

EVA MARTIŠOVÁ
BARVY



Zadání bakalářské práce

Barvy

Jméno a příjmení:

Eva Martišová

Osobní číslo:

A13000030

Studijní program:

B8206 Výtvarná umění

Studijní obor:

Vizuální komunikace

Zadávající katedra:

Katedra výtvarného umění

Akademický rok:

2017/2018

Zásady pro vypracování:

Výstupem práce bude webová aplikace, která se budu zabývá barevnými geometrickými tvary. A to jejich náhodným generováním a uspořádáním ve webovém prostoru.

1. Prostorová světelná instalace (světelné objekty)
2. Průvodní teoretická zpráva ve formátu A4. Zpráva obsahuje mezi jinými inspirační zdroje, souvislosti, vlastní definici díla a projekt samotný.
3. Průvodní technická zpráva ve formátu A4 v pevné vazbě, včetně originálu zadání práce a prohlášení o autorském právu. Zpráva obsahuje detailní technický popis užitě technické prostředky, výkresovou dokumentaci, zdrojové texty a další.
4. Elektronická podoba všech částí bakalářské práce na CD-ROM (akceptovatelné formáty: pdf, mp3, mp4)
5. V systému STAG (Moje studium-Kvalifikační práce-Doplňit údaje o práci) vložit veškerá technická data o práci a soubor obsahující kompletní výkresovou a textovou dokumentaci, průvodní zprávu, technickou zprávu a doplnit související textová pole

Rozsah grafických prací:
Rozsah pracovní zprávy:
Forma zpracování práce:

viz výše
viz výše
tištěná/elektronická



Seznam odborné literatury:

Vedoucí práce: doc. Stanislav Zippe
Katedra výtvarného umění
Datum zadání práce: 2. října 2017
Předpokládaný termín odevzdání: 25. května 2018


prof. Ing. arch. Zdeněk Fránek
děkan

L. S.



doc. Stanislav Zippe
vedoucí katedry

V Liberci 2. října 2017

BARVY

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta umění a architektury

Bakalářská práce

2018

Autor práce: **Eva Martišová**

Vedoucí práce: **doc. Stanislav Zippe**

Studijní program: B8206 - Výtvarná umění

Studijní obor: 8206R067 - Vizuální komunikace

PROHLÁŠENÍ

Byla jsem seznámena s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména §60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL. Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědoma povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracovala samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí vloženou do IS STAG

Datum 25.5.2018

Podpis Markišová

PODĚKOVÁNÍ

Tímto děkuji panu doc. Stanislavu Zippemu za trpělivé vedení a cenné rady. Dále bych ráda poděkovala i MgA. Jaroslavu Prokešovi za vždy ochotnou pomoc a panu PhDr. Karlu Srpovi, Ph.D. za zajímavé diskuze a podporu s teoretickou problematikou. V neposlední řadě děkuji svému bratrovi Filipu Dokládaloovi z Flumpy s.r.o. za technickou podporu.

OBSAH

- 1/ Teoretická reflexe
- 2/ Obrazová dokumentace
- 3/ Technická dokumentace
- 4/ Portfolio
- 5/ Životopis

1/ TEORETICKÁ REFLEXE

Barvy

Náhoda je nepředvídatelná, může být ohraničená nebo mít nekonečný počet možností, je nevyzpytatelná. Spojením náhody, barevnosti a geometrického rastru se zabývám už několik let.

Webová aplikace s názvem Barvy je další z prací navazující na můj Net Art pojmenovaný 1974, jež vychází z mechanického hlavolamu Rubikovy kostky. Společným znakem těchto uměleckých webových programů je princip stisku tlačítka způsobující náhodnou změnu barev s velkým počtem kombinací. Statická forma díla je tak přeměňována na dynamickou.

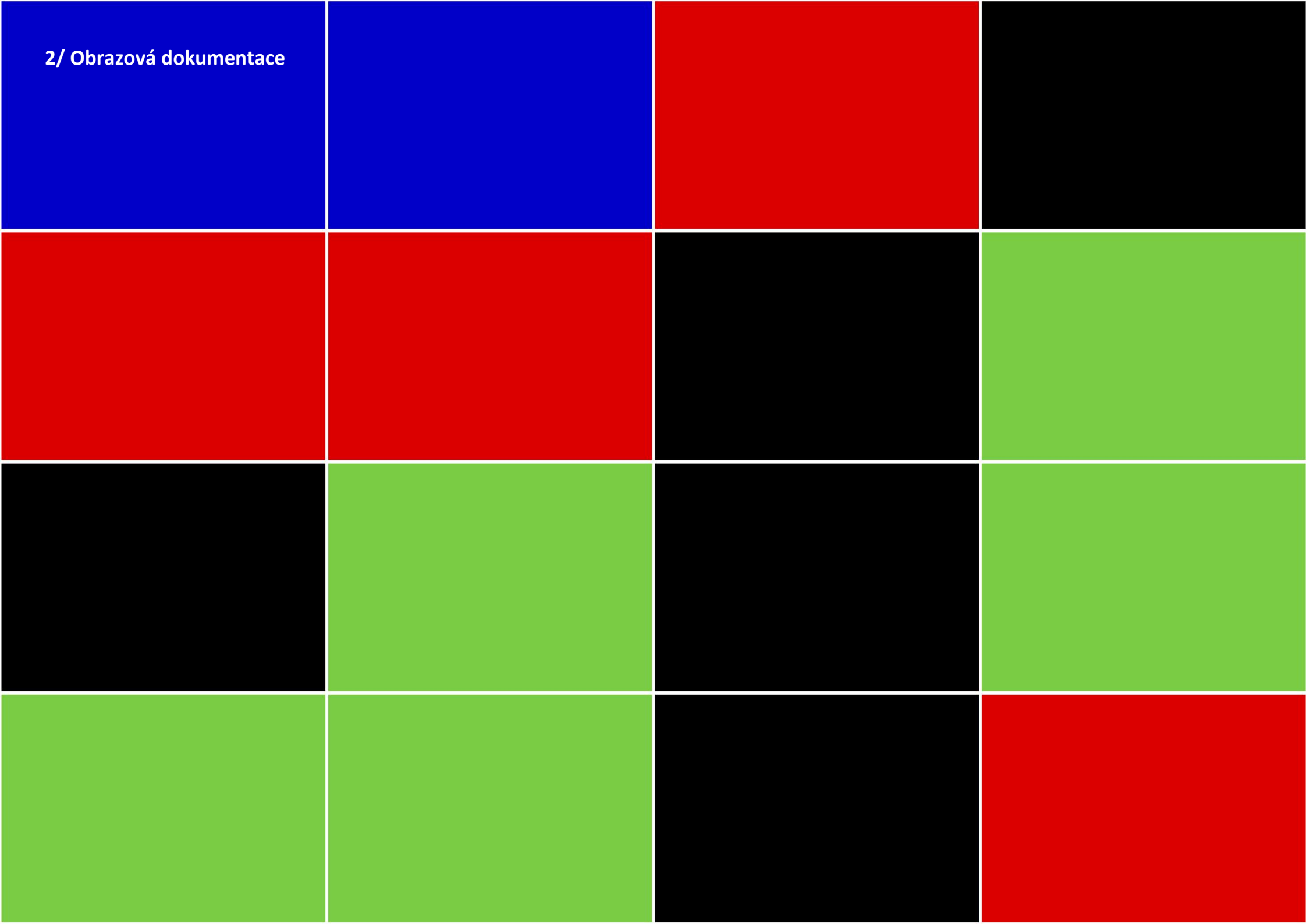
Cílem práce, jež je inspirovaná díly zabývajícími se také náhodou a obrovskou škálou barev Gerharda Richtera ze série Colour Charts, je zkoumat vztahy a chování barev v režimu pseudonáhodného generování.

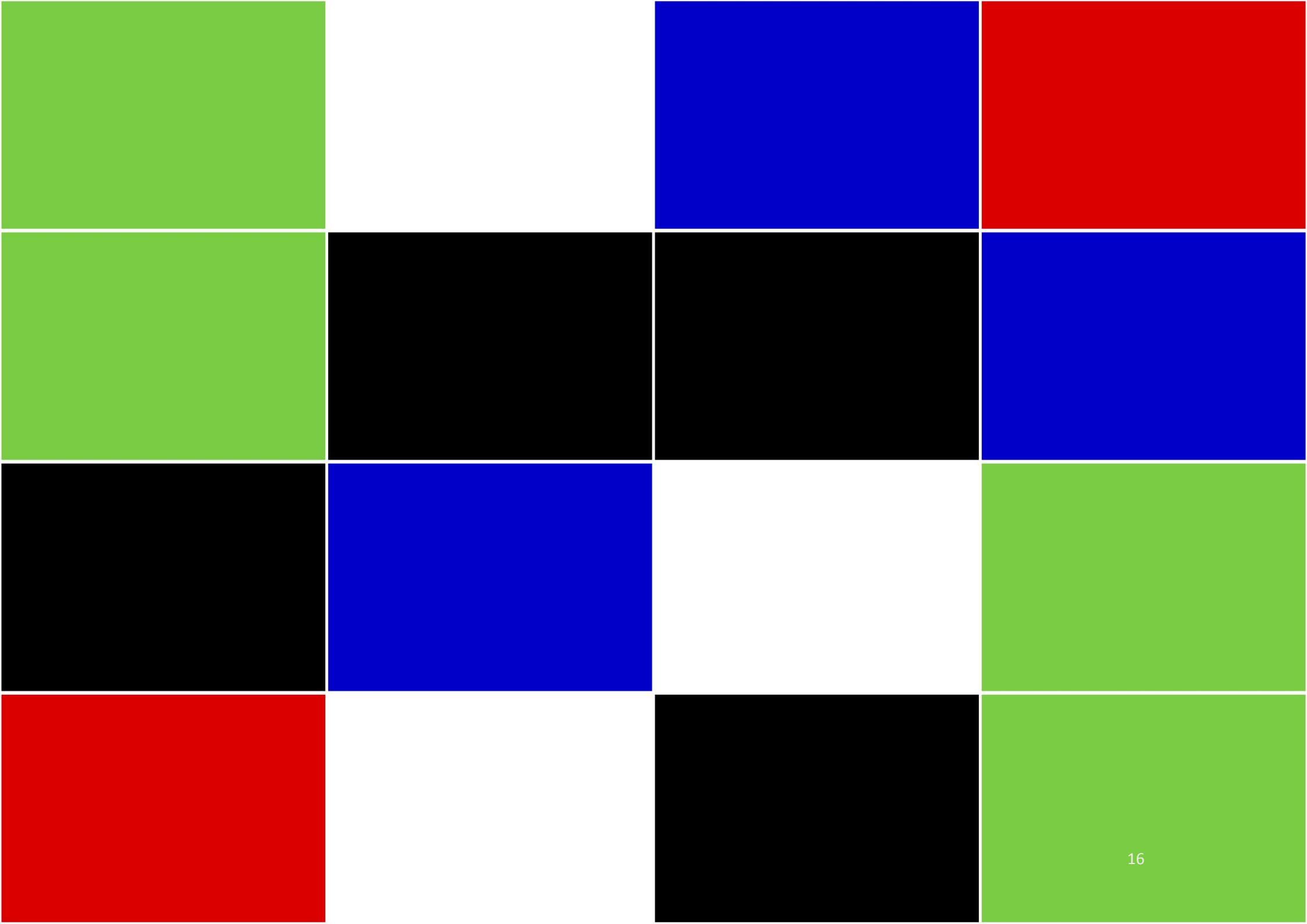
Spojitost této aplikace s obrazy z edice Colour Charts Gerharda Richtera, jež začal malovat v 60. letech, spočívá v zacházení s geometrickým rastrem složeného z barevně vykreslených předem určených polí. Dalšími společnými znaky jsou použití náhodného generování a variabilita barev.

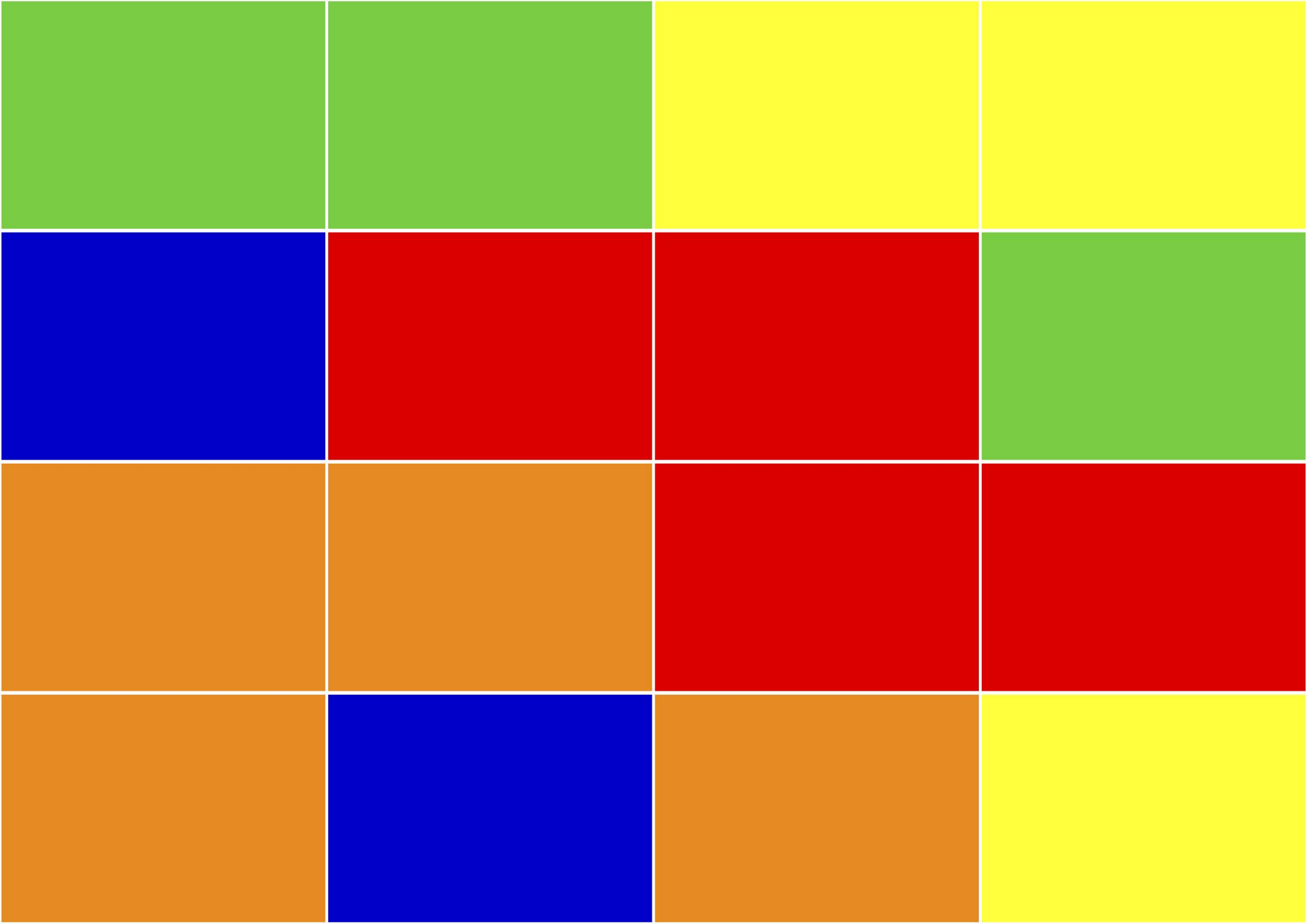
Webová aplikace je tvořena z barevných polí. Barvy jednotlivých segmentů jsou generovány náhodně z předem dané barevné škály. Bylo vytvořeno několik variant s různým počtem polí a zvolených barev. Od jednodušších až po ty složitější. Výsledná verze, jež je převedena do webového prostoru, je složená z 16 segmentů a dochází v ní k náhodnému generování těchto barev - červené, zelené, modré a černé.

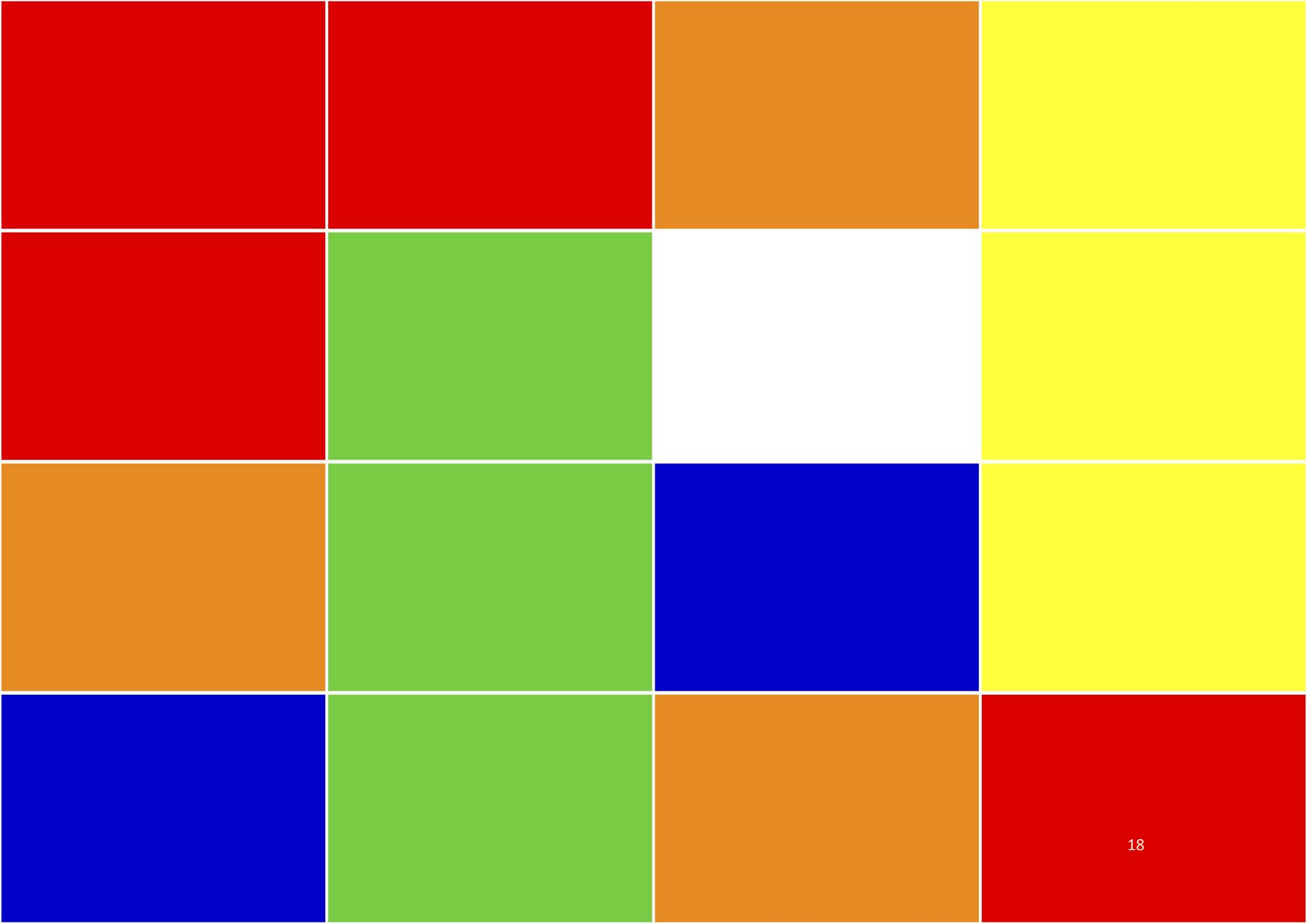
Systém, který zde probíhá, způsobuje náhodné zanikání a objevování vygenerovaných obrazců. Divák by snad předvídal, že části, které se v náhodné barvě objeví zpět, budou ty již skryté. Avšak svou barvu náhodně mění také už vyobrazené segmenty.

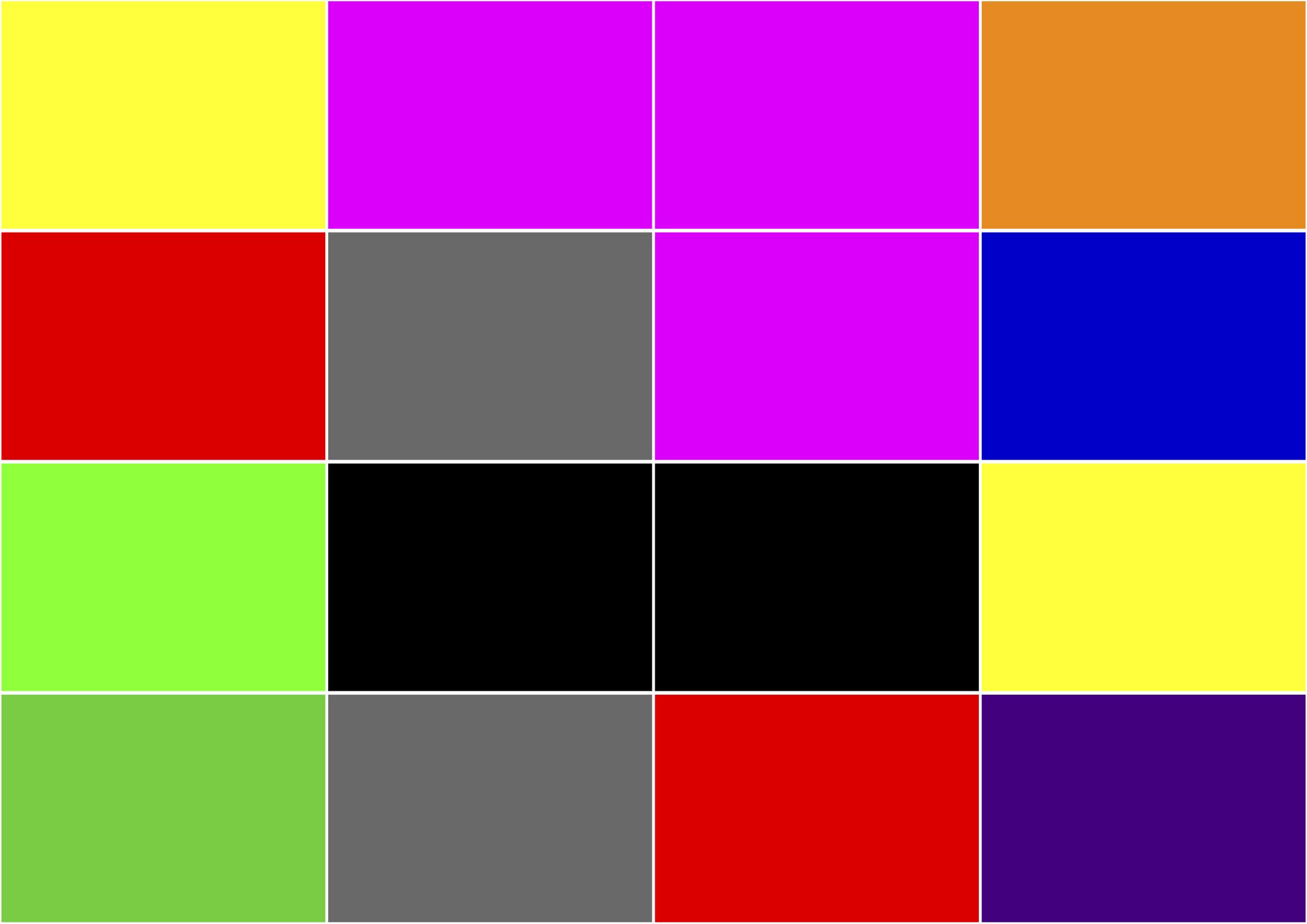
2/ Obrazová dokumentace

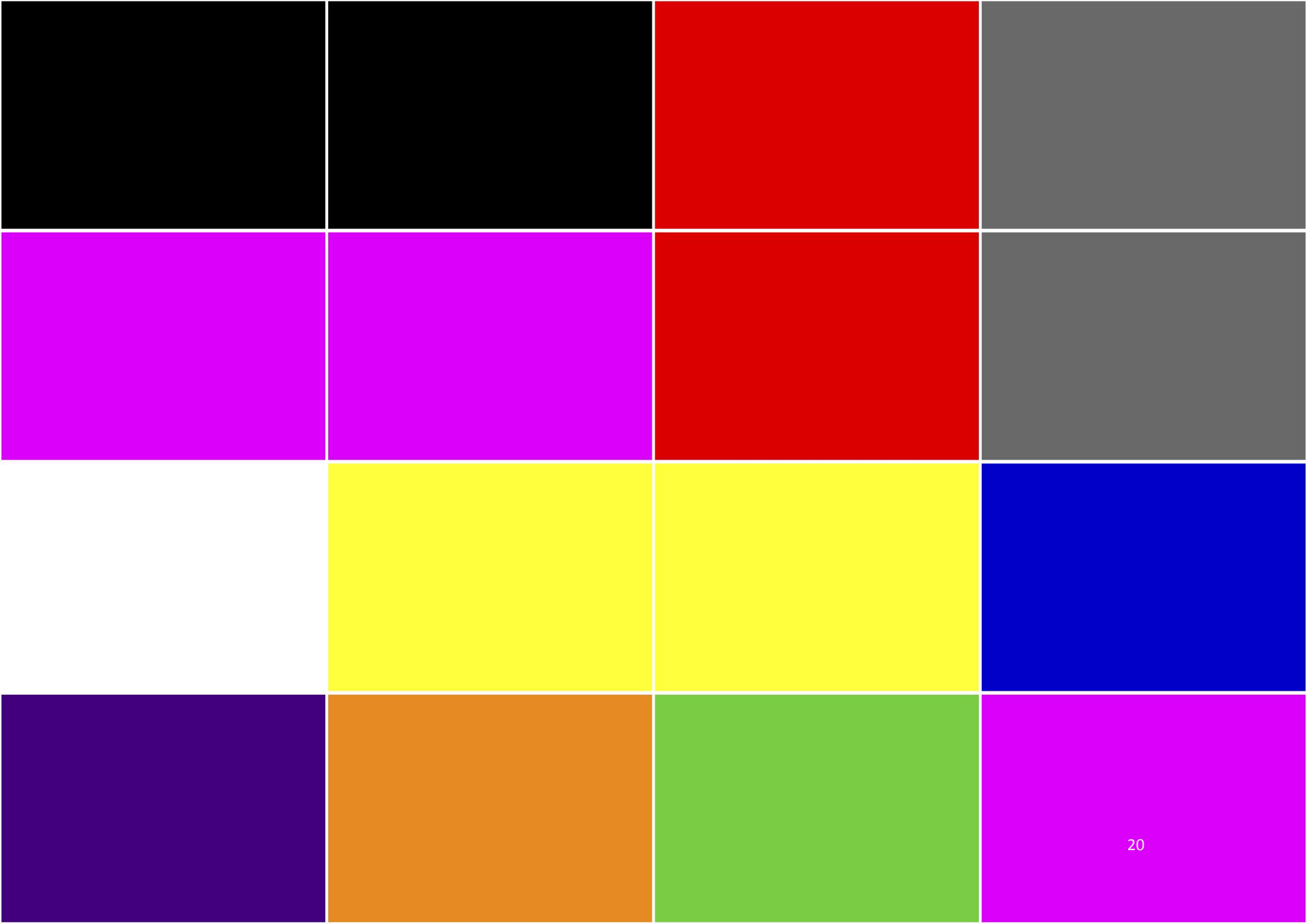


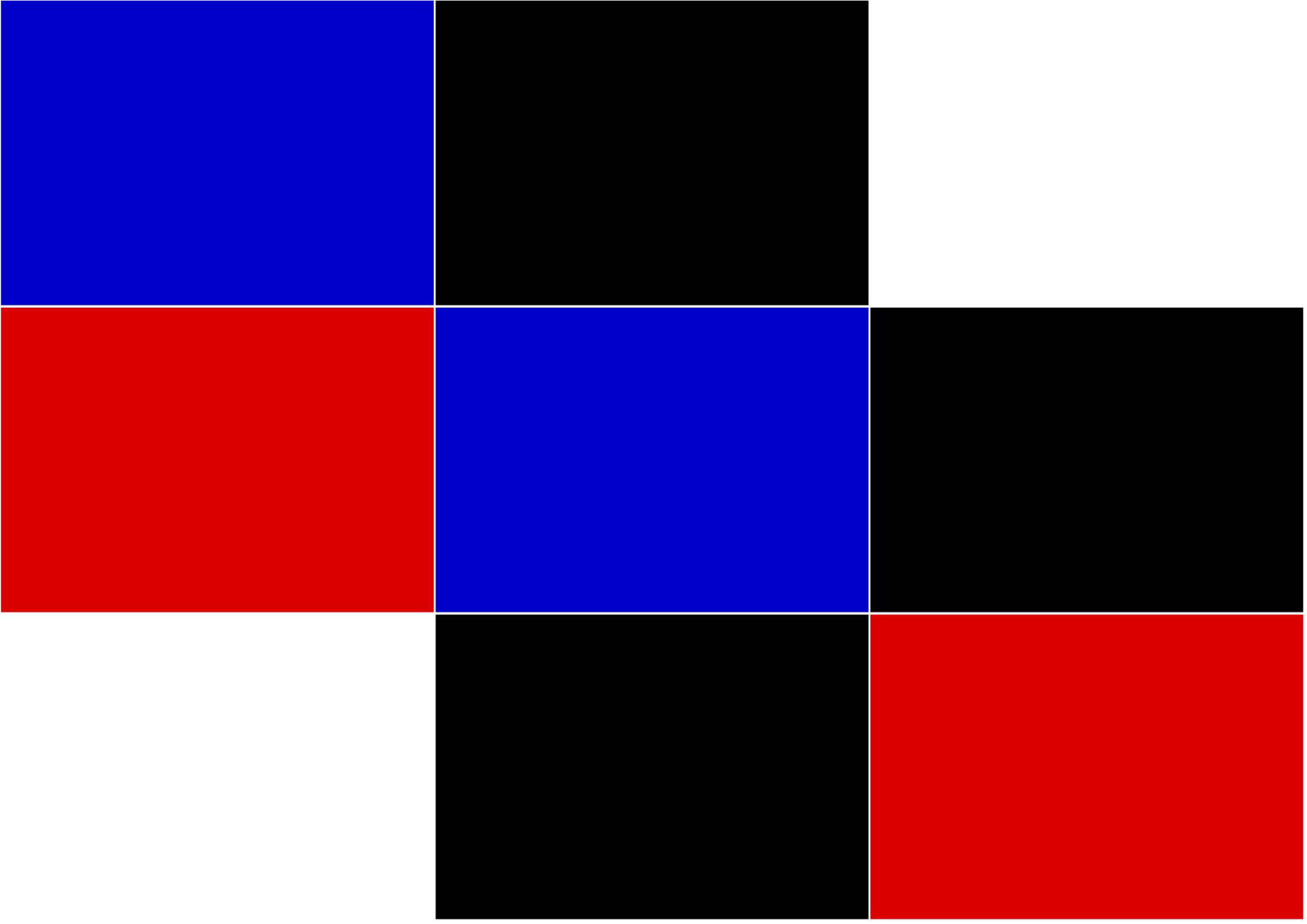


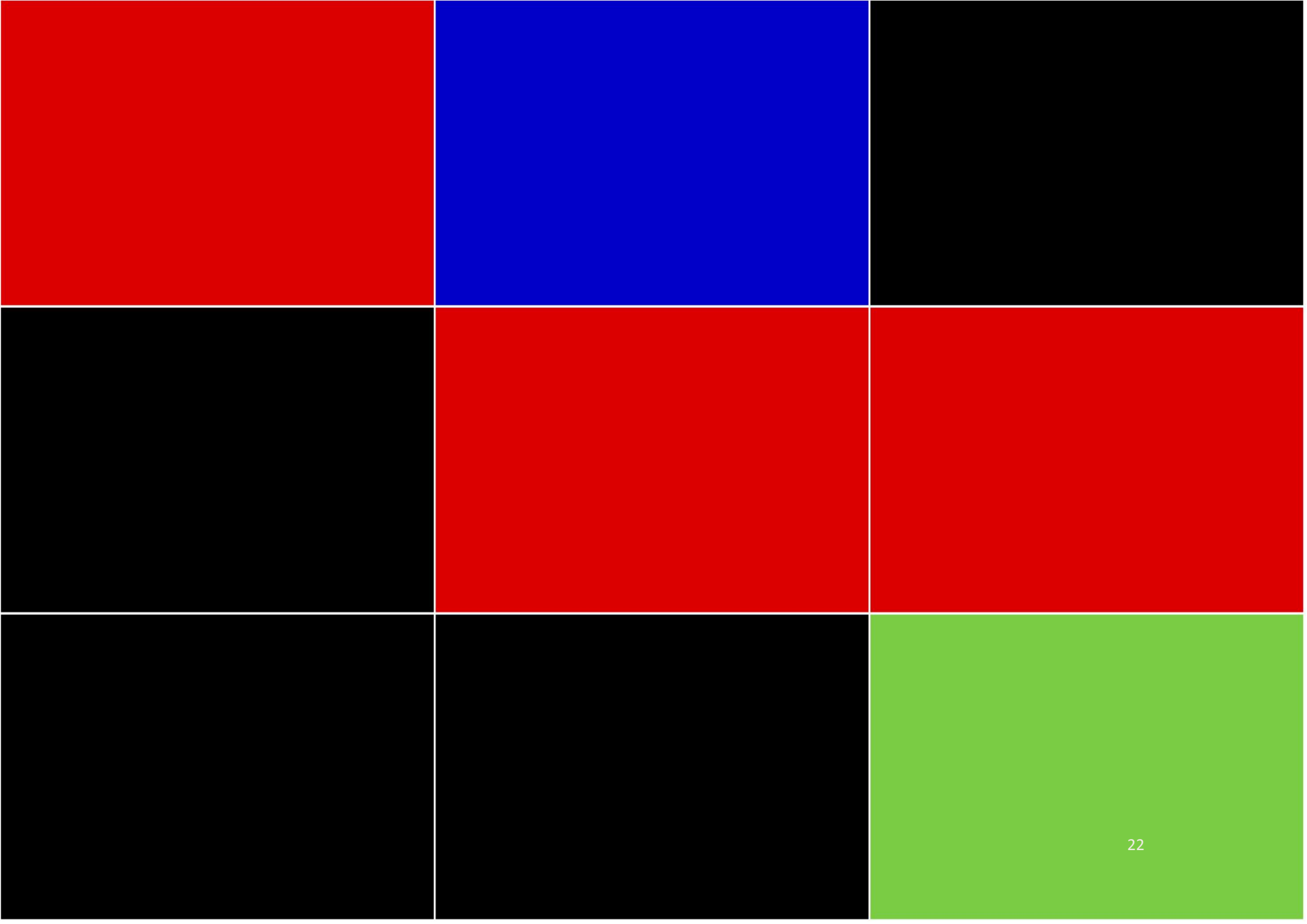


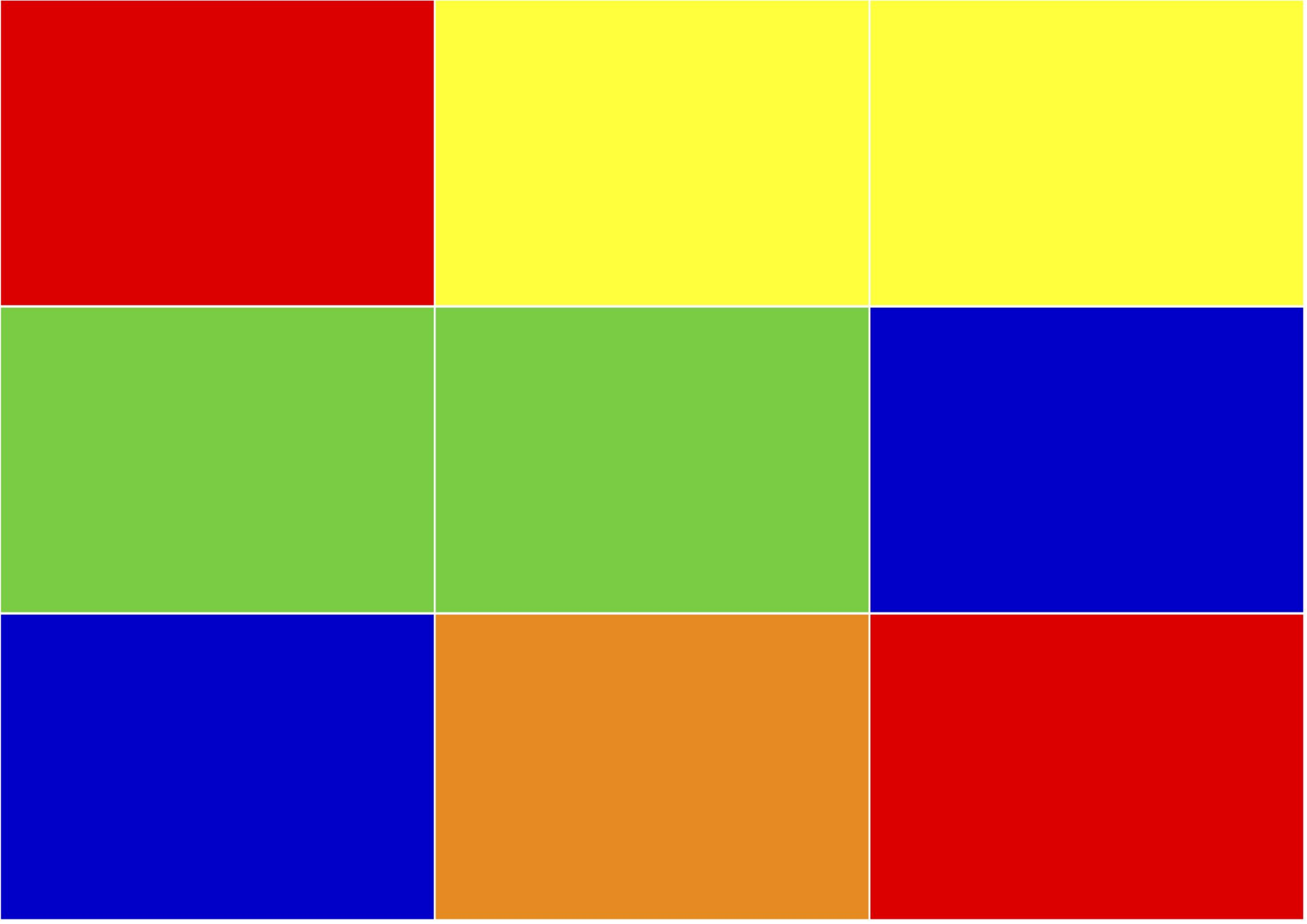


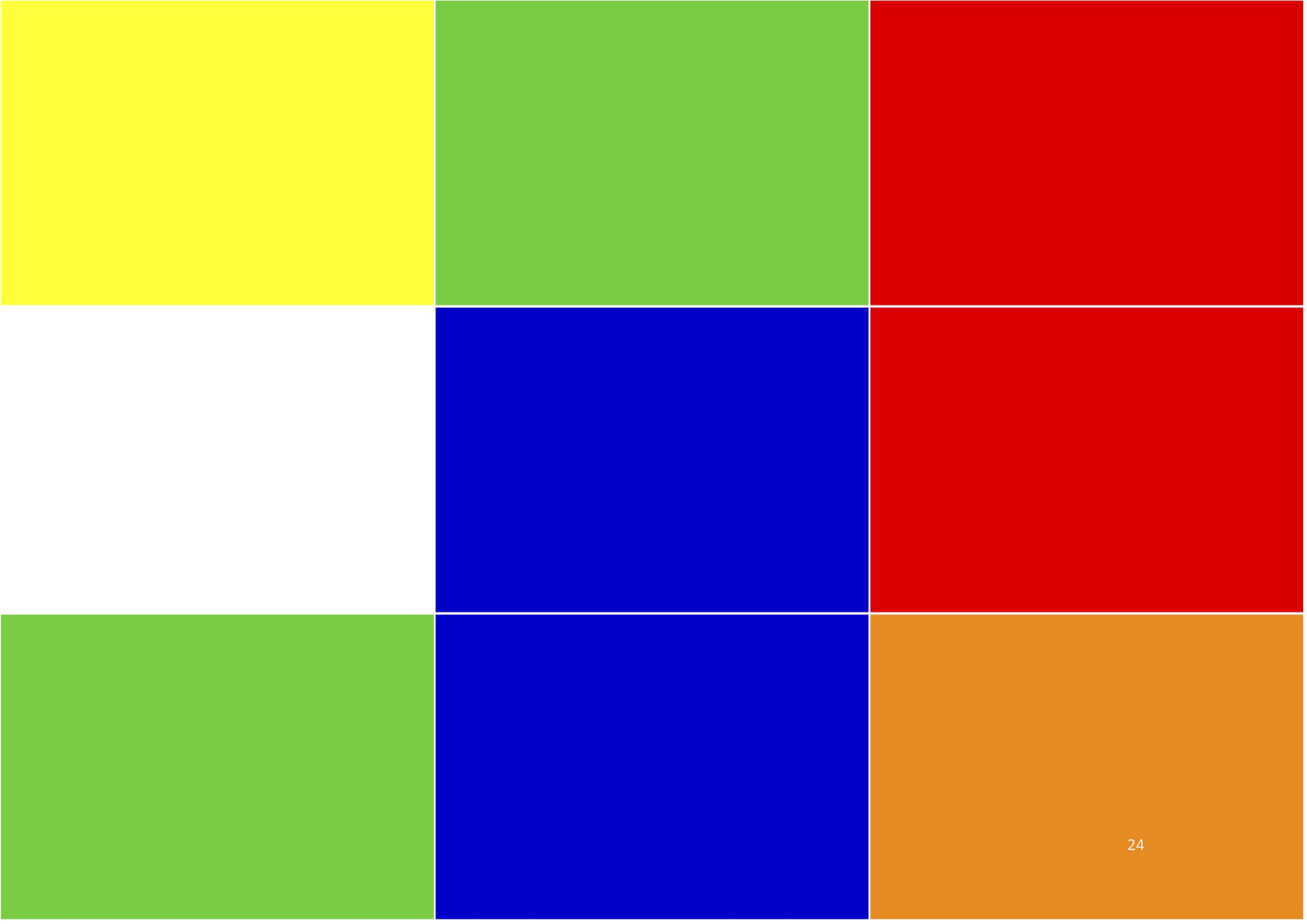


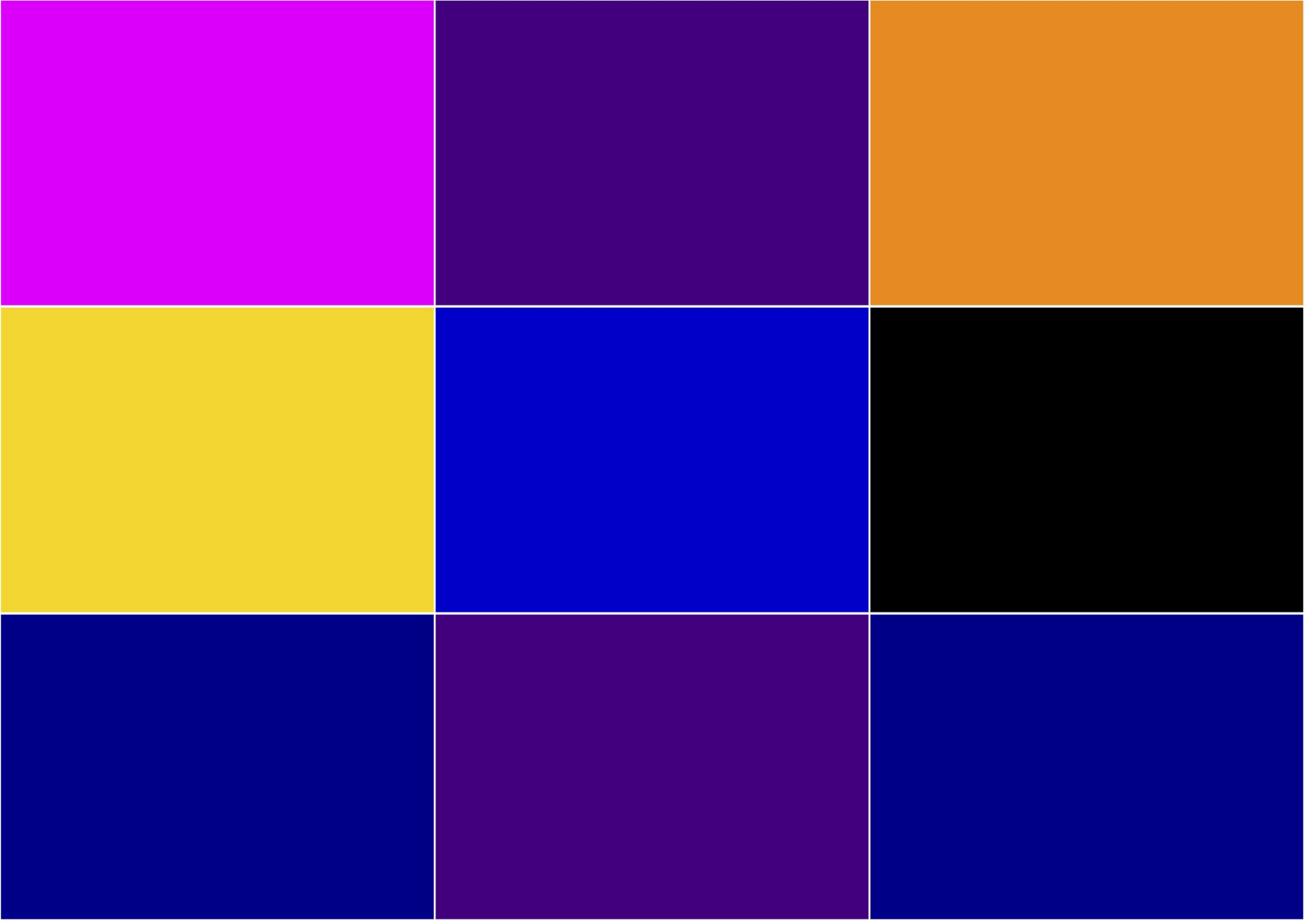


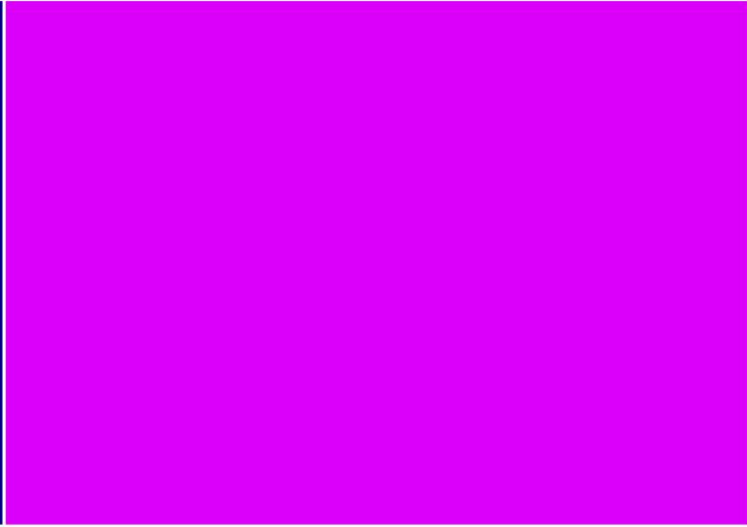
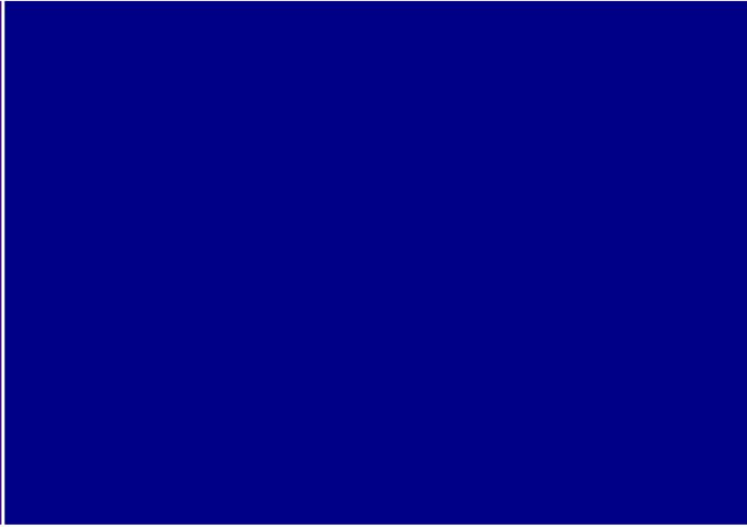
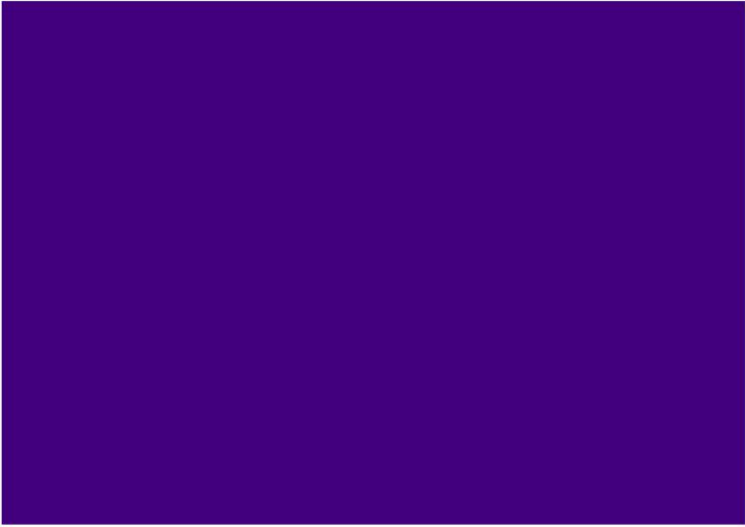


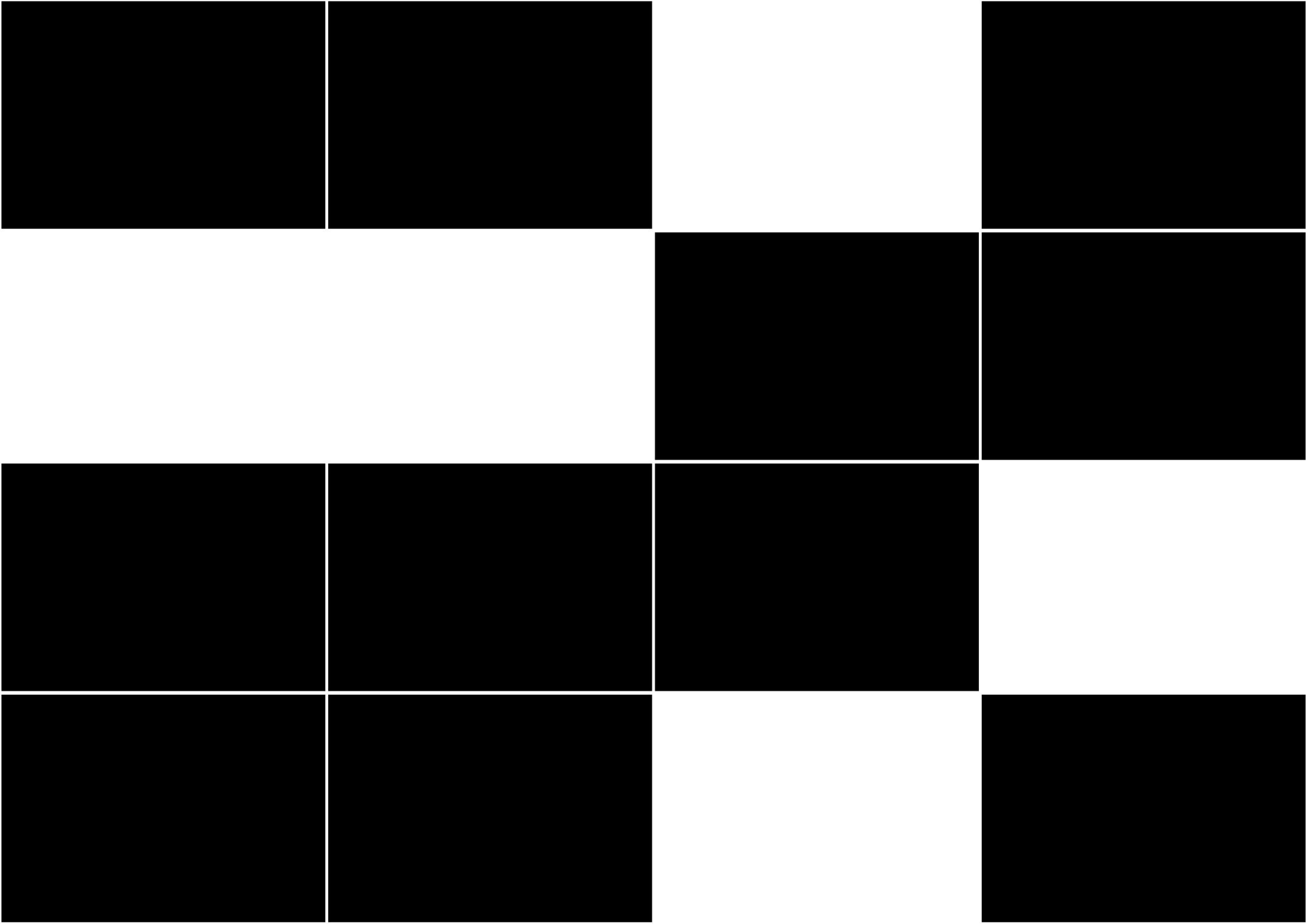


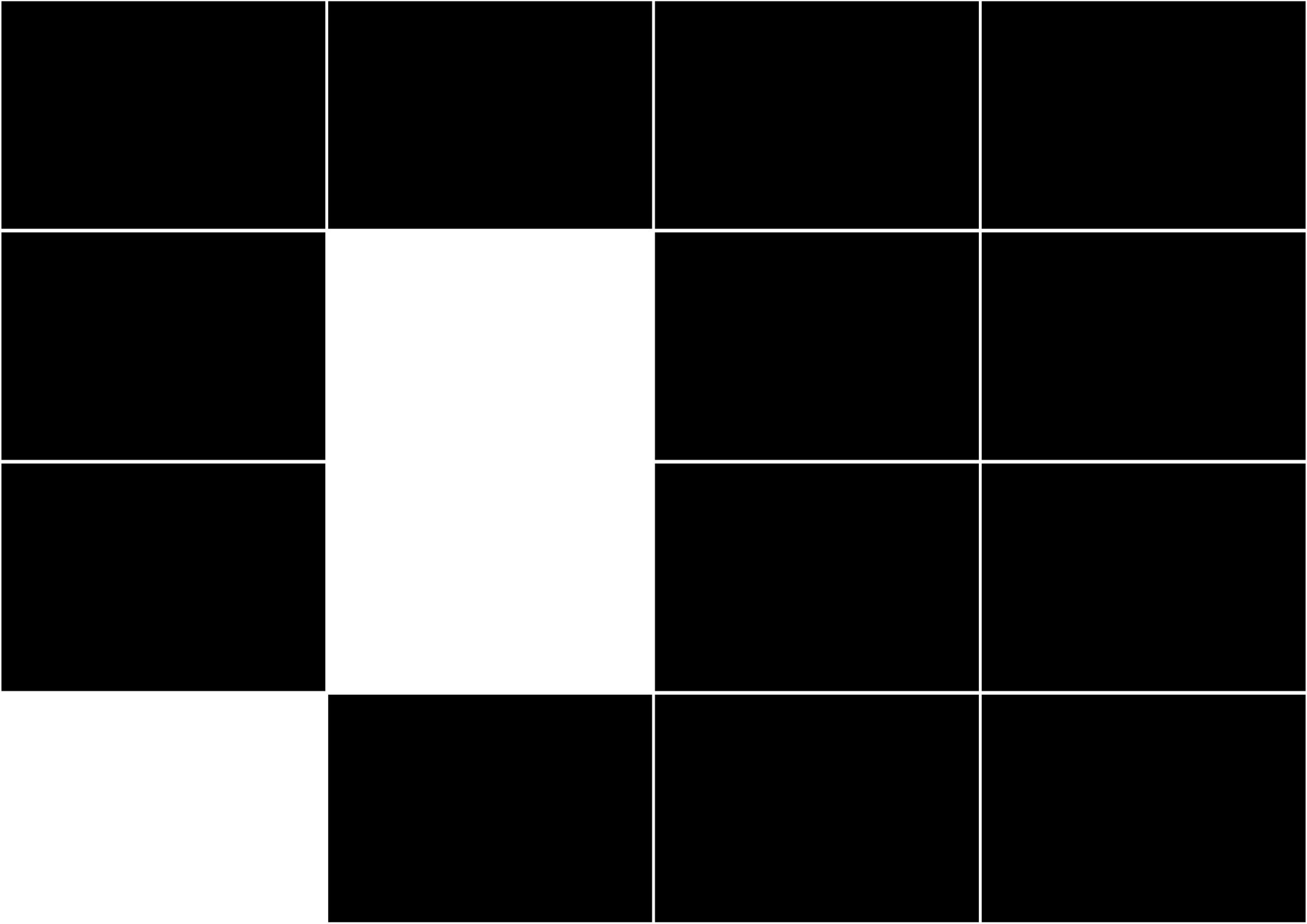


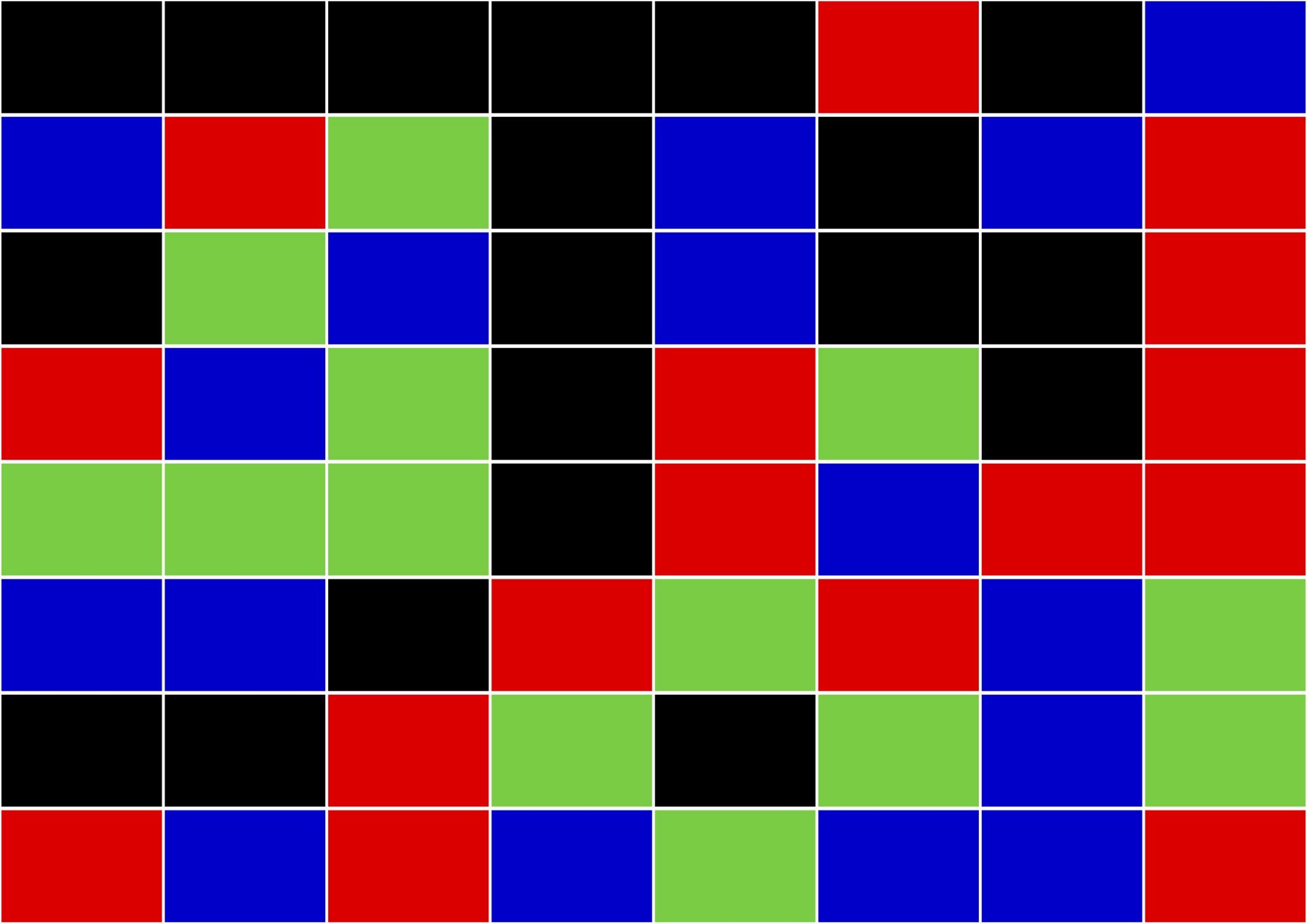


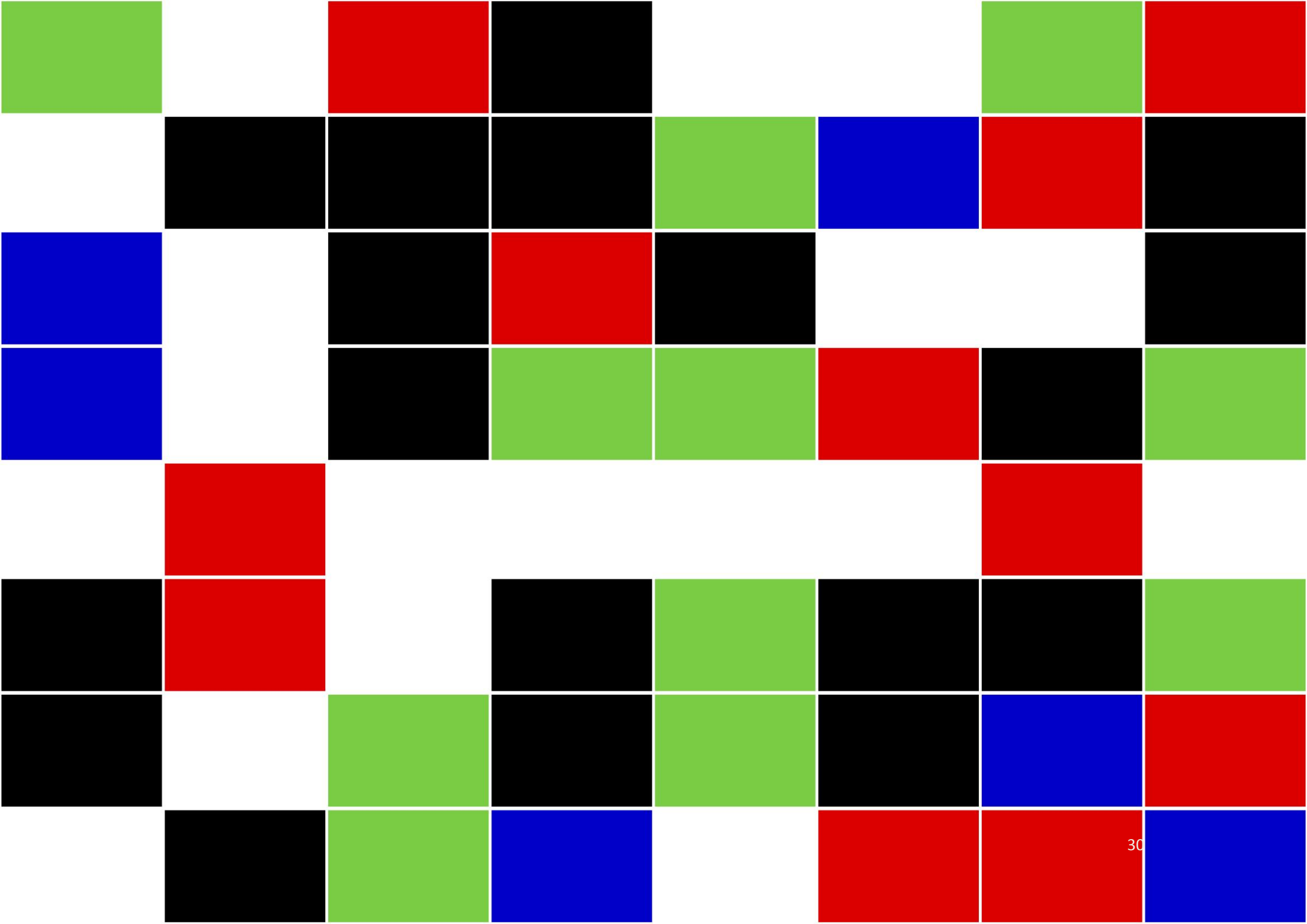


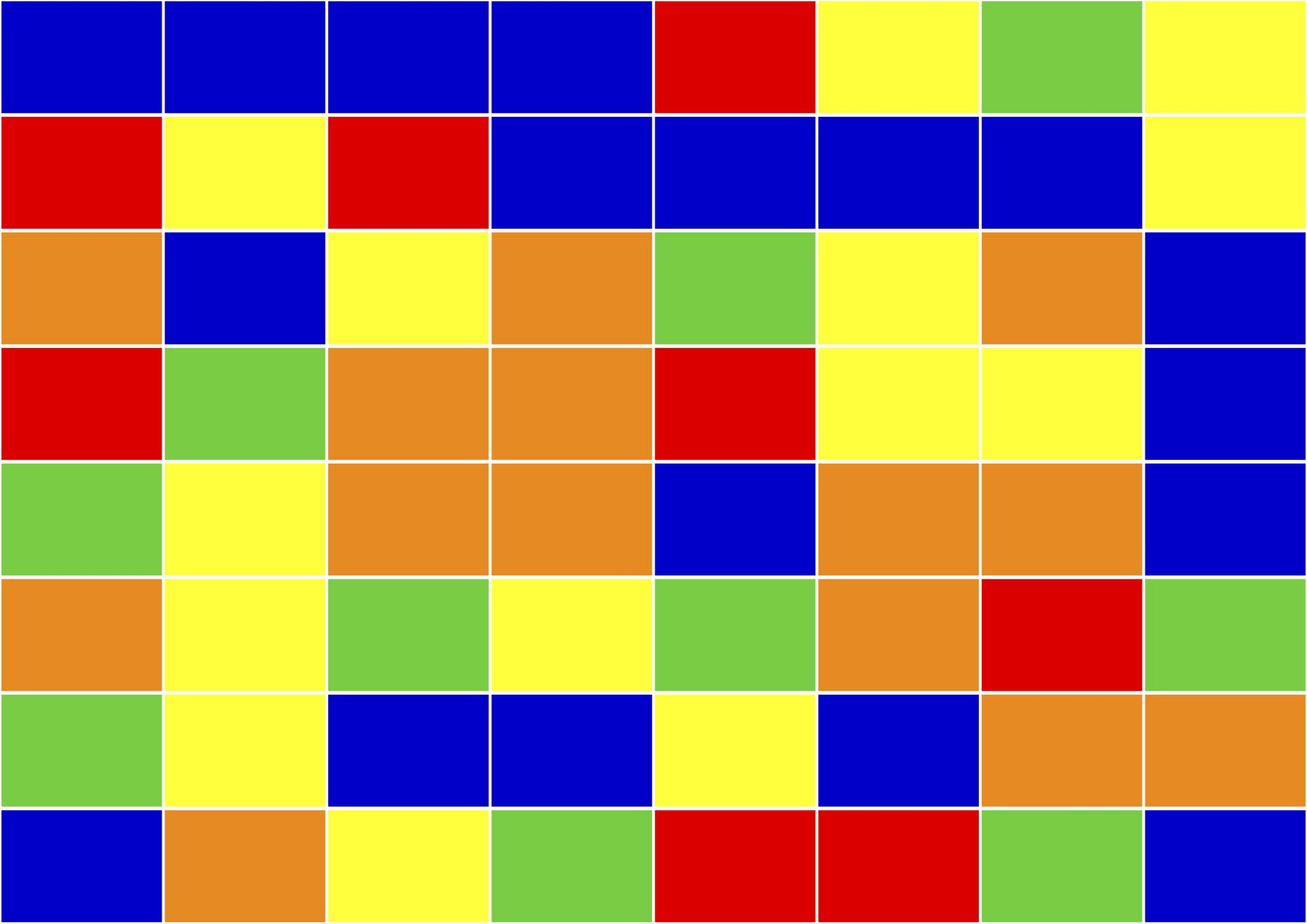




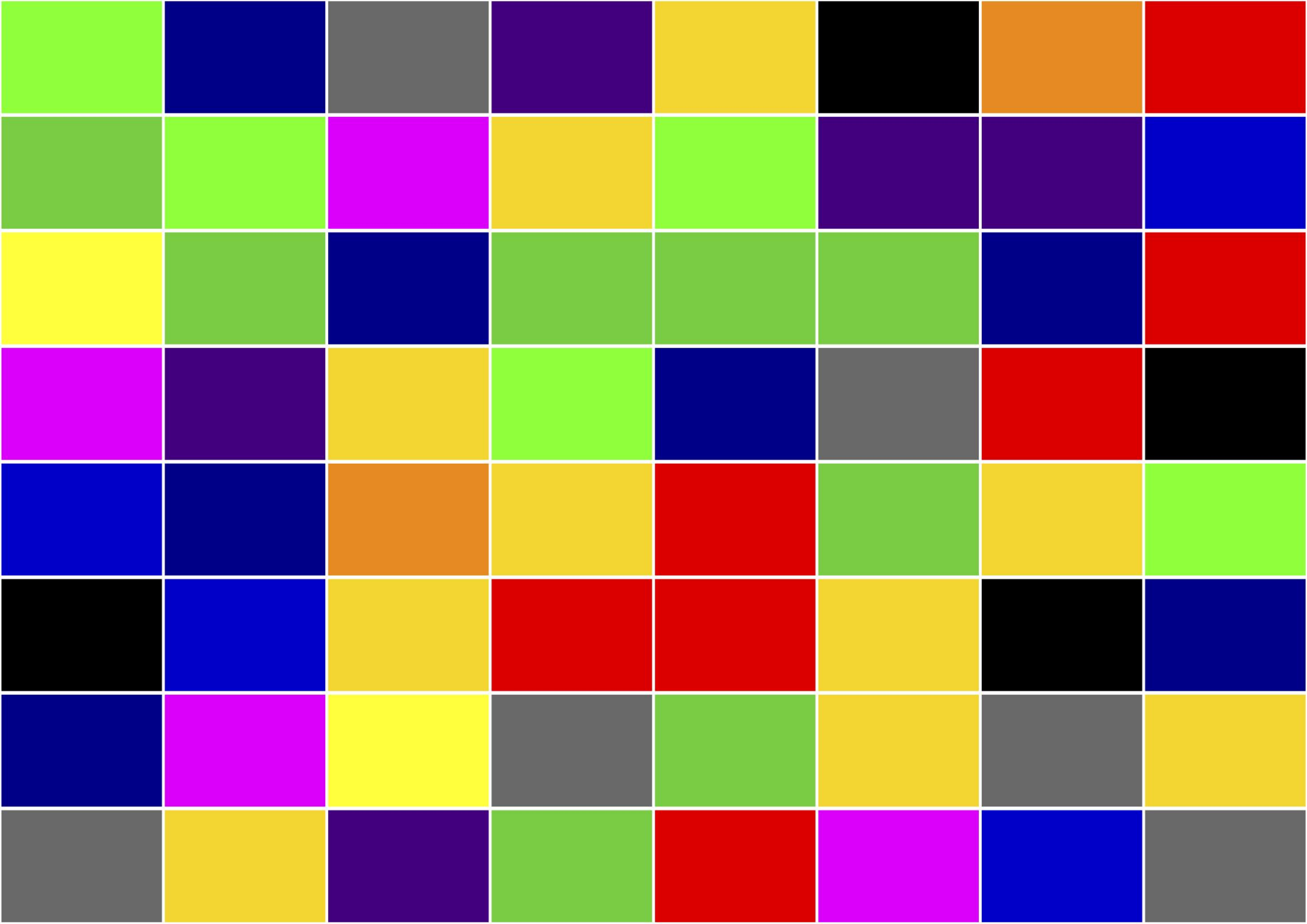


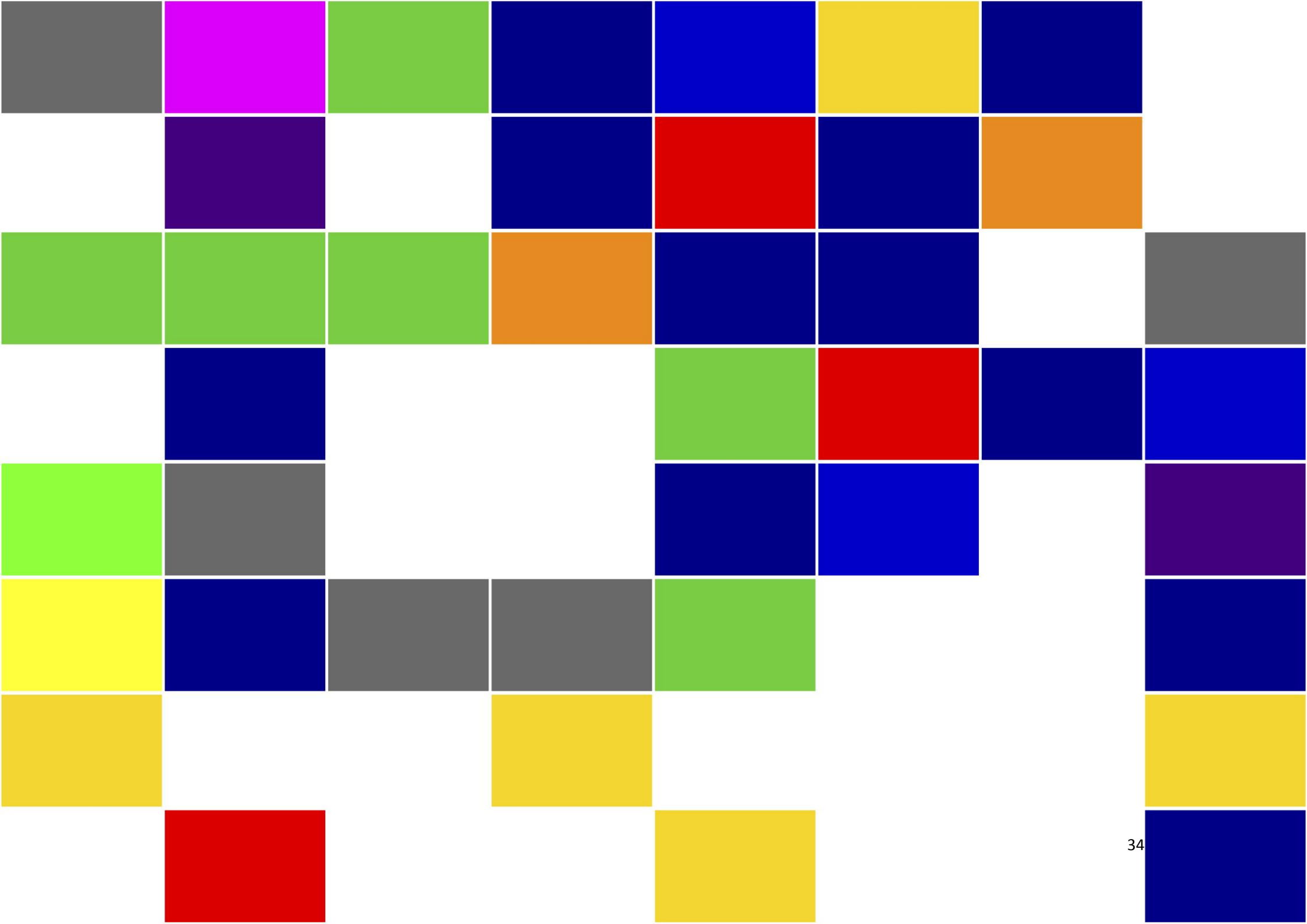


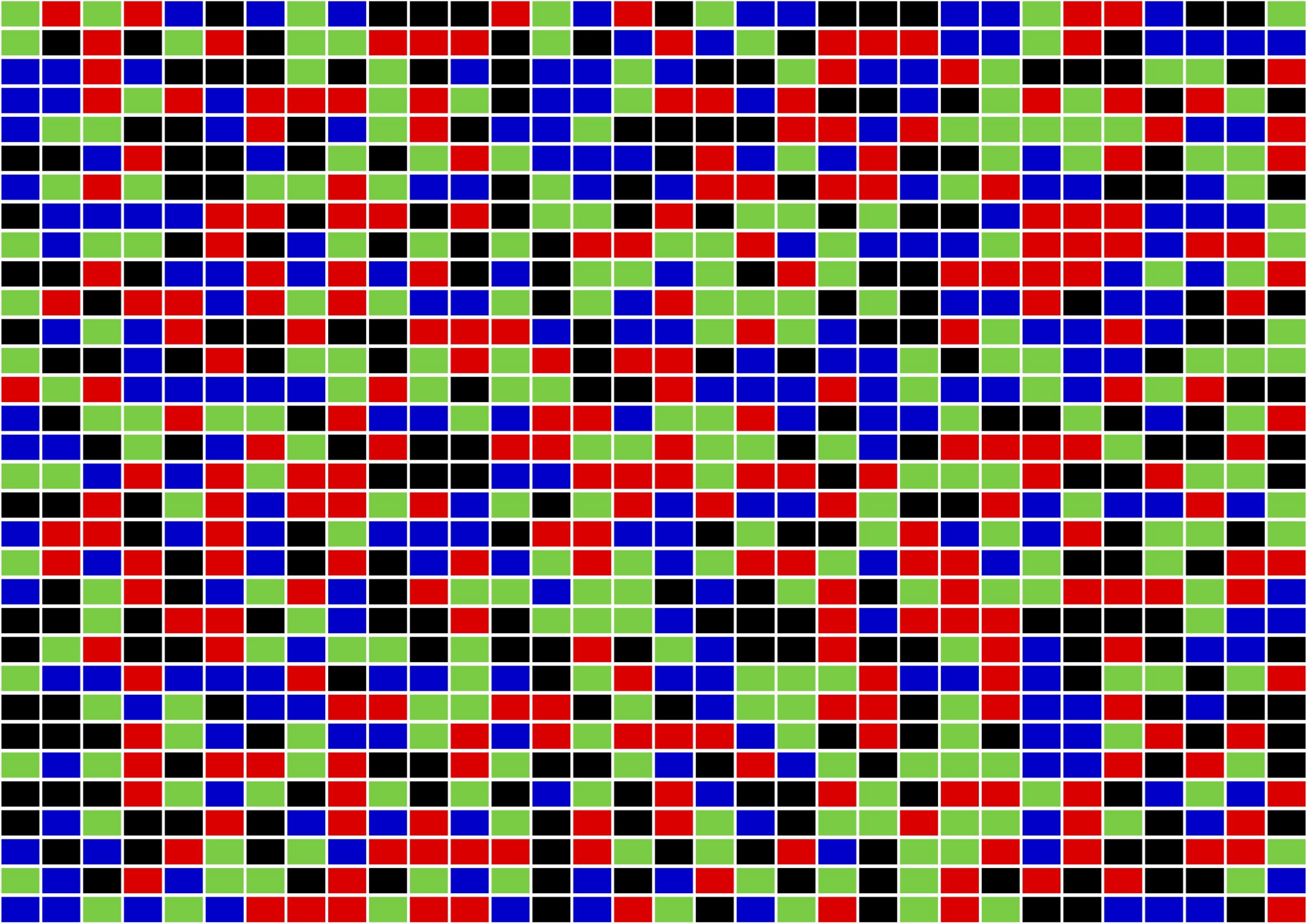


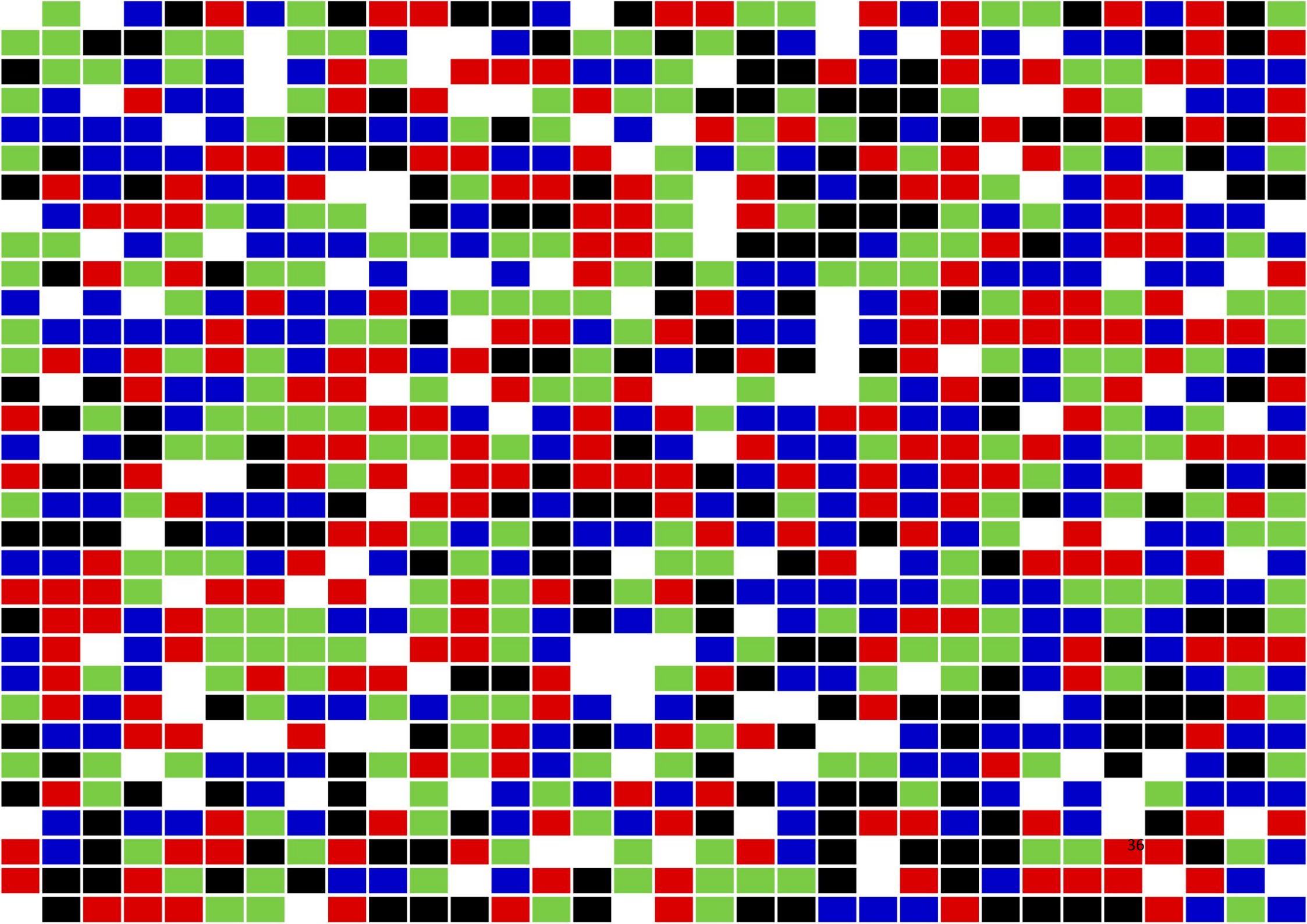


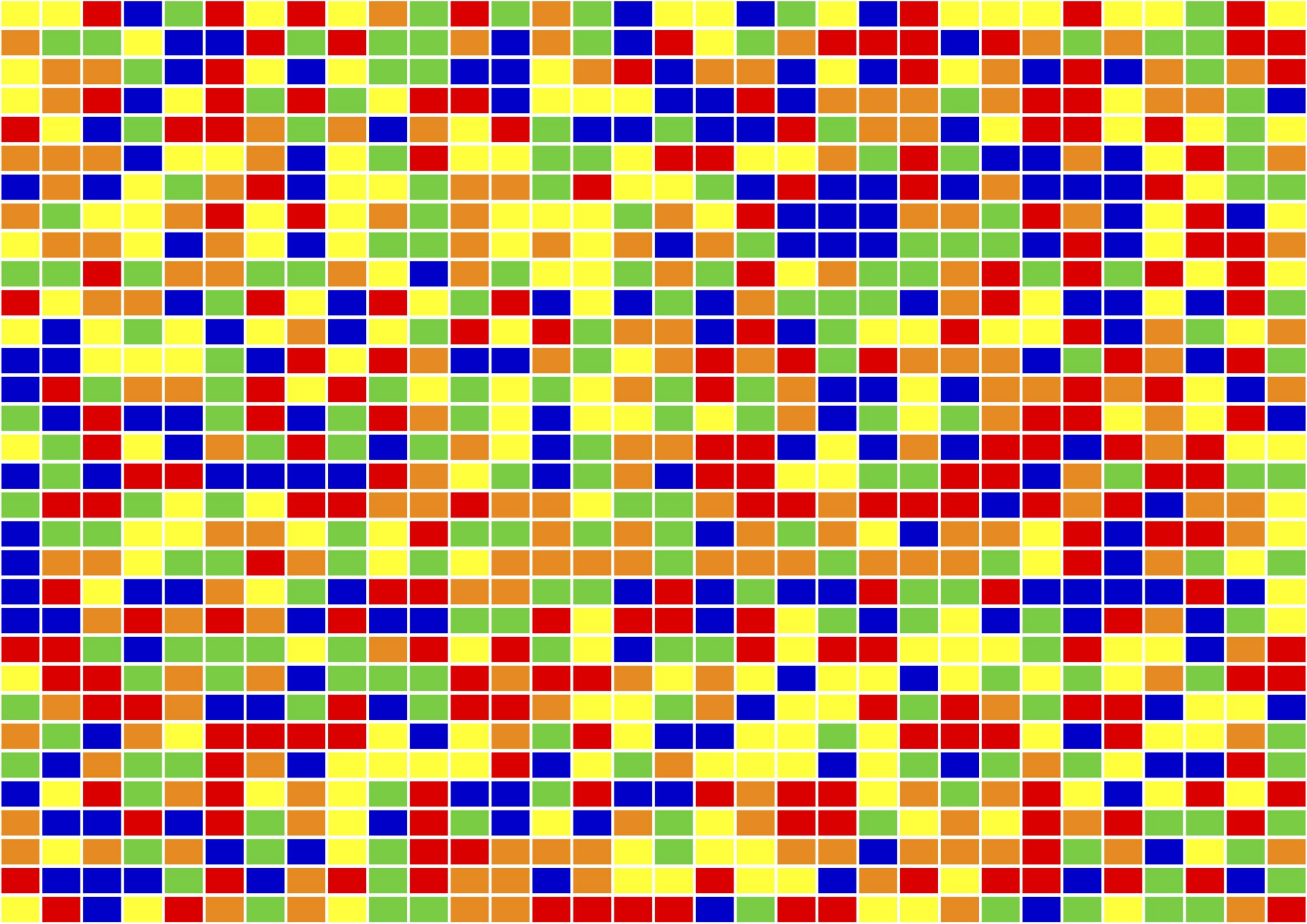


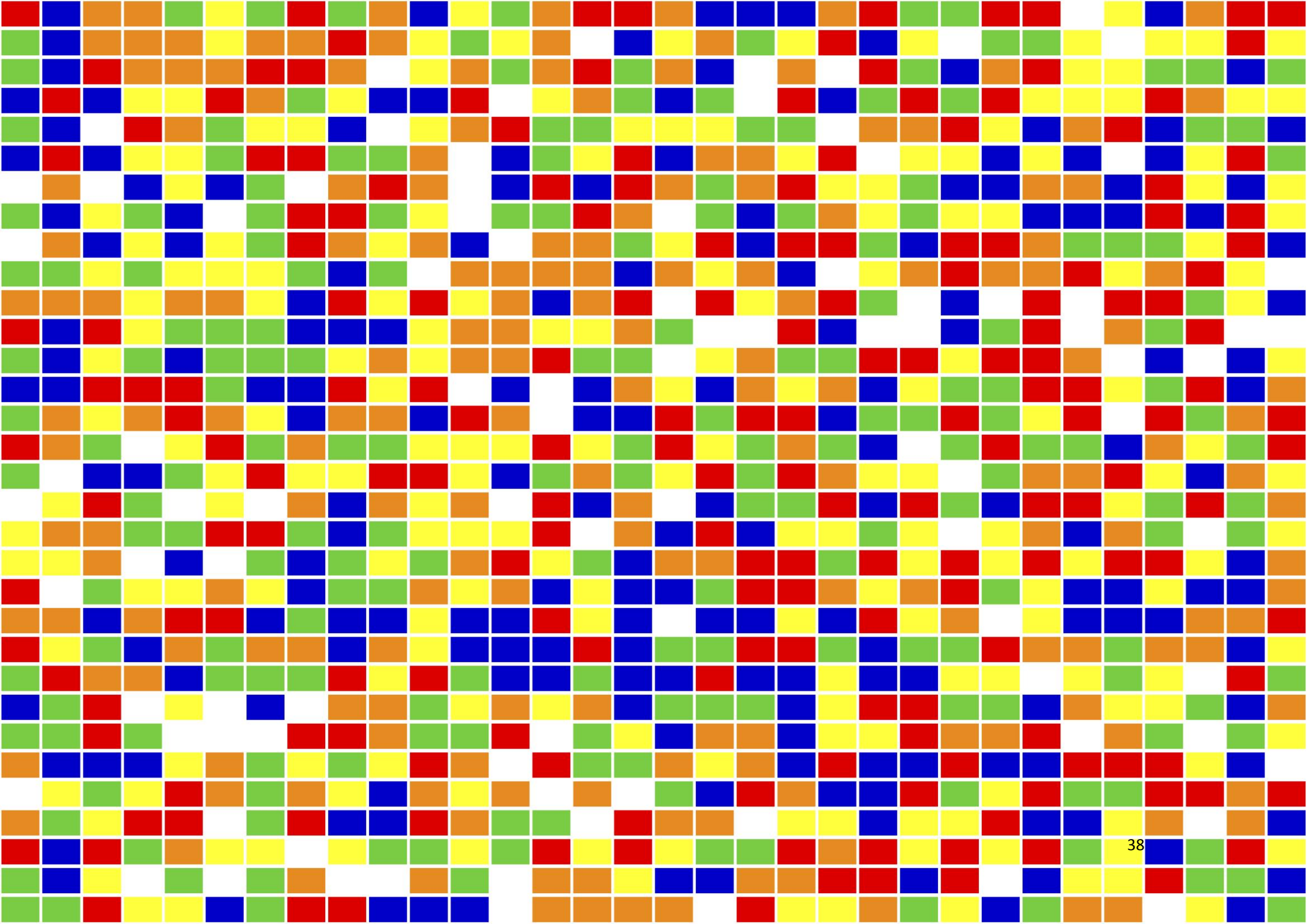


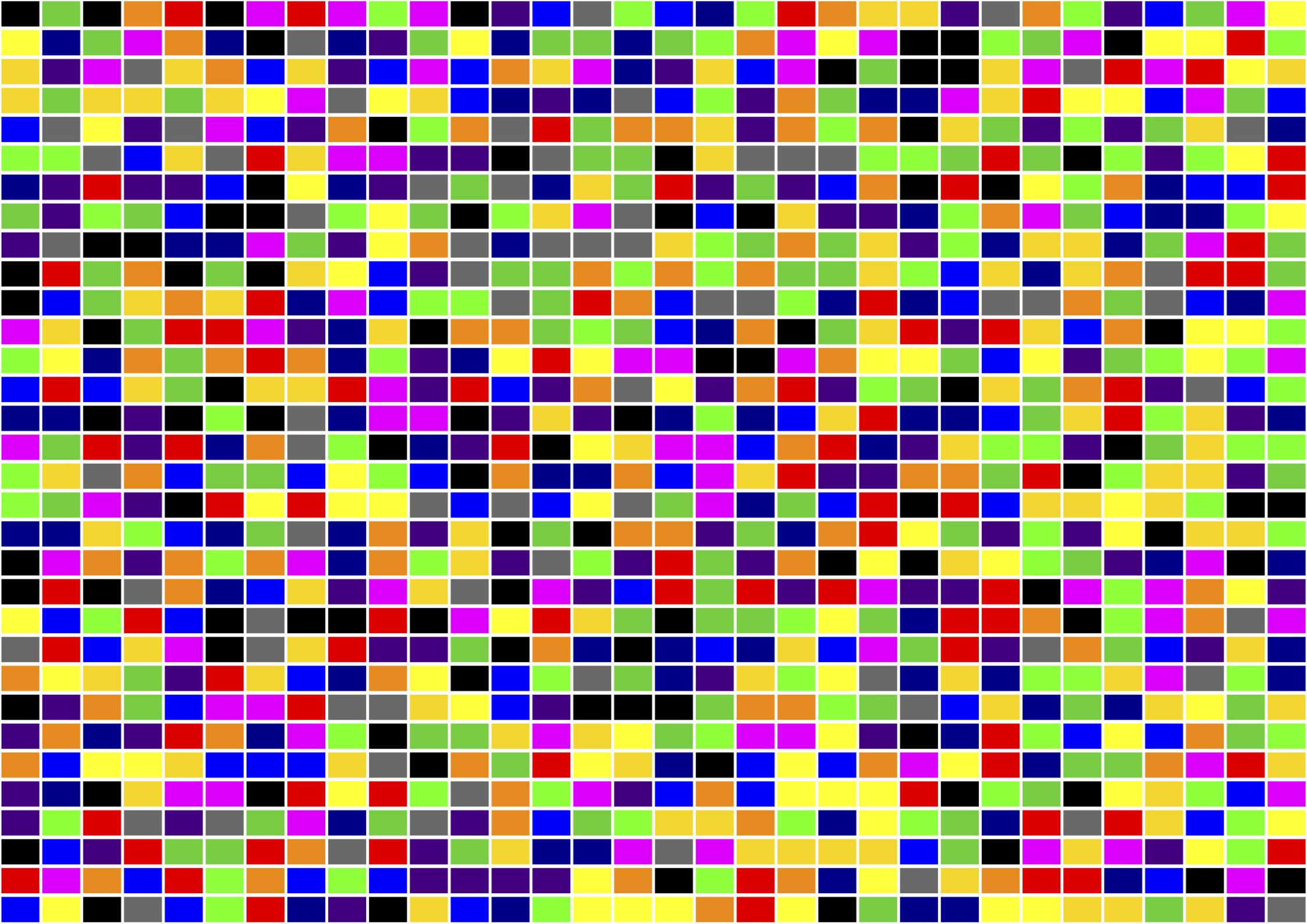


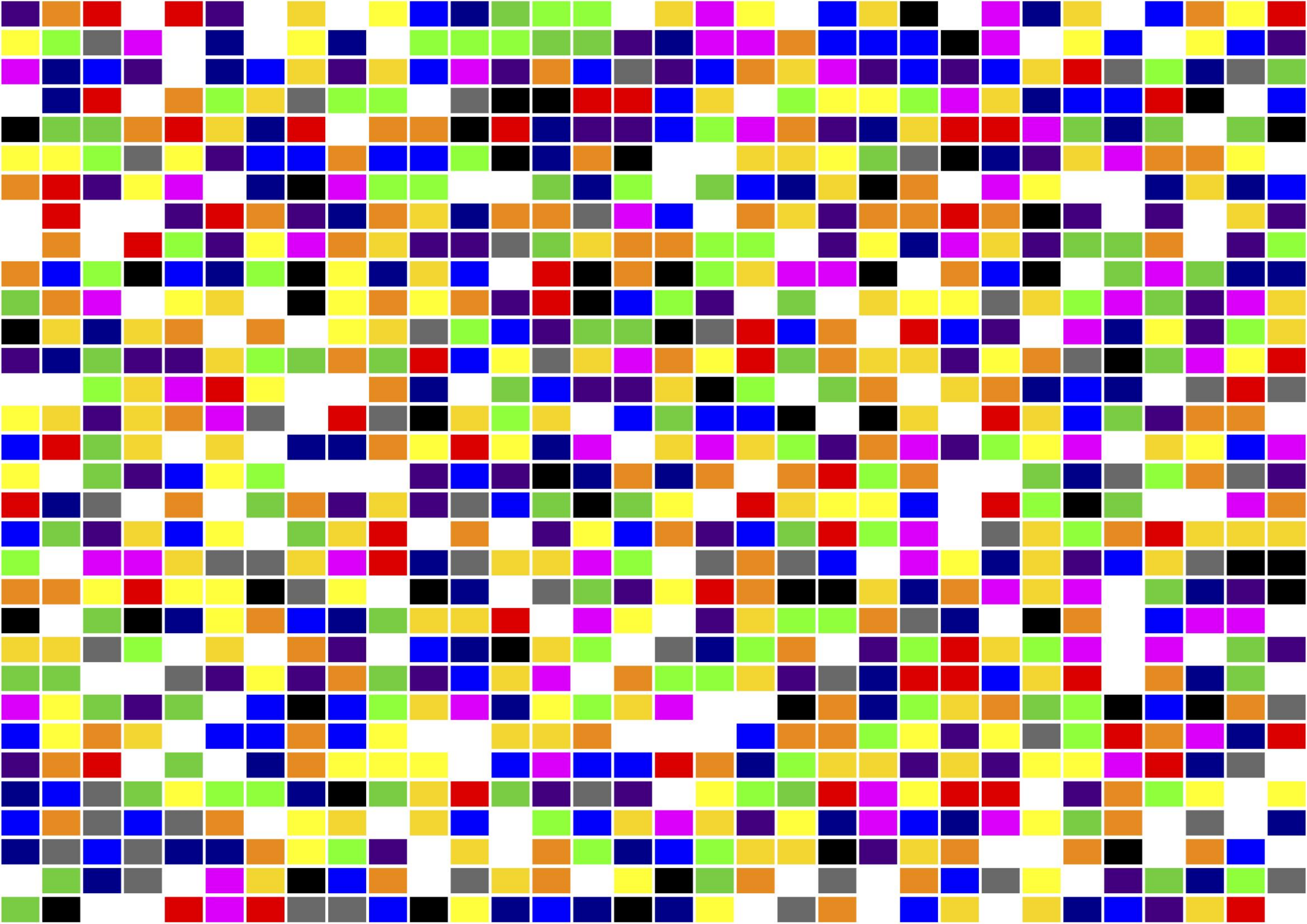


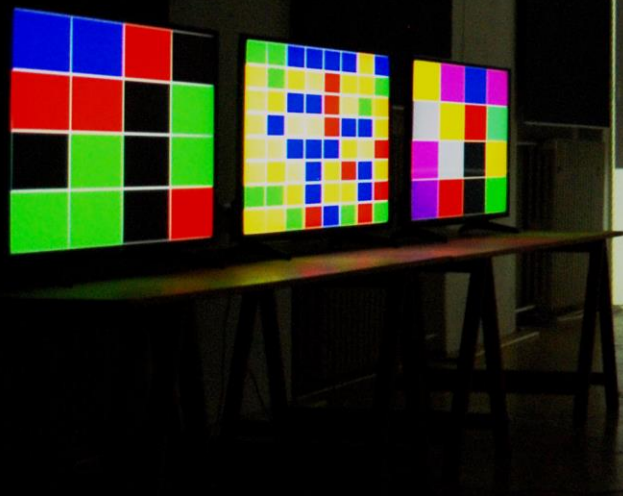






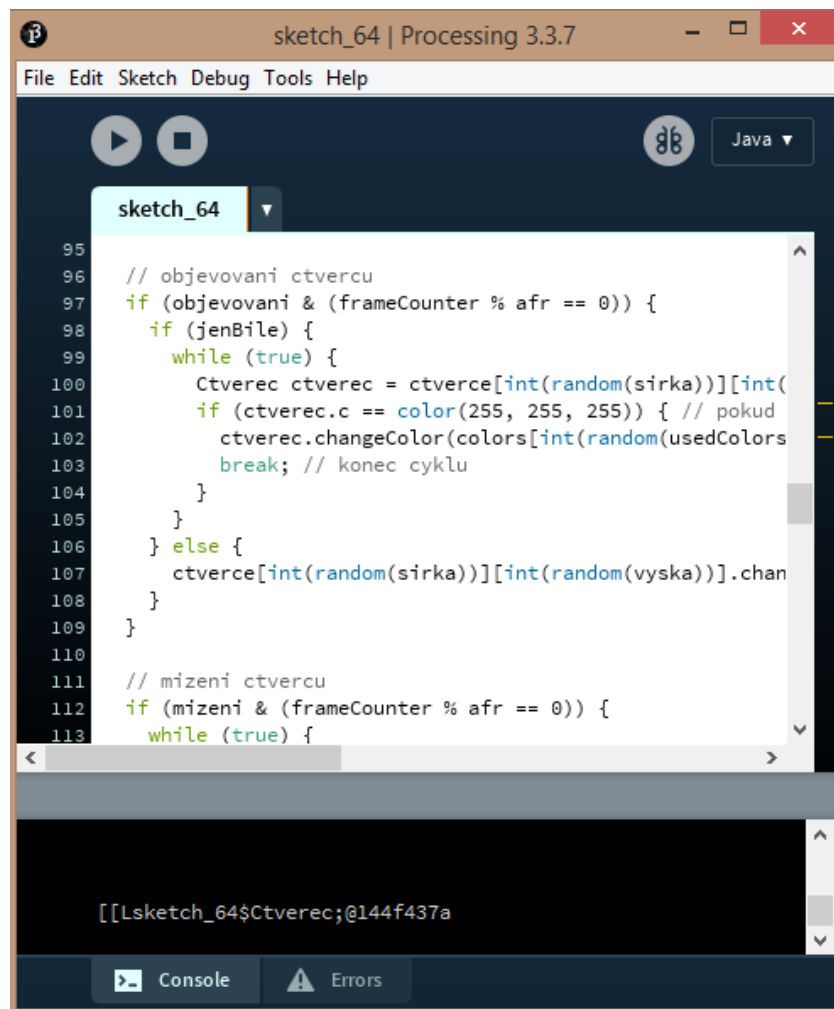












3/ TECHNICKÁ DOKUMENTACE

Práce je vytvořena v počítačovém programu Processing 3.0 pomocí programového jazyka JavaScript. Následně je pak umístěna na web. Výsledná varianta je na základě propojení s počítačem prezentována na LCD televizoru pro úplné vyniknutí barevných segmentů.

```
import processing.pdf.*;
```

```
boolean record;  
Ctverec[][] ctverce;  
color[] colors = new color[1024];  
color[] usedColors = new color[4];  
IntList colorIndexes = new IntList();  
int[] odskok = new int[2]; //odskok platna od kraje  
float[] velikost = new float[2]; //velikost ctverce  
float mezera = 3;  
// nesmi byt vice nez 1024 ctvecu  
int vyska = 4; // pocet ctvercu na vysku  
int sirka = 4; // pocet ctvercu na sirku  
color pozadi = color(255, 255, 255); // barva pozadi
```

```
// velikost okna se musi dopocitat rucne a vyplnit do size()  
// sirka okna = (sirka * velikost + mezera * (sirka + 1))  
// vyska okna = (vyska * velikost + mezera * (vyska + 1))
```

```
int mizeniPosun = 2; // kolik sekund po spusteni zacnou mizet ctverce  
int objevovaniPosun = 2; // kolik sekund po spusteni se zacnou ctverce objevovat,  
musi byt vetsi nez posun mizeni  
boolean mizeni = false;  
boolean objevovani = false;  
int ctvercuZaVterinu = 1; // kolik ctvercu za vterinu ma mizet/objevit se, max 60!  
int afr;  
int frameCounter = 0;  
boolean pause = false;  
boolean jenBile = false; // objevuji se jen bile(skryte) ctverce
```

```
void setup() {  
  fullScreen();  
  odskok[0] = odskok[1] = 0;  
  
  ctverce = new Ctverec[sirka][vyska];  
  frameRate(60);  
  afr = int(60/ ctvercuZaVterinu);
```

```
  if (objevovaniPosun <= mizeniPosun) { // pokud je posun objevovani mensi nebo  
    roven posunu mizeni, tak to opravi  
    objevovaniPosun = 2 * mizeniPosun;  
  }
```

```
  // vytvoreni indexu pro barvy  
  for (int i = 0; i < (sirka*vyska); i++) {  
    colorIndexes.append(i);  
  }  
  // zamichani indexu  
  colorIndexes.shuffle();
```

```
  //pouzite barvy  
  usedColors[0] = color(255,0,0);  
  usedColors[1] = color(0,128,0);  
  usedColors[2] = color(0,0,255);  
  usedColors[3] = color(0,0,0);
```

```
  // naplneni pole barev  
  int tmp = 0;  
  for (int a=0; a<(sirka*vyska); a++) {  
    colors[a] = usedColors[floor(random(usedColors.length))];  
  }
```

```
  velikost[0] = ((float)width/sirka) - mezera;  
  velikost[1] = ((float)height/vyska) - mezera;
```

```
  // naplneni pole ctvercu  
  for (int y = 0; y < vyska; y++) {  
    for (int x = 0; x < sirka; x++) {  
      ctverce[x][y] = new Ctverec(x, y, colors[colorIndexes.get(x + vyska*y)]);  
    }  
  }  
  print(ctverce);  
  noStroke();  
}
```

```

void draw() {
  if (record) {
    beginRecord(PDF, "frame-####.pdf");
  }
  background(pozadi);
  noStroke();

  // kontrola posunu mizeni a objevovani
  if (!mizeni) {
    if (frameCounter / frameRate > mizeniPosun) {

      mizeni = true;
    }
  }

  if (!objevovani) {
    if (frameCounter / frameRate > objevovaniPosun) {
      objevovani = true;
    }
  }

  // objevovani ctvercu
  if (objevovani & (frameCounter % afr == 0)) {
    if (jenBile) {
      while (true) {
        Ctverec ctverec = ctverce[int(random(sirka))][int(random(vyska))];
        if (ctverec.c == color(255, 255, 255)) { // pokud ma nahodny ctverec bilou barvu
          ctverec.changeColor(colors[int(random(usedColors.length))]);
          break; // konec cyklu
        }
      }
    }
    else {

      ctverce[int(random(sirka))][int(random(vyska))].changeColor(colors[int(random(usedColors.length))]); // vymeni barvu nahodnemu ctverci
    }
  }
}

```

```

// mizeni ctvercu
if (mizeni & (frameCounter % afr == 0)) {
  while (true) {
    Ctverec ctverec = ctverce[int(random(sirka))][int(random(vyska))];
    if (ctverec.c != color(255, 255, 255)) { // pokud nema nahodny ctverec bilou
      barvu
      ctverec.changeColor(color(255, 255, 255));
      break; // konec cyklu
    }
  }

  boolean allWhite = true;
  for (int y = 0; y < vyska; y++) {

    for (int x = 0; x < sirka; x++) {

      if(ctverce[x][y].c != color(255, 255, 255)) {
        allWhite = false;
        break;
      }
    }
    if (allWhite == false) {
      break;
    }
  }

  if(allWhite){
    for (int a=0; a<(sirka*vyska); a++) {
      colors[a] = usedColors[floor(random(usedColors.length))];
    }
    colorIndexes.shuffle(); // zamichani barev
    for (int y = 0; y < vyska; y++) {
      for (int x = 0; x < sirka; x++) {
        ctverce[x][y].changeColor(colors[colorIndexes.get(x + vyska*y)]);
      }
    }
  }
}
}

```

```

// zobrazeni ctvercu
for (int y = 0; y < vyska; y++) {
    for (int x = 0; x < sirka; x++) {
        ctverce[x][y].display();
    }
}

frameCounter++;

if (record) {
    endRecord();
    record = false;
}
}

void keyPressed() {
    if (keyCode == 'S' || keyCode == 's') {
        record = true;
    }
}

// kontrola stisknuti klavesy

void keyReleased() {
    if (key == 'p' || key == 'P') { // po stisknuti p
        if (pause) {
            pause = false;
            loop();
        } else {
            pause = true;
            noLoop();
        }
    }
}

void mouseClicked() {
    if (mouseButton == LEFT) {
        colorIndexes.shuffle(); // zamichani barev
        for (int y = 0; y < vyska; y++) {
            for (int x = 0; x < sirka; x++) {
                ctverce[x][y].changeColor(colors[colorIndexes.get(x + vyska*y)]);
            }
        }
    }
}

```

```

}
    frameCounter = 0; // vynulovani pocitadla
    mizeni = false; // zruseni mizeni
    objevovani = false; // zruseni objevovani
}
}

class Ctverec {
    int x, y; // pozice ctverce
    color c; // barva ctverce

    Ctverec ( int xTmp, int yTmp, color cTemp) {
        x = xTmp;
        y = yTmp;
        c = cTemp;
    }

    // zmeni barvu ctverce
    void changeColor(color cTemp) {
        c = cTemp;
    }

    // vykresli ctverec na platno
    void display() {
        pushStyle();

        pushMatrix();

        fill(c);
        translate((x * velikost[0] + x*mezera + mezera/2), (y * velikost[1] + y*mezera +
        mezera/2));
        rect(0, 0, velikost[0], velikost[1]);
        popMatrix();
        popStyle();
    }
}

```

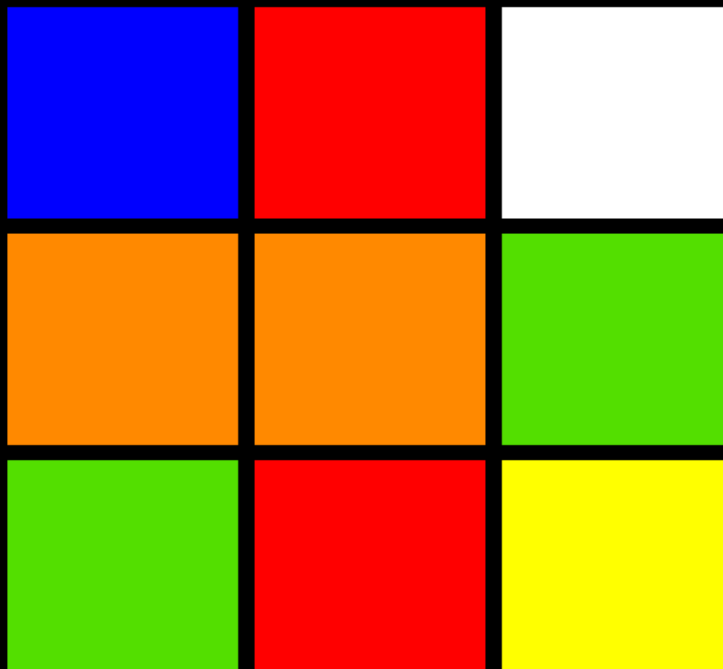

4/ PORTFOLIO

1974

NET ART

2014

Net art 1974 vychází z Rubikovy kostky. Uživatel pomocí klikání myši zachází s náhodou, jež generuje barvy – žlutou, červenou, modrou, zelenou, bílou a oranžovou.

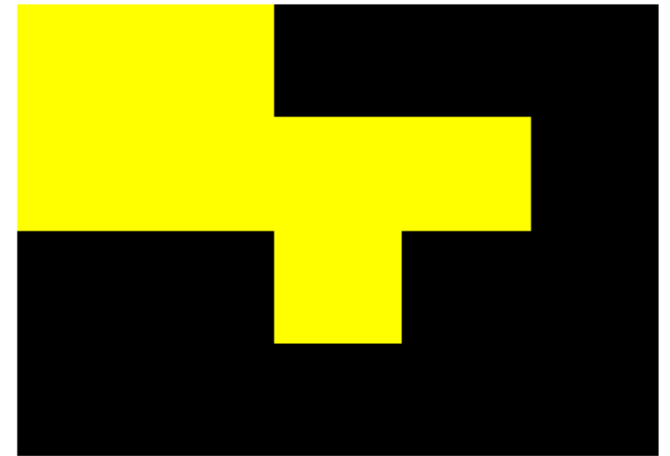
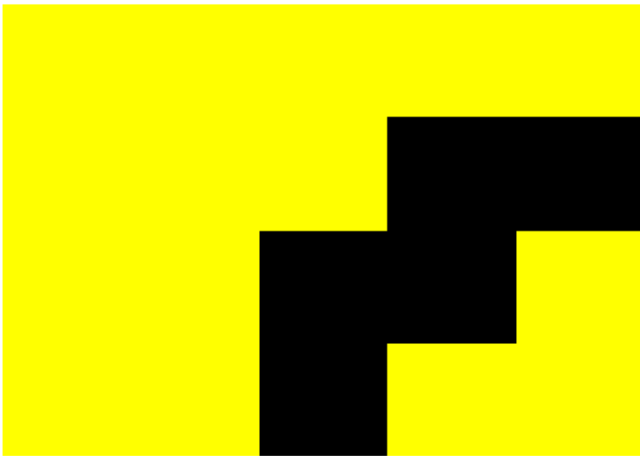


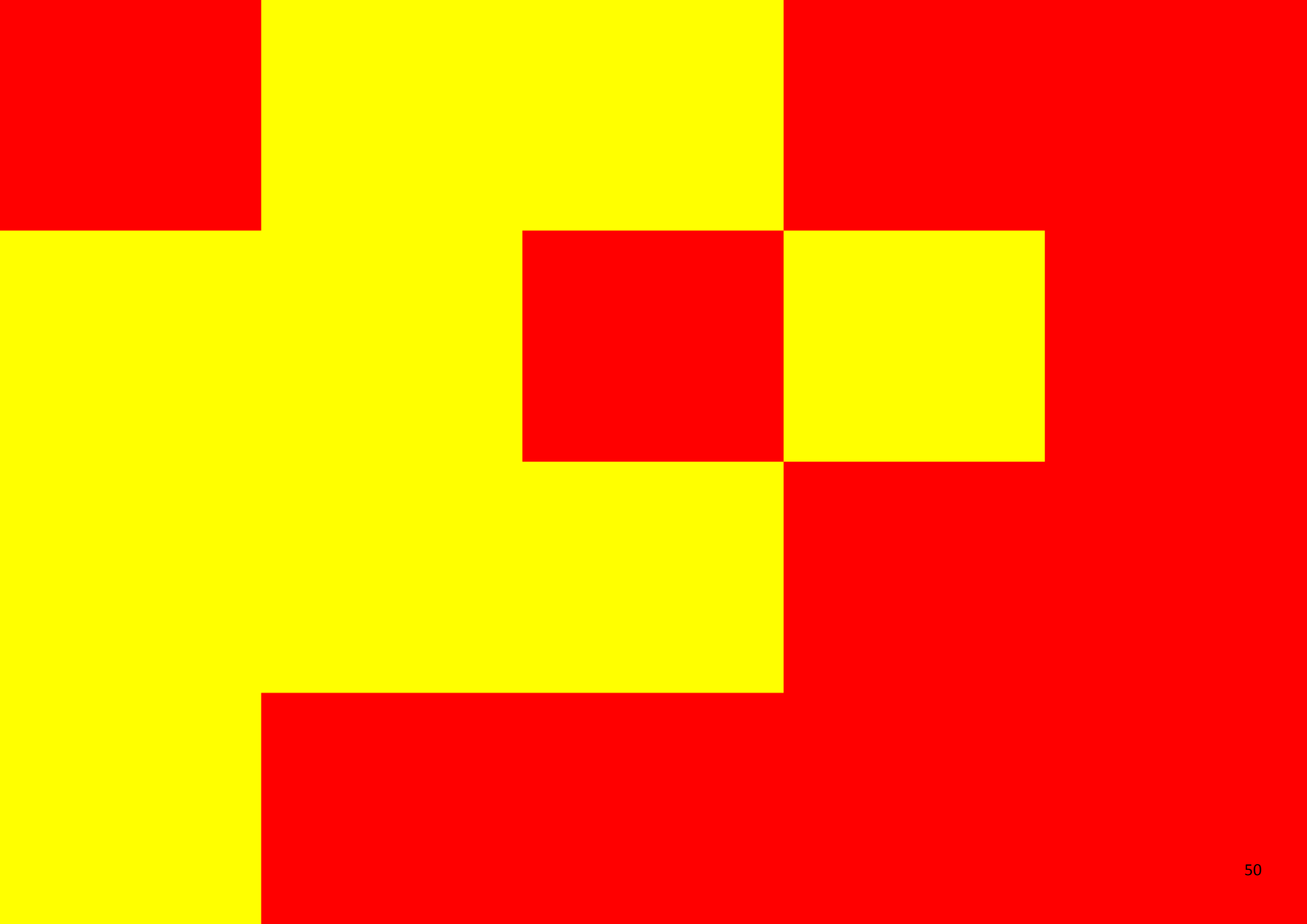
COM.POZICE

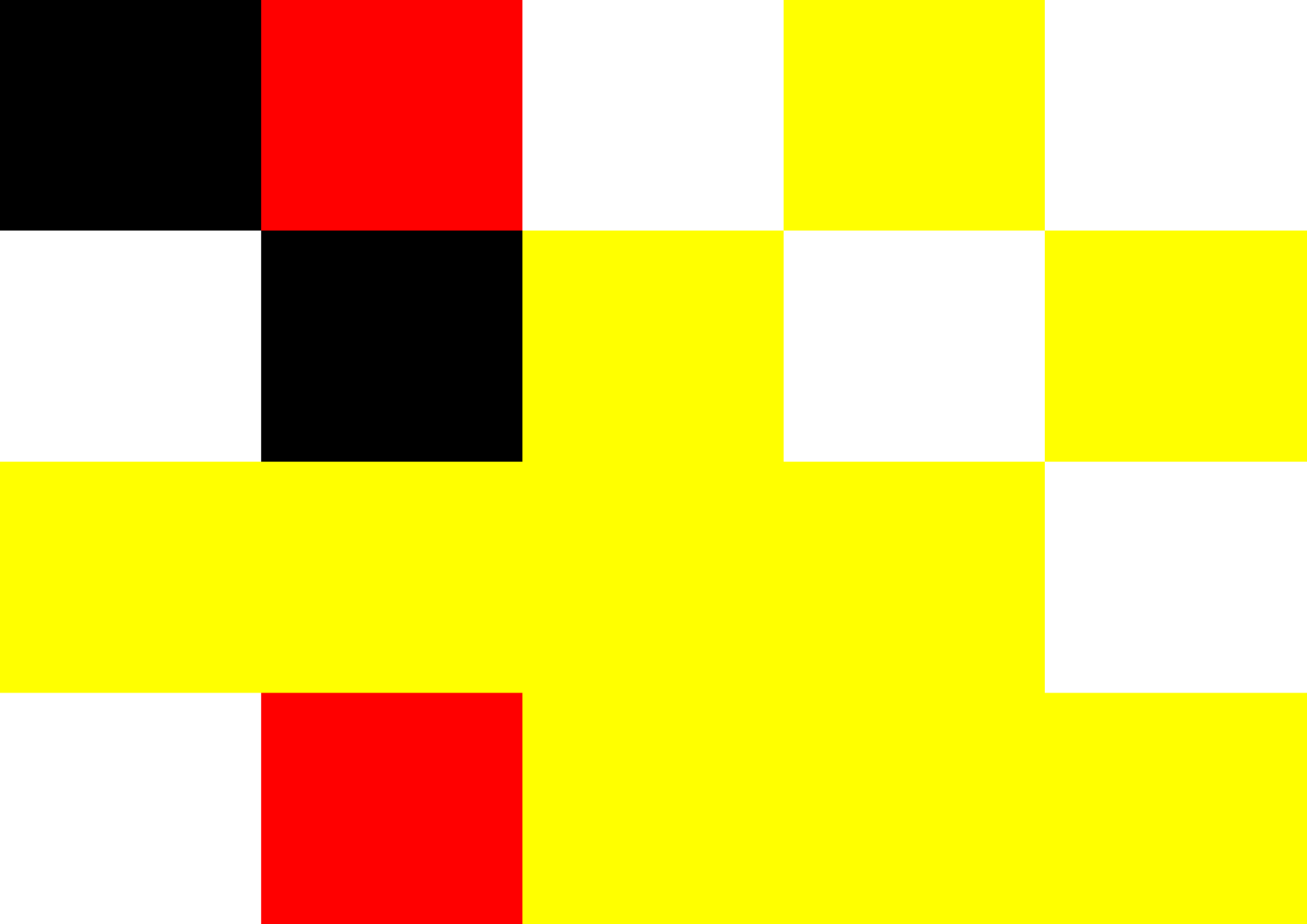
webová aplikace
2015

Webová aplikace com.pozice vychází z Net Artu 1974. Princip klikání a náhodné změny barev s obrovským počtem kombinací přeměňuje statickou formu díla a rozšiřuje tak obzory diváka.

Odchýlení se od pravidelných čtverců a množství barev vedlo ke vzniku jednodušších, ale pozoruhodných variant této aplikace.
Variabilita náhodného generování posunula mé schopnosti v JavaScriptu.



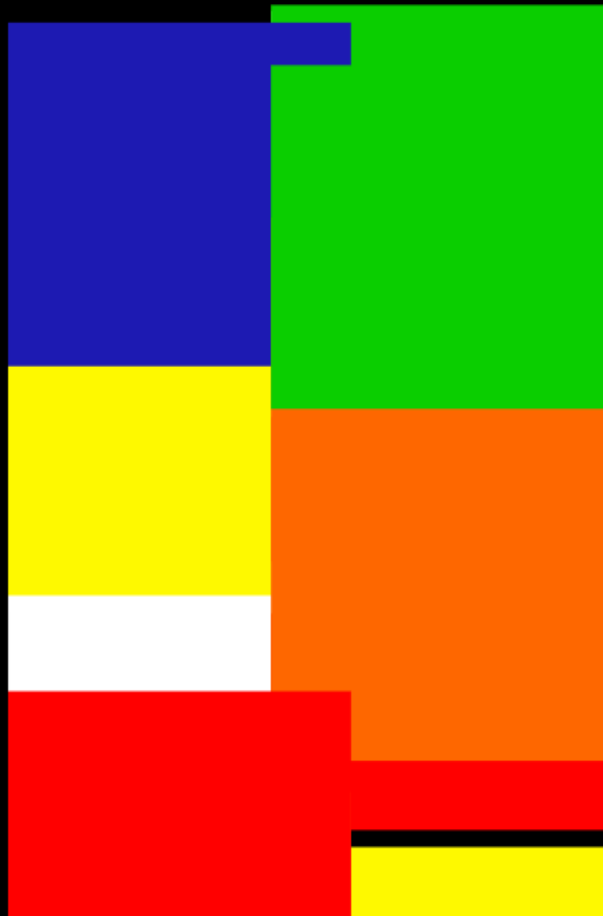


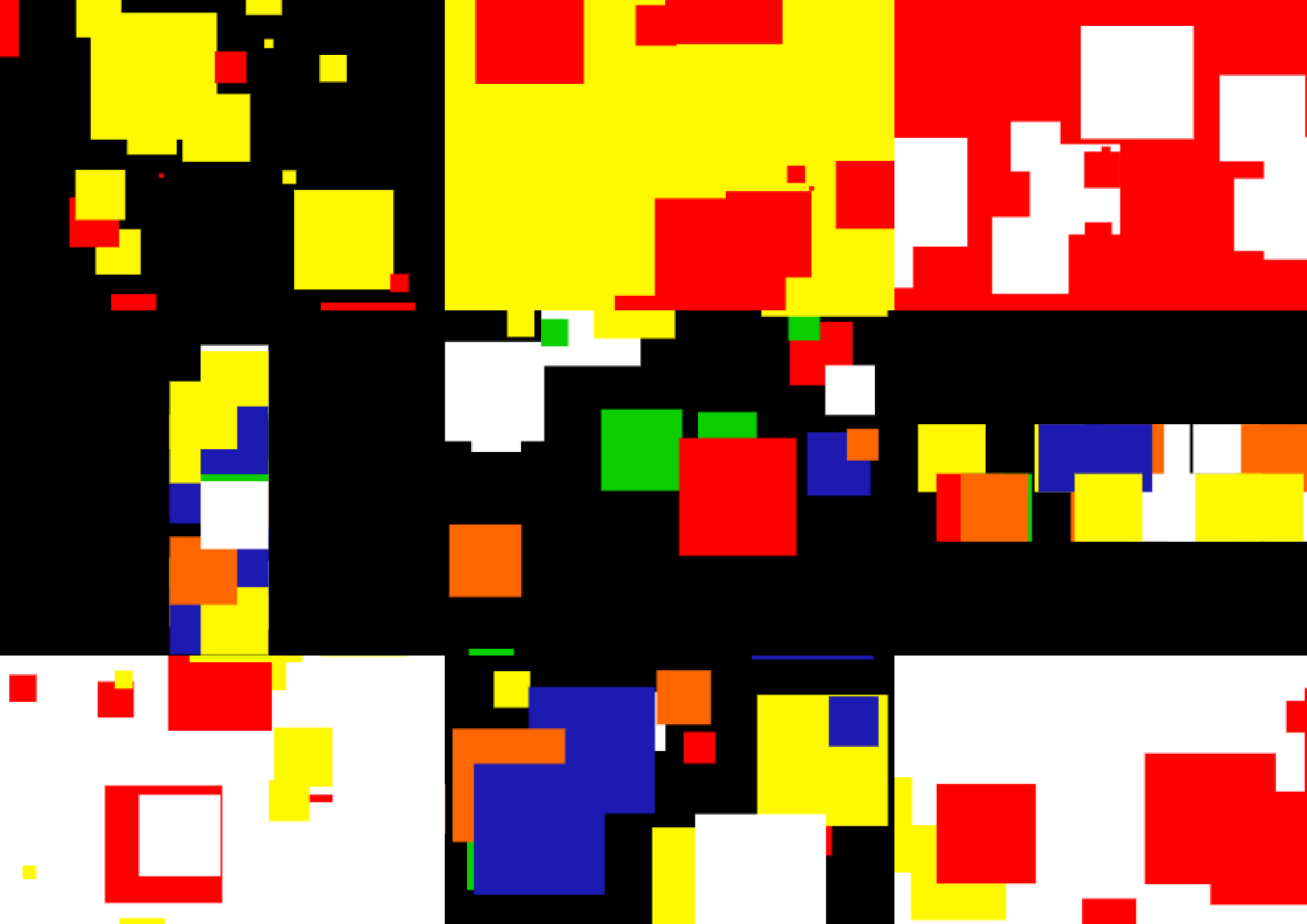


COM.POZICE V POHYBU

webová aplikace
2015

Webová aplikace „com.pozice v pohybu“ vychází z práce „com.pozice“. Princip klikání je zde nahrazen stálým pohybem čtverců náhodně se generujících ze stran. Tématem díla již nejsou obrazce samotné, ale následné jejich prolnutí. Divák nikdy dopředu neví, který čtverec bude v popředí, který v pozadí. Každá sekvence má proto svůj osobitý význam.



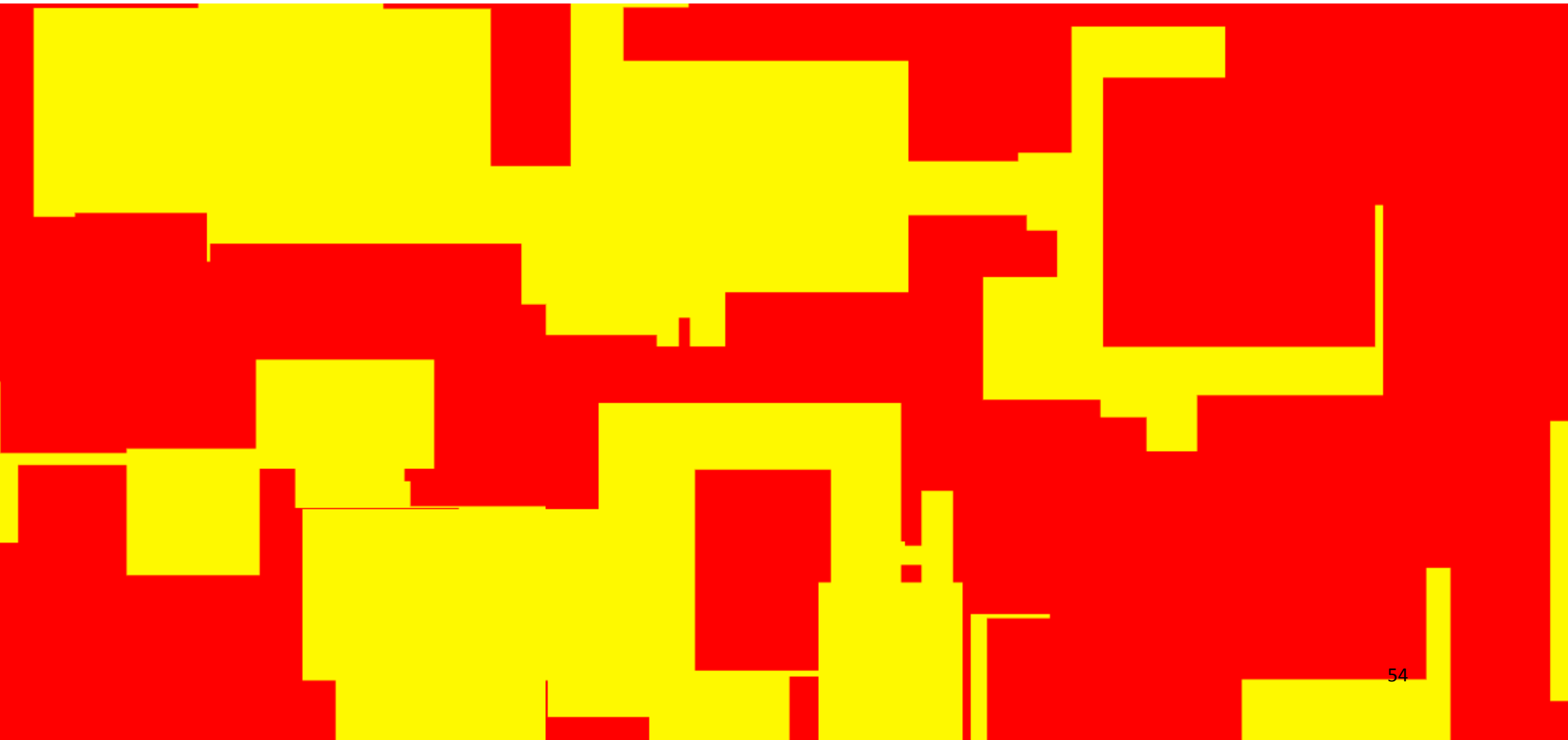


COM.POZICE V ČASE

2016

webová aplikace

Com.pozice v čase je webová aplikace, jež náhodně generuje obrazce, které po daných časových úsecích rozpohybuje a různě s nimi zachází. Divák opět může zasáhnout pomocí klikání myší a přidávat tak nové segmenty do stále opakujícího se procesu.





KOSTKY

2016

světelná instalace

Moje první práce, která vychází ze souboru `com,pozice`, jež ale není tvořena pro webový prostor. Jedná se o čtyři svítící plastové krychle naprogramované tak aby náhodně měnily barvu. Variabilita světelných odstínů je obrovská.



5/ ŽIVOTOPIS

EVA MARTIŠOVÁ

* 14 .7.1994 Bruntál

2013 - současnost Technická univerzita v Liberci Fakulta umění a architektury Vizuální komunikace / digitální média doc. Stanislav Zippe

2009 - 2013 – Gymnázium Olomouc – Hejčín

2000 – 2009 – Základní umělecká škola Bruntál

2016 – Preciosa Crystal Challenge (2. místo)

2016 – Brandz Friendz Czech s.r.o. – praxe

2015, 2017 - rodičovství

