

In questo articolo esamineremo un particolare array numerico, cioè quello del punteggio, per vedere come viene gestito, inoltre vedremo come far terminare una partita.

Sarà un articolo il cui argomento proseguirà nella prossima lezione in modo da proporre tutto (o quasi) ciò che è relativo al punteggio.

Cominciamo con questo codice sorgente:

```
! Costantie array da indicare SEMPRE
```

```
! con questi PRECISI NOMI
```

```
! (notare che sono dichiarati prima dell' Include "Parser"; !!!
```

```
Constant TASKS_PROVIDED;
```

```
Constant NUMBER_TASKS = 2;
```

```
Constant MAX_SCORE = 3;
```

```
Array task_scores -> 1 2;
```

```
Include "Parser";
```

```
Include "Verblib";
```

```
Include "Replace";
```

```
Object stanza "stanza"
```

```
with description "Ti trovi in una stanza.",
```

```
has light;
```

```
Object -> cranio "teschio"
```

```
with name 'teschio' 'cranio',
```

```
initial "Puoi vedere un cranio. Se lo raccogli guadagni un punto.",
```

```
description "Un cranio antico.",
```

```
after [;
```

```
Take: Achieved(0);
```

```
];
```

```
Object -> candela "candela"
```

```
with name 'candela',
```

```
initial "Puoi vedere una candela. Se la raccogli guadagni due punti.",
```

```
description "Puoi vedere una candela. Se la raccogli guadagni due punti.",
```

```
after[;
Take: Achieved(1);
],
has female;

Object -> serpente "serpente"
with name 'serpente',
initial "Puoi vedere un serpente. Se lo tocchi perdi la partita.",
description "Un serpente velenoso e aggressivo.",
before [;
Touch: deadflag=1; print "Prendi il serpente, che innervosito ti morde e ti fa morire avvelenato.^";
rtrue;
];
```

```
Object -> fiore "fiore"
with name 'fiore',
initial "Puoi vedere un fiore. Se lo annusi vinci la partita.",
description "Un fiore viola.",
before [;
Smell: deadflag=2; print "Annusi il fiore e vinci la partita!^"; rtrue;
];
```

```
[ Initialise;
location = stanza;
];
```

```
Include "ItalianG";
```

Come abbiamo già detto nell'articolo precedente, l'array che gestisce il punteggio è nominato (**necessariamente**) `task_scores`, e vediamo che lo abbiamo inizializzato coi valori 1 e 2. **Sempre** associato a questo array ci sono alcune costanti:

Constant TASKS_PROVIDED: permette di gestire autonomamente e adeguatamente i punteggi

Constant NUMBER_TASKS: indica quanti elementi sono presenti nell'array `task_scores`

Constant MAX_SCORE: indica il punteggio massimo totalizzabile, cioè il suo valore corrisponde alla somma dei valori indicati nell'array `task_scores`

Da notare che questo Array e le relative costanti sono indicati prima della direttiva *Include "Replace"*;

Spostiamoci all'oggetto cranio. Per modificare i verbi vengono utilizzate le proprietà **after** e **before**: la loro trattazione è lunga, non standardizzata e non inerente a questo argomento, quindi per il momento quando dovete assegnare un punteggio utilizzate la *after* e quando dovete far concludere la partita utilizzate la *before*.

Nell'esempio abbiamo modificato il verbo *Prendi* (indicato dall'inglese *Take*, che è tra quelli preimpostati) con l'istruzione **Achieved()**. All'interno delle parentesi bisogna indicare l'indice della cella dell'array *task_scores* che corrisponde al punteggio associato. Per il cranio il valore è zero perché in un array i valori degli indici partono da zero, identificando così il primo valore, cioè uno. Con la routine *Achieved()* la **prima volta** che si compie l'azione si conquista anche il punteggio corrispondente. Le costanti che abbiamo definito, infatti, evitano che il punteggio venga riassegnato se l'azione viene compiuta una seconda volta.

Lo stesso identico discorso vale per la candela.

Guardiamo invece l'oggetto serpente. Qui, anziché il verbo *Prendi* abbiamo modificato il verbo **Tocca** (che corrisponde all'inglese preimpostato **Touch**) [attenzione: chiaramente se si prende il serpente lo si tocca, così come se lo si spinge, tira... quindi sarebbe opportuno modificare tutti questi verbi (preimpostati) per dare un senso al comando!].

In questo caso non viene modificato il punteggio, ma compare la variabile **deadflag**. Questa variabile di default può assumere i valori 0, 1 e 2. Il suo valore di default è 0 e non produce effetti. Quando modifichiamo il suo valore in 1, come nel caso del serpente, comparirà un messaggio di morte. Quando invece il suo valore viene modificato in 2, come nel caso del fiore, comparirà un messaggio di vittoria.

Notiamo che la vittoria non corrisponde necessariamente all'aver conquistato il punteggio massimo (ad esempio si potrebbe annusare il fiore senza prendere il teschio o la candela o entrambi).

Quando, durante il gioco, si digita *> punti* o *> punteggio* vengono indicati i punti totalizzati rispetto al punteggio massimo. Si può introdurre una migliona nel codice sorgente permettendo di indicare le azioni compiute che ci hanno fatto aumentare il punteggio.

Per fare ciò bisogna **prima di Include "Parser"**; aggiungere **Replace PrintTaskNameSub**; che ci permette di modificare la routine associata alla stampa del punteggio.

Questa routine si indica (generalmente dopo) la *Initialise* ed ha la seguente forma:

```
[ PrintTaskName achievement;
```

```
    switch(achievement)
```

```
    { 0: "hai preso il teschio";
```

```
      1: "hai preso la candela";
```

```
    };
```

```
];
```

Vediamo che come le routines dell'articolo precedente viene aperta la quadra, indicato il nome (*PrintTaskName*) e indicata anche una variabile (*achievement*). Per ora lasciamo da parte una trattazione approfondita dell'istruzione **switch** e vediamo solo come agisce in questo caso particolare: se l'azione che corrisponde al punteggio della prima casella dell'array *task_scores* (di indice zero) è stata compiuta, viene stampato il relativo messaggio, così per le altre azioni. Ossia: i numeri prima dei

due punti corrispondono agli indici dell'array utilizzato per il punteggio, e di conseguenza alle rispettive azioni.

Allo stesso modo, per dare un tocco di colore all'avventura, è possibile modificare il messaggio di morte preimpostato (che generalmente è ***** SEI MORTO *****). Per fare ciò bisogna creare la routine **DeathMessage**, anche questa (generalmente) dopo la *Initialise*, ed ha questo aspetto:

```
[ DeathMessage;  
if (deadflag == 3) print "muori avvelenato";  
];
```

In questo caso abbiamo voluto modificare il messaggio di morte provocato dal toccare il serpente. Vediamo che compare ancora la variabile *deadflag*, ma col valore 3. Questo perché per ottenere l'effetto non dobbiamo più utilizzare il valore 1 nella modifica del verbo *Touch* (che darebbe la risposta di default), ma far assumere un nuovo valore a questa variabile (diverso anche da 0 e 2). Quindi, immaginando, nell'oggetto serpente, di aver scritto

```
Object->serpente "serpente"  
with name 'serpente',  
initial "Puoi vedere un serpente. Se lo tocchi perdi la partita.",  
description "Un serpente velenoso e aggressivo.",  
before [;  
Touch: deadflag=3; print "Prendi il serpente, che innervosito ti morde e ti fa morire avvelenato.^";  
rtrue;  
];
```

una volta toccatolo otterremo il nostro nuovo messaggio di morte.

[Da notare la condizione all'interno dell' *if*: quando si vuole verificare che una variabile abbia un determinato valore (o che due variabili abbiano lo stesso valore) va indicato con l'operatore **==** anziché col solo uguale!)]

Quindi, per ricapitolare, il nostro codice sorgente, con gli "abbellimenti" apportati sarebbe il seguente:

```
Constant TASKS_PROVIDED;  
Constant NUMBER_TASKS = 2;  
Constant MAX_SCORE = 3;  
Array task_scores -> 1 2;  
  
Replace PrintTaskNameSub;  
  
Include "Parser";
```

Include "Verblib";

Include "Replace";

Object stanza "stanza"

with description "Ti trovi in una stanza.",

has light;

Object -> cranio "teschio"

with name 'teschio' 'cranio',

initial "Puoi vedere un cranio. Se lo raccogli guadagni un punto.",

description "Un cranio antico.",

after [;

Take: Achieved(0);

];

Object -> candela "candela"

with name 'candela',

initial "Puoi vedere una candela. Se la raccogli guadagni due punti.",

description "Puoi vedere una candela. Se la raccogli guadagni due punti.",

after [;

Take: Achieved(1);

],

has female;

Object -> serpente "serpente"

with name 'serpente',

initial "Puoi vedere un serpente. Se lo tocchi perdi la partita.",

description "Un serpente velenoso e aggressivo.",

before [;

*Touch: deadflag=3; print "Prendi il serpente, che innervosito ti morde e ti fa morire avvelenato.^";
rtrue;*

```
];
```

```
Object->fiore "fiore"
```

```
with name 'fiore',
```

```
initial "Puoi vedere un fiore. Se lo annusi vinci la partita.",
```

```
description "Un fiore viola.",
```

```
before [;
```

```
Smell: deadflag=2; print "Annusi il fiore e vinci la partita!^"; rtrue;
```

```
];
```

```
[ Initialise;
```

```
location = stanza;
```

```
];
```

```
[ PrintTaskName achievement;
```

```
switch(achievement)
```

```
{ 0: "hai preso il teschio";
```

```
1: "hai preso la candela";
```

```
};
```

```
];
```

```
[ DeathMessage;
```

```
if (deadflag == 3) print "muori avvelenato";
```

```
];
```

```
Include "ItalianG";
```

Anche per questa volta è tutto. La prossima volta per finire il capitolo sugli array numerici (in cui è confluito l'argomento del punteggio) vedremo come modificare negativamente i punti totalizzati e come stampare un "rango" associato alla bravura del giocatore.

Gli esercizi, quindi, al prossimo arti

