Belgium





UGENT VOLKSSTERRENWACHT ARMAND PIEN

Workshop Astro Pi Mission ZERO

Lespakket voor basisonderwijs en secundair onderwijs Lerarengids



Programmeer in Python een minicomputer en stuur je code naar het ISS
 Ideale beginnersoefening om coderen te leren kennen
 Volg onze korte video's met gerichte instructies en uitleg







OVER ESERO BELGIUM

ESERO is een scholenprogramma van de Europese Ruimtevaartorganisatie ESA. Het doel van dit programma is leraren van basisonderwijs en middelbaar onderwijs helpen om het populaire thema ruimtevaart in de klas te brengen, binnen hun lesopdracht. Dit doen we op drie manieren: **lesmateriaal** (online), **lerarenvormingen**, en **STEM projecten voor scholen**. Het aanbod is volledig gratis voor leraren in beroep en leraren in opleiding, en is afgestemd op de eindtermen in het onderwijs. Hedendaagse en inspirerende ruimtevaartmissies vormen de context diverse schoolvakken.

WWW.ESERO.BE



ESA Education beheert en coördineert alle ESERO's in Europa. Elke ESERO bestaat dankzij een cofinanciering van ESA en nationale partners. Het federaal wetenschapsbeleid (BELSPO) is de cofinancierende partner voor ESERO Belgium.







Workshop Astro Pi Misson Zero

Handleiding voor leraren

Kenmerken

Doelgroep	Leraren lager en secundair					
Туре	Computeropdracht rond coderen van een Astro Pi simulator.					
Hoeveel lestijden?	1 tot 2 lesuren.					
Benodigdheden	 Computer met internetverbinding per kind Beamer Account voor leraar op Astro Pi Mission Zero platform Werkblaadje: Spiekblaadje bij het coderen 					
Wat de leerlingen gaan leren	 Een Astro Pi is een minicomputer Waarom codeertaal gebruiken? Introductie in Python Toepassingen van Python in het dagelijkse leven 					
Samenvatting	Via deze les proberen we de leerlingen via een eenvoudige oefening warm te maken voor coderen. Ze laten een afbeelding van 8x8 pixels op een LED-schermpje tevoorschijn komen. De achtergrond van de afbeelding laten ze door gebruik van een kleurensensor veranderen. Hun beeld blijft maximum 28 seconden zichtbaar.					





Colofon

Eerste uitgave	Juni 2024
----------------	-----------

Tweede uitgave /

Gebruik Deze cursus mag gratis gebruikt worden voor educatieve doeleinden. Als je onderdelen eruit kopieert, dan moet dit gebeuren met een verwijzing naar het origineel. De recentste versie van de cursus kan je downloaden op www.esero.be

AUTEURS

ESERO Belgium Cursusinhoud, lay-out, (Leonie De Clercq, Pieter Mestdagh).

UGent Deze les is samengesteld en wordt uitgevoerd tijdens klasbezoeken door de educatieve medewerkers van de volkssterrenwacht.

Uw mening is belangrijk	ESERO Belgium werkt altijd aan een betere kwaliteit. Gebruikers van onze cursussen worden aangemoedigd om feedback te geven via de contact gegevens op
	www.esero.be. Wanneer uw reactie bijdraagt aan een belangrijke verbetering van de cursus, dan wordt uw naam opgenomen in de auteurslijst (colofon) in de nieuwe online editie. Op die manier helpen gebruikers de andere, toekomstige gebruikers om beter lesmateriaal te krijgen.



Inhoud

Kenmerken	4
Colofon	5
Over dit lespakket	8
Doel	8
Waarom Pythoncode in onderwijs?	8
2 Voorbereiding van de les	9
Account voor de leerkracht	9
Stap 1: astro-pi.org	9
Stap 2: Mission Zero	9
Stap 3: Register for Mission Zero	10
Klasopstelling	10
Computers die de leerlingen mogen gebruiken	11
Stap 1: Step-by-step guide	11
Stap 2: Raspberry Pi foundation	11
Stap 3: Display an image	12
Stap 4: Mission Zero starter project	12
3 Lesactiviteit	13
Inleiding	13
Doel van de dag	13
Wat is een Astro Pi?	13
Een Astro Pi in het ISS v.s. in de klas	14
Wat is programmeren in Python?	14
Voorbeelden van Python in het dagelijkse leven	15
Instructie & Verwerking	16
Herhaling van de opdracht: tonen van de simulatie	16
Overlopen van de code op hun blaadje	17
Openen van de simulator op hun computer	18
Verschillende # noteren in de code	19
Liberaries	20
STAP 1: Kleuren aanmaken	21
STAP 2: Maak de tekening	24
STAP 3: Toon op display en sleep	28
EERSTE RUN TEST	29
STAP 4: Kleuren waarnemen met sensor	31



TWEEDE RUN TEST	32
STAP 5: Loop	32
DERDE RUN TEST	33
STAP 6: Stopscherm	33
VIERDE RUN TEST	34
Afronding	35
Code indienen	35
Toonmoment	36
OPGELET	38



1 Over dit lespakket

Doel

Een introductie in de wereld van programmeren met Python

Het doel van dit lespakket is om de leerlingen een eerste kennismaking te geven met de codeertaal Python. Dit doen ze door een simulatie van een Astro Pi computer te coderen op een internetapplicatie. De verschillende codeonderdelen worden stapsgewijs aangebracht om de nodige denkwijze bij coderen te stimuleren.

Waarom Pythoncode in onderwijs?

Meest gebruikte codetaal in de wereld

Python heeft zijn populariteit in de wereld te danken aan zijn eenvoud. De taal is zodanig ontwikkeld dat het makkelijk te lezen en te leren valt. Bovendien is het gebruiksvriendelijk voor verschillende toepassingen.

Python is gratis en bevat heel wat kant en klare coderegels die je vrij kan gebruiken en aanpassen. Dit maakt coderen nog vlotter.

Python wordt nu door heel wat verschillende bedrijven gebruikt om data te verwerken. Hierbij kunnen ze onderzoeken hoe goed hun product presteert op de markt en waar ze verbeteringen moeten aanbrengen. Je kan er ook gemakkelijk simulaties mee ontwikkelen om een toekomstbeeld te scheppen en zo de groei in kaart te brengen.

Kunnen werken met Python wordt door heel wat bedrijven gezien als 'basisskill'. Er vroeg bijzijn is dus erg belangrijk. Daar kan het onderwijs een belangrijke rol in spelen. Hoe meer kinderen bekend zijn met de basisprincipes van het programmeren, hoe vlotter zij later hun weg kunnen vinden in deze wereld.





2 Voorbereiding van de les

Account voor de leerkracht

Als begleider dien je een account aan te maken op het AstroPi-platform.Via dit account krijg je een klascode die de leerlingen moeten gebruiken om hun programmaatje in te dienen. Later kan jij als leerkracht na het indienen de verschillende werkjes ook bekijken of tonen aan de medeleerlingen.

Stap 1: astro-pi.org



Je komt op de startpagina van Astro Pi terecht. Daar krijg je kort info over de verschillende missies die je met je klas kan uitvoeren.

Stap 2: Mission Zero



Klik bovenaan op 'Mission Zero'. Hier vind je alle informatie die je nodig hebt alvorens je meedoet aan deze missie.

- Data
- Leeftijd
- Duur van de activiteit
- Criteria waaraan je moet voldoen
- Hoe je een account aanmaakt

Juni 2024



Stap 3: Register for Mission Zero



Wanneer je wat naar onderen scrollt, vind je een link terug om als mentor een account aan te maken. Via deze weg krijg je je klascode die de leerlingen kunnen gebruiken om het werkje in te dienen.

Vul de vragenlijst in. Op het einde krijg je de klascode te zien. Die krijg je ook via mail of kan je terugvinden op je account.

Via deze link kan je alle stappen nog eens bekijken die nodig zijn bij het aanmaken van een account.

https://www.youtube.com/watch?v=-K7xLDD6B3I&ab_channel=ESEROBelgium

Klasopstelling

Zorg ervoor dat alle computers voorzien zijn van internettoegang en stroom. We raden ook aan om computermuizen te voorzien. We merken dat leerlingen na verloop van tijd gefrustreerd raken met een mousepad omdat je voor coderen heel nauwkeurig je cursor moet kunnen plaatsen in de codelijnen.

Zorg ervoor dat alle leerlingen het beeldscherm vooraan kunnen meevolgen.





Computers die de leerlingen mogen gebruiken

Plaats een directe link naar de simulator op de computer. Dit kan door een link van de simulator door te sturen via mail naar de leerlingen of via een intern leerplatform van de school. Zo vermijd je dat je leerlingen terechtkomen op de officiële AstroPi-pagina, waar ze alle stappen van de code al zouden terugvinden.

Leerlingen kunnen ook zelfstandig aan dit project werken als ze dit wensen.

Stap 1: Step-by-step guide



- Ga naar de astropi.org pagina.
- Klik op Mission Zero.
- Scroll wat naar onder en klik op de grote roze knop 'Step-by-step guide'.

Stap 2: Raspberry Pi foundation

You will make	
What is an Astro Pi?	
Display an image	Mission Zero 2023/24 has now closed.
Sense a colour	You can still complete the activity but will not be able to submit your program to run on the ISS.
What next – more Astro Pi projects	The next round of the challenge will launch in September 2024. Check social media and sign up to the newsletter on the <u>Astro Pi website</u> for updates.
	Or use this project to create something you love for Coolest Projects! .
Print this project	You will make
Save your progress & collect badges!	Your project will set the background colour of an image to the colour that the Astro Pi detects. This will make the International Space Station (ISS) more colourful for the astronauts on board. Your code will use the colour luminosity sensor on the new Mark II Astro Pi computer's Sense HAT to make this happen.
) 📀 🛞 😴 (Here is an example of the kind of program you could make to run on an Astro Pi in space.
Sog in or sign up	

Je komt op de Rasberry Pi Foundation pagina terecht. Deze pagina leidt je door de oefening. Alles wordt stap voor stap uitgelegd en telkens krijg je de nodige code die je verder kan aanvullen om de oefening te vervolledigen.



Stap 3: Display an image

You will make	
What is an Astro Pi?	Mission Zaro 2023/24 has now closed
Display an image Sense a colour What next - more Astro Piprojects	You can still complete the activity but will not be able to submit your program to run on the ISS. The next round of the challenge will launch in September 2024. Check social media and sign up to the newsletter on the <u>Astro Pi website</u> for updates.

Links op de pagina vind je snelkoppelingen naar de verschillende stappen in de oefening. Klik op 'Display an image' om de link naar de simulator te vinden.

Stap 4: Mission Zero starter project

Open the Mission Zero starter project.

Link= https://missions.astro-pi.org/mz/code_submissions/newb

Wanneer je bij 'Display an image' verder naar beneden scrolt, zie je bovenstaande link in een groene kader staan. Deze link brengt je naar de simulator om een Astro Pi te coderen.



3 Lesactiviteit

Inleiding

Doel van de dag

Programma overlopen

Dia 1

In deze les gaan de leerlingen een Astro Pi leren coderen. De les bestaat uit drie belangrijke delen:

- Wat is een Astro Pi?
- Wat is coderen?
- Coderen.

Doel van de missie

Dia 2

Jouw leerlingen programmeren een Astro Pi computer zo dat die een afbeelding toont die de maken heeft met fauna of flora. De computer heeft een 8X8 pixel display, dat per pixel verschillende kleuren kan tonen. Ook die kleuren programmeren je leerlingen. Meer nog: je klas zal worden uitgedaagd om terwijl hun programma loopt, één van de kleuren aan te passen aan de omgeving waarin de Astro Pi computer staat. Daarvoor kunnen ze een kleursensor gebruiken, die de omgeving voortdurend waarneemt. Dat stelt de leerlingen in staat om de kleursensor te laten 'kijken' naar de achtergrondkleuren, en die tevoorschijn te toveren in hun programma. In totaal mag hun creatie niet langer dan 30 seconden duren. Daarna verdwijnt hun tekening en toont het display een effen kleur. A

Wat is een Astro Pi?

Minicomputer

Dia 3-4

Een Astro Pi is een minicomputer die uit twee delen bestaat. Het onderste gedeelte is de computer zelf, die programmeertaal verwerkt. Je vindt er verschillende poorten, zoals een internetpoort, HDMI-aansluiting, USB-poorten, Ook de voeding loopt via dit gedeelte en stuurt het tweede deel aan.

Sensehat

Dia 5

Dat tweede deel noemen we sensehat. De letterlijke vertaling is het 'voelhoedje'. De naam spreekt voor zichzelf. Het is een deel dat je op de astropicomputer zet. Dit tweede deel heeft als doel om de omgeving te 'voelen'. Het kan de temperatuur, de luchtdruk, de



luchtvochtigheid meten. De sensor die de leerlingen vandaag gaan gebruiken is de kleurensensor.

Er bestaan zo verschillende soorten sensehats. Zo heb je ook hoedjes waar je lampjes kan op zetten, een ventilator kan opzetten of zelfs wielen kan aan bevestigen. Noem maar op. Je kan het zo gek niet bedenken of er bestaat een hoedje voor.

Een Astro Pi in het ISS v.s. in de klas

Astro Pi in het ISS

Dia 6-7

De Astro Pi die op aarde gebruikt wordt, is niet dezelfde als degene die je in het ISS vindt. De Astro Pi in het ISS heeft een 'hoesje' aan. Deze casing beschermt de Astro Pi tegen de 'harde' omgeving in het ISS. Het moet tegen trillingen, botsingen, vocht, ... kunnen. Als de astronauten per ongeluk tegen de Astro Pi zouden bosten, kan dat niet veel kwaad.

Astro Pi op aarde

Dia 8

De hardware van een Astro Pi is een 'naakte' versie. Hier zie je de bedrading, pinnen en bevestigen zitten. Belangrijk, want zoals hiervoor al vermeld kan het 'hoedje' van een Astro Pi aangepast worden. Je kan er extra hardware of andere sensoren aan toevoegen. Je kan meerdere Astro Pi's met elkaar verbinden.

Astro Pi simulator

Dia 9

Een Astro Pi is niet zo goedkoop in de aankoop. Daarom heeft ESA een simulator voorzien zodat ook scholen met enig budget aan deze missie kunnen meedoen. De simulatie ziet er uit zoals een Astro Pi in het ISS. Alleen is deze natuurlijk virtueel. Het is zodanig gemaakt dat de virtuele Astro Pi net hetzelfde kan als de echte Astro Pi in het ISS.

Wat is programmeren in Python?

Domme Astro Pi

Dia 10

Een Astro Pi computer is net zoals alle andere computers... dom. De leerlingen moeten de Astro Pi letterlijk alles commanderen. Ze moeten de Astro Pi duidelijk maken welke sensehat er op bevestigd is. Ze moeten de Astro Pi vertellen welke sensor hij hiervan moet gebruiken. Ze moeten duidelijk maken dat de sensehat ook een LEDscherm heeft. Ze moeten elke led aanspreken. Alles, alles, alles moet tot in het detail uitgelegd worden aan de domme computer. Van zodra er iets niet duidelijk gecommandeerd is, weet de computer niet wat hij moet doen.





Codeertaal

Dia 10

Computertaal en mensentaal zijn alles behalve hetzelfde. Mensen spreken met klanken. Die klanken vormen woorden. Woorden die in een bepaalde volgorde staan, zijn zinnen. Door intonatie en gezichtsuitdrukkingen kunnen wij communiceren met elkaar. Een computer werkt met binaire code. Binair wil zeggen 2 waarden. Je hebt de waarde nul en één. Dat staat voor aan en uit, ja en neen, start en stop. Maar hoe kan je nu met twee tekens gecompliceerde commando's geven aan die domme computer? Daarvoor dient codeertaal. Codeertaal is de tussenstap om mensentaal om te zetten naar computertaal.

Mensen kunnen met eenvoudige tekens en woorden een commando schrijven.

Verschillende codeertalen

Dia 10

Vandaag heb je heel wat verschillende codeertalen. Je hebt Python, Java, C++, Assembly, ... Elke code heeft zijn specifieke toepassingen. Elke codeertaal heeft zijn voordelen en nadelen, gemakken en ongemakken. De leerlingen zullen gebruik maken van Python. Deze taal wordt vandaag door heel wat bedrijven gebruikt. Wij kozen bewust voor deze taal omdat het kinderen een eerste zet kan zetten in coderen die hun hele leven bruikbaar zal zijn.

Voorbeelden van Python in het dagelijkse leven

Waarom Python

Dia 11

Python heeft zijn populariteit in de gewonnen door zijn eenvoud. De taal is zodanig ontwikkeld dat het eenvoudig te lezen en te leren valt. Bovendien is het gebruiksvriendelijk voor verschillende toepassingen.

Python is gratis en bevat heel wat kant en klare coderegels die je vrij kan gebruiken en aanpassen. Dit maakt coderen nog vlotter.

Toepassing

Dia 11

Python wordt in eerste instantie gebruikt om data te verwerken. Hier heb je een overzicht van verschillende toepassingen:

- Websitebeheer
- Verwerken van wetenschappelijke data
- Al en robotica
- Educatie: leren van codeertaal
- Game Development
- Software Development

Juni 2024



Hieronder enkel voorbeelden uit het dagelijkse leven waarbij Python werd/wordt gebruikt:

- Dropbox
- Reddit
- Yahoo
- Netflix
- Instagram
- Uber
- Facebook
- Spotify
- Lyft
- Youtube
- NASA: Perseverance etc, ...
- Battlefield 2
- Mount and blade
- Sims 4
- MODs voor games
- Tesla

Instructie & Verwerking

Herhaling van de opdracht: tonen van de simulatie

Dia 12

Het doel is dat de leerlingen een Astro Pi computer gaan programmeren. Dit doen ze via een simulatie op de computer, niet op een echte Astro Pi.

De leerlingen moeten een afbeelding op dit computertje laten tevoorschijn toveren. Deze tekening moet te maken hebben met Fauna en Flora'. Dat wil zeggen 'dieren en planten'. De tekening bestaat uit 8x8 pixels. Elke pixel wordt voorgesteld met een LED-lampje op de AstroPi.

In hun tekening laten ze een kleur veranderen moet te maken hebben via de kleurensensor. De kleurensensor kijkt naar de kleur die op dat moment voor hem zichtbaar is. Zweeft er op dat moment een astronaut met een blauw pak voorbij de sensor, dan gaat er een kleur in de tekening in blauw veranderen. Stel dat er een rode bal passeert en de kleurensensor ziet dit, dan gaat er een kleur in de leerlingen hun tekening veranderen naar die rode kleur.

De tekening is maximum 30 seconden zichtbaar op de Astro Pi. Het programmaatje moet eindigen met een leeg scherm om het einde van hun code te duiden

Wanneer hun code aan alle eisen voldoet, mag het ingediend worden.

Indien er nog tijd over is, kan er nog één code afgespeeld worden op de hardware.

Let op: wanneer de data van indienen verstreken is bij het project, wordt de code niet meer in het ISS afgespeeld.





Overlopen van de code op hun blaadje

Spiekbriefje

Dia 13

Deel de leerlingen het werkblaadje 'Spiekbriefje bij het coderen' uit. Op dit werkblaadje staat alle info die ze nodig hebben om mee te volgen bij het programmeren. Elke lijn code staat zodanig genoteerd dat de leerlingen deze enkel nog moeten aanvullen en overtypen.

Instructie bij het spiekbriefje

Dia 13

Op het spiekbriefje kan je zien dat de code die de leerlingen moeten schrijven uit zes grote stappen bestaan.

STAP	CODE					
>	# Import the libraries					
>	# Set up the Sense HAT					
→	# Set up the colour sensor					
→	# Add colour variables and image					
Stap 1	# Kleuren aanmaken					
Stap 5	# <u>Herhalen</u> in <u>een</u> Loop					
Stap 4	# Kleur waarnemen met sensor					
Stap 2	# <u>Tekening maken</u>					
→ Stap3	# Display the image					
Stap 6	# <u>Stopscherm</u> tonen					

Merk op dat de stappen op het blaadje niet in de juiste volgorde staan. Dat is normaal, want coderen vereist een specifieke volgorde.

Eerst moet je de computer duidelijk maken hoe kleuren samengesteld zijn voordat je ze in de tekening kunt gebruiken. Vervolgens teken je de afbeelding in codetaal. Daarna moet je het LED - display duidelijk maken welke kleur uit d tekening bij welke LED moet terechtkomen.

Pas daarna kun je de kleurensensor gebruiken, want deze moet weten welke kleuren/LED's je wilt aanpassen in je tekening. Vervolgens herhaal je alles in een loop, zodat de code meerdere keren door de Astro Pi wordt gelezen en uitgevoerd. Tot slot eindigt je programma met een leeg stopscherm.

Later zal duidelijk worden waarom de verschillende stappen op die specifieke plaats in de code staan.



Openen van de simulator op hun computer

Dia 14

Laat de leerlingen de simulator openen. Overloop met hen het scherm dat ze voor zich zien.



Bron: ESERO BE VL

Zoals al eerder vermeld, zullen de leerlingen geen gebruik maken van een echte Astro Pi, maar van een virtuele.

Links van het scherm hebben leerlingen plaats om code te schrijven. Zoals je ziet, zijn er hier al enkele lijnen op ingevuld. Daar blijven ze af. Deze lijnen zijn op voorhand gegeven om de oefening korter en eenvoudiger te maken. Wanneer de leerlingen die aanpassen of verwijderen, za de simulatie niet meer werken.

Rechts onder de Astro Pi zie je het controlepaneel. Daar kan je alle sensoren manipuleren. Voor de opdracht zijn temperatuur, vochtigheid of luchtdruk niet nodig. Deze sliders kunnen we negeren. De kleursensor onderaan hebben we dan weer wel nodig. Wanneer de leerlingen op het gekleurde vakje klikken, zien ze een kleurenraster. Door met de muis hierover te bewegen, veranderen ze de kleur die hun Astro Pi 'ziet'.



Verschillende # noteren in de code

Betekenis van de #

Dia 15

Staat er een # voor een lijn? Dat betekent dat dat een lijn is die de computer niet zal lezen. Je kan dus een nota voor jezelf of voor anderen maken.

Je kan de tekst achter een # zien als een notitie voor de codeerder of de externen. In deze code gebruiken wij de # om de leerlingen duidelijk te maken wat de verschillende stappen zijn in de oefening en maken we het de leerlingen makkelijker om mee te volgen met de les.

Noteren in simulatie

Dia 15

In de simulatie zelf staan er al enkele #. Die vind je op het spiekbriefje terug met een oranje kleur. Nu is het de bedoeling dat de leerlingen de zwarte lijnen code op de juiste plaats in de simulatie typen.

De leerlingen moeten goed opletten dat ze geen leestekens gebruiken in hun code.

Niet:

- # Kleuren aanmaken.
- # Kleuren aanmaken

Wel:

- # Kleuren aanmaken
- # kleuren aanmaken





Het zou er bij hen zo moeten uitzien:

	in.py	Vis
	Import the libraries	
2	rom sense_hat import SenseHat	
	rom time import sleep	
4		
-	Set up the Sense HAT	
(ense = SenseHat()	
7	ense.set_rotation(270, False)	
	Sat up the colour concor	
10	ense color gain - 60 # Set the sensitivity of the s	
11	ense color integration cycles = $64 \text{ # The interval a}$	
12	enservoron rincegi deron_cycres = 04 # The rince var e	
1	Add colour variables and image	
14		
19	Kleuren aanmaken	
16		
17	Herhalen in een loop	
18		
19	Kleur waarnemen in een sensor	
26		
21	Tekening maken	
22		Spac
23		
24	Display the image	
25		
26		
27	Stopscherm tonen	

Let op: Het kan zijn dat de lijnen die na de # komen van kleur veranderen, dat kan geen kwaad.

Liberaries

Wat zijn Libraries?

Dia 15

Python heeft bibliotheken met voorgeschreven codes die je kan gebruiken. Dat spaart je tijd doordat je geen 20-tal extra regels moet uitschrijven of zelf op zoek moet gaan naar de ideale code. Deze libraries worden aangevuld door andere Python-gebruikers die het niet erg vinden om hun werk te delen.



Betekenis in onze missie

Dia 15

1 # Import the libraries
2 from sense_hat import SenseHat
3 from time import sleep

We vermeldden eerder al dat een computer 'dom' is: je moet tot in de kleinste details duidelijk maken wat je wil bereiken en wat de computer voor je moet doen. Zo ook welk 'hoedje' hij heeft om te gebruiken. De code op lijn 2 is nodig om de Astro Pi duidelijk te maken wat er allemaal aanwezig is op dit hoedje. Het schrijft alle LEDs aan, het schrijft de sensoren aan. Indien we deze code zelf moeten uitschrijven, zijn we tientallen lijnen verder. Het feit dat we dit dus uit de libraries kunnen halen, spaart ons werk uit.

Op lijn 3 staat de code die ervoor zorgt dat onze Astro Pi na het runnen van de code even in pauze gaat. Verder in de code wordt er dan gespecifieerd voor hoelang de pauze doorgaat. Gedurende die tijd kan er geen nieuw signaal van buitenaf opgenomen worden.

STAP 1: Kleuren aanmaken

RGB-waarden

Dia 16

LEDs werken als volgt: ze hebben ofwel een rode, groene of blauwe kleur. Samen maken ze, naargelang hun lichtintensiteit, alle kleuren van de regenboog.

De LEDs die gebruikt worden in onze Astro Pi hebben elk de mogelijkheid om ofwel rood, groen of blauw licht te schijnen. Zo één ledje kan dus alle kleuren van de regenboog creëren.





Primaire kleuren bij verf Subtractieve kleurenmenging

Deze werking kan je terugvinden in heel wat verschillende schermen. Denk maar aan een tv, monitor, gsm, smartphone, ... als je met een vergrootglas naar zo'n scherm zou kijken, zou je zien dat elke pixel bestaat uit drie kleuren (rood, groen en blauw). Heb je geen vergrootglas? Dan kan je het ook eens proberen door een druppeltje water op een scherm te plaatsen.



Tip: Je kan additieve kleurenmenging aantonen met een 3-kleuren lamp die gebruikt wordt tijdens de lessen optica.

Onderaan de virtuele Astro Pi kan je de kleurensensor manipuleren. Laat de leerlingen daar verschillende kleuren kiezen. Wijs hen erop dat elke kleur gemaakt wordt door drie verschillende kleurwaarden te mengen: rood, groen en blauw. (vandaar de afkorting RGB-kleuren).

Vandaar de RGB kleuren. Elke waarde gaat van een minimum van nul (geen intensiteit) naar een maximum waarde van 255 (volle intensiteit).



Kleuren kiezen

Dia 17-18

Om een kleur te coderen, heb je dus telkens drie RGB-waarden nodig. Op het spiekblaadje hebben we enkele voorbeelden van verschillende kleuren geplaatst waaruit de leerlingen kunnen kiezen. Als ze willen, mogen ze via de kleurensensor ook andere kleuren kiezen.

Om een kleur aan te maken in code, moet je eerst de kleur een naam geven. Wij hebben ervoor gekozen om een letter uit het alfabet te nemen. Dit maakt het coderen van de tekening een pak eenvoudiger.

Wel:

- a = (0, 0, 0)
- a=(0,0,0)
- A = (0, 0, 0)

Niet:

- Wit = (0,0,0)
- Wit = (0,0,0)
- WIT = (0,0,0)
- A=(0.0.0)
- A=(0;0;0)
- A: (0,0,0)
- -1=(0,0,0)





Zoals je kan zien in de voorbeelden maakt het niet uit of je spaties gebruikt in de code. Python is echter hoofdletter-, leesteken- en cijfergevoelig. Op die drie aspecten moet je dus zeker letten als je codetaal invoert.

Als je een kleur een naam met een hoofdletter hebt gegeven, moet je verder in de code ook steeds een hoofdletter gebruiken als je naar die kleur verwijst.

Maak geen gebruik van punten of puntkomma's, want elk teken heeft een eigen taak in de Pythoncode. Cijfers zijn in Python enkel waarden.

Jij als leerkracht mag de leerlingen laten kiezen uit hoeveel kleuren hun tekening mag bestaan. Wij raden uit ervaring aan dat te veel kleuren het coderen enkel ingewikkeld en onoverzichtelijk maakt. Daarom verkiezen wij minimum drie maximum vijf kleuren per tekening.

Kleuren aanmaken - coderen

Dia 19

De leerlingen coderen hun kleuren onder bovenstaande lijn.

Ze kunnen er zelf voor kiezen of ze met een # de naam van het kleur erbij noteren. Het zou er ongeveer zo moeten uitzien:



Een kleur krijgt in de code een naam. In het spiekbriefje hebben die telkens een letter van het alfabet gekregen. Ze kunnen ook een andere naam krijgen, maar om het coderen simpel te houden, kiezen we voor een eenvoudige letter en geen voluit geschreven kleur.



STAP 2: Maak de tekening

Pixilart: met kleurpotloden

Dia 20

De tekening die de AstroPi moet tonen bestaat uit 8x8 pixels en beeldt een plant of dier uit.

Op hun spiekbriefje kunnen ze tot vier keer een leuke tekening proberen maken vooraleer die echt te gaan programmeren.

Bijvoorbeeld:

Maak hier je 8X8 pixeltekening



Pixelart: met online tool

Dia 20

De leerlingen kunnen ook gebruik maken van een internetapplicatie.

- Daar gaan de leerlingen als volgt te werk:
- Tik in de zoekbalk 'pixilart.com'
- Klik op 'Start Drawing'





Klik in de linker menu op 'New' •

Autosave Loaded		×
New Save Drawing Download .png.gif Settings Key Bindings Custom Font Mobile App About Pixilart Advertisement - <u>Go Ad Erred</u>	Looks like you didn't save your latest changes and autosave has loaded them for you. Click 'New Drawing' if you would like to start a new drawing. You can disable autosave in settings (of). CONTINUE	
ING Fietsverzekering		
ING Belgie Meer info		

- Geef twee maal de waarde 8 in bij 'Width' en 'Height'. Klik op 'New Drawing' •
- •

New Drawing		×									
P Black Canyos	Blank Canvas Start a new drawing. Max 1000x1000 size. Width B Height B										
Drawing Bases	Ig Bases Color Malette Select octor PALETTE										
🖕 Open .pixil 🗈 Open Image	pixil NEW DRAWING										
o¢ Settings ♣ Tutorials IC ❶ Undates	Today's drawing challenge is <u>Dragon</u> . See today's entries										
	Canvas Presets FAVICON RANDOM 32X32 645464 tatite active a										
	128X128 100X100 256X144 500X500 128x128 100x100 256x144 500x500 Want to learn pixel art? Pixel art tutorials 100x100 100x100 100x100										



De leerlingen hebben de nu de kans om hun tekening te maken.



Opmerking: Omdat deze site gratis is, heb je wel last van reclame-pop-ups.

Coderen op papier (spiekbriefje)

Dia 21

De leerlingen hebben nu een tekening gemaakt op papier of online waarbij elk vakje van het 8X8 grid een pixel voorstelt. Ze hebben nu elke pixel een kleur toegekend.

Elke kleur hebben ze in het voorgaande deel een naam gegeven door middel van een letter uit het alfabet.

Maar hoe ziet dat er nu uit, kleuren met code? Hiervoor hebben ze de keuze of ze eerst op papier hun code gaan invullen, of onmiddellijk over gaan naar de simulatie.

Manier 1: op tekening spiekbriefje							rief	je	Ν	Manier	r 2: in code spiekbriefje
			Ρ	og	ing	1			Stap 2	2 # <u>Tekening</u> maken	
	U	3	8	3	Э	8	W	V			$[\omega, \chi, \chi, \chi, \chi, \chi, \chi, \psi, \psi],$
	w	ω	۲	۲	۲	r	V	W			μ, μ, τ, τ, τ, τ, ψ, ψ,
	W	٢	r	r	r	r	r	V			$\psi, \tau, \tau, \tau, \tau, \tau, \tau, \tau, h, \psi$
	۲	3	r	З	Ł	8	۲	r			r, g, r, g, r, g, r, v, ,
	P	۲	r	۲	۲	ŀ	r	ω			(1) [1] [2]
	W	r	3	ŀ	ð	r	2	W			$(\mathcal{A}, \mathcal{A}, A$
	ω	ω	r	8	r	w	L	ω			$[\psi, \psi, \omega, \kappa, \omega, \psi, \psi, w, v]_1$
	ω	w	w	r	ω	w	ω	w	1		

Ze kunnen op twee manieren de code eerst op papier schrijven:



Coderen in simulatie

Dia 22

De leerlingen hebben ook de keuze om hun code onmiddellijk in de simulatie te zetten. De code bestaat uit twee delen:

- Naam van de tekening
 Wanneer je de tekening een naam geeft, wordt het makkelijker om later naar dit stuk code te verwijzen. Doe je dat niet, dan moet je alles terug kopiëren. Dit is niet alleen onbegonnen werk, het maakt de volledige code ook nog eens onoverzichtelijk.
- Kleuren van de leds
 Vervolgens moet je elke led een kleur toewijzen. hiervoor typ je telkens de letter van je kleur, gescheiden door een komma Elke komma bakent dus een led af.

De code zou er zo moeten uitzien:





STAP 3: Toon op display en sleep

De tekening is in Python omgezet, nu is het zaak om ervoor te zorgen dat die op het LED-schermpje getoond wordt. Dat vereist een nieuw stuk code, die uit twee stappen bestaat:

Pixels aanschrijven

Dia 23

In de vorige stap hebben we elke led een kleur toegewezen. Nu moeten we nog de code schrijven die die kleuren overzet naar de leds.

Onder stap 3 '# Display the image' gaan leerlingen hun code schrijven. Die code ziet er als volgt uit: sense.set_pixels(...)

De 'sense' is de naam die we aan onze SenseHat(), het 'hoedje', gegeven hebben. Zie onderstaande lijn code.

```
5 # Set up the Sense HAT
6 sense = SenseHat()
```

- '.set' wil 'settings' zeggen. Je gaat de SenseHat() programmeren met volgende code.
- '_pixels' wil zeggen dat je de pixels gaat coderen van de SenseHat().
- '(...)' wil zeggen dat je hier je commando gaat neerschrijven. Hier noteren ze de naam van hun tekening in. Bij ons voorbeeld is dat 'Aardbei'. Merk op dat er hier gelet wordt op de hoofdletters.

Slaapmodus

Dia 23

Wanneer de code gespeeld heeft, moet de Astro Pi eerst even pauzeren. Dat kan door de commando 'sleep(...)' te gebruiken. Hier verwijzen we graag terug naar de eerste lijnen code waar we de libraries hebben aangesproken.

```
1 # Import the libraries
2 from sense_hat import SenseHat
3 from time import sleep
```

Tussen de haakjes schrijf je de waarde '1'. Deze waarde stelt het aantal seconden voor dat je Astro Pi gaat pauzeren alvorens hij nog eens door de code gaat. In deze code gaat hij dus 1 seconde lang 'slapen'.



De code zou er als volgt moeten uitzien:



EERSTE RUN TEST

Dia 24

Druk op de groene knop 'Run'. Indien er een fout in de code zit, zal er rechts boven de Astro Pi een rode balk komen met een errorcode.





Errorcodes

Links zie je een voorbeeld. De error zegt dat er iets mis is met de syntax op lijn 36. Dat klopt, want er is één komma te kort op de laatste lijn. Rechts is er een fout bij de naamgeving. Klopt, want op lijn 39 staat 'aardbei' en niet 'Aardbei'. Hier kan je merken dat hoofdletters binnen Python een groot belang hebben.





STAP 4: Kleuren waarnemen met sensor

Een belangrijke stap in de missie is een kleur op de tekening laten veranderen door de kleurensensor. Dit onderdeel bestaat uit twee verschillende codelijnen:

Kleurensensor aanspreken

Dia 25-26

De volwaardige naam van de kleurensensor is 'sense.color'. Wij gaan hem een kortere naam geven. Dat doen we omdat we later in de code deze naam tot driemaal toe moeten schrijven. Dit maakt de code onoverzichtelijk. De nieuwe naam voor onze sense.color is rgb.

De code komt onder '# kleur waarnemen met sensor' en ziet er als volgt uit:



Welke kleur moet er verandert worden?

Dia 26

Nu kiezen de leerlingen een kleur in hun tekening die door de sensor mag veranderd worden. Welke kleur maakt niet uit: de achtergrondkleur, of een accentkleur in de tekening zelf. Denk bijvoorbeeld aan de ogen van een dier, een veranderende snavel, blaadje op een bloem, ...

In onze code kiezen we ervoor om de achtergrond van de tekening te veranderen. Dat is de witte kleur. De code ziet er als volgt uit: 'w = (rgb.red, rgb.green, rgb.blue)'

- 'w' is de naam die we gegeven hebben aan wit.
- '=' hierdoor geven we aan dat we dit gaan 'overschrijven' met een nieuwe code.
- '()' geeft aan de nieuwe code aan.
- 'rgb.rood, rgb.green, rgb.blue': 'rgb' is de kleurensensor, '.rood' is de kleurwaarde. De komma geeft de drie verschillende waarden aan.

De code zou er in de simulatie als volgt moeten uitzien:





TWEEDE RUN TEST

Dia 26

Boven de Astro Pi simulator wordt er in het rood aangegeven of er ergens een fout in de code zit.

STAP 5: Loop

Wat doet het?

Dia 27-28

Wat we tot hiertoe hebben geschreven, duurt ongeveer één seconde op de display. Dit is natuurlijk veel te kort. Het geeft de sensor niet echt de tijd om een kleur waar te nemen en dit weer te geven op het display. We moeten onze Astro Pi computer dus duidelijk maken dat hij dit voor een langere periode op het scherm moet laten zien. Onze code mag maximum gedurende 30 seconden lopen.

We zouden de code dus 30 keer na elkaar in onze simulator kunnen schrijven. Maar dat maakt de code heel lang en onoverzichtelijk. We kunnen hier ook een commando voor gebruiken! Dat is de 'Loop'. Letterlijk vertaalt 'de lus'. Dit zorgt ervoor dat de code zoveel als jij wil herhaald wordt.

Wat is de code?

Dia 28

De code ziet er als volgt uit: 'for i in range (...):'

- 'For in': herhaalt een stukje code voor een gegeven aantal keer.
- 'i': is een variabele die wij hier niet gebruiken. In een ander code kan je hier een waarde aan geven.
- 'Range (...)': geeft aan hoeveel keer de 'for'-loop door de code moet gaan.
- ':' geeft aan welke blok code er moet herhaald worden.

Deze code ziet er in de simulator als volgt uit:





Hoe zet ik alles onder deze loop?

Dia 29

Na de dubbele punt van de loop komt de code die de loop moet herhalen. Dat doe je door de code blok met behulp van tabs onder de code te plaatsen. Voor die blok code moet er dan een lijn tevoorschijn komen.

De code die achter zo een lijn moeten staan zijn:

- # Kleur waarnemen met sensor
- # Tekening maken
- # Display the image

In de simulator ziet het er als volgt uit:



DERDE RUN TEST

Dia 30

Boven de Astro Pi simulator wordt er in het rood aangegeven of er ergens een fout in de code zit.

STAP 6: Stopscherm

Het laatste deel van de missie is dat het scherm 'leeg' moet zijn wanneer de Astro Pi de code heeft afgespeeld. Op die manier is het voor de astronauten duidelijk dat er nieuwe code afgespeeld zal worden.

Voor het lege scherm kunnen de leerlingen een kleur naar keuze nemen.



Deze codeblok bestaat opnieuw uit twee delen en komt terecht onder volgende lijn '# Stopscherm tonen'.

Kleur kiezen

Dia 31

Net zoals we de vorige kleuren gecodeerd hebben, doen we nu hetzelfde. De leerlingen kiezen een nieuwe kleur, geven die een naam en drie RGB-waardes. Bijvoorbeeld: z = (0,0,0) # zwart

Display leegmaken

Dia 31

Om de computer duidelijk te maken dat hij het display moet 'legen', moet je de naam ervan gebruiken: 'sense' en moet je zeggen dat hij het display leeg moet maken: 'clear'. 'Clear' in het Engels betekent 'wissen'.

De code ziet er als volgt uit: 'Sense.clear(z)'

- Sense: SenseHat
- .clear(): wissen naar een bepaalde waarde
- (z): de waarde van het kleur zwart die we erboven hebben gecodeerd.

De code zou er in de simulator als volgt moeten uitzien



VIERDE RUN TEST

Dia 32

Boven de Astro Pi simulator wordt er in het rood aangegeven of er ergens een fout in de code zit.





Afronding

Code indienen

Dia 33

Indien de leerlingen na hun laatste run geen foutmeldingen Krijgen je leerlingen na hun laatste run geen foutmeldingen meer? Dan kunnen ze onderaan hun Astro Pi controleren of alle vakjes zijn aangevinkt.

Vervolgens kunnen ze de klascode invullen die de begeleider doorgestuurd heeft gekregen na het aanmaken van een account. Vervolgens vullen ze nog enkele gegevens in zoals teamnaam, leeftijd en geslacht. De leraar krijgt op zijn account een overzicht van alle ingediende werkjes en krijgt later op het schooljaar de melding dat de codes goed aangekomen zijn op het ISS.

Via deze weg kan de leraar de werkjes van de leerlingen klassikaal tonen.



Let op: Indien de code langer dan 30 seconden loopt, dan kan je in for-loop de waarde 28 verminderen. Dan ga je dus de code een keer minder herhalen waardoor het minder lang duurt om het programmaatje te doorlopen.



Toonmoment

De leraar kan er hiervoor kiezen om via zijn/haar account de codes aan de leerlingen te tonen. Dit gaat als volgt:

Stap 1: log in op account

Cesa S		Log in	~	Inloggen
Astr Click Your can r	Vou have logged out. o Pi k Log in to continue your journey with Astro Pi. r progress will be saved after submitting each step, and you resume at any time by logging back in. Log in		E-mailadn Wachtwoo 	er of gebruikersnaam ord Inloggen Inloggen een account? <u>Maak er een</u> .

Stap 2: Kies Mission Zero



Stap 3: Kies een ingediend werkje





Stap 4: Download de code



Stap 5: Open de download in kladblok







Stap 6: Kopieer de code



Stap 7: Kleef in de simulator, overschrijf hier alle bestaande code.

Voila.

OPGELET

De codes worden niet automatisch opgeslagen in de simulator. Indien de codes niet binnen de lestijd afgerond geraken, moeten de leerlingen de code eerst opslaan in een kladblok document. Dit slaat tekst op zonder opmaak.

In een volgende les kunnen ze hun code terug uit het kladblok kopiëren in de simulator.

Kopieer geen tekst in een Word - document. Dit kan foutmeldingen geven doordat Word automatisch hoofdletters toevoegt of tekens aanpast.

Als de leerlingen per ongeluk hun tabblad sluiten op de webbrowser, zijn ze ook hun code kwijt op de simulator. Vandaar dat we aanraden de code ook steeds op het spiekbriefje te schrijven.

