Maths for Games

|  |  |
| --- | --- |
| **Assessment Task Number:** Part 1 – Custom Maths Data Types | |
| **Unit Code(s):** | **Unit Title(s):** |
| PGDMTH6005 | Apply fundamental games programming mathematics skills |
| CUADIG511 | Coordinate testing of interactive media products |
| **Instructions to Learners:** | |

Write a collection of custom data types in C#. These will implement the required data types (either as structures or classes) and functions listed below.

Your math data types (i.e., structs or classes) must include the following types. The maths operations should be in column-major order, using a right-handed coordinate system, with the following names of types and functions to ensure that your code works correctly with the Unit Test Application used in Assessment Task 2:

* Vector3 (3-dimensional vector using 3 floats)
* Vector4 (3-dimensional homogeneous vector using 4 floats)
* Matrix3 (2-dimensional 3x3 rotation matrix using 9 float)
* Matrix4 (3-dimensional 4x4 transform matrix using 16 float)
* Colour (RGBA values stored as a 4 byte integer)

Your types must overload the following mathematical operators and include the additionally mentioned member functions, where **V** represents a Vector, **M** represents a Matrix, **n** represents an index and **f** represents a float:

* V = V + V (point translated by a vector)
* V = V – V (point translated by a vector)
* V = V x f (vector scale)
* V = f x V (vector scale)
* V = M x V (vector transformation)
* M = M x M (matrix concatenation)
* f = V.Dot( V )
* V = V.Cross( V )
* f = V.Magnitude()
* V.Normalize()
* M.setRotateX( f )
* M.setRotateY( f ) and
* M.setRotateZ( f )

If applicable, your types should all have default constructors, and constructors that allow each float element to be set individually, for example:

* Vector3 v3; // default
* Vector4 v4( 0, 0, 0, 1 );
* Matrix3 m3 = Matrix3( 1, 0, 0, 0, 1, 0, 0, 0, 1 );

(Default constructors won't be possible if you are implementing your types as structs, but will be expected if you are writing classes).

Your Color type will have get and set functions/accessors/methods for each colour component that employ bit shifting and bitwise operators to query or modify the individual colour components of the 4 byte integer. For example:

* byte Red { get{…} {set{…} }

or, alternatively

* byte GetRed();

void SetRed(byte value);

On C*anvas* you will find a Unit Test Project. This project contains a series of pre-programmed tests that will validate the implementation of your custom math data types.

To ensure that your code functions correctly, you will use a Unit Test Project to test the accuracy of your mathematical methods in the code you have written for your Maths Data Types.

To demonstrate competency in this subject your custom data types must pass all unit tests provided.

|  |  |  |
| --- | --- | --- |
| **Items to Submit** | | **Evidence Criteria and Assessment Decision-Making Rules** |
| 1. | Vector types for 3D vectors | Write Vector data types for 3D vectors, including homogeneous 3D vectors.  Types implement methods for, in all instances, translation, scale, magnitude, normalisation, cross product and dot product.  All written code conforms to the provided brief (in *Instructions to Students)*. |
| 2. | Matrix types for 3D matrices | Write Matrix data types for 3D matrices, including homogeneous 4D matrices.  Types implement methods for multiplication, vectors transformation, and transpose.  Types implement methods for setting up as rotation matrices.  All written code conforms to the provided brief (in *Instructions to Students)*. |
| 3. | Color type | Write a Color data type.  As part of the Color type, functions for manipulating a bitfield implemented using bit shift operations  All written code conforms to the provided brief (in *Instructions to Students)*. |
| 4. | Unit Testing | All custom maths data types pas all unit tests in the provided Unit Testing Project. |
| **Submission Requirements:** | | |
| You will need to submit the following:   * All source code * Your complete Visual Studio project   Be sure to remove any temporary build folders (i.e., the Debug and Release folders). Only project files, source code files, and any resource files used should be included in your submission.  Package all files in a single compressed archive file (.zip, .7z, or .rar) | | |