

## Lab02 – Mathematics for Computer Graphics

### Task 1

- 1) A straight line segment starts at  $p_1(5, 11)$  and ends at  $p_2(13, 8)$ .
  - a) Draw it in the Cartesian co-ordinate system on a sheet of paper.
  - b) Calculate its length.
  - c) Calculate its gradient.
- 2) For two vectors  $\mathbf{p} = 10\mathbf{i} + 7\mathbf{j}$  and  $\mathbf{q} = 2\mathbf{i} - 5\mathbf{j}$ 
  - a) Calculate the magnitude of  $\mathbf{p}$  and  $\mathbf{q}$  respectively.
  - b) Calculate  $\mathbf{p} - \mathbf{q}$ .
  - c) Calculate the dot product of  $\mathbf{p}$  and  $\mathbf{q}$ ,  $\mathbf{p} \bullet \mathbf{q}$ .
  - d) Find the angle between  $\mathbf{p}$  and  $\mathbf{q}$ .
- 3) For matrices, explain the following
  - a) Square matrices
  - b) Identity matrices
  - c) Inverse matrices
- 4) For matrix multiplication,
  - a) Explain the rule for matrix multiplication.
  - b) If the rule is satisfied for matrices for both  $\mathbf{AB}$  and  $\mathbf{BA}$ , would  $\mathbf{AB} = \mathbf{BA}$ ?
  - c) For two matrices,  $\mathbf{A} = [1 \ 2 \ 3 \ 4]$  and  $\mathbf{B} = \begin{bmatrix} 5 \\ 6 \\ 7 \\ 8 \end{bmatrix}$ , would the rule be satisfied for  $\mathbf{AB}$  and  $\mathbf{BA}$ ? If yes, work out the resultant matrices; if not, explain the reason.

### Task 2

- 1) Try to read and understand the following code, and then run it in MS VS.

```
// File ID: Lab02.cpp
// Title: Interactive program for calculating area of a circle

#include <iostream>
#include <math.h>

using namespace std;

int main() // Define function of int type
{
    float r, area; // Declare variables for radius and area
    cout << "Please enter radius of circle: "; // Prompt user to enter value of radius
    cin >> r; // Read in value of radius
    area = 3.14 * r * r; // Calculate area
    cout << "Area of circle is: " << area << endl; // Print on screen a message and area of circle
    return 0; // Return an integer to indicate successful
               // completion of function call
}
```

- 2) Write a simple program in VC++, which
  - a) takes in x and y co-ordinate values for two points  $P_1$  and  $P_2$  in turn from the user,
  - b) calculates the length of Line  $P_1P_2$ ,
  - c) calculates the gradient of Line  $P_1P_2$ , and
  - d) prints out the results on screen.