



UHASSELT

KNOWLEDGE IN ACTION

School voor Educatieve Studies

Educatieve master in de wetenschappen en technologie

Masterthesis

Leren via artificiële intelligentie

Benny Kerkhofs

Scriptie ingediend tot het behalen van de graad van Educatieve master in de wetenschappen en technologie, afstudeerrichting engineering en technologie

PROMOTOR :

Prof. dr. Katrien STRUYVEN

COPROMOTOR :

Prof. dr. Elke EMMERS
dr. Mario GIELEN



UHASSELT

KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

2023
2024



School voor Educatieve Studies

Educatieve master in de wetenschappen en technologie

Masterthesis

Leren via artificiële intelligentie

Benny Kerkhofs

Scriptie ingediend tot het behalen van de graad van Educatieve master in de wetenschappen en technologie, afstudeerrichting engineering en technologie

PROMOTOR :

Prof. dr. Katrien STRUYVEN

COPROMOTOR :

Prof. dr. Elke EMMERS
dr. Mario GIELEN

Abstract	3
Literatuurstudie en probleemstelling	4
Zoekplan	4
Gebruikte databanken.....	4
Zoektermen en zoekvragen	4
Criteria voor selectie bronnen (inclusiecriteria/exclusiecriteria):	4
Literatuurstudie	5
Probleemstelling	8
Onderzoeksvraag	8
Onderzoeksoptzet en -methoden	9
Eerste fase: design thinking	9
Tweede fase: verdiepende enquête.....	10
Onderzoeksresultaten	11
Eerste fase: Design Thinking	11
Iteratie 1: online invulformulier.....	11
Iteratie 2: beperkter online invulformulier	16
Iteratie 3: complexere softwarecode uitfilteren	20
Iteratie 4: gebruikte softwarecode als lesmateriaal inzetten.....	21
Iteratie 5: basisvoorbeeld + plan van aanpak + 'prompting'-instructies.....	23
Iteratie 6: geen "opdrachten", maar mogelijkheden	28
Tweede fase: verdiepende enquête	30
Conclusie en discussie	31
Conclusie	31
Discussie	32
Opstart van een nieuw leeronderwerp/-hoofdstuk.....	32
Na de opstart: project- en leercyclus	32
Evalueren	33
Toekomst	34
Referentielijst	35
Bijlagen	37
Creatief eindproduct: promptgenerator.....	37
Handleiding	37
Link naar het bestand	37
Printscreen	38
Resultaat.....	39

Abstract

Door de introductie van generatieve artificiële intelligentie (AI) ontstaan er verschillende mogelijkheden en uitdagingen in het onderwijs.

Onder andere het lesgeven in softwarevakken kan extra uitdagend zijn. Als er bijvoorbeeld een softwareprogramma geschreven moet worden, bestaat de mogelijkheid dat AI dit kan doen in plaats van de student. De student hoeft dit dan enkel nog te kopiëren, waardoor er geen echt leerproces meer plaatsvindt.

AI kan ervoor zorgen dat studenten sneller in contact komen met moeilijkere softwarecode, wat de omvang van de leerstof kan vergroten. Tegelijkertijd kan dit het leerpotentieel en de intrinsieke motivatie van studenten verhogen, als deze leerstof op een adaptieve en ervaringsgerichte manier kan worden aangebracht.

De nood dient zich aan om manieren te zoeken om op een zinvolle manier gebruik te maken van generatieve AI en dit in de juiste context.

De hoofdvraag die in dit onderzoek gesteld wordt is : "Wat is een goede aanpak om studenten te leren programmeren, in tijden waar generatieve AI beschikbaar is?"

Het onderzoek focust zich op een iteratief experiment dat gedaan werd volgens het principe van 'Design Thinking'. Daarnaast werd er een verdiepende enquête gehouden. Hierbij werden verschillende manieren uitgetest die ondersteunend kunnen zijn: zowel voor studenten in hun groeiproces en in het opbouwen van expertise, als voor de docent in het evalueren van de studenten.

De studenten waardeerden het om een basisvoorbeeld te krijgen bij een nieuw lesonderwerp. Generatieve AI helpt de studenten in hun leerproces om van start te geraken, op hun eigen tempo te leren, en om specifieke vragen te kunnen stellen om code beter te begrijpen.

Het laten indienen van code door de studenten werkte goed en geeft de mogelijkheid om leerstof op een adaptieve manier aan te brengen (aan de hele klas).

Als voorstel wordt aangegeven om eerst een introductie te doen, waarbij de 'fundamenten' gelegd worden, door een absoluut basisvoorbeeld aan te leren. Daarnaast zijn er hulpbronnen nodig, zoals bijvoorbeeld richtlijnen om uit probleemsituaties te geraken. Na de fundamenten, stellen we een continue project- en leercyclus voor, die adaptief van aard is. De ene student verkiest meer verdieping, de andere meer ontdekking. De docent volgt elk project op, en verzamelt gemeenschappelijkheden tussen studenten zodat ze ook van elkaar kunnen leren, of samen projecten kunnen uitwerken. Daarnaast worden verschillende ideeën gegeven om volwaardig te evalueren.

Literatuurstudie en probleemstelling

Zoekplan

Gebruikte databanken

[SciSpace](#), [ResearchGate](#), [Semantic Scholar](#), [UHasselt Discovery Service](#)

Zoektermen en zoekvragen

Gebruikte zoektermen:

- AI education
- AI education expertise
- AI programming learning
- AI programming assistants
- AI assessment education
- personalized learning
- adaptive learning

Criteria voor selectie bronnen (inclusiecriteria/exclusiecriteria):

- Inclusief:
 - o Onderzoek speelt zich af binnen een onderwijzende/lerende context
 - o Expertise opbouwen
 - o Evalueren
- Exclusief:
 - o Automatische evaluatie ("automatic grading")
 - o AI-systemen binnen gezondheidszorg en industrie.
 - o Energieverbruik van AI-systemen
 - o Privacy en security van AI-systemen

Literatuurstudie

Op relatief korte tijd is er een grote maatschappelijke evolutie op vlak van generatieve AI, via Large Language Models (LLM's). Er zijn enkele onderzoeken die een grote invloed hebben gehad in de ontwikkeling van deze technologie, en die ook helpen in het begrijpen van hoe een LLM werkt.

Eenzijds is dit het werk van Vaswani et al. (2017), getiteld "Attention is all you need". Hierbij hebben ze de Transformer-architectuur ontwikkeld. Deze softwarearchitectuur is gericht op vertalingen, samenvattingen maken en natuurlijke taalgeneratie. Het belangrijkste principe in het model is dat van "self-attention". Hierdoor kunnen relaties tussen woorden in een zin begrepen worden, waardoor niet de volledige tekst hoeft geïnterpreteerd te worden om te weten waar het over gaat. Dit model zorgde voor een hele efficiëntieverbetering, waardoor er ook minder computerkracht nodig was dan bij andere modellen. Dit onderzoek heeft de basis gelegd voor de meeste Large Language Models, waaronder de GPT-reeksen van OpenAI.

Via het werk van Devlin et al. (2018), en van Howard et al. (2018) is er een verdere verfijning en efficiëntieverhoging mogelijk gemaakt in het trainen van deze modellen op grote hoeveelheden tekstdata. Hierdoor werden de modellen ook betrouwbaarder.

Het werk van Radford et al. (2018) heeft gezorgd voor een beter basisbegrip van taal, via de techniek van 'unsupervised pre-training'.

Daarnaast is het werk van Brown et al. (2020), getiteld "Language Models are Few-Shot Learners" ook heel belangrijk geweest. Door een LLM te trainen op een heel grote hoeveelheid tekstdata, gaat het beter kunnen antwoorden op vragen waarop het niet specifiek getraind is, maar waarbij er wel enkele voorbeelden ('few shots') gegeven worden. In hun onderzoek werd GPT-3 getraind, dat de basis werd voor ChatGPT en veel andere AI-tools op dit moment.

Er zijn de laatste jaren verschillende publicaties gebeurd i.v.m. het gebruiken van AI in het onderwijs. Er zijn hierbij verschillende invloeden die AI kan hebben en die besproken worden.

Een interessant onderzoek is gebeurd bij jonge leerlingen door Se-Jung et al. (2023). Hier is bekeken hoe artificiële intelligentie zwakkere leerlingen kan helpen bij het leren van Engels in de lagere school. Aangezien het leeftijdsverschil is er mogelijks een ander verloop dan bij hogeschoolstudenten, maar toch geeft het een interessant beeld.

In het begin vertoonden de leerlingen een grote afhankelijkheid t.o.v. AI. Na een tijd, ontwikkelden ze eigen routines en raakten ze meer gewoon aan het integreren van AI in hun leerproces. Naarmate het onderzoek vorderde, werden de leerlingen minder afhankelijk van AI. De angst bij de laagpresterende leerlingen verminderde, en het zelfvertrouwen ging omhoog, doordat AI extra opties gaf om leermoeilijkheden te overwinnen.

Doordat het zelfvertrouwen omhoog gaat, gaat het leerresultaat omhoog (Oktafiani & Yusri, 2021 en Acosta-Gonzaga, 2023).

AI kan helpen om flexibel in te spelen op leerbehoeftes van studenten. Algemeen is dit gekend onder de naam "flexibel leren", "adaptief leren", "adaptive teaching", "adaptive learning". Hieronder enkele publicaties met betrekking tot dit onderwerp.

- In 2009 is er een studie gebeurd door Graf et al. waarin het belang geuit is van adaptieve leersystemen.
- Adaptive Learning Platform (Yanlin & Dylan, 2015): hierin wordt aangegeven dat adaptief leren een meer interactieve, persoonlijke en boeiende leerervaring kan bieden.
- Adaptive Learning (Karun, 2017): hierin wordt een leersysteem ontworpen dat gebruik maakt van 'analytics' om te bepalen welke instructiemethodes en welke pedagogische ondersteuning voor elke student "het best" zou zijn.
- Adaptive Learning Technologies (Nicola & Santi, 2020): adaptieve leertechnologieën passen de cursusinhoud dynamisch aan op basis van individuele vaardigheden. Hierdoor worden de leerprestaties verhoogd.
- Een vorm van adaptief leren kan ook zijn door het voorzien van een systeem, zoals een "personal" e-learningmodule, zie bijvoorbeeld in "Personalized learning through AI" (Jian, 2023).
- "Role of AI in Personalized Education" (Mote, 2024): citaat: *"Personalized Education is based on individual interest, learning styles and self-pace. It provides more focused and self-directed learning."*

Daarnaast wordt via i-Learn Vlaanderen adaptief (digitaal) leren gepromoot, ondersteund door de Vlaamse overheid.

Een ander belangrijk punt om mee te nemen, is dat men ervan uit zou gaan dat er weinig expertise ontwikkeld wordt door de studenten, als AI het meeste werk doet. Dit is bijvoorbeeld aangegeven in een onderzoek door Alexandre (2022) binnen de professionele sector, waar de introductie van AI ervoor zorgt dat boekhouders wel productiever worden, maar er tegelijkertijd expertise verloren kan gaan.

'Snap!' is een praktisch project, wat komt uit het onderzoek genaamd "Learning by Enhancing Half-Baked AI Projects" (Kahn & Winters, 2021). De insteek die hierin interessant is, is de manier waarop AI gebruikt wordt om toch expertise te ontwikkelen. Doordat de projecten "half-baked" (halfbakken/gebrekkig) zijn, moeten deze verbeterd worden. Het verbeteren ervan, kan een startpunt zijn om expertise op te bouwen (mits goede begeleiding door een leerkracht).

Direct gelinkt aan het opbouwen van expertise, ligt ook het evalueren van (al dan niet) opgebouwde expertise. Een interessante aanzet hierin is hier die van Furze et al. (2024) via "The Artificial Intelligence Assessment Scale" (AIAS). Dit geeft een kader dat leerkrachten kunnen gebruiken voor het definiëren van de mate waarin AI gebruikt mag worden in een opdracht. Zie Figuur 1 verderop.

1	NO AI	The assessment is completed entirely without AI assistance. This level ensures that students rely solely on their knowledge, understanding, and skills. AI must not be used at any point during the assessment.
2	AI-ASSISTED IDEA GENERATION AND STRUCTURING	AI can be used in the assessment for brainstorming, creating structures, and generating ideas for improving work. No AI content is allowed in the final submission.
3	AI-ASSISTED EDITING	AI can be used to make improvements to the clarity or quality of student created work to improve the final output, but no new content can be created using AI. AI can be used, but your original work with no AI content must be provided in an appendix.
4	AI TASK COMPLETION, HUMAN EVALUATION	AI is used to complete certain elements of the task, with students providing discussion or commentary on the AI-generated content. This level requires critical engagement with AI generated content and evaluating its output. You will use AI to complete specified tasks in your assessment. Any AI created content must be cited.
5	FULL AI	AI should be used as a “co-pilot” in order to meet the requirements of the assessment, allowing for a collaborative approach with AI and enhancing creativity. You may use AI throughout your assessment to support your own work and do not have to specify which content is AI generated.

Figuur 1: “The AI Assessment Scale” (Furze et al., 2024)

Er is ook onderzoek gebeurd in het gebruik van ChatGPT om te leren programmeren (Ramazan et al., 2023). Hierin zijn vooral de meningen van de studenten bevraagd. De studenten vonden dat ChatGPT hen hielp in het begrijpen van programmeerconcepten, en om efficiënt programmeerproblemen op te lossen. Er wordt in die paper gesuggereerd dat de integratie van ChatGPT de programmeervaardigheden kan verbeteren van de studenten. Het onderzoek duidt erop dat verder onderzoek nodig is in het gebruik van ChatGPT in het leren programmeren.

Probleemstelling

AI-tools bieden uitdagingen, maar ook mogelijkheden op vlak van het ontwikkelen van expertise.

AI-tools zorgen voor mogelijkheden op vlak van adaptief leren. Adaptief leren wordt meestal geassocieerd met het gebruik van analysemethodes om studentenprofielen te maken, en op basis hiervan de leerinhoud individueel bij te sturen. Hierbij wordt over het algemeen niet gesproken over generatieve AI.

Er is verder (praktisch) onderzoek nodig, zeker op het gebruik van AI bij het programmeren van microcontrollers.

Daarnaast is het goed om nieuwe manieren van evalueren uit te proberen die aangepast zijn aan het werken met AI.

Onderzoeksvraag

Hoofdvraag

Wat is een goede aanpak om studenten te leren programmeren, in tijden waar generatieve AI beschikbaar is?

Deelvragen

- Welke expertise is er (nog) nodig bij de studenten?
- Hoe kunnen studenten deze expertise opbouwen terwijl ze met AI werken?
- Hoe kan er een correcte evaluatie van deze opgebouwde expertise gebeuren door de docent, als studenten met AI werken? (rekening houdend met kopiëren van code enz.)
- Hoe kan generatieve AI bijdragen in het algemene groeiproces? (cfr. adaptief leren, volgens niveau en behoeftes)

Onderzoeksopzet en -methoden

Het onderzoek is gebeurd in een klas van 11 studenten in het hoger onderwijs (9 mannen en 2 vrouwen, van 18 à 20 jaar), in de opleiding 'Graduaat in het Internet of Things'. Deze studenten hebben over het algemeen weinig programmeerervaring.

Vóór de start van het experiment, is de volgende vraag gesteld aan elke student:

"Wil je met AI-programmeerassistenten werken, en daarbij ook moeilijkere opdrachten krijgen?"

OF:

"Wil je zonder AI-programmeerassistenten werken."

Het antwoord van elke student was "met" AI-programmeerassistenten.

Er zijn daarna verschillende programmeerassistenten voorgesteld aan de studenten, maar ze mochten er ook zelf kiezen. De voorgestelde AI-assistenten waren: Codey.online, Github Copilot, ChatGPT en Microsoft Copilot.

Er zijn twee fases in dit onderzoek, waarbij de eerste fase is opgebouwd uit achtereenvolgende subfases (in de vorm van iteraties).

Eerste fase: design thinking

Het onderzoek speelt zich af binnen een op dit moment zeer snel evoluerend onderwerp zoals generatieve AI. Daarnaast is het onderzoek experimenteel van opzet, voor het vinden van manieren die werken voor zowel de studenten als de docent in een onderwijscontext.

De eerste fase maakt daarom gebruik van "design thinking" in combinatie met het houden van observaties en bevestigingen. Design thinking loopt standaard doorheen verschillende **iteraties**, als een zoektocht om een complex probleem op te lossen. Doorheen de verschillende iteraties wordt de oplossing telkens bijgestuurd of uitgebreid, op een 'agile' manier. Dit helpt om flexibel te kunnen inspelen op wat zich aandient, en van hieruit tot een constructief eindresultaat te komen dat vanuit het ervaringsgerichte ontstaat.

In elke iteratie van het design thinking onderzoek zijn er vaste stappen die doorlopen worden. In dit onderzoek zijn de volgende stappen gebruikt:

- 'Empathize': de behoeftes begrijpen
- 'Define': op basis van de behoeftes het kernprobleem formuleren
- 'Ideate': ideeën genereren voor het probleem
- 'Prototype': prototype/oplossing maken voor het probleem
- 'Test': de oplossing testen in de praktijk, inclusief observaties en bevestigingen

Tweede fase: verdiepende enquête

In de tweede fase wordt er een verdiepende enquête gedaan onder de studenten. Hierbij wordt er meer toegewerkt naar een finale, werkende aanpak voor het programmeren en leren via AI. De enquête wordt via een invulformulier aangeboden, met een afwisseling van open vragen, keuzeopties, en meerkeuzevragen. Bij verschillende keuzevragen wordt er aansluitend een extra toelichting gevraagd onder de vorm van een open vraag, voor een juister begrip van wat er leeft. De enquête gebeurt in een fysieke ruimte, en wordt rechtstreeks geïnterpreteerd zodat bijvragen gesteld kunnen worden, mondeling of via een extra invulformulier.

Onderzoeksresultaten

Eerste fase: Design Thinking

Iteratie 1: online invulformulier

Empathize

Het semester voorafgaand aan dit onderzoek werd bij deze klas vastgesteld dat sommige studenten in een ander vak een AI-programmeerassistent gebruikten om opdrachten op te lossen, in plaats van deze opdrachten zelf te maken. Vanuit de hogeschool wordt generatieve AI toegestaan (onder bepaalde voorwaarden). De docent zelf wil de studenten de kans geven om met AI te leren werken, en te ontdekken in welke situatie dit zinvol is en wanneer niet. Tegelijkertijd was er een noodzaak om dit gestructureerd aan te pakken, zodat de studenten goede vaardigheden en expertise opbouwen. Na navraag bij elke student voor het onderzoek, heeft elke student expliciet gekozen om met AI-programmeerassistenten te werken om microcontrollers te programmeren.

Define

Een gestructureerde aanpak vinden voor het gebruik van generatieve AI, die zowel werkt voor de student als voor de docent en de hogeschool.

Daarnaast, volgens het onderzoek van Liang et al. (2023) en van Tan et al. (2024) is gebleken dat AI-programmeerassistenten dikwijls niet de verwachte antwoorden geven.

In deze iteratie wordt een antwoord gezocht op de onderzoeksvraag welke expertise er nodig is, en hoe deze kan opgebouwd worden.

Ideate

Het idee is om een online invulformulier te voorzien voor de studenten. In dit invulformulier kan elke student verschillende informatie invullen. Hierdoor kunnen studenten opvolgen wat werkt en hierbij stilstaan. Ook kunnen ze zo leren van andere studenten.

Ideeën voor een "promptsdatabank" voor AI-assistenten:

- Ofwel in Excel (online)
- Ofwel in Microsoft Lists (online gedeeld met de hele klas)
- Mogelijke kolommen:
 - o vraagstelling invullen
 - o antwoord
 - o fouten in ontvangen antwoord
 - o nog niet gekende code
 - o eigen verklaring van niet gekende code
- controle/extra uitleg door docent kunnen geven door docent
- links naar externe websites en informatie, die ondersteunend kan zijn voor het leerproces en het begrip van de studenten.

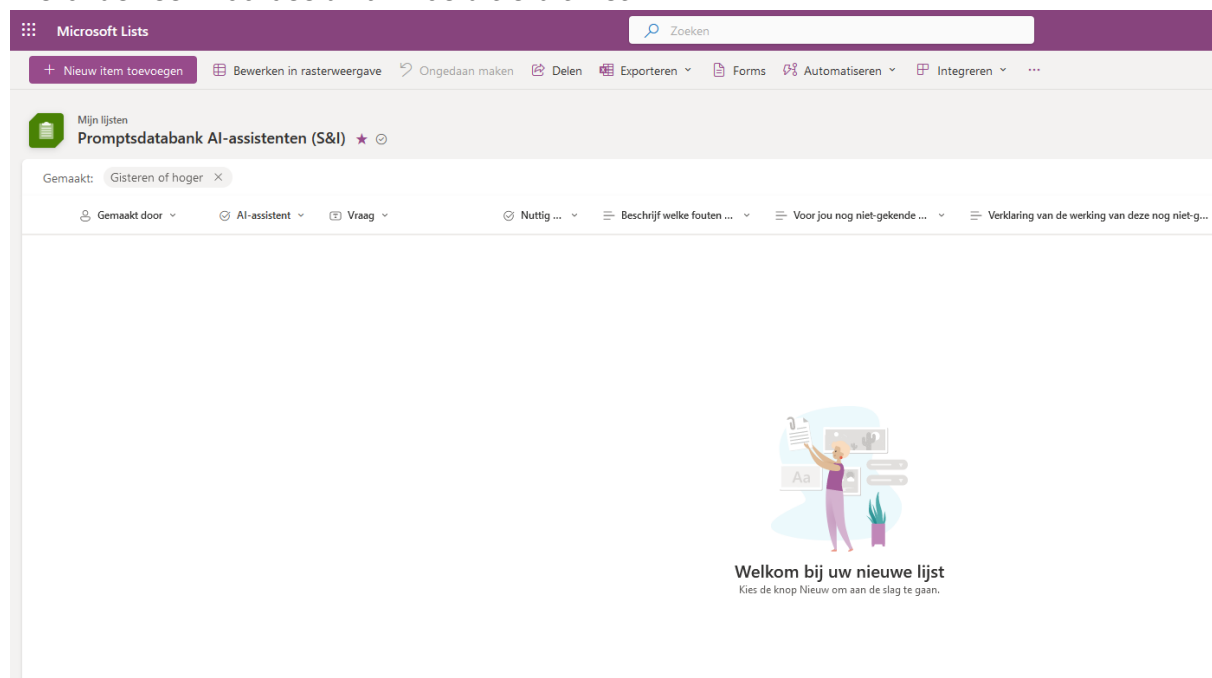
De expertise die ze hierbij kunnen opbouwen, is enerzijds het leren kennen van verschillende AI-programmeerassistenten, anderzijds het leren stellen van de juiste vragen 'prompting'. Daarnaast leren ze uit foute antwoorden van AI en kunnen ze leren uit moeilijke softwarecode.

Prototype

"Promptsdatabank AI-assistenten":

- in Microsoft Lists (online gedeeld met de hele klas)
 - o kolommen:
 - Gemaakt door (studentnaam) (*automatisch*)
 - Gebruikte AI-assistent
 - Gestelde vraag ("prompt")
 - Nuttig antwoord ontvangen? (Ja/Nee/Redelijk)
 - Beschrijving van de fouten in het ontvangen antwoord
 - Voor de student nog niet gekende code
 - Eigen verklaring van de student, van de nog niet gekende code
 - Kopie van het ontvangen antwoord
 - Nuttige link
 - Extra uitleg (bv. achteraf door de docent in te vullen, voor meer diepgaand begrip)
 - Datum waarop de lijn is ingevoerd (*automatisch*)

Hieronder een voorbeeld van hoe dit eruit ziet:



Figuur 2: prototype "promptsdatabank"

Test

Dit online invulformulier werd met de studenten gedeeld tijdens de les. Ze kregen een opdracht om op te lossen. In de eerste les was de opdracht het programmeren van een microcontroller, die temperatuurwaarden uit een specifieke sensor moest uitlezen. Dit was een praktische opdracht, waarbij ook een schakeling gebouwd moest worden. Ter herinnering: de studenten hebben over het algemeen weinig tot geen voorkennis van het programmeren van microcontrollers.

Telkens als ze een vraag stelden aan een AI-programmeerassistent, werd verwacht dat ze het online formulier aanvulden:

- Gemaakt door (studentnaam) (*automatisch*)
- Gebruikte AI-assistent
- Gestelde vraag ("prompt")
- Nuttig antwoord ontvangen? (Ja/Nee/Redelijk)
- Beschrijving van de fouten in het ontvangen antwoord
- Voor de student nog niet gekende code
- Eigen verklaring van de student, van de nog niet gekende code
- Kopie van het ontvangen antwoord
- Nuttige link
- Extra uitleg (bv. achteraf door de docent in te vullen, voor meer diepgaand begrip)
- Datum waarop de lijn is ingevoerd (*automatisch*)

Naarmate het formulier meer werd ingevuld door de studenten, werd het duidelijk dat de kolom "Nuttig antwoord ontvangen?" het het minst op "Ja" stond, maar meer op "Redelijk" en "Nee". Een extract van het ingevuld formulier is verderop te zien, in Figuur 3. Dit ligt in lijn met wat Liang et al. (2023) en van Tan et al. (2024) aangaven, namelijk dat AI-programmeerassistenten dikwijls niet de verwachte antwoorden geven.

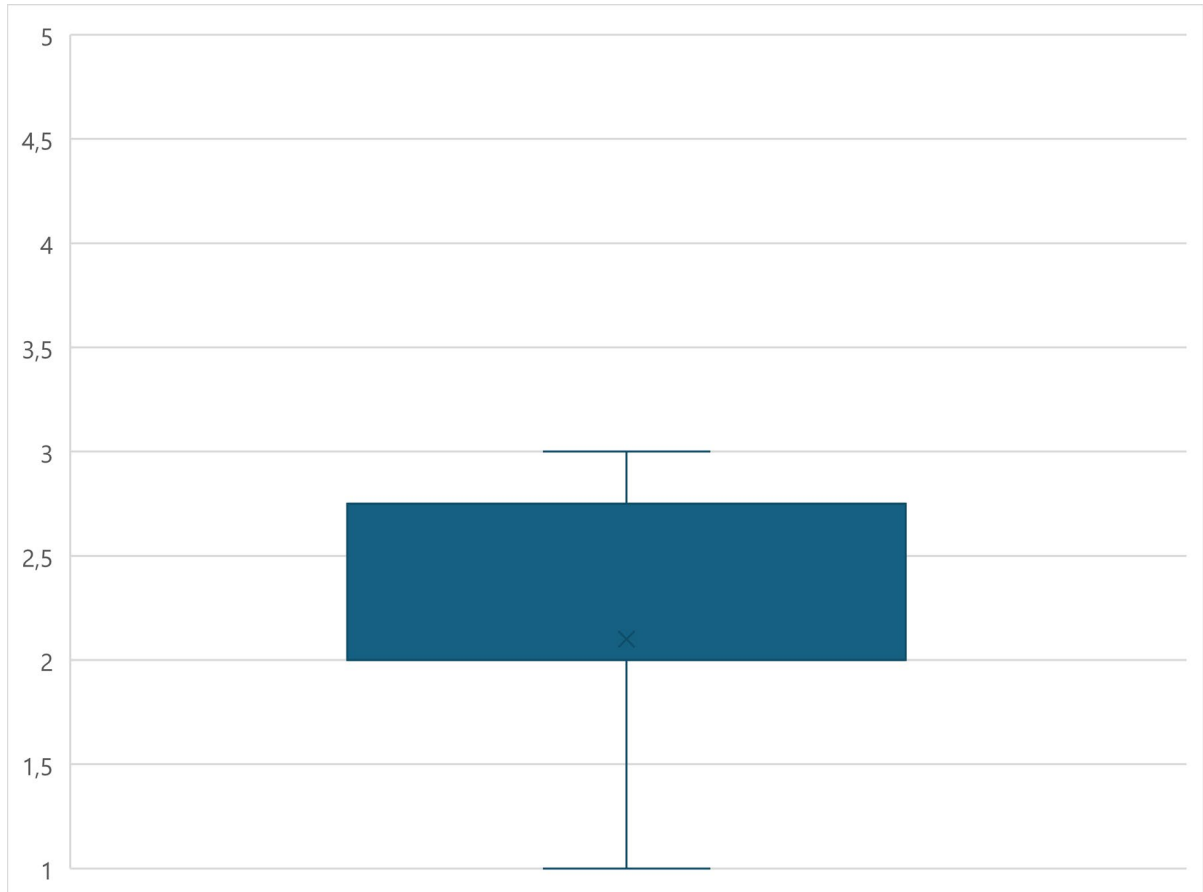
Hier in dit onderzoek, en deze iteratie gaat het over het programmeren van microcontrollers, wat niet exact vergelijkbaar is met dat onderzoek, maar de observaties liggen wel in dezelfde lijn. Hierbij moet vermeld worden, dat dit afhankelijk is van de ontwikkeling van AI. Bijvoorbeeld hebben de studenten bij deze iteratie ook gebruik gemaakt van Codey.online, die op dat moment gebaseerd was op een oudere GPT-versie, die minder goede resultaten gaf.

De studenten toonden soms frustraties omdat ze niet direct de verwachte resultaten kregen. De tweede les werd dit invulformulier bijna niet meer ingevuld.

AI-assistent	Vraag	Nuttig antwoord ?	Beschrijf welke fouten de gegeven code bevat ...	Voor jou nog niet-gekende...	Verklaring van de werking van deze nog niet-gekende code (in eigen woorden)
Codey/online	Maak een code met 1 Wire-interface voor 1 bus met GND pin, 3.3V pin en data pin op een ESP32. Maak het eenvoudig.	Redelijk	Geen pinnen beschreven voor GND en 3.3V, en verkeerde pin als DATA pin gebruikt.	Gebruik van bytes in het code.	De bytes in de code worden gebruikt om gegevens van de 1-Wire-bus te lezen en te verwerken. In het specifieke geval van het lezen van de temperatuur van een DS18B20-sensor via de 1-Wire-interface, worden de bytes gebruikt om de ruwe gegevens van de sensor te ontvangen en om te zetten naar een bruikbare temperatuurwaarde. De DS18B20-sensor stuurt 9 bytes met gegevens terug wanneer er om de temperatuur wordt gevraagd. Deze gegevens bevatten informatie over de temperatuur in een bepaald formaat dat moet worden verwerkt om de daadwerkelijke temperatuurwaarde te verkrijgen. Door de bytes op de juiste manier te combineren en te verwerken, kunnen we de temperatuur berekenen en deze vervolgens afdrucken via de seriële monitor. Kortom, de bytes in de code worden gebruikt voor het ontvangen, opslaan en verwerken van gegevens van de DS18B20-sensor via de 1-Wire-bus.
ChatGPT	Maak een code met 1 Wire-interface voor 1 bus met GND pin, 3.3V pin en data pin op een ESP32. Maak het eenvoudig.	Redelijk	Geen pinnen voor GND en 3.3V weergegeven, maar geeft wel het waarschuwing om de DATA pin te wijzigen indien we een andere willen gebruiken	Het gebruik van bytes in de code	byte i; Dit is een tellervariabele die wordt gebruikt in loops om door arrays te itereren. byte present = 0; Dit wordt gebruikt om de aanwezigheid van een 1-Wire-apparaat aan te geven. Het wordt ingesteld op 0 (false) aan het begin en kan later worden gewijzigd in 1 (true) als er een apparaat wordt gedetecteerd. byte data[12]; Dit is een array van bytes die wordt gebruikt om gegevens van een 1-Wire-apparaat op te slaan. In deze code wordt het niet expliciet gebruikt, maar je zou het kunnen gebruiken om gegevens van een 1-Wire-sensor te lezen en op te slaan. byte addr[8]; Dit is een array van bytes die wordt gebruikt om het adres van een 1-Wire-apparaat op te slaan. Wanneer een apparaat wordt gedetecteerd, wordt het adres opgeslagen in deze array, zodat het later kan worden gebruikt voor communicatie met dat specifieke apparaat.
Codey/online	schrijf een programma voor de ds1820 temperatuursensor.	Redelijk	if (tempC != DEVICE_DISCONNECTED_C) { } else { Serial.println("Fout bij het lezen van de temperatuur."); } de if en else geven een fout bij de uitvoering.		

Figuur 3: extract ingevuld prototype "promptsdatabank"

De reacties van de studenten op de vraag "Hoe zinvol vind jij het formulier Promptsdatabank AI-assistenten" zijn verdeeld over verschillende studenten (zie Figuur 4 hieronder, waarbij 1=niet zinvol, 5=heel zinvol). Het gemiddelde resultaat is gelijk aan 2,1 en de standaardafwijking is gelijk aan 0,7. Dit prototype wordt als eerder niet zinvol ervaren wordt door de studenten.



Figuur 4: resultaten van de studentenbevraging over de zinvolheid van het prototype "Promptsdatabank AI-assistenten". Waardes Y-as: 1=niet zinvol, 5=heel zinvol.

Bijkomende feedback van de studenten:

De meesten gaan zelf op zoek, en stellen bijvoorbeeld vragen aan de AI-tool om hun probleem op te lossen.

Interpretatie:

De studenten willen vooral zelf ervaring opdoen, en hierbij niet teveel extra 'administratie' doen.

Iteratie 2: beperkter online invulformulier

Empathize

Het invulformulier uit de vorige fase werd door de studenten niet meer ingevuld.

Voor de docent is de prioriteit vooral dat de studenten leren uit de fouten die naar boven komen, en dat ze hun gebruikte code begrijpen. Ook als die code van een AI-tool komt. Dit ligt in lijn van het beleid van de hogeschool.

Voor het beleid van de hogeschool zijn dit enkele regels voor de student op vlak van het gebruik van generatieve AI ("GenAI"):

"

- De student is **volledig verantwoordelijk** voor wat hij indient.
- De student **zorgt dat de lector ondubbelzinnig kan evalueren** welke competenties hij verworven heeft.
- De student **kent de mogelijkheden en beperkingen** van GenAI en **maakt er verantwoord gebruik van** binnen de kaders die door de opleiding zijn vastgelegd.

"

Define

De studenten moeten een meerwaarde en motivatie vinden in het leren uit fouten en het opbouwen van expertise. Ze moeten onafhankelijker worden van de beperkingen van AI-tools, aangezien ze zelf eindverantwoordelijk zijn voor hun ingediende opdracht.

Ideate

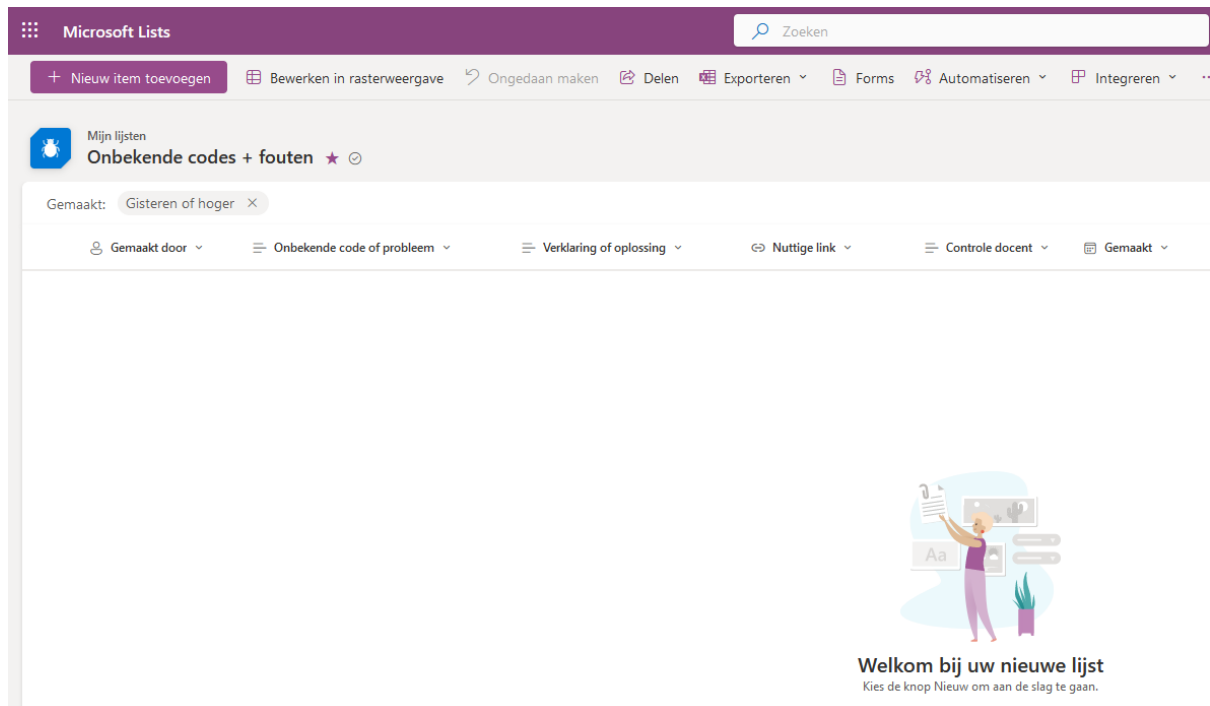
Een nieuw formulier, dat niet meer direct gelinkt is aan een AI-tool, maar vooral gericht is op de student. De focus ligt hier op het leren van onbekende code, en fouten die ontstaan bij het maken van softwareopdrachten (door de studenten). De vraag is hier aan de studenten om deze zelf te verklaren. Daarnaast volgt de docent deze lijst op, en stuurt bij waar nodig.

Prototype

Er wordt een beperkter invulformulier gegeven aan de studenten, getiteld "Onbekende codes + fouten":

- in Microsoft Lists (online gedeeld met de hele klas)
- kolommen:
 - o onbekende code of probleem
 - o verklaring of oplossing (gegeven door student)
 - o nuttige link naar externe website
 - o controle door docent

Voorbeeld van het invulformulier:



The screenshot displays the Microsoft Lists web application. At the top, there's a purple header bar with the 'Microsoft Lists' logo and a search bar labeled 'Zoeken'. Below the header, a navigation bar contains several action buttons: '+ Nieuw item toevoegen', 'Bewerken in rasterweergave', 'Ongedaan maken', 'Delen', 'Exporteren', 'Forms', 'Automatiseren', and 'Integreren'. The main content area shows a list titled 'Onbekende codes + fouten' under the heading 'Mijn lijsten'. Below the title, there's a filter bar with 'Gemaakt: Gisteren of hoger'. A table header is visible with columns: 'Gemaakt door', 'Onbekende code of probleem', 'Verklaring of oplossing', 'Nuttige link', 'Controle docent', and 'Gemaakt'. The table body is currently empty. At the bottom right, there's a welcome message: 'Welkom bij uw nieuwe lijst' followed by 'Kies de knop Nieuw om aan de slag te gaan.' and an illustration of a person holding a document.

Figuur 5: prototype invulformulier "Onbekende codes + fouten"

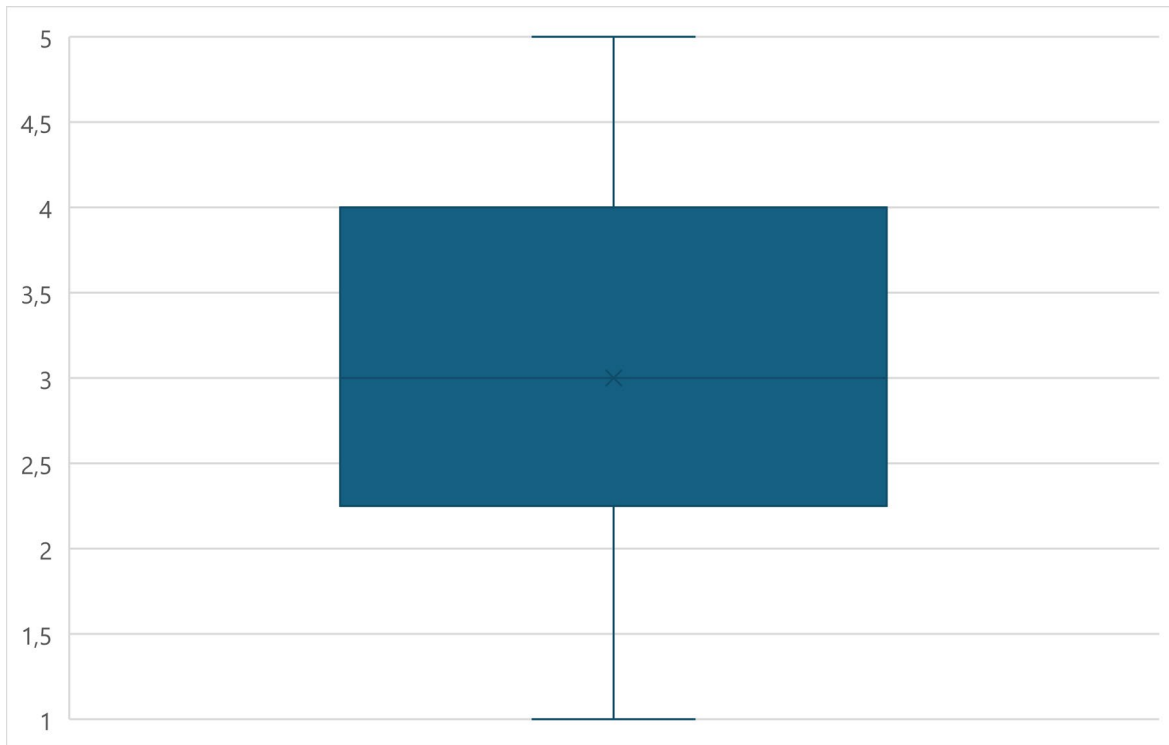
Test

De studenten tonen over het algemeen meer motivatie voor dit invulformulier dan voor het andere uit iteratie 1. In Figuur 6 hieronder is een extract te zien van het ingevulde prototype.

Onbekende code of probleem	Verklaring of oplossing	Nuttige link	Controle
	<pre>volatile bool newData = false; declareert een boolean variabele genaamd `newData` die aangeeft of er nieuwe gegevens beschikbaar zijn. Het sleutelwoord `volatile` geeft aan dat de variabele op elk moment kan veranderen, zelfs als het programma er niet expliciet iets aan verandert. Dit kan handig zijn bij bijvoorbeeld onderbrekingen.</pre>		
Wire.requestFrom	Deze functie wordt gebruikt om gegevens aan te vragen van een specifiek slave-apparaat op het I2C-bus en deze gegevens naar de master (het Arduino-board dat de communicatie initieert) te lezen.		
SerialTeleplot teleplot("127.0.0.1");	Gegevens naar Teleplot sturen via een seriële verbinding met het IP-adres "127.0.0.1"		Check dat je de juiste "Teleplot" hebt die in de opdracht gevraagd was. Je kan ook zoeken op de klasse "SerialTeleplot" bijvoorbeeld op Github om te zien of die wel bestaat.

Figuur 6: prototype invulformulier "Onbekende codes + fouten", met aanvullingen van studenten (+ inclusief opmerking docent in het blauw)

De reacties van de studenten op de vraag "Hoe zinvol vind jij het formulier Onbekende codes + fouten" zijn verdeeld. Zie Figuur 7 hieronder, waarbij 1=niet zinvol, 5=heel zinvol. Het gemiddelde is gelijk aan 3 en de standaardafwijking is gelijk aan 1,3.



Figuur 7: resultaten van de studentenbevraging over de zinvolheid van het invulformulier "Onbekende codes + fouten". Waardes Y-as: 1=niet zinvol, 5=heel zinvol.

De toelichting van de studenten hierbij:

Voor sommigen is het niet duidelijk wat er door de andere studenten wordt ingevuld.

Sommige studenten willen zelf hun eigen problemen oplossen (maar vinden het wel belangrijk om uit fouten te leren).

Sommigen willen de combinatie van zowel zelf verklaringen ingeven, en ook leren van andere studenten.

Anderen willen vooral info eruit halen die andere studenten erin hebben gezet.

Link met de onderzoeksvragen:

Dit invulformulier kan een aanzet zijn voor studenten om expertise op te bouwen op vlak van het programmeren van software, en het werken met AI-assistenten. Het kan een aanzet zijn voor de docent om tot evaluatievormen te komen, omdat studenten in het formulier ingeven welke fouten en speciale code ze tegenkomen.

Iteratie 3: complexere softwarecode uitfilteren

Empathize

De meningen van de studenten zijn verdeeld over het beperkter invulformulier "Onbekende codes en fouten".

Voor de docent is het nog steeds de prioriteit dat de studenten leren uit de fouten die naar boven komen, en dat ze hun gebruikte code begrijpen. Ook als die code van een AI-tool komt.

Voor het beleid van de hogeschool, is het ook nog steeds zo dat de student zelf verantwoordelijk is voor wat hij indient.

Define

Los van het invulformulier ervoor zorgen dat toch alle studenten leren uit de fouten die ontstaan uit het werken met AI-programmeerassistenten. Daarnaast dit ook kunnen opvolgen.

Ideate

Een uploadzone maken waarin de studenten hun softwarecode moeten uploaden. Via de ingezonden code, kan de docent zien welke fouten de studenten nog tegenkomen. Daarnaast kan de docent zien welke speciale code er gebruikt wordt en of deze correct is of niet. De docent neemt deze informatie mee naar het begin van de volgende les, en spreekt de betrokken studenten hier (klassikaal) over aan om te zien of ze dit kunnen uitleggen. Op deze manier is er meer interactie en opvolging.

Prototype

Standaard opdracht/uploadzone in Toledo/Blackboard.

Standaard PowerPoint-presentatie met op elke slide telkens de speciale softwarecode in kleur aangeduid, met de namen erbij van de studenten die dit gebruikt hebben. In de klas wordt hen gevraagd om dit uit te leggen.

Test

De studenten konden meestal de speciale code die ze gebruikt hadden niet uitleggen.

Link met de onderzoeksvragen:

Deze aanpak kan een aanzet zijn voor studenten om expertise op te bouwen op vlak van het programmeren van software, en het werken met AI-assistenten. Het kan een aanzet zijn voor de docent om tot evaluatievormen te komen, omdat studenten hun werk individueel geanalyseerd wordt en ze hier mondeling over worden ondervraagd. De andere studenten leren hier ook uit, omdat de mondelinge ondervraging klassikaal gebeurt. Tijdens de observatie werd wel duidelijk dat sommige studenten zich hier ongemakkelijk bij voelen.

Iteratie 4: gebruikte softwarecode als lesmateriaal inzetten

Empathize

Voor de docent is het nog steeds de prioriteit dat de studenten leren uit de fouten die naar boven komen, en dat ze hun gebruikte code begrijpen. Ook als die code van een AI-tool komt.

Voor het beleid van de hogeschool, is het ook nog steeds zo dat de student zelf verantwoordelijk is voor wat hij indient.

Define

De studenten moeten weten waar hun gebruikte softwarecode over gaat, omdat ze hier zelf verantwoordelijk voor zijn. Het is nodig dat de studenten weten waar de softwarecode over gaat omdat generatieve AI op dit moment niet betrouwbaar genoeg is om volledig de softwarecode te schrijven. Zie ook de resultaten uit iteratie 1, en de resultaten uit het onderzoek van Liang et al. (2023) en van Tan et al. (2024).

Er is een andere aanpak nodig dan in iteratie 3.

Ideate

De docent neemt de informatie uit de ingediende opdrachten mee naar het begin van de volgende les, en legt deze code klassikaal uit, zodat de studenten eruit kunnen leren.

Het idee is om te werken met een vorm van flexibel/adaptief leren (Graf et al., 2009).

Concreet betekent dit dat er flexibel wordt ingespeeld op de code die de studenten gebruiken. Uit alle opdrachten die de studenten indienen, wordt er een extractie gemaakt. De speciale code hierin wordt door de docent omgevormd naar leerstof. Deze leerstof wordt direct de volgende les aangeboden aan alle studenten.

Prototype

Enerzijds blijft de standaard opdracht/uploadzone in Toledo/Blackboard behouden uit de vorige iteratie.

Daarnaast wordt een standaard PowerPoint-presentatie voorzien met een uitgebreide uitleg van de speciale code van de vorige lessen. De docent legt hierbij de leerstof uitgebreid uit.

Hieronder in Figuur 8 is een voorbeeld te zien van een slide uit de PowerPoint-presentatie. Dit voorbeeld is rechtstreeks genomen uit de softwarecode die een student heeft ingediend voor zijn opdracht.

```
const int ledPins[] = {25, 26, 27, 14, 12};
```

- **const** : constante variabele (waarde kan achteraf niet gewijzigd worden)
- **int** : datatype integer (= geheel getal)
 - *extra info i.v.m. datatypes, zie Toledo bij 'Microcontroller programmeren', en hier : <https://www.geeksforgeeks.org/cpp-data-types/>*
- **[]** : vierkante haken duidt op een **array**. Dit is een verzameling van variabelen, vergelijkbaar met een ladekast. Als er een getal tussen de vierkante haken staat, dan zijn het een vast aantal lades, als er geen getal tussen staat, staat het aantal lades (nog) niet vast.
 - **Omdat "const int" ervoor staat, is het een array van constante integers.**
 - **Na het gelijkheidsteken worden hier tussen accolades de waardes gegeven. In dit geval voor 5 ledpinnen.**



Figuur 8: uitleg van softwarecode, gebruikt in de les in de context van 'adaptief leren'

Test

Aan de houding van de studenten was op te merken dat enkelen sterk betrokken waren, en anderen minder.

De vraag werd daarna aan elke student gesteld of hij wil leren van problemen van anderen in de klas, zoals via een presentatie in het begin van de les.

De meeste studenten willen deze presentatie volgen, o.w.v. volgende redenen:

- als het probleem veel voorkomt bij verschillende studenten;
- om dezelfde fouten niet te maken;
- om eruit te kunnen leren.

Ze willen dat de presentatie kort is (beperkt in tijd), zodat ze langer de tijd hebben om effectief aan een project te werken.

Link met de onderzoeksvragen:

Deze aanpak is een aanzet naar adaptief leren, maar dan gebaseerd op softwarecode van de hele klas. Zoals de studenten aangeven in hun feedback, zou dit vooral interessant zijn als de code eerder veel voorkomt.

De expertise die studenten hiermee opbouwen, is vooral op vlak van het leren van softwarecode. Het is mogelijk om dit te evalueren. Hier kan eenvoudigweg een formatieve evaluatie over gedaan worden, of een soort van quiz aan het begin of het eind van elke les.

Iteratie 5: basisvoorbeeld + plan van aanpak + ‘prompting’-instructies

Empathize

De studenten zijn veel aan het zoeken om een opdracht op te kunnen lossen. Na een les is er feedback gekomen van studenten dat ze niet direct vooruit geraken door het werken met een AI-tool. De softwarecode die de studenten van de AI-tool krijgen, is complex opgebouwd in verhouding tot de voorkennis van de studenten.

Define

Als de totale AI-gegenereerde softwarecode te omvangrijk en complex is om te bevatten voor de student, moeten ze dit kunnen vereenvoudigen.

Ideate

Richtlijnen voorzien voor studenten waarmee ze hun code kunnen vereenvoudigen.

Een basisvoorbeeld voorzien voor studenten, zodat ze iets hebben om op terug te vallen.

De studenten op een juiste manier leren vragen stellen.

Prototypes

Prototype 1: plan van aanpak voor het vermijden of oplossen van programmeerproblemen (met of zonder AI). Zie Figuur 9 hieronder.

Tips bij het programmeren (met of zonder AI)

- **Vertrekken van de basis, van iets dat werkt. Stelselmatig functies en extra complexiteit toevoegen.**
- **De vraag of het probleem opdelen in verschillende delen.**
 - Bijvoorbeeld als de vraag is om waarden van een sensor te schrijven op een microSD-kaart, zijn dit 2 deelproblemen:
 - 1) sensorwaardes uitlezen en
 - 2) schrijven op de microSD-kaart.

Het belangrijkste is hier om op de microSD-kaart te schrijven, en dit kan ook door gewoon losse waardes (bv. continu '1') te schrijven. Zo beperk je de complexiteit van het probleem.
- Als er een **stuk(je) moeilijke code is die wel nodig is: deze analyseren tot je deze begrijpt**, zodat dit geen probleem meer geeft in het begrijpen van je volledige code. Zo bouw je ook expertise op voor de toekomst.
- **Indien nodig, teruggaan naar de absolute basis.**
 - Bij de absolute basis kan je **gemakkelijker onderscheiden** waar de fout ligt (in de aansluitingen, in één specifiek stuk code, ...)
 - **Moeilijke code weglaten als die niet essentieel is** om je resultaat te bereiken.
 - **Tijdelijk stukken code uitcommentariëren** om te checken vanaf waar de code fout loopt.
- **Stapsgewijs doorheen de code gaan.**
 - Dit door **Serial.print("hier ben ik")**; toe te voegen, om te weten tot waar je geraakt in de code.
 - Ofwel via **debugging met breakpoints**. Zie : <https://code.visualstudio.com/Docs/editor/debugging>

O.a. via de voorgaande stappen zoveel mogelijk uitfilteren van code die mogelijks problemen kan geven, zodat je project beheersbaar blijft. **Indien nodig, stappen terugschakelen. Het is altijd beter dat er iets werkt (en dat je dit volledig begrijpt) dan dat er niets werkt (en je niet weet waarom).**

Figuur 9: tips bij het programmeren (met of zonder AI)

Prototype 2: een basisvoorbeeld dat werkt, en dat de studenten eerst kunnen testen. Met een korte uitleg tijdens de les. Zie Figuur 10 hieronder.

```
#include <Arduino.h>
#include <TFT_eSPI.h>

TFT_eSPI tft = TFT_eSPI(); // tft aanmaken (instantie van de TFT_eSPI-klasse)

void setup(void) {
  tft.init(); // Initialiseer het scherm
  tft.setRotation(3); // Stel de draaiing in (normaal 1, bij sommige schermtypes is dit standaard 3)
  tft.fillScreen(TFT_LIGHTBLUE); // Stel de achtergrondkleur in
  tft.fillCircle(100, 100, 50, TFT_ORANGE); // Teken een gevulde cirkel met de gekozen straal en kleur
}

void loop() {
  // niks nodig als er niks verandert
}
```

Figuur 10: basisvoorbeeld dat werkt

Prototype 3: Een korte handleiding met 'prompting'-instructies, specifiek voor het programmeren van een ESP32-microcontroller zoals in de lessen. Deze handleiding is gegenereerd door Microsoft Copilot. Zie Figuur 11 hieronder.

Voor de microcontroller ESP32

Gegenereerd door Microsoft Copilot:

- Wees specifiek:
 - Geef zoveel mogelijk details over wat je wilt bereiken met je ESP32. Wil je bijvoorbeeld verbinding maken met een WiFi-netwerk, gegevens lezen van een sensor, een webserver opzetten, of iets anders?
- Gebruikte bibliotheken:
 - Vermeld de bibliotheken die je gebruikt of van plan bent te gebruiken. Dit helpt mij om code te genereren die compatibel is met je huidige setup.
- Hardware details:
 - Geef informatie over eventuele extra hardware die je gebruikt, zoals sensoren, displays, etc.
- Foutopsporing:
 - Als je problemen ondervindt met je huidige code, deel dan de foutmeldingen of beschrijf het gedrag dat je ziet.
- Veiligheid eerst:
 - Zorg ervoor dat je project veilig is. Dit betekent dat je geen gevoelige informatie zoals wachtwoorden in je code moet opnemen. Als je bijvoorbeeld verbinding maakt met een WiFi-netwerk, zorg er dan voor dat je het wachtwoord op een veilige manier opslaat.

Figuur 11: beknopte prompthandleiding, specifiek voor de microcontroller 'ESP32'

Test

De studenten vinden het veel beter dat er een standaard voorbeeld beschikbaar is waar ze van kunnen vertrekken.

I.v.m. de tips bij het programmeren: als de studenten op problemen stootten tijdens de lessen, en de docent wees op dit document of een instructie erin, hielp dit de student om het probleem op te lossen. Uit de feedback van alle studenten bleek dat ze het document duidelijk (genoeg) vinden. De ervaring van verschillende studenten in hoe ze best problemen oplossen of vermijden: stapsgewijs werken, de code opdelen in eenvoudige stukken, en de code simpel houden als het kan.

Eén student vindt het nuttige instructies, maar wijst erop dat het zelfs dan soms niet lukt voor AI om een duidelijk antwoord te geven. Dit kan te maken hebben met de manier van vraagstelling.

Toch bleek tijdens de bevraging dat de studenten zelf geen vragende partij waren om uitgebreide uitleg te krijgen i.v.m. 'prompting'.

Link met de onderzoeksvragen:

Deze aanpak geeft meer rust aan de studenten en komt eraan tegemoet dat de AI-tools op dit moment nog verkeerde antwoorden geven.

Het kan helpen in de opbouw van expertise, omdat er op deze manier een correcte basis is die de studenten begrijpen, van waarop ze verder kunnen bouwen.

De manier van aanpak en de tips bij het programmeren kunnen helpen bij het adaptief leren, omdat een manier van aanpak los staat van de kennis. Generatieve AI is getraind op grote hoeveelheden kennis, en bevat deze in principe al. Vaardigheden in de omgang met AI kunnen hierdoor belangrijker worden, afhankelijk van de kwaliteit van het LLM-framework (zie hoofdstuk Literatuurstudie en probleemstelling, p.4).

Iteratie 6: geen “opdrachten”, maar mogelijkheden

Empathize

De intrinsieke motivatie van de studenten lijkt te verlagen op het moment dat er opdrachten aan te pas komen. Volgens Sultan et al. (2017), zorgt het geven van meer autonomie en vrijheid aan studenten hen meer wilskracht voor moeilijke leertaken.

Define

De motivatie van de studenten verhogen door het verhogen van de vrijheid en autonomie.

Ideate

Vanuit de aangeleverde basisvoorbeelden, kunnen studenten zelf een nieuw project/oefening kiezen die ze willen maken, volgens wat hen zelf interesseert. Ze kunnen terecht bij de docent voor extra inspiratie.

Er wordt geen “opdracht” meer voorzien, maar een vrije uploadzone waarin projecten doorgestuurd kunnen worden. In deze uploadzone kunnen de studenten hun werk tonen van zodra ze dat als ze dat willen, als ze hier klaar voor zijn en/of als ze feedback willen.

De docent bekijkt de projecten en stuurt bij waar nodig. Als iets niet volledig af is binnen de voorziene tijdsspanne, maar de volgende les toch een nieuw onderwerp aan de orde is, dan worden de projecten die niet af zijn gestopt. Deze kunnen evt. wel verder worden geïntegreerd in een volgende fase. Of de student kan dit zelf thuis verder afwerken.

Deze iteratie is zeer sterk gelinkt met flexibel/adaptief leren.

Prototype

Standaard opdracht/uploadzone in Toledo/Blackboard, met een onbeperkt aantal pogingen ingesteld. De bedoeling hiervan is géén vastgelegde “opdracht”. Het is gericht op “zelf op ontdekking gaan”.

Test

Bij de start van de les, zei de docent dat de studenten mochten kiezen wat ze wilden maken, zolang het met een project met touchscreen was.

Over het algemeen hadden geen idee van wat zelf wilden maken. De docent is rondgegaan bij elke student om te kijken hoever ze staan in hun leerniveau, en wat hun interesses zijn.

De studenten hadden elk verschillende sensoren ter beschikking, die de docent in het begin van het semester verdeeld had volgens de interesse van elke student. Hierop aansluitend, gaf de docent ideeën i.v.m. wat de studenten konden maken. Hierin werd relatief veel vrijheid gegeven, maar dit moest zich wel afspelen binnen de context van het leerstof van dat moment (nl. het uitwerken van een project met touchscreen).

Feedback studenten:

De vraag werd achteraf gesteld aan de studenten wat voor hen het best werkt:

- A: Vaste opdrachten maken
- B: Vrij projecten maken volgens eigen interesses (met leerondersteuning waar nodig)
- C: Combinatie van vaste opdrachten en vrije projecten

De verdeling van de antwoorden van de studenten:

- A: 60%
- B: 20%
- C: 20%

Extra toelichtingen hierbij door de studenten, waarom ze hiervoor kiezen:

- Vaste opdrachten:
 - o Om basiskennis op te bouwen
 - o Samen met de klas kunnen nadenken
 - o Zo leert iedereen dezelfde leerstof en kunnen we elkaar helpen wanneer nodig.
 - o Meteen een concreet idee om mee aan de slag te gaan
- Vrij projecten maken
 - o Nadat er een vaste opdracht gemaakt is.
 - o Zodat de docent kan zien hoever de student staat met de leerstof, en hoe bereid de student is om ingewikkelde opdrachten te maken.
 - o Altijd leuk om voor een bepaalde periode je eigen interesses achterna te gaan.
 - o Dat je zelf iets kan ontwikkelen waar je echt trots op bent!

Interpretatie:

Het lijkt afhankelijk van het type persoonlijkheid te zijn wat er het beste werkt. Hetgeen wel duidelijk is, is dat er een basis nodig is. Voor sommigen is het basisvoorbeeld voldoende, zoals in een vorige iteratie besproken. Anderen willen (eerst) meer verdieping voordat ze vrij projecten maken. Daarnaast kan het ook te maken hebben met zelfvertrouwen en met persoonlijke vaardigheden.

Tweede fase: verdiepende enquête

Uit deze enquête is naar boven gekomen dat een AI-tool de studenten helpt om van start te geraken met een (nieuwe) opdracht. Daarnaast helpt het om op eigen leertempo bij te leren en om de softwarecode beter te begrijpen. Ook kan het helpen om sneller te leren. Ze kunnen bijvoorbeeld specifieke vragen stellen, extra verduidelijking krijgen, enz. Een AI-tool helpt sommige studenten ook om nieuwe softwarecode te ontdekken.

De helft van alle studenten zegt dat het maximum 3 minuten tijd kost tot ze een goed antwoord hebben van een AI-tool. Dit heeft volgens hen zelf ook te maken met de manier van vraagstelling aan een AI-tool. De studenten zijn zich hiervan bewust. Er is een onderscheid tussen verschillende studenten in de vaardigheid hierin. De studenten hebben volgens hen geen hulp nodig met het 'prompten', de manier van vraagstelling aan een AI-tool. Ook hebben ze zelfstandig doorheen de eerste lessen geleerd welk type van vraagstelling beter werkt. Er is een verschil in aanpak tussen verschillende studenten, hetgeen interessant kan zijn voor studenten om hierin van elkaar te leren.

Er is een verschil tussen studenten in de mate waarin ze AI-tools willen gebruiken. De meesten antwoorden dat ze enkel een AI-tool gebruiken, als ze ervan uitgaan die ook echt kan helpen. Een derde van de studenten wil zoveel mogelijk AI gebruiken.

De meeste studenten gebruiken geen online forums, terwijl dit vóór de intrede van AI-tools veel gebruikt werd om problemen op te lossen.

De studenten willen over het algemeen de lessen fysiek volgen (niet online).

Conclusie en discussie

Conclusie

“Wat is een goede aanpak om studenten te leren programmeren, in tijden waar generatieve AI beschikbaar is?”

Hierop is een antwoord gezocht in dit onderzoek, via enerzijds een iteratief experimenteel onderzoek volgens de principes van ‘Design Thinking’, en anderzijds via een verdiepende enquête. Richtpunten hierbij waren het achterhalen welke expertise er (nog) nodig is bij studenten, hoe ze deze kunnen opbouwen, en hoe deze expertise geëvalueerd kan worden. Ook adaptief leren was een focuspunt.

Wat geen goede aanpak bleek, is het laten invullen van een uitgebreid online invulformulier (zie Iteratie 1: online invulformulier, p. 11).

Wat de studenten wel heel goed vonden, is om een basisvoorbeeld te krijgen voor een nieuw lesonderwerp. Dit is te begrijpen, omdat generatieve AI soms nog foute of onnodig moeilijke softwarecode genereert (zie ook Liang et al. (2023) en Tan et al. (2024)). Doordat de studenten nog geen echte basiskennis hebben, maakt dit het moeilijker om foute antwoorden te verwerken. Dit bleek ook uit de observaties.

Daarnaast gaven de studenten aan dat generatieve AI hen helpt in hun leerproces. Ze kunnen bijvoorbeeld specifieke vragen stellen, extra verduidelijking krijgen, enz. Dit helpt hen in de eerste plaats om van start te geraken met een (nieuwe) opdracht. Verder helpt het om op eigen leertempo bij te leren en om de softwarecode beter te begrijpen. Ook kan het helpen om sneller te leren en om nieuwe informatie te ontdekken. Dit bevestigt dat generatieve AI bijdraagt aan adaptief leren (i-Learn Vlaanderen, 2021): afgestemd op de student en beschikbaar waar nodig. Daarnaast bevestigt dit ook het onderzoek van Ramazan et al. (2023).

Wat ook werkte, is het scannen van softwarecode van studenten op speciale softwarecode en/of fouten. Deze code en/of fouten worden daarna samengenomen. De meest voorkomende zaken worden eruit gehaald, en gebruikt als leerstof voor de les erna. Dit ligt in lijn van de doelstellingen van zowel de docent, de hogeschool, als de studenten. Ook ligt dit in lijn van adaptief leren, met het verschil dat de leerstof hier klassikaal wordt samengenomen, terwijl dit bij adaptief leren in principe individueel is.

Discussie

Hieronder staan de aanbevelingen voor het aanleren van software, waarbij de studenten gebruikmaken van generatieve AI.

Opstart van een nieuw leeronderwerp/-hoofdstuk

Basisvoorbeeld

Een correct basisvoorbeeld is op dit moment nog steeds zinvol. Als een AI-tool dit niet direct kan geven in hoge kwaliteit, is het belangrijk dat de docent dit kan geven.

Voorzien van handvaten

Studenten moeten weten hoe ze uit het kluwen van AI moeten geraken, en handvaten hiervoor krijgen. Enerzijds zijn dit richtlijnen of een schema. Anderzijds kunnen dit tips zijn in het 'prompten'. Dit kan via uitwisseling van ervaringen van studenten, aangevuld met ervaring van de docent. Deze handvaten moeten niet opgedrongen worden, maar beschikbaar zijn, ook doorheen de rest van de projectcyclus (zie hieronder).

Na de opstart: project- en leercyclus

Keuze project

De student kiest zelf een eigen project/oefening om te maken, dat in lijn ligt van de leercontext.

Opvolgen en documenteren van hetgeen geleerd werd

Een platform of formulier voor opvolging van problemen en speciale code.

Dit kan zijn via een uploadzone in een Learning Management System zoals Blackboard, ofwel via een ticketsysteem of andere waarin de studenten hun code insturen.

De docent volgt dit op, controleert, en brengt samen. Fouten en moeilijke code die veel voorkomen, worden samengebracht, en klassikaal uitgelegd in de les.

Evaluëren

Tijdens het onderzoek is gebleken dat studenten soms al de volledige werkende code van AI krijgen. Op dat moment moet er gekeken worden naar een juiste manier van evalueren.

Algemene voorstellen voor manieren om te kunnen evalueren of de student de noodzakelijke vaardigheden bezit:

- Zorgen voor individuele projecten voor studenten (die bij voorkeur in lijn liggen van de eigen interesses).
- Een bepaald doel stellen dat ze moeten behalen binnen een bepaalde tijd. Het doel is bijvoorbeeld een werkend resultaat uitwerken dat aan bepaalde vereisten voldoet.

Voorstellen voor evaluatie, door middel van simulatie van situaties die zich ook in de professionele wereld voordoen:

- Het project onderwerpen aan "extremere" externe factoren, om te zien of het programma blijft werken, en indien niet, kijken in welke mate de student dit kan hanteren.
- Het project laten uitbreiden, waardoor eventueel een herstructurering van de softwarecode nodig is. Afhankelijk van de capaciteiten van AI, kan het zijn dat dit extra inzicht van de student vereist.

De manier van evalueren kan zowel proces- als productgericht zijn, aangevuld met een mondelinge evaluatie. Hierbij moet de student dan de gebruikte aanpak toelichten. De docent kan dan nagaan of deze aanpak verbeterd kan worden.

Op dit moment zijn AI-tools nog in volle ontwikkeling. Daarom is het op vlak van de expertise die (nog) nodig is, moeilijk om een eenduidig antwoord te geven. AI-tools werken met een benadering (zie bv. de Transformer-architectuur van Vaswani et al. (2017)). Er is daarna een verbeterde betrouwbaarheid gekomen door onder andere Devlin et al. (2018) en Howard et al. (2018). Later is een manier gevonden om beter te kunnen antwoorden op niet-getrainde data, door Brown et al. (2020). Momenteel zijn er nog onderzoeken lopende, onder andere om de betrouwbaarheid te verbeteren. Een van de recentere projecten is GPT-4 (Achiam et al, 2023), en de ontwikkeling van 'Retrieval-Augmented Generation' (RAG). Omdat AI getraind is op kennis, en hoe meer AI betrouwbaar wordt op vlak van kennis, hoe minder de mens kennis nodig zou moeten hebben. In kader van evaluatie wordt daarom aangeraden om meer te evolueren richting het ontwikkelen van vaardigheden bij de studenten.

Daarnaast, als de capaciteiten van AI-tools verbeteren, wordt het belangrijk om hier verantwoord mee om te springen.

Toekomst

Verder onderzoek

Het onderzoek is nu gebeurd in een klasgroep van 11 studenten binnen de opleiding graduaat Internet of Things. Het is aangewezen om dit verder te testen met verschillende groottes en samenstellingen van klasgroepen.

Participatory Action Research (PAR)

Het kan interessant zijn om studenten nauwer te betrekken bij de ontwikkeling van nieuwe manieren van werken met AI. Dit wil zeggen dat de studenten mee vorm zullen geven aan hoe AI in hun leerproces gebruikt kan worden.

Online adaptief leerpad via AI gestuurd

Na verloop van tijd is het een idee om een online leerpad te ontwikkelen dat via AI gestuurd is. Gedeeltes die de student niet goed kan, kunnen door AI achterwege gelaten worden, of net meer herhaald. AI kan dan voorstellen doen over leerstof die de student meer zou liggen. Het doel hierbij is dat de student snel én gefundeerd groeit in een richting die hem/haar past.

Reeds bestaande tools die mogelijks verder die richting uitgaan in de toekomst:

- <https://forms.office.com/>
- <https://www.anthology.com/en-emea> (<https://Blackboard.com>)
- <https://www.i-learn.vlaanderen/>
- <https://feedbackfruits.com/>
- <https://www.gradescope.com/>

Gezamenlijke cursus-/informatiebouwers

Instructies, basisvoorbeelden en flexibel leerstof aanbieden, kan automatisch gestuurd worden vanuit een centrale dataset van de hogeschool zelf. Zo weet de school dat de kwaliteit van de leerstof die studenten krijgen gelinkt is aan eigen ontwikkeld lesmateriaal.

Service Learning

De groep studenten in deze klas had weinig tot geen ervaring op voorhand. In een volgende stap kan het principe van Service Learning mee geïntegreerd worden. Hierbij wordt een concrete praktische situatie als uitgangspunt genomen voor het uitwerken van een project. Als AI zich daarnaast verder kwalitatief kan ontwikkelen, kan het mogelijk zijn dat meer gevorderdere projecten kunnen uitgewerkt worden door studenten doordat ze meer op AI kunnen vertrouwen.

<https://www.servicelearningvlaanderen.be/service-learning/>

PS: AI blijft een gereedschap, dat vooral in specifieke situaties een antwoord kan zijn op een probleem. In andere situaties zullen andere manieren misschien beter werken.

Referentielijst

- Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... & McGrew, B. (2023). Gpt-4 technical report. arXiv preprint arXiv:2303.08774.
- Acosta-Gonzaga E. (2023). The Effects of Self-Esteem and Academic Engagement on University Students' Performance. Behavioral sciences (Basel, Switzerland), 13(4), 348. <https://doi.org/10.3390/bs13040348>
- Alexandre, Ardichvili. (2022). The Impact of Artificial Intelligence on Expertise Development: Implications for HRD. Advances in Developing Human Resources, 24(2):78-98. doi: 10.1177/15234223221077304
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165
- Bui, Trinh. (2022). Enhancing Students' Interaction with AI Teaching Assistants: Roles of Novelty and Innovativeness. Academy of Management Proceedings. 2022. 10.5465/AMBPP.2022.15728.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
- Furze, L., Perkins, M., Roe, J., & MacVaugh, J. (2024). The AI Assessment Scale (AIAS) in action: A pilot implementation of GenAI supported assessment. ArXiv, abs/2403.14692.
- Graf, Sabine & Lan, Chung & Liu, Tzu-Chien & Kinshuk, Dr. (2009). Investigations about the Effects and Effectiveness of Adaptivity for Students with Different Learning Styles. Proceedings - 2009 9th IEEE International Conference on Advanced Learning Technologies, ICAIT 2009. 415-419. 10.1109/ICAIT.2009.135.
- Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146.
- i-Learn Vlaanderen. (2021, 30 december). Adaptief leren = leren op maat - i-Learn Vlaanderen. <https://www.i-learn.vlaanderen/nieuws/edtech/adaptief-leren-leren-op-maat/>
- Jian, M. J. K. O. (2023). Personalized learning through AI. Advances in Engineering Innovation, 5(1), None. <https://doi.org/10.54254/2977-3903/5/2023039>
- Liang, J.T., Yang, C., & Myers, B.A. (2023). Understanding the Usability of AI Programming Assistants. ArXiv, abs/2303.17125.
- Mote, Asst. (2024). Role of AI in Personalized Education. International Journal of Advanced Research in Science, Communication and Technology. 108-111. 10.48175/IJARSCT-15919
- Nicola, Capuano., Santi, Caballé. (2020). Adaptive Learning Technologies. Ai Magazine, 41(2):96-98. doi: 10.1609/AIMAG.V41I2.5317
- Oktafiani, Zakia & Yusri, Yusri. (2021). The Relationship of Self Confidence to Students Learning Achievement. Counseling and Humanities Review. 1. 20-26. 10.24036/000411chr2021.
- Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pretraining. OpenAI Blog.

Ramazan, Yilmaz., Fatma, Gizem, Karaoglan, Yilmaz. (2023). Augmented intelligence in programming learning: Examining student views on the use of ChatGPT for programming learning. Computers in Human Behavior, doi: 10.1016/j.chbah.2023.100005

Se-Jung, Lee., Kyungsuk, Chang., Kyungmee, Lee. (2023). A Classroom-based Study on the Application of Artificial Intelligence Translator: The Use and Perception of AI by Low-achieving Primary School English Learners. Co'deung yeong'eo gyo'yug, 29(1):31-53. doi: 10.25231/pee.2023.29.1.31

Tan, X., Long, X., Ni, X., Zhu, Y., Jiang, J., & Zhang, L. (2024). How far are AI-powered programming assistants from meeting developers' needs? ArXiv, abs/2404.12000.

UCLL. (2024). Beleidskader Generatieve Artificiële Intelligentie.
<https://intranet.ucll.be/nl/node/52323>

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

Vereijken, J. (2022, 25 maart). Wat is Design Thinking? Proces en kracht van deze methode. Spring Company. <https://springcompany.nl/wat-is-design-thinking-het-proces-en-de-kracht-van-deze-populaire-innovatiemethode/>

Bijlagen

Creatief eindproduct: promptgenerator

Handleiding

Deze promptgenerator is ontwikkeld specifiek vanuit ervaring op vlak van de problemen die kunnen ontstaan bij het vragen van code aan een AI-tool.

In het groene gedeelte kan je opgeven welke componenten of functionaliteiten je wil gebruiken. Door het opgeven van een basisvoorbeeld kan je optioneel het resultaat al op je huidige (basis-)voorbeeld afstemmen.

In het gele gedeelte staat de gegenereerde 'hoofd'-prompt eerst. Eronder staan mogelijke bijvragen om verder te finetunen of om meer informatie te krijgen. De 'hoofd'-prompt kan je eenvoudig kopiëren en plakken (in tekst) in een generatieve AI-tool.

Het resultaat dat uit de AI-tool komt, bevat de softwarecode en de meeste praktische informatie die je nodig hebt om het project uit te voeren.

De promptgenerator kan zowel nuttig zijn voor leerkrachten als lerenden. Voor leerkrachten kan het helpen om verschillende voorbeelden uit te proberen, om lessen voor te bereiden (door bijvoorbeeld de speciale code en mogelijke fouten al op voorhand te verwerken). Daarnaast kan het in de lessen gebruikt worden door studenten, waarbij het nog gemakkelijker wordt om van start te geraken. Via de prompt wordt ook automatisch uitleg gegeven over speciale code en mogelijke fouten, wat het adaptief leren bevordert.

Link naar het bestand

https://docs.google.com/spreadsheets/d/1nNL0S9S-JunnXI7mInAVEqSYdOMRs3f5/edit?usp=drive_link&ouid=116276579782709564087&rtpof=true&sd=true

Printscreen

Keuzemenu	
Kies een sensor	temperatuur-, druk- en vochtigheidssensor
Kies een actuator (bv. LED, motor, ...)	LED
Kies een microcontroller	ESP32 DOIT DEVKIT V1
Speciale functionaliteiten (optioneel)	via WiFi en MQTT de waardes doorsturen naar een centraal punt
Eigen basisvoorbeeld toevoegen (optioneel)	

Hoofdvraag	
Gegenereerde prompt (kopiëren en plakken als vraag in een AI-tool, bv. via Ctrl+Shift+V)	<i>Ik wil een project maken in Visual Studio Code, m.b.v. de extensie PlatformIO.</i> <i>Het project moet bestaan uit:</i> - microcontroller: ESP32 DOIT DEVKIT V1 - sensor: temperatuur-, druk- en vochtigheidssensor DHT - actuator: LED. - speciale functionaliteiten: via WiFi en MQTT de waardes doorsturen naar een centraal punt 1. Geef me hiervoor de softwarecode (zonder Engels hierin te gebruiken). Vereenvoudig deze code zoveel mogelijk. 2. Zet beschrijvende commentaren mee in de code zodat duidelijk wordt hoe de code werkt. Zet deze commentaar zoveel mogelijk rechts van de lijn code, behalve als de verklaring over verschillende lijnen code gaat. Zorg dat de commentaar mooi is uitgelijnd. 3. Geef de bibliotheken die in PlatformIO aan mijn project moeten gekoppeld worden, zonder de manier waarop te geven. 4. Geef de aanpassingen die ik zelf nog moet doen in deze bibliotheken. 5. Geef de pinnummers van de microcontroller die ik moet gebruiken, en hoe ik deze moet aansluiten. 6. Geef voor elke sensor het detail van hoe elke pin moet aangesloten worden 7. Geef voor elke actuator het detail van hoe elke pin moet aangesloten worden 8. Geef aan welke aandachtspunten en problemen er mogelijks zijn. 9. Geef een lijst van alle componenten die ik nodig heb, inclusief types van weerstanden enz. 10. Geef een opsomming van elke functie en lijn code in je voorbeeld die een absolute beginner niet kent, en geef hiervan telkens een korte uitleg.

Bijvragen	
Actualisering van de gegeven code	<i>Gebruik nu de meest gangbare bibliotheken van dit moment.</i>
Mogelijke uitbreiding van het project	<i>Geef mogelijke variaties binnen de context van dit project.</i>
Prompt voor het vragen van het uitleggen van het basisvoorbeeld (kopiëren en plakken als vraag in een AI-tool)	<i>Leg dit voorbeeld in detail uit, zodat ook iemand met weinig of geen basiskennis van software dit kan begrijpen.</i>
Prompt voor het uitleggen van één specifieke lijn code	<i>Wat betekent "..."</i>

Resultaat

Dit is een voorbeeld van hoe je een project kunt opzetten met de ESP32 DOIT DEVKIT V1 microcontroller, DHT sensor, en een LED actuator. De code is geschreven in C++ voor de PlatformIO extensie in Visual Studio Code.

```
// Inclusief de nodige bibliotheken
#include <WiFi.h> // Voor WiFi-functionaliteit
#include <PubSubClient.h> // Voor MQTT-functionaliteit
#include <DHT.h> // Voor de DHT sensor

// Definieer de pinnen
#define DHTPIN 4 // Pin waar de DHT sensor is aangesloten
#define LEDPIN 2 // Pin waar de LED is aangesloten

// Maak een DHT-object
DHT dht(DHTPIN, DHT11);

// WiFi-informatie
const char* ssid = "jouw_ssid"; // Vervang dit door jouw SSID
const char* password = "jouw_wachtwoord"; // Vervang dit door jouw wachtwoord

// MQTT-informatie
const char* mqtt_server = "mqtt_server"; // Vervang dit door jouw MQTT-server
const char* mqtt_user = "mqtt_gebruiker"; // Vervang dit door jouw MQTT-gebruikersnaam
const char* mqtt_password = "mqtt_wachtwoord"; // Vervang dit door jouw MQTT-wachtwoord
const char* mqtt_topic = "mqtt_onderwerp"; // Vervang dit door jouw MQTT-onderwerp

WiFiClient espClient;
PubSubClient client(espClient);

void setup() {
    // Start de seriële communicatie
    Serial.begin(115200);

    // Start de DHT sensor
    dht.begin();

    // Stel de LED-pin in als OUTPUT
    pinMode(LEDPIN, OUTPUT);

    // Verbind met WiFi
    WiFi.begin(ssid, password);
```

```

while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Verbinden met WiFi...");
}
Serial.println("Verbonden met WiFi");

// Stel de MQTT-server in
client.setServer(mqtt_server, 1883);
}

void loop() {
    // Als de client niet is verbonden met de MQTT-server, probeer dan
    // opnieuw te verbinden
    if (!client.connected()) {
        reconnect();
    }
    client.loop();

    // Lees de temperatuur en vochtigheid
    float h = dht.readHumidity();
    float t = dht.readTemperature();

    // Verzend de temperatuur en vochtigheid via MQTT
    String payload = "Temperatuur: " + String(t) + " °C, Vochtigheid: " +
String(h) + " %";
    client.publish(mqtt_topic, payload.c_str());

    // Wacht 1 seconde
    delay(1000);
}

void reconnect() {
    // Probeer opnieuw te verbinden met de MQTT-server
    while (!client.connected()) {
        Serial.print("Verbinden met MQTT-server...");
        if (client.connect("ESP32Client", mqtt_user, mqtt_password)) {
            Serial.println("verbonden");
        } else {
            Serial.print("mislukt, rc=");
            Serial.print(client.state());
            Serial.println(" probeer het over 5 seconden opnieuw");
            delay(5000);
        }
    }
}
}

```

Deze code maakt verbinding met een WiFi-netwerk en een MQTT-server, leest de temperatuur en vochtigheid van de DHT-sensor, en stuurt deze waarden via MQTT naar een centraal punt. De LED-pin is ingesteld als OUTPUT, maar wordt in deze code niet gebruikt. Je kunt de LED aansturen door `digitalWrite(LEDPIN, HIGH);` te gebruiken om de LED aan te zetten, en `digitalWrite(LEDPIN, LOW);` om de LED uit te zetten.

De bibliotheken die je nodig hebt zijn:

- WiFi.h
- PubSubClient.h
- DHT.h

Je moet de volgende waarden aanpassen aan jouw situatie:

- ssid
- password
- mqtt_server
- mqtt_user
- mqtt_password
- mqtt_topic

De DHT-sensor moet worden aangesloten op pin 4 van de ESP32. De LED moet worden aangesloten op pin 2. Zorg ervoor dat je de juiste weerstanden gebruikt voor de LED om te voorkomen dat deze doorbrandt.

Let op dat de DHT-sensor een beetje traag kan zijn bij het lezen van waarden, dus het is een goed idee om wat vertraging in je code op te nemen om te voorkomen dat je te snel probeert te lezen. Ook kan de WiFi-verbinding soms wegvallen, dus het is belangrijk om te controleren of de verbinding nog steeds actief is voordat je probeert gegevens te verzenden.

De componenten die je nodig hebt zijn:

- ESP32 DOIT DEVKIT V1
- DHT-sensor
- LED
- Weerstanden voor de LED (meestal rond 220 Ohm, maar dit kan variëren afhankelijk van de specificaties van je LED)

Elke functie en lijn code in dit voorbeeld is vrij standaard voor iemand die bekend is met de Arduino IDE en C++, maar als je een absolute beginner bent, zijn hier enkele dingen die je misschien niet kent:

- `#include <...>`: Dit is een preprocessor commando dat vertelt de compiler om de bibliotheek op te nemen in de code.
- `#define ...`: Dit is een andere preprocessor commando dat een macro definieert.
- `const char* ...`: Dit definieert een constante string.

- `void setup() {...}`: Dit is een functie die één keer wordt uitgevoerd bij het opstarten van de ESP32.
- `void loop() {...}`: Dit is een functie die continu wordt uitgevoerd zolang de ESP32 aan staat.
- `Serial.begin(115200);`: Dit start de seriële communicatie met een baudrate van 115200.
- `pinMode(LEDPIN, OUTPUT);`: Dit stelt de LEDPIN in als een output pin.
- `WiFi.begin(ssid, password);`: Dit maakt verbinding met het WiFi-netwerk.
- `client.setServer(mqtt_server, 1883);`: Dit stelt de MQTT-server in.
- `dht.readHumidity();` en `dht.readTemperature();`: Dit leest de vochtigheid en temperatuur van de DHT-sensor.
- `client.publish(mqtt_topic, payload.c_str());`: Dit stuurt de payload naar het MQTT-onderwerp.
- `delay(1000);`: Dit wacht 1 seconde.