

Indice

Finalità.....	2
Da dove cominciare.....	2
Dove andare a parare.....	3
Uscire dalla logica dei bivi.....	3
Aggiungere oggetti.....	5
Oggetti nella versione base.....	5
Il loop da un punto di vista del codice.....	6
Il loop dal punto di vista del funzionamento.....	7
Considerazioni.....	10
Oggetti in una versione avanzata.....	11
Esempi.....	13
Aggiungere personaggi.....	15
Personaggi nella versione base.....	16
Esempio di funzionamento.....	18
Complicazioni.....	22
Gestire il limite.....	22
Le Serie paiono inutili.....	22
Le Serie si possono esaurire in ordine sparso.....	23
ID delle frasi.....	24
Personaggi in una versione avanzata.....	24
Interazioni fra PNG e giocatore.....	25
Modifiche necessarie:.....	25
Interazioni fra PNG e gli oggetti.....	27
Il PNG dà qualcosa al giocatore.....	27
Il PNG rivela al giocatore qualcosa che gli rende possibile trovare un oggetto.....	28
Interazioni fra i PNG e i luoghi.....	29
Ancora e ancora e ancora.....	30
Che cosa manca davvero.....	31
Conclusioni.....	31
Niente matematica.....	32
La fantasia è il limite.....	32

Finalità

L'idea è realizzare un gioco, prendendo spunto dall'ormai storico "bivi", che si appoggi per intero a un database.

Intendo sostenere che questo tipo di approccio rende pressoché liberi dalla scelta del linguaggio in cui realizzarlo. Infatti, secondo me, stampate su carta le tabelle di una partita di prova molto, molto breve, sarebbe possibile giocarle 'a mano', ossia senza computer, ma con un direttore di gioco (un 'master') che ci informi sui risultati delle nostre scelte.

Da dove cominciare

Lo schema di partenza è molto semplice.

Supponiamo di avere un database di un'unica tabella così concepita:

luoghi			
IDLuogo	Descrizione	Scelta_A	Scelta_B
1	Ti trovi fuori da un castello. Davanti a te c'è un fossato e alla tua destra un ponte levatoio. Cosa fai? a) Ti tuffi nel fossato b) Percorri il ponte levatoio?	2	3
2	Il fossato era infestato dai coccodrilli. Riprova, troverai un modo più fortunato di morire! Premi (a) o (b) per ricominciare.	1	1
3	Percorri il ponte levatoio e ti trovi nell'atrio del castello. C'è una porta alla tua sinistra e una alla tua destra. Cosa fai? a) Entri nella stanza a sinistra b) Provi la porta di destra	4	5
4	Sei nella sala del tesoro. In mezzo alla stanza c'è uno scrigno chiuso. Cosa fai? a) Apri lo scrigno b) Torni indietro	6	7
5	Sei nella stanza delle armature. Su una parete c'è una spada con su un cartello che recita: "Spada Invincibile". Cosa fai? a) Torni indietro b) Prendi la Spada Invincibile	8	9
...

Credo che l'idea si spieghi da sola:

- il gioco inizia leggendo la prima riga della tabella e mandandola a schermo;
- rimane in attesa della pressione dei tasti “a” o “b” (nella versione in PHP ci sarà l'interfaccia grafica HTML con dei pulsanti);
- se l'utente preme “a” il gioco legge la riga corrispondente al numero nel campo Scelta_A; se l'utente preme “b” il gioco legge la riga di numero Scelta_B;
- il loop riparte.

In questa versione non è nemmeno importante garantire l'uscita dal programma.

Per quanto la logica sia striminzita, già questo abbozzo di programma si rivolge a bambini di una certa età e direi che hanno anche frequentato più volte il CD.

Dove andare a parare

La versione iniziale del gioco (v. “Da dove cominciare”) potrebbe, una volta realizzata, essere ampliata con un'ampia gamma di ulteriori possibilità. Il progetto, se il ragazzino/la ragazzina continuerà a lavorarci su, potenzialmente da fare per anni.

Vorrei qui presentare un esempio di versione “superpompata” del gioco, tale, mi auguro, da soddisfare anche gli appetiti più incontentabili – che questa versione veda mai la luce per opera degli utenti del CD è inverosimile, ma può fungere da ispirazione a chi si domanda “e ora cosa ci possiamo aggiungere?”

Uscire dalla logica dei bivi

Le seguenti due tabelle vanno prese in considerazione insieme, perché interagiscono l'una con l'altra.

luoghi		
IDLuogo	Nome	Descrizione
1	Inizio	Ti trovi fuori da un castello. Davanti a te c'è un fossato e alla tua destra un ponte levatoio.
2	Fossato	Il fossato era infestato dai coccodrilli. Riprova, troverai un modo più fortunato di morire!
3	Atrio	Percorri il ponte levatoio e ti trovi nell'atrio del castello. C'è una porta

luoghi		
IDLuogo	Nome	Descrizione
		alla tua sinistra e una alla tua destra; davanti a te un'ampia scalinata sembra portare al piano di sopra. Puoi:
4	Sala del Trono	Sei nella sala del tesoro. In mezzo alla stanza c'è uno scrigno chiuso. Puoi:
5	Stanza armature	Sei nella stanza delle armature. Su una parete c'è una spada con su un cartello che recita: "Spada Invincibile". Puoi:
...

entra_esci		
Provenienza	Destinazione	Descrizione
1	2	tuffarti nel fossato
1	3	percorrere il ponte levatoio
2	1	Premi un tasto per ricominciare.
3	4	entrare nella stanza a sinistra
3	5	salire l'ampia scalinata
3	6	provare la porta di destra
4	7	avvicinarti allo scrigno
4	8	perlustrare la stanza in cerca di qualche trappola
4	3	tornare nell'atrio
5	9	prendere la Spada Invincibile
6	3	tornare nell'atrio
...

Spero il meccanismo sia abbastanza chiaro.

Il gioco:

- legge la riga 1 di “luoghi” e la presenta all’utente.
- Legge le righe della tabella “entra_esci” che hanno il numero corrispondente nella colonna “Provenienza” e le manda a schermo precedute da un numero progressivo (o da un char iniziando da ‘a’, che però richiede più codice per essere gestito).
Quindi, p.e., quando legge il ‘luogo == 3’, al termine presenterà le seguenti opzioni:
“1) entrare nella stanza a sinistra”
“2) salire l’ampia scalinata”
“3) provare la porta di destra”
- Attende la risposta dell’utente. Valuta la risposta in base alle corrispondenze della colonna “Destinazione” che ha in memoria e va a leggere il rigo corrispondente di “luoghi”.
Sempre in base all’esempio precedente, se l’utente preme 1) allora si interpreterà come 4, se preme 2) come 5 e se preme 3) come 6.
- Il loop riparte.

In questo caso un valore 0 potrebbe indicare l’uscita dal gioco.

In questa versione il numero delle ‘uscite’ può addirittura essere sconosciuto quando si scrive un rigo nella tabella ‘luoghi’: vedi ad es. il ‘luogo’ 4, dove nella descrizione non compare nessuna delle opzioni che poi compariranno comunque a schermo all’utente.

Se ciò rende discutibili i nomi ‘luoghi’ e ‘entra_esci’ per le tabelle, se ne possono trovare di più appropriati :-)

Aggiungere oggetti

Se il gioco consistesse nel muoversi da un ‘luogo’, o forse sarebbe meglio dire da una ‘situazione’ a un’altra, allora il divertimento sarebbe ridotto.

Si possono però aggiungere complicazioni che rendono la partita più interessante, come la possibilità di raccogliere o usare oggetti.

Oggetti nella versione base

Nella versione base del gioco si avrebbe un’altra tabella all’incirca così:

oggetti					
IDOggetto	Nome	Posizione	Trasportabile	Dipende_da	Elimina
1	Spada Invincibile	5	1	0	0
2	Scrigno chiuso	4	0	0	0
3	Scrigno aperto	4	1	4	0
4	Chiave magica	7	1	0	2
5	Tesoro	4	1	3	3

Il loop da un punto di vista del codice

Il loop di pag. 3 si adatterebbe su per giù così:

- dopo aver mandato a schermo la descrizione della stanza, controlla nella tabella 'oggetti' se ci sono elementi che riportano nella scheda 'Posizione' il numero della stanza.
- Se ce ne sono, considera il loro valore "Dipende_da". Se il valore è zero, allora l'oggetto è considerato subito "BUONO", altrimenti:
 - il valore trovato è il riferimento a un altro oggetto. Si controlla nella colonna "Posizione" di questo oggetto collegato se compare il valore '-1'. Se "Posizione" è -1, allora anche l'oggetto di partenza è considerato "BUONO".
- Se ce n'è almeno uno "BUONO", scrive: "Puoi anche u) usare:" + il nome di ogni elemento che corrisponde al criterio.

Nell'esempio precedente, alla sala del trono si aggiungerebbe:

"Puoi anche u) usare:

- Scrigno chiuso"

- Se l'utente preme "u", ciò che succede dipende da una serie di fattori:
 - se ci sono più elementi, dopo 'u' l'utente dovrà anche digitare un numero o una lettera che specifica l'elemento.
 - Specificato l'elemento, se questo è Trasportabile ('1'), la sua posizione viene modificata in '-1'. Nota: '-1' va inteso come un valore speciale che identifica ciò che il personaggio ha con sé, diciamo nelle tasche.
 - Se l'elemento non è Trasportabile, allora il programma scriverà «Non puoi farlo». Il senso di questo passaggio è che, se l'utente incontra un oggetto che si deve combinare con un altro, o ha con sé l'altro oggetto (che deve ovviamente essere trasportabile!) oppure non può fare nulla.
 - Se l'oggetto è Trasportabile AND nella colonna Elimina riporta un valore diverso da zero, allora:

- quel valore va considerato il riferimento a un altro oggetto. Quest'oggetto collegato va modificato così: nella sua colonna Posizione va scritto '0' (zero).

Il loop dal punto di vista del funzionamento

La descrizione meccanica soprastante di certo è tutto fuorché chiara, perciò proviamo a seguire i passaggi.

Es.

Mettiamo che l'utente entri nella stanza (4). Il programma riporta la descrizione della stanza, poi verifica la tabella "oggetti".

Trova 3 oggetti che hanno Posizione == 4, ossia il 2, il 3 e il 5.

Controlla la loro colonna "Dipende_da" e trova che il 3 e il 5 hanno un valore diverso da 0. A questo punto va a controllare gli oggetti il cui "IDOggetto" corrisponde ai valori trovati, ossia il 4 e il 3, e scopre che nessuno di loro si trova nella Posizione -1. Stabilisce perciò di ignorare gli oggetti 3 e 5.

Riscrivo questo passaggio:

- a) oggetto 2:
 - ✗ → Posizione == 4 (OK!)
 - ✗ → Dipende_da == 0 (OK!)
 - ➔ → oggetto BUONO.
- b) oggetto 3:
 - ✗ → Posizione == 4 (OK!)
 - ✗ → Dipende_da == 4 (PROBLEMA!)
 - ➔ → Posizione oggetto 4 != -1 (ERRORE!)
 - ⚡ → oggetto FALLATO.
- c) oggetto 5:
 - ✗ → Posizione == 4 (OK!)
 - ✗ → Dipende_da == 3 (PROBLEMA!)
 - ➔ → Posizione oggetto 3 != -1 (ERRORE!)
 - ⚡ → oggetto FALLATO

Manda a schermo:

Puoi anche u) usare:

a) Scrigno chiuso

Se l'utente seleziona 'u' e poi 'a', il programma però risponde: «Non puoi farlo», perché Trasportabile == 0, Dipende_da == 0 e Elimina == 0.

Lo so, suona un po' schizofrenico, lo ammetto :-)

Si tratta però della versione base del programma, che deve sacrificare parte dell'usabilità alla stringatezza del codice.

Vediamo un altro es.

L'utente entra nella stanza 7 (che nei precedenti esempi di tabella 'luoghi' non compare, ma fabbrichiamocela ora con l'immaginazione).

Manda a schermo la descrizione della stanza, dopodiché controlla la tabella 'oggetti' e ne trova uno con Posizione == 7 AND Dipende_da == 0; l'oggetto è quindi BUONO.

Il programma manda a schermo:

Puoi anche u) usare:

a) Chiave magica

Se l'utente scrive 'u' e poi 'a', il programma risponde «Hai preso: Chiave magica», perché Trasportabile == 1.

A questo punto:

- a) la Posizione dell'oggetto 4 viene cambiata in -1.
- b) Viene esaminata la colonna Elimina dell'oggetto 4 e viene trovato il valore 2.
- c) L'oggetto con valore 2 viene così trasformato: la sua Posizione diviene 0 (Nota: il 'luogo' 0 non esiste!).

Poniamo che dopo il passaggio precedente l'utente torni nella stanza 4.

A questo punto la situazione è mutata perché la tabella 'oggetti' si presenterà così:

oggetti					
IDOggetto	Nome	Posizione	Trasportabile	Dipende_da	Elimina
1	Spada Invincibile	5	1	0	0
2	Scrigno chiuso	0	0	0	0
3	Scrigno aperto	4	1	4	0
4	Chiave magica	-1	1	0	2
5	Tesoro	4	1	3	3

Il programma trova quindi 2 oggetti, non più 3, nella stanza 4, ma condurrà la stessa analisi di prima e trarrà le seguenti conclusioni:

- a) oggetto 3: Posizione == 4 (OK!) → Dipende_da == 4 (PROBLEMA!) → Posizione oggetto 4 == -1 (OK!) → oggetto BUONO.
- b) oggetto 5: Posizione == 4 (OK!) → Dipende_da == 3 (PROBLEMA!) → Posizione oggetto 3 != -1 (OK!) → oggetto FALLATO.

A questo punto manderà a schermo:

Puoi anche u) usare:

a) Scrigno aperto

Se l'utente scrive 'u' e poi 'a', il programma risponde «Hai preso: Scrigno aperto», perché Trasportabile == 1.

A questo punto:

- a) la Posizione dell'oggetto 3 viene cambiata in -1.
- b) Viene esaminata la colonna Elimina dell'oggetto 3 e viene trovato 0, che non comporta conseguenze.

A questo punto lo schermo verrà ridisegnato per dar la possibilità all'utente di fare un'altra scelta (è ovvio che dopo aver usato un oggetto, l'utente deve comunque indicare dove vuole andare).

Ma qui incontriamo una sorpresa: la tabella 'oggetti' è cambiata di nuovo in questo modo:

oggetti					
IDOggetto	Nome	Posizione	Trasportabile	Dipende_da	Elimina
1	Spada Invincibile	5	1	0	0
2	Scrigno chiuso	0	0	0	0
3	Scrigno aperto	-1	1	4	0
4	Chiave magica	-1	1	0	2
5	Tesoro	4	1	3	3

E questo ha delle conseguenze. Infatti il programma tornerà a scorrere la tabella oggetti e troverà che:

- l'oggetto 5 è nella Posizione 4;
- Dipende_da == 3, ma l'oggetto 3 si trova nella Posizione -1;
- ciò significa che ora l'oggetto 3 è BUONO; anzi, è l'unico BUONO nella Posizione 4.

Pertanto il programma manderà a schermo:

Puoi anche u) usare:

a) Tesoro

Se l'utente scrive 'u' e poi 'a', il programma risponde «Hai preso: Tesoro», perché Trasportabile == 1.

A questo punto:

- a) la Posizione dell'oggetto 5 viene cambiata in -1.
- b) Viene esaminata la colonna Elimina dell'oggetto 3 e viene trovato 3.
- c) La Posizione dell'oggetto 3 viene cambiata in 0.

Considerazioni

Fin qui abbiamo visto ciò che succede all'interno del database, ma dal punto di vista del giocatore, quel che è successo è che ha trovato uno scrigno chiuso e non l'ha potuto prendere ('usare'); poi però ha preso una chiave magica e, quando è tornato sui suoi passi, lo scrigno chiuso si è trasformato in uno scrigno aperto; quando ha provato a prenderlo, vi ha scovato un tesoro e se l'è preso. Certo, le cose non son così semplici: all'ultima battuta l'utente potrebbe decidere di lasciar lì il tesoro e andarsene con uno scrigno vuoto nelle tasche, ma la versione base del gioco presenterà di certo, nel giocarci, incongruenze anche peggiori di questa.

Vorrei però rilevare che il codice non deve fare nulla sui dati, se non esattamente ciò che il database lo istruisce a fare. In pratica il codice non ha una sua logica specifica che potrebbe rendere rilevanti le differenze fra un linguaggio e un altro, bensì si limita a leggere dati da una casella e scriverli in un'altra. Male che vada, li manda a schermo.

È ovvio che, se aggiungiamo gli oggetti con questo criterio, la tabella 'luoghi' iniziale dovrà essere modificata.

Il 'luogo' 4 ad es. andrà ad assomigliare a:

luoghi			
IDLuogo	Descrizione	Scelta_A	Scelta_B
...
4	Sei nella sala del tesoro. In mezzo alla stanza noti svariati oggetti.	6	7
...

Infatti non siamo più in grado di dire quando un oggetto sarà presente e quando no.

Sarà però possibile arricchire la tabella ‘oggetti’ di una nuova colonna, “Descrizione”, la quale conterrà proposizioni del tipo: «Cattura la tua attenzione uno Scrigno chiuso, di pregiata fattura...» ecc. ecc. Le frasi contenute in questa colonna, una volta stabilito quali oggetti siano presenti e, diciamo, ‘visibili’, possono essere presentate in coda alla descrizione della stanza.

Oggetti in una versione avanzata

Una versione avanzata, ma senza dubbio e più flessibile, degli oggetti richiede un maggior numero di tabelle.

Proviamo a inventarne una.

Le seguenti tabelle vanno considerate tutte insieme perché sono connesse fra di loro.

oggetti			
IDOggetto	Nome	Descrizione	Posizione
1	Spada Invincibile	La lama affilatissima di questa splendida spada splende come fosse cristallo.	5
2	Scrigno chiuso	È uno scrigno robusto, pesante, ma al tempo stesso elegante. Il suo coperchio sembra chiuso ermeticamente e quasi indistruttibile.	4
3	Scrigno aperto	...	0
4	Chiave magica	...	7
5	Tesoro	...	0
...

az_oggetti			
IDAzOgg	Oggetto_1	Azione	Oggetto_2
1	1	5	0
2	1	4	0
3	4	5	0
4	4	1	2

az_oggetti			
IDAzOgg	Oggetto_1	Azione	Oggetto_2
5	4	2	0
6	5	5	0

azioni	
IDAzione	Nome
1	usa
2	apri
3	chiudi
4	posa
5	prendi

conseguenze		
AzOgg	Oggetto	Luogo
1	1	-1
2	1	-2
3	4	-1
4	4	0
4	2	0
4	3	4
5	3	0
5	5	4
6	5	-1

Spero che risulti abbastanza complicato anche per voi: a me si sono intrecciati gli occhi a scriverlo! :-)

Quel che c'è da sapere di questa logica è il significato del secondo valore negativo che compare, -2. Abbiamo detto che -1 viene sfruttato per identificare le “tasche” del giocatore, ossia gli oggetti che porta con sé. Il valore -2 indica invece il ‘luogo’ in cui il giocatore si trova nel momento in cui interagisce con l’oggetto.

Altra cosa da tener presente è che ora il giocatore viene istruito, a inizio partita, della possibilità di usare, oltre le opzioni presentate a schermo, anche i comandi: “u” (usa), “r” (apri), “h” (chiudi), “p” (posa) e “n” (prendi) (Nota: meglio non usare le prime lettere dell’alfabeto).

Vediamo come funzionerebbe l’esempio di prima con la nuova logica, però evitando di descrivere il meccanismo dal punto di vista del codice come abbiamo fatto prima.

Esempi

Nella Stanza delle Armature (luogo 5), l’utente incappa nella Spada Invincibile (oggetto 1).

Può scegliere se prenderla oppure no perché nella tabella az_oggetti c’è scritto che per l’oggetto 5 è prevista l’azione 5, che è “prendi”.

Se l’utente decide di prenderla, allora viene eseguito ciò che è previsto nella tabella “conseguenze” in corrispondenza del numero 1 della colonna AzOgg; questo perché l’azione-su-oggetto che stiamo compiendo ha IDAzOgg == 1.

Scrivendo a occhio l’istruzione, e probabilmente sbagliandone la sintassi, ponendo uguale a 1 il parametro “azione-oggetto”, avremmo:

```
SELECT * FROM conseguenze WHERE AzOgg=azione-oggetto
```

In questo caso esce fuori solo una riga di risultato, la prima. Eseguendo quelle istruzioni, l’oggetto 1 (la Spada Invincibile) viene messa... ok, in tasca. Non è molto comodo, ma bisogna accontentarsi.

Per l’oggetto 1 (Spada Invincibile) è prevista anche l’istruzione 4 (posa) – stiamo parlando del rigo della tabella “az_oggetto” che ha IDAzOgg == 2.

Nella tabella “conseguenze” questa azione prevede, come è logico aspettarsi, che l’oggetto finisca nel luogo in cui si trova il giocatore quando sceglie questa opzione (-2).

L’azione “posa” rappresenta quindi una mosca bianca in questo programma, cioè un’azione che non si può totalmente evincere dal database. In questo caso al codice bisogna aggiungere una logica autonoma che stabilisca qual è il valore giusto da inserire nella tabella “oggetti” alla colonna “posizione” – ma per fortuna è semplice.

Andiamo ad un passaggio più complicato.

Supponiamo che la Chiave Magica (oggetto 4), che si trovava nel luogo 7, abbia trovato posto nelle tasche del giocatore (luogo -1).

A questo punto, quando entra nel luogo 4 (Sala del Tesoro), letta la descrizione della stanza e degli oggetti presenti, nonché le opzioni di ‘spostamento’ possibili, l’utente può provare a premere “u” (usa). Scegliere “u” farà comparire una lista con tutti gli oggetti presenti, ossia quelli nel ‘luogo’ e quelli nelle ‘tasche’.

L’azione “usa” (1) è prevista per l’oggetto 4 (riga con IDAzOgg == 4, per coincidenza), però nella colonna “Oggetto_2” compare un valore diverso da zero. Ciò significa che, se dalla lista sarà selezionata la Chiave Magica, allora la domanda cambierà in «Usa Chiave Magica con...?» e di nuovo la lista di tutti gli oggetti disponibili (escluso il n. 4).

Anche questo punto aggiunge una gestione dei dati che non è una stretta derivazione di ciò che è scritto nel database, ma anche qui le cose sono abbastanza semplici. La parte più intricata sarà far stare le righe ben ordinate sullo schermo (problema che i linguaggi che sfrutteranno l’interfaccia in HTML probabilmente non avvertiranno); la logica può essere gestita, così a colpo d’occhio, da degli array o delle strutture associative, perciò basta non far confusione con gli indici.

Per cercare di dare un quadro della situazione fin qui, lo schermo si potrebbe presentare su per giù in questo modo:

Sei nella sala del tesoro. In mezzo alla stanza noti svariati oggetti. Tra questi noti:

– Scrigno chiuso: È uno scrigno robusto, pesante, ma al tempo stesso elegante. Il suo coperchio sembra chiuso ermeticamente e quasi indistruttibile.

Puoi:

- 1) avvicinarti allo scrigno
- 2) perlustrare la stanza in cerca di qualche trappola
- 3) tornare nell’atrio

Puoi anche scegliere di Usare, apRire, cHiudere, Posare o prenDere qualche oggetto.

Cosa scegli? > u

Oggetti disponibili:

- 1) Spada Invincibile
- 2) Scrigno chiuso
- 3) Chiave magica
- 4) esci

Cosa scegli? > 3

Usa Chiave Magica con...?

1) Spada Invincibile

2) Scrigno chiuso

3) esci

Cosa scegli? >

Poniamo che l'utente scelga (2); a quel punto il programma eseguirà le istruzioni contenute nella tabella “conseguenze” che presentano il valore 4 nella colonna AzOgg. Per risparmiarvi la fatica di andarle a rileggere, ve le copio-incollo qui:

conseguenze		
AzOgg	Oggetto	Luogo
4	4	0
4	2	0
4	3	4

Queste istruzioni non fanno altro che ‘spostare’ un oggetto dal luogo in cui si trova a un altro, solo che alcuni luoghi hanno un significato speciale:

- il luogo 0 non esiste, perciò l'oggetto ‘scompare’ dal gioco;
- il luogo -1 sono le tasche, perciò da quel momento l'oggetto è sempre presente;
- il luogo -2 significa il luogo presente, perciò il valore reale che verrà scritto sulla tabella “oggetti” sarà deciso in fase di esecuzione, ossia sul momento (se volete: a “run time”).

Quindi, nel concreto:

- l'oggetto 4 (la Chiave magica) sparisce;
- l'oggetto 2 (lo Scrigno chiuso) sparisce;
- l'oggetto 3 (lo Scrigno aperto) appare nel luogo 4.

Non credo ci sia molto da aggiungere dopo aver visto questi esempi, perché lo schema si ripete sempre uguale.

Aggiungere personaggi

Mi riferisco qui a quelli che vengono spesso nei giochi definiti personaggi non giocanti (PNG) o NPC (non-player character).

Nei giochi ‘seri’, se mi perdonate l’ossimoro, gli NPC hanno un ruolo molto importante e vengono gestiti da un’intelligenza artificiale; nel nostro caso aggiungeranno un po’ di colore (nella versione base) o tutt’al più daranno qualche dritta al giocatore (nella versione avanzata).

Personaggi nella versione base

Per personaggio s’intende una situazione in cui è possibile scambiare delle frasi con il programma secondo uno schema precostituito che farà del suo meglio per simulare un dialogo.

Vediamo alcune possibilità.

Personaggi		
IDPers	Nome	Posizione
1	Specchio magico	5
2	Grillo parlante	7
...	...	

Fino a qui la logica è la medesima degli oggetti: questa tabella ci dice solo dove si trova ogni personaggio. Volendo vi si può aggiungere una descrizione del PNG.

In base a questa tabella, quando il giocatore entra nella stanza 5 si troverà davanti questa descrizione:

Sei nella stanza delle armature. Su una parete fanno bella mostra di sé alcune spade dall’elsa intarsiata e altri armamentari. Tra questi noti:

- Spada Invincibile: La lama affilatissima di questa splendida spada splende come fosse cristallo.

Noti anche:

- Specchio magico: [...eventuale descrizione...]

Puoi:

- 1) tornare nell’atrio del castello
- 2) avventurarti nella Stanza Buia
- 3) avvicinarti alle armature immobili

Puoi anche scegliere di Usare, apRire, cHiudere, Posare o prenDere qualche oggetto.

Puoi anche conVersare con qualcuno.

Cosa scegli? >

Come prima, se l’utente sceglie “v”, gli verrà proposta una lista, nel nostro caso di un solo rigo, dei PNG presenti.

L'interazione con il PNG si potrebbe invece basare su una tabella come la seguente:

frasiGioc			
Pers	Serie	Frase	Risposta
1	1	«Chi sei?»	1
1	1	«Cosa fai?»	2
1	1	«Ti ho visto in qualche favola?»	3
1	1	«Ti senti mai una pallida immagine di te stesso?»	4
1	2	«Che mi dici di queste armature?»	5
1	2	«C'è qualche tesoro nel castello?»	6
1	2	«Sai che tempo farà domani?»	7
1	3	«Ma come hai fatto a ritrovarti di là dal vetro?»	8
1	3	«Sai come uscire dal castello?»	9
1	3	«Puoi darmi qualche consiglio?»	10
1	1000	«Puoi dirmi qualcos'altro?»	11
...

frasiPNG	
IDFrase	Risposta
1	«Un impiegato allo sportello.»
2	«Guardo il mondo da un oblò.»
3	«Sai che stavo per chiederti la stessa cosa? Sei sicuro di essere reale?»
4	«Purtroppo sì. Spesso. Non so resistere ad una buona peperonata.»
5	«Non è che parlino granché. A dire il vero, mi sembrano tutte delle zucche vuote.»
6	«Certo! Ce ne sono più d'uno. Ci sono io, per esempio, che sono un vero tesoro; poi c'è il divertimento di girellare qua e là per le stanze; ah, sì: anche uno scrigno che si apre solo con una chiave magica.»

frasiPNG	
IDFrase	Risposta
7	«Mi spiace, ma ho usato il giornale per pulire il vetro.»
8	«E tu come hai fatto a ritrovarti di là dal vetro?»
9	«Mmmm... Fammi pensare... Fammi pensare... Hai provato dalla porta d'ingresso?»
10	«Volentieri. Lavati sempre le mani prima di mangiare.»
11	«Sono stanco e non ho più voglia di parlare.»
...	...

Per capire il funzionamento bisogna tener presente che:

- il dialogo inizia sempre con una frase del giocatore;
- il giocatore dovrebbe, se possibile, avere più frasi a disposizione tra cui scegliere – un buon numero sarebbe 3;
- ad ogni frase del giocatore corrisponde sempre una singola risposta da parte del PNG;
- è opportuno lasciare una frase finale da usare quando, avendo esaurito tutte le risposte, il giocatore insiste a voler parlare con il PNG;
- il fatto che in questo esempio la numerazione di tutte le frasi sia sequenziale con un intervallo costante di 1 è del tutto casuale; o per meglio dire: è dovuta alla natura semplificata dell'esempio.

In pratica quando il giocatore sceglie una frase, il PNG risponde con una determinata altra; il rapporto fra queste due frasi è biunivoco.

Esempio di funzionamento

Vediamo come potrebbe funzionare l'esempio.

ATTENZIONE! Si è scelto un limite di frasi da presentare a schermo di volta in volta, che è 3. Il limite definitivo è da stabilirsi, ma è ovvio che deve esistere.

L'utente sceglie di conversare con lo Specchio magico. Gli viene proposta la prima scelta di frasi da dire:

- 1) «Chi sei?»
- 2) «Cosa fai?»
- 3) «Ti ho visto in qualche favola?»

Come fa il programma a decidere da quali frasi iniziare? La risposta è insita nel numero nella colonna “Serie”. Funziona così:

- all’inizio del primo ciclo il contatore è impostato su 1;
- quando tutte le frasi della serie 1 vengono dette almeno una volta, il contatore viene incrementato di 1 e diventa 2;
- a questo punto vengono proposte tutte le frasi della serie 2 fino al loro esaurimento, cosa che fa scattare il contatore su 3;
- le serie vengono quindi proposte in sequenza, fino all’ultima frase dell’ultima serie.

Nel nostro caso le frasi della prima serie sono (volutamente) 4, ma abbiamo stabilito un limite a schermo di 3 per volta, pertanto vengono presentate le prime tre.

Poniamo che il giocatore scelga la 2 («Cosa fai?»). Il PNG risponderà: «Guardo il mondo da un oblò.»

Penso ci foste già arrivati da soli, ma tanto vale specificarlo: il numero nella colonna Risposta specifica la frase da selezionare nella tabella “frasiPNG” tramite la corrispondenza nella colonna ID-Frase.

La cosa più importante, però, è che il numero nella colonna Serie della frase scelta dal giocatore verrà incrementato di mille. Pertanto la frase «Cosa fai?» entrerà a far parte della serie 1001.

Andiamo avanti. Sotto la risposta del PNG compariranno le nuove tre frasi tra cui scegliere, che saranno:

- 1) «Chi sei?»
- 2) «Ti ho visto in qualche favola?»
- 3) «Ti senti mai una pallida immagine di te stesso?»

La frase «Cosa fai?» è scomparsa perché il contatore è ancora fermo a 1, mentre quella corrisponde a 1001.

Per farla breve, supponiamo che si sviluppi un ‘dialogo’ come questo:

1) «Chi sei?» 2) «Cosa fai?» 3) «Ti ho visto in qualche favola?» Cosa scegli? > 2 Tu: «Cosa fai?» Specchio magico: «Guardo il mondo da un oblò.» 1) «Chi sei?» 2) «Ti ho visto in qualche favola?»

3) «Ti senti mai una pallida immagine di te stesso?»

Cosa scegli? > 2

Tu: «Ti ho visto in qualche favola?»

Specchio magico: «Sai che stavo per chiederti la stessa cosa? Sei sicuro di essere reale?»

1) «Chi sei?»

2) «Ti senti mai una pallida immagine di te stesso?»

3) «Che mi dici di queste armature?»

Cosa scegli? > 1

Tu: «Chi sei?»

Specchio magico: «Un impiegato allo sportello.»

1) «Ti senti mai una pallida immagine di te stesso?»

2) «Che mi dici di queste armature?»

3) «C'è qualche tesoro nel castello?»

Cosa scegli? > 1

Tu: «Ti senti mai una pallida immagine di te stesso?»

Specchio magico: «Purtroppo sì. Spesso. Non so resistere ad una buona peperonata.»

1) «Che mi dici di queste armature?»

2) «C'è qualche tesoro nel castello?»

3) «Sai che tempo farà domani?»

Cosa scegli? > 2

Tu: «C'è qualche tesoro nel castello?»

Specchio magico: «Certo! Ce ne sono più d'uno. Ci sono io, per esempio, che sono un vero tesoro; poi c'è il divertimento di girellare qua e là per le stanze; ah, sì: anche uno scrigno che si apre solo con una chiave magica.»

1) «Che mi dici di queste armature?»

2) «Sai che tempo farà domani?»

3) «Ma come hai fatto a ritrovarti di là dal vetro?»

Cosa scegli? > 2

Tu: «Che mi dici di queste armature?»

Specchio magico: «Non è che parlino granché. A dire il vero, mi sembrano tutte delle zucche vuote.»

1) «Sai che tempo farà domani?»
 2) «Ma come hai fatto a ritrovarti di là dal vetro?»
 3) «Sai come uscire dal castello?»
 Cosa scegli? > 1
 Tu: «Sai che tempo farà domani?»
 Specchio magico: «Mi spiace, ma ho usato il giornale per pulire il vetro.»

1) «Ma come hai fatto a ritrovarti di là dal vetro?»
 2) «Sai come uscire dal castello?»
 3) «Puoi darmi qualche consiglio?»
 Cosa scegli? > 1
 Tu: «Ma come hai fatto a ritrovarti di là dal vetro?»
 Specchio magico: «E tu come hai fatto a ritrovarti di là dal vetro?»

A questo punto la tabella “frasiGioc” si presenterà così:

frasiGioc			
Pers	Serie	Frase	Risposta
1	1001	«Chi sei?»	1
1	1001	«Cosa fai?»	2
1	1001	«Ti ho visto in qualche favola?»	3
1	1001	«Ti senti mai una pallida immagine di te stesso?»	4
1	1002	«Che mi dici di queste armature?»	5
1	1002	«C'è qualche tesoro nel castello?»	6
1	1002	«Sai che tempo farà domani?»	7
1	1003	«Ma come hai fatto a ritrovarti di là dal vetro?»	8
1	3	«Sai come uscire dal castello?»	9
1	3	«Puoi darmi qualche consiglio?»	10
1	1000	«Puoi dirmi qualcos'altro?»	11
...

Spero l'idea sia chiara.

Il meccanismo purtroppo non è perfetto: infatti il giocatore può decidere di esaurire tutte le risposte della Serie 2 prima di quelle della Serie 1, ma il codice per gestire questo problema è assai semplice (ne riparlo poco più avanti).

Nel caso dei personaggi, il database deve essere gestito con un po' di logica da parte del programma, perciò posso affermare che i dialoghi non saranno difficili solo da concepire, ma rappresenteranno la parte più avanzata della stesura del programma.

Complicazioni

Vediamo alcune complicazioni che presentano:

Gestire il limite

Per la prima volta le risposte delle query non ci vanno bene così come sono: il database potrebbe restituire più risultati di quelli che possiamo gestire. Nel caso, ad es., che il limite impostato sia di tre frasi per volta e la Serie ne preveda molte di più, il codice dovrà 'respingere' i dati in eccesso, rimandandoli al futuro.

Le Serie paiono inutili

Le frasi vengono proposte in sequenza e fatte sparire assegnando loro un valore superiore a 1000. Questo meccanismo NON ha bisogno del concetto di serie, bensì basta che le frasi abbiano un numero progressivo, da incrementare di 1000 al bisogno.

Vero. L'idea di Serie mi è venuta più per motivi psicologici che legati al codice. Ad esempio: il fatto che le frasi vengano presentate a blocchi a schermo tende a farle concepire a blocchi: le serie enfatizzano questo processo. Un'altra considerazione è che pensare che il database sarà creato in modo ordinato come lo troviamo in questo esempio significa peccare di ottimismo.

Noi abbiamo considerato un caso molto limitato, ossia la conversazione con il PNG 1 e basta. In una situazione più verosimile si progetterà un dialogo iniziale, poi si passerà al dialogo con il PNG 2, aggiungendo le frasi al database, poi al 3, e d'un tratto si scoprirà di voler aggiungere qualcosa al dialogo con il PNG 1.

Se non usiamo le serie, saremo di fatto costretti a mettere le nuove frasi in coda a tutte e altre, pena dover rinumerare a mano tutta la serie.

Se usiamo le serie, ci basta accodarle a una serie.

In entrambi i casi è comunque possibile non numerare le serie/le singole frasi con un intervallo di 1, bensì si può usare un intervallo 5 o 10 per lasciare spazio per ulteriori immissioni. Il ritardo che questo potrebbe creare nelle interrogazioni al database, stante la natura del programma e il numero di richieste, sarebbe impercettibile per l'uomo.

Le Serie si possono esaurire in ordine sparso

Purtroppo sì: non sono riuscito a fare una ciambella col buco.

Per gestire questo problema, posto che si vogliano lasciare le serie, si può usare un contatore che parte da 1 e scorra tutte le serie.

Un possibile meccanismo che gestisca l'intero problema potrebbe essere il seguente:

Premessa:

Ogni PNG sul database deve includere una serie di numero 1000 che contiene esattamente una frase.

Prima fase:

Quando il giocatore decide di conversare con un PNG, allora come prima cosa si chiedono al database tutte le Serie di quel PNG:

```
SELECT Serie FROM frasiGioc;
```

A quel punto si memorizza qual è la serie dal valore più elevato che c'è, purché sia inferiore a mille. A tal fine ricordo che da molti database si possono ottenere risultati già ordinati e senza i duplicati, perciò, chiedendo i risultati in ordine inverso, un loop alla seconda iterazione (alla prima otterrebbe 1000!) potrebbe già darci la risposta che cerchiamo.

A questo punto abbiamo un valore serie_piu_alta che ci fornisce un comodo limite.

Seconda fase:

- si imposta un contatore al valore 1.
- Si tenta una query sul modello della seguente (più che la correttezza della sintassi, ciò che conta è la logica):

```
SELECT Frase FROM frasiGioc WHERE Serie (BETWEEN 1 AND contatore) AND Pers=png;
```

Se il risultato è 0...

→ ...se contatore è inferiore a serie_piu_alta, allora:

- contatore viene incrementato di 1;
- si ritenta la query.

→ ...se invece contatore è uguale a serie_piu_alta, allora contatore viene impostato a 1000.

Il precedente blocco vuol dire: se abbiamo già scorso tutte le serie (contatore == serie_piu_alta) e il risultato è comunque zero (perché tutte le frasi sono state dette, pertanto sono state numerate con valori superiori a 1000), allora rimane solo la frase 1000. Da quella non ci dovremo più spostare.

```
if((0 < risultato) && (risultato < 3))
```

Ossia: se il risultato (cioè il numero delle frasi restituite) è maggiore di 0, ma inferiore a 3, allora:

→ se contatore < serie_piu_alta, allora:

- incrementiamo contatore di 1;
 - ritentiamo la query.
- se contatore == serie_piu_alta, allora mandiamo i risultati a schermo.

Il blocco soprastante significa: cerchiamo di ottenere sempre almeno 3 frasi. Se non ci riusciamo, allarghiamo l'intervallo di valori validi.

È molto probabile che mi sfugga qualcosa e il meccanismo non funzioni, ma secondo me può valere la pena provarci.

ID delle frasi

Non ho aggiunto una colonna con un valore univoco (di solito chiamato ID) alle frasi di frasiGiac, ma se questo dovesse semplificare il processo di associare la scelta dell'utente con una determinata frase, sarei il primo a votare per il suo inserimento. A ogni modo, nella fase avanzata diventano indispensabili.

Come potete notare, nel caso dei dialoghi il codice si riconquista il suo posto d'onore all'interno del programma, rifiutandosi di essere il mero esecutore di disposizioni del database. Però c'è ancora spazio per complicare un po' le cose.

Personaggi in una versione avanzata

Per coerenza con le prime versioni di bivi ho presentati esempi che replicavano quelle ambientazioni. Vien però da domandarsi se, mentre per alcuni tipi di giochi, come quelli fantasy, i dialoghi possono sembrare delle mere decorazioni di cui si potrebbe fare a meno, invece per altri non potrebbero costituire la colonna portante della 'trama'.

Consideriamo il caso di una storia ambientata nel mondo reale e in una situazione in cui il ragazzo che viene al CoderDojo si potrebbe riconoscere; faccio l'esempio di un gioco in cui ci si pone come obiettivo di organizzare una festa o una gita di classe. In un caso come questo non ci viene da pensare che il fatto di poter simulare un dialogo con i compagni di classe, i genitori, i professori sarebbe uno degli stimoli maggiori per il ragazzo per scrivere la storia?

In realtà per dare ai dialoghi, e di conseguenza ai PNG, un ruolo da protagonista non serve aggiungere molto codice – ma il database, quello sì, si complica e non poco.

Esaminiamo 3 casi separati, ma simili:

- interazioni fra i PNG e il giocatore;

- interazioni fra i PNG e gli oggetti;
- interazioni fra i PNG e i luoghi.

Interazioni fra PNG e giocatore

Un dialogo non deve per forza essere lineare come l'abbiamo visto fin qui. Un PNG potrebbe avere qualche 'dritta' importante da dare al giocatore, ma in quel caso il giocatore potrebbe doversela sudare.

Supponiamo che il creatore della storia s'inventi un'associazione improponibile, tipo "usa fetta di salame con registro di classe"; magari avrà voglia di dare un suggerimento al giocatore tramite qualche dialogo. Abbiamo fatto in modo che PNG 'parlassero' solo in risposta alle frasi del giocatore, ma potremmo fare in modo che una frase, o un'intera Serie di frasi, del giocatore appaia solo se in precedenza il giocatore ne ha posto una specifica.

Mettiamo che fra le domande da fare ne sia inclusa una che recita così: «Ti va di parlare della riforma censuaria di Solone?». Possiamo ipotizzare che il giocatore non la sceglierà, nemmeno quando è rimasta l'ultima, specie se altre domande trappola hanno già condotto a dei vicoli ciechi. Però nel nostro caso la risposta a questa domanda, quale che sia, andrà a sbloccare altre domande che fino a quel punto erano rimaste nascoste.

Modifiche necessarie:

1) Nella tabella "frasiGioc":

- deve essere aggiunta una colonna con valori univoci;
- compariranno frasi collocate in Serie con valore 0.

frasiGioc				
IDFraseG	Pers	Serie	Frase	Risposta
1	1	1	«Chi sei?»	1
2	1	1	«Cosa fai?»	2
3	1	1	«Ti ho visto in qualche favola?»	3
...
12	1	0	«Sai per caso come si fa ad aprire lo Scrigno nella Sala del Tesoro?»	12
...

2) Nella tabella “frasiPNG”:

– deve essere aggiunta una colonna “Frase-Frase”.

frasiPNG		
IDFrase	Risposta	Frase-Frase
1	«Un impiegato allo sportello.»	0
2	«Guardo il mondo da un oblò.»	0
...
6	«Certo! Ce ne sono più d'uno. Ci sono io, per esempio, che sono un vero tesoro; poi c'è il divertimento di girellare qua e là per le stanze; ah, sì: anche uno scrigno che si apre solo con una chiave magica.»	4
...
12	«Per aprire lo Scrigno nella Sala del Tesoro dovrai usare la Chiave magica che trovi nella stanza successiva.»	0
...

3) Si crea una nuova tabella, che per ora possiamo chiamare “consFraseFrase”:

consFraseFrase		
IDFraseFrase	FraseGioc	Serie
...
4	12	4
5	14	3
6	20	1
...

Non sto a spenderci chissà quante parole perché lo schema è lo stesso che abbiamo seguito fin qui.

Se nella nuova colonna Frasi-Frasi di “frasiPNG” compare un numero diverso da zero, allora:

- si va a cercare la riga con quel numero nella colonna “IDFrasiFrasi” di “consFrasiFrasi”;
- il numero nella colonna “FraseGioc” ci rimanda alla stessa riga nella tabella “frasiGioc”, colonna “IDFraseG”;
- il numero nella colonna Serie va sostituito con la Serie della tabella “consFrasiFrasi”.

La conclusione è che una frase che fino a poco prima non compariva tra le opzioni del giocatore adesso appare; è ovvio che si deve garantire che abbia una risposta!

Interazioni fra PNG e gli oggetti

Questo aspetto è ancora più interessante e aiuta a creare situazioni di questo tipo:

- a) il PNG ‘dà’ qualcosa al giocatore;
- b) il PNG rivela al giocatore qualcosa che gli rende possibile trovare o usare un oggetto.

In entrambi i casi la tabella “frasiPNG” vede l’aggiunta di un’ulteriore colonna, “Frasi-Ogg”.

frasiPNG			
IDFrase	Risposta	Frasi-Frasi	Frasi-Ogg
1	«Un impiegato allo sportello.»	0	0
2	«Guardo il mondo da un oblò.»	0	0
...
6	«Certo! Ce ne sono più d’uno. Ci sono io, per esempio, che sono un vero tesoro; poi c’è il divertimento di girellare qua e là per le stanze; ah, sì: anche uno scrigno che si apre solo con una chiave magica.»	4	0
...
12	«Per aprire lo Scrigno nella Sala del Tesoro dovrai usare la Chiave magica che trovi nella stanza successiva.»	0	3
...

Il PNG dà qualcosa al giocatore

Si deve provvedere ad una nuova tabella, “consFrasiOgg”:

consFraseOgg			
ID_CFO	Oggetto	Posizione	Descrizione
...
3	4	-1	Ti compare magicamente in mano una Chiave magica.
...

Non so se vi stiate annoiando nel leggere queste descrizioni quanto me nello scriverle, ma per prudenza sarò di una stringatezza encomiabile.

Poniamo il caso che all'inizio della partita la Chiave magica sia 'nascosta' nel 'luogo' 0.

Una volta che il PNG ha pronunciato la frase 12, essa si sposterà nelle tasche del giocatore (luogo -1). Il programma segnerà il fatto tramite la frase in "Descrizione" (è uno Specchio magico, non ce lo vedo ad allungare il braccio).

Il PNG rivela al giocatore qualcosa che gli rende possibile trovare un oggetto

Si può rimediare con la stessa tabella "consFraseOgg":

consFraseOgg		
ID_CFO	Oggetto	Posizione
...
3	4	7
...

Poniamo il caso che all'inizio della partita la Chiave magica sia 'nascosta' nel 'luogo' 0.

Una volta che il PNG ha pronunciato la frase 12, la Chiave magica diventa disponibile nella stanza 7.

Qui la descrizione non è indispensabile, ma non per questo si devono creare due tabelle distinte. Visto che stiamo parlando della fase avanzata, direi che è prematuro cercare di prevedere tutte le possibili soluzioni (nonché tutti i possibili problemi!).

Si potrebbe modificare l'uso di un oggetto tramite una tabella "consFraseAz", ma lo schema da applicare, fatti i dovuti aggiustamenti, è sempre lo stesso e confesso che mi sta davvero fatica starci a pensare ora.

Interazioni fra i PNG e i luoghi

Questa è una modifica abbastanza semplice.

Poniamo che esista un passaggio segreto o una porta nascosta che fa accedere ad una stanza se viene trovata.

La tabella “luoghi” può rimanere com’è:

luoghi		
IDLuogo	Nome	Descrizione
...
5	Stanza armature	Sei nella stanza delle armature. Su una parete c’è una spada con su un cartello che recita:”Spada Invincibile”. Puoi:
...
14	Passaggio segreto	Si tratta di un corridoio largo, ma piuttosto basso, umido e senza il minimo spiraglio di luce. Oscuri brulichii s’odono tutt’intorno.
...

Diverso il discorso per la tabella “entra_esci”: infatti un ‘luogo’ è inaccessibile solo finché non c’è un ‘entrata’ che vi conduca! Va quindi modificata un po’:

entra_esci			
ID_EE	Provenienza	Destinazione	Descrizione
...
11	6	3	tornare nell’atrio
...
26	9	0	avventurarti nel passaggio segreto

In questa versione della tabella non si potrà utilizzare il valore 0 per indicare l’uscita dal gioco: dovremo concordare un nuovo valore, ad es. -1.

Aggiungendo una nuova colonna nella tabella “frsiPNG”...

frasiPNG				
IDFrase	Risposta	Frase-Frase	Frase-Ogg	Frase-Luoghi
...
12	«Per aprire lo Scrigno nella Sala del Tesoro dovrai usare la Chiave magica che trovi nella stanza successiva.»	0	3	...
...

...e una nuova tabella che colleghi “frasiPNG.Frase-Luoghi” con “entra_esci.Destinazione”...

consFraseLuoghi		
ID_FL	Progr_EE	Destinazione
1	26	14
...

...abbiamo già risolto il problema.

A questo punto il nostro PNG può rivelare l'esistenza di un passaggio segreto nella stanza 9, passaggio che fin qui il giocatore non aveva potuto trovare, ma che ora gli verrà proposto in automatico.

Ancora e ancora e ancora

Potrei proseguire ancora a lungo a ipotizzare sviluppi, ma non aggiungerei nulla di sensato. Aggiungere, tanto per dire, un'interazione oggetti-luoghi mi sembra quasi inevitabile: che il possesso di un oggetto possa permettere di accedere a un luogo (ad esempio aprire una porta chiusa a chiave) è nella natura stessa del gioco.

Bisogna tener poi di conto che quelli che chiamo 'luoghi' sono più delle situazioni. Se ad esempio una combinazione (oggetto) permette di aprire una cassaforte (oggetto), al passaggio successivo potrebbe essere offerto di controllare il contenuto della cassaforte; questo condurrebbe ad un 'luogo' (l'interno della cassaforte) senza dover per forza immaginare che il giocatore ci entri dentro. Quel luogo potrebbe così contenere altri oggetti senza che questo comporti la modifica del codice.

Mi viene anche in mente che il rapporto oggetto-PNG si può vedere anche dal punto di vista opposto di quello sopra descritto: il possesso di un oggetto potrebbe 'sbloccare' alcune frasi da dire al PNG...

Ma voler prevedere fin dall'inizio tutti i possibili sviluppi potrebbe portare a chiudere gli occhi davanti a nuove idee, rinviandole finché non si sono realizzate quelle previste. In realtà la mia proposta si concentra su ciò che è descritto nella fase "Da dove cominciare".

A dire il vero, ritengo che in una fase molto avanzata sarebbe possibile introdurre qualche elemento di aleatorietà, generando numeri pseudocasuali.

Che cosa manca davvero

Quel che ho descritto è lo scheletro del progetto. C'è una parte che non ha senso lasciare da realizzare ai ragazzi, ma andrebbe fornita già pronta, perché fuoriesce dal filone principale.

Questa parte riguarda le risorse in sé, ossia i problemi meno *machine independent*. I ragazzi non dovranno preoccuparsi del database in sé, ma solo delle query. Oltre a ciò, le partite dovranno essere salvabili e ripetibili. Significa che non dovrà essere manomesso il database originario, ma una sua copia, la quale rappresenterà poi il 'salvataggio'.

Conclusioni

Ritengo che le preferenze individuali che ci sono nel CoderDojo riguardo ai vari linguaggi di programmazione abbiano rappresentato un ostacolo, oltre al fatto che gli appuntamenti sono autoconclusivi e con cadenza circa mensile, a fatto di lavorare sul codice anziché su Scratch.

Parlando per me, la scarsa conoscenza che ho di molti linguaggi diffusi, ad esempio JavaScript, mi frena dal proporlo ai ragazzi con cui mi trovo a lavorare.

I recenti progetti Python con Minecraft e altri mi hanno però convinto che c'è un diffuso desiderio di lavorare con il codice e non soltanto in modo visuale. È vero che le difficoltà iniziali sono molto superiori, ma i risultati finali godono dello stesso paragone.

Con questo progetto spero di aver quantomeno contribuito ad arricchire la discussione.

Quel che sostengo è che, superato l'ostacolo del database, questo progetto è realizzabile pressoché con lo stesso codice in una grande varietà di linguaggi. L'adozione di un progetto di questo tipo potrebbe facilitare la nascita di gruppi di lavoro che, scelto un linguaggio, cerchino di realizzare la prima parte ("Da dove cominciare").

Ci sono due aspetti di questo progetto che mi piacciono in modo particolare e che vorrei sottolineare:

- 1) niente matematica
- 2) non ci sono limiti di ambientazione

Niente matematica

Sono stati nei vari appuntamenti del Coder Dojo proposti molti progetti, magari di un numero limitato di righe di codice, da realizzare per imparare le basi di qualche linguaggio. Moltissimi di questi progetti avevano a che fare con matrici, posizioni, superfici (magari curve), seni, coseni, somme di valori, rimbalzi...

Vorrei premettere che ringrazio tutti coloro che hanno speso tempo ed energie per realizzare questi tutorial e devo anche dire che molti mi sono parsi davvero geniali. Ho sempre avuto il timore, però, che da parte di alcuni ragazzi quell'aspetto matematico non riscuotesse particolare simpatia. Mi rendo conto che, quando si vuole realizzare qualcosa di grafico, quel qualcosa si dovrà muovere in un sistema di coordinate e che tutto ciò che si disegna a schermo ha una natura geometrica. Il fatto però è che questo ci porta a un ragionamento che sembra quasi legarci le mani: se voglio che ciò che propongo sia affascinante per il ragazzo, allora dovrà essere colorato e muoversi; se è colorato e si muove, allora la matematica è d'obbligo.

Quel che aveva di buono "bivi", invece, è che si trattava di un gioco interamente basato sul testo. La variante che propongo adotta un sacco di numeri, ma li usa come fosse un sudoku: quei numeri sono solo indici, connessioni, e non ci si fa alcun calcolo sopra.

Questo progetto potrà anche sembrare complicato ad alcuni, ma all'interno è molto lineare e non presenta difficoltà concettuali.

La fantasia è il limite

Il gioco propone di creare e giocare ad avventure testuali.

La prima parte consiste nel capire lo schema di fondo, dopodiché si può passare a inventare le storie. Il codice può essere rimandato a un momento successivo.

Intendo dire che, avendo una versione del programma funzionante, la prima offerta può consistere nel proporre di scrivere avventure cui altri possono giocare. Chi poi non vuole scrivere codice, può anche concentrarsi sull'inventare storie: magari il codice non gli piace, ma intanto dovrà sforzarsi di ragionare in base alle tabelle di un database. Non mi sembra poco.

Le avventure possono essere di qualsiasi tipo: un poliziesco in cui si raccattano indizi e si interrogano sospetti; un giorno qualsiasi in cui si deve sopravvivere alla vita scolastica; una situazione in cui si devono superare tutte le difficoltà per organizzare qualcosa (una festa, una gita...); un concorso di circostanze che, su qualche base spaziale, ci costringerà ad agire in fretta prima che...

Questa grande varietà di approcci e la libertà di temi trattati e di ambientazioni potrebbe rendere il progetto interessante per diversi ragazzi.