I.T.T Montani
ITT "MONTANI" Fermo Corso Marconi, 35 63900 Fermo (FM)
numero: 0734/622632
email certificata: APTF010002@pec.istruzione.it

Sistemi e reti

Riepilogo della relazione

Data della relazione:	Nome relazione:	Preparato da:
15/10/2021	Quiz a scelta multipla con comunicazione client/server	Nucci Enrique, Raffaeli Giulio

Indice

ANALISI

Si richiede di sviluppare un'applicazione client-server con implementato un classico gioco quiz a domanda e risposta. Poiché stiamo sviluppando un'applicazione client-server abbiamo bisogno di separare le due parti, creando appunto un software client, ed un software server. Innanzitutto bisogna trovare un modo per creare la connessione client-server. Il metodo piú opportuno è quello di utilizzare i socket, dichiarando manualmente il port number e un indirizzo ip (che in questo caso sarà quello locale, poiché il server viene aperto nella stessa rete del client), necessari per effettuare la connessione.
Una volta aperto il server ci verrà notificato tramite un messaggio che il server è pronto per la connessione di uno o più client.
Dopo aver acceso il server si potrà finalmente eseguire il client, che è predisposto di un'interfaccia grafica, la prima cosa che ci verrà mostrata è una schermata iniziale con un video in riproduzione in background, un logo rappresentante il nome del programma e un tasto per andare avanti nell'esecuzione.
Dopo aver premuto il tasto avanti verrà visualizzata una nuova schermata dove ci verrà richiesto di inserire un nickname, Il nickname è univoco, infatti se provassimo ad inserire un nickname già presente nell'archivio il gioco visualizzerebbe un messaggio di errore, chiedendo di inserire un nuovo nickname.

Dopo di ciò il programma offre quattro diverse opzioni:

- Partita classificata;
- Partita per argomento;
- Classifica;
- Esci.

Nel caso in cui venisse premuto il pulsante "Partita classificata", inizierà il gioco, ci saranno una sequenza di dieci domande casuali di quattro argomenti principali (Geografia, Storia, Informatica, Scienze).
L'utente si ritroverà quindi a dover rispondere alla domanda tramite uno dei quattro pulsanti contenenti le possibili risposte, se la risposta data dall'utente è corretta verranno aggiunti 10 punti al punteggio totale, in caso contrario non verranno aggiunti punti e si procederà direttamente alla prossima domanda.
Al fine del quiz ci sarà una schermata che visualizza il punteggio totale e un tasto per tornare al menù principale, il punteggio ed il nickname dell'utente verranno inseriti all'interno della classifica.

Se l'utente seleziona il pulsante "Partita per argomento" verrà disposto ad una scelta tra quattro argomenti diversi:

- Storia;
- Geografia;
- Informatica;
- Scienze;

Selezionando un argomento, il programma procederà con il quiz, le domande che verranno visualizzate saranno tutte inerenti all'argomento che si sceglie nel menù precedente. Le modalità di gioco sono le stesse presenti nella modalità "Partita classificata", tranne per il fatto che il punteggio e il nickname dell'utente non verranno inseriti all'interno della classifica.
Infine, premendo il tasto "Classifica" il programma visualizza una schermata contenente la classifica dei 3 migliori giocatori in base al punteggio accumulato.

FUNZIONALITÀ AGGIUNTIVE

- INTERFACCIA GRAFICA
  È stato scelto di supportare il quiz con un'interfaccia grafica per facilitare l'esperienza dell'utente nel navigare nei vari menu del quiz, usufruendo di pulsanti interattivi sia per scegliere le categoria, sia per scegliere le diverse risposte all'interno del quiz
- SUONI
  Sono stati aggiunti suoni in caso l'utente clicchi su un pulsante e a seconda che la risposta selezionata sia Corretta o Sbagliata
- CLASSIFICA
  E stata aggiunta una classifica per permettere all'utente di visualizzare i migliori risultati ottenuti e, dopo aver effettuato una partita, poter salvare il proprio risultato

CLIENT

Il Software client è utilizzato per gestire l'interazione con l'utente e contiene tutte le funzioni necessarie al funzionamento corretto del gioco.
Per sviluppare l'interfaccia grafica é stata utilizzata la libreria aggiuntiva di python "pygame", che contiene funzionalità quali gestione del refresh dello schermo, creazione di elementi software come rettangoli e gestione delle interazioni dell'utente.
Inoltre al fine di rendere le variabili e le funzioni del programma riutilizzabili, il gioco è stato creato all'interno di una classe denominata "Game".
Cosa analoga è avvenuta nella creazione dei pulsanti con cui l'utente può interagire nel corso del gioco, salvati nella classe "Pulsanti".
Per prima cosa bisogna creare la finestra su cui sarà visualizzato il gioco; nel nostro caso é stata creata una finestra di dimensione 1440 x 720 pixel, rinominata "QUIZ TIME".
Successivamente si parte con il metodo setup(), che al suo interno gestisce la connessione iniziale con il Server tramite un Socket avente host e port number preconfigurati.
Dopo aver verificato che la connessione sia stata instaurata con successo, si passa al metodo Title(), che ha il compito di mostrare la schermata iniziale all'utente.
In questa schermata è presente un video in riproduzione in background, un'immagine che mostra il titolo del gioco ed un pulsante.
Per mostrare il video in background è stato necessario configurare un clock che ricarica la finestra 60 volte al secondo e utilizzare la libreria cv2. Grazie a questa libreria è stato possibile acquisire un video dalla cartella "Assets" tramite la funzione VideoCapture() e successivamente dividere il filmato e mostrarlo frame per frame, tramite la funzione read(). Ogni volta che un frame viene visualizzato inoltre aumenta un contatore che, in caso raggiunga il numero di frame totali del video, lo fa ricominciare da capo, evitando che il video si stoppi dopo un paio di secondi.
Il pulsante invece é un rettangolo della libreria pygame.Rect a cui é stata assegnata un'immagine e che controlla quando l'utente clicca col mouse su di esso.
In caso ció avviene si passa al metodo getNick(). Lo scopo di questa funzione è di acquisire il nome utente che verrá successivamente visualizzato sulla classifica.

Per acquisire il nome é stata creata un' input box che acquisisce le lettere che l'utente digita sulla tastiera. Per evitare errori di lunghezza del nome, sono stati inseriti appositi controlli che verificano se il nome ha meno di 3 caratteri o ha piú di 6 caratteri.

In entrambi casi sarà visualizzato un messaggio di avviso che informa l'utente. In caso l'utente digita un nickname valido, il Client manda il suddetto nome al server e si mette in ascolto, aspettando che il server verifichi se esso sia già presente all'interno della classifica o meno.

In caso ció sia vero, il metodo getNick() verrà richiamato e verrà mostrato un messaggio all'utente informandolo che il nome é già in uso, altrimenti si procede al metodo MenuPrincipale().

All'interno di questa funzione verranno mostrati quattro pulsanti (creati usando la classe "Pulsanti" citata all'inizio) :

-Partita Classificata

-Partita Per Argomento

-Classifica

-Esci

Per migliorare la visualizzazione di tali pulsanti, è stato creato un metodo mouseSopra(), che in caso il mouse ci passi sopra (anche senza cliccare), li faccia illuminare.

A seconda di quale pulsante l'utente clicchi, si passa ad una funzione diversa, in ogni caso peró, per segnalare la selezione dell'utente, viene inviato un messaggio contenente un valore da 1 a 3, corrispondente alla scelta effettuata.

In caso si scelga di visualizzare la classifica, si passerà al metodo Classifica().

Al suo interno sono state create tre variabili, top 1, top 2 e top 3, il cui valore è stato acquisito tramite un recv() da parte del server e verranno visualizzate a schermo una sopra l'altra colorate di oro, argento e bronzo.

Per tornare indietro è stato creato un pulsante (utilizzando una classe "PulsanteIndi", che però ha le stesse funzionalità della classe "Pulsanti") che in caso venga cliccato, richiama il metodo MenuPrincipale().

In caso si scelga il pulsante "Partita Classificata", si passa al metodo PartitaVeloce().

Prima di fare ciò vengono inizializzate inoltre le variabili round (che una volta arrivato a 10 termina il gioco) e score.

Una volta entrati nella funzione , riceve dal server 5 messaggi, il primo corrisponde alla domanda e i successivi alle quattro possibili risposte.

A questo punto vengono generati quattro pulsanti utilizzando l'omonima classe e mostra a schermo il tutto.

Una volta che l'utente seleziona uno dei bottoni, invia una stringa al server, contenente la selezione dell'utente e attende una risposta.

In caso il messaggio che riceve è "Giusto", aumenta la variabile score di 10, altrimenti incrementa solo il contatore round e passa alla prossima domanda, ricominciando la funzione.

Una volta che il contatore round arriva a 10 si passa al metodo schermataScore(), che calcola il punteggio totale e lo converte in stringa. Fatto ciò manda la suddetta stringa al server che salverà il risultato nella classifica.

A questo punto comparirà un tasto "Continua", che se cliccato riporterà l'utente al menuPrincipale().

Ultimo pulsante da poter premere nel menu è "Partita per argomento". Una volta cliccato si passerà al metodo selezionaCategoria(), dove compariranno altri quattro pulsanti corrispondenti alle quattro possibili categorie (storia, geografia, scienze, informatica). A seconda del pulsante premuto, il Client manderá al server un valore compreso tra 1 e 4, che indicherá al server quale categoria selezionare.

A questo punto si passerà alla funzione PartitaVeloce() e l'utente giocherà come se fosse una partita normale.

SERVER

Il software server server per permettere ai diversi client di connettersi e ricevere tutti i dati necessari per il funzionamento del gioco.
Come prima cosa per far diventare il programma un server a cui i client possono connettersi è necessario creare un Socket a cui verrà poi inserita una porta ed un indirizzo ip (che in questo caso è il local host).
Dopo fatto ciò il server rimarrà in attesa della connessione di un client e dopo di che assegnerà un thread a gestirlo, permettendo così la possibilità di connettere più client.
Il server rimarrà in ascolto finché un client non invierà un nickname che verrà utilizzato come nome del thread.
Il nickname viene controllato tramite un loop che cercherà il nome inserito all'interno di una lista contenente il file dove sono salvati tutti i nickname e la classifica, controllando se il nickname inserito dall'utente sia diverso da tutti quelli presenti nel file. Il controllo viene effettuato cercando carattere per carattere la stringa all'interno della lista e restituendo 1 al client in caso il nickname sia uguale ad uno presente nel file, chiedendo nuovamente all'utente di inserire un nuovo nickname.
Dopo di ciò il server rimarrà in ascolto aspettando una scelta da parte del client, questo viene fatto in un ciclo infinito così da mantenere la ricezione sempre aperta fino a chiusura del server.
Da qui il server dovrà semplicemente calcolare e inviare dati al client, infatti se nel client viene selezionata la modalità di gioco "Partita classificata" il server inizierà un loop che verrà eseguito 10 volte. In questo loop il server prende tutti gli elementi dei diversi file contenenti le domande e le risposte divisi per argomento, inserendoli in una lista.
Ci sarà poi un'altra lista che conterrà solo le posizioni prestabilite delle domande, quindi il procedimento sarà:

1. viene assegnata ad una variabile una posizione randomica di una domanda presa dalla lista delle posizioni;
2. la domanda scelta viene rimossa dalla lista per evitare ricorrenze;
3. la domanda scelta viene inviata al client;

Per poi andare in un ciclo che verrà ripetuto per 4 volte:

1. viene pre assegnata una variabile con il valore della posizione della domanda scelta randomicamente;
2. nel ciclo questa variabile viene incrementata di 1;
3. si salva in una variabile l'elemento corrispondente alla posizione data dalla precedente variabile nella lista contenente le domande e le risposte;
4. si invia questa variabile al client;

Il file delle domande e delle risposte si presenta in questa maniera:
<Domanda>
<possibile risposta>
<possibile risposta>
<possibile risposta>
<possibile risposta>
<risposta esatta>
Sapendo ciò si saprà sempre che la domanda si trova ogni 6 posizioni nella lista e che la risposta giusta si trova 5 posizioni dopo la domanda.
Infatti per effettuare il controllo della risposta data dall'utente basterà comparare la risposta data con la posizione della domanda all'interno della lista +5.
Dopo di che al server verrà restituito un valore, che rappresenta il punteggio totalizzato dall'utente nel quiz. Questo valore verrà inserito insieme al nickname all'interno di un file chiamato Classifica.
Nel caso in cui al server venisse restituito 2 come valore, partirebbe uno script simile a quello della Partita classificata tranne per alcuni cambiamenti.
Mentre in partita classificata tutti i file comprendenti gli argomenti vengono inseriti in una lista, qui l'utente può scegliere il singolo argomento su cui si può "allenare", per cui in base alla sua scelta il server inserirà in una lista solo gli elementi presenti nel file dell'argomento scelto.
Un altro piccolo cambiamento è che in questa modalità di gioco il punteggio non viene inserito nella classifica essendo che viene considerato allenamento.

Nell’ultimo caso possibile il client restituirà il valore 3 che rappresenta la classifica.
In questa parte di codice viene effettuata una lettura del file score.txt salvando anche qui tutti gli elementi in una lista.
Essendo il file composto in questo modo:
<score>,<nickname>
<score>,<nickname>
<score>,<nickname>
sarà possibile utilizzare una funzione di python che permette di ordinare in modo decrescente tutti gli elementi che verranno poi salvati in una lista.
Dopo di che si svolgerà per tre volte un ciclo che invierà al client gli elementi presenti nella lista.
Essendo che il ciclo viene ripetuto per sole tre volte si potrà avere una top 3 dei migliori giocatori.

CENNI TEORICI

Architettura Client-Server

In informatica il termine sistema client-server indica un'architettura di rete nella quale genericamente un computer client o terminale si connette ad un server per la fruizione di un certo servizio, quale ad esempio la condivisione di una certa risorsa con altri client, appoggiandosi alla sottostante architettura protocollare.

Client

In informatica, il client indica una determinata componente hardware o software che accede alle risorse o ai servizi erogati da un’altra componente detta server

Server

In informatica, il server indica una componente hardware o software che fornisce i dati richiesti da una o più altre componenti dette client. In altre parole, un server non è altro che un computer o un programma in grado di rispondere alle richieste fatte da altri computer o da altri programmi.

Socket

Un socket è un oggetto software che permette l'invio e la ricezione di dati, tra host remoti (tramite una rete) o tra processi locali. Socket locali e remoti in comunicazione formano una coppia (pair), composta da indirizzo e porta di client e server; tra di loro c'è una connessione logica.

Python

Python è un linguaggio di programmazione ad alto livello, supporta diversi paradigmi di programmazione, come quello object-oriented, quello imperativo e quello funzionale, ed offre una tipizzazione dinamica forte.

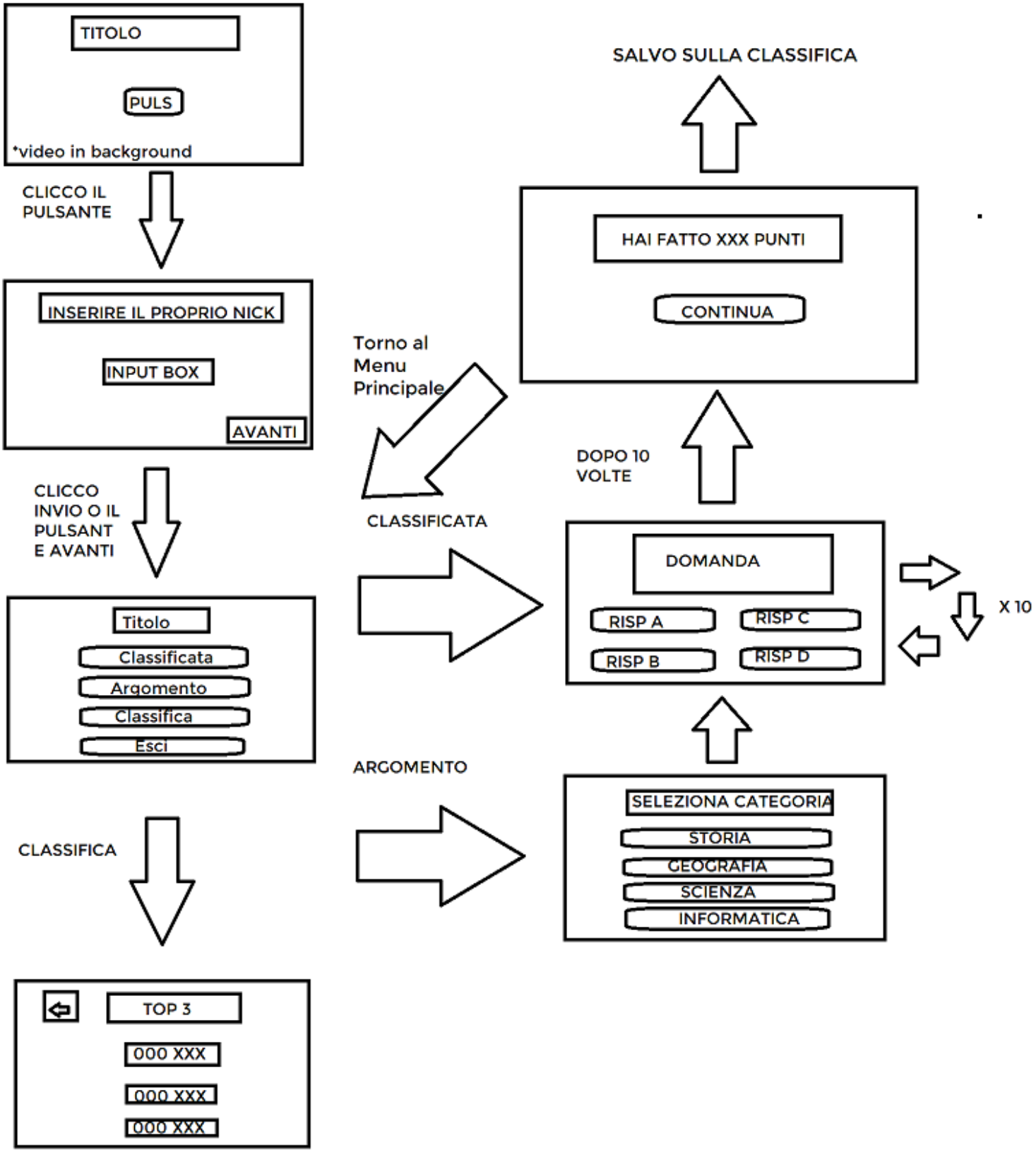
Pygame

Pygame è un insieme di moduli Python progettato per la scrittura di giochi. Offre diverse funzionalitá quali gestione delle finestre e gestione delle interazioni tramite elementi software come rettangoli.

OpenCV

OpenCV è una libreria Open-source per la gestione della Computer vision e del machine learning. È utilizzato per processare immagini e video per identificare volti, oggetti o semplicemente mostrarli a schermo.

PROGETTAZIONE CLIENT GRAFICA



ORGANIGRAMMA

- Enrique Nucci: Sviluppo lato server, documentazione.
- Giulio Raffaeli: Sviluppo lato client, implementazione interfaccia grafica.