

Shapecraft2 Hackathon Submission

Katachi Gen

Katachi (形) + Gen (現) = Shape Revealed

An on-chain NFT collection of algorithmically generated 3D Origami forms based on your wallet's ShapeL2 participation.

Katachi (形) simply means shape or form in Japanese.

Gen (現) means to appear, to manifest, or to become present.

Together, **Katachi Gen** (形現) can be translated to: “Shape Revealed” or “Shape Manifest”, evoking the transformation from flat pattern to dimensional object.

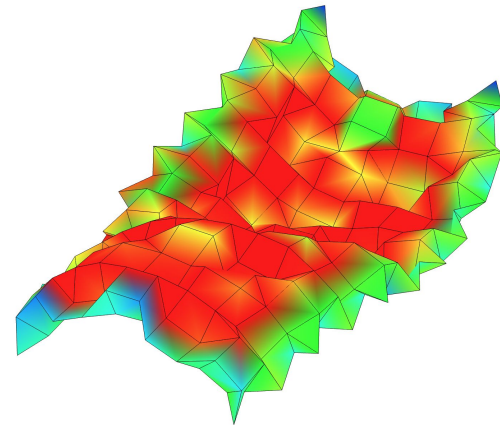
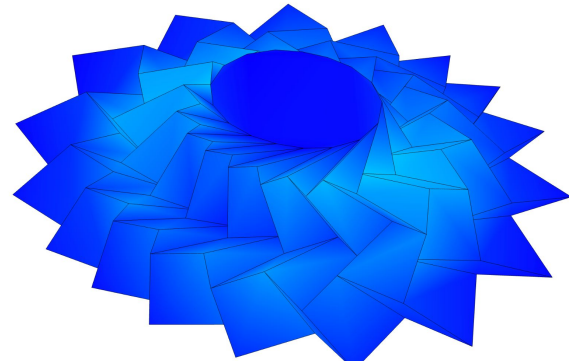
This project interprets data from ShapeL2, and generating physical shapes featuring art from the blockchain.



Origami Kabuto is a traditional origami very common for Japanese youth. We love the playful nature of what Katachi Gen artifacts represent.

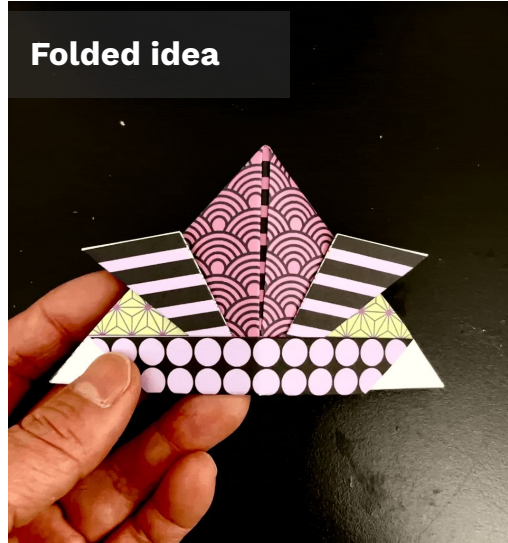
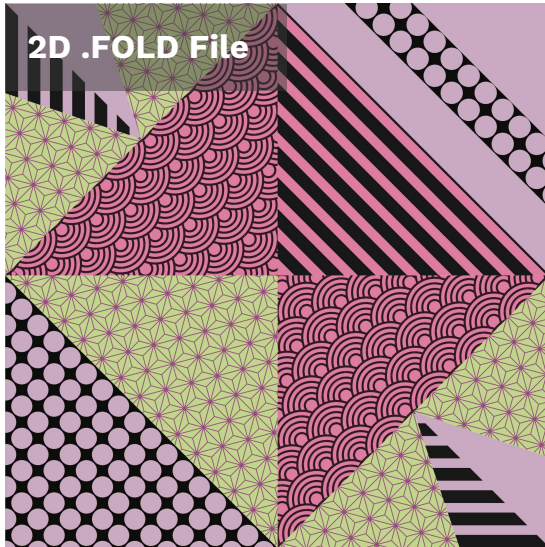
A 2D origami pattern (FOLD file) is generated at time of mint.

The fold complexity is determined by Shape MCP data about the wallet that is minting the token. Data used to generate fold lines includes Stack rank, Shape NFTs owned, and more MCP data.



Art

NFTs owned by the wallet at time of mint (as well as stack achievements) will be graphically represented on the 2D asset, which could be printed and folded by the collector (files included in token metadata).



Tech

In preliminary research we found some useful tools we'll use to build this:

- **White Paper by Jun Mitani:** A Method for Designing Crease Patterns for Flat-Foldable Origami with Numerical Optimization:
 - <https://www.jst.go.jp/erato/igarashi/publications/001/j15h2mita.pdf>
- **Origami Simulator**
 - <https://github.com/amandaghassaei/OrigamiSimulator>
- **Rabbit Ear**, computational origami
 - <https://github.com/rabbit-ear>
- **Shape MCP Server**
 - <https://github.com/shape-network/mcp-server>



Every Wallet with a Stack NFT would be able mint 1 Katachi Gen (price TBD) – Their own uniquely foldable, graphically designed (featuring their owned art), representing their participation on ShapeL2.

Thoughts? Ideas?

Team

Joe ([@josdotph](#)) and

Sembo ([@1000b](#))

