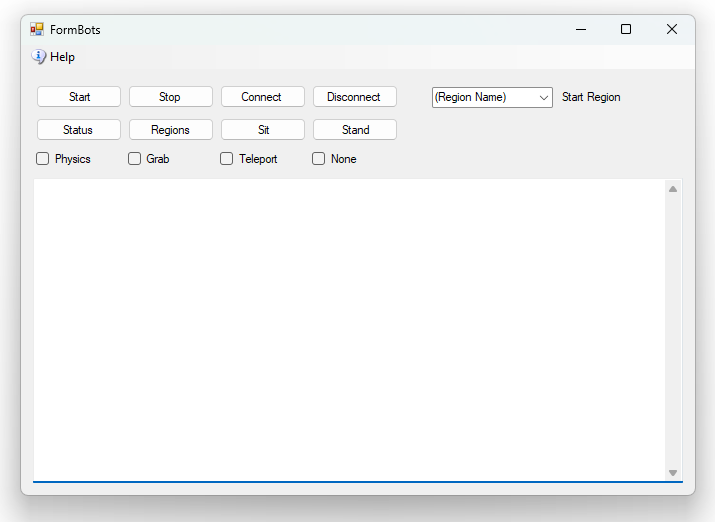
# Outworldz DreamBot server

Outworldz DreamGrid has a pCampBot Server. A bot is a computer-controlled Avatar, like an NPC. However, a Bot is an actual Avatar.

This is the Control Panel from Setup-Settings-DreamBots:

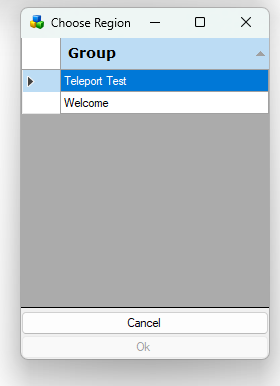


Bots are easy to run with this panel.

## Run the Bots

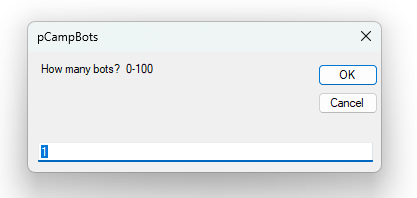
Click Start. The system will ask you to choose a region.

Poick a region and click Okay, or double click the region to selecty it quickly.



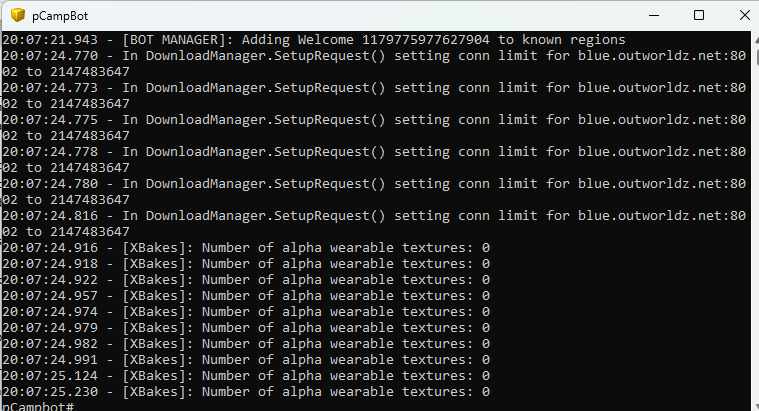
## Click Start

DreamGrid will ask how many bots you wish to run. The default is 1 bot. It takes a high performnce c omputer to run 100 bots,



Click OK and Dreamgrid will create the bots in Robust and then run the Bot control panel.

When using this you will get a console command line like the one in OpenSimulator and Robust. Type "help" at this prompt to see further commands.



You do not need to interact with this panel.

A help menu will sppear:

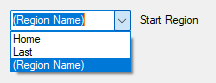
For more information, type 'help all' to get a list of all commands,or type help <item>' where <item> is one of the following:  
add behaviour <abbreviated-name> [<bot-number>] - Add a behaviour to a bot.  
connect [<n>] - Connect bots.  
disconnect [<n>] - Disconnect bots.  
quit - Shutdown bots and exit.  
remove behaviour <abbreviated-name> [<bot-number>] - Remove a behaviour from a bot.  
set bots <key> <value> - Set a setting for all bots.  
show bot <bot-number> - Shows the detailed status And settings of a particular bot.  
show bots - Shows the status of all bots.  
show regions - Show regions known to bots.  
show status - Shows status.  
shutdown - Shutdown bots And exit.  
sit - Sit all bots on the ground.  
stand - Stand all bots.

You do not need to know all these commands. There are buttons for all of them already set up.

## Region Selection



The region selector has three choices. Home which is your Welcome Region, Last which h is the bots last region it logged in, and {Region Name}



If you choose this last setting, or make no choice, the system ask you for the region:

## Start, Stop, Connect and Disconnect

This row of buttons controls the Bot Windows:



* Start - This asks for how many Bots, then if it does not find them, makes them in Robust. It then starts the Bot Window. The bots start and you will see a familiar console command line as you would on a simulator or Robust console. As with the other consoles, you can type "help" to get a list of commands. The bot log onto (connected) to the region. You will see the connection messages on the console and if all goes well, you'll see the bots appear on the region.
* Stop – Disconnects all Bots, then closes the Bot Window.
* Connect – Reconnects the Bots to the simulator.
* Disconnect - logs the bots out but leaves the window open. You can change the bot settings and click Connect to start them.

## Status, Regions, Sit and Stand Buttons

The second row has buttons to see and control Bots



### Status:

Runs the show bots command:

pCampbot# show bots  
Name Region Status Conns Behaviours  
Iama Bot0 Teleport Test Connected 1 p  
Disconnected : 0  
Connecting : 0  
Connected : 1  
Disconnecting : 0

## Regions

Shows all registered regions.

## Sit

Issues a sit command to all Bots. This will cancel all other behaviour.

## Stand

Issues a stand command to all Bots. The bots will resume other behaviors, such as clicking prims, walking, flying and teleport, if these are enabled.

## Behavior Control Checkboxes

The check boxes control what the bots will be doing once they are connected



Physics: bots constantly move and jump around

Grab: Bots randomly click prims whether set clickable or not.

Teleport: bots regularly teleport between regions on the grid. If you select Teleport behavior, the bots will travel to all the registered regions. This may not be what you want to do. It may be best to go to Setup-Settings->Regions and deregister all regions, and then start the system.

## Performance Checkboxes

There are two checkboxes for more control for putting a load on the server.

### Send Agent Updates checkbox

Controls whether bots should regularly send agent updates. Not doing this will reduce CPU requirements but reduces the realism compared to viewers which are constantly sending Agent Update UDP packets.

### Request Object Textures

Controls whether bots will request textures when receiving object information. Not doing this will reduce CPU requirements but reduce the realism compared to viewers which requests such texture data if not already cached.

## Assessing Performance

There is currently no automated way of assessing simulator or connection performance. Manual ways to assess this include logging in a viewer to the same region to get a feel for avatar movement (e.g. is it jerky, is there rubberbanding, is chat delayed, etc.) and to see viewer statistics such as Ping Sim, Packet Loss, Physics Time (frame) and Spare Time (frame).

However, one needs to be extremely careful if logging in using a viewer on the same machine from which the bots are running. The bots should really be running from a different machine or ideally even from a different network to get a more exact picture of current simulator performance.

One can also look at server-side monitoring, particularly "show stats clientstack" which will show various client stack related statistics (this is the part of the OpenSimulator code that handles receiving and sending of UDP messages from viewers). In particular, the InboxPacketsCount should be constantly zero or near 0 - a build up here means that your simulator is receiving UDP packets quicker than it can process them.

Another useful command is *debug lludp packet <level> <avatar-first-name> <avatar-last-name> -* Turn on received packet logging.

Once you have finished, you can disconnect the bots in an orderly fashion with the Stop button.

A pCampBot synthetic bot load is still an extremely poor proxy for a 'real' connection (i.e., a real human being logging in with a viewer and doing all the things they expect to be able to do in a virtual world). Real connections are very likely to place different stresses on the simulator so real-world performance may be considerably different and poorer than synthetic load tests would suggest.

There are some things you can do to improve this in pCampbot. For instance, if you have RequestObjectTextures then pCampbot will request textures for all objects it sees. This is both much less stressful than a real scenario (since for a particular set of bots, textures are only requested once) and more stressful (since there is never any caching).

pCampbot does currently not work well if there are any surrounding regions. Avatars will currently just walk off into oblivion. You must either build fences or remove such regions. In testing so far, the presence of neighboring regions does not have any significant impact on performance in the region occupied by bots. However, these tests were performed with blank surrounding regions - surrounding regions containing objects will place more initial stress on the bot connections as object data is downloaded (and textures if this option is active).