Zeus Documentation

By Kaukassus & H4b4sch

Inhalt

[Introduction 1](#_Toc385781856)

[How it works 1](#_Toc385781857)

[Buyable Objects 1](#_Toc385781858)

[Command Points Synchronization 2](#_Toc385781859)

[Zones 2](#_Toc385781860)

# Introduction

Zeus joined the game. Currently (as of 20.04.2014) there can be up to 5 Zeus players which are just the first 5 player slots. Later there will be explicit slots for Zeus’.

A Zeus player can buy and control the same Units as a normal Player in the HQ could. They can be placed around the HQ and each FOB. Additional to that are buyable structures. They can be separated by Defending Structures and Appearance Structures. Defending Structure, like Walls or a HMG, cost Command Points just like units. Appearance Structures, for example a mapboard, are just to make it look more real, so they are free to buy.

The Resources of each Zeus are always the same as the Command Points. The Resources of each Zeus get synchronized to them.

# How it works

### Buyable Objects

In a new update, I changed the way the buyable Object are determined. Now there’s the script *setBuyable.sqf* which contains an array for each Object.  
While I changed the scripts for the “Request Unit” in HQ and FOBs a created a new one for Zeus, *Zeus\setPrice.sqf.* The setting of which objects can be buyed and their price is made in an Event-Handler which is executed when the player of the respective Zeus enters the Zeus Interface.  
More Information on that: <https://community.bistudio.com/wiki/Curator#Assigning>

### Command Points Synchronization

Pretty much the most complicated part. Mainly, the Command Points are the central resource and every Zeus gets synchronized to them. However of course changes on the Zeus resources must be synchronized back to the command points. This is done in *Zeus\commandPointsToZeus.sqf*  
Actually, in every Cycle (each second), I firstly look up if there are changes on the resource of each Zeus. For that I saved the resource value per Zeus in an array in the cycle before. If there happened a change, the Command Points change their value respective to the difference. The Zeus Points, at least at this point, stay the same.  
But, after I changed the Command Points by every Zeus, every Zeus is synchronized to it. Here I do not change the Zeus resources respective to the difference, but first set them to zero and then just fully add the Command Points value. **At the moment Zeus Points can only be added, but not set to a value.**

### Zones

The Zeus shouldn’t be able to build objects everywhere. Because of that there are zones, which are always a circle. Around the HQ there’s a circle with a radiant of 100 meters. This is set directly in *init.sqf*. For the FOBs, there’s a script *Zeus\zones\_fob.sqf.*I just go through each entry in a global array of FOBs (this already existed) and add an editing area with a radiant of 50 meters. Because the Zones have an ID, I made a variable which starts by 1 every cycle and counts ab with every FOB. If the FOB already exists the respective Zone gets changed. Because it has the same position and radiant it will not actually change (of course).  
Also there’s a small zone around the MHQ.