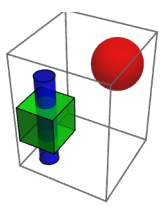
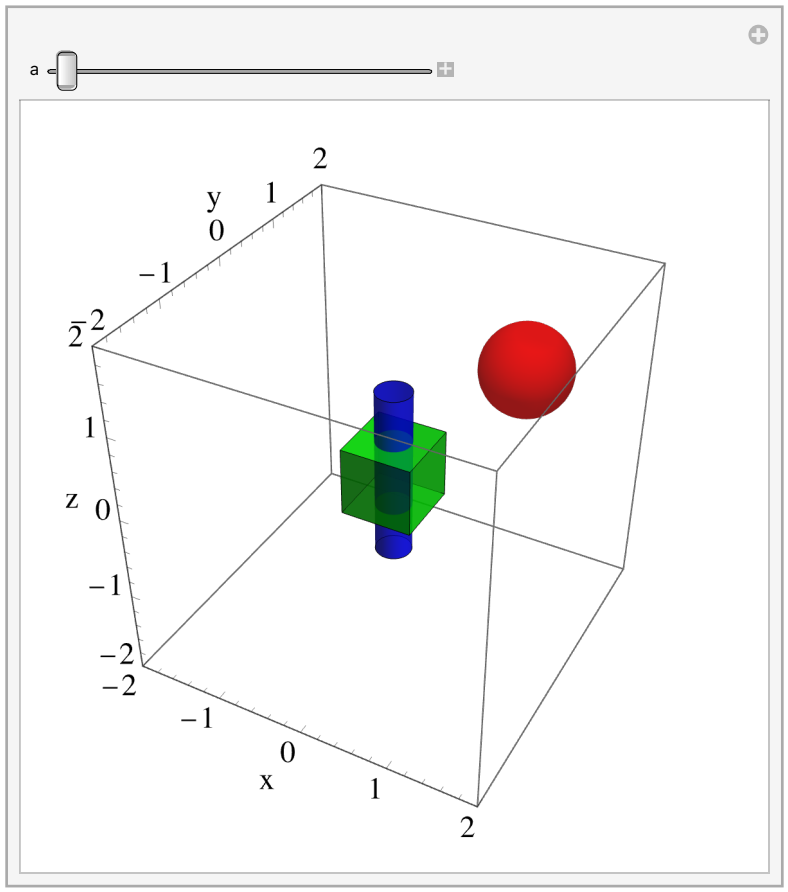
```
SetOptions[EvaluationNotebook[], ShowCellLabel → False];

SetOptions[{Plot, Plot3D, ContourPlot, DensityPlot, ParametricPlot, ParametricPlot3D, ListPlot, ListLinePlot,
    VectorPlot, VectorPlot3D, StreamPlot, RevolutionPlot3D, ContourPlot3D, Graphics, Graphics3D},
  BaseStyle → {16, FontFamily → "Times"}, ImageSize → Medium];

(*********************************)(* PROBLEM SET 1 *)(*********************************)

(*
Problem 1.1: Create the following primitive objects:
   a) A sphere with radius 0.5 at position {1, 1, 1} (Red Color)
   b) A cylinder with radius 0.2 and height 2, positioned along the z-axis (Blue Color)
   c) A cube with side length 0.8 centered at the origin (Green Color)
Then rotate the entire set around the y-axis
*)

(* Create basic primitives *)
[redacted]

(* Combine objects *)

compoundObject1 = {sphere1, cylinder1, cube1};

(* Check *)
[redacted]
```



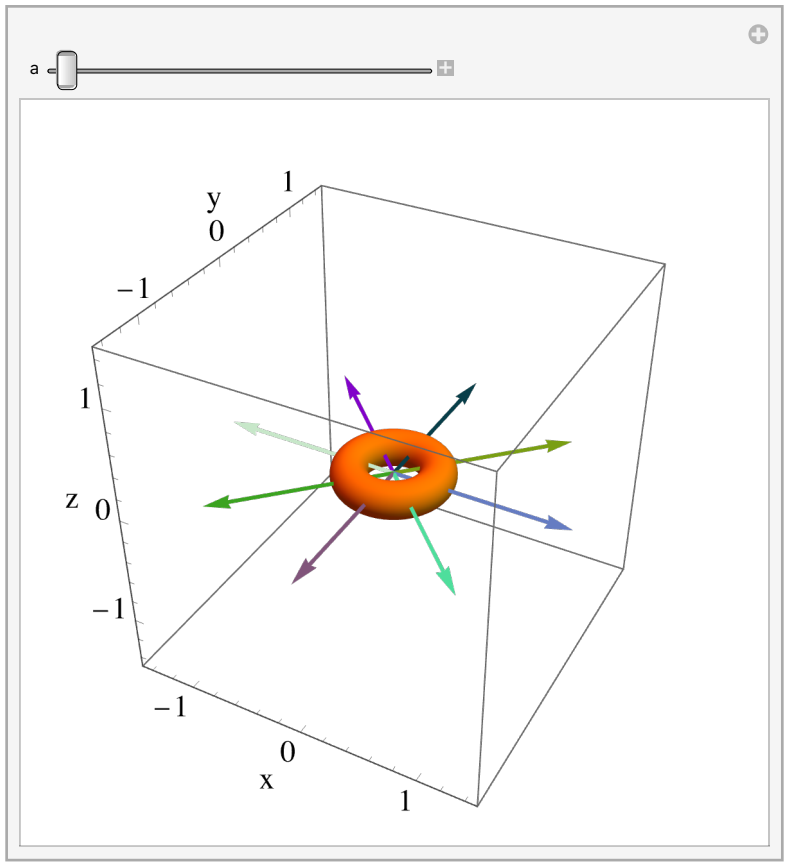
```
(* Rotation function around y-axis using GeometricTransformation *)
[redacted]

(* Graphics function to display the rotated object *)
[redacted]

(* Manipulate to animate the rotation 0-2π *)
```

```
Manipulate[displayRotated1[a], {a, 0, 2 * Pi}]
```




```
(*
Problem 1.2: Create a dynamic scene with:
   a) A table of 8 arrows pointing outward from origin (with random color)
   b) A torus at the origin
   c) Animate to rotate the arrows around the x-axis while keeping the torus static (don't move)
*)

(* Create table *)
[redacted]

(* Create 8 arrows pointing outward from origin *)
[redacted]

(* Rotation function for just the arrows *)
[redacted]

(* Manipulate to animate the rotation *)
```

```
Manipulate[displayScene1[a], {a, 0, 2 Pi}]
```




```
(*
Problem 1.3: Load an external STL file (CubCr.stl)
   a) Import the STL file and display it in 3D space
   b) Create a function to rotate the model around an anchor point {5, 0, 0}
   c) Implement a geometric transformation that combines translation and rotation
   d) Visualize the rotation with a yellow sphere (radius = 1) showing the anchor point
   e) Create an animation that rotates the model around the x-axis
*)
```

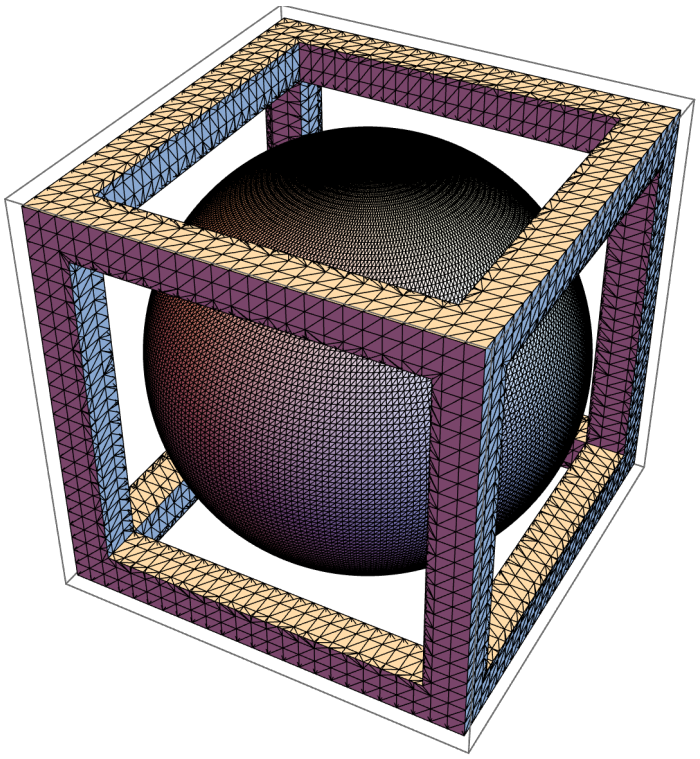
```
SetDirectory["[redacted]"];
```

```
(* Import the STL file as polygon objects *)
```

```
[redacted]
```

```
(* Check *)
```

```
Graphics3D[cube]
```



```
Paround1 = {5, 0, 0};

RR1 = {0, 0, 5.5};

(* Create a function that applies geometric transformations to rotate the object *)
[redacted]

(* Create a visualization function that displays both rotated object *)
[redacted]

(* Manipulate *)
```

```
Manipulate[displayAnchoredRotation[a], {a, 0, 2 * Pi}]
```