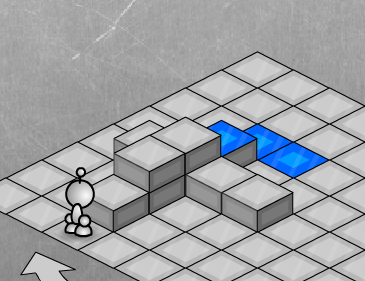
Midterm Exam: BCUSP 110

**(20%) Light Bot 2.0 Programming:**

Given that the instructions for the light bot are:

**S**-Step, **J**-Jump, **L**-Left, **R**-Right, **P**-Power

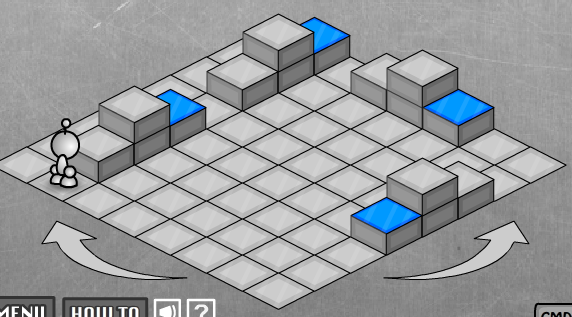
1. (10%) Solve the following puzzle, make sure your solution is optimized and you use the iteration symbol whenever possible:



1. (10%) Given the definition of **ProcessBlock** function:

// Jump Jump Jump Power

Def **ProcessBlock**() J J J P.



Solve the above puzzle by calling the given **ProcessBlock** function. This is to say, you can use any instructions, but you should call **ProcessBlock** whenever possible. Make sure your solution is optimized.

**(40%) Binary number, CD-ROM, and Color Representation**

1. (10%) Conversion to binary:
   1. (5%) Convert decimal 59 to binary
   2. (3%) How many bits do we need to represent 59?
   3. (2%) What is the largest decimal number that can be represented with 4 bits?
2. (10%) Binary number arithmetic: compute the following summation and leave your results in ***binary***
   1. (5%) 11012 + 100012
   2. (5%) 11012 + 101112
3. (15%) On a CD-ROM, we read off the following sequence (first bit read is on the left), you are told each number is represented by 6 bits.



Start:   
first bit read

* 1. (5%) How many numbers does the above sequence of bits represent?
  2. (5%) Convert the very first number recorded on this CD ROM to decimal.
  3. (5%) Draw 3 circles around the bits representing the first three numbers. Each circle should surround the bits representing each of the first three numbers. You do ***not*** need to convert these numbers to decimal, just draw the circles around the bits that represent each number.

1. (5%) For floating point RGB color representation, connect the following color description to the ***best*** RGB values:

|  |  |
| --- | --- |
| Almost black | (1.0 0.0 0.0) |
| Bright red | (0.8 0.8 0.8) |
| Light red | (1.0 0.7 0.7) |
| Not a red color | (0.2 1.0 0.8) |
| Almost white | (0.1 0.1 0.1) |

**(25%) Programming:**

1. (5%) Given the following code, what are the values of x, y, and z?

int x, y, z;

x = 5;

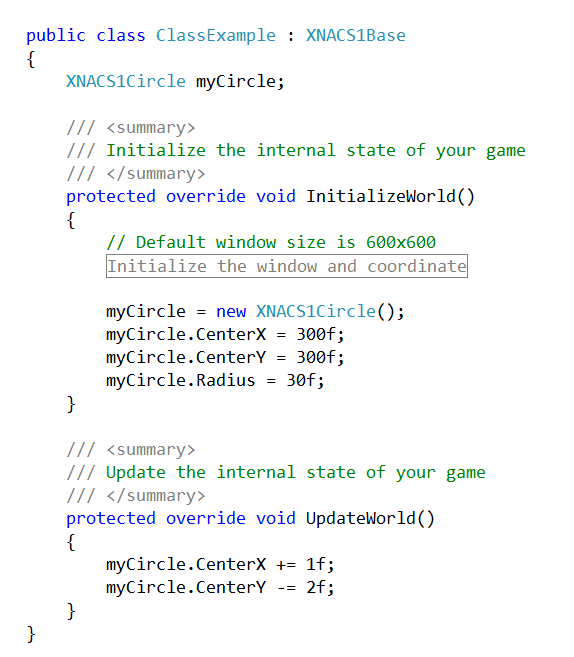
y = 10;

z = x + y;

y += z;

x += x;

1. (20%) Refer to the following code, notice that the window size is 600x600, and that we know UpdateWorld() is called 60 times a second



* 1. (5%) Describe what you will see when the program first begins?
  2. (5%) Because we change the centerX/Y value by constants at each update, we know the circle will move and eventually leave the window. On which boundary (top, bottom, left, right) will myCircle exit from the Window?
  3. (5%) How many second will it take the circle to move to the boundary that it will exit from?

* 1. (5%) We know: depending how the user manipulate the left thumbstick (or left/right arrow):

GamePad.ThumbSticks.Right.X

will give a number between -1 and +1. Insert code in the above given listing to show how you can allow the user to control the radius of myCircle while it is moving.

**(15%) Others**

1. (5%) Give an example of information that is not wise to post on your facebook page, and explain why.
2. (5%) Explain what is the difference between double clicking on *ClassEample.sln* and *ClassExample.cs*.
3. (3%) Each transistor can store/control one bit of information. Which of the following is the most reason number to use to describe number of transistors in the computer in front of you?
   1. 1 transistor
   2. hundreds of transistors
   3. hundreds of thousands of transistors
   4. millions of transistors
   5. billions of transistors
4. (2%) In the first two questions we see that the Bot supports 5 different instructions: S J L R P. Which of the following numbers (circle the correct one) best describe the number of different instructions supported by the computer you have in front of you?
   1. 2 different instructions
   2. 100 different instructions
   3. 10,000 different instructions
   4. Millions of different instructions
   5. Billions of different instructions