# Challenges on the Construction of Topological Interlocking Configurations on Solids and Meshes

**Hello!**

# I am Andres Bejarano

Interested in Computer Graphics and Geometry. Currently working on interlocking configurations, advised by professor **Christoph Hoffmann**.

Content

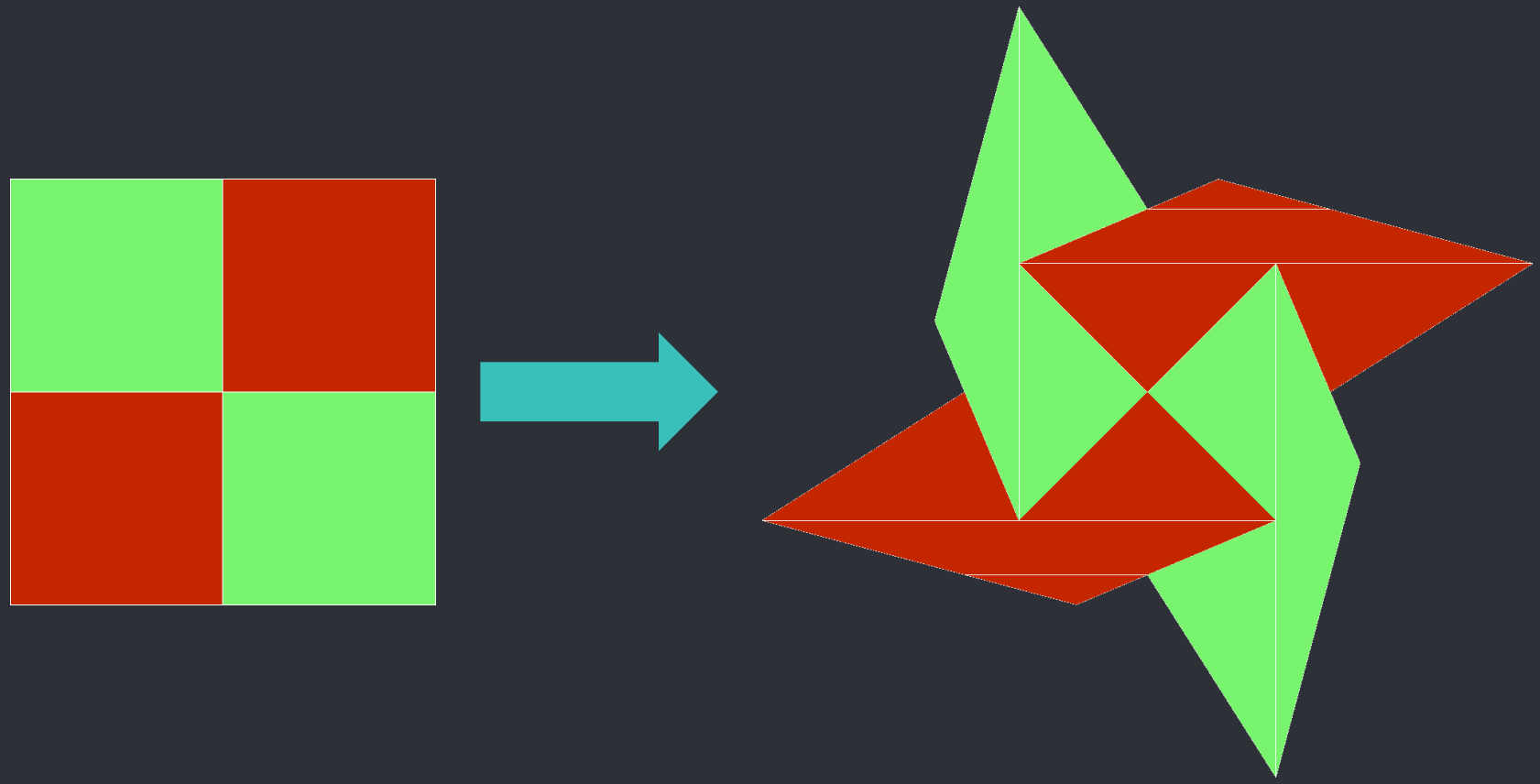
- Concept
- Challenges
- TIGER (yep, we have one)
- Examples

1

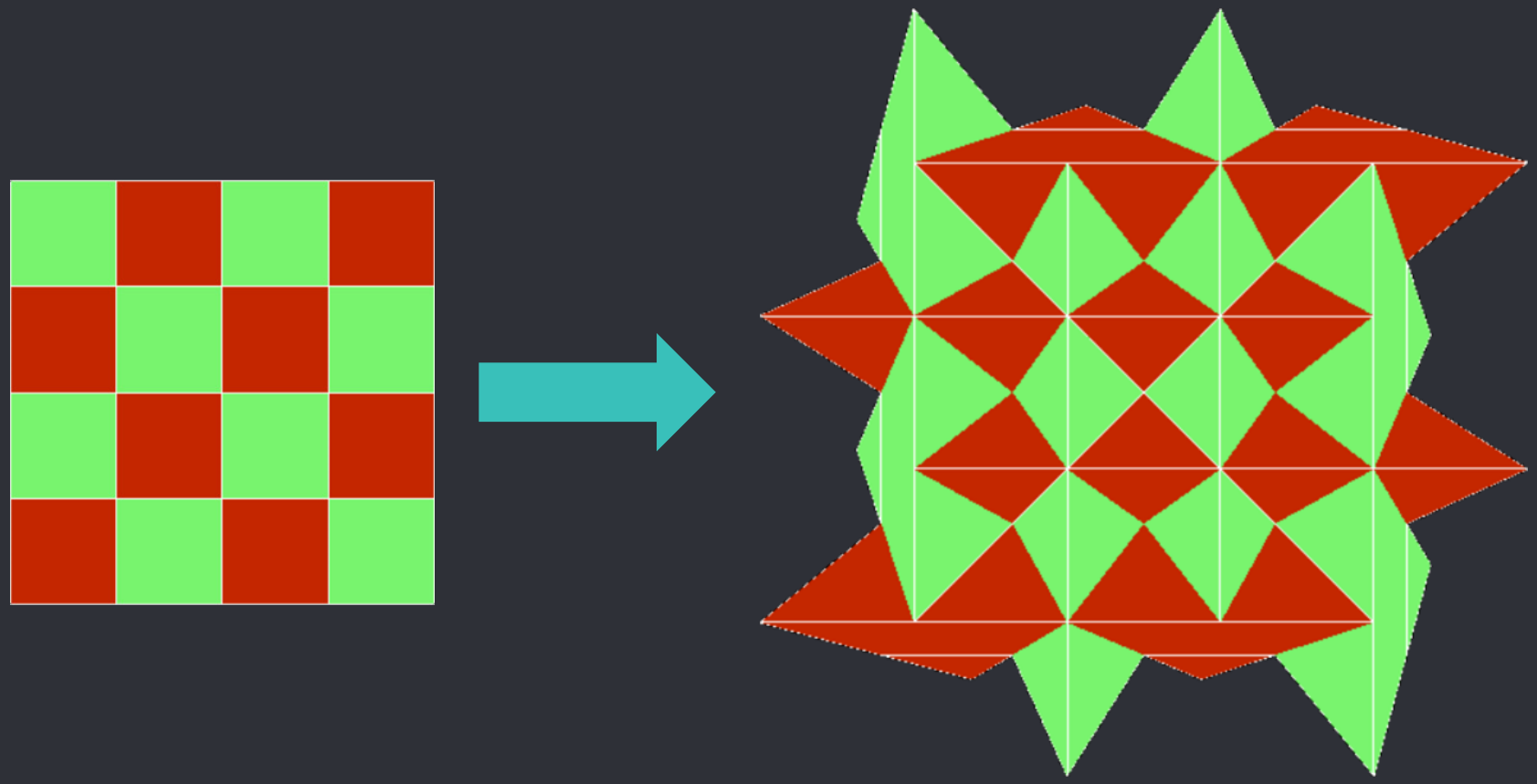
# CONCEPT

What is this all about

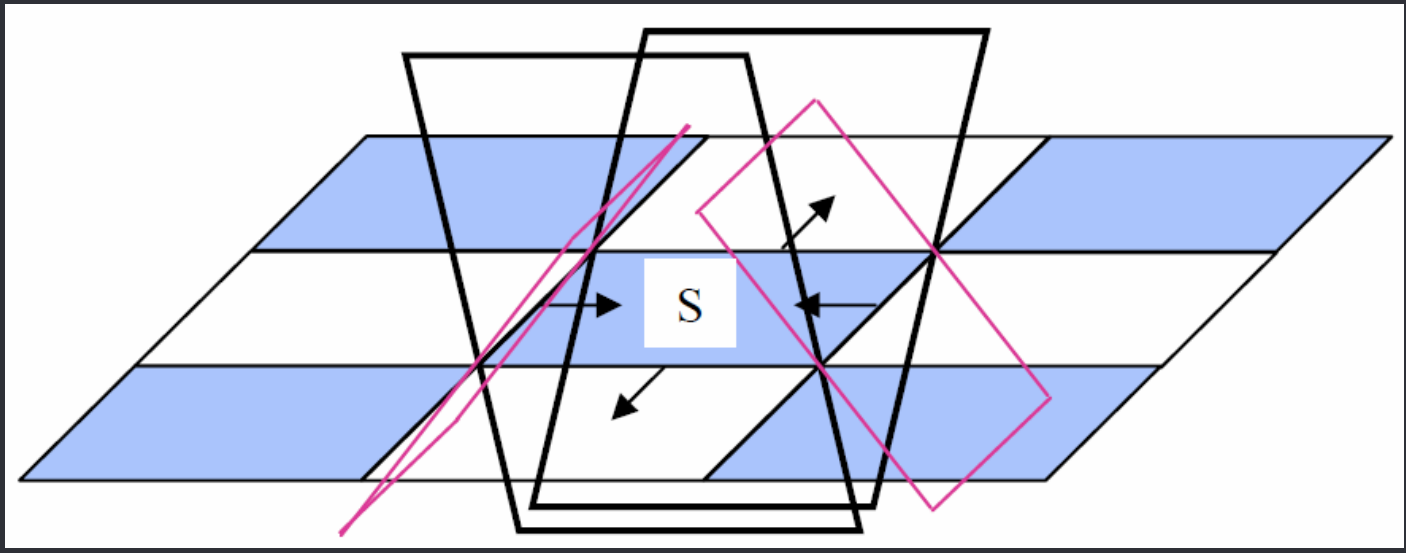
Can interlocking occur using only convex shapes?



- Repeat process until infinity... or some physical border



# Generating a TIC on the plane




It requires a chessboard, some directions on each square based on its color, and four overlapping planes per square.

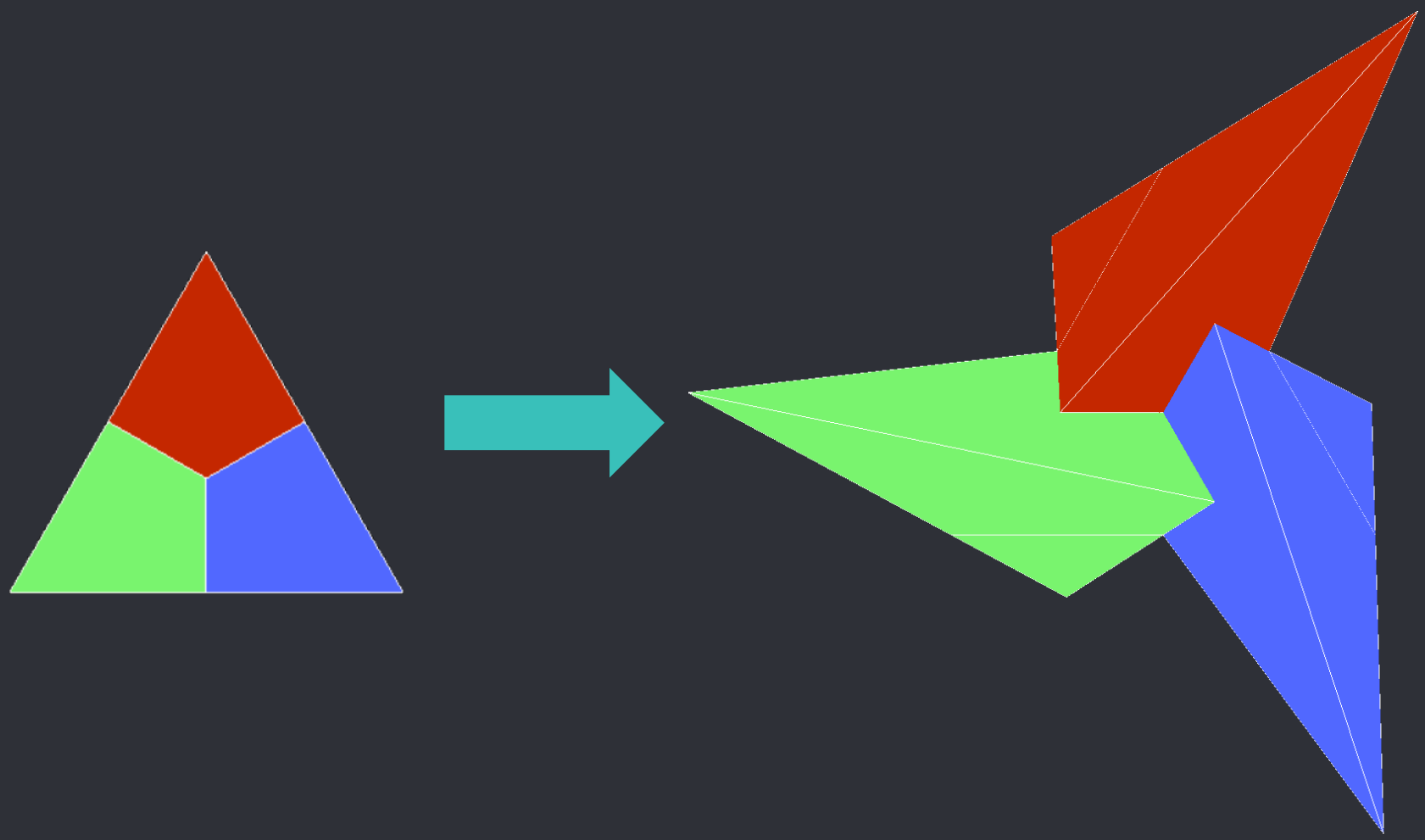
(Kanel-Belov, 2008)

“

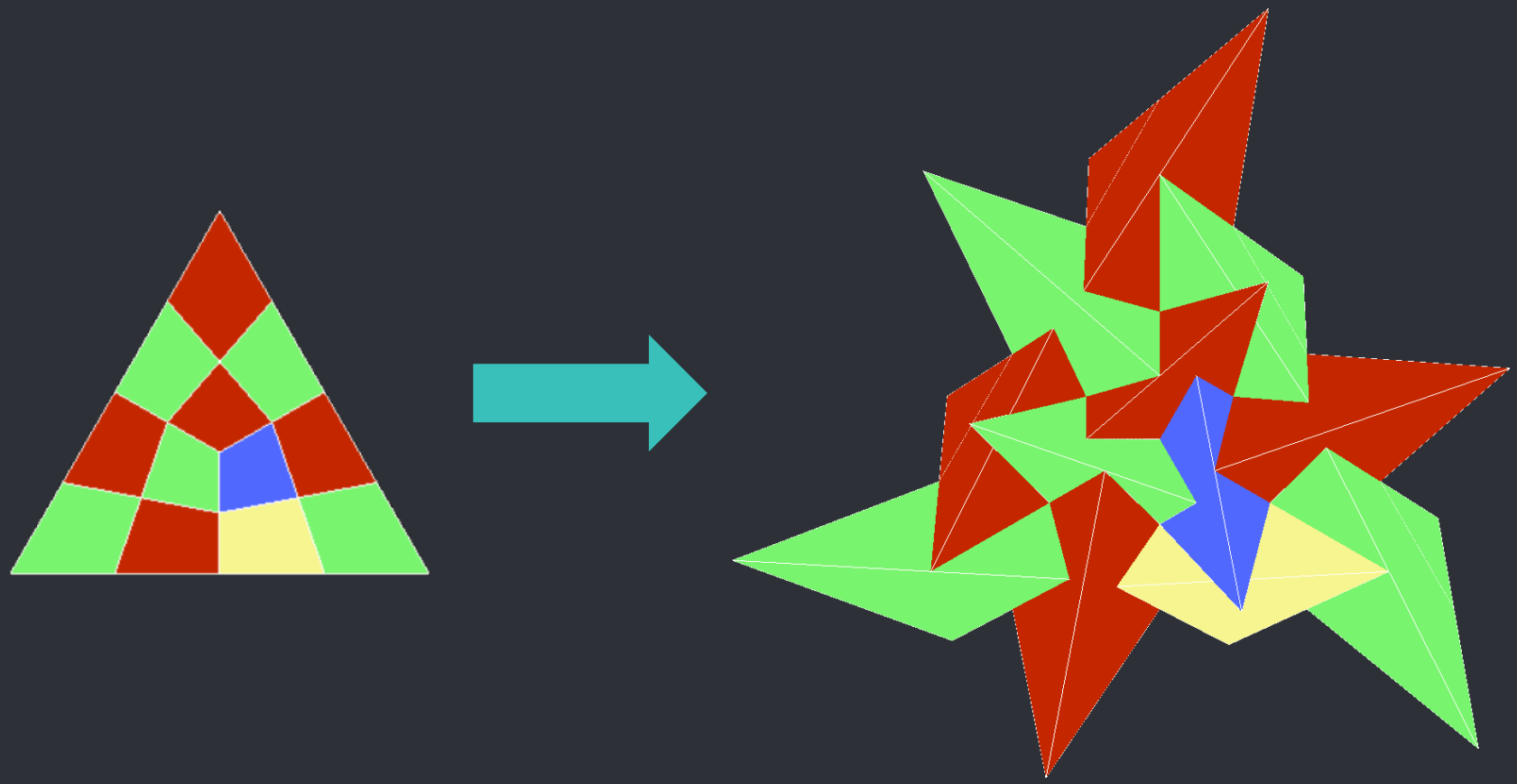
***Interlocking*** *is achieved if in every row of elements one can identify two sections normal to the assembly plane such that while one section ensures* ***kinematic constraint*** *in one direction (normal to the assembly plane), the other section provides the same elements with* ***constraint in the opposite direction****."*

*(Kanel-Belov et al., 2008)*

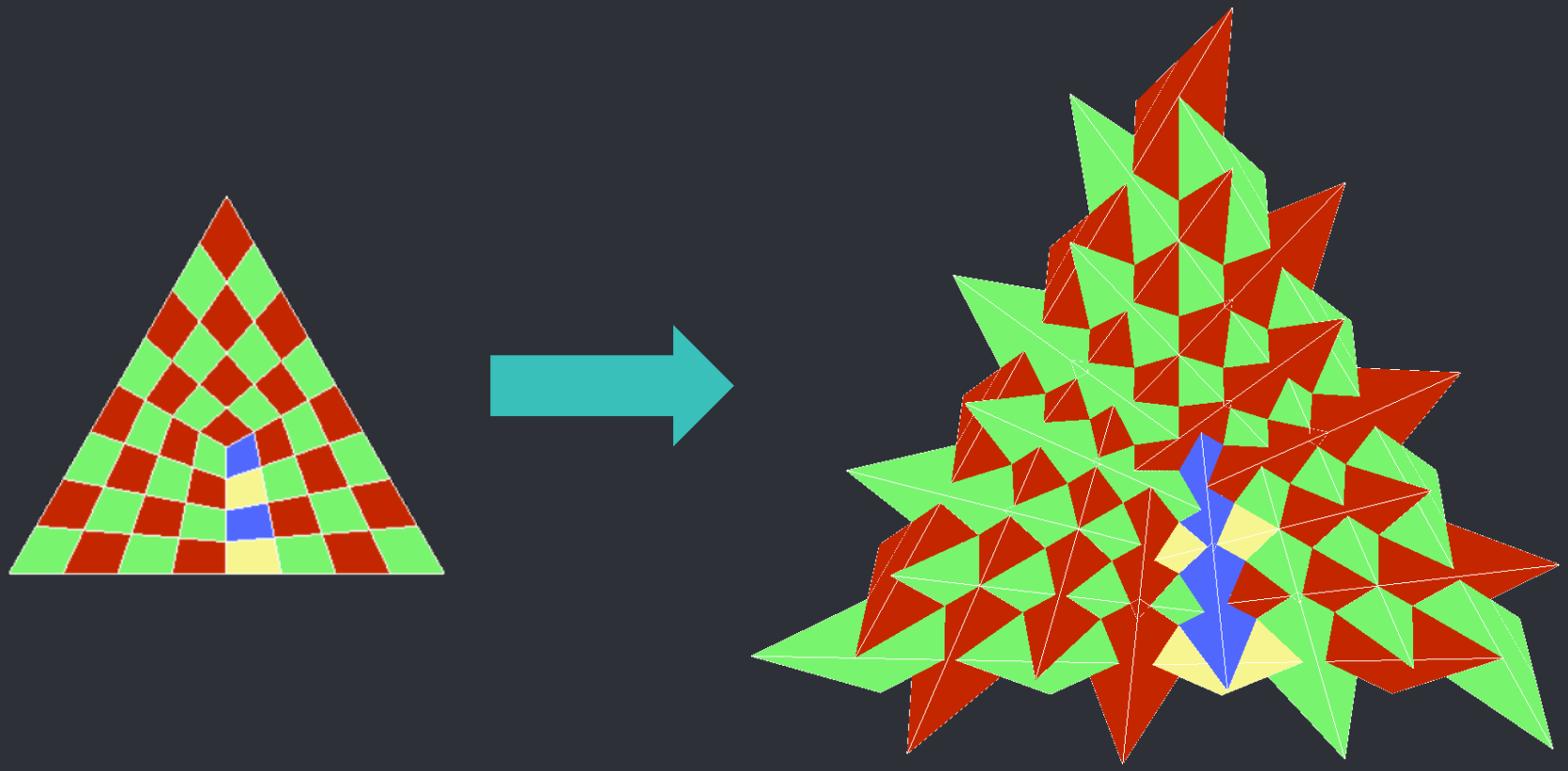
- In theory, any quadrilateral should work



- Looking good!

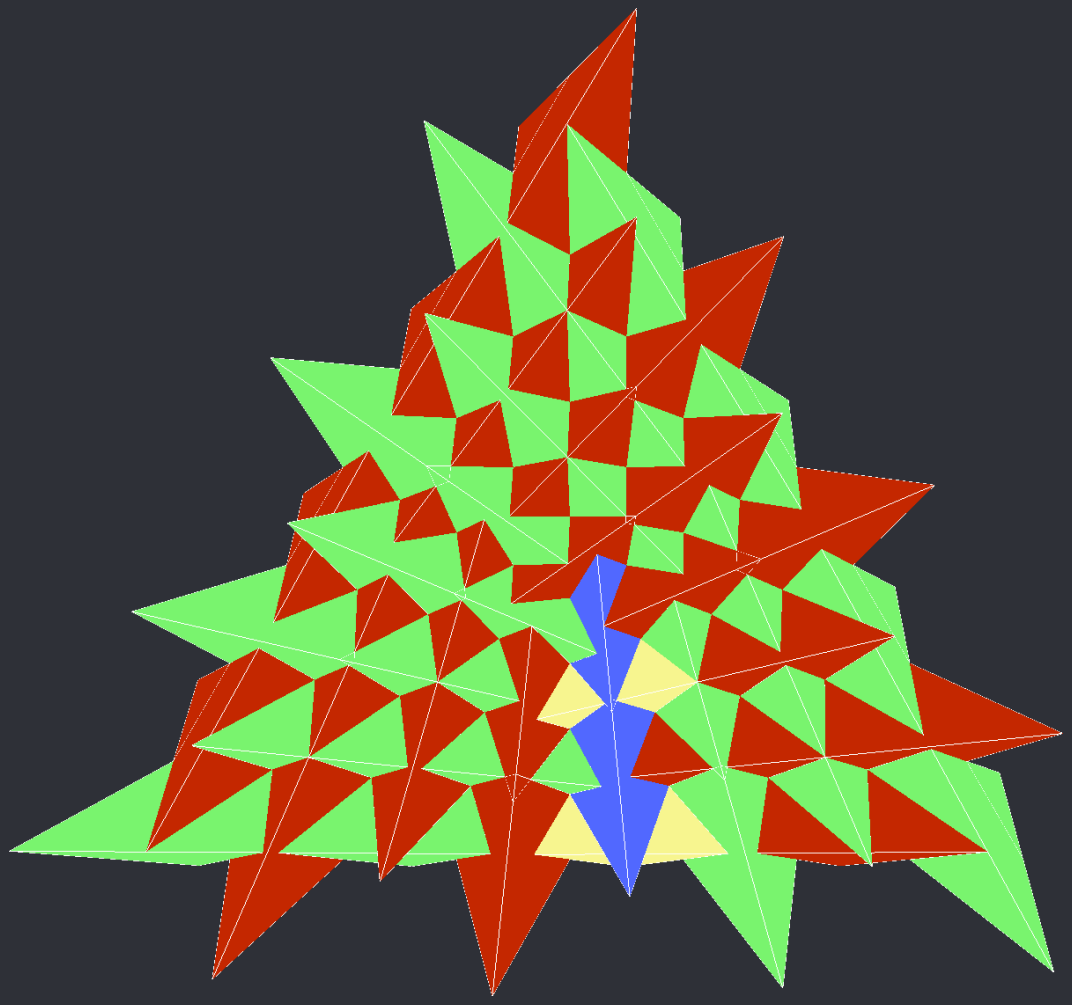


## Houston, we have a problem



Losing symmetry and nonlinearity generate

**Overlappings**

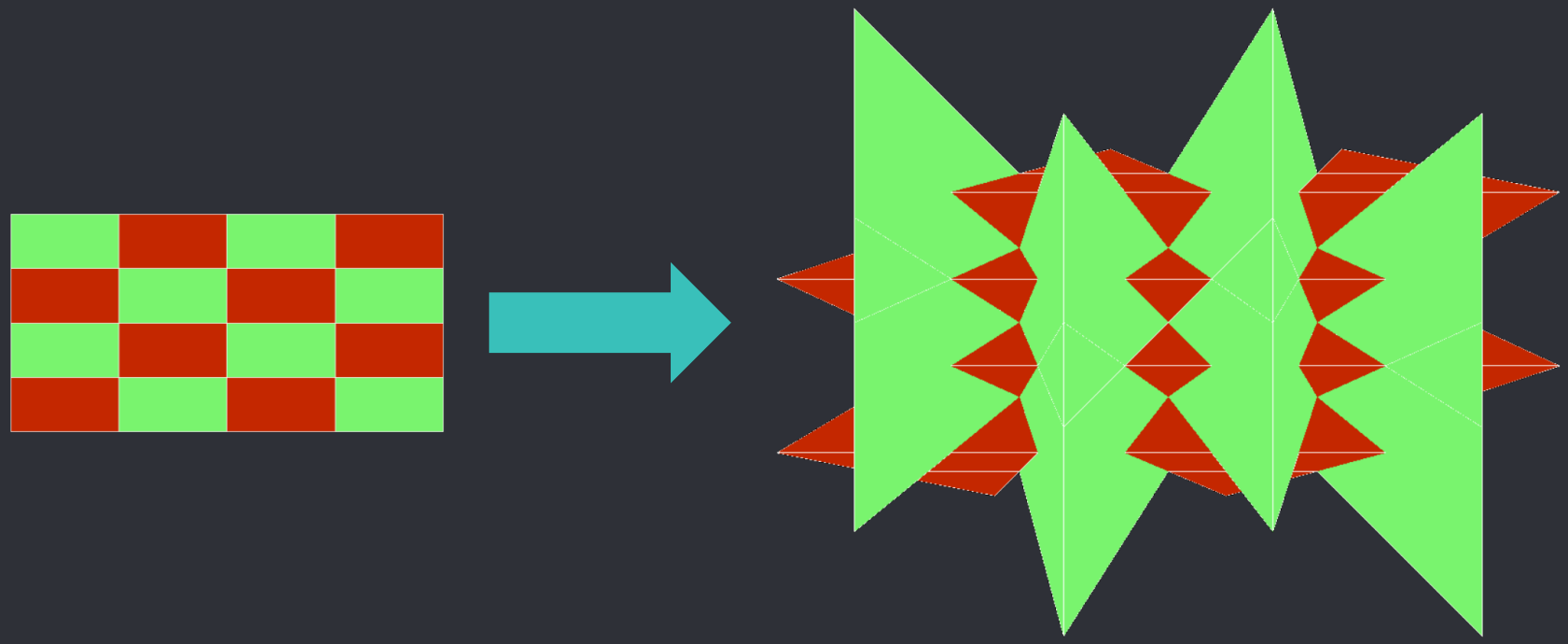


2

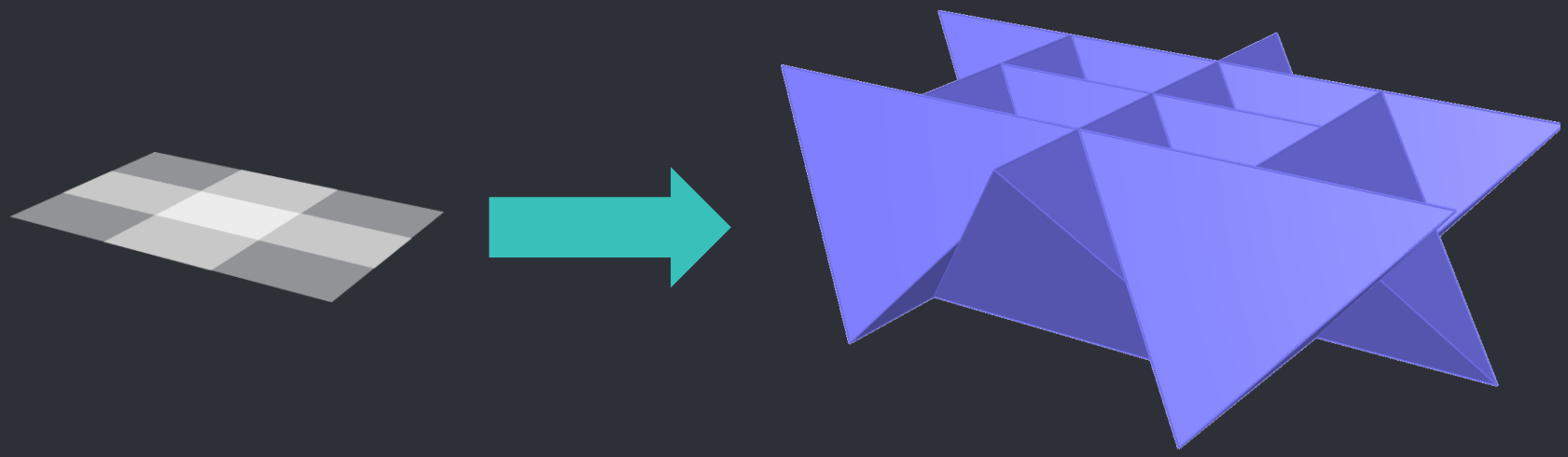
# CHALLENGES

A little bit of Math

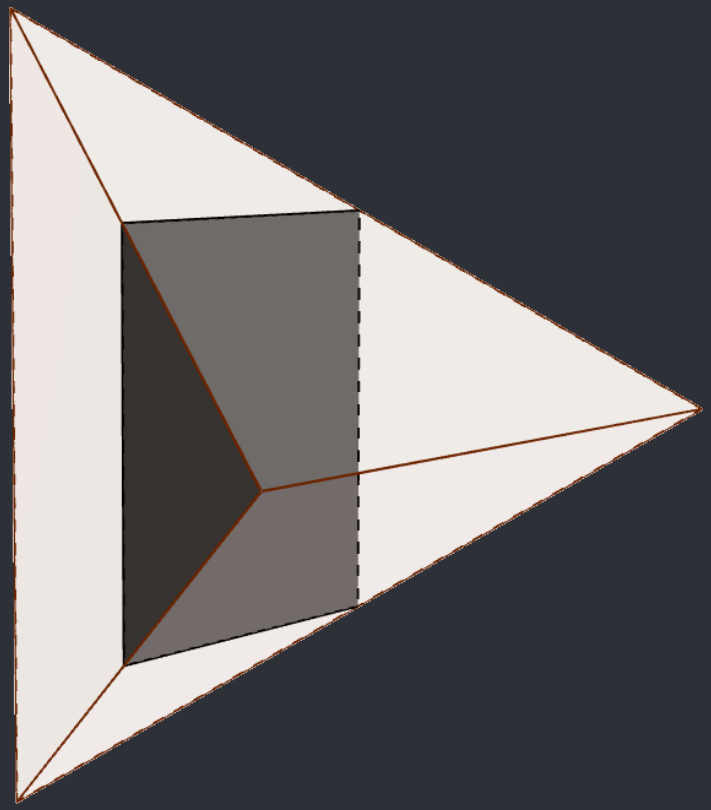
- Even an innocent rectangle suffers overlappings



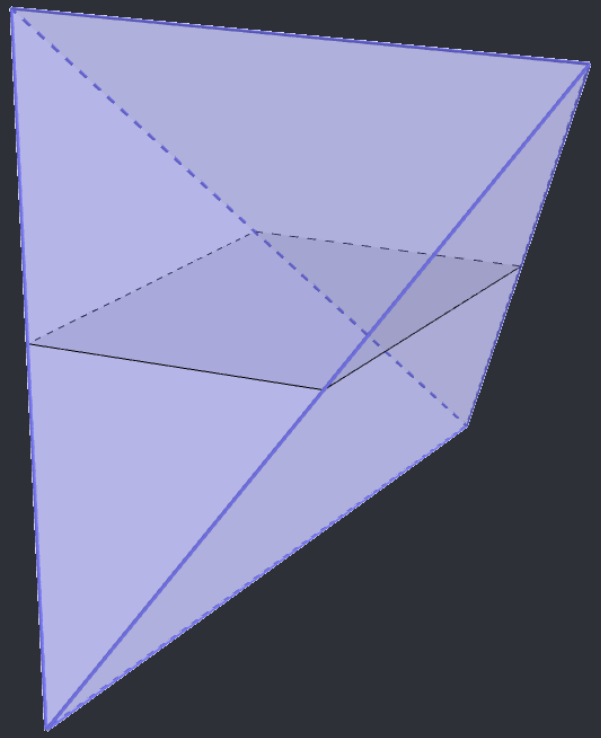
- Having two different angles solves the problem



Problem: Find angles such that the quadrilateral is exactly the midsection of the generated tetrahedron



Square: 60°



Rectangle: 60°, 49.1°

# Problem Formulation (angle based)

Given a quadrilateral $Q = \{V, E\}, V = \{v_1, v_2, v_3, v_4\}, E = \{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_1)\}$ in space (vertices assumed to be coplanar), a direction value $dir(E_i) \in \{1, -1\}, i = 1,2,3,4$ assigned to each edge in $Q$ such that not two consecutive edges have the same direction value, and rotation angles $\theta_i, i = 1,2,3,4$ assigned to each edge. A tetrahedron whose midsection is the given quadrilateral is generated as follows:

Let rotation axes $R_i$ be the direction vectors for each edge $E_i$. Set the normal $NE_i$ associated to each edge be the normal $N$ of the quadrilateral. The normal for each plane incident to each edge is given by expression $NP_i = AxisAngleRotation(R_i, dir(E_i) * \theta_i, NE_i)$ where

$$AxisAngleRotation(K, \theta, V) = Vcos(\theta) + (K \times V)sin(\theta) + K(K \cdot V)(1 - cos(\theta))$$

is the rotation of vector $V$ around vector axis $K$ for an angle $\theta$. This is also known as Rodrigue's Rotation.

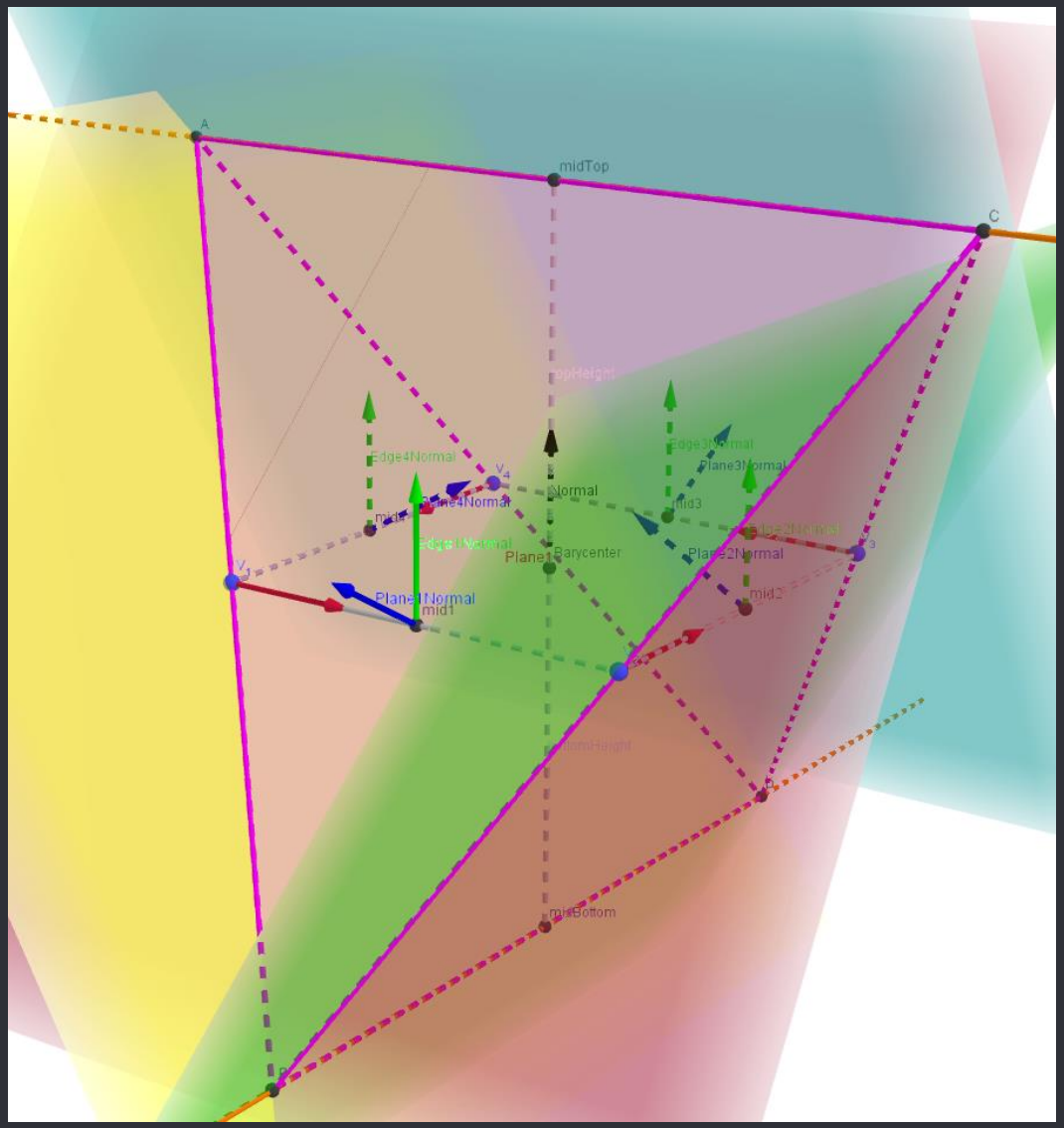
Let $mid_i$ the midpoint for each edge $E_i$. The plane incident to each edge is $P_i = NP_i \cdot (p - mid_i) = 0$. The vertices of the tetrahedron $T$ are given by the intersection of three consecutive planes, that is:

$$A = Intersection(P_1, P_2, P_3)$$
$$B = Intersection(P_2, P_3, P_4)$$
$$C = Intersection(P_3, P_4, P_1)$$
$$D = Intersection(P_4, P_1, P_2)$$

Let $midTop$ and $midBottom$ be the midpoints for the top and bottom edges of $T$. Then, $Q$ is the midsection of $T$ if $dist(center(Q), midTop) = dist(center(Q), midBottom)$.

Problem Formulation

**Interactive**



3

# TIGER

So, we have a TIGER...

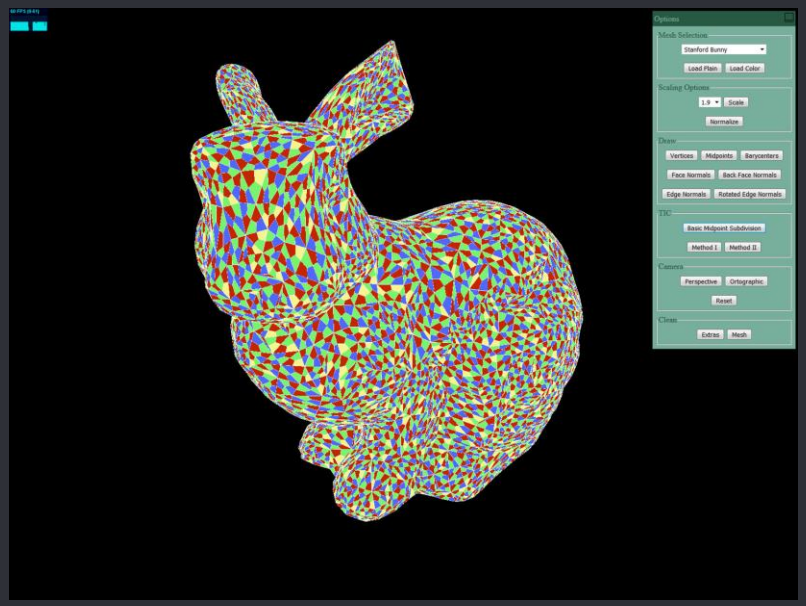
# TIGER

## Topological Interlocking GEneratoR

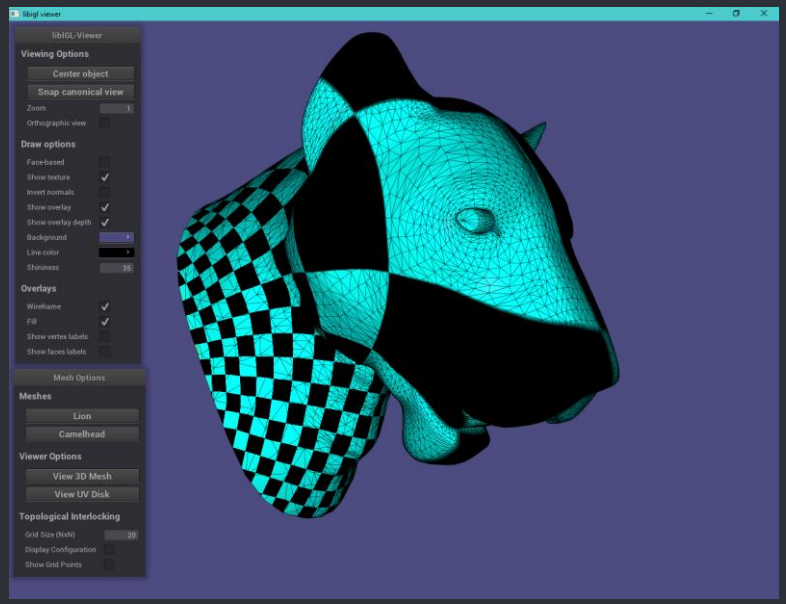
Logo by **HundredMelanie**
https://hundredmelanie.deviantart.com/art/Triangle-Tiger-Vector-470750954

## Versions

# Web (master)




# Desktop (branch)



Technical Aspects

**Web (master)**

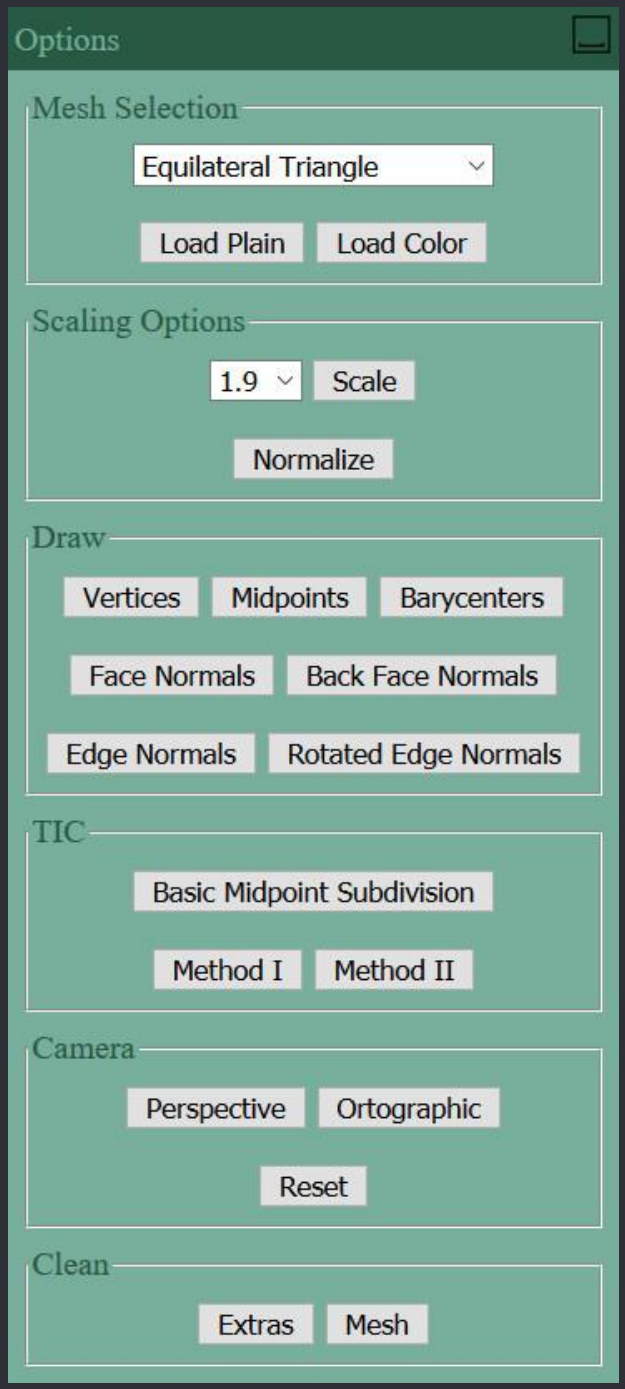
Coded in JavaScript. Uses Three.js and math.js.

**Desktop (branch)**

Coded in C++. Uses libigl (includes Eigen + NanoGUI).

Simple Menu

**Complex Results**



4

# EXAMPLES

One TIGER to rule them all

Let’s see some

**Demos**

**Thanks!**

# ANY QUESTIONS?

You can find me at

@andresbeja87

https://www.cs.purdue.edu/homes/abejara/

# CREDITS

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash