

Dino Run and Jump

| | | |
|-------|--|----|
| 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 | 7 |
| 2.4 | Pianificazione | 8 |
| 2.5 | Analisi dei mezzi | 9 |
| 3 | Progettazione | 10 |
| 3.1 | Design dell'architettura del sistema | 10 |
| 3.2 | Design dei dati e database..... | 10 |
| 3.3 | Design delle interfacce..... | 11 |
| 3.3.1 | Design interfacce telefono: | 11 |
| 3.3.2 | Design interfacce computer | 13 |
| 3.4 | Design procedurale | 15 |
| 4 | Implementazione | 16 |
| 5 | Test | 17 |
| 5.1 | Protocollo di test..... | 17 |
| 5.2 | Risultati test..... | 24 |
| 5.3 | Mancanze/limitazioni conosciute..... | 24 |
| 6 | Consuntivo | 25 |
| 7 | Conclusioni..... | 26 |
| 7.1 | Sviluppi futuri..... | 27 |
| 7.2 | Considerazioni personali..... | 27 |
| 7.2.1 | Michea | 27 |
| 7.2.2 | Nadia..... | 27 |
| 7.2.3 | Thomas | 28 |
| 8 | Sitografia | 29 |
| 9 | Allegati..... | 29 |

1 Introduzione

1.1 Informazioni sul progetto

- Allievi coinvolti nel progetto: Michea Colautti, Nadia Fasani, Thomas Sartini.
- Classe: I3AA/BB/BC Scuola Arti e Mestieri Trevano.
- Docenti responsabili: Geo Petrini
- Data inizio: 27 gennaio 2022.
- Data di fine: 05 maggio 2022.
- Linguaggio: JavaScript

1.2 Abstract

We all know the famous Chrome Dino, that little black dinosaur that jumps a lot of cactuses endlessly and tells us that we are not connected to the internet. We've all hated him and loved him at some point in our lives. With this project the Chrome Dino is taken to another level: starting from a previously created project we have improved the user experience and added a new interesting game mechanic. With this project, that's even multiplayer, the players will be able to jump over cacti not by pressing a button, but by jumping into real life. Using the phone's sensors his dinosaur will jump over obstacles. New features have also been added, such as the possibility to customize the dinosaur and earn rewards. But the project is also available to those who cannot, for one reason or another, jump. For this reason, the phone can also be used, via a button, to control your own dinosaur.

1.3 Scopo

Lo scopo del progetto è quello di creare una versione multiplayer del famoso Chrome Dino dove gli utenti si possano connettere ad una partita e possano giocare tutti assieme. Il numero di giocatori è quindi variabile, da un minimo di uno ad un massimo di 10. Ci sarà quindi un utente "host" che si occupa di creare la partita e di mostrarla agli altri utenti, idealmente su uno schermo sufficientemente grande. Man mano che si aggiungono giocatori i loro dinosauri appariranno sullo schermo principale.

Il centro del progetto è però il fatto che è possibile far saltare il proprio dinosauro saltando personalmente. Tramite i sensori di movimento del telefono e alla struttura a sessioni del programma i salti verranno trasmessi al server, che comunicherà alla pagina l'evento. Tuttavia per quelle persone che non possiedono un telefono con i giusti sensori oppure sono impossibilitate nel saltare bisogna introdurre una modalità di gioco basilare, come un click, per saltare.

Gli utenti devono poter inoltre poter creare un account tramite il quale potranno personalizzare l'aspetto del proprio dino e, una volta giocata una o più partite, guadagnarsi delle medaglie che poi saranno visibili in una pagina apposita. Per coloro che non vogliono fare il login saranno creati degli utenti ospiti che verranno eliminati quando l'utente esce dal sito.

Ci deve essere infine la possibilità di vedere la partita come spettatori, senza interagire con il gioco vero e proprio.

2 Analisi

2.1 Analisi del dominio

Non dovremo sviluppare questo progetto da zero, come base avremo infatti il Chrome Dino realizzato da Manuel Grosso (vedi sitografia), nel corso del primo semestre dell'anno scolastico 2021/2022. Questo progetto e quello precedente anno in comune l'aspetto multiplayer, ma per la versione sviluppata da Manuel il numero di giocatori era prescritto a 4. Noi dovremmo rendere questo aspetto dinamico, in modo da permettere maggiore flessibilità. Inoltre per questo progetto la meccanica di salto cambia notevolmente; infatti se prima si premevano dei tasti nel nostro progetto, come già detto, bisognerà saltare veramente. Attualmente non ci sono quindi progetti simili al nostro. Esso non risolve un problema vero e proprio, ma non per questo è poco importante; troviamo infatti molto interessante questa meccanica di gioco, in quanto complica abbastanza un gioco di per sé semplice. Il progetto si rivolge a tutti coloro che hanno voglia di provare qualcosa di nuovo e, avendo introdotto una meccanica di salto alternativa, non esclude coloro che non possono fisicamente saltare o hanno un telefono che non ha i sensori adeguati.

2.2 Analisi e specifica dei requisiti

| ID: REQ-01 | |
|-----------------|--|
| Nome | I giocatori devono poter creare una partita con URL o sessione |
| Priorità | 1 |
| Versione | 1.0 |
| Note | |

| ID: REQ-02 | |
|-----------------|-------------------------|
| Nome | È possibile registrarsi |
| Priorità | 1 |
| Versione | 1.0 |
| Note | |

| ID: REQ-03 | |
|-----------------|-------------------------------|
| Nome | È possibile eseguire il login |
| Priorità | 1 |
| Versione | 1.0 |
| Note | |

| ID: REQ-04 | |
|-----------------|---|
| Nome | I giocatori devono potersi unire alla partita |
| Priorità | 1 |
| Versione | 1.0 |
| Note | |

| ID: REQ-05 | |
|-----------------|---|
| Nome | Funzioni aggiuntive di login |
| Priorità | 2 |
| Versione | 1.0 |
| Note | |
| Sotto Requisiti | |
| 001 | Personalizzazione del dinosauro salvata |
| 002 | Punteggio salvato nel profilo |
| 003 | Bacheca per visualizzare le medaglie |

| ID: REQ-06 | |
|-----------------|-------------------------------|
| Nome | La GUI deve essere responsive |
| Priorità | 2 |
| Versione | 1.0 |
| Note | |

| ID: REQ-07 | |
|-----------------|------------------------------------|
| Nome | Possibilità di giocare come ospite |
| Priorità | 2 |
| Versione | 1.0 |
| Note | |

| ID: REQ-08 | |
|-----------------|---|
| Nome | Il dinosauro deve essere personalizzabile prima della partita |
| Priorità | 3 |
| Versione | 1.0 |
| Note | |

| ID: REQ-09 | |
|-----------------|--|
| Nome | Il dinosauro deve saltare sfruttando i sensori del telefono |
| Priorità | 2 |
| Versione | 1.0 |
| Note | |
| Sotto Requisiti | |
| 001 | Alternativa di gioco in caso di handicap o assenza di sensori. |

| ID: REQ-10 | |
|-----------------|--|
| Nome | Il numero di giocatori deve essere dinamico: da uno a dieci. |
| Priorità | 1 |
| Versione | 1.0 |
| Note | |

| ID: REQ-11 | |
|-----------------|---|
| Nome | Il nome dei giocatori deve apparire a schermo |
| Priorità | 3 |
| Versione | 1.0 |
| Note | |

| ID: REQ-12 | |
|-----------------|---|
| Nome | Alla fine del gioco deve essere visualizzata una classifica |
| Priorità | 2 |
| Versione | 1.0 |
| Note | |
| Sotto Requisiti | |
| 001 | Punteggi in ordine decrementale |
| 002 | Visualizzare la/le medaglie direttamente nella classifica |

| ID: REQ-13 | |
|-----------------|--|
| Nome | Al vincitore viene assegnata una medaglia |
| Priorità | 3 |
| Versione | 1.0 |
| Note | |
| Sotto Requisiti | |
| 001 | Medaglia generata con un algoritmo per rendere il più univoche possibili |

| ID: REQ-14 | |
|-----------------|--|
| Nome | Possibilità di vedere la partita da remoto |
| Priorità | 3 |
| Versione | 1.0 |
| Note | |

2.3 Use case

Ecco lo *use case* da noi definito:

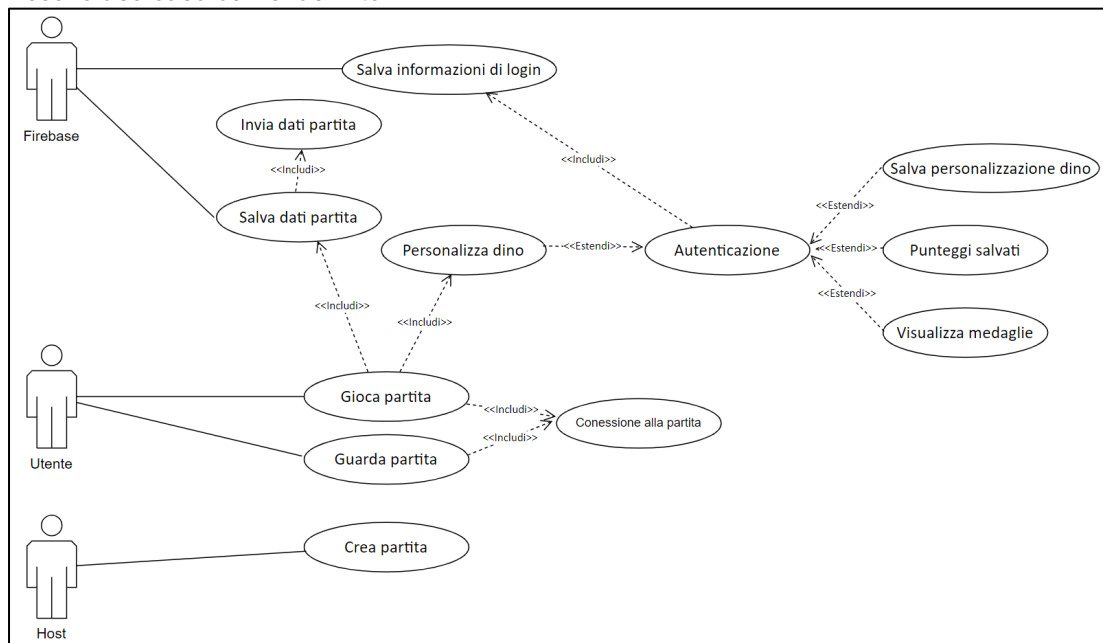


Figura 1 Use case

2.4 Pianificazione

Per quanto riguarda la pianificazione alleghiamo il diagramma di Gantt iniziale. Per lo sviluppo del progetto abbiamo deciso di utilizzare un approccio *Waterfall*.

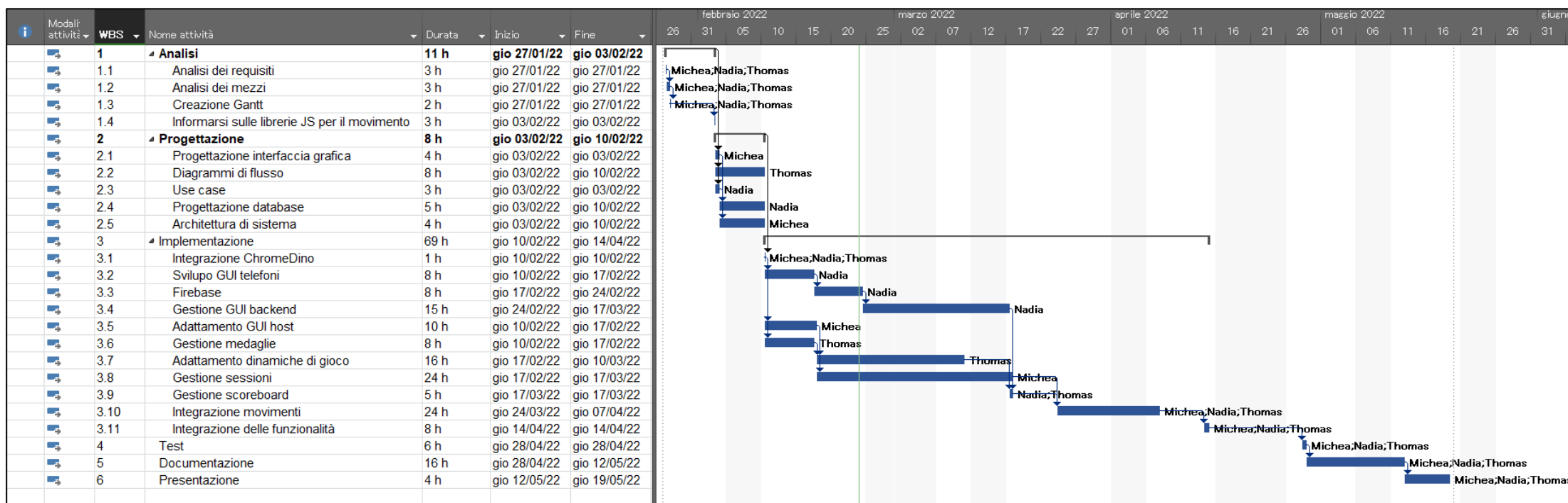


Figura 2 Gantt preventivo

2.5 Analisi dei mezzi

Software

- FireBase 8.2.1
- MockFlow 1.4.7
- Draw.io
- VS Code 1.65.2
- GIMP 2.10.24
- GitHub
- GitHub Desktop 2.9.12

Hardware

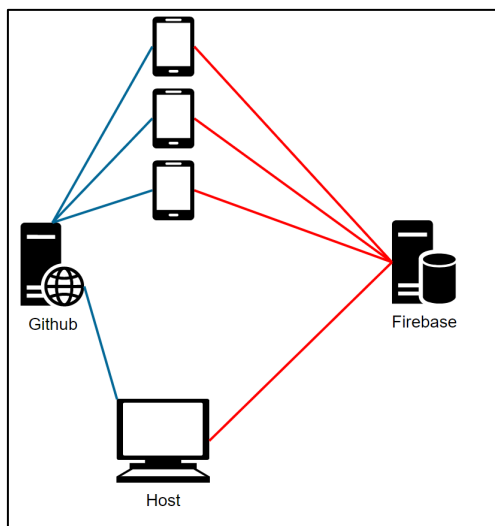
- Laptop personali
- PC scolastici

Come abbiamo già scritto il progetto è scritto in JavaScript, sarà quindi eseguibile da tutti i sistemi operativi. Per quanto riguarda la possibilità di saltare tramite i sensori di movimento invece sui terminali che eseguono IOS non sarà possibile usufruire di questa funzione. Questo a causa delle restrizioni e delle misure di sicurezza applicate da Apple.

3 Progettazione

La progettazione è sempre un punto importante in ogni progetto; perciò, abbiamo voluto dedicare tutto il tempo necessario ad essa, definendo tutti gli aspetti che ci sono venuti in mente.

3.1 Design dell'architettura del sistema



Per l'architettura di sistema abbiamo voluto adottare una struttura abbastanza semplice ma funzionale. Il primo passo della comunicazione, in ordine cronologico, è l'host che crea una partita e la mostra agli utenti. Tutta la parte di gestione del server e delle connessioni è gestita grazie ai server di GitHub. Infatti questa piattaforma offre infatti un servizio chiamato "GitHub Pages". Esso svolge la funzione di un Web server, togliendo così l'incombenza all'utente. Contemporaneamente alla creazione della partita l'host comunica al server FireBase le istruzioni necessarie per il buon funzionamento della stessa. Una volta che la partita è stata creata gli utenti si collegano alla pagina, collegandosi quindi ai server GitHub, ma instaurano anche una comunicazione con il server FireBase

Figura 3 Architettura di sistema

3.2 Design dei dati e database

3.3 Design delle interfacce

Per la progettazione delle interfacce abbiamo deciso di dividere i mockups in 2 famiglie: quelle pesate per il telefono e quelle per il computer. Per le interfacce che dovevano essere visualizzate da entrambi i terminali abbiamo realizzato entrambe le versioni.

3.3.1 Design interfacce telefono:

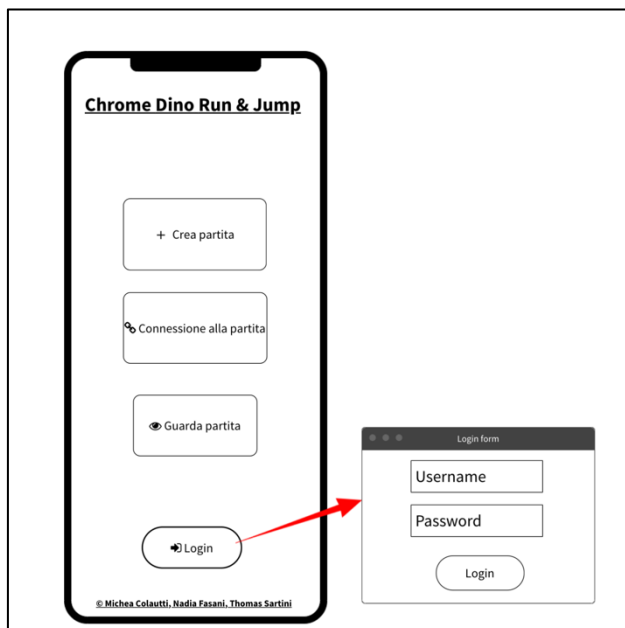


Figura 4 Home page con pop-up di login

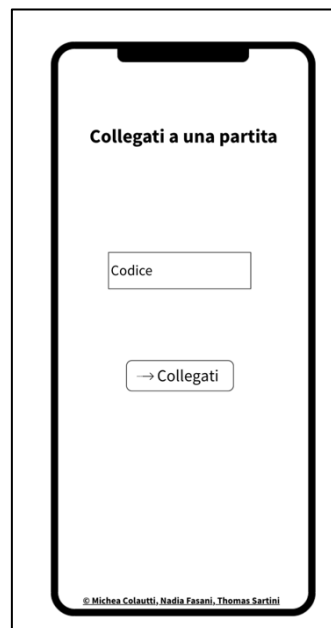


Figura 6 Pagina per collegamento ad una partita



Figura 5 Home page con utente loggato

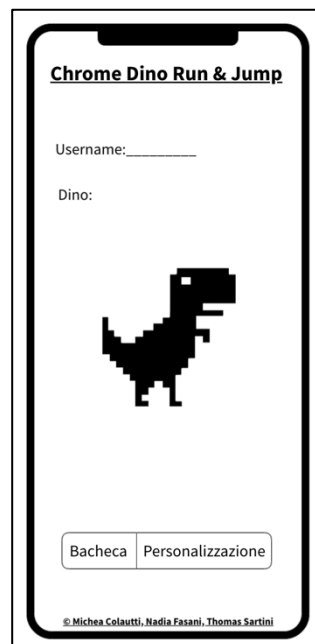


Figura 7 Pagina utente

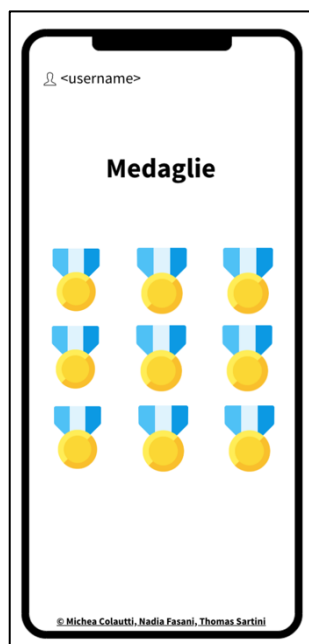


Figura 8 Bacheca medaglie

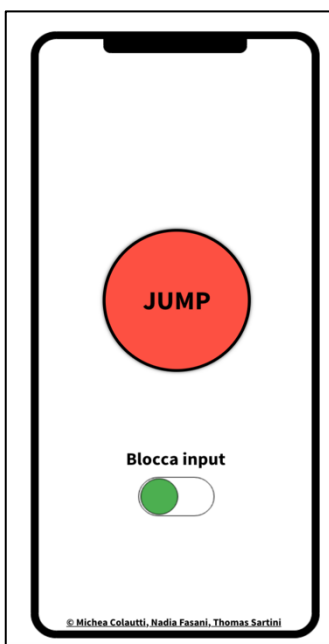


Figura 9 Pagina personalizzazione utente

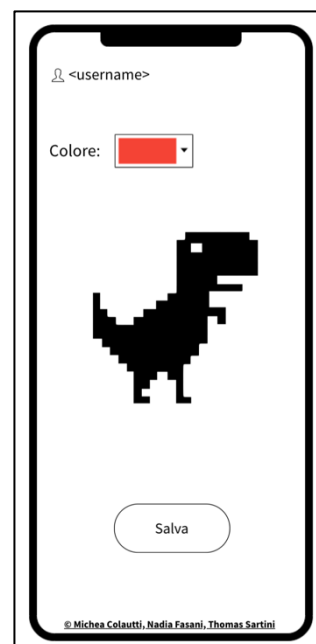


Figura 10 Pagina di gioco

3.3.2 Design interfacce computer

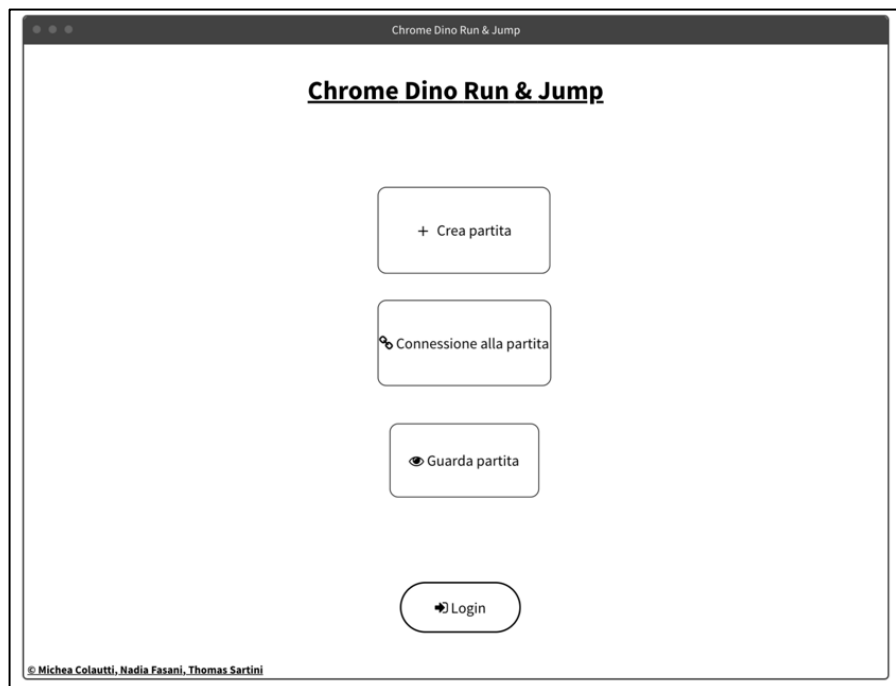


Figura 11 Home page

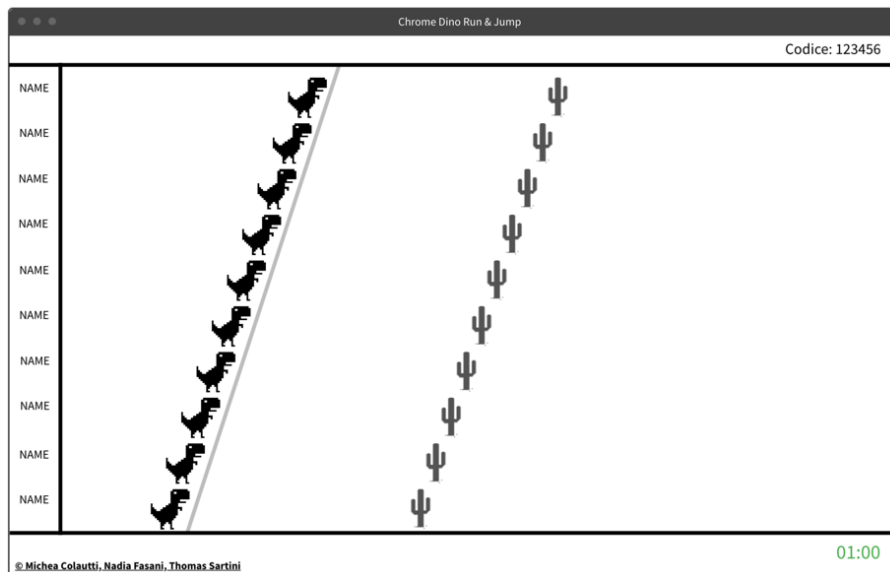


Figura 12 Creazione di una partita

Chrome Dino Run & Jump

Collegati a una partita

Codice

→ Collegati

© Michea Colautti, Nadia Fasani, Thomas Sartini

Figura 13 Collegamento ad una partita

Chrome Dino Run & Jump

Score Board

| Posizione | Username | Medaglia |
|-----------|----------|----------|
| 1° | Dino 1 | |
| 2° | Dino 2 | |
| 3° | Dino 3 | |
| 4° | Dino 4 | |
| 5° | Dino 5 | |
| 6° | Dino 6 | |
| 7° | Dino 7 | |
| 8° | Dino 8 | |
| 9° | Dino 9 | |
| 10° | Dino 10 | |

© Michea Colautti, Nadia Fasani, Thomas Sartini

Figura 14 Classifica

3.4 Design procedurale

Per quanto riguarda il design procedurale alleghiamo il diagramma di flusso da noi pensato.

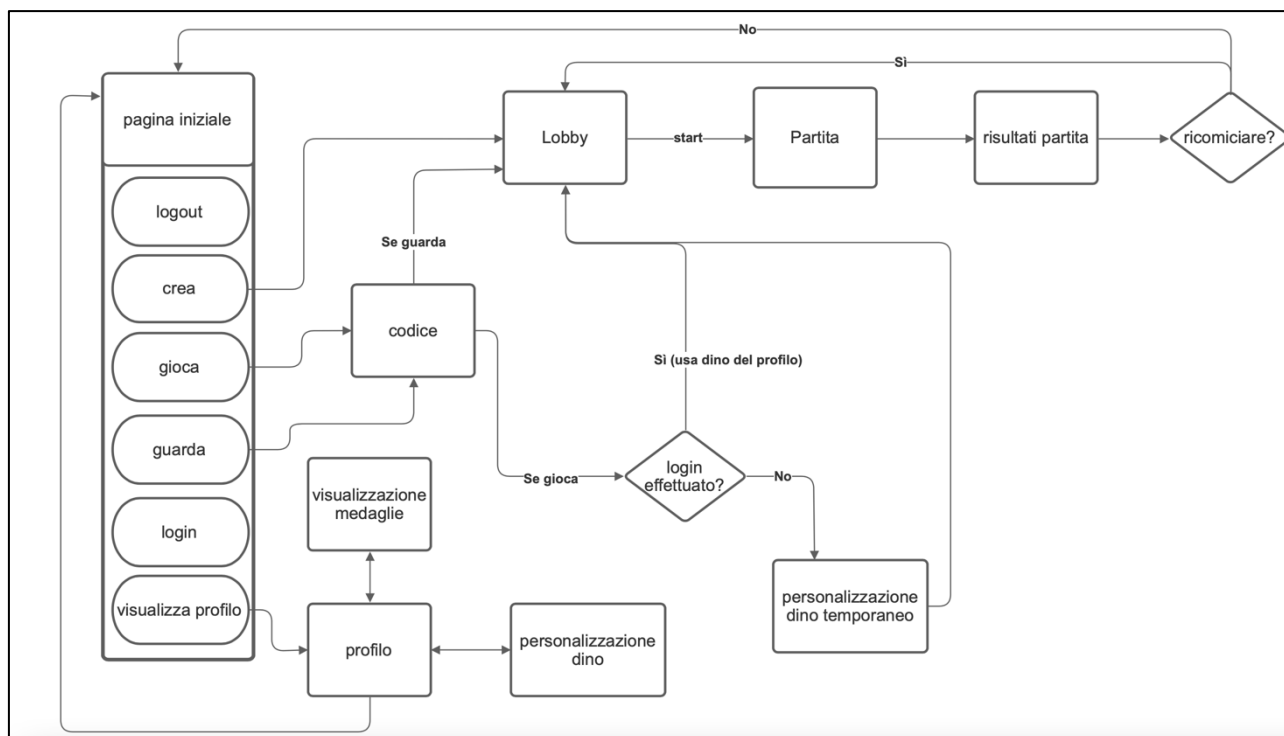


Figura 15 Diagramma di flusso

Lo schema del design è di per sé abbastanza esplicativo, ma spendiamo comunque qualche parola per commentarlo. Questo design ci è stato molto utile in fase di implementazione, poiché ci ha permesso di strutturare meglio il codice. Il punto di partenza è la *pagina iniziale*, da qui si può arrivare alla pagina di login oppure fare il logout. Nella *pagina iniziale* sono inoltre presenti altre funzionalità, che sono descritte di seguito:

- Creare una partita
 - Questo collegamento aprirà la partita nella lobby, dove gli utenti si possono connettere.
 - In seguito, quando viene fatta partire la partita, il gioco parte.
 - Una volta finito il gioco viene mostrata la classifica, e si può ricominciare o terminare.
- Giocare una partita
 - Questa parte del gioco prevede come prima cosa l'inserimento del codice della partita.
 - Se a giocare è un guest si viene portati ad una pagina di personalizzazione del dinosauro, altrimenti si viene portati direttamente alla partita.
- Guardare una partita
 - Questo collegamento aprirà la stessa pagina di inserimento del codice usato per chi vuole giocare.
 - Una volta inserito il codice si viene portati alla partita.
- Visualizzare il proprio profilo.
 - Questa funzione invece permette all'utente di vedere il profilo, e contiene due sotto-funzioni.
 1. Visualizzare la bacheca con le medaglie
 2. Personalizzare il proprio dinosauro.

4 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.

5 Test

5.1 Protocollo di test

| | | | |
|-------------------|--|-------|---------------------------|
| Test Case: | TC-01 | Nome: | Creazione di una sessione |
| Riferimento: | REQ-01 | | |
| Descrizione: | L'utente raggiunge il sito e crea una partita | | |
| Prerequisiti: | - | | |
| Procedura: | <ol style="list-style-type: none"> 1. Raggiungere il sito dell'applicativo 2. Premere il pulsante "Crea una partita" | | |
| Risultati attesi: | <ol style="list-style-type: none"> 1. La pagina deve mostrare la schermata di gioco 2. La console di Firebase deve mostrare la sessione creata | | |

| | | | |
|-------------------|--|-------|----------------------|
| Test Case: | TC-02 | Nome: | Registrazione utente |
| Riferimento: | REQ-02 | | |
| Descrizione: | L'utente deve potersi registrare nel sito | | |
| Prerequisiti: | - | | |
| Procedura: | <ol style="list-style-type: none"> 1. Raggiungere il sito dell'applicativo 2. Premere sul pulsante "Sign Up" 3. Immettere username e password 4. Confermare | | |
| Risultati attesi: | <ol style="list-style-type: none"> 1. L'utente deve essere loggato 2. Nella console di Firebase deve essere presente un nuovo utente con l'username corretto (la password è cifrata) | | |

| | | | |
|-------------------|---|-------|-------|
| Test Case: | TC-03 | Nome: | Login |
| Riferimento: | REQ-03 | | |
| Descrizione: | L'utente deve potersi autenticare | | |
| Prerequisiti: | Un utente creato | | |
| Procedura: | <ol style="list-style-type: none"> 1. Raggiungere il sito dell'applicativo 2. Premere sul pulsante "Sign In" 3. Immettere username e password 4. Confermare | | |
| Risultati attesi: | L'utente deve essere loggato | | |

| | | | |
|-------------------|--|-------|-----------------------|
| Test Case: | TC-04 | Nome: | Unione ad una partita |
| Riferimento: | REQ-04 | | |
| Descrizione: | Un utente deve poter giocare una partita | | |
| Prerequisiti: | <ol style="list-style-type: none"> 1. Un utente creato 2. Una partita creata ed aperta su uno schermo | | |
| Procedura: | <ol style="list-style-type: none"> 1. Raggiungere il sito dell'applicativo 2. Effettuare il login 3. Premere sul pulsante "Connessione alla partita" 4. Immettere il codice della partita 5. Confermare | | |
| Risultati attesi: | L'utente deve apparire nella partita nella pagina dell'host | | |

| | | | |
|-------------------|---|-------|-----------------------------|
| Test Case: | TC-05 | Nome: | Funzione aggiuntive login 1 |
| Riferimento: | REQ-05 | | |
| Descrizione: | L'utente deve poter personalizzare il suo dinosauro | | |
| Prerequisiti: | Un utente creato | | |
| Procedura: | <ol style="list-style-type: none"> 1. Raggiungere il sito dell'applicativo 2. Effettuare il login 3. Recarsi nella pagina utente premendo sull'icona dello stesso 4. Premere su "Personalizza" 5. Scegliere un colore per il dinosauro 6. Premere "Salva" | | |
| Risultati attesi: | Il colore del dinosauro deve essere cambiato nella pagina utente | | |

| | | | |
|---------------|---|-------|-----------------------------|
| Test Case: | TC-06 | Nome: | Funzione aggiuntive login 2 |
| Riferimento: | REQ-05 | | |
| Descrizione: | Il punteggio dell'utente deve essere salvato | | |
| Prerequisiti: | <ol style="list-style-type: none"> 1. Un utente creato che non ha mai giocato una partita 2. Una partita creata ed aperta su uno schermo | | |
| Procedura: | <ol style="list-style-type: none"> 1. Raggiungere il sito dell'applicativo 2. Effettuare il login 3. Premere sul pulsante "Connessione alla partita" 4. Immettere il codice della partita 5. Confermare 6. Avviare la partita dall'host 7. Giocare | | |

| | |
|-------------------|---|
| | 8. Quando la partita finisce recarsi nella pagina dell'utente |
| Risultati attesi: | Il punteggio fatto deve essere visibile |

| | | | |
|-------------------|--|-------|-----------------------------|
| Test Case: | TC-07 | Nome: | Funzione aggiuntive login 3 |
| Riferimento: | REQ-05 | | |
| Descrizione: | Le medaglie devono essere presenti nella bacheca | | |
| Prerequisiti: | <ol style="list-style-type: none"> 1. Un utente creato con almeno una medaglia 2. Una partita creata ed aperta su uno schermo | | |
| Procedura: | <ol style="list-style-type: none"> 1. Raggiungere il sito dell'applicativo 2. Aprire la pagina utente 3. Premere su "Bacheca" | | |
| Risultati attesi: | Le medaglie dell'utente devono visibili | | |

| | | | |
|-------------------|---|-------|----------------|
| Test Case: | TC-08 | Nome: | GUI responsive |
| Riferimento: | REQ-06 | | |
| Descrizione: | La GUI del sito deve essere responsive | | |
| Prerequisiti: | Un utente loggato | | |
| Procedura: | <ol style="list-style-type: none"> 1. Raggiungere il sito dell'applicativo 2. Effettuare il login 3. Provare a ridimensionare la pagina e verificare che nessun elemento venga nascosto 4. Ripetere il test per tutte le pagine eccetto quella di gioco | | |
| Risultati attesi: | Tutti gli elementi nella pagina si spostano/ridimensionano correttamente | | |

| | | | |
|-------------------|--|-------|--|
| Test Case: | TC-09 | Nome: | Pagina di gioco delle dimensioni massime |
| Riferimento: | REQ-06 | | |
| Descrizione: | La pagina di gioco deve essere sempre delle dimensioni massime della pagina | | |
| Prerequisiti: | - | | |
| Procedura: | <ol style="list-style-type: none"> 1. Raggiungere il sito dell'applicativo 2. Creare una nuova partita 3. Verificare che la pagina occupi tutto lo spazio disponibile 4. Provare a ridimensionare la pagina e poi refreshare | | |
| Risultati attesi: | La pagina assume sempre le dimensioni massime consentite | | |

| | | | |
|-------------------|---|-------|------------------|
| Test Case: | TC-10 | Nome: | Gioco come guest |
| Riferimento: | REQ-07 | | |
| Descrizione: | Si deve poter giocare anche senza aver effettuato il login | | |
| Prerequisiti: | Una partita creata ed aperta su uno schermo | | |
| Procedura: | <ol style="list-style-type: none"> 1. Raggiungere il sito dell'applicativo 2. Premere sul pulsante "Connessione alla partita" 3. Immettere il codice della partita 4. Confermare 5. Avviare la partita dall'host 6. Giocare | | |
| Risultati attesi: | L'utente deve poter giocare normalmente | | |

| | | | |
|-------------------|--|-------|-----------------------------------|
| Test Case: | TC-11 | Nome: | Personalizzazione dinosauro guest |
| Riferimento: | REQ-08 | | |
| Descrizione: | Si deve poter personalizzare il proprio dinosauro anche senza aver effettuato il login, appena prima di unirsi alla partita | | |
| Prerequisiti: | Una partita creata ed aperta su uno schermo | | |
| Procedura: | <ol style="list-style-type: none"> 1. Raggiungere il sito dell'applicativo 2. Premere sul pulsante "Connessione alla partita" 3. Immettere il codice della partita 4. Selezionare il colore del dinosauro 5. Confermare | | |
| Risultati attesi: | Il dinosauro deve apparire nella pagina dell'host con il colore prescelto. | | |

| | | | |
|---------------|--|-------|------------------------------------|
| Test Case: | TC-12 | Nome: | Giocare con i sensori del telefono |
| Riferimento: | REQ-09 | | |
| Descrizione: | Si deve poter giocare sfruttando i sensori del telefono | | |
| Prerequisiti: | <ol style="list-style-type: none"> 1. Una partita creata ed aperta su uno schermo 2. Un telefono con i sensori di movimento appropriati (NO IOS) 3. Un utente creato | | |
| Procedura: | <ol style="list-style-type: none"> 1. Raggiungere il sito dell'applicativo 2. Premere sul pulsante "Connessione alla partita" 3. Immettere il codice della partita 4. Confermare 5. Provare a saltare con il telefono in mano o in tasca 6. Ripetere il test con un utente loggato | | |

| | |
|-------------------|---|
| Risultati attesi: | Il dinosauro deve saltare nella pagina dell'host. |
|-------------------|---|

| | | | |
|-------------------|---|-------|-------------------------|
| Test Case: | TC-13 | Nome: | Giocare con il pulsante |
| Riferimento: | REQ-09 | | |
| Descrizione: | Si deve poter giocare sfruttando i sensori del telefono. | | |
| Prerequisiti: | <ol style="list-style-type: none"> 1. Una partita creata ed aperta su uno schermo 2. Un utente creato | | |
| Procedura: | <ol style="list-style-type: none"> 1. Raggiungere il sito dell'applicativo 2. Premere sul pulsante "Unisciti ad una partita" 3. Immettere il codice della partita 4. Confermare 5. Provare a premere il pulsante "Jump" 6. Ripetere il test con un utente loggato | | |
| Risultati attesi: | Il dinosauro deve saltare nella pagina dell'host. | | |

| | | | |
|-------------------|---|-------|------------------------------|
| Test Case: | TC-14 | Nome: | Numero di giocatori dinamico |
| Riferimento: | REQ-10 | | |
| Descrizione: | Si deve poter giocare in più persone. | | |
| Prerequisiti: | <ol style="list-style-type: none"> 1. Una partita creata ed aperta su uno schermo 2. Almeno un utente creato | | |
| Procedura: | <ol style="list-style-type: none"> 1. Raggiungere il sito dell'applicativo 2. Premere sul pulsante "Connessione alla partita" 3. Immettere il codice della partita 4. Confermare 5. Ripetere più volte, fino ad un massimo di 11 6. Utilizzare anche almeno un utente loggato 7. Provare a saltare/premere jump su i vari dispositivi 8. Avviare la partita | | |
| Risultati attesi: | <ol style="list-style-type: none"> 1. Ad ogni connessione deve apparire un nuovo dinosauro 2. Ogni dinosauro deve saltare quando il dispositivo che lo controlla ordina un salto 3. Ci possono essere un massimo di 10 giocatori 4. Quando la partita viene avviata tutti i dinosauri devono ancora poter saltare | | |

| | | | |
|-------------------|---|-------|------------------------------|
| Test Case: | TC-15 | Nome: | Nome dei giocatori a schermo |
| Riferimento: | REQ-11 | | |
| Descrizione: | Quando ci si unisce ad una partita, oltre al dinosauro, appare anche il nickname | | |
| Prerequisiti: | <ol style="list-style-type: none"> 1. Una partita creata ed aperta su uno schermo 2. Almeno un utente creato | | |
| Procedura: | <ol style="list-style-type: none"> 1. Raggiungere il sito dell'applicativo con 2 dispositivi 2. Effettuare il login 3. Premere sul pulsante "Connessione alla partita" 4. Immettere il codice della partita 5. Confermare 6. Ripetere il test con un utente loggato e con più utenti contemporaneamente | | |
| Risultati attesi: | I nickname degli utenti devono apparire correttamente | | |

| | | | |
|-------------------|--|-------|-------------------|
| Test Case: | TC-16 | Nome: | Classifica finale |
| Riferimento: | REQ-12 | | |
| Descrizione: | Quando una partita finisce deve apparire una classifica | | |
| Prerequisiti: | <ol style="list-style-type: none"> 1. Una partita creata ed aperta su uno schermo 2. Almeno un utente creato | | |
| Procedura: | <ol style="list-style-type: none"> 1. Raggiungere il sito dell'applicativo con 2 dispositivi 2. Effettuare il login 3. Premere sul pulsante "Connessione alla partita" 4. Immettere il codice della partita 5. Confermare 6. Giocare | | |
| Risultati attesi: | La classifica appare alla fine della partita | | |

| | | | |
|---------------|--|-------|----------------------------|
| Test Case: | TC-17 | Nome: | Classifica finale ordinata |
| Riferimento: | REQ-12 | | |
| Descrizione: | Quando una partita finisce deve apparire una classifica ordinata per punteggio | | |
| Prerequisiti: | <ol style="list-style-type: none"> 1. Una partita creata ed aperta su uno schermo 2. Almeno un utente creato | | |
| Procedura: | <ol style="list-style-type: none"> 1. Raggiungere il sito dell'applicativo con 2 dispositivi 2. Effettuare il login 3. Premere sul pulsante "Connessione alla partita" 4. Immettere il codice della partita 5. Confermare | | |

| | |
|-------------------|---|
| | 6. Unirsi con un secondo utente (loggato o normale) 7. Giocare |
| Risultati attesi: | La classifica appare alla fine della partita ed è ordinata |

| | | | |
|-------------------|---|-------|---------------------------|
| Test Case: | TC-18 | Nome: | Medaglie nella classifica |
| Riferimento: | REQ-12 | | |
| Descrizione: | Quando una partita finisce deve apparire una classifica con una medaglia per il primo classificato | | |
| Prerequisiti: | 1. Una partita creata ed aperta su uno schermo | | |
| Procedura: | 1. Raggiungere il sito dell'applicativo con 2 dispositivi 2. Premere sul pulsante "Connessione alla partita" 3. Immettere il codice della partita 4. Confermare 5. Unirsi con un secondo utente (loggato o normale) 6. Giocare | | |
| Risultati attesi: | La classifica appare alla fine e solo il primo classificato ha una medaglia | | |

| | | | |
|-------------------|--|-------|---------------------------------|
| Test Case: | TC-19 | Nome: | Medaglie assegnate al vincitore |
| Riferimento: | REQ-12 | | |
| Descrizione: | Quando una partita finisce la medaglia viene assegnata al vincitore | | |
| Prerequisiti: | 1. Un utente creato con almeno una medaglia 2. Una partita creata ed aperta su uno schermo | | |
| Procedura: | 1. Raggiungere il sito dell'applicativo 2. Premere sul pulsante "Connessione alla partita" 3. Immettere il codice della partita 4. Confermare 5. Avviare la partita dall'host 6. Giocare 7. Quando la partita finisce verificare la presenza della medaglia nella classifica e recarsi nella pagina dell'utente 8. Premere su "Bacheca" | | |
| Risultati attesi: | La medaglia appena vinta deve essere nella bacheca | | |

5.2 Risultati test

| Test Case | Risultato | Commenti |
|-----------|-----------|-----------------------------|
| TC-01 | Passato | |
| TC-02 | Passato | |
| TC-03 | Passato | |
| TC-04 | Passato | |
| TC-05 | Passato | |
| TC-06 | FALLITO | Punteggi assenti |
| TC-07 | Passato | |
| TC-08 | Passato | |
| TC-09 | Passato | |
| TC-10 | Passato | |
| TC-11 | Passato | |
| TC-12 | FALLITO | |
| TC-13 | Passato | Perfezionare utente loggato |
| TC-14 | FORSE | Max 10? |
| TC-15 | Passato | |
| TC-16 | Passato | |
| TC-17 | Passato | |
| TC-18 | Passato | |
| TC-19 | Passato | |

5.3 Mancanze/limitazioni conosciute

Penso che la mancanza, o limitazione, principale del nostro progetto sia la struttura del codice. È stato fatto, a mio parere, un ottimo lavoro di refactoring che ci ha permesso di portare questo progetto fino in fondo con successo; il problema non sta quindi nel refactor in sé, quanto nella modalità dello stesso.

Un approccio a classi infatti, dove ogni parte del gioco sarebbe stata rappresentata da un oggetto e non da un array, avrebbe semplificato e abbellito ulteriormente il codice. Forse un approccio simile avrebbe risolto anche il problema del ghost. Per quest'ultimo non intendiamo dilungarci in quanto abbiamo già spiegato il problema in maniera esauriente nel capitolo 4 e pur non essendo questa, a conti fatti, una vera e propria limitazione, è sicuramente un punto a sfavore di questo progetto.



6 Consuntivo

Per questo capitolo alleghiamo il nostro Gantt consuntivo.

7 Conclusioni

Arrivando alla fine del progetto arriva il momento di tirare le somme e trarre le conclusioni. Guardando indietro siamo soddisfatti del lavoro da noi svolto. Non solo perché ci siamo impegnati a rispettare il più fedelmente possibile le specifiche, ma anche perché pensiamo di aver realizzato, nel complesso, un buon gioco. Per due di noi era la prima esperienza a lavorare in gruppo ad un progetto semestrale; abbiamo sì già svolto dei piccoli progetti in gruppo o in coppia, ma un progetto come questo, con uno sviluppo puntuale sotto più aspetti, come progettazione, codice, documentazione, ecc., è un'altra cosa.

Per quello che riguarda i rapporti nel gruppo è andato tutto bene, pur non avendo mai lavorato assieme abbiamo trovato una certa chimica; probabilmente è anche questo uno dei fattori che ci ha portato ad avere un progetto funzionante e quasi completo.

Tornando a parlare del progetto siamo anche contenti di aver imparato tutti moltissime nuove importanti nozioni. Non solo per quanto riguarda JavaScript, ma anche per la gestione del tempo, per la divisione del lavoro, e per lo spirito di gruppo. Oltre a JavaScript abbiamo imparato anche ad usare il framework Phaser e due di noi hanno avuto anche la possibilità di conoscere Firebase.

L'uso di strumenti come il Gantt, o GitHub, ci hanno aiutato bene a spartire i compiti e a creare un ambiente di lavoro funzionante.

Un altro aspetto che riteniamo sia stato importante per la buona riuscita del progetto è l'unione delle conoscenze. Infatti pensiamo che durante lo sviluppo del progetto, chi più chi meno, abbia messo a disposizione una sua conoscenza personale, o un lavoro già svolto, per semplificare il lavoro agli altri. Per esempio avere Nadia nel progetto ci ha aiutato per la parte di comunicazione con il database Firebase: avendo lei già sviluppato un progetto con questo sistema è stato più semplice implementare il codice e apprendere nuove nozioni.

Thomas ha portato un'ottima conoscenza di JavaScript che, unita a molta pazienza, ci ha permesso di migliorare e convertire il codice prodotto in precedenza da Manuel Grosso. Michea invece ha portato qualcosa in più dal lato "burocratico", quindi ciò che riguarda documentazione e diari.

Questo forse è dovuto al fatto che, bocciando un anno, ha scritto più documentazioni in questi anni.

Altro aspetto interessante che vogliamo citare per questo progetto è che, come abbiamo appena accennato, per la prima volta non ci siamo confrontati con uno sviluppo da zero di un gioco. Infatti, siccome abbiamo usato come base il progetto di Manuel, prima di sviluppare si è presentata una fase di comprensione e di ristrutturazione del codice. Questa fase è stata forse una delle più complesse, infatti comprendere del codice scritto da un'altra persona, con una documentazione piuttosto scarsa, è stato molto difficile. Soprattutto perché l'interezza del codice era stata sviluppata con un approccio *Hard coded*. Questo ha alzato la difficoltà, ma pensiamo di aver fatto un buon lavoro, anche se c'è ancora margine di miglioramento.

Dobbiamo però dire che forse potevamo dare qualcosa di più in fase di progettazione. Ci siamo trovati a metà progetto, come detto nel capitolo 4, a dover escludere dal QdC una parte del gioco. Questo è dovuto in parte al fatto che nella fase di ristrutturazione del codice e di realizzazione delle basi dell'app, non abbiamo considerato tutti gli aspetti che il nostro approccio avrebbe portato. Forse, vista la problematica che ci ha impedito di realizzare la funzione *ghost*, il risultato non sarebbe cambiato, ma avremmo sicuramente risparmiato tempo.

Infine dobbiamo per forza dire che parte del merito della buona riuscita del progetto va sicuramente assegnato al professor Petrini, che per tutta la durata ci ha seguito e ci ha sempre fornito tutti i mezzi per risolvere tutti i problemi; sia che si parli di spunti che di dritte vere e proprie. Cogliamo quindi l'occasione per ringraziarlo, dato che senza di lui il progetto non sarebbe probabilmente riuscito così bene.

7.1 Sviluppi futuri

Per quanto riguarda gli sviluppi futuri di questo progetto, abbiamo identificato alcune opzioni:

La prima non riguarda una funzione o aggiunta al progetto in sé, quanto il codice di base. Come abbiamo già detto nel refactor abbiamo deciso di utilizzare degli array per rendere l'applicazione multiplayer, ma un migliore approccio sarebbe stato quello di utilizzare le classi. È quindi questa la prima miglioria che può essere fatta al progetto; ciò permetterebbe non solo di avere un codice molto più comprensibile, ma semplificherebbe anche future modifiche.

Un altro sviluppo è da ricercare nell'implementazione della funzione *ghost*. Ci è dispiaciuto molto infatti non poter realizzare questa funzione, in quanto ritenevamo molto interessante la possibilità di assistere da remoto ad una partita come spettatore. Tuttavia, se ciò non risultasse comunque possibile tramite un'implementazione ad oggetti o tramite migliorie al codice, sarebbe necessario cambiare la piattaforma di base del database, ovvero Firebase. Nel caso si prendesse questa strada bisognerebbe ripensare l'applicazione quasi da zero, modificando tutte le logiche per il passaggio e l'ottenimento dei dati.

Altri sviluppi futuri che ci sono venuti in mente, e che riguardano di più il progetto nel suo aspetto principale, sono la possibilità di personalizzare il proprio dinosauro, magari implementando l'inserimento di *skin* o la possibilità di utilizzare più colori per il proprio personaggio. Senza dimenticare che si potrebbe pensare addirittura all'aggiunta di oggetti esterni da aggiungere, come cappelli o occhiali: si potrebbe per esempio fare in modo che solo gli utenti con un punteggio massimo che supera una certa soglia possono accedere a questi elementi.

Ispirandoci invece al vero Chrome Dino ci è venuto in mente che forse sarebbe molto carino implementare anche diverse tipologie di cactus, differenziandoli per altezza e magari generandoli anche in coppia. Si potrebbero inserire anche gli uccelli che sono presenti nel gioco originale, sviluppando anche la possibilità di potersi abbassare, in modo da variare un po' i movimenti che l'utente deve fare. Infine, come piccola aggiunta, si potrebbe implementare lo scorrimento del tempo, in modo che ogni tanto il gioco passi dal giorno alla notte.

7.2 Considerazioni personali

7.2.1 Michea

Personalmente la cosa che più ho apprezzato nel corso del progetto è stata la sfida, per me praticamente nuova, di creare un gioco. Pur avendo fatto ormai -nel bene e nel male- ormai 3 progetti, non mi era mai capitato di dover sviluppare un gioco. Sono sempre stato convinto che lo sviluppo dei giochi fosse una branca piuttosto complessa e a me non vicina, dato che non sono mai stato un videogiocatore, ma sviluppare il progetto mi ha intrigato molto. Sono contento di come è uscito il progetto alla fine e di come è stato portato avanti. Trovo infatti che nel gruppo siamo sempre andati d'accordo, abbiamo iniziato bene e ciò ci ha permesso di pianificare tutto il lavoro e di dividerlo efficientemente, spartendo i vari compiti in base alle qualità e alle doti individuali.

Spero che questo progetto, prima o poi, venga riproposto ad altri allievi, sia in ottica di un possibile miglioramento, sia come progetto da sviluppare da zero.

Un ultimo motivo per cui ho apprezzato molto questo progetto è stato il fatto che dopo un anno e mezzo sono tornato ad occuparmi di un progetto in JavaScript, linguaggio che mi è sempre piaciuto e che nell'ultimo periodo ho perso un po'. Spero quindi che questo possa essere, oltre che un arricchimento per la mie conoscenze, un punto di ripartenza per tornare a masticare JavaScript e dintorni.

7.2.2 Nadia**7.2.3 Thomas**

8 Sitografia

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

9 Allegati

- Diari di lavoro
- QdC
- Prodotto