Subject: Re: Question about pick behaviour
Posted by mr_ped on Mon, 28 Mar 2011 14:23:17 GMT

ross_tang: once your two references would start to live in quite different parts of complex app on their own, it would be quite difficult to decide which one is responsible to finish life cycle of that memory block, ie. you would need GC to make the life easier. Which means you failed to find an U++ way of design where you know where that memory block belongs and when you are done with it and you can lost it completely, so there's no need for GC, you just exit from that stack window away.

It means U++ gives less freedom in implementation to force you for a bit more thoughtful design in exchange for no GC performance penalty and the final source is usually simpler and easier to maintain.