Custom Structures & AI in BFME1

Introduction

Let's imagine you have just created a new and impressive Barracks structure for the Rohan faction, but your pesky AI system doesn't want to use it! Time for a fix…

I've been working on the optimisation of the predictive building AI lately, and thought I'd produce this tutorial on what is the definitive, and most efficient way of getting your AI system to build and use your custom structures at run time.

Here goes:

NOTE: Before this will work you need to make sure you have added all of the appropriate unit reference scripts in "lib_objects_lists.map" and team entries in "ai_rohan.map" for all of your new hordes/heroes or else the AI system won't build them regardless of what we do here! Anyway, on with the show...

There are 3 main parts involved in getting your "factory" structures to work. The first places us smack bang in "ai_rohan.map".

Part 1:

Open the script editor, and navigate to the "[ns A]Pred Build Overrides" script folder. Look down the list until you see the "[S A nd] [E N H]g_PredBuild - Build Archer Building" script.

Copy it.

Immediately change its name to "[S A nd] [E N H]g_PredBuild - Build Barracks Building". Then modify the condition/action statements in accordance with the following:

*** IF ***
Can player Player '<This Player>' build at base Unit 'AI_CURRENT_CONSTRUCTION_SITE'

*** THEN ***
Build building 'RohanBarracks' at a foundation that is the most near, relative to the object of type, 'CastleRight' which is part of the base named, Unit 'AI_CURRENT_CONSTRUCTION_SITE' ... then finally reference the new building as: UnitRef 'AI_BARRACKS' .

The above script will force a build of any particular structure we desire (in this case it refers to a RohanBarracks). At this point, if you run your game, you should see your new structure pop up inside an enemy base, but it will not be used… we now need to tie the structure into the AI combat reasoning system.

The first step in achieving this forces us to include our new structure in the basic attack strategy used by the AI system.

Open the “[ns A] Attack Rohan AI” script folder.

The scripts in this folder allow us to build multiple factory structures of the same type. In the case of Rohan, we are effectively talking about Archery Ranges and Stables. Note that the Archery Range scripts can only be activated twice, and the Stables three times.

We can go the whole hog and make three entries for our new Barracks Building.

Copy the first “[ns A D][E N H]a_BB - Stables 1” script and then change its name to “[ns A D][E N H]a_BB – Barracks 1”. Modify the condition/action statements in accordance with the following:

*** IF ***
Flag named 'AI_On' IS TRUE
AND Counter 'g_Attack_Urgency' IS Greater Than or Equal To counter 'g_ATTACK_BUILD_WAVE_1'
AND 400 is Less Than the number of credits possessed by Player '<This Player>'
AND Can player Player '<This Player>' build at base Unit 'AI_CURRENT_CONSTRUCTION_SITE'
AND Unit 'AI_CURRENT_CONSTRUCTION_SITE' has Less Than threat level 25.00 within radius 500.00
AND Player '<This Player>' has Less Than 1 unit or structure of type 'RohanBarracks'

*** THEN ***
Build building 'RohanBarracks' at a foundation that is the most near, relative to the object of type, 'Flank1' which is part of the base named, Unit 'AI_CURRENT_CONSTRUCTION_SITE' ... then finally reference the new building as: UnitRef 'AI_BARRACKS_1' .

Enable Script 'a_BB - Barracks 1 ReEnable'.

Note the final statement in the event section of the above script… We now need to create a new “a_BB – Barracks 1 ReEnable” script, so the building can be replaced should the need arise during combat.

Copy the “[ns na D][E N H]a_BB - Stables 1 ReEnable” script file and then change the name to “[ns na D][E N H]a_BB - Barracks 1 ReEnable”. Modify the condition/action statements in accordance with the following:

*** IF ***
Counter 'g_Attack_Urgency' IS Greater Than or Equal To counter 'g_ATTACK_BUILD_WAVE_1'
AND Player '<This Player>' has Greater Than or Equal To 1 unit or structure of type 'RohanBarracks'

*** THEN ***
Enable Script 'a_BB - Barracks 1'.

Repeat as many times as you feel necessary… I would recommend three times for starters, but take care to replicate the correct scripts and values.

That's it for part 1, and the process is generally the same for all four factions in BFME1 with the exception of Mordor. For some reason, EA kindly decided to replace the "Attack Faction AI" folder with one called "Production Buildings" just to confuse us… so watch out for it.

Part 2:

In part one, we created the basic structure building script, and then made the new Barracks available when the AI system is in attacking mode. Now we need to play around with the build priority scripts and values so that our Barracks is considered when weighing up the type of units we need to create when under attack, or forming an attack of our own.

Save and close the "ai_rohan.map" file, and open "ai_predictive_building.map".

Open the "[ns A] Predictive Building Init" script folder, and add the following entry to the "[ns A D][E N H]g_PredBuild - Variable Inits" script file:

```
Set 'l_PredBuild_Barracks_Building_Priority' to  0
```

The above value can be increased from zero when required depending on the nature of the game being played, and the threats being posed to the AI system at any given time.

Next we need to add a repeat entry of our original structure building script to the "[ns A] Predictive Build Subs" script folder. In this case, we only need the condition, because the action is already set up for us in "ai_rohan.map".

Add a script called "[S A nd] [E N H]g_PredBuild - Build Barracks Building" with the following contents:

```
*** IF ***
    Can player Player '<This Player>' build at base Unit 'AI_CURRENT_CONSTRUCTION_SITE'

*** THEN ***
  Null operation. (Does nothing.)
```

Now we need to set up a script that will activate our build command based on priority data that ways up economy, archer, cavalry, artillery and upgrade necessity at all given times during a game. We want our structure to be built only if it is of an equal or higher priority than the others…

Open the "[S A nd] [E N H]g_Predictive Build Execution" script folder, and copy the "[S A nd] [E N H]g_PredBuild - Infantry Building Is Highest Priority" script file and change the name to "[S A nd] [E N H]g_PredBuild - Barracks Building Is Highest Priority". Then modify the condition/action statements in accordance with the following:

```
*** IF ***
   Counter 'g_PREDBUILD_MONEY_NEEDED' IS Less Than counter 'g_Economy_Current_Money'
   *AND* Counter 'g_PREDBUILD_MAX_INFANTRY' IS Greater Than counter
'g_PredBuild_My_Soldier_Building_Count'
   *AND* Counter 'l_PredBuild_Infantry_Building_Priority' IS Not Equal To  0
   *AND* Counter 'l_PredBuild_Infantry_Building_Priority' IS Greater Than or Equal To counter
'l_PredBuild_Archer_Building_Priority'
   *AND* Counter 'l_PredBuild_Infantry_Building_Priority' IS Greater Than or Equal To counter
'l_PredBuild_Artillery_Building_Priority'
   *AND* Counter 'l_PredBuild_Infantry_Building_Priority' IS Greater Than or Equal To counter
'l_PredBuild_Cavalry_Building_Priority'
   *AND* Counter 'l_PredBuild_Infantry_Building_Priority' IS Greater Than or Equal To counter
'l_PredBuild_Economy_Building_Priority'
   *AND* Counter 'l_PredBuild_Infantry_Building_Priority' IS Greater Than or Equal To counter
'l_PredBuild_Upgrade_Only_Building_Priority'

*** THEN ***
  Run Subroutine 'g_PredBuild - Build Barracks Building'.
  Run Subroutine 'g_PredBuild - Clear Pred Build Priorities'.
```

You can then add the reference to your new subroutine in the “[S A nd] [E N H]g_PredBuild - Run Build Scripts” with the following code in the action section of the script:

```
Run Subroutine 'g_PredBuild - Barracks Building Is Highest Priority'.
```

Note that we are classing our Barracks as a type of infantry factory… this is by design as we don’t want to have to set up a whole new set of scripts, we can piggy-back on the infantry detection system.

Move on to the “[ns A]g_Predictive Monitoring” script folder, and take a look at the “[S A nd] [E N H]g_PredBuild - Monitor Buildings” script file. Note that the AI system is constantly keeping track of the number of factories and units of both itself and opponents during run time… this is the infantry detection system we are using since our Barracks is a type of “Infantry Factory”.

On a side note: The “Infantry Factory” object is available from “Object Lists” section of the system hierarchy. To use the infantry method correctly I have made sure that my “RohanBarracks” is declared as a factory, and any hordes that it can build are declared as infantry (NOT archers – they are different!) in “lib_object_lists.map”.

Finally, we need to write a script that will force the build priority of our Barracks in relation to other unit types during run time. Copy the “[S A nd] [E N H]g_PredBuild - Set Infantry Building Priority” script file and change the name to “[S A nd] [E N H]g_PredBuild - Set Barracks Building Priority”. Then modify the condition/action statements in accordance with the following:

```
*** IF ***
   Counter 'g_PredBuild_My_Soldier_Building_Count' IS Less Than counter
'g_PREDBUILD_MAX_INFANTRY'
   *AND* Counter 'g_PredBuild_My_Soldier_Building_Count' IS Equal To  0
  *** OR ***
```

```
   Counter 'g_PredBuild_My_Soldier_Building_Count' IS Less Than counter
'g_PREDBUILD_MAX_INFANTRY'
   *AND* Counter 'g_PredBuild_Barracks_Building_Count' IS Greater Than or Equal To counter
'g_PREDBUILD_OPP_SINGLE_UNIT'
  *** OR ***
   Counter 'g_PredBuild_My_Soldier_Building_Count' IS Less Than counter
'g_PREDBUILD_MAX_INFANTRY'
   *AND* Counter 'g_PredBuild_Barracks_Building_Count' IS Less Than counter
'g_PredBuild_Archer_Building_Count'
   *AND* Counter 'g_PredBuild_Barracks_Building_Count' IS Less Than counter
'g_PredBuild_Artillery_Building_Count'
   *AND* Counter 'g_PredBuild_Barracks_Building_Count' IS Less Than counter
'g_PredBuild_Infantry_Building_Count'
   *AND* Counter 'g_PredBuild_Economy_Building_Count' IS Greater Than or Equal To counter
'g_PredBuild_Factory_Building_Count'

*** THEN ***
  Counter 'l_PredBuild_Barracks_Building_Priority' , Add with value  2
```

Again, note that we are using the infantry method that ties into the Soldier_Building_Counter that allows us to monitor how close we are the maximum number of appropriate infantry types at any given time.

Repeat the same process for the “[S A nd] [E N H]g_PredBuild - Set Infantry Building Filler Priority” script file. Don’t forget to change the name, and then add the following code:

```
*** IF ***
   Counter 'g_PredBuild_My_Soldier_Building_Count' IS Less Than counter
'g_PREDBUILD_MAX_INFANTRY'

*** THEN ***
  Counter 'l_PredBuild_Barracks_Building_Priority' , Add with value  1
```

That’s pretty much it for the main scripting necessary for our new structure to function, with the exception of one last thing…

Part 3:

We now need to initialise any working values we require when the map is first loaded. Open “ai_initialise.map”. In our case, the only value left outstanding is that of the “Barracks_Building_Count” we have used previously.

Open the “[ns A]Non Over Ride-able Paper Work” script folder, and add the following line to the “[ns A D][E N H]g_Init - Initialize Global Variables (NonOverride)” script file:

```
  Set 'g_PredBuild_Barracks_Building_Count' to  0
```

Save and close the file, and you're all done. You should now see your Rohan AI system build and utilise your new structure effectively i.e. in conjunction with the battle strategy as opposed to randomly spilling out units.

For maximum effectiveness, you need to make sure that you plan your "ai_rohan.map" teams appropriately… ensure that you take into account the initial build waves of units, and DO NOT set highly priced objects to early!

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

Enjoy!

Hebblewhite. ☺