Vediamo ora come si fa a connettere le stanze e introduciamo anche l'istruzione **if else**

```
Include "Parser";
Include "Verblib";
Include "Replace";

Object stanza_1 "Una stanza"
with description "Ti trovi in una stanza.^Puoi andare ad est e a sud.",
e_to stanza_2,
s_to [;
if (teschio in player) { PlayerTo(stanza_3); rtrue; }
else "Devi possedere il teschio per andare in quella stanza.";
],
has light;

Object -> teschio "teschio"
with name 'teschio';

Object stanza_2 "La stanza ad est"
with description "Ti trovi nella stanza ad est.^Puoi ritornare ad ovest.",
w_to stanza_1,
cant_go "Cerca un'altra strada.",
has light;

Object stanza_3 "La stanza a sud"
with description "Ti trovi nella stanza a sud.^Puoi ritornare a nord.",
n_to stanza_1,
has light;

[ Initialise;
location = stanza_1;
player.description = "Sei il tipico avventuriero.";
];

Include "ItalianG";
```

Vediamo che abbiamo creato tre stanze col solito metodo visto negli esempi passati. Per connetterle tra di loro basta utilizzare le istruzioni che richiamano le direzioni:

n, s, e, w, ne, se, nw, sw

a cui bisogna aggiungere *_to* seguito dal nome della stanza che vogliamo collegare. Le istruzioni per muoversi tra le stanze vanno indicate sempre prima degli attributi. Vediamo che la dicitura è anglofona, essendo l'ovest identificato dalla lettera w ("west"). Esistono in realtà altre due direzioni: *u_to* e *d_to* rispettivamente per andare in alto e in basso.

Notiamo anche che, (così come per le istruzioni *initial* e *description*, anche se fino ad ora non lo abbiamo mai visto) è possibile complicare un po' il codice sorgente: abbiamo deciso di non permettere al giocatore di andare a sud se non possiede il teschio: dopo *s_to*, come fosse una routine, si apre una parentesi quadra seguita da un punto e virgola (vedremo come si crea una routine prossimamente), si scrivono le istruzioni, poi si richiude e, come al solito, si aggiunge una virgola.

[Nota di game design: nel gioco bisognerebbe giustificare il perché si può proseguire solo se si possiede il teschio!]

Naturalmente, però, abbiamo bisogno di verificare che il giocatore abbia o meno il teschio. L'istruzione **if else** è il **cuore delle avventure testuali** ed è proprio quella che ci serve. La sua sintassi è

if (*condizione*) *prima alternativa*;
else *seconda alternativa*

questo nel caso in cui la prima alternativa sia costituita da una sola istruzione: nel nostro esempio ce ne sono due, e quindi abbiamo dovuto includerle tra graffe per farle "considerare come una".
[Le graffe si ottengono tenendo premuto il tasto Alt e premendo in sequenza i tasti 1 2 3, oppure 1 2 5]
Attenzione: in questo caso la sintassi cambia leggermente:

if (condizione) {
prima_istruzione_prima_alternativa;
seconda_istruzione_prima_alternativa;
}
else *seconda_alternativa*;

[cioè non è più presente il punto e virgola prima dell' *else*]

Allo stesso modo se invece avessimo dovuto inserire più istruzioni per quanto riguarda la seconda alternativa

if (condizione) prima_alternativa;
else {
prima_istruzione_seconda_alternativa;
seconda_istruzione_seconda_alternativa:
}; ! da notare quest'ultimo punto e virgola

[non è necessario andare a capo ad ogni istruzione, ma è consuetudine farlo per rendere più leggibile e comprensibile il listato]

In questo caso dovevamo testare se un oggetto apparteneva al giocatore. Abbiamo detto che anche i personaggi vengono identificati come oggetti, e lo stesso accade per il giocatore, quindi la domanda corrispondente è: il teschio è nel giocatore? che, tradotto praticamente alla lettera, diventa proprio *teschio in player* (*player* è il nome predefinito per identificare l'oggetto giocatore).

[avremmo potuto anche pensare alla condizione opposta, ovvero non permettere al giocatore di andare a sud qualora possedesse il teschio, in questo caso *in* va sostituito con *notin*]

Oltre all'istruzione *if else* è possibile utilizzare anche la sola **if**, nel caso si voglia semplicemente "aggiungere" qualcosa e non "proporre un'alternativa" (vedi esercizio alla fine).

Vediamo anche che se "complichiamo" l'istruzione per muoversi tra le stanze, non è più sufficiente indicare la stanza di arrivo, ma abbiamo bisogno di una istruzione apposta: **PlayerTo()** seguita, come al solito, da *rtrue*, altrimenti anche in questo caso dopo esserci recati nella nuova locazione ci viene

fornito anche il messaggio standard di una locazione irraggiungibile (qualcosa come "Non è possibile andare da quella parte.").

Inform 6 mette a disposizione anche la possibilità di modificare questo messaggio: nella stanza ad est vediamo infatti l'istruzione *cant_go*: essa viene richiamata quando il giocatore (che si trova nella stanza in cui è descritta) digita una direzione in cui non può andare, permettendo di dare una risposta personalizzata anziché quella di default. Non è forse un metodo molto più comodo che scrivere per ogni stanza una risposta per tutte le direzioni in cui (si può e) non si può andare?
Esiste inoltre un modo per applicare questa personalizzazione una volta per tutte, piuttosto che definire una *cant_go* per ogni stanza: lo vedremo più avanti.

Saltiamo poi alla routine *Initialise* e vediamo una nuova istruzione. Ebbene sì: come è possibile descrivere le locazioni tramite l'istruzione *description*, è possibile dare anche una descrizione del giocatore (giocando si ottiene digitando *> esamina me stesso*). In questo caso bisogna far precedere la parola *description* da *player.* (*player.* indica al programma a chi appartiene la descrizione - questo è un meccanismo utilizzatissimo per le variabili, che vedremo prossimamente). Notiamo anche che in questo caso dopo *description* è presente il segno uguale, a differenza di quanto accade nelle stanze.
[Ciò è necessario perché questa descrizione non è una proprietà ma una variabile vera e propria (che contiene al suo interno non un valore ma una sequenza di caratteri)]

[Un'ultima nota per chi ha già sbirciato il manuale, o comunque una nota da tenere buona in futuro: all'interno dell'istruzione *if* è possibile valutare anche il valore delle variabili.
Immaginiamo che da qualche parte ci sia una variabile (chiamiamola: variabile_test) che viene incrementata di uno in seguito a qualche evento e che qualcosa accada non appena questa variabile raggiunge il valore 3, allora in questo caso bisogna scrivere

if (variabile_test == 3) …

e non

if (variabile_test = 3) …

dal momento che il semplice segno uguale serve per assegnare un valore a una variabile (come per la *player.description* - anche se abbiamo detto che a venire assegnato non è un valore ma una stringa di caratteri)]

Per oggi (e per un po' di tempo) è tutto: c'è comunque molta carne al fuoco su cui meditare e materiale sufficiente per creare già qualcosa di leggermente elaborato.
Come usuale, vi lascio uno screenshot: cercate di compilare un codice sorgente che permetta di ottenere le stesse risposte

Una stanza
Ti trovi in una stanza. Puoi andare ad est.

Puoi vedere un teschio qui.

>e
Dovresti prima prendere il teschio.

>prendi teschio
Preso.

>guarda

Una stanza
Ti trovi in una stanza. Una volta qui c'era anche un teschio. Puoi andare ad est.

>esamina teschio
Chissà di chi sarà stato?

>nord
Sei sordo?
Ti ho detto che puoi andare solo ad est!

>e

La stanza ad est
Ti trovi nella stanza ad est.
Puoi ritornare ad ovest.

>esamina teschio
Il teschio emana una strana aura!

>