

Università degli Studi di Udine
Dipartimento di Scienze matematiche, informatiche e fisiche
Internet of Things, Big Data & Machine Learning
A.A. 2022-23

Relazione del progetto di Game Programming

Gianluca Fabris 151884
Andrea Martin 147800

Indice:

- 1) Guida al gioco
 - 1.1) Sports inc
 - 1.2) Minigioco atletica
 - 1.3) Minigioco basket
 - 1.4) Minigioco calcio
 - 1.5) Minigioco pallavolo
 - 1.6) Minigioco tennis
 - 1.7) Minigioco motorsport
 - 1.8) Strategia di gioco
 - 1.9) Significato del gioco
- 2) Scelte implementative
 - 2.1) Sports inc
 - 2.2) Minigioco atletica
 - 2.3) Minigioco basket
 - 2.4) Minigioco calcio
 - 2.5) Minigioco pallavolo
 - 2.6) Minigioco tennis
 - 2.7) Minigioco motorsport
 - 2.8) metodologie di lavoro
- 3) Future implementazioni

1. Guida al gioco

1.1. Sports inc

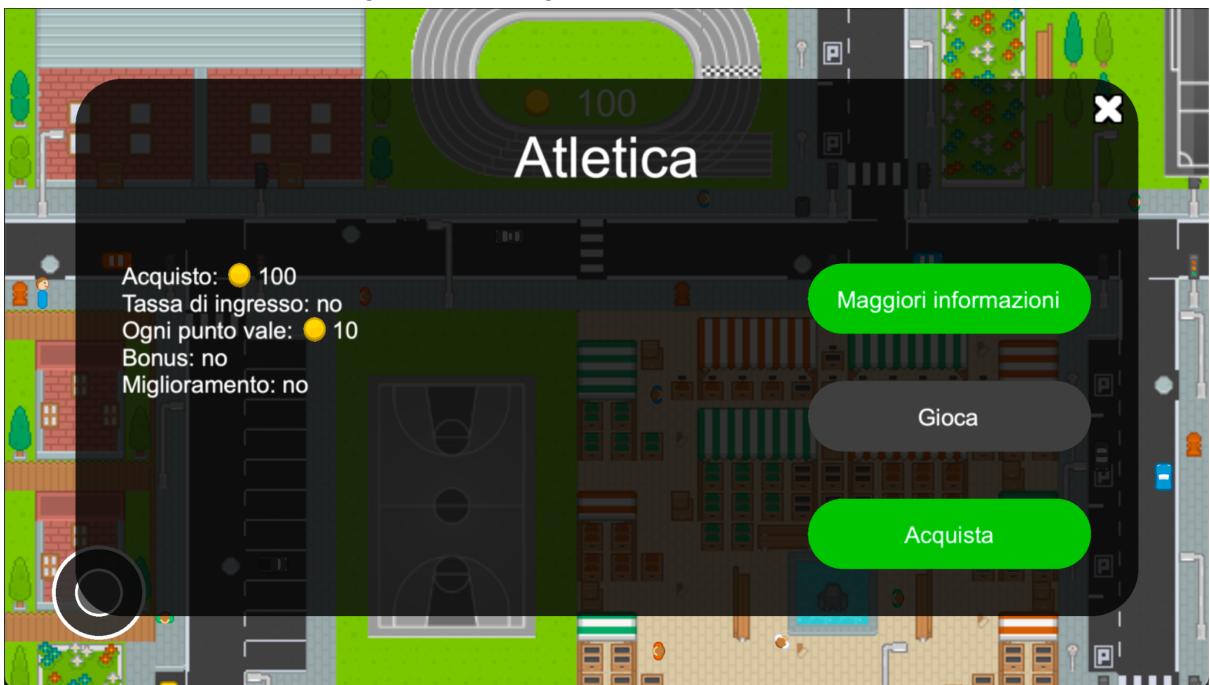
Il gioco all'avvio per la prima volta presenta una schermata di benvenuto, con un video per introdurre al gioco, e vengono accreditate 100 monete, sufficienti per comprare la palestra di atletica.



Esplorando la mappa, è possibile trovare le varie palestre sparse in essa.

Premendo su una di esse, viene presentata una schermata di informazioni.

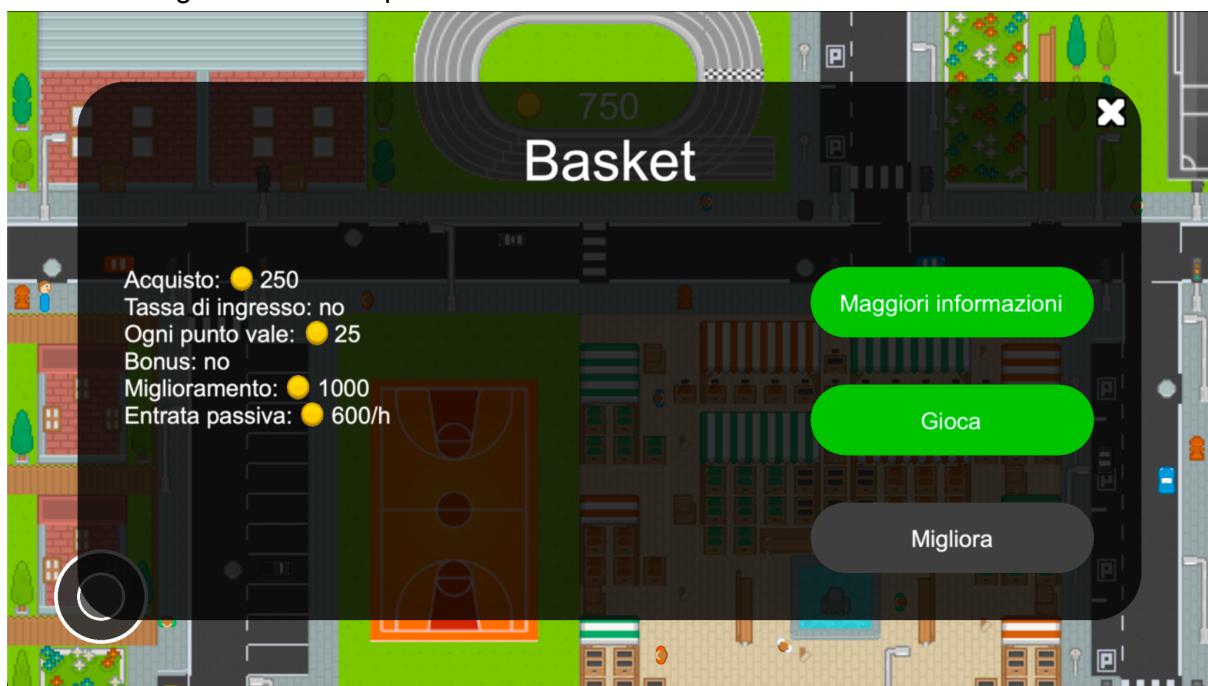
Da tale schermata è possibile avere maggiori informazioni riguardo lo sport (istruittivo per i bambini più piccoli) e come giocare il minigioco con un video dimostrativo.

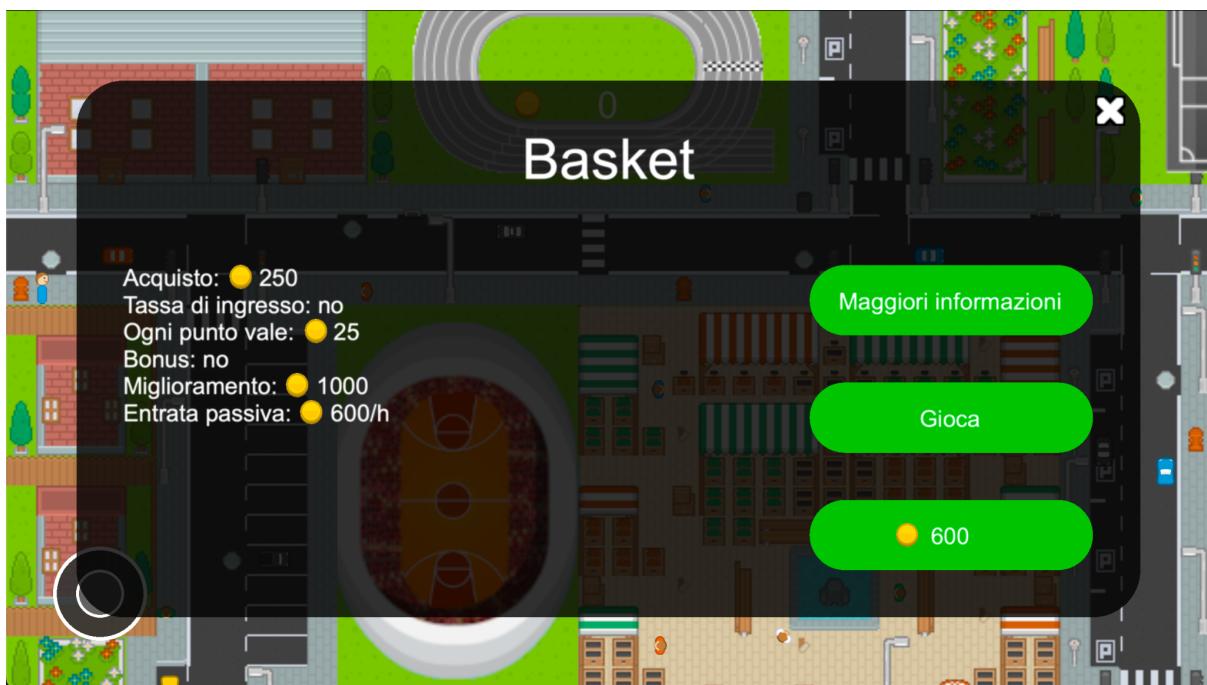




È possibile avviare il minigioco se si possiede la palestra.

Inoltre, è possibile comprare, migliorare la palestra o raccogliere l'entrata passiva, in base allo stato del gioco e se si dispone di abbastanza monete.



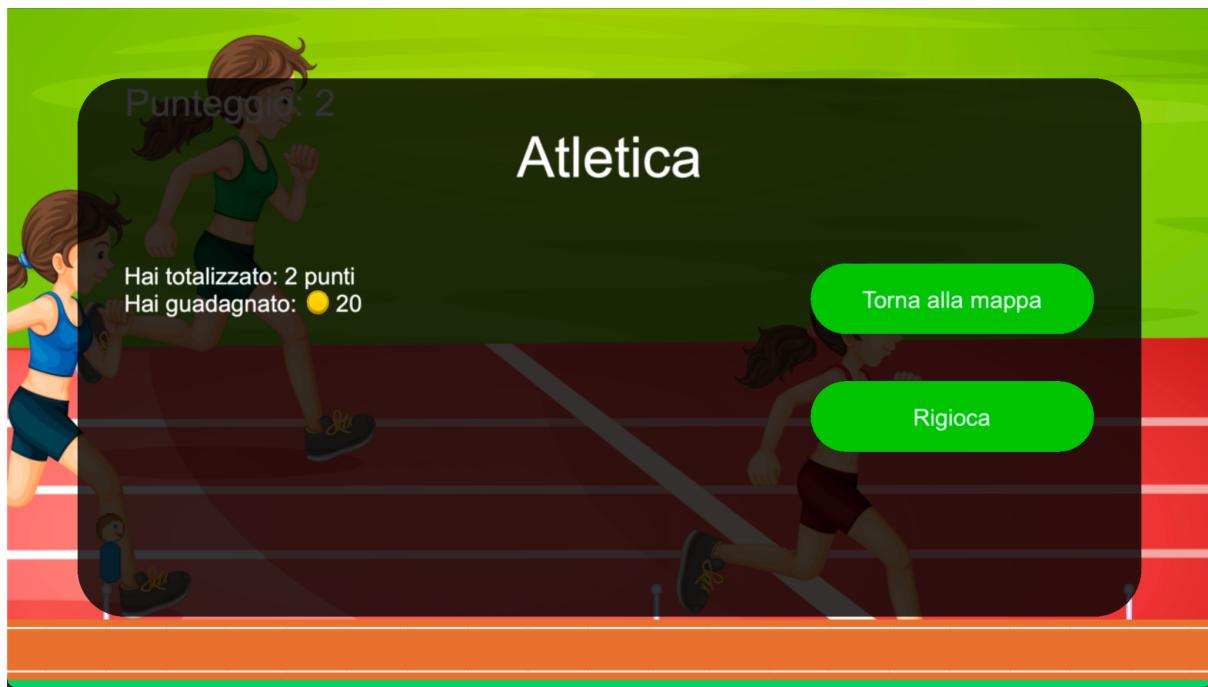


Una volta comprata la prima palestra viene presentata una schermata introduttiva ai minigiochi, mentre se si completa il gioco (acquistando e migliorando tutte le palestre) viene presentata una schermata di completamento del gioco indicando che si può continuare a giocare ai vari minigiochi.





Al termine di ogni minigioco viene presentata una schermata che riassume il punteggio ottenuto e le monete guadagnate con la possibilità di rigiocare il minigioco o di ritornare alla mappa principale.



1.2. Minigioco atletica

Abbiamo scelto questo sport perché volevamo inserirne uno presente alle olimpiadi e praticato nelle scuole.

Il gioco consiste nel saltare gli ostacoli, che continuano a scorrere da destra verso sinistra, facendo saltare l'atleta cliccando con il dito sullo schermo; ad ogni ostacolo saltato l'atleta guadagna 1 punto che viene mostrato in alto a sinistra.

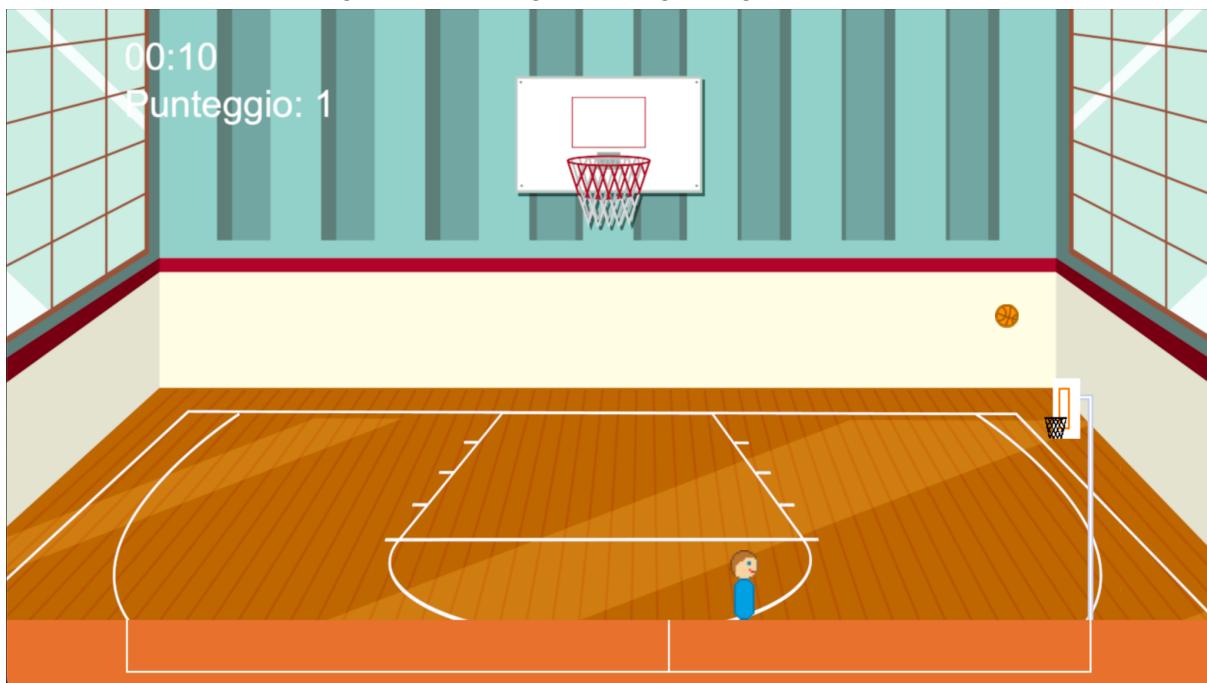


1.3. Minigioco basket

Abbiamo scelto questo sport perché pensiamo sia uno dei più seguiti a livello internazionale.

L'atleta ha tempo 1 minuto per aggiudicarsi il maggior numero possibile di punti; per effettuare un canestro il giocatore deve premere sulla palla, trascinare il dito per regolare la mira e rilasciare il tocco per lanciarla a canestro.

Inoltre, si ricorda che ad ogni canestro il giocatore guadagna 1 punto.



1.4. Minigioco calcio

Abbiamo scelto questo sport perché il più amato dagli italiani.

Anche in questo minigioco, il giocatore avrà a disposizione solamente 1 minuto per effettuare il maggior numero di goal.

Il giocatore, per fare goal, deve lanciare la palla effettuando uno swipe verso la porta, ad ogni goal segnato gli sarà dato 1 punto.



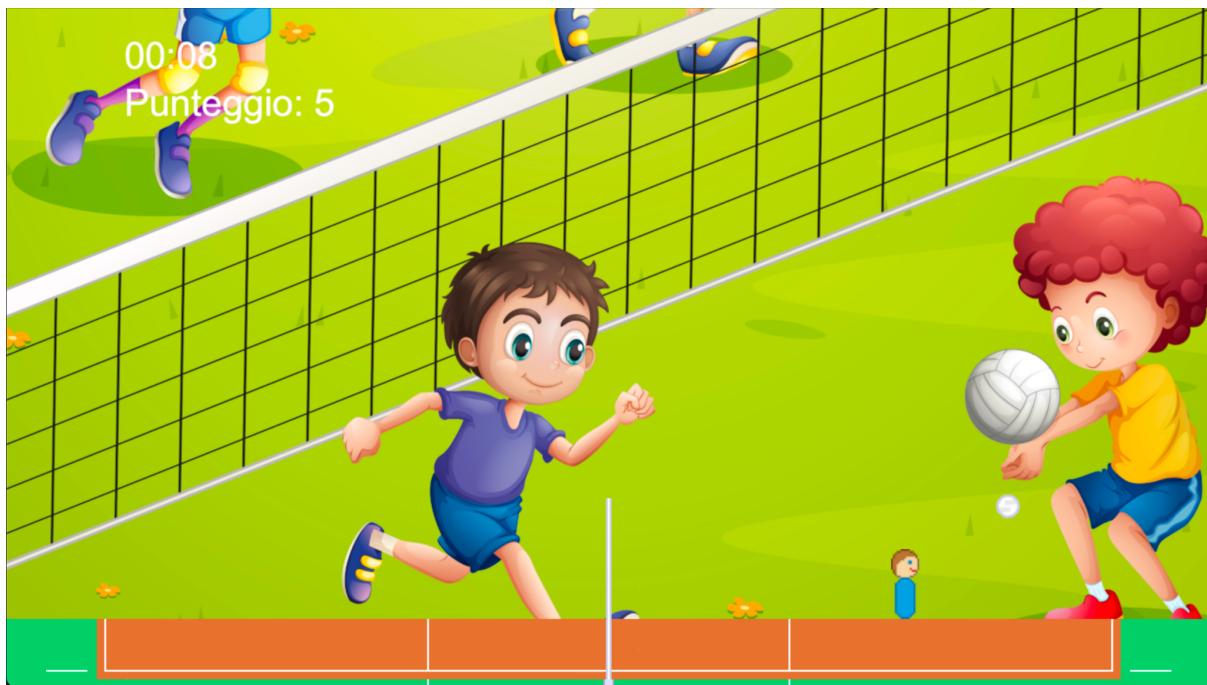
1.5. Minigioco pallavolo

Abbiamo scelto pallavolo e tennis perché riteniamo siano due sport famosissimi (sia nel mondo reale che nei videogiochi) e molto spesso praticati dagli studenti.

Lo scopo del gioco è quello di tenere la palla il più possibile in aria ed evitare di farle toccare terra.

Per poter interagire con il minigioco bisogna posizionare il dito sopra il giocatore e spostarlo sotto alla palla.

Ricordando che c'è solo 1 minuto di gioco, si guadagna 1 punto ad ogni palleggio sulla testa del giocatore e si guadagna 10 punti; se si riesce a fare rimbalzare la palla sulla rete si guadagnano 10 punti, invece se la palla tocca terra si perde 1 punto e se si fa scappare la palla ai lati essa verrà riposizionata sullo schermo, senza alcun effetto sul punteggio.



1.6. Minigioco tennis

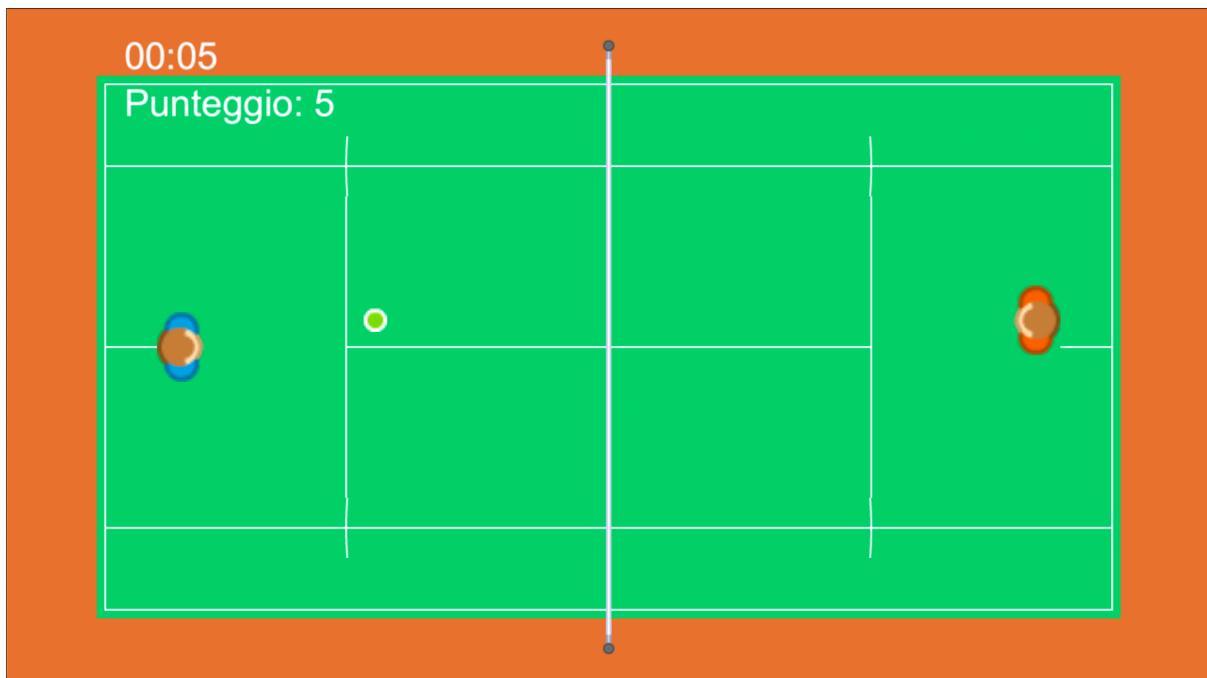
Questo minigioco è stato inserito anche per rendere omaggio al famosissimo Pong (uno dei primi videogiochi).

Inclinando il telefono avanti e indietro il giocatore può controllare il movimento del player in gioco e colpire la pallina.

Per guadagnare punti in questo minigioco ci sono due possibilità:

- colpire la pallina con il giocatore e mandarla oltre la rete (1 punto)
- mandare la pallina nell'area di fuorigioco dell'avversario (25 punti)

Inoltre il giocatore ha solo 1 minuto di tempo per fare più punti possibili.



1.7. Minigioco motorsport

Abbiamo scelto questo sport perchè ne siamo entrambi appassionati e volevamo rappresentare questa categoria di atleti, che seppure si allenino come i giocatori di calcio, basket, ecc., spesso per poter seguire la loro passione non vengono pagati, ma devono trovare degli sponsor o pagare per poter gareggiare, in quanto, oltre la preparazione dell'atleta, c'è un elemento in più, la macchina, che può influenzare l'esito della competizione sportiva.

Tramite il joystick è possibile controllare il movimento della macchina blu, l'obiettivo è quello di arrivare per primo al traguardo. Il giocatore può scegliere se fare tutto il giro della pista o se prendere una scorciatoia sullo sterrato.

In alto a sinistra viene mostrato il cronometro del minigioco e la posizione (quest'ultima viene aggiornata quando si passa il traguardo).



1.8. Strategia di gioco

Inizialmente è possibile comprare solo la palestra di atletica.

Bisogna giocare al minigioco atletica e guadagnare monete per poter comprare altre palestre.

Il giocatore una volta raccolte abbastanza monete può sviluppare il proprio gioco in maniera indipendente e non vincolata; noi consigliamo di seguire questo percorso:

- acquisto atletica
- acquisto basket
- acquisto pallavolo
- miglioramento pallavolo
- acquisto tennis
- miglioramento tennis
- acquisto calcio
- miglioramento basket
- miglioramento calcio
- acquisto motorsport

- miglioramento motorsport

1.9. Significato del gioco

Con questo gioco vogliamo trasmettere ai più piccoli che con impegno e costanza (giocando ai minigiochi) e anche partendo da poco (iniziando con sole 100 monete) si può arrivare a raggiungere i propri desideri (avere tutte le palestre) e speriamo che questo messaggio possa essere compreso e astratto nella vita reale.

2. Scelte implementative

2.1. Sports inc

Abbiamo strutturato il gioco usando le scene di composer, inizialmente viene presentata la scena di intro, contenente la menzione Uniud e il corso Laboratorio di Game Programming e un suono di benvenuto, successivamente si passa alla scena del gioco principale e, in base alla scelta dell'utente, si passa alla scena del minigioco selezionato; una volta terminato il minigioco si ritorna alla scena del gioco principale.

Il file game è strutturato in varie funzioni per suddividere il lavoro e rendere più leggibile il codice: la funzione movePlayer si occupa di spostare il giocatore nella mappa e la funzione moveMap si occupa di spostare la mappa se il giocatore sta per uscire dallo schermo, la funzione informazioniPalestraShow viene usata negli eventListener delle palestre e serve per visualizzare le informazioni della palestra (se non vi sono altre schermate già aperte), la funzione aggiornaTestoSoldi serve per aggiornare il testo che mostra le monete in base allo stato del gioco, la funzione aggiornaVisualizzazioneSport serve per aggiornare la visualizzazione delle palestre in base allo stato del gioco.

Le funzioni di per gestire la scena di composer: scene:create crea la scena di composer e istanzia le variabili, scene:show se event.phase è "will" carica la mappa, mentre se event.phase è "did" attiva gli eventListener sulle palestre e fa partire la musica di sottofondo, scene:hide se event.phase è "will" rimuove gli eventListener dalle palestre e mette in pausa la musica, mentre se event.phase è "did" rimuove tutta la grafica dal display, scene:destroy ripulisce la memoria, queste quattro funzioni sono presenti anche nei vari minigiochi con gli stessi compiti.

I file introduzione e informazioniPalestra servono a creare le schermate di introduzione al gioco e ai minigiochi e le schermate di informazioni dei minigiochi, dove è possibile avere maggiori informazioni, riguardo lo sport e a come giocare il minigioco con video dimostrativo, è possibile avviare il minigioco se si possiede la palestra ed è possibile comprare, migliorare la palestra o raccogliere l'entrata passiva, in base allo stato della palestra, se si dispone di abbastanza monete e se è passata un'ora dall'ultima volta che è stata raccolta l'entrata passiva.

Nonostante utilizziamo l'ultima versione di ponytiled, abbiamo riscontrato un bug di visualizzazione dei tile in solar 2D, probabilmente dovuto alla dimensione dei tile non standard (16x16 invece di 32x32), questo problema è stato risolto salvando i layer coinvolti come immagini e sostituendoli. Questo problema si è riscontrato anche nella mappa del motorsport presumibilmente per lo stesso problema (con tile da 64x64) e risolto alla stessa maniera.

Per rendere i minigiochi più interessanti e interattivi, abbiamo scelto di creare un minigame per ogni modalità di interazione vista a lezione:

- atletica -> tap
- basket -> fisica
- calcio -> swipe
- pallavolo -> touch
- tennis -> giroscopio/accelerometro
- motorsport -> controller

2.2. Minigame atletica

Il minigame di atletica è costituito da 2 funzioni principali che rispettivamente gestiscono le collisioni tra l'ostacolo e il giocatore e generano randomicamente gli ostacoli; oltre a queste funzioni sono presenti anche delle funzioni marginali che gestiscono il salto compiuto dal giocatore al tocco sullo schermo da parte dell'utente e lo spostamento del giocatore e la generazione di nuovi ostacoli.

La generazione di nuovi ostacoli si divide in 2 funzioni principali che sono `createObstacle` e `scrollObstacles`.

La prima genera fisicamente l'ostacolo, l'area di collisione e l'area adiacente (che indica la corretta esecuzione del salto) in una posizione random mentre la seconda sposta gli ostacoli da destra verso sinistra. Una volta che l'ostacolo esce dalla visualizzazione viene distrutto.

Il cuore pulsante del minigame è la funzione `collisionHandler` che, oltre a verificare l'esecuzione corretta del salto da parte del giocatore e assegnare i punti, richiama la funzione che genera nuovi ostacoli e distrugge quelli usciti dalla visualizzazione.

2.3. Minigame basket

Il minigame del basket è realizzato da 3 funzioni principali che interoperano per permettere all'utente di effettuare dei tiri a canestro e collezionare punti.

Le 3 funzioni principali sono la funzione `basketShooting` che applica un impulso sulla palla e riproduce un suono, la funzione `drawTrajectory` che disegna una traiettoria fittizia dalla palla verso il canestro, e richiama la funzione precedentemente illustrata per lanciare la palla, e la funzione `collisionHandler` che verifica se viene effettuato correttamente un canestro.

2.4. Minigame calcio

Nel minigame del calcio, come nel basket, sono presenti 3 funzioni principali (una che lancia la palla, una che disegna la traiettoria e una che controlla se il giocatore ha guadagnato un punto).

In questo caso le funzioni si chiamano rispettivamente `soccerGoalShot`, `swipeTrajectory` e `collisionHandler` quest'ultima funzione è ricorrente in ogni minigame ed il suo scopo è quello di verificare le collisioni con superfici o oggetti.

La differenza principale tra il minigame del basket e quello del calcio è la realizzazione della funzione `swipeTrajectory` che non deve più disegnare una vera e propria mira, ma deve rilevare uno swipe e richiamare la funzione che lancia la palla.

2.5. Minigame pallavolo

Il minigioco della pallavolo deve permettere al giocatore di spostare il giocatore lungo il bordo inferiore dello schermo per evitare di fare cadere la palla, per permettere all'utente ciò, sono state implementate 3 funzioni principali.

La prima funzione è la `volleyHit` che avvia il suono della palla, la seconda funzione si chiama `playerMovement` che permette di spostare il giocatore con il dito e l'ultima funzione implementata è la `collisionHandler` che come sempre gestisce le collisioni e l'assegnazione dei punti.

2.6. Minigioco tennis

Nell'implementazione del minigioco del tennis è possibile trovare una tipologia di interazione mediante hardware differente dal touch screen: il giroscopio.

La funzione, che permette al giocatore di spostare il giocatore inclinando il telefono in avanti e indietro, si chiama `playerMovementListener` e tramite `event.xGravity` sposta il giocatore sull'asse delle y per cercare di intercettare e colpire la pallina.

Questo gioco sfrutta anche la funzione `aiMovement` che fa muovere in maniera autonoma la "ai" avversaria; questa funzione, per comodità, attribuisce la posizione su y della palla alla "ai" in maniera tale da far seguire la pallina.

La funzione che sposta la pallina si chiama `ballMovement` e si occupa di dare un impulso iniziale alla pallina per poi lasciarla rimbalzare tra i due giocatori.

Come su tutti i minigiochi è presente la funzione `collisionHandler` che si occupa delle collisioni e delle differenti tipologie di assegnazione dei punti (nel caso di punteggi bonus).

2.7. Minigioco motorsport

Il file `motorsport` è strutturato in varie funzioni per suddividere il lavoro e rendere più leggibile il codice: le funzioni `pointToAngle` e `map` sono funzioni di utility e servono rispettivamente a trasformare una coordinata in un angolo e per convertire un valore da un intervallo ad un altro.

La funzione `movePlayer` si occupa di spostare il giocatore nella mappa, la funzione `moveAiTo` serve per muovere una "ai" verso il suo target e `moveAis` è la funzione che viene richiamata dal `eventListener`; mentre la funzione `rotatePlayerAi` serve per far ruotare le macchine del giocatore e delle "ai" verso dove stanno andando, al fine di rendere più reale il movimento.

Per il movimento delle "ai" avevamo preso in considerazione la libreria `jumper`, ma la abbiamo ritenuta una soluzione valida in caso di ostacoli dinamici ed essendo la mappa statica abbiamo deciso di implementare lo spostamento delle "ai" autonomamente.

La funzione `engineSoundVolume` serve per regolare i volumi dei motori in base alla velocità delle macchine e la funzione `cronometroUpdate` serve ad aggiornare il testo del cronometro.

La funzione `collisionHandler` svolge diverse attività: modifica la velocità del giocatore se entra o esce in una delle due zone scivolose, crea path da seguire per le "ai", controlla cronometro e posizione del giocatore e alla fine del minigioco presenta la schermata riassuntiva.

2.8. Metodologie di lavoro

Durante lo sviluppo del progetto, ci siamo suddivisi i compiti e abbiamo lavorato individualmente sui compiti e, in caso di problematiche, ci siamo confrontati per poter capire

quale potesse essere la soluzione. Abbiamo utilizzando GitHub per poter aggiornare da remoto il progetto e per tenerci aggiornati sui progressi, anche grazie a riunioni costanti.

I ruoli ricoperti sono stati:

- Gianluca: mappe, grafica (immagini e video) e implementazione gioco principale e minigioco motorsport
- Andrea: audio e implementazione minigiochi atletica, basket, calcio, pallavolo e tennis

Per gli elementi grafici abbiamo principalmente usato i siti kenney e freepik, alcune immagini sono state create da noi (quelle presenti nelle cartelle extra) e i video sono registrazioni del gioco, mentre per gli audio abbiamo usato principalmente freesound, il file game.mp3 è stato creato da noi.

Per la parte di produzione dei contenuti multimediali (prodotti da noi) abbiamo utilizzato dei tool come Garage Band (per gli audio), iMovie (per i video) e Gimp (per le immagini).

Abbiamo scelto di frequentare questo insegnamento (pur non essendo iscritti al corso di laurea in Scienze e Tecnologie Multimediali) per curiosità, da amanti dei videogiochi e della programmazione, abbiamo voluto sperimentare la fusione di questi due mondi.

3. Future implementazioni

Per una futura evoluzione del gioco, abbiamo pensato che si potrebbero integrare altri sport (palestre), con altri livelli di miglioramento delle singole palestre per poter avere dei guadagni maggiori nel gioco.

Una possibile implementazione della gestione dei punteggi e di una possibile classifica online potrebbe essere tramite una API Rest sviluppata in python, per quanto riguarda il front-end (il gioco in lua) basterebbe modificare il file salvataggio, effettuando delle http request all'api mediante le funzioni messe a disposizione dal linguaggio lua, mentre per il back-end (un server in python) che permetta al gioco di registrare gli avanzamenti e, a richiesta, di fornire il salvataggio.

Quest'ultimo potrebbe essere utilizzato per controllare l'erogazione delle entrate passive (in maniera tale da evitare le tipologie di sabotaggio o imbroglio).

```
1 local http = require("socket.http")
2 local ltn12 = require 'ltn12'
3
4 local body = {}
5 local res, code, headers, status = http.request{
6     url = "...",
7     sink = ltn12.sink.table(body)
8 }
9
10 local response = table.concat(body)
```