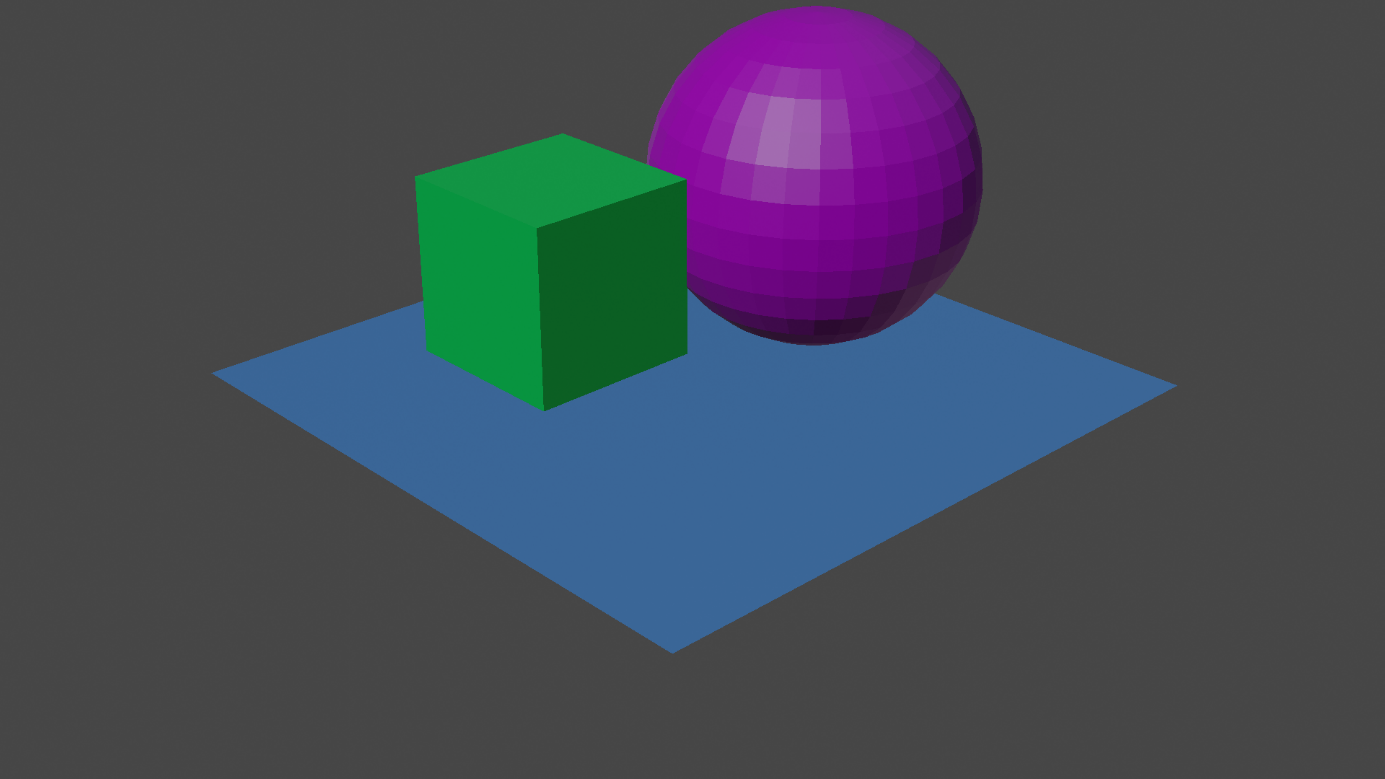
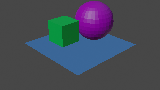
**Project 3**

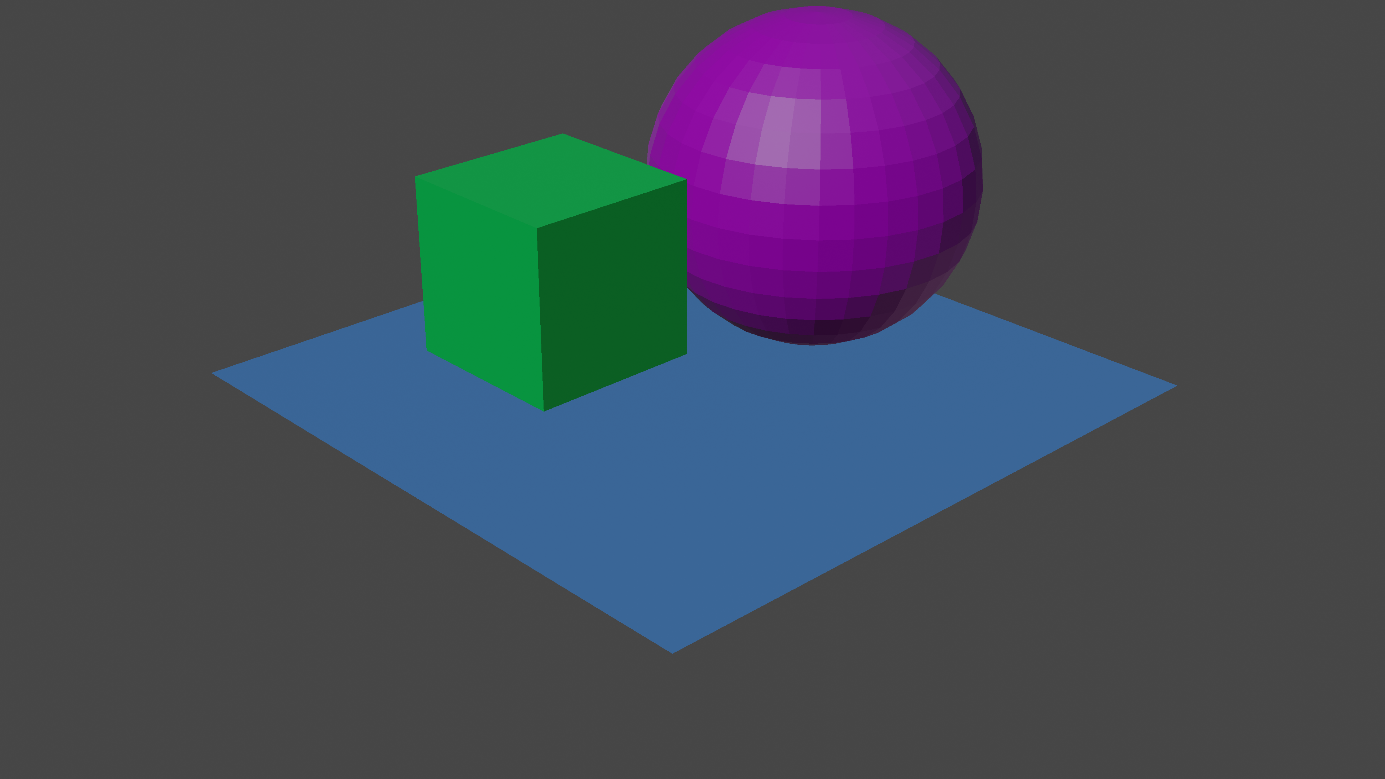
**Part 1-**

Resolution Comparison:

After setting the scene in Blender, the image was rendered three times with different resolutions. One is the 1920x1080 resolution which will be used as the default case on the following sections for the same .blend file. The second image is rendered with resolution 160x90 and the third image is rendered with resolution 3840x2160. The differences that the differing resolutions caused will be discussed in this part.

1. Rendered image with 1920x1080 resolution
2. Rendered image with 160x90 resolution



1. Rendered image with 3840x2160 resolution
2. Comparison:

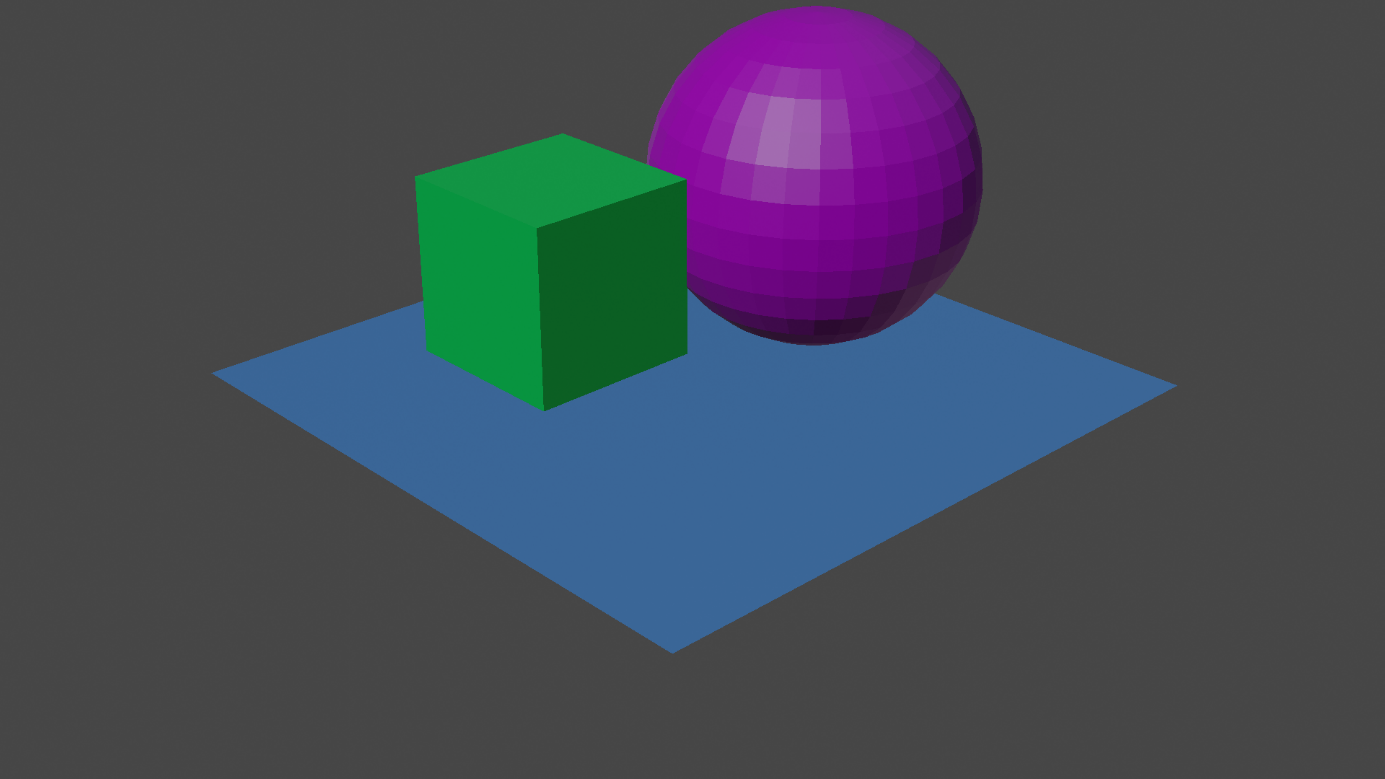
Increasing or decreasing resolution changes the amount of pixels used to display the image. Halving 1920x1080 would give us 960x540. If we did the multiplication, we would see that the pixel amount was reduced to 25% of its original amount.

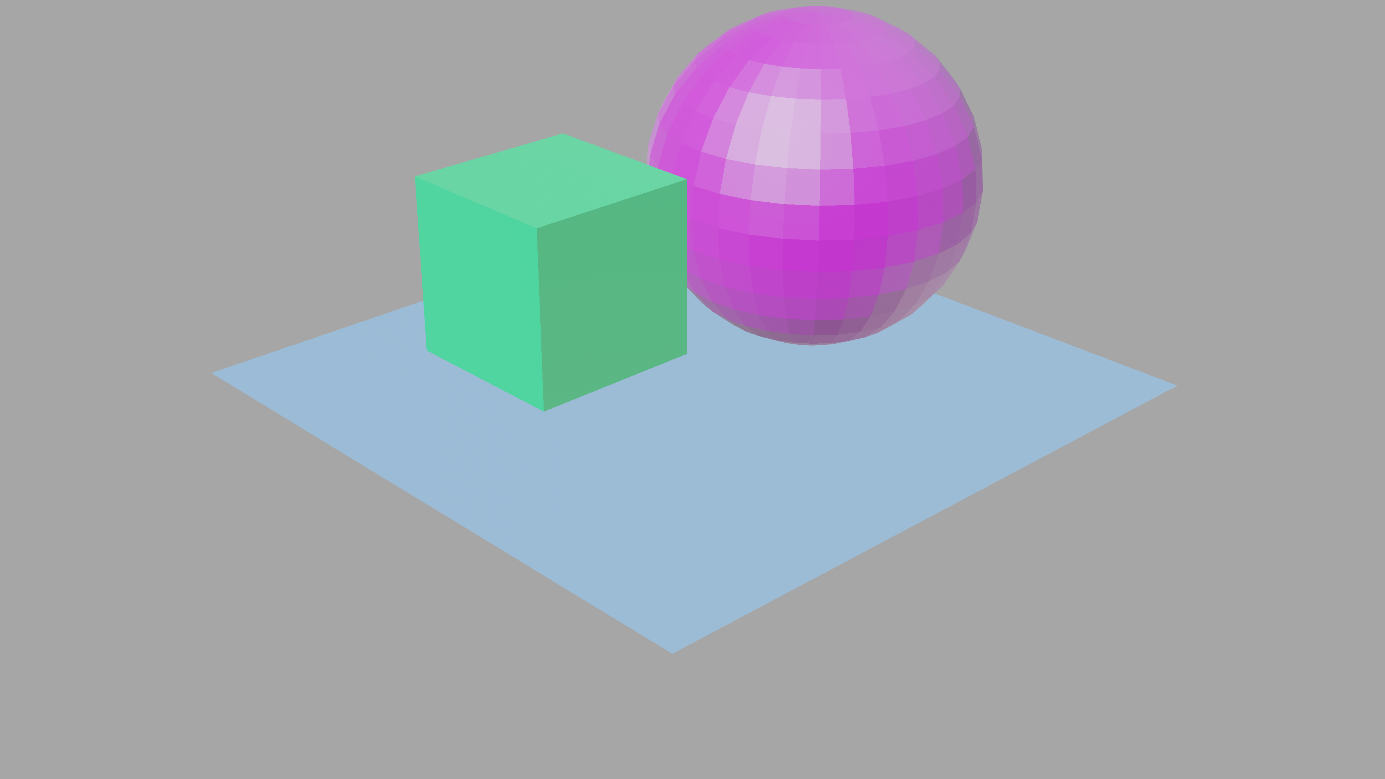
With more pixels used to draw on the same area on a display, the image will be more detailed on the image with more resolution. On this example, Image C has four times the pixels Image A has, wheras Image A has 12 times 144 times the pixels Image B has.

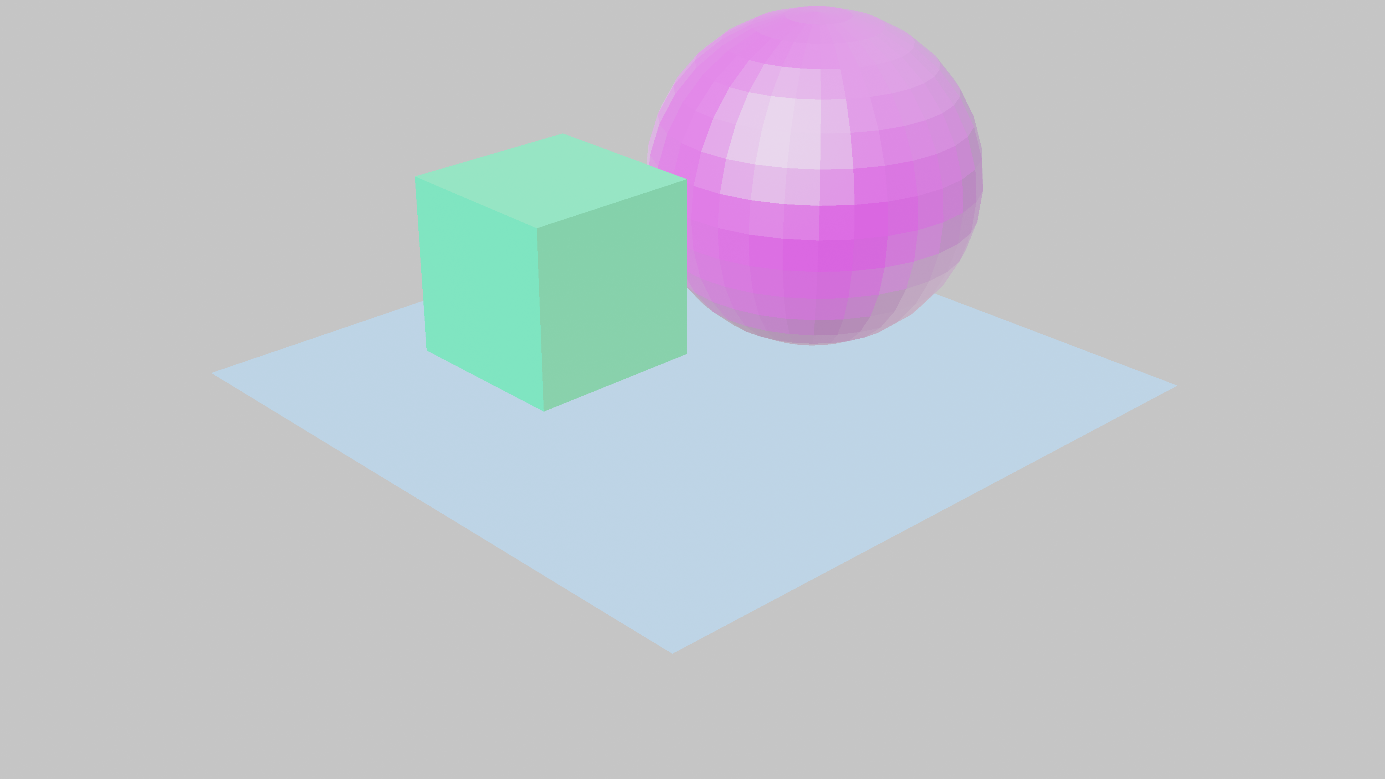
It is apparent how the Image B was affected by the low resolution as the vertices and edges are more noticably jagged and the image itself is shrinked. If Image B was to be resized to Image A or C’s size, it would look way low quality.

Gamma Comparison:

The image on the 1920x1080 resolution was rendered with different gamma values, the values being 1, 3 and 5. The gamma values’ difference in the visuals will be discussed in the following part.

1. Rendered image with gamma 1
2. Rendered image with gamma 3



1. Rendered image with gamma 5
2. Comparison:

Increasing the gamma caused the images to appear brighter. While the parts of the image that were already white/black colours didn’t change much, the other colours have had a whitening of sorts that makes the scene appear brighter.

1. What’s the advantage of using YUV colour space?

YUV colour spaces are more efficient in capturing colours than RGB. While the U and V signify axes that make combinations of colours, the Y value is used for luminance and it has a bigger sample rate than U or V.

**Part 2-**

**Order of Rotations**

1. pxy

Rx x Ry :

m11 = (1\*cos)+(0\*0)+(0\*(-sin) = cos

m12 = (1\*0)+(0\*1)+(0\*0) = 0

m13 = (1\*sin)+(0\*0)+(0\*cos) = sin

m21 = (0\*cos)+(cos\*0)+((-sin\*(-sin) = sin2

m22 = (0\*0)+(cos\*1)+((-sin\*0) = cos

m23 = (0\*sin)+(cos\*0)+((-sin\*cos) = -sin\*cos

m31 = (0\*cos)+(sin0)+(cos(-sin) = -sin\*cos

m32 = (0\*0)+(sin1)+(cos0) = sin

m33 = (0\*sin)+(sin0)+(cos) = cos2

px = 0.7071 + 0.7071 = 1.4142

py = = 0.5 + 0.7071 – 0.5 = 0.7071

pz = = -0.5 + 0.7071 + 0.5 = 0.7071

1. pyx

Ry x Rx :

m11 = (cos\*1)+(0\*0)+(sin\*0) = cos

m12 = (cos\*0)+(0\*cos)+(sin\*sin) = sin2

m13 = (cos\*0)+(0\*(-sin))+(sin\*cos) =sincos

m21 = (0\*1)+(1\*0)+(0\*0) = 0

m22 = (0\*0)+(1\*cos)+(0\*sin) = cos

m23 = (0\*0)+(1\*(-sin))+(0\*cos) = -sin

m31 = (-sin)+(0\*0)+(cos) = -sin

m32 = (-sin)+(0\*cos)+(cos) = sin\*cos

m33 = (-sin)+(0\*(-sin))+(cos\*cos) = cos2

px = cos = 0.7071 + 0.5 + 0.5 = 1.7071

py = = 0 + 0.7071 – 0.7071 = 0

pz = = -0.7071 + 0.5 + 0.5 = 0.2929

1. Explain how you got these results

Matrix multiplication orders affect the outcome, by applying a certain transformation before the second one -or vice versa- one can obtain entirely different results. To demonstrate this, the rotations on x and y axes were applied two times: on the first example, Ry was applied first and Rx was applied second, resulting in the Rx xRy = Rxy. Whereas on the second example, the Rx rotation was applied first and the transformation was thus Ry x Rx = Ryx. Applying these two different transformation matrices to the vector from the center of the cube to a vertex gave the position of that vertex after the respective transformation.

**Parenting Objects**

In this section, a cube that was given the translation was parented by a plane. This made the original transformation the cube’s local position according to the center of the plane. This brought about the question of how the global position of the cube changes when transformations are applied to the plane.

Since the cube is a child of the plane, it also undergoes whatever transformation the plane undergoes. Translating the plane by , the global location vector of the cube should be added to this new translation.

This time, the plane is rotated along it’s local X axis by -45 degrees. So, the cube must also go through the same rotation, according to the plane’s center. Since the cube’s position according to the plane is already known to be , we must first translate the cube with , then apply the and finally take the cube back to where it is supposed to be with .

:

m11

m12

m13

m14

m21

M22

M23

M24

M31

M32

M33

M34

M41

M42

M43

M44

Rotated -45 x axis

1. Explain how you got these results