

Verslag opkomende technologieën

Inleiding	3
Uitleg project	3
Oefeningen.....	3
Oefening 1: RGB-LED aansturen (één kleur)	3
Oefening 2: Kleur tonen bij knopdruk.....	5
Schakeling:	5
Oefening 3: RGB LED verandert van kleur na druk op de knop.....	6
Schakeling:	6
Code:	6
Oefening 4: Willekeurige kleurvolgorde tonen	8
Schakeling:	8
Code:	8
Oefening 5: 4 RGB leds groen laten oplichten in combinatie met 4 knoppen d.m.v. 2 shift registers.....	10
Schakeling:	10
Code:	10
Conclusie:	11
Oefening 6: Neopixel kleurt groen	11
Schakeling:	11
Code:	12
Oefening 7: Neopixel verandert om de seconde van kleur	12
Schakeling:	12
Code:	12
Oefening 8: elke Neopixel linken aan een knop, die ervoor zorgt dat de pixel uitgaat	14
Schakeling:	14
Code:	14
Oefening 9: Geef een random kleurencode van 4 kleuren en geef elke knop een kleur van deze volgorde.	15
Schakeling:	15
Code:	15
Oefening 10: Denkspel met verbetering.....	17
Schakeling:	17

Code:	17
Oefening 11: De volledige code en schakeling met extra Neopixel	19
Schakeling:	19
Code:	19

Inleiding

Tijdens de lessen praktijk wordt gewerkt aan het elektronische aspect van ‘Project Gebruiksgericht Ontwerpen’. Daarbij werden op voorhand duidelijke doelen opgesteld, waar naartoe kon worden gewerkt tijdens de lessen. Dit gebeurde door te starten met kleine oefeningen die stelselmatig complexer werden, zodat uiteindelijk in verschillende stappen de finale schakeling en bijhorende code tot stand kwamen.

Uitleg project

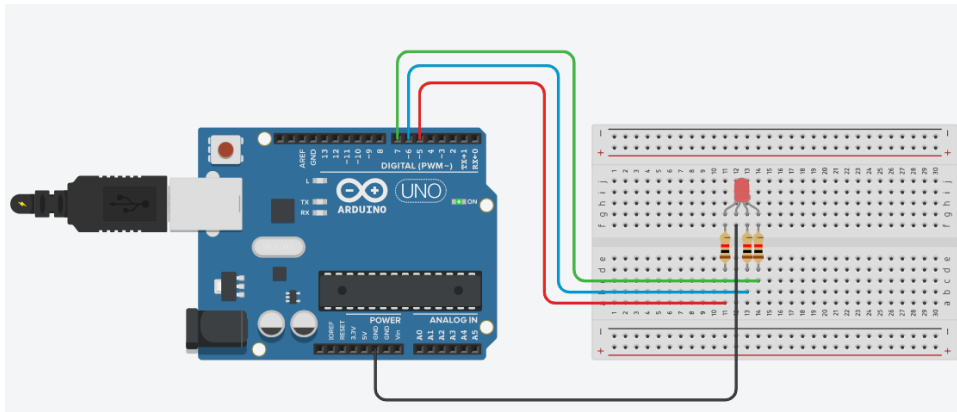
De bedoeling van het project is om de mentale gezondheid van ouderen te stimuleren d.m.v. een denkspel. Zo wordt op het scherm van de spelconsole een random kleurenvolgorde getoond van 4 kleuren waarna de 4 knoppen ook een random kleur van de kleurenvolgorde krijgen. Deze moet ingegeven worden door de speler. Vervolgens kleuren de juist ingegeven knoppen groen, en de foute knoppen rood. Als alles juist was ingegeven komt er op het scherm ‘succes’, indien niet: ‘fail’. De besturing van de spelconsole gebeurt met een 5e knop, die ook beschikt van een lichtje. Die geeft aan wanneer op de knop kan gedrukt worden. Het scherm wordt bestuurd door een Protopie-programma die gelinkt is aan de code, die beschikt over waarden die naar de seriële monitor worden gestuurd. Om dit visueel duidelijker te maken staat hieronder een afbeelding:

AFBEELDING EINDPRODUCT

Oefeningen

Oefening 1: RGB-LED aansturen (één kleur)

Schakeling:



Code (LED rood laten knipperen):

```
int redPin = 5;
int greenPin = 7;
int bluePin = 6;

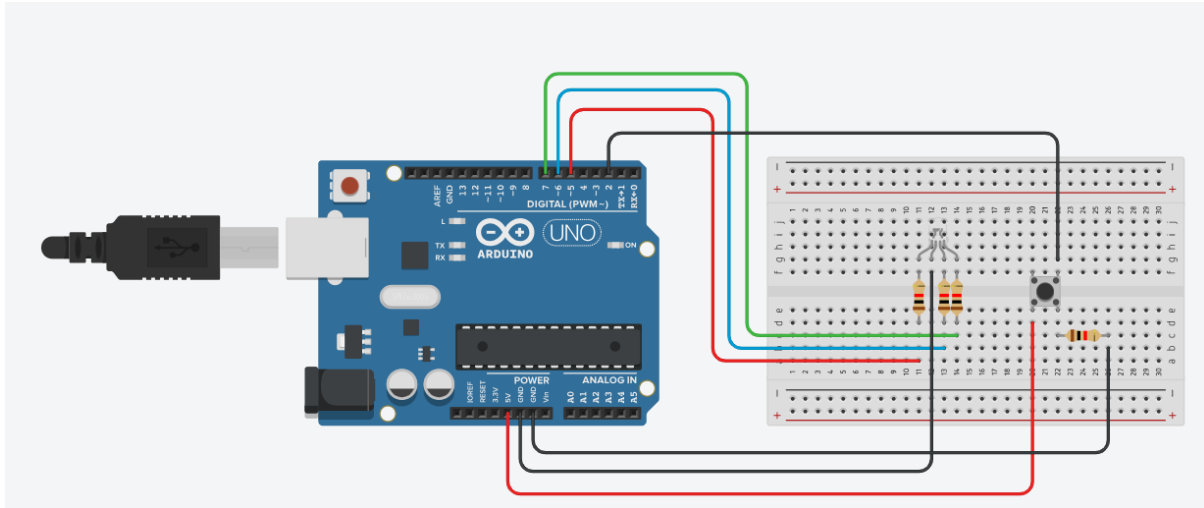
void setup() {
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}

void loop() {
  // Rood aan
  analogWrite(redPin, 255);
  analogWrite(greenPin, 0);
  analogWrite(bluePin, 0);
  delay(1000);

  // LED uit
  analogWrite(redPin, 0);
  delay(1000);
}
```

Oefening 2: Kleur tonen bij knopdruk

Schakeling:



Code:

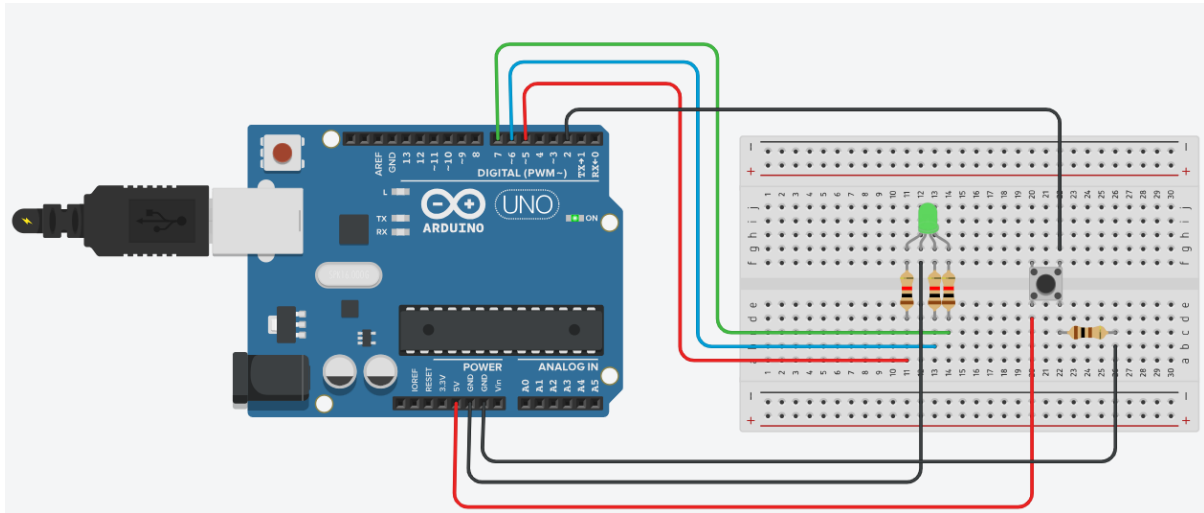
```
int buttonPin = 2;
int redPin = 5;
int greenPin = 7;
int bluePin = 6;

void setup() {
  pinMode(buttonPin, INPUT_PULLUP);
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}

void loop() {
  if (digitalRead(buttonPin) == LOW) {
    analogWrite(redPin, 0);
    analogWrite(greenPin, 0);
    analogWrite(bluePin, 0);
  }
  else {
    analogWrite(redPin, 0);
    analogWrite(greenPin, 255);
    analogWrite(bluePin, 0);
  }
}
```

Oefening 3: RGB LED verandert van kleur na druk op de knop

Schakeling:



Code:

```
int knopPin = 2;
int roodPin = 5;
int groenPin = 7;
int blauwPin = 6;

bool wachtOpKleur = false;
bool knopWasIngedrukt = false;
unsigned long startWachtTijd = 0;
const unsigned long wachttijd = 500; // halve seconde

void setup() {
  pinMode(knopPin, INPUT_PULLUP);
  pinMode(roodPin, OUTPUT);
  pinMode(groenPin, OUTPUT);
  pinMode(blauwPin, OUTPUT);
  randomSeed(analogRead(A0)); // Zorg voor willekeur
}

void loop() {
  bool knopIngedrukt = digitalRead(knopPin) == LOW;

  if (knopIngedrukt && !knopWasIngedrukt && !wachtOpKleur) {
    wachtOpKleur = true;
    startWachtTijd = millis();
  }
```

```

}

if (wachtOpKleur && millis() - startWachtTijd >= wachttijd) {
    int kleurCode = random(0, 4);
    toonKleur(kleurCode);
    wachtOpKleur = false;
}
knopWasIngedrukt = knopIngedrukt;
}

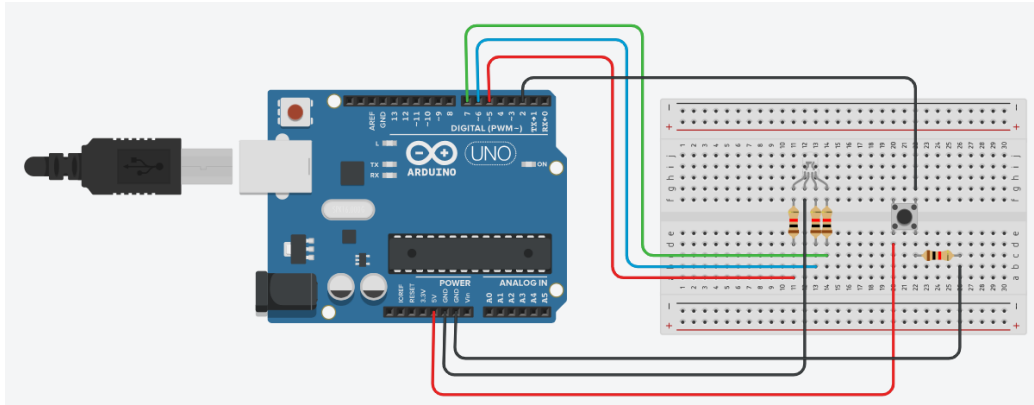
void toonKleur(int code) {
    analogWrite(roodPin, 0);
    analogWrite(groenPin, 0);
    analogWrite(blauwPin, 0);

    switch (code) {
        case 0: // geel
            analogWrite(roodPin, 255);
            analogWrite(groenPin, 255);
            break;
        case 1: // groen
            analogWrite(groenPin, 255);
            break;
        case 2: // blauw
            analogWrite(blauwPin, 255);
            break;
        case 3: // paars
            analogWrite(roodPin, 255);
            analogWrite(blauwPin, 255);
            break;
    }
}
}

```

Oefening 4: Willekeurige kleurvolgorde tonen

Schakeling:



Code:

```
int knopPin = 2;
int roodLed = 5;
int groenLed = 7;
int blauwLed = 6;

bool isAanHetWachten = false;
bool knopWasAlIngedrukt = false;
unsigned long tijdToenWachtenStartte = 0;
const unsigned long wachtVertraging = 500;

void setup() {
  pinMode(knopPin, INPUT_PULLUP);
  pinMode(roodLed, OUTPUT);
  pinMode(groenLed, OUTPUT);
  pinMode(blauwLed, OUTPUT);
  randomSeed(analogRead(A0));
}

void loop() {
  bool knopIngedrukt = digitalRead(knopPin) == LOW;

  // Start wachtmoment als knop net werd ingedrukt
  if (knopIngedrukt && !knopWasAlIngedrukt && !isAanHetWachten) {
    isAanHetWachten = true;
    tijdToenWachtenStartte = millis();
  }

  if (isAanHetWachten && millis() - tijdToenWachtenStartte >= wachtVertraging) {
```



```
int kleurCode = random(0, 4);
toonKleur(kleurCode);
isAanHetWachten = false;
}

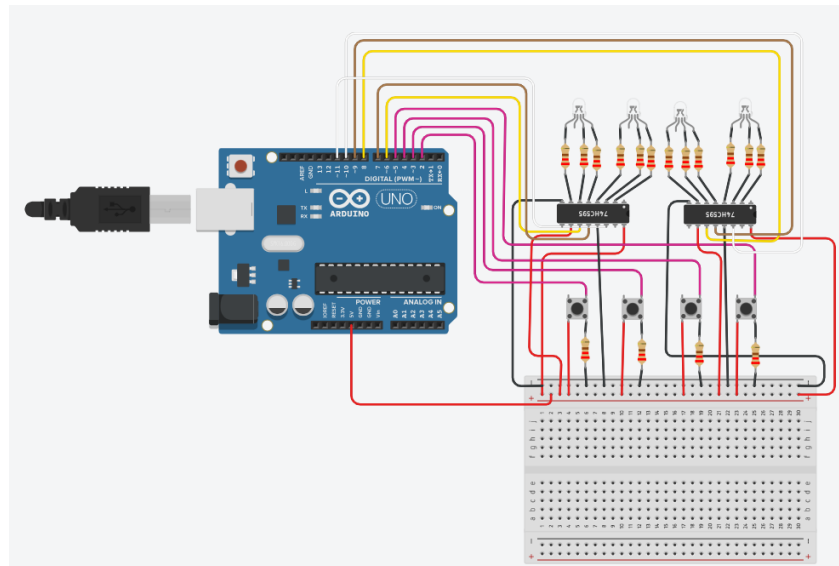
knopWasAllIngedrukt = knopIngedrukt;
}

void toonKleur(int code) {
  // Eerst alle leds uitzetten
  analogWrite(roodLed, 0);
  analogWrite(groenLed, 0);
  analogWrite(blauwLed, 0);

  switch (code) {
    case 0: // Geel (rood + groen)
      analogWrite(roodLed, 255);
      analogWrite(groenLed, 255);
      break;
    case 1: // Groen
      analogWrite(groenLed, 255);
      break;
    case 2: // Blauw
      analogWrite(blauwLed, 255);
      break;
    case 3: // Paars (rood + blauw)
      analogWrite(roodLed, 255);
      analogWrite(blauwLed, 255);
      break;
  }
}
```

Oefening 5: 4 RGB leds groen laten oplichten in combinatie met 4 knoppen d.m.v. 2 shift registers

Schakeling:



Code:

```
const int latchPin = 8;
const int klokPin = 12;
const int dataPin = 11;

void setup() {
  pinMode(latchPin, OUTPUT);
  pinMode(klokPin, OUTPUT);
  pinMode(dataPin, OUTPUT);

  byte groenLeds1 = 0b01100100;
  byte groenLeds2 = 0b10010010;

  schrijfNaarShiftregisters(groenLeds1, groenLeds2);
}

void loop() {
}

void schrijfNaarShiftregisters(byte patroon1, byte patroon2) {
  digitalWrite(latchPin, LOW);
  shiftOut(dataPin, klokPin, MSBFIRST, patroon1);
  shiftOut(dataPin, klokPin, MSBFIRST, patroon2);
}
```

```
digitalWrite(latchPin, HIGH);  
}
```

Conclusie:

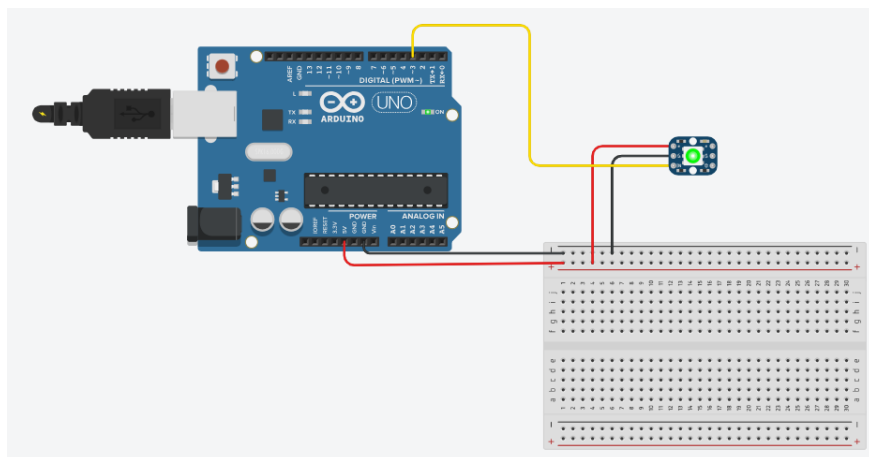
Deze code werkt niet en ook de opstelling is heel ingewikkeld. Hierdoor werd gezocht naar een oplossing, zodat zowel het programmeren als het schakelen eenvoudiger wordt zonder gevolgen voor het resultaat.

Zo werd ons 'Neopixels' aangeraden. Hierdoor zijn er veel minder aansluitingen nodig en kunnen de 2 shift registers achterwege gelaten worden. Ook de code kan eenvoudiger geschreven.

Omdat er verder wordt gegaan met Neopixels, wordt hiermee geëxperimenteerd in de volgende oefeningen om zodoende tot het finale eindresultaat te komen.

Oefening 6: Neopixel kleurt groen

Schakeling:



Code:

```
#include <Adafruit_NeoPixel.h>

#define PIN 6

NUMPIXELS 1

Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

void setup() {

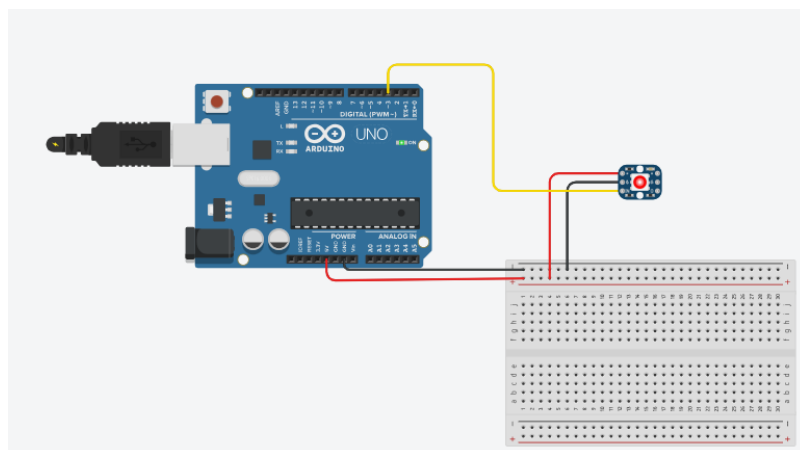
  pixels.begin();
  pixels.setPixelColor(0, pixels.Color(0, 255, 0));
  pixels.show();

void loop() {

}
```

Oefening 7: Neopixel verandert om de seconde van kleur

Schakeling:



Code:

```
#include <Adafruit_NeoPixel.h>
#define PIN 6
#define NUMPIXELS 4
Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
```

```
int kleurIndex = 0;

void setup() {
    pixels.begin();
}

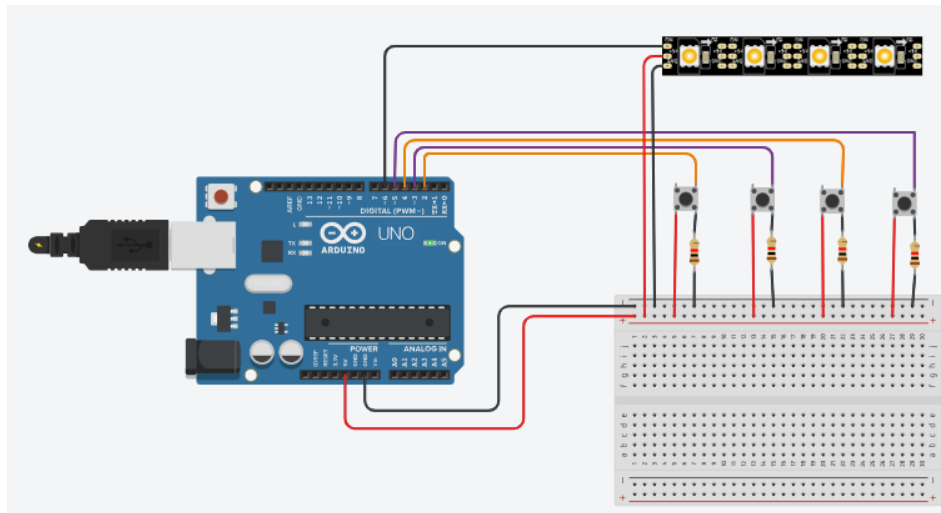
void loop() {
    uint32_t kleuren[5] = {
        pixels.Color(255, 0, 0),
        pixels.Color(0, 255, 0),
        pixels.Color(0, 0, 255),
        pixels.Color(255, 255, 255),
        pixels.Color(0, 0, 0)
    };

    for (int i = 0; i < NUMPIXELS; i++) {
        pixels.setPixelColor(i, kleuren[kleurIndex]);
    }

    pixels.show();
    delay(1000);
    kleurIndex = (kleurIndex + 1) % 5;
}
```

Oefening 8: Elke Neopixel linken aan een knop, die ervoor zorgt dat de pixel uitgaat

Schakeling:



Code:

```
#include <Adafruit_NeoPixel.h>

#define PIN 6
#define NUMPIXELS 4

Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

const int knopPinnen[NUMPIXELS] = {2, 3, 4, 5};

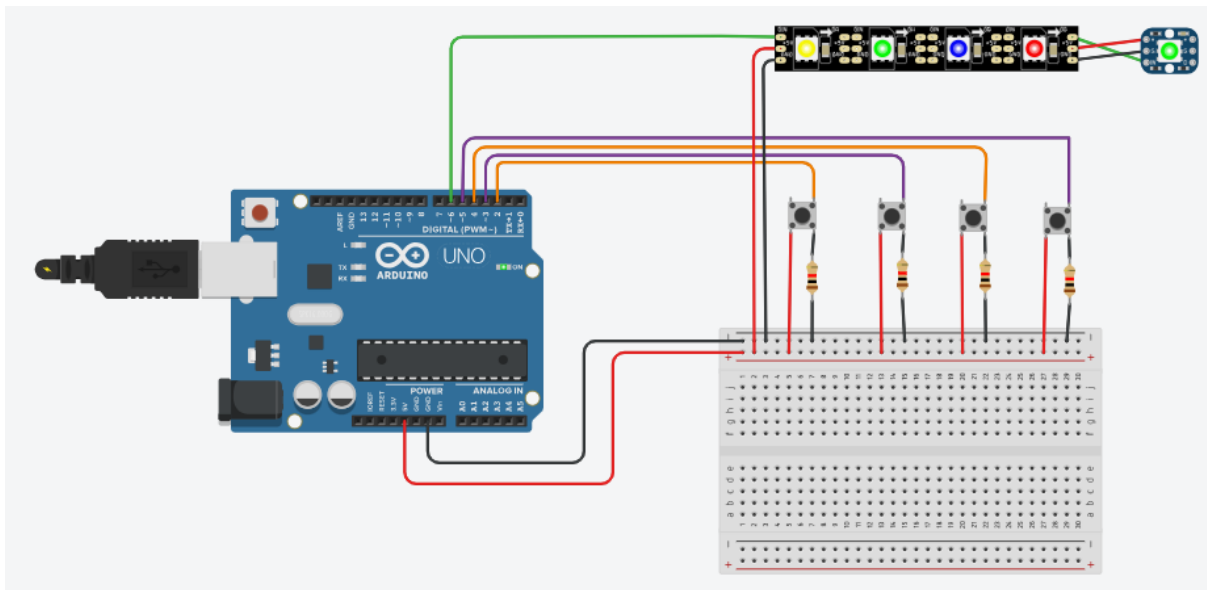
void setup() {
  pixels.begin();
  for (int i = 0; i < NUMPIXELS; i++) {
    pinMode(knopPinnen[i], INPUT);
  }
}

void loop() {
  for (int i = 0; i < NUMPIXELS; i++) {
    if (digitalRead(knopPinnen[i]) == HIGH) {
      pixels.setPixelColor(i, pixels.Color(0, 0, 0));
    } else {
      pixels.setPixelColor(i, pixels.Color(255, 100, 0));
    }
  }
}
```

```
}  
pixels.show();
```

Oefening 9: Geef een random kleurencode van 4 kleuren en geef elke knop een kleur van deze volgorde.

Schakeling:



Code:

```
#include <Adafruit_NeoPixel.h>  
  
#define LED_PIN 6  
#define AANTAL_PIXELS 5  
  
Adafruit_NeoPixel pixels(AANTAL_PIXELS, LED_PIN, NEO_GRB + NEO_KHZ800);  
  
const int knopPinnen[4] = {2, 3, 4, 5};  
int kleurVolgorde[4];  
bool knopWasIngedrukt[4] = {false, false, false, false};  
  
uint32_t kleuren[4] = {  
  pixels.Color(255, 0, 0),  
  pixels.Color(0, 0, 255),  
  pixels.Color(255, 255, 0),  
  pixels.Color(0, 255, 0)  
};
```

```

void shuffleKleuren() {
    for (int i = 0; i < 4; i++) {
        kleurVolgorde[i] = i;
    }

    for (int i = 3; i > 0; i--) {
        int j = random(i + 1);
        int temp = kleurVolgorde[i];
        kleurVolgorde[i] = kleurVolgorde[j];
        kleurVolgorde[j] = temp;
    }
}

void setup() {
    pixels.begin();
    randomSeed(analogRead(0));

    for (int i = 0; i < 4; i++) {
        pinMode(knopPinnen[i], INPUT);
    }

    shuffleKleuren();

    for (int i = 0; i < 4; i++) {
        pixels.setPixelColor(i, kleuren[kleurVolgorde[i]]);
    }

    for (int i = 0; i < 4; i++) {
        pixels.setPixelColor(4, kleuren[kleurVolgorde[i]]);
        pixels.show();
        delay(500);
    }

    pixels.setPixelColor(4, 0);
    pixels.show();
}

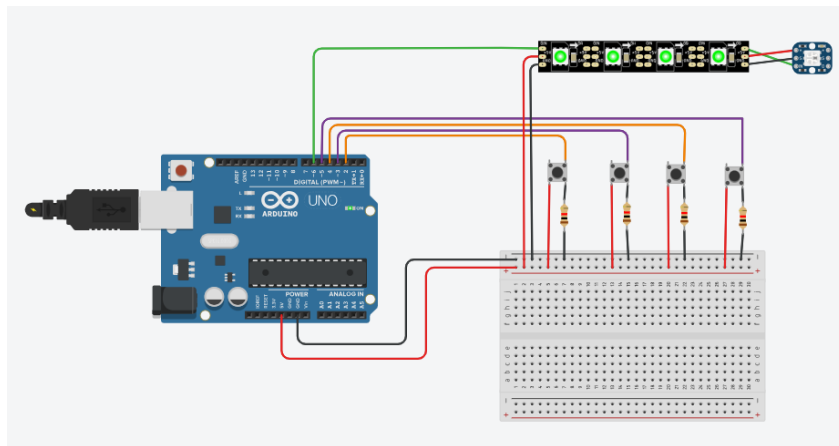
void loop() {
    for (int i = 0; i < 4; i++) {
        if (digitalRead(knopPinnen[i]) == HIGH && !knopWasIngedrukt[i]) {
            pixels.setPixelColor(i, 0);
            knopWasIngedrukt[i] = true;
            pixels.show();
        }
    }
}

```


Oefening 10: Denkspel met verbetering

Bij deze oefening wordt eerst via de 5e Neopixel de random kleurenvolgorde getoond. Daarna krijgt elke knop een random kleur van de kleurenvolgorde. Na de ingegeven volgorde wordt dit gecontroleerd en het bijhorende neopixel groen opgelicht als het juist was, en rood wanneer het fout was.

Schakeling:



Code:

```
#include <Adafruit_NeoPixel.h>

#define PIN 6
#define NUMPIXELS 5

Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

const int knopPinnen[4] = {2, 3, 4, 5};
int kleurenIndex[4];
bool knopIngedrukt[4] = {false, false, false, false};
bool knopStatus[4] = {false, false, false, false};
int juisteKleurVolgorde[4];

uint32_t kleuren[4] = {
  pixels.Color(255, 0, 0),
  pixels.Color(0, 0, 255),
  pixels.Color(255, 255, 0),
  pixels.Color(0, 255, 0)
};

void shuffleKleuren() {
  for (int i = 0; i < 4; i++) {
    kleurenIndex[i] = i;
  }
}
```

```

    for (int i = 3; i > 0; i--) {
        int j = random(i + 1);
        int temp = kleurenIndex[i];
        kleurenIndex[i] = kleurenIndex[j];
        kleurenIndex[j] = temp;
    }
}

void setup() {
    pixels.begin();
    randomSeed(analogRead(0));

    for (int i = 0; i < 4; i++) {
        pinMode(knopPinnen[i], INPUT);
    }

    shuffleKleuren();

    // Toon volgorde via pixel 4
    for (int i = 0; i < 4; i++) {
        juisteKleurVolgorde[i] = kleurenIndex[i];
        pixels.setPixelColor(4, kleuren[juisteKleurVolgorde[i]]);
        pixels.show();
        delay(600);
        pixels.setPixelColor(4, 0);
        pixels.show();
        delay(200);
    }

    delay(500);
    for (int i = 0; i < 4; i++) {
        pixels.setPixelColor(i, kleuren[kleurenIndex[i]]);
    }
    pixels.setPixelColor(4, 0);
    pixels.show();
}

void loop() {

    int count = 0;
    for (int i = 0; i < 4; i++) {
        if (!knopIngedrukt[i] && digitalRead(knopPinnen[i]) == HIGH) {
            knopIngedrukt[i] = true;
            knopStatus[i] = (kleurenIndex[i] == juisteKleurVolgorde[i]);
            pixels.setPixelColor(i, 0);
            pixels.show();
        }
        if (knopIngedrukt[i]) {
            count++;
        }
    }

    if (count == 4) {
        delay(500);

        for (int i = 0; i < 4; i++) {
            if (knopStatus[i]) {

```

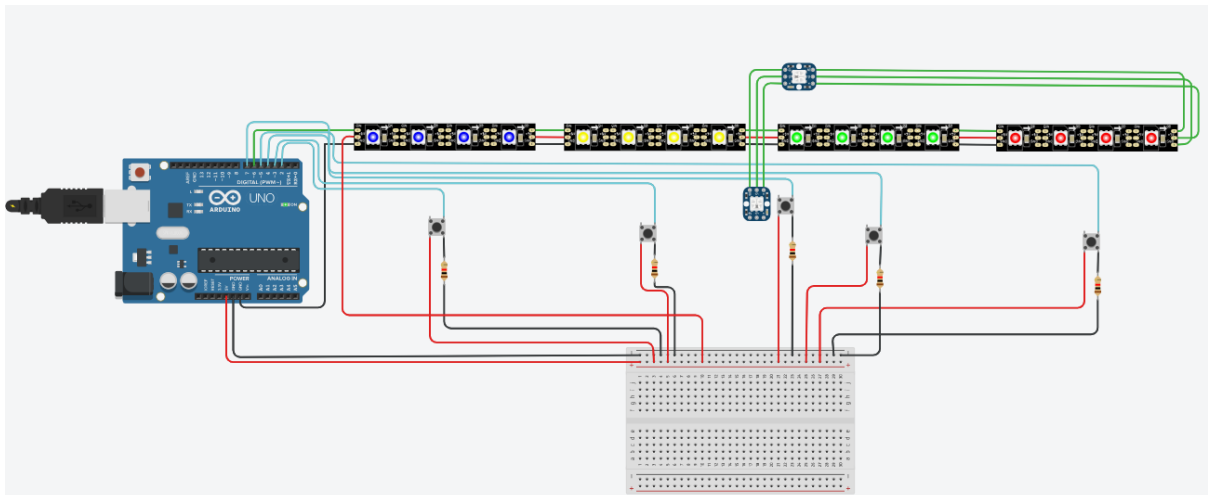
```

        pixels.setPixelColor(i, pixels.Color(0, 255, 0));
    } else {
        pixels.setPixelColor(i, pixels.Color(255, 0, 0));
    }
}
pixels.show();
while (true); // stop na resultaat
}
}

```

Oefening 11: De volledige code en schakeling van het project met extra Neopixel + weerstanden

Schakeling:



Code:

```

#include <Adafruit_NeoPixel.h>

#define LED_PIN 6

#define AANTAL_LEDS 18

#define KNOP1 2

#define KNOP2 3

#define KNOP3 4

#define KNOP4 5

#define START_KNOP 7

```

```
#define ROOD_PIN 10
#define GROEN_PIN 11
#define BLAUW_PIN 12

Adafruit_NeoPixel leds(AANTAL_LEDS, LED_PIN, NEO_GRB + NEO_KHZ800);

int segmentBegin[4] = {0, 4, 8, 12};
int segmentEinde[4] = {3, 7, 11, 15};

int knoppen[4] = {KNOP1, KNOP2, KNOP3, KNOP4};
bool vorigeKnopStatus[4] = {true, true, true, true};
bool vorigeStartStatus = true;

const uint32_t KLEUREN[4] = {
    Adafruit_NeoPixel::Color(255, 0, 0),
    Adafruit_NeoPixel::Color(255, 255, 0),
    Adafruit_NeoPixel::Color(0, 255, 0),
    Adafruit_NeoPixel::Color(0, 0, 255)
};

const char* KLEURNAMEN[4] = {"rood", "geel", "groen", "blauw"};

int kleurPerKnop[4] = {0, 1, 2, 3};
int doelVolgorde[4];
int spelerInput[4];
int invoerIndex = 0;

int volgordeIndex = 0;
bool volgordeKlaar = false;
bool wachtenOpNieuweStart = false;
```

```

void setup() {

  Serial.begin(9600);

  leds.begin();

  leds.show();

  for (int i = 0; i < 4; i++) {

    pinMode(knoppen[i], INPUT_PULLUP);

  }

  pinMode(START_KNOP, INPUT_PULLUP);


  pinMode(ROOD_PIN, OUTPUT);
  pinMode(GROEN_PIN, OUTPUT);
  pinMode(BLAUW_PIN, OUTPUT);


  randomSeed(analogRead(0));


  nieuweRonde();
  toonIndicator(true);


  while (digitalRead(START_KNOP) == LOW);
  vorigeStartStatus = digitalRead(START_KNOP);
}


void loop() {

  bool huidigeStart = digitalRead(START_KNOP);

  if (huidigeStart == LOW && vorigeStartStatus == HIGH) {

    if (wachtenOpNieuweStart) {

      Serial.println("start");

      nieuweRonde();

      wachtenOpNieuweStart = false;

      toonIndicator(true);

    } else if (!volgordeKlaar && volgordeIndex < 4) {

```

```

    toonVolgendeKleur();

}

}

vorigeStartStatus = huidigeStart;


for (int i = 0; i < 4; i++) {

    bool nulIngedrukt = digitalRead(knoppen[i]) == LOW;

    if (nulIngedrukt && vorigeKnopStatus[i] == true && volgordeKlaar) {

        verwerkDruk(i);

    }

    vorigeKnopStatus[i] = nulIngedrukt;

}

}

void nieuweRonde() {

    int indices[4] = {0, 1, 2, 3};


    // shuffle volgorde
    for (int i = 3; i > 0; i--) {

        int j = random(i + 1);

        int temp = indices[i];

        indices[i] = indices[j];

        indices[j] = temp;

    }

    for (int i = 0; i < 4; i++) {

        doelVolgorde[i] = indices[i];

    }


    // shuffle kleurtoewijzing aan knoppen
    for (int i = 3; i > 0; i--) {

        int j = random(i + 1);

```

```
int temp = kleurPerKnop[i];

kleurPerKnop[i] = kleurPerKnop[j];

kleurPerKnop[j] = temp;
}

invoerIndex = 0;

volgordeIndex = 0;

volgordeKlaar = false;

wisAlleSegmenten();

wisPixel(16);

wisPixel(17);
}

void toonVolgendeKleur() {

int kleurIndex = doelVolgorde[volgordeIndex];

Serial.println(KLEURNAMEN[kleurIndex]);

leds.setPixelColor(16, KLEUREN[kleurIndex]);

leds.show();

if (volgordeIndex == 3) {

    delay(600);

    wisPixel(16);

    Serial.println("einde_volgorde");

    for (int i = 0; i < 4; i++) {

        zetSegment(i, KLEUREN[kleurPerKnop[i]]);

    }

    volgordeKlaar = true;

    toonIndicator(false);

} else {
```

```
    toonIndicator(true);
}

    volgordeIndex++;
}

void verwerkDruk(int knopIndex) {
    if (invoerIndex >= 4) return;

    zetSegmentUit(knopIndex);
    spelerInput[invoerIndex] = kleurPerKnop[knopIndex];
    invoerIndex++;

    if (invoerIndex == 4) {
        delay(300);
        controleerAntwoord();
    }
}

void controleerAntwoord() {
    bool juist = true;
    for (int i = 0; i < 4; i++) {
        if (spelerInput[i] == doelVolgorde[i]) {
            zetSegment(i, Adafruit_NeoPixel::Color(0, 255, 0)); // groen = juist
        } else {
            zetSegment(i, Adafruit_NeoPixel::Color(255, 0, 0)); // rood = fout
            juist = false;
        }
    }

    if (juist) {
        Serial.println("succes");
    }
}
```



```

} else {

    Serial.println("fout");

}

delay(2000);

wisAlleSegmenten();

wachtenOpNieuweStart = true;

volgordeKlaar = false;

toonIndicator(true);

}

void zetSegment(int segmentIndex, uint32_t kleur) {

    for (int i = segmentBegin[segmentIndex]; i <= segmentEinde[segmentIndex]; i++) {

        leds.setPixelColor(i, kleur);

    }

    leds.show();

}

void zetSegmentUit(int segmentIndex) {

    for (int i = segmentBegin[segmentIndex]; i <= segmentEinde[segmentIndex]; i++) {

        leds.setPixelColor(i, 0);

    }

    leds.show();

}

void wisAlleSegmenten() {

    for (int i = 0; i < 4; i++) {

        zetSegmentUit(i);

    }

}

```

```

void wisPixel(int index) {

  leds.setPixelColor(index, 0);

  leds.show();
}

void toonIndicator(bool aan) {

  if (aan) {

    leds.setPixelColor(17, Adafruit_NeoPixel::Color(255, 100, 0));

  } else {

    leds.setPixelColor(17, 0);

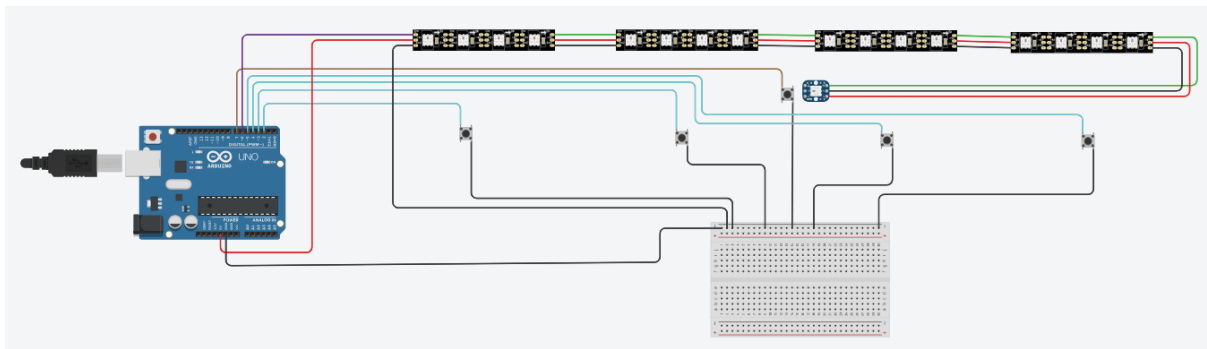
  }

  leds.show();
}

```

Oefening 12: De volledige code en schakeling van het project zonder extra Neopixel en zonder weerstanden

Schakeling:



Code:

```

#include <Adafruit_NeoPixel.h>

#define LED_PIN 6
#define AANTAL_LEDS 17

#define KNOP1 2
#define KNOP2 3
#define KNOP3 4
#define KNOP4 5

```

```

#define START_KNOP 7

#define ROOD_PIN 10
#define GROEN_PIN 11
#define BLAUW_PIN 12

Adafruit_NeoPixel leds(AANTAL_LEDS, LED_PIN, NEO_GRB + NEO_KHZ800);

int segmentBegin[4] = {0, 4, 8, 12};
int segmentEinde[4] = {3, 7, 11, 15};

int knoppen[4] = {KNOP1, KNOP2, KNOP3, KNOP4};
bool vorigeKnopStatus[4] = {true, true, true, true};
bool vorigeStartStatus = true;

const uint32_t KLEUREN[4] = {
  Adafruit_NeoPixel::Color(255, 0, 0),
  Adafruit_NeoPixel::Color(255, 255, 0),
  Adafruit_NeoPixel::Color(0, 255, 0),
  Adafruit_NeoPixel::Color(0, 0, 255)
};

const char* KLEURNAMEN[4] = {"rood", "geel", "groen", "blauw"};

int kleurPerKnop[4] = {0, 1, 2, 3};
int doelVolgorde[4];
int spelerInput[4];
int invoerIndex = 0;

int volgordeIndex = 0;
bool volgordeKlaar = false;
bool wachtOpVolgendeStart = false;

void setup() {
  Serial.begin(9600);
  leds.begin();
  leds.show();

  for (int i = 0; i < 4; i++) {
    pinMode(knoppen[i], INPUT_PULLUP);
  }
  pinMode(START_KNOP, INPUT_PULLUP);

  pinMode(ROOD_PIN, OUTPUT);
  pinMode(GROEN_PIN, OUTPUT);
  pinMode(BLAUW_PIN, OUTPUT);

```

```

randomSeed(analogRead(0));
nieuweRonde();
toonIndicator(true);

while (digitalRead(START_KNOP) == LOW);
vorigeStartStatus = digitalRead(START_KNOP);
}

void loop() {
    bool huidigeStart = digitalRead(START_KNOP);

    if (huidigeStart == LOW && vorigeStartStatus == HIGH) {
        if (wachtOpVolgendeStart) {
            Serial.println("start");
            nieuweRonde();
            wachtOpVolgendeStart = false;
            toonIndicator(true);
        } else if (!volgordeKlaar && volgordeIndex < 4) {
            toonVolgendeKleur();
        }
    }
    vorigeStartStatus = huidigeStart;

    for (int i = 0; i < 4; i++) {
        bool huidig = digitalRead(knoppen[i]);
        if (huidig == LOW && vorigeKnopStatus[i] == HIGH && volgordeKlaar) {
            verwerkKnopdruk(i);
        }
        vorigeKnopStatus[i] = huidig;
    }
}

void nieuweRonde() {
    int indices[4] = {0, 1, 2, 3};
    for (int i = 3; i > 0; i--) {
        int j = random(i + 1);
        int temp = indices[i];
        indices[i] = indices[j];
        indices[j] = temp;
    }
    for (int i = 0; i < 4; i++) {
        doelVolgorde[i] = indices[i];
    }

    for (int i = 3; i > 0; i--) {
        int j = random(i + 1);
        int temp = kleurPerKnop[i];

```

```

    kleurPerKnop[i] = kleurPerKnop[j];
    kleurPerKnop[j] = temp;
}

invoerIndex = 0;
volgordeIndex = 0;
volgordeKlaar = false;

wisAlleSegmenten();
wisPixel(16);
}

void toonVolgendeKleur() {
    int kleurIndex = doelVolgorde[volgordeIndex];
    Serial.println(KLEURNAMEN[kleurIndex]);

    if (volgordeIndex == 3) {
        delay(600);
        Serial.println("einde_volgorde");

        for (int i = 0; i < 4; i++) {
            zetSegment(i, KLEUREN[kleurPerKnop[i]]);
        }

        volgordeKlaar = true;
        toonIndicator(false);
    } else {
        toonIndicator(true);
    }

    volgordeIndex++;
}

void verwerkKnopdruk(int knopIndex) {
    if (invoerIndex >= 4) return;

    zetSegmentUit(knopIndex);
    spelerInput[invoerIndex] = kleurPerKnop[knopIndex];
    invoerIndex++;

    if (invoerIndex == 4) {
        delay(300);
        controleerAntwoord();
    }
}

void controleerAntwoord() {

```

```

bool allesJuist = true;
for (int i = 0; i < 4; i++) {
    if (spelerInput[i] == doelVolgorde[i]) {
        zetSegment(i, Adafruit_NeoPixel::Color(0, 255, 0));
    } else {
        zetSegment(i, Adafruit_NeoPixel::Color(255, 0, 0));
        allesJuist = false;
    }
}

if (allesJuist) {
    Serial.println("succes");
} else {
    Serial.println("fout");
}

delay(2000);
wisAlleSegmenten();
wachtOpVolgendeStart = true;
volgordeKlaar = false;
toonIndicator(true);
}

void zetSegment(int index, uint32_t kleur) {
    for (int i = segmentBegin[index]; i <= segmentEinde[index]; i++) {
        leds.setPixelColor(i, kleur);
    }
    leds.show();
}

void zetSegmentUit(int index) {
    for (int i = segmentBegin[index]; i <= segmentEinde[index]; i++) {
        leds.setPixelColor(i, 0);
    }
    leds.show();
}

void wisAlleSegmenten() {
    for (int i = 0; i < 4; i++) {
        zetSegmentUit(i);
    }
}

void wisPixel(int index) {
    leds.setPixelColor(index, 0);
    leds.show();
}

```

```

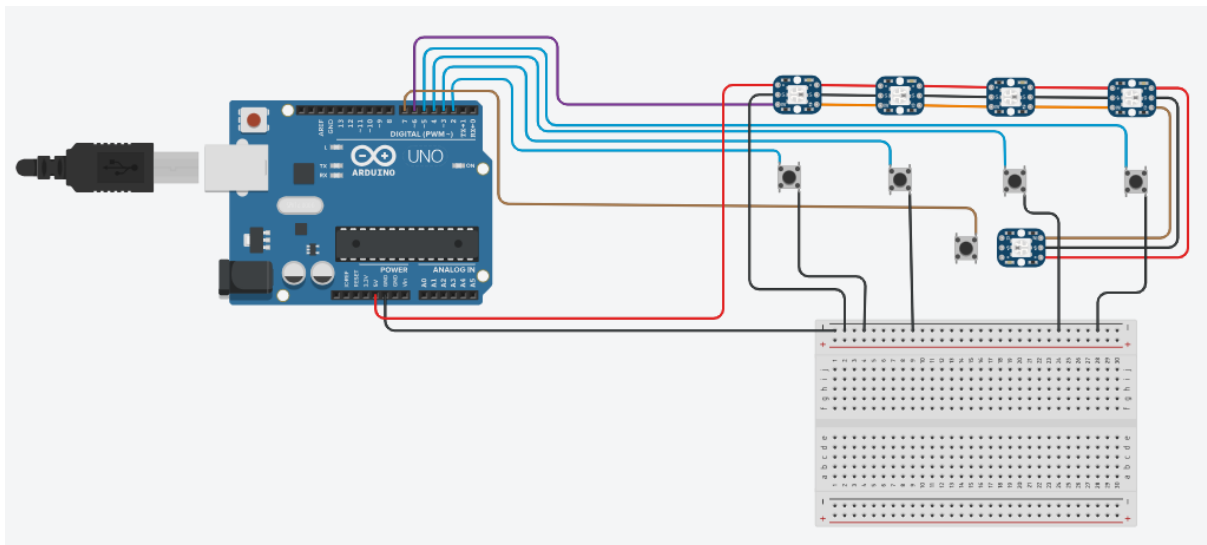
void toonIndicator(bool aan) {
  if (aan) {
    leds.setPixelColor(16, Adafruit_NeoPixel::Color(255, 100, 0));
  } else {
    leds.setPixelColor(16, 0);
  }
  leds.show();
}

```

Oefening 13: De volledige code en schakeling van het project met telkens 1 (i.p.v. 4) Neopixel(s) bij de bijhorende knop (FINAAL)

Door gebruikerstesten werd geconcludeerd dat er niet 4 Neopixels nodig zijn per knop, maar slechts 1. Dit werd aangepast in zowel de schakeling als in de code.

Schakeling:



Code:

```

#include <Adafruit_NeoPixel.h>

#define LED_PIN 6
#define AANTAL_LEDS 5 // 4 gekleurde + 1 oranje indicator

#define KNOP1 2
#define KNOP2 3

```

```

#define KNOP3 4
#define KNOP4 5
#define START_KNOP 7

Adafruit_NeoPixel leds(AANTAL_LEDS, LED_PIN, NEO_GRB + NEO_KHZ800);

int knopPinnen[4] = {KNOP1, KNOP2, KNOP3, KNOP4};
bool vorigeKnopStatus[4] = {true, true, true, true};
bool vorigeStartStatus = true;

const uint32_t KLEUREN[4] = {
  Adafruit_NeoPixel::Color(255, 0, 0),
  Adafruit_NeoPixel::Color(255, 255, 0),
  Adafruit_NeoPixel::Color(0, 255, 0),
  Adafruit_NeoPixel::Color(0, 0, 255)
};

const char* KLEUR_NAAM[4] = {"rood", "geel", "groen", "blauw"};

int kleurPerKnop[4] = {0, 1, 2, 3};
int doelVolgorde[4];
int spelerInput[4];
int invoerIndex = 0;

int volgordeIndex = 0;
bool klaarOmInTeVoeren = false;
bool wachtOpStart = false;

void setup() {
  Serial.begin(9600);
  leds.begin();
  leds.show();

  for (int i = 0; i < 4; i++) {
    pinMode(knopPinnen[i], INPUT_PULLUP);
  }
  pinMode(START_KNOP, INPUT_PULLUP);

  randomSeed(analogRead(0));
  nieuweRonde();
  zetIndicator(true);

  while (digitalRead(START_KNOP) == LOW);
  vorigeStartStatus = digitalRead(START_KNOP);
}

void loop() {

```



```

bool huidigeStart = digitalRead(START_KNOP);

if (huidigeStart == LOW && vorigeStartStatus == HIGH) {
    if (wachtOpStart) {
        Serial.println("start");
        nieuweRonde();
        wachtOpStart = false;
        zetIndicator(true);
    } else if (!klaarOmInTeVoeren && volgordelIndex < 4) {
        toonVolgendeKleur();
    }
}
vorigeStartStatus = huidigeStart;

for (int i = 0; i < 4; i++) {
    bool knopNu = digitalRead(knopPinnen[i]);
    if (knopNu == LOW && vorigeKnopStatus[i] == HIGH && klaarOmInTeVoeren) {
        verwerkKnopdruk(i);
    }
    vorigeKnopStatus[i] = knopNu;
}

void nieuweRonde() {
    int indices[4] = {0, 1, 2, 3};
    for (int i = 3; i > 0; i--) {
        int j = random(i + 1);
        int temp = indices[i];
        indices[i] = indices[j];
        indices[j] = temp;
    }
    for (int i = 0; i < 4; i++) {
        doelVolgorde[i] = indices[i];
    }

    for (int i = 3; i > 0; i--) {
        int j = random(i + 1);
        int temp = kleurPerKnop[i];
        kleurPerKnop[i] = kleurPerKnop[j];
        kleurPerKnop[j] = temp;
    }

    invoerIndex = 0;
    volgordelIndex = 0;
    klaarOmInTeVoeren = false;

    wisLeds();
}

```

```

    zetIndicator(true);
}

void toonVolgendeKleur() {
    int kleurIndex = doelVolgorde[volgordeIndex];
    Serial.println(KLEUR_NAAM[kleurIndex]);

    if (volgordeIndex == 3) {
        delay(600);
        Serial.println("einde_volgorde");

        for (int i = 0; i < 4; i++) {
            leds.setPixelColor(i, KLEUREN[kleurPerKnop[i]]);
        }
        leds.show();

        klaarOmInTeVoeren = true;
        zetIndicator(false);
    } else {
        zetIndicator(true);
    }

    volgordeIndex++;
}

void verwerkKnopdruk(int knopIndex) {
    if (invoerIndex >= 4) return;

    leds.setPixelColor(knopIndex, 0);
    leds.show();

    spelerInput[invoerIndex] = kleurPerKnop[knopIndex];
    invoerIndex++;

    if (invoerIndex == 4) {
        delay(300);
        controleerAntwoord();
    }
}

void controleerAntwoord() {
    bool juist = true;
    for (int i = 0; i < 4; i++) {
        if (spelerInput[i] != doelVolgorde[i]) {
            leds.setPixelColor(i, Adafruit_NeoPixel::Color(0, 255, 0));
        } else {
            leds.setPixelColor(i, Adafruit_NeoPixel::Color(255, 0, 0));
        }
    }
}

```

```
    juist = false;
  }
}
leds.show();

if (juist) {
  Serial.println("succes");
} else {
  Serial.println("fout");
}

delay(2000);
wisLeds();
wachtOpStart = true;
klaarOmInTeVoeren = false;
zetIndicator(true);
}

void wisLeds() {
  for (int i = 0; i < 4; i++) {
    leds.setPixelColor(i, 0);
  }
  leds.show();
}

void zetIndicator(bool aan) {
  if (aan) {
    leds.setPixelColor(4, Adafruit_NeoPixel::Color(255, 100, 0));
  } else {
    leds.setPixelColor(4, 0);
  }
  leds.show();
}
```