

MODULO GESTIRE_MENU- ALGORITMI

Moduli inclusi:

GESTIRE_FILE

GESTIRE_CLASSIFICA

GESTIRE_PARTITE

GESTIRE_PERCORSO

GESTIRE_MENU

Costanti:

PROMPT_ERRORE= "Nessuna Partita in corso"

SCELTA_ESCI=0

SCELTA_PARTITA_C=1

SCELTA_PARTITA_P=2

SCELTA_SALVA=1

SCELTA_S_E=2

SCELTA_ABBANDONA=3

SCELTA_CONTINUA=4

Algoritmi:

Funzione `gestire_menu_partita`:

INPUT:

`file_menu`, file contenente le opzioni del menu, file di testo

OUTPUT:

flag, indicatore di completamento della funzione, intero

LAVORO:

scelta, scelta effettuata dall'utente riguardo alla funzionalità, intero $\geq 0 \leq 4$

partita, partita configurata dall'utente, Record competizione_oca

vincitore, giocatore vincitore della partita, Record giocatore

INIZIO

```
flag = 1
```

```
ESEGUI
```

```
scelta = scegliere_opzione_di_menu(file_menu, SCELTA_ESCI,  
SCELTA_PARTITA_P)
```

```
SE (scelta = SCELTA_PARTITA_C)
```

```
partita = configurare_partita(partita)
```

```
vincitore = gestire_turno(partita, vincitore)
```

```
flag = aggiornare_classifica(vincitore)
```

```
flag = 0
```

```
ALTRIMENTI SE (scelta = SCELTA_PARTITA_P)
```

```
partita = configurare_partita(partita)
```

```
vincitore = gestire_turno(partita, vincitore)
```

```
flag = aggiornare_classifica(vincitore)
```

```
ALTRIMENTI SE (scelta = SCELTA_ESCI)
```

```
flag = 0
```

```
fine SE
```

```
FINCHÈ ((flag = 1) && (flag != ERRORE_FILE))
```

FINE

Funzione `gestire_partita_in_corso`:

INPUT:

`partita`, `partita` attualmente in corso, Record
`competizione_oca`

OUTPUT:

`fine`, variabile che assume valore 1 qualora si decida di abbandonare la partita in corso, intero = 1

LAVORO:

`scelta_menu_g`, scelta effettuata dall'utente, intero ≥ 0
 ≤ 3

`file_menu_carica`, file che contiene il menu, file di testo
`file_partita_in_corso`, file contenente il record della partita attuale, file di testo

`flag`, variabile che indica l'uscita dalla pausa o meno, Intero

`giocatore_turno`, giocatore che sta disputando il turno della partita, Record giocatore

`tabellone_percorso`, tabellone della partita in corso, Record tabellone

`turno`, valore che indica il turno del giocatore, Intero
 $\text{MIN_GIOCATORI} \leq \text{turno} \leq \text{MAX_GIOCATORI}$

`pos_turno`, posizione del giocatore in un determinato turno, Intero ≥ 0

INIZIO

`fine` = 0

```

file_partita_in_corso=ricercare_il_menu(PARTITA_IN_CORSO,
LETTURA_TESTO)

SE(file_partita_in_corso = 0)

fine = ERRORE_FILE

ALTRIMENTI

flag = 1

tabellone_percorso = leggere_tabellone_percorso (partita,
tabellone_percorso)

turno = leggere_turno (partita)

giocatore_turno=leggere_giocatori(partita,turno,giocatore
_turno)

pos_turno = leggere_posizione (giocatore_turno)

ESEGUI

ESEGUI

visualizzare_partita(partita, pos_turno)

stampare_file_di_testo(file_partita_in_corso)

scelta_menu_g = leggere_da_tastiera()

SE(scelta_menu_g = SCELTA_SALVA)

file_menu_carica=ricercare_il_menu(CARICA,LETTURA_TESTO)

stampare_file_di_testo(file_menu_carica)

fine=salvare_partita(partita)

FINCHÈ((scelta_menu_g!=SCELTA_CONTINUA)AND (scelta_menu_g
!= SCELTA_SALVA) AND (scelta_menu_g != SCELTA_S_E) AND
(scelta_menu_g != SCELTA_ABBANDONA))

ESEGUI

SE(scelta_menu_g = SCELTA_ABBANDONA)

fine=1

flag=0

ALTRIMENTI SE (scelta_menu_g = SCELTA_S_E)

```

```

file_menu_carica=ricercare_il_menu(CARICA,LETTURA_TESTO)
stampare_file_di_testo(file_menu_carica)
fine=salvare_partita(partita)
fine=1
flag=0
ALTRIMENTI
flag=0
fine SE
FINCHÈ ((flag=1) AND (fine !=ERRORE_FILE))
FINCHÈ((scelta_menu_g = SCELTA_SALVA) OR (scelta_menu_g =
SCELTA_S_E) OR (scelta_menu_g = SCELTA_ABBANDONA) AND
(fine != 1) AND (fine != ERRORE_FILE)
FINE

```

Funzione salvare_partita:

INPUT:

partita_salvata, partita che si intende salvare nello slot
scelto, Record competizione_oca

OUTPUT:

fine, variabile che assume valore 1 qualora si decida di
abbandonare la partita in corso, intero = 1

LAVORO

slot, indica lo slot in cui salvare la partita attualmente
in corso, Intero, >0 <= MAX_SLOT

file_salvataggio, file binario contenente un singolo
record partita, file binario

INIZIO

```
PROMPT_SLOT = ""
fine=0
ESEGUI
stampare_a_video(PROMPT_SLOT)
slot = leggere_tastiera_int_verificato()
FINCHÈ ((slot > MAX_SLOT) OR (slot <= 0))
file_salvataggio=ricercare_slots(slot,SCRITTURA_BINARIA)
SE(file_salvataggio= 0)
fine=ERRORE_FILE
ALTRIMENTI
file_salvataggio=scrivere_file(file_salvataggio,partita_s
alvata)
fine SE
FINE
```

Funzione gestire_menu_aiuto:

INPUT:

menu_aiuto, file contenente la scelta tra le regole e il manuale di gioco, file di testo

OUTPUT:

flag, flag che indica alla funzione chiamante di ritornare al menu principale, intero

LAVORO:

scelta, scelta effettuata dall'utente riguardo al menu, intero >=0 <=4

file_regole, file contenente le regole del gioco, file di testo

file_manuale, file contenente il manuale del gioco, file di testo

INIZIO

SCELTA_REGOLE = 1

SCELTA_MANUALE = 2

flag = 1

ESEGUI

scelta=scegliere_opzione_di_menu(menu_aiuto,SCELTA_ESCI,
SCELTA_MANUALE)

SE (scelta = SCELTA_REGOLE)

file_regole = ricercare_il_menu(REGOLE, LETTURA_TESTO)

SE (file_regole = 0)

flag = ERRORE_FILE

ALTRIMENTI

flag = gestire_sottomenu_aiuto(file_regole)

flag = 1

fine SE

ALTRIMENTI SE (scelta = SCELTA_MANUALE)

file_manuale=ricercare_il_menu(MANUALE, LETTURA_TESTO)

SE (file_manuale = 0)

flag = ERRORE_FILE

ALTRIMENTI

flag = gestire_sottomenu_aiuto(file_manuale)

flag = 1

fine SE

ALTRIMENTI

flag = 0

```
fine SE  
FINCHÈ ((flag = 1) AND (flag != ERRORE_FILE))  
FINE
```

Funzione gestire_sottomenu_aiuto:

INPUT:

sottomenu, sottomenu da stampare, file di testo

OUTPUT:

flag, indica la volontà di ritornare indietro, intero

INIZIO

flag=1

ESEGUI

stampare_file_di_testo(sottomenu)

stampare_a_video(SOTTOMENU)

flag = leggere_tastiera_int_verificato()

FINCHÈ (flag != SCELTA_ESCI)

FINE