

3D Graphics And Animation (CM3045)

Course Notes

Felipe Balbi

April 22, 2021

Contents

Week 1	3
1.104 Vectors - 1	3
1.105 Vectors - 2	3
1.106 Vectors - 3	4
Week 2	5
1.304 Objects in Unity	5
1.305 Reading: Computer graphics	5
Week 3	6
2.101 Transforms	6
2.102 Compound Objects	6
Week 4	7
2.301 Matrices	7
2.302 Matrix Multiplication	8
2.303 Introduction to Transformation Matrix	8
2.304 Scaling	9
2.305 Rotation	9
2.306 Transformation Matrix - translation and summary	9
2.308 Transforms in Unity	10
2.309 Reading: Computer graphics	10

Week 1

Key Concepts

- Create simple scenes from basic primitives or imported assets
- Evaluate and select 3D graphics engines for use in a project.
- Explain the basic idea of 3D graphics

1.104 Vectors - 1

A 3D vector \vec{a} has three coordinates, (x_a, y_a, z_a) . The magnitude of a vector is given by $|\vec{a}| = \sqrt{x_a^2 + y_a^2 + z_a^2}$.

A unit vector for \vec{a} can be computed by dividing each coordinate of \vec{a} by its magnitude (i.e. normalising), in other words $\hat{a} = \frac{\vec{a}}{|\vec{a}|}$. The unit vector \hat{a} also serves as the direction vector for vector \vec{a} .

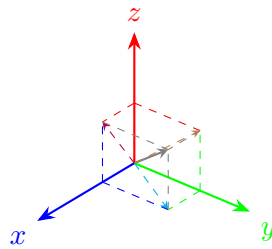


Figure 1: 3d Vector

1.105 Vectors - 2

We can scale a vector by multiplying each coordinate by a scalar, a single number, for example $2 \cdot (2, 1) = (2 \cdot 2, 1 \cdot 2) = (4, 2)$. If the scaling factor is negative, the resulting vector will be pointing at the opposite direction as the original vector.

Vector addition simply adds matching pairs: $(x_a + x_b, y_a + y_b, z_a + z_b)$. The operation is similar to vector subtraction (just *add* negative numbers). Note, however, that vector addition is commutative, but the same does **not** apply to vector subtraction.

1.106 Vectors - 3

The Dot Product is a product operation between two vectors. It results in a scalar. The operation is simple:

$$\vec{a} \cdot \vec{b} = x_a \cdot x_b + y_a \cdot y_b + z_a \cdot z_b$$

When we compute the dot product between two unit vectors, the resulting scalar is the cosine of the angle between those unit vectors. In other words, when we want to find out the angle between any two vectors, we can simply normalize them and compute the dot product. In general, we have:

$$\cos \Phi = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|}$$

The Cross Product between two vector \vec{p} and \vec{q} , is given by $\vec{p} \times \vec{q} = |\vec{p}||\vec{q}|\sin\theta\hat{n}$, the result can be expressed in a matrix form:

$$\begin{bmatrix} y_a \cdot z_b - z_a \cdot y_b \\ z_a \cdot x_b - x_a \cdot z_b \\ x_a \cdot y_b - y_a \cdot x_b \end{bmatrix}$$

Week 2

Key Concepts

- Explain the basic idea of 3D graphics
- Create simple scenes from basic primitives or imported assets
- Evaluate and select 3D graphics engines for use in a project.

1.304 Objects in Unity

- <https://docs.unity3d.com/Manual/GameObjects.html>
- <https://docs.unity3d.com/Manual/comp-MeshGroup.html>
- <https://docs.unity3d.com/Manual/PrimitiveObjects.html>
- <https://docs.unity3d.com/Manual/AssetWorkflow.html>

1.305 Reading: Computer graphics

- Read Boreskov, A. and E. Shikin Computer Graphics: From Pixels to Programmable Graphics Hardware. (Boca Raton: CRC Press, 2013) Chapter 1.

Week 3

Key Concepts

- Use transforms and objects to create 3D graphics scenes
- Explain the basic mathematical and theoretical principles of 3D models and transforms
- Create simple scenes from basic primitives or imported assets

2.101 Transforms

Transforms allow us to position objects within a 3D scene.

2.102 Compound Objects

Compound objects allow us to position groups of objects as a single thing.

Week 4

Key Concepts

- Create simple scenes from basic primitives or imported assets
- Use transforms and objects to create 3D graphics scenes
- Explain the basic mathematical and theoretical principles of 3D models and transforms

2.301 Matrices

A matrix is a way of arranging numbers in rows and columns forming of a rectangular array. We can use matrices to store information about transformations in a 3D scene.

Matrices are defined in terms of their number of rows and columns, we write rows first. The following in a 3×3 matrix:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

And here's a 3×2 matrix:

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix}$$

And a 2×3 matrix:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}$$

Vectors are a special of matrices where they either have 1 row or 1 column.

There are several operations that can be carried out on matrices, for example they can be transposed (denoted A^T), scaled (e.g. $3 \cdot A$), multiplied by other matrices ($A \cdot B$), and so on.

2.302 Matrix Multiplication

When we multiply matrices A and B , the result will be another matrix. In order to multiply matrices, the number of **columns** in the first matrix (i.e. A) has to be the same as the number of **rows** in the second matrix (i.e. B). The resulting matrix will have the number **rows** from the first matrix and the number of **columns** from the second matrix. The example shown in figure 2 gives an idea of how this is carried out.

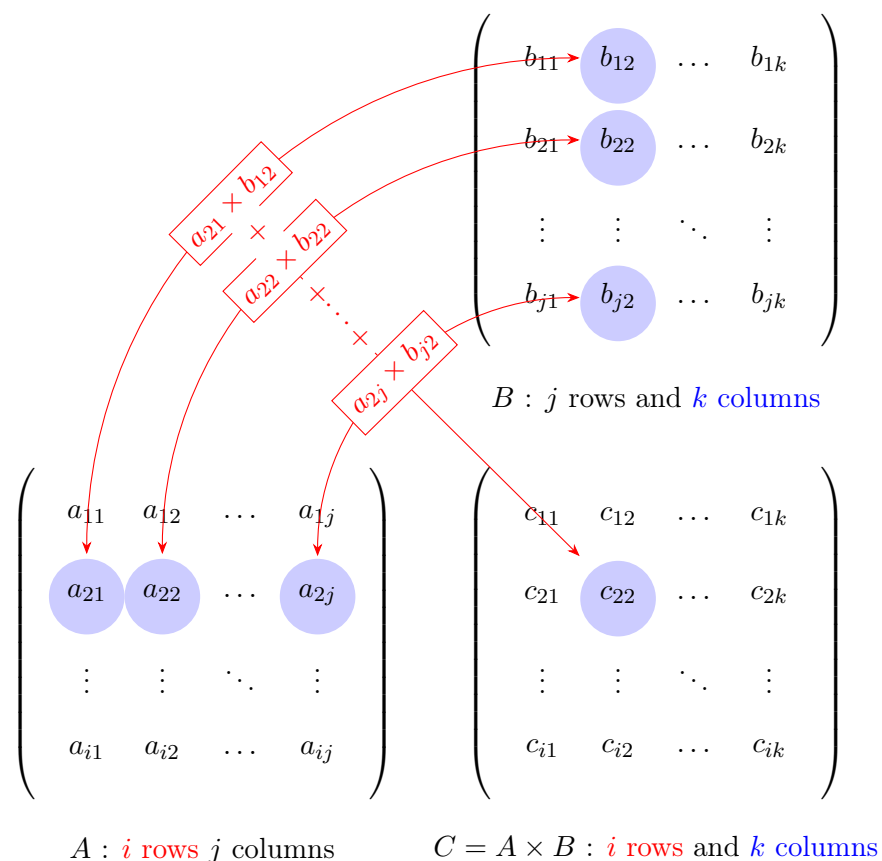


Figure 2: Matrix Multiplication

Note that for a multiplication between matrices $A_{i \times j}$ and $B_{j \times k}$, the resulting matrix will have sizes $C_{i \times k}$ (and both j must be the same).

2.303 Introduction to Transformation Matrix

There are three basic transformations: translate, rotate and scale. Behind the scenes, these are all stored as matrices and transformations are applied as matrix multiplications.

2.304 Scaling

To scale a 3D vector, the scaling factors will be held in the diagonal of the matrix, every other element is zero. A scaling matrix will look like this:

$$\begin{pmatrix} x_{scale} & 0 & 0 \\ 0 & y_{scale} & 0 \\ 0 & 0 & z_{scale} \end{pmatrix}$$

When $x_{scale} = y_{scale} = z_{scale}$ we end up with a uniform scaling, otherwise we have a non-uniform scaling.

2.305 Rotation

A rotation matrix will be composed of sin and cos of the angle θ^1 . Depending on the axis of rotation, the matrix is different.

Here's the matrix for rotation around the z axis:

$$\begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Here's the matrix for rotation around the y axis:

$$\begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}$$

Here's the matrix for rotation around the x axis:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix}$$

The order of rotations matters, which means rotation transformations are **not** commutative.

2.306 Transformation Matrix - translation and summary

For translations, we must add an extra column and extra row to our transformation matrix and one more row to our 3D vector. The operation looks like so:

¹In practice, however, rotations in 3D are carried out with the use of Quaternions.

$$\begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix}$$

where t_x , t_y , and t_z store the translation amount for each axis.

2.308 Transforms in Unity

- [Unity Game Objects Tutorial](#)
- [Unity Transform Class Reference](#)

2.309 Reading: Computer graphics

- Read Boreskov, A. and E. Shikin Computer Graphics: From Pixels to Programmable Graphics Hardware. (Boca Raton: CRC Press, 2013) Chapter 2 and sections 4.1-4.8.