de Casteljau

- de Casteljau algorithm evaluates a point on a Bezier curve by scaffolding lerps.
- Blossoming renames the control and intermediate points, like $p_{12}$, using a polar form, like $p(0, t, 1)$. 
• de Casteljau algorithm evaluates a point on a Bezier curve by scaffolding lerps
• Blossoming renames the control and intermediate points, like $p_{12}$, using a polar form, like $p(0,t,1)$
de Casteljau

- de Casteljau algorithm evaluates a point on a Bezier curve by scaffolding lerps
- Blossoming renames the control and intermediate points, like $p_{12}$, using a polar form, like $p(0,t,1)$
de Casteljau

- de Casteljau algorithm evaluates a point on a Bezier curve by scaffolding lerps
- Blossoming renames the control and intermediate points, like $p_{12}$, using a polar form, like $p(0,t,1)$
de Casteljau

- de Casteljau algorithm evaluates a point on a Bezier curve by scaffolding lerps.
- Blossoming renames the control and intermediate points, like $p_{12}$, using a polar form, like $p(0,t,1)$. 

![Bezier Curve Diagram]
Blossoming Rules

1. # of parameters = degree

Cubic: $p(\_,\_,\_)$
1. # of parameters = degree

   Cubic: \( p(\_,\_,\_) \)

2. Order doesn't matter

   \( p(a,b,c) = p(b,a,c) \)
Blossoming Rules

1. # of parameters = degree
   Cubic: \( p(\_,\_,\_) \)

2. Order doesn’t matter
   \( p(a,b,c) = p(b,a,c) \)

3. Setting up the board
   \( p(0,0,0), p(0,0,1), p(0,1,1), p(1,1,1) \)
Blossoming Rules

1. # of parameters = degree
   Cubic: $p(\_,\_,\_)$

2. Order doesn’t matter
   $p(a,b,c) = p(b,a,c)$

3. Setting up the board
   $p(0,0,0), p(0,0,1), p(0,1,1), p(1,1,1)$

4. Winning the game
   $p(t,t,t)$
Placing Blossoms

Find two blossoms whose values match except for one

Rewrite the blossoms in a consistent order and draw a line segment between them

Interpolate the differing blossom value at interpolated points along the line segment
Evaluation

\[ p(t) = p(t, t, t) \]
Evaluation

\[ p(t) = p(t,t,t) \]
\[ = (1-t) p(t,t,0) + t p(t,t,1) \]
Evaluation

\[ p(t) = p(t,t,t) \]
\[ = (1-t) \ p(t,t,0) + t \ p(t,t,1) \]
\[ = (1-t)[(1-t) \ p(t,0,0) + t \ p(t,0,1)] \]
\[ + t \ [(1-t) \ p(t,0,1) + t \ p(t,1,1)] \]
Evaluation

\[ p(t) = p(t,t,t) \]

\[ = (1-t) \ p(t,t,0) + t \ p(t,t,1) \]

\[ = (1-t)[(1-t) \ p(t,0,0) + t \ p(t,0,1)] \]

\[ + t [(1-t) \ p(t,0,1) + t \ p(t,1,1)] \]

\[ = (1-t)^2 \ p(t,0,0) + 2 (1-t) t \ p(t,0,1) + t^2 \ p(t,1,1) \]
Evaluation

\[ p(t) = p(t, t, t) \]
\[ = (1-t) p(t, t, 0) + t p(t, t, 1) \]
\[ = (1-t) [(1-t) p(t, 0, 0) + t p(t, 0, 1)] \]
\[ + t [(1-t) p(t, 0, 1) + t p(t, 1, 1)] \]
\[ = (1-t)^2 p(t, 0, 0) + 2 (1-t) t p(t, 0, 1) + t^2 p(t, 1, 1) \]
\[ = (1-t)^2 [(1-t)p(0, 0, 0)+tp(1, 0, 0)]+2(1-t)t[(1-t)p(0, 0, 1)+tp(1, 0, 1)]+t^2[(1-t)p(0, 1, 1)+tp(1, 1, 1)] \]
**Evaluation**

\[ \mathbf{p}(t) = \mathbf{p}(t,t,t) \]
\[ = (1-t) \mathbf{p}(t,t,0) + t \mathbf{p}(t,t,1) \]
\[ = (1-t)[(1-t) \mathbf{p}(t,0,0) + t \mathbf{p}(t,0,1)] \\
+ [t [(1-t) \mathbf{p}(t,0,1) + t \mathbf{p}(t,1,1)] \\
= (1-t)^2 \mathbf{p}(t,0,0) + 2 (1-t) t \mathbf{p}(t,0,1) + t^2 \mathbf{p}(t,1,1) \]
\[ = (1-t)^2[(1-t)\mathbf{p}(0,0,0)+t\mathbf{p}(1,0,0)]+2(1-t)t[(1-t)\mathbf{p}(0,0,1)+t\mathbf{p}(1,0,1)]+t^2[(1-t)\mathbf{p}(0,1,1)+t\mathbf{p}(1,1,1)] \]
\[ = (1-t)^3 \mathbf{p}(0,0,0) + 3 (1-t)^2 t \mathbf{p}(0,0,1) + 3 (1-t) t^2 \mathbf{p}(0,1,1) + t^3 \mathbf{p}(1,1,1) \]