

Background (non influisce sull'avventura):

A metà del XIII secolo, l'esercito del regno di Holte è l'unica forza militare ad opporsi all'avanzata dell'impero Garay, che ha ormai conquistato l'intera area scandinava. La famiglia reale del regno di Holte ha ormai radunato l'ultimo corpo dell'esercito nel castello situato in un isolotto dell'area per opporre strenua resistenza ma da lì a poco tutti i soldati saranno sconfitti e la famiglia reale sarà prima imprigionata e poi sterminata.

Trama:

Il giocatore impersona il predestinato erede al trono Aelwith Holte, l'unico superstite a sfuggire all'assalto del castello. Egli non ha più niente da perdere e proverà, dopo mezzo secolo dalla conquista, ad assassinare il Re dell'impero e riprendersi la corona che gli spetta.

Protagonista: Aelwith Holte.

Nell'avventura assumerà un atteggiamento spietato e furtivo, tipico di un assassino. Varcherà l'entrata del castello nascondendosi in una carovana di rifornimenti, ed entrerà in azione nella notte.

Missione del protagonista:

Entrare nel Palazzo Reale, trovare e assassinare il Re. Infine prendere la Corona Reale. L'impresa non sarà facile perché per varcare all'interno del Palazzo e andare da una stanza all'altra dovrà recuperare oggetti, chiavi, leggere indizi e uccidere le guardie.

Sistema di checkpoint:

Il protagonista ha un numero di vite iniziali pari a 3. Vale inoltre la vita 'zero'.

Egli perde una vita se, dopo un combattimento, i suoi punti salute si azzerano oppure quando esegue una determinata azione su un oggetto.

Dopo aver perso una vita:

- se ne ha ancora almeno un'altra riprende il gioco dalla stanza in cui l'ha persa, con la quantità normale di punti salute e con l'inventario intatto.
- se non rimangono vite, il giocatore deve re-iniziare la partita, perdendo i progressi del gioco e ripartendo da 3 vite iniziali.

Mappa: di seguito sono elencate due rappresentazioni delle mappe. La prima ha l'unico scopo di rendere l'idea dell'ambientazione. La seconda indica il collegamento tra le varie stanze.



Attributi + Tabella Oggetti:

S = Si può salire sull'oggetto

s = Si può scendere dall'oggetto

i = Oggetto ispezionabile

p = Oggetto prendibile

A = Oggetto apribile e in tal caso il protagonista si sposta oltre l'oggetto in questione (senza dare il comando vai)

a = Oggetto apribile e in tal caso contiene altri oggetti

u = Oggetto usabile

e = Oggetto equipaggiabile

l = Oggetto leggibile

m = Oggetto mangiabile

w = Oggetto con cui il protagonista può scontrarsi.

NOTA: Se una stessa porta/portone/scala/grata collega le stanze x e y, essendo visibile da entrambe le stanze, risulta essere duplicata e quindi presente in entrambe le stanze x e y.

NOTA: In grassetto gli oggetti il cui funzionamento è simile a una chiave generica.

ID stanza	ID oggetto genitore	ID oggetto [PK]	Nome	Attributi
1	-1	10	Scala (1-70)	i, S
	-1	11	Chiocciola (1-6)	i, S
	-1	12	Prigioni	i
	-1	13	Cassetto	i, a
2	-1	20	Scultura	i, l
	-1	21	Candelabro	i, p, u
	-1	22	Falcastro	i, p, e
4	-1	40	Trono	i
	-1	41	Statua	i
	-1	42	Gufo	i, p
5	-1	50	Portale (5-10)	i, A
	-1	51	Scala (5-50)	i, S
6	-1	60	Chiocciola (6-1)	i, s
	-1	61	Porta (6-11)	i, A
	-1	62	Fuocherello	i
7	-1	70	Portone (7-12)	i, A
	-1	71	Inginocchiatoio	i
	-1	72	Grata (7-70)	i, A
8	-1	80	Porta (8-13)	i, A
	-1	81	Ritratto	i
	-1	82	Camino	i, s
	-1	83	Statuetta	i
	-1	84	Teca	i, a
9	-1	90	Portone (9-14)	i, A
	-1	91	Porta_a_bilico (9-10)	i, A
	-1	92	Tavolo	i
	-1	93	Sedie	i
	-1	94	Cassetto	i, a
10	-1	100	Porta_a_bilico (10-9)	i, A
	-1	101	Tavolino	i
	-1	102	Statuetta	i, l
	-1	103	Sedia	i
	-1	104	Quadro	i
	-1	105	Cassetto	i, a
	-1	106	Libreria	i

	-1	107	Portale (10-5)	i,A
11	-1	110	Porta (11-6)	i,A
	-1	111	Guardia	i,w
	-1	112	Chiave_palazzo	i,p
12	-1	120	Porta_saloon (12-120)	i,A
	-1	121	Portone (12-7)	i,A
13	-1	131	Statua	i,l
	-1	132	Pozzo	i
	-1	133	Passaggio_fune (13-130)	i,s
	-1	134	Porta (13-8)	i,A
14	-1	140	Portone (14-9)	i,A
	-1	141	Porta (14-19)	i,A
	-1	142	Tavolo	i
	-1	143	Sedie	i
	-1	144	Caffe	i,m
15	-1	150	Re	i,w
	-1	151	Corona	i,p
	-1	152	Portone (15-20)	i,A
17	-1	170	Cipresso (17-70)	i,S
18	-1	180	Carovana	i,a
19	-1	190	Porta (19-14)	i,A
	-1	191	Bancone	i
	-1	192	Cassa	i,a
20	-1	200	Scalino	i,S
	-1	201	Portone (20-15)	i,A
	-1	202	Teca	i,a
130	-1	1300	Passaggio_fune (130-13)	i,S
	-1	1301	Chiave_grata	i,p
120	-1	1200	Porta_saloon (120-12)	i,A
	-1	1201	Cavalli	i
	-1	1202	Guardiano	i,w
	-1	1203	Fune	i,p,u
	-1	1204	Chiave_porta_saloon	i,p
170	-1	1700	Cipresso (170-17)	i,s
	-1	1701	Baccarance	i,p,m
70	-1	700	Grata (70-7)	i,A
	-1	701	Scala (70-1)	i,s
50	-1	500	Scala (50-5)	i,s
	-1	501	Scalino	i,s
	-1	502	Guardia	i,w
	-1	503	Chiave_dorata	i,p
-1	13	130	Ciondolo	i,p,u
-1	84	840	Piuma	i,p,u
-1	94	940	Mappa_palazzo	i,p,l
-1	94	941	Medaglie	i,p
-1	94	942	Chiave_biblioteca	i,p
-1	105	1050	Diamante_grezzo	i,p,u
-1	180	1800	Mela	i,p,m
-1	192	1920	Mandorle	i,p,m
-1	202	2020	Spada	i,p,e
-1	202	2021	Ascia	i,p,e
-1	202	2022	Pugnale	i,p,e

Vincoli legati all'operazione "scendere":

(8) Se si scende da "Camino", si perde una vita.

Vincoli legati all'operazione "prendere":

(4) Se si prende "Gufo Reale", si perde una vita.

(15) Se si prende "Corona" (visibile dopo aver ucciso "Re") si vince il gioco.

Vincoli legati all'operazione "usare":

(2) "Candelabro (spento)" si utilizza dopo aver ispezionato "Fuocherello".

In tal caso diventa "Candelabro (acceso)"

(10a) Dopo aver ispezionato "Libreria", si deve utilizzare l'oggetto "Piuma di Gufo".

Se si utilizzano altri oggetti, si perde una vita.

(10b) "Portale" non visibile fino a quando non si ispeziona "Libreria" e si utilizza "Piuma di Gufo"

(13) "Passaggio fune" non visibile fino a che si ispeziona "Pozzo" e si utilizza "Fune"

Vincoli legati all'operazione "saltare":

(170) Saltando dalla stanza 170, ci si trasporta in stanza 120

Vincoli legati all'operazione "aprire":

(11) "Chiave palazzo" e "candelabro (acceso)", permette di aprire "Porta (13-8)"

(120) Permette di aprire l'oggetto "Porta saloon (120-12)"

(130) "Chiave grata" permette di aprire "Grata (7-70)"

Dimensione inventario: illimitata, ovvero ha una capienza massima pari ad 80 e il numero di oggetti prendibili nel gioco è minore di 80

Oggetti nell'inventario del protagonista (a inizio del gioco): nessuno

Oggetti inutili: Tutti gli oggetti della Tabella Oggetti aventi come unico attributo 'i'

Caratteristiche Protagonista e Nemici:

ID	Nome	Salute	Forza	Salute Max	Forza Max	ID oggetti rilasciati
1	Aelwith Holte	20	5 (senza armi equipaggiate)	illimitato	15	
1202	Guardiano Scuderia	30	1			1203, 1204
111	Guardia Torre OVEST	40	8			112
502	Guardia Torre EST	40	10			503
150	Re	22	13			151

Caratteristiche armi:

ID	Nome	Bonus Salute	Bonus Forza
22	Falcastro	0	+4
2020	Spada	+12	+3
2021	Ascia	+10	+5
2022	Pugnale	+6	+6

Sistema di combattimento:

Nel gioco, il sistema di combattimento si avvia quando il protagonista si vuole scontrare con nemico, cioè quando esegue il comando 'uccidi'. Il funzionamento è descritto di seguito:

Input	Output
Status Protagonista (Salute1, Forza1)	Danno che il Protagonista subisce affrontando il nemico.
Status Nemico (Salute2, Forza2)	

- $0 \leq \text{danno} < \text{Salute1}$
Il protagonista uccide il Nemico ma i suoi punti Salute sono decrementati in quantità pari al danno
- $\text{danno} \geq \text{Salute1}$
Il protagonista è sconfitto dalla guardia. Dunque perde una vita e i suoi punti salute ritornano 20.

Il sistema di combattimento funziona nella maniera seguente: finché uno dei due avversari non muore si tira un dado da 1-15:

- ▽ se il valore del dado è minore o uguale a Forza1 il protagonista infligge al nemico almeno un danno pari a Forza1.
- ▽ se il valore del dado è compreso tra Forza1 e Forza2 si confrontano Forza1 e Forza2:
se $\text{Forza1} < \text{Forza2}$ il protagonista subisce un danno pari al valore del dado.
se $\text{Forza1} > \text{Forza2}$ il protagonista infligge un danno casuale compreso tra le due forze.
- ▽ se il valore del dado è maggiore di Forza2 si confrontano Forza 1 e Forza2:
se $\text{Forza1} < \text{Forza2}$ il protagonista subisce un danno pari alla metà del dado o di Forza2
se $\text{Forza1} > \text{Forza2}$ uno a caso tra il protagonista e il nemico subiscono un danno pari a 1.

Sequenza azioni per finire il gioco:

Il giocatore inizia in stanza 18.

Apri carovana, prendi mela, vai nord, vai ovest, vai nord, vai nord, prendi falcastro, equipaggia falcastro, prendi candelabro, vai sud, vai sud, vai sud, sali quercia, prendi ghiande, salta, uccidi guardiano, mangia mela, prendi fune, prendi chiave porta saloon, apri porta saloon, vai est, ispeziona pozzo, usa fune, scendi passaggio fune, prendi chiave grata, sali passaggio fune

Vai ovest, vai nord, apri grata, scendi scala, sali chiocciola, ispeziona fuocherello, usa candelabro, vai sud, uccidi guardia, mangia ghiande, prendi chiave palazzo, vai nord, scendi chiocciola, sali scala, apri grata, vai sud, vai est, apri porta.

Apri teca, prendi piuma, vai est, apri cassetto, prendi mappa palazzo, prendi chiave biblioteca, leggi mappa palazzo, apri porta a bilico, ispeziona libreria, usa piuma, apri portale, sali scala, uccidi guardia, prendi chiave dorata, scendi scalino, apri teca, prendi pugnale, equipaggia pugnale, vai nord, uccidi re, prendi corona.

Comandi spostamenti**1 – vai + input**

Input	Output
nord/est/sud/ovest/porta	esito (il protagonista può/non può andare nella direzione scelta)
inventario del protagonista	nuova stanza dove si trova il protagonista (1.1)

(1.1) Se lo spostamento del protagonista avviene con successo nella nuova stanza, avvia in automatico il comando “guarda”

2 – sali + input

Input	Output
oggetto nella stanza corrente con attributo ‘S’	nuova stanza dove si trova il protagonista

3 – scendi + input

Input	Output
oggetto nella stanza corrente con attributo ‘s’	nuova stanza dove si trova il protagonista

Comandi generali**1 – guarda / guarda intorno**

Input	Output
stanza dove si trova il protagonista	descrizione narrata della stanza
	lista di oggetti presenti e contenuti nella stanza
	direzioni possibili

2 – ispeziona + input (può essere usato in coppia con il comando ‘usa’)

Input	Output
oggetto nella stanza corrente con attributo ‘i’	Descrizione narrata dell’oggetto

3 – prendi + input

Input	Output
oggetto nella stanza corrente con attributo ‘p’ oppure oggetto con attributo ‘p’ contenuto nella stanza	spostamento dell’oggetto di input nell’inventario del protagonista

4 – apri + input

Input	Output
oggetto nella stanza corrente con attributo ‘a’/‘A’	Se l’input ha attributo ‘a’: Lista oggetti contenuti nell’oggetto input Se l’input ha attributo ‘A’: Stesso output del comando ‘vai’
inventario del protagonista	

(2.4a) Un oggetto con attributo ‘a’ è un oggetto che può essere aperto e in tal caso contiene altri oggetti.

Un oggetto con attributo ‘A’ è un oggetto che può essere aperto e in tal caso il protagonista si sposta oltre l’oggetto senza che sia necessario usare il comando ‘vai’.

Ad es. il comando ‘apri porta’ è equivalente al comando ‘vai nord’ se la porta collega la stanza

corrente con una stanza a nord.

(2.4b) Il comando 'apri + input' dove input ha attributo 'A' controlla se esiste un oggetto nell'inventario (ad es. una chiave particolare) che consente di aprire (varcare) l'oggetto di input.

Comandi particolari

1 – usa + input2

Può essere utilizzato solo se il comando precedente è "ispeziona + input1"

Input	Output
input2: oggetto nella stanza corrente con attributo 'u'	Descrizione narrata dell'azione
	Output che dipende da input1 e input2 (vedi elenco puntato)

(3.1) Per una esperienza di gioco meno macchinosa le chiavi non vengono usate, vedi nota (2.4b)

- Se l'input1 è 'Fuocherello' e input2 è 'Candelabro (spento)': vedi vincolo (2)
- Se l'input1 è 'Libreria' e input2 è 'Piuma di Gufo Reale': vedi vincolo (10b)
- Se l'input1 è 'Libreria' e input2 è qualsiasi altro oggetto: vedi vincolo (10a)
- Se input1 è 'Pozzo' e input2 è 'Fune': vedi vincolo (13)

2 – mangia + input

Input	Output
oggetto nell'inventario del protagonista con attributo 'm' <i>oppure</i> oggetto nella stanza corrente con attributo 'm'	Cambiamento dello status Salute del protagonista

3 – equipaggia + input

Input	Output
oggetto nell'inventario del protagonista con attributo 'e'	Cambiamento dello status Forza del protagonista

(3.3) Si può equipaggiare solo un'arma alla volta.

4 – leggi + input

Input	Output
oggetto nell'inventario del protagonista con attributo 'l' <i>oppure</i> oggetto nella stanza corrente con attributo 'l'	Messaggio scritto sull'input

5 – uccidi + input

Input	Output
oggetto nella stanza corrente con attributo 'w'	Cambiamento dello status Salute del Protagonista
	Si rendono visibili gli oggetti che il nemico rilascia nella stanza corrente

6 – salta

Input	Output
id della stanza corrente del giocatore	nuova stanza dove si trova il protagonista

(3.6) Si può saltare solo nella stanza 170.

Comandi per salvare/caricare partita e comando di aiuto

1 – salva partita

Input	Output
nome del giocatore	File (unico per giocatore) che contiene: numero vite corrente del protagonista, id stanza corrente del protagonista, struttura dati del protagonista, inventario, id arma equipaggiata, mappa, stanze della mappa

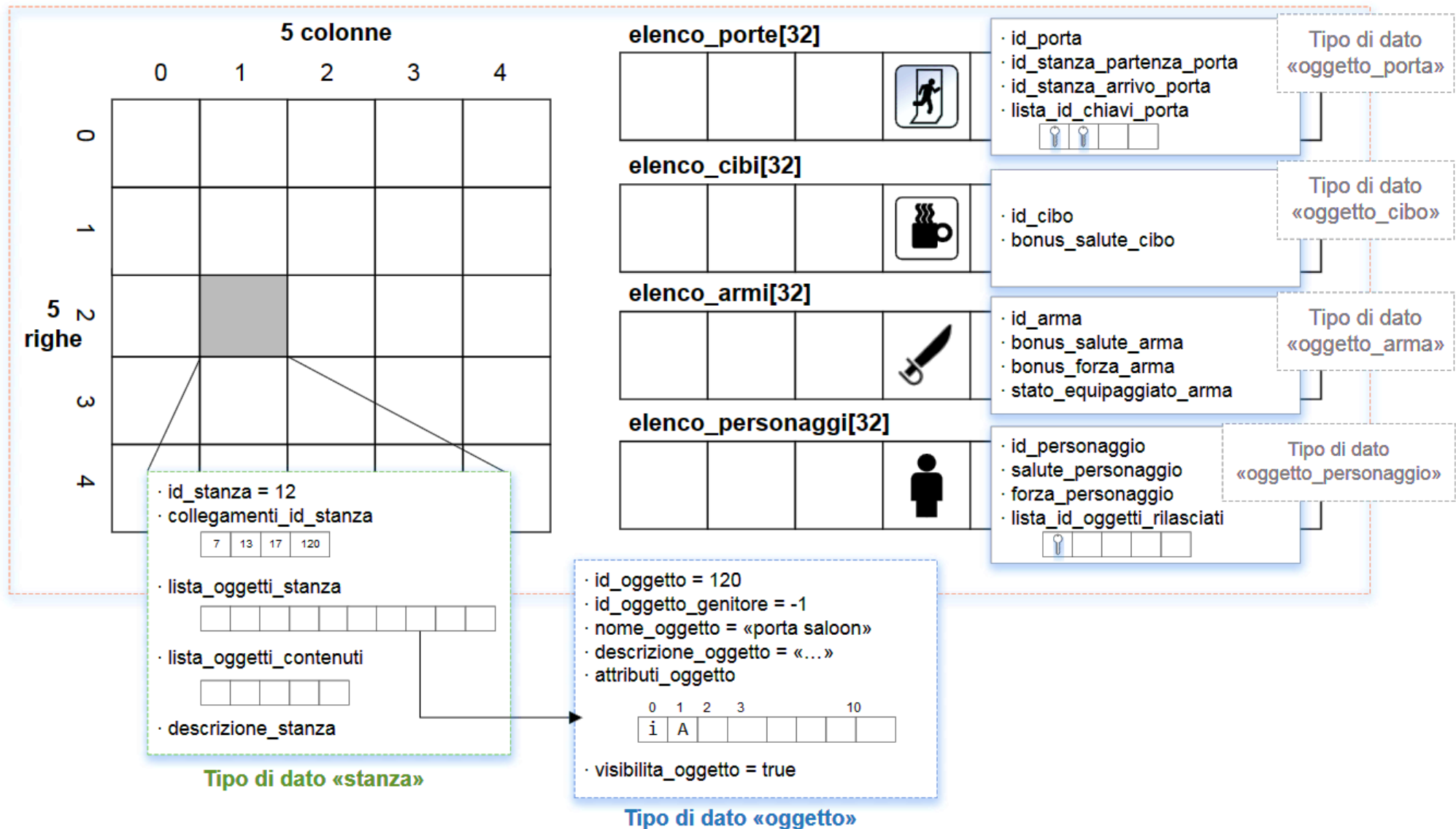
2 – carica partita

Input	Output
nome del giocatore	Lettura da file (unico per giocatore) di: numero vite corrente del protagonista, id stanza corrente del protagonista, struttura dati del protagonista, inventario, id arma equipaggiata, mappa, stanze della mappa

3 – aiuto

Input	Output
	Guida del gioco

Tipo di dato «mappa»



Tipo di dato “mappa”

Nome	Descrizione	Tipo	Vincoli
elenco_stanze	Tabella di stanze, ciascuna delle quali avente un numero di riga e di colonna	array 2 dimensioni di stanza (statico)	
righe	Numero di righe della mappa	intero	>0
col	Numero di colonne della mappa	intero	>0
elencoporte	Elenco di tutte le porte della mappa	array 1 dimensione di oggetto_porta (statico)	DIM=32
elenco_cibi	Elenco di tutte i cibi sparsi nella mappa	array 1 dimensione di oggetto_cibo (statico)	DIM=32
elenco_armi	Elenco di tutte le armi sparse nella mappa	array 1 dimensione di oggetto_arma (statico)	DIM=32
elenco_personaggi	Elenco di tutte i personaggi sparsi nella mappa	array 1 dimensione di oggetto_personaggio (statico)	DIM=32

Tipo di dato “stanza”

Nome	Descrizione	Tipo	Vincoli
id_stanza	ID univoco della stanza	intero	univoco
collegamenti_id_stanza	Elenco di ID delle stanze direttamente collegate alla stanza corrente	array 1 dimensione di interi (statico)	DIM=4, devono essere ID di stanze esistenti
lista_oggetti_stanza	Elenco degli oggetti della stanza corrente	array 1 dimensione di oggetti (statico)	DIM=10
lista_oggetti_contenuti	Elenco degli oggetti contenuti in qualsiasi oggetto apribile nella stanza corrente	array 1 dimensione di oggetti (statico)	DIM=5
descrizione_stanza	Descrizione narrata della stanza	array 1 dimensione di caratteri (statico)	DIM=350

Tipo di dato “oggetto”

Nome	Descrizione	Tipo	Vincoli
id_oggetto	ID univoco dell'oggetto	intero	univoco, =-1 se l'oggetto non è valido
id_oggetto_genitore	ID dell'oggetto che contiene l'oggetto in questione	intero	=-1 se non è contenuto in nessun oggetto
nome_oggetto	Nome dell'oggetto	array 1 dimensione di caratteri (statico)	DIM=80
descrizione_oggetto	Descrizione narrata dell'oggetto	array 1 dimensione di caratteri (statico)	DIM=350
attributi_oggetto	Elenco degli attributi dell'oggetto	array 1 dimensione di caratteri (statico)	DIM=11
visibilita_oggetto	Flag che indica se l'oggetto è visibile o meno	booleano	vero=visibile falso=no visibile

Tipo di dato “oggetto_porta”

Nome	Descrizione	Tipo	Vincoli
id_porta	ID univoco della porta nella mappa	intero	univoco
id_stanza_partenza_porta	ID della stanza di partenza della porta	intero	deve essere un ID di una stanza esistente
id_stanza_arrivo_porta	ID della stanza dove la porta conduce	intero	deve essere un ID di una stanza esistente
lista_id_chiavi_porta	ID degli oggetti che aprono la porta	intero	DIM=4, devono essere ID di oggetti esistenti

Tipo di dato “oggetto_cibo”

Nome	Descrizione	Tipo	Vincoli
id_cibo	ID univoco del cibo	intero	univoco
bonus_salute_cibo	Numero di punti salute che si vanno ad aggiungere ai punti salute correnti	intero	>0

Tipo di dato “oggetto_arma”

Nome	Descrizione	Tipo	Vincoli
id_arma	ID univoco dell'arma	intero	univoco
bonus_salute_arma	Numero di punti salute che si vanno ad aggiungere ai punti salute correnti	intero	>=0
bonus_forza_arma	Numero di punti forza che si vanno ad aggiungere ai punti salute correnti	intero	>0
stato_equipaggiato_arma	Flag che indica se l'arma è equipaggiata al protagonista	booleano	vero=equipaggiato falso=no equipaggiato

Tipo di dato “oggetto_personaggio”

Nome	Descrizione	Tipo	Vincoli
id_personaggio	ID univoco del personaggio	intero	univoco
salute_personaggio	Numero di punti salute del personaggio	intero	>0
forza_personaggio	Numero di punti forza del personaggio	intero	compreso tra 0 e 15
lista_id_oggetti_rilasciati	Elenco di tutti gli ID degli oggetti resi visibili se il personaggio muore	array 1 dimensione di interi (statico)	DIM=5, devono essere ID di oggetti esistenti

Tipo di dato “comando”

Nome	Descrizione	Tipo	Vincoli
id_comando	ID del comando	intero	univoco, >=0
regex_verbo	Espressione regolare che descrive il linguaggio a cui devono appartenere i verbi corretti sintatticamente	array 1 dimensione caratteri (statico)	DIM=350
argomentooggetto	Flag che indica se dopo il verbo deve essere espresso il nome di un oggetto o una locuzione	booleano	vero=oggetto falso=locuzione
regex_argomento	Espressione regolare che descrive il linguaggio a cui devono appartenere gli argomenti corretti sintatticamente	array 1 dimensione caratteri (statico)	DIM=350
argomentooggetto_in_stanza	Flag che indica se l'argomento del comando deve essere un nome di un oggetto presente nella stanza	booleano	vero=nome oggetto nella stanza falso=non è un nome di un oggetto in stanza
argomentooggetto_in_inventario	Flag che indica se l'argomento del comando deve essere un nome di un oggetto presente nell'inventario	booleano	vero=nome oggetto nell'inventario falso=non è un nome di un oggetto in inventario

Due esempi di comandi e le relative interpretazioni:

1 - Comando “vai”

id_comando := 0

regex_verbo :=

^(andare)\$|^(vai)\$|^(recarsi)\$|^(recati)\$|^(proseguire)\$|^(proseguì)\$|^(visitare)\$|^(visita)\$

argomentooggetto := falso

regex_argomento := ^(nord)\$|^(est)\$|^(sud)\$|^(ovest)\$

argomentooggetto_in_stanza := falso

argomentooggetto_in_inventario := falso

Il **comando** vai è il primo comando riconosciuto dal gioco.

La suddetta **espressione regolare del verbo** sta a significare che il verbo può essere:

la stringa intera “andare” OPPURE la stringa intera “vai” OPPURE la stringa intera “recarsi” ecc...

NOTA: Questo significa che un comando avente verbo “testandaretest” NON è riconosciuto come comando “vai”.

Il comando vai non necessita un **argomento che è il nome di un oggetto**, pertanto l'argomento del comando deve essere una LOCUZIONE.

Le **locuzioni** possibili sono: “nord” OPPURE “est” OPPURE “sud” OPPURE “ovest”

L'argomento del comando “vai” non è un oggetto, quindi gli ultimi due attributi sono irrilevanti in questo caso.

2 – Comando “prendi”

id_comando := 5

regex_verbo :=

^(prendere)\$|^(prendi)\$|^(ottenere)\$|^(ottieni)\$|^(acquisire)\$|^(acquisisci)\$|^(prelevare)\$|^(preleva)\$|^(raccogliere)\$|^(raccogli)\$

argomento_oggetto := vero

regex_argomento := p

argomento_oggetto_in_stanza := vero

argomento_oggetto_in_inventario := falso

Il **comando** vai è il sesto comando riconosciuto dal gioco.

La suddetta **espressione regolare del verbo** sta a significare che il verbo può essere:

la stringa intera “prendere” OPPURE la stringa intera “prendi” ecc...

NOTA: Questo significa che un comando avente verbo “testprenditest” NON è riconosciuto come comando “prendere”.

Il comando vai necessita un **argomento che è il nome di un oggetto**.

L'argomento del comando “prendi” deve essere un nome di un oggetto che abbia ‘p’ tra i suoi attributi

L'argomento del comando “prendi” deve essere un nome di un oggetto presente nella stanza e non nell'inventario.

Casi di test avventura testuale**Gruppo 11****1 – leggere_nome_giocatore**

Nome	Descrizione	Tipo	Vincoli
nome_giocatore	Nome del giocatore che l'utente ha immesso	array 1 dimensione di caratteri (statico)	dimensione=10

Nome	Descrizione	Tipo	Vincoli
esito	flag che indica se nome_giocatore immesso dall'utente e' corretto	booleano	vero=corretto falso=errato

Input da testare	Output Funzione	Output desiderato?	Commento
<i>(stringa vuota)</i>	esito := falso	si	CORRETTO
►	esito := falso	si	CORRETTO
stringa con più di 10 caratteri	esito := falso	si	CORRETTO
test 12	esito := falso	si	CORRETTO
test	esito := vero	si	CORRETTO
1test2444	esito := vero	si	CORRETTO
gianni ciao	esito := falso	si	CORRETTO

2 – eseguire_scanner

Nome	Descrizione	Tipo	Vincoli
comando_utente	Comando immesso dall'utente	array 1 dimensione di caratteri (statico)	dimensione=64

Nome	Descrizione	Tipo	Vincoli
regex_match	flag che indica se una o più parole del linguaggio descritto dall'espressione regolare <code>[^a-zA-Z][^a-zA-Z]*</code> e' sottostringa del comando	booleano	vero=c'è un match falso= non c'è un match

Nome [DATI LAVORO]	Descrizione	Tipo	Vincoli
verbo_comando	Verbo estratto da comando_utente	array 1 dimensione di caratteri (statico)	dimensione=64
argomento_comando	Argomento estratto da comando_utente	array 1 dimensione di caratteri (statico)	dimensione=64

Poiche' la funzione `eseguire_scanner` modifica i dati di lavoro `verbo_comando` e `argomento_comando` che saranno l'input della funzione `eseguire_parser`, facciamo un controllo anche su quelli.

Input da testare	Output Funzione	Output desiderato?	Commento
1243	<code>regex_match := vero</code>	si	CORRETTO
!£%%	<code>regex_match := vero</code>	si	CORRETTO
<i>(stringa vuota)</i>	<code>regex_match := falso</code>	si	CORRETTO
guarda	<code>regex_match := falso</code> <code>verbo_comando := "guarda"</code> <code>argomento_comando := ""</code>	si	CORRETTO
APRI carovana	<code>regex_match := falso</code> <code>verbo_comando := "apri"</code> <code>argomento_comando := "carovana"</code>	si	CORRETTO
prendere MELA	<code>regex_match := falso</code> <code>verbo_comando := "prendere"</code> <code>argomento_comando := "mela"</code>	si	CORRETTO
PREN\ER\E M11111LA	<code>regex_match := vero</code>	si	CORRETTO
ucCIDERE tesT	<code>regex_match := falso</code> <code>verbo_comando := "uccidere"</code> <code>argomento_comando := "test"</code>	si	CORRETTO

3 – eseguire_parser

Nome	Descrizione	Tipo	Vincoli
verbo_comando	Verbo estratto dal comando immesso dall'utente	array 1 dimensione caratteri (statico)	dimensione=64
argomento_comando	Argomento estratto dal comando immesso dall'utente	array 1 dimensione caratteri (statico)	dimensione=64

Nome	Descrizione	Tipo	Vincoli
regex_match	flag che indica se una o più parole del linguaggio descritto dalle espressioni regolari memorizzate nella struttura dati array di tipo comando sono sottostringhe di verbo_comando	booleano	vero=c'è un match falso= non c'è un match

Input da testare verbo_comando argomento_comando	Output Funzione	Output desiderato?	Commento
<p><i>Esempio di stanza dove e' stata testata la funzione: stanza 18.</i></p> <p>Oggetti: carovana (attributi i,a), mela (attributi i,p) (presente negli oggetti contenuti della stanza)</p>			
"scendi"	regex_match := vero (significa che il comando "scendi" esiste)	si	CORRETTO
"stqlla"	<i>Non vedo nessun stqlla nella stanza</i>		
"prendi"	regex_match := vero (significa che il comando "prendi" esiste)	si	CORRETTO
"carovana"	<i>Non posso fare questa azione su carovana</i>		
"scendi"	regex_match := vero (significa che il comando "scendi" esiste)	si	CORRETTO
"carovana"	<i>Non posso fare questa azione su carovana</i>		
"soffoca"	regex_match := vero (significa che il comando "soffoca" esiste)	si	CORRETTO
"carovana"	<i>Non posso fare questa azione su carovana</i>		
"scendi"	regex_match := vero (significa che il comando "scendi" esiste)	si	CORRETTO
" "	<i>Su quale oggetto devo fare l'azione?</i>		
"blalblal"	regex_match := falso (significa che il comando "blalblal" non esiste)	si	CORRETTO
"test test g "	<i>Non capisco cosa vuoi fare</i>		
"equipaggia"	regex_match := vero (significa che il comando "equipaggia" esiste)	si	CORRETTO
"carro armato"	<i>Non vedo nessun carro armato nell'inventario</i>		

"salta"	regex_match := vero (significa che il comando "salta" esiste)	si	CORRETTO
""	Si esegue la funzione eseguire_salta		
"salta"	regex_match := vero (significa che il comando "salta" esiste)	si	CORRETTO
"bla bla"	Devo solo saltare. Digita 'salta'		
PRIMO CASO DI TEST SUL COMANDO "prendi mela": Prima di eseguire il parser con input "prendi" e "mela", si esegue il parser con input "apri" e "carovana"			
"apri"	regex_match := vero (significa che il comando "apri" esiste)	si	CORRETTO
"carovana"	Si rendono visibili gli oggetti contenuti della stanza.		
"prendi"	regex_match := vero (significa che il comando "prendi" esiste)	si	CORRETTO
"mela"	Preso		
SECONDO CASO DI TEST SUL COMANDO "prendi mela": Si esegue il parser con input "prendi" e "mela" senza aver prima eseguito il parser con input "apri" e "carovana"			
"prendi"	regex_match := vero (significa che il comando "prendi" esiste)	si	CORRETTO
"mela"	Non vedo nessun mela nella stanza		

4 – avviare_sistema_combattimento

Nome	Descrizione	Tipo	Vincoli
protagonista	Protagonista del gioco che deve battersi contro il nemico	oggetto_personaggio	deve avere gli attributi salute e forza >0
oggetto_nemico	Nemico del protagonista	oggetto_personaggio	deve avere gli attributi salute e forza >0

Nome	Descrizione	Tipo	Vincoli
danno	Danno che protagonista subisce battendosi contro il nemico	intero	>= salute protagonista significa che protagonista perde, vince altrimenti

Poiche' la funzione avviare_sistema_combattimento basa il suo funzionamento su numeri casuali, nella colonna "Output Funzione" sono mostrati piu' output (separati dalla virgola)

Input da testare 1^ riga: salute/forza Protagonista 2^ riga: salute/forza Nemico	Output Funzione	Output desiderato?	Commento
28,9	9,9,7,11,6,10	si	CORRETTO
40,8			
20,5	4,8,8,8,7,6,3,8,7	si	CORRETTO
30,1			
30,5	9,9,10,6,5,8,7,6,7,5,9	si	CORRETTO
30,1			
26,9	9,7,7,8,9,9,8,6,8	si	CORRETTO
40,8			
10,2	16,12,11,14,11,12	si	CORRETTO
40,8			
10,15	13,3,10,9,11,9,8	si	CORRETTO
40,8			