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. Chapter 5 is provided as a download for lecture 5.2.

Getting Started

Double-click the index file in the Help folder and click the Lab6 link in the pane on the left; this is the documentation for the `TeddyBear` class I provided to you in the Lab6 materials.

Problem 1 – Create a project and add content

Start up the IDE and create a new MonoGame Windows Project (or appropriate MonoGame project for your OS) 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.

Use the Pipeline tool to build content for the two teddy bear images I provided and add that content to your Lab6 project.

Copy the TeddyBear.cs file into the Lab6 project folder (not the folder that contains the Lab6 SLN file, the folder that contains the Game1.cs file).

Add the TeddyBear.cs file to the Lab6 project by right-clicking the project, selecting Add > Existing Item … (Mac and Linux Users: Add > Add Files ...), selecting the TeddyBear.cs file, and clicking the Add button.

Problem 2 – Create and draw teddy bears

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

```
public const int WindowWidth = 800;
public const int WindowHeight = 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 = WindowWidth;
graphics.PreferredBackBufferHeight = WindowHeight;
```

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 Lab 6 documentation to figure out how to call the constructor properly.

Caution: The `TeddyBear` constructor for the class I gave you for the lab is different from the `TeddyBear` constructor in the videos; that's why you need to use the documentation.

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. Don't forget to begin and end the `spriteBatch` appropriately.

Problem 3 – Make teddy bears move

Add code to the `Game1` `Update` method to have the two teddy bears update themselves.