**Documentação CXP Framework**

***By Casperento***

A seguir trago algumas instruções acerca de como configurar a framework (ou missão) apropriadamente. Lembre-se de que esta missão não é 100% igual à distribuída no repositório oficial AsYetUntitled no GitHub. Com mais de 1 ano de desenvolvimento, nesta versão (7.0.0) temos algumas novidades como a passagem de funções do cliente para o servidor. Confesso que não é a melhor forma de proteger códigos no Arma 3, mas é suficiente para evitar que aqueles editores idiotas da comunidade brasileira não copiem nosso trabalho.

Vamos lá. Começaremos pela pasta do **Cliente (cxpf.Altis).** Aqui você encontrará arquivos de mídia e alguns códigos pertinentes para a montagem do cliente dentro do servidor.

***Pasta cxpf.Altis/comunidade:***

Esta pasta contém algumas funções que eu decidi pegar no fórum do altisliferpg.com e redistribuir com minhas modificações.

***Pasta cxpf.Altis/config:***

Esta pasta contém todos os arquivos de configuração da missão. Dentre eles devemos destacar o arquivo **CfgMaster.hpp** e **CfgAddActions**. O primeiro envolve váriaveis básicas de configuração do cliente como quanto deve ser o salário de cada um, quantos policiais são necessários para fazer uma reserva, etc… O segundo é novo nesta versão da framework e deve ser manipulado com **MUITO CUIDADO** para não quebrar o servidor inteiro! Nele você colocará o argumento que a função nativa (**addAciton**) do jogo requer para exibir aquelas opções do scroll das placas. No geral, quando estamos no editor, abrimos o menu de contexto da plaquinha e colocamos o seguinte código no campo “Init”:

this addAction[“Abrir menu da loja de armas”,cxp\_fnc\_funcaoDeAbrirLojaDeArmas,(argumento da funcao anterior),0,FALSE,FALSE,“”,'isNull objectParent player && player distance \_target < 4'];

Dessa maneira damos a opção de abrir um menu para a plaquinha em questão. Já no **CfgAddActions.hpp** faremos a mesma coisa porém com pequenas modificações:

(Pegaremos o colchete e seu conteúdo e colocaremos ele dentro de aspas duplas “”, lembre-se de que cada linha é um array que segue a seguinte estrutura:

{nomedaplaca,“primeiro addaction”,“segundo addaction”, etc…}):

{weapon\_shop, “[“Abrir menu da loja de armas”,cxp\_fnc\_funcaoDeAbrirLojaDeArmas,(argumento da funcao anterior),0,FALSE,FALSE,'','isNull objectParent player && player distance \_target < 4']”}

(Agora temos um problema de sintaxe de substrings, para resolver trocaremos as aspas duplas de dentro da string principal para um apóstrofo *'* )

“['Abrir menu da loja de armas',cxp\_fnc\_funcaoDeAbrirLojaDeArmas,(argumento da funcao anterior),0,FALSE,FALSE,'','isNull objectParent player && player distance \_target < 4']”

(OBS: Os apóstrofos em vermelho não precisam ser modificados!)

Agora que você sabe configurar todas as opções do scroll do mapa inteiro basta colocar uma plaquinha em algum lugar e dar um nome para ela de acordo com o que você digitar no array do arquivo de configuração dos addActions. No nosso caso teríamos que criar uma placa com o nome de **“weapon\_shop”**. Sendo assim, sua placa já deverá estar funcionando dentro do servidor!