

MECAC WEEKLY REPORT (April 1 – April 7)

This week we have worked separately to accomplish several tasks in Virtual Turkey project. Mert has worked on SVN, MSI builder, and Test project with Umit. We have setup a local SVN in MODSIMMER. Checkout problems related to older svn versions in MODSIMMER has been alleviated. We were unable to check our source code to CENG servers. This is because the commit operation failed in Subversion software. We are investigating the cause of this problem, and promise to deal with this issue next week.

We have downloaded an MSI builder and started to specify the locations of the files during a setup environment. As Client solution has 4.458 files in its debug folder, we have decided not to use default MSI builders but to start our own from scratch. We are also considering the possibility of buying Install Shield for installation, but that's a far possibility as its expensive.

Umit has worked on increasing number of simultaneous test clients in the test project. For this, we have worked on test client software to decrease memory and CPU usage for each client. We have tried to remove windows forms graphics objects such as buttons and labels from test clients. However, this caused new problems in network logic as they are somehow connected. Moreover, memory footprint was not affected much by these modifications. Following addition has caused significant improvement in reducing resource usage:

```
private void UpdateViewMatrix()
{
    if (!test)
    {
        Matrix cameraRotation = Matrix.CreateRotationX(updownRo
```

We have modified the test clients not to paint the virtual world and update the view matrix. CPU usage of each client is reduced by half. Surprisingly memory usage has been also decreased by this modification. Before this change, a test client would consume around 60MB of memory and 2% of CPU time in an intel i7 computer. After we have prevented the test clients from painting canvas and using model matrix parameters, each test client started to consume around 40MB of memory and 1% of CPU time.

That modification has enabled us to increase the number of concurrent test clients in a test environment. We have seamlessly tested the server with over 15 clients on a single i7 CPU. Although the clients have consumed around 550MB of memory and 13% percent of CPU time, server has consumed much of its resources to simulate the network connections. Next week we plan to reduce the memory usage as well to support more clients simultaneously.

Cinar has worked on server database to alleviate problems occurred during testing the game server. He has examined the source code to find out database connections via SQL. We have found out that the server actually uses MySQL rather than Microsoft SQL to store its information. We have connected to MySQL through TCP socket with the credentials we have observed from the source code. We have deleted the erroneous accounts and their associated information from the database in the server side. As a result of this change, we were able to add imaginary test clients to project and simulate the network traffic with more than 15 clients concurrently. We have configured the test project to send the client index via command line argument.

```
⊞ unsigned __stdcall Thread(void *ArgList) {  
    HANDLE hEvent = *((HANDLE*) ArgList);  
    int first = 1;  
    while (WaitForSingleObject(hEvent, 0) != WAIT_OBJECT_0) {  
        first = 0;  
        char buff[255];  
        char buff2[255] = {'\0'};  
        int radix = 10;  
  
        strcat(buff2, "Client.exe test ");  
        strcat(buff2, itoa(ClientIndex++, buff, radix));  
  
        system(buff2);  
    }
```

Test project runs a number of threads to start number of test clients. Each thread supplies its index to client software. Client accordingly figures out its username and password from its client index.

First client uses mecacrobot0 as a username and password, second client uses mecacrobot1 as its username and password. Therefore, each test client logs into the server with its credentials. Since Cinar was able to log into SQL and insert imaginary test clients to the server database, test clients are able to log in game server without a problem.