |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Sotto requisiti** | |
| **001** | Single player (Player vs PC) |
| **002** | Dual player hotseat |
| **ID: REQ-02** | |

|  |  |
| --- | --- |
| **ID: REQ-03** | |
| **Nome** | Movimento |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Sotto requisiti** | |
| **001** | Si può solo accelerare |
| **002** | Si può curvare |
| **ID: REQ-03** | |

|  |  |
| --- | --- |
| **ID: REQ-04** | |
| **Nome** | Ostacoli opzionali |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Sotto requisiti** | |
| **001** | Pianeta al centro |
| **002** | Gravità |
| **ID: REQ-04** | |

|  |  |
| --- | --- |
| **ID: REQ-05** | |
| **Nome** | Poteri speciali |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Sotto requisiti** | |
| **001** | Azzero l’energia e ricarico la vita |
| **002** | Turbo |
| **ID: REQ-05** | |

|  |  |
| --- | --- |
| **ID: REQ-06** | |
| **Nome** | Scelta navicella |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Sotto requisiti** | |
| **001** | 2 tipi di navicella |
| **ID: REQ-06** | |

|  |  |
| --- | --- |
| **ID: REQ-07** | |
| **Nome** | Exit |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note:** | Chiude il gioco |
| **ID: REQ-07** | |

### Spiegazione elementi tabella dei requisiti:

**ID**: identificativo univoco del requisito.

**Nome**: breve descrizione del requisito.

**Priorità**: l'importanza con la quale deve essere svolto un requisito.

**Versione**: indica la versione del requisito.

**Note**: eventuali osservazioni.

**Sotto requisiti**: elementi che compongono il requisito.

## Use case

Figura : Use-Case

## Pianificazione Di seguito c’è la pianificazione preventiva, nel Gantt preventivo è stimato il tempo necessario per completare le task del progetto.

Figura : Gantt preventivo

## Analisi dei mezzi

Per realizzare il progetto è stato utilizzato un computer con Visual Studio Code, Phaser3 come Framework, e un server php per visualizzare il gioco.

### Software

Per realizzare questo progetto sono stati utilizzati i seguenti software:

* OS Computer - Windows 10: Per la creazione di tutto il progetto.
* Visual Studio Code - 1.72: Per tutto ciò che riguarda la scrittura del codice.
* Phaser – 3.55.2: Utilizzato come framework.
* Server php: Utilizzato per la visualizzazione del gioco.

### Hardware

Un computer con:

* CPU Intel Core i7-7700
* RAM 16 GB

# Progettazione

Questo capitolo descrive esaustivamente come deve essere realizzato il prodotto fin nei suoi dettagli. Una buona progettazione permette all’esecutore di evitare fraintendimenti e imprecisioni nell’implementazione del prodotto.

## Design del gioco

Il gioco è strutturato nel modo seguente:

**Interfaccia home page**

La home page è l’interfaccia che si visualizza non appena si avvia il gioco, da qui posso scegliere le varie opzioni.

****

Figura : Home page

**Interfaccia bottone play**

Nell’interfaccia del bottone play si hanno le due scelte per poter giocare, single player o dual player.

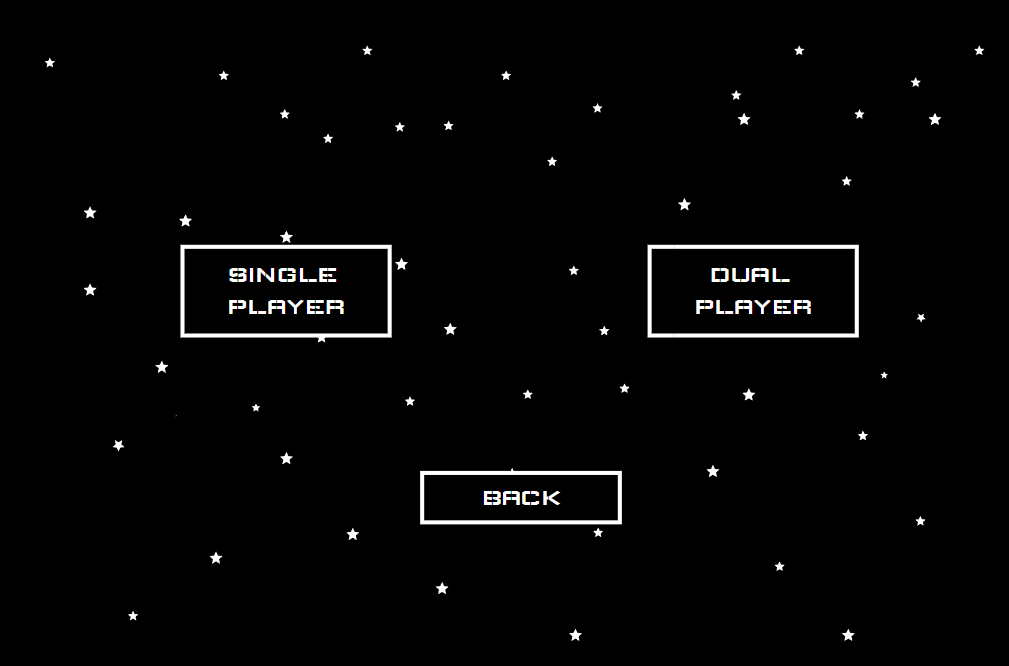


Figura : Interfaccia bottone play

**Interfaccia bottone options**

Immagine che contiene testo, nero, oggetto da esterni, notte

Descrizione generata automaticamenteNell’interfaccia options ci sono le varie opzioni da selezionare, si può scegliere di utilizzare il pianeta e la gravità e quale navicella utilizzare

Figura : Interfaccia bottone options

**Interfaccia bottone instructions**

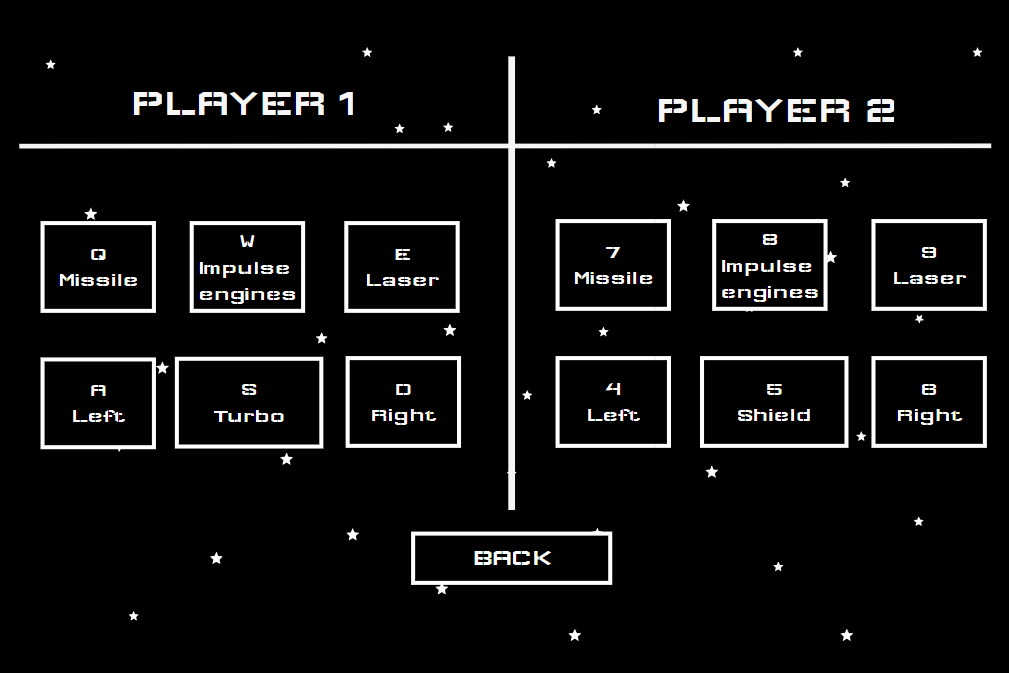
Nell’interfaccia del bottone instructions vengono mostrati i comandi per poter muovere, far sparare e utilizzare il potere speciale della navicella.

Figura : Interfaccia bottone instructions

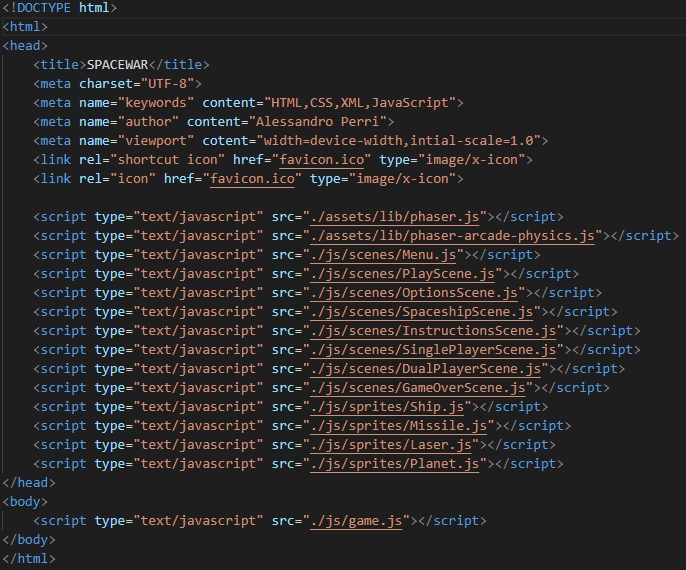
**Interfaccia bottone exit**

Appena il bottone exit viene cliccato si chiuderà la pagina.

# Implementazione

## File index

Nel file index vengono importate tutte le classi che ho implementato al fine di visualizzarle grazie alla classe presente nel body game.js.



### Utilizzo Phaser 3

Per questo gioco, come già detto precedentemente ho utilizzato il framework Phaser nella sua ultima versione 3.55.2. Per poter utilizzare tutte le sue funzionalità ho dovuto importarlo nel codice.

## Classe game

Nella seguente classe troveremo tutte le configurazioni per la creazione del gioco.

Essa comprende una serie di variabili globali come pure le varie scene (menu, impostazioni, gioco).

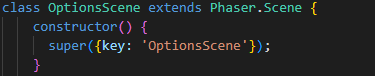
**Configurazioni generali**

## Scene

Il codice del gioco è contenuto nelle diverse scene, esse ti permettono di rendere il codice più leggibile e nel mio caso di passare da una scena all’altra semplicemente cliccando un bottone.   
Ogni scena ha bisogno di una key per essere riconosciuta e deve dichiarare i metodi preload(), create(), update(),

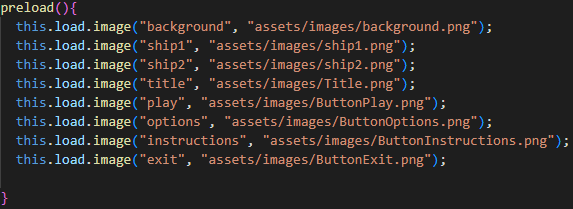
### Scena menu

Il menu è la scena principale che viene visualizzata appena si apre il gioco, da qui cliccando i vari pulsanti posso passare alle scene successive.

**Constructor**

Inizialmente creo la classe estendo Phaser.Scene siccome la uso come scena.

Dopodiché nel costruttore dichiaro la key che mi servirà per poter passare da una scena all’altra.

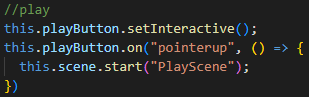
**Preload**

Per lo stile del pulsante ho utilizzato delle immagini dunque nel preload bisogna caricarle.

Come primo argomento passo una key che mi permette poi di andare a prendere l’immagine, mentre come secondo argomento inserisco il percorso dell’immagine.

**Create**

Dopo aver caricato le immagini, assegno l’immagine ad una variabile mettendo la posizione x-y e la chiave che ho inserito nel preload per poter prendere l’immagine.



Successivamente metto setInteractive() alla variabile, perché questa funzione mi permette di trasformare l’immagine in un vero e proprio pulsante.

Dopodiché utilizzo la funzione “pointerup” che mi permette di passare alla scena play non appena clicco il pulsante.

Infine eseguo lo stesso procedimento per i pulsanti options, instructions e exit che però al click non passerà a nessuna scena ma chiuderà l’applicazione.

### Scena play

In questa scena ci sarà la scelta di gioco, ovvero in single player o in dual player mode.

Per l’implementazione dai pulsanti single player, dual player e back, ho eseguito esattamente gli stessi passaggi del punto 4.3.1.

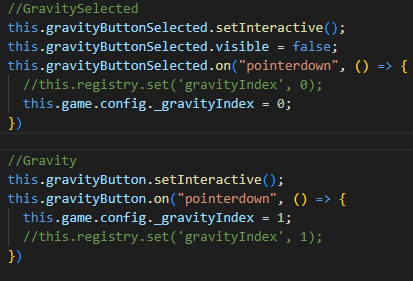
### Scena instructions

La scena instructions ti permette di visualizzare tutti i comandi del player 1 e 2 per poter muovere e far sparare la navicella.

Per il pulsante back ho eseguito gli stessi passaggi del punto 4.3.1.

### Scena options

Nella optionsScene ho la possibilità di scegliere le opzioni se utilizzare il pianeta, se utilizzare la gravità e quale navicella usare.

**Create**

Per la scelta delle opzioni ho utilizzato le variabili globali create nella classe game.

Quando clicco il pulsante gravityButton esso sparirà e apparirà il pulsante gravityButtonSelected.

La variabile creata la utilizzo scrivendo this.game.confif.\_gravityIndex e successivamente vado ad assegnarli il valore 1 se viene cliccata.

# Test

## Protocollo di test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-01 | **Nome:** | Navicelle che sparano laser e missili |
| **Descrizione:** | Una navicella quando si sposta deve poeter sparare sia laser che missile per eliminare l’avversario. | | |
| **Procedura:** | Player 1   * Cliccare il tasto “Q” e verificare se spara i missili * Cliccare il tasto “E” e verificare se spara i laser   Player 2   * Cliccare il tasto “7” e verificare se spara i missili * Cliccare il tasto “9” e verificare se spara i laser | | |
| **Risultati attesi:** | La navicella deve sparare sia I missile che I laser. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-002  REQ-01 | **Nome:** | Specifiche laser e missili |
| **Descrizione:** | I laser devono avere una lunghezza massima e se colpisco un missile o un player esso si distrugge.  I missili non hanno una lunghezza massima ma quando colpiscono il bordo dell’area di gioco o un player si distrugge. | | |
| **Procedura:** | Verifiche missile   * Si distrugga quando tocca il bordo dell’area di gioco e se presente quando tocca il pianeta. * Se quando colpisce un player esso perda una vita.   Verifiche laser   * Se il laser abbia una lunghezza massima. * Se colpisco un player esso perda mezza vita. | | |
| **Risultati attesi:** | I laser e I missili soddisfano Ie loro specifiche. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-003  REQ-02 | **Nome:** | Modalità di gioco |
| **Descrizione:** | Selezionare la modalità di gioco, ovvero single player o dual player. | | |
| **Procedura:** | 1. Cliccare play 2. Scegliere la modalità 3. Se seleziono single player verificare che il player 2 si muova da solo 4. Se seleziono dual player verificare che il player 2 si muova i numeri | | |
| **Risultati attesi:** | In modalità single player il player 2 deve essere controllato dal pc, mentre in modalità dual player il player 2 deve essere controllato dall’utente tramite ï numeri. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-004  REQ-03 | **Nome:** | Movimento |
| **Descrizione:** | La navicella può solo andare avanti e curvare. | | |
| **Procedura:** | 1. Iniziare a giocare 2. Verificare che posso andare solo avanti e curvare con i vari tasti. 3. Verificare che il movimento sia graduale e non a scatti. 4. Per il Movimento vedi GUI instructions. | | |
| **Risultati attesi:** | La navicella deve curvare e accellerare. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-005  REQ-03 | **Nome:** | Collisioni |
| **Descrizione:** | Le navicelle se si scontrano o sbattono contro il pianeta esplodono automaticamente. | | |
| **Procedura:** | 1. Iniziare a giocare 2. Andare contro l’altra navicella e verificare che entrambe esplodano. 3. Andare contro il pianeta e verificare che la navicella esploda. | | |
| **Risultati attesi:** | La navicella deve soddisfare le collisioni che ha nella sua specifica. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-006  REQ-04 | **Nome:** | Ostacoli opzionali |
| **Descrizione:** | Negli ostacoli posso scegliere se avere il pianeta al centro o la gravità. | | |
| **Procedura:** | 1. Cliccare options 2. Scegliere se usare una di queste opzioni 3. Entrare nel gioco e verificare se ci sono | | |
| **Risultati attesi:** | Le opzioni se scelte dovrebbero apparire quando inizio a giocare. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-007  REQ-05 | **Nome:** | Poteri speciali |
| **Descrizione:** | Ogni giocatore se clicca un tasto dovrebbe poter attivare la sua specialità. | | |
| **Procedura:** | Player 1   * Cliccare il tasto “s” e vedere se la navicella utilizza la sua specialità.   Player 2   * Cliccare il tasto “5” e vedere se la navicella utilizza la sua specialità. | | |
| **Risultati attesi:** | Quando clicco il relativo tasto la navicella deve attivare la sua specialità. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-008  REQ-06 | **Nome:** | Scelta navicella |
| **Descrizione:** | Nelle opzioni si può scegliere la navicella da usare | | |
| **Procedura:** | 1. Cliccare option 2. Cliccare spaceship 3. Selezionare la navicella che si desidera | | |
| **Risultati attesi:** | Quando inizio a giocare dovrei utilizzare la navicella che ho scelto. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-009  REQ-07 | **Nome:** | Exit |
| **Descrizione:** | Cliccando exit chiudo il gioco. | | |
| **Procedura:** | 1. Cliccare exit | | |
| **Risultati attesi:** | Il gioco dovrebbe chiudersi. | | |

## Risultati test

Tabella riassuntiva in cui si inseriscono i test riusciti e non del prodotto finale. Se un test non riesce e viene corretto l’errore, questo dovrà risultare nel documento finale come riuscito (la procedura della correzione apparirà nel diario), altrimenti dovrà essere descritto l’errore con eventuali ipotesi di correzione.

## Mancanze/limitazioni conosciute

Descrizione con motivazione di eventuali elementi mancanti o non completamente implementati, al di fuori dei test case. Non devono essere riportati gli errori e i problemi riscontrati e poi risolti durante il progetto.

# 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

Migliorie o estensioni che possono essere sviluppate sul prodotto.

## Considerazioni personali

Cosa ho imparato in questo progetto? ecc.

# Glossario

## 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).

**Esempio:**

* http://standards.ieee.org/guides/style/section7.html, *IEEE Standards Style Manual*, 07-06-2008.

# 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
* …