

CS250 Assignment 4 Circle Animation

Date assigned: Monday, March 3, 2014
Date due: Wednesday, March 12, 2014
Points: 30

In class we saw that one easy way to animate a circle is:

```
void DarkGDK ()
{
    int x = 20, y = 20;
    dbSyncOn ();
    dbSyncRate (60);
    while (LoopGDK ())
    {
        dbClear (0, 0, 255);
        dbCircle (x, y, 10);
        ++x;
        ++y;
        dbSync ();
    }
}
```

Here are some interesting things to think about:

1. How might we have multiple circles animating on the screen?
2. How do we maintain each circle's attributes?
3. What attributes and behaviors do we want each circle to have?
4. What if we want to expand our animation to rectangles, triangles, and other objects?

Problem: You are to implement each interface found in the CS250Assignment4 folder which can be found in your class sections Public folder. The interfaces **Direction2D.h**, **Move2D.h**, and **Speed.h** are to be implemented in a project **Game2DUtilities**. The interface **Circle.h** is to be implemented in a project called **CircleAnimation**. In the CircleAnimation project, you are to write a driver called `circledriver.cpp` that places a circle of random size between 15 and 50 for a radius in the center of the screen and animates the circle off the screen in a random direction.

To complete this assignment you must

1. Create two new C++ projects **Game2DUtilities** and **CircleAnimation** in Visual Studio 2010 in your existing solution PUNetIDAssignments and write the interface as described above.
2. Type the solution (fully documented/commented) to the problem into your projects.
3. Remember to enter in your name as the author of the program. Also, each file is the have file header documentation.
4. Make sure that your program compiles and runs correctly. If you get any errors, double check that you typed everything correctly. Be aware that C++ is case-sensitive. Also, there must not be any warnings when compiling your program (other than those produced by Dark GDK) or you will lose points.
5. Once you are sure that the program works correctly, it is time to submit your solution. You do this by logging on to Turing and placing your complete solution folder with all projects working correctly in the proper CS250 Drop folder. Make sure that you copy your program folder and don't move the folder. If you move the folder, then you will not have your own copy!

Additional Notes

1. You must follow the coding standards found on the main CS250 Web page. Each method must be documented as shown below. The method is either a Constructor or a Method.

```
//*****  
// Constructor: Rational  
//  
// Description: Initializes data members to default values  
//  
// Parameters:  numerator   - the numerator of the rational number  
//              denominator - the denominator of the rational number  
//  
// Returned:   None  
//*****
```

2. Each file needs file heading documentation
3. You must use constants when possible.

4. Dark GDK projects are Win32 Projects NOT Win32 Console Applications

WARNING: This assignment is hard because you will most likely have several compiler and linker errors to fix during the development process. Start Early!!!!