

---

Subject: Re: Syncing or changing BuildingGameObj 'IsDestroyed' state for clients  
Posted by [Jerad2142](#) on Fri, 16 May 2014 14:48:19 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

danpaul88 wrote on Fri, 16 May 2014 07:56Jerad Gray wrote on Fri, 16 May 2014  
14:53danpaul88 wrote on Sun, 04 May 2014 10:02

All GameObjects have a guaranteed lifecycle whereby they transition from Created -> Destroyed, usually via Killed unless removed from the level by a script or other event. Scripts attached to that object are free to make the assumption that they can clean up resources once Killed or Destroyed are called. Subsequently causing that GameObject to be alive again and triggering Damaged events or another Killed / Destroyed event in this case would cause that script to be in an invalid state and could result in a crash.

I don't think this statement is quite right, unless the lifecycle does not have to wait for all scripts with a Created function to complete; because, I have encountered plenty of times (both in 3.4.4 and 4.1) that Created fired after Destroyed, I've got a few tricks that prevent destroyed from continuing if created of a script hasn't been completed yet as I've had crashing issues in the past due to uninitialized objects and variables

I don't see how that could happen unless you're destroying objects in the initial Created cycle for all game objects, which is inadvisable and rather inefficient (why create objects if you're just going to destroy them instantly on game start?). If you can provide a reproduction case for that we will look into adding a fix in the code to ensure Created is always called before Destroyed (even if we end up doing something like if (!created) {Created();} Destroyed();)  
Last time I ran into it was caused by changing the player character on spawn, but that is one of the rare occurrences where I had complete control and could have found a way around it besides just making a flag external to the script to check.

In the wild, I've seen it caused by things ranging from a separate timer doing clean up, to something a player does (for example once I saw a player die while at a PT (due to lag or a lockup which ever the case was they were still in the PT after their char died) and just as the player's character spawned they bought a character from the PT, the destroy flag on the scripts attached to the spawn character were called before the create due to the character being 'destroyed' and replaced with a new character.

Mind you, this is only a problem if your storing an array index, or a pointer in your script (playerId for efficiency, or a pointer to a list being the most common occurrence).

Normally I just protect against it now by attaching a dummy script as the final step of create and then checking for the attached script on any classes that I've found can be called before create (Custom, Damaged, Killed, and Destroyed are the ones that come to mind) on any sensitive scripts (in all of ECW I've done this less than 50 times, and that's a good 4.27MB of code.

---