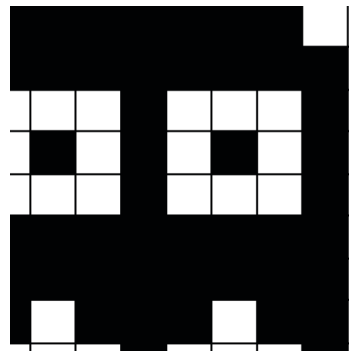

Handleiding Pixeltekeningen



jxxi

Beste leerkracht,

Met deze lesbundel willen we je op weg zetten om het onderwerp “pixeltekeningen” aan te brengen in je klas, met als doel een originele pixeltekening te kunnen ontwerpen als inzending voor de wedstrijd “5 jaar Fyxxi”¹.

De bundel is zo opgebouwd dat ze gebruikt kan worden in zowel het kleuteronderwijs, het lager onderwijs, als het secundair onderwijs. Ze start met een algemene inleiding rond pixeltekeningen, en gaat daarna dieper in op de mogelijkheden voor de verschillende leeftijden. Het lager onderwijs kan aan de slag met numerieke pixeltekeningen, het secundair onderwijs behandelt achtereenvolgens de numerieke en de binaire pixeltekeningen.

Vier winnende klassen worden uitgenodigd om op **9 mei 2019** “5 jaar Fyxxi” met ons te komen vieren. Ze worden de allereerste klassen die in onze spiksplinternieuwe locatie een scholenworkshop zullen volgen - en dit **volledig gratis**² ! Wie niet waagt, niet wint, dus. We wensen je alvast...

A pixelated graphic consisting of a checkmark on the left and the words 'VEEL SUCCES!' in a bold, blocky, pixelated font on the right.

Het Fyxxi-team

¹ Er is geen verplichting om deel te nemen aan de wedstrijd na het gebruik van deze bundel.

² We geven 5 gratis workshops weg: 1 voor kleuteronderwijs, 2 voor basisonderwijs, en 2 voor secundair onderwijs. Opgelet: De workshops voor lager en secundair onderwijs vinden plaats op *9 mei 2019* in het Fyxxilab in het kader van onze verjaardagsweek. Het vervoer van en naar de workshoplocatie is niet inbegrepen in de prijs en valt volledig onder de verantwoordelijkheid van de winnende klas.

De workshop voor het kleuteronderwijs wordt aangeboden op de school zelf, op een onderling af te spreken datum.

Fyxxi bepaalt de inhoud van de workshops samen met winnende klassen.

DEEL 1: PIXELS

Insteek

Wat zie je hieronder?



*(Geen paard maar een **foto** van een paard)*

En wat zie je wanneer je heel hard inzoomt?

(pixels, komt van picture elements)

Wat zijn dat eigenlijk, pixels?

(Kleine onderdeeljes van een foto, vierkantjes die elk hun eigen kleurwaarde hebben)



DEEL 2: PIXELTEKENINGEN VOOR KLEUTERS

Insteek

Zie *Deel 1*.

Probleem

Computers kunnen eigenlijk geen beelden onthouden, alleen maar getallen. Elk onderdeelje (= pixel) van een foto is dus opgeslagen als code en moet 'gerenderd' worden door die code om te zetten in beeld via een afgesproken systeem.

Opdracht

In een monochrome afbeelding (afbeelding in 1 kleur) is iedere pixel gekleurd of wit. Als een computer een beeld moet opslaan onthoudt hij precies welke puntjes gekleurd moeten worden en welke wit.

De afbeelding van het mannetje hieronder is een 7x7 afbeelding (7 rijen, 7 kolommen). Om deze afbeelding te maken, moeten de kleuters eerst twee vakjes wit laten, dan drie vakjes rood kleuren, en dan opnieuw twee vakjes wit laten. Op die manier wordt de afbeelding rij per rij opgebouwd.

Opdracht:

Oplossing:

	3													
	2			1										
	5													
	1			2										

Je kan werken met een in te kleuren raster zoals hierboven, of je kan werken met een vast raster waarop de kleuters de tekening maken met gekleurde post-it'jes, origamipapier, ... Voor jongere kleuters kan je het aantal pixels bijvoorbeeld aanduiden met een aantal stippen in plaats van met getallen.

Op het internet vind je tal van afbeeldingen waardoor je je kan laten inspireren. Behalve diertjes en figuurtjes kan je (voor de oudere kleuters) ook letters of cijfers laten maken.

Opmerking

Deze tekeningen van 49 pixels (7x7) in 2 kleuren (gekleurd en wit) manueel renderen duurt voor een kleuter enkele minuten.

Een gemiddelde smartphone neemt een foto van 5 megapixels (2560 x 1920 = 4.915.200 pixels) tot 13 megapixels (4160 x 3120 = 12.979.200 pixels) in "true colors" ($= (2^8)^3 = 2^{24} = 16.777.216$ mogelijke kleuren) met 1 druk op de knop.

Wedstrijd

Ontwerp met je klas een **originele pixeltekening van 10x10**, neem er een foto van, en stuur deze **voor 26 april 2019** naar wedstrijd@educentrum.be. Je maakt kans op een Codeer Lab voor je klas op je eigen school, op een onderling af te spreken datum.

DEEL 3: NUMERIEKE PIXELTEKENINGEN

A. PIXELTEKENING

Insteek

Zie Deel 1.

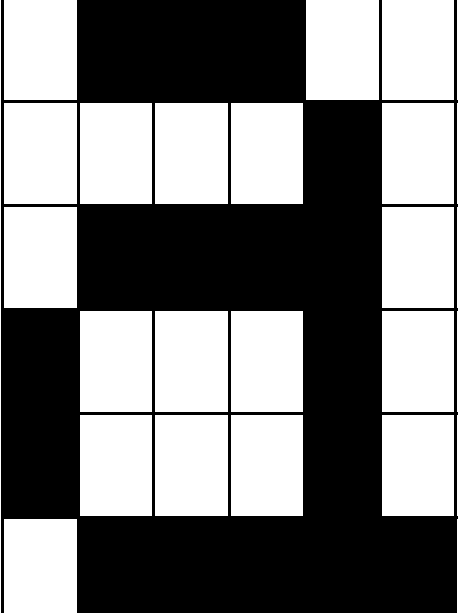
Probleem

Computers kunnen eigenlijk geen beelden onthouden, alleen maar getallen. Elk onderdeelje (= pixel) van een foto is opgeslagen als code en moet dus 'gerenderd' worden door die code om te zetten in beeld via een afgesproken systeem.

Opdracht

In een zwart-wit afbeelding is iedere pixel zwart of wit. Als een computer een beeld moet opslaan, onthoudt hij precies welke puntjes zwart moeten worden en welke wit.

Voorbeeld

	
	4 - 1
	1 - 3 - 1 - 1
	5

De letter 'a' is hier vergroot om de pixels te laten zien in een 6x6 raster (6 rijen, 6 kolommen). De rechterkolom laat zien hoe dit beeld weergegeven kan worden door getallen. We spreken af dat het eerste getal weergeeft hoeveel witte pixels er zijn, het tweede hoeveel zwarte, het derde hoeveel witte, enzovoort.

De eerste regel bestaat uit een witte pixel, dan drie zwarte, en dan weer twee witte. De eerste regel wordt daarom bewaard als 1 - 3 - 1.

Als de eerste pixel zwart is, zoals op regel 4 en 5, dan begint de regel met een nul. Het eerste cijfer geeft immers altijd het aantal witte pixels aan, en er zijn eerst nul witte pixels.

Op het einde van dit hoofdstuk staan nog enkele voorbeelden die je leerlingen kunnen ontcijferen met de bovenstaande methode. Op het internet vind je nog veel meer voorbeelden terug.

Opmerking

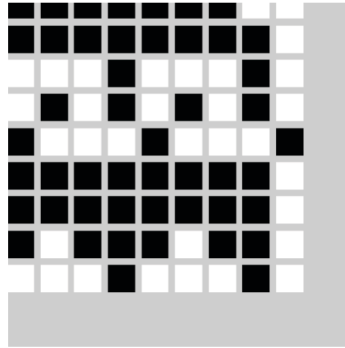
Deze tekeningen van 100 pixels (10x10) in 2 kleuren (zwart en wit) manueel renderen duurt zowat 5 minuten. Een gemiddelde smartphone neemt een foto van 5 megapixels (2560 x 1920 = 4.915.200 pixels) tot 13 megapixels (4160 x 3120 = 12.979.200 pixels) in "true colors" ($= (2^8)^3 = 2^{24} = 16.777.216$ mogelijke kleuren) met 1 druk op de knop.

Wedstrijd Lager Onderwijs

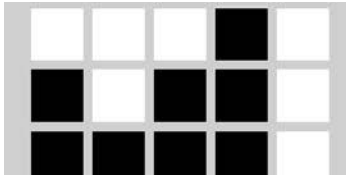
Ontwerp met je klas een **originale pixeltekening van 10x10**, zet deze om naar de juiste code, en stuur de code voor **26 april 2019** naar wedstrijd@educentrum.be.

B. OEPS... FOUTJE!

Soms maken de leerlingen bij het renderen eens een foutje. Bij computers is dat niet anders. Maar terwijl wij onmiddellijk zien dat er een fout zit in de onderstaande tekening, kan een computer dat niet. Computers kunnen namelijk niet interpreteren. Daarom is er een systeem ontworpen om mogelijke foutjes snel op te sporen, namelijk een controlerij en -kolom!

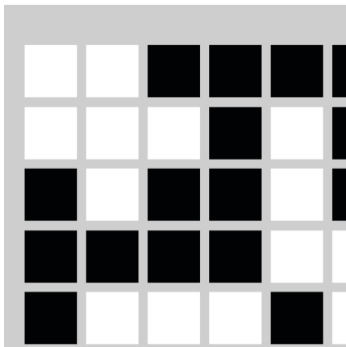


Om dit concept duidelijk te maken, kan je volgende oefening klassikaal maken, bijvoorbeeld aan bord.



‘tekening’.

1. Een leerling maakt een willekeurige



2. Voeg een extra rij en kolom toe en maak in die rij en kolom pixels zwart zodat op elke rij en op elke kolom een even aantal zwarte pixels staan (maar verklap dit nog niet op het moment dat je dit doet).

DEEL 4: BINAIRE PIXELTEKENINGEN

A. BINAIR TELLEN

Probleem

Computers kennen dan wel getallen, maar niet zoals wij die kennen. Computers kennen in feite enkel 0 en 1: geen elektriciteit of wel elektriciteit, uit of aan, nee of ja. We hebben dus een systeem nodig dat al die kleine wel- of niet-signaaltjes omzet in interessante data, te beginnen met onze getallen.

Opdracht

5 leerlingen krijgen een bordje met een waarde op: 1, 2, 4, 8, of 16. Zij zijn nu bits en kunnen enkel aan- of uitstaan.

Bordje omhoog (= |) betekent "AAN": de waarde wordt meegeteld.

Bordje omlaag (= 0) betekent "UIT": de waarde wordt *niet* meegeteld.

Door de waardes op te tellen van de bits die "AAN" staan, kunnen nu getallen worden gevormd.

Voorbeelden:

$$000|0 = ?$$

Positie bord	UIT	JIT	UIT	AAN	UIT

$$\text{Totaal.} \quad 0 \times 16. + 0 \times 8. + 0 \times 4. + 1 \times 2 + 0 \times 1 = 2$$

$$0|||0 = ?$$

$$\text{Totaal.} \quad 0 \times 16 + 1 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1 = 14$$

$$||0|0 = ?$$

Waarde			0		0

Totaal. $1 \times 16 + 1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1 = 26$

Hoe meer bits je gebruikt, hoe groter het getal dat je kan maken. Als je bijvoorbeeld gebruik maakt van vier bits, gebruik je enkel bits 1, 2, 4, en 8. Het grootste getal dat je kan maken is dan $(1 \times 8) + (1 \times 4) + (1 \times 2) + (1 \times 1) = 15$. Gebruik je 5 bits (1, 2, 4, 8, en 16), dan is het grootste getal $(1 \times 16) + (1 \times 8) + (1 \times 4) + (1 \times 2) + (1 \times 1) = 31$

Opmerking

Bij binaire getallen worden de nullen die links staan, vaak weggelaten. "000|0" wordt dus "|0", "|||0" wordt "|||0", ... Om een binair getal te lezen, start je daarom altijd van rechts naar links. Maak eventueel gebruik van een schema zoals hieronder:

Om de waarde van de volgende bit te kennen, vermenigvuldig je de grootste bit met 2. Dus voor 8 komt 16, voor 16 komt 32, voor 32 komt 64, ...

Extra

Wat is het hoogst mogelijke getal dat je kan maken met 6 bits? Hoeveel bits heb je nodig om het getal 1000 te kunnen maken?

Waarom?

B. BINAIR RENDEREN

Probleem

In deel 2 las je het volgende:

“Computers kunnen eigenlijk geen beelden onthouden, alleen maar getallen. Elk onderdeelje (= pixel) van een foto is dus opgeslagen als code en moet ‘gerenderd’ worden door die code om te zetten in beeld via een afgesproken systeem.”

In het voorgaande deel dit:

“Computers kennen dan wel getallen, maar niet zoals wij die kennen. Computers kennen in feite enkel 0 en 1: geen elektriciteit of wel elektriciteit, uit of aan, niet of wel. We hebben dus een systeem nodig dat al die kleine wel- of niet-signaaltjes omzet in interessante data, te beginnen met onze getallen.”

Nu wordt het tijd om beide te combineren.

Opdracht

De pixeltekening wordt nog steeds gedefinieerd door het aantal witte en zwarte pixels, alleen wordt het aantal pixels nu uitgedrukt in bits, en niet in decimale getallen.

Voorbeeld:

					0 - - 0
					0 - - -
					- - - -

Werkwijze:

Zet eerst de binaire getallen om in leesbare (decimale) getallen en pas vervolgens het systeem uit het vorige deel toe om het beeld te renderen tot een tekening.

Oplossing:

		■			2 - 1 - 2
	■		■		
■				■	
	■		■		1 - 1 - 1 - 1 - 1
■	■	■	■	■	

Opmerking

Doe deze opdracht in groepjes van 2 en je kan volgende zaken observeren:

- Wie is inzichtelijk sterk of minder sterk?
- Wie kan goed samenwerken?
- Wie ziet zijn/haar eigen sterkte en zwakte? (rekenen vs. kleuren)
- Wie is competitief ingesteld?
- ...

Wedstrijd Secundair Onderwijs

Ontwerp met je klas een **originale pixeltekening van 20x20**, zet deze om naar de juiste binaire code, en stuur de code **voor 26 april 2019** naar wedstrijd@educentrum.be.

C. BINAIR RENDEREN... IN KLEUR!

Probleem

Totnogtoe hebben we enkel zwart-wit tekeningen gemaakt. Hoe kunnen we kleur in onze code brengen?

Opdracht

Vanaf nu werken we standaard met 8 bits (ook wel 1 byte genoemd). Deze 8 bits splitsen we op in 2 delen: 3 bits + 5 bits. De eerst 3 bits voorzien ons van kleurinformatie, de laatste 5 bits geven het aantal pixels dat die kleur krijgt.

We maken de volgende kleurafspraken:

- 0 = wit
- 1 = geel
- 2 = groen
- 3 = blauw
- 4 = rood
- 5 = roze
- 6 = bruin
- 7 = zwart

Voorbeeld:

					000 , 000 0000 , 000
					0 0000 , 0000 , 000 0000 , 0000 , 0 0000

Werkwijze:

1. Splits de bytes in 3 bits en 5 bits.
2. Zet beide binaire getallen om in leesbare (decimale) getallen. De eerste 3 bits geven informatie over de kleur, de laatste over het aantal pixels in die kleur.
3. Pas het systeem uit deel 1 toe om het beeld te renderen tot een tekening.

Oplissing:

					$7 \times 2 - 0 \times 1 - 7 \times 2$
					$0 \times 2 - 7 \times 1 - 0 \times 2$

					wit x 1 - bruin x 3 - wit x 1
					wit x 2 - zwart x 1 - wit x 2
					wit x 1 - zwart x 1 - rood x 1 - zwart x 1 - wit x 1

Wedstrijd

Zin in een grotere uitdaging? Ontwerp dan met je klas een **gekleurde pixeltekening van 20x20** voor onze wedstrijd!