Lab 16 – Text IO

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.

Problem 1 – Create a MonoGame project, add assets, load content, and draw

Start up the IDE and create a new MonoGame Windows Project (or appropriate MonoGame project for your OS) named Lab16. Save the project in a reasonable location on the computer.

Download or create a png file, use the Pipeline tool to convert it into an xnb file, and add the xnb file as content to your project.

Declare `Texture2D` and `Rectangle` fields in the `Game1` class to hold/draw your asset.

In the `Game1` `LoadContent` method, load your asset into your `Texture2D` field and create an appropriate `Rectangle` object for your `Rectangle` field so the sprite is centered in the screen.

In the `Game1` `Draw` method, draw your asset.

Problem 2 – Add code to move the asset around the screen

Declare a constant field in the `Game1` class that specifies how many pixels your asset will move on each update (if the appropriate keys are pressed).

In the `Game1` `Update` method, add code to get the current keyboard state and put it in a variable.

Use the variable from the previous step to decide how to move your asset. Use standard WASD key controls.

Problem 3 – Add text display and centering

Declare a field called `offScreenCount` to keep track of how many times the player moves your asset partially off the screen. Initialize the field to 0.

Declare a `SpriteFont` field to hold a sprite font for your text display.

Declare and initialize a `Vector2` field for the text display position in the window.

Use the Pipeline tool to build a new Sprite Font with whatever characteristics you want. Add the xnb file for the sprite font as content to your project.

Linux Users: I've provided a compiled xnb file for an Arial 20 point sprite font in the zip file.

In the `Game1` `LoadContent` method, load your Sprite Font asset into your `SpriteFont` field.

In the `Game1` `Draw` method, draw a text message that tells the number of times the asset has gone off the screen. At this point, that number will always be 0.

In the `Game1` `Update` method, add code to increment the counter if any part of the asset is off the screen. Now your text display should work properly.

NOTE: Of course, the counter doesn't really give a count of how many times you went off screen because it gets incremented on every update while you're off the screen. You don't have to do a true counter for this, but take a moment to think about how you'd do that.

Make it so if the player presses the space bar, the sprite is moved to the center of the screen.