

SISTEMA EDITOR DE BOVINO
Renato Degiovani

sistema
EDITOR
RENATO DEGIOVANI

SISTEMA EDITOR DE ADVENTURES
Renato Degiovani

Para uma correta compreensão da funcionalidade do sistema EDITOR é necessário ler este livro em conjunto com o livro deste manual. Em seguida o usuário deverá implementar e instalar o sistema que proporciona esse sistema além de ler os conceitos de construção de um programa em uma linguagem clara.

Este sistema foi desenvolvido para criação e editoração de programas adventures.

A versão para IBM PC, do sistema EDITOR, é compatível com o hardware Microsoft de 16 bits. Versão 3.4 - 1986. Este livro contém informações que ocorrem durante a sua instalação e execução para o usuário.

Programação: Renato Degiovani
Produção: PRO KIT software
Distribuição: PRO KIT software

Agradecimentos:

Cláudio Costa e José Eduardo Neves

<<< TODOS OS DIREITOS RESERVADOS >>>

ESTE MANUAL :

Para uma perfeita compreensão da funcionalidade do sistema EDITOR é necessário que se faça uma leitura completa deste manual. Em seguida o usuário deverá implementar e analisar o exemplo que acompanha esse sistema afim de que os conceitos de construção de um adventure fiquem mais claros.

A versão para MSX disco, do sistema EDITOR, é compatível com o padrão Microsoft de interface de drive. Qualquer dúvida operacional que ocorra durante a sua operação escreva para:

RENATO DEGIOVANI - Coluna ADVENTURES
Av. Pres. Wilson, 165 gr 1210
Rio de Janeiro - RJ
CEP 20030

ÍNDICE:

1. INTRODUÇÃO	6
1.1 A funcionalidade do jogo	6
1.2 Procedimento de carga	7
1.3 Inicialização do EDITOR	7
1.4 Organização do sistema	8
1.5 Elementos estruturais do jogo	8
1.6 A funcionalidade do EDITOR	9
1.7 Tabela dos comandos do EDITOR	9
1.8 Edição de alfabetos	11
2. VÍDEO E EDITORAÇÃO	12
2.1 Campos funcionais	12
2.2 Editor de tela	13
2.3 Comandos de edição de tela	14
3. REGISTRADORES	15
3.1 Registradores especiais	15
3.2 Registradores pré-definidos	15
4. VERBOS	17
4.1 Divisão funcional dos verbos	18
5. OBJETOS	19
5.1 Objetos pré-definidos	19
5.2 Situação de um objeto	19
5.3 Byte de consistência	20
6. POSIÇÕES	22
6.1 Movimento entre posições	22
7. COMANDOS	23

7.1 Comandos já definidos	23
8. INSTRUÇÕES DO SISTEMA	24
8.1 Edição das instruções	24
8.2 Tabela de instruções	25
9. MENSAGENS	27
9.1 Mensagens especiais	27
9.2 Mensagens pré-definidas	28
9.3 A estrutura de uma mensagem	29
10. FUNÇÕES	30
10.1 Funções especiais	30
11. UM EXEMPLO PRÁTICO	34
11.1 Criando um mapa	34
11.2 A inicialização	36
11.3 As passagens do jogo	37
11.4 Criando Objetos	42
11.5 Projetando a saída	48
11.6 Conclusão	49
12. INFORMAÇÕES COMPLEMENTARES	51

1. INTRODUÇÃO

O sistema EDITOR foi concebido para monitorar a criação e edição de programas aventureiros. A sua potencialidade está diretamente vinculada ao conhecimento dos diversos elementos que o compõem.

Todo o trabalho de programação, ou de uso de uma linguagem de computação, foi minimizado afim de que o autor possa usar seu esforço unicamente na elaboração funcional do jogo. No entanto, as práticas e noções elementares de programação são úteis quando da criação e manipulação dos COMANDOS e FUNÇÕES do sistema.

1.1 A FUNCIONALIDADE DO JOGO

O adventure é um jogo no qual o jogador é sempre o agente criador e idealizador das ações. O computador é passivo e apenas responde a um enredo previamente estruturado pelo autor do jogo.

A forma de interagir com o jogo é através de frases comando que expressam o desejo do jogador, ou seja, se o jogador deseja pegar um objeto, uma caneta por exemplo, ele simplesmente induz o computador a essa ação, na forma PEGUE A CANETA. O computador se restringe a executar a ordem e a fornecer um relatório do seu desempenho. O computador também se encarrega de orientar o jogador em relação ao local onde ele se encontra, descrevendo-o e manipulando os movimentos através de comandos do tipo VÁ PARA O NORTE, por exemplo.

Os adventures criados pelo EDITOR permitem não só uma diversidade de ações, mas também de estruturas de construção das frases. Em função das letras fornecidas pelo jogador, o sistema procura no seu banco um conjunto que satisfaça a intenção da frase. Por exemplo, basta digitar EX que o sistema compreende a palavra EXAMINE, ou apenas LOC, que é interpretado como LOCAL.

O sistema possui também formas de simplificação para a abreviação dos comandos mais usados, principalmente as movimentações, com a substituição das frases do tipo VÁ PARA O NORTE por uma composição mais simples: NORTE ou apenas N. A tecla ENTER, pressionada sem nenhuma frase para interpretação, assume a função de fazer uma descrição sumária dos objetos que estão no local onde o jogador se encontra.

O jogo permite também a referência indireta ao objeto, ou seja, a construção PEGUE A CANETA pode ser estruturada também como PEGUE-A, desde que o objeto "caneta" tenha sido referenciado anteriormente.

A versão 3.4 trabalha exclusivamente com textos e produz adventures no estilo do AMAZÔ-

NIA e SERRA PELADA.

O uso da acentuação obedece aos procedimentos normais de cada versão de computador. Uma vez que o modelo EXPERT 1.0, da Gradiente, não possui todas as letras acentuadas e nem permite a acentuação correta via teclado, é aconselhável definir as teclas SHIFT NÚMERO como letras acentuadas. Veja maiores detalhes no item sobre edição de alfabetos (1.8).

1.2 PROCEDIMENTO DE CARGA

O procedimento de carga (LOAD) do sistema EDITOR é bastante simples, porém deve ser executado exatamente como descrito abaixo:

- desligue o computador e aguarde alguns segundos para que os capacitores internos descarreguem.
- coloque o disco EDITOR no drive A.
- ligue o micro e feche a portinhola do drive.

1.3 INICIALIZAÇÃO DO EDITOR

O sistema EDITOR é carregado automaticamente quando se reseta o computador. A primeira intervenção do operador diz respeito à pergunta "Editar (S/N)".

Neste momento o usuário deve optar entre editar um jogo que já existe ou criar um novo jogo. O disco EDITOR possui um jogo exemplo chamado MANSÃO, que pode ser acessado neste momento, pressionando-se a tecla "S" e a tecla ENTER a seguir..

Caso se deseje editar um jogo, o sistema solicitará a presença, no drive, de um disco que contenha tal jogo. Caso a tecla pressionada seja "N", o sistema solicitará um nome para o jogo a ser criado e, logo após, solicitará a presença de um disco previamente formatado.

Em ambos os casos o sistema trabalhará com uma organização própria para o disco. Tal organização não responde a comandos do tipo DIR, FILES, COPY, etc.

Quando em edição, é dispensável a presença do disco EDITOR no drive, com excessão da ocasião em que for solicitada a carga de um conjunto novo de letras (veja tópico sobre edição de alfabetos).

Não é aconselhável, durante a criação de um jogo, usar o disco EDITOR como arquivo de trabalho. Tanto o EDITOR quanto um jogo por ele criado utilizam apenas um drive.

Após a operação de carga, ou criação, de um jogo, será apresentada ao operador uma tela com um balanço da situação dos elementos das tabelas, bem como o espaço de memória utilizado e o espaço disponível.

1.4 ORGANIZAÇÃO DO SISTEMA

O sistema EDITOR compõe-se de um módulo de jogo, que é o jogo propriamente dito, um módulo de edição e das tabelas dos elementos funcionais do jogo. Quando em edição, estando no jogo, o retorno ao EDITOR é feito mediante o acionamento da tecla ESC.

A cópia do jogo, gerada pelo EDITOR, pode ser reeditada apenas se ambos possuírem uma versão compatível. A gravação do jogo, efetuada pelo comando CJ, não carrega consigo o módulo EDITOR.

1.5 ELEMENTOS ESTRUTURAIS DO JOGO

Um jogo criado pelo EDITOR é composto por uma série de elementos organizados em tabelas. São eles:

- REGISTRADORES** são variáveis de um byte cuja função é permitir o registro de estados diversos. Eles contêm informações sobre as condições de determinados objetos ou estão associados a alguma circunstância relativa a uma POSIÇÃO. Por exemplo, o registrador 9 indica se o local onde o jogador se encontra está claro ou escuro.
- Eles podem servir também como contadores ou temporizadores e se assemelham, na sua aplicação, às variáveis de um programa BASIC.
- VERBOS** são os verbos reconhecidos pelo sistema e compõem sempre a primeira palavra da frase comando do jogador.
- Eles são reponsáveis pela definição da ação pretendida, pelo jogador, durante o jogo.
- OBJETOS** são os elementos, ou apenas palavras, que podem ser manipulados, ou referenciados, pelo jogador. Por exemplo, vela, mala, chão, árvore, etc.
- POSIÇÕES** são os locais onde é possível a presença física do jogador.
- COMANDOS** são séries de procedimentos, ou instruções de programação, que configuram a sistemática operacional de uma determinada frase do jogador. Por exemplo, para a frase PEGUE A CANETA existe um COMANDO que permitirá ao sistema "pegar a caneta".
- MENSAGENS** são as mensagens emitidas pelo sistema para se comunicar com o jogador, dando o resultado de uma determinada frase comando.
- FUNÇÕES** são séries de procedimentos, semelhantes aos COMANDOS, cuja requisição é determinada por um acontecimento ou situação especial. Podem estar associadas à passagem de um local para outro ou podem ser acionadas por um COMANDO qualquer.
- Funcionam também como subrotinas de um comando.

1.6 A FUNCIONALIDADE DO EDITOR

O sistema operacional do EDITOR se baseia no uso direto de comandos, acompanhados ou não de parâmetros, constituídos na forma abreviada da ação. Por exemplo, para Listar as Mensagens basta executar o comando LM (ENTER). Para Listar a Mensagem número 5 basta executar LM 5 (ENTER).

Quando o EDITOR está em processo de criação ou edição de um determinado elemento (MENSAGENS, POSIÇÕES ou VERBOS) a tecla ESC fica desativada não permitindo, dessa forma, que o comando seja cancelado. Essa restrição não se aplica às FUNÇÕES e nem aos COMANDOS.

Toda a funcionalidade do sistema do jogo está baseada na indexação dos elementos. Assim, para qualquer referência basta citar o valor correspondente, ou seja, OBJETO 3, POSIÇÃO 5, MENSAGEM 10, FUNÇÃO 18, REGISTRADOR 17, VERBO 9, etc. Os COMANDOS não são indexados e são acessados pela sua construção efetiva, por exemplo: PEGUE A CANETA.

Uma vez criado um elemento, ele não mais poderá ser deletado do jogo. No caso de não ser preciso a sua presença, basta deixá-lo com um valor qualquer.

1.7 TABELA DOS COMANDOS DO EDITOR

Os comandos de listagem funcionam mediante o uso das setas, de acordo com a tabela abaixo:

SETA p/ BAIXO - avança um elemento.
SETA p/ CIMA - retrocede um elemento.
SETA p/ DIREITA - edita o elemento.
ESC - encerra o comando.

MNEMÔNICO

FUNÇÃO

LR X

Listar Registrador - lista o registrador X. Se X=0 ou não for dado X a listagem será integral onde os registradores são listados um a um.

MR X, Y

Mudar Registrador - modifica o valor do registrador X para Y. Ambos valores devem estar numa faixa aceitável ($0 < X < 200$ e $0 < Y < 256$).

LV X

Listar Verbo - lista o verbo X. Se X=0 ou não for dado X a listagem será integral onde os verbos são listados um a um.

CV

Criar Verbo - cria mais um verbo na tabela de definição dos verbos e atribui-lhe o último índice.

EV X

Editar Verbo - permite a edição do verbo X se X estiver dentro da faixa de verbos criados.

LP X	Listar Posição - lista a posição X. Se X=0 ou não for dado X a listagem será integral onde as posições são listadas uma a uma.
CP	Criar Posição - cria uma posição na tabela de definição das posições e atribui-lhe o último índice.
EP X	Editar Posição - permite a edição da posição X se X estiver dentro da faixa de posições criadas.
CC	Criar Comando - cria um comando na tabela de definição dos comandos. A identificação do comando será solicitada a seguir.
EC X	Editar Comando - permite a edição de um comando existente.
LM X	Listar Mensagem - lista a mensagem X. Se X=0 ou não for dado X a listagem será integral onde as mensagens são listadas uma a uma.
CM	Criar Mensagem - cria uma mensagem na tabela de definição das mensagens e atribui-lhe o último índice.
EM X	Editar Mensagem - permite a edição da mensagem X se X estiver dentro da faixa de mensagens criadas.
CF	Criar Função - cria uma função na tabela de definição das funções e atribui-lhe o último índice.
EF X	Editar Função - permite a edição da função X se X estiver dentro da faixa das funções criadas.
LO X	Listar Objeto - lista o objeto X. Se X=0 ou não for dado X a listagem será integral onde os objetos são listados um a um.
CO	Criar Objeto - cria um objeto na tabela de definição dos objetos e atribui-lhe o último índice.
EO X	Editar Objeto - permite a edição do objeto X se X estiver dentro da faixa de objetos criados.
CJ	Copiar Jogo - grava uma cópia do jogo que está em edição.
TJ	Testar Jogo - permite o teste do jogo em edição desde a sua fase inicial.

RJ	Retornar Jogo - retorna ao jogo em edição no ponto em que houve uma interrupção para edição.
PO X,Y	Pokear - modifica o endereço de memória X com o valor Y.
EA	Editar Alfabeto - permite a edição do alfabeto em uso ou a carga de um alfabeto arquivado no disco EDITOR.

1.8 EDIÇÃO DE ALFABETOS

A edição/criação de alfabetos permite que se use um desenho de letras mais adequado ao tema do adventure. Ao ser dado o comando EA, será apresentado ao usuário todo o set de caracteres utilizados pelo EDITOR e pelo jogo.

Um quadro em destaque ampliado indicará o desenho do caracter sob o cursor e os comandos disponíveis nesta etapa são:

SETAS	- movimentam o cursos pelo set de caracteres.
CTRL S	- salva, no disco EDITOR, o set de caracteres.
CTRL L	- carrega, do disco EDITOR, um set de caracteres.
CTRL R	- anula as alteração efetuadas e recupera o alfabeto original.
ESC	- cancela o comando EA e retorna ao EDITOR sem alterações.
SELECT	- aceita as alterações e retorna ao EDITOR.
ENTER	- edita o caracter.

Os comandos de edição do caracter são:

@@ tecla	- copia o caracter correspondente à tecla.
ESC	- cancela a edição do caracter sem alterações.
CTRL @	- inverte o caracter.
CTRL L	- limpa o caracter.
CTRL I	- espelha o caracter.
SETAS	- movem o cursor pixel.
ESPAÇO	- muda o estado do pixel sob o cursor.
ENTER	- aceita as modificações e retorna ao set de caracteres.

O disco EDITOR possui três buffers de alfabetos já definidos. O programa GRAPHOS III pode fornecer alfabetos para o jogo criado pelo EDITOR. Consulte o manual do GRAPHOS para maiores informações.

2. VÍDEO E EDITORAÇÃO

O sistema EDITOR é baseado na utilização do vídeo como processo interativo de jogo e edição. Nesse sentido a tela é sempre dividida em três campos principais: superior, central e inferior, divididos por duas barras horizontais.

2.1 CAMPOS FUNCIONAIS

Durante o jogo os campos assumem as seguintes características:

CAMPO INFERIOR - recebe a frase do jogador e a analisa enquanto comando. O cursor pode se deslocar até a última posição da linha, quando então retorna à sua primeira posição.

CAMPO SUPERIOR - apresenta a frase do jogador após a mesma ter sido interpretada. Todos os excessos, ou palavras não identificadas, são eliminados e o sistema passa a processar o comando.

CAMPO CENTRAL - apresenta o resultado, em forma de mensagem, do comando identificado na frase do jogador.

A figura abaixo ilustra essa divisão:

JEXAMINE LOCAL

=====

Neste local tem:

uma vela
uma mala

=====

IC

Durante a edição os campos assumem as seguintes características:

CAMPO SUPERIOR - apresenta o nome e a versão do sistema.

CAMPO INFERIOR - opera a entrada dos comandos do editor. Sua funcionalidade é igual à do jogo.

CAMPO CENTRAL - apresenta um balanço da memória, as listagens dos elementos e permite a edição das mensagens e frases diretamente na sua posição funcional. Além disso esse campo é usado para a edição dos COMANDOS e FUNÇÕES.

A figura abaixo ilustra essa divisão:

JSISTEMA EDITOR DE ADVENTURES - Versão 3.4 - Fev 86C

=====

Verbos: 34
Objetos: 3
Posições: 1
Comandos: 3
Mensagens: 19
Funções: 20

Nome: TESTE/CMD Tamanho: 12344 área livre: 20641

=====

JC

2.2 EDITOR DE TELA

Todas as operações que apresentam como resultado a impressão de uma mensagem, ou frase, acessam uma função de edição de tela interna ao sistema EDITOR.

Para a edição/criação das MENSAGENS, ou das frases de descrição das posições, o sistema apresenta um cursor não destrutivo e a posição padrão inicial de impressão (caracter "). Essa posição é assumida automaticamente, porém pode ser modificada mediante o deslocamento do caracter ")" para o local desejado. Durante a varredura de incorporação da mensagem, ao jogo, o sistema procurará um caracter ")" a partir do canto superior esquerdo do vídeo. Se este não for encontrado, então o sistema volta a assumir a posição padrão.

Toda a tela pode ser acessada pelo operador, porém é preciso manter as mensagens dentro do campo central a fim de evitar problemas durante o jogo. é preciso ter cuidado também com as mudanças de posição de impressão das mensagens, pois a rotina de CLS do jogo limpa a tela apenas da posição padrão ")" até o primeiro caracter antes da linha divisória do campo inferior.

2.3 COMANDOS DE EDIÇÃO DE TELA

Os comandos para edição de telas obedecem a um padrão funcional segundo a tabela abaixo:

INS	Inserir um espaço na posição do cursor. A tela, até a última posição do campo central, é deslocada uma posição.
DEL	Deleta um carácter na posição do cursor. A tela, até a última posição do campo central, é deslocada e na última posição é inserido um espaço.
=	Inserir um carácter indicativo de Line Feed.
@	Inserir um carácter especial indicativo de CLS.
ENTER	Finaliza a operação de edição da tela e produz a incorporação da mensagem na tabela correspondente. Para efeito de compilação, a mensagem válida inicia no primeiro carácter ">" até a posição atual do cursor.
HOME	Coloca o cursor no topo do campo central.
CAPS	Chave maiúsculas/minúsculas.
CTRL D	Inserir uma marca de mensagem especial. É apresentado o sinal "#" e o sistema espera uma tecla de 1 a 9, correspondente a uma mensagem. ESC cancela.
CTRL I	Inverte o vídeo.
CTRL T	Repõe o carácter ">" indicativo de posição padrão de impressão de mensagem..

3. REGISTRADORES

Os registradores são variáveis internas do jogo que podem assumir diversos estados. Eles são numerados de 1 a 99, sendo que alguns já tem funções pré-definidas. Um registrador pode assumir qualquer valor de 0 a 255, mediante a atuação de uma determinada instrução de programação (item 8.).

3.1 REGISTRADORES ESPECIAIS

O sistema do jogo, quando se reinicia uma partida, repõe numa tabela interna a condição de existência de cada objeto. Dessa forma, em um byte indexado são guardadas todas as informações referentes a um determinado objeto, tais como posição em que se encontra, existência ou não, se está sendo carregado ou se está dentro de outro objeto, etc (ver também item 5.2).

O acesso a essa tabela pode ser feito mediante a referência ao objeto, como se ele fosse um registrador. A fórmula para o acesso é indexar um objeto como REGISTRADOR+100. Por exemplo, se desejarmos manipular diretamente o objeto 7, basta usar uma instrução de programação com um índice 107.

Essa forma de manipulação também pode ser usada pelos comandos do EDITOR (LR X e MR X).

3.2 REGISTRADORES PRÉ-DEFINIDOS

Os registradores 1 a 14 possuem o seguinte significado:

- | | |
|---------------|----------------------------------------------------------------------------------------------------------------|
| REGISTRADOR 1 | corresponde ao número da posição em que o jogador se encontra. |
| REGISTRADOR 2 | é o byte menos significativo do contador de jogadas. É incrementado a cada ENTER. |
| REGISTRADOR 3 | é o byte mais significativo do contador de jogadas. |
| REGISTRADOR 4 | é um contador especial, de um byte, que é incrementado a cada frase comando, sempre que for diferente de zero. |

- REGISTRADOR 5** é um contador especial, de um byte, que é decrementado a cada frase comando sempre que for diferente de zero. Ele é acionado com um valor diferente de zero e quando a contagem atingir zero a FUNÇÃO 2 é executada automaticamente. Seu mecanismo é semelhante a uma bomba relógio.
- REGISTRADOR 6** é um contador de passos no escuro, ou seja, todas as frases comandos inseridas enquanto se está no escuro causam o incremento desse registrador. Quando o contador atinge o valor 5 a FUNÇÃO 3 é executada automaticamente.
- REGISTRADOR 7** conta quantos objetos estão dentro do objeto 3. Normalmente são possíveis apenas três objetos.
- REGISTRADOR 8** conta quantos objetos estão sendo carregados pelo jogador. Normalmente o jogador carrega apenas 5 objetos.
- REGISTRADOR 9** flag de iluminação do local onde se encontra o jogador. Se o seu valor for 0, o jogador está numa posição clara e se o seu valor for 1, o jogador está no escuro.
- REGISTRADOR 10** flag de estado do objeto 2. Se o seu valor for 0, o objeto 2 está apagado e se o seu valor for 1, o objeto 2 está aceso.
- REGISTRADOR 11** contador de minutos do relógio do jogo.
- REGISTRADOR 12** contador de horas do relógio do jogo.
- REGISTRADOR 13** contador de dias do relógio do jogo.
- REGISTRADOR 14** valor de ajuste para o relógio interno do sistema. Determina a cadência dos minutos pela fórmula: $\text{REGISTRADOR14} * 65536 * \text{FATOR}$. O FATOR corresponde ao tempo de execução das instruções em código de máquina no ciclo de "real time" do sistema multiplicado pelo clock do computador.

No MSX, o valor 1 corresponde a aproximadamente 1 segundo.

4. VERBOS

Os verbos são os elementos que iniciam a frase comando do jogador e devem definir claramente a ação que ele quer que o sistema execute.

Os verbos são referenciados por uma numeração que inicia com o valor 1 e pode terminar com o valor 209. Essa numeração é atribuída automaticamente, durante a criação dos verbos, e indica a posição, na tabela de definição dos verbos, que o verbo ou conjunto de sinônimos ocupa.

Um verbo é definido pelo processo normal de edição do campo central da tela, sendo que os verbos sinônimos são definidos em sequência. Por exemplo, o verbo PEGUE ocupa a posição 20. Para criar um conjunto de sinônimos para o ato (pegar), basta editar o verbo 20 (comando EV 20) e acrescentar os sinônimos desejados:

```
>PEGUE:CARREGUE:APANHE
```

é imprescindível não esquecer do delimitador ':' entre os verbos.

Todos os comandos e funções de manipulação da tela (itens 2.2 e 2.3) estão também disponíveis na criação/edição dos verbos.

Não há necessidade de se criar uma forma simplificada de reconhecimento, do tipo:

```
>PEGUE:PEG:PEGU
```

Durante o jogo o sistema só aceitará frases com letras maiúsculas e portanto só defina verbos desta forma.

Os verbos, ou o conjunto de sinônimos, devem refletir sempre um mesmo conceito de ação. Por exemplo, dado o conjunto:

```
>FAÇA:CONSTRUA:MONTE:CRIE
```

o verbo MONTAR está conceitualmente errado pois a montagem de alguma coisa não é uma ação tão global quanto a ação refletida pelos outros verbos. Se durante um jogo quisermos fazer uma flecha, por exemplo, a frase MONTE UMA FLECHA terá o mesmo efeito que a frase FAÇA UMA FLECHA. Apesar disto, o sentido de MONTE UMA FLECHA não é idêntico ao de FAÇA UMA FLECHA.

Por outro lado, MONTE O CAVALO é bastante explícito e FAÇA UM CAVALO provoca um erro conceitual grave. Nesses casos, o mais correto é separar os verbos que tendem a gerar situações dúbias e, para as ações comuns - FAÇA UMA BALSA e MONTE UMA BALSA - definir dois comandos distintos. O próprio sistema EDITOR possui uma instrução (CMD X) específica para esses casos, afim de que não seja necessária a digitação de todas as instruções do comando.

4.1 DIVISÃO FUNCIONAL DOS VERBOS

Os verbos possuem uma distribuição espacial, na tabela, que não pode ser alterada até o verbo 34. Essa distribuição atende a aspectos técnicos de funcionalidade do sistema do jogo e é a seguinte:

1 NORTE	10 DESÇA	19 CORRA	28 EXAMINE
2 SUL	11 HORAS	20 PEGUE	29 PROCURE
3 LESTE	12 QUANTO	21 COLOQUE	30 OFEREÇA
4 OESTE	13 TEMOS	22 TROQUE	31 FAÇA
5 GRAVE	14 RECOMECE	23 COMPRE	32 JOGUE
6 RECUPERE	15 HÁ	24 ROUBE	33 CONSERTE
7 ENTRE	16 GARIMPE	25 TIRE	34 VENDA
8 SUBA	17 PENSE	26 QUEBRE	
9 SAIA	18 GRITE	27 SOLTE	

5. OBJETOS

Todas as palavras que necessitarem de reconhecimento e não puderem ser definidas como VERBO, devem ser definidas como OBJETO, mesmo que a sua existência dentro do jogo não se efetue.

O sistema permite a criação e manipulação de até 99 objetos, indexados de 1 a 99, sendo que os 3 primeiros fazem parte do sistema de uma forma pré-definida.

5.1 OBJETOS PRÉ DEFINIDOS

Por motivos técnicos o objeto 1 é a palavra LOCAL, a fim de se obter uma melhor performance dos comandos que fazem a verificação dos objetos que se encontram no mesmo local que o jogador.

O objeto 2 deve ser, no jogo, o objeto para a iluminação dos locais escuros. A verificação de existência, posse ou colocação e condição de luminância são automáticas e não precisam ser estruturadas pelo autor.

O objeto 3 deve ser, no jogo, um objeto passível de receber outros dentro de si. A manipulação e operação desse objeto também é automática.

5.2 SITUAÇÃO DE UM OBJETO

Os objetos podem ser manipulados de diversas formas durante um jogo. A situação inicial é definida em edição e, durante a inicialização do jogo, é transferida para uma tabela interna do sistema.

A situação atual de um objeto é determinada por uma variável indexada, cujo valor pode ser lido/alterado pelo uso das instruções de programação (ver também item 3.). A interpretação desses valores deve obedecer aos seguintes padrões:

0 objeto não existe ou é uma palavra apenas para possibilitar o reconhecimento de um comando.

1 a 99 objeto se encontra na posição correspondente a esse valor, podendo ser listado pelo comando EXAMINE O LOCAL.

101 a 199	objeto na posição valor -100 porém não será listado pelo comando EXAMINE O LOCAL.
AA ou 250	objeto sendo carregado.
BB ou 251	objeto dentro do objeto 3 liberado para ser pego (i.e. objeto 3 está aberto).
DD ou 253	objeto dentro do objeto 3 sem liberação para ser pego (i.e. objeto 3 está fechado).

5.3 BYTE DE CONSISTÊNCIA

Todos os objetos, quando são criados, recebem um byte de consistência onde os bits de 0 a 6 servem para determinar algumas particularidades relativas aquele objeto.

Eles indicam se o objeto possui, ou não, uma pré-disposição para a ação correspondente ao verbo que o bit representa. Se não houver nenhum comando de interceptação e o bit correspondente for 1, então para aquela ação+objeto não há necessidade de estruturar um comando específico pois o objeto é considerado um objeto comum.

Isto significa, em síntese, que para cada bit do byte de consistência há uma FUNÇÃO associada e que é responsável pela execução da ação correspondente.

Por exemplo, um objeto (VELA) cujo bit 0, do byte de consistência, possui o valor 1 será considerado como um objeto comum e passível de ser PEGO, pela FUNÇÃO 5, sem quaisquer condições pré-definidas (salvo se não houver um comando PEGUE A VELA).

A funcionalidade das ações relativas aos bits está associada a uma determinada FUNÇÃO, de acordo com a tabela abaixo:

Bit	FUNÇÃO	Significado
0	5	Pegar um objeto.
1	6	Colocar um objeto dentro do objeto 3.
2	7	Trocar um objeto por outro.
3	8	Comprar um determinado objeto.
4	9	Roubar um determinado objeto.
5	10	Tirar um objeto de algum lugar.
6	11	Quebrar um objeto.

5.4 CRIANDO OU EDITANDO UM OBJETO

Quando for acionado um comando de criação/edição de objetos, o sistema solicitará o nome pelo qual tal objeto será reconhecido. Podem ser usados sinônimos para desequinar um mesmo objeto, por exemplo: BALA:MUNICÃO:CARTUCHO. é imprescindível a presença do delimitador ":" entre os sinônimos.

Logo após é solicitada a condição de existência do objeto. O parâmetro digitado deve corresponder à tabela de situação do objeto (item 5.2).

A seguir é solicitado cada um dos 6 bits de consistência. Nesta operação são aceitas apenas as teclas "1" e "0". O zero indica que o objeto em questão irá exigir comandos específicos ou que as FUNÇÕES correspondentes aos bits não serão possíveis para tal objeto. O um indica que a FUNÇÃO padrão correspondente deverá ser acionada pelo sistema.

Após a definição do byte de consistência, o sistema solicitará o texto da mensagem que irá descrever o objeto. Nesta operação são possíveis os mesmos comandos e recursos presentes na edição de mensagens.

6. POSIÇÕES

Posições são locais onde o jogador pode estar fisicamente. Elas não precisam ser necessariamente um local geograficamente definido, com saídas ou entradas.

O sistema permite a criação de no máximo 99 posições numeradas de 1 a 99.

A vizinhança das posições não precisa obedecer a nenhuma ordem numérica, podendo coexistir a posição 3 ao norte da posição 75 e ao sul da posição 16, por exemplo.

Toda posição tem definidos seus quatro limites cardeais no formato N/S/L/O, onde os valores decimais representam a ligação efetiva entre as posições. Um exemplo seria 75/16/00/00, onde o valor 0 (zero) corresponde a uma não vizinhança naquela direção, normalmente provocando uma resposta do tipo "NÃO É POSSÍVEL IR NESTA DIREÇÃO" quando se tentar seguir por ali.

Toda posição tem uma mensagem de descrição que é estruturada da mesma forma que as MENSAGENS. Essas descrições são acionadas quando o jogador pressiona a tecla ENTER, no jogo, sem que haja uma frase comando para ser processada.

6.1 MOVIMENTO ENTRE POSIÇÕES

Todo movimento é feito naturalmente desde que exista uma posição na sua direção, ou seja, se o jogador entrar com o comando NORTE, o sistema procurará o valor correspondente na tabela de definição geográfica da posição e se ele não for zero então o registrador 1 assumirá esse valor. Na prática temos a sensação de movimento do jogador.

Um movimento condicional é reconhecido se o valor encontrado na direção desejada for superior a 100. Nesse caso o movimento deve ser efetuado pela FUNÇÃO (valor - 100).

7. COMANDOS

Os comandos são definidos como séries de instruções executáveis pelo sistema. O comando é composto por um cabeçalho de identificação, a ser estruturado da frase composta pelo jogador, com os seguintes itens: VERBO + 1# OBJETO + 2# OBJETO. A ausência dos objetos implica no zeramento das suas posições relativas no cabeçalho.

Os COMANDOS não são numerados e portanto não tem restrições de quantidade, sendo referenciados pela sua própria estrutura de reconhecimento. Isso vale tanto para a criação quanto para a edição, ou seja: para criar o comando de pegar a vela, por exemplo, basta entrar com o mnemônico CC (Criar Comando) e reponder à questão "Qual comando?" com a sintaxe idêntica a do jogo: PEGUE A VELA.

O processo de digitação das instruções é idêntico ao descrito no item 8.1 (edição das instruções programáveis).

Os COMANDOS são executados antes do cheque das operações padronizadas e portanto sua funcionalidade tem prioridade. Por exemplo: para um objeto X, cujo bit 0 do byte de consistência tiver o valor 1, então esse objeto será 'prego' pela FUNÇÃO 6. Se o ato de 'pegar' tiver alguma particularidade então deverá ser criado um COMANDO 'PEGUE OBJETO X'.

7.1 COMANDOS JÁ DEFINIDOS

O sistema já vem com três comandos definidos. São eles:

EXAMINE LOCAL	faz uma descrição sumária dos objetos que se encontram na mesma posição do jogador.
TEMOS	faz uma listagem dos objetos que estão sendo carregados pelo jogador.
REINICIE	reinicia a partida, resetando todas as variáveis.

8. INSTRUÇÕES DO SISTEMA

O sistema EDITOR possui uma série de instruções internas para a composição dos COMANDOS e das FUNÇÕES. Elas são referenciadas por mnemônicos e podem ter ou não parâmetros adicionais.

A construção, ou estruturação, de um COMANDO ou FUNÇÃO obedece aos conceitos estruturais da programação normal de uma linguagem. A referência a uma determinada instrução é feita por labels numerados de A a T, ou seja, toda instrução, ou parte da estrutura, pode ser considerada como um ponto de desvio e reentrada na execução do COMANDO ou da FUNÇÃO.

8.1 EDIÇÃO DAS INSTRUÇÕES

Ao editar um COMANDO ou uma FUNÇÃO, o sistema apresenta um prompt ">" no centro e no alto do campo central. Esse cursor indica que a instrução por ele apontada, ou sua posição na listagem, está em evidência. Todas as inserções, modificações, anulações e definições serão assumidas nessa posição.

Os comandos desse editor de instruções são:

SETA P/BAIXO	avança uma instrução da listagem.
SETA P/CIMA	retrocede uma instrução até o topo do campo central.
HOME	reinicia a listagem a partir da primeira instrução.
SETA P/DIREITA	insere um espaço na posição do cursor. Pode ser usada a tecla INS.
SETA P/ESQUER	deleta uma instrução na posição do cursor. Pode ser usada a tecla DEL.
ESC	cancela o comando sem alteração nas tabelas.
SELECT	compila a listagem.
QQ TECLA	aceita uma instrução. Se a instrução digitada não corresponder a uma instrução do sistema, na posição do cursor o sistema imprime um asterisco (*). Se o primeiro caracter da instrução for "#" a próxima letra será interpretada como label. Se os parâmetros não foram digitados o sistema assume o valor 0. O espaço entre o mnemônico e os parâmetros não é optativo e deve ser digitado. A tecla ESC cancela a digitação da instrução.

8.2 TABELA DE INSTRUÇÕES

NOP	sem efeito; não produz resultado.
MSG X	imprime a mensagem X.
NVC	aguarda novo comando.
LLIST	lista os objetos que estão no local e aguarda novo comando.
CLIST	lista os objetos que estão sendo carregados e aguarda novo comando.
DLIST	lista os objetos que estão dentro do 3 e aguarda novo comando.
OBJ X	imprime o objeto X. Se X=0 usar, para efeito de execução, o objeto em evidência.
INC X	incrementa o registrador X.
DEC X	incrementa o registrador X.
LDR X, Y	grava o registrador X com o valor Y.
SOMA X, Y	soma ao registrador X o valor Y.
RND X, Y	soma ao registrador X um valor aleatório entre 0 e Y.
REG= X, Y, Z	se o registrador X tiver um valor igual a Y salta para o label Z, caso contrário continua a execução.
REG> X, Y, Z	se o registrador X tiver um valor maior que Y salta para o label Z, caso contrário continua a execução.
REG< X, Y, Z	se o registrador X tiver um valor menor que Y salta para o label Z, caso contrário continua a execução.
AQUI X, Y	se o objeto X estiver no local salta para o label Y. Se X=0 usar, para efeito de execução, o objeto em evidência.
LOCAL X, Y	se X for o local corrente salta para o label Y.
TEMOS X, Y	se o objeto X estiver sendo carregado salta para o label Y. Se X=0 usar, para efeito de execução, o objeto em evidência.
SOLTA X	solta o objeto X; se X=100 solta todos os objetos que estão sendo carregados. Se X=0 usar, para efeito de execução, o objeto em evidência.
PEGA X	carrega o objeto X. Se X=0 usar, para efeito de execução, o objeto em evidência.
CRIA X	coloca o objeto X no local corrente. Se X=0 usar, para efeito de execução, o objeto em evidência.
APAG X	apaga o objeto X do local ou sendo carregado. Se X=0 usar, para efeito de execução, o objeto em evidência.
GOSUB X	desvia execução para a FUNÇÃO X.
LIBR	libera todos os objetos que estão dentro do objeto 3.

TRC	tranca todos os objetos que estão dentro do objeto 3.
POE X	coloca o objeto X dentro do objeto 3. Se X=0 usar, para efeito de execução, o objeto em evidência.
ESV	esvazia o objeto 3.
OK	imprime 'Ok.' e aguarda novo comando.
REGN X	imprime o valor decimal do registrador X.
NVF	aguarda novo comando sem testar a FUNÇÃO 5.
9. MENSAGENS	
REF	retorna da FUNÇÃO 4.
FIM	finaliza a partida.
NEW	reinicia a partida.
DESC	descreve o local e aguarda novo comando.
RET	retorna para a chamada GOSUB. O retorno só será efetivo para uma única chamada. Não se deve usar mais do que um GOSUB por COMANDO ou FUNÇÃO.
GOTO X	salta para o label X.
PAUSA X	atrasa a execução do jogo em X segundos.
FLAG X, Y	grava a Flag X com o valor Y.
EVID X	coloca o objeto do buffer X em evidência. O buffer, nesse caso, refere-se ao primeiro ou segundo objeto que estiver compondo a frase comando do jogador.
7.1 AS MENSAGENS	
CLS	limpa a tela.
EVD= X, Y	se o objeto X estiver em evidência salta para o label Y, caso contrário continua a execução.
CHRS X	imprime o caracter do código X.
PRT X, Y	imprime a mensagem X, o objeto Y e aguarda novo comando.
DNT X, Y	se o objeto X estiver dentro do objeto 3, salta para o label Y.
CMD X	equivale ao comando X.

9. MENSAGENS

As MENSAGENS são os elementos mais importantes dentro de um jogo pois é através delas que o sistema informa o resultado operacional de uma frase comando, ou o estado em que se encontra o jogador.

Isso implica em que toda a construção gramatical das mensagens deve servir aos propósitos de orientar o jogador durante uma partida. Frases mal elaboradas podem dar margem a dúvidas ou situações embaraçosas, onde a confusão termina por prejudicar a qualidade geral do jogo. Boas mensagens, portanto, são essenciais para a obtenção de um jogo de qualidade.

As MENSAGENS são estruturadas através dos comandos de editoração de tela (itens 2.2 e 2.3). Elas são indexadas numa faixa entre 1 e 209, sendo que algumas tem uma representação especial para o sistema.

9.1 AS MENSAGENS ESPECIAIS

As MENSAGENS numeradas de 1 a 9 são especiais e servem para otimizar o espaço alocado em memória. Dessa forma, toda construção gramatical repetitiva, do tipo "Nós estamos", pode ser substituída por um código de valor 1 a 9 referente a uma dada mensagem.

Para definir uma mensagem especial basta editá-la (MENSAGEM 1 a 9) como se fosse uma MENSAGEM comum.

A sua utilização se dá da seguinte forma: durante a criação/editoração de uma outra MENSAGEM, ou da descrição de uma POSIÇÃO, ou até mesmo da descrição de um objeto, é necessário digitar toda a frase desejada. Ex:

)Nós estamos num quarto de hotel.

Após a formatação da tela, a construção gramatical que é usada como mensagem especial deve ser deletada por completo, deixando apenas um espaço para receber o código de identificação. Ex:

) num quarto de hotel.

Com o cursor entre o caracter ")" e a palavra "num" deve ser acionado o comando CTRL D que colocará na posição do cursor o caracter "#". Nesse ponto o sistema aguarda uma tecla de 1 a 9 que representará o código de uma MENSAGEM. Após a escolha da tecla, o cur-

sor deve ser levado até o final da frase e a operação deve ser encerrada com a tecla ENTER.

Toda vez que for solicitada a apresentação da MENSAGEM criada, ela se dará na sua forma completa. Ex:

>Nós estamos num quarto de hotel.

Normalmente a MENSAGEM 1 corresponde a "Nós estamos" e a MENSAGEM 2 corresponde a "Sinto muito mas".

9.2 AS MENSAGENS PRÉ-DEFINIDAS

As MENSAGENS 11 a 21 são mensagens pré-definidas pois a sua significação é interna ao sistema, ou seja, o próprio sistema do jogo as utiliza de uma forma autônoma. Elas correspondem a:

- MSG 10 é a apresentação inicial do jogo e deve conter o texto que introduz o jogador na partida.
- MSG 11 corresponde a "Achei o que você queria." e é usada sempre que a procura por um determinado objeto resulta positiva.
- MSG 12 é usada sempre que o jogador está numa posição escura sem um objeto para a iluminação do local. Normalmente "Está muito escuro aqui. é melhor arranjar alguma luz ou poderemos ter problemas."
- MSG 13 é usada sempre que a frase comando do jogador não foi integralmente interpretada pelo sistema. Normalmente "Perdão, não entendi...".
- MSG 14 é usada sempre que o jogador não pode se mover na direção desejada. Normalmente "é impossível ir nesta direção."
- MSG 15 é usada sempre que o sistema decodifica a frase comando mas não acha um COMANDO correspondente ou a ação do jogador não está enquadrada como uma ação já estabelecida como padrão. Normalmente "Isto não é possível."
- MSG 16 é usada sempre que o sistema precisa informar ao jogador de ele não está carregando o objeto mencionado na frase comando. Normalmente "Nós não temos (objeto)."
- MSG 17 é usada sempre que é solicitada a aquisição de um objeto que o jogador já possui. Normalmente "Nós já temos (objeto)."
- MSG 18 é usada sempre que é solicitado um objeto e este não se encontra no local e nem está com o jogador. Normalmente "Eu não estou vendo..."
- MSG 19 é usada sempre que é solicitado o exame de um determinado objeto e, não havendo uma descrição pormenorizada a ser impressa, o sistema assume que o objeto é um objeto comum. Normalmente "É apenas (objeto)."
- MSG 20 é usada sempre que o jogador tentar carregar mais objetos do que a quantidade permitida pelo sistema. Normalmente "Não dá para carregar mais nada."

é usada sempre que o jogador tentar colocar mais objetos, dentro do objeto 3, do que a quantidade permitida pelo sistema. Normalmente "Não cabe mais nada."

9.3 A ESTRUTURA DE UMA MENSAGEM

Uma mensagem é constituída de palavras, letras e caracteres gráficos cujo propósito é transmitir uma idéia ao jogador. Nesse sentido, toda a tela central do vídeo pode ser usada e para isso alguns caracteres do micro tem um significado especial.

Toda mensagem, descrição, etc, é automaticamente impressa a partir do ponto imediatamente posterior ao caracter "). Para que uma mensagem seja impressa a partir de outro ponto basta apagá-lo de sua posição original e imprimi-lo na posição imediatamente anterior à desejada.

É importante lembrar sempre que a rotina de CLS do sistema apaga a tela apenas a partir da posição inicial padrão. Tudo que hover antes será conservado na tela até que seja sobreposto por outra mensagem.

Todas as mensagens são impressas, durante o jogo, exatamente como foram formatadas no campo central do vídeo. Elas são compiladas, nas suas respectivas tabelas, do ponto inicial de impressão ("") até onde estiver o cursor quando ENTER for pressionado. Nesse momento o sistema compila a mensagem, compactando-a de acordo com uma técnica especial de compressão de espaços.

Existem dois caracteres especiais que possuem uma significação própria: o caracter "a" significa que será feito um CLS antes de imprimir o que o suceder e o caracter "=" significa que o texto a seguir deve ser tabulado para o início da próxima linha.

10. FUNÇÕES

As FUNÇÕES são séries de instruções programáveis que são executadas sob condições especiais. Ao contrário dos COMANDOS, que estão associados a uma intenção do jogador, as FUNÇÕES são solicitadas de acordo com a situação ou a fase em que se encontra o jogo.

Elas são indexadas de 1 a 209 e a sua programação é idêntica à dos COMANDOS (item 7.). As duas únicas diferenças funcionais entre FUNÇÕES e COMANDOS são o índice e a ausência de um cabeçalho. No mais a estrutura de ambos é idêntica.

As 20 primeiras FUNÇÕES do sistema possuem uma significação especial e seu conteúdo pode ser editado de acordo com o jogo a ser criado pelo usuário.

10.1 FUNÇÕES ESPECIAIS

- FUNÇÃO 1** reset do jogo. Essa FUNÇÃO é executada sempre que se inicializa o jogo.
- | | | |
|-----|-----|-------------------------------------|
| MSG | 10 | Imprime a mensagem de apresentação. |
| LDR | 1,1 | Coloca o jogador na posição 1. |
| NVC | | |
- FUNÇÃO 2** o REGISTRADOR 5, quando recebe um valor diferente de 0 é decrementado automaticamente, a cada frase comando, até conter zero novamente. Nesse momento a FUNÇÃO 2 é executada. Sua estrutura de acionamento é semelhante a uma bomba relógio.
- FUNÇÃO 3** esta FUNÇÃO é executada sempre que o jogador der mais do que 8 passos no escuro. é uma espécie de FUNÇÃO castigo por não acender uma luz.
- FUNÇÃO 4** esta FUNÇÃO é executada constantemente, desde que sua primeira instrução seja diferente de OP. Ela simula o "real time" do sistema.
- FUNÇÃO 5** esta FUNÇÃO é executada toda vez que o jogador entra uma frase comando, desde que sua primeira instrução seja diferente de NOP.

FUNÇÃO 6

esta é a FUNÇÃO padrão de 'pegar' objetos comuns.

EVID	1	Põe em evidência o primeiro objeto da frase.
AQUI	0,C	Verifica se ele se encontra no local.
TEMOS	0,A	Verifica se já temos o objeto.
MSG	18	Imprime "Eu não vejo..."
GOTO	B	
A MSG	17	Imprime "Nós já temos..."
B OBJ	0	Imprime o nome do objeto em questão.
CHRS	46	Imprime um ponto final.
NVC		
C PEGA	0	Pega o objeto requisitado.
OK		

FUNÇÃO 7

esta é a FUNÇÃO padrão para 'colocar' objetos dentro do objeto 3.

MSG 15
NVC

FUNÇÃO 8

esta é a FUNÇÃO padrão para 'trocar' objetos com alguém.

MSG 15
NVC

FUNÇÃO 9

esta é a FUNÇÃO padrão para 'comprar' alguma coisa.

MSG 15
NVC

FUNÇÃO 10

esta é a FUNÇÃO padrão para 'roubar' algum objeto.

MSG 15
NVC

FUNÇÃO 11

esta é a FUNÇÃO padrão para 'tirar' um objeto de dentro de outro (normalmente o 3).

MSG 15
NVC

FUNÇÃO 12

esta é a FUNÇÃO padrão para 'quebrar' algum objeto.

MSG 15
NVC

FUNÇÃO 13 esta é a FUNÇÃO padrão para 'soltar' um objeto.

EVID	1	Põe o primeiro objeto da frase em evidência.
TEMOS	0,A	Checa se o jogador tem o objeto.
MSG	16	Imprime "Nós não temos...".
OBJ	0	Imprime o nome do objeto.
CHRS	46	Imprime o ponto final.
NVC		
A SOLTA	0	Solta o objeto em questão.
OK		

FUNÇÃO 14 esta é a FUNÇÃO padrão para 'examinar' um objeto.

EVID	1	Põe o primeiro objeto da frase em evidência.
TEMOS	0,B	Checa se o jogador tem o objeto.
AQUI	0,B	Checa se o objeto está na posição do jogador.
MSG	18	Imprime "Eu não vejo...".
A OBJ	0	Imprime o nome do objeto.
CHRS	46	Imprime um ponto final.
NVC		
B MSG	19	Imprime "é apenas...".
GOTO	A	

FUNÇÃO 15 esta é a FUNÇÃO padrão para 'procurar' objetos.

EVID	1	Põe o primeiro objeto da frase em evidência.
AQUI	0,C	Checa se o objeto está na posição.
TEMOS	0,A	Checa se o jogador está com o objeto.
MSG	18	Imprime "Eu não vejo...".
GOTO	B	
A MSG	17	Imprime "Nós já temos...".
B OBJ	0	Imprime o objeto em questão.
CHRS	46	Imprime um ponto final.
NVC		
C MSG	11	Imprime "Achei o que você queria..".
NVC		

FUNÇÃO 16 esta é a FUNÇÃO padrão de 'dar' um objeto.

MSG	15
NVC	

FUNÇÃO 17 esta é a FUNÇÃO padrão de 'fazer' algum objeto.

MSG	15
NVC	

FUNÇÃO 18

esta é a FUNÇÃO padrão de 'jogar' algum objeto.

MSG 15
NVC

FUNÇÃO 19

esta é a FUNÇÃO padrão de 'consertar' um objeto.

MSG 15
NVC

FUNÇÃO 20

esta é a FUNÇÃO padrão de 'vender' um objeto.

MSG 15
NVC

11. UM EXEMPLO PRÁTICO

Para que as fases de criação e edição de um adventure sejam facilmente assimiladas, esse capítulo trata exclusivamente da construção de um exemplo simples.

Vamos tomar como ponto de partida um enredo voltado para uma situação normal. O jogador está preso em sua casa e deverá achar uma maneira de sair dela a tempo de assistir a decisão do campeonato nacional de futebol, onde seu time é um forte candidato à vitória.

Para efeito de simplificação, vamos desconsiderar a existência de janelas e outros elementos comuns a uma casa.

11.1 CRIANDO UM MAPA

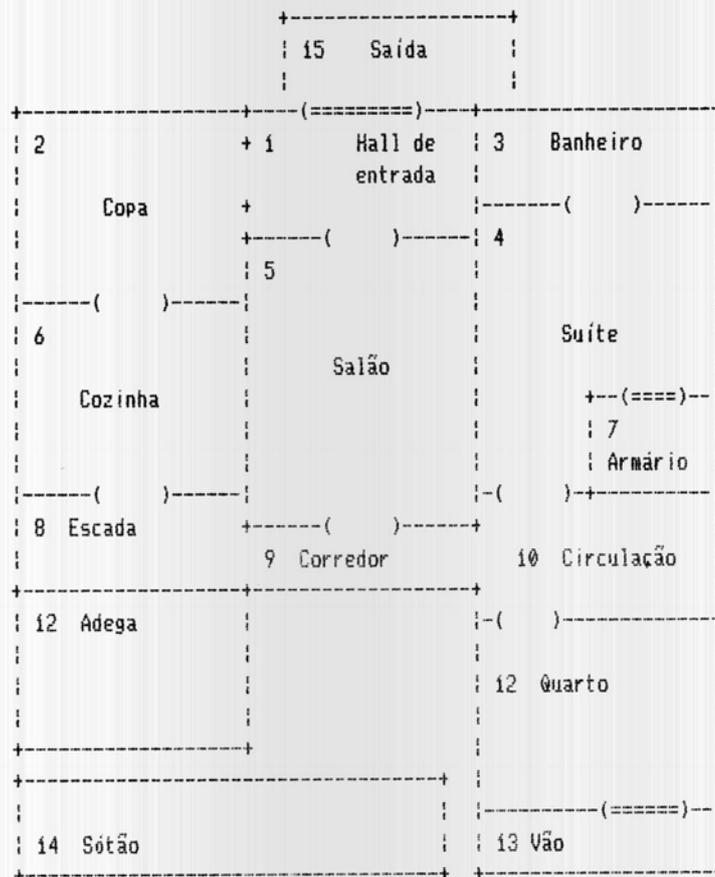
A primeira providência, para se escrever um adventure, é a criação de um mapa do jogo, que seja o mais completo possível. A figura na página seguinte ilustra a nossa MANSÃO e suas dependências.

A seguir é necessário criar os códigos de interrelação das posições (limites geográficos) e a descrição de cada uma delas. É preciso não esquecer da compactação das mensagens especiais (item 9.1) quando da digitação da descrição das posições.

Uma vez que a POSIÇÃO 1 já está criada, basta editá-la com o comando EP 1. As outras posições são criadas com o comando CP. As POSIÇÕES para digitação são:

POSIÇÃO 1	121/05/00/02
)Nós estamos no hall de entrada da casa.
POSIÇÃO 2	00/06/01/00
)Nós estamos na copa.
POSIÇÃO 3	00/04/00/00
)Nós estamos no banheiro.

MAPA DA MANSÃO:



POSICÃO 4

03/10/00/00

)Nós estamos na suíte.

POSICÃO 5

01/09/00/00

)Nós estamos no salão principal.

POSICÃO 6

02/08/00/00

)Nós estamos na cozinha.

POSICÃO 7

00/00/00/00

)Nós estamos dentro do armário.

POSICÃO 8

06/00/09/00

)Nós estamos perto da escada.

POSIÇÃO 9	05/00/10/08	>Nós estamos no corredor.
POSIÇÃO 10	04/11/00/09	>Nós estamos na circulação.
POSIÇÃO 11	10/00/00/00	>Nós estamos no quarto.
POSIÇÃO 12	00/00/00/00	>Nós estamos na adega.
POSIÇÃO 13	00/00/00/00	>Nós estamos num vão da parede. Ao norte existe um buraco que vai dar no quarto.
POSIÇÃO 14	00/00/00/00	>Nós estamos no sótão.
POSIÇÃO 15	00/00/00/00	>Nós estamos na entrada da casa.

11.2 A INICIALIZAÇÃO

O jogo começa executando a FUNÇÃO 1, que é responsável pelo reset. Inicialmente é apresentado o enredo do jogo (MENSAGEM 10) e a seguir o jogador é colocado numa das salas da casa. Note que a numeração das salas é importante pois o jogador não poderá iniciar o jogo nas posições 12, 13, 14 e 15 por motivos óbvios.

A FUNÇÃO 1 ainda é responsável pelo ajuste do relógio interno do jogo. Uma vez que essa partida deverá ter o seu desenvolvimento das 9 horas da manhã até às 15 horas, será necessário ajustar o relógio do sistema interno para uma cadência mais vagarosa.

Os elementos da inicialização são:

MENSAGEM 10 >Você mora, com um amigo, numa grande casa de um bairro afastado do centro da cidade.

Seu amigo foi viajar e levou a sua chave por engano. Foi dessa forma que você ficou preso na casa.

Agora você precisa achar uma maneira de sair dela a tempo de assistir a decisão do campeonato nacional de futebol, pois o seu time é um forte concorrente ao título.

O jogo começa às 5 horas da tarde. Boa sorte...

FUNÇÃO 1:

Inicialização do jogo.

MSG	10	Apresenta o jogo.
LDR	1,1	Coloca o jogador numa posição qualquer da casa.
RND	1,10	
LDR	12,9	Acerta o relógio para 9:00h do dia 3.
LDR	13,3	
LDR	14,80	Ajuste de cadência do relógio.
NVC		

11.3 AS PASSAGENS DO JOGO

A porta principal da casa é o objeto que se interpõe entre a saída e o jogador e, nesse caso, está associada ao REGISTRADOR 15, onde 0 significa "porta fechada" e 1 significa "porta aberta". A FUNÇÃO 2i é a responsável pela saída e finalização da partida e é solicitada quando o jogador comanda NORTE, estando na POSIÇÃO 1.

O OBJETO porta, na verdade, não existe e a sua criação é exigida apenas para o reconhecimento da palavra PORTA. O OBJETO em questão não pode ser pego e nem destruído, porém o comando EXAMINE A PORTA deve ser hábil o suficiente para reconhecer se ela está aberta ou fechada.

Existe também uma porta na suíte, que é a porta do armário. Ela está associada ao REGISTRADOR 16, onde 0 significa fechada e 1 significa aberta. Note-se que a movimentação da suíte para o armário não será comandada por uma frase SUL e sim por uma frase ENTRE NO ARMÁRIO. Para que isso funcione será necessário definir armário como um OBJETO.

Os elementos dessas duas passagens são:

OBJETO 4:

PORTA Ext: 0 Cons: 0000000 uma porta

PASSAGENS E MOVIMENTOS ESPECIAIS:

OBJETO 5:

ARMÁRIO Ext: 0 Cons: 0000000 um armário

MENSAGEM 22

)A porta está fechada.

MENSAGEM 23

)A porta está aberta.

MENSAGEM 24

)Parabéns, você conseguiu.

MENSAGEM 25

)= Você vai assistir a um jogão. Tomara que o seu time seja o campeão.

MENSAGEM 26)== Que pena, mas desta vez você não irá ver o seu time sagrar-se campeão. Melhor sorte na próxima.

MENSAGEM 27)== Joga novamente (S/N)?

FUNÇÃO 21: Saída pela porta principal.

REG= 15,1,A A porta está aberta? Sim-)A.
MSG 22 "A porta está fechada."
NVC
A LDR 1,15 Move o jogador para a POSIÇÃO 15.
MSG 24 "Parabéns, você conseguiu."
REG> 12,17,B O jogo já começou? Sim-)B
MSG 25 "Você vai assistir a um jogo..."
GOTO C
B MSG 26 "Perdeu o jogo..."
C PAUSA 10 Delay
MSG 27 "Joga novamente?"
FIM

COMANDO: ENTRE NO ARMÁRIO

LOCAL 4,A Nós estamos na suíte? Sim-)A
LOCAL 7,A Nós estamos dentro do armário? Sim-)A
MSG 18 "Eu não vejo um armário."
OBJ 5
CHRS 46
NVC
A REG= 16,1,B A porta do armário está aberta? Sim-)B
MSG 22 "A porta está fechada."
NVC
B LDR 1,7 Entra no armário.
DESC

COMANDO: ABRA A PORTA DO ARMÁRIO

LOCAL 4,A
LOCAL 7,A
MSG 18
OBJ 5
CHRS 46
NVC
A LDR 16,1 Abre a porta.
OK

COMANDO: FECHER A PORTA DO ARMÁRIO

LOCAL 4,A
LOCAL 7,A
MSG 18

OBJND: 5
CHRS 46
NVC
A LDR 16,0 Fecha a porta.
OK

COMANDO: SAIA DO ARMÁRIO

LOCAL 7,A Nós estamos dentro do armário? sim-)A
MSG 28 "Ficou maluco???"
NVC
A LDR 1,4 Vai para a suíte.
DESC

MENSAGEM 28)Ficou maluco???

Podemos otimizar a movimentação na suíte da seguinte forma:

FUNÇÃO 22: Movimento suíte/armário.

LOCAL 4,A Nós estamos na suíte? sim-)A
LOCAL 7,A Nós estamos dentro do armário? sim-)A
MSG 18 "Eu não vejo um armário."
OBJ 5
CHRS 46
NVC
A RET

COMANDO: ENTRE NO ARMÁRIO

GOSUB 22
REG= 16,1,A
MSG 22
NVC
A LDR 1,7
DESC

COMANDO: ABRA A PORTA DO ARMÁRIO

GOSUB 22
LDR 16,1
OK

COMANDO:**FECHE A PORTA DO ARMÁRIO**

GOSUB 22
LDR 16,0
OK

O acesso à adega é feito através da escada no fim do corredor (POSIÇÃO 8) e portanto é necessária a criação do OBJETO ESCADA para que a palavra seja reconhecida. Dessa forma podemos ir e voltar da adega sem maiores problemas.

Os itens relativos à esses movimentos são:

COMANDO:**DESCA A ESCADA**

LOCAL 8,A
MSG 18
OBJ 6
CHRS 46
NVC
A LDR 1,12
DESC

Nós estamos no fim do corredor? sim->A
"Eu não vejo uma escada."

Vai para a adega.

COMANDO:**SUBA A ESCADA**

LOCAL 12,A
MSG 18
OBJ 6
CHRS 46
NVC
A LDR 1,8
DESC

Nós estamos na adega? sim->A
"Eu não vejo uma escada."

Sobe a escada.

OBJETO 6:

ESCADA Ext: 0 Cons: 0000000 uma escada

A ida ao sótão é uma movimentação um pouco mais complexa. Suponhamos que o quarto tem uma parede falsa e que existe um vão atrás dela. A descrição da POSIÇÃO deverá insinuar algo para que o jogador se sinta tentado a fazer um buraco na parede. Esse buraco, que servirá como passagem está associado ao registrador 17, onde o valor 0 significa que ele não existe e o valor 1 significa que ele já foi feito.

A POSIÇÃO 10 deverá ser editada para comportar essa nova estrutura e o buraco deverá ser feito por uma picareta, que se encontra na adega. Note-se que a descrição da POSIÇÃO 11 será feita pelas MENSAGENS 27, 28 e 29 através da FUNÇÃO 23.

Após a feitura do buraco, o jogador poderá se deslocar até o vão atrás da parede onde, subindo num caixote, ele poderá ir ao sótão através de uma abertura no teto.

POSIÇÃO 10

4/123/0/9

>Nós estamos na circulação.

OBJETO 7:	BURACO	Ext: 0	Cons: 0000000	um uraco
OBJETO 8:	PAREDE	Ext: 0	Cons: 0000000	uma parede
OBJETO 9:	PICARETA	Ext: 12	Cons: 0000001	uma picareta
OBJETO 10:	CAIXOTE	Ext: 13	Cons: 0000000	um caixote

COMANDO:

FAÇA UM BURACO NA PAREDE

LOCAL 11,A	Nós estamos no quarto?
MSG 15	"Isto não é possível".
NVC	
A TEMOS 9,B	Nós temos uma picareta?
MSG 16	"Nos não temos uma picareta."
OBJ 9	
CHRS 46	
NVC	
B LDR 17,1	Faz o buraco.
OK	

COMANDO:

ENTRE NO BURACO

LOCAL 11,B	Nós estamos no quarto?
A MSG 18	"Eu não vejo um buraco."
OBJ 7	
CHRS 46	
NVC	
B REG= 17,0,A	O buraco está fechado?
LDR 1,13	Entra no vão.
DESC	

COMANDO:

SUBA NO CAIXOTE

LOCAL 13,A	Nós estamos no vão?
MSG 15	"Isto não é possível"
NVC	
A LDR 1,14	Vai para o sótão.
MSG 32	
NVC	

COMANDO:

DESÇA

LOCAL 14,A	Nós estamos no sótão?
MSG 15	"Isto não é possível."
NVC	

A LDR	1,13	Desce para o vão.
DESC		
MENSAGEM	29)Nós estamos no quarto.
MENSAGEM	30)== A parede me parece estranha.
MENSAGEM	31)== Há um buraco enorme na parede.
MENSAGEM	32)Nós chegamos ao sótão da casa através de uma abertura no teto do vão.

11.4 CRIANDO OBJETOS

Alguns objetos deverão estar na casa e a sua existência deve ser decidida com antecedência. Alguns deles desempenham um papel específico no jogo e outros existem apenas para constar como elementos figurativos. Vejamos caso a caso.

VELA:

A vela será o objeto iluminante do jogo e o local escuro será a adega. Isso implica em alterar o movimento de descer e subir a escada que dá acesso à adega. Para acender a vela será necessário que o jogador possua um isqueiro.

Para dificultar um pouco o jogo, vamos considerar que a vela já tenha sido usada anteriormente e que sobrou apenas um 'cotoco'. O tamanho do cotoco será definido pelo tempo que ele puder ficar aceso. Isso será feito da seguinte forma: ao acender a vela, o seu tempo de vida útil será igual aos minutos que faltarem para o relógio do jogo completar mais uma hora. Isso fará com que o tamanho da vela seja aleatório e dependente do momento em que a mesma foi acesa.

O controle dessa fase ficará a cargo da FUNÇÃO 4, que é o 'real time' do sistema.

Os elementos ligados ao OBJETO VELA são:

OBJETO	2:	VELA	Ext: 6	Cons: 0000011	uma vela
OBJETO	11:	ISQUEIRO	Ext: 5	Cons: 0000011	um isqueiro

COMANDO:

DESCA A ESCADA

LOCAL	8,A	Nós estamos no fim do corredor? sim-)A
MSG	18	"Eu não vejo uma escada."
OBJ	6	
CHRS	46	
NVC		
A LDR	1,12	Vai para a adega.
LDR	9,1	Escurece a posição.
LDR	6,0	Zera o contador de passos no escuro.
DESC		

COMANDO:

LOCAL 12,A
 MSG 18
 OBJ 6
 CHRS 46
 NVC
 A LDR 1,8
 LDR 9,0
 DESC

SUBA A ESCADA

Nós estamos na adega? sim->A
 "Eu não vejo uma escada."

Sobe a escada.
 Ilumina a posição.

COMANDO:

TEMOS 2,A
 MSG 16
 OBJ 2
 CHRS 46
 NVC
 A TEMOS 11,B
 MSG 16
 OBJ 11
 CHRS 46
 NVC
 B LDR 10,1
 FLAG 0,0
 OK

ACENDA A VELA

Nós temos a vela?
 "Nós não temos uma vela."

Nós temos um isqueiro?
 "Nós não temos um isqueiro."

Acende a vela.

COMANDO:

TEMOS 2,A
 MSG 16
 OBJ 2
 CHRS 46
 NVC
 A LDR 10,0
 FLAG 0,1
 NVC

APAGUE A VELA

Nós temos a vela?

Apaga a vela

FUNÇÃO 4:

REG= 10,1,A
 REF
 A REG= 11,0,B
 REF
 B APAG 2
 LDR 10,0
 MSG 35
 REF

Controle da vela acesa.

A vela está acesa?

O relógio marcou mais uma hora?

Derrete a vela.

"Ih! a vela derreteu."

MENSAGEM 35

>Ih! a vela derreteu-se toda.

CAMA:

A cama é um objeto que deverá existir tanto no quarto quanto na suíte. Como o sistema não admite dois objetos com a mesma designação, deve-se editar a FUNÇÃO 5 para que ela estabeleça em qual posição estará a cama. Essa posição deverá ser a mesma que a do jogador.

Outro ponto importante é a descrição da cama. Para isso será criado um comando EXAMINE A CAMA afim de que sejam diferenciadas as camas de casal, na suíte, e solteiro, no quarto. Esse sistema implica em que as camas não poderão ser carregadas pelo jogador.

OBJETO 12:

CAMA Ext: 0 Cons: 0000000 uma cama

FUNÇÃO 5:

Local onde está a cama.

LOCAL 4,A
LOCAL 11,A
NVF
A CRIA 12
NVF

Nós estamos na suíte?
Nós estamos no quarto?
Coloca a cama na posição do jogador.

COMANDO:

EXAMINE A CAMA

LOCAL 4,A
LOCAL 11,B
MSG 18
OBJ 12
CHRS 46
NVC
A MSG 33
NVC
B MSG 34
NVC

Nós estamos na suíte?
Nós estamos no quarto?
"Eu não vejo uma cama."

é uma cama de casal."
"É uma cama de solteiro."

MENSAGEM 33

>É uma cama de casal.

MENSAGEM 34

>É uma cama de solteiro.

RÁDIO:

O rádio é um objeto que terá um desempenho especial. Ele deverá ser ligado pelo jogador para que este saiba quanto tempo ainda lhe resta até a hora do jogo. A programação da Rádio FM Endado é basicamente música e hora certa.

Para que o rádio funcione o jogador deverá colocar-lhe pilhas que, evidentemente, estão na geladeira recarregando. A programação da Rádio é controlada pela FUNÇÃO 4.

Os REGISTRADORES associados ao rádio são:

REGISTRADOR 18 0=rádio sem pilhas / 1=rádio com pilhas
REGISTRADOR 19 0=rádio desligado / 1=rádio ligado
REGISTRADOR 20 1 a 3=tipo de música que está tocando.

OBJETO 13: RÁDIO Ext: 7 Cons: 0000011 um rádio
OBJETO 14: PILHAS Ext: DD Cons: 0000011 pilhas

VERBOS: LIGUE e DESLIGUE

COMANDO: COLOQUE PILHAS NO RÁDIO

TEMOS	13,A	Nós temos o rádio?
MSG	16	"Nós não temos um rádio."
OBJ	13	
CHRS	46	
NVC		
A TEMOS	14,B	Nós temos as pilhas?
MSG	16	"Nós não temos pilhas."
OBJ	14	
CHRS	46	
NVC		
B APAGA	14	Coloca as pilhas no rádio.
LDR	18,1	
OK		

COMANDO: LIGUE O RÁDIO

TEMOS	13,A	Nós temos o rádio?
MSG	16	"Nós não temos um rádio."
OBJ	13	
CHRS	46	
NVC		
A REG=	18,1,B	O rádio tem pilhas?
MSG	36	"O rádio não funciona."
NVC		
B LDR	19,1	Liga o rádio.
MSG	37	"Boa tarde ouvintes..."
REGN	12	Imprime as horas.
CHRS	58	
REGN	11	
CHRS	104	
NVC		

COMANDO: DESLIGUE O RÁDIO

TEMOS	13,A	Nós temos o rádio?
MSG	16	"Nós não temos um rádio."

OBJ	13	
CHRS	46	
NVC		
A LDR	19,0	Desliga o rádio.
OK		

MENSAGEM 36)O rádio não funciona.

MENSAGEM 37)Boa tarde ouvintes da sua Rádio FM Endado. A nossa programação de hoje é especial para ouvintes nervosos: música para acalmar as neuras.

No pedaço são exatamente:

MENSAGEM 38)== A sua FM preferida informa: são

MENSAGEM 39)== E agora uma seleção de músicas para meditar.

MENSAGEM 40)== É rock paulera minha gente.

MENSAGEM 41)== Vamos ouvir agora muito samba e pagode.

FUNÇÃO 4: Controle da vela e da Rádio.

REG=	10,0,A	A vela está apagada?
REG>	11,0,A	O relógio contou mais uma hora?
APAGA	2	Derrete toda a vela
LDR	10,0	
MSG	35	Ih! derreteu toda a vela."
A REG=	19,1,B	O rádio está ligado?
REF		
B REG=	11,0,C	Anuncia as horas?
REG=	11,15,C	
REG=	11,30,C	
REG=	11,45,C	
LDR	20,0	
REF		
C REG=	20,0,D	É para imprimir a seleção musical e a hora?
REF		
D LDR	20,1	Seleciona a música.
RND	20,2	
MSG	38	Imprime as horas.
REGN	12	
CHRS	58	
REGN	11	
CHRS	104	
REG=	20,1,E	Imprime a música escolhida.
REG=	20,2,F	
MSG	37	Meditação
REF		
E MSG	40	Rock
REF		
F MSG	41	Samba
REF		

GELADEIRA:

A geladeira será o objeto 3 e portanto poderá receber outros objetos dentro dele. A definição do objeto é idêntica aos demais. O REGISTRADOR 21 indicará o seu estado: 0 significa que a geladeira está fechada e 1 significa que ela está aberta.

O processo de colocação dos objetos dentro da geladeira é automático, ou seja, ele é definido pela FUNÇÃO 7 e depende do bit 1 do byte de consistência de cada objeto. Toda frase comando do tipo COLOQUE O OBJETO NA GELADEIRA será desviado para a FUNÇÃO 7 onde o objeto será obtido da própria frase do jogador, pela instrução EVID 1. A FUNÇÃO 7 deverá ser construída sem especificações de objeto (índice 0), pois assim o objeto citado na frase será utilizado como default.

Esse processo deverá ser utilizado também para as outras FUNÇÕES ESPECIAIS.

OBJETO 3:

GELADEIRA Ext: 6 Cons: 0000000 uma geladeira

COMANDO:**ABRA A GELADEIRA**

LOCAL	6,A	Nós estamos na cozinha?
MSG	18	"Eu não vejo uma geladeira."
OBJ	3	
CHRS	46	
NVC		
A LDR	21,1	Abre a geladeira.
LIBR		
OK		

COMANDO:**FECHE A GELADEIRA**

LOCAL	6,A	Nós estamos na cozinha?
MSG	18	"Eu não vejo uma geladeira."
OBJ	3	
CHRS	46	
NVC		
A LDR	21,0	Fecha a geladeira.
TRC		
OK		

COMANDO:**EXAMINE A GELADEIRA**

LOCAL	6,A	Nós estamos na cozinha?
MSG	18	"Eu não vejo uma geladeira."
OBJ	3	
CHRS	46	
NVC		
A REG=	21,1,B	Ela está aberta?
MSG	43	"A geladeira está fechada."
NVC		
B MSG	44	Descreve o conteúdo.
DLIST		

MENSAGEM 42)Ela está aberta.
MENSAGEM 43)Ela está fechada.
MENSAGEM 44)Na geladeira tem:

FUNÇÃO 7: Colocar objetos na geladeira.

EVID	1	Coloca objeto em evidência.
AQUI	0,A	O objeto está no local?
TEMOS	0,A	Estamos carregando-o?
MSG	18	"Eu não vejo o objeto."
OBJ	0	
CHRS	46	
NVC		
A LOCAL	6,B	Nós estamos na cozinha?
MSG	18	"Eu não vejo uma geladeira."
OBJ	3	
CHRS	46	
NVC		
B REG=	21,1,C	A geladeira está aberta?
MSG	43	"Ela está fechada."
NVC		
C AQUI	0,D	O Objeto está no local?
SOLTA	0	
D POE	0	
OK		

Os objetos a seguir são apenas elementos figurativos e não desempenham nenhuma função especial.

OBJETO 15:	GELO	Ext: 00	Cons: 0000011	gelo
OBJETO 16:	FACA	Ext: 6	Cons: 0000011	uma faca
OBJETO 17:	MESA	Ext: 2	Cons: 0000000	uma mesa
OBJETO 18:	CADEIRA	Ext: 2	Cons: 0000001	uma cadeira
OBJETO 19:	TAPETE	Ext: 5	Cons: 0000011	um tapete
OBJETO 20:	ESPELHO	Ext: 3	Cons: 0000011	um espelho

11.5 PROJETANDO A SAÍDA

A saída, ou finalização da partida, é um dos pontos mais importantes na programação dos adventures. Um adventure pode perder toda a sua magia se o seu final não for projetado para premiar a paciência e inteligência do jogador. No exemplo desse capítulo vamos considerar um modo bem simples para a finalização do jogo: o jogador deverá achar a chave, que inicialmente estará no sótão, e com ela abrir a porta da frente.

O comando para abrir a porta deverá considerar que ele poderá ser utilizado para abrir outra porta no jogo, a porta do armário.

OBJETO 21:	CHAVE	Ext: 14	Cons: 0000011	uma chave
------------	-------	---------	---------------	-----------

COMANDO =**ABRA A PORTA**

LOCAL	4,A	Nós estamos na suíte?
LOCAL	7,A	Nós estamos dentro do armário?
LOCAL	1,B	Nós estamos no hall?
MSG	18	"Eu não vejo uma porta."
OBJ	5	
CHRS	46	
NVC		
A LDR	16,1	Abre a porta do armário.
OK		
B REG=	15,0,C	A porta está fechada?
MSG	23	"A porta está aberta."
NVC		
C TEMOS	21,D	Nós temos a chave?
MSG	16	"Nós não temos uma chave."
OBJ	21	
CHRS	46	
NVC		
D LDR	15,1	Abre a porta.
OK		

11.6 CONCLUSÃO

O exemplo criado nesse capítulo ilustra bem o processo de desenvolvimento de um adventure. No entanto ele não é um jogo completo e acabado.

Propositadamente alguns itens foram deixados incompletos ou foram omitidos. Após uma leitura atenta da construção desse exemplo, é possível identificar onde estão os pontos que podem gerar dúvidas durante o desenrolar da partida. O usuário deverá editá-los de acordo com a sua conveniência.

A seguir são dados algumas dicas sobre pontos críticos:

- O comando EXAMINE A PORTA deveria responder se ela está aberta ou fechada.
- O comando SAIA DO ARMARIO não verifica se a porta do mesmo está aberta. Dessa forma é possível sair dele mesmo que a porta esteja fechada.
- Quando o jogador se encontra na adega, o comando DESÇA A ESCADA irá resultar na resposta "Eu não vejo uma escada.". O mais natural seria avisar ao jogador que a escada, naquela posição, permite apenas subir. O mesmo acontece quando o jogador está na posição 8 e tenta subir a escada. Nesse caso a escada permite apenas descer para a adega.
- No quarto é possível fazer uma série de buracos na parede, quando o mais natural seria o comando informar que o buraco já havia sido feito anteriormente.
- O jogo não está programado para responder a um comando simples FAÇA UM BURACO. Nesse caso poderia haver a pergunta "Em que lugar?".
- O sistema do jogo poderia reconhecer os comandos ENTRE, SUBA, DESÇA e ENTRE sem a referência ao lugar.
- Se a vela derreter numa posição diferente da posição do jogador esse, ainda assim,

será informado. O mais natural seria considerar que o jogador não viu a vela derreter-se toda.

- O rádio informa as horas independentemente de estar com o jogador. Isso só deveria ocorrer quando o jogador possuísse o rádio ou se o mesmo estivesse na posição do jogador.

12. INFORMAÇÕES COMPLEMENTARES

Esse capítulo trata de algumas particularidades relativas ao sistema e também a cada computador. Sua leitura é essencial para uma operação satisfatória do EDITOR.

- Quando o jogador está numa posição escura e não possui um objeto para a iluminação do local, o sistema executa todas as frases comando, porém não fornece o relatório de desempenho. Nesse caso é impressa apenas a mensagem "Está escuro aqui..."

- A ordem funcional do loop de comando, do sistema do jogo, é a seguinte:

- 1) Executa a FUNÇÃO 5.
- 2) Recebe a frase comando do jogador.
- 3) Incrementa o REGISTRADOR 4.
- 4) Decrementa o REGISTRADOR 5 e testa se é zero.
- 5) Reconhece o conteúdo da frase comando.
- 6) Executa o COMANDO pertinente.
- 7) Volta ao item 1.

- Quando se está testando um jogo, o retorno ao EDITOR é feito mediante o acionamento da tecla ESC.

- O EDITOR permite a acentuação gramatical da mesma forma que a operação normal do computador. A primeira versão do EXPERT, no entanto, não possui todas as letras acentuadas e portanto algumas palavras poderão gerar problemas quando o jogo criado pelo EDITOR estiver sendo executado em outro computador, ou vier de outra versão de EXPERT.

- Alguns endereços importantes:

26.003 / 6593H: quantidade máxima de objetos na mão.

26.004 / 6594H: quantidade máxima de objetos dentro do objeto 3.