GHOST RESOLUTION

Exercise Files

Starter – "Kit/gpgt/server/scripts/gpgt/chapter3/exercise003.cs"
Answers – "Kit/gpgt/server/scripts/gpgt/chapter3/answers/exercise003_f.cs"

Exercise Mission

Chapter 3: "003_GameConnection: Ghost Resolution"

Synopsis

In these exercises, we will learn to use the various ghost resolution features provided by the engine by applying what we discussed in the book to some concrete examples of ghost resolution.

Prerequisites

1. *ch1_001.pdf "Using The Kit"*
2. *ch3_001.pdf "Named Connections"*
3. *ch3_002.pdf "ClientGroup"*

Exercises

GHOST RESOLUTION

1 Counting Ghosts

Goal: Modify the final code from exercise "ClientGroup" to count the number of server objects that are being ghosted on each individual client connection.

Starter Code: You are provided with a mission starter that places three AIPlayer objects in the scene. Figure 1 below shows the scene as it should appear when you load the mission for this exercise.

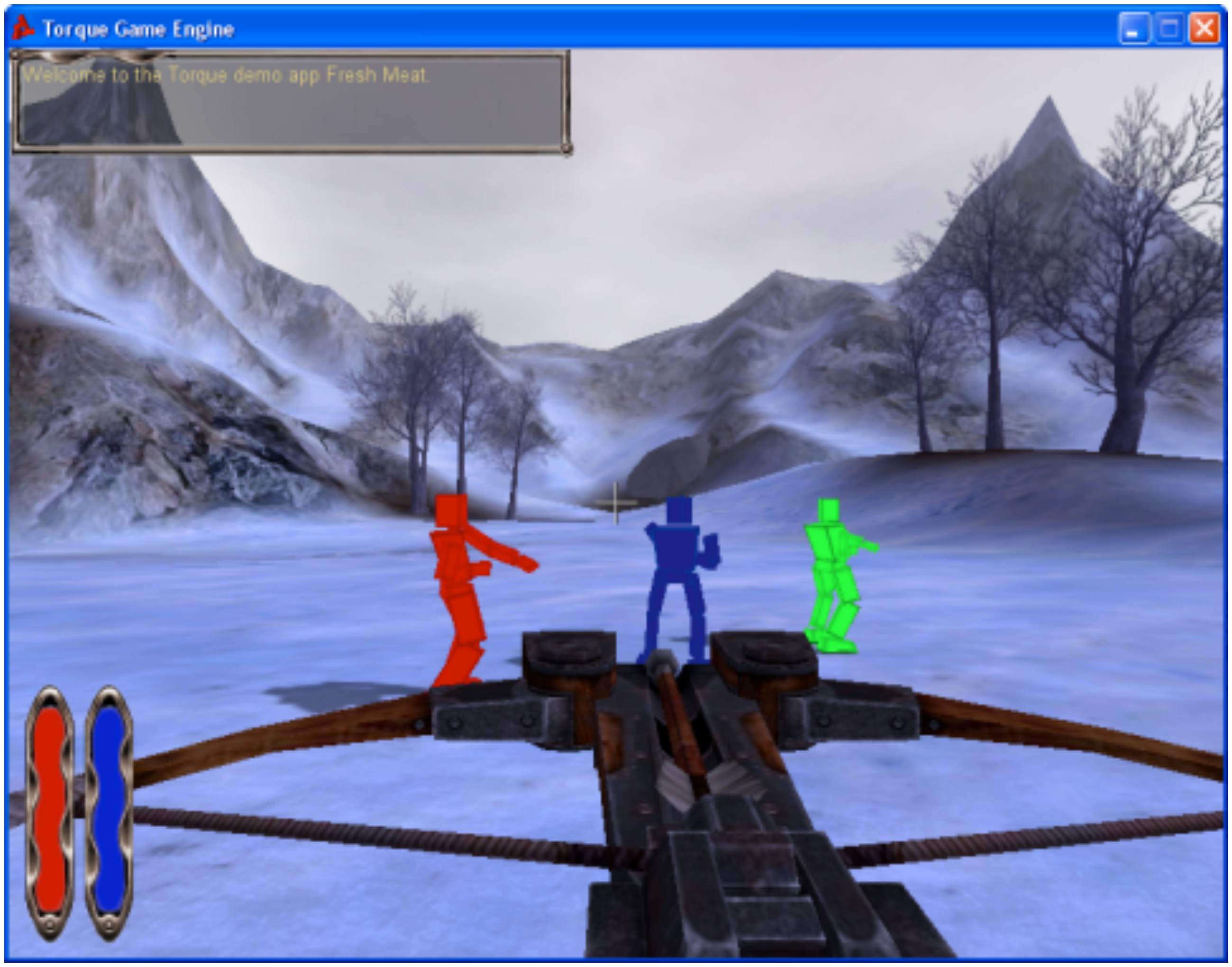

Figure 1. Three bots, red, blue, and green.

Additionally, you are supplied with a single empty function named dumpClientGroup4().

```
function dumpClientGroup4()
{
}
```

GHOST RESOLUTION

Steps:

1. Please duplicate the answer you provided for dumpClientGroup3() in exercise “ClientGroup”.
2. Please add the code necessary to count ghosts on each client and to print that information out as shown below (output goal).

Output Goal:

When you have completed this code, you can test it as follows.

1. Run the mission associated with this exercise.
2. Open the console (~).
3. Type: “dumpClientGroup4();” and press return. You should see something similar to the following. (Your IDs may be different.)

```
==>dumpClientGroup4();
Client Group (1034) contains these clients:
Client #0
 > ClientConnection: 1585 (localClientConnection)
 > ServerConnection: 1584
 >    Active Ghosts: 5
```

Questions:

1. What are the five server objects that are being ghosted to the local client?

2 Dumping Ghost Indexes

Goal: Using the provided code, determine the ID of the local client's camera and player object IDs and then convert those IDs into indexes.

Starter Code: You are provided with a single partially completed function named dumpClientGroup2().

```
function dumpLocalGhostIndexes()
{
   //%cameraID = ?????
   //%playerID = ?????

   //%cameraGhost = ?????.?????( %cameraID );
   //%playerGhost = ?????.?????( %playerID );

   echo("The server object ID of the local camera is: ", %cameraID );
   echo("The server object ID of the local player is: ", %playerID );

   echo("The ghost index for the local camera is: ", %cameraGhost );
   echo("The ghost index for the local player is: ", %playerGhost );

}
```

Steps:

1. Please uncomment the commented out lines and fill in the "?????" segments with the appropriate code.

Output Goal:

When you have completed this code, you can test it as follows.

1. Run the mission associated with this exercise.
2. Open the console (~).
3. Type: "dumpLocalGhostIndexes();" and press return. You should see something similar to the following. (Your IDs may be different.)

```
==>dumpLocalGhostIndexes();
The server object ID of the local camera is: 1636
The server object ID of the local player is: 1637
The ghost index for the local camera is: 0
The ghost index for the local player is: 1
```

Hints:

1. Don't forget. The standard starter kit scripts automatically store away key information about the player and camera in the connection object. I talked about this in the chapter.

GHOST RESOLUTION

3 Resolve Server Object IDs

Goal: Write the code necessary to allow a client to count the ghosts on its server connection and to directly convert those indexes into the server object IDs as they exist on the server.

Starter Code: You are provided with a single partially completed function named dumpServerObjectIDs(). This function is designed to count the ghosts that were sent to the current client and to convert those indexes into server object IDs.

```
function dumpServerObjectIDs()
{
   //1
   //%ghostCount = serverConnection.?????();

   echo("Resolving ghost indexes to server object IDs.");

   for( %count = 0; %count < %ghostCount; %count++ )
   {
      // 2
      //%serverObjectID = serverConnection.?????( %count );
      echo("Ghost Index #", %count, " translates to server object ID: ",
           %serverObjectID ,
           " Class Name: ", %serverObjectID.getClassName());
   }
}
```

Steps:

1. Please uncomment the marked line (matches step number) and fill in the code marked by "?????" to count the active ghosts on the named instance of GameConnection.
2. Please uncomment the marked line (matches step number) and fill in the code marked by "?????" to convert the indexes (0 .. ghost count - 1) into server object IDs.

GHOST RESOLUTION

Output Goal:

When you have completed this code, you can test it as follows.

1. Run the mission associated with this exercise.
2. Open the console (~).
3. Type: "dumpServerObjectIDs()" and press return. You should see something similar to the following. (Your IDs may be different.)

```
==>dumpServerObjectIDs();
Resolving ghost indexes to server object IDs.
Ghost Index #0 translates to server object ID: 1654 Class Name: Camera
Ghost Index #1 translates to server object ID: 1638 Class Name: Player
Ghost Index #2 translates to server object ID: 1650 Class Name: AIPlayer
Ghost Index #3 translates to server object ID: 1646 Class Name: AIPlayer
Ghost Index #4 translates to server object ID: 1642 Class Name: AIPlayer
```

4 Resolve Ghost Object IDs

Goal: Convert the code from the last part of this exercise into code that prints ghost IDs instead.

Starter Code: You are provided with a single empty completed function named dumpServerObjectIDs().

```
function dumpGhostIDs()
{
   // 1
   //%ghostCount = serverConnection.?????();

   echo("Resolving ghost indexes to ghosts for local server and client.");

   for( %count = 0; %count < %ghostCount; %count++ )
   {
      // 2
      //%ghostID = localClientConnection.?????( %count );
      echo("Ghost Index #", %count, " translates to ghost object ID: ",
           %ghostID ,
           " Class Name: ", %ghostID.getClassName());
   }
}
```

Steps:

1. Please uncomment the marked line (matches step number) and fill in the code marked by "?????" to count the active ghosts on the named instance of GameConnection.
2. Please uncomment the marked line (matches step number) and fill in the code marked by "?????" to convert the indexes (0 .. ghost count - 1) into ghost object IDs.

Output Goal:

When you have completed this code, you can test it as follows.

1. Run the mission associated with this exercise.
2. Open the console (~).
3. Type: "dumpGhostIDs();" and press return. You should see something similar to the following. (Your IDs may be different.)

```
==>dumpGhostIDs();
Resolving ghost indexes to ghosts for local server and client.
Ghost Index #0 translates to ghost object ID: 1636 Class Name: Camera
Ghost Index #1 translates to ghost object ID: 1637 Class Name: Player
Ghost Index #2 translates to ghost object ID: 1588 Class Name: AIPlayer
Ghost Index #3 translates to ghost object ID: 1587 Class Name: AIPlayer
Ghost Index #4 translates to ghost object ID: 1586 Class Name: AIPlayer
```