Graphics Programming (Graphics and Visual Computing)
Assignment list: July-Dec 2013

INSTRUCTIONS:

1. **Turbo C++ NOT ALLOWED. Program preferably using OpenGL**
2. **Codes should be well commented with header lines.**
3. **Version and the Date of last modification should be mensioned.**
4. **Group No, Roll No, Names, should be part of the header info.**
5. **Problem statement should also be part of the header**
6. **Comment the variable when defined the top of the code**
7. **Most of the code should be as functions.**
8. **"main();" should just call the functions (minimum code).**
9. **Maintain a Lab Notebook where you will write the Main gist of your code and which will be checked.**
   This list of Instruction will increase so regularly checks updates.

Assignment 1: *Submission on.*

1.1 Draw a point **point(x: float, y : float); .** **[5]**

1.2 Draw a Line **line(x1: float, y1 : float, x2: float, y2 : float, Np: int);**
Draw a line using closely placed points. If **dx** is the spacing between 2 points in **x,** then **dx=|x1-x2|/ Np.** **Np** is the number of points in the line. Now keep changing N and check the time taken to execute your code (Use Large **N**).
Plot Time take with respect to **N**. **[5]**

1.3 Draw a square/rectangle
Rectangle(x1: float, y1 : float, x2: float, y2 : float);.
Here only the coordinate of the diagonal is provided. **[5]**

1.4 Draw a Polygon with n sides : **PoligonN(&X, &Y, n).** Where **X** and **Y** are arrays. **[5]**

1.5 Given a data set of X_i and y_i points. Write a program to display these points and draw the least square (***LSq***) line for these points. If the equation of the ***LSq*** is $y = mx + b$, display the computed m and b with its respective errors $m \pm \Delta m$ and $b \pm \Delta b$ **[10]**

1.6 Search on the net for different types of random numbers (i.e. simple rand, Uniform rand (white noise), Normal rand (Gaussian), Poisson rand …..). Understand these rand functions. Write or download the random number generator and use them (only after you have understood it. You may see the fallowing site:
http://www.cs.wm.edu/~va/software/park/

Use these random number generators to generate x_i and y_i. Plot the point (x_i, y_i) for large number of (i)s and different random number generators. (Use coordinate **(0,0)** at the center of the figure. **[10]**

1.7 Make Conversion Functions between different coordinate systems:

i) ***Cartesian-to-Polar(x, y, z, r, theta, phi)***
ii) ***Polar-to-Cartesian(r, theta, phi, x, y, z)***
iii) ***Cartesian-to-Cylindrical(x, y, z, r, theta, h)***
iv) ***Cylindrical-to-Cartesian(r, theta, h, x, y, z)***
v) ***Polar-to-Cylindrical(r0,theta0,phi0, r, theta, h)***
vi) ***Cylindrical-to-Polar(r, theta, h, r0, theta0, phi0)***

1.8 Given a surface equation such as Cone, Hour-Glass, Sphere, find Cartesian surface points ***x***, ***y***, ***z***, at regular interval on the surface. Plot using **glvertex(x,y,z)**.

Assignment 2: *Submission on.*

2.1 Draw a Line **line(x1: float, y1 : float, x2: float, y2 : float);**
Use DDA algorithms, Bresenham algorithms and 2-step algorithm by Xiaolin Wu.
Repeat Problem 1.3, 1.4 and 1.5. **[5+5+5=15]**

2.2 Draw a dash and dotted line.
Dashedline(x1, y1, x2, y2, Dashsize, Dashgaps : float);
Dottedline(x1, y1, x2, y2, Dotgaps : float);

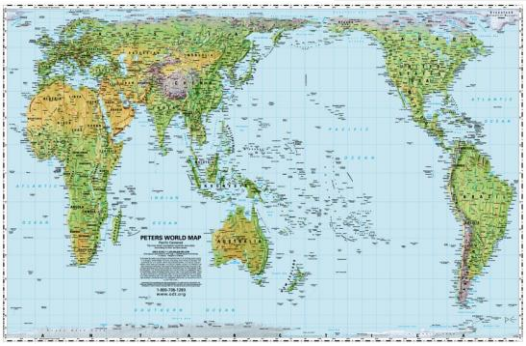
2.3 Use DDA and Bresenham algorithms. Use Bresenham algorithm to plot different functions (Circle, Parabola, Sine function, Gaussian function, ...).
[5+5=10]

2.4 Draw and Fill a Polygon **n** sides with (points/lines) : **Fill_Polygon(&x, &y, n, Color).**
[10]

2.5 Repeat Problem No 1.6 from previous assignment, but this time in colour. Define a Matrix of pixels that represents the graphics output. Compute the Number of hits ***N(i,j)*** per pixel.
If ***N(i,j)=1 to 100*** then the colour **BLUE** should vary from **0 to 1**, **Green=0**, **Red =0**.
If ***N(i,j)=101 to 200*** then the colour **GREEN** should vary from **0 to 1**, **Blue=0**, **Red =0**.
If ***N(i,j)=201 to 301*** then the colour **RED** should vary from **0 to 1**, **Blue=0**, **Green=0**.
[5]

2.6 Draw a globe **Globe(x0, y0, z0, R, dTheta, dPhi)** as in the figure bellow. **[5]**

2.7 Take an image of the world map from the web and map/paste it on the globe. Check the longitude and latitude on the image of the map and map it accordingly.



[20]

2.8 Make a sky map using the bright star catalogue which you can get from:
http://www-kpno.kpno.noao.edu/Info/Caches/Catalogs/BSC5/catalog5.html
Please Note **RA** is similar to **Longitude** but instead of being **-180 to 0 to +180** it is **0hrs to 24hrs.** Do the changes accordingly. **DEC** is similar to **Latitudes -90 to +90**. This Bright star catalog contains **9110 stars**. The brightness of the star is given by ***MAG = -2.5 Log(I / I_0)***. **[10]**

Assignment 3: *Submission on: .*

3.1 Make your own **Colour Alias Library Function** using combination the fundamental colours **RGB**. A look up table may be constructed which could be read by your Colour function to determine the aliasing. **[10]**

3.2 Write an Antialiasing function using weighted area filtering.
Antialiasing (x1: float, y1: float, x2: float, y2: float, d0: float);

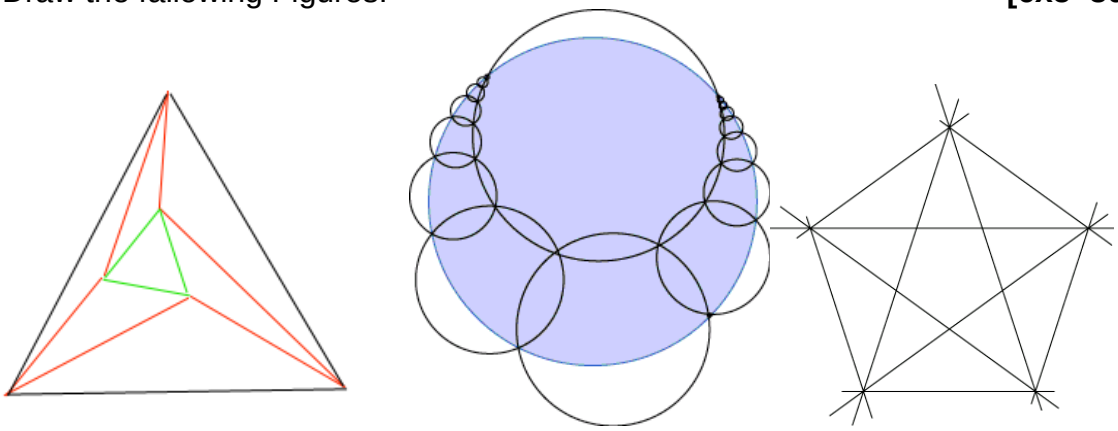
3.3 Where **d0** is the diameter of the line (use 2 pixels, so overlap occurs). You need to calculate distance **d** from pixel centre to line. **[5]**

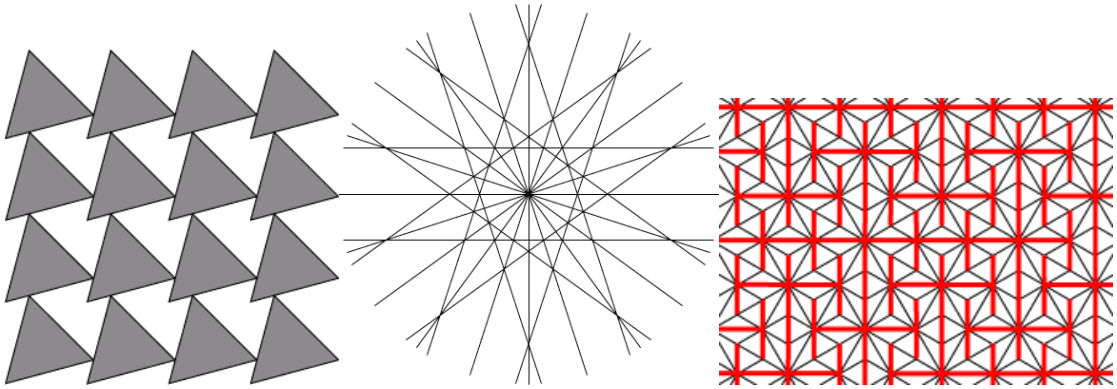
3.4 Write an Antialiasing function using Box, Cone, Gaussian $sinc^2$ filtering.
AntialiasingGauss (x1: float, y1: float, x2: float, y2: float, Sigma: float, d0: float);

3.5 Where **d0** is the diameter of the line and **Sigma** comes fro the Gaussian Function. You need to calculate distance **d** from pixel centre to line. **[10]**

3.6 Repeat Q-7 using Convolution Method. **[10]**

3.7 Draw the fallowing Figures: **[6x5=30]**



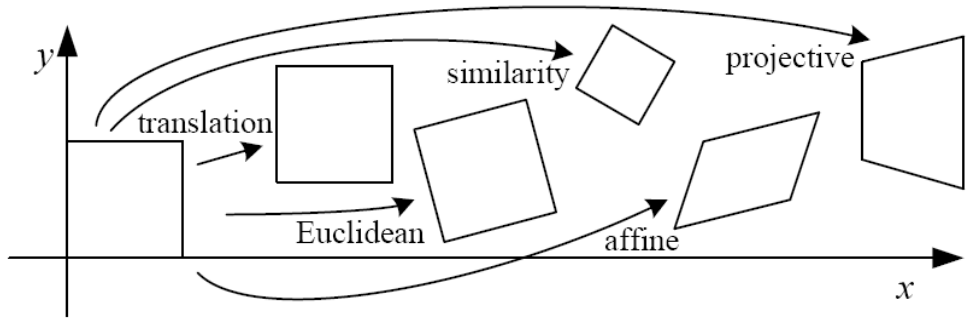


Assignment 4: *TRANSFORMATIONS.*

4.1. Implement generalized 2D transformations with translation, rotation, reflection (about standard and arbitrary axes), shearing and scaling. Use simple objects such as rectangles, triangles etc, as input to test your program. Origin should be considered at the centre of the screen. Write Transformation Functions:

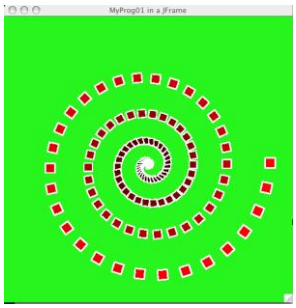
Transform_Identity(&X, &Y, n,);
Transform_Translate(&X, &Y, n,.....);
Transform_Rotate(&X, &Y, n,.....);
Transform_IsotropicScalling(&X, &Y, n,…..);
Transform_AnIsotropicScalling (&X, &Y, n,……);
Transform_Reflaction(&X, &Y, n,…..);
Transform_Shear(&X, &Y, n,…..);
Transform_Projective(&X, &Y, n,…..);

Use **PoligonN(&X, &Y, n)** and **Fill_Polygon(&x, &y, n, Color)** which you have already developed in Assignment 2 to represent your output.



[8x1.5=12]

4.2. Draw the falloing figure using the above Transformation Functions:



Equation of a Spiral:

$$R = At \cdot e^{j(\omega t+\theta_0)}$$

$$x = At \cdot \cos(\omega t + \theta_0)$$

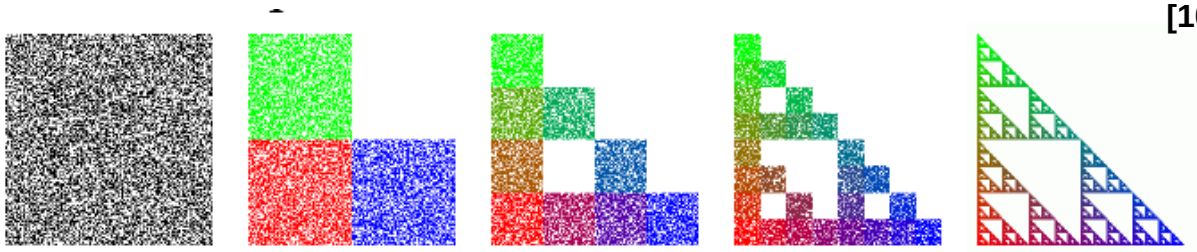
$$y = At \cdot \sin(\omega t + \theta_0)$$

[8]

4.3 Build the fallowing figure by scaling the figure by **scale(0.5,0.5)**, make 3 copies with the translations **Translate(0, 0.5)**, **Translate(0.5, 0)** and **Translate(0.5, 0.5)**. Initially the figure consists of 3 squares of RGB colours: Red: (1,0,0), Green: (0,1,0) and Blue:

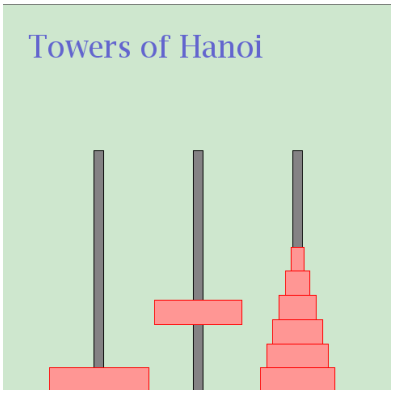
(0,0,1). Add the colours of the superposing squares. Since the maximum colour value is 1, normalise the colours with the maximum colour value after addition. [For example if the colour after addition is (1, 0, 2), it should be represented as (0.5, 0, 1)]

[10]



Assignment 5: *ANIMATION Submission on: .*

5.1 Design the game of "Towers of Hanoi". The **Tower of Hanoi** or **Towers of Hanoi** (also known as **The Towers of Brahma**) is a mathematical game or puzzle. It consists of three rods, and a number of disks of different sizes which can slide onto any rod. The puzzle starts with the disks neatly stacked in order of size on one rod, the smallest at the top, thus making a conical shape.



The objective of the puzzle is to move the entire stack to another rod, obeying the following rules:

- Only one disk may be moved at a time.
- Each move consists of taking the upper disk from one of the rods and sliding it onto another rod, on top of the other disks that may already be present on that rod.
- No disk may be placed on top of a smaller disk.

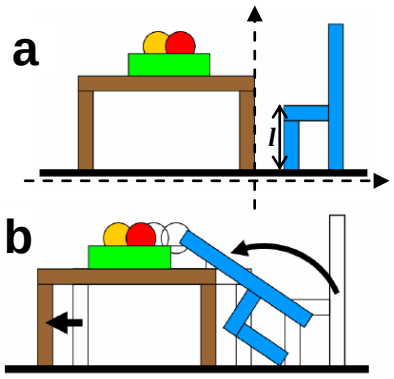
Please check the fallowing website:

http://www.dynamicdrive.com/dynamicindex12/towerhanoi.htm

You will have to design a similar game as shown in the above website but show the movement of the disks from one tower to another. Provide the choice of number of disk.

[10]

5.2 **a)** Draw the figure-(a) with primitives like rectangles (polygons) and circles. Use different transformations to place the primitives to make the figure-(a). Define your own distances and sizes of the primitives. The coordinate origin is provided (along the right edge of the table). You can assume a "**Fill(object, color)**" function to color the primitives. The chair is blue; the table is brown, the platter on the table is green and the circles in the platter are red and ocher.

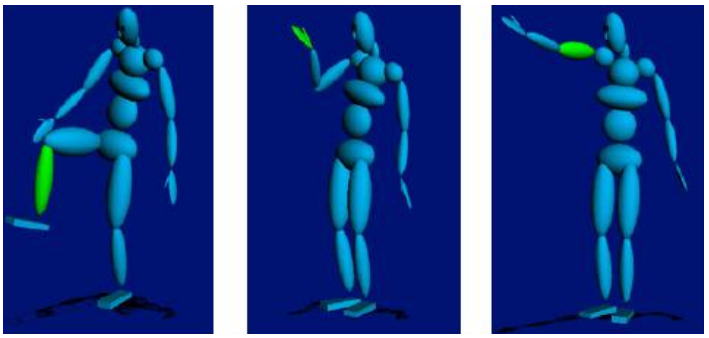


b) Animated the figure (a) as shown in figure (b). you might have to change the origin of the coordinates. Maintain the sequence of animation. First the chair hits the table at angle α (which is to be calculated in function

of distances that you have defined in (a)). Then the table moves with the platter and the spheres. The chair can topple with constant angular velocity ω until it touches the table. After which the chair has to decelerate and stop with the table at an inclined angle of **30°** from the floor. The initial velocity of the table will be $v_0 = d/dt\left[l\cos(\alpha)\right]$, where $\frac{d\alpha}{dt} = \omega.$ The Table will decelerate and stop.

[10]

5.3 Group Problems *(To be done as group where each member developes a module)*



As shown in the figure, build a human being with scaled spheres (oblates). Use hierarchical model to stitch the different limbs of the Human being. Add another sphere to represent a football. Animate your human being so that it should kick the football. You are free to colour your graphics to make it attractive. You are also free to design other movements of your human being. **[30]**

5.4 In continuation with the previous problem, assume an Ascii table of 8 column containing angles of rotation with time for its eight joints (Left and Right Shoulders, Elbows, Hips and Knees), This Table should be read and the limbs should be moved accordingly. (The table will be provided later. For the time being you could create the table in Excel and put some 8 different sin and cos function to represent the positive and negative movement around the static point.) **[20]**

Assignment 6: *ANIMATION Submission on: February 22, 2012.*

6.1 Rotate the Globe that you have designed in Assignment 2. The rotation is of ωt around its pole which tiled by (θ_p, φ_p). $\omega, \theta_p, \varphi_p$ are the input parameters along with the spheres parameters $(R, \Delta\theta_l, \Delta\varphi_L)$. **[10]**

6.2 **a)** Animate a jet of particles emanating from a specific Longitude and Latitudes (θ_l, φ_L). The jet is moving out radially with a constant velocity v_R .

b) The emanating jet which was moving out radially with a constant velocity v_R experiences a force and is accelerated by (a_x, a_y, a_z). **[20]**

6.3 The table bellow provides the coordinates of lines starting from (**Xa, Ya, Za**) and ending at (**Xb, Yb, Zb**). The Table contains coordinates for 2+32=34 lines (Where **k** varies from 1 to 16). The functional dependencies of the vertices are provided in the table. The display region is

X_{min} = -75, Y_{min}=0, Z_{min}=+10; X_{max}=+75, Y_{max}=+100, Z_{max}=+100.

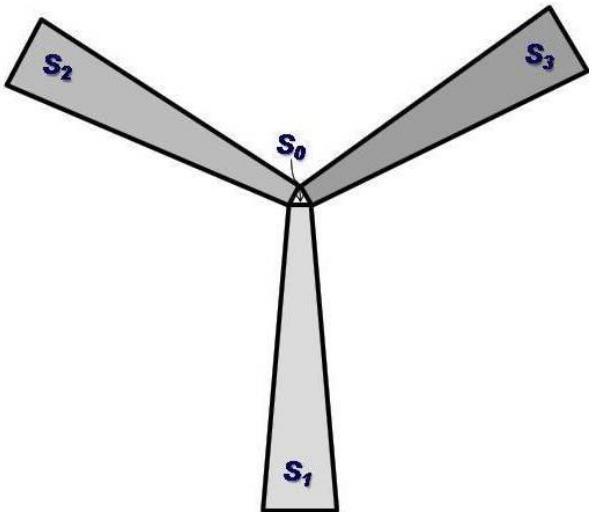
The Camera or the Observer is at ***(0,0,0)*** at a distance ***d= Z_{min}=+10*** form the display screen.

Line	Xa	Ya	Za	Xb	Yb	Zb
A	-80	0	0	-80	0	200
B	80	0	0	80	0	200
1	-80	0	0	-80	50	0
2	80	0	0	80	50	0
. . . .	. . . .	. . .	. . .	. . .	. . .	. . .
2k-1	**X**a(2k-1) = **X**a(1)	**Y**a(2k-1) = **Y**a(1)	**Z**a(2k-1) = **Z**a(1)+10(k-1)	**X**b(2k-1) = **X**b(1)	**Y**b(2k-1) = **Y**b(1)	**Z**b(2k-1) = **Z**b(1)+10(k-1)
2k	**X**a(2k) = **X**a(2)	**Y**a(2k) = **Y**a(2)	**Z**a(2k) = **Z**a(2)+10(k-1)	**X**b(2k) = **X**b(2)	**Y**b(2k) = **Y**b(2)	**Z**b(2k) = **Z**b(2)+10(k-1)
.	.	.	.	.	.	.
k=16	(32 Lines)					

Create a **3D** experience of the above scene. Two projected perspective images as seen by both of our eyes need to be created. Assume that the above coordinates are in **meters** and our both eyes are separated by **5cm** and at a human height of **5.5feet**. We center our eyes at the previous Camera/observer location ***(0,0,0)*** (Note the eyes are aligned in x-direction). Describe the method of creating both the images using short notations. [i) How do I create the 2 viewer orthonormal coordinates; ii) How to transform the vertex coordinates; iii)Create perspective coordinates; iv) Select / Reject / Clip. ……]. (*1feet = 30.48cm*).

You may create the image one beside the other and bring a cardboard to isolate one image from one eye. Or you may create 1 image in Blue and the other in red super imposed on the same image and sees it through special spectacle where one eye sees through RED cellophane paper while the other sees through a blue screen.

You may beautify the scene as per you imagination and artistic liberty. **[20]**

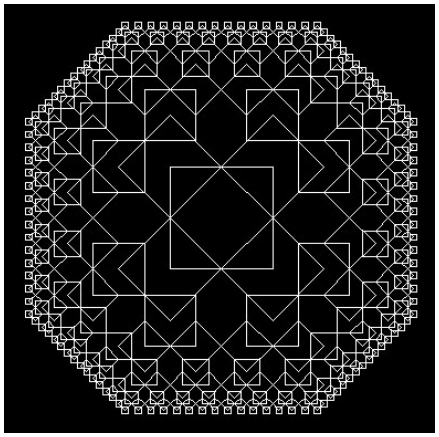
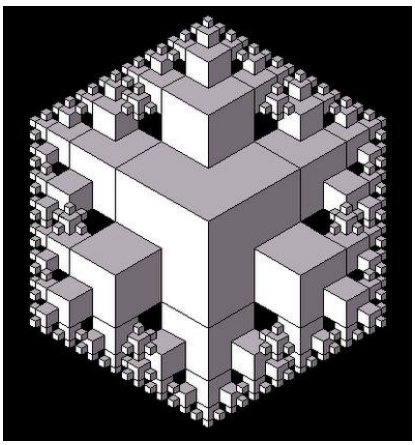


6.4 A Helicopter Propellers needs to be designed using 3 equal isosceles trapezoids as shown in the figure. The dimensions of each trapezoid are height=100 and the bases are 10 and 30. A vector $\vec{r}$ formed from the vertex ***(x_1=20, y_1=30, z_1=25)*** and ***(x_2=170, y_2=180, z_2=175)*** is the axis of rotation of the propeller. The center of the

propeller is placed at ***(x_2, y_2, z_2).*** The propeller is rotated by and angle ωt around $\vec{r}$. You may beautify the scene as per you imagination and artistic liberty.

[15]

6.5 *Make the following FRACTALS*

Mandel Broth Set $z_{n+1} = z_n^2 + c$. Where "c" is a complex number**.** **[5x5=25]**