USING THE KIT

Exercise Files

Starter - "Kit/gpgt/server/scripts/gpgt/chapter1/exercise001.cs"
Answers - "Kit/gpgt/server/scripts/gpgt/chapter1/answers/exercise001_f.cs"

Exercise Mission

Chapter 1: "001_Introduction: Using The Kit"

Synopsis

In this exercise, we will explore the kit and come up to speed on how exercises are organized.

Prerequisites

None.

Exercises

USING THE KIT

1 Setting Up

Did you copy the contents of the accompanying disk to your hard-drive yet? No. Well then, now is the time. The easiest way to handle this is a follows.

1. Create a new directory named “gpgt” on your desktop (Windows or OSX).
2. Place the CD in your reader.
3. Open the CD and cut-copy-paste the entire contents into your new desktop directory “gpgt”.

Now, you should be ready to try out the Kit.

2 Running The Kit

To try out the kit, simply do the following:

1. Open the directory/folder "gpgt/Kit"
2. Double click on the following executable:
   - OS X Users - "Torque Demo OSX"
   - Windows Users - "torqueDemo.exe"
3. Wait for the kit to start. When it is done opening you should have an image like figure 1 below. The individual Chapter buttons are used to open a chapter-specific dialog box. When a chapter-specific dialog box is open, you will see a list of the exercise missions for that chapter. At that point, you can double-click any entry (or select an entry and click the 'Launch Mission!' button) to run the mission. (Note: I modified this dialog to allow double-clicking. The standard missions dialog requires that you use the 'Launch Mission!' button to launch a mission.)

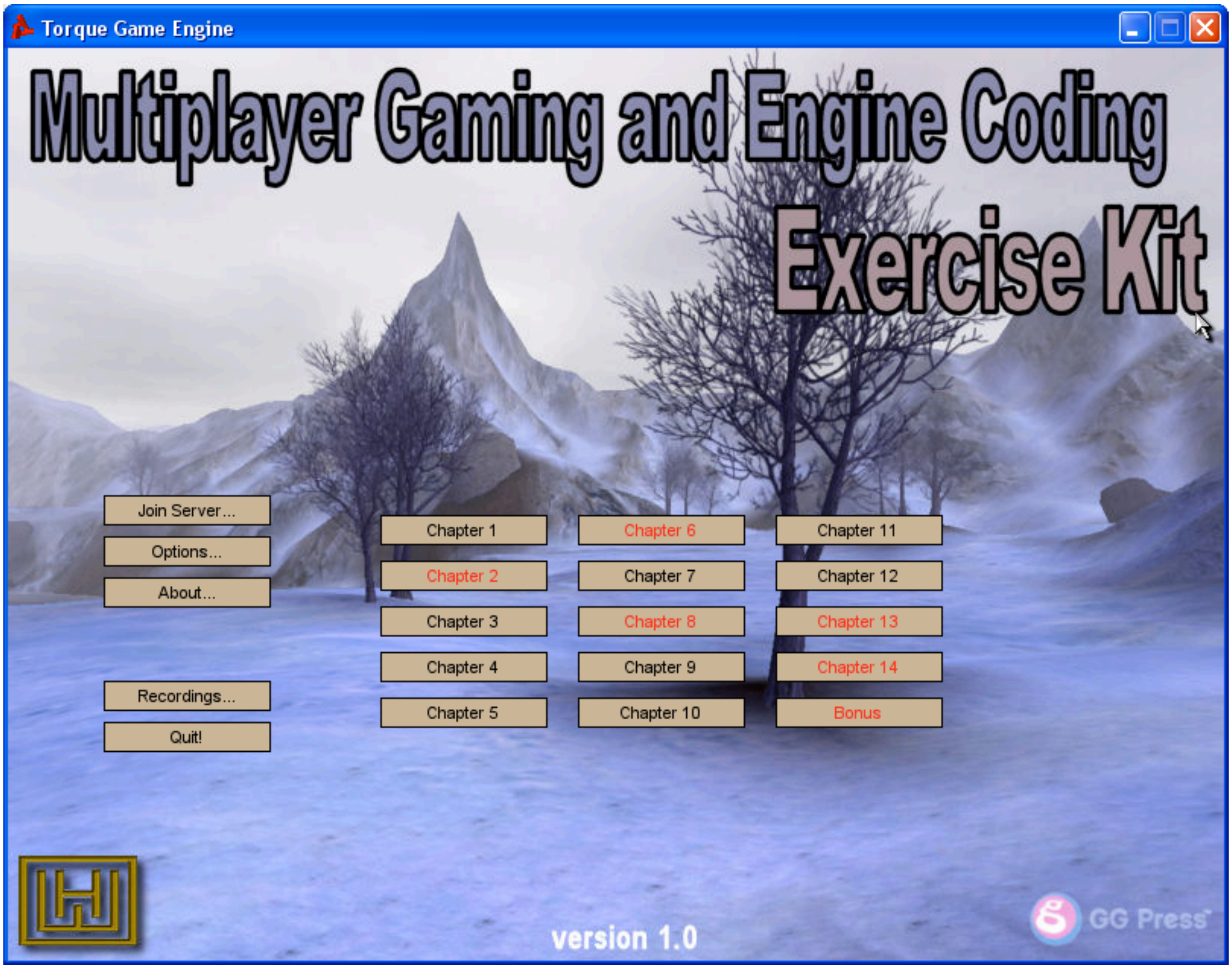
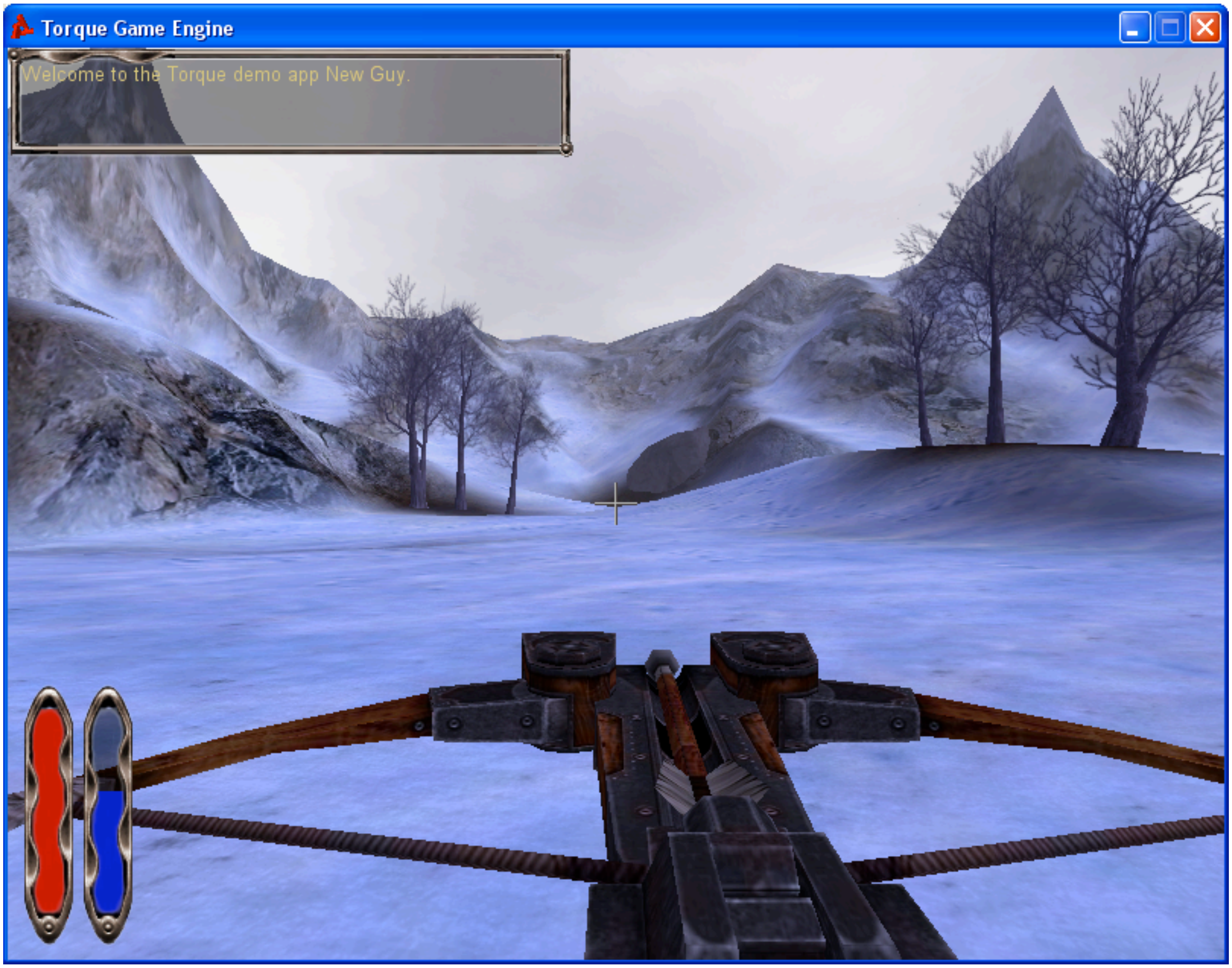


Figure 1. The Kit Is Started!

USING THE KIT

4. To see the chapter-specific mission selection in action, please click the "Chapter 1" button and then click the one mission that is listed there: "001_Introduction: Using The Kit". As soon as the mission gets done loading, you should see something like the picture in figure 2.



Welcome to the Torque demo app New Guy.

Figure 2. Your first exercise mission.

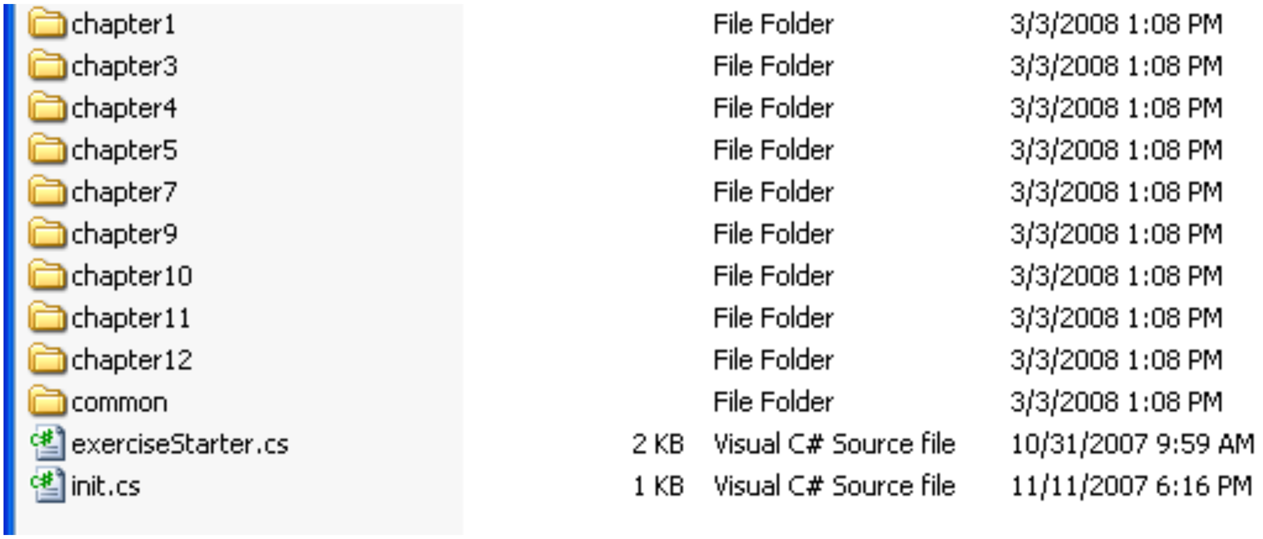
That is about it for loading a mission's exercise. Next, let's talk about locating exercise files. Then, we can run through an example exercise and see how to check out work.

3 Locating Exercise Source Files

During the course of examining other exercises, you will need to edit some of the exercise files and to add your own content to them. Individually, the exercise PDF files will list any script or C++ files you need to edit. However, in general, you will find the exercise files in the following directory.

"Kit/gpgt/server/scripts/gpgt/"

In this directory you will find directories matching the chapters. (See figure 3 below.)



chapter1		File Folder	3/3/2008 1:08 PM
chapter3		File Folder	3/3/2008 1:08 PM
chapter4		File Folder	3/3/2008 1:08 PM
chapter5		File Folder	3/3/2008 1:08 PM
chapter7		File Folder	3/3/2008 1:08 PM
chapter9		File Folder	3/3/2008 1:08 PM
chapter10		File Folder	3/3/2008 1:08 PM
chapter11		File Folder	3/3/2008 1:08 PM
chapter12		File Folder	3/3/2008 1:08 PM
common		File Folder	3/3/2008 1:08 PM
exerciseStarter.cs	2 KB	Visual C# Source file	10/31/2007 9:59 AM
init.cs	1 KB	Visual C# Source file	11/11/2007 6:16 PM

Figure 3. Chapter directories containing exercise source files.

Now, let's do a simple exercise together and then check our work.

USING THE KIT

4 Test Exercise

At this point, you should still have the Chapter 1 mission running. If not, please start it again.

OK, now that you have the first mission running, let's pretend we are doing an exercise where you need to edit a script file.

Using your favorite editor, please open the starter file for this exercise. (Starter files are always listed at the top of the exercise.)

Once you get this file open, you should see some code like this.

```
package exercisePackage_001
{
function startexercise001()
{
}
};
```

How Exercises Work (50 cent tour)

As you can see, this code doesn't seem to do anything. What you are seeing is a lesson starter function enclosed in a lesson package. As you will see, all missions have these two parts. Without going into gory details, let me explain how these two pieces of code are useful.

Although you will not have to edit them, every exercise has a mission file (*.mis). All of these mission files are almost exactly alike. They only differ in one small detail. Within every mission files is a piece of code like this.

```
new ScriptObject(exerciseStarter) {
      packageName = "001";
      exerciseName = "exercise001";
};
```

As you can see, this code creates a scriptObject named “exerciseStarter”. Additionally, this object will be created with two fields, packageName and exerciseName. The values in these fields are similar for all exercises, except that the numbers “001” will match the (current) exercise number.

Elsewhere in the code (”Kit/gpgt/server/scripts/gpgt/exerciseStarter.cs”), I have defined an onAdd() and an onRemove() callback for the “exerciseStarter” namespace. So, whenever one of these objects is created, the onAdd() callback is called. Likewise, when it is destroyed, the onRemove() callback is called.

USING THE KIT

The onAdd() callback does two important things.

1. It activates a package whose name matches this TorqueScript formula: "exercisePackage_" @ exerciseStarter.packageName

   Ex: For exercise 001, this would make the string: "exercisePackage_001"

2. It then calls the function whose name matches this TorqueScript formula: "start" @ exerciseStarter.exerciseName

   Ex: For exercise 001, this function is called: "startexercise001();"

As you might imagine, the onRemove() callback has one job, deactivate the package that was activated by onAdd().

The point of this discussion is that any code placed in a lesson package will be loaded and unloaded automatically. As well, every lesson has a starter function which can be used for various purposes.

Edit The Exercise Code

To continue this exercise, please modify the starter function to look like this.

```
function startexercise001()
{
   AIPlayer::spawn( exerciseCenter.getTransform() , BlueGuy );
}
```

(No, don't worry about what this code does. I just want you to see how lessons work.)

OK, now that you have this code typed in and saved, please quit the mission and then reload it. If you have typed in the code correctly, you will see something like figure 4 (on next page). Is there a blue guy standing in front of you? Good job.

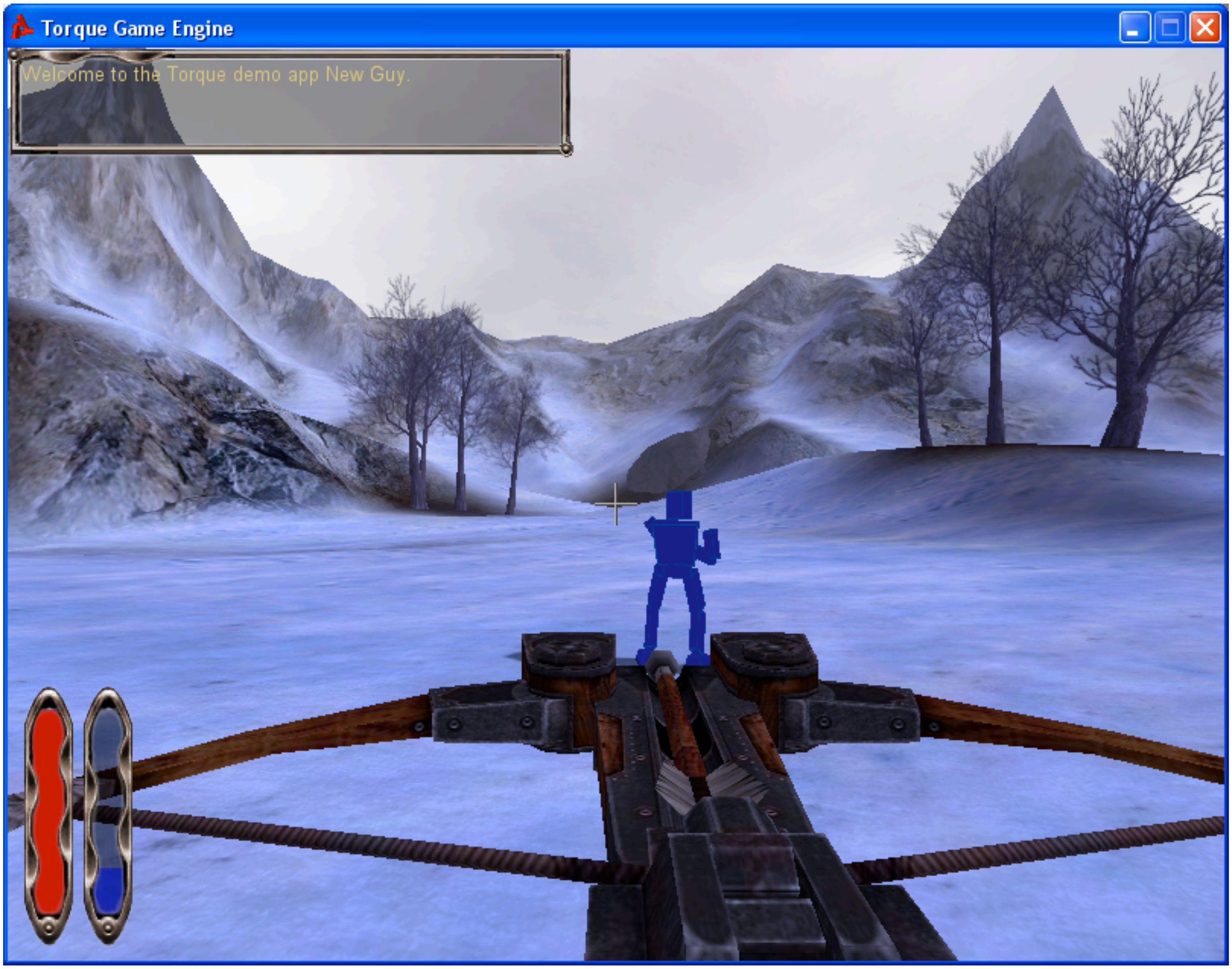


Figure 4. You typed in the code properly.

Running Answer Code

As I stated in chapter 1, all exercises come with answers, either Torque Scripts or C++ code (for chapters 10 and 11). All answer files for an exercise are listed in the "Exercise Files" section (at the top of the exercise) next to the Answers label.

To run the answer (if it is a .cs file), simply copy the file into the same directory as the Starter. In this case, we would copy the file "Kit/gpgt/server/scripts/gpgt/chapter1/answers/exercise001_f.cs" into the directory "Kit/gpgt/server/scripts/gpgt/chapter1/". Then, having done this, we could just re-run the mission. The kit is smart enough to see that there is an answer file and will run it instead of the file we just edited.

Try this out. You should see a blue guy, just like in figure 4.