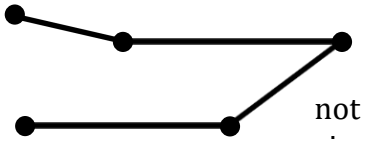
Describing Shapes

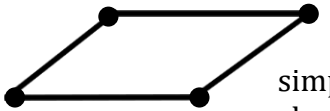
Constructing Objects in Computer Graphics

2D Object Definition (1/3)

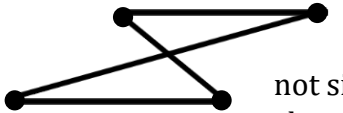
- Lines and polylines:
  - Polylines: lines drawn between ordered points
  - A closed polyline is a polygon, a simple polygon has no self-intersections



not closed,
simple polyline

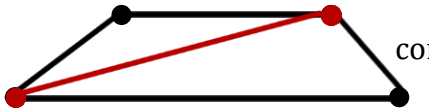


simple polygon and
closed polyline

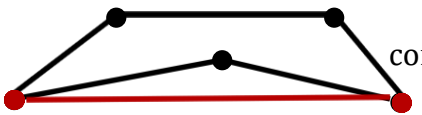


not simple polygon,
closed polyline

- Convex and concave polygons:
  - Convex: Line between any two points is inside polygon
  - Concave: At least one line between two points crosses outside polygon



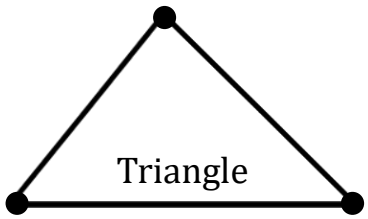
convex

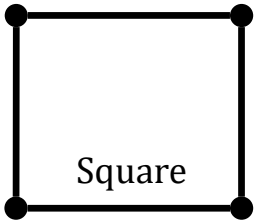


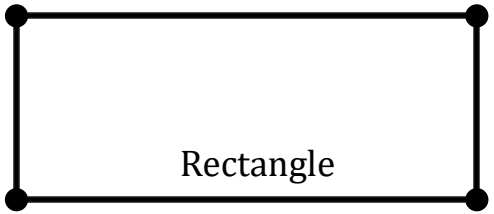
concave

2D Object Definition (2/3)

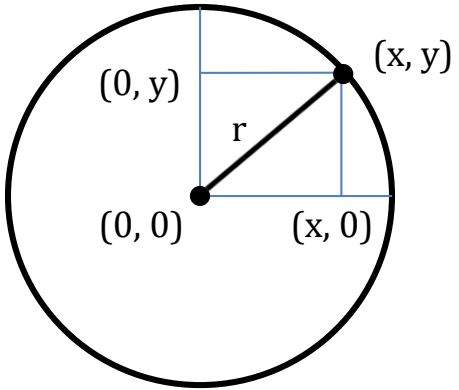
- Special Polygons:





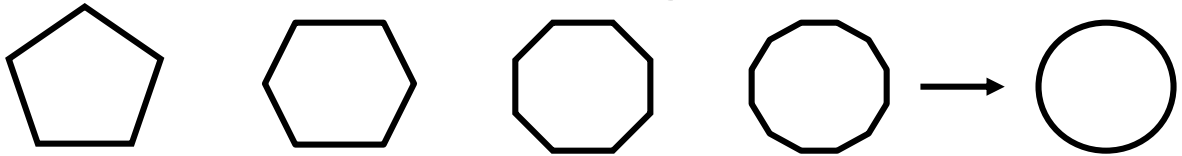


- Circles:
  - Set of all points equidistant from one point called the center
  - The distance from the center is the radius *r*
  - The equation for a circle centered at (0, 0) is $r^2 = x^2 + y^2$

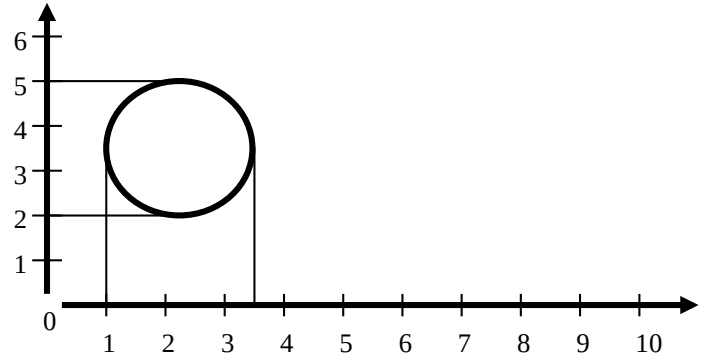


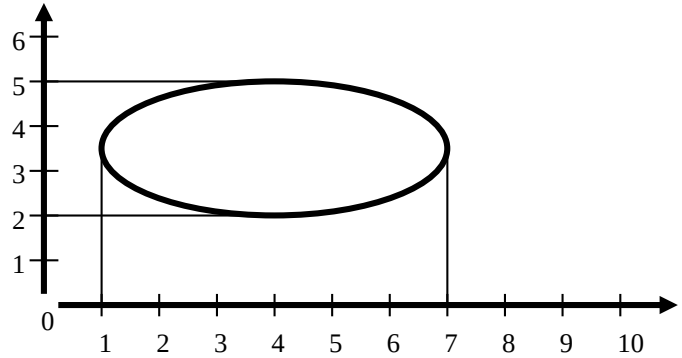
2D Object Definition (3/3)

- A circle can be approximated by a polygon with many sides.



- Axis aligned ellipse: a circle scaled in the x and/or y direction





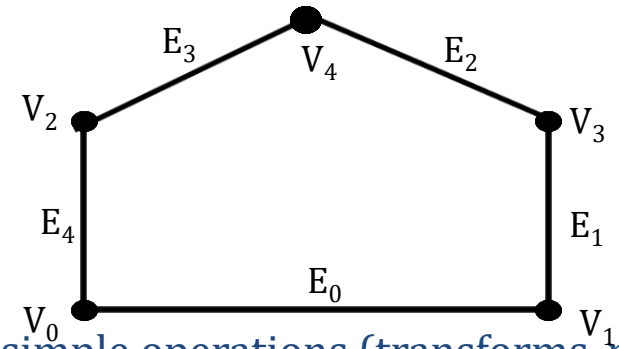
Scaled by a factor of 2 in the x direction and not scaled in the y direction. Width changes from 3.5 to 7.

Representing Shapes

- Vertex and edge tables:
  - General purpose, minimal overhead, reasonably efficient
  - Each vertex listed once
  - Each edge is an ordered pair of indices to the vertex list

	Vertices
0	(0, 0)
1	(2, 0)
2	(0, 1)
3	(2, 1)
4	(1, 1.5)

	Edges
0	(0, 1)
1	(1, 3)
2	(3, 4)
3	(4, 2)
4	(2, 0)



  - Sufficient to draw shape and perform simple operations (transforms, point inside/outside)
  - Edges listed in counterclockwise winding order for consistency with 3D where we need to compute outward-facing normals

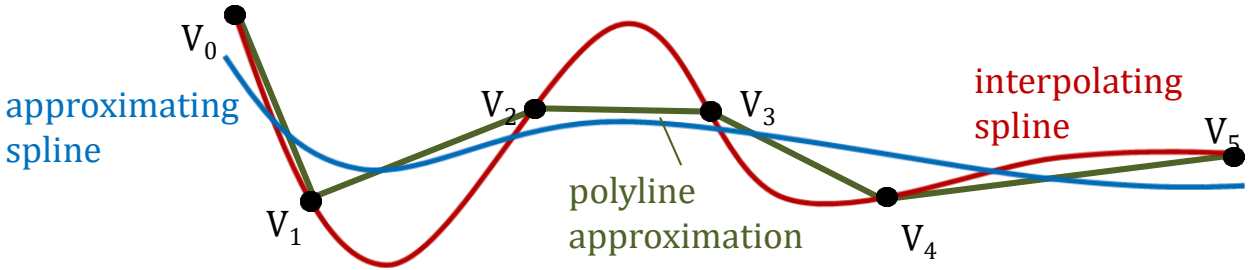
Splines (1/5) – Representing General Curves

- We can represent any polyline with vertices and edges. What about curves?
  - Don't want to store curves as raster graphics (aliasing, not scalable, memory intensive). We need a more efficient mathematical representation
  - Store control points in a list, find some way of smoothly interpolating between them
  - Closely related to curve-fitting of data, done by hand with "French curves", or by computation

- Piecewise Linear Approximation
  - Not smooth, looks awful without many control points
- Trigonometric functions
  - Difficult to manipulate and control, computationally expensive
- Higher order polynomials
  - Relatively cheap to compute, only slightly more difficult to operate on than polylines

Splines (2/5) - Spline Types and Uses

- Polynomial interpolation is typically used. Splines are parametric curves governed by control points or control vectors, third or higher order
- Used early on in automobile and aircraft industry to achieve smoothness – even small differences can make a big difference in efficiency and look



Splines still exist outside of computers. They're now called flexible curves.

- Used for:
  - Representing smooth shapes in 2D as outlines or in 3D using "patches" parameterized with two variables: *s* and *t* (see slide 13)
  - Animation paths for "tweening" between keyframes
  - Approximating "expensive" functions (polynomials are cheaper than log, sin, cos, etc.)

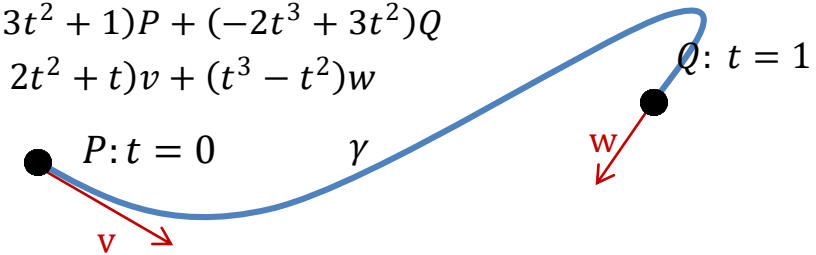
Splines (3/5) - Hermite Curves

- Polylines are linear (1st order polynomial) interpolations between points
  - Given points P and Q, line between the two is given by the parametric equation:

$$x(t) = (1-t)P + tQ, \qquad 0 \le t \le 1$$

  - $(1-t)$ and t are called weighting functions of P and Q
- Splines are higher order polynomial interpolations between points
  - Like linear interpolation, but with higher order weighting functions allowing better approximations/smoother curves
- One representation - Hermite curves (Interpolating spline):
  - Determined by two control points P and Q, an initial tangent vector v and a final tangent vector w.
  - Satisfies:
    - $\gamma(0) = P$
    - $\gamma(1) = Q$
    - $\gamma'(0) = v$
    - $\gamma'(1) = w$

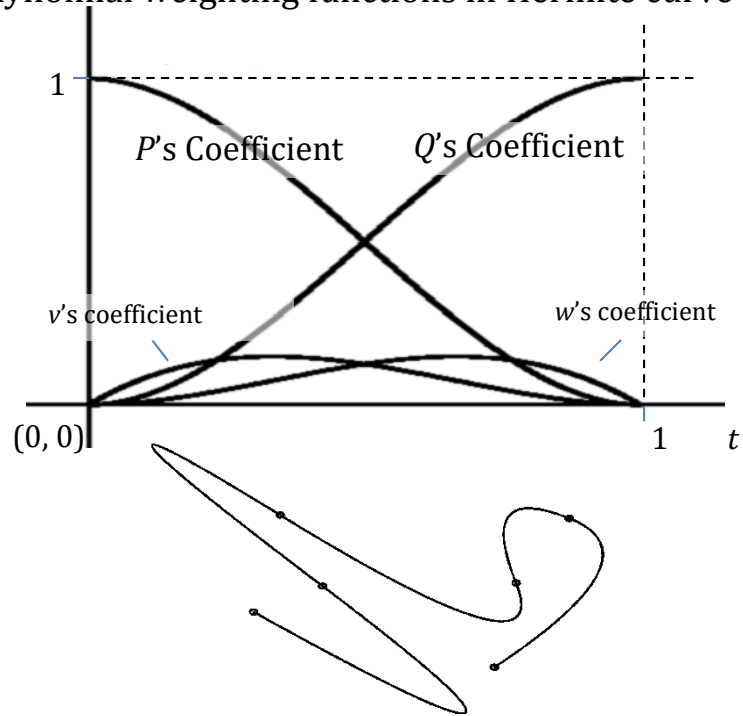
$$\gamma(t) = (2t^3 - 3t^2 + 1)P + (-2t^3 + 3t^2)Q + (t^3 - 2t^2 + t)v + (t^3 - t^2)w$$



Splines (4/5) - Hermite Weighting Explained

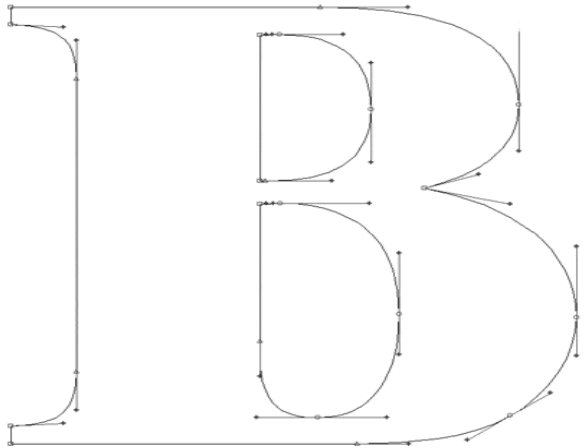
- Polynomial splines have more complex weighting functions than lines
  - Coefficients for *P* and *Q* are now 3rd degree polynomials
- At $t = 0$
  - Coefficients of *P* is 1, all others 0
  - Derivative of coefficient of *v* is 1, derivative of all others is 0
- At $t = 1$
  - Coefficient of *Q* is 1, all others 0
  - Derivative of coefficient of w is 1, derivative of all others 0
- Can be chained together to make more complex curves

Polynomial weighting functions in Hermite curve equation



Splines (5/5) - Bezier Curves

- Bezier representation is similar to Hermite
  - 4 points instead of 2 points and 2 vectors ($P_1 \ldots P_4$)
  - Initial position P_1, tangent vector is $P_2 - P_1$
  - Final position P_4, tangent vector is $P_4 - P_3$
  - This representation allows a spline to be stored as a list of vertices with some global parameters that describe the smoothness and continuity
- Bezier splines are widely used (Adobe, Microsoft) for font definition
- See chapters 23 and 24 for more on splines

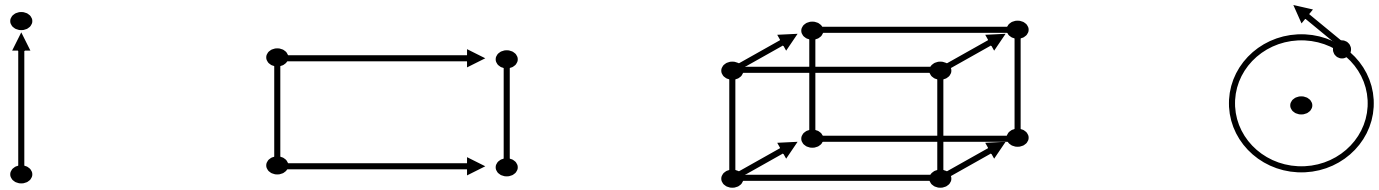


https://www.jasondavies.com/animated-bezier/

Image credit: *http://miphol.com/muse/2008/04/25/Bezier-courbes-anim.gif*

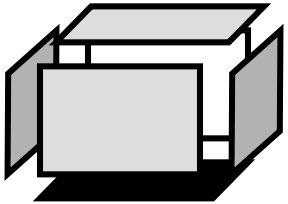
"Vertices in Motion" – Object Definition

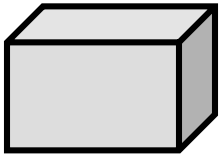
- A line is drawn by tracing a point as it moves (1st dimension added)
- A rectangle is drawn by tracing the vertices of a line as it moves perpendicularly to itself (2nd dimension added)
- A rectangular prism is drawn by tracing the vertices of a rectangle as it moves perpendicularly to itself (3rd dimension)
- A circle is drawn by tracing a point swinging at a fixed distance around a center point.



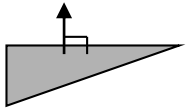
Building 3D Primitives

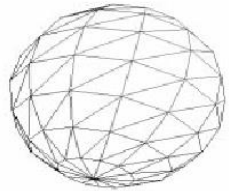
- Made out of 2D and 1D primitives

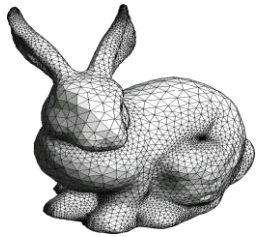




- Triangles are commonly used
- Many triangles used for a single object is a triangular mesh

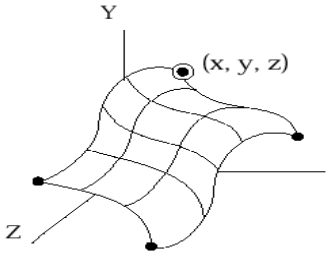






- Splines used to describe boundaries of "patches" – these can be "sewn together" to represent curved surfaces

$$x(s,t) = (1-s)^3 \times (1-t)^3 \times P_{1,1} + (1-s)^3 \times 3t(1-t)^2 \times P_{1,2} + \cdots$$



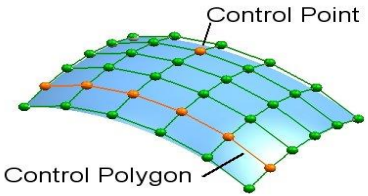


Image credit (Stanford Bunny): http://mech.fsv.cvut.cz/~dr/papers/Habil/img1007.gif

Triangle Meshes

- Most common representation of shape in three dimensions
- All vertices of triangle are guaranteed to lie in one plane (not true for quadrilaterals or other polygons)
- Uniformity makes it easy to perform mesh operations such as subdivision, simplification, transformation etc.
- Many different ways to represent triangular meshes

- Meshes
  en.wikipedia.org/wiki/polygon_mesh
  - Mesh transformation and deformation
  - Procedural generation techniques (upcoming labs on simulating terrain)

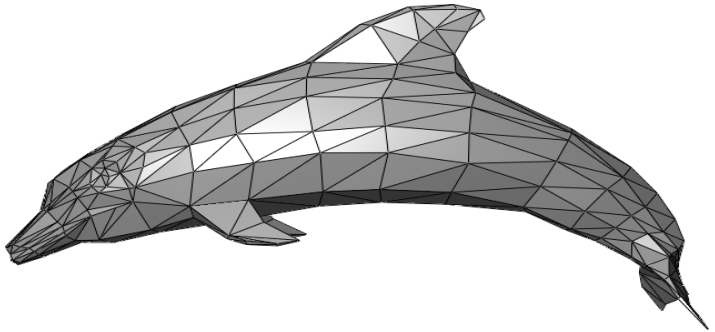


Image credit:
http://upload.wikimedia.org/wikipedia/commons/f/fb/Dolphin_triangle_mesh.png

Triangular Mesh Representation

- Vertex and face tables, analogous to 2D vertex and edge tables
- Each vertex listed once, triangles listed as ordered triplets of indices into the vertex table
  - Edges inferred from triangles
  - It's often useful to store associated faces with vertices (i.e. computing normals: vertex normal as average of surrounding face normals)
- Vertices listed in **counter** clockwise order in face table.
  - No longer just because of convention. CCW order differentiates front and back of face

Vertex List

v0	0, 0, 0	f0 f1 f12 f15 f7
v1	1, 0, 0	f2 f3 f13 f12 f1
v2	1, 1, 0	f4 f5 f14 f13 f3
v3	0, 1, 0	f6 f7 f15 f14 f5
v4	0, 0, 1	f6 f7 f0 f8 f11
v5	1, 0, 1	f0 f1 f2 f9 f8
v6	1, 1, 1	f2 f3 f4 f10 f9
v7	0, 1, 1	f4 f5 f6 f11 f10
v8	.5, .5, 1	f8 f9 f10 f11
v9	.5, .5, 0	f12 f13 f14 f15

Face List

f0	v0 v5 v4
f1	v0 v5 v1
f2	v1 v6 v5
f3	v1 v2 v6
f4	v2 v7 v6
f5	v2 v3 v7
f6	v3 v4 v7
f7	v3 v0 v4
f8	v8 v4 v5
f9	v8 v5 v6
f10	v8 v6 v7
f11	v8 v7 v4
f12	v9 v1 v0
f13	v9 v2 v1
f14	v9 v3 v2
f15	v9 v0 v3

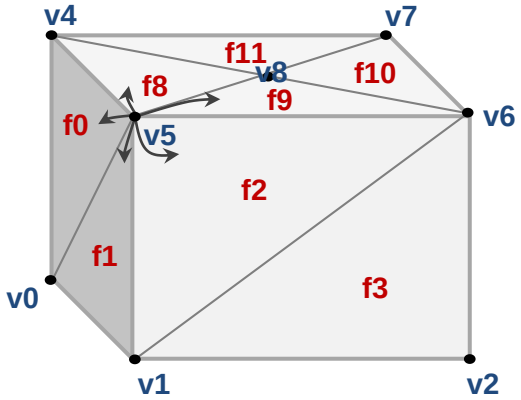


Diagram licensed under Creative Commons Attribution license. Created by Ben Herila based on http://upload.wikimedia.org/wikipedia/en/thumb/2/2d/Mesh_fv.jpg/500px-Mesh_fv.jpg