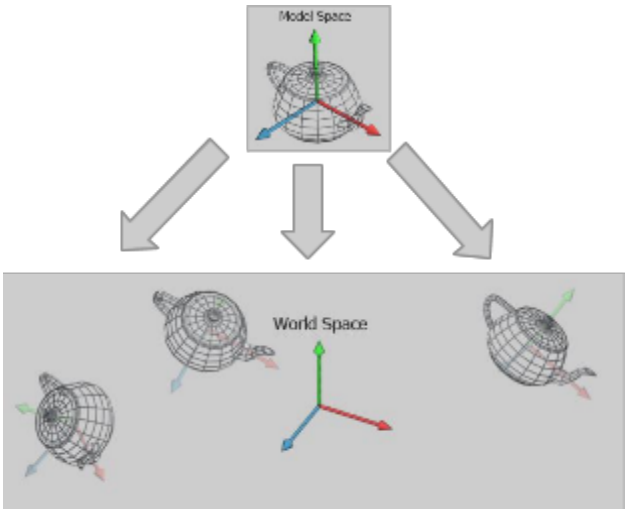
CMSC 161 UV-1L
Interactive Computer Graphics
Meeting 05 - Graphical Projection

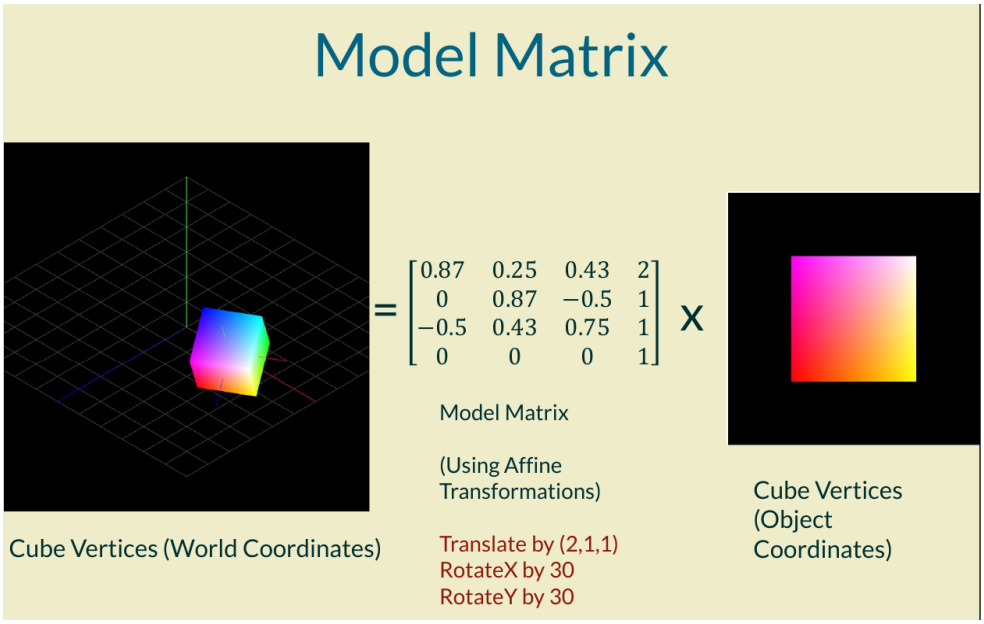
Model Transform

File: 05-projection01.html
05-projection02.html

Models/Objects are always defined on its own coordinate system. The model transform is the one responsible for the placement of the objects from its own coordinate system (object coordinates) to the world coordinate system (world coordinates).



Model transform is done by multiplying an affine transformation matrix to the actual model.

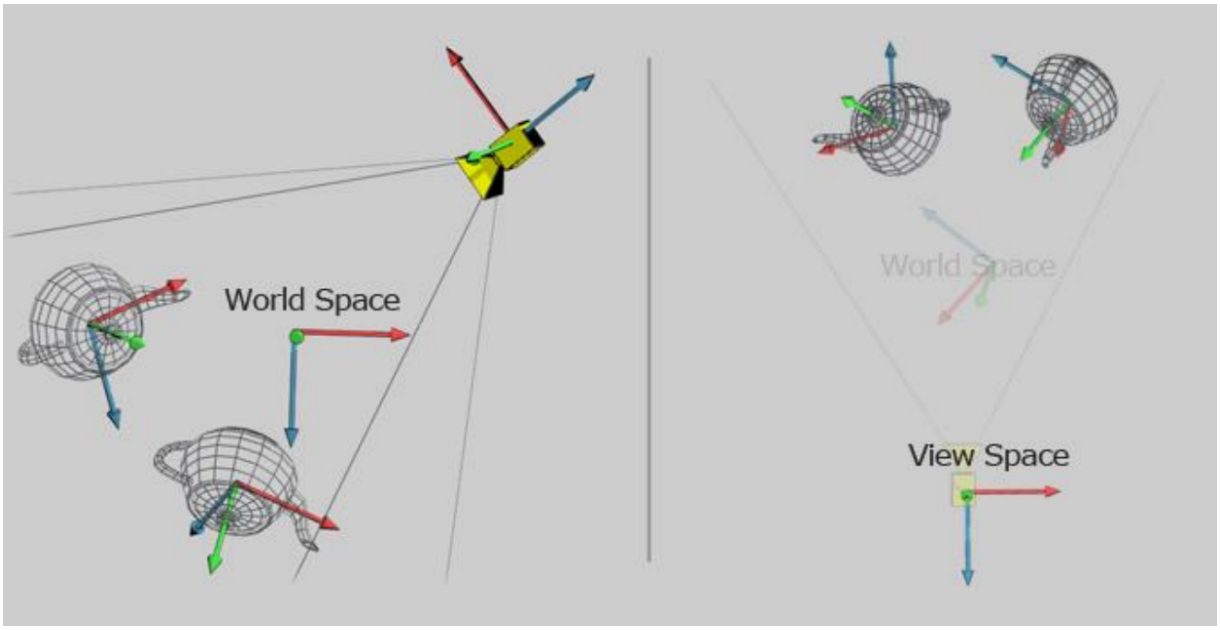


CMSC 161 UV-1L
Interactive Computer Graphics
Meeting 05 - Graphical Projection

View Transform

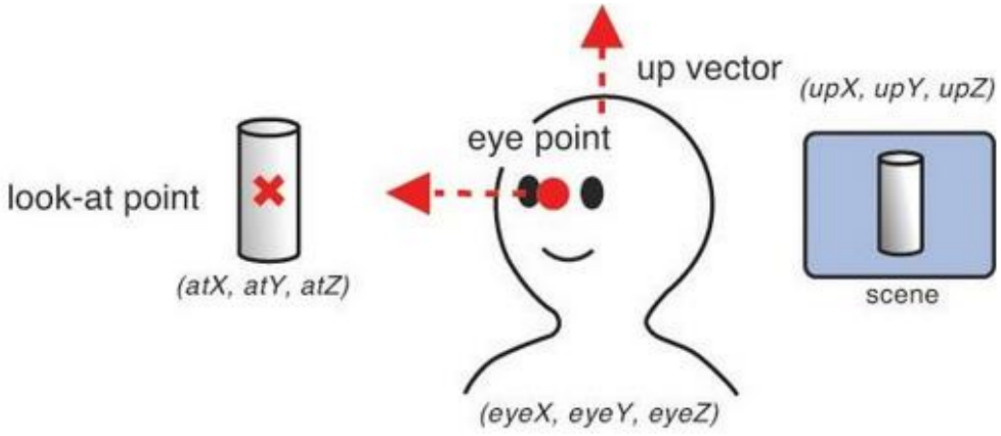
File: 05-projection03.html

View transform emulates the camera/eye location on the world coordinates. View transform utilizes the view matrix.



3 parameters are needed to create the correct view matrix:

1. Look at point - a point where the camera is looking
2. Camera/Eye point - a point where the camera is located
3. Up vector - a vector that represents the orientation of the camera



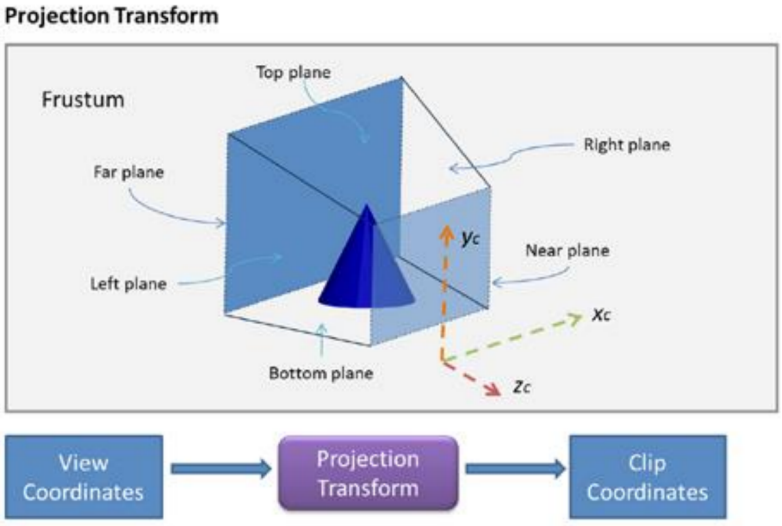
CMSC 161 UV-1L
Interactive Computer Graphics
Meeting 05 - Graphical Projection

Projection Transform and Perspective Division

File: 05-projection03.html
05-projection04.html

Projection Transform

Projection transform determines the amount of view space that will be rendered using a square frustum. This utilizes the concept of the projection matrix.

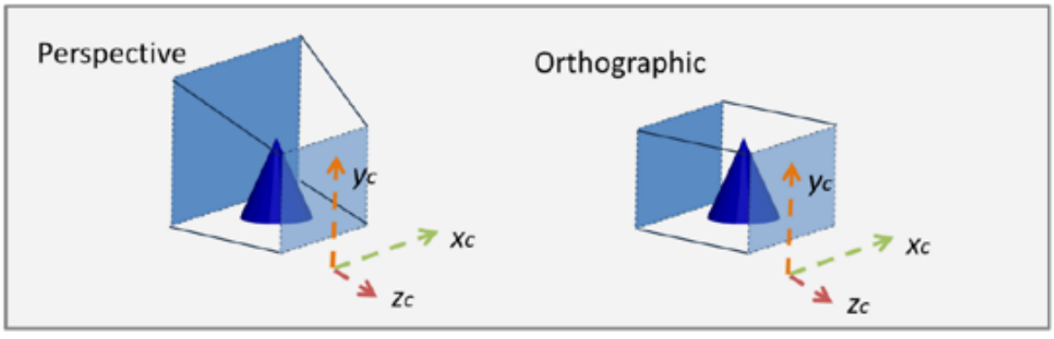


The frustum determines which objects or portion of objects will be *clipped out* and discarded.

2 types of projection transform can be implemened

1. Orthographic Projection (Parallel) - size of the near plane == far plane
2. Perspective Projection - size of the near plane < far plane

Frustum shape



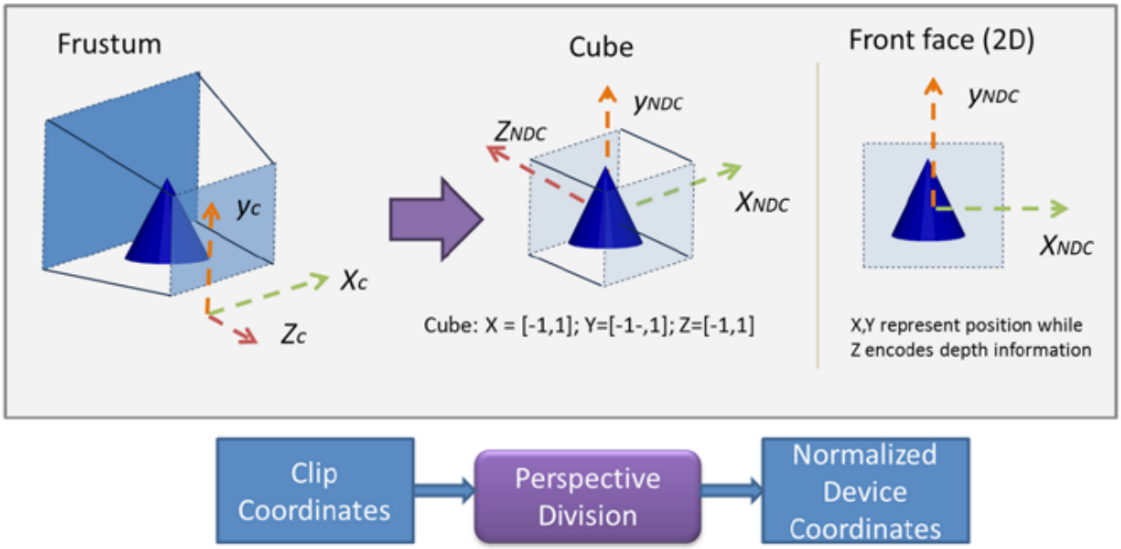
The extent and shape of the frustum determines how much of the 3D view space Is mapped to the screen and the type of 3D to 2D projection that takes place.

CMSC 161 UV-1L
Interactive Computer Graphics
Meeting 05 - Graphical Projection

Perspective Division

The result of the projection transform (clip coordinates) will be mapped to the normalized device coordinates (range of values from -1 to 1)

Normalized Device Coordinates

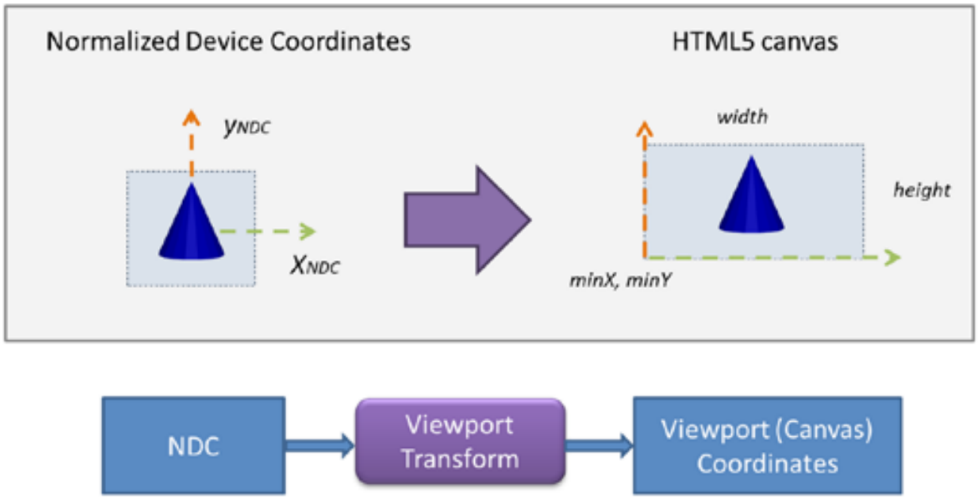


Viewport Transform

File: *05-projection05.html*

The `gl_position` variable expects the vertices to be the normalized device coordinates (NDC) form. The NDC are mapped to the viewport coordinates.

Viewport Transform



Depth Testing
File: 05-depth01.html

By default, WebGL does not consider the z-coordinate of the scene when handling depth. It draws the objects in order and overwrites them respectively.

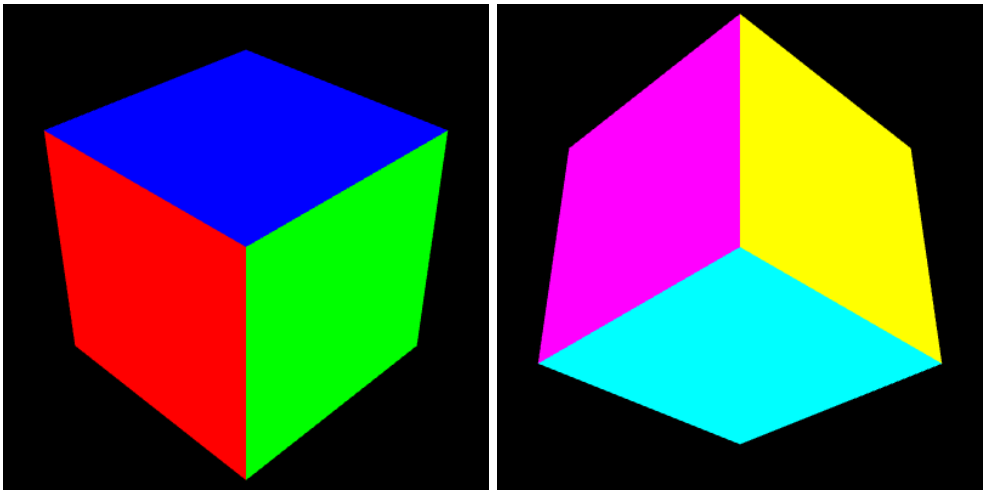
To enable the depth testing algorithm, the following lines must be included

```
gl.enable(gl.DEPTH_TEST);
gl.clear(gl.DEPTH_BUFFER_BIT);
```

CMSC 161 UV-1L
Interactive Computer Graphics
Meeting 05 - Graphical Projection

Exercise (Part 1)

Using WebGL, draw a scene with a cube the looks like the image below



The cube must be rotating randomly showcasing its different colored sides.

Scoring
4 points - cube properly modeled with its different colors
7 points - cube can be perfectly seen using the correct graphical projections
10 points - cube is rotating randomly on some axis to showcase its different colored sides