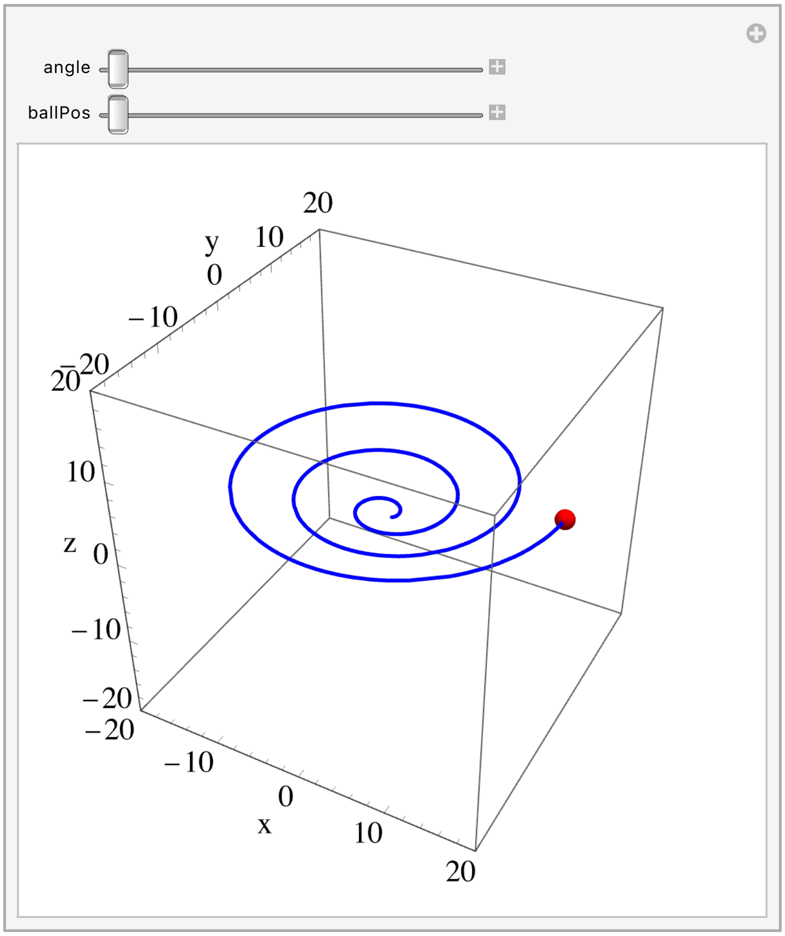
```
SetOptions[EvaluationNotebook[], ShowCellLabel → False];

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

(********************************)
(* PROBLEM SET 2 *)(********************************)

 (*
Problem 2.1: Create a scene with multiple objects:
   a) A parametric curve
      following {t*Cos[t], t*Sin[t], t/3} for t from 0 to 6π
   b) A sphere that moves along the curve
   c) Animate the rotation of the entire setup around axis {1,1,1}
*)

paramCurve[t_] := {t * Cos[t], t * Sin[t], t / 3}

tMax = 6 * Pi;

(* Creates a 3D plot of the parametric curve + Extract *)

[redacted]⟦1⟧;

(* Create function (rotation transformation) *)

[redacted]

[redacted]

(* Manipulate *)
```

```
Manipulate[displayRotated2[angle, 1 - ballPos], {angle, 0, 2 Pi}, {ballPos, 0, 1}]
```

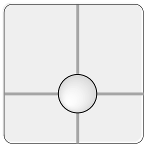


```
(* Problem 2: Creating and Transforming a Helix
   Create a helix curve defined by {Sin[t], Cos[t], t/4} for t from 0 to 8π.
   Implement a rotation around a user-controlled anchor point on the xy-plane.
   The anchor point position is controlled by Slider2D.
   The rotation is around the vector {0,0,1} (z-axis).

   Start with Clear[anchorX, anchorY, angle] where:
   - anchorX, anchorY are the x,y coordinates of the anchor point
   - angle is the rotation angle

   Use PlotRange → 5, Boxed → False, AxesOrigin → {0,0,0}
*)
```

```
helix[t_] := {Sin[t], Cos[t], t / 4}
```

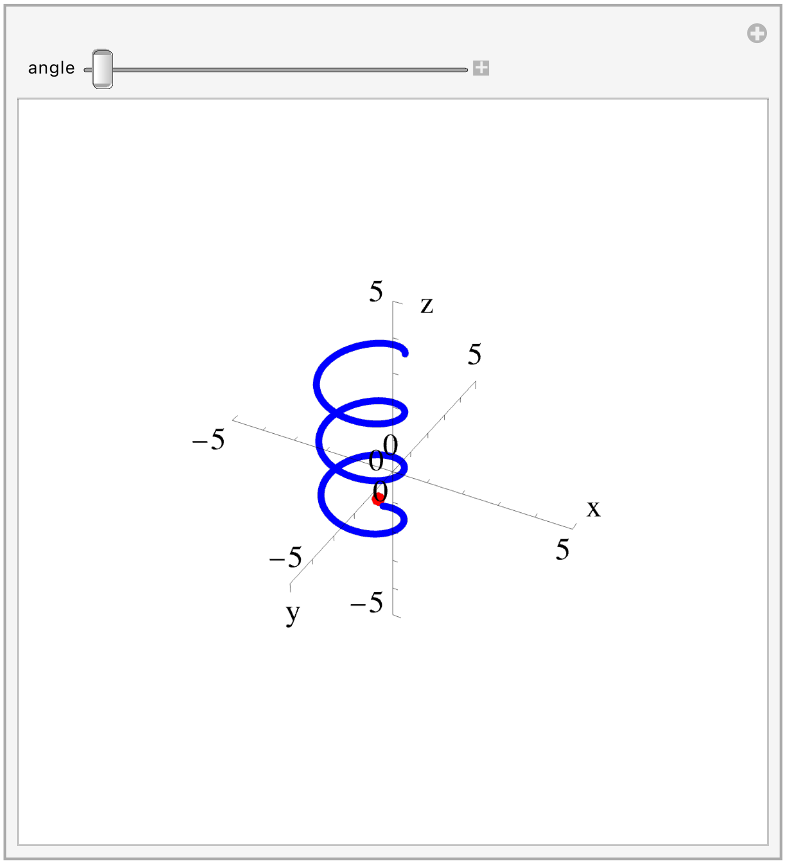
```
(* Creates a 3D plot *)
```

```
[redacted]

(* Extracts the graphical primitive *)

[redacted]

(* Transformation function *)

rotateAroundAnchor[obj_, x_, y_, a_] :=
  [redacted]

displayRotatedHelix[x_, y_, a_] :=
  [redacted]

(* Slider 2D *)

anchorSlider = Slider2D[Dynamic[{anchorX, anchorY}], {{-3, -3}, {3, 3}}]
```

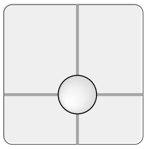


```
{Manipulate[displayRotatedHelix[anchorX, anchorY, angle], {angle, 0, 2*Pi}],
 anchorSlider, Dynamic[{anchorX, anchorY}]}
```

{



,



, {0.24, -1.2}}

```
(* Problem 3: Create a interpolated 3D curve with a moving point
   1. Create a 3D spline curve from a given set of points
   2. Add a slider to control a point that moves along the curve
   3. Display both the curve and the moving point
*)
```

```
dataPoints = {{0, 0, 0}, {1, 2, 1}, {3, 1, 2},
   {4, -1, 3}, {2, -2, 2}, {0, -1, 1}, {-1, 1, 0}, {2, 1, 5}};

pointPlot =
  ListPointPlot3D[dataPoints, PlotStyle → {Red, PointSize[0.03]}, PlotRange → 5];

(* Extract x, y and z *)
```

[redacted]

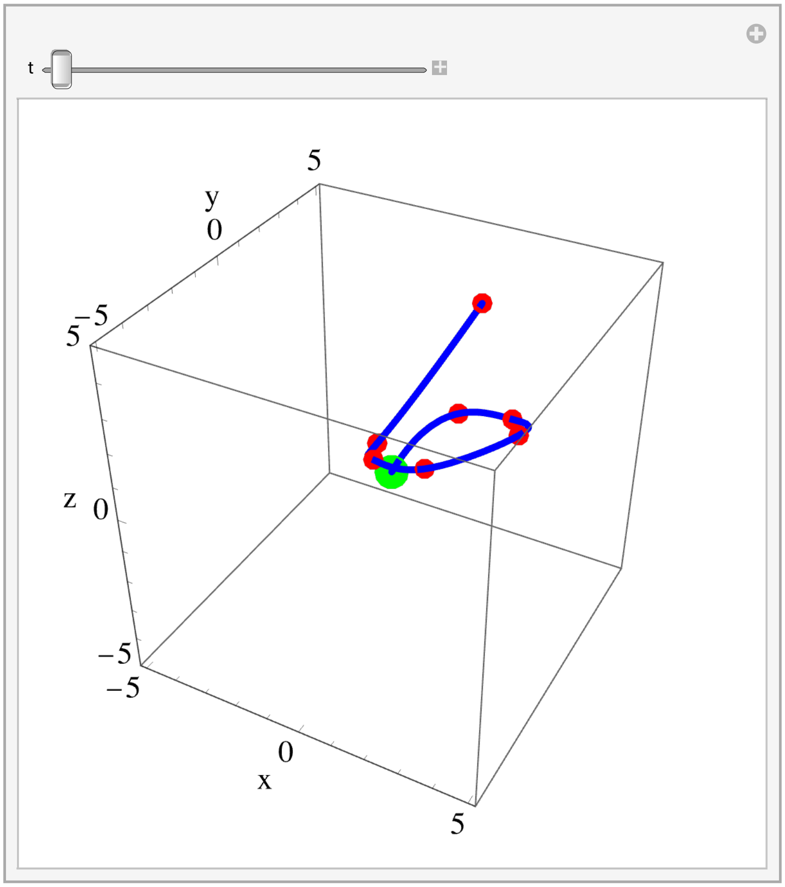
```
(* Parametric Interpolation *)
```

[redacted]

```
(* 3D parametric function *)

splineCurve[t_] = {xInterp[t], yInterp[t], zInterp[t]};

tmax = Length[dataPoints];

(* Create a function to plot the curve *)

curvePlot = ParametricPlot3D[splineCurve[t],
   {t, 1, tmax}, PlotStyle → {Blue, Thickness[0.01]}];

(* Create a function to show a point at a specific position on the curve *)

movingPoint[t_] := [redacted]
```

[redacted]

```
(* Manipulate *)

Manipulate[showCurveWithPoint[t], {t, 1, tmax, 0.1}]
```


t
5
y
0
−5
5
z
0
−5
−5
0
x
5