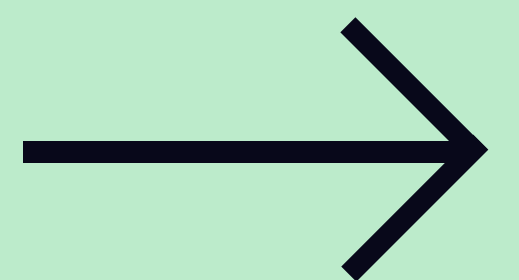


CYCLES

Markus Schnugg, Suha1 Enayat
Generative Gestaltung



Idee

Anfänge

Nutzer kann Linien setzen, die durch Spiegelung ein neues Gesamtbild erzeugen können.

Inspiration:

Karl Gerstner - Programme entwerfen
adaptiert and herausgegeben von
Harald Geisler 2020

MIDI Board

Besitzt verschiedene Elemente, wie Tasten, Schiebe- und Drehregler

Keyboard/Tastatur

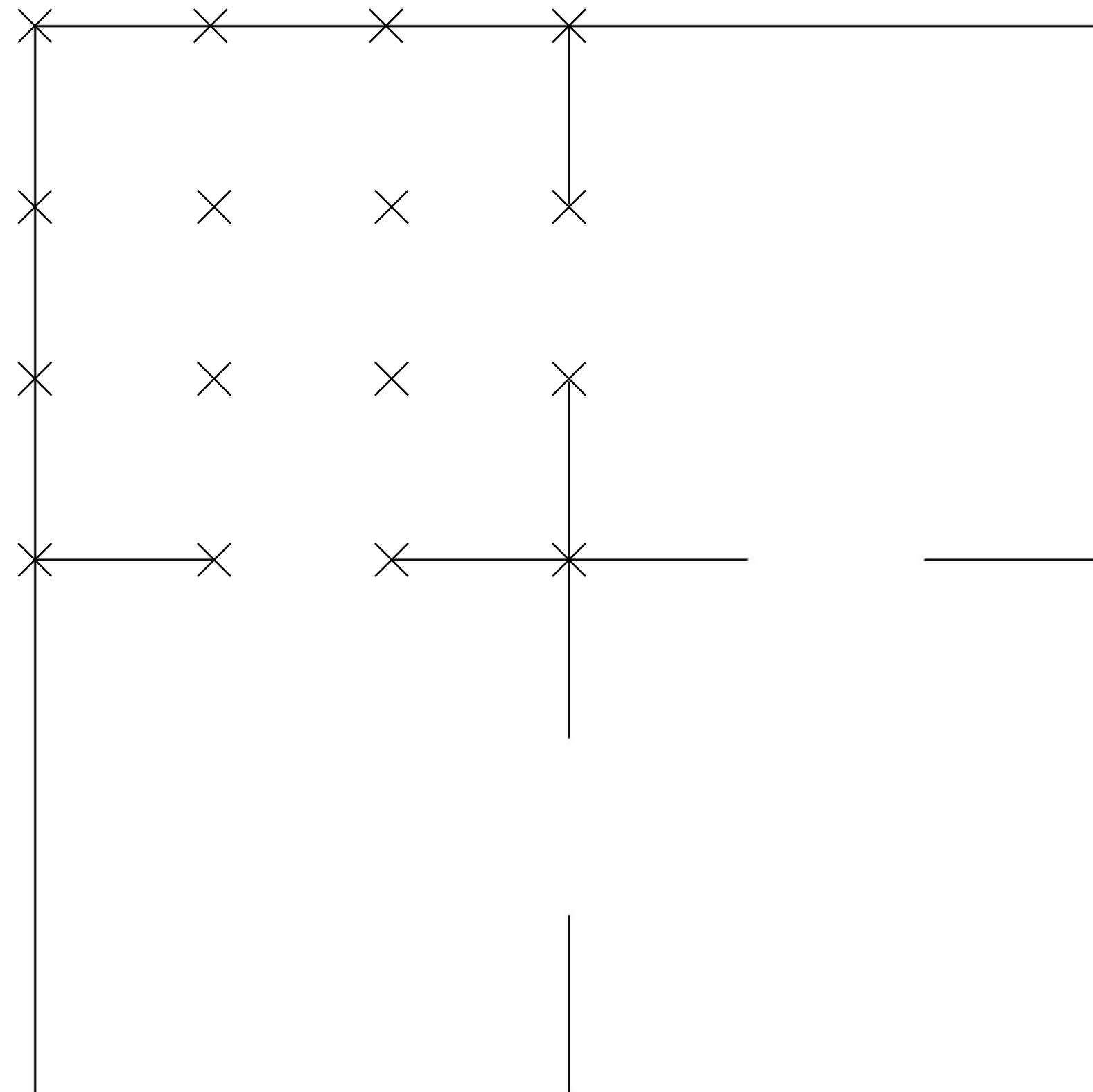
Für die Anwendung ohne MIDI

Processing

Vorgegebenes Tool



Anfänge



1. 4x4 Raster
2. 2x Gespiegelt
3. Linien erzeugen mit MIDI
Tasten-Kombination



Weitere Iteration

Aus Linien wird ein Graph erzeugt.

Ein Algorithmus erkennt Cycles im Graph -> Flächenerkennung

Fächen werden vom Programm gezeichnet

Was ist hierbei des Nutzers
eigene Entscheidung und was wird
vom Programm automatisiert
übernommen?

Kernfrage

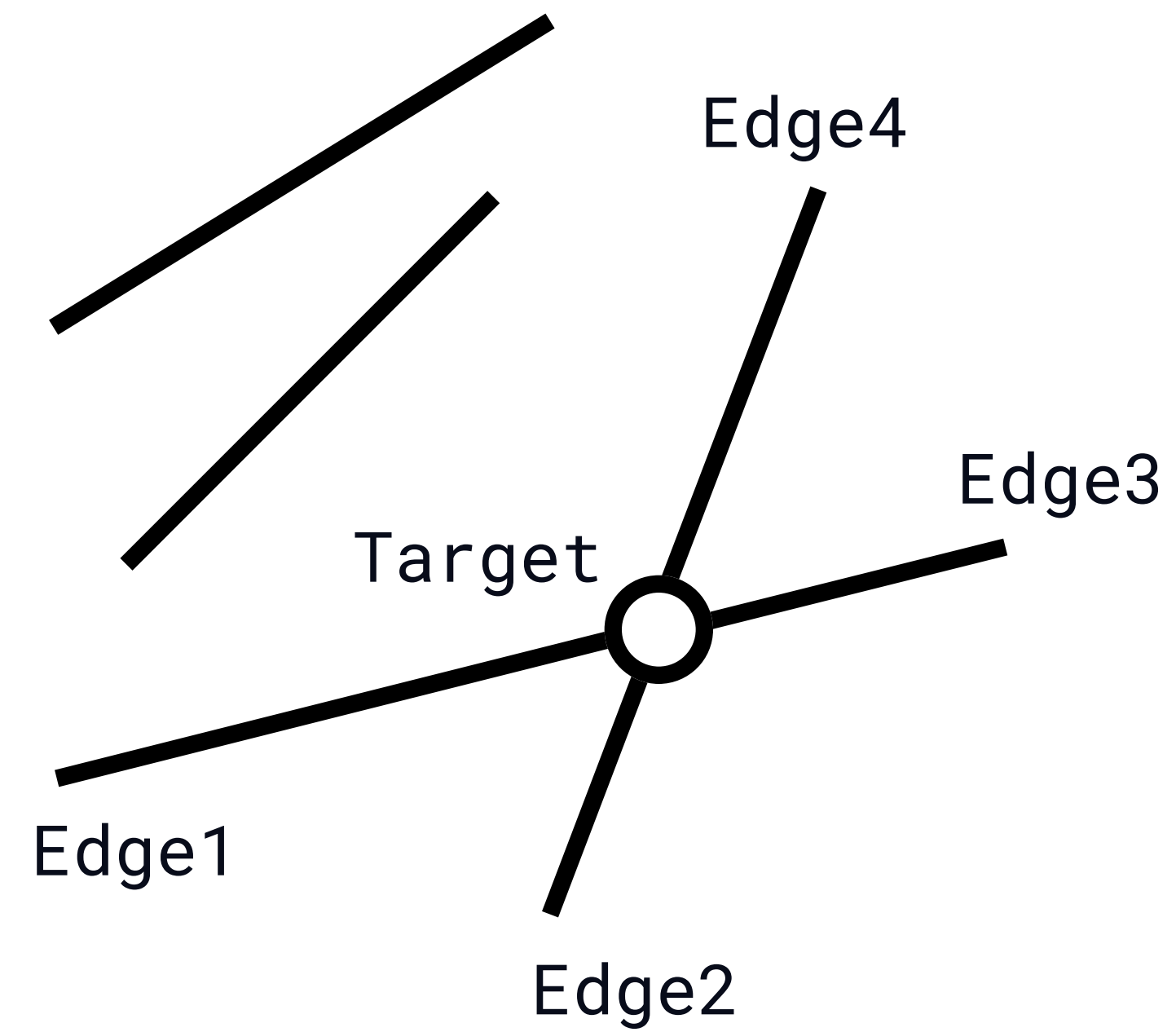
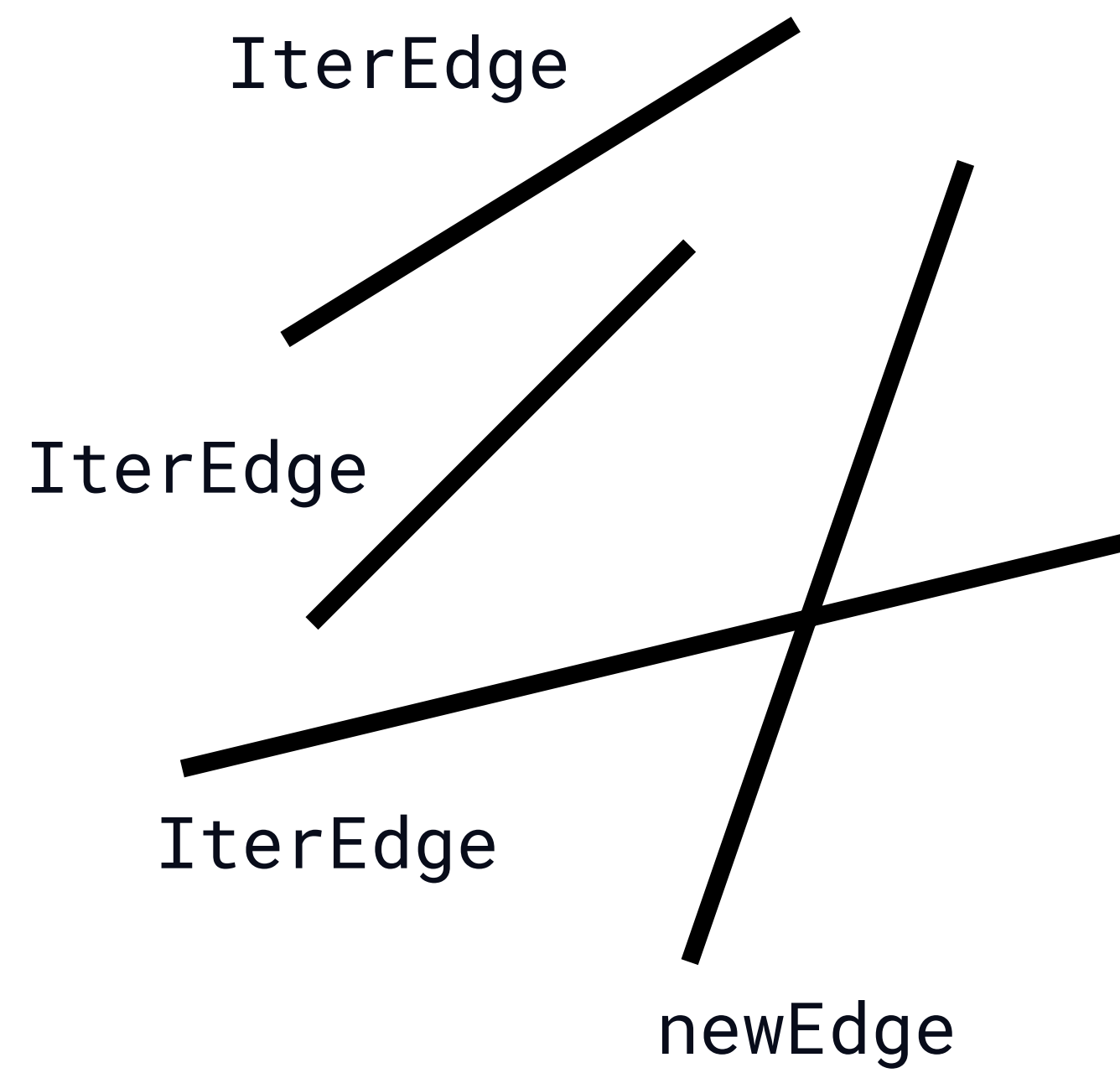
Nutzer:

- _kann Kanten setzen
- _kann erzeugte Flächen auswählen und deren visuelles Erscheinungsbild beeinflussen

Programm:

- _erkennt automatisiert Flächen aus den gesetzten Linien
- _bei mehreren Schnittpunkten steigt die Anzahl der möglichen Formen so stark an, dass es sogar wie unmöglich wird, exakt gewünschte Flächen zu erhalten
- _Flächen werden durch randomisierte Bezier Kontrollpunkte verzerrt.

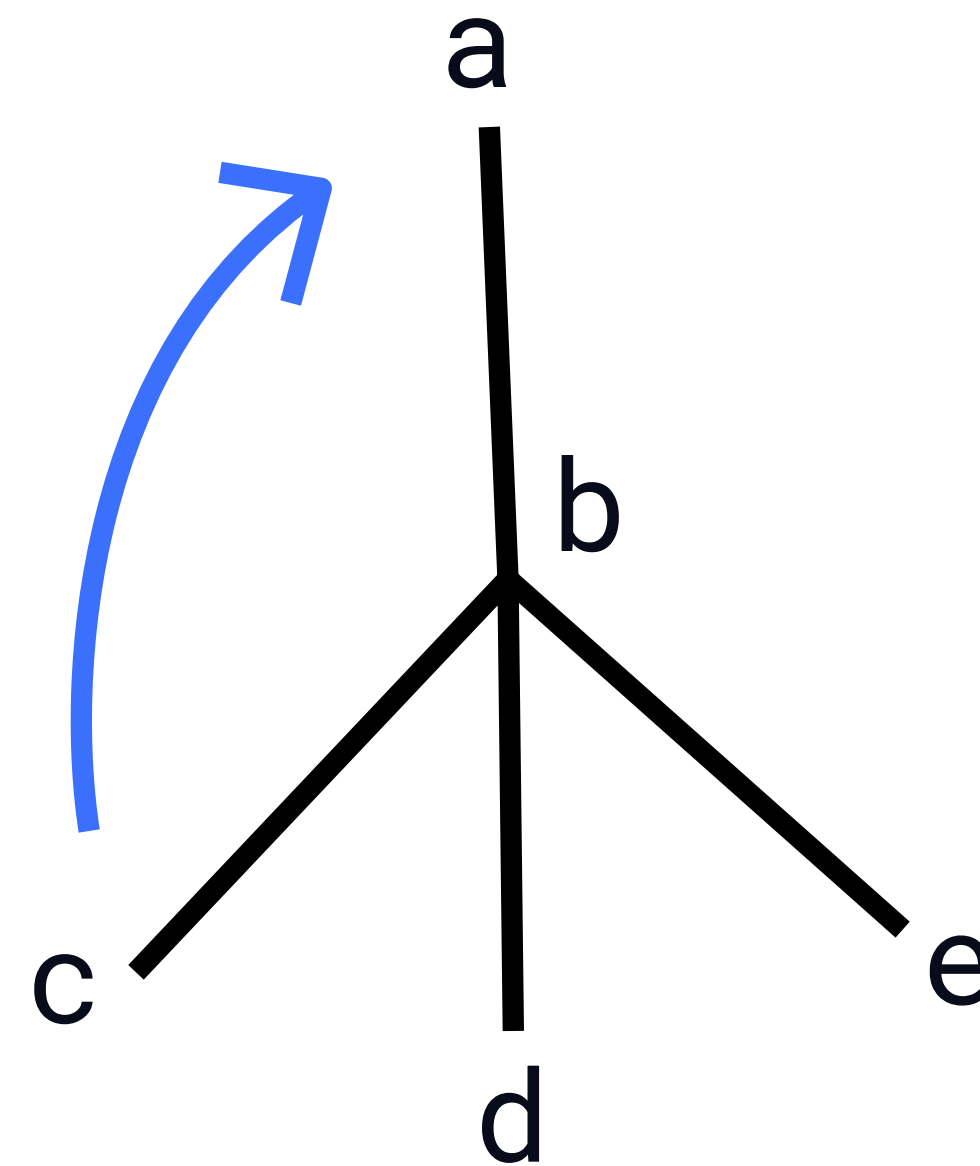
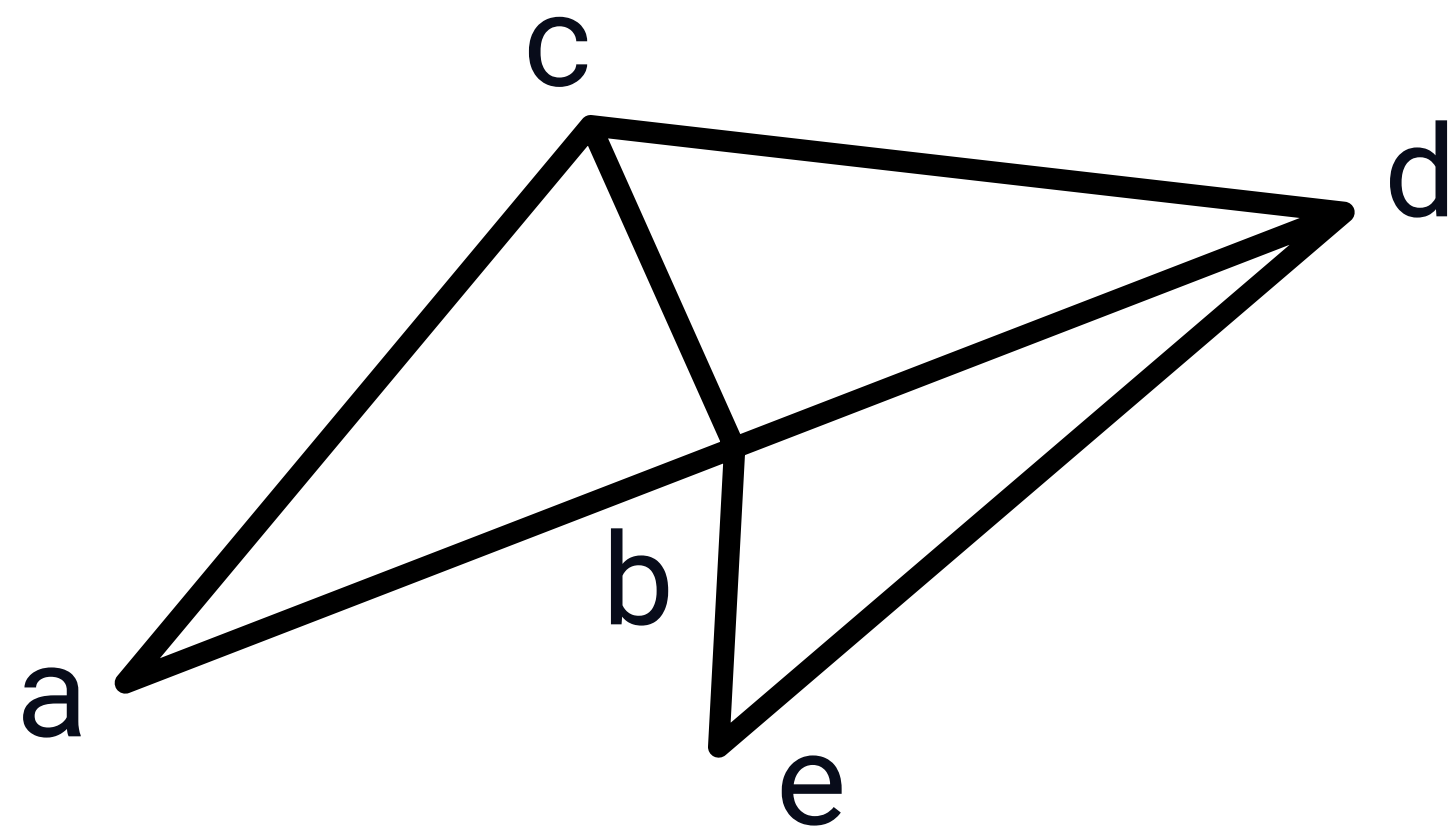
Intersection

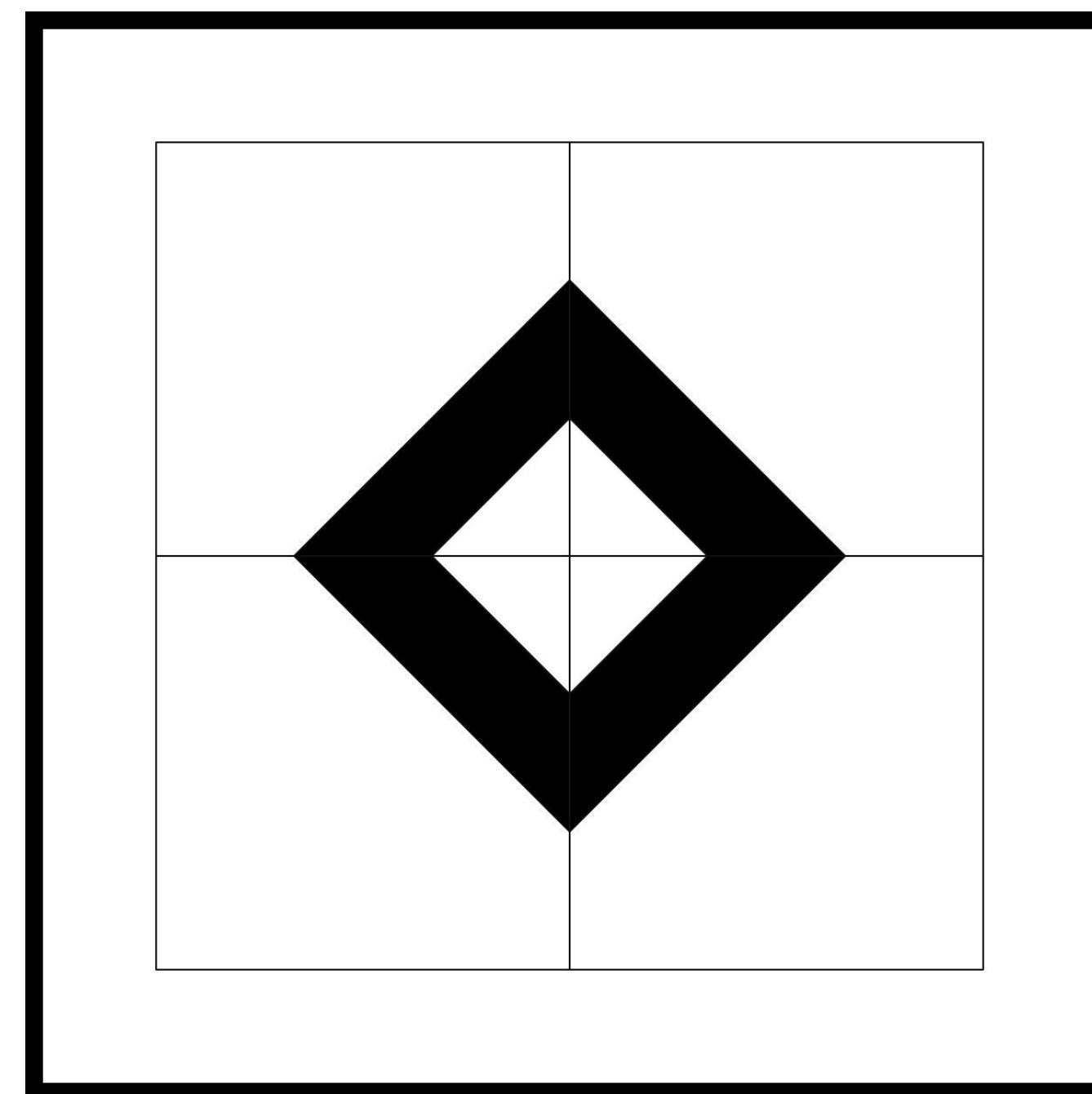
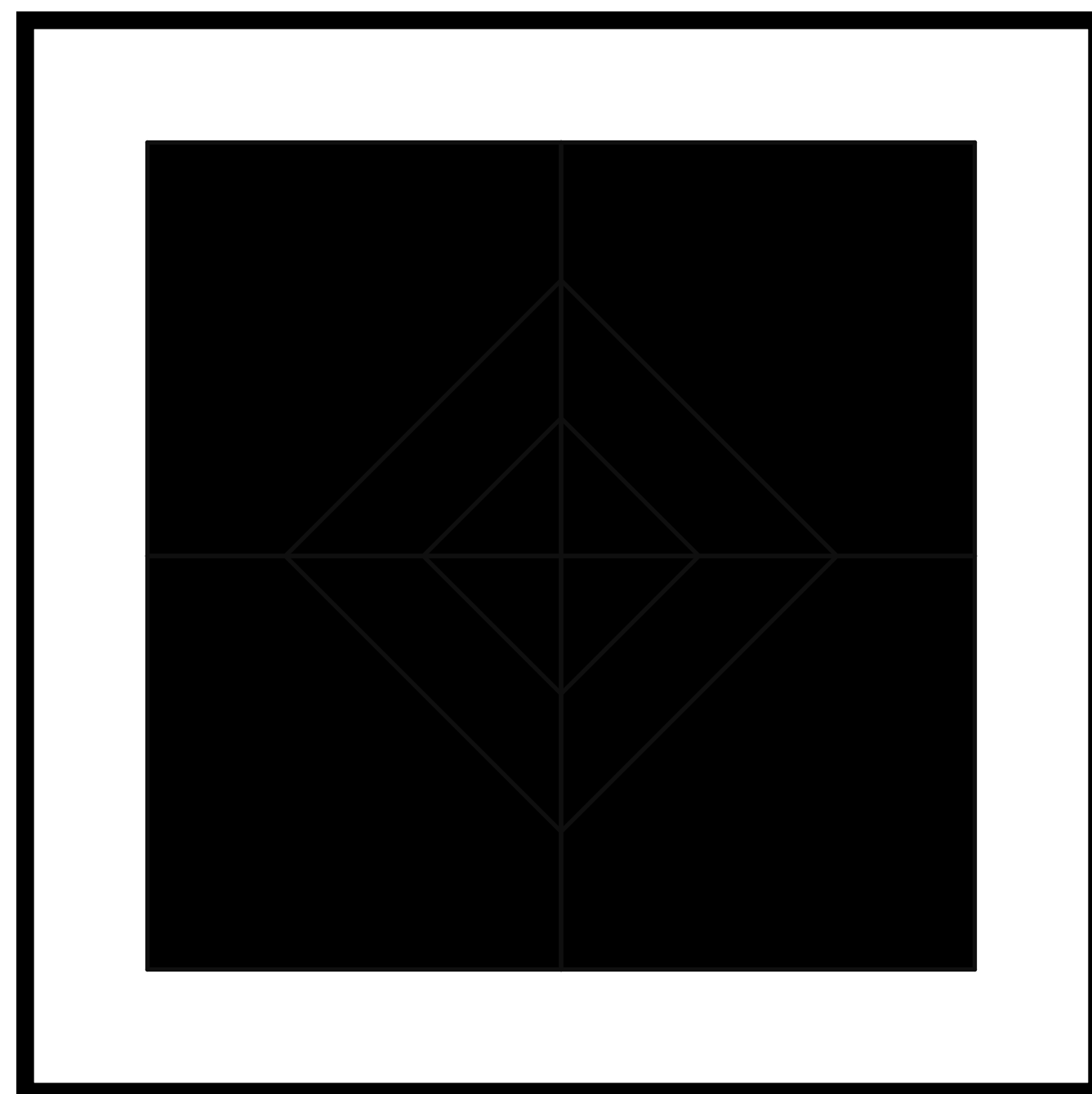
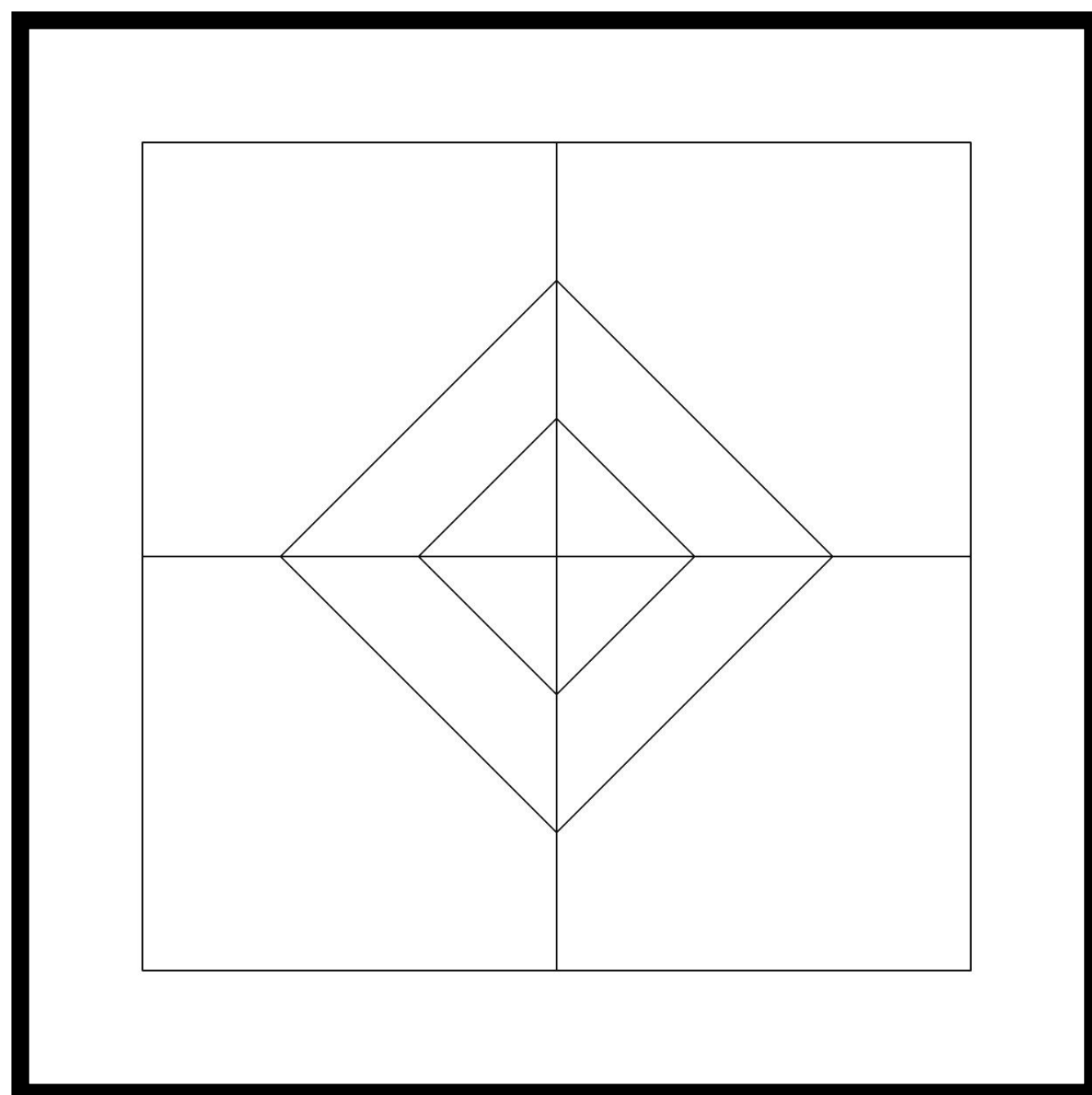


Intersection

```
// Only add edges that don't already exist
if (Graph.addEdge(newEdge))
    // Go through all the edges
    for (IterEdge in Graph)
        // If the edges intersect
        if (target = intersect(newEdge, IterEdge))
            // Delete Edges
            Graph.removeEdge(newEdge)
            Graph.removeEdge(IterEdge)
            //Add target vertex
            Graph.addVertex(target)
            //Add new edges from target
            Graph.addEdge(Edge(newEdge.p1, target))
            Graph.addEdge(Edge(newEdge.p2, target))
            Graph.addEdge(Edge(IterEdge.p1, target))
            Graph.addEdge(Edge(IterEdge.p2, target))
```

Cycle Detection (DFS)





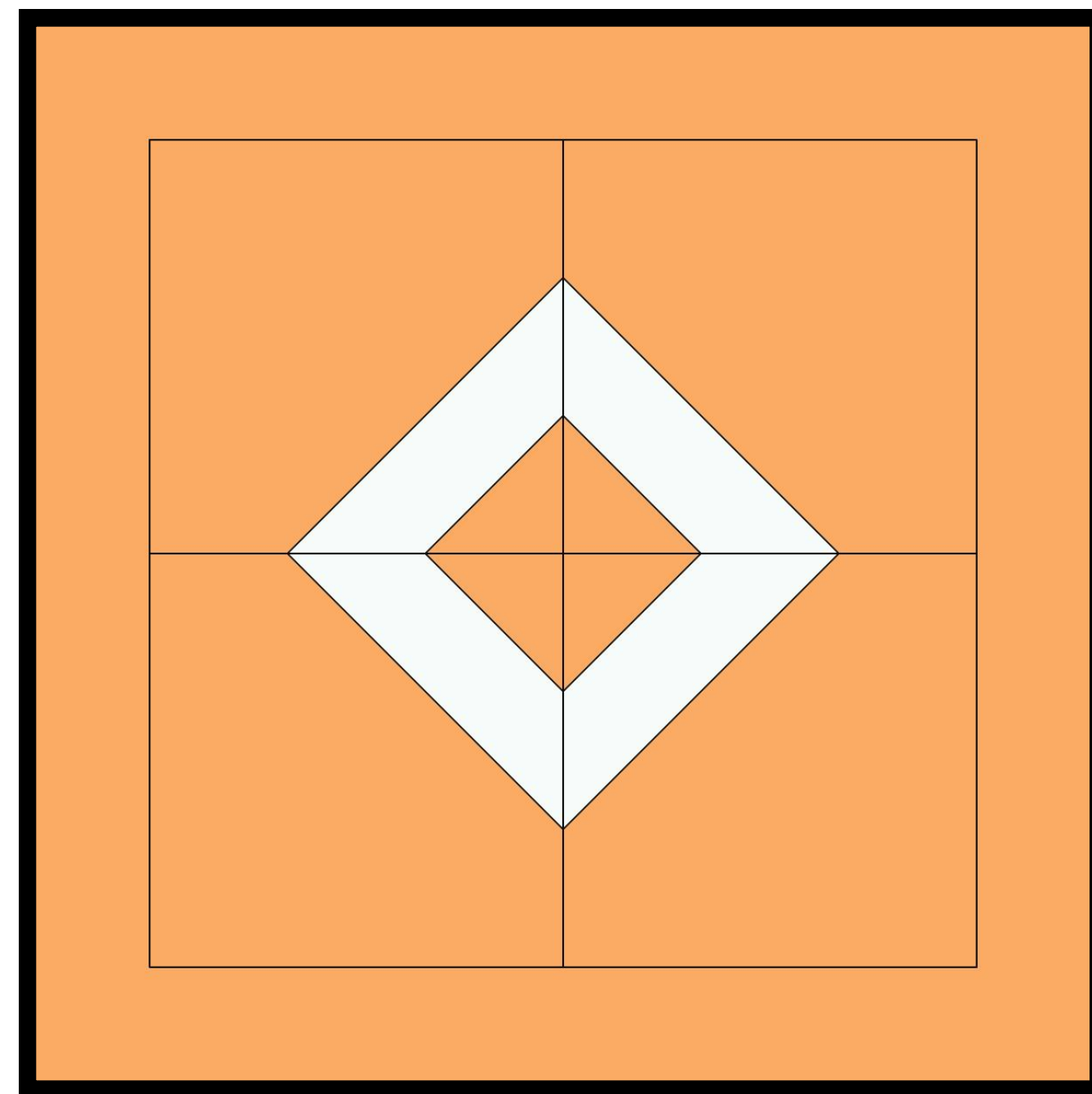
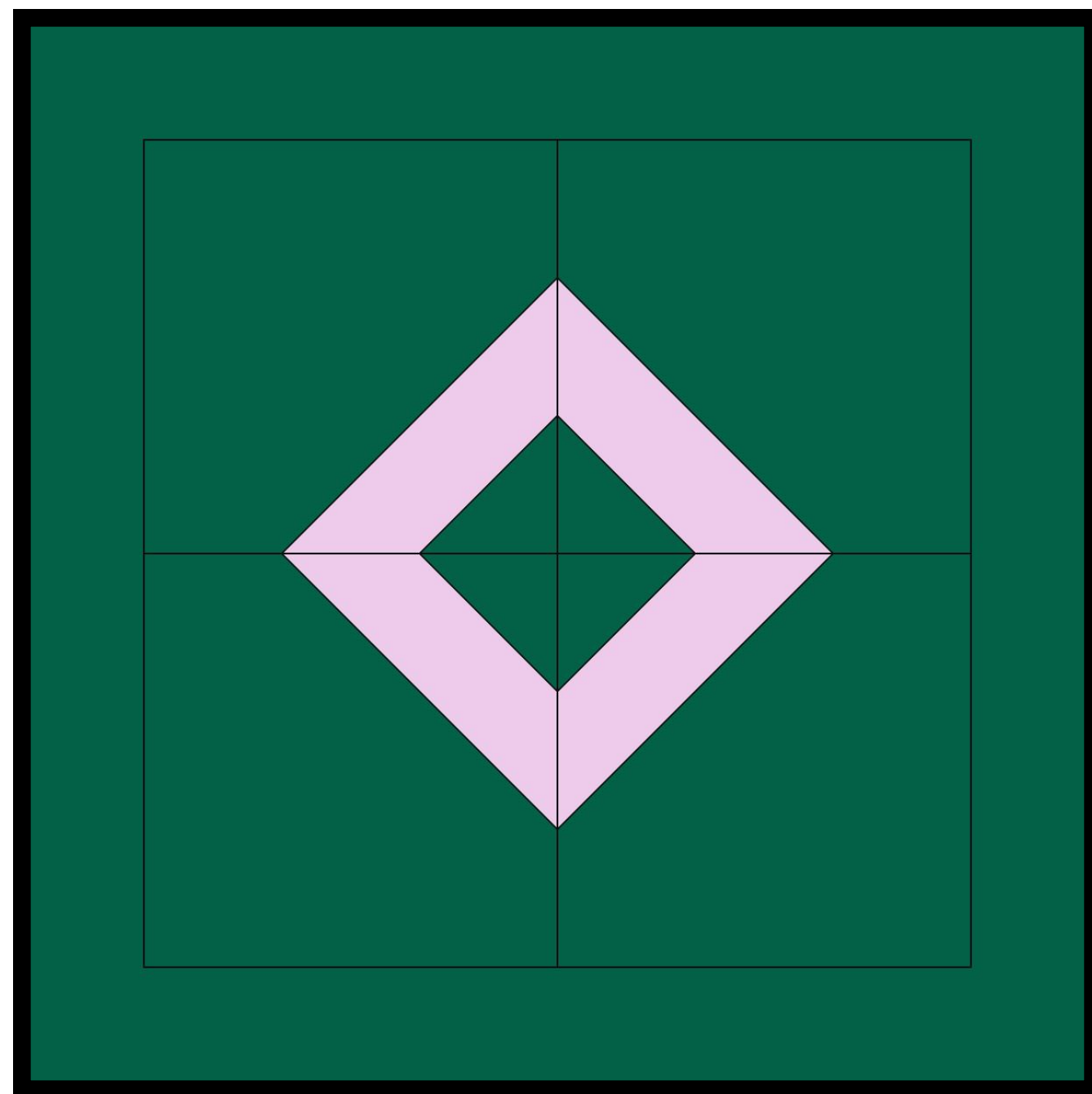
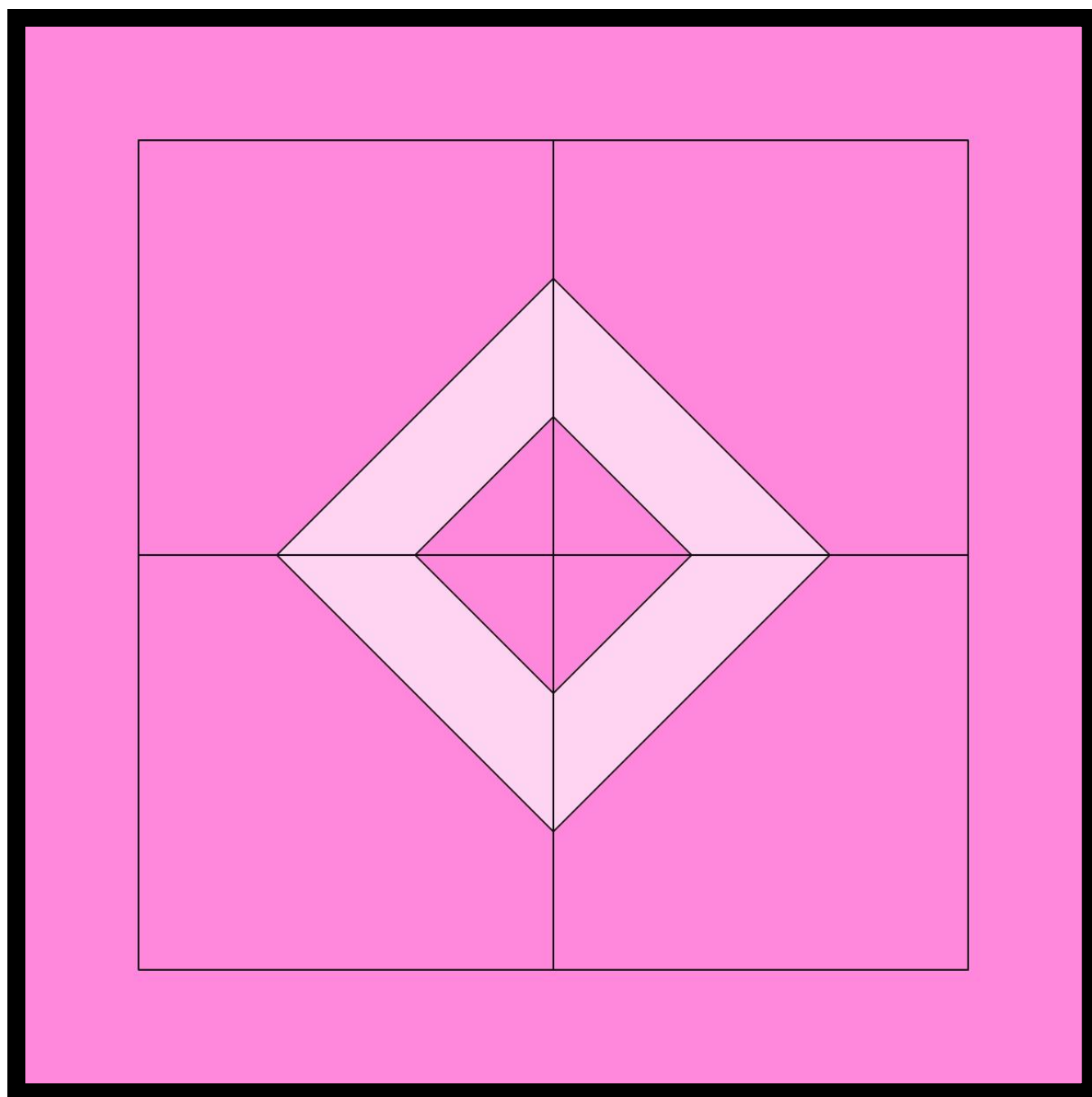


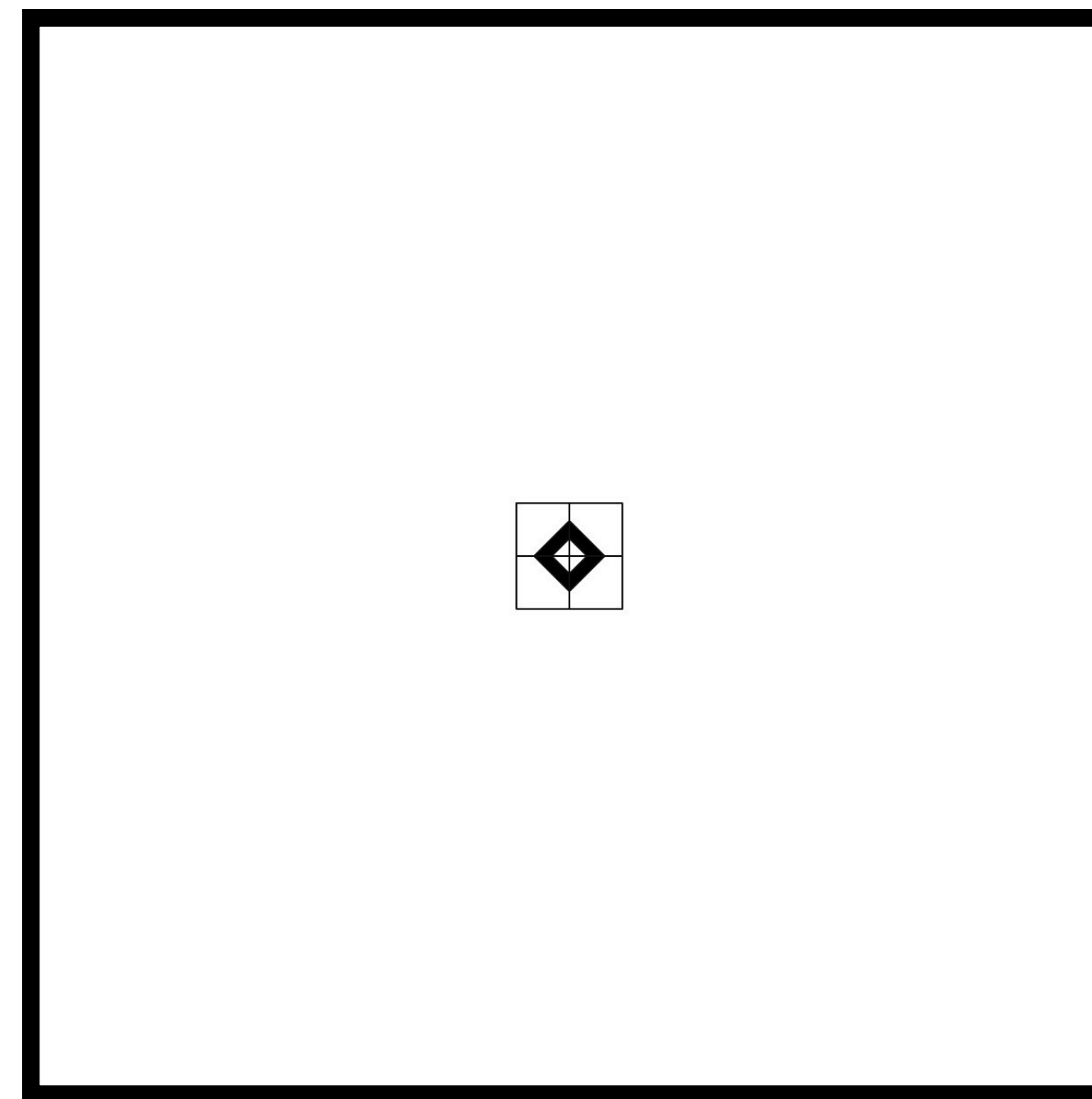
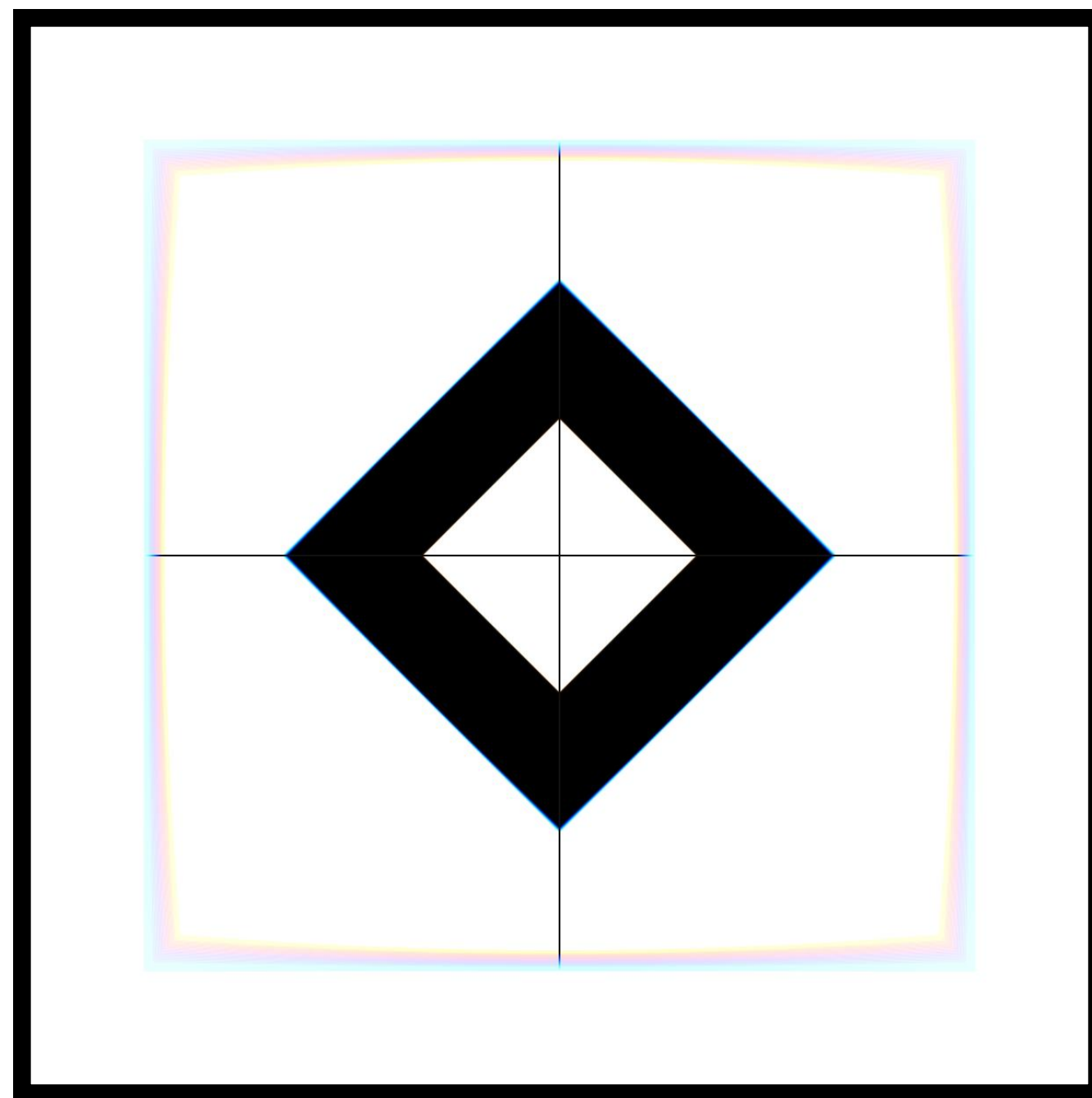
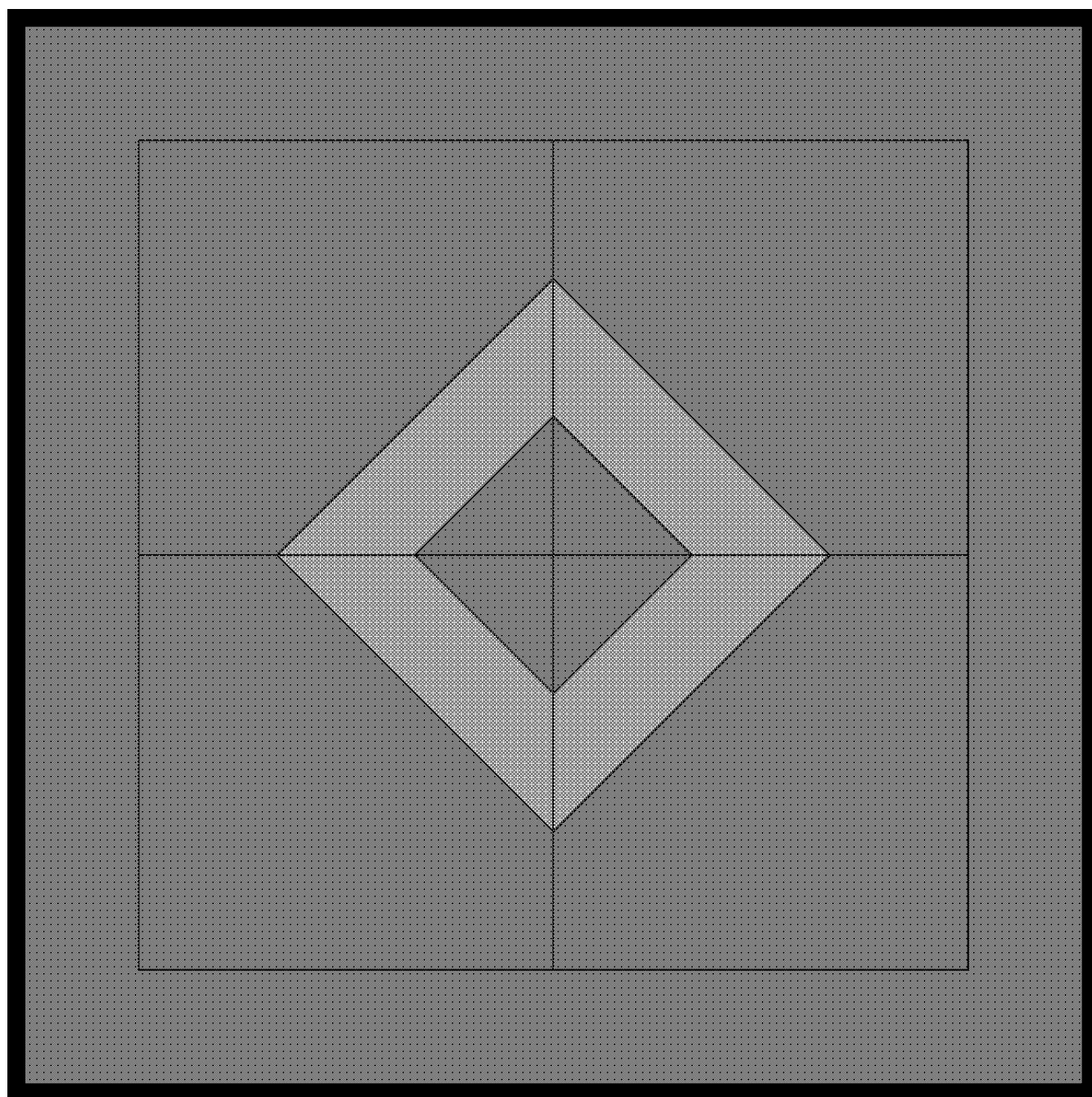
Weitere Iteration

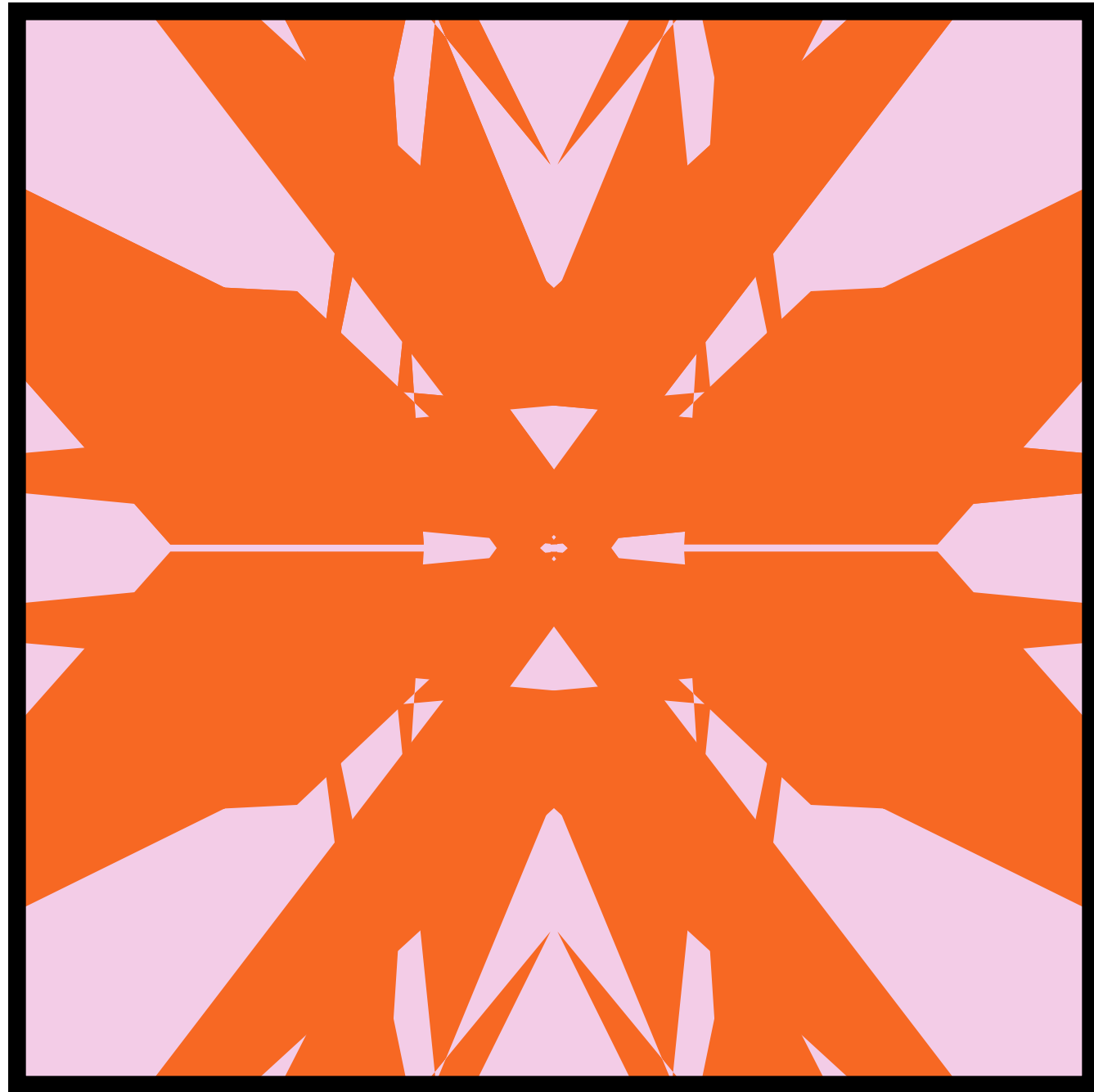
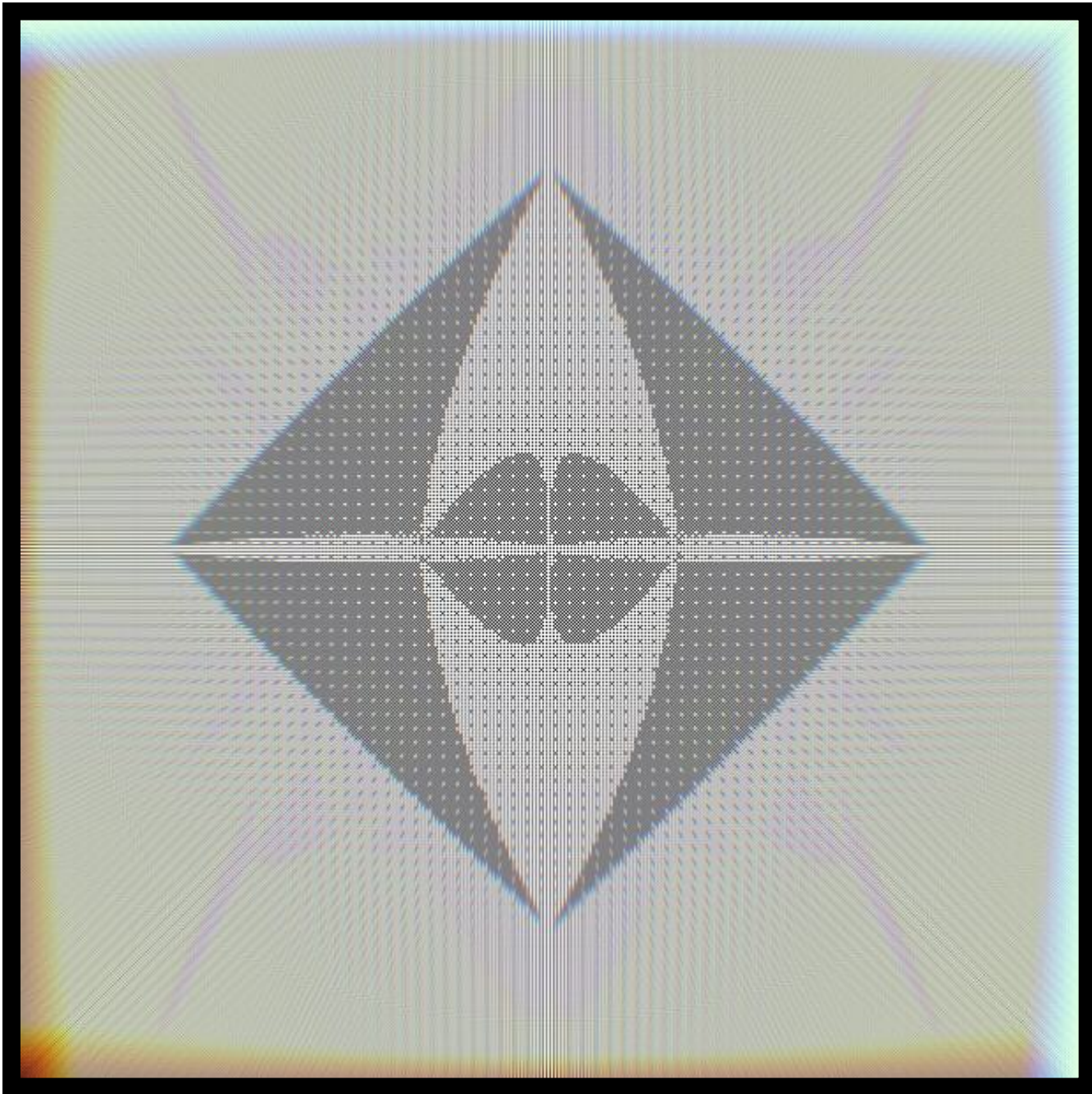
Dem Nutzer wird eine Auswahl an Farbkombinationen zur Verfügung gestellt.

Die Formen können mithilfe von zufällig erzeugten Bezierkurven verzerrt werden

Barreldistortion und Dithering
Filter als visuelle Effekte







Main Manipulations

Color Change

Bezier

Shape Switch

Shaders

Weitere Funktionen

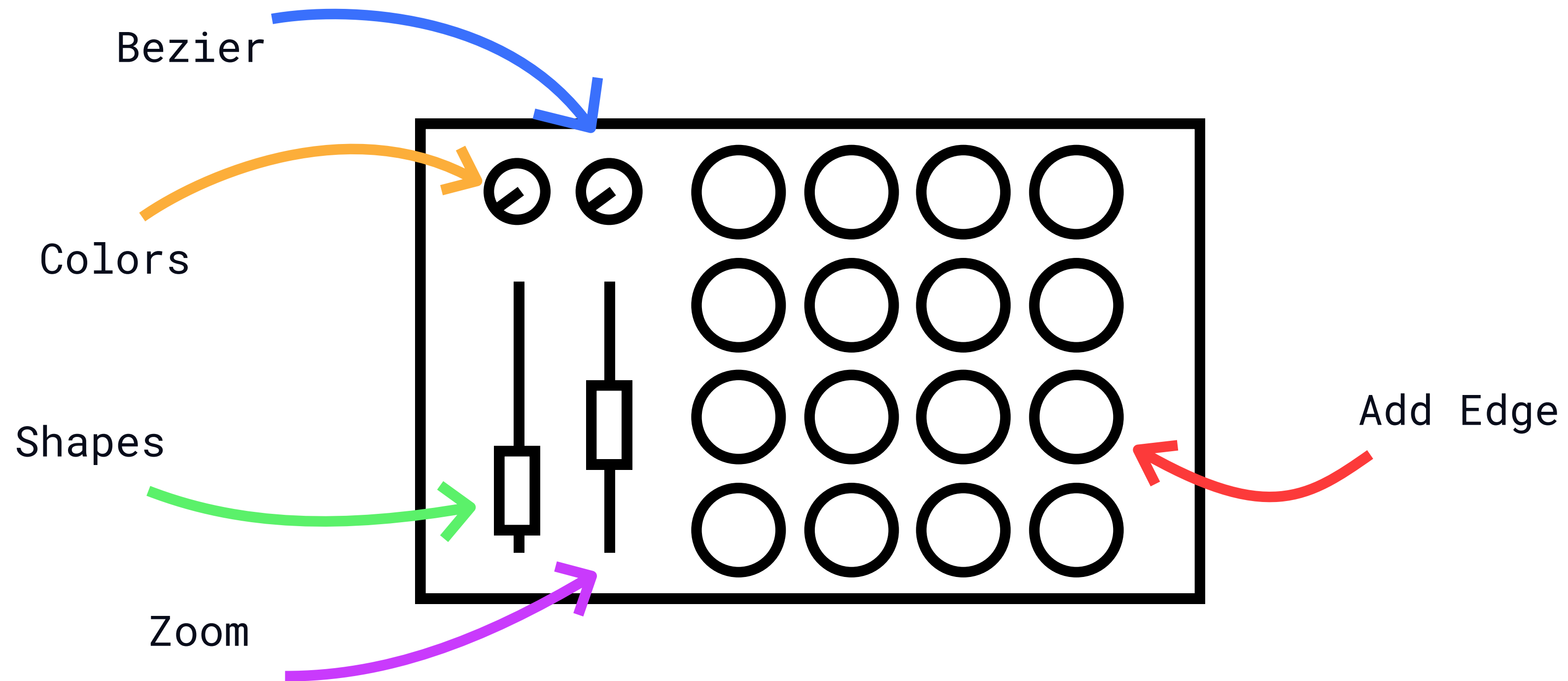
Bilder speichern (.png / .svg)

Lines togglen

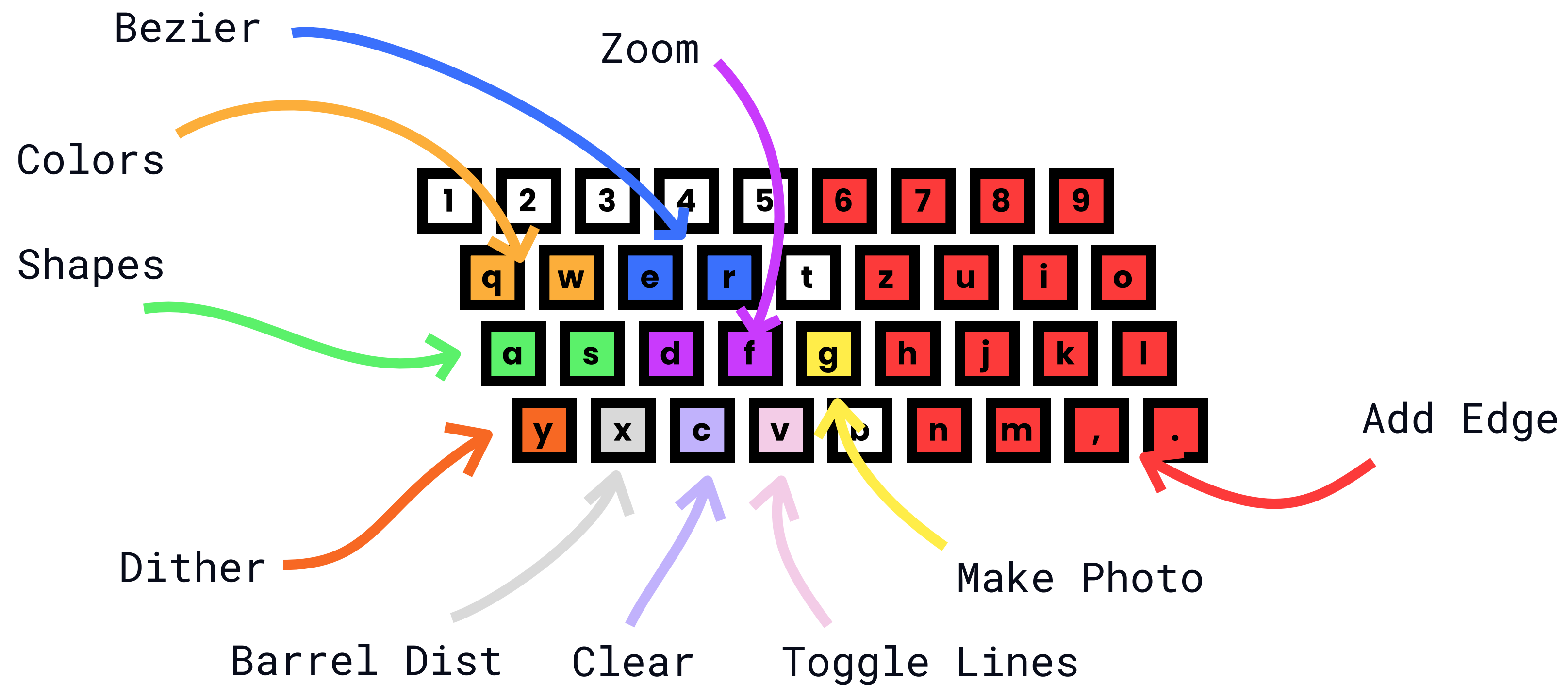
Löschen

Zoom

Controller Mapping



Keyboard Mapping





Programm als Formenlehre

Unbegrenzte Flächen

Ein Beispiel.

Um wenigstens in einem Falle die erstaunliche Mannigfaltigkeit und Schönheit solcher geometrischer Muster zu zeigen, sind in untenstehendem Schema 16 Quadrate dargestellt, eines davon mit einem innern Raster von je $3 \times 3 = 9$ Quadraten. Daraus lassen sich 16 Knotenpunkte ableiten. Diese Anzahl ist zufällig ebenso gross wie die Anzahl der verschiedenen Verbindungslinien zwischen ihnen und daher auch die Anzahl der Muster erster Ordnung. Jedes Gesamtquadrat ist viermal nebeneinander wiederholt, um die dadurch entstehende Verbindung der Einzelmuster zu zeigen. (Fig. 5)

In den Mustern 5 bis 14 liegt das «Thema», d. h. die Linie, die gesetzliche Vervielfältigung erfährt (hier ist die vierfache Spiegelung des Quadrats benutzt worden), entweder bar einer Quadratseite oder in einer Spiegellinie, so dass nur je vier Wiederholungen entstehen. Sie bilden die einfacheren, allgemein bekannten Formen. Die andern Muster sind aus Linien in anderer Lage entwickelt, welche je acht Wiederholungen ergeben. Die so entstehenden Formen sind grossenteils unbekannt.

Jede der 16 Formen lässt sich mit jeder anderen zu einem Muster zweiter Ordnung verbinden, wobei 120 neue Formen entstehen. Man kann sie leicht zeichnen, wenn man in das Quadratnetz zuerst das eine Muster einträgt und dann das andere darüberzeichnet.

Zeichnet man je drei aufeinander, so gewinnt man Muster dritter Ordnung, deren Anzahl 560 ist.

Aus: «Harmonie der Formen» von Wilhelm Ostwald. Verlag Unesma, Leipzig 1927.

