ChromeDino

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

Analisi 4

1.4 Analisi del dominio 4

1.5 Analisi e specifica dei requisiti 4

1.6 Use case 6

1.7 Pianificazione 6

1.8 Analisi dei mezzi 6

1.8.1 Software 6

1.8.2 Hardware 6

2 Progettazione 7

2.1 Design dell’architettura del sistema 7

2.2 Design dei dati e database 7

2.3 Design delle interfacce 7

2.4 Design procedurale 7

3 Implementazione 8

4 Test 8

4.1 Protocollo di test 8

4.2 Risultati test 9

4.3 Mancanze/limitazioni conosciute 9

5 Consuntivo 9

6 Conclusioni 9

6.1 Sviluppi futuri 9

6.2 Considerazioni personali 9

7 Bibliografia 9

7.1 Bibliografia per articoli di riviste: 9

7.2 Bibliografia per libri 9

7.3 Sitografia 10

8 Allegati 10

# Introduzione

## Informazioni sul progetto

Sono Manuel Grosso, un allievo della scuola informatica Arti e Mestieri di Trevano.

In questo progetto ho il ruolo di produttore ed il progetto è stato supervisionato dal professor Geo Petrini.

Il progetto ha avuto la durata di quasi un semestre scolastico più precisamente dal 09.09.21 al 23.12.21

## Abstract

In questo documento potete trovare tutte le informazioni spiegate in maniera dettagliate che si trovano alla base dello sviluppo del ChromeDino a partire dalla progettazione con Acivity Diagram e Gant fino ad arrivare alla spiegazione del codice metodo dopo metodo.

## Scopo

Questo progetto non ha un vero e proprio scopo essendo semplicemente un videogioco con la quale puoi divertirti a sfidare i tuoi amici dallo stesso computer.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-01** | |
| **Nome** | Multiplayer dallo stesso pc |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Massimo 4 giocatori. |
| **Sotto requisiti** | |
| **011** | Interfaccia diversa per numero di utenti. |

|  |  |
| --- | --- |
| **ID: REQ-02** | |
| **Nome** | Grafica 8bit |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Grafica semplice |
| **Sotto requisiti** | |

|  |  |
| --- | --- |
| **ID: REQ-03** | |
| **Nome** | Ogni dinosauro visibile a schermo |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Numero dinosauri visibili uguale a numero giocatori |
| **Sotto requisiti** | |
| **031** | Sapere numero giocatori |
| **032** | Creare campi di gioco |

|  |  |
| --- | --- |
| **ID: REQ-04** | |
| **Nome** | Ogni player possibilità inserire nome e personalizzare dinosauro |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Se non viene inserito nessun nome default (es. Guest1) |
| **Sotto requisiti** | |
| **041** | Immagini dinosauri di diversi colore |
| **042** | File dove salvare nome |

|  |  |
| --- | --- |
| **ID: REQ-05** | |
| **Nome** | Più controlli di input |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Input già scelti di default |
| **Sotto requisiti** | |
| **051** | Tastiera e Mouse |

|  |  |
| --- | --- |
| **ID: REQ-06** | |
| **Nome** | 2 livelli di sfondo in parallasse |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | 2 velocità differenti per i due sfondi |
| **Sotto requisiti** | |
| **061** | Avere i 2 sfondi |

|  |  |
| --- | --- |
| **ID: REQ-07** | |
| **Nome** | Generazione terreno infinita |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Finché non si muore si continua a generare lo sfondo |
| **Sotto requisiti** | |
| **071** | Controlli di collisione |

|  |  |
| --- | --- |
| **ID: REQ-08** | |
| **Nome** | Highscore e cancellare i risultati |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Top 20 highscore con nome del giocatore e possibilità resettare tutto |
| **Sotto requisiti** | |
| **071** | File per salvarsi i dati |

**Spiegazione elementi tabella dei requisiti:**

**ID**: identificativo univoco del requisito

**Nome**: breve descrizione del requisito

**Priorità**: indica l’importanza di un requisito nell’insieme del progetto, definita assieme al committente. Ad esempio, poter disporre di report con colonne di colori diversi ha priorità minore rispetto al fatto di avere un database con gli elementi al suo interno. Solitamente si definiscono al massimo di 2-3 livelli di priorità.

**Versione**: indica la versione del requisito. Ogni modifica del requisito avrà una versione aggiornata.

Sulla documentazione apparirà solamente l’ultima versione, mentre le vecchie dovranno essere inserite nei diari.

**Note**: eventuali osservazioni importanti o riferimenti ad altri requisiti.

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

## Activity Diagram



## Pianificazione

|  |
| --- |
| Figura 2: Diagramma di Gantt. |

## Analisi dei mezzi

Per realizzare il progetto è stato utilizzato un computer con Sublime e Phaser3 come Framework mentre per visualizzare il gioco Chrome con un server php.

### Software

OS Computer: Windows 10 versione 20H2 (build OS 19042.1415)

Sublime:

Phaser: version 3.55.2

Chrome: version 96.0.4664.110 (Build ufficiale) (a 64 bit)

### Hardware

Processore: Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz

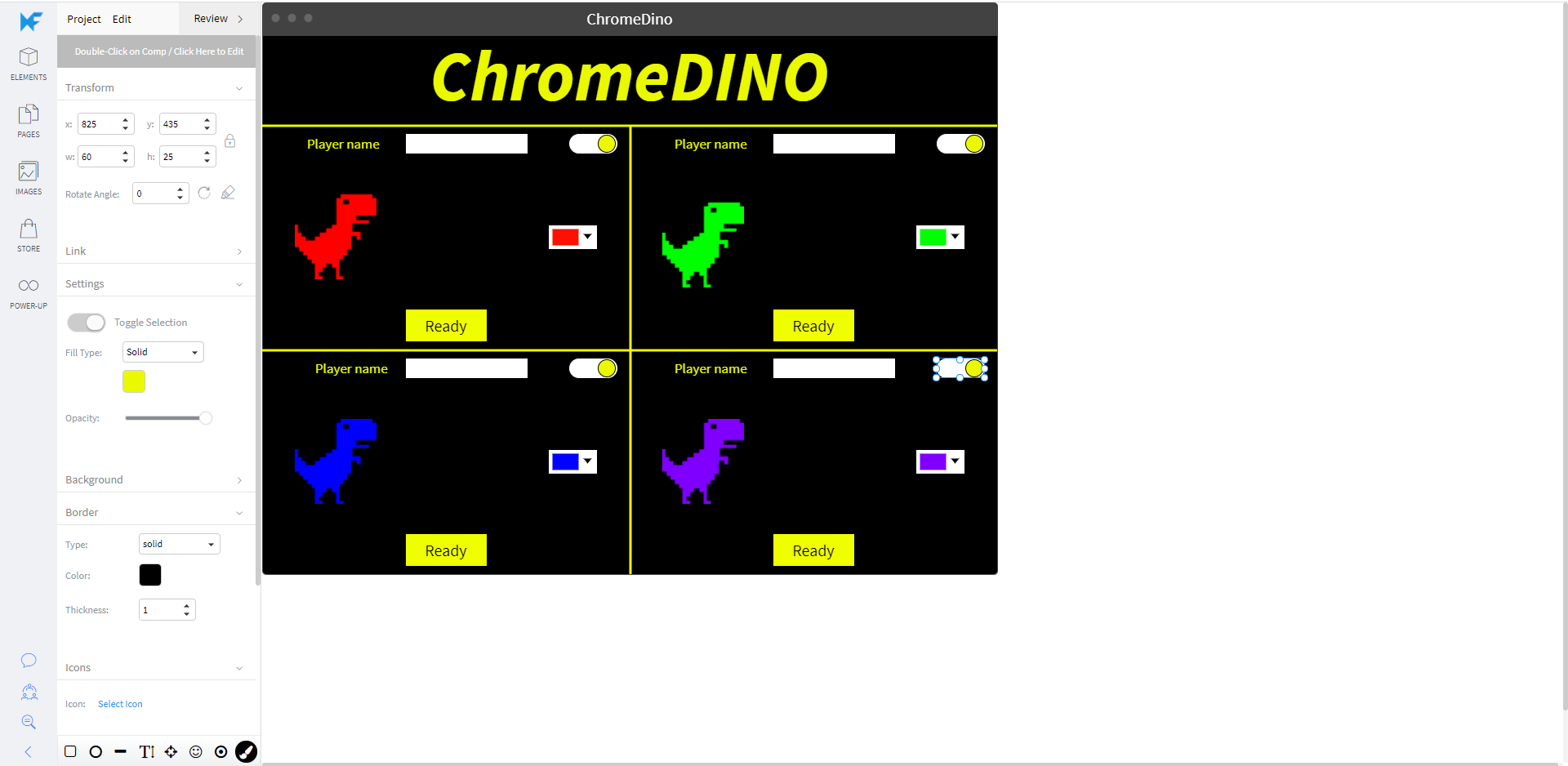
Ram: 16GB

# Progettazione

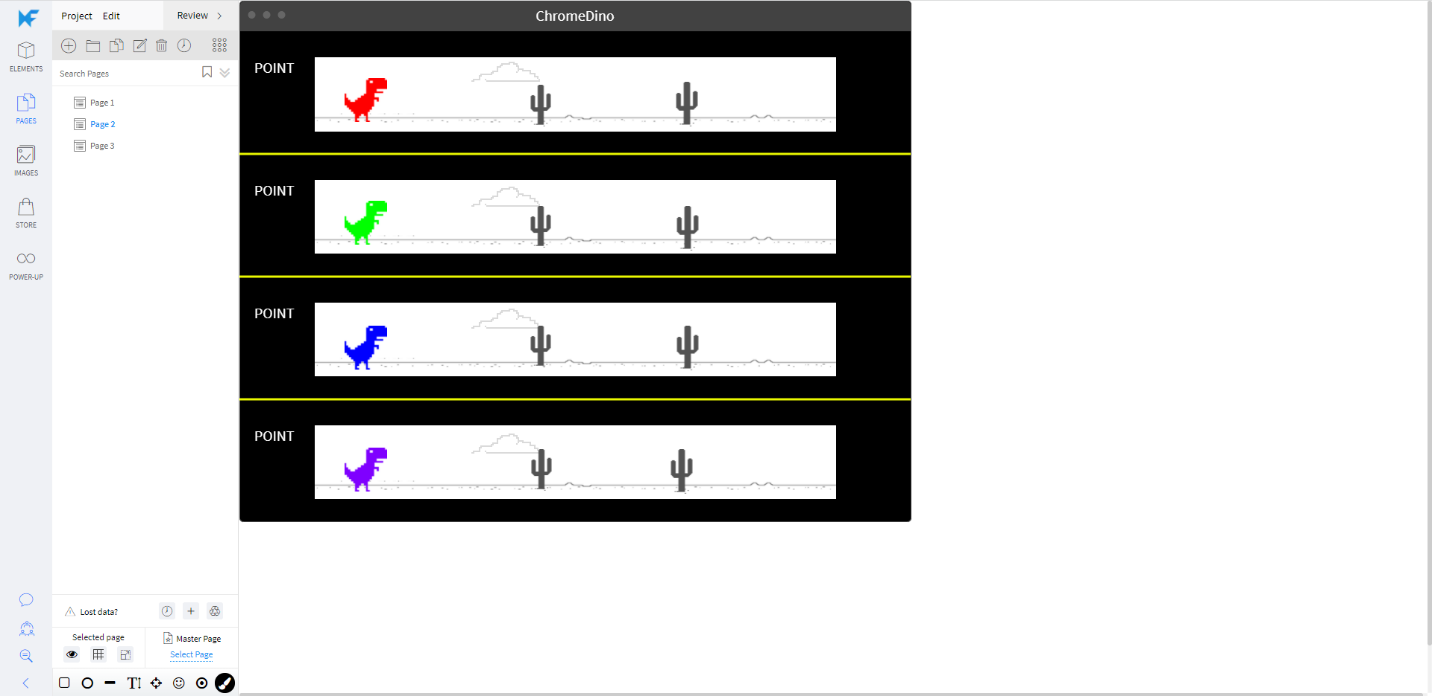
## Design delle interfacce

### Design Iniziale

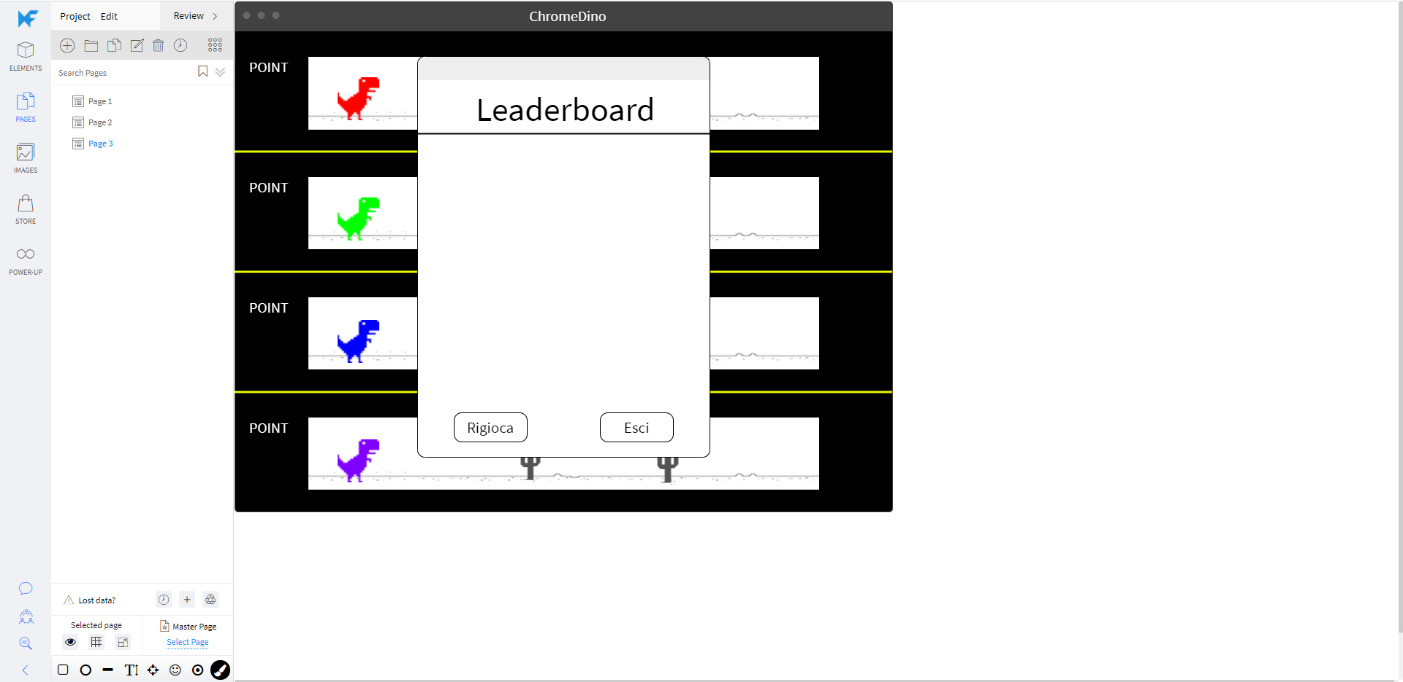
Interfaccia di personalizzazione



Interfaccia di gioco

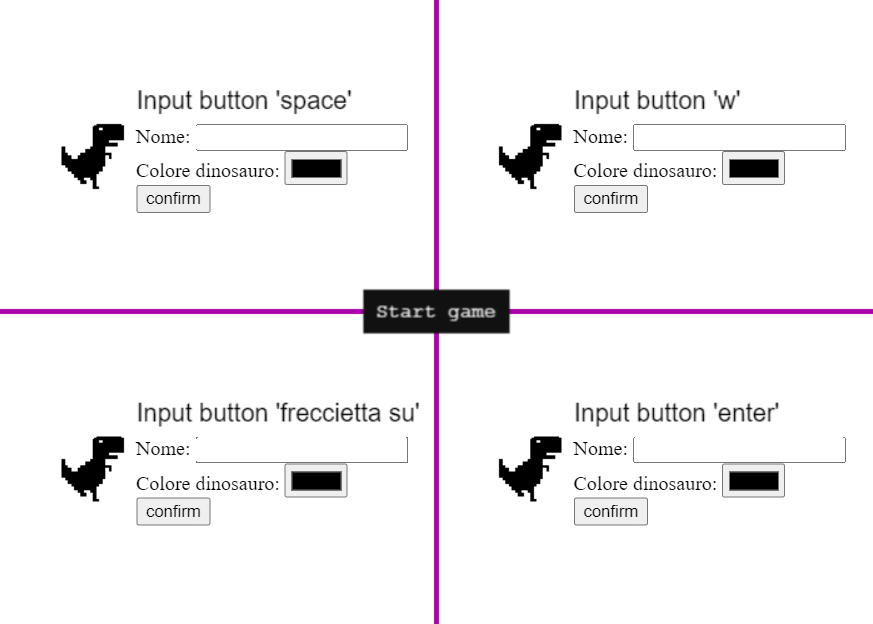


Interfaccia Leaderboard

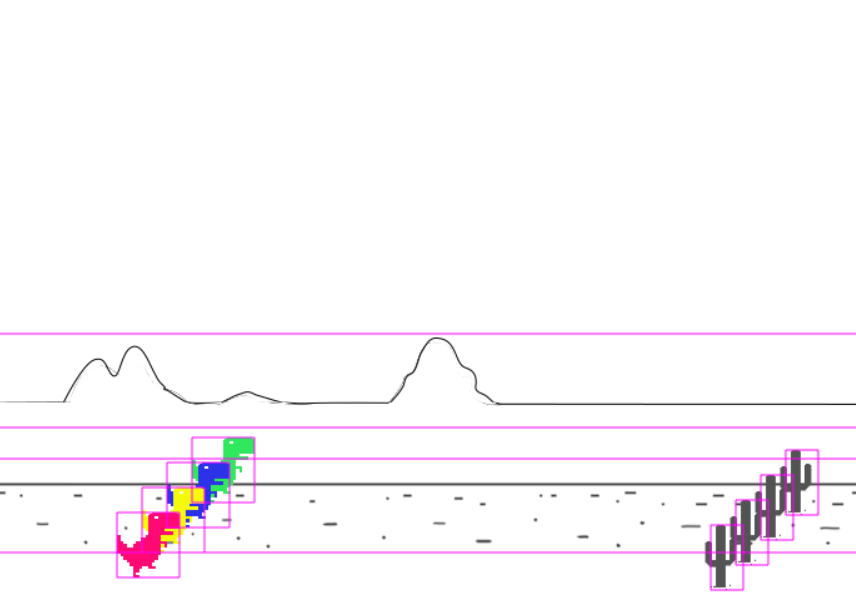


### Design Finale

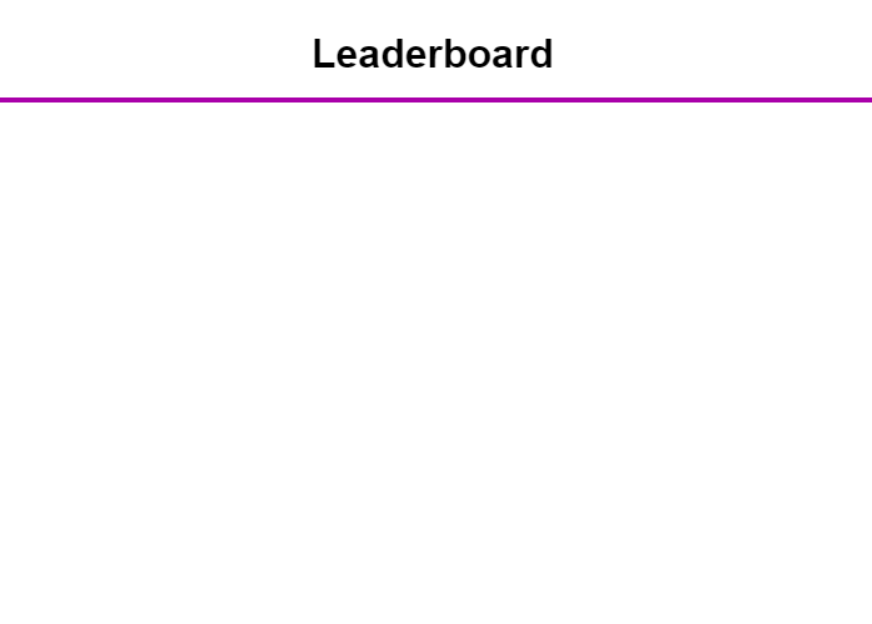
Interfaccia di personalizzazione



Interfaccia di gioco



Interfaccia Leaderboard



# Implementazione

## Utilizzo di Phaser3

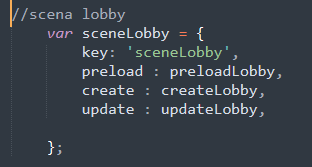
Per creare questo gioco ho utilizzato un framework di nome Phaser3 .

Per utilizzare tutte le funzionalità di questo framework ho dovuto importarlo nel codice, nelmio caso i file gli ho messi nella cartella js.



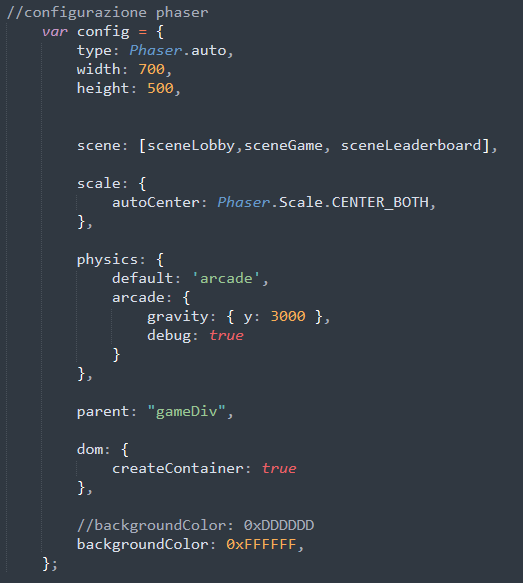
## Creazione scene

Una volta importato correttamente i file per utilizzare phaser bisogerà creare le scene desiderate, nel mio caso la scena per la lobby quella per il gioco e la leaderboard finale. Ognuna di queste scene deve avere una key per essere riconosciuta e deve dichiarare i metodi preload, create e update.



Questa operazione va fatta per tutte le scene.

## Configurazione del gioco

In una variabile mettiamo tutta la configurazione per il nostro gioco.

### Parametri più importanti

Scene: È l’ordine con la quale le scene vengono visualizzate nel gioco.

Physics: attiva la possibilità di dare una fisica agli oggetti come applicargli la gravità.

Parent: Il parent è un div che bisogna posizionare nel body per riuscire a prendere gli input da un file html.

Dom: stessa cosa del parent, crea un contenitore per l’html.

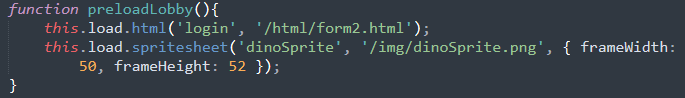
## Creazione del gioco

Per creare il canvas nella quale andremo a creare il nostro gioco passandogli la variabile di configurazione.

## Scena Lobby

### preloadLobby()

la funzione preloadLobby come tutti i preload serve a caricare tutti gli assets che andremo ad utilizzare nella scena lobby.

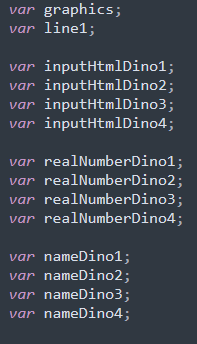
Nella scena Lobby ho importato il file html da dove ho preso gli input e lo spritesheet del dinosauro.

Per importarli bisisogna specificare sempre id e percorso del file.

Per lo spritesheet anche a dove inizia e dove fisce un immagine.

### Variabili globali scena lobby

Queste sono le variabili che ho dichiarato globalmente ma che vengono usate quasi solo in lobby.



#### Lines e Graphics

Queste vengono utilizzate solo per dividere il canvas i 4 parti per i 4 giocatori.

#### InputHtmlDino

Le variabili InputHtmlDino sono 4, una per ogni dino.

Queste vengono utilizzate per mettere nel canvas il file html dalla quale verrano presi gli input.

#### realNumberDino

Le variabili realNumberDino sono 4, una per ogni dino.

In queste variabili ci ho salvato il valore dei colori selezionati dalutente, sono state dichiarate globalmente perche vengono riutilizzate nella scena game.

#### NameDino

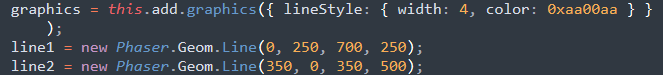
Le variabili nameDino sono 4, una per ogni dino.

In queste variabili ho salvato i nomi impostati dagli utenti oppure Player1 se l’utente non inserisce niente, sono state dichiarate globalmente per poterle riutilizzare nella scena leaderboard.

### createLobby()

La funzione createLobby serve per posizionare in maniera grafica tutto quello che ci serve nel canvas.

#### Linee di separazione

Con questo codice mettiamo le linee nel canvas ma per visualizzarle bisognerà fare ancora un passaggio nella funzione update.

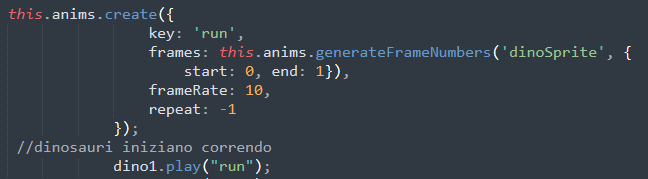
Quando creiamo una nuova linea c’è da passare (x0,y0,x1,y1)

#### Dinosauri

Per inserire i dinosauri bisogna specificare la x, la y e l’id messo nel preloadLobby()

il setOrigin posizione l’origine dell’immagine nell’angolo alto a sinistra.

Tutti i dinosauri che ho posizionato nella lobby hanno l’animazione di corsa, questo l’ho fatto creando un animazione e poi facendola partire.



key: è utilizzata per poi poter richiamare la animazione.

frames: specifica qual’è lo spritesheet da cambiare e fino dove deve arrivare

ramerate: Ogni quanto deve cambiare frame.

# Test

## Protocollo di test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-01 | **Nome:** | Multiplayer dallo stesso PC |
| **Descrizione:** | Tutti i giocatori possono giocare contemporaneamente dallo stesso PC. | | |
| **Prerequisiti:** | Interfaccia di gioco iniziale, scelta numero di giocatori e avvio partita completati. | | |
| **Procedura:** | 1. Inserire il numero di giocatori che parteciperanno alla partita. 2. Personalizzare il colore del dinosauro per ogni giocatore. 3. Avviare partita quando tutti sono pronti. 4. Controllare che ogni giocatore può giocare. | | |
| **Risultati attesi:** | Tutti i giocatori possono svolgere la loro partita | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-002  REQ-03 | **Nome:** | Tutti i dinosauri presenti a schermo |
| **Descrizione:** | Una volta avviata la partita tutti i dinosauri devono essere presenti a schermo con i rispettivi campi di gioco. | | |
| **Prerequisiti:** | Interfaccia di gioco iniziale, scelta numero di giocatori e avvio partita completati. | | |
| **Procedura:** | 1. Inserire il numero di giocatori che parteciperanno alla partita. 2. Personalizzare il colore del dinosauro per ogni giocatore. 3. Avviare partita quando tutti sono pronti. 4. Controllare che tutti i dinosauri con i rispettivi campi di gioco siano visibili a schermo. | | |
| **Risultati attesi:** | Numero di campi di gioco uguale al numero di giocatori. | | |

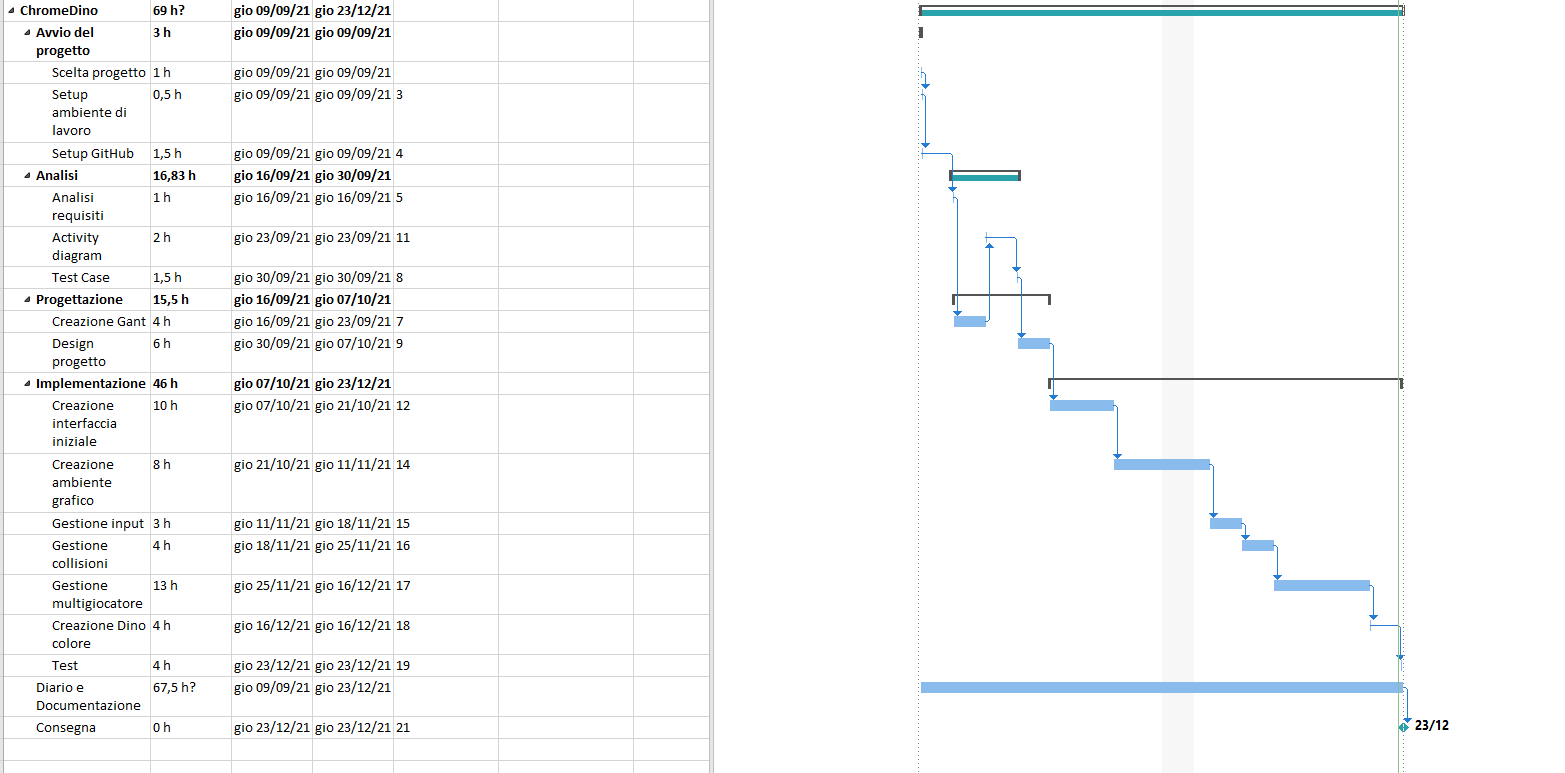
|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-003  REQ-04 | **Nome:** | Personalizzare il nome e il colore del dinosauro |
| **Descrizione:** | Tutti i giocatori prima dell’avvio della partita possono personalizzare nome e colore del dinosauro. | | |
| **Prerequisiti:** | Interfaccia di gioco iniziale, scelta numero di giocatori e avvio partita completati. | | |
| **Procedura:** | 1. Inserire il numero di giocatori che parteciperanno alla partita. 2. Personalizzare il colore del dinosauro e eventualmente il nome per ogni giocatore. 3. Avviare partita quando tutti sono pronti. 4. Controllare che tutti i dinosauri abbiano nome e colore corretto. | | |
| **Risultati attesi:** | I nomi e i colori dei dinosauri sono uguali a quelli scelti prima del avvio della partita. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-004  REQ-05 | **Nome:** | Input |
| **Descrizione:** | Tutti i giocatori hanno i loro input | | |
| **Prerequisiti:** | Interfaccia di gioco iniziale, scelta numero di giocatori e avvio partita completati. | | |
| **Procedura:** | 1. Inserire il numero di giocatori che parteciperanno alla partita. 2. Personalizzare il colore del dinosauro e eventualmente il nome per ogni giocatore. 3. Avviare partita quando tutti sono pronti. 4. Controllare che tutti i dinosauri ricevono i comandi corretti dalla tastiera. | | |
| **Risultati attesi:** | Ogni giocatore può abbassarsi e alzarsi in base ai suoi input. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-005  REQ-06 | **Nome:** | Sfondi in parallasse |
| **Descrizione:** | Tutti i campi di gioco devono avere i due sfondi in parallasse, il terreno deve essere più veloce del cielo. | | |
| **Prerequisiti:** | Interfaccia di gioco iniziale, scelta numero di giocatori e avvio partita completati. | | |
| **Procedura:** | 1. Inserire il numero di giocatori che parteciperanno alla partita. 2. Personalizzare il colore del dinosauro e eventualmente il nome per ogni giocatore. 3. Avviare partita quando tutti sono pronti. 4. Controllare che i due sfondi si muovano correttamente. | | |
| **Risultati attesi:** | Il terreno si deve muo | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-006  REQ-08 | **Nome:** | Highscore |
| **Descrizione:** | Alla fine di ogni partita deve apparire la top 20 degli Highscore | | |
| **Prerequisiti:** | Tutta la parte di gioco deve essere completata. | | |
| **Procedura:** | 1. Inserire il numero di giocatori che parteciperanno alla partita. 2. Personalizzare il colore del dinosauro e eventualmente il nome per ogni giocatore. 3. Avviare partita quando tutti sono pronti. 4. Finire la partita 5. Controllare che appaia correttamente la top 20 | | |
| **Risultati attesi:** | Top 20 highscore che appare a schermo. | | |

# Consuntivo



# Conclusioni

## Conclusioni

In conclusione posso dire che mi sono divertito a fare questo progetto perché era un progetto che mi dava stimoli e che facevo volentieri.

Non è stato facile imparare ad usare un Framework ma una volta capito mi ha semplificato di gran lunga la vita.

## Sviluppi futuri

Gli sviluppi che vorrei apportare in futuro sono rendere disponibile la leaderboard e la possibilità di fare un rematch con gli stessi nomi e dinosauri. Inoltre vorrei implementare la possibilità di scegliere in quanti giocatori giocare.

# Bibliografia

## Sitografia

<https://phaser.io/tutorials/making-your-first-phaser-3-game/part1> , 2021.10.07

<https://codeburst.io/create-a-secret-dino-chrome-game-in-1-hour-with-js-and-phaser-3-2caebb1abe2a>, 2021.10.07

<https://shakuro.com/blog/phaser-js-a-step-by-step-tutorial-on-making-a-phaser-3-game/> , 2021.10.14

<https://rexrainbow.github.io/phaser3-rex-notes/docs/site/> , 2021.10.14

<https://labs.phaser.io/> , 2021.10.21

<https://photonstorm.github.io/>, 2021.10.21

<https://www.w3schools.com/tags/att_input_type_color.asp> 2021.11.18

<https://phaser.io/examples/v3/view/textures/edit-texture-silhouette> 2021.11.18

<https://phaser.io/examples/v3/view/animation/create-animation-from-sprite-sheet> 2021.12.02

<https://phaser.io/examples/v3/view/scenes/swapping-scenes> 2021.12.02

<https://phaser.io/examples/v3/view/scenes/passing-data-to-a-scene> 2021.12.02

<https://newdocs.phaser.io/docs/3.55.2/focus/Phaser.GameObjects.DOMElement-createFromCache> 2021.12.09

<https://newdocs.phaser.io/docs/3.55.2/Phaser.GameObjects.DOMElement> 2021.12.09

<https://labs.phaser.io/index.html?dir=game%20objects/dom%20element/&q>= 2021.12.16

<https://phaser.discourse.group/t/adding-html-elements/278/2> 2021.12.16

<https://phaser.io/news/2019/07/html-buttons-in-phaser-games> 2021.12.16

<https://photonstorm.github.io/phaser3-docs/Phaser.GameObjects.DOMElement.html> 2021.12.16

.

# Allegati

* Diari
* Activity Diagram
* Design
* Gant
* QDC
* Src