A wireframe map of the Netherlands, composed of a dense network of white lines and dots on a black background, forming the outline of the country.

Kernmodule 1 IxD openFrameworks

edwin.vanouwerkerkmoria@hku.nl

Vorige keer...

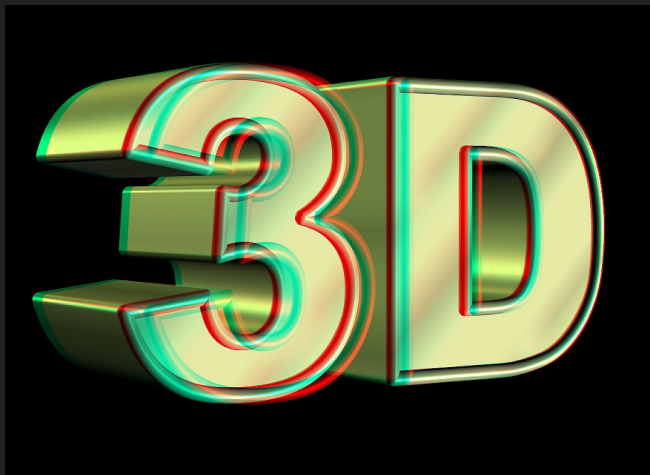
- OpenFrameworks
 - basis structuur van OF programma*
 - Simpel programma: *Bounce*
- oF koppelen aan Arduino: Firmata
- oF installatie/configuratie perikelen...
 - OS/X: *nightly build* van 24 sept: <http://openframeworks.cc/download> -> nightly builds

<https://goo.gl/WiFB6U>

Opdrachten vorige keer

- Mail mij de link naar je GitHub repository
- Koppel de *Bounce* app aan je Arduino
Laat bv een LEDje aan/uitgaan als je 'bouncet', of stel de snelheid of kleur in met een potmeter, start/stop de bal met een button, etc.
- Bekijk deze installatie van Erik Pirolt:
<https://www.youtube.com/watch?v=66A5uGBqJ6o>
Hoe zou je dit bouwen? Wat is er voor nodig? Beschrijf de hardware en software in een kort documentje in je GitHub repo.

Vandaag



- Simpele GUI's maken



THE THE AND THE
GOOD BAD UGLY

Firmata: the Good, the Bad, the Ugly

Good

Al je code op 1 plek: in oF applicatie

Firmata is compact & robuust

Bad

Je kunt niet direct op je Arduino kijken (black box).

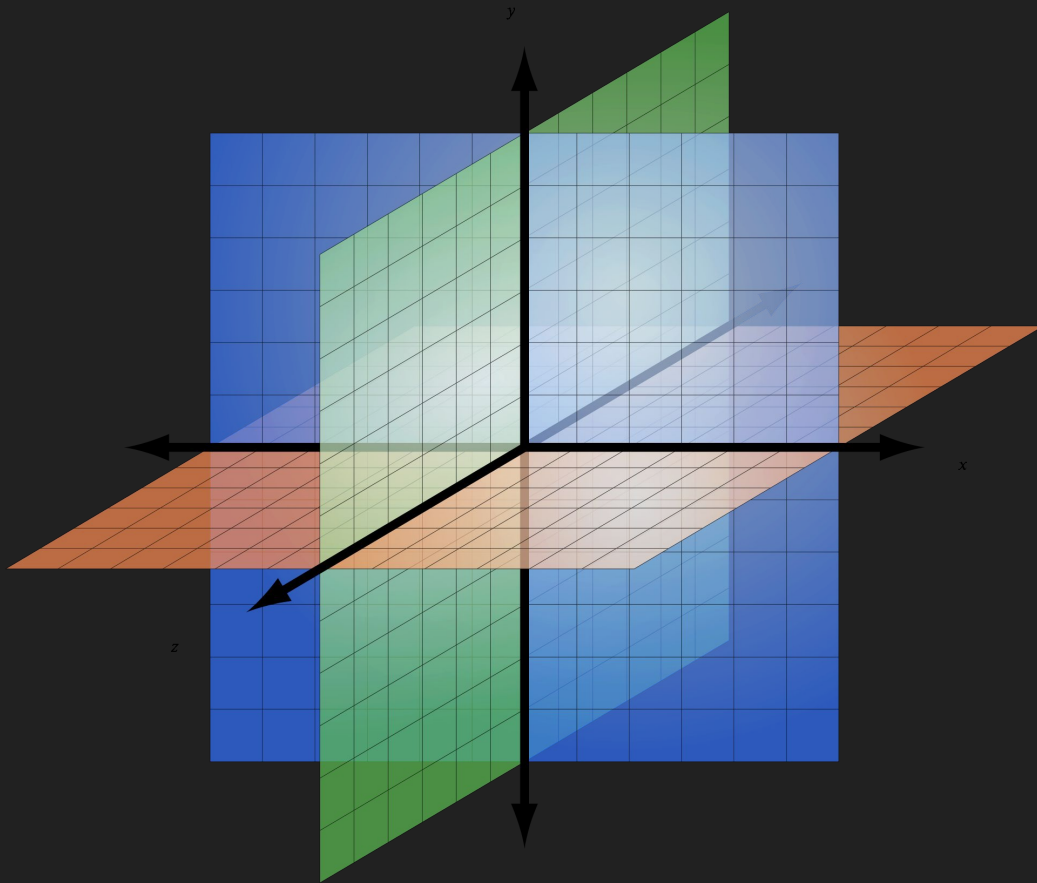
Soms toch test-programma voor Arduino schrijven...

Ugly

Je kunt alleen gebruik maken van wat er ingebouwd zit in Firmata.

Als sensors eigen libraries nodig hebben, zitten die meestal niet in Firmata...

Firmata is wel aan te passen (functionaliteit er bij/er af)



2D vs 3D graphics

Totnogtoe: 2D graphics: **x** & **y** coördinaten

3D graphics: extra coördinaat: **z** (diepte)

Coördinaten-ruimte: 3 vlakken:

Assen zijn verplaatsbaar over 't scherm: `ofTranslate()`

Je kunt ook alles draaien om die assen: `ofRotate()`

openFrameworks werkt eigenlijk altijd in 3D - maar die zie je niet altijd



Reageren op de buitenwereld

'events' : dingen die buiten je programma gebeuren

- toetsaanslagen
- muisklikken
- sensorwaarden van je Arduino...

OF vertelt je deze events via '*callback methodes*'

- `ofApp::keyPressed(int key);`
- `ofApp::mousePressed(int x, int y, int button);`



GUI's maken

Dynamische instellingen aan je programma

- grootte van de bal(len)
- snelheid x en y
- ...

Als je ambitieus bent: verschillende vormen laten 'stuiteren' (rechthoeken, driehoeken , etc...)

Gebruik `ofParameter<>` om variabelen aan GUI elementen te binden

Opdrachten volgende keer:

- Maak een 3D variant van de Arduino Bounce-app die je gemaakt hebt.

Zie evt nog de voorbeeldcode op GitHub

- Ga uit van Planetarium code die we samen bekeken hebben.
Voeg toe dat de planeet (of planeten?) daadwerkelijk om de zon draaien.
hint: denk aan de 3DCoords voorbeeld applicatie.

Koppel de app aan je Arduino. Kies zelf wat je wilt koppelen: draaisnelheid, belichting aan/uit, in/uitzoomen...