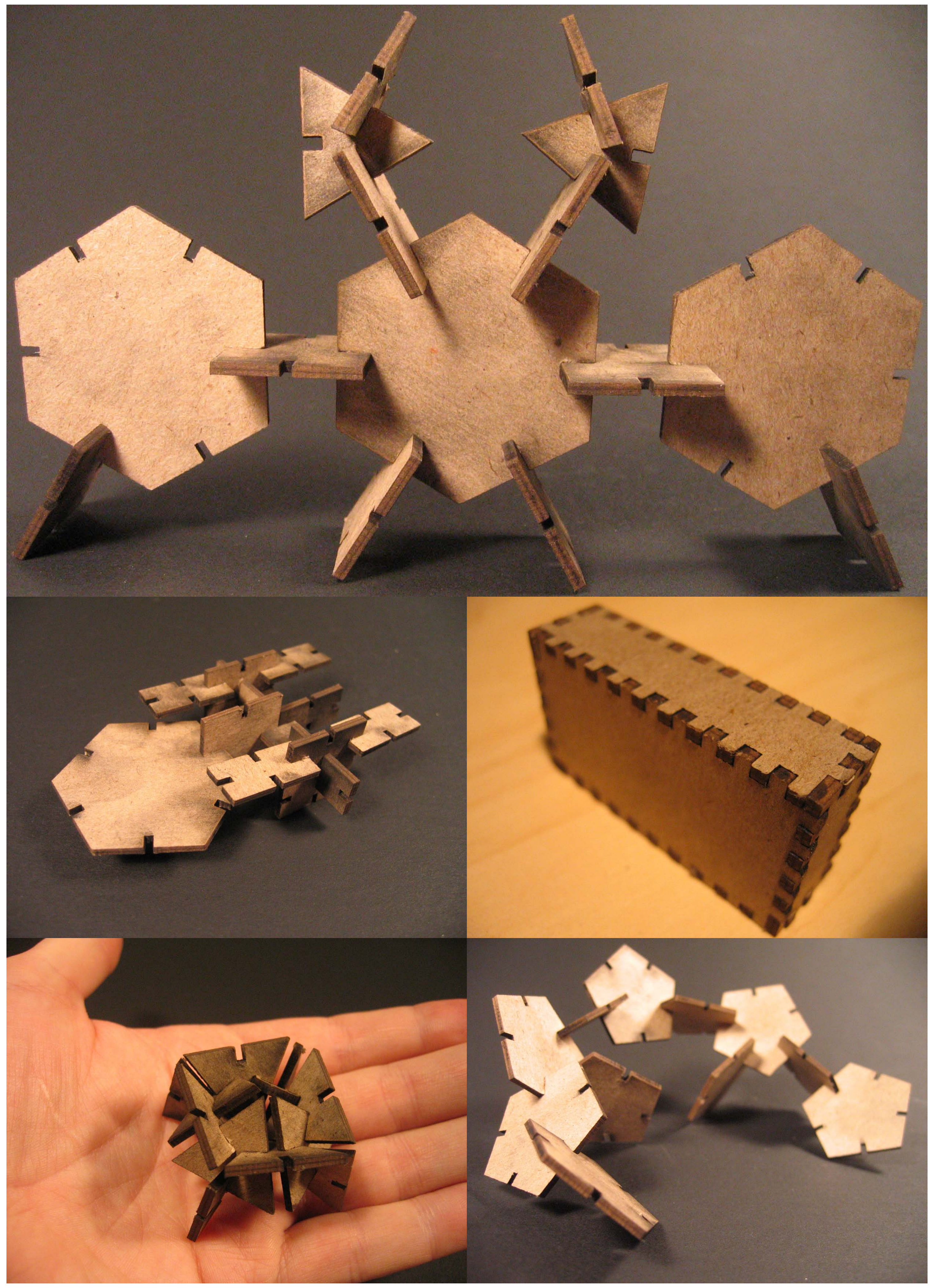
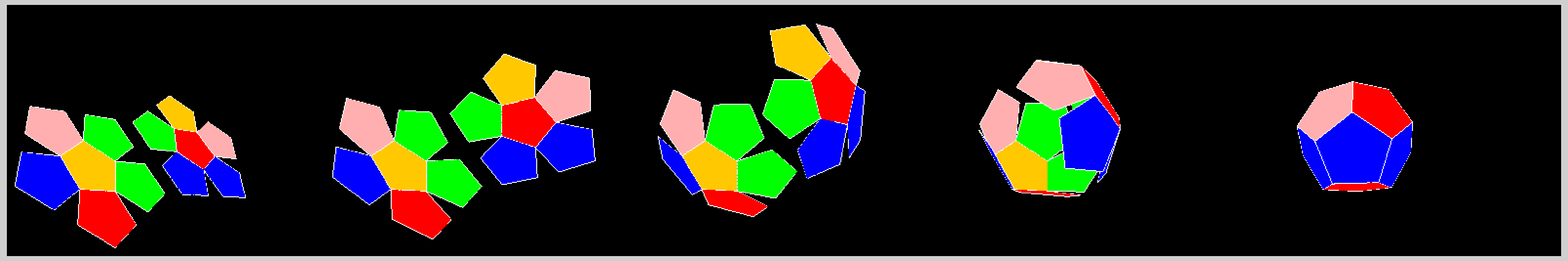
# FlatCAD kits by code

Gabe Johnson, Mark D Gross
johnsogg@cmu.edu



The FlatCAD system lets you create your own construction kit by coding in the LOGO-like FlatLang language. No longer must construction kits be merely a product designed by someone else: if you can write a simple FlatLang program, you can design a kit.

You can also unfold solid geometries using FlatCAD. Here, we unroll a dodecahedron by changing the dihedral angles between the pentagon faces.

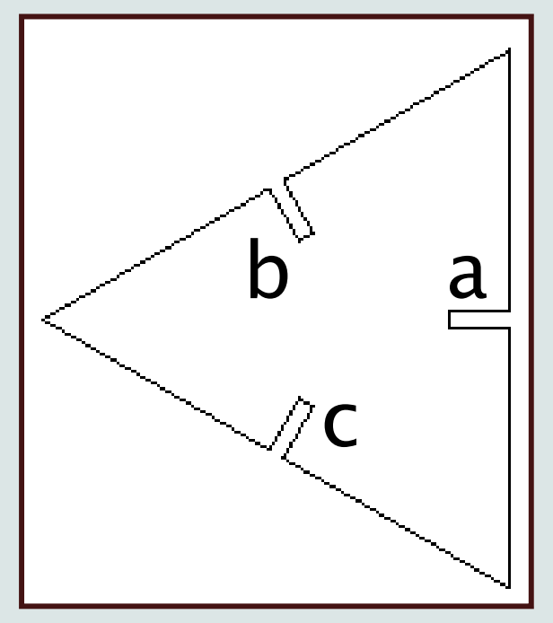


## FlatLang Example

```
; triangle.fl
;
; This is a notched triangle with named points
; a, b, and c at the bottom of the notches

; 'notch' creates one side of a polygon, leaving a
; parameterized slot in the middle.
define notch(len, notchDepth, notchWidth, name)
  fAmt = (len / 2) - (notchWidth / 2)
  forward (fAmt)
  left(90)
  forward (notchDepth)
  right(90)
  forward (notchWidth / 2)
  mark(name)
  forward (notchWidth / 2)
  right(90)
  forward (notchDepth)
  left(90)
  forward (fAmt)
done

define triangle(len, notchDepth, notchWidth)
  angle = 360 / 3
  notch(len, notchDepth, notchWidth, "a")
  left(angle)
  notch(len, notchDepth, notchWidth, "b")
  left(angle)
  notch(len, notchDepth, notchWidth, "c")
  left(angle)
done
```

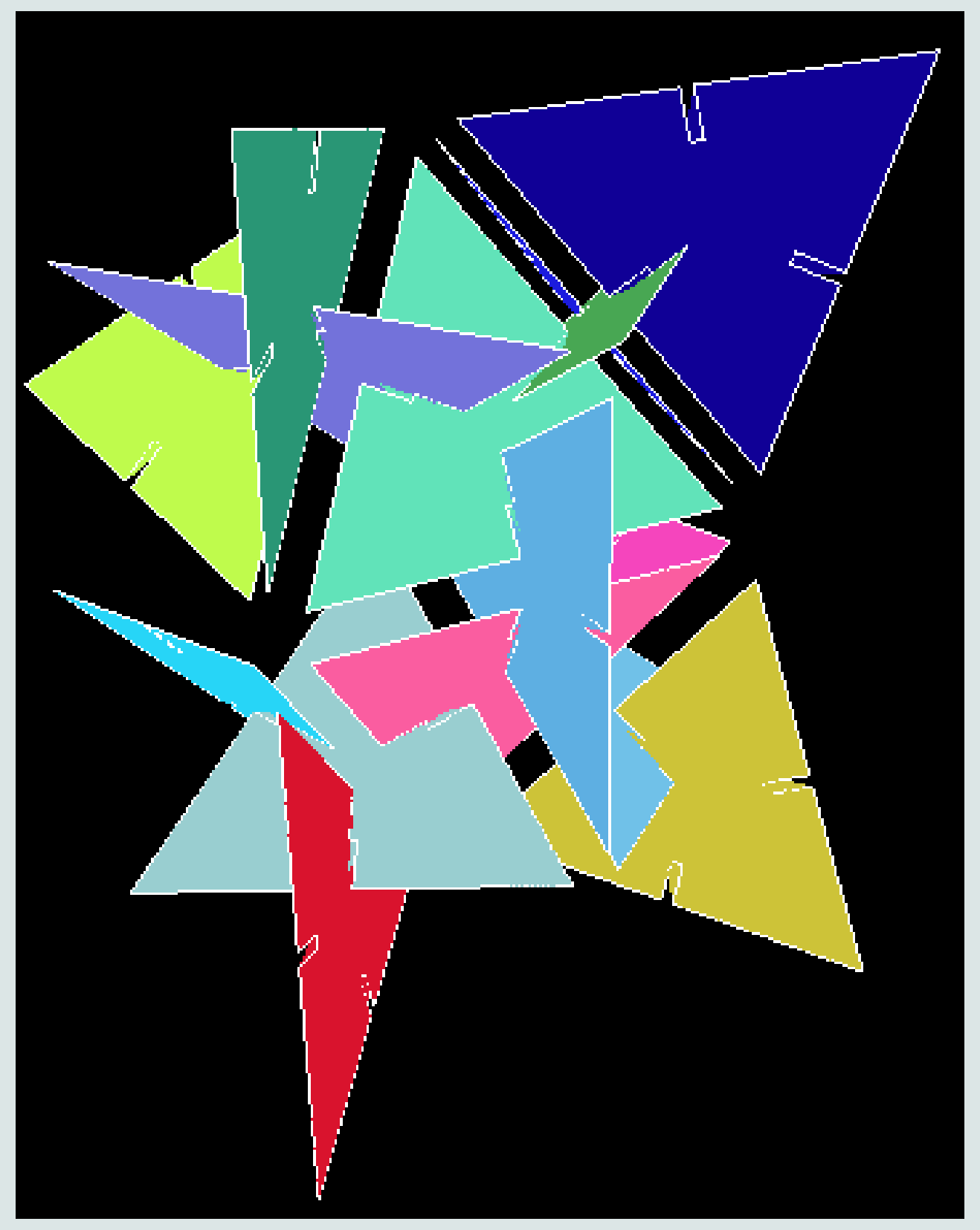


```
load(”kit/triangle.fl”)
showPolygons(true)

shape(”tri”)
  triangle(2.7, 0.3, 0.08)
done

define go(s, ttl)
  draw(s, “a”)
  from(”b”, “c”)
    pitch(90)
    left(180)
    if(ttl > 0)
      go(s, ttl - 1)
    done
  done
done

go(”tri”, 3)
```



This work is part of the Codelab’s Computationally Enhanced Construction Kits and Craft Grant.

Computational
Design
Lab

Funded by NSF Grant ITR-0326054