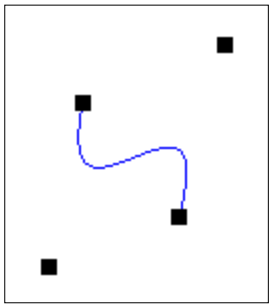
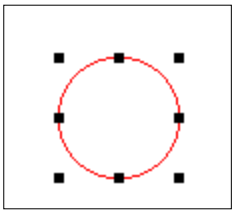
Exercise 11: Geometric Modeling – Parametric Curves and Surfaces

Reading	Angel: Angel: chap.10
Purpose	The purpose of the exercise is to understand the theory behind parametric curves and surfaces and how to use this theory to create geometric models. The theory includes Bezier, uniform B-spline, non-uniform B-spline and NURBS (non-uniform rational B-spline) curves and surfaces. The advantage of using a parametric representation of curves and surfaces is that they can be represented using only a few control points instead of a detailed mesh.
Part 1 Bezier curves 	Exercise 02561-11-01 contains a parametric line and 4 control points that the user can move around the screen using the mouse. The parameter domain u is currently defined to be between 0.0 and 1.0. Change the program to use cubic Bezier to map from the parameter domain to the Bezier curve. • First compute the curve matrix in the `renderScene()` function of the C++ file (this should be $\mathbf{M_B P}$ – note that P is a 4x4 matrix) • Modify the vertex shader to compute the position using the bezier formula (remember that the curve matrix contains the $\mathbf{M_B P}$ matrix). $$p(u) = (1, u, u^2, u^3)^T \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} P1 \\ P2 \\ P3 \\ P4 \end{bmatrix} = u^T M_B P$$ **Formula for cubic Bezier**

Exercise 11: Geometric Modeling – Parametric Curves and Surfaces

Part 2
NURBS circle



One of the advantages of using NURBS is that it is possible to create an exact circle.

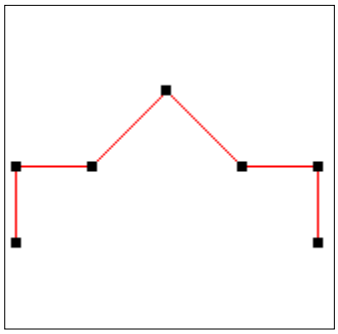
Exercise 02561-11-02 currently creates a Bezier curve using NURBS.

- Change the `setupCurve` function to create a circle using the input given below. Note that you have to use `vec4` for control points where the weight is stored in the `w` field of the `vec4`.

Circle control points
Position 9 control points in a rectangle; 4 in the corners, 4 between the corners (as the figure to the left) and the last control point should equal the first control point (to close the circle). The starting/end point should be one of the midpoints. The corner control points should have the weight $\sqrt{2}/2$ and midpoints should have the weight 1.

Circle knot vector
{ 0, 0, 0, π/2, π/2, π, π, 3π/2, 3π/2, 2π, 2π, 2π }

Part 3
NURBS curves



Exercise 02561-11-03 shows 7 control points. The program has the following different modes, which can be changed using keypress '1' to '5':

1. Linear: Uses linear interpolation between the control points
2. Bezier
3. Uniform B-Spline
4. Non-uniform B-Spline
5. NURBS (Non-Uniform Rational B-Spline)

Linear interpolation is already implemented. Implement the functions:

- `setupBezier()`. Use only the first 4 control points and the order of the curve should be 4.
- `setupUniformBSpline()`. All 7 control points must be used. The degree should be 3.
- `setupNonUniformBSpline()`. All 7 control points must be used. The curve should start at the first control point and end in the last control point.
- `setupNURBS()` - use weigths after your choice

Exercise 11: Geometric Modeling – Parametric Curves and Surfaces

Part 4
NURBS surface patch

Exercise 02561-11-04 shows a non uniform b-spline surface patch using 7 x 2 control points.

- Change the number of control points to be 7 x 7 and update the knot vectors to fit the new control points. The control points should not be linear but should have some curvature in both direction.

The image on the left shows an example of how this could look.