Custom Heroes & AI

Basic Functionality

Ever wondered how to get that pesky BFME1 AI system to build those new heroes you've spent hours perfecting…? Then read on…

The actual level of scripting needed to make this work is fairly minimal, and many people are surprised by just how easy this is! There are basically 2 main challenges to overcome:

1) Getting the AI system to build the hero.

2) Getting the AI system to use the hero effectively (including all of the special abilities).

This guide will focus on the 1st challenge and get you started with some of the basic AI scripts needed for 2nd one. For challenge no.1, we will be editing the following two files from within the main "Libraries.big":

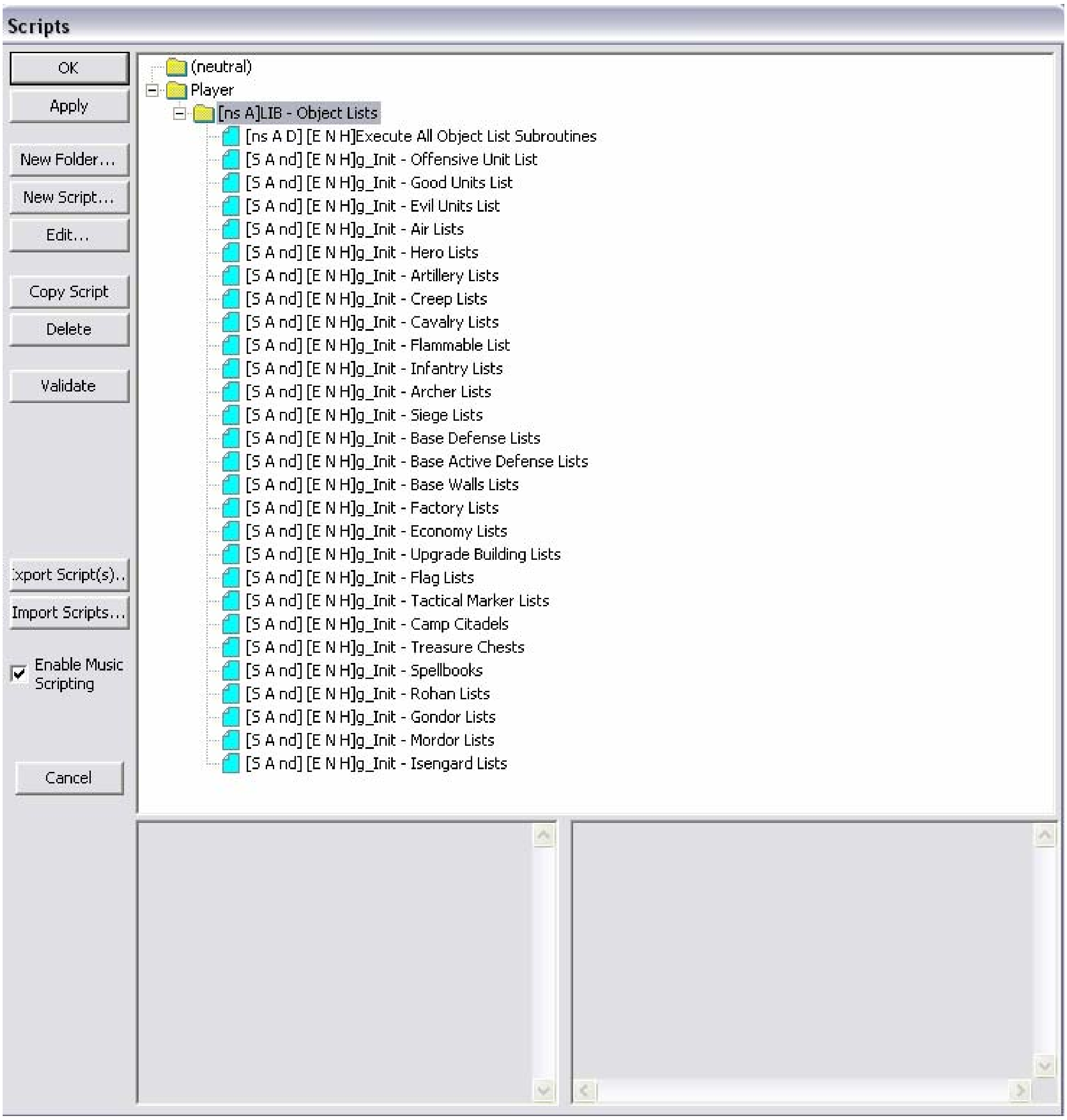
- lib_object_lists.map
- ai_mordor.map

The first thing we need to do is add the "MordorSauron" object to the BFME object build lists. This is fairly straight forward and will be achieved by adding some new scripts to "lib_object_lists.map". Go ahead and open the file with Worldbuilder. Select "Edit→Scripts…".

We want to edit 3 specific script files inside the "[ns A] LIB – Object Lists" folder:

- Evil Units List
- Hero Lists
- Mordor Lists

By now, you should be staring at something like this:



(neutral)
Player
[ns A]LIB - Object Lists
[ns A D] [E N H]Execute All Object List Subroutines
[S A nd] [E N H]g_Init - Offensive Unit List
[S A nd] [E N H]g_Init - Good Units List
[S A nd] [E N H]g_Init - Evil Units List
[S A nd] [E N H]g_Init - Air Lists
[S A nd] [E N H]g_Init - Hero Lists
[S A nd] [E N H]g_Init - Artillery Lists
[S A nd] [E N H]g_Init - Creep Lists
[S A nd] [E N H]g_Init - Cavalry Lists
[S A nd] [E N H]g_Init - Flammable List
[S A nd] [E N H]g_Init - Infantry Lists
[S A nd] [E N H]g_Init - Archer Lists
[S A nd] [E N H]g_Init - Siege Lists
[S A nd] [E N H]g_Init - Base Defense Lists
[S A nd] [E N H]g_Init - Base Active Defense Lists
[S A nd] [E N H]g_Init - Base Walls Lists
[S A nd] [E N H]g_Init - Factory Lists
[S A nd] [E N H]g_Init - Economy Lists
[S A nd] [E N H]g_Init - Upgrade Building Lists
[S A nd] [E N H]g_Init - Flag Lists
[S A nd] [E N H]g_Init - Tactical Marker Lists
[S A nd] [E N H]g_Init - Camp Citadels
[S A nd] [E N H]g_Init - Treasure Chests
[S A nd] [E N H]g_Init - Spellbooks
[S A nd] [E N H]g_Init - Rohan Lists
[S A nd] [E N H]g_Init - Gondor Lists
[S A nd] [E N H]g_Init - Mordor Lists
[S A nd] [E N H]g_Init - Isengard Lists

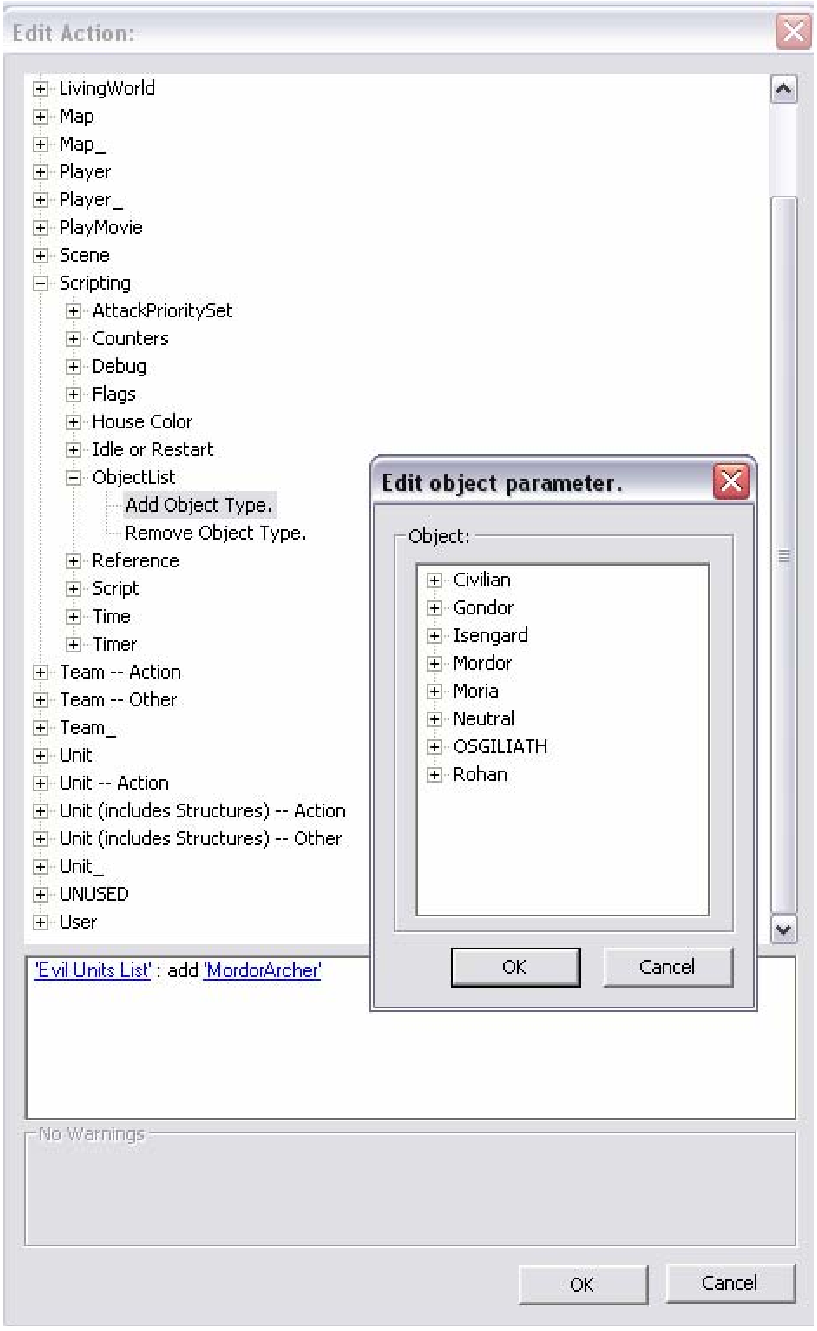
Starting with “Evil Units List”, select the script file and open the “Actions If True” tab. You are looking at a complete list of any unit in BFME1 that has a classification of evil… we need to add “MordorSauron” to this list.

In most cases, the easiest way to add a new script file is to copy an existing one, and then modify it (when making more complex scripts it can also be best to start from scratch so be careful!).

Just select, and then copy the first item in the list (in my case it is “MordorArcher”), and open the “Edit Action” dialog box.

We want to replace the “MordorArcher” object with “MordorSauron”. Just select it and then browse to the appropriate object in the “Edit Object Parameter” box shown below.

You will find Sauron hiding in “Mordor→Unit”. Simple as that!



'Evil Units List' : add 'MordorArcher'

We now need to do the same thing in both the "Hero List" and "Mordor Lists" files. Just be sure to copy an existing file so that your new object is using the exact same parameters as the existing objects.

Now, the second thing we need to do is add a new team for "MordorSauron" to the "ai_mordor.map" file. Here, we will also add some extra scripts so that the AI system will actually utilise our hero, and not just leave him standing around like an idiot.

Again, open the file in world builder. Take a moment to glance at the "Scripts" editor again.

This time, you'll notice that many of the script file directories have the shortcut icon and the script files themselves are not editable. This is because these particular files are inherited from another ".map" file, and can only be modified from within the actual parent file. Just in case you were wondering!

Select “Edit→Edit Teams…”.

Along with “Scripts” editor, the “Team Builder” dialog is an important part of the Worldbuilder environment, and if you intend to build your own maps in the future, you will find yourself using this tool a lot more.

Our first task is to add a new team to our player ironically called “Player”. When you select it, you will be greeted by a large list of teams for use within the Mordor faction. You will notice that they are all broken down into specific categories. Some examples are:

- **Attack** – Any units that the AI system will send to your base (or towards you on the battlefield) during an attack. There are however several forms of attack, so you may see several sub-categories of attack teams… each team with its own set of objectives and motivations.

- **Defense** – Obviously the opposite of attack. There are also many forms of defense such as a counter for siege attacks or hero attacks to name but a few. The way the AI system defends its base is achieved using several sub-categories.

- **Hero** – Used for anything hero related. Each hero will usually have its own team.

- **Patrol** – Used to control the little bunches of units that run about the map during a skirmish. They can be told to patrol specific routes, and look out for specific types of enemy units. Again, look out for several sub-categories.

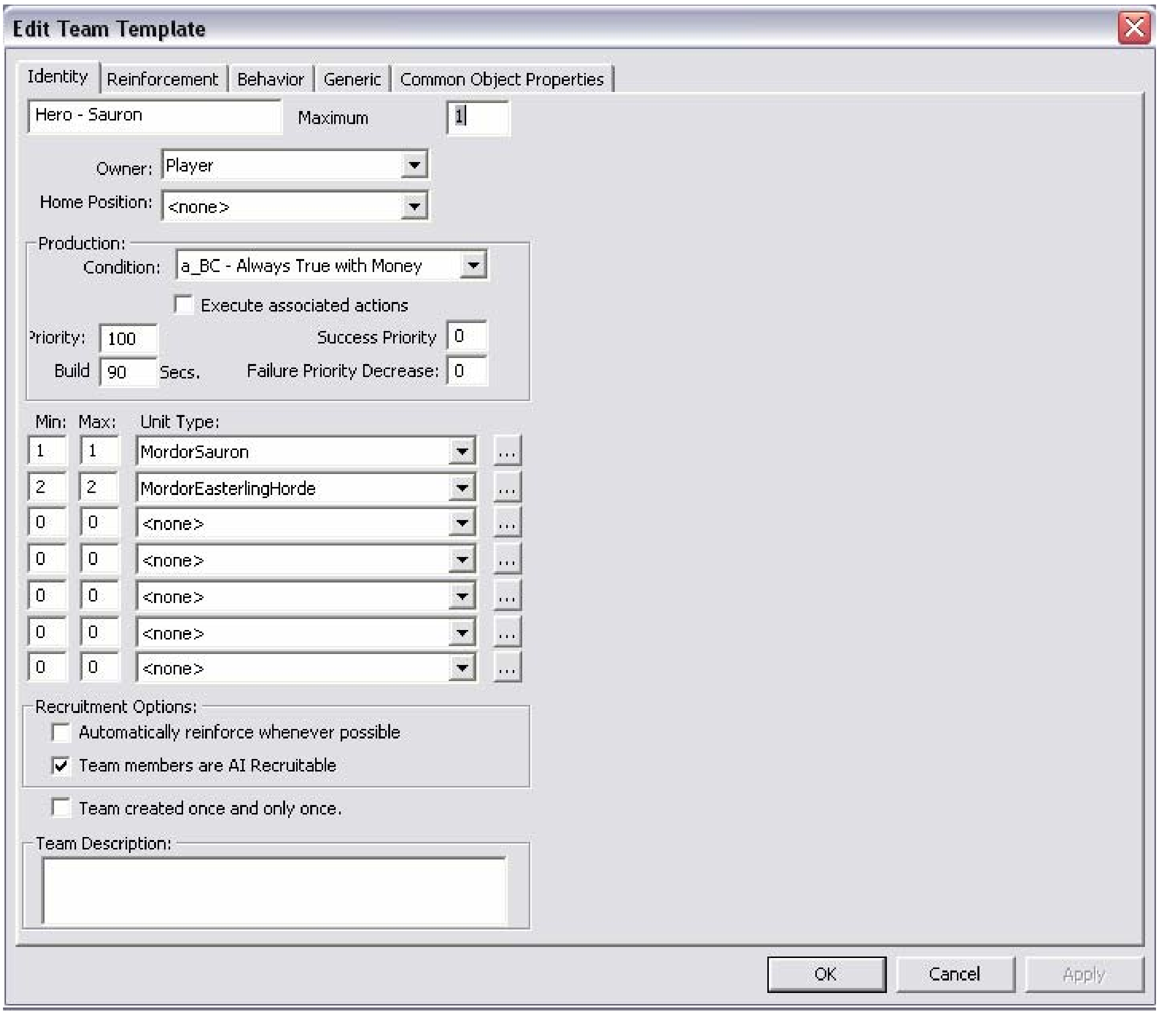
There are quite a few more team types, and you can learn a lot by experimenting with them. Basically, just think of a team as a bunch of hordes/units that hang around together acting in accordance with a specific set of instructions.

One last example (I know I’m dragging on a bit now!) would be an “Anti-Cavalry” team made up of a bunch of Uruk Pikemen. They could patrol about looking to get themselves into the porcupine formation whenever a cavalry-based unit approaches them. It’s that simple.

Hero based teams can be thought of in the same way… they not only control the actions of a specific hero but also the conditions under which they should be built and utilised. More expensive/powerful heroes should only be constructed during the mid or late game, after a stronger economy has been created.

For now, copy the “Hero – Witch King” team and then select your duplicate version. Using the “Edit Team Template” dialog, we firstly need to replace the “MordorWitchkingOnFellBeast” object with “MordorSauron”. Also, make sure you re-name this team using the text-box in the top-left hand corner.

You should now be looking at something like this:



Hero - Sauron Maximum 1

Owner: Player

Home Position: <none>

Production:

Condition: a_BC - Always True with Money

□ Execute associated actions

'riority: 100 Success Priority 0

Build 90 Secs. Failure Priority Decrease: 0

Min:	Max:	Unit Type:
1	1	MordorSauron
2	2	MordorEasterlingHorde
0	0	<none>
0	0	<none>
0	0	<none>
0	0	<none>
0	0	<none>

Recruitment Options:

□ Automatically reinforce whenever possible

☑ Team members are AI Recruitable

□ Team created once and only once.

Team Description:

Notice that I have added a “MordorEasterlingHorde” to the team (2 hordes)… personal bodyguards!

The maximum number of Saurons we want any one faction to build at any given time is one (hence both the “Min” and “Max” values are set to 1). The Nazgul heroes are set up in the same way except they are set to the value of 2… told you it was easy!

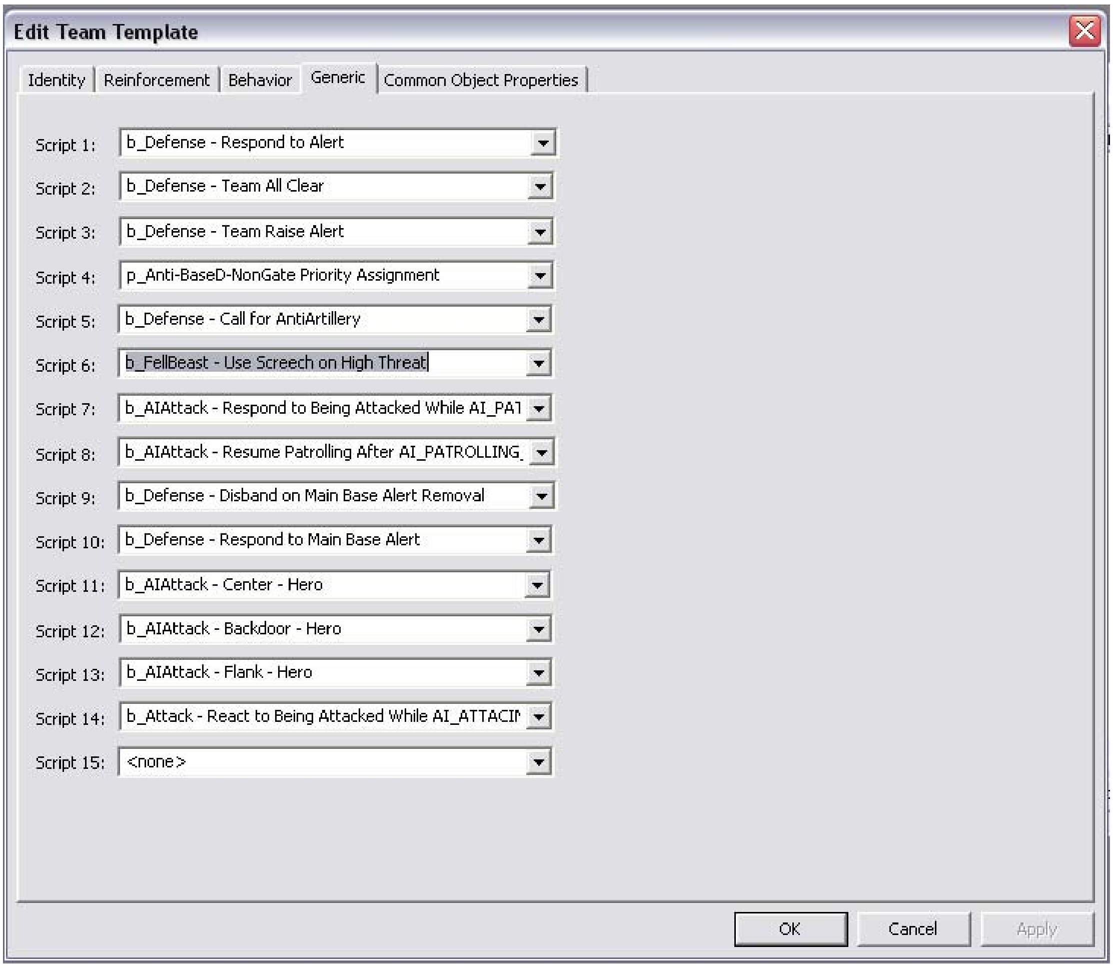
Depending on how you wanted the Nazgul to act, you could create a single team of 2 Nazgul heroes, thus forcing them to attack and operate together, or change the “Maximum” field to 2, thus allowing 2 teams of 1 Nazgul (the default setting used in BFME1).

If you want to go crazy… just increase the max number of Gollums’ to 100, then let a couple of AI controlled players go at each other. Pure Smeagolic Joy!

The most important part of this section is arguably the “Production Condition”. In other words, under what circumstance should we build Sauron…? In this case, I have used the “a_BC – Always True with Money” condition that (as its name suggests) forces the AI system to build Sauron whenever it happens to have enough money. Experiment with this to achieve a variety of different outcomes.

Now, switch over to the “Behaviour” tab. From the “On Create” combo box select “b_AIAttack – Patrol Combat Areas”. This will force the AI system to send Sauron out on patrol after he has been constructed… the combat areas themselves are taken from the combat area nodes scattered around the map by the designer.

Last but not least, select the “Generic” tab. Here you will find the core set of scripts that control the behaviour of the hero during a game. See below:



Script 1: b_Defense - Respond to Alert

Script 2: b_Defense - Team All Clear

Script 3: b_Defense - Team Raise Alert

Script 4: p_Anti-BaseD-NonGate Priority Assignment

Script 5: b_Defense - Call for AntiArtillery

Script 6: b_FellBeast - Use Screech on High Threat

Script 7: b_AIAttack - Respond to Being Attacked While AI_PAT

Script 8: b_AIAttack - Resume Patrolling After AI_PATROLLING_

Script 9: b_Defense - Disband on Main Base Alert Removal

Script 10: b_Defense - Respond to Main Base Alert

Script 11: b_AIAttack - Center - Hero

Script 12: b_AIAttack - Backdoor - Hero

Script 13: b_AIAttack - Flank - Hero

Script 14: b_Attack - React to Being Attacked While AI_ATTACIN

Script 15: <none>

Since we already have a bunch of scripts relating to the “MordorWitchKingOnFellBeast” object, we only really need to remove the ones that cannot be utilised by Sauron himself.

Firstly, you will notice something very interesting:

“b_FellBeast - Use Screech on High Threat”

The “Generic” tab is also where the final scripts that allow heroes to use their special abilities are located… however, we need to do some more work in order for those abilities to show up in our list of available scripts…

For now, remove this by selecting “None” from the combo box. The relevant script should automatically disappear.

As you begin to experiment, you will find an absolute mother-load of scripts, each designed for specific elements of unit/hero behaviours.

Looking at our modified “MordorWitchkingOnFellBeast” (now “MordorSauron”) object, we can start to build a picture of how this hero will act on the battlefield. Some example scripts:

- **b_Defense - Respond to Alert -** Calls the unit/hero back to defend the base when under heavy attack. Actual threat levels of enemy attacks are worked out using additional scripts

- **b_Defense - Team All Clear** AND **b_Defense - Team Raise Alert -** These 2 scripts allow a team to send out an all clear signal after an attack has been repelled or raise the alarm (and call back existing units) in the event of an attack.

  Different teams will react in different ways depending on the nature of the attack, and some teams will resume patrol patterns when an all clear signal is given.

- **p_Anti-BaseD-NonGate Priority Assignment -** Attack the enemy base via one of the 3 attack routes (Center, Flank or Backdoor) but leave the gate alone. This works great for Nazgul, Witcking or Eagles because they can fly over walls. You will have problems if you apply this script to ground based units/heroes because they will simply run to their deaths against Gondor or Rohan players.

- **b_Defense - Respond to Main Base Alert -** Get your ass back to the base if it is under attack.

- **b_AIAttack - Center - Hero** – Attack an enemy base via the “Center” attack path nodes (placed by map designers), and attack heroes before anything else.

  Add the same script for “Flank” and “Backdoor” to randomise the way in which this unit/hero will attack the enemy.

- **b_AIAttack - Resume Patrolling After AI_PATROLLING_HUNT** – Go back to your general patrol route if you tried to chase an enemy (that you spotted whilst patrolling) and decided to give up on it.

Working with Hero Abilities

As mentioned earlier, we need to play around with a few more script files if we want the BFME1 AI system to make use of our special hero abilities during run time.

Firstly, open “lib_hero_behaviours.map” and take a look inside the “[ns A]LIB - Hero Abilities” script folder. You’ll notice a complete list of all of the original heroes that EA went with in the final release of BFME1. This area is basically where the magic happens…

Copy the “[ns A]Hero Ability – Boromir” script file, and then change the name to “[ns A]Hero Ability – Sauron”.

In order for the AI system to sue our Sauron character effectively, we will need to implement three things:

- A Unit Reference for the AI to recognise the object in subsequent scripts
- A basic Combat Retreat script to allow our hero to run away when heavily damaged
- Any special abilities our character may possess

To make things a little easier, we can recycle some of the Boromir script files for use with our own Sauron character. Delete all of the script files in our new folder with the exception of the Unit Reference and Combat Retreat scripts. Change the name of the “[S A nd][E N H]b_Boromir - Set Unit Ref” script file to “[S A nd][E N H]b_Sauron - Set Unit Ref”.

We can then modify the condition/action set to reflect the following:

```
*** IF ***
    Team '<This Team>' has one or more units.

*** THEN ***
  Find 'MordorSauron' Owned by Player '<This Player>' nearest team Team '<This Team>' and reference as UnitRef 'AI_HERO_SAURON'
```

When the game is running, our character will now always be known as AI_HERO_SAURON whenever decisions need to be made, or abilities need to be activated.

We can then modify the Boromir combat retreat script by firstly changing the name to “[S A nd][E N H]b_Sauron - Retreat to Combat Area”, and then modifying the condition/action set to reflect the following code:

```
*** IF ***
    Unit '<This Object>' Health IS Less Than  25  percent.

*** THEN ***
   Team '<This Team>' executes Script 'be_Hero - Retreat to Combat Area' sequentially.
```

As you can see from the above script, our character will be forced to retreat from combat if it’s health should become less than 25 percent at any given time. You can play around with these values to achieve different effects (remember that some heroes are much more durable than others).

Now we can go about enabling some of our hero abilities… let’s assume for a second that our Sauron character has the ability to wield the mighty one ring that boosts his armour and attack strength for a limited time.

The trick here is to decide on when is best to use this ability… for example when our hero is under threat from multiple enemies, or perhaps when we get close enough to an enemy encampment. In this case, we’ll go with the first option since it’s the most reliable.

Create a new script by pressing the “New Script…” button.

In the “Script Name” field enter “b_Sauron - Use One Ring On High Threat”. Make sure that the script is a subroutine, active and available in all three difficulty levels by checking all of the appropriate check boxes.

We can then add our ability condition as follows:

```
*** IF ***
     Team '<This Team>' has Greater Than or Equal To threat level 30.00 within radius 200.00
    *AND*  Player '<This Player>"s Team '<This Team>' are ready to use Ability
'Command_SauronOneRing' (at least one member).

*** THEN ***
   Team '<This Team>' use Ability 'Command_SauronOneRing'.
```

Obviously, the above condition refers to a special ability in the characters ini code including the actual command button found in the Sauron command set. You will need to make sure you have all of the necessary code completed for your own abilities before you can add them to the AI system.

The threat level is calculated by taking the sum of the threat values of all hordes and heroes within the specified target area relevant to our character (in this case, a radius of 200). If this value is equalled or exceeded and our unit has the ability to use the special power, it will be activated immediately.

Other types of abilities can be activated in a similar way… generally speaking; the action is always the same i.e. use ability ‘x’. The key is to experiment with different kinds of conditions that will affect how and when a hero should activate an ability.

Special powers such as “Kings Favour” should not have any strict conditions bound to them, and should be activated as soon as they are available… Others may need to more specific conditioning… it’s all down to trial and error.

We now need to save and close the map file, and add the files into our BFME mod. You will need to start the actual game up, and then exit from the main menu immediately in order to have access to our new script files from within another map file.

This is important, and leads us to our final point.

Earlier in this tutorial, we mentioned the “Generic” tab of our Sauron team file in “ai_mordor.map”. When we first copied the team, we noticed that our unit had a “b_FellBeast - Use Screech on High Threat” script that we deleted because it was useless to us…

Return to the “ai_mordor.map” file, and navigate to the “Generic” tab of our Sauron team.

Select the last combo box in the list entitled “None” and search for the “b_Sauron - Use One Ring On High Threat” script file that we created previously.

Select it, save and close the file.

You’re all done! It’s worth re-iterating at this point that you will not see any new scripts in the combo box list of the “Generic” tab unless you start/exit the game first…

The next time you run the game, you should see your new hero in full flight! Enjoy.

As always, feel free to contact me if you have any questions about BFME1 AI.

Hebblewhite.