PROGETTO INFORMATICA 3M

**Ideato da:**

Nicola Tonelli, Si è occupato dell’interfaccia con l’utente, ha inserito le funzioni e ha creato il file principale.

Diego Amati, Si è occupato del principale lavoro del sistema di combattimento, ha gestito i file esterni e si è occupato dei salvataggi

Paolo D’Elia, Si è occupato dei suoni e dell’analisi e ha anche aiutato con il resto.

Filippo Maria Cornacchia, ha tradotto il programma in inglese

**Idee:**

Nicola Tonelli, ha avuto l’idea dei titoli di coda, della formula per calcolare il danno e della colorazione dei testi.

Diego Amati, ha avuto l’idea di fare il progetto su pokémon, l’inserimento delle immagini (implementato da Nicola Tonelli), la selezione della lingua, l’easter egg e l’inserimento della barra di caricamento.

Paolo D’Elia, ha avuto l’idea di mettere i suoni (implementati da Nicola Tonelli) e l’inserimento della scritta pokémon in Text art.

**INDICE**

* Menù
* Creazione personaggio
* Scheda pokémon del giocatore
* Scheda pokémon avversario
* Sistema di combattimento
* Sistema di cura
* Sistema punti esperienza
* Funzioni principali

**Specificazione schede**:

Scheda pokémon del giocatore:

* Statistiche del pokémon.
* Livello del pokémon che parte dal livello 5.
* Il tipo che varia dal pokémon scelto.

Scheda pokémon avversario:

La scheda del Pokémon avversario è quasi uguale cambia solo il livello che viene scelto casualmente da 1 a 10

**Descrizione del gioco:**

Il tema del progetto è basato sui famosi giochi Pokémon. I primi giochi, Pokémon Rosso e Verde furono rilasciati nel 1996 facendo appassionare tantissime persone. Nel programma si vuole ricreare una porzione di un gioco Pokémon, scegliendo all’inizio il proprio personaggio tra Calem oppure Serena e subito dopo decidere quale pokémon prendere tra i tre starter (quelli di prima generazioni) per incominciare la propria esperienza di gioco. Appena si è scelto il proprio compagno si formerà una scheda con le sue statistiche del tutto casuali e subito dopo ci si potrà cimentare in un uno o più combattimenti contro dei pokémon selvatici.

**Specifiche aggiuntive:**

Abbiamo ricreato un gioco Pokémon ispirandoci a quelli ufficiali prendendo spunto dai diversi giochi della saga. Il nostro programma riproduce qualche meccanica del gioco vero .Alcuni dei pokémon avversari hanno un solo tipo anziché due come nel gioco originale, non abbiamo utilizzato tutti i tipi esistenti e abbiamo usato una nostra formula per far incrementare le statistiche del pokémon ad ogni aumento di livello, anche il sistema di guadagno dell’esperienza non corrisponde all’originale così come il sistema di schivata e di fuga.

**Macro Analisi:**

Il programma è composto da diverse funzioni le più importanti sono la procedura “game\_i” presente nel file pokémon\_ita.py e “game\_e” presente nel file pokémon\_eng.py la prima usata per definire il programma in italiano e la seconda per il programma in inglese dove al loro interno sono presenti le altre funzioni e anche altre procedure che verranno spiegate singolarmente più in basso.

**Tabella delle variabili:**

**Nome tipo utilizzo descrizione**

pokemon\_name testo I/O/L nome del tuo pokémon

enemy\_pokemon testo I/O/L nome pokémon avv.

enemy\_pokemon\_list intero I//L variabile utilizzata per far si che un pokémon venga scelto a random fra quelli della lista.

pokemon\_lvl, enemy\_pokemon\_lvl intero I/O/L livello del pokémon

pokemon\_exp, enemy\_pokemon\_exp intero I/O/L exp del pokémon

battle\_exp intero I/O/L exp ottenuti

enemy\_attack intero I//L attacco del nemico

pokemon\_type, enemy\_pokemon\_type testo I/O/L tipo del pokémon

modifier, enemy\_modifier reale I//L variabile utilizzata per capire se un pokémon è superefficace, poco efficace oppure non ha effetto.

power intero I//L potenza della mossa

dmg reale I/O/L danni inflitti/ricevuti

lp1 reale I/O/L variabile usata per il sistema di cura

miss\_perc intero I//L probabilità che un pokémon possa schivare un attacco

lp, elp intero I/O/L punti vita

att, eatt intero I/O/L punti attacco

dif, edif intero I/O/L punti difesa

satt, esatt intero I/O/L punti attacco speciale

sdif, esdif intero I/O/L punti difesa speciale

spe, espe intero I/O/L punti velocità

inv vettore O/L oggetti dell’inventario

invn lista I/O/L quantità di oggetti nell’inventario

stick lista I/O/L usata per inserire i salvataggi

save\_list lista I/O/L variabili che sono da salvare

tot\_life intero I/O/L

type\_heal testo I/L tipo di cura

heal intero O/L tot punti vita ricevuti grazie al tipo di item

saves testo I/O/L

player\_name testo O/L

**Schema:**

game\_i

Main

game\_e

**ANALISI DI TUTTE LE FUNZIONI/PROCEDURE PIU IMPORTANTI UNA ALLA VOLTA**

Analisi di save: questa procedura viene utilizzata per il salvataggio dei dati

|  |  |  |  |
| --- | --- | --- | --- |
| player\_name | testo | O/L |  |
| character | testo | O/L |  |
| pokemon | testo | O/L |  |
| pokemon\_exp | intero | O/L |  |
| pokemon\_lvl | intero | O/L |  |
| pokemon\_type | testo | O/L |  |
| lp | intero | O/L |  |
| att | intero | O/L |  |
| dif | intero | O/L |  |
| satt | intero | O/L |  |
| sdif | intero | O/L |  |
| spe | intero | O/L |  |
| save\_list | lista | I/O/L |  |
| f | testo | L | Variabile di supporto |

Schema:

game\_i save

**Pseudo della procedura save**:

Scrivi: "Salvataggio in corso...";

time.sleep(2)

save\_list = [player\_name, character, pokemon, str(pokemon\_exp), str(pokemon\_lvl), pokemon\_type, str(lp), str(att), str(dif), str(satt), str(sdif), str(spe)]

f = open("salvataggi.txt", "w")

per i fino a (lunghezza(save\_list)):

f.write(save\_list[i] + "\n")

f.close()

Scrivi: "Salvataggio riuscito.";

**ANALISI DI TUTTE LE FUNZIONI/PROCEDURE PIU IMPORTANTI UNA ALLA VOLTA**

Analisi di cura: questa procedura serve per gestire la salute del tuo pokemon nel caso tu voglia curarlo durante il combattimento utilizzando un oggetto.

|  |  |  |  |
| --- | --- | --- | --- |
| tot\_life | intero | I/O/L |  |
| type\_heal | testo | I/L |  |
| invn | lista | I/O/L |  |
| heal | intero | O/L |  |

Schema:

game\_i cura

**Pseudo della procedura cura**:

Scrivi: "Cosa vuoi usare tra Pozione, Super Pozione, Iper Pozione o Pozione Max.";

type\_heal=input()

global heal

se type\_heal == "pozione" or type\_heal=="Pozione" or type\_heal=="POZIONE": allora

heal = 30

invn[0]=invn[0]-1

playsound("heal\_sound.mp3")

Scrivi: "Il tuo pokémon ha recuperato 30 di vita.";

Fine allora

Altrimenti se type\_heal == "super pozione" or type\_heal=="Super Pozione" or type\_heal=="SUPER POZIONE": allora

heal = 60

invn[1] invn[1]-1

playsound("heal\_sound.mp3")

Scrivi: "Il tuo pokémon ha recuperato 60 di vita.";

fine altrimenti se

altrimenti se type\_heal == "iper pozione" or type\_heal=="Iper Pozione" or type\_heal=="IPER POZIONE": allora

heal = 120

invn[2] invn[2]-1

playsound("heal\_sound.mp3")

Scrivi: "Il tuo pokémon ha recuperato 120 di vita.";

fine altrimenti se

altrimenti se type\_heal == "pozione max" or type\_heal=="Pozione Max" or type\_heal=="POZIONE MAX": allora

heal = tot\_life

invn[3] invn[3]-1

playsound("heal\_sound.mp3")

Scrivi: "Il tuo pokémon è tornato in piene forze.";

fine altrimenti se

**ANALISI DI TUTTE LE FUNZIONI/PROCEDURE PIU IMPORTANTI UNA ALLA VOLTA**

Analisi di damage: questa procedura gestisce il danno in combattimento utilizzando una formula.

|  |  |  |  |
| --- | --- | --- | --- |
| pokemon\_lvl | intero | I/O/L |  |
| power | intero | I/L |  |
| modifier | reale | I/L |  |
| dmg | reale | I/O/L |  |

Schema:

game\_i damage

**Pseudo della procedura damage**:

global dmg

dmg ((((((2\*pokemon\_lvl)/5)+2)\*power\*stat\_1/stat\_2)/50)+2)\*modifier

**ANALISI DI TUTTE LE FUNZIONI/PROCEDURE PIU IMPORTANTI UNA ALLA VOLTA**

Analisi di lvl\_up: questa funzione viene usata solo nel caso di un aumento di livello da parte del pokemon.

|  |  |  |  |
| --- | --- | --- | --- |
| pokemon\_lvl | intero | O/L |  |
| pokemon\_exp | intero | O/L |  |
| lp | intero | O/L |  |
| att | intero | O/L |  |
| dif | intero | O/L |  |
| satt | intero | O/L |  |
| sdif | intero | O/L |  |
| spe | intero | O/L |  |
| pokemon | testo | O/L |  |

Schema:

game\_i lvl\_up

**Pseudo della funzione lvl\_up:**

Se pokemon\_exp >= 100: allora

pokemon\_exp = 0

pokemon\_lvl pokemon\_lvl+1

playsound("lvl\_sound.mp3")

Scrivi: pokemon, “è salito al livello”, pokemon\_lvl “!")

lp lp+(lp/100\*7)

att att+(att/100\*7)

dif dif+(dif/100\*7)

satt satt+(satt/100\*7)

sdif sdif+(sdif/100\*7)

spe spe+(spe/100\*7)

ritorna pokemon\_lvl, pokemon\_exp

fine allora

**Pseudo del Main:**

Finchè language!="ITA" or language!="Ita" or language!="ita" or language!="italiano" or language!="i" or language!="ITALIANO" or language!="Italiano" or language!="I" or language!="ENG" or language!="Eng" or language!="eng" or language!="english" or language!="e" or language!="ENGLISH" or language!="English" or language!="E":

se language == "ITA" or language == "Ita" or language == "ita" or language == "italiano" or language == "i" or language == "ITALIANO" or language == "Italiano" or language == "I": allora

Pokémon\_ita.game\_i()

break

Fine allora

Altrimenti se language == "ENG" or language == "Eng" or language == "eng" or language == "english" or language == "e" or language == "ENGLISH" or language == "English" or language == "E": allora

Pokémon\_eng.game\_e()

break

Fine altrimenti se

altrimenti:

Scrivi: "Invalid input./Input invalido.";

fine altrimenti

**Fonti:**

Come fonte principale abbiamo usato <https://wiki.pokemoncentral.it>

Abbiamo usato anche diverse librerie come:

* Climage, utilizzato per stampare a schermo delle immagini;
* Random,
* Time, utilizzata per mettere il programma in pausa per tot secondi;
* Colored, utilizzando le funzioni “**fg”** (foreground, per il colore del testo) “**bg**” (background, per colorare lo sfondo e “**attr**” (per interrompere i comandi precedenti);
* Playsound, usata per inserire file audio nel programma;
* Pygame, usata per la colonna sonora con la funzione “mixer” che serve per mettere i file audio in background;
* Pyfiglet, usata per trasformare una frase in Text art