

## Doomsday Engine - Feature #1986

### [MP] Configure what the server does when an episode ends

2015-03-02 08:52 - skyjake

<b>Status:</b>	New	<b>Start date:</b>	2015-03-02
<b>Priority:</b>	High	<b>% Done:</b>	0%
<b>Assignee:</b>	skyjake		
<b>Category:</b>	Enhancement		
<b>Target version:</b>	3.x		
<b>Description</b>			
Currently if an episode ends, the server will get stuck in the finale animation and the only way forward is to manually change to a different map on the server using a console command. It would be much preferable if the server admin could choose what happens next, e.g., continue to the next episode or start again from the first episode.			
In an ad-hoc server setup, the clients should have the option to vote on which episode to start next.			
<b>Related issues:</b>			
Copied to Feature #2413: [MP] Configure what the server does after a timeout ... <b>New</b> <b>2015-03-02</b>			

#### History

##### #1 - 2015-03-02 08:52 - skyjake

- Target version set to 50

##### #2 - 2015-03-03 06:57 - danij

Perhaps this functionality should be combined into the map cycle expression? The map cycle is already functioning at a level which assumes a "current" episode, so if the cycle supported a start-new-episode directive then one could use a single expression to describe the whole sequence (autogenerating cycles from the available data would be convenient and straightforward).

##### #3 - 2015-03-03 10:39 - skyjake

If we were to retain the current map cycle implementation, sure, that would make sense. However, I would like to replace it with Doomsday 2 components and take advantage of Doomsday Script for more powerful behaviors (thus getting rid of the "cycle script" syntax). In broad terms, the map cycle should be both configurable from a GUI with some basic rules, and it should have scriptable hooks where one can specify additional logic and actions.

However, looking at this from an even wider perspective, the map cycle is just a special kind of custom Episode — it would make sense to enhance the map progression logic to allow handling MP map cycles & rules, too (with support for scripts and InFine triggers).

##### #4 - 2015-03-03 10:42 - skyjake

However, to tackle this issue for the near term (with least amount of effort), even just restarting the cycle (if a cycle is defined) or continuing to the next episode as a hard-coded behavior would suffice.

##### #5 - 2015-03-03 12:58 - danij

I sincerely hope you are joking re using Episodes to represent a cycle of episodes. It makes zero sense other than in an extremely loose theoretical case.

##### #6 - 2015-03-03 22:03 - danij

I'm struggling to see a tangible benefit in implementing such cycles as scripts in and of themselves. Conceptually, a "map cycle" only ever made sense in deathmatch. TBH I would prefer to remove all the map cycle mechanics entirely as they no longer make sense at all IMO.

In fact I don't see the need for a system to handle what happens when the episode the server admin has configured is completed. The obvious solution is to simply start the episode over following a suitable interlude. Changing episode is to reconfigure the server, so should be addressed using suitable remote administration mechanisms plus a voting system.

##### #7 - 2015-03-04 11:43 - vermil

STAGE 1: Dealing with the main issue

My opinion is that by default, an episode should loop back to its beginning when the end is reached (i.e the server admin has specified absolutely no options for map progression on the server). This should be defined by the episode definitions the server has loaded and should be used even if the server admin has started the server on or warped the clients to a map in the middle of an episode once the server has started.

i.e. E1M1, E1M2...E1M7, E1M8 loop E1M1...  
i.e.2 Warp E1M5, E1M6, E1M7, E1M8 loop E1M1...  
i.e.3 E1M1 Warp E2M7, E2M8 loop E2M1...

I believe implementing something that deals with the hanging by default, is by far the most important thing in this report. Anything else (i.e. below this point) is more of a feature request.

STAGE 2: Displaying this information for clients

The first stage in communicating the above to clients should be the server information simply saying either 'original map progression' (i.e the episode def's for the lwads are being used) or 'custom' (i.e the server admin is using a custom episode def).

The second stage would be adding an optional additional drop down in the server information that lists each map in the episode def the server is using.

STAGE 3: Future RFE ideas

The server admin should get an additional option to make map progression loop back to the map they started the server on, if they started the map in the middle of an episode.

i.e. E1M5, E1M6, E1M7, E1M8 loop E1M5, E1M6...

ADDENDUM: Map Cycles; the existing additional option

I personally believe Map Cycles are useful in all MP modes and should remain usable by the server admin in all game modes.

The question is what happens when the end of the map cycle is reached; should Dday automatically loop the cycle or should it allow map progression to continue on to the end of the episode the last map in the cycle is a part of? Personally, I think it should loop the cycle.

I also think the Map Cycle options should be expanded upon to include the ability to make a cycle out of whole episodes (or hubs), rather than just map numbers . But this is something for the upcoming Shell GUI: i.e the user loads an add-on in the GUI and various GUI options such as 'start map', 'map cycle' etc, attempt to read the maps and episode def's in said add-on and automatically update (and by default already contain the original map progressions if no add-on is loaded, such as the start maps of each hub in the original HeXen).

#### **#8 - 2015-03-04 12:52 - danij**

I agree that the most logical course of action would be to simply restart the current episode once the finale has ended. Clients should be made aware the episode is starting over should they wish to take action beforehand, so a short countdown timer would be ideal.

With regard to the other points raised thus far, however, it is important to bear in mind the server admin's desire to limit what clients can do on their server. We can't simply allow clients to randomly mix up episodes and maps without the admin's prior consent. Warping to another map in the same episode has no affect on this. However, changing episodes once the server has started may potentially circumvent the admin's control (assuming no whitelist mechanism is employed).

Also, now Doomsday has a Ring Zero and clients are free to leave and join servers as they wish - there is much less need to dynamically reconfigure the current server. If clients want to play a different episode then joining another server is much less of a issue (assuming suitably configured server is running of course. Although the availability thereof should not dictate high-level behavior!).

If one considers the map cycle as an alternative to an Episode definition then the user should define it as such (or we could autogenerate, whatever...). The important thing to note is that a Episode def supports everything the existing map cycle does other than the timed exit conditions. I agree that it would be useful to extract this functionality into scripts, however it makes no sense to script up the old map cycle instead when it makes far more sense to leverage the new custom episode stuff.

#### **#9 - 2015-03-04 12:55 - vermil**

To make clear, everything I said above was meant to be server admin controlled; none of it was meant to be client controlled.

A client side voting system comes after admin controls are fully implemented IMO.

#### **#10 - 2015-03-04 13:02 - danij**

Thinking about the "timed exit conditions" of the map cycle - these are actually game rules. If the exit conditions are converted to game rules and the cycle itself is defined in an Episode def, then there would be nothing remaining of the old map cycle mechanism.

#### **#11 - 2015-03-04 13:32 - skyjake**

danij wrote:

implementing such cycles as scripts

That is not what I meant. The idea I failed to describe in my haste writing the comment was that episodes, map exit conditions (for single or multiplayer), and the server's map "playlist" (cf. music playlist) are all very closely related concepts. In object-oriented programming, closely related concepts usually have common base classes or other common functionality — therefore these features should not be thought of as separate things, but aspects of one system.

The scripts come into play when you want to do things like show InFine animations, determine special victory conditions for MP games, do advanced in-hub warps, etc.

I get the feeling that we are indeed talking about the Game Rules system.

**#12 - 2015-03-04 13:41 - danij**

danij wrote:

If the exit conditions are converted to game rules and the cycle itself is defined in an Episode def, then there would be nothing remaining of the old map cycle mechanism.

Actually, that is not entirely correct - the randomized map progression made possible via the map cycle would have no counterpart in an Episode + Game Rules solution. However, one must consider whether it is logical to randomize the map progression - to do so in co-op makes no sense - its only really in deathmatch modes where maps are played in isolation, with otherwise no other ongoing progression of the high-level game state. (Although with that said, a Game Rule could well be "when an exit condition is met - override the defined progression and pick a map at random from the current episode" -- this would support the existing use case for a randomized map cycle in deathmatch).

**#13 - 2015-05-20 16:27 - vermil**

- File *heretic-ext-001.jpg* added

I have attached an image that I suppose is sort of related to this.

**#14 - 2016-03-27 08:04 - skyjake**

- Target version changed from 50 to 2.1 (Late 2018)

**#15 - 2018-07-29 16:53 - skyjake**

- Target version changed from 2.1 (Late 2018) to Multiplayer

**#16 - 2020-02-05 07:19 - skyjake**

- Copied to Feature #2413: [MP] Configure what the server does after a timeout when no clients connected added

**#17 - 2020-02-05 07:19 - skyjake**

- Assignee set to skyjake

- Target version changed from Multiplayer to 2.3

**#18 - 2020-12-08 08:27 - skyjake**

- Target version changed from 2.3 to 3.x

**Files**

---

heretic-ext-001.jpg	344 KB	2015-05-20	vermil
---------------------	--------	------------	--------