

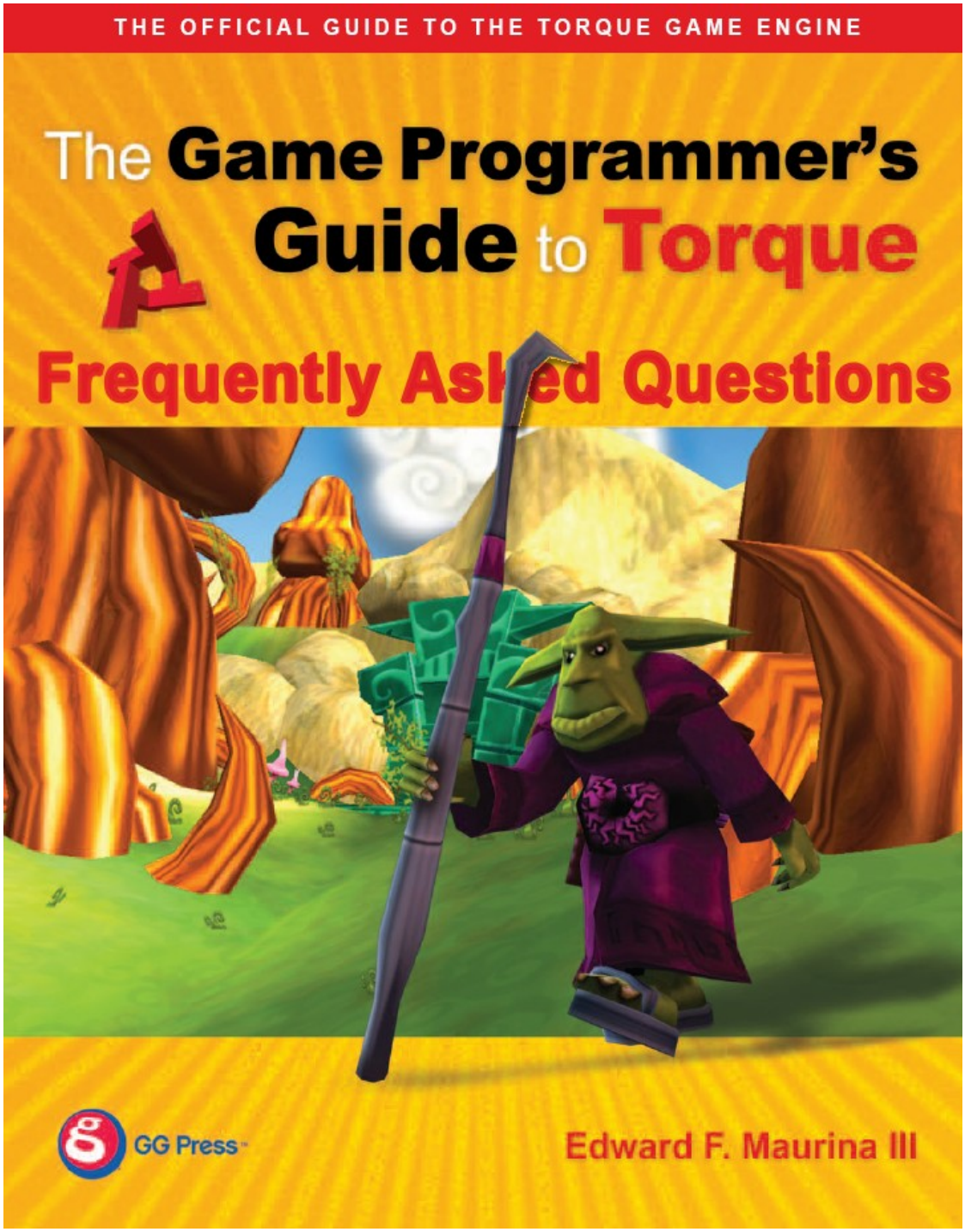

GPGT FAQ as of 21 JUL 2006

Table of Contents

Question #1 – Why won't "XYZ" run?

Question Credit: General

Problem Description:

This is a pretty easy mistake to make. Many people have tried to run the GPGT lesson kit or one of the other applications directly off of the CD.

Problem Resolution:

Copy the **entire** contents of the CD to your hard-drive. Now, run the applications there. This is necessary because the applications (GPGT Lesson Kit and Maze Runner Advanced) need to create DSO files, which they cannot do on the CD.

Question #2 – What do I do if I have a problem with my CD?

Question Credit: General

Description:

I have received a handful of direct e-mails from individuals who either did not get a disk with their books, or when the did get a disk, the disk was damaged.

Note: This has only happened for 0.3% of all books, that is about 3 in 1000 book/cd combos have a problem. Nonetheless, if you are one of those 3 in 1000 people, this can be really frustrating and disappointing.

Problem Resolution:

AK Peters has been very good about resolving this problem. Simple visit this webpage:

http://www.akpeters.com/contactus.asp

Describe your problem by giving this information:

1. **Product** – The Game Programmer's Guide to Torque (ISBN 1-56881-284-1)
2. **Problem** – Missing or Damaged Disk.
3. e-mail contact Info: your_email_address
4. mailing address: address to mail replacement CD to.

Now, submit!

Depending on where you are, you should receive a new disk in about a week or less.

If you encounter ANY problems in this process, please feel free to contact me via e-mail at:

gpgt@hallofworlds.com

Question #3 – Why Do GUI's Move When Editing?

Question Credit: Matthew Johnson

Issue Description: Matt has noticed that sometime while previewing his work GUI controls will shift. He has further noticed that this is true when he uses a horizSizing and/or vertSizing of 'relative'.

The specific problem Matt is seeing is this:

1. Open an interface using the GUI Editor (press F10).
2. Manipulate a control that uses 'relative' sizing or change the control to use this sizing method.
3. Preview your work by pressing F10.
4. Open the editor and edit some more.
5. Repeat steps 3 and 4.

At some time in this editing process, you may see that controls that use horizSizing are moving away from the position they were originally placed at. So, what is going on here?

Issue Cause: What you're seeing is the effect of floating-point to integer rounding errors. C programmers will recall that when a floating point value is stuffed into an integer, the entire mantissa is dropped. It is isn't rounded, it is just dropped. As of 1.4, TGE doesn't account for this dropping vs. rounding behavior when scaling the integer values of pixel sizes/positions.

Let me elaborate further in case the above statement is unclear. Below I'll talk in general about what is happening and the provide some solutions.

Parent Controls Resize While Editing/Previewing

When editing interfaces, we often use one resolution for our game/session. Meanwhile, we run the GUI editor at a lower resolution. For example, I often run TGE at 1024 x 768 and edit my GUIs at 800 x 600. So, when I start the GUI Editor, the interface I am editing is resized. Then, when I preview, the interface is resized again.

Pixel Size/Position Truncation Errors

When a parent control is resized it has the job of resizing and repositioning all of its children. If you are using a HorizSizing and/or VertSizing of relative, you may experience truncation errors that cause controls to translate and resize every time you rescale. To see this 'in action', let us do a hypothetical example.

Example – Bad Resizing/Repositioning

In our example, the following things are true:

- Our interface consists of a button (GuiButtonCtrl) contained in a GuiControl.
- The button is positioned at < 100 , 100 >
- The button size is < 50 , 18 >
- The button has a HorizSizing and a VertSizing of 'relative'.
- The TGE application is being run at 1024 x 768.
- The interface will be edited at 800 x 600.

In the first step of this example, we will (hypothetically) start the GUI Editor by pressing F10. This will cause the parent control and therefore our button to be resized. It will also cause the button to be repositioned. Because the factor between these two resolutions when reducing resolution is 0.78125, the engine will calculate size and position as follows:

- Calculated Size: 0.78125 * < 50 , 18 > == < 39.0625 , 14.0625 >
- Calculated Position: 0.78125 * < 100 , 100 > == < 78.125 , 78.125 >

Because stuffing floating-point values into integers causes the mantissas to be truncated, the actual values we will get are:

- Truncated Size: < 39 , 14 >
- Truncated Position: < 78 , 78 >

Fine, this probably won't be all that noticeable. However, when we turn off the editor by pressing F10, the button is going to be resized and repositioned again. This time, the multiplying factor will be 1.28:

- Truncated Size: 1.28 * < 39 , 14 > == < 50 , 18 >
- Truncated Position: 1.28 * < 78 , 78 > == < 99 , 99 >

Hey, what the heck is going on here? It looks like we recovered our previous size, but lost a pixel off the horizontal and vertical position. In fact, what has occurred is a truncating error. If we were to switch back and forth a couple more times (between editing and previewing), we would find that we lost a pixel off horizontal and vertical positions each time: < 98 , 98 >, < 97 , 97 >, ...

Can We Fix This?

So, the questions that arise are, "Is there a way to fix this problem?", and, "Are there save pixel sizes and positions?" The answer to both of these questions is a definitive, Yes!

Fix Without Engine Changes

If we decide to fix this problem without modifying the engine, we simply have to find pixel values that will not experience rounding issues when we toggle back and forth between two resolutions. That is, we need to choose a viewing resolution and and editing resolution and then consistently use these resolutions while we edit our interfaces. Then, armed with a list of X and Y pixel values that will not be affected by rounding errors, we can edit to our heart's content.

The first step is to determine which resolutions are best for viewing and editing. To do this we will calculate the scaling factors (both up and down) for the most commonly used resolutions: 640 x 480, 800 x 600, 1024 x 768, and 1280 x 1024.

This first table shows the X resolutions and the factor by which values are scaled from any one X resolution to another. The values were obtained by dividing columns by rows.

Pixel Size	640	800	1024	1280
640	1	1.25	1.6	2
800	0.8	**1**	**1.28**	0.5
1024	0.625	**0.78125**	**1**	1.25
1280	0.5	0.625	0.8	1

This second table shows the Y resolutions and the factor by which values are scaled from any one Y resolution to another. The values were obtained by dividing columns by rows.

Pixel Size	480	600	768	1024
480	1	1.25	1.6	2.133...
600	0.8	**1**	**1.28**	1.7066...
768	0.625	**0.78125**	**1**	1.33...
1024	0.46875	0.5859375	0.75	1

As should be readily apparent, the lowest and highest resolutions will not be appropriate because they produce ratios that will obviously not produce many values that do not round.

So, having only two resolutions left, we shall view our work at 1024 x 768 and edit at 800 x 600. Now, we need to find a list of numbers between (0 and 800) that will not experience rounding errors when multiplied by **1.28** and then by **0.78125**. The easiest way to do this is to simply write a script to do it for us:

```
function bfValueListNoFix( %max , %upFactor , %downFactor )
{
    for( %count = 0 ; %count <= %max; %count++)
    {
        %up   = mFloor( %count * %upFactor );
        %down = mFloor( %up * %downFactor  );
        if( %count == %down)  %valueList = %valueList SPC %count;
    }
    echo( %valueList );
}
```

If we run the above code with these arguments:

```
bfValueListNoFix( 800 , 1.28 , 0.78125 );
```

, we will get this list of values:

```
0 25 50 75 100 125 150 175 200 225 250 275 300 325 350 375 400 425 450 475 500 525 550 575 600 625
650 675 700 725 750 775 800
```

We can safely use any of these values for either a control's size and/or position and it will not migrate while we are editing and previewing using a previewing resolution of 1024 x 768 and an editing resolution of 800 x 600.

Hmm... You might be thinking that this isn't all that great. i.e., You'd rather have a more robust solution. Well, to get that solution you'll have to edit the engine.

Fix With Engine Changes

The better solution to this problem is to, fix problem at its root.

Because I cannot post source-code in a public purview (EULA restrictions), you can find a complete description of the engine fix in the TGE private forums here:

http://www.garagegames.com/mg/forums/result.thread.php?qt=42489

Question #4 – Which Executable Do I Run?

Question Credit: Roto

Issue Description: Roto posts:

> *"I am frustrated at the errors/glossing over important steps in the book."*
>
> *...*
>
> *"I have found a few paths in the books that do not exist on the CD."*
>
> *...*
>
> *"Right off the bat, Step #2. It does not tell you how to start the Maze Runner Mission. Which file do I double click to start this mission, or do I load something and then search for this mission, which by the way is close to impossible to find since many of the same name exist."*

As the author, I hear you and agree with your frustration. That said, let's see if I can't help clarify a few things.

Issue Cause:

This book was written to cover a lot of ground. It was designed to do all of the following:

1. Teach new users how to use Torque.
2. Teach any user how to write a game (Maze Runner) using Torque.
3. Act as a reference (printed material and e-appendicies).
4. Provide samples and examples of (almost) all topics discussed (GPGT Lesson Kit).

While attempting to cover all this ground, I will have your run either the GPGT lesson kit, or your Maze Runner prototype. However, sometimes it is very unclear which one should be run. What is needed is a rule of thumb for using this book and the accompanying materials.

Rules Of Thumb:

#1 WHEN TO USE Maze Runner PROTOTYPE – If you are examining numbered lesson #1 through #21, you should always be running the game prototype we set up (following the directions in section 14.4).

#2 WHEN TO USE GPGT – If you are reading through any of the printed material (besides numbered lessons), use the GPGT lesson kit.

#3 WHEN TO USE STARTER.FPS – Never. At one time, this book used the starter.fps for all of its lessons, and discussions, but I abandoned it because the terrains was "changing under my feet". Thus I created the GPGT lesson kit instead. Certainly, you can use the starter.fps on occasion and many examples will work there, but while you are following samples in the book, it is suggested that you follow rules #1 and #2 above.

Before Starting The Numbered Lessons

During the course of this book you may optionally create a game (Maze Runner). However before you can do this, you need follow all of the directions in section "14.4 Setting Up Or Workspace", starting on page 574 and ending on page 577. If you have not created a working area for your prototype, none of the numbered lessons will work.

A Suggestion (For Very New Users)

I strongly suggest, if you are new to Torque, that you read the book first and SKIP all of the numbered lessons. Then, when you feel ready, go back and do them in this order:

- Set-up Steps (Section 14.4 pages 574 – 577)
- #1 ... #21

I suggest this, because in the lessons we apply skills we just discussed in a specific chapter. To be fair, using these skills may require a bit of time, so reading the book, goofing around a bit, and then doing the lessons will probably be easier.

A Second Suggestion

If you are doing the Maze Runner Lessons, you may find it much easier to print the file:

"\Appendicies\Appendix C - Combined Maze Runner Lessons.pdf"

This file (found on the accompanying disk) contains all of the steps you need to follow IN ORDER to create the Maze Runner prototype.

Question #5 – Why Doesn't The Tilde (~) Key Work In The 'Sample Script Console'?

Question Credit: Mark Barner

Description: Mark asks, "... when you hit "~" [it] sends you into the console mode.... Is there a special way to enter the "~" in without it going into the console mode?".

Resolution: The tilde (~) key is tied to the console function 'toggleConsole()'. It is the job of this function to open and close the console window. Furthermore, when the tilde (~) key is bound to an action, both the tilde (~) key and the grave (`) key produce the same action. i.e., We lose the grave (`) key when we bind the tilde (~) key.

To resolve this issue, you can simply do the following:

1. Open the file "/gpgt/client/interfaces/SampleScriptConsole/SampleScriptConsole.cs"
2. Page down to the end of this file and you will see an implementation of 'onWake()':

```
function SampleScriptConsole::onWake( %theControl )
{
   //ssConsole.position = "2 2";
   //ssConsole.extent= "800 800";

   // cls();
}
```

As can be seen, I was experimenting and didn't clean up after myself. Tsk...tsk... Well, it doesn't matter, because our next step will wipe this all away.

3. Remove the old onWake() implementation. Yes, delete it.
4. Add these two code snippets:

```
function SampleScriptConsole::onWake( %theControl )
{
    GlobalActionMap.unbind(keyboard, "tilde");
}

function SampleScriptConsole::onSleep( %theControl )
{
    GlobalActionMap.bind(keyboard, "tilde", toggleConsole);
}
```

5. Save your changes.
6. Delete all .dso files.
7. Test the change (see **'Testing The Change'**) below.

What we have done here is to temporarily remove the binding for the tilde(~) key whenever the SampleScriptConsole tool is open. Then, when the tool gets closed, we simply re-add the binding. If you're curious, the original binding is done in the file 'default.bind.cs'.

Testing The Change: Now, the tool should work as expected. To test if the fix is working, you can do the following:

1. Add a file named "/gpgt/test.cs" with this contents:

```
echo("it works!");
```

2. Open the 'GPGT Lesson Kit' and run the 'Sample Script Console Tool'.
3. In the left window, type: echo("~/test.cs");
4. Click on the 'cls()' button.
5. Click on the 'exec()' button.
6. Viola!

Question #6 – Why Doesn't The 3D Lesson Lightning Stop?

Question Credit: Mark Barner

Description: Mark is very fast. He noted this problem before I had time to release the updated lesson. Basically, this lesson, once selected, does not stop. The lightning continues the run in all lessons thereafter.

Problem: I modified this lesson about a day before we shipped the book off for printing and did not add the correct cleanup code in the lesson. The lesson manager should delete all 'objects' associated with the lesson on exit, but I did not record the ID of the lightning object and thus the manager cannot delete it.

Resolution: We need to delete the current lightning object when the lesson is shutdown (deleted). To do this, we will make the following change to the lesson:

1. Open the file "".
2. Find the function 'LightningLesson::onRemove()'. Note that the body is empty.
3. Modify the body of the function to look like this:

```
function LightningLesson::onRemove(%this)
{
    if( isObject(LightningLesson.CurrentLightning) )
    {
        LightningLesson.CurrentLightning.delete();
    }
}
```

4. Save your changes.
5. Delete all .dso files.
6. Re-run the 'GPGT Lesson Kit' and test the 3D Lesson.

We have simply forced the 'CurrentLighting' object to be deleted when the lesson is deleted. Easy.

Question #7 – Why Can't I Manually Add A 'BoxHover' Vehicle

Question Credit: Mark Barner

Description: Mark has found that you cannot manually add a 'BoxHover' Vehicle using the Creator.

Problem: Before spelling out the error, let me help you (the reader) learn how to fish. i.e., How to debug this kind of problem. Because this kind of problem is commonly encountered when adding new 'creatable' objects, let's figure out my mistake together.

1. Run the 'GPGT Lesson Kit'
2. Click on the 'Start Mission...' button.
3. Click on the 'Launch Mission' button.
4. Wait for the mission to load.
5. Press F11 followed by F4 to start the Creator.
6. Open the console and type: cls();
7. Close the console.
8. Attempt to add a 'BoxHover' (found under Shapes->baseVehicles).
9. Open the console:

You should see a message like this:

```
gpgt/server/scripts/GPGTBase/Vehicles/HoverVehicle/HoverVehicleDataMethods.cs (16): Register object
failed for object (null) of class HoverVehicle.
```

This message is telling us that the script "HoverVehicleDataMethods.cs" has some bad code it in. Furthermore, this code tried to register an object with an ID of 0.

Huh?

OK, in English:

1. There is a bug in the script file.
2. This particular bug results from a failed call to **new.**

OK, let's investigate further.

1. Open the file "\gpgt\server\scripts\GPGTBase\Vehicles\HoverVehicle\HoverVehicleDataMethods.cs"
2. Examine the code and you should see this:

```
function HoverVehicleData::create(%HoverVehicleDB)
{
   %obj = new HoverVehicle()
   {
      dataHoverVehicleDB = %HoverVehicleDB;
   };

   %obj.mountable = true;
   return(%obj);
}
```

It may not be obvious at first, but the line in bold should have the word 'datablock' in front of the assignment operator.

This function is supposed to create a new hovervehicle object and initialize it using the datablock that is passed in. This initialization is accomplished by assigning the ID of the datablock to the object's datablock field.

As you can see I must have typoed this code, or replaced it with a global search-and-replace.

Resolution: To resolve this problem, simply change your code to look like this:

```
function HoverVehicleData::create(%HoverVehicleDB)
{
   %obj = new HoverVehicle()
   {
      datablock = %HoverVehicleDB;
   };

   %obj.mountable = true;
   return(%obj);
}
```

Question #8 – Why Does The 3D Lesson Sound Stop?

Question Credit: Rex

Description:

Rex noted that when you run the 'ShapeBase Sounds' 3D lesson from the GPGT lesson kit, that switching back and forth between sounds does not work.

Specifically, when you start the 'ShapeBase Sound' 3D Lesson, you are presented with this dialog:

However, if one follows the directions and presses 2 followed by pressing a 1, suddenly the sound stops playing.

Correction: This is on my TBD list. I have not yet determined if this is an engine bug or (more likely) a bug in my lesson. I'll update this answer when I have a resolution. **(02 MAY 2006 – TBD)**