Chrome Dino

**Sommario**

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

2.3 Use case 6

2.4 Pianificazione 1

2.4.1 Analisi 1

2.4.2 Progettazione 1

2.4.3 Sviluppo / Implementazione 1

2.4.4 Test e collaudo 1

2.4.5 Documentazione 1

2.4.6 Consegna 1

2.5 Analisi dei mezzi 2

2.5.1 Software 2

2.5.2 Hardware 2

3 Progettazione 3

3.1 Design dell’architettura del sistema 3

3.2 Design dei dati e database 3

3.3 Design delle interfacce 4

3.3.1 Main Menu 5

3.3.2 Game 6

3.3.3 Ranking 7

3.4 Design procedurale 7

4 Implementazione 10

4.1 Main Menu 10

4.1.1 Script 10

4.2 Gioco 10

4.2.1 Creazione del terreno 11

4.2.2 Creazione sfondi in parallasse 11

4.2.3 Creazione del giocatore 11

4.2.4 Gestione della multi-giocabilità 11

4.2.5 Scritte e punteggi 11

4.3 Classifica 12

4.4 Ultime modifiche e build del gioco 12

5 Test 13

5.1 Protocollo di test 13

5.2 Risultati test 13

5.3 Mancanze/limitazioni conosciute 13

6 Consuntivo 1

7 Conclusioni 1

7.1 Sviluppi futuri 1

7.2 Considerazioni personali 1

8 Bibliografia 1

8.1 Bibliografia per articoli di riviste 1

8.2 Bibliografia per libri 1

8.3 Sitografia 1

9 Allegati 1

# Introduzione

## Informazioni sul progetto

Allievo coinvolto nel progetto: Sara Bressan

Classe: 3AC della sezione Informatica presso la Scuola Arti e Mestieri di Trevano

Docente responsabile: Geo Petrini

Data di inizio: 2020-09-03

Data di consegna: 2020-09-17

## Abstract

Vuoi divertirti da solo o con degli amici? Crome Dino potrebbe essere ciò che stai cercando, questo progetto consiste in una versione hotseat del gioco del dinosauro di Chrome. Puoi giocare a questo gioco da solo, oppure giocare fino a 4 giocatori tutti sullo stesso computer, ogni giocatore può giocare con un solo tasto e personalizzare il proprio dinosauro. Ogni giocatore sarà in grado di dare qualsiasi nome al proprio dinosauro e se risulterà tra i migliori apparirà nella classifica Top 20 visibile alla fine del gioco.

“Do you want to have fun alone or with friends? Chrome Dino Hotseat might be for you, this project consists of a hotseat version of chrome's dinosaur game. With this game you can play alone, or you can play up to 4 players on the same computer, each player can simply play with a button and can customize their dinosaur.

Each player will be able to write any name to their dinosaur and if it is among the best it will appear in the Top 20 ranking visible at the end of the game.”

## Scopo

Lo scopo del progetto “Chrome Dino Hotseat” consiste nell’ottenere un versione hotseat, cioè una versione multiplayer locale, del gioco Chrome Dino, il gioco di Chrome che si avvia solitamente in assenza di una connessione ad Internet.

Il progetto sarà standalone, quindi non avrà la necessità di essere eseguito all’interno di un browser come il gioco originale, inoltre il gioco sarà munito di classifica contenente i 20 migliori risultati i quali potranno in seguito essere resettati.

I comandi di gioco saranno molteplici e tutti i giocatori potranno giocare contemporaneamente sulla stessa macchina, i loro dinosauri, i quali avranno un nome e un aspetto personalizzabile, saranno visibili per tutta la durata del gioco.

Saranno presenti due sfondi in parallasse e la creazione del terreno sarà infinita.

# Analisi

## Analisi del dominio

Questo progetto ha come obbiettivo la realizzazione di una versione hotseat del gioco Chrome Dino, questa versione deve aver la possibilità di essere giocata in modalità multigiocatore locale.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-1** | |
| **Nome** | Il gioco deve essere una versione di Chrome Dino |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il gioco deve essere multigiocatore hotseat |

|  |  |
| --- | --- |
| **ID: REQ-2** | |
| **Nome** | Possibilità di avere più giocatori contemporaneamente |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Il numero di giocatori minimo è 1 |
| **002** | Il numero massimo di giocatori deve essere almeno 4 |
| **003** | Tutti dinosauri devono essere visibili contemporaneamente per tutta la durata del gioco, lo schermo viene diviso in base al numero di giocatori |

|  |  |
| --- | --- |
| **ID: REQ-3** | |
| **Nome** | La grafica deve essere 8-Bit |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | La grafica può essere semplice |

|  |  |
| --- | --- |
| **ID: REQ-4** | |
| **Nome** | I Dinosauri devono essere personalizzabili |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Si può cambiare il colore del dinosauro |
| **002** | Si può cambiare la forma del dinosauro (mantenendo delle grandezze standard) |
| **003** | Il giocatore può inserire il proprio nome, se il nome non viene inserito viene messo un nome di default (“Dino” + numero del giocatore) |

|  |  |
| --- | --- |
| **ID: REQ-5** | |
| **Nome** | I giocatori possono usare molteplici controlli di input |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Basta un tasto per giocatore |

|  |  |
| --- | --- |
| **ID: REQ-6** | |
| **Nome** | Lo sfondo deve avere due livelli in parallasse |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | La generazione del terreno deve essere infinita |

|  |  |
| --- | --- |
| **ID: REQ-7** | |
| **Nome** | Il gioco deve essere standalone |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-8** | |
| **Nome** | Il gioco deve avere una classifica dei punteggi migliori |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Si devono vedere gli ultimi 20 migliori risultati |
| **002** | Accanto a un risultato ci deve essere il nome del giocatore |
| **003** | Tutti i valori sono resettabili |

## Use case

L’utente potrà giocare facendo semplicemente partire il gioco (ChromeDinoHotseat.exe), in seguito gli verrà chiesto di selezionare il numero di giocatori e avrà la possibilità di personalizzare i dinosauri.  
Per far saltare i dinosauri il giocatore potrà semplicemente utilizzare il tasto a lui dedicato, il gioco terminerà automaticamente una volta che tutti giocatori avranno fatto Game Over, a quel punto sarà possibile vedere la classifica contenente i migliori 20 giocatori. La classifica sarà resettabile, in alternativa si potrà riavviare il gioco oppure uscire dal gioco tramite il tasto “esc”.

## Pianificazione

Figura 1 - GANTT Preventivo

### Analisi

La fase di analisi consiste in una serie di attività che sono state svolte all’inizio del progetto, prima dell’implementazione e della pianificazione.

In questa fase consiste nel capire in cosa consiste il progetto, quali sono i suoi obbiettivi e quali sono i requisiti del cliente.

Durante questa fase sono state inoltre effettuate al docente responsabile eventuali domande e chiarimenti sul progetto.

### Progettazione

La progettazione consiste nella scelta dei mezzi e la creazione del Gantt preventivo, inoltre è stato fatto il design del progetto.

#### Design

Il design del progetto consiste nel design dell’architettura del sistema, nel design del salvataggio di dati, il design delle interfacce e il design procedurale che verranno descritti nei capitoli seguenti.

### Sviluppo / Implementazione

L’implementazione del progetto sarà la fase più lunga del progetto, siccome è la parte in cui il gioco verrà creato, inoltre questa parte non può essere eseguita bene se la progettazione non è stata fatta al meglio.

### Test e collaudo

I test verranno eseguiti durante lo sviluppo, soprattutto dopo attività importanti per il progetto, i test verranno documentati nell’apposito capitolo di questa documentazione

### Documentazione

La documentazione verrà fatta durante tutto lo sviluppo del progetto e l’andamento del lavoro verrà documentato tramite i diari di lavoro.

### Consegna

La consegna verrà eseguita giovedì 17 dicembre 2020 via GitHub.

## Analisi dei mezzi

Il gioco è stato creato con Unity, siccome Unity non era intallato sulle macchine scolastiche ho dovuto creare una macchina virtuale con Windows 10 utilizzando VMWare Worstation Pro.

Per effettuare la documentazione è stato utilizzato Windows 10 e MS Project, mentre per i diagrammi è stato utilizzato l’applicativo web Draw.io.

### Software

* Microsoft Word
* Microsoft Project
* Macchina Virtuale VM Ware con Windows 10 x64
* Unity versione 2017.4.26f1
* Visual Studio 2017
* Draw.io (<https://draw.io>)
* GitHub

### Hardware

* PC Scolastico (A423-08)
* SSD

# Progettazione

## Design dell’architettura del sistema

Il gioco sarà composto da 3 Scene principali:

* La pagina del menù la quale conterrà il menù tramite il quale modificare le impostazioni di gioco;
* La pagina principale del gioco che conterrà il gioco in sé e utilizzerà le impostazioni create nel menù;
* La pagina che conterrà la classifica dei migliori 20 giocatori e tramite la quale sarà possibile riavviare il gioco o cancellare la classifica.

Queste tre Scene saranno collegate tra loro, infatti dalla prima pagina si potrà accedere alla seconda facendo partire il gioco, dalla seconda si andrà alla terza una volta che tutti i giocatori saranno sconfitti e dalla terza Scene si potrà andare alla prima riavviando il gioco.

Figura 2 - Design

## Design dei dati e database

Le informazioni dei giocatori, le quali impostate dal menù di gioco, verranno convertite in binario e salvate su un file, in seguito questo file verrà letto durante la fase di gioco in modo che i giocatori abbiano le impostazioni desiderate (colore e forma del dinosauro, nome del giocatore, numero di giocatori e punteggio), il file con i dati dei giocatori verrà eliminato al termine della partita.

Figura 3 - Diagramma di salvataggio dei dati dei giocatori

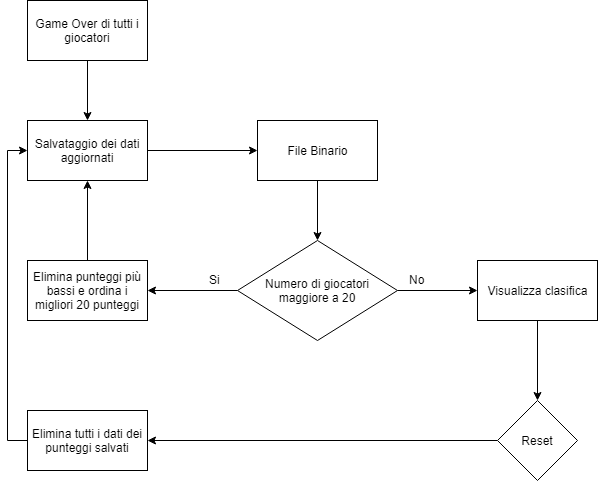
I punteggi dei giocatori verranno salvati su un file binario quando tutti i giocatori perdono la vita, una volta aperta la classifica il file contenente i punteggi e i nomi dei giocatori verrà letto e coloro che non sono tra i primi 20 verranno eliminati dalla classifica, il file con i punteggi viene eliminato solo quando si effettuerà un reset dei punteggi, altrimenti rimane accessibile a tutte le partite seguenti.

Figura 4 - Diagramma di utilizzo dei dati dei giocatori

Figura 5 - Diagramma di salvataggio dei punteggi

## Design delle interfacce

Il gioco sarà munito di 3 scene, la prima scena (la prima ad apparire all’avvio del gioco) consisterà nel menù di gioco nella quale sarà possibile settare le impostazioni del gioco e modificare i dinosauri dei giocatori.

La seconda scena consiste nel gioco nella quale si potranno vedere tutti i giocatori contemporaneamente, mentre nell’ultima scena, la quale verrà visualizzata quando tutti giocatori andranno in Game Over, consiste nella classifica dei 20 migliori giocatori.

### Main Menu

Il menu avrà un bottone “Start” per far partire il gioco e il numero di giocatori sarà gestito tramite uno slider.

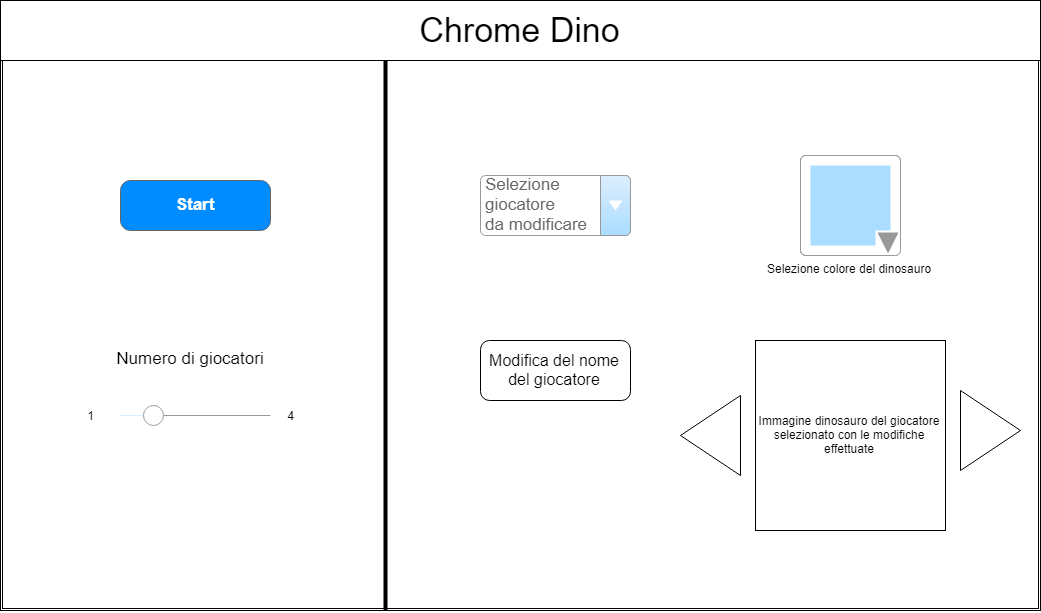
Tramite il menu sarà inoltre possibile selezionare il nome del giocatore e modificarlo, oltre a che cambiare il colore del dinosauro con un color picker e cambiare la forma del dinosauro (usare le frecce), le impostazioni inserite in questa scena verranno salvate per le altre scene.

Figura 6 - Main Menù

### Game

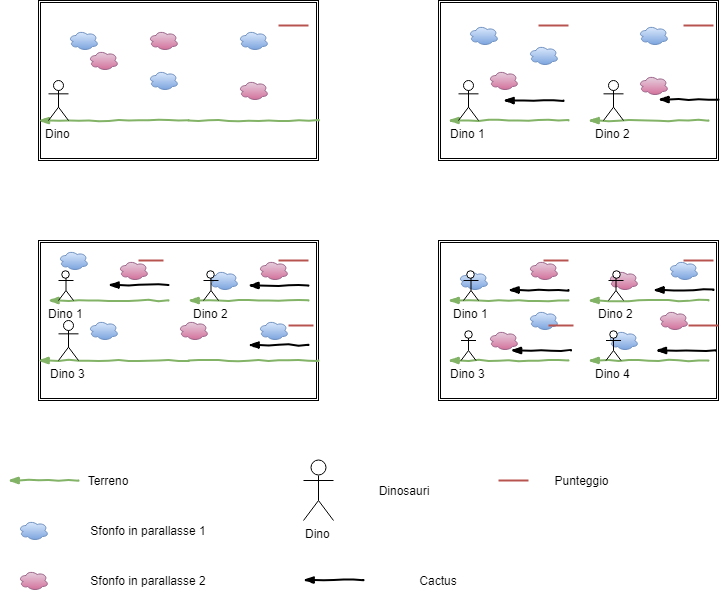


Figura 7 – Game

Una volta partito il gioco i giocatori potranno vedere tutti i dinosauri in gioco temporaneamente, la scena verrà creata in base al numero di giocatori indicati nel menù.

Il terreno scorrerà all’infinito, mentre il dinosauro rimarrà fermo orizzontalmente e potrà soltanto saltare.

Ogni giocatore potrà vedere il suo punteggio in tempo reale e potrà riconoscere il proprio dinosauro leggendo il label con il nome deciso in precedenza.

Saranno presenti, inoltre, due sfondi in parallasse con scorrimento orizzontale (nuvole), i due sfondi avranno velocità diverse e i cactus, i quali verranno creati casualmente, si muoveranno con il terreno.

### Ranking

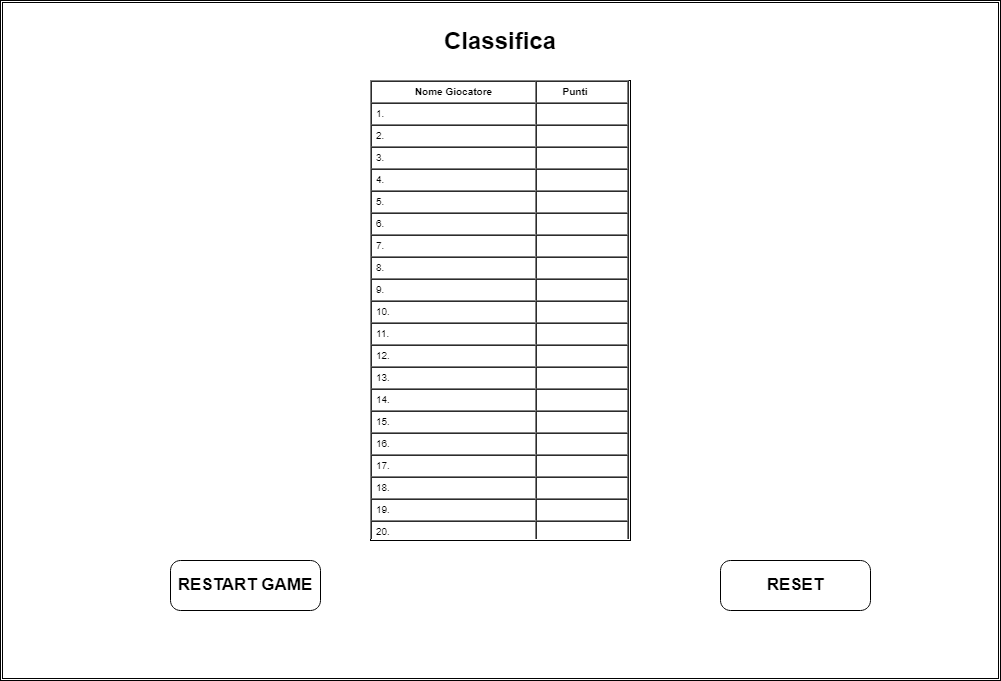


Figura 8 – Classifica

Quando tutti i giocatori avranno perso si aprirà la scena con la classifica, questa scena conterrà una tabella contenente i nomi dei migliori 20 giocatori e i loro punteggi.

Tramite questa schermata sarà possibile riavviare il gioco oppure azzerare i punteggi cliccando sugli appositi bottoni.

## Design procedurale

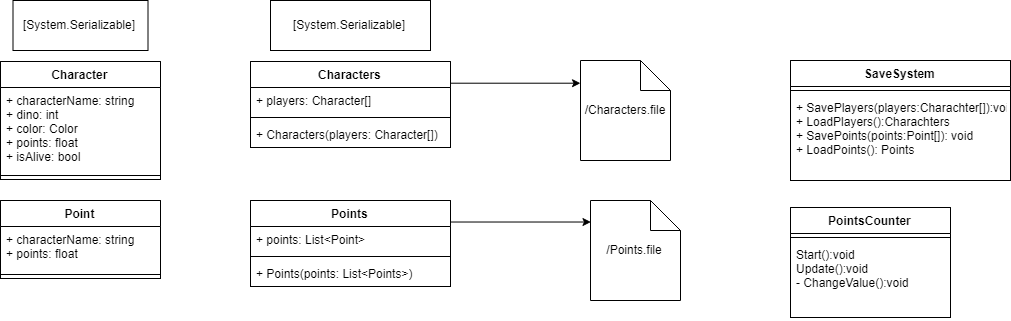


Figura 9 - Classi e salvataggio su file

La classe SaveSystem serve a salvare sui file i dati dei giocatori e i punteggi, a questo scopo gli oggetti da salvare saranno Serializzabili (utile per trasformare in binario le informazioni).

SaveSystem avrà quindi un metodo che serve a salvare e un metodo per ottenere le informazioni dei personaggi e i punteggi.

PointsCounter servirà invece ad aggiornare i punteggi durante l’avanzamento del gioco e a caricare la classifica una volta che tutti i dinosauri saranno morti, inoltre aggiornerà la lista dei punteggi e se ci saranno più di 20 punteggi nel file eliminerà i peggiori, infine salverà i dati dei punteggi aggiornati sul file.

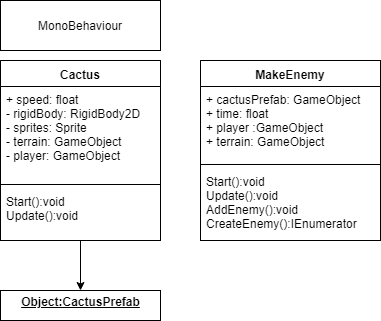


Figura 10 – Cactus

I cactus verranno creati a intervalli di tempo casuali da MakeEnemy, essi verranno eliminati una volta usciti dall’area di terreno di gioco e si fermeranno in caso di eliminazione di un giocatore.

I cactus saranno muniti di RigidBody e saranno sensibili alle collisioni con i dinosauri.

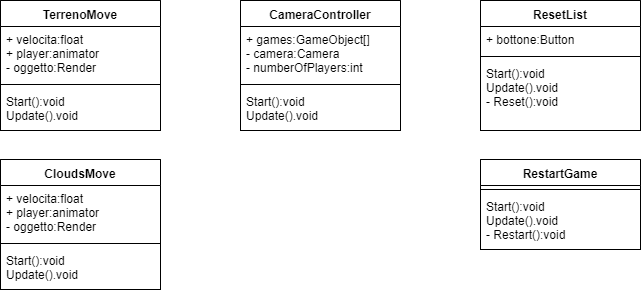


Figura 11 – Classi

Il terreno e i due sfondi in parallasse utilizzeranno due metodi simili per muoversi orizzontalmente e si fermeranno nel caso i dinosauri dovessero essere eliminati.

CameraController gestirà la telecamera alla partenza del gioco, cambiando la visuale in base al numero di giocatori.

ResetList servirà a resettare la classifica dei punteggi se cliccato il bottone di reset, mentre RestartGame servirà per far ripartire il gioco.

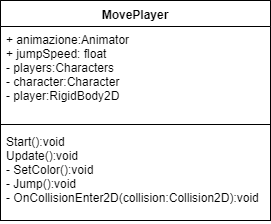


Figura 12 - Move Player

La classe MovePlayer si occuperà di gestire il giocatore, creerà il dinosauro in base a le caratteristiche scelte del Menù, controllerà le collisioni del dinosauro con gli ostacoli e lo farà saltare o morire, inoltre gestirà il cambio di animazione a seconda dello stato del dinosauro.

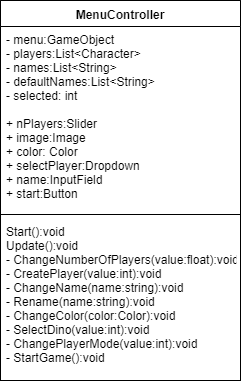


Figura 13 – MenuController

La classe MenuController si occuperà invece di gestire le impostazioni inserite dall’utente nel menù, inoltre salverà queste informazioni sul file contenente le impostazioni di gioco e farà partire il gioco.

# Implementazione

Per prima cosa è stata creata la macchina virtuale con VMWare Workstation Pro, questa macchina virtuale lavora con un sistema operativo Windows 10, ha 3 GB di RAM.

La ISO di Windows 10 che avevo inizialmente non era funzionante, di conseguenza un mio compagno di classe mi ha passato un’altra ISO.

Durante la durata di tutto il progetto alla fine di ogni lezione è stato effettuato uno snapshot della macchina virtuale.

Sulla macchina virtuale è stato installato Unity e ho creato un nuovo progetto 2D nominato “ChromeDinoHotseat”.

## Main Menu

Per prima cosa ho creato una nuova scena chiamata “Menu” dentro il quale ho creato un pannello che funge da contenitore per tutti gli altri componenti della pagina.

In seguito, ho creato il pannello “Comandi” all’interno della quale ho inserito il bottone per far partire il gioco e uno slider per il controllo del numero di giocatori.

In un altro pannello ho messo il menù nella quale i giocatori possono modificare il proprio dinosauro.

Questo pannello contiene a sua volta due nuovi pannelli:

* Il primo pannello contiene un menù a tendina con la quale si potrà selezionare il giocatore da modificare, un TextBox nella quale è possibile modificare il nome del giocatore corrente e un altro menù a tendina per selezionare l’immagine del dinosauro.
* Il secondo pannello contiene tre Slider che servono a selezionare il colore del dinosauro (siccome non esiste un color picker già prefabbricato l’utilizzo dei tre Slider è stato un passo obbligato) e un canvas nella quale verrà mostrata l’immagine del dinosauro e le sue modifiche in tempo reale.

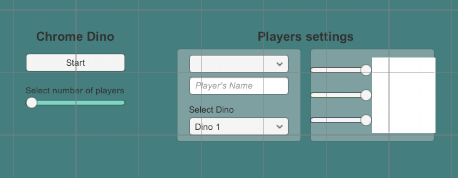


Figura 14 - Grafica del Menù

### Script

Dopo aver creato l’interfaccia grafica ho creato la logica dietro al menù.

#### Menù controller

All’interno del Pannello principale ho creato lo script “Menu Controller” il quale gestirà gli eventi del menù.

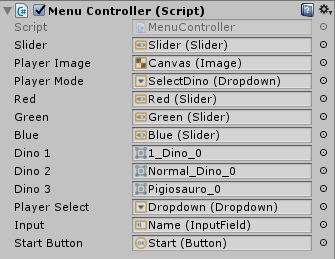


Figura 15 - Menù controller

#### Pannello dei comandi

Il bottone Start non contiene alcuno script, mentre il testo dell’Handle dello Slider del numero di giocatori contiene lo script “Value to Text” che serve a mostrare in tempo reale il numero di giocatori scelto.

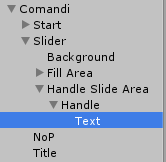


Figura 16 - Pannello dei comandi

NoP corrisponde al label che dice “Select number of players”, mentre title corrisponde al titolo “Chrome Dino” entrambi i label sono statici, quindi al loro interno non ho messo alcuno script.

#### Pannello “PlayerM” – Player Menu

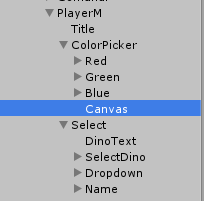


Figura 17 - Player Menù

Il pannello PlayerM non ha nessuno script, mentre come in precedenza i tre slider dei colori contengono lo script “Value to Text” per scrivere il loro valore attuale nell’Handle.

Il canvas (che conterrà l’immagine del dinosauro) ha lo script “Pick Color” così da tenere l’immagine sempre aggiornata per il giocatore selezionato.

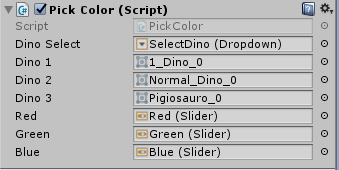


Figura 18 - Pick Color

DinoText essendo un label da non modificare che dice “Select Dino” non contiene alcuno script, mentre nel DropDown “SelectDino” ho inserito i valori assumibili da esso e le immagini ad essi associati (vedi Figura 19).

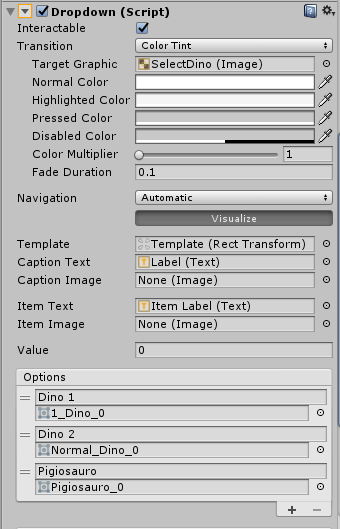


Figura 19 - Dropdown SelectDino

Nell’altro menù a tendina non ho messo niente siccome i suoi valori vengono gestiti dallo script “Menù controller” che si trova nel Panel come detto in precedenza, anche il Text Input “Name” viene gestito dallo stesso script.

## Gioco

Per prima cosa ho cercato su internet degli spunti per disegnare gli sfondi, il terreno e i dinosauri, a quel punto ho deciso che tutti i dinosauri sarebbero stati di dimensioni 44x45 pixel così da avere tutti i dinosauri della stessa dimensione.

Il terreno è stato ridisegnato secondo il terreno del gioco originale (<https://chromedino.com/>), mentre le nuvole sono la stessa nuvola con colori diversi ripetute con dimensioni variabili.

Dopo aver disegnato tutti i dinosauri (il dinosauro originale, il dinosauro che saluta e il pigiosauro) ho creato una nuova scena che conterrà il gioco chiamata “1”.



Figura 20 – Pigiosauro

Una volta creata la scena ho iniziato a creare la grafica del gioco come se ci fosse un solo giocatore: ho quindi creato un GameObject chiamato “Game1”.

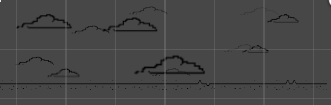


Figura 21 - Game 1

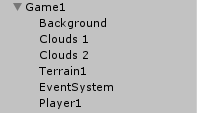


Figura 22 - Struttura Game1

### Creazione del terreno

All’interno di “Game1” ho creato un altro GameObject chiamato “terrain” il quale conterrà il nostro terreno, in questo nuovo terreno ho messo lo script “Terreno Move” che gli consentirà di scorrere orizzontalmente.

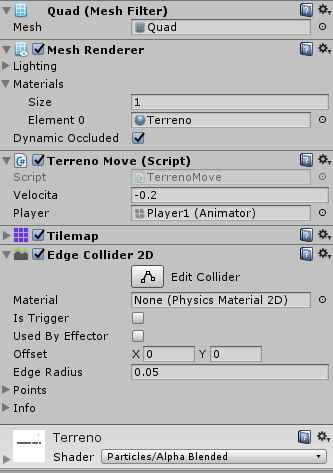


Figura 23 - Terreno1

### Creazione sfondi in parallasse

I due sfondi in parallasse (le nuvole) sono stati creati come il terreno, solo che a questi è stato messo lo script “Clouds Move” che controlla lo spostamento orizzontale dello sfondo, inoltre la velocità di “Clouds 1” è stata settata a -0.01 mentre quella di “Clouds 2” a -0.02.

### Creazione del giocatore

Il giocatore è un GameObject chiamato “Player”, ad esso ho dovuto mettere un Animator così da gestire le animazioni del dinosauro nelle diverse occasioni (ad esempio il cambio di immagine quando esso salta).

“Player” ha inoltre lo script “Move player” che serve a gestire il giocatore, un RigidBody2D e un Boy Collider 2D per la gestione delle collisioni.

“Game1” avrà inoltre lo script “Make Enemy” che serve a creare i nemici.

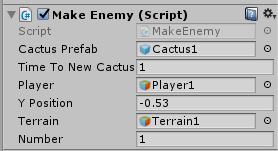


Figura 24 - Make Enemy del Game1

#### Animator

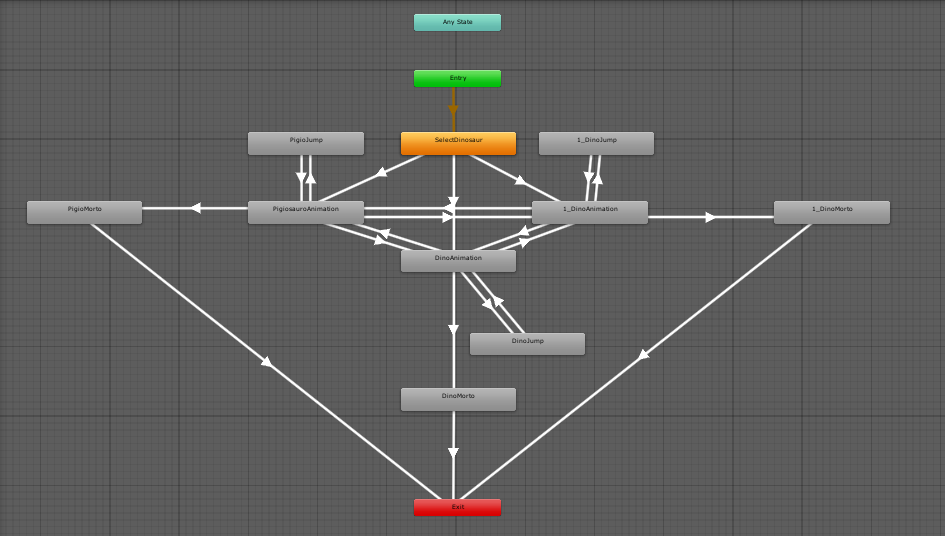


Figura 25 – Animator

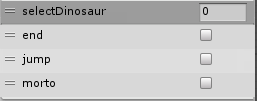


Figura 26 – Parameter

Le animazioni sono gestite attraverso l’Animator, in base ai parametri della Figura 26 il dinosauro cambia animazione, ad esempio se il giocatore decide di utilizzare il pigiosauro, lo script “Move Player” assegnerà a selectDinosaur il valore 0 che corrisponde al Pigiosauro, in seguito se il giocatore deciderà di saltare jump diventerà “true” e l’animazione passerà a “PigioJump” per poi tornare a “PigiosauroAnimation” una volta terminato il salto (il parametro Jump tornerà a false) e così via.

### Gestione della multi-giocabilità

Per gestire la multi-giocabilità ho copiato “Game1” così da avere 4 giochi uguali, in seguito li ho disposti come volevo, i nomi degli oggetti in ogni Game sarà seguito dal numero di gioco.

La gestione dell’aspetto multigiocatore è gestita dallo script “Camera Controller” che ho messo nella Main Camera, questo script eliminerà i Game di troppo.



Figura 27 - 4 Giochi



Figura 28 - Camera Controller

### Scritte e punteggi

I punteggi e i nomi dei giocatori sono stati messi come ultima cosa in un Canvas, all’interno di questo Canvas ho inserito 4 pannelli, uno in alto a sinistra per il giocatore 1, uno in alto a destra per il giocatore 2, uno in basso a sinistra per il giocatore 3 e l’ultimo in basso a destra per il giocatore 4. All’interno di questi pannelli ho messo il Label “Name” e il Label “Points” che conterranno il nome del giocatore e i suoi punti.

Per aggiornare i dati all’interno del label ho messo nel canvas lo script “Info Conroller”.

## Classifica

Per ultima cosa ho creato la terza scena “Ranking” la quale mostrerà la classifica contenente i 20 migliori giocatori.

Questa scena consiste in un Panel nella quale ho messo lo script Reset List dove l’attributo Button è il bottone di reset che ho aggiunto nel pannello.

Nel pannello ho aggiunto pure bottone Restart che serve a far ripartire il gioco, infatti questo bottone ha al suo interno lo script Restart Game.

All’interno del pannello ho inoltre inserito un’immagine con al suo interno il titolo della scena “Score” e un altro pannello chiamato “Scores”.

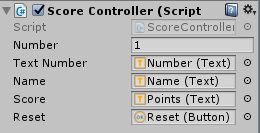
Scores ha un Horizontal Layout così da ottenere tre colonne, nella prima colonna ho messo tutte le righe contenenti i Label dei punteggi, nella seconda colonna il bottone di reset e nell’ultima colonna il bottone di restart del gioco.

Figura 29 - Score Controller del primo in classifica

Ogni riga dei punteggi ha al suo interno 3 Label disposti con un Horizontal Layout, inoltre ogni riga ha al suo interno lo script Score Controller che gestisce i Label.

## Ultime modifiche e build del gioco

Alla fine, ho fatto la build del gioco (le tre scene ordinate) anche se non funzionante.

# Test

## Protocollo di test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-1  REQ-1 | **Nome:** | Il gioco è una versione di Chrome Dino multigiocatore hotseat |
| **Descrizione:** | Verificare che il gioco assomigli a Chrome Dino | | |
| **Prerequisiti:** | Il gioco deve avere una build (file .exe), deve essere quindi terminato. | | |
| **Procedura:** | Fare partire il file ChromeDinoHotseat.exe | | |
| **Risultati attesi:** | Il gioco parta correttamente e si possano mettere più giocatori, tutti i giocatori riescono a giocare. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-2  REQ-2 001 | **Nome:** | Il numero di giocatori minimo è 1 |
| **Descrizione:** | Verificare che il numero di giocatori minimo sia 1. | | |
| **Prerequisiti:** | Il Main Menù deve essere stato creato | | |
| **Procedura:** | * Fare partire la Scena “Menu” * Cambiare il valore dello slider per selezionare il numero di giocatori al minimo | | |
| **Risultati attesi:** | Il valore minimo settabile risulta 1. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-3  REQ-2 002 | **Nome:** | Il numero di giocatori massimo è di almeno 4 giocatori |
| **Descrizione:** | Verificare che il numero di giocatori massimi sia almeno 4. | | |
| **Prerequisiti:** | Il Main Menù deve essere stato creato | | |
| **Procedura:** | * Fare partire la Scena “Menu” * Cambiare il valore dello slider per selezionare il numero di giocatori al massimo | | |
| **Risultati attesi:** | Il valore massimo settabile sia almeno 4. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-4  REQ-2 003 | **Nome:** | Tutti i dinosauri devono essere visibili contemporaneamente per tutta la durata del gioco, lo schermo viene diviso in base al numero di giocatori. |
| **Descrizione:** | Verificare che i giocatori selezionati nel menù siano visibili a schermo | | |
| **Prerequisiti:** | La scena “Menu” e la scena di gioco devono essere ultimati | | |
| **Procedura:** | * Fare partire la Scena “Menu” * Selezionare il numero di giocatori (da 1 a 4) * Modificare i dinosauri e il nome dei giocatori * Far partire il gioco * Giocare * Rifare il test con valori differenti (da 1 a 4 giocatori) | | |
| **Risultati attesi:** | I giocatori richiesti devono apparire a schermo e devono giocare poter contemporaneamente | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-5  REQ-3 | **Nome:** | La grafica deve essere 8-bit / semplice |
| **Descrizione:** | Verificare che la grafica sia semplice e 8-bit | | |
| **Prerequisiti:** | Aver creato la parte grafica | | |
| **Procedura:** | Guardare la grafica | | |
| **Risultati attesi:** | La grafica è 8-bit | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-6  REQ-4 | **Nome:** | I dinosauri devono essere personalizzabili |
| **Descrizione:** | Verificare che il gioco accetti le modifiche dei dinosauri e che queste modifiche si vedano durante il gioco. | | |
| **Prerequisiti:** | Aver creato il menù di gioco e il gioco | | |
| **Procedura:** | * Nel menù selezionare 4 giocatori * Creare il primo personaggio senza nome, blu e che sia pigiosauro * Creare il secondo personaggio chiamandolo P2, rosso e con il dinosauro normale * Creare il terzo personaggio chiamandolo P3, verde e con il dinosauro che saluta * Creare il quarto personaggio senza nome, con il colore di default e il dinosauro di default * Far partire il gioco (“Tasto Start Game”) | | |
| **Risultati attesi:** | * Il primo giocatore si chiama “Dino1”, è blu e l’immagine è di un pigiosauro * Il secondo giocatore si chiama “P2”, è rosso e l’immagine è di un dinosauro normale * Il terzo giocatore si chiama “P3”, è verde e l’immagine è di un dinosauro che saluta * Il quarto giocatore si chiama “Dino4”, è nero su sfondo bianco e l’immagine è di un dinosauro di default | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-7  REQ-5 | **Nome:** | Con un tasto i dinosauri saltano. |
| **Descrizione:** | Verificare che basti un tasto a giocatore per giocare | | |
| **Prerequisiti:** | Aver creato il menù e il gioco | | |
| **Procedura:** | * Creare 4 giocatori * Giocare con i tasti A, Spazio, Freccia in su e il tasto a sinistra del mouse | | |
| **Risultati attesi:** | I giocatori possono giocare con un solo tasto a testa | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-8  REQ-6 | **Nome:** | Ci sono due sfondi in parallasse e il terreno infinito |
| **Descrizione:** | Verificare che gli sfondi e il terreno si muovano, il terreno all’infinito, gli sfondi devono essere in parallasse | | |
| **Prerequisiti:** | Aver creato il gioco (Scena 1) | | |
| **Procedura:** | * Far partire il gioco | | |
| **Risultati attesi:** | La generazione del terreno è infinita (a meno che i dinosauro non muoia) e gli sfondi si muovono in parallasse | | |
| **Test Case:**  **Riferimento**: | TC-9  REQ-7 | **Nome:** | Il gioco deve essere standalone |
| **Descrizione:** | Verificare che il gioco sia standalone | | |
| **Prerequisiti:** | Aver creato la build del gioco (.exe) | | |
| **Procedura:** | * Far partire ChromeDinoHotseat.exe | | |
| **Risultati attesi:** | Il gioco deve partire correttamente ed essere standalone | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-10  REQ-8 | **Nome:** | La classifica deve mostrare i 20 migliori risultati, accanto al risultato ci deve essere il nome del giocatore e i valori sono resettabili. |
| **Descrizione:** | Verificare che classifica mostri i 20 migliori risultati, accanto al risultato ci deve essere il nome del giocatore e i valori sono resettabili. | | |
| **Prerequisiti:** | Aver creato la build del gioco (.exe) | | |
| **Procedura:** | * Giocare 21 volte con nomi diversi * Marcare i giocatori che hanno vinto e i loro punteggi esternamente al gioco * Visualizzare la classifica * Resettare la classifica | | |
| **Risultati attesi:** | La classifica mostrata alla fine deve essere coerente con quella scritta a mano esternamente al gioco.  La classifica si azzera se si resettano i valori | | |

## Risultati test

|  |  |  |
| --- | --- | --- |
| Test Case | Risultato | Annotazioni |
| TC-1 | Non funzionante | Il gioco non è stato terminato, di conseguenza non funziona. |
| TC-2 | Funzionante |  |
| TC-3 | Funzionante |  |
| TC-4 | Non funzionante | Non funziona siccome i dati del file nella quale vengono salvati i dati dei giocatori non viene caricato correttamente. |
| TC-5 | Funzionante |  |
| TC-6 | Non Funzionante | Non funziona siccome i dati del file nella quale vengono salvati i dati dei giocatori non viene caricato correttamente. |
| TC-7 | Non Funzionante | Non funziona siccome i dati del file nella quale vengono salvati i dati dei giocatori non viene caricato correttamente. |
| TC-8 | Funzionante |  |
| TC-9 | Non Funzionante | Non funziona siccome i dati del file nella quale vengono salvati i dati dei giocatori non viene caricato correttamente. Inoltre non vengono caricati i punteggi sul file siccome “Scores” non è serializzabile. |

## Mancanze/limitazioni conosciute

La gestione dei punteggi non è stata terminata e il tempo è stato gestito male, di conseguenza non c’è stato abbastanza tempo per gestire gli errori.

I file con i dati da salvare non vengono letti correttamente (deserializzati), l’errore non è ancora stato risolto.

# Consuntivo

Figura 30 - GANTT Consuntivo

Come si può vedere dal GANTT Consuntivo il progetto non è stato eseguito come pianificato inizialmente, infatti il progetto non è stato terminato.

Errori durante l’ultima fase di implementazione non hanno permesso al gioco di funzionare correttamente.

Il progetto è stato consegnato in ritardo (il 22.12.2020 al posto del 17.12.2020) a causa di una mia assenza per malattia e la mancanza di materiale a casa con la quale proseguire il lavoro.

# Conclusioni

Questo progetto una volta ultimato sarà un divertente passatempo, nel frattempo è stata un’ottima preparazione per i progetti futuri.

## Sviluppi futuri

Nel futuro desidero migliorare graficamente il gioco e soprattutto trovare e correggere gli errori che ne impediscono il funzionamento.

## Considerazioni personali

Durante la creazione di questo progetto ho preso un po' di dimestichezza all’utilizzo di Unity, oltre a che imparare meglio a programmare in C#.

Inoltre ho imparato a gestire meglio il tempo a disposizione.

# Bibliografia

## Bibliografia per articoli di riviste

## Bibliografia per libri

## Sitografia

<https://chromedino.com/>

<https://youtu.be/wBol2xzxCOU>

<https://youtu.be/AqXC-QmhRXQ>

<https://youtu.be/XOjd_qU2Ido>

https://youtu.be/E7gmylDS1C4

https://docs.unity3d.com/Manual/index.html

https://www.sitepoint.com/saving-data-between-scenes-in-unity/

https://www.sitepoint.com/saving-and-loading-player-game-data-in-unity/

# Allegati

Allegati a questo documento sono presenti:

* I diari di lavoro;
* La VM contenente il progetto;
* I codici sorgente;
* Il mandato;
* I GANTT in Project.