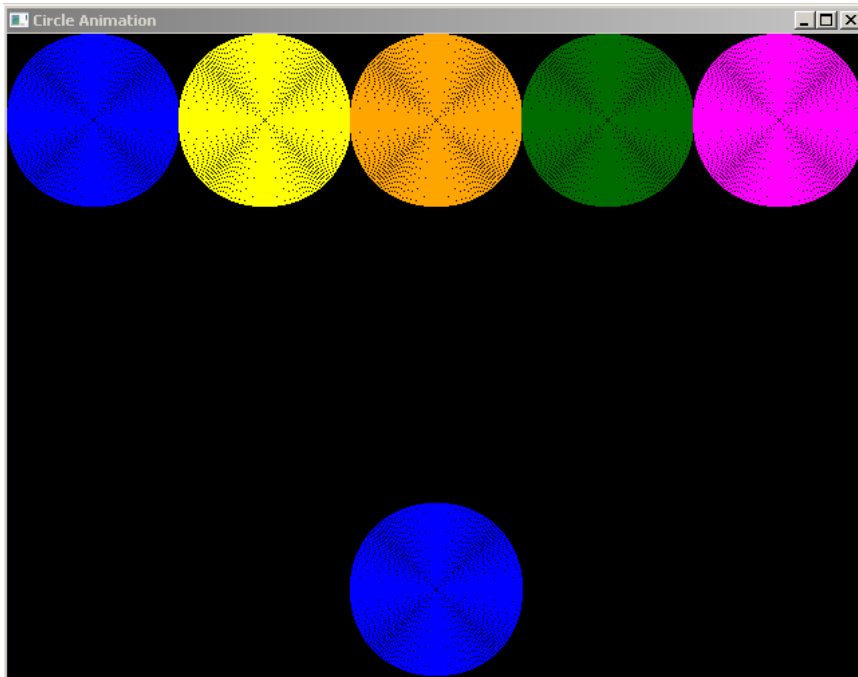


## CS250 Assignment 5 More Game Utilities

**Date assigned:** Monday, March 11, 2013  
**Date due:** Monday, March 18, 2013  
**Points:** 30

Before we can implement a simple Dark GDK game, we need a few more 2D graphic utility classes. Consider the screen below:



You are to modify your CircleAnimation driver to produce the above screen.

a) The top row of balls consists of random colors from the following random possibilities:

Color	RGB Value
White	255, 255, 255
Black	0, 0, 0
Red	255, 0, 0
Blue	0, 0, 255
Green	0, 108, 0
Purple	108, 0, 108
Yellow	255, 255, 0
Orange	255, 165, 0
Violet	238, 130, 238
Silver	192, 192, 192
Fuchsia	255, 0, 255
Aqua	0, 255, 255

b) Create a Game class whose constructor accepts the number of balls to be placed in the top row. This value can be thought of as the level of the game. The higher the level, the more balls and possibly the more difficult the game will be. The above game shows a level of 5. You will notice that a level of 5 creates five balls with the same diameter. The diameter is calculated based on the screen width and the number of balls to be placed in the top row of the screen. For example, the screen width is 640 and there are 5 balls, so the diameter of the balls is  $640/5$  or 128. The width of the screen is not always evenly divisible by the number of balls. Take this into account when displaying the balls evenly on the screen.

c) The ball on the bottom of the screen is selected from the random colors from the top row.

d) In order to successfully complete this assignment, you need to write a minimum of three more classes.

(1) The **Color** class will allow the creation of a Color object that will set and get a basic color as well as return the RGB value for an existing color. The constructor accepts a basic color with a default color of Black.

(2) Since you will have to maintain the random colors selected, you are to implement a set of Colors. The **ColorSet** class is to have a constructor and methods `isInSet`, `getRandomColorInSet`, `size`, as well as overloaded operators `+` for adding a Color to the set and `-` for subtracting a Color from the set.

(3) A Game class initializes the objects for the game and displays the above screen inside of the game loop. There can be at most 25 balls.

### ***To complete this assignment you must***

1. Create two new C++ classes (interface & implementation) in **Game2DUtilities** and one new class (interface & implementation) in **CircleAnimation** using Visual Studio 2010 in your existing solution PUNetIDAssignments.
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 to 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 (**01\_DNA**, **02\_WordSearch**, **Random**, **Rational**, **Game2DUtilities**, and **CircleAnimation**) 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!