Lab 6 – More XNA Practice

Instructions: Complete each problem. If you're struggling with a problem, feel free to ask questions on the class forum.

This lab is optional, but it gives you valuable programming experience. You should definitely complete the lab if you can.

If you run into problems, you should look at the Section 5.4 in the book before asking someone else for help.

Getting Started

Download the Lab6Materials.zip file from the Labs course page. Unzip the file somewhere on your computer. Copy the Lab 6 Help chm file from the Help folder onto your desktop.

Double click the Lab 6 Help file on the desktop to open the file.

If you get an error message in the right-hand pane instead of documentation links, it means you're currently blocking access to the documentation. To fix this, right-click on the Lab 6 Help file on the desktop, select Properties, and select the General tab. Click the Unblock button near the lower right corner of the popup.

Problem 1 – Create a project and add content

XNA Users

Start up the IDE and create a new Windows Game (4.0) project named Lab6. (If you name your project something other than Lab6, you won't be able to use the `TeddyBear` class I've provided).

Save the project in a reasonable location on the computer.

Copy the two teddy bear images into the Lab6Content project folder.

Add the two teddy bear images to the Lab6Content project by right-clicking the project, selecting Add -> Existing Item …, selecting both image files, and clicking the Add button.

Copy the TeddyBear.cs file into the Lab6 project folder.

Add the TeddyBear.cs file to the Lab6 project by right-clicking the project, selecting Add -> Existing Item …, selecting the TeddyBear.cs file, and clicking the Add button.

MonoGame Users

Start up the IDE and create a new MonoGame solution named Lab5.

Mac Users: Read the "Creating a New Mac MonoGame Solution" document on the MonoDevelop Resources course page to learn how to do this.

Linux Users: Read the "Creating a New Linux MonoGame Solution" document on the MonoDevelop Resources course page to learn how to do this.

Save the project in a reasonable location on the computer − and remember where that location is!

Read the "Adding Content to a MonoGame Project" document on the MonoDevelop Resources course page to learn how to add content to your project. I've provided compiled teddy bear xnb assets for both Mac and Linux users in the zip file; make sure you use the right ones!

Copy the TeddyBear.cs file into the Lab6 project folder.

Add the TeddyBear.cs file to the Lab6 project by right-clicking the project, selecting Add -> Add files …, selecting the TeddyBear.cs file, and clicking the Open button.

Problem 2 – Create and draw teddy bears

Add two constants to the `Game1` class just above the declaration of the `graphics` variable:

```
const int WINDOW_WIDTH = 800;
const int WINDOW_HEIGHT = 600;
```

Just below the line that says `SpriteBatch spriteBatch;` near the top of the `Game1` class, add variable declarations for two `TeddyBear` objects.

Just below the line that says `Content.RootDirectory = "Content";` near the top of the `Game1` constructor, add the following two lines of code:

```
graphics.PreferredBackBufferWidth = WINDOW_WIDTH;
graphics.PreferredBackBufferHeight = WINDOW_HEIGHT;
```

In the `Game1 LoadContent` method, replace the comment that says

```
// TODO: use this.Content to load your game content here
```

with a comment and the code to create two new teddy bear objects using the `TeddyBear` constructor that gives the teddy bears a random velocity. The teddy bears should use different sprites so we can tell them apart. Use the Lab6Help file to figure out how to call the constructor properly.

In the `Game1 Draw` method, replace the comment that says

```
// TODO: Add your drawing code here
```

with a comment and the code to have the two teddy bears draw themselves.

Problem 3 – Make teddy bears move

Add code to the `Game1 Update` method to have the two teddy bears update themselves. Don't forget to begin and end the `spriteBatch` appropriately.