Chapter 16

Story Adventure Maker

In this chapter, we have gathered a number of tools and editors that will help you create data for your adventure games, and we have given these tools the name Story Adventure Maker.

These tools will form the basis of your own toolset and will help you speed up your adventure game development. These tools are aimed mainly at the text adventure and story adventure games.

Map Creator

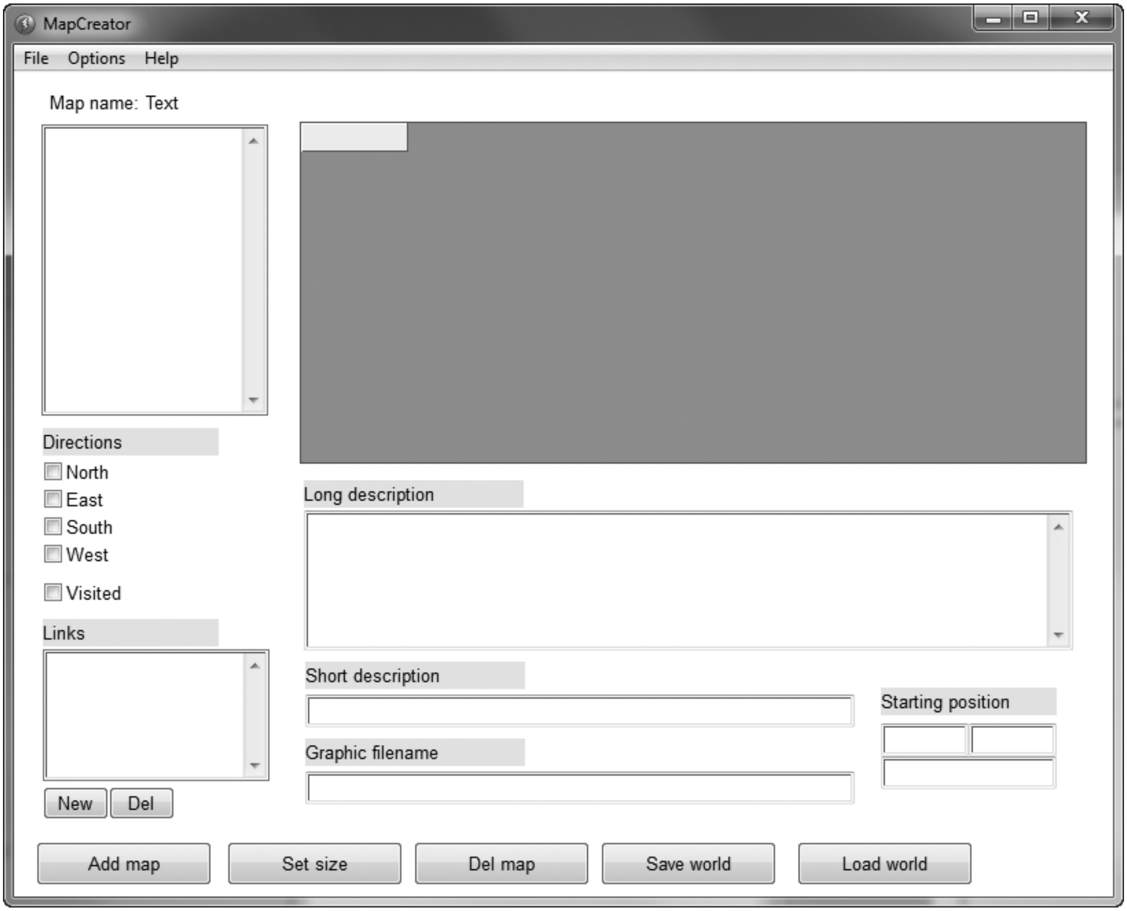
The Map Creator allows you to create a text/graphic adventure map using a grid system. The actual editor comes from the powerful brother of MMF2 called MMF2 Developer and allows us to utilize the advantages of the Data Grid object. But there are other methods, such as using active boxes to represent the grid, that you can potentially use if you only had MMF2 standard.

You can see an example of the Map Creator in Figure 16.1.

The Map Creator allows you to save and load map-based information into a single easy-to-read screen. Everything on this screen is loading and storing data into the Text Adventure Map object, which you can then save into a file. You would then take this file and load it directly into your game.

The Map Creator has the following features:

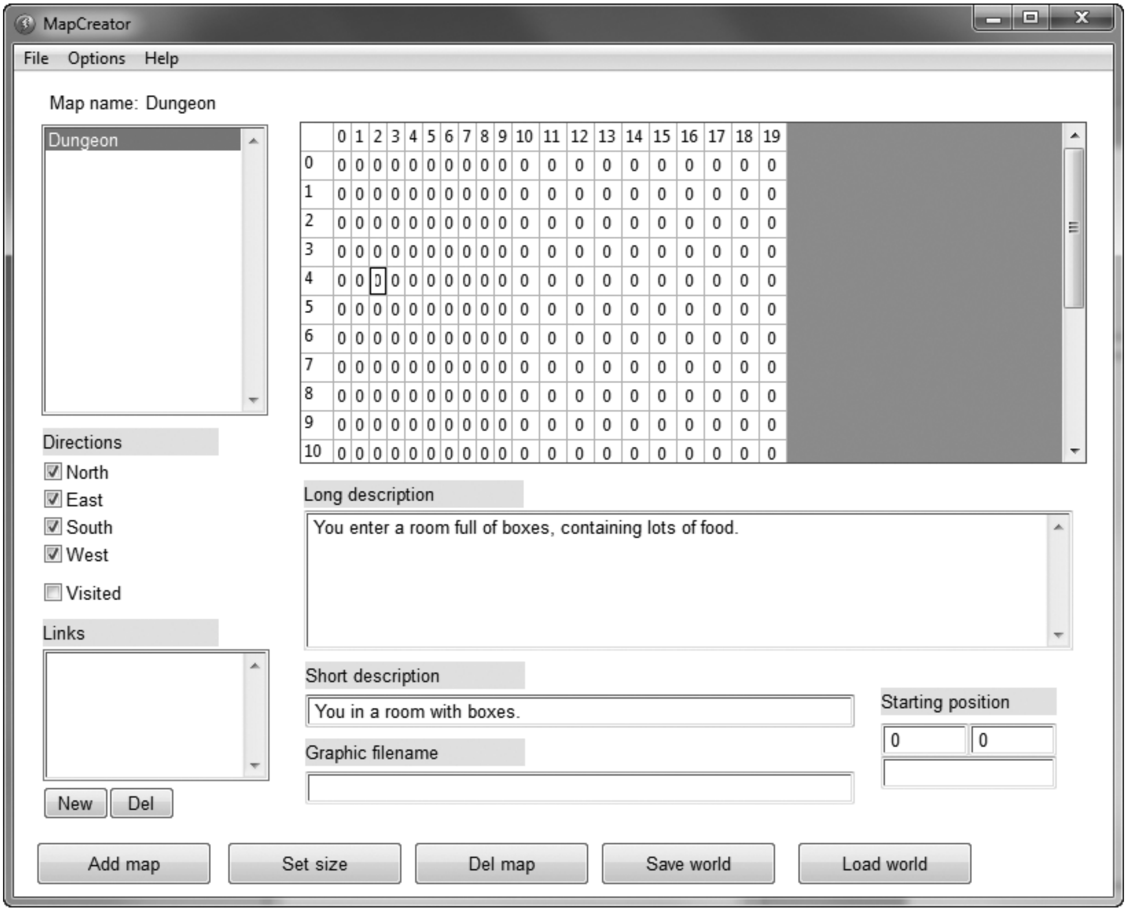
- **Maps:** When you add a map, you enter the name and it will appear in a list. You need to highlight the map name to see the details.



Clickteam

Figure 16.1
The Map Creator editor.

- **Directions:** A set of check boxes that you can use to create links to other connected cells.
- **Visited:** A check box that lets you know if you have previously visited a cell.
- **Links:** Links to other cells in the same map or to a cell on another map.
- **Cells:** Using the Data Grid object, you can see each of the cells, select them, and configure them. You can see an example map with some data already entered in Figure 16.2.
- **Long Description:** The long description for the current room.
- **Short Description:** A short description for the current room.
- **Graphic Filename:** The filename of an image for this particular room.
- **Starting Position:** The player's starting position.
- **Add Map button:** An Add Map button to create a map.



Map name: Dungeon

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Directions

☑ North
☑ East
☑ South
☑ West

☐ Visited

Links

Long description

You enter a room full of boxes, containing lots of food.

Short description

You in a room with boxes.

Graphic filename

Starting position

0 0

Clickteam

Figure 16.2
The Map Creator with some details entered.

- **Set Size:** Change the default size of a map.
- **Del Map:** Delete the selected map.
- **Save World:** Save the current world data to a file. The world is saved with the file extension .taw.
- **Load World:** Load a world in from a file.

If you have downloaded the files for this book, you will find the Map Creator within the files, called MapCreator.exe.

Pages Editor

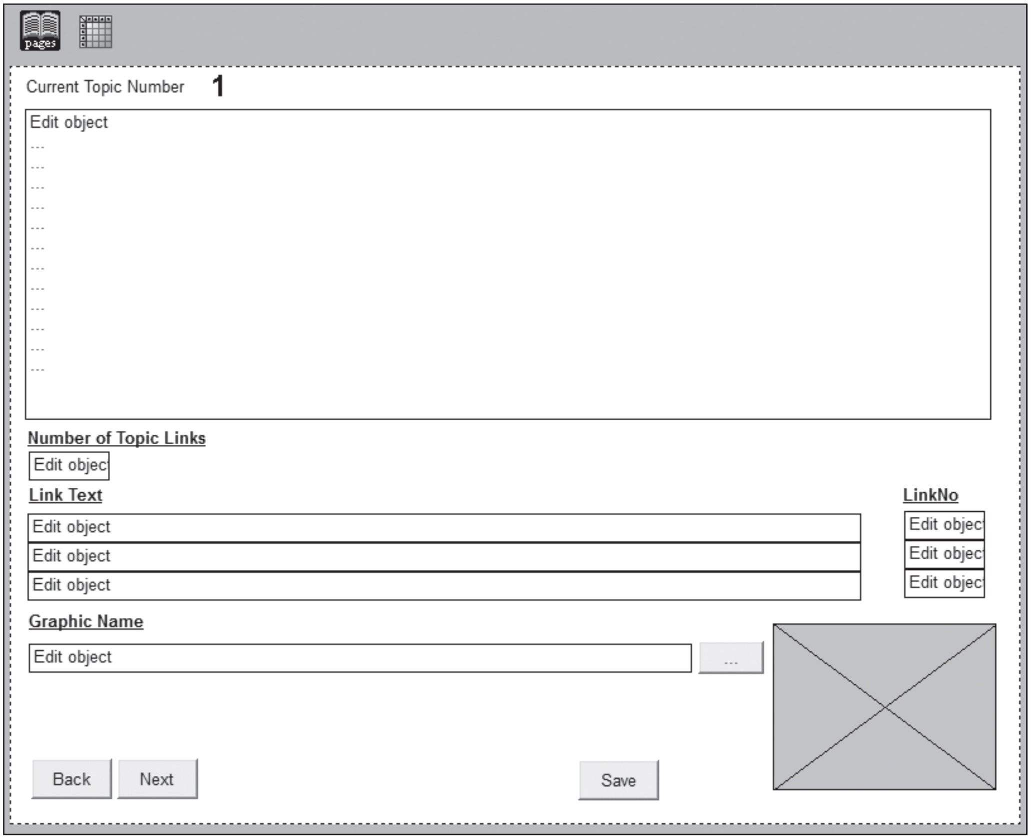
In Chapters 10 and 11, we looked at editors, saving and loading data, and in particular, we created the Pages editor to create the content for our own adventure story books/games.

We've made some changes to the original program to give it a little bit more functionality as well as increasing the size of the application to 800 × 600. By increasing the size of the application, we will be able to fit more content into the window. The other new functionality is the ability to assign an image to a particular topic; this saves the image name in the Text Adventure Pages object and then reloads it when the user clicks through the topic numbers.

We've also added the following new objects and features:

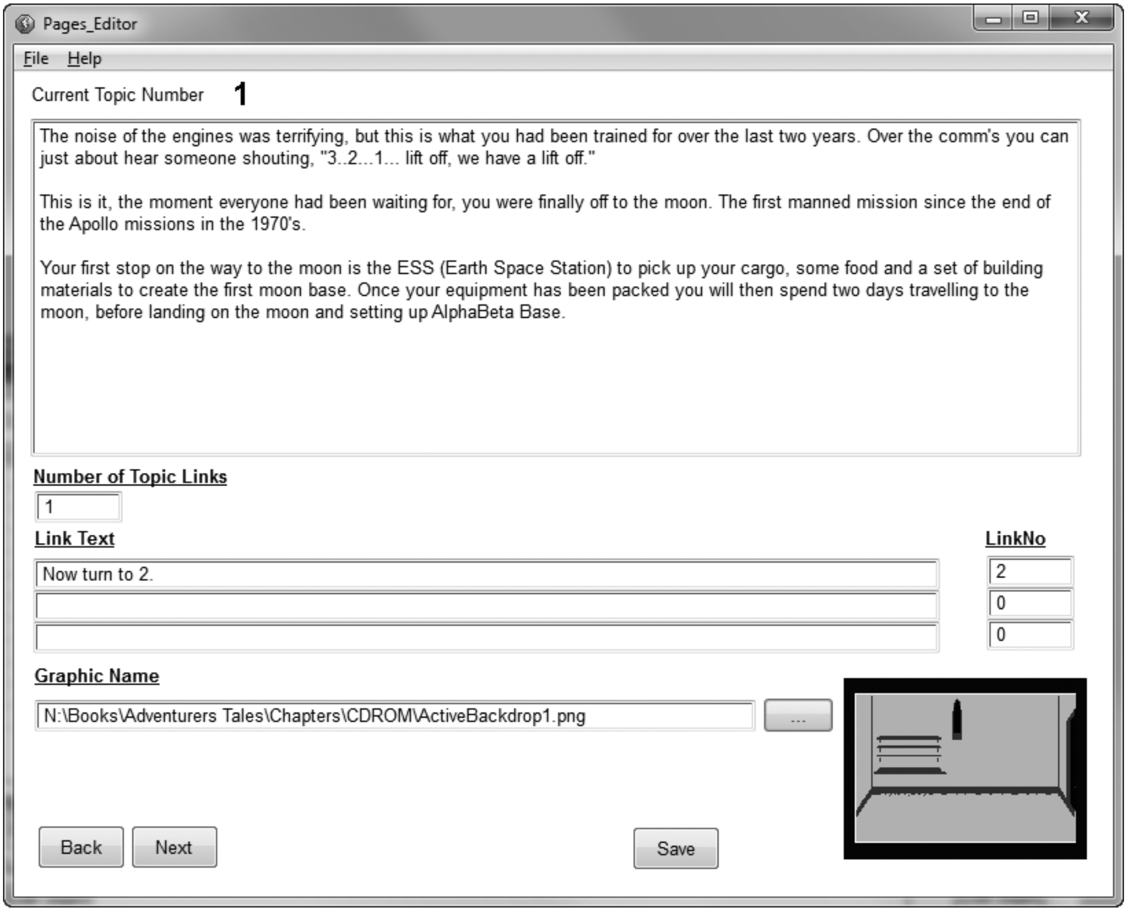
- **Graphic Name:** An edit box that is used to display the filename of an image that we wish to apply to a particular topic.
- **Button:** A button that will launch a dialog box (using the Picture object) so that you can browse for a particular file.
- **Picture Object:** A Picture object that will display the currently selected picture.

You can see the Frame editor for the improved editor in Figure 16.3 and what it looks like when running it in Figure 16.4.



Clickteam

Figure 16.3
The Frame editor for our updated Pages editor.



Current Topic Number 1

The noise of the engines was terrifying, but this is what you had been trained for over the last two years. Over the comm's you can just about hear someone shouting, "3..2...1... lift off, we have a lift off."

This is it, the moment everyone had been waiting for, you were finally off to the moon. The first manned mission since the end of the Apollo missions in the 1970's.

Your first stop on the way to the moon is the ESS (Earth Space Station) to pick up your cargo, some food and a set of building materials to create the first moon base. Once your equipment has been packed you will then spend two days travelling to the moon, before landing on the moon and setting up AlphaBeta Base.

Number of Topic Links: 1

Link Text	LinkNo
Now turn to 2.	2
	0
	0

Graphic Name: N:\Books\Adventurers Tales\Chapters\CDROM\ActiveBackdrop1.png

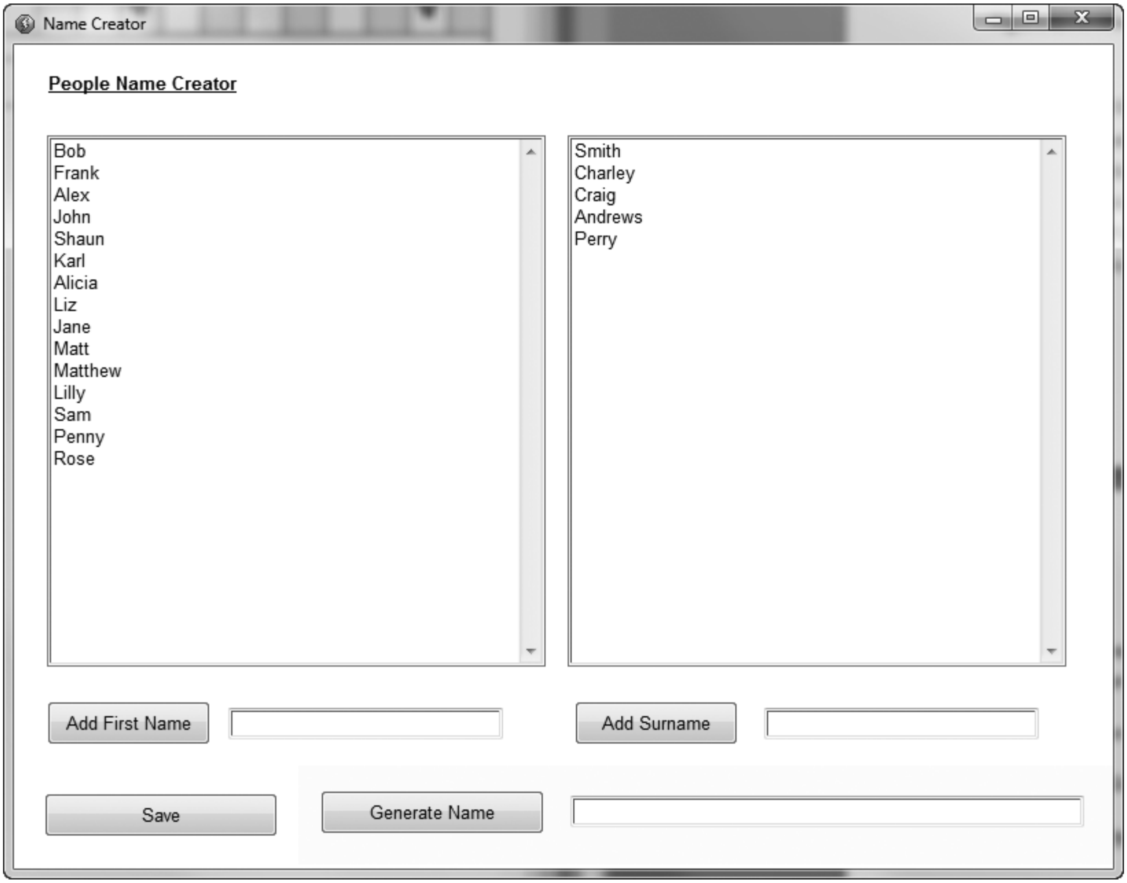
Clickteam. Game: Jason Darby

Figure 16.4
The Pages editor running.

Name Creator

The Name Creator is a simple program that will help you generate names for your characters in your games. Sometimes generating content, such as character names and locations, can be a bit of a chore. If you are making a game with lots of different characters, it can also be quite time-consuming to think of different names. With the Name Creator, you can enter many first names and surnames and then click on a button to generate a name randomly.

Using the same code and by changing the data, you can create a name location random generator system. This provides you with a way of creating fictional place names based on real places. You can see an example of the program in Figure 16.5.



Bob
Frank
Alex
John
Shaun
Karl
Alicia
Liz
Jane
Matt
Matthew
Lilly
Sam
Penny
Rose

Smith
Charley
Craig
Andrews
Perry

Clickteam

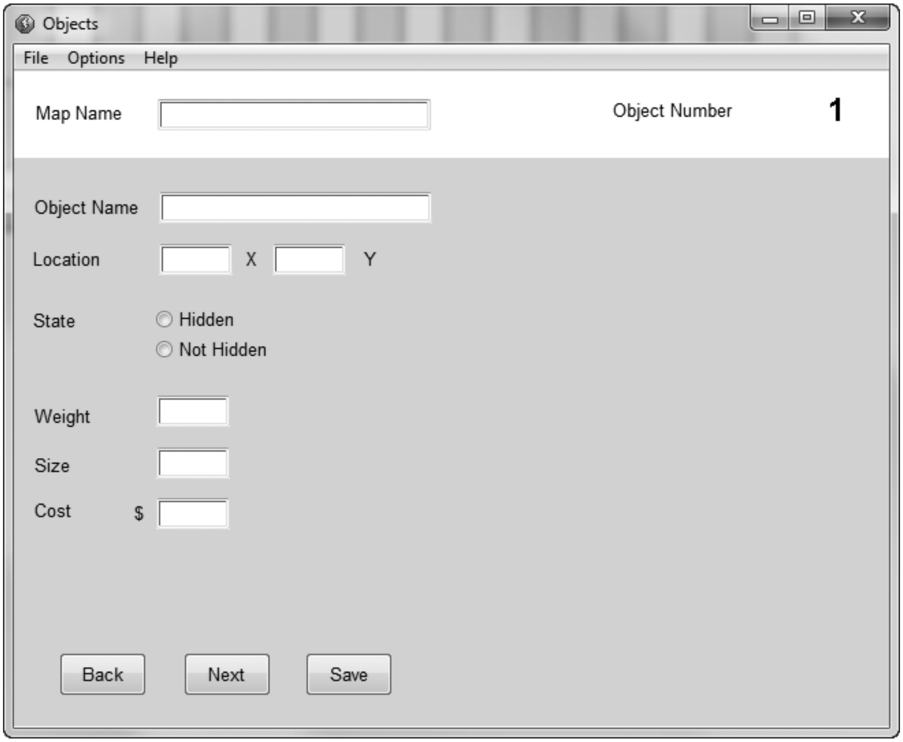
Figure 16.5
The Name Creator.

You can find the file called Namecreator.mfa in the downloaded files if you wish to view the source code or make additional changes and improvements.

Object System

This system uses the Text Adventure Map object and utilizes the object/inventory part of the object. In this editor, you will create the objects that are within your game and place them within a particular location.

All objects are assigned to a particular map, and the contents saved to a file that can then be loaded into a game. This editor follows the same principles as all of the other editors. By creating a way to enter and store data, you can load it directly into your game with only a little amount of coding work. You can make changes quickly and easily to your data without needing to change your code or, at the very least, not have



Map Name

Object Number 1

Object Name

Location X Y

State
○ Hidden
○ Not Hidden

Weight

Size

Cost $

Clickteam

Figure 16.6
The Objects System editor

to re-write large parts of your data, which you would normally have had to manually enter if you weren't using an editor.

You can see in Figure 16.6 the layout of the editor. It contains the following items and can access the code in the MMF2 file called Objects.mfa:

- **Map Name:** The Text Adventure Map objects key identifier is the map name, so we need to enter this so that we can store all objects to this particular map.
- **Object Number:** We are using a counter to keep track of each object, so that we can easily load in the data for the correct object number at the right time.
- **Object Name:** The name given to the object, for example, "Rusty sword" or "Apple."
- **Location:** When you create your map, you give it a grid size. Here is where you will enter the X and Y coordinates for the object.
- **State:** Is the object hidden or visible? This is important if you want the object to be hidden until the player has completed another task, such as searching the room, or solving a puzzle.

- **Weight:** What is the weight of the object? You can create a game that takes into account the total weight of all the objects, or checks if the object can be carried. You can consider adding RPG elements into your adventure game, such as strength, and if the player is strong enough, he can carry a particularly heavy object. You can also have a potion that a player drinks that will allow him to pick up heavy objects; you can then create some interesting puzzles based on this mechanic.
- **Size:** How big is the object, and is it too big to fit in the player's inventory?
- **Cost:** How expensive is the object, which allows you to consider a shop-based mechanic where the player can sell an object for money, or perhaps the adventure game uses the item's total value to test a player's success. Many early adventure games used turns/points to work out how well a player had played the game. You can create an adventure game where the player needs to find objects of value. The more valuable the objects, the better the player has played the game.
- **Buttons:** Three buttons allow you to move forward through the objects, another to move backward, and a final one to save the data to a file.

There are some additional features in the Text Adventure Map object that you can use to improve this editor, allowing the ability to add further values to an object. Perhaps you can add items such as object wear-and-tear or quality.

Additional Example Files

When we were making the Adventure Game objects for MMF2, we created a number of data-driven files that tested the code to ensure it worked correctly. These files have been included with the download files so that you can take a look at the code and see how it works, but also to get an idea how some of the data is handled.

These example files are more complex than those of the editors shown in this chapter, but they should be a good source of inspiration and knowledge of how the objects work behind the scenes.

The following files are available in the Examples folder:

- **Flags test:** This example shows how the flags system works, which is an extremely powerful system for identifying when things should be on/off or when using toggles.
- **Map test:** This is a powerful editor shown earlier in this chapter where you can quickly create the data for your map, including setting directions and room descriptions.

- **Messages test:** This example file shows how you can use messages within your games by using states and entering parameters to add words to a sentence.
- **Pages test:** A simple example showing the randomize pages (topics) system in the Text Adventure Pages object.
- **Pages:** Another example showing how you can use other objects to create topic links for the Text Adventure Pages object.
- **Words test:** A simple example of how to use different words to trigger text replies. In this example, you can examine a potion or lotion, and it will return a particular set of text.

Note

The Pages example is only available for the MMF2 Developer demo/full product because it uses the Data Grid object. You can download a demo of MMF2 Developer from the following website: www.clickteam.com.

Chapter Summary

In this chapter, we looked at some of the editors that have been written to help you generate data for your adventure games. Using and improving these editors or making your own will ensure you can quickly create your own adventure games. Speed is very important as this allows you to focus more time and effort on writing the story and creating your game's puzzles rather than spending time entering code.

Always make a note of how each of your editors works so that you can easily get this content into your game. Making comments in the source files and writing Word documents will help ensure that if you come back to an editor to make improvements at a later stage, you will remember what each part of the editor does.

In the next chapter, we will look at how you can compile your games into different formats.