Computer Science 591B, Graphics

http://www.cs.umass.edu/∼elm/graphics/

Assignment 2: Spaceships in Two Dimensions

Part I.

You will be using the SDL environment again for this assignment. You will be creating a spaceship and writing a program to move it around the screen in two dimensions.

1. Define the shape of a two-dimensional spaceship. Do this by drawing it as any sort of polygon on a sheet of graph paper (if you want) and writing down the coordinates of the points as you trace the outside of the spaceship. The spaceship can be as simple as a single triangle if you want, but hopefully you can come up with something more creative that that. Thus, the space ship will just be defined by a bunch of two-dimensional coordinates. One of the coordinates should be $(0,0)$.

2. Make sure you can draw your spaceship on the screen, by calling your line-drawing function from the last assignment. Make sure you can draw it at any particular coordinates you want. Write a function

   ```
   void draw_spaceship(int xpos,int ypos,double angle,double warpfactor);
   ```

   which draws your spaceship at a given location $(xpos, ypos)$ on the screen. For the moment, just leave the angle and warpfactor out of it. Make sure you do something reasonable at the screen borders. In other words, your program shouldn't crash when the space ship is near the screen border.

3. *Rotating the spaceship.* Your spaceship will have an orientation R. R should start with a value of 0, which should make the spaceship appear in the default orientation. It will probably be easier for you if you make the spaceship face directly to the right of the screen when R=0.

   Use two keys on the keyboard (you can use the arrow keys if you want) to change the rotation of the ship. Have one key rotate the ship to the right, and another key rotate the ship to the left. At each time step, you should call draw_spaceship to update the appearance of your ship. Pass the rotation value R to the function to control the rotational appearance of the ship.

4. *Moving the spaceship.* Each time you execute the main loop of your program, your spaceship will have a position P and also a velocity V. Remember from physics that velocity is not just a speed, but a speed and a direction. This will be represented by a vector V. Each time through your loop, you should add the velocity vector to the position vector to get a new position for your spaceship, and then call draw_spaceship again to draw the spaceship in the new position.

   When the spaceship gets near the edge of the screen, I want you to move it to the opposite edge of the screen so that it goes off one edge and onto the other. This is the way many video games work.

   The fun part is changing the velocity. Choose a new key to control the thrust of the ship. When the key is hit, you should add a vector to the current velocity. The vector should be a unit vector pointing in the same direction as the current orientation R of the ship. In other words, use the formula

   $$V_{new} \leftarrow V + (\cos(R), \sin(R)),$$

   where $(\cos(R), \sin(R))$ is a vector with two components.

5. *Compressing your ship.* Imagine that your spaceship has a flexible hull, and when you turn on the thrusters, the ship compresses in the direction of R. Make a variable C that defines the degree of compression of your ship, and make C larger when thrusting is used. Let C slowly return to its default value (no compression) when thrusting is not being used. In your draw_spaceship function, C should be passed in as the warpfactor, and it should cause a scaling of your ship in the direction it is facing when the thrust is applied.

Part II. Duplicate the program you just wrote, but this time instead of using "vectorized" graphics, use a bitmap (i.e. a picture) for your spaceship. You can use any bitmap you want. I will suggest some methods for getting a bitmap in class. NOTE: you shouldn't have to change many parts of your program, just some initializations and the draw_spaceship function.

Extra credit. I will assign up to 20% extra credit for SIGNIFICANT additions to your spaceship and "game". Be creative!