



UNIVERSITÀ
DI PARMA



Super Mario P2P

Anno Accademico 2020-2021

Filippo D'Addeo 294524

Filippo Gessati 294625

| | |
|---|-----------|
| INTRODUZIONE | 4 |
| FUNZIONAMENTO GENERALE | 4 |
| MODALITÀ DI GIOCO | 4 |
| HUD ED ELEMENTI DI GIOCO | 5 |
| HUD | 5 |
| MONETE | 5 |
| NEMICI | 6 |
| PIATTAFORME MOBILI | 7 |
| POWER UP | 7 |
| CHECKPOINT | 8 |
| FINE LIVELLO | 8 |
| SINGLEPLAYER: REGOLAMENTO E GAMEPLAY | 9 |
| MOVIMENTI CONSENTITI | 9 |
| NEMICI, VITE E RINASCITA | 9 |
| RACCOGLIERE MONETE | 10 |
| RACCOGLIERE POWER UP | 10 |
| FINIRE IL LIVELLO | 11 |
| FINIRE IL GIOCO E PUNTEGGIO | 11 |
| RIGIOCARE | 12 |
| MULTIPLAYER: REGOLAMENTO E GAMEPLAY | 13 |
| CONNESSIONE | 13 |
| MOVIMENTI CONSENTITI | 13 |
| NEMICI | 14 |
| GESTIONE VITE E RINASCITA | 14 |
| RACCOGLIERE MONETE | 15 |
| RACCOGLIERE POWER UP | 16 |
| PIATTAFORME MOBILI | 16 |
| FINIRE IL LIVELLO | 17 |
| FINIRE IL GIOCO E PUNTEGGIO | 17 |
| RIGIOCARE | 18 |
| TECNOLOGIE UTILIZZATE | 19 |
| PYTHON | 19 |
| PYTHON ARCADE | 19 |
| TILED | 19 |
| PYTHON BANYAN | 20 |
| PYTHON MYSQL-CONNECTOR | 20 |
| INSTALLAZIONE DEL GIOCO | 21 |
| REQUISITI | 21 |
| <i>Python</i> | 21 |
| <i>Python Arcade</i> | 21 |
| <i>Python Banyan</i> | 21 |
| <i>Python MySQL Connector</i> | 21 |
| <i>Xampp</i> | 21 |
| <i>MySQL Community</i> | 22 |
| <i>Requisiti di rete</i> | 22 |
| INSTALLARE SUPER MARIO P2P | 22 |

| | |
|------------------------|----|
| <i>Xampp</i> | 22 |
| <i>MySQL Community</i> | 25 |
| AVVIARE IL GIOCO | 26 |

Introduzione

Funzionamento generale

Lo sviluppo di Super Mario P2P ha avuto l'obiettivo di prendere un gioco che amiamo e aggiungere la possibilità non solo di poterci giocare su un computer, ma soprattutto di poterlo fare in multigiocatore con amici. Abbiamo cercato di offrire ai giocatori un'esperienza il più possibile simile al gioco ufficiale, aggiungendo ovviamente qualche elemento di originalità. Lo scopo è quello di completare tutti e quattro i differenti livelli nel minor tempo e con il maggior numero di vite rimaste possibili, così da comparire e rimanere in cima alla classifica dei migliori giocatori. I livelli aumentano gradualmente di difficoltà e, ovviamente, non mancano i nemici che cercano di mettere i bastoni tra le ruote: *Koopa*, *Goomba*, lava, piattaforme moventi e punte di ferro sono solo alcune delle difficoltà che il giocatore dovrà superare per completare i diversi livelli.

Modalità di gioco

Il gioco, una volta avviato, si presenta al giocatore con il menu principale fornendo, con tre possibilità di selezione, due modalità di gioco differenti: la prima è la classica modalità *singleplayer*, con la quale ci si può cimentare nei 4 differenti livelli di Super Mario P2P da soli; la seconda è invece quella della modalità cooperativa, cliccando infatti sul pulsante "*Multiplayer*" presente nel menu verrà visualizzato il proprio indirizzo IP su schermo, che andrà successivamente comunicato all'altro giocatore, il quale, a sua volta, dovrà selezionare dal menu la voce "*Friend code*", in modo che gli sia permesso di inserire l'indirizzo IP comunicatogli e di unirsi così alla partita.

HUD ed elementi di gioco

HUD

Lo HUD è costituito, in ordine da sinistra a destra, da tre elementi: un timer, un contatore per le vite e un contatore per le monete. Il primo indica il tempo trascorso dal momento in cui è iniziata la partita, il secondo informa il giocatore sulle vite rimaste, il terzo tiene traccia delle monete raccolte durante il gioco.



Figura 1: Gli elementi dello HUD cerchiati in rosso

Monete

Elemento fondamentale del gioco sono le monete, rappresentate come gettoni d'oro girevoli, che possono essere raccolti da qualsiasi giocatore per poter usufruire di alcuni vantaggi.

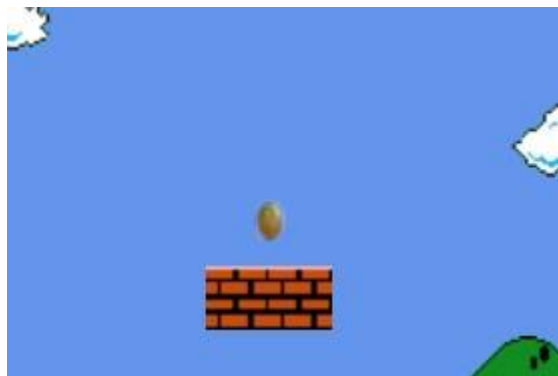


Figura 2: Una moneta

Nemici

All'interno del gioco sono inseriti diversi ostacoli il cui scopo è quello di aumentare il coefficiente di difficoltà di ogni livello. In caso di collisione tra uno di questi e il giocatore la conseguenza sarà sempre la medesima, ovvero la fine del gioco.

Questi ostacoli possono essere suddivisi in due categorie differenti: ostacoli statici e ostacoli dinamici.

La prima categoria, come suggerisce anche il nome, è caratterizzata da elementi che sono sempre nella stessa posizione. Ne fanno parte le pozzanghere di lava e le punte di ferro.



Figura 3: Una pozzanghera di lava con alcuni blocchi superiori utili per evitarla



Figura 4: Delle punte di ferro presenti nel livello 3

La seconda categoria invece è rappresentata da elementi che si muovono in determinate zone del livello. Ne fanno parte i personaggi che prendono il nome di *Goomba* e di *Koopa*.



Figura 5: Un Goomba



Figura 6: Un Koopa

Piattaforme mobili

Nel corso del quarto livello sono inserite alcune piattaforme moventi in prossimità delle quali, per evitare di finire nella lava o cadere nel vuoto, è fondamentale il tempismo del salto.



Figura 7: Una piattaforma mobile

Power up

In ogni livello è inserito un fiorellino che, se raccolto, permette di sparare palle di fuoco. Il giocatore che lo raccoglierà vedrà cambiare il colore della propria maglietta da verde a bianco.



Figura 8: Il fiorellino del power up



Figura 9: Il giocatore che ha raccolto il fiorellino e che ora vede la propria maglietta bianca



Figura 10: Il giocatore che ha raccolto il fiorellino e che ora può sparare le palle di fuoco

Checkpoint

A metà di ogni livello è collocato un checkpoint, simboleggiato da un'asta. Una volta raggiunto, il giocatore sentirà un suono caratteristico e vedrà issarsi la bandiera.

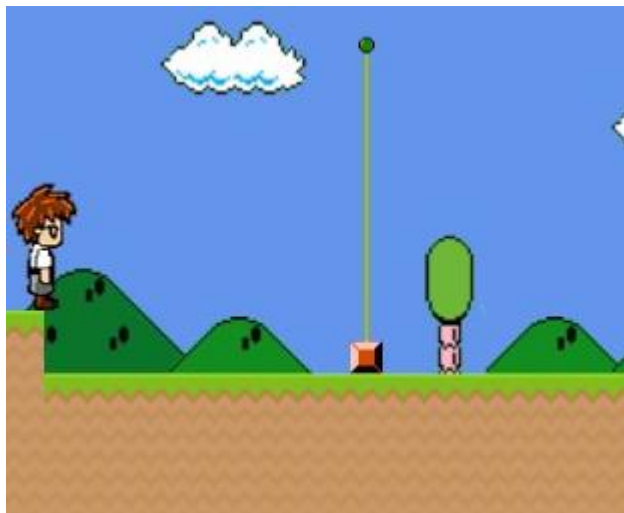


Figura 11: Il checkpoint che deve essere ancora raggiunto



Figura 12: Il checkpoint raggiunto con la bandiera issata

Fine livello

Alla fine di ogni livello è presente un portone. Una volta raggiunto dal giocatore, le due porte si apriranno e verrà suonata una musica trionfale.



Figura 13: Il portone di fine livello

Singleplayer: Regolamento e gameplay

Movimenti consentiti

I tasti principali con cui comandare il proprio personaggio sono quattro: il classico “W A S D” oppure in alternativa le frecce direzionali. Per effettuare un salto sarà necessario premere il tasto “W” o il tasto “freccia in alto”; per andare avanti e indietro si utilizzano rispettivamente “D” e “A” oppure “freccia destra” e “freccia sinistra”.

Qualora si raccolga un power up, sarà possibile premere “barra spaziatrice” per sparare una palla di fuoco.

Nemici, vite e rinascita

In caso di collisione tra il giocatore e uno qualsiasi dei nemici si perderà una vita. Se non si è ancora raggiunto il checkpoint si ritornerà all’inizio del livello, in caso contrario si rinascerà dal checkpoint stesso.

Le vite inizialmente a disposizione del giocatore sono cinque. Una volta finite, in caso di collisione con un nemico, oppure in caso di caduta nel vuoto, verrà mostrata una schermata di *game over* dalla quale sarà possibile premere il tasto “invio” per tornare al menu principale e poter così iniziare, se si desidera, una nuova partita.



Figura 14: Viene mostrato il giocatore che ha perso una vita (cerchiato in rosso) ma che è rinato al checkpoint in quanto era stato precedentemente raggiunto

Raccogliere monete

Le monete sono un aiuto dato al giocatore in quanto, una volta che si avrà collezionato un totale di dieci monete, queste verranno scambiate automaticamente con una nuova vita, che verrà aggiunta al contatore sempre presente e costantemente aggiornato nello HUD.

In particolare, per raccogliere una moneta sarà sufficiente entrare in contatto con una di essa.



Figura 15: Viene mostrato l'istante prima di raccogliere le monete



Figura 16: Viene mostrato l'istante in cui si raccolgono le monete con il contatore aggiornato

Raccogliere power up

I power up sono un secondo aiuto fornito al giocatore in quanto gli permettono di combattere contro i nemici dinamici, ovvero *Goomba* e *Koopa*. Una volta raccolto sarà possibile scagliare nella direzione in cui è rivolto il giocatore, premendo il tasto "barra spaziatrice", delle sfere infuocate, che se entreranno in contatto con un nemico ne determineranno l'eliminazione.

Una volta giunti alla fine del livello, oppure nel caso si dovesse perdere una vita, il power up andrà perso e sarà necessario aspettare il livello successivo per ottenerlo nuovamente, in quanto per ogni livello ne è inserito uno solo.



Figura 17: Il giocatore che ha raccolto il fiorellino e che ora può sparare le palle di fuoco

Finire il livello

Una volta raggiunto il portone di fine livello, esso si aprirà suonando una musica trionfale. Verrà così salvato il tempo impiegato per terminare il livello corrente e una volta terminata la musica verrà caricato e visualizzato il livello successivo.



Figura 18: Viene mostrata la fine del livello con le porte aperte

Finire il gioco e punteggio

Una volta concluso il quarto e ultimo livello il gioco terminerà e verrà visualizzata una schermata contenente il tempo totale impiegato e le vite rimanenti al termine della partita. In tale schermata sarà presente anche una piccola classifica che visualizzerà le cinque migliori partite precedenti con il relativo nome del giocatore, tempo impiegato e vite rimaste.

Per poter salvare i propri risultati e quindi avere la possibilità di apparire nella classifica, sarà sufficiente inserire il proprio nome composto da non più di tre lettere nell'apposito spazio, e premere il pulsante "Records" così da salvare i dati della partita appena conclusa in un database locale.

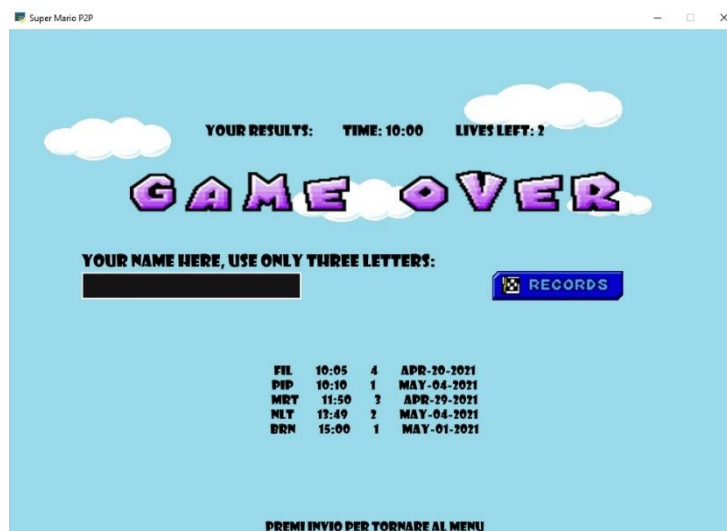


Figura 19: Viene mostrata la schermata di fine gioco con la classifica delle partite singleplayer ordinata per tempo impiegato e il proprio risultato

Rigiocare

È possibile ritentare il gioco dalla schermata di *game over* o da quella finale. Per farlo sarà sufficiente premere il tasto “Invio” per tornare al menu principale, da dove si potrà scegliere se iniziare una nuova partita o in alternativa chiudere il gioco.



Figura 20: La schermata di game over con le istruzioni per tornare al menu cerchiate in rosso

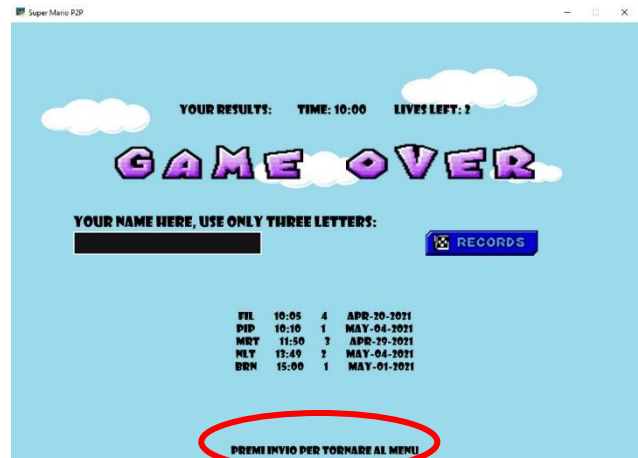


Figura 21: La schermata di fine gioco con le istruzioni per tornare al menu cerchiate in rosso

Multiplayer: Regolamento e gameplay

Connessione

Per poter giocare in modalità multiplayer sarà necessario connettersi al backplane. Quest'ultimo verrà creato dal giocatore che nel menu principale selezionerà la voce "Multiplayer", al quale successivamente verrà mostrata una schermata contenente il proprio indirizzo IP. Il secondo giocatore, che dovrà selezionare la voce "Friend Code", inserirà questo indirizzo IP nell'apposito spazio mostratogli a schermo per connettersi al compagno e iniziare così la partita.



Figura 22: La schermata con il proprio indirizzo IP che viene visualizzata dopo aver premuto sulla voce "Multiplayer"



Figura 23: La schermata dove inserire l'indirizzo IP al quale ci si vuole connettere che viene visualizzata dopo aver premuto sulla voce "Friend code"

Movimenti consentiti

I tasti principali con cui comandare il proprio personaggio sono quattro: il classico "W A S D" oppure in alternativa le frecce direzionali. Per effettuare un salto sarà necessario premere il tasto "W" o il tasto "freccia in alto"; per andare avanti e indietro si utilizzano rispettivamente "D" e "A" oppure "freccia destra" e "freccia sinistra".

Qualora si raccolga un power up sarà possibile premere "barra spaziatrice" per sparare una palla di fuoco.

In quanto ci si trova in modalità "Multiplayer" qualsiasi azione effettuata da un giocatore sarà trasmessa anche all'altro tramite uno scambio di messaggi; di conseguenza ogni giocatore nel proprio schermo vedrà anche il personaggio dell'altro giocatore muoversi, identificandolo con la maglietta rossa.



Figura 24: Il proprio personaggio viene mostrato con la maglietta verde mentre il personaggio dell'altro peer viene mostrato con la maglietta rossa

Nemici

Il giocatore che ha avviato il *backplane* è incaricato, con il proseguire della partita, di comunicare il movimento dei *Goomba* e dei *Koopa* all'altro peer. In quest'ottica è dunque possibile vedere il giocatore che ha dato il via al *backplane* come una sorta di *super peer*, in quanto ha dei doveri maggiori rispetto a chi si connette. Tale incarico ovviamente è completamente trasparente all'utente e viene gestito esclusivamente dal software.

Esattamente come per il singleplayer, il contatto con uno qualsiasi dei nemici porterà alla perdita di una vita.

Gestione vite e rinascita

Come per il singleplayer le vite inizialmente disponibili sono cinque e in comune ad entrambi i giocatori. In questo modo nel caso un giocatore dovesse perdere una vita, dovrà comunicarlo al secondo peer che provvederà a sua volta a decrementare il numero di vite rimaste. Tale numero sarà sempre visibile e costantemente aggiornato nello HUD di ciascun peer. Anche in questo caso il meccanismo di comunicazione tra i due peer sarà completamente trasparente per l'utente, essendo gestito via software.

Quando un giocatore entra in contatto con un nemico, diversamente dal singleplayer, sarà necessario distinguere due diversi casi: un caso in cui uno dei due peer sia ancora in vita e un caso in cui entrambi non lo siano più. Nel primo caso, il giocatore rinascerà sotto forma di lapide nella posizione del secondo peer, muovendosi in su e in giù finché non verrà toccato dal proprio compagno venendo riportato così in vita; nel secondo caso invece, il giocatore rinascerà all'inizio del livello oppure in prossimità del checkpoint nel caso sia stato precedentemente raggiunto da uno dei due giocatori.

Una volta terminate le vite comuni, in caso di collisione con un nemico oppure in caso di caduta nel vuoto da parte di uno dei due peer, verrà mostrata una schermata di *game over* ad entrambi i giocatori decretando la fine della partita e dalla quale sarà possibile premere "invio" per tornare al menu principale.



Figura 25: Si vede la lapide comparire nel punto in cui ci troviamo nel momento in cui l'altro giocatore ha perso una vita

Raccogliere monete

Come per le vite anche le monete sono in comune ad entrambi i giocatori. Queste sono per loro un aiuto, in quanto una volta che si avrà collezionato un totale di dieci monete, queste verranno scambiate automaticamente con una nuova vita, che verrà aggiunta al contatore sempre presente e costantemente aggiornato nello HUD di ogni peer.

Essendo comuni, le monete raccolte da un peer scompariranno dallo schermo dell'altro ed entrambi vedranno il contatore delle monete raccolte aggiornarsi.



Figura 26: Viene mostrato l'istante in cui l'altro giocatore non ha ancora raccolto le monete



Figura 27: Viene mostrato l'istante in cui l'altro giocatore sta raccogliendo le monete con il contatore che si incrementa

Raccogliere power up

I power up sono un secondo aiuto fornito ai giocatori in quanto permettono di combattere contro i nemici dinamici, ovvero *Goomba* e *Koopa*.

Una volta raccolto da un giocatore gli sarà possibile scagliare nella direzione in cui è rivolto, premendo il tasto "barra spaziatrice", delle sfere infuocate, che se entreranno in contatto con un nemico ne determineranno l'eliminazione dal gioco di entrambi i peer.

Il power up può essere raccolto solo da un giocatore ad ogni livello e inoltre il peer che lo raccoglierà potrà sparare una sola palla di fuoco alla volta, aspettando dunque che quella precedente esploda prima di poter lanciare quella successiva.

Una volta giunti alla fine del livello, oppure nel caso si dovesse perdere una vita, il power up andrà perso e sarà necessario aspettare il prossimo livello per ottenerlo nuovamente in quanto per ogni livello ne è inserito uno solo.



Figura 28: Viene mostrato il power up che deve essere ancora raccolto



Figura 29: Viene mostrato il power up è stato raccolto dall'altro giocatore che ora ha cambiato il colore della maglietta

Piattaforme mobili

Come accade per i nemici il giocatore che ha avviato il backplane è incaricato, con il proseguire della partita, di comunicare il movimento delle piattaforme mobili presenti nel quarto livello all'altro peer, rafforzando così la sua figura di *super peer*.

Finire il livello

Una volta raggiunto il portone di fine livello da entrambi i peer si aprirà suonando una musica trionfale udibile da entrambi i giocatori. Verrà così salvato il tempo impiegato per concludere il livello e una volta terminata la musica verrà caricato e visualizzato il livello successivo.

Essendo un gioco cooperativo come tempo viene preso in considerazione quello impiegato dall'ultimo giocatore che giunge al portone, così da incentivare la cooperazione tra i due peer.



Figura 30: Viene mostrata la fine del livello con le porte aperte

Finire il gioco e punteggio

Una volta concluso il quarto e ultimo livello il gioco terminerà e verrà visualizzata ad entrambi i peer una schermata contenente il tempo totale impiegato per concludere la partita e le vite rimanenti. In tale schermata sarà presente anche una piccola classifica che visualizzerà le cinque migliori partite precedenti con il relativo nome dei giocatori, tempo impiegato e vite rimaste.

Per poter salvare i propri risultati e quindi avere la possibilità di apparire nella classifica, sarà sufficiente inserire il proprio nome composto da non più di tre lettere nell'apposito spazio, e premere il pulsante "Records" così da salvare i dati della partita appena conclusa in un database locale.

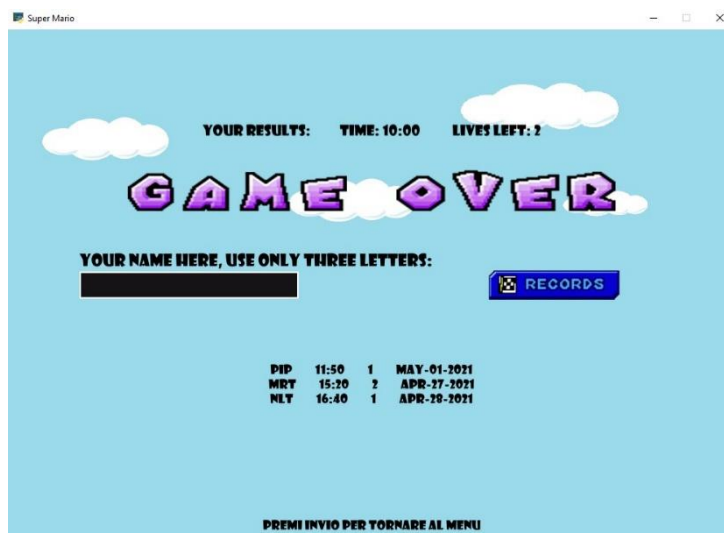


Figura 31: Viene mostrata la schermata di fine gioco con la classifica delle partite multiplayer ordinata per tempo impiegato e il proprio risultato

Rigiocare

È possibile ritentare il gioco dalla schermata di *game over* o da quella finale. Per farlo sarà sufficiente premere il tasto “Invio” per tornare al menu principale, da dove si potrà scegliere se iniziare una nuova partita o in alternativa chiudere il gioco.

In questo caso se si dovesse scegliere di ripetere il gioco in modalità *multiplayer* allora sarà necessario ripetere la procedura di connessione.



Figura 32: La schermata di game over con le istruzioni per tornare al menu cerchiare in rosso



Figura 33: La schermata di fine gioco con le istruzioni per tornare al menu cerchiare in rosso

Tecnologie utilizzate

Python

La scelta di Python rispetto ad altri linguaggi di programmazione è frutto di diversi ragionamenti e motivazioni. Primo fra tutti l'esperienza: avendo già scritto una nostra versione del famoso videogioco "Frogger" eravamo già in possesso delle basi necessarie per lo sviluppo di giochi Arcade. Inoltre, il linguaggio gode di buone librerie, semplici ed intuitive, per il design di giochi in 2D. Volendo offrire un gioco in versione desktop, senza l'ausilio del browser, abbiamo appurato che Python offre semplicità e velocità nella stesura del codice, con buone prestazioni in quanto non si tratta di un gioco tridimensionale con alti requisiti prestazionali.

Fonte: <https://www.python.org/>

Python Arcade

Arcade è una libreria *easy-to-learn* di Python per creare videogiochi in 2D costruita al di sopra di Pyglet e OpenGL.

La scelta della libreria da utilizzare per lo sviluppo era proprio tra Python Arcade e Pygame, tenendo presente che il secondo risulta maggiormente utilizzato dalla community. Tuttavia, Arcade può vantare di un sito web ben fatto che include tutta la documentazione necessaria per poter comprendere ed utilizzare la libreria. Essendo sviluppata con l'obiettivo di essere utilizzata per la realizzazione di giochi arcade 2D, sul sito sono presenti molteplici esempi ben commentati e facilmente analizzabili. Oltre alla documentazione, sicuramente molto valida, Python Arcade presenta una sintassi molto più concisa e facile da comprendere, basandosi inoltre su OpenGL3+, contrariamente a Pygame che si basa su una libreria grafica più vecchia ed obsoleta: SDL1.

Un aspetto chiave che ci ha fatto propendere per Arcade è stato il supporto e gli aggiornamenti costanti che questa libreria riceve, al contrario di Pygame, che essendo un po' più datato, viene aggiornato non costantemente. Inoltre, Arcade incoraggia il programmatore a separare il codice che regola la logica di gioco dal codice che permette il *display* dei vari elementi, rendendo il programma più leggibile.

Fonte: https://arcade.academy/get_started.html

Tiled

Tiled è un programma *open source* per sviluppare livelli di gioco in modo grafico, importando nel software le proprie immagini, in modo da utilizzarle come unità costituenti della mappa. Il livello può essere salvato in vari formati, tra cui, quello da noi scelto, il CSV. Il programma velocizza enormemente il *game design* in quanto si ha una vista completa della mappa e non sono presenti necessità di nessun tipo di rendering e/o attese, infatti il blocco inserito all'interno della mappa viene visualizzato immediatamente.

In maniera simile ad altri programmi di grafica è possibile definire vari *layer*, ovvero una gerarchia tra gli elementi che fanno parte dello stesso livello. In questo modo nella mappa di gioco gli elementi appartenenti ad un *layer* più alto andranno a coprire quegli elementi che fanno parte dei *layer* inferiori, permettendo così di inserire, per esempio, elementi di *foreground* e *background*.

Questa divisione risulta molto utile in quanto tutti gli oggetti di un determinato *layer* possono essere importati su Python Arcade ed inseriti direttamente in una lista specifica; questo è fondamentale per la logica

del gioco dato che, nel nostro caso, tutti i nemici saranno contenuti nella lista *“enemy_list”* riempita utilizzando gli elementi che fanno parte del *layer “enemy”* di Tiled.

Il programma offre anche la possibilità di definire oggetti in movimento, specificando la loro velocità e gli estremi sinistro e destro della loro traversata.

Fonte: <https://www.mapeditor.org/>

Python Banyan

Python Banyan è un framework leggero e reattivo utilizzabile per creare applicazioni flessibili, non bloccanti, asincrone, *event-driven*. Utilizza un'estensione del modello orientato agli oggetti chiamato modello *component-based service-oriented architectural*. Un'applicazione *component-based* consiste in un insieme di componenti indipendenti e *loosely coupled*. Per python-banyan questi componenti comunicano fra di loro attraverso un insieme di messaggi di protocollo definiti dall'utente e *platform-independent* trasmessi lungo un *network TCP*.

Banyan utilizza un meccanismo centrale di distribuzione dei messaggi chiamato *Backplane*. Quando si esegue un'applicazione Banyan, viene avviato come prima cosa il backplane, seguito dai componenti dell'applicazione. All'avvio ogni componente si collega automaticamente al Backplane aprendo un canale di comunicazione TCP/IP tra sé stesso e il *backplane*. I componenti di un'applicazione Banyan possono essere eseguiti su un singolo computer oppure attraverso diversi computer.

I messaggi di protocollo di Banyan consistono in due parti: un *topic* e un *payload*. Il *topic* del messaggio consiste in una semplice stringa. Affinché un componente riceva messaggi con determinati *topics*, deve sottoscrivere per ricevere messaggi ad ognuno dei *topic*. Un componente sottoscritto riceve tutti i messaggi pubblicati che contengono quel particolare *topic* desiderato. Il *payload* del messaggio consiste in una struttura di tipo *Python Dictionary*, contenente una o più coppie chiave/valore. Il vantaggio di Banyan nell'utilizzare un *dictionary* consiste nel fatto che i dati del messaggio possono essere velocemente de-referenziati e processati.

Fonte: https://mrslab.github.io/python_banyan/

Python mysql-connector

Python mysql-connector implementa il MySQL driver che abilita i programmi Python ad accedere ai databases MySQL, utilizzando una API che è conforme alla *Python Database API Specification V2.0*.

Fonte: <https://pypi.org/project/mysql-connector-python/>

Installazione del gioco

Requisiti

Per poter giocare a Super Mario P2P è prima necessario installare:

Python

Per installare Python è necessario aprire il proprio browser e recarsi a questo link: <https://www.python.it/download/>

Una volta scaricato il file eseguibile corretto per il proprio sistema operativo, lo si dovrà lanciare e sarà necessario eseguire tutti i passaggi che verranno richiesti. Una volta terminato il *setup* Python sarà installato sul vostro computer e potrete procedere con i prossimi requisiti.

Python Arcade

Su Windows aprire un terminale ed eseguire il comando:

```
pip install arcade
```

Su MacOS aprire un terminale ed eseguire il comando:

```
pip3 install arcade
```

Python Banyan

Su Windows aprire un terminale ed eseguire il comando:

```
pip install python-banyan
```

Su MacOS aprire un terminale e digitare il comando:

```
pip3 install python-banyan
```

Python MySQL Connector

Su Windows aprire un terminale ed eseguire il comando:

```
pip install mysql-connector-python
```

Su MacOS aprire un terminale ed eseguire il comando:

```
pip3 install mysql-connector-python
```

Xampp

In alternativa a MySQL Community è possibile installare Xampp.

Per installarlo sarà sufficiente andare al seguente link: <https://www.apachefriends.org/it/download.html>

Una volta scaricato l'eseguibile apposito per il proprio sistema operativo bisognerà proseguire con l'installazione seguendo le istruzioni a schermo.

MySQL Community

In alternativa a Xampp è possibile installare MySQL Community.

Per installarlo sarà sufficiente andare al seguente link: <https://dev.mysql.com/downloads/mysql/>

Una volta scaricato l'eseguibile apposito per il proprio sistema operativo bisognerà proseguire con l'installazione seguendo le istruzioni a schermo.

Requisiti di rete

Per poter giocare online è necessario verificare che i computers siano visibili in rete. In particolare, se si dispone di un pc Windows, è fondamentale controllare che il gruppo Home sia corretto e congruente con le impostazioni di rete.

Installare Super Mario P2P

Se si possiede un computer Windows come prima cosa bisogna scaricare e installare Git, reperibile a questo indirizzo: <https://gitforwindows.org/>.

Successivamente aprire un terminale nella cartella in cui si desidera installare Super Mario P2P e digitare il comando:

```
git clone https://github.com/tomanabeach/SuperMarioP2P.git
```

Una volta scaricato sarà necessario procedere alla creazione del database locale contenente le informazioni necessarie per popolare i livelli. Per farlo è sufficiente utilizzare i due file .sql che si trovano all'interno della cartella "/DBCcreation".

Xampp

Se si utilizza Xampp è necessario innanzitutto avviare i moduli apache e MySQL all'interno del pannello di controllo. Successivamente si dovrà aprire il proprio browser e digitare: <http://localhost/dashboard/>, per poi cliccare sulla voce "phpMyAdmin".

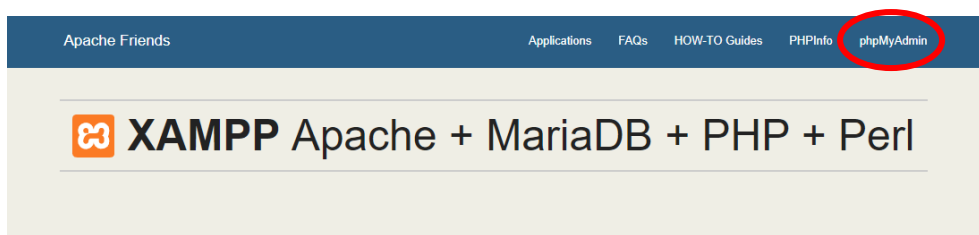


Figura 34: Ciò che viene visualizzato all'indirizzo <http://localhost/dashboard/>

In questo modo si verrà reindirizzati alla pagina di gestione dei database, dalla quale si dovranno importare i due file .sql per poter creare e popolare il database.

Si dovrà dunque cliccare sulla voce “importa”.

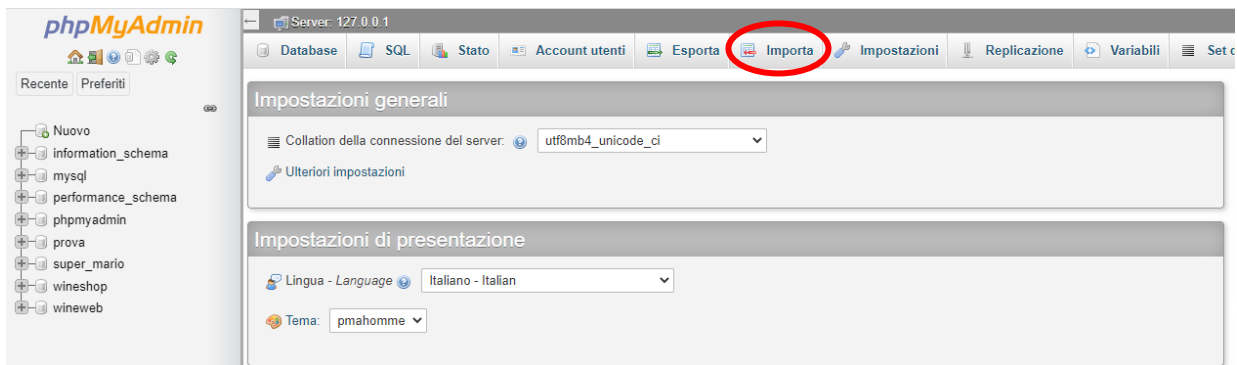


Figura 35: La voce "Importa" cerchiata in rosso

Successivamente bisognerà cliccare su “scegli file” e selezionare il file “createTable.sql” all’interno della cartella “/DBCcreation”. Premendo il tasto “esegui” si completerà l’operazione.

Ripetere poi lo stesso procedimento per il file “insertQuery.sql”.

Importazione nel server corrente in corso

File da importare:

Il file può essere compresso (gzip, bzip2, zip) o non compresso.
Il nome di un file compresso deve terminare in `.[formato].[compressione]`. Ad esempio: `.sql.zip`

Cerca sul tuo computer: **Scegli file** Nessun file selezionato (Dimensione massima: 40MiB)

Puoi anche trascinare e rilasciare un file su qualsiasi pagina.

Set di caratteri del file: `utf-8`

Importazione parziale:

☒ Consenti l'interruzione del processo di importazione nel caso lo script rilevati di essere troppo vicino al timeout di PHP. (Questo potrebbe essere un buon modo di importare grandi file, tuttavia può interrompere le transazioni.)

Salta questo numero di query (per SQL) partendo dalla prima:

Altre opzioni:

☒ Abilita i controlli sulle chiavi esterne

Formato:

`SQL`

Opzioni specifiche al formato:

Modalità di compatibilità SQL: `NONE`

☒ Non usare AUTO_INCREMENT per il valori zero

Esegui

Figura 36: Vengono mostrati i tasti da premere cerchiati in rosso

In questo modo il database sarà pronto. È importante notare che Xampp non dovrà assolutamente essere chiuso per tutta la durata della sessione di gioco.

Il prossimo e ultimo passaggio è quello di accertarsi che la password all'interno dello script Python per accedere al database sia corretta. Nelle prime righe di codice all'interno dei file "single_player.py", "multi_player.py" e "game_won.py" è definita la funzione "database_connection" utile per connettersi al database. Qui si dovrà controllare che la password sia "".

```
def database_connection():  
    """ Connect to our MySQL database"""  
  
    connection = None  
    try:  
        connection = mysql.connector.connect(host = "localhost", database = "super_mario", user = "root", password = "")  
        if connection.is_connected():  
            print("Connected to MySQL database")  
  
    except Error as e:  
        print(e)  
  
    return connection
```

Figura 37: La funzione database_connection

MySQL Community

Se si utilizza MySQL Community è necessario come prima cosa avviare “MySQL Workbench CE”. Successivamente, se non si dispone già di una *Local instance MySQL*, sarà necessario crearne una.

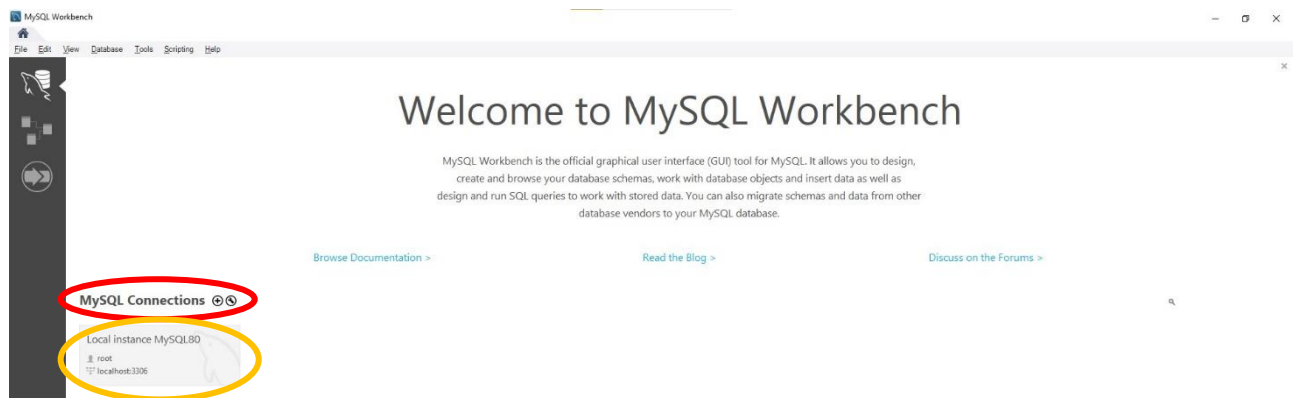


Figura 38: In rosso il pulsante per creare una nuova connessione e in arancione l'istanza locale già presente e per la quale non è necessario creare una nuova connessione

A questo punto bisognerà cliccare sull'istanza appena creata, oppure che era già presente, e verificare nel pannello di amministrazione che il server MySQL sia in esecuzione. In caso contrario sarà necessario avviarlo premendo sulla voce “Startup/Shutdown” e poi sul pulsante “Start server” inserendo la password utilizzata al momento della creazione dell'istanza stessa.



Figura 40: La pagina per avviare il server MySQL con il pulsante cerchiato in rosso

Figura 39: Il pannello di amministrazione con la voce per avviare il server MySQL cerchiata in rosso

Ora bisognerà importare i due file .sql che contengono le query per la creazione e la popolazione del database. Per farlo sarà necessario cliccare sul pulsante di importazione posto nel menu in alto a sinistra. Una volta selezionati i due file all'interno della cartella "/DBCreation", questi verranno aperti e visualizzati all'interno dell'editor di MySQL Workbench. Ora non rimarrà altro che eseguire le query contenute cliccando il relativo bottone.



Figura 41: Il pulsante per l'importazione dei file

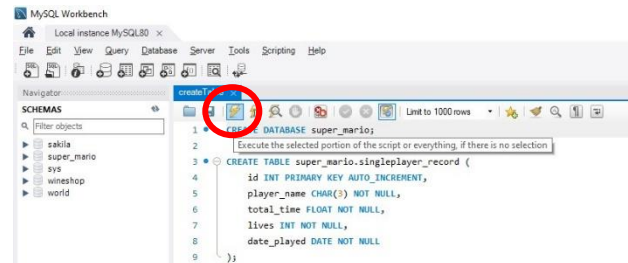


Figura 42: Il pulsante per l'esecuzione delle query

In questo modo il database sarà pronto. È importante notare che MySQL non dovrà essere chiuso per tutta la durata della sessione di gioco.

Il prossimo e ultimo passaggio è quello di accertarsi che la password all'interno dello script Python per accedere al database sia corretta. Nelle prime righe di codice all'interno dei file "single_player.py", "multi_player.py" e "game_won.py" è definita la funzione "database_connection" utile per connettersi al database. Qui si dovrà controllare che la password sia la stessa utilizzata al momento della creazione dell'istanza locale di MySQL Workbench.

```
def database_connection():
    """ Connect to our MySQL database """

    connection = None
    try:
        connection = mysql.connector.connect(host = "localhost", database = "super_mario", user = "root", password = "root")
        if connection.is_connected():
            print("Connected to MySQL database")

    except Error as e:
        print(e)

    return connection
```

Figura 43: La funzione database_connection, in questo caso la password è "root"

Avviare il gioco

Come prima cosa avviare Xampp o MySQL Workbench.

Su Windows aprire un terminale nella directory in cui è stato scaricato il gioco ed eseguire i seguenti comandi:

```
cd SuperMarioP2P-main/
./menu.py
```

Su MacOS aprire un terminale nella directory in cui è stato scaricato il gioco ed eseguire i seguenti comandi:

```
cd SuperMarioP2P-main/
python3 menu.py
```