Advanced Games Engine Creation

3D Maths

In this practical you will

- Practice doing vector operations with pen and paper
- Practice doing matrix multiplications with pen and paper
- Implement and test a Vector3D struct in C++

Hint – use a site such as http://www.mathportal.org/calculators/matrices-calculators/ to check your answers as you go

1. Vector operations
   Use pen and paper to perform the following calculations:

a) Find the vector AB between the point A(4, -2, 8) and B(-1, 4, 6)

b) Add the vectors (-7, 3, 2) and (-2, -1, 4)

c) Multiply the vector (2, -5, 3) by -2

d) Find the magnitude of the vector (2, 1, 3)

e) Normalise the vector (3, 4, 0)

2. Multiplying vectors by matrices - Use pen and paper to perform the following calculations.

a) $\begin{bmatrix} 3 & 1 & 6 \\ 0 & 2 & 1 \\ 1 & 0 & 5 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 3 \end{bmatrix} = ?$

b) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 0 \\ 1 \end{bmatrix} = ?$

c) $\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 0 \\ 1 \end{bmatrix} = ?$

d) $\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 0 \\ 1 \end{bmatrix} = ?$

e) $\begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 0 \\ 1 \end{bmatrix} = ?$

f) $\begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 0 \\ 1 \end{bmatrix} = ?$

What is the effect of each matrix on the vector in questions b)-f) (ignore the final 1, it will be explained next week)? Try drawing a sketch of the vector before and after the transformation.

3. Multiplying matrices:
Use pen and paper to perform the following calculations:

a) $\begin{bmatrix} 2 & -1 & 6 \\ 3 & 5 & 2 \\ 6 & 4 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & 3 \\ 4 & 0 & 2 \\ 2 & 1 & -1 \end{bmatrix} = ?$

b) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 4 & 2 & 2 \\ 3 & 2 & -2 & 8 \\ 0 & 9 & 1 & 1 \\ 1 & -3 & 6 & 2 \end{bmatrix} = ?$

c) $\begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = ?$

d) $\begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 0 \\ 1 \end{bmatrix} = ?$

e) $\begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = ?$

Next week we will discuss the how 3D transformations can be expressed by matrix operations such as the ones in questions 2 and 3.

4. Implement a 3D Vector struct

Create a new C++ project in Visual Studio – use the Empty Project template.

Outside of Visual Studio (in Windows Explorer), copy the file Commons.h from last week's OpenGL_SDL Base Project to the project file of your new project.

Within Visual Studio, add Commons.h to your new project by right-clicking on the Header Files folder in Solution Explorer, choosing Add -> Existing Item, and selecting Commons.h.

Add a new item (a C++ tiles) called CommonsTest.cpp to the Source Files folder of your project. Copy into it the following code:

```
#include <iostream>
#include "Commons.h"
using namespace::std;

void display(const Vector2D v) {
	cout << "[" << v.x << "," << v.y <<  "]" << endl;
}

int main(void) {
	char x;
	Vector2D v1(2.0f, 3.0f);
	display(v1);
	cout << "Enter any char to exit";
	cin >> x;
}
```

Compile and run your program. You should see the output:

```
[2,3]
Enter any char to exit
```

Now create a Vector3D structure in Commons.h analogous to the existing Vector2D structure. Test your new structure by adding code to the main method to create an instance of the structure and display its values.

As discussed in the lecture, add overloads to the +, -, and * (where * is scalar multiplication) operators. Add one method at a time and test using the main method. Be sure that each operator behaves as expected.

Challenge question – also overload and test the +=, -= and *= operators.

Add two more methods to the Vector3D structure as follows:

- A method Magnitude(), which calculates and return the vector's magnitude as a float. You will need to use the cmath libraray square root function – see http://www.cplusplus.com/reference/cmath/ for documentation
- A method Normalise(), which calculates and return the corresponding unit vector. This method should make use of the Magnitude() method.

Test each method by writing appropriate code in the main method, and displaying the output.

Once you have thoroughly tested the methods, you can add your new Vector3D struct to your SDL base project, by pasting the Vector3D structure code into Commons.h after the Vector2D structure code.