

# Nile Visualization

Bret Victor (2013)

An interactive visualization of the Gezira/Nile software by Dan Amelang, a graphics renderer (Gezira) written in a domain specific language (Nile).

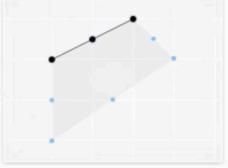
This project points to ways in which software description, data, and behavior, and be represented and made tangible as deeply and vivaciously interlinked representations.

**initial input**



5 Points

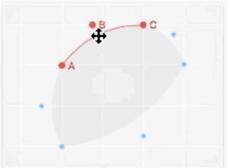
**MakePolygon ()**



processed 5 Points  
output 4 Beziers

```
MakePolygon () : Point >> Bezier
p:Point = 0
first = true
∀ p'
  first' = false
  if ¬first
    >> (p, p ~ p', p')
```

**RoundPolygon ()**

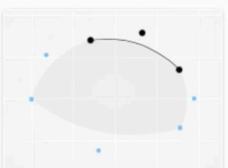


processed 4 Beziers  
output 4 Beziers

```
RoundPolygon () : Bezier >> Bezier
∀ (A, B, C)
  n = (A ⊥ C) / 4
  >> (A, B + n, C)
```

A: (0.2), B: (0.75, 3), C: (2, 3)

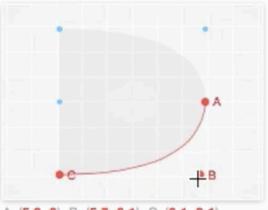
**TransformBeziers ()**



processed 4 Beziers  
output 4 Beziers

```
TransformBeziers (M:Matrix) : Bezier >> Bezier
∀ (A, B, C)
  >> (MA, MB, MC)
```

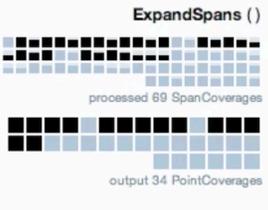
**initial input**



3 Beziers

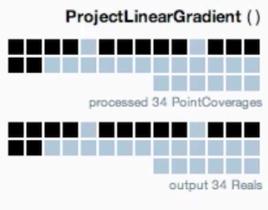
A: (5.9, 3), B: (5.7, 0.1), C: (0.1, 0.1)

**ExpandSpans ()**



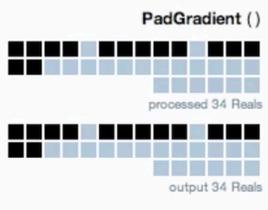
processed 69 SpanCoverages  
output 34 PointCoverages

**ProjectLinearGradient ()**



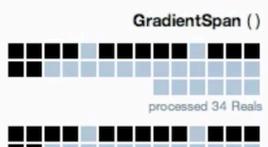
processed 34 PointCoverages  
output 34 Reals

**PadGradient ()**



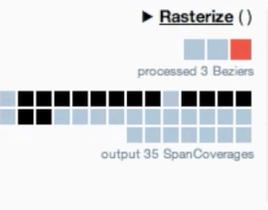
processed 34 Reals  
output 34 Reals

**GradientSpan ()**



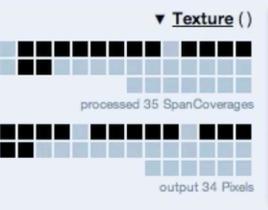
processed 34 Reals  
output 34 Reals

**Rasterize ()**



processed 3 Beziers  
output 35 SpanCoverages

**Texture ()**



processed 35 SpanCoverages  
output 34 Pixels

```
ExpandSpans ()
  ∀ (x, y, c,
    if c > 1
    >>
    <<
```

```
ProjectLinearGr
  v = B - A
  Δs = v / (
  s00 = A · Δ
  ∀ (P,_)
  >> P ·
```

```
PadGradient ()
  ∀ s
  >> 0 ▷
```

```
GradientSpan (A
  ∀ s
  >> sA +
```