

Single Player Mission Process

A break down of Good Mission 1: Rivendell.

This Tutorial provides general guidelines on the process of building Battle For Middle Earth 2's first mission. We'll talk about map layout and general scripting of the mission. If you are not familiar with the basics of WorldBuilder, you should check out BFMEII_basicscriptingwb.doc beforehand. It covers essential tips you'll need to know and understand before reading further.

- **Lesson 1 – Layout.** When creating a mission like Good Rivendell, many things must be taken into account. When you first look at the layout of the mission, you can see that there really is only one way for the player to go. The south side of the map has a large area where the player can roam, but ultimately the player **MUST** cross the bridge.

Having only one way to go helps an exploratory player follow the correct mission progression. This bridge is also an indicator that the mission has changed. There are 3 acts to this mission, and every time you cross the bridge, you enter a new act.

- In general, when making a layout for a mission you have to take into account a lot of obvious things like available tech and special abilities. Technical constraints must also be taken into account.
 - Choke points should be at least 2 ½ hordes wide. There should never be a case where a path-able area is less than 2 hordes wide...ever.
 - Take into account high ground and wall bonuses. They actually matter in this game, so a wall on top of a hill overlooking a pathable ravine will be almost impossible to take down, which is no fun for anyone.
 - If you want to call attention to something, frame it. Important set pieces like the Tower of Orthanc or the House of Elrond should be framed. The Tower has the roads surrounding and leading to it, drawing your attention towards it. The house of Elrond has been raised up on a sort of cliff, helping illustrate that it is an important place.
 - If there is a river in the map, make sure there are adequate fords to cross. Only use bridges if absolutely necessary. Fords should be at least 3 hordes wide and no deeper than 5 ft. below the water plane.
- Act I begins on the south side of the river. The first part of the mission is a scouting mission. Very simple and straight forward, so the layout must be equally simple. There are no big rocks or stands of trees here, lending the area a subtle sense of simplicity. There is a path that leads the player along this

edge of the cliff. The choice to put the path along the cliff was made to give the player a natural border and a not so subtle sense of direction.

The Act I space also had to be large enough to support the Goblin base in Act III, which was another important consideration when creating the layout.

- Act II considerations for layout include the placement of the House of Elrond, the proximity of that to the incoming Goblin waves and the gate. Because buildings die so rapidly in this game, we didn't want to have a lot of travel time between the House of Elrond and the west side entrance where the base attack begins. This enables the player to react quickly enough to where casualties are generally minimal. In a mission 1 the player is still getting a feel for the game and we don't want to come straight out and hit them as hard as we can...that comes later. The river that runs through also serves to help confine the area and give it a sense of claustrophobia. However, the player still needed enough room for maneuvering and tactics so the gardens to the west are rather open and flat.
- Act III was actually quite simple to lay out -- it had to be large enough for a decent sized Goblin base, but the north end also had to be shared with Act I. More emphasis was put into the base layout than the actual map layout here. Planning the Goblin base buildup was a lot of fun, but also required a lot of thought in respect to the layout. It demonstrates how the AI will expand outwards after building up a main base, and show what those expansions look like, but it must also string the player along in a bread crumb fashion, leading the player to the final conflict.
- **Lesson 2 – Setup.** The layout is important, but what happens really defines that mission. Most importantly, scripting is what makes the mission fun. For this tutorial, I am going to take you through all 3 acts start to finish. These acts cover Pre-placing units and ambushes, spawning waves in and general practices when doing the basics.
- **Placing and Naming Objects** -When making a single player mission, the first things to be placed (and inevitably changed a hundred times) are the players starting units. In this case, the player starts with Glorfindel and a battalion of Lorien Warriors. After these units have been placed in the world, we need to set up various Players (both AI and the User) required by the map. We placed the units in the world before doing this because it is easier to add a player by placing objects down, than by going through the Edit Player setup screen.

Once you have your starting units give them each a unique name. I called mine Glorfindel and Elf 2. One thing to keep in mind when naming your units is that your units will show up in a list, so keeping a naming convention helps when searching your unit lists and keeps them organized. Also name your



units something very obvious like, “Eastern Attack Wave 1”. That tells you exactly what the unit or team does.

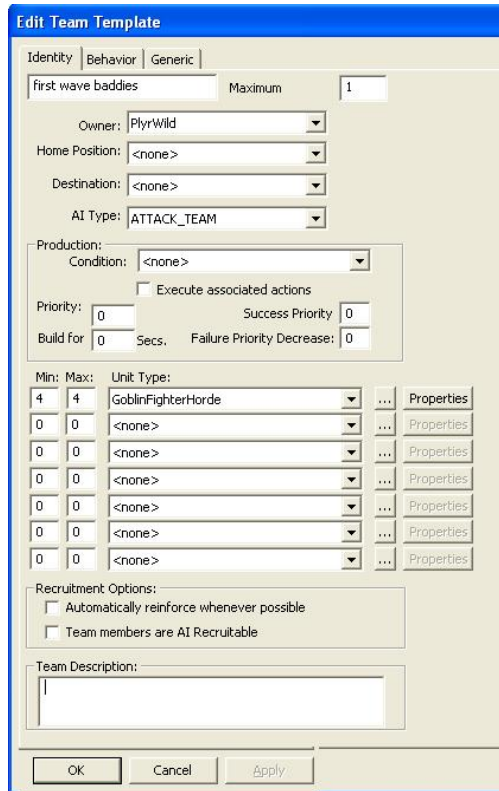
In Good Rivendell, I have the pre-placed goblins named First Encounter, Second Encounter, Third encounter fighters, Third encounter archers... you get the picture. When I am scripting the actions for these guys, I know exactly what they do and where they are in the mission flow.

- **ACT I, pre-placed encounters and ambushes.** – So, first thing you’ll want to do is set the stage for how the level is going to play out. In Good Rivendell, Goblins are invading and Glorfindel spots them while on a routine patrol.
 - **Leading the player** -The player’s first objective is to return to the House of Elrond and inform him of the goblin invasion. So we have the starting forces facing the direction they should move, and we have an enemy horde set up just outside the vision range. The player encounters are also set up to string the player along in the right direction. When making single player missions it is a great idea to attack the player from the direction you want them to go.
 - **Sleep** - As you progress through the mission you encounter many small groups of goblins. These are thought of as the “forward scouts” of the large force. All of these guys are pre-placed on the map. You will notice most of them are set to the AI mood “Sleep”. Units are set to Sleep as an optimization to help keep the framerate higher. When a unit is set to Sleep it does not track any information pertaining to interacting with other units.
 - **Waking Units** - For each sleeping group of units a trigger area has been set up. When a player’s unit enters this area, the units will change from Sleep to Aggressive. This enables them to attack and interact and seek out player units. Player recruitable units work in the same way. They start off in Sleep then are set to Aggressive when the player enters the trigger.
 - **Gags** – Use gags to keep the mission from being repetitive. The Mallorn tree gag at the beginning is a perfect example. Instead of encountering units standing around or jumping out of trees, you see these guys attacking something already. The tree is actually set to invulnerable, and the goblins are attacking from the mission start. When the player gets close, the tree is destroyed and the goblins will now acquire the players units.
 - **Rewards** -Once the player crosses the bridge they are rewarded with an in-game cut scene. You generally want to give the player something cool when they finish an objective, whether it’s a new type of unit, a reinforcement wave or a bit of story. During this cut scene you’ll notice that control is taken away from the player so they can enjoy the scene that plays out. You’ll also notice that whoever is talking will begin flashing, this helps the players identify what unit is speaking and gives

the player the idea that these units are important. The cut scene as a whole gives the player a sense of accomplishment.

○ **ACT II – Spawning attack waves.**

- **Telegraphing** - After teaching the player the basics of unit building you begin to hear war drums. While this is also dramatic, it signals the player that something bad is about to happen. This is called telegraphing. Whether you use a big creepy drum or a long playing horn or a siren you will always want to telegraph big and important events that happen. Usually this is done with Audio.
- **Creating the Path to follow** – Click the following button to bring up the Path Tool.  After you click this button, click the ground to place a waypoint. Click and drag to connect waypoints. When placing your path, keep in mind how many units are in the team that uses it. If there are a lot of units meant to follow it, make sure there is plenty of space between the path and any obstacles. And just like everything else, make sure the name is distinct.
- **Creating the team** - Spawning in teams is rather simple, all you need is a team and a waypoint. Let's create the team first. Bring up the team window by clicking this button:  Once you are in the team window click the “add new team” button, located at the bottom of the team window. This brings up the team template window. At the top, name your team. I called mine “First Wave Baddies.” This instantly lets me know this is the first wave of enemy attackers any time I refer to them later.



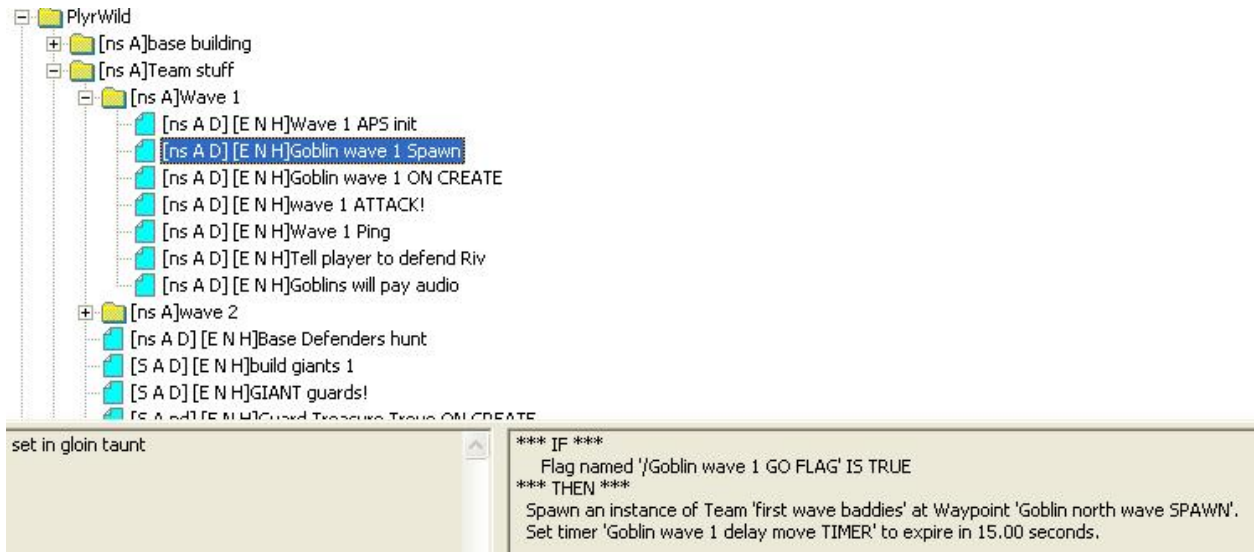
- Set up a Spawning team**– Once you have your team set up, you need to throw down a waypoint (WP) where you want them to spawn in. In this case, I have a few teams spawning on the north side of the river, so I named the WP “Goblin north wave spawn.” Once again, being as descriptive as possible helps you keep everything straight. WP’s are cheap so it often helps to throw down one for each team spawning there, even if it’s the same general area.

Having separate WP’s will help keep your mission running clean. Here you see I have placed the WP off the map next to a few other WP’s that spawn elves later. You should ALWAYS spawn things in off the map and march them in. This will help keep the player immersed in the reality you are creating for them – if you spawn them on map, no matter how cleverly you do it, someone, somewhere will find out.



You will also notice that the Path that the units are meant to use begins inside the map border. Doing this ensures that the team will be able to path onto the map at the appropriate point.

- **Scripting a Spawned team** – Once you have the WP and Teams actually set up, you need to set up the scripts. Check out the picture below:

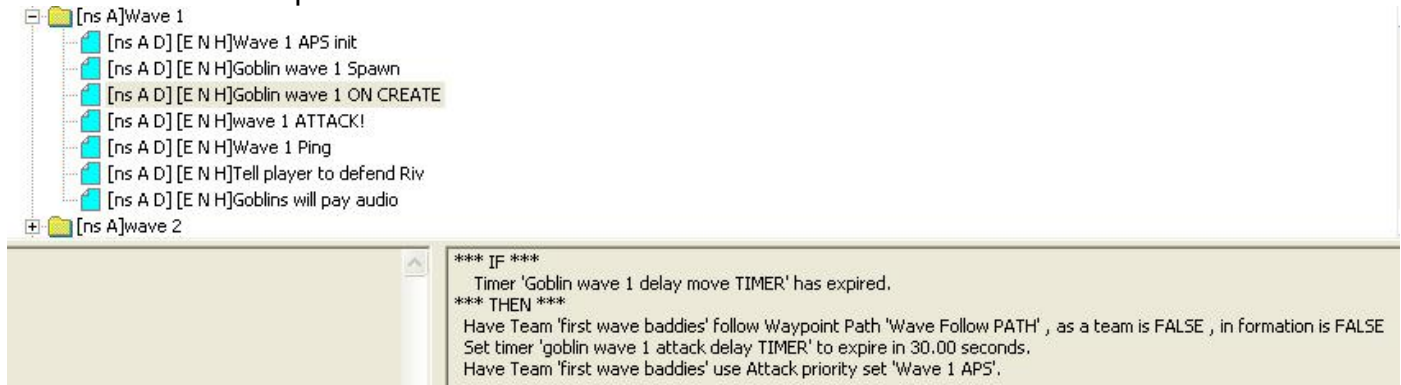


- You will notice I have my scripts organized into folders. I can't stress enough how important it is to organize your scripting. They are also organized by firing order top to bottom. You can see the spawn script is highlighted. It triggers when a Flag is set to True. Flags and timers are good to use when spawning in a team. Keep your condition statements simple.

It is a good rule of thumb not to have it check for more than 3 conditions. You can see that I set a timer in the same script also, "Goblin Wave 1 Delay move

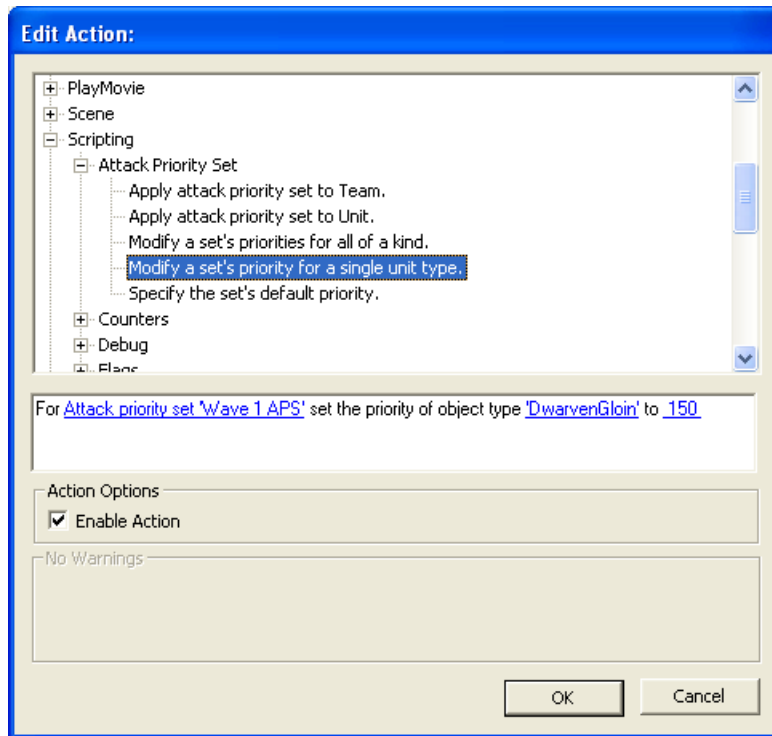
TIMER”. This timer is as it states, how long the delay is before the team move. The flag is set during the cut scene. It’s a good rule of thumb to spawn in your teams 15-30 seconds before you are going to use them. This helps spread out the processor hit and guarantees the team is ready to go when you need it.

- The next script that fires is below.



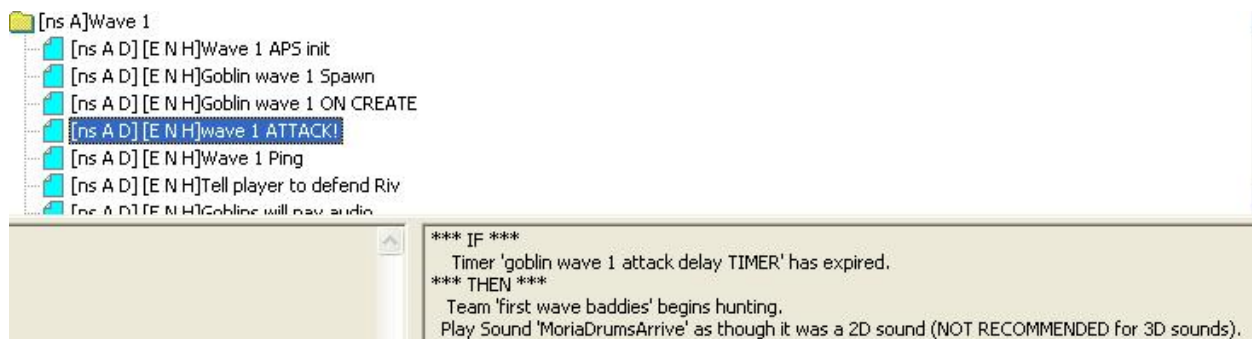
This next script fires when the “Goblin Wave 1 Delay move TIMER” expires. First thing this script does is move the team down a Path. This action can be found in Team-action → Move → Team follow waypoint path. You’ll notice the “as a team” and “in formation” are both false, Goblins are not an organized faction. Having them run around willy nilly is definitely in their idiom.

Once the little goblins are on their way, another timer is set. This one is the attack delay. When setting up attack delays, it is a good practice to have the delay timer slightly shorter than the time it takes to traverse the path. This is done so that the units are still in motion when they receive their attack orders. Otherwise you can have teams stop of a few seconds, then attack. If the unit stops, it does not look very realistic. This is probably the only time it is actually a GOOD practice to interrupt another script. The last action is telling the team to use an APS or Attack Priority Set. APS’ will tell the team what kind of targets to prefer and are populated by using the script shown below:



- Attack Priority Sets** -The way APS' work is simple. First you give the actual priority set a name, this one is "Wave 1 APS". Then, you specify an object type and associate a number to that object type. The number is a priority designation. Higher is means more likely to attack. Here we set DwarvenGloin to have a priority of 150. When you play the mission, you will notice that the first wave runs right past the player's buildings and straight for the player's heroes. You can even change the Priority of an object on the fly, once a team is told to use an APS it looks to that before acquiring its next target.

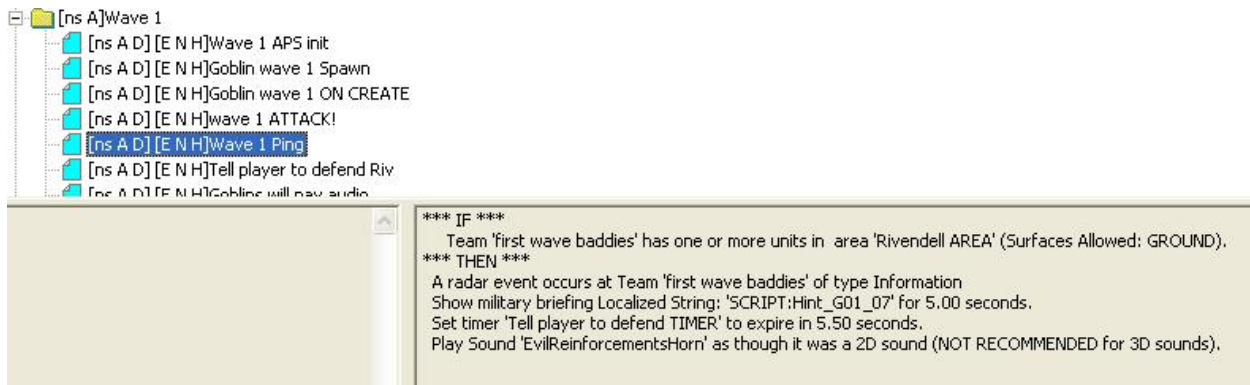
Another important thing to note: *APS' are *ONLY* used when the team is hunting. It will not affect the team's normal behavior in any way.* Use the Team – Actions → Attack or hunt → Team begins Hunting.



As you can see, this is how the attack is set up. The first action tells the team to hunt (using the APS assigned to it) The Sound playing is telegraphing the attack.

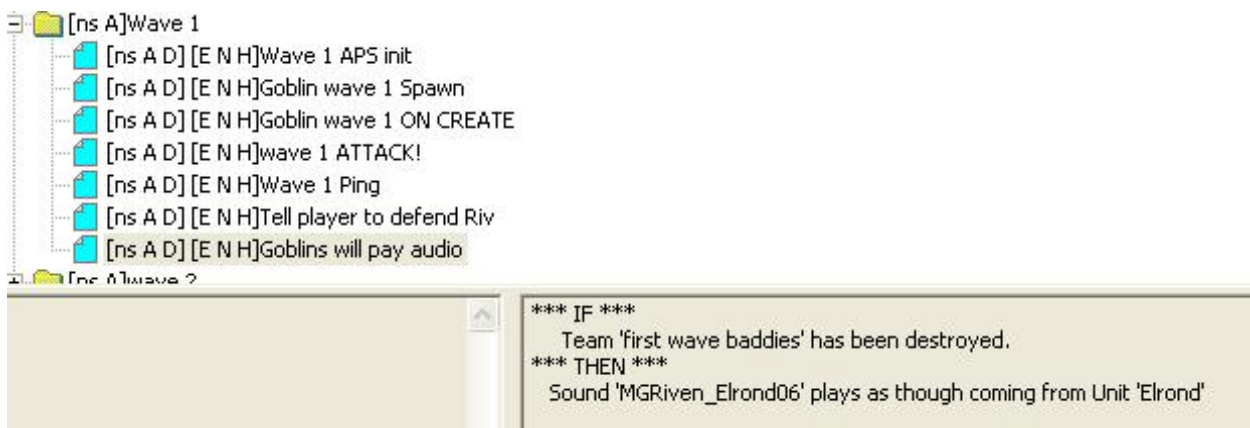
The second wave is set up the same way. The only major difference is the units that compose the team.

- **Keeping the player informed** - The next thing done for every attack is actually informing the player they are under attack.



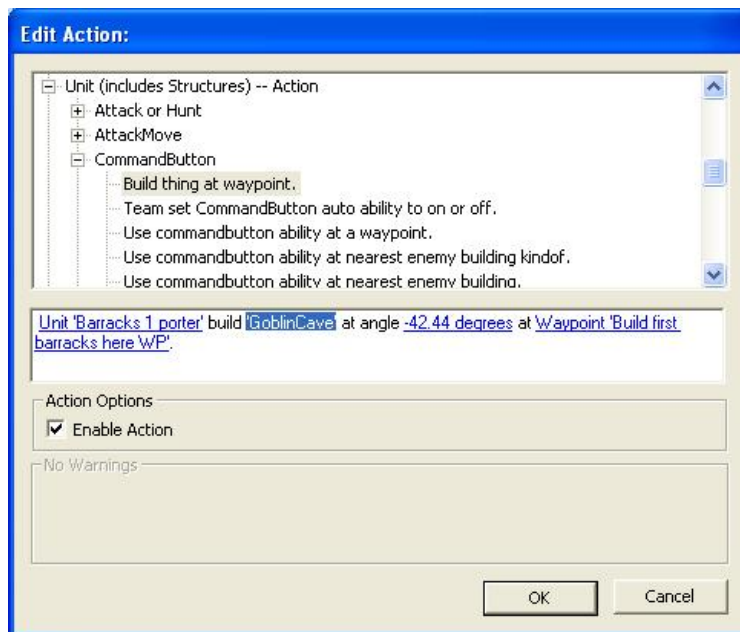
As you can see, once the attack team has entered the Rivendell perimeter the player is informed in every way possible of the incoming attacks short of reaching out of the computer and smacking him. You get a text display, a mini map ping, and multiple audio cues. Sneak attacks are cool, but only when used sparingly. These types of scripts usually are set to repeat every 40-60 seconds to help remind the player where they should be defending. Once the team is destroyed another script will disable the repeating script.

And of course, the player gets a mini- reward in the shape of Elrond talking some smack when the wave is destroyed.

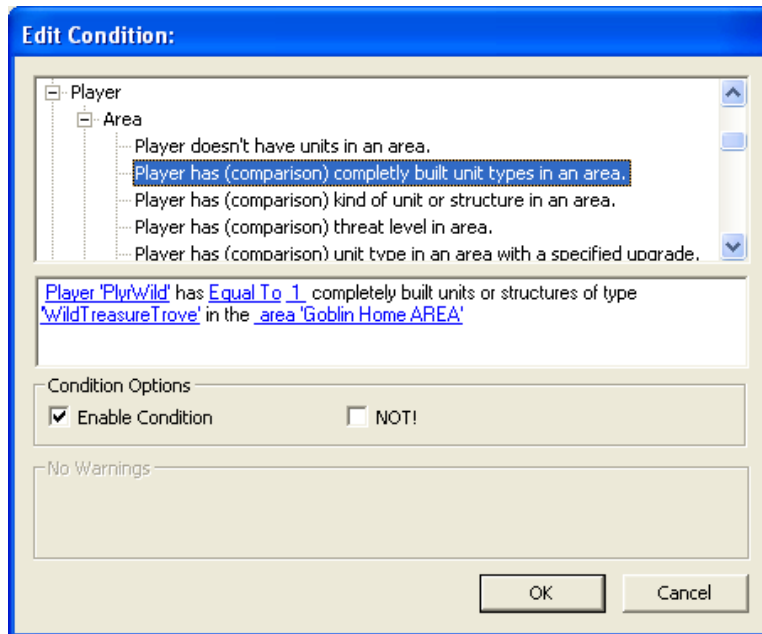


- **Base Building** - After Act II is finished, Act III is heralded by a lone goblin fighter horde running into the gate. Before I get into that attack though, lets back up a bit. While the player is occupied fighting off the two attack waves from the west, 4 lone porters walk on to the south. Each porter is pre-placed off the side of the map and named individually. They are also all on the 'PlyrWild/Porter Team'. This was done so I could control each individual independently, but also the team as a whole. These porters build up the goblin base in the south.

One thing to always keep in mind while making a mission in any RTS is that you never know where the player is looking at any given time. So whenever you're scripting actions, make sure you try to emulate what a player would actually do. Instead of just spawning in buildings and calling it a day, these porters walk on and use a special build script that allows you to build a structure at a WP with an angle. You can find this script here in:



The easiest way of figuring out the angle of placement is actually placing the object in the world. Once you get the angle you want, just copy the angle into the script parameters. Now it gets a little tricky. We have to detect when each structure has completed its build then give the porter an order to build the next structure. I put down an area trigger that will cover all of the area that the structures are going to be building. It is called 'Goblin Home AREA'. Here is the condition for testing whether or not the building exists:



As you can see, it basically detects for the correct number of ‘buildings of type’ in the area trigger. This is done instead of checking for player ownership because I have the porters set up to build a specific set of buildings. Trove Porter builds the structures around the Trove. Once it is done with the trove, it moves on to build the fissure. Once the fissure is complete it builds the guard tower. This is all done by making variations of the following script:

```

*** IF ***
  Player 'PlyrWild' has Equal To 1 completely built units or structures of type 'WildTreasureTrove' in the area 'Goblin Home AREA'
*** THEN ***
  Unit 'Trove porter' build 'GoblinFissure' at angle -42.44 degrees at 'Waypoint 'Build Fissure here''.
  Find unnamed 'WildTreasureTrove' Owned by Player 'PlyrWild' nearest team Team 'Porter Team' and reference as UnitRef 'Wild Treasure Trove'

```

You’ll also notice that the last action names whatever building was just finished. I’ll get to why I do that in “Ending a mission”.

- **Communicating threat to the player** - So, all the base building actually happens in Act II. Now we move on to Act III and back to that Goblin fighter horde spawning in. This little gag was set up with 2 simple scripts, a spawn script followed by an attack script. After the unit was spawned it, it set a timer. After the timer expired the unit was given and attack-move to WP. The WP was placed next to the House of Elrond. This baby attack was done to alert the player that there was a base to the south and to remind them not to get

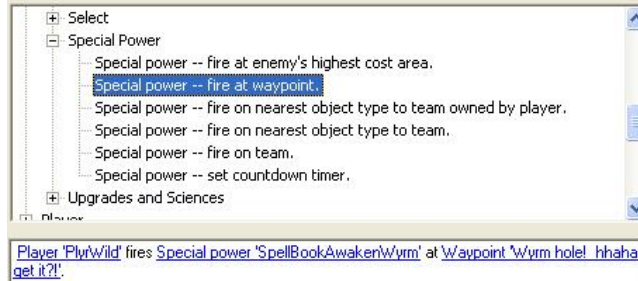
complacent. Having a unit actually attack from that direction makes the threat real.

- **Ending a mission** - The player is now told to ride out and destroy the goblin base. This is done once again with multiple audio cues, visual map pings and text. Remember how each goblin building was named after it was built? Well, here is why they were named individually:

```
*** IF ***
  Flag named 'Repeat Ride out objective FLAG' IS TRUE
  *AND* # of objects owned by player Player 'PlyrElves' with model condition ModelCondition State: ATTACKING is Less Than or Equal 0 .
*** THEN ***
  Show military briefing Localized String: 'SCRIPT:Announcement_G01_04' for 8.00 seconds.
  Sound 'MGRiven_ElronD07' plays as though coming from Unit 'Elrond'
  A radar event occurs at Unit 'PlyrWild/Fissure' of type Information
  A radar event occurs at Unit 'PlyrWild/Wild Barracks 1' of type Information
  A radar event occurs at Unit 'PlyrWild/Wild Barracks 2' of type Information
  A radar event occurs at Unit 'PlyrWild/Wild Barracks 3' of type Information
  A radar event occurs at Unit 'PlyrWild/Wild Barracks 4' of type Information
  A radar event occurs at Unit 'PlyrWild/Wild Mill 1' of type Information
  A radar event occurs at Unit 'PlyrWild/Wild Mill 2' of type Information
```

This is how I set up all the objective pings. Each objective has a name. Each name is pinged on the radar map. If the unit no longer exists, no ping will occur. So if the player destroys buildings, but maybe misses one along the way, that building will ping on the radar. This gives them only the information they need, as opposed to pinging an X,Y,Z coordinate which will ping the location whether there is anything there or not.

- **Boss Fight!** – To give the mission a little kick at the end, and to introduce and reinforce some of the new creatures you will encounter in BFME 2 in the first mission, we chose to end it with a Wyrms.
 - **Bringin' in the Wyrms** – This was a simple matter of casting the spell at a waypoint. In early play tests the the Wyrms was cast at the player unit closest to the goblin base and didn't pull the camera away to where the summon occurs. This resulted in the player missing the actual summon about half the time, and then the player was left wondering what to do next. So we went with the camera pull and predetermined summon spot. The players never missed the Wyrms and knew exactly what needed to be done to finish the mission.
 - **Scripting the Wyrms** – This was a simple matter of using the “<Player> fires special power 'SpellbookAwakenWyrms' at Waypoint” that can be found in Player→ Special Power.



- The whole sequence was 3 scripts.
 - The first Moves the Camera to the WP where the wyrm will be summoned

```

*** IF ***
  Player 'PlyrWild' has Less Than or Equal 0 unit or structure of type 'player wild LIST'
*** THEN ***
  Start letterbox mode (hide UI, add border).
  Move camera to Camera 'Worm hole! hhaha get it?!' in 2.00 seconds, camera shutter 0.00 seconds, ease-in 0.50 seconds, ease-out 1.00 seconds.
  Set timer 'Worm camera moving TIMER!' to expire in 2.00 seconds.
  The map is permanently revealed at Waypoint 'Worm hole! hhaha get it?!' with a radius of 300.00 for Player 'PlyrElves'. (Afterwards referred to as Reveal Name: Wyrms REVEAL).
  Disable Script 'PlyrElves/Repeating ride out!'.
  Disable mouse and keyboard input.

```

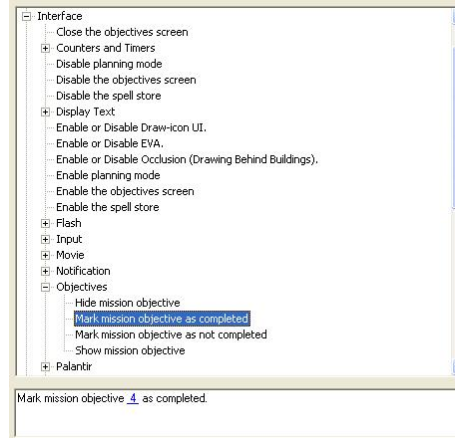
- You can see that this triggers when the Goblin Player has zero structures. This was done to ensure the player ONLY had to worry about the Wyrms.
- The first action in the script enables the letterbox mode. That brings in the black bars at the top and bottom and gives the cinematic feel.
- After that is enabled, it begins moving the camera to the same WP that the wyrm summons at. The 2 second camera move ensures that the player doesn't miss the summon animations. A quick camera move also gives the player a sense of urgency. The timer is set to trigger the next script in the series.
- The last actions reveal the area so the player doesn't have to worry about sight ranges during the bossfight. It then turns off the previous objective dialogue script. The last action turns off the User Interface so the player can't scroll or interrupt the camera move.
- Once I bring the Wyrms in I enable the win condition script, set a timer to reference the wyrm and then another to enable the win condition. You can see the trigger for this script is the timer set up in the previous script.

```

*** IF ***
    Timer 'Wyrms camera moving TIMER' has expired.
*** THEN ***
    Player 'PlyrWild' fires Special power 'SpellBookAwakenWyrms' at Waypoint 'Wyrms hole! hahaha get it?!'.
    Set timer 'Wyrms reff TIMER' to expire in 10 frames.
    Set timer 'win enable delay TIMER' to expire in 5.00 seconds.
    Mark mission objective 4 as completed.

```

And then finally I mark the previous mission objective complete. That script can be found in the Interface → Objectives tree.



- The final script for this gag sets a reference to the wyrm. This is done to detect when the player destroys it to trigger the win condition.

```

*** IF ***
    Timer 'Wyrms reff TIMER' has expired.
*** THEN ***
    Find 'Wyrms' Owned by Player 'PlyrWild' nearest team Team 'PlyrElves/teamPlyrElves' and reference as UnitRef 'Da WYRM'
    Show military briefing Localized String: 'SCRIPT:Hint_G01_11' for 8.00 seconds.
    Play sound Sound 'MGRiven_ElfOfficer063' as though coming from a member of Team 'PlyrElves/teamPlyrElves'.
    Set timer 'use eagles hint TIMER' to expire in 10.00 seconds.
    End letterbox mode (show UI, remove border).
    Enable mouse and keyboard input.

```

As you can also see, it gives the player some audio and text lines to inform that player of the wyrm. You'll notice that the audio plays from "Team -'PlyrElves\TeamPlyrElves" to guarantee that the player will hear the audio as if it is coming from his soldiers. The last and most important thing enables UI and also ends the letterbox mode.

I'll get to why I do that in "Ending a mission".

- **Map.ini** - The Wyrms were modified to give them a big bang. It has a modified body that has increased hitpoints to make it more of a challenge. We set this via the Map.ini file. Below are the actual lines from the map.ini with a bit of explanation as to what is going on here.

```
Object Wyrms
  RemoveModule ModuleTag_LifetimeUpdate
  ReplaceModule ModuleTag_Body
    Body = ActiveBody ModuleTag_tuffwyrms
      MaxHealth = 30000 ;WYRM_HEALTH
  End
END
END
```

- **The first line is the name of the actual object to be modified.**
- **The second removes the un-summon timer. So the Wyrms does not Unsummon after 30 seconds.**
- **The third line replaces the old body module. The ModuleTag name MUST be that of the old module that it is to replace.**
- **The fourth line names the type of module “body” and the new name of the module “ModuleTag_tuffwyrms”**
- **The fifth line sets the new health to 30,000**
- **That last few lines end each module. The first end is for the body module, the second for the replace module command and the last for the entire object itself.**

That was a basic insight into Good Mission 1: Rivendell. I covered the basic layout of scripting ambushes, a few of our basic design rules of thumb and a little insight as to why I made the choices I made. We hope you enjoyed it and now feel ready to make some fantastic single player missions on your own!

-Adam Hayes, Level Designer