

# Anatomy of Complex Encounter Editor

From time to time you may want to have some action that goes beyond the simple things that can take place in an Action Point. For that you need some type of encounter.

This section deals with the Complex Encounters. Next to the Rogue Encounter, this is one of the most difficult, to both understand and script, aspects of a Realmz scenario. I'm dreading the idea of having to explain this in words but here goes.

As you may have seen in Realmz, some encounters prompt you to do something with the Encounter Bar as in Figure 5.0. In Divinity these are referred to as Complex Encounters. A Complex Encounter can contain up to 5 separate types of actions that can be used to solve the puzzle. These are represented below this list in Figure 9.0:

- 1) Cast some sort of spell either from memory or from a scroll.
- 2) Use some item
- 3) Perform some physical Action
- 4) Use some special skill
- 5) Speak some special word or phrase

Figure 9.0

Each of these actions must be accounted for in some way. Figure 9.1 shows you the Complex Encounter editor. Each of the above actions has a separate area and it can be a bit confusing as to what does what. These separate sections have been grouped together with a color background to help them stand out better.

## Figure 9.1

Before we get into how to fill out each type of area, lets make a few points clear.

The Complex Encounter consists of tests needed for each type of action as well as 4 scriptable Action Points referred to as Result Codes #1 to #4. The scripts for these Action Points are actual parts of the Complex Encounter code and are self contained.

Whenever a player attempts some type of action, Realmz determines if they were successful or not. If the player failed, then the game branches to Result Code #4 (The Default Fail Result) and begins to execute the script like any other Action Point.

Result Code #4 is the default FAIL script for any section of the Complex Encounter. The other 3 result codes can be used for scripts that play out what will happen if the player was successful at any particular section of the encounter.

Lets look at some of the miscellaneous features of this screen first.

**Prompt String:** This is the string you want to use for the prompt to the encounter whenever a player is confronted with the Action Bar as in figure 9.0. They should also have some text to describe to them what it is you want them to solve. The string ID of the prompt can be entered here.

**Max Times:** This is the number of times they can attempt some solution before it will default to the FAIL Result Code #4 and give them the boot. This will allow you to limit the number of times they can try something.

**Can Backout:** If this box is checked, Realmz will allow the player to back off on this encounter and try it again later. i.e. They will get

n the end of the Action Bar. If this is NOT checked, they will have no option to back out and will be forced to try and resolve the encounter right then and there.

Physical Actions: Section 1: Red background is used for Physical Actions. You can specify up to 8 actions that you offer the player. You can then hit the check box to the left of each action that is needed in order to have a successful outcome. In the above example, there are only two choices for actions. Only the second action, "Try and force the door" will trigger some type of result other than FAIL.

You fill in the field, ACTION RESULT, with the Result Code # you want to take place if they are successful in selecting the correct actions. In this case, if they select "Try and force the door" the game will jump to Result Code #2 and begin executing the codes like any other Action Point. If they fail, it will go to Result Code #4 by default.

Word Action: Section 2: Purple background is used for Word Actions. You can specify up to 40 characters that will trigger a successful response. Do not use any capital letters in this area. If the player enters capital letters, Realmz will convert them to lower before it compares them.

If they player enters the same word or phrase as you specify, then the game will jump to the Result Code # you specify in Word Result. In this example it would jump to Result Code #1 if the player enters the word "Break!"

Don't mistake this for a full featured text based mechanism. It is very basic and does not even attempt to comply with many text based game engines.

ogue Action: Section 3: Blue background is used for Rogue Action. Here you can specify whether or not there is the possibility of a Rogue Action. If you want one, then check the box "Has Rogue Action" AND "Rogue Reset Flag," then fill in the field Rogue Encounter ID with the ID number of the Rogue encounter you want to be activated when they select Special from the action bar.

Rogue Actions are special in that they are scripted on a separate screen than the rest of the Complex Encounter. There are several reasons for this, but the main reason is that there is just not enough room to fit it all on one screen. For more information on how to script a Rogue Encounter see Anatomy of a Rogue Encounter.

You will notice that there is not a Result Code field for the Rogue Action. This is because there can be a number of different things that can happen with the Rogue Action and it is explained in more detail in Anatomy of a Rogue Encounter.

Use Item Action: Section 4: Green background is for Item Actions where you can specify up to 5 items that will cause some type of result. In addition, each item can send the encounter to a different Result Code # if you so choose. In the above example only one item is listed. If the player were to use the Necklace of Keys it would send the encounter to Result Code #1. Any other item would send it to the default of FAIL or Result Code #4

Cast Spell: Use Scroll Actions: Section 4: Yellow background is used for Spell Actions where you can specify up to 10 spells that will cause some type of result. In addition, each spell can send the encounter to a different Result Code # if you so choose. In the above example only two spells are listed. If the player were to cast either "Open Lock" or "Destroy Trap" it would send them to Result Code #1. Any other spell would send it to the default of FAIL or Result Code #4.

Special Note: Though not shown here, there is also a way of covering an entire spell class and assigning a Result Code # to that as well.

If you place the number 1 through 7 as the SPELL ID. It will treat any spell of the appropriate spell class as being a success. Example: If you give it the following code:

## Figure 9.2

This would cause it to go to Result Code #1 if they cast Open Lock or to Result Code #3 if they cast ANY spell of Spell Class 1 (Which happens to be fire based spells).

If you look on the CODE PAGE it shows Spell Classes under the DRVs Type heading. The number corresponds to the Spell Class referred to here.

That pretty much covers Complex Encounters except for the Rogue Actions which appear under the heading Anatomy of a Rogue Encounter.

This is one of those things where a little trial and error and looking at how some of them are defined for some of the other existing scenarios will really help you understand it a lot better.