Role Playing Database
A Computerized Role Playing Database System
User Manual

Robert Heller
Deepwoods Software
Wendell, MA, USA

March 8, 2002

This documentation was prepared with LaTeX.

This document describes version 2.0 of the Role Playing Database package.

The author, Robert Heller, may be contacted electronically (E-Mail) via the following:

FidoNet 1:321/153, Locks Hill BBS.

InterNet heller@deepsoft.com

Web site URL: `http://www.deepsoft.com/`.

Contents

List of Figures

List of Tables

Preface

RPGs[1] are a popular pastime among many people these days. Maybe they are a form of escape from the rather mundane lives many people live, at least during the workday. An RPG allows the players to escape into a world where some things are simpler, and some things more complex, in interesting ways.

I have played AD&D a few times and was dismayed at the amount of paperwork needed to keep track of everything. Being a computer person, it seemed to me that most of this paperwork could be replaced by a computer and the information managed by a clever database system. Given that now there are high-powered laptop computers business people use to keep track of and manage large corporations, it should be possible to manage the odd imaginary universe on such a machine. So I wrote the *Role Playing DataBase System* to manage all of the information that goes with an RPG.

The *Role Playing DataBase System* maintains a database describing an RPG "universe". This "universe" contains a group of "characters", some player and some non-player, a collection of "monsters", and one or more "places" (dungeons usually) where the "monsters" reside, generally guarding some treasure. The *Role Playing DataBase System* helps game masters and players keep track of the various things in the make-believe universe in which the RPG takes place.

If you have *any* comments about this package, please let me know. My electronic mail addresses are listed on the back side of the title page. My postal address is listed in Appendix D. I would be very interested in any comments users of the *Role Playing DataBase System* package might have.

Robert Heller
Deepwoods Software

[1] RPG: Role Playing Game, a game where the players take on the roles of persons who might have lived (or may yet live) in a different time and place. See [2, 3].

Wendell, MA, USA
January 1999

Chapter 1

Introduction

1.1 What Is the *Role Playing DataBase System*?

The *Role Playing DataBase System* is a specialized database system with a GUI front end designed to aid people who play RPGs. Both the players and the masters can find uses for this package, to manage the information that describes the players' characters and the game environment and its contents.

The system stores the data describing the various game components (characters, monsters, spells, etc.) in a collection of data files. The data file formats are described in detail in Appendix A. Except for the Map data file, the data files contain the serialized contents of the various C++ class structures. See [6] for a detailed description of the C++ class library used to manage these data files.

The rest of the system consists of a collection of Tcl/Tk ([7]) script files that implement a GUI. See [4] for a detailed description of these script files. The Tcl/Tk scripts interface to the C++ class library via a *SWIG* ([1]) interface file, which is described in detail in [5].

1.2 How this Manual Is Organized

Most of this manual describes how to use the GUI.

The GUI has eight main toplevel GUI windows:

1. The *Main* window. This is the main window and it is described in

detail in Chapter 3.

2. The *Character Editing* window. This window is used to create and edit Character Object data files and it is described in detail in Chapter 4.

3. The *Monster Editing* window. This window is used to create and edit Monster Object data files and it is described in detail in Chapter 5.

4. The *Spell Editing* window. This window is used to create and edit Spell Object data files and it is described in detail in Chapter 6.

5. The *Map Editing* window. This window is used to create and edit Map Object data files and it is described in detail in Chapter 10.

6. The *Treasure Editing* window. This window is used to create and edit Treasure Object data files and it is described in detail in Chapter 9.

7. The *Trick / Trap Editing* window. This window is used to create and edit Trick or Trap Object data files and it is described in detail in Chapter 8.

8. The *Dressing Editing* window. This window is used to create and edit Dressing Object data files and it is described in detail in Chapter 7.

Common GUI elements are described in Chapter 2.

Chapter 2

Common GUI Elements

This chapter describes the common GUI elements used by the *Role Playing DataBase System*.

2.1 Standard Menu Bar

All of the main toplevel windows use a common menu bar. This menu bar has five pull-down menus as shown in Figure 2.1. These menus are:

1. `File` The File menu, shown in Figure 2.2, follows the Motif standard by having `New`, `Open...`, `Save`, `Save As...`, `Print...`, `Close`, and `Exit` menu items.

2. `Edit` The Edit menu, shown in Figure 2.3, follows the Motif standard by having `Undo`, `Cut`, `Copy`, `Paste`, `Clear`, `Delete`, `Select All`, and `De-select All` menu items.

3. `View` The View menu is presently empty. This menu is included for future functionality.

4. `Options` The Options menu is also presently empty. This menu is included for future functionality.

Figure 2.1: Standard Menubar

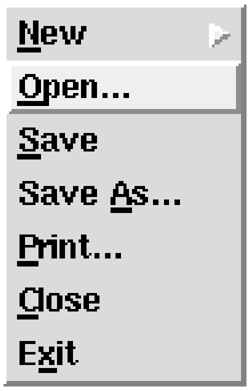

Figure 2.2: File Menu

Figure 2.3: Edit Menu

Figure 2.4: Help Menu

5. `Help` The Help menu, shown in Figure 2.4, follows the Motif standard by having `On Context`..., `On Help`..., `On Window`..., `On Keys`..., `Index`..., `Tutorial`..., and `On Version`... menu items, with the additional menu items `Warranty`..., `Copying`..., and `Registering`....

Chapter 3

Main GUI Window

The Main window, shown in Figure 3.1 contains a collection of eight buttons:

Make or Edit Character This button starts up the *Character* edit GUI toplevel window.

Make or Edit Monster This button starts up the *Monster* edit GUI toplevel window.

Make or Edit Spell This button starts up the *Spell* edit GUI toplevel window.

Make or Edit Map This button starts up the *Map* edit GUI toplevel window.

Make or Edit Treasure This button starts up the *Treasure* edit GUI toplevel window.

Make or Edit Trick or Trap This button starts up the *Trick / Trap* edit GUI toplevel window.

Make or Edit Dressing This button starts up the *Dressing* edit GUI toplevel window.

Exit This button closes all open windows and exits from the application.

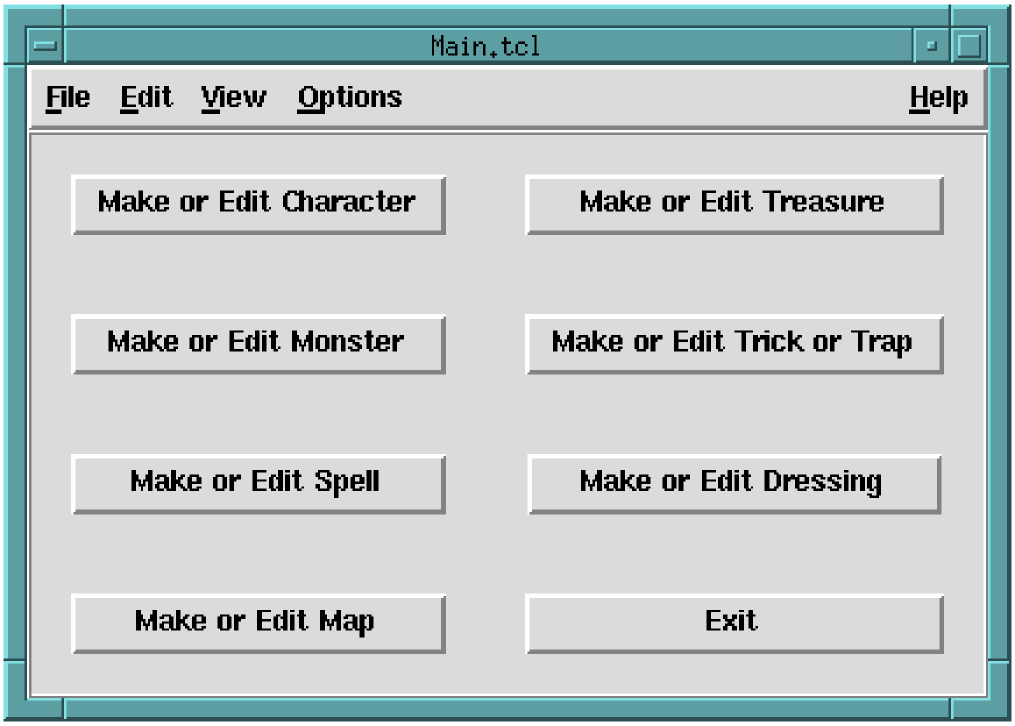

Figure 3.1: Main Window

Chapter 4

Character Data Object Editor GUI Window

The Character Data Object is used to represent characters, both player and non-player. The player characters are controlled by the corresponding player, whose goal is to survive and accumulate treasure, gold, and experience points. As the experience point total of the player's character increases, the character advances in levels, which means more hit points, increased access to spells (magic users) and so on. Non-player characters are the additional people in the environment and are “played” by the game master.

4.1 New Character Roll Dialog

When the Character Data Object Editor GUI Window is opened from the `New` menu item on the `File` menu or via the `Make or Edit Character` button on the Main GUI Window, a New Character Roll Dialog box appears, shown in Figure 4.1. This dialog box can be used to generate the attribute statistics for a new character, and has entries to select the attribute dice, the initial hit dice, and the maximum number of hit dice. There is a button to re-roll the attribute statistics. By default the attribute dice are 3d6, the initial hit dice are 1d6, and the maximum number of hit dice is 10.

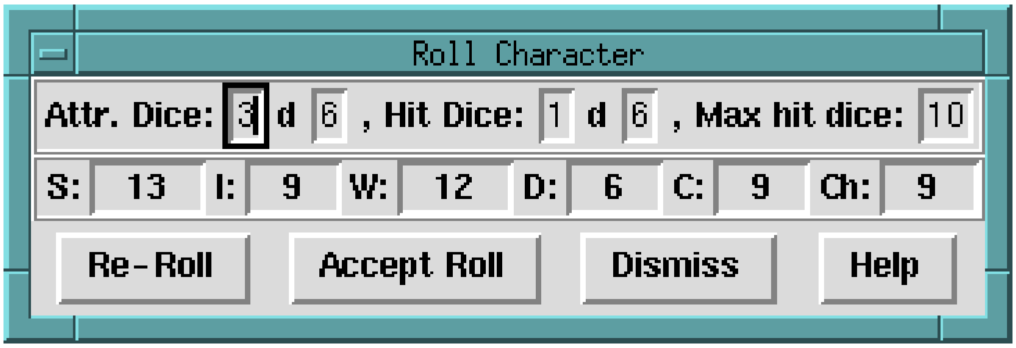

Figure 4.1: New Character Roll Dialog

4.2 Character Data Object Editor GUI Window

The Character Data Object Editor GUI Window contains entries to edit all of the settable fields in a Character Data Object as shown in Figure 4.2.

The settable fields include:

Strength The character's strength attribute. There is a second entry for a fighter's exceptional strength.

Intelligence The character's intelligence.

Wisdom The character's wisdom.

Dexterity The character's dexterity.

Constitution The character's constitution.

Charisma The character's charisma.

Character Name The name of the character.

Player Name The real name or nickname of the player.

Character Race The race of the character.

Character Alignment The character's alignment.

Character Sex The character's gender.

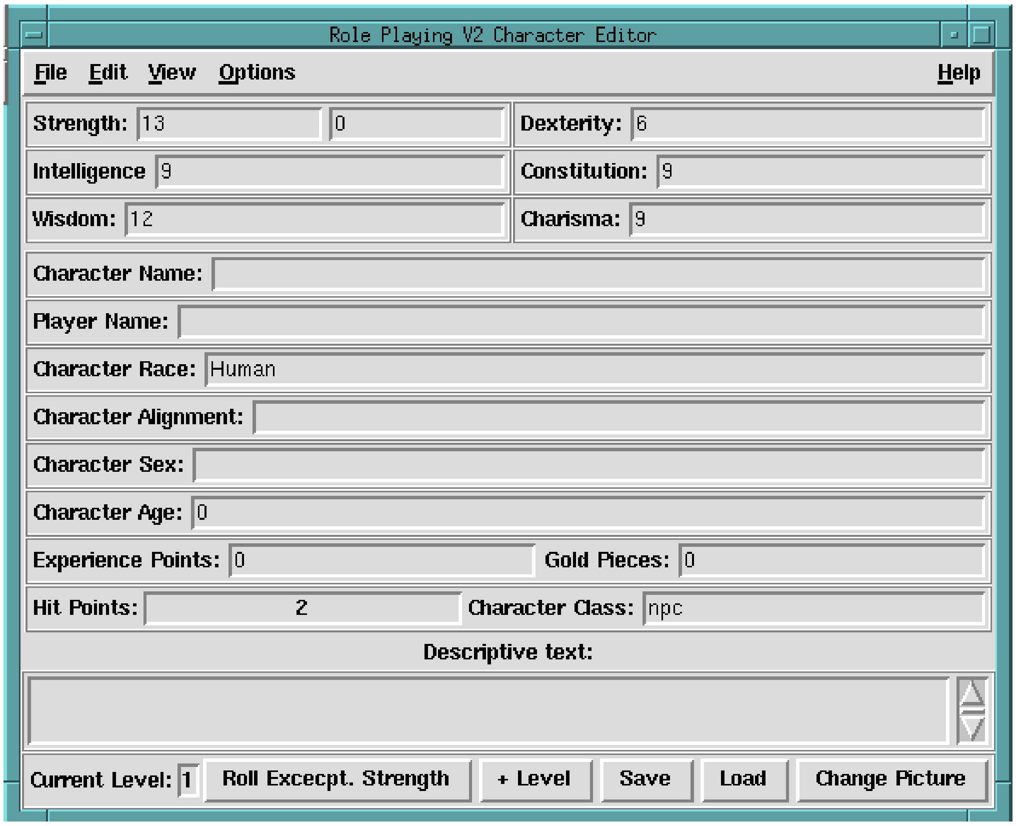

Figure 4.2: Character Data Object Editor GUI Window

Character Age The character's age.

Experience Points The number of experience points the character has accumulated.

Gold Pieces The number of gold pieces the character has accumulated.

Hit Points The character's hit points. This is a read-only field.

Character Class The class of the character, such as fighter or magic user.

Descriptive Text Additional text that further describes the character in detail.

There are buttons to advance the character's level, roll exceptional strength, save and load to and from data files, and a button to set or change the character's picture (GIF file).

Chapter 5

Monster Data Object Editor GUI Window

The Monster Data Object represents the monsters that the player characters might encounter. These monsters can be only a minor hindrance or a major problem, spelling disaster for the adventure.

The Monster Data Object Editor GUI Window contains entries to edit all of the settable fields in a Monster Data Object as shown in Figure 5.1.

The settable fields include:

Name / Type The name or type of the monster.

Spec. Attacks The special attacks the monster is capable of.

Alignment The monster's alignment.

Spec. Defenses The special defenses the monster is capable of.

Treas. Type The type of treasure the monster has or is guarding.

Psionics The monster's Psionic capabilities.

Hit Points The monster's hit points. This applies to special monsters, ones that are unique or exist in very small numbers. Most monsters have hit dice instead.

Hit Dice The monster's hit dice.

Armor Class The monster's armor class.

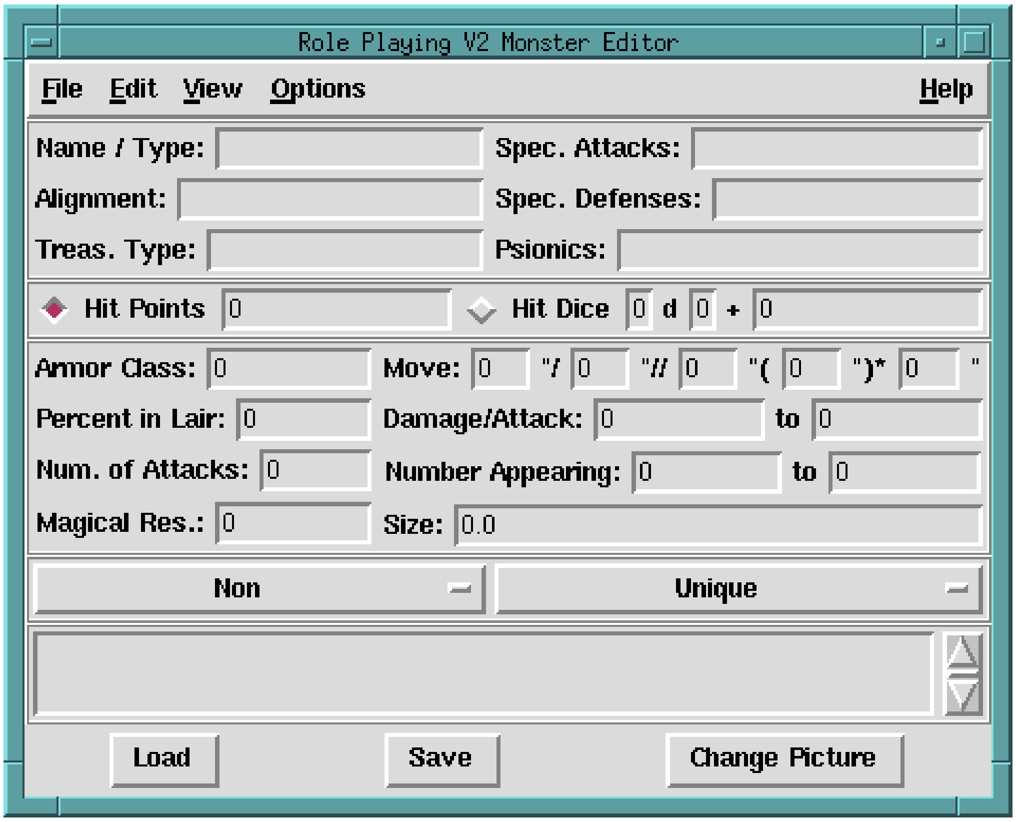

Figure 5.1: Monster Data Object Editor GUI Window

Move The monster's movement speed. This includes on land, in the water, flying, burrowing, and web speed.

Percent in Lair Probability that the monster is in its lair.

Damage / Attack The amount of damage the monster can inflict for each attack.

Num. of Attacks The number of attacks the monster can make in each melee round.

Number Appearing The number of this type of monster that will appear at any given time.

Magical Res. The monster's resistance to magical attacks.

Size The monster's size, in feet.

Intelligence The monster's intelligence rating:

- Non – Not intelligent, this is an intelligence score of 0.
- Animal – Animal intelligence, this is an intelligence score of 1.
- Semi – Semi-intelligent, this is an intelligence score of 2-4.
- Low – Low intelligence, this is an intelligence score of 5-7.
- Average – Average (Human) intelligence, this is an intelligence score of 8-10.
- Very – Very intelligent, this is an intelligence score of 11-12.
- Highly – Highly intelligent, this is an intelligence score of 13-14.
- Exceptionally – Exceptionally intelligent, this is an intelligence score of 15-16.
- Genius – Genius, this is an intelligence score of 17-18.
- SupraGenius – Supra-Genius, this is an intelligence score of 19-20.
- Godlike – Godlike intelligence, this is an intelligence score of 21 or more.

Frequency The monster's frequency rating:

- Unique – This monster is unique, only one exists.
- VeryRare – This monster is very rare, only a few exist; 4% probability of occurrence.
- Rare – This monster is rare, not many exist; 11% probability of occurrence.
- Uncommon – This monster is uncommon, some exist but are not seen often; 20% probability of occurrence.
- Common – This monster is common, they are all over the place; 65% probability of occurrence.

Description This is a text field for any additional descriptive commentary.

In addition to the data entry fields, there are buttons to load and save a Monster Data Object file. There also is a button to set or change the monster's picture (GIF file).

Chapter 6

Spell Data Object Editor GUI Window

The Spell Data Object represents magic spells. These spells can be “cast” or invoked by special character classes, Magic Users and Clerics. Different spells have different effects. Some are offensive, some are defensive, and others are informational.

The Spell Data Object Editor GUI Window contains entries to edit a Spell Data Object as shown in Figure 6.1.

The settable fields include:

Class This is the character class that can cast this spell.

Level The level of the spell.

Name The name of the spell.

Type The type of the spell.

Effected Area The area the spell has an effect in.

Range The range of the spell.

Casting Time How long it takes to cast the spell.

Saving Throw The saving throw needed to counteract the spell, if any.

Duration How long the spell lasts.

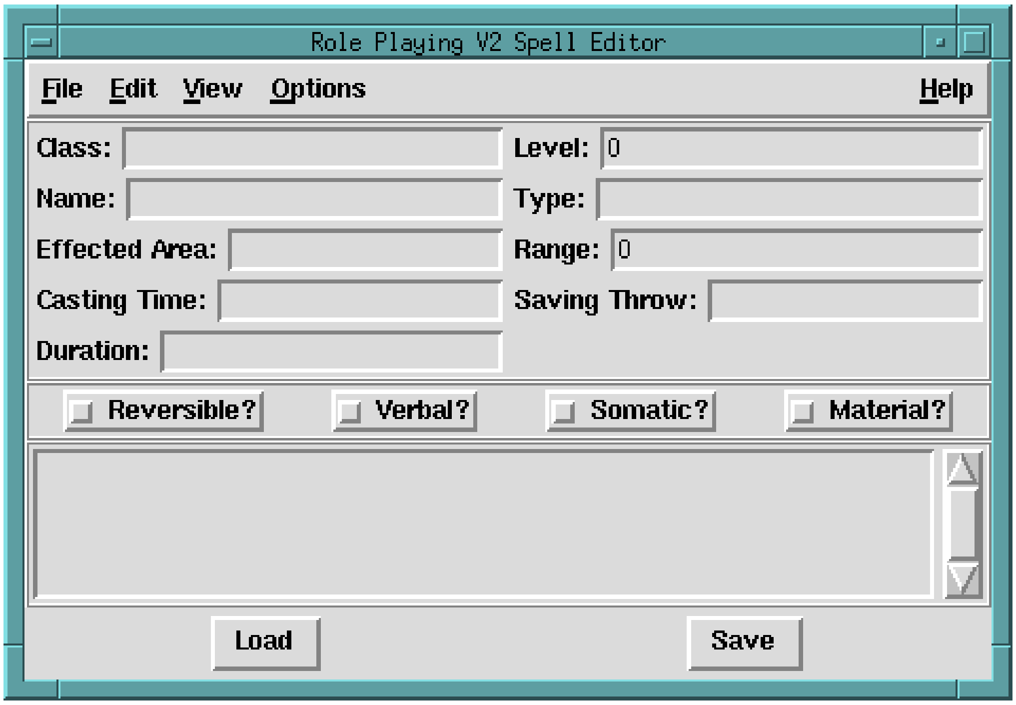

Figure 6.1: Spell Data Object Editor GUI Window

Reversible? A flag indicating if the spell is reversible.

Verbal? A flag indicating if the spell has a verbal component.

Somatic? A flag indicating if the spell has a somatic component.

Material? A flag indicating if the spell has a material component.

Description A text field for entering descriptive commentary about the spell.

In addition to the data entry fields, there are buttons to load and save the Spell Data Object to a disk file.

Chapter 7

Dressing Data Object Editor GUI Window

The Dressing Data Object contains information about the various things that might be scattered about in the game environment, such as furnishings and related items.

The Dressing Data Object Editor GUI Window contains entries to edit all of the settable fields in a Dressing Data Object as shown in Figure 7.1.

The settable fields include:

Name The name of the object.

Value The item value in gold pieces.

Description Descriptive text describing the item.

In addition to the data entry fields, there are buttons to load and save the data object and to set or change its picture (GIF file).

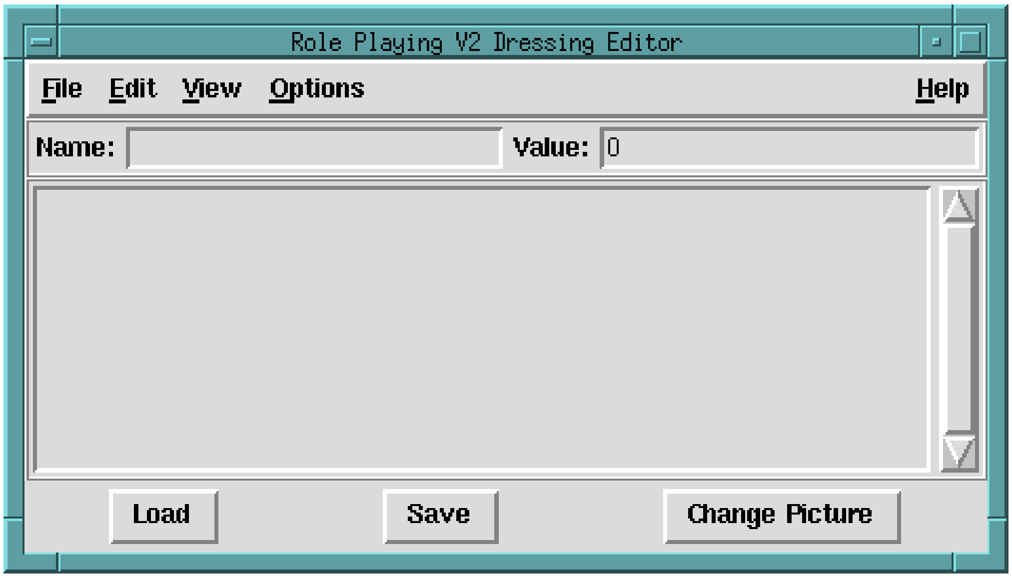

Figure 7.1: Dressing Data Object Editor GUI Window

Chapter 8

Trick / Trap Data Object Editor GUI Window

The TrickTrap Data Object contains information about tricks and traps that the game master has scattered about the playing environment. These tricks and traps generally protect treasures, but sometimes are just there to keep the players on their toes.

The Trick / Trap Data Object Editor GUI Window contains entries to edit all of the settable fields in a Trick / Trap Data Object as shown in Figure 8.1.

The settable fields include:

Name The name of the trick or trap.

Type The type of the trick or trap.

Description Descriptive text describing the trick or trap.

In addition to the data entry fields, there are buttons to load and save the data object and to set or change its picture (GIF file).

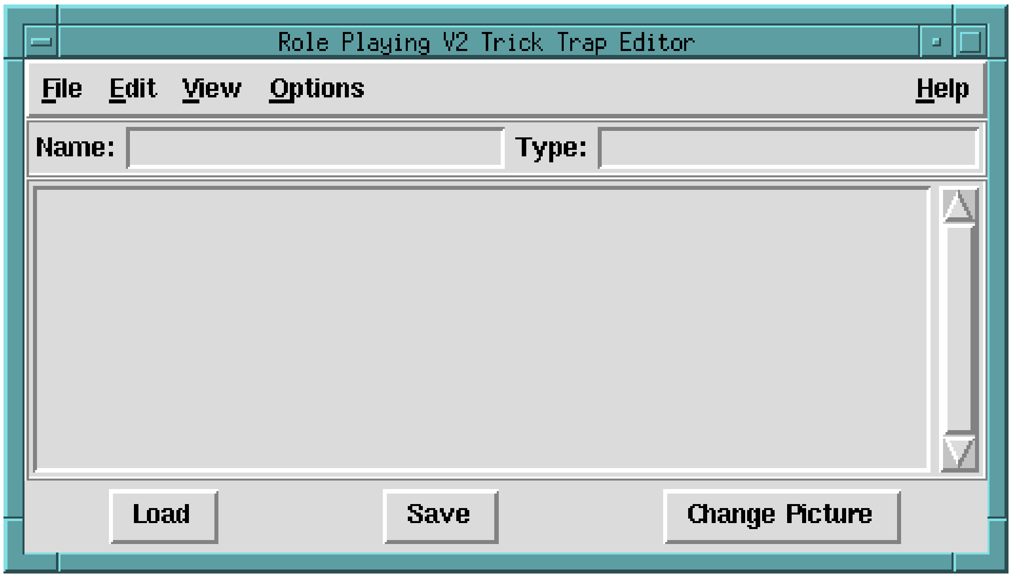

Figure 8.1: Trick / Trap Data Object Editor GUI Window

Chapter 9

Treasure Data Object Editor GUI Window

The Treasure Data Object contains information about treasures that the game master has scattered about the playing environment. These treasures are what the player characters are seeking. The treasures need not be gold or gems, but could be useful (or not so useful) magical items.

The Treasure Data Object GUI Window contains entries to edit all of the settable fields in a Treasure Data Object as shown in Figure 9.1.

The settable fields include:

Name The name of the treasure.

Strength +/− The strength adjustment the treasure affords, if any.

Damage Protection +/− The damage protection adjustment the treasure affords, if any.

Weight The weight, in gold pieces, of the treasure.

Intelligence +/− The intelligence adjustment the treasure affords, if any.

Movement +/− The movement (land speed) adjustment the treasure affords, if any.

Armor Class +/− The armor class adjustment the treasure affords, if any.

Wisdom +/− The wisdom adjustment the treasure affords, if any.

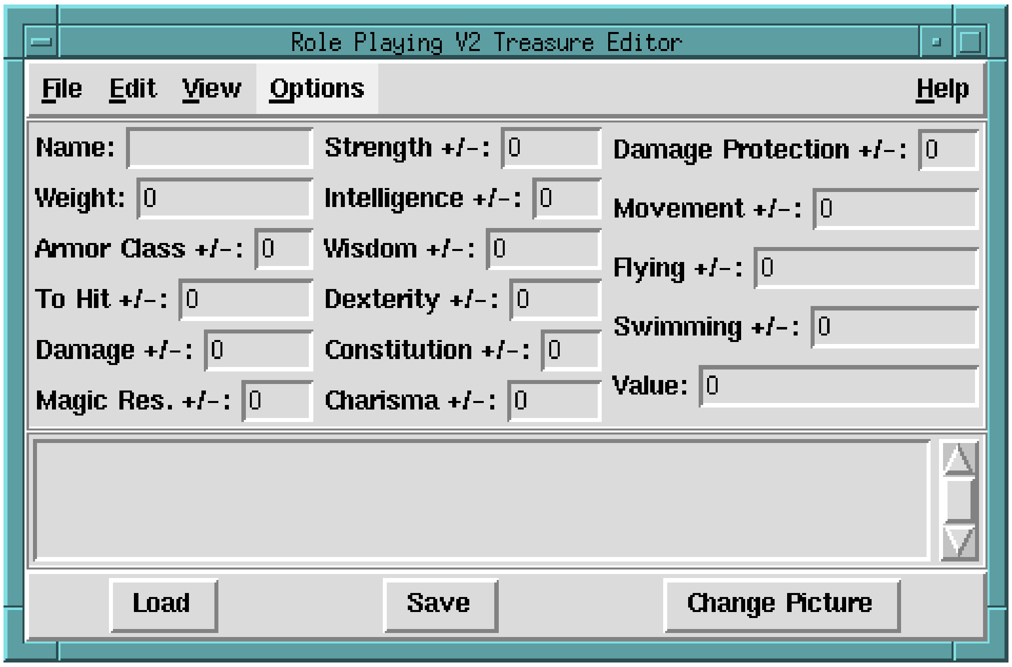

Figure 9.1: Treasure Data Object Editor GUI Window

Flying +/− The flying speed adjustment the treasure affords, if any.

To Hit +/− The to hit adjustment the treasure affords, if any.

Dexterity +/− The dexterity adjustment the treasure affords, if any.

Swimming +/− The swimming speed adjustment the treasure affords, if any.

Damage +/− The damage adjustment the treasure affords, if any.

Constitution +/− The constitution adjustment the treasure affords, if any.

Value The treasure's value in gold pieces.

Magic Res. +/− The magical attack resistance adjustment the treasure affords, if any.

Charisma +/− The charisma adjustment the treasure affords, if any.

Description Descriptive commentary about the treasure.

In addition to these data entry fields, there are buttons to load and save a Treasure Data Object and to change its picture (GIF file).

Chapter 10

Map Editor GUI Window

Map files contain information about the geographical environment in which the role playing game takes place. A map can have a number of levels, each of which can have a number of spaces. The spaces can be either squares or hexagons. All of the spaces in a given map must be the same shape.

10.1 Create Map Dialog

When a new map is created, a Create Map Dialog box appears as shown in Figure 10.1. This dialog box contains an entry for the name of the map, buttons to select the shape of the spaces, a text entry area for the descriptive commentary of the map, and entries for the X and Y extents of the map.

10.2 Map Editor GUI Window

The Map Editor GUI Window, shown in Figure 10.2, contains a window showing the spaces on the current level, an entry for the map's name, a scale indicating the current level, a text area for the map's descriptive commentary, and buttons for creating and deleting spaces.

The spaces in the window showing the spaces are sensitive to mouse clicks. One click selects the space, which outlines the space in red and displays the space's name next to the `Space Delete` button. This button can be used to delete the space. Two clicks pops up the Space Data Object Editor GUI Window, as shown in Figure 11.2 and described in Chapter 11. A new space

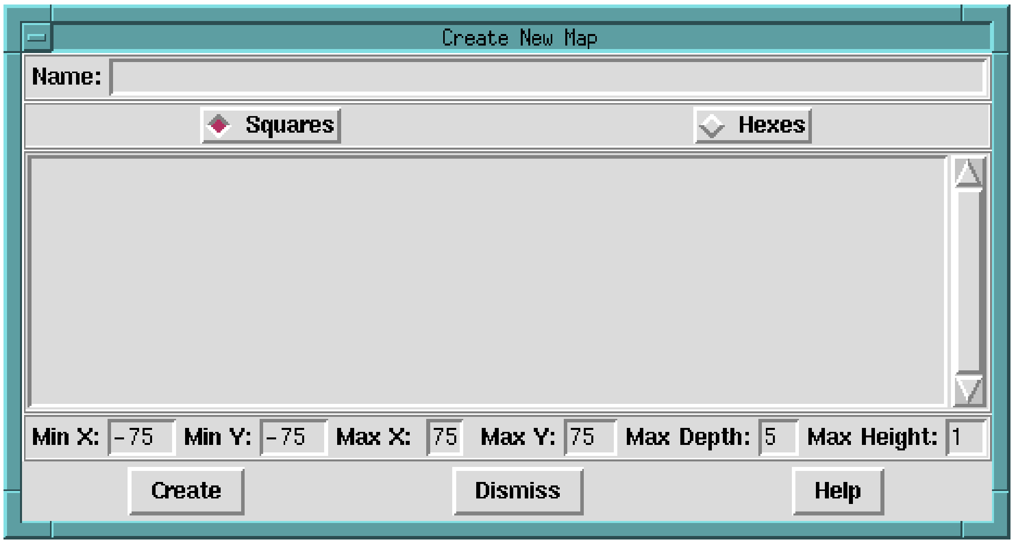

Figure 10.1: Create Map Dialog

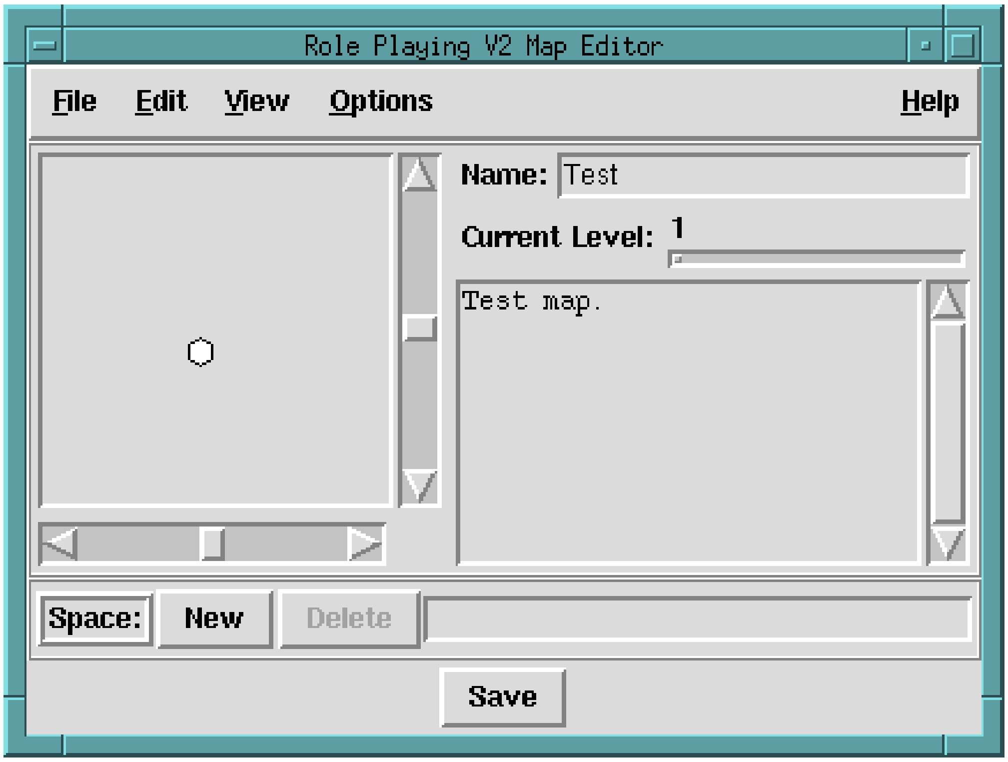

Figure 10.2: Map Editor GUI Window

can be added with the `New` space button. This button brings up the Create New Space dialog box, as shown in Figure 11.1 and described in Chapter 11.

Chapter 11

Space Data Object Editor GUI Window

11.1 Create New Space

When the new space button on the Map Editor GUI Window (see Chapter 10, Figure 10.2) is clicked, the Create New Space dialog box appears, as shown in Figure 11.1.

This dialog box contains entries for the location of the center of the space, the name of the space, and the background color.

11.2 Space Data Object Editor GUI Window

The Space Data Object Editor GUI Window, shown in Figure 11.2, contains a text entry area for descriptive text, and a graphic area showing the space and its contents – Exits and Items (Characters, Monsters, Treasures, Tricks / Traps, or Dressings). A set of crosshair lines is drawn on the graphic area, along with the coordinates of the intersection of the crosshairs.

To the right of the graphic area is a pair of lists (Exits and Items) with buttons. The upper one lists exits; the buttons are for adding and deleting exits. The lower one lists items; the buttons are for adding and deleting items.

Each exit and item is displayed in the space graphic area. Each exit and item responds to the left mouse button by selecting the corresponding list

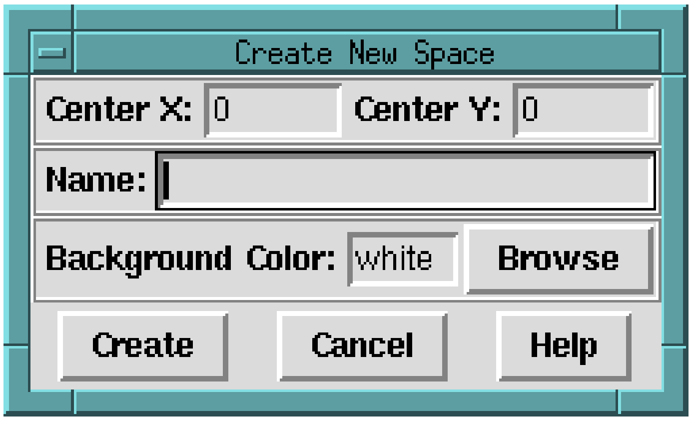

Figure 11.1: Create New Space

item. The right mouse button[1] brings up an Exit Information dialog box (described in Section 11.3) for exits and a GUI editor for items.

The `Add` exit button brings up the Creating New Exit dialog box, described in Section 11.4. The `Add` item button brings up the Creating New Item dialog box, described in Section 11.5.

11.3 Exit Information Dialog Box

The Exit Information Dialog box, shown in Figure 11.3, displays the information about the selected exit – the type, the X and Y position, the description, the image, and the next space index.

11.4 Creating New Exit Dialog Box

The Creating New Exit dialog box, shown in Figure 11.4, collects the information needed to create a new exit point from one space to another space. The data fields consist of the X and Y position, the wall alignment flag, the image of the exit, descriptive text, and the index of the next space. There is also a button to compute the next space[2].

[1] Command plus button on the Mac.

[2] This button does an approximate computation. Most of the time it should be correct, but there are some conditions that might confuse it. You should check the result and make whatever necessary corrections.

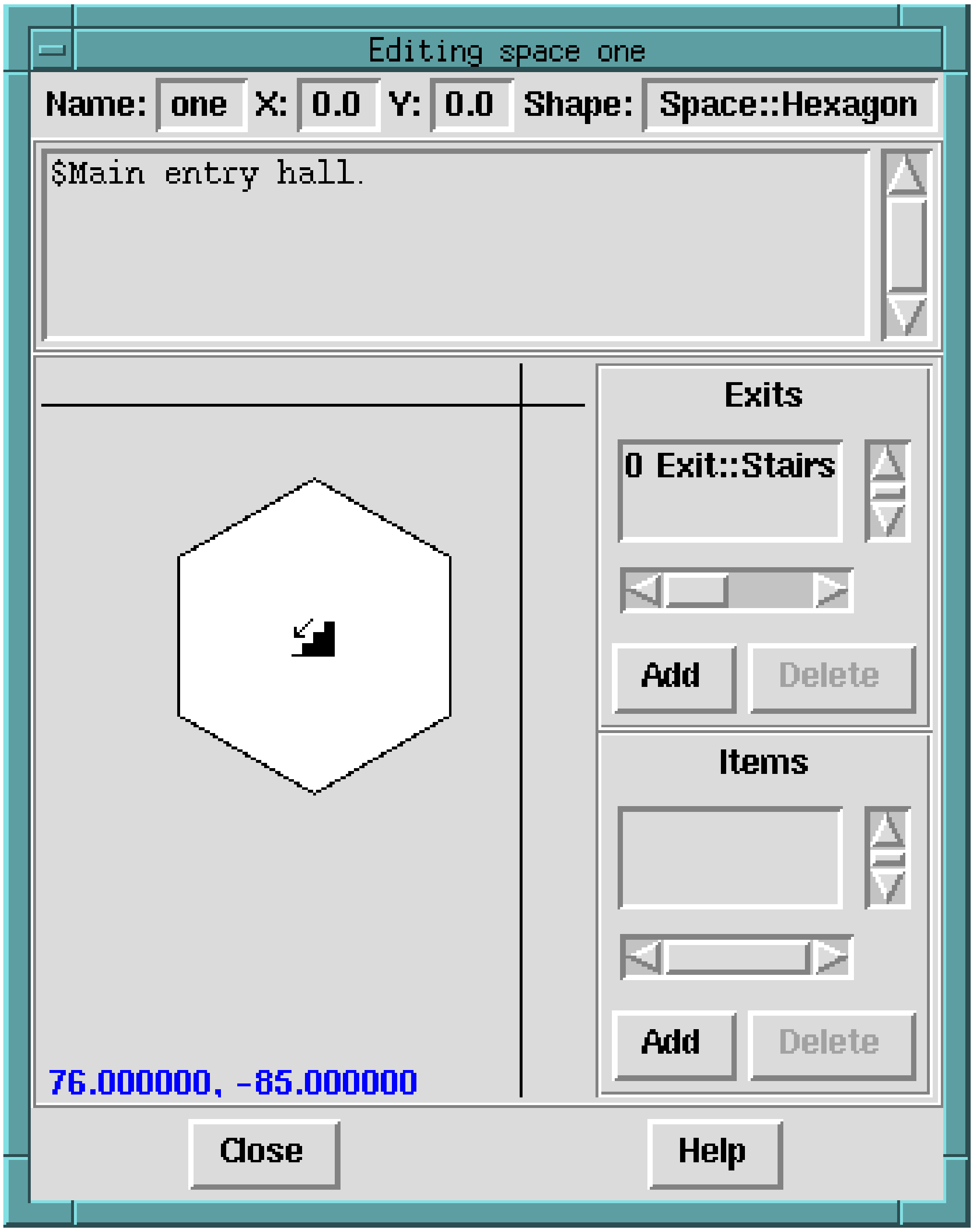

Figure 11.2: Space Data Object Editor GUI Window

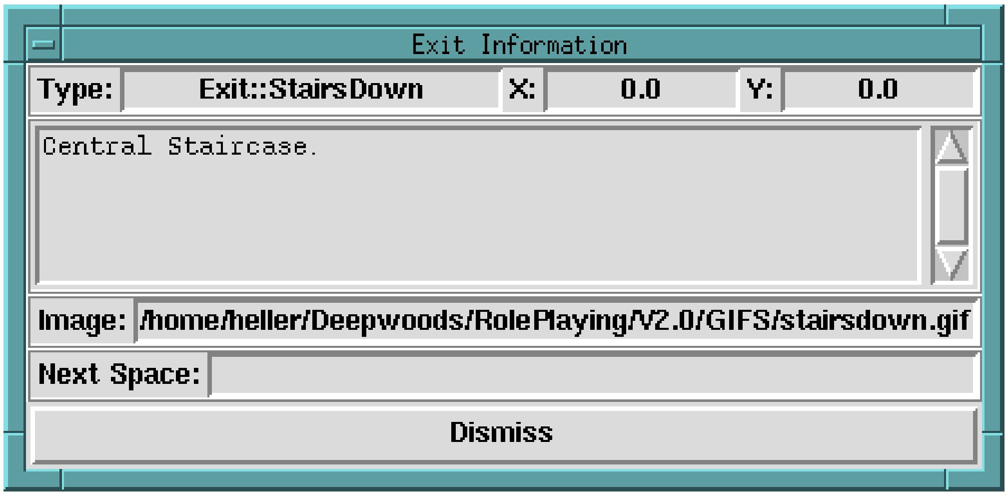

Figure 11.3: Exit Information GUI Window

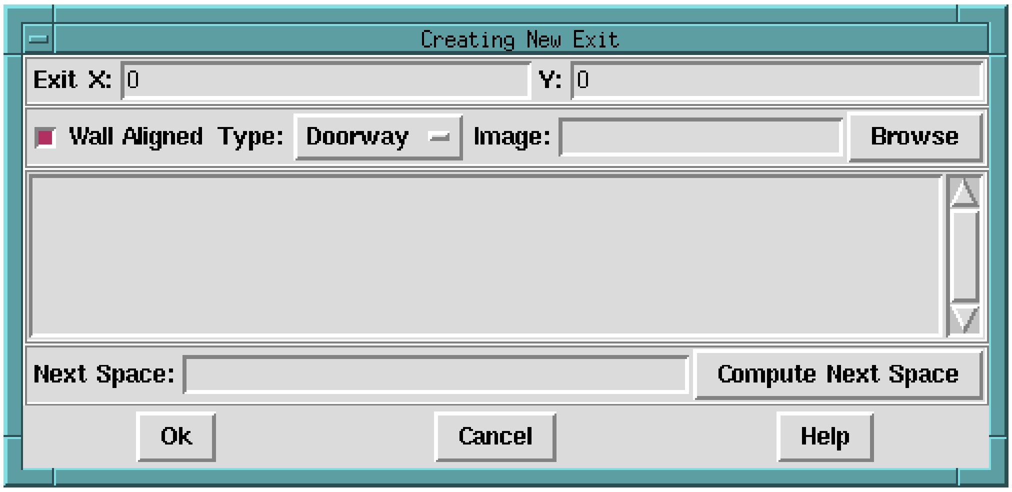

Figure 11.4: Creating New Exit dialog box

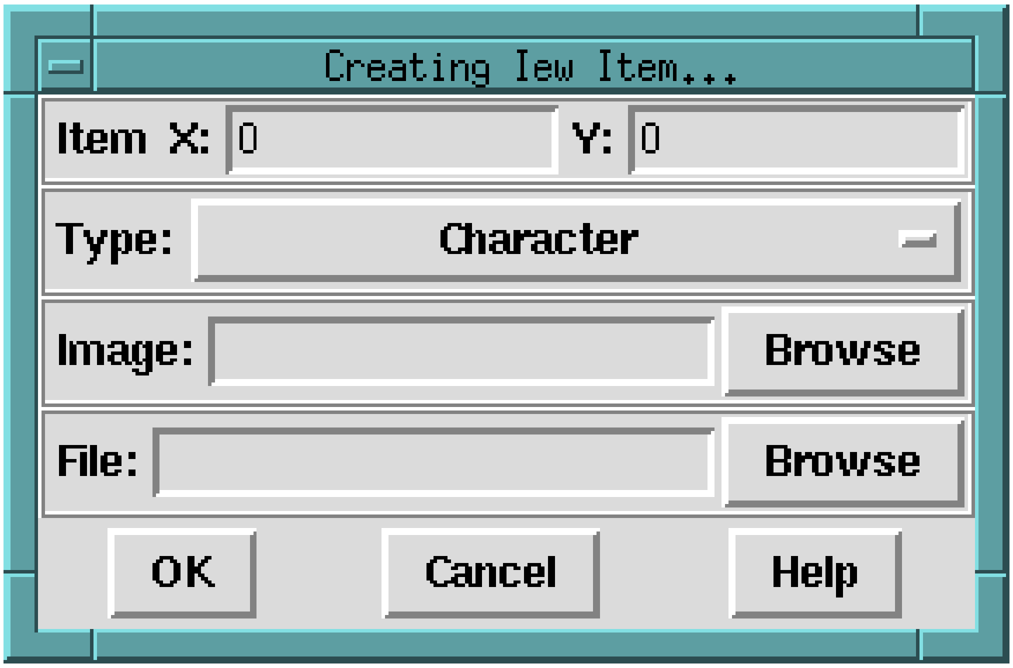

Figure 11.5: Creating New Item dialog box

11.5 Creating New Item Dialog Box

The Creating New Item dialog box, shown in Figure 11.5, collects the information needed to create a new item. The data fields consist of the X and Y position, the type of the item, the image of the item, and the data file of the item. The type is one of the types listed in Table 11.1.

Type	Description
Character	The item is a non-playing character (for example a barkeeper).
Monster	The item is a monster – this is its lair or it is guarding something.
Treasure	The item is a treasure.
TrickTrap	The item is a trick or a trap.
Dressing	The item is some random dressing.

Table 11.1: Item Types.

Appendix A

Data File Formats

A.1 Class Data Object Record Files

Except for the Map Data files, all of the other data files are serializations of C++ class structures. These files are read and written using Tcl's file I/O functions from internal Record class structure members in each of the class structures. Each of these records consists of a header key string starting with an asterisk (*) followed by the name of the class as a NUL terminated string. The rest of the record is a sequence of NUL terminated strings that start with a key character, a percentage character (%), and a data value string. The record is terminated with a line-feed character and a NUL byte.

A.1.1 Character Data Object Files

The Character Data Object File contains the key characters listed in Table A.1. The file starts with the key string `*Character`.

A.1.2 Monster Data Object Files

The Monster Data Object File contains the key characters listed in Table A.2. The file starts with the key string `*Monster`.

A.1.3 Spell Data Object Files

The Spell Data Object File contains the key characters listed in Table A.3. The file starts with the key string `*Spell`.

Char	Value	Type of data
s	Strength	Decimal integer string
i	Intelligence	Decimal integer string
w	Wisdom	Decimal integer string
d	Dexterity	Decimal integer string
c	Constitution	Decimal integer string
h	Charisma	Decimal integer string
e	Exceptional Strength	Decimal integer string
l	Level	Decimal integer string
D	Number Hit Dice	Decimal integer string
N	Hit Die Sides	Decimal integer string
G	Gold Pieces	Decimal integer string
n	Character Name	Text string
p	Player Name	Text string
r	Character Race	Text string
C	Character Class	Text string
a	Character Alignment	Text string
S	Character Gender	Text string
A	Character Age	Decimal integer string
T	Commentary	Text string
I	Image	Text string, a GIF file name

Table A.1: Key Characters used in Character Data Object Files

Char	Value	Type of data
n	Monster Name	Text string
a	Monster Alignment	Text string
t	Treasure Type	Text string
x	Special Attacks	Text string
d	Special Defenses	Text string
p	Psionics	Text string
T	Commentary	Text string
I	Image	Text string, a GIF file name
H	Hit Points	Decimal integer string
D	Number Hit Dice	Decimal integer string
N	Hit Die Sides	Decimal integer string
A	Hit Point Adjustment	Decimal integer string
X	Armor Class	Decimal integer string
m	Land Speed	Decimal integer string
f	Flying Speed	Decimal integer string
s	Swimming Speed	Decimal integer string
b	Burrowing Speed	Decimal integer string
w	Web Speed	Decimal integer string
l	Percent in Lair	Decimal integer string
u	Number of Attacks	Decimal integer string
M	Magical Resistance	Decimal integer string
+	Minimum Damage per Attack	Decimal integer string
−	Maximum Damage per Attack	Decimal integer string
!	Minimum number appearing	Decimal integer string
@	Maximum number appearing	Decimal integer string
i	Intelligence Rating	Text string (rating string)
q	Frequency	Text string (frequency class)
h	Hit Type	Text string ("Points" or "Dice")
S	Size	Decimal floating point string

Table A.2: Key Characters used in Monster Data Object Files

Char	Value	Type of data
l	Level	Decimal integer string
R	Range	Decimal integer string
F	Flag bits	Two hexadecimal characters
n	Spell Name	Text string
C	Spell Class	Text string
t	Spell Type	Text string
d	Spell Description	Text string
a	Spell Area of Effect	Text string
T	Spell Casting Time	Text string
D	Spell Duration	Text string
s	Saving Throw	Text string

Table A.3: Key Characters used in Spell Data Object Files

A.1.4 Treasure Data Object Files

The Treasure Data Object File contains the key characters listed in Table A.4. The file starts with the key string `*Treasure`.

A.1.5 TrickTrap Data Object Files

The TrickTrap Data Object File contains the key characters listed in Table A.5. The file starts with the key string `*TrickTrap`.

A.1.6 Dressing Data Object Files

The Dressing Data Object File contains the key characters listed in Table A.6. The file starts with the key string `*Dressing`.

A.1.7 Space Data Object Files

The Space Data Object File contains the key characters listed in Table A.7. The file starts with the key string `*Space`.

Char	Value	Type of data
W	Weight	Decimal integer string
A	Armor Class Adjustment	Decimal integer string
T	To Hit Adjustment	Decimal integer string
D	Damage Adjustment	Decimal integer string
M	Magical Resistance Adjustment	Decimal integer string
P	Damage Protection Adjustment	Decimal integer string
s	Strength Adjustment	Decimal integer string
i	Intelligence Adjustment	Decimal integer string
w	Wisdom Adjustment	Decimal integer string
d	Dexterity Adjustment	Decimal integer string
c	Constitution Adjustment	Decimal integer string
h	Charisma Adjustment	Decimal integer string
g	Land Speed Adjustment	Decimal integer string
f	Flying Speed Adjustment	Decimal integer string
S	Swimming Adjustment	Decimal integer string
v	Value	Decimal integer string
n	Treasure Name	Text string
C	Description	Text string
I	Image	Text string, a GIF file name

Table A.4: Key Characters used in Treasure Data Object Files

Char	Value	Type of data
n	Trick or Trap Name	Text string
T	Trick or Trap Type	Text string
C	Description	Text string
I	Image	Text string, a GIF file name

Table A.5: Key Characters used in TrickTrap Data Object Files

Char	Value	Type of data
n	Dressing Name	Text string
v	Value	Decimal integer string
C	Description	Text string
I	Image	Text string, a GIF file name

Table A.6: Key Characters used in Dressing Data Object Files

Char	Value	Type of data
x	Center X coordinate	Decimal floating point string
y	Center Y coordinate	Decimal floating point string
s	Shape Type	Text string (“Square” or “Hexagon”)
n	Space Name	Text string
C	Space Description	Text string
b	Background Color	Text string (color name)
e	Exit Element	Text string (exit description string, see Figure A.1)
i	Item Element	Text string (item description string, see Figure A.2)

Table A.7: Key Characters used in Space Data Object Files

```
%ex.xx\0y.yy\0A\0type\0descr\0image\0nextspi\0\r\0
```

Where x.xx and y.yy is the relative offset of the exit, A is the alignment flag and is either a T or an F, type is the type of exit (listed in Table A.8), descr is the description, image is the name of a GIF file to display for the exit, and nextspi is the next space index string.

Figure A.1: Exit Description String

Type	Description
Doorway	Open doorway
Door	Normal door
LockedDoor	Locked door
SecretDoor	Secret (hidden) door
OnewayDoor	One way door
TrapDoorUp	Trap door in ceiling
TrapDoorDown	Trap door in floor
StairsUp	Ascending stairs
StairsDown	Descending stairs
WindowUnglazed	Unglazed window
WindowGlazed	Glazed window
Chimney	Chimney or hole in the ceiling
Pit	Pit or hole in the floor

Table A.8: Exit Types

```
%ix.xx\0y.yy\0type\0image\0filename\0\r\0
```

Where x.xx and y.yy is the relative offset of the item, type is the type of the item (Character, Monster, Treasure, Trick / Trap, or Dressing), image is the name of a GIF file to display for the item, and filename is the name of the file with the full description of the item.

Figure A.2: Item Description String

A.2 Map Data Files

Map data files are written directly from Tcl – there is no C++ class structure for maps. Maps are implemented using a Tcl "array" variable. Tcl "arrays" are really associative tables implemented with a hash table.

A Map data file contains a set of name value pairs as shown in Table A.9.

Name	Value
shape	The shape of the spaces, either Space::Hexagon or Space::Square.
name	The name of the map.
description	The description of the map.
deepestLevel	The level deepest in the ground (most positive numerical value).
highestLevel	Highest level at or above the ground (0 = ground level, negative values are above ground).
maxX	Maximum (rightmost or “easternmost”) X coordinate.
maxY	Maximum (bottommost or “southernmost”) Y coordinate.
minX	Minimum (leftmost or “westernmost”) X coordinate.
minY	Minimum (topmost or “northernmost”) Y coordinate.
Spaces,l,x,y	Filename of the Space Data Object at level l at x,y.

Table A.9: Name Value Pairs in a Map Data File.

Appendix B

License

B.1 Copyright

B.2 License.

B.2.1 License to use

The software package known as the *Role Playing DataBase System* package is licensed to anyone to use at no charge or fee, on any number of machines, by any number of users.

B.2.2 License to copy

The *Role Playing DataBase System* package may be freely copied, **SO LONG AS THE COMPLETE PACKAGE OF THE SOURCE CODE, SCRIPTS AND THE DOCUMENTATION FILES ARE COPIED TOGETHER AS A UNIT**[1]. No charge or fee may be demanded for a copy of this pack-

[1] The binary executable may also be copied instead of the sources, so long as the source code is also available from the same source (media, ftp site, etc.)

age[2].

B.2.3 No Support

Unless the package is registered, no customer support of any sort is provided.

B.3 No Warranty

BECAUSE THE *Role Playing DataBase System* PACKAGE IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PACKAGE, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PACKAGE "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PACKAGE IS WITH YOU, THE USER. SHOULD THE PACKAGE PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY REDISTRIBUTE THE PACKAGE AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF USE OF THE USE OR INABILITY TO USE THE PACKAGE (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PACKAGE TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

[2] A modest media or transmission charge or fee is allowed. This charge or fee should not exceed the cost of the media or the cost of the transmission.

Appendix C

Registration: Why Bother?

C.1 What Is Shareware?

Shareware is software that is distributed freely, but with a voluntary fee. Generally, this fee gets some extra benefits. In some cases, source code is provided. In others a nice manual and/or customer support is provided. Sometimes the freely distributed version has limited functionality and registration provides a fully functional version of the package or program.

C.2 What You Get When You Register

The *Role Playing DataBase System* package is fully functional as distributed freely. You don't need to register it to make full use of the package. There are advantages to registering your copy of the *Role Playing DataBase System* package.

When you fill in the registration form (see Appendix D) and send in your registration fee you will get customer support. This means you can ask me questions and I will try to answer them and try to help you with whatever problems you might have. It also means I will really listen to your suggestions and comments.

C.3 If You Decide Not To Register

This is okay. You are entirely on your own if you have problems or questions.

C.4 Why Not Simply Make It a Commercial Package?

Producing a slick, shrink-wrapped commercial package requires a certain amount of investment in time and resources. It would make the package more costly to the user (much more than the registration fee) and would put me in debt. If it did not sell I could be stuck with a large debt. It is possible, if this package is extremely popular, that a commercial version might be produced. As a shareware package, the *Role Playing DataBase System* package can be distributed widely at low cost. A lot of people can get a chance to try it out without having to pay. If the package proves to be useful, people can choose to pay the modest registration fee and get customer support.

Appendix D

Registration Form

This is the Shareware Registration Form for the *Role Playing DataBase System* package. Copy this form and mail it with \$50.00[1] to:

Robert Heller
D/B/A Deepwoods Software
51 Locke Hill Road
Wendell, MA 01379-9728
USA

[1] Check or money order, in USA funds, made payable to "Robert Heller".

Figure D.1: Registration Form

Please answer all 5 of the following questions:

1. Your name: ______________________________

2. Your mailing address: ______________________________
____
____

3. Your e-mail address(es): ______________________________
____
____

4. Which operating system are you using?

4a. MS-Windows: 3.11 ☐ '95 ☐ '98 ☐ 'NT ☐.

4b. MacOS: System 7 ☐ MacOS 8.x ☐. 68K ☐ PPC ☐.

4c UNIX: Linux ☐ other UNIX ☐ Name of UNIX system: ______________.

5. Name of ftp site, BBS, or information service you downloaded the *Role Playing DataBase System* from: ______________________________.
Purchased CD-ROM: ☐.

These remaining questions are optional. You don't have to answer any of them, but please try to answer as many as you feel like. These answers will help me to make improvements in the *Role Playing DataBase System* package.

6. Which RPG system(s) do you use? ______________________________

7. What do you *like* about the user interface of *Role Playing DataBase System* ______________________________

8. What do you *not like* about the user interface of *Role Playing DataBase System*? ______________________________

Bibliography

[1] Dave Beazley. *Simplified Wrapper and Interface Generator*. Department of Computer Science, University of Utah, Salt Lake City, UT, 1998.

[2] Gary Gygax. *Players Handbook*. TSR Games, Lake Geneva, WI, 1978.

[3] Gary Gygax. *Dungeon Masters Guide*. TSR Games, Lake Geneva, WI, 1979.

[4] Robert Heller. *Role Playing Database, A Computerized Role Playing Database System, Tcl Internals*. Deepwoods Software, Wendell, MA, 1999.

[5] Robert Heller. *Role Playing DB V2.0 – Tcl Interface*. Deepwoods Software, Wendell, MA, 1999.

[6] Robert Heller. *RPG DB C++ Internals*. Deepwoods Software, Wendell, MA, 1999.

[7] John K. Ousterhout. *Tcl and the Tk Toolkit*. Addison-Wesley Publishing Company, Reading, MA, 1994.

Index