

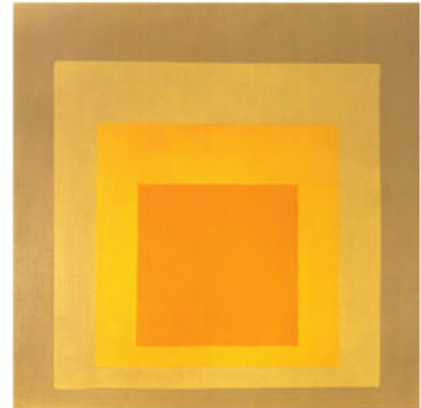
## Aufgabe 8

Dennis Blöte, 22.01.2007

### Homage to the square

Josef Albers begann die Arbeit an der Serie von Bildern, die er "Homage to the Square" genannt hat, im Alter von 62 Jahren. Er entwickelte bis zu seinem Tod 1975 hunderte Varianten dieses Themas. [...]

Im Zusammenspiel der Farben entwickelte Albers aus dieser Grundstruktur die komplexe Gesamtwirkung seiner Bilder. Er trug die Farben unvermischt aus der Tube mit einem Spachtelmesser auf die Masonit-Platten auf. Die Platten präparierte er mit mehrschichtigen Kreidegründen, um den Eindruck aus sich heraus leuchtender Farben zu erreichen.

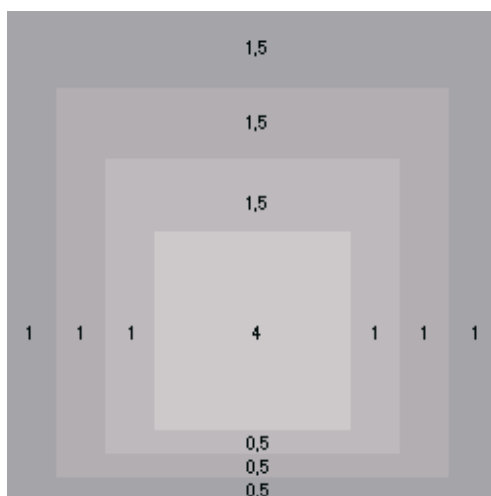


[...] Der Kontrast von bunten und unbunten Farben, von Farben gleicher Helligkeitsstufe in den beiden äußeren und den beiden inneren Quadraten erzeugen eine Spannung, die Albers sehr schätzte. Die gesamte Farbigkeit ist einheitlich und harmonisch, aber dennoch kraftvoll und spannungsgeladen. [1]

Die Farben sind nie gemischt, sondern direkt aus industriell hergestellten Farben aufgetragen, deren Artikel-Nummern der Maler auf der Rückseite der Bilder vermerkte. Dadurch wird ersichtlich, dass ein und dieselbe Farbe je nach Umgebung völlig unterschiedlich auf den Betrachter wirkt. Daher wird Albers auch zu den Vertreter von Hard edge gezählt. [2]

In seinen Serien wie "Homage to the Square" geht es um die Interaktion der Farben, die durch bloße Gegenüberstellungen ihre Wirkung unendlich verändern können. [3]

### Bildaufbau



Die Quadrate der Serie "Homage to the Square" sind nach diesem Schema konstruiert. Mehrfach wird eins der vier Quadrate weggelassen, gelegentlich werden die Quadrate noch umrandet, manchmal werden noch Diagonalen eingezeichnet.

-----  
[1] <http://www.arithmeum.uni-bonn.de/de/events/64>  
[2] [http://de.wikipedia.org/wiki/Josef\\_Albers](http://de.wikipedia.org/wiki/Josef_Albers)  
[3] <http://www.josef-albers.de/>  
[4] <http://www.laurentianum.de/ldalbe02.htm>

## Das Programm

Das Programm erzeugt 24 verschiedene Variationen des Bildes, sogenannte Permutationen. Die Anzahl der verschiedenen möglichen Kombinationen der Farben lässt sich über die Fakultät der verwendeten Farben berechnen: Werden wie in diesem Programm vier verschiedene Farben verwendet, so ergeben sich  $4! = 1 \cdot 2 \cdot 3 \cdot 4$  Möglichkeiten, die Farben anzuordnen.

Die initiale Homage gibt die Farben für das Programm vor:

```
color[] homageColors = {
    color(161, 182, 223),
    color(153, 153, 102),
    color(255, 255, 255),
    color(204, 255, 51)
};
```

In der Funktion `homage` ist der Bildaufbau festgelegt. Je nach Anzahl der Farben, welche als Parameter `c` übergeben werden, werden die Quadrate in einer Schleife nach innen gehend auf die Bühne gezeichnet. Die Positionsberechnung erfolgt nach dem auf Seite 1 abgebildeten Schema.

```
void homage(color[] c)
{
    for(int i=0; i<c.length;i++) {
        int rectSize = SIZE-(SIZE/5)*i;
        int rectYpos = int(SIZE/2+((SIZE/20)*i));
        drawSquare(c[i], rectYpos, rectSize);
    }
}
```

Die Funktion `drawSquare`, welche aus `homage` heraus aufgerufen wird, setzt die Farbe des Quadrates und zeichnet es horizontal zentriert auf die Bühne.

```
void drawSquare(color c, int y, int s)
{
    fill(c);
    rect(SIZE/2, y, s, s);
}
```

Mit der Funktion `permute` lassen sich die Permutationen durch Vertauschung einzelner Arrayelemente erzeugen. Leider funktioniert diese Fassung nicht richtig und liefert nur 12 verschiedene Folgen.

```
void permute(int index)
{
    int i = (index == 0) ? 0 : index % homageColors.length;
    int j = (i == homageColors.length-1) ? 0 : i+1;
    swap(i, j);
}
```