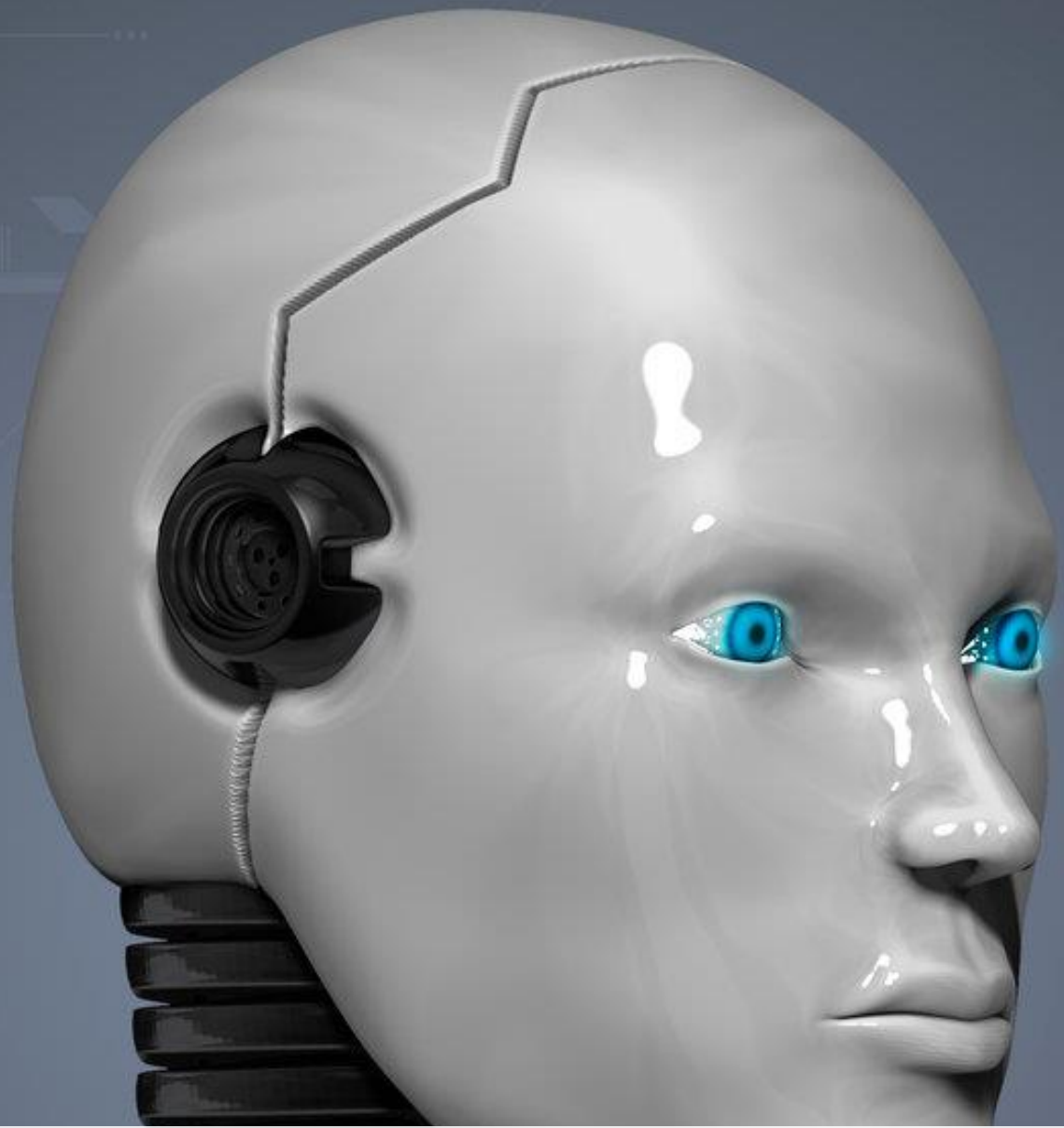




Master Thesis

Onderzoek van Onderwijs

FACULTY OF BEHAVIOURAL MANAGEMENT AND SOCIAL SCIENCES
MASTER SCIENCE EDUCATION AND COMMUNICATION – INFORMATICA

EXAMINATION COMMITTEE

dr. J.T. van der Veen

dr. L.E.I. Breymann

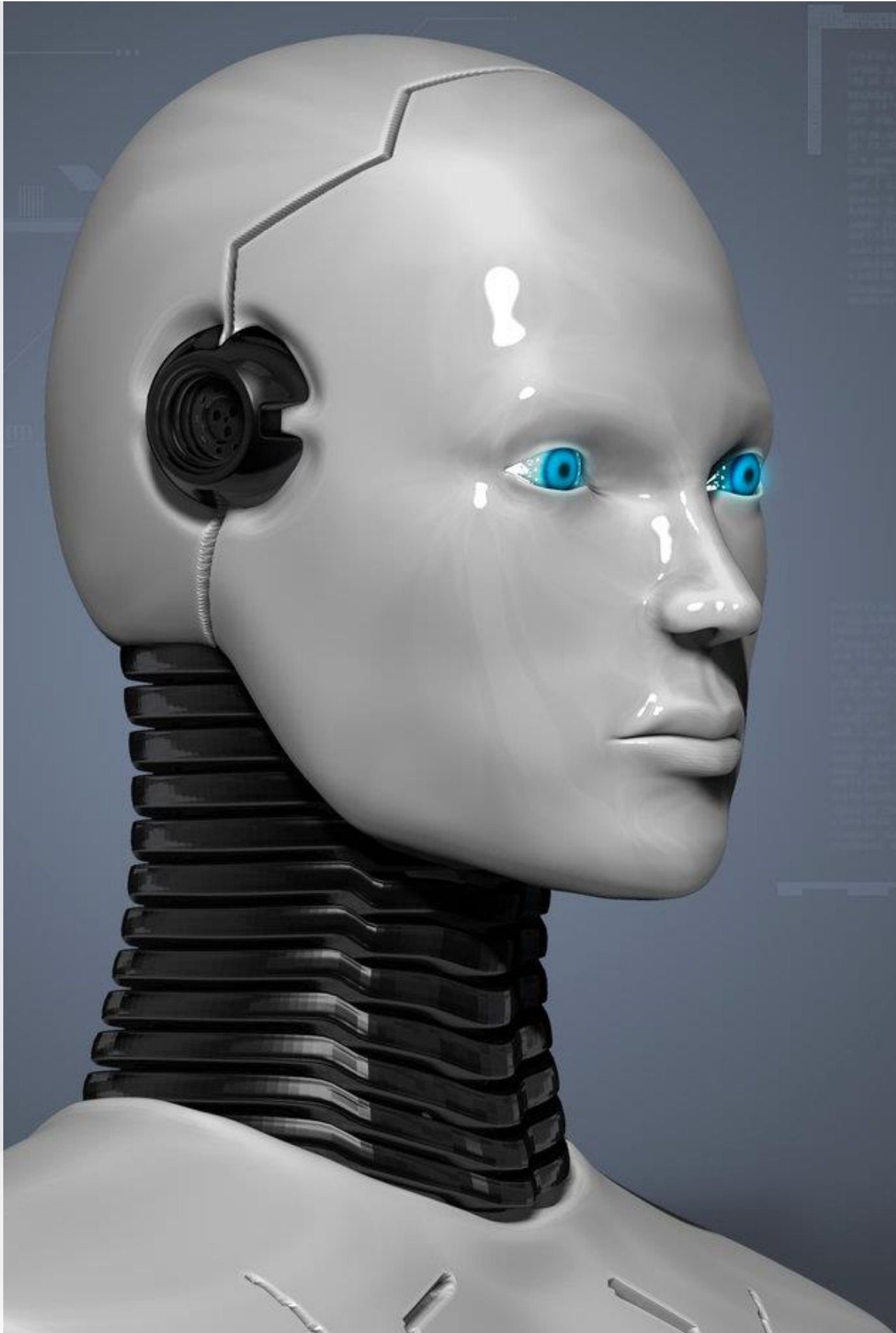
Complete les module

Dit is het verzamel document van de module gezichtsuitdrukkingen ontworpen voor het Impuls project. Dit document bestaat uit drie onderdelen:

1. Ontwerp verslag,
Dit deel bevat het ontwerpproces van de module
2. Docenten materiaal
Dit deel geeft een handvat aan de docent hoe hij de lessen van de module kan inrichten
3. Leerling materiaal
Dit deel bevat de opdrachten die de leerlingen moeten uitvoeren.

Het docenten en leerlingen materiaal zijn onderling sterk verbonden en zullen vaak in combinatie ingezet worden. Het ontwerp verslag is zelfstandiger, en dient puur ter verdieping van de ontwerpkeuzes van de module.

Ontwerp verslag



Lesmodule Sociale Robotica

Gezichtsuitdrukkingen

Ontwerp verslag

Deze module is ontworpen voor het Impuls project in het kader Onderzoek van Onderwijs. Op het moment van start was reeds een ut-onderzoek en een richting gekozen voor de module: er kwamen twee informatica modules gericht op sociale robotica waarvan één het onderwerp “sociale robots in de ouderen zorg” en de andere “Emotieherkenning/gezichtsuitdrukkingen”.

Deze ontwerpopdracht heeft de modulevoorzet van “Emotieherkenning/gezichtsuitdrukkingen” als basis genomen, en dit uitgewerkt tot een complete les module met docent en leerling materiaal.

De lessen zijn uitgetest op het Stedelijk Lyceum, locatie Kottenpark in Enschede, in een 5vwo klas onder leiding van Dirk-Jan van de Poppe. Op basis van de leerpunten en uitkomsten van deze try-out is het materiaal verbeterd.

Allereerst zal in het hoofdstuk onderzoeksmethodiek worden uitgelegd op basis van welke methodiek het onderzoek is opgezet, en wat de bijbehorende ontwerp vraag is. In het hoofdstuk daarna zijn enkele onderzoeksvragen behandeld om een theoretisch kader binnen het ontwerp vast te stellen. Vervolgens (Ontwerpfase) is uitgewerkt hoe de module ontworpen is en waarom bepaalde ontwerpkeuze genomen zijn. Waarna door de evaluatie van de pilot en het materiaal, aangevuld met een hoofdstuk geplaatst met verbeteren die reeds gedaan zijn op het materiaal n.a.v. de pilot en verbeteren die nog gedaan kunnen worden. Aan het einde van het verslag is de conclusie en discussie te vinden.

Inhoudsopgave

Ontwerp verslag	1
Inhoudsopgave	2
Onderzoeksmethodiek	3
Onderzoekfase (Problem investigation).....	5
Ontwerpfase (Treatment design)	9
Evaluatiefase (Design validation / Design implementation)	14
Bijstellen materiaal n.a.v. Pilot (Implementation evaluation)	16
Conclusie & Discussie	17
Verwijzingen	18
Bijlage A - Opzet lesmodule – Dirk-Jan van de Poppe en Jurre Moor	19
Bijlage B – Leerdoelen matrix.....	22
Bijlage C - Relevante exameneisen concept advies-examenprogramma	23
Bijlage D - Relevante exameneisen huidige examenprogramma.....	25

Onderzoeksmethodiek

Dit onderzoek is gestructureerd volgens de Design Science Methodology (Wieringa, 2010) van Wieringa. Design Science is het ontwerpen van een artefact binnen een bepaalde context om een 'real world' probleem op te lossen. Het artefact zal in dit geval een lesmodule zijn. Volgens Wieringa kan je een ontwerp probleem in twee activiteiten splitsen: Het ontwerp van je artefact en kennis problemen beantwoorden. Deze kennis vragen geven de basis voor het ontwerpen van je artefact.

Het ontwerpen van het artefact wordt gedaan door het volgen van de designcyclus zoals Wieringa hem heeft gedefinieerd. Binnen de verschillende stappen van de designcyclus zal je kennisproblemen definiëren voor het opdoen van kennis welk nodig is voor het ontwerp van je artefact. Hieronder zijn alle stappen van de designcyclus beschreven.

Designcyclus

Hierin wordt een ontwerp cyclus van 4 stappen aangehouden: Problem Investigation, Treatment design, Design Validation en design implementation.

Problem investigation

In de eerste fase ga je het probleem onderzoeken en vaststellen:

- Wat is het probleem.
- In welke context vindt het probleem plaats
- Wat is het doel binnen het probleem
- Welke stakeholders zijn bij het probleem betrokken.

Treatment design

Zodra het probleem bekend is ga je een artefact ontwerpen dat een oplossing is voor het benoemde probleem. De oplossing is altijd een interactie tussen het ontworpen artefact en de probleem context.

Design validation

Nu er een artefact ontworpen is moet deze gevalideerd worden. Dit doe je door het ontworpen artefact in de context te plaatsen en te kijken of de effecten daarvan het benoemde probleem oplossen. Deze validatie kan op verschillende manieren uitvoeren:

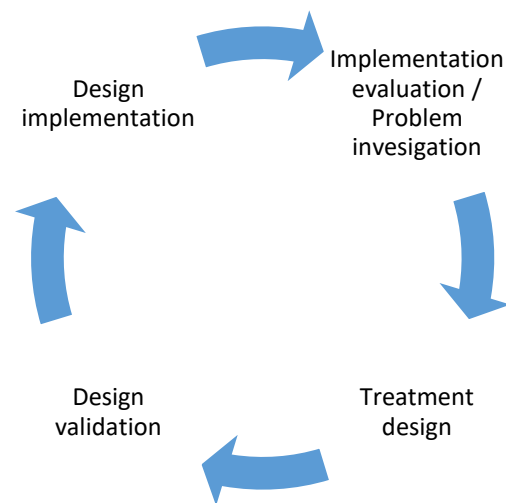
- Experts opinion (bijv. focus groepen, expertinterviews etc.)
- Simulatie
- Field experiment

Design implementation

Zodra tot een artefact gekomen is die het probleem oplost, zal dit geïmplementeerd moeten worden in de "real world" om het probleem daadwerkelijk op te lossen.

Implementation evaluation

Zodra de implementatie voltooid is, kan de cyclus opnieuw beginnen om te kijken of het probleem nog bestaat of om te onderzoeken of er nog andere gerelateerde problemen zijn om op te lossen.



In dit verslag zal vooral gefocust worden op de fases “treatment design” en “design validation”. De problem investigation fase was al reeds een onderdeel van het overkoepelende IMPULS-project en was als basis voor dit ontwerp een gegeven. Op basis van dit gegeven is een ontwerp vraag opgesteld en de benodigde kennis vragen om de ontwerp vraag te kunnen beantwoorden.

Ontwerp vraag

Maak een informatica lesmodule gerelateerd aan sociale robotica voor het Impuls project gebaseerd op het reeds gemaakte ontwerp voorstel.

Kennisvragen

1. Welke opzet en literatuur is er beschikbaar vanuit IMPULS
2. Welk lesmateriaal bestaat er al op het gebied van sociale robotica
3. Welke leerdoelen moet de module behalen en hoe verhouden deze zich tot het eindexamenprogramma.
4. Wat is de doelgroep (leerlingen en docent) en wat is hun voorkennis.

Onderzoekfase (Problem investigation)

1. IMPULS

Eind 2015 is vanuit ELAN, de vakgroep docentontwikkeling van Universiteit Twente, het IMPULS-project gestart. Dit project heeft tot doel om lesmateriaal te ontwikkelen dat aansluit bij de hedendaagse wetenschapscontexten.

Hiervoor zijn voor 5 beta-vakken: Wiskunde; Natuurkunde; Scheikunde; Informatica en NLT, ontwikkelgroepen gestart voor het ontwerpen van nieuw onderwijsmateriaal voor vwo-klassen op de middelbare school dat samenhangt met 1 of meerdere vakgroepen die onderzoek doen in die richting.

Voor informatica zijn twee modules ontwikkeld: 'Sociale Robotica – Ouderen hulp in de zorg' en 'Sociale Robotica – Gezichtsuitdrukkingen'. Deze modules zijn inhoudelijk verbonden aan de vakgroep Human Machine Interaction, die onderzoek doet naar Sociale Robotica.

Hieronder staan enkele publicaties vanuit de vakgroep die relevant zijn voor dit ontwerp onderzoek:

- Karreman, D.E. (2016) Beyond R2D2 - The design of nonverbal interaction behavior optimized for robot-specific morphologies. PhD thesis, University of Twente. CTIT Ph.d-thesis series No. 16-404 ISBN 978-90-365-4184-8
- Pantic, M. and Evers, V. and Deisenroth, M and Merino, L. and Schuller, B. (2016) Social and Affective Robotics Tutorial. In: Proceedings of the 2016 ACM on Multimedia Conference (MM 2016), 15-19 Oct 2016, Amsterdam, The Netherlands. pp. 1477-1478. ACM. ISBN 978-1-4503-3603-1
- Vroon, J.H. and Englebienne, G. and Evers, V. (2016) Responsive Social Agents: Feedback-Sensitive Behavior Generation for Social Interactions. In: Proceedings of the 8th International Conference on Social Robotics, ICSR 2016, 01-03 Nov 2016, Kansas City, MO, USA. pp. 126-137. Lecture Notes in Computer Science 9979. Springer International Publishing. ISSN 0302-9743 ISBN 978-3-319-47437-3
- Sagonas, C. and Panagakis, Y. and Zafeiriou, S. and Pantic, M. (2016) Robust statistical frontalization of human and animal faces. International Journal of Computer Vision, 122 (2). pp. 270-291. ISSN 0920-5691 *** ISI Impact 4,270 ***
- Cameron, D. and Fernando, S. and Collins, E. and Millings, A. and Moore, R. and Sharkey, A. and Evers, V. and Prescott, T. (2015) Presence of Life-Like Robot Expressions Influences Children's Enjoyment of Human-Robot Interactions in the Field. In: Proceedings of Fourth International Symposium on "New Frontiers in Human-Robot Interaction", 21-22 April 2015, Canterbury, UK. 3. The Society for the Study of Artificial Intelligence and Simulation of Behaviour (AISB). ISBN not assigned
- Joosse, M.P. and Knuppe, R.A. and Pinggen, G.L.J. and Varkevisser, R.A. and Vukoja, J. and Lohse, M. and Evers, V. (2015) Robots Guiding Small Groups: The Effect of Appearance Change on the User Experience. In: Proceedings of the 4th International Symposium on New Frontiers in Human-Robot Interaction, 21-22 April 2015, Canterbury, U.K.. 6. University of Kent. ISBN not assigned

2. Bestaand lesmateriaal

Robots zijn in fabrieken al enige tijd gemeengoed, maar langzaam komen ze ook bij de “gewone mens” terecht. Sociale robotica richt zich op alles wat te maken heeft met het feit dat robots zich in je dagelijkse omgeving bevinden. Hoe moeten de robots gemaakt worden dat mensen er mee om kunnen gaan, en dat de robots met de mensen om kunnen gaan. Dit onderwerp bevat naast de technische vaardigheden ook veel soft skills zoals psychologische aspecten of juist ethische bezwaren.

Omdat sociale robotica steeds meer voorkomt in het dagelijkse leven, is het belangrijk dat leerlingen hiermee bekend worden maar ook nadenken over wat er allemaal bij komt kijken om het mogelijk te maken in een snel veranderende wereld. Iedereen ziet dat alles om zich heen steeds slimmer wordt, maar heeft niet direct door wat voor impact dit heeft.

Het (voor het publiek) bekendste onderzoek hiernaar wordt gedaan door Vanessa Evers (Evers, 2017), hoogleraar op Universiteit Twente. Er is veel materiaal beschikbaar waarin ze toelicht wat sociale robotica is en waarom het belangrijk is. (bijv. <https://www.youtube.com/watch?v=QZV8claK0Hk>)

Over de technische kant van robotica is verschillend lesmateriaal te vinden:

- <http://probotics.eu/onderwijs>
- <http://www.robocupjunior.nl/download-document/124-nlt-robotica-ev3/nxt-2015-deel-1-lesmateriaal.html>
- <http://www.legoleerlijn.nl/mindstorms/>

Echter is er weinig lesmateriaal beschikbaar over het inzetten van sociale robots, die je in het dagelijkse leven tegenkomt, zoals: een stofzuig robot; robots in de zorg of robots in het onderwijs.

3. Leerdoelen¹

De leerdoelen zijn opgesteld zodat ze aansluiten op het eerste module voorstel van Dirk-Jan van de Poppe en Jurre Moor (Bijlage A - Opzet lesmodule – Dirk-Jan van de Poppe en Jurre Moor). In het ontwerpvoorstel wordt de focus gelegd op twee domeinen: modelleren vaardigheden en programmeren vaardigheden.

Naast deze twee domeinen, is er een derde domein bij gekomen die concreter ingaat op het sociale deel van de robotica.

Hieronder zijn alle hoofd leerdoelen die in de module behaald moeten worden uiteengezet. In de bijlage (Bijlage B – Leerdoelen matrix) is een matrix te vinden van alle leerdoelen en in welke lessen deze behandeld worden.

Modelleren vaardigheden

- De leerling moet zelfstandig een eenvoudig software model kunnen opstellen met enkele parameters

Programmeren vaardigheden

- De leerling moet een algoritme kunnen uitleggen voor sentiment bepaling.
- De leerling moet kunnen uitleggen wat een webservice is
- De leerling moet documentatie kunnen raadplegen om tot een oplossing te komen.
- De leerling moet json documenten begrijpen en kunnen lezen
- *De leerling moet de volgende programmeren concepten kunnen toepassen:*
 - *Arrays/Collections/Lists*
 - *Code becommentariëren*
 - *Bestaande functies aanroepen*
 - *Zelf code in functies opsplitsen*
 - *Het kunnen definiëren en gebruiken van variabelen en types (String, Int, Double)*

Ethische vaardigheden

- De leerling moet enkele ethische gevaren en risico's van robotica kunnen herkennen en aangeven.

Tevens is deze module ook uiteengezet tegen het examenprogramma in bijlage C (concept advies examenprogramma) en bijlage D (huidige examenprogramma). Bij het huidige en het concept examenprogramma zie je dat de les module verschillende (sub)domeinen behandelt, maar voornamelijk aansluit bij de ontwerp en programmeren vaardigheden. Hierdoor is de module goed in te zetten naast of ter vervanging van de huidige stof die deze onderwerpen behandelt, of inzetbaar als een keuze module binnen het vak.

¹ De leerdoelen zijn opgesteld volgens de SMART methodiek

4. Doelgroep en voorkennis

De module gaat gegeven worden aan 5VWO bij het schoolvak informatica en de module wordt gegeven door de informatica docent van de school. De module moet dus aansluiten bij de voorkennis van de leerlingen maar ook van de docent.

Binnen informatica onderwijs heerst een grote achterstand in de inhoud van het vak en de scholing van de docenten die het vak geven volgens SLO (SLO, 2014). Het huidige examenprogramma is erg vrij, wat de docent de mogelijkheid biedt om onderdelen te behandelen waar de docent sterk in is. Dit betekent echter dat onderwerpen waar de docent minder bedreven in is niet of minder aan bod komen.

Het verschil in voorkennis van de leerlingen en de docent die de module gaan gebruiken, welke kan verschillen per school of zelfs per docent binnen de school, maakt het lastig om een breed inzetbare module te ontwerpen die precies aansluit bij iedereen binnen de doelgroep.

De module wordt ontworpen voor een 5VWO klas. Op basis van het examenprogramma kan worden aangekomen dat de leerlingen reeds basis programmeerkennis hebben en dat hierop voort gebouwd kan worden. Echter welke programmeer concepten er daadwerkelijk besproken zijn is daarbij weer geheel afhankelijk van de docent.

De pilot wordt uitgevoerd halverwege het 5^e jaar op het VWO. De docent van de pilot school is het meest bedreven in Visual Basic for Applications (VBA) en de leerlingen hebben dus ook in deze taal eerder programma's ontwikkeld. De basis programmeer concepten zijn uitgelegd, maar waarschijnlijk al wel weer weggezakt doordat de afgelopen periodes andere onderwerpen aan bod zijn geweest.

Ontwerpfase (Treatment design)

Module opbouw

De module bestaat uit twee onderdelen, soft skills en programmeer skills, die samen 18 lessen van 45 minuten vullen en er vanuit gaan dat de leerlingen ook als huiswerk opdrachten af maken. De programmeer lessen zijn hieronder in lichtblauw aangegeven, de softskills lessen zijn geel. Afhankelijk van het niveau en de voorkennis van de leerlingen kunnen lessen weggelaten of toegevoegd worden.

Les	Onderwerp
1	Inleiding Robotica
2	Gezichtsmodel definiëren
3	Model visualiseren (paper prototyping)
4 -5	Herhalen programmeer skills
6	<i>“Wat is de uitdrukking”</i> en sentiment parameter
7 - 9	Gezicht model programmeren
10 - 11	Sentiment bepaling
12	Sentiment en gezicht combineren
13	Webservices (Twitter) Inleiding
14 - 15	Webservices (Twitter) implementeren
16 - 17	Afronding
18	Theorie toets + inleveren verslag

Het einddoel van de module is dat de leerlingen een “robot” gezicht hebben, waarvan de gezichtsuitdrukking veranderd op basis van de sentiment van een twitter feed.

De module begint met een inleiding in de robotica, om de leerlingen bekend te laten worden met het concept en de opdracht die bij de module hoort en waarom dit nuttig is. Zodra de leerlingen dit weten kunnen ze bezig met het opstellen van een model voor het weergeven van gezichtsuitdrukkingen van hun “robot”. Dit model moeten ze zelf maken en testen aan de hand van paper prototyping.

Vervolgens wordt tussendoor de programmeerkennis van de leerlingen opgehaald, en door middel van een sturende opdracht worden alle concepten die aan het einde nodig zijn herhaald zodat ze deze zelf later kunnen toepassen.

Door de lessen met programmeerherhaling is er ruimte voor de docent om aan de hand van de door de leerlingen gemaakte modellen een quiz te maken waarin de andere leerlingen de gezichtsuitdrukkingen gaan raden van de modellen. Met de feedback die de leerlingen hierop krijgen kunnen ze hun modellen nog verbeteren en/of aanpassen.

Nu er een gezichtsmodel is, moet beter worden nagedacht over een hoofd-parameter die verschillende gezichtsuitdrukkingen kan weergeven. Dit is belangrijk omdat in de volgende lessen de leerlingen hun gezicht model gaan programmeren op basis van deze parameter.

Zodra de leerlingen een gezicht hebben welke aanpast a.d.h.v. een sentiment parameter, wordt de volgende stap genomen om het sentiment van een zin te bepalen, welke vervolgens weer het gezicht kan aanpassen.

Als laatste stap zal een introductie in webservices gegeven worden, zodat de leerlingen (met hulp) een koppeling kunnen maken met witter welke weer aansluit op hun eerdere sentiment analyse. Zodra alle onderdelen samengevoegd zijn zou een werkend programma moeten zijn ontstaan.

De module wordt afgerond door het inleveren van de code en een verslag, en een schriftelijke toets.

Ontwerpbeslissingen

In dit hoofdstuk wordt van verschillende onderdelen van de module toegelicht waarom ze op deze manier zijn opgezet.

Werkvormen

De module bestaat voornamelijk uit klassikale of zelfstandige werkzaamheden, maar niet uit groepswork. In de basis moet elke leerling elke opdracht zelf uitgevoerd hebben, om daadwerkelijk de leerdoelen te behalen. Dit onder het motto “Het lijkt zo logisch als je het antwoord ziet, maar het is moeilijk om zelf te bedenken”.

Programmeertaal

De lessenserie is zo opgezet dat deze geschikt is voor meerdere programmeertalen, echter om de lessen concreet te maken en de pilot te kunnen uitvoeren is gekozen om de opdrachten uit te werken in Visual Basic for Applications.

De keuze voor VBA is genomen omdat de leerlingen en docent op de school van de pilot er reeds ervaring mee hadden, maar ook omdat deze goed beschikbaar is op de vaak beperkte schoolnetwerken. Het voordeel van VBA, dat onderdeel is van de office for Windows suite, is tevens het nadeel. Dit betekent namelijk dat leerlingen zonder Microsoft office op Windows, niet in staat zijn om hier thuis voor te ontwikkelen.

Les duur

Deze module bestaat uit 18 lessen van elk 45 minuten, voor deze opbouw is gekozen omdat tijdens de pilot drie lessen van 45 minuten per week gegeven werden. Waardoor in het geval van de pilot de module 6 weken heeft geduurd.

Afhankelijk van de voorkennis van de leerlingen kunnen lessen samengenomen of juist gesplitst worden.

Leerlingen materiaal

Deze handleiding heeft een leerlingen hoofdstuk, met materiaal voor de leerlingen. Dit leerlingen materiaal dient ter aanvulling en ondersteuning van de lessen. Echter is klassikale uitleg benodigd om de opdrachten te maken, anders zullen de leerlingen niet genoeg kennis en informatie hebben. Deze keuze is gemaakt zodat de leerlingen wel een leidraad en overzicht hebben van opdrachten en wat er moet gebeuren, maar de regie van de lessen blijft bij de docent liggen. Daardoor is ook door de docent makkelijker in te spelen op onderwerpen welke hij/zij belangrijk vindt of al eerder behandeld heeft.

Didactiek

Les en module opbouw

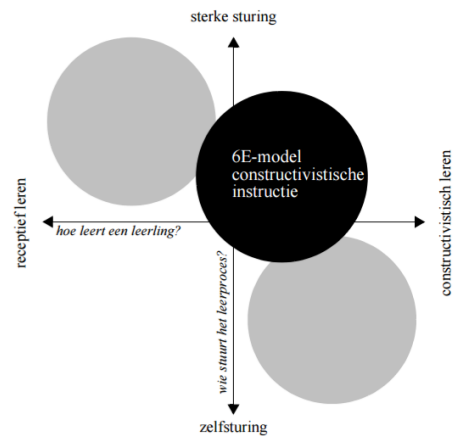
De module en de individuele lessen zijn opgebouwd volgens het 6^e-model (Windels, 2011).

Dit instructiemodel richt zicht op een constructivistische les met een sterkere sturing van de docent. Dit wordt ondersteund door 6 stappen die in 1 of meerdere keren uitgevoerd worden gedurende een les.

Er is voor dit model gekozen, omdat het informatica onderwijs erg op het wiskundeonderwijs lijkt (Zwaneveld, 2005). Dezelfde vaardigheden als modelleren, probleem oplossen en toepassen worden gedoceerd.

In het 6E model zijn de volgende stappen de basis:

- E1. Evalueren
- E2. Engageren
- E3. Exploreren
- E4. Empirische kennis toepassen
- E5. Expliciteren
- E6. Eindcompetenties consolideren



Figuur 1 Bron: Nieuwe Wiskrant 32-2/december 2012

Het 6E model binnen de lessen

De lessen binnen de module volgen zo veel mogelijk het 6E model, om de leerlingen nieuwe kennis bij te brengen. Ter illustratie zijn hieronder twee verschillende lessen uiteengezet tegen het 6E instructie model. Hierbij zijn sommige stappen herhaald voor verschillende onderdelen van de les, of juist samengenomen omdat 1 activiteit meerdere stappen gelijktijdig vervuld. De leerdoelen en exacte invullingen van de lessen is te vinden in het docenten materiaal.

6E instructie model in les 1

Les onderdeel	Stap	Toelichting
Zelf uitzoeken wat robotica is	E1	In dit onderdeel gaan de leerlingen zelf nadenken wat (sociale) robotica is en wat ze er al over weten
Video's over robotica	E2	D.m.v. de video's krijgen leerlingen het nut en de toepassingen te zien
Ethische dilemma's intro	E3	A.d.h.v. concrete vragen op basis van voorbeelden wordt de leerlingen duidelijk gemaakt wat ze er al van weten en kunnen
Ethische dilemma's discussie	E4 – E5	De leerlingen denken zelf na over ethische dilemma's en naarmate discussie wordt concreet
Gezichtsuitdrukkingen	E3	Voorbeeld a.d.h.v. video
	E4	Zelf nadenken over voorbeelden
	E5	Goede voorbeelden bespreken als antwoord
LES 2	E6	In de volgende les gaan ze de kennis toepassen

6E instructie model in les 13 (en 14/15)

Dit is één van de latere lessen waar nieuwe kennis word behandeld, namelijk webservices. In deze les zijn meer stappen samengenomen binnen 1 activiteit. Dit komt onder andere doordat de leerlingen zelfstandiger werken en doordat het om grotere hoeveelheden nieuwe kennis gaat.

Les onderdeel	Stap	Toelichting
Introductie webservices	E1 – E2	In dit onderdeel wordt uitgelegd wat webservices zijn en wordt dit gerelateerd naar wat dicht bij ze staat (Social media)
Opdracht Twitter	E3 – E4	D.m.v. een zelfstandig te volgen opdracht wordt eerst kennis van een webservice getest met simpele opdrachten om vervolgens dit zelf toe te passen op de API van twitter.
Bespreken twitter opdracht	E5	Tijdens de bespreking worden de concepten losgetrokken zodat de kennis breder inzetbaar is.
Les 14/15 -> Opdracht	E6	Tijdens de volgende opdracht, moet de opgedane kennis in een andere context worden toegepast.

6E instructie model over de module

Het 6E instructie model is gericht op een instructie model binnen een les, met stappen van 1 tot 10 minuten. Maar over de gehele module is het concept ook toegepast, waarbij in het begin van de module leerlingen concrete voorbeelden krijgen, en ze naarmate de module vordert deze concepten en kennis moeten toepassen in hun eigen context.

Opdrachten

De opdrachten zijn gemaakt om na een klassikale introductie zelfstandig door de leerlingen uit te voeren. De bedoeling is ook dat iedereen de opdrachten zelf uitvoert (behalve als dit anders is aangegeven bij de opdracht), dit omdat binnen programmeer onderwijs het erg makkelijk is om iemand anders het te zien doen en het dan te begrijpen. Maar de lastigheid ligt er om het zelf te kunnen doen.

Wel kunnen de leerlingen overleggen met elkaar, zolang ze allemaal maar zelf de opdrachten maken, en dus ook kunnen toelichten.

Evaluatiefase (Design validation / Design implementation)

Pilot

De evaluatie heeft plaatsgevonden in de vorm van een pilot: De module is uitgetest op het Stedelijk Lyceum locatie Kottenpark onder begeleiding van dhr. van de Poppe. De groep was een 5vwo klas, van 22 leerlingen die drie keer per week een les van 45 minuten hadden. De leerlingen hadden eerder reeds geprogrammeerd in Python en VBA, mede door de vaardigheden van de docent is daarom voor VBA als leidende programmeertaal gekozen.

Een globale opbouw van de module was klaar en uitgewerkte lessen zijn ontworpen tijdens de pilot fase, hierdoor was het mogelijk om nieuwe inzichten van de eerste lessen gelijk te verwerken in het materiaal van de lessen later in de module.

Feedback is verworven op verschillende manieren:

- Expert opinion, waarbij de docent en een TOA feedback geven op het materiaal en de lessen.
- Field experiment (de pilot), waarbij gekeken is naar de vragen die de leerlingen stelden tijdens de opdrachten; de kwaliteit van de opdrachten die ze maakten; de feedback die de leerlingen gegeven hebben over de module en het onderwerp; en een schriftelijke toets

Expert opinion

Gedurende de module is het materiaal en de lessen bekeken door de docent en de TOA. De experts hebben het materiaal beoordeeld door het te lezen en de opdrachten uit te voeren, maar ook door de reacties van de leerlingen te bekijken. De gekregen feedback had vooral betrekking op de voorkennis van de leerlingen, in het materiaal werden aannames gedaan over voorkennis of te snel door bepaalde stappen heen gegaan. De conclusie hieruit was dat op sommige punten het materiaal te snel ging of het niet geheel duidelijk was voor de leerlingen wat er van ze verwacht werd. Aan de andere kant was de mening van de experts erg positief over het feit dat het materiaal de leerlingen een abstracte denkwijze probeert aan te leren, door het uitleggen van programmeer concepten in plaats van een “voor doen en namaken”-opdrachten.

Field Experiment

Vragen v.d. leerlingen

Tijdens de pilot is er gekeken welke vragen de leerlingen tijdens de les stelden. Deze vragen waren in twee categorieën op te delen: 1. Onduidelijk en/of te moeilijk materiaal en 2. Een andere manier van lesmethode die de leerlingen niet gewend waren.

De vragen van punt 1 kwamen overeen met de vergaarde informatie van de expert opinion, waar ook bleek dat soms uit was gegaan van een te hoge voorkennis. Ook bleek hier dat sommige vragen ambigu opgevat konden worden, waardoor de leerlingen met een verkeerd antwoord verder werkten.

Ook waren er veel vragen over de gebruikte programmeertaal (VBA). Die niet direct in verband stonden met de lesstof maar die wel het verder werken belemmerden. Zoals vreemde foutmeldingen door ontbrekende haakjes en/of puntkomma's. Hierdoor kregen de leerlingen het gevoel dat ze de opdrachten niet begrepen, terwijl de problemen vooral lagen bij rare/gekke fouten die horen bij VBA.

Daarnaast (punt 2) kwamen er veel vragen dat de leerlingen niet wisten wat ze moesten doen, of dat ze vooral al veronderstelden dat ze het niet begrepen. Op veel van deze vragen was als antwoord: “Heb je de opdracht wel goed gelezen” of “Heb je al op het onderwerp gezocht op internet” vaak voldoende. De leerlingen waren niet gewend dat in/bij de vragen vaak een hint staat naar hoe ze het

probleem moesten oplossen in plaats van een directe oplossing. Een hint kan bijvoorbeeld zijn een link naar API-documentatie van een klasse/methode die je nodig hebt.

Kwaliteit v.d. ingeleverde opdrachten

De leerlingen moesten tussendoor opdrachten inleveren, op basis van dit ingeleverde week is gekeken of dit overeenkwam met de verwachtingen. Bij de eerste ingeleverde opdrachten bleek dat sommige leerlingen de opdracht anders geïnterpreteerd hadden dan de bedoeling was. Dit samengenomen met de geleerde punten van de vragen van de leerlingen bleek dat de opdrachten niet strikt genoeg beschreven waren en dat er te veel ruimte was voor interpretatie. In de loop van de module zijn daarom de opdrachten ook aangescherpt en hier was ook duidelijk verbetering te zien op het ingeleverde werk.

Feedback v.d. leerlingen

Gedurende de module is op een informele manier feedback verkregen van de leerlingen, door tijdens het zelfstandig werken en na de les met de leerlingen te praten. In deze gesprekken kwamen veel punten naar boven die de eerdere bevindingen staafden: De opdrachten waren anders geformuleerd dan ze gewend waren, waardoor het in het begin zeker onduidelijk was; de leerlingen vonden VBA een onduidelijke programmeertaal en vonden al dat ze hier niet zo goed mee overweg konden; tevens werd aangegeven dat ze soms de makkelijke weg kozen, gelijk om hulp vragen, in plaats van zelf eerst uitzoeken hoe het werkte. De leerlingen die goed meededen hadden wel het gevoel dat ze echt wat geleerd hadden in de module; en dat ze door de opdrachten ook verder konden werken als ze het al af hadden.

Schriftelijke toets

Aan het einde van de module is een schriftelijke toets geweest, dit om te bepalen of de leerlingen de leerdoelen van de module bereikt hebben. Uit de resultaten van de toets werd de indruk bevestigd dat leerlingen die meegedaan hadden tijdens de lessen en een goed eindproduct hebben opgeleverd ook hoog scoorden op de schriftelijke toets. De leerlingen die vak niet in de le aanwezig of bezig waren met andere dingen hebben een onvoldoende op de toets gehaald. Dit bevestigt dat, indien de leerling mee doet, hij aan het eind de leerdoelen heeft bekijkt van de module. Maar ook dat de toets niet te makkelijk was, omdat er daadwerkelijk kennis opgedaan moest zijn om de toets te halen.

Bijstellen materiaal n.a.v. Pilot (Implementation evaluation)

Naar aanleiding van de evaluatie is het materiaal bijgesteld. Hieronder zijn de wijzigingen binnen het materiaal en attentie punten voor bij het geven van de module opgesteld.

Verwerkte leerpunten

1. Tijdens de pilot bestond het leerlingen deel niet uit één hoofdstuk, maar uit los aangeleverde opdrachten. Hierdoor was het niet altijd duidelijk voor de leerlingen aan welke opdrachten ze moesten werken. Om dit probleem op te lossen is het leerlingen materiaal beter gestructureerd en samengebonden voor de leerlingen.
2. In het begin tijdens de pilot was het een groepsopdracht, echter doordat niet iedereen zelf aan de slag ging maar het werk gedaan werd door de “sterkere” leerlingen, is het aangepast naar een individuele opdracht
3. Voor het gebruik van de Twitter API, zijn OAuth tokens nodig, echter voor het aanmaken van deze tokens hebben de leerlingen een Twitteraccount met gekoppeld telefoonnummer nodig. Aangezien je niet van alle leerlingen kan verwachten dat ze dit maken, zijn aan de opdracht tokens toegevoegd die de docent kan aanmaken.
4. Bij het vak informatica is vaak een groot niveauverschil merkbaar, om dit op te vangen zijn er aan de opdracht enkele bonus opdrachten toegevoegd die de sterke leerlingen extra uitdaging kunnen geven.
5. Gedurende de opdrachten van de mode-robot en sentiment analyse bleek dat sommige vragen te ruim of niet duidelijk waren gesteld. Een voorbeeld hiervan is vraag 8 van de module robot, dit was *“Maak een 2e knop, die de functionaliteit krijgt dat elk onderdeel een andere kleur krijgt”*, nu is hier extra toelichting bij gegeven hoe de leerlingen dit moeten doen.

Attentie punten

1. Het is belangrijk dat de leerlingen goed lezen, tijdens de pilot kwamen veel vragen waarvan de antwoorden in de opdracht stonden. Dit kon zijn in de vorm van vraagstellen maar soms staat er ook hints of links naar documentatie in de opdrachten.
2. Enkele leerlingen hadden thuis geen Microsoft Office, het is handig om van tevoren duidelijk te maken wat de leerlingen nodig hebben om thuis te kunnen werken of wanneer ze er op school aan kunnen werken.
3. Tijdens de pilot bleek het nodig om tussendoor de leerlingen ook opdrachten te laten inleveren en dit te becijferen. Dit zodat de minder serieuze leerlingen extrinsiek “gemotiveerd” werden om aan de slag te gaan.

Conclusie & Discussie

Op basis van de pilot en de evaluatie kan geconcludeerd worden dat in de basis er een module staat die voldoet aan het ontwerpvoorstel voor het Impuls project. Het introduceert verschillende concepten met betrekking tot sociale robotica gezichtsuitdrukkingen en sentiment analyse maar ook de ethische problemen met robotica. Maar het sluit ook aan bij het huidige en nieuwe concept examenprogramma.

Uit de evaluatie punten in het vorige hoofdstuk komt dat er nog verbeterpunten zijn voor de module, door het doorvoeren van deze punten kan de module inhoudelijk nog verbeterd worden.

Ook is er een discussie punt met betrekking tot de programmeertaal die het beste gebruikt kan worden voor deze module. De module is opgezet onafhankelijk van de programmeertaal, echter de opdrachten richten zich nu op VBA. Echter levert deze taal enkele problemen op zoals: Je hebt er office voor nodig; de taal levert geen duidelijke foutmeldingen; de taal heeft enkele vage syntactische "eigenaardigheden". Waarschijnlijk zijn talen zoals Java, c# of python / javascript met HTML een geschiktere keuze om deze module te onderwijzen.

Daar hangt aan vast dat de voorkennis van de docent een belangrijke voorwaarde is voor het geven van de module. De module bevat veel leidende opdrachten voor de leerlingen, maar een deel van de kennis zal door de docent tijdens klassikale uitleg overgedragen moeten worden. Daarnaast is bij een programmeer opdracht die op meerdere manieren opgevat kan worden vaak geen eenduidige lijst met problemen die kunnen voorkomen, het is belangrijk dat de docent voldoende boven de stof staat om de leerlingen te kunnen helpen met deze problemen.

Verwijzingen

Evers, V. (2017). Opgehaald van <https://vanessaevers.wordpress.com/about/>

SLO. (2014). *Informatica in de bovenbouw van havo/VWO*. Opgehaald van <http://www.slo.nl/downloads/2014/informatica-in-de-bovenbouw-havo-vwo.pdf/download>

Wieringa, R. (2010). Design science methodology: principles and practice. . *2nd ACM/IEEE International Conference on Software Engineering-Volume 2*, 493-494.

Windels, B. (2011). Het 6E-model: een compromis tussen positivistisch en constructivistisch wiskundeonderwijs? *Tijdschrift voor lerarenopleiders VELON-VELOV* 32(3), 17-24.

Zwaneveld, G. &. (2005). *Wiskunde en informatica: innovatie en consolidatie: over vragen in het wiskunde-en informaticaonderwijs*. . Open Universiteit Nederland.

Bijlage A - Opzet lesmodule – Dirk-Jan van de Poppe en Jurre Moor

De voorzet van deze lesmodule is gekomen van Dirk-Jan van de Poppe en Jurre Moor, hieronder vind je het document waar deze module uit voort is gekomen.

Lesmodule Sociale robotica, emotieherkenning en beslisboom

Jurre Moor en Dirk Jan van de Poppe

Oktober 2016

Doel: een practicum opdracht uitvoeren waarin de computer een vorm van emotie moet kunnen simuleren. Daarbij moeten de leerlingen nadenken hoe emoties zichtbaar worden en in eerste instantie op een monitor een emotie uitdrukken. De volgende stap is dat de computer een ingetypte vraag analyseert, en daarbij een emotie toont. Dit moet inpasbaar zijn in het curriculum van 5 vwo/ 6 vwo. Adequaat en van voldoende niveau. Het moet bruikbaar zijn op meerdere scholen, zodat het systeemafhankelijk moet kunnen worden uitgevoerd.

Onderwerpen die aan de orde komen:

- Het simuleren van een gezicht met parameters voor bepaalde emotietoestanden.
- Opzetten onderzoek:
 - Welke grootheden bepalen de gezichtsuitdrukking?
 - Het maken van een fotoreeks of tekening en het samenstellen van een hoofd.
 - Het opzetten van een algoritme om tendensen te bepalen.
 - Het beantwoorden op een ingetypte zin met bijbehorende emotie.
- Ontwerpcyclus.

Context:

Het vertalen van het onderzoek van Bob Schadenborg, Enigma, *robots bij autistische leerlingen* naar middelbare schoolniveau, met name 5 vwo, met uitloop naar 6 vwo.

Probleemstelling:

Het volgende scenario beschrijft de context van deze opdracht:

Een robot gebruik je als huishoudelijke hulp. Dan zou van kunstmatige intelligentie sprake zijn wanneer de robot je begroette met: "wat ben je vrolijk vandaag" of na het zeggen van een antwoord waarbij de gebruiker een verbaast gezicht opzet, de vraag stelt: "zeg ik iets verkeeds?" De robot zou blij moeten reageren op een compliment, en verdrietig op een standje. Het antwoord van een robot moet afhangen van de gemoedstoestand die hij bij de gebruiker detecteert.

Opdracht:

De opdracht bestaat uit vier deelopdrachten: een *simulatieprogramma*, een *gezicht-samenstelling programma*, een *lerend programma*, een *beslisboom waarbij de geschiedenis bijgehouden wordt en keuzen aangepast aan de hand van de geschiedenis* en een *koppelprogramma*. Eventueel kan, als de beschikbare tijd minder is dan 8 weken, een deel van de deelopdrachten worden uitgevoerd of als de opdrachten meer tijd kosten, de deelopdrachten parallel worden ontwikkeld.

1. Simulatieprogramma (2 weken):

Stap 1: bedenk parameters die emotie uitdrukken. Te denken valt aan mondhoeken, positie wenkbrauwen en verhouding lengte breedte van de oogleden. Maak een programma die een gezicht simuleert met de bijbehorende parameters.

2. Gezicht-samenstelling-programma: (2 weken)

Maak foto's van proefpersonen met verschillende gezichtsuitdrukkingen bij verschillende emoties. Geef parameters daaraan en vergelijk de foto's met het simulatieprogramma.

3. Lerend programma: (2 weken)

Werk vanuit de basis met tien vragen. Beantwoordt de vragen op verschillende manieren, afhankelijk van de gemoedstoestand. Laat het programma feedback vragen en neem betere alternatieven op bij een grotere groep gebruikers. Gooi de wat mindere antwoorden niet weg, maar laat met een kansverdeling de wat mindere antwoorden terugkomen.

4. Koppelprogramma: (2 weken)

- Combineer het lerend programma met de gezichtsuitdrukking. Het is de bedoeling om de robot emotie te laten uiten. Maak een algoritme waarbij ook de vorige vragen een rol spelen in het gezicht dat de robot toont.

Evalueer na afloop het product. Wat zijn de sterke en minder sterke punten van jouw systeem?

Uitgangspunt/achtergrond:

Vanessa Evers (http://hmi.ewi.utwente.nl/Member/vanessa_evers/CV)

Enigma project

Autism Spectrum Conditions (ASC, frequently defined as ASD - Autism Spectrum Disorders) are neurodevelopmental conditions, characterized by social communication difficulties and restricted and repetitive behavior patterns. There are over 5 million people with autism in Europe – around 1 in every 100 people, affecting lives of over 20 million people each day. Alongside their difficulties, individuals with ASC tend to have intact and sometimes superior abilities to comprehend and manipulate closed, rule-based, predictable systems, such as robot based technology. Over the last couple of years, this has led to several attempts to teach emotion recognition and expression to individuals with ASC, using humanoid robots. This has been shown to be very effective as an integral part of the psycho-educational therapy for children with ASC.

The main reason for this is that humanoid robots are perceived by children with autism as being more predictable, less complicated, less threatening, and more comfortable to communicate with than humans, with all their complex and frightening subtleties and nuances. The proposed project aims to create and evaluate the effectiveness of such a robot-based technology, directed for children with ASC. This technology will enable to realize robust, context-sensitive (such as user- and culture-

specific), multimodal including facial, bodily, vocal and verbal cues) and naturalistic human-robot interaction (HRI) aimed at enhancing the social imagination skills of children with autism. The proposed will include the design of effective and user-adaptable robot behaviors for the target user group, leading to more personalized and effective therapies than previously realized. Carers will be offered their own supportive environment, including professional information, reports of child's progress and use of the system and forums for parents and therapists.

Info vanuit Delft over enigma : <http://www.robotblog.nl/category/educatie-robotica/>

Toepassing emotieherkenning: http://www.intermediair.nl/vakgebieden/onderwijs/robots-helpen-autistische-leerlingen?utm_referrer=https%3A%2F%2Fwww.google.nl%2F en <http://www.newscientist.nl/nieuws/vmbo-leerlingen-krijgen-robot-als-leraar/>

Bijlage B – Leerdoelen matrix

De leerdoelen voor deze module zijn in drie categorieën op te delen: programmeer vaardigheden, modelleer vaardigheden en ethische vraagstukken over robotica. Hieronder staat een matrix van elk leerdoel en in welke les deze behandeld wordt.

Les:	1	2	3	4,5	6	7,8,9	10,11,12	13,14,15
De leerling moet een eenvoudig software model kunnen opstellen met enkele parameters		X	X			X		
De leerling moet enkele ethische gevaren en risico's van robotica kunnen herkennen en aangeven.	X				X			
De leerling moet een algoritme kunnen uitleggen voor sentiment bepaling.							X	
De leerling moet kunnen uitleggen wat een webservice is								X
De leerling moet json documenten begrijpen en kunnen lezen								X
<i>De leerling moet de volgende programmeer concepten kunnen toepassen:</i>								
- Arrays/Collections/Lists				X		X	X	X
- Code becommentariëren								X
- Bestaande functies aanroepen				X		X	X	X
- Zelf code in functies opsplitsen				X		X	X	X
- Het kunnen definiëren en gebruiken van variabelen en types (String, Int, Double)				X		X	X	X
De leerling moet documentatie kunnen raadplegen om tot een oplossing te komen.				X		X	X	X

Bijlage C - Relevante exameneisen concept advies-examenprogramma²

In dit hoofdstuk is de les module tegen het concept examenprogramma van het SLO aangehouden. Hieronder zie je welke domeinen terugkomen in deze les module en hoe dit vormgegeven is.

Kern programma

Domein A – Vaardigheden

Binnen dit domein wordt onderscheid gemaakt tussen drie soorten vaardigheden: ontwerpen en ontwikkelen, informatica hanteren als perspectief, en samenwerken en interdisciplinariteit. Deze les module focus zicht vooral op de ontwerp en ontwikkelvaardigheden van de leerlingen. Binnen deze categorie zijn voor

- in een context mogelijkheden zien voor het inzetten van digitale artefacten;
- deze mogelijkheden vertalen tot een doelstelling voor ontwerp en ontwikkeling, daarbij technische factoren, omgevingsfactoren en menselijke factoren betrekken;
- een digitaal artefact ontwerpen;
- bij het ontwerp van een digitaal artefact keuzes afwegen via onderzoeken en experimenteren;
- een digitaal artefact implementeren;

Naast deze ontwerp en ontwikkel vaardigheden, komt ook “Informatica hanteren als perspectief “ en “Ethisch handelen “ aan bod als we het hebben over robotica in de praktijk.

Domein B - Grondslagen

Dit domein bestaat uit verschillende grondslagen, waarbij één hiervan relevantie heeft op deze module.

- een oplossingsrichting voor een probleem uitwerken tot een algoritme

deze vaardigheden worden ontwikkeld en getoetst bij het onderdeel taal analyse

Domein D – Programmeren

Programmeren is een belangrijk deel van informatica maar ook van deze module. Alle leerdoelen die voor programmeren gesteld worden zijn dan ook relevant voor dit les materiaal.

De kandidaat kan, voor een gegeven doelstelling,

- programmacomponenten ontwikkelen in een imperatieve programmeertaal;
- daarbij programmeertaalconstructies gebruiken die abstractie ondersteunen;
- programmacomponenten zodanig structureren dat ze door anderen gemakkelijk te begrijpen en te evalueren zijn.

De kandidaat kan

- structuur en werking van gegeven programmacomponenten uitleggen;
- zulke programmacomponenten aanpassen op basis van evaluatie of veranderde eisen

Domein E- Interactie

De module gaat over sociale robotica, waar interactie tussen mens en machine belangrijk is. Echter is dit niet het type interactie dat hier in het examenprogramma bedoeld wordt. De focus ligt hier vooral op usability, i.p.v. sociale robotica.

² <http://downloads.slo.nl/Repository/advies-examenprogramma-informatica-havo-vwo.pdf>

Keuze modules

Domein R - Computational science

De concepten van deze module komen uit wetenschappelijke hoek, namelijk de vakgroep Human-Machine-Interaction van Universiteit Twente. Deze concepten gaan de leerlingen concretiseren en een prototype van bouwen.

Bijlage D - Relevante exameneisen huidige examenprogramma³

Hieronder staan de onderdelen van het huidige examenprogramma die betrekking hebben tot deze module. Deze onderdelen worden één of meerdere malen behandeld gedurende het hele programma.

Subdomein A2:

Maatschappij 2. De kandidaat kan beschrijven wat de rol is van informatica en ICT bij maatschappelijke ontwikkelingen zowel in het verleden als nu.

Subdomein B3:

Software 7. De kandidaat beheerst eenvoudige datatypen, programmastructuren en programmeertechnieken.

Subdomein C6:

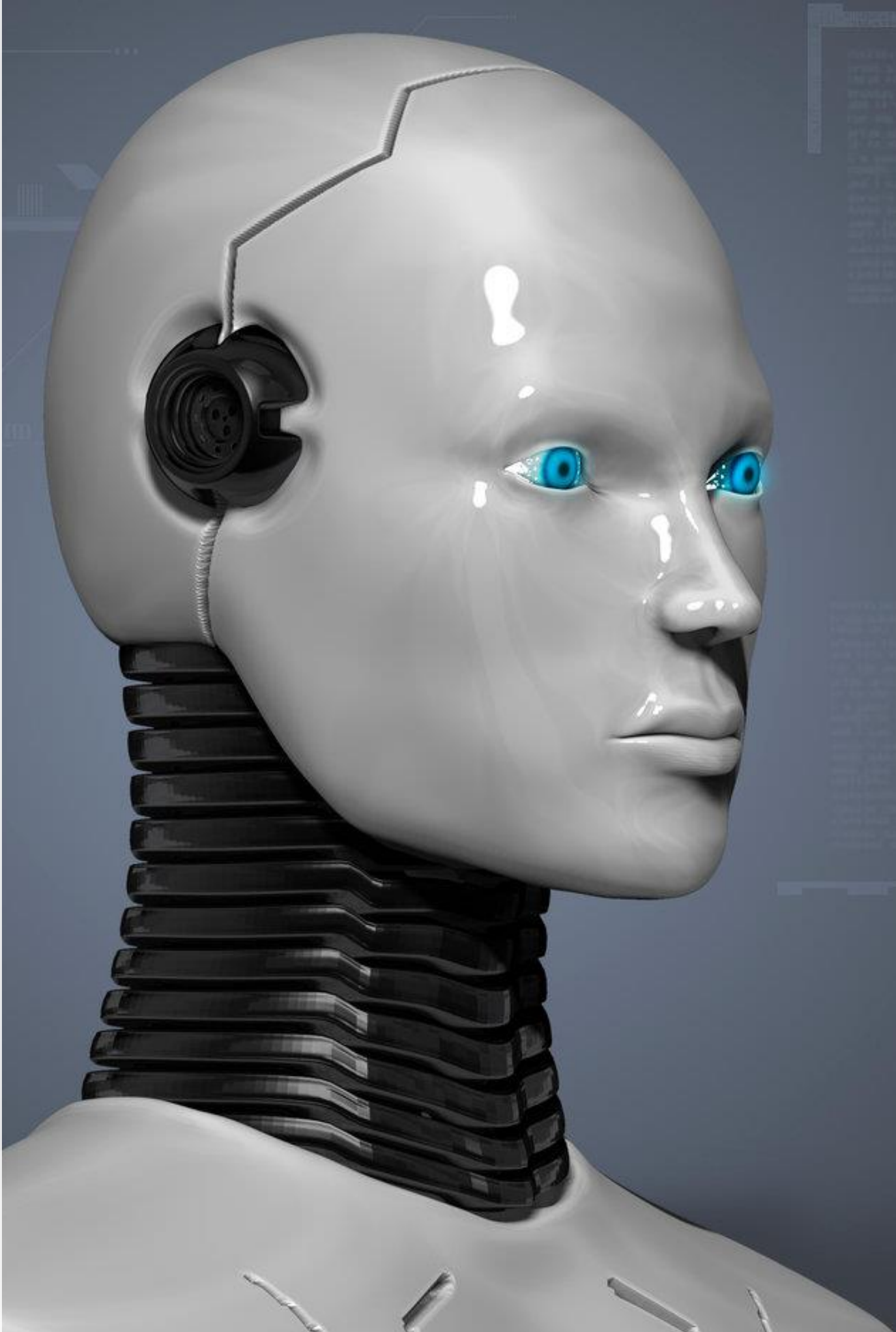
Informatieanalyse 14. De kandidaat kan informatie en informatiebehoeftes analyseren en het bijbehorende informatiemodel bouwen/aanpassen.

Subdomein C8:

Interactie mens-machine 16. De kandidaat kan mens-machine interactie bij informatiesystemen herkennen, de kenmerken ervan benoemen en keuzecriteria in het ontwerp van gebruikersdialogen benoemen en hanteren.

³ http://www.slo.nl/downloads/archief/Examenprogramma_informatica_DEFINITIEF.pdf/

Docenten handleiding



Lesmodule Sociale Robotica

Gezichtsuitdrukkingen

Docenten materiaal

Deze module is ontworpen in het kader van het IMPULS project. De module focust zich op de sociale kant van robotica en probeert leerlingen hiermee in aanraking te brengen. In dit document is de opbouw en lesmateriaal te vinden voor het geven van de module, en is theoretisch kader toegevoegd ter ondersteuning van het geheel.

Alle opdrachten zijn ontworpen met het oog op een 5vwo klas met een basiskennis in programmeren. De opdrachten zijn in het begin erg sturend, maar verder in de module en de opdrachten worden ze steeds ruimer.

De module is onder te verdelen in 2 aspecten: het technische en het sociaal ethische aspect. Beide worden behandeld en zullen bijdragen aan het eindresultaat. Het technische aspect van de module is gebaseerd op de programmeertaal VBA (Visual Basic for Applications), maar zou zonder veel problemen ook toegepast kunnen worden op andere talen.

Het einddoel van de module is dat de leerlingen een robot gezicht bouwen die reageert op het sentiment van Twitter berichten. Technische vaardigheden die hierbij aanbod komen zijn: modelleren, softwareontwikkeling en begrip van webservices.

In de rest van dit hoofdstuk is voor elke les uitgewerkt wat de inhoud van deze les kan zijn. De lessen zijn aangevuld met suggesties voor opdrachten en/of video's die de theorie kunnen verduidelijken. De lessen gaan uit van een duur van 45 minuten, waarbij de leerlingen die het werk niet af hebben dit thuis doen als huiswerk. Afhankelijk van de duur van de lessen en het niveau van de leerlingen is het natuurlijk altijd mogelijk om lessen in te korten of samen te voegen.

Bij het voorbereiden van de lessen, is het het beste om het leerlingen materiaal naar het docenten materiaal te leggen om een compleet beeld te krijgen.

Enkele lessen, bijvoorbeeld vier en vijf, zijn in de omschrijving hieronder en in het leerlingen materiaal samengenomen. De activiteiten hebben dan meerdere lessen nodig om alle opdrachten af te ronden. Het is mogelijk om de leerlingen zelfstandig deze gecombineerde lessen aan de slag te laten gaan, met als opdracht om aan het einde alles af te hebben. Ook kan je de leerlingen meer duidelijkheid geven door aan het einde van de les te vertellen wat die les af moet zijn.

De lessen zijn uitgetest op het Stedelijk Lyceum, locatie Kotten park in Enschede, in een 5vwo klas onder leiding van Dirk-Jan van de Poppe. De leerpunten en uitkomsten van deze try-out zijn meegekomen in het huidige lesmateriaal, en zijn verder toegelicht later in dit document.

Inhoudsopgave

Docenten materiaal.....	1
Inhoudsopgave.....	2
Leerlingen materiaal.....	3
Les 1 – Inleiding Robotica.....	4
Les 2 – Gezichtsmodel definiëren	6
Les 3 – Model visualisatie.....	7
Les 4 & 5 – Herhaling programmeer skills.....	8
Les 6 - “Wat is de uitdrukking” en sentiment parameter	9
Les 7, 8 & 9 – Model implementeren	10
Les 10 & 11 – Sentiment analyse	11
Les 12 – Gezicht en sentiment combineren	12
Les 13 – Webservice inleiding	13
Les 14 & 15 – Twitter koppeling bouwen.....	14
Les 16 en 17 - Afronding.....	15
Les 18 - Toets.....	16

Leerlingen materiaal

Deze module bevat een los onderdeel voor de leerlingen, dit materiaal is zo opgebouwd dat het voor de verschillende lessen de leerlingen een handvat en/of opdrachten geeft wat ze moeten doen. Dit materiaal is er niet voor gemaakt dat de leerlingen de gehele module zelfstandig kunnen doorwerken, maar ter ondersteuning dat de leerlingen een leidraad hebben wat ze moeten doen.

De docent zal dus op basis van de lessen dus moeten inleiden en sturen op basis van het eerdere hoofdstuk. Sommige lessen (zoals les 1) die vooral klassikaal ingericht zijn, zijn daarom ook niet terug te vinden in het leerlingen materiaal.

Afhankelijk van de docent en klas, kan besloten worden dat de leerlingen ook tussentijds al hun werk moeten inleveren. De huidige opbouw gaat er echter vanuit dat de leerlingen aan het einde van de 18 lessen hun werk inleveren en dat dan de beoordeling plaats vindt. Omdat alle lessen op elkaar voortbouwen zal het eindproduct het werk van de gehele module bevatten.

Les 1 – Inleiding Robotica

Leerdoelen

- De leerling kan uitleggen wat robotica is
- De leerling kan enkele Ethische dilemma's met betrekking tot robotica opnoemen
- De leerling kan het toelichten met voorbeelden waarom gezicht uitdrukkingen belangrijk zijn tijdens communicatie.

Lesopbouw en werkvormen

- 5 min – Klassikale uitleg – Inleiding
- 10 min – Tweetallen zelfstandig bezig – Uitzoeken wat robotica is
- 5 min – Klassikale discussie – Wat is robotica
- 5 min – Klassikale uitleg – Ethische dilemma's gespreken
- 5 min – Video – Ethische dilemma's van een zelfrijdende auto
- 10 min – Klassikale discussie – Belang van emoji's

Inhoud

De eerste les gaan de leerlingen kennis maken met robotica en ethische aspecten ervan. Hieronder staan voorbeeld onderwerpen om de leerlingen kennis te laten maken met robotica, en zelf na te laten denken over de dilemma's

Wat is robotica

Voordat je er dieper op in kan gaan, moet je weten wat robotica is. Het is belangrijk dat de leerlingen dit ook weten. Dit kan je bereiken door bijvoorbeeld de leerlingen zelf in 10 minuten uit te laten zoeken wat robotica is en wat voorbeelden ervan zijn. Vervolgens kan je discussiëren over wat onder robotica valt. Mogelijke discussie onderwerpen zijn:

1. Skype
2. Siri
3. Nummertjes-trek-apparaat bij de IKEA
4. Spraakherkenning in de auto ("navigeer naar XXX")
5. Een rumba (zelf stofzuigende stofzuiger)
6. Tesla auto pilot

Ethische dilemma's

Naast technische uitdagingen heeft robotica ook Ethische uitdagingen. Laat leerlingen zelf bedenken welke ethische problemen er kunnen ontstaan.

Later kan je de volgende voorbeelden erbij pakken:

- https://nl.wikipedia.org/wiki/Drie_wetten_van_de_robotica
- <http://webwereld.nl/development/83911-ethische--robot-laat-mensen-te-pletter-vallen>
- <https://www.youtube.com/watch?v=ixloDYVfKA0>

Emoties / Emoji

Dat emoties en uitdrukkingen belangrijk zijn in het digitale tijdperk, kunnen we wel afleiden van het belang van emoji's. Je komt emoji's tegenwoordig overal tegen (geschiedenis van emoji <https://www.youtube.com/watch?v=SoZlB9pFV2M>). Om de leerlingen hier meer het belang van te laten inzien van lichaamstaal en gezichtsuitdrukkingen bij tekst, kan je ze (whatsapp) berichten laten

bedenken die een compleet andere connotatie krijgen afhankelijk van de emoji's (Jeej 😊 versus Jeej 😞)

Video's robots met emoties

- iRobot: <https://www.youtube.com/watch?v=9A8lIA3jHJw>
- Wall-e: <https://www.youtube.com/watch?v=FvA78Z-W9UE>

Les 2 – Gezichtsmodel definiëren

Leerdoelen

- De leerling kan een simpel visueel model opstellen
- De leerling kan parameters voor een model definiëren

Lesopbouw en werkvormen

- 20 min – Klassikale uitleg – Wat is een model
- 5 min – Klassikale uitleg – Inleiding opdracht
- 20 min – Tweetallen zelfstandig bezig – Het model bedenken.

Inhoud

In de eerste les is de introductie met robotica gemaakt en het belang van emoties (emojis) duidelijk gemaakt. Deze les gaan de leerlingen een model maken van het menselijk gezicht voor het simuleren van gezichtsuitdrukkingen.

Binnen informatica ben je altijd bezig met het maken van modellen (versimpelde weergaven die de werkelijkheid representeren). In dit geval gaan de leerlingen een model maken van het menselijke gezicht, waarmee we vervolgens door middel van parameters emoties kunnen uitdrukken. Om het thema modelleren theoretisch te ondersteunen kan je verwijzen naar hoofdstukken in het boek die voor het vak gebruikt wordt. Indien de lesmethode hier geen theorie over heeft, kan je de volgende lessen serie op youtube gebruiken:

1. <https://www.youtube.com/watch?v=xHtsuOB-TPw>
2. <https://www.youtube.com/watch?v=iYSw2NPruFg>
3. <https://www.youtube.com/watch?v=Nc0IrpVXUT4>
4. <https://www.youtube.com/watch?v=oQTrwJWtc4M>

De leerlingen moeten deze les een model van een gezicht maken m.b.t. emoties. Dus welke onderdelen van een gezicht kunnen veranderen, en wat voor impact heeft dat op een gezichtsuitdrukking/emotie.

De bedoeling is dat de leerlingen zelfstandig een model leren opstellen en hierbij parameters bedenken die invloed hebben op de gezichtsuitdrukking. Het model is in dit geval een weergave van een gezicht, en bij parameters kan je denken aan:

- Hoogte van de wenkbrauwen
- Vorm van de wenkbrauwen
- Grootte van de ogen
- Richting van de pupillen
- Vorm van de mond
- In hoeverre je mond open staat
- Kleur van het gezicht
- Tranen/geen tranen

De leerlingen kunnen zelf deze parameters uitproberen met hun eigen gezicht, om te kijken wat het effect ervan is. Laat de leerlingen hiervan enkele foto's maken.

Les 3 – Model visualisatie

Leerdoelen

- De leerlingen kan een paper prototype maken van een idee
- De leerling kan een model valideren door het te prototypen.

Lesopbouw en werkvormen

- 5 min – Klassikale uitleg – inleiding
- 20 min – Tweetallen zelfstandig bezig – Afbeeldingen/onderdelen zoeken
- 20 min – Tweetallen zelfstandig bezig – Paper prototyping

Inhoud

In de vorige les hebben de leerlingen nagedacht over welke parameters geschikt zijn om een gezichtsuitdrukking te modelleren. In deze les gaan ze dit model maken en testen d.m.v. plaatjes/foto's/tekeningen. De bedoeling is dat ze deze afbeeldingen later gebruiken wanneer ze het model gaan implementeren.

Het doel is dat de leerlingen deze les als soort “paper prototyping” aan de slag gaan door, de verschillende onderdelen uit te printen en door “knippen en plakken” verschillende gezichtsuitdrukkingen samenstellen.

Hiervoor moeten ze plaatjes of foto's zoeken op internet om het gezicht uit op te bouwen, **het is nadrukkelijk niet de bedoeling om kant en klare gezichten te downloaden en te gebruiken**. Ze gaan met de verschillende onderdelen minstens 5 verschillende gezichtsuitdrukkingen visualiseren, en hierbij aangeven hoe de verschillende parameters dan ingesteld zouden moeten zijn. Hiervan moeten ze een foto maken om dit later in het verslag toe te voegen.

Laat de leerlingen 4 foto's van hun mooiste/beste uitdrukkingen opsturen, deze worden in les 6 gebruikt als “raad de uitdrukking en parameters”.

Les 4 & 5 – Herhaling programmeer skills

Leerdoelen

- De leerling kan lijsten toepassen om een programmeer probleem op te lossen.
- De leerling weet hoe hij commentaar bij code kan plaatsen en waar het voor dient.
- De leerling weet hoe hij bestaande functies van een programmeer kan gebruiken toepassen.
- De leerling kan zelf functies aanmaken en gebruiken.
- De leerling kan variabelen gebruiken en weet het verschil tussen types (String, Int, Double)

Lesopbouw en werkvormen

- 10 min – Klassikale uitleg – Korte herhaling van eerder uitgewerkte stof
- 35 min – Zelfstandig werken – Opdrachten volgen

- 5 min – Klassikale uitleg – Introductie, paar veel voorkomende problemen klassikaal behandelen
- 40 min – Zelfstandig werken – Opdrachten volgen

Inhoud

De vorige drie lessen stonden in het teken van de menselijke kant van de robotica. Deze twee lessen staan in het kader om alle benodigde programmeerkennis weer op te halen. De gemaakte opdracht is bedoeld voor VBA maar een vergelijkbare opdracht kan je voor andere programmeertalen maken. De volgende programmeer concepten komen naar voren in de opdracht

- Een UI kunnen maken en eigenschappen van UI elementen kunnen veranderen.
- Het kunnen definiëren en gebruiken van variabelen
- Het snappen van arrays en het gebruiken van for-loops
- Het kunnen gebruiken van bestaande functies (bijvoorbeeld voor random getal)
- Het kunnen definiëren en gebruiken van zelf gemaakte functies.

Bij het leerlingen materiaal staat een uitgebreide opdracht voor de leerlingen om deze vaardigheden te herhalen in Visual Basic for Applications. De opdracht begint met veel handvaten en wordt langzaam ruimer zodat de leerlingen meer zelf moeten bedenken

Note: Er is een voorbeeld uitwerking te vinden in het bestand "Voorbeelduitwerking – Moderobot.docm"

Les 6 - “Wat is de uitdrukking” en sentiment parameter

Leerdoelen

- De leerling kan feedback ontvangen en verwerken.
- De leerling kan samenhang tussen verschillende parameters bedenken en modelleren.

Lesopbouw en werkvormen

- 5 min – Klassikale uitleg – Introductie van de les
- 10 min – Kahoot – “Raad de uitdrukking”
- 5 min – Klassikale discussie – De uitkomsten van de kahoot bespreken
- 5 min – Klassikale instructie – “Hoofd parameter opstellen”
- 20 min – Tweetallen zelfstandig – Hoofd parameter bedenken en opstellen.

Inhoud

Deze les bestaat uit twee delen: “feedback op het model” en “hoofd parameter opstellen”

Feedback op het model

Tijdens de 3^e les hebben de leerlingen allemaal foto's opgestuurd van zelf samengestelde gezichtsuitdrukkingen en hun gebruikte parameters. Het doel van dit deel is dat leerlingen ook de modellen anderen te zien krijgen en met feedback hun eigen model kunnen beoordelen

Eerst kan je klassikaal “raad de gezichtsuitdrukking” doen. Hierbij zien de leerlingen een willekeurige van de ingeleverde gezichtsuitdrukkingen op het bord en geven dan aan wat zij denken dat de uitdrukking is. Hiervoor kan je bijvoorbeeld kahoot.it of edmodo.com gebruiken. Je kan de leerlingen extra motiveren door de persoon die de meeste goed heeft (of de persoon die de meest duidelijke emoties had) een bonus punt of prijs te geven.

Daarna kan worden gediscussieerd waarom bepaalde gezichten beter waren dan andere. En de verschillende parameters en de invloed hiervan bespreken.

Hoofd parameter opstellen

De leerlingen hebben nu parameters die de verschillende onderdelen van je gezicht aanpassen en/of verplaatsen. Echter komt uit de sentiment analyse van de volgende lessen maar 1 parameter die het sentiment uitdrukt. Het is daarom belangrijk dat de leerlingen 1 variabele maken die alle andere parameters zo aanpast dat je het gezicht in meerdere stappen van boos naar blij kan aanpassen.

Bijvoorbeeld -5 is heel boos, en 5 is heel blij, dan moet het gezicht voor alle waarden daar tussen een overgangsuitdrukking hebben.

Indien deze les sneller verloopt, kunnen de leerlingen bezig met de volgende opdrachten:

1. Verbeter de herkenbaarheid van je gezichtsuitdrukkingen
2. Schrijf een stukje over de herkenbaarheid van je uitdrukkingen voor je eindverslag
3. Probeer literatuur/bronnen te zoeken over facial expressions / gezichtsuitdrukkingen. En je eigen model hiermee te onderbouwen.

Les 7, 8 & 9 – Model implementeren

Leerdoelen

- De leerling kan een concept omzetten tot code

Lesopbouw en werkvormen

3x

- 5 min – Klassikale instructie – Korte introductie, wat ga je deze les doen en wat moet je reeds af hebben.
- 40 min – Zelfstandig met opdracht bezig – Elke leerling gaat zelf de opdracht maken. Echter mogen ze overleggen als ze vast zitten.

Inhoud

In deze drie lessen implementeren de leerlingen hun model. Aan het einde van deze drie lessen is het de bedoeling dat de leerlingen een applicatie hebben met een gezicht, waarbij ze in de userinterface de waarden van de parameters kunnen aanpassen om verschillende gezichtsuitdrukkingen te kunnen samenstellen.

Een belangrijk aspect is dat de leerlingen ook hun extra parameter die ze in de vorige les hebben bedacht implementeren. De sentiment analyse van het volgende onderdeel, moeten ze namelijk aan deze parameter koppelen.

De bouw van het gezicht kan in de volgende stappen opgezet worden:

1. Zorg dat je een applicatie bouwt met alle onderdelen van je gezicht er op.
2. Maak vier vaste knoppen, die de waarden van de parameters zo aanpassen dat je de gezichtsuitdrukking kan krijgen door op een knop te drukken blij, boos, neutraal, verdrietig.
3. Zorg dat d.m.v. invul velden en/of scrolbalken de parameters los aangepast kunnen worden, zodat je alle mogelijke gezichtsuitdrukkingen kan maken.
4. Maak je extra parameter voor blij/boos, die de andere parameters aanpast en zo je gezichtsuitdrukkingen in stappen van blij naar boos kan veranderen.

Les 10 & 11 – Sentiment analyse

Leerdoelen

- De leerling kan een simpele taal (sentiment) analyse maken.

Lesopbouw en werkvormen

- 5 min – Klassikale instructie – Introductie op de opdracht
- 40 min – Zelfstandig werken – De opdracht maken

- 5 min – Klassikale instructie – Voortgang bewaken, extra uitdaging voor betere leerlingen introduceren.
- 40 min – Zelfstandig werken – De opdracht afmaken.

Inhoud

Tijdens 10^e en 11^e lessen gaan de leerlingen met een sentiment analyse aan de gang. De leerlingen implementeren hier zelf een programma waar ze een zin kunnen invoeren en het programma een sentiment waarde teruggeeft. Deze opdracht bevat een voorbeeld basis die de leerlingen zelf verder moeten uitbouwen.

Het eerste deel van de opdracht laat de leerlingen een zin ontleden en vervolgens vergelijken met een basis lijst van woorden die een bepaald sentiment hebben. De betere leerlingen kunnen uitgedaagd worden om de analyse hoofdletter ongevoelig te maken of om leestekens te negeren of soms zelfs de woorden te matchen met behulp van een levenshtein distance⁴ (bijv <2) algoritme, waardoor je rekening kan houden met spelfouten.

Het tweede deel van de opdracht is om een “Algemeen sentiment” te berekenen, zodat je sentiment door 1 slechte zin niet gelijk omlaag schiet, maar dat je het gemiddelde van meerdere zinnen pakt voor de gemoedsuitdrukking. Dit kan op verschillende manieren gebouwd worden, in de opdracht wordt aangeraden om een publieke variabele te maken waarin je alle sentiment waardes bij houdt, en als eind sentiment het gemiddelde van de laatste 5 waarden pakt.

Ter ondersteuning van deze twee lessen is een stap-voor-stap leerlingen opdracht voor VBA waar de leerlingen hun sentiment machine mee maken.

Note 1: Voor deze opdracht hebben de leerlingen naast de opdracht ook het bestand “Leerling basis – sentiment.docm” nodig

Note 2: Er is een voorbeeld uitwerking te vinden in het bestand “Voorbeelduitwerking – sentiment.docm”

⁴ <https://nl.wikipedia.org/wiki/Levenshteinafstand>

Les 12 – Gezicht en sentiment combineren

Leerdoelen

- De leerling is in staat om meerdere applicaties samen te voegen tot een werkend geheel

Lesopbouw en werkvormen

- 5 min – Klassikale instructie – Uitleg wat de leerlingen moeten doen
- 40 min – Zelfstandig werken – Werken aan de opdracht

Inhoud

Deze les combineren de leerlingen hun gemaakte sentiment analyse met hun gezicht applicatie. Hiervoor moeten ze de onderdelen van de sentiment machine naar hun eigen applicatie toe kopiëren.

Aan het einde van deze les moeten de leerlingen een applicatie hebben waar je een zin kan invullen het sentiment bepaald wordt van de zin, en het gezicht zijn gezichtsuitdrukking hierop aanpast.

Voor de snelle leerlingen is een bonus opdracht toegevoegd, voor het uitzoeken van verschillende algoritmes voor het bepalen van het sentiment. De leerlingen kunnen dit aan hun eindverslag toevoegen voor bonus punten. Ook kunnen de bonus opdrachten van de vorige lessen nog gedaan worden

Les 13 – Webservice inleiding

Leerdoelen

- De leerling kan uitleggen wat een webservice is en waar deze voor gebruikt kan worden
- De leerling kan een JSON-bestand lezen en interpreteren
- De leerling kan uitleggen wat OAuth is en waarom het belangrijk is
- De leerling kan een query opstellen om een REST webservice aan te roepen.

Lesopbouw en werkvormen

- 5 min – Klassikale introductie – Waar staan we in het project wat gaan we vandaag doen
- 20 min – Klassikale instructie – Webservices
- 20 min – Zelfstandig werken – De twitter opdracht maken.

Inhoud

Deze les is een inleiding in webservices die bestaat uit een stukje theorie, en een opdracht waarmee de leerlingen zelf met de twitter API bezig gaan.

Theorie

Het is belangrijk dat de leerlingen weten wat een webservice is, waar je het voor kan gebruiken en hoe ze werken. De volgende punten zijn relevant:

- Wat is een (REST) webservice en hoe werken query parameters
- Wat is OAuth en waarvoor is OAuth belangrijk
- Wat zijn XML en JSON en waarvoor is het handig
- Hoe ziet JSON eruit

De volgende linkjes kan je gebruiken voor de uitleg, maar er kan ook verwezen worden naar relevante hoofdstukken uit lesmethode die standaard wordt gebruikt.

- Wat is webservice waarvoor gebruik je het
<https://www.youtube.com/watch?v=u80uPzhFYvc>
- Data formaten (json/xml)
<https://nl.wikipedia.org/wiki/JSON> / <https://nl.wikipedia.org/wiki/XML>
- Autorisatie -> OAuth
<https://blog.surf.nl/oauth-voor-beginners/>

Praktijk

De opdracht in het leerlingen materiaal laat de leerlingen zelf de twitter API aanroepen, en hier vervolgens vragen over beantwoorden. Twitter heeft een REST api die als autorisatie OAuth1a gebruikt. Voor het kunnen aanroepen van de API zijn twee dingen nodig:

1. Een applicatie om REST requests te kunnen doen
2. OAuth tokens om te kunnen authenticeren

Voor het eerste punt zijn verschillende applicaties beschikbaar zoals Postman, RestClient en SoapUI. En ook verschillende websites die dit kunnen zoals hurl.it

In de leerlingen opdracht staat hoe je OAuth tokens kan krijgen, hiervoor heb je echter wel een twitter account met gekoppeld telefoonnummer nodig. Daarom kan de docent ook algemene tokens aanmaken die de klas dan kan gebruiken.

Les 14 & 15 – Twitter koppeling bouwen

Leerdoelen

- De leerling is in staat d.m.v. een library een webservice aan te roepen
- De leerling is in staat om JSON output d.m.v. een library uit te lezen.

Lesopbouw en werkvormen

Les 14

- 5 min – Klassikale introductie – Herhaling vorige les, introductie deze opdracht
- 10 min – Bespreken van de Twitter opdracht
- 30 min – Zelfstandig werken – Zelfstandig de opdrachten uitvoeren

Les 15

- 5 min – Klassikale introductie – Herhaling vorige les, introductie deze opdracht
- 40 min – Zelfstandig werken – Zelfstandig de opdrachten uitvoeren

Inhoud

Als eerste kan de opdracht van de vorige les nog kort doorgesproken worden, zodat de concepten die de leerlingen moesten leren ook duidelijk worden gemaakt. Ze moeten nu namelijk de daar opgedane kennis gaan toepassen in een andere context.

De leerlingen weten nu hoe de twitter API werkt, ze kunnen met behulp van de twitter api verschillende zoekopdrachten uitvoeren, en kunnen de uitkomst van de API interpreteren. Deze lessen gaan ze een koppeling met Twitter bouwen om tweets te tonen naar aanleiding van een zoekopdracht.

Er is een voorbeeld bestand die reeds een library geïmplementeerd heeft voor de afhankelijk van OAuth en de rest requests. Het is aan de leerlingen zelf om deze library aan te roepen en de (json) output vervolgens uit te lezen. Het voorbeeld bestand bevat de stappen en uitleg om dit te doen.

De laatste stap voor de leerlingen is om hun sentiment & gezicht aan deze twitter api te koppelen, zodat de tweets in de sentiment analyse kunnen worden ingelezen. Het is het verstandigst voor de leerlingen om hun sentiment & gezicht naar de twitter applicatie te kopiëren, omdat hierin de complete twitter library zit opgenomen.

Na deze twee lessen moeten de leerlingen dus een applicatie hebben waarbij een tweet wordt opgehaald, het sentiment wordt bepaald en hierop de gezichtsuitdrukking wordt aangepast.

Note 1: Voor deze opdracht hebben de leerlingen naast de opdracht ook het bestand “Leerling basis – twitter.docm” nodig

Note 2: Er is een voorbeeld uitwerking te vinden in het bestand “Voorbeelduitwerking – twitter.docm”

Les 16 en 17 - Afronding

De laatste twee lessen kunnen de leerlingen gebruiken om hun applicatie te verbeteren en om het verslag te schrijven. Alle eisen aan de applicatie en het verslag zijn bij het leerlingen materiaal te vinden.

Les 18 - Toets

De laatste les moeten beide eindproducten ingeleverd worden (verslag en applicatie) en zal er een korte theorie toets gehouden worden over de lessen serie. Hieronder zijn twee voorbeeld toetsen gemaakt die voor de module gebruikt kunnen worden. Tevens zijn er enkele vragen gemaakt die de leerlingen kunnen gebruiken ter voorbereiding van de toets, deze vragen zijn in het leerlingen materiaal te vinden. Deze vragen komen overeen met het soort vragen die in de proefwerken zitten.

Beide toetsen bevatten 4 vragen met de volgende onderwerpen:

1. Robotica en ethische bezwaren
2. Maak een visueel model
3. Tekstanalyse
4. Webservices

De toets bevat dus 4 vragen, die samen een tijdsduur van 45 minuten in beslagen nemen.

TOETS A

Deze toets bestaat in totaal uit 4 vragen en totaal 35 punten. De normering is: Cijfer = $9 * [\text{punten}] / 35 + 1$

1. Deze vraag gaat over het begrip robotica. (5p)
 - a. Hieronder staat een stuk uit het woordenboek, je ziet dat er twee verschillende definities zijn van het woord robot. Licht toe of het door jou gemaakte gezicht aan deze definitie voldoet of niet
ro·bot (meervoud: *robots*)
1 (de; m) geheel van werktuigen en machines dat op een mens lijkt
2 (de; m,v) mens zonder individuele trekken die gedwee opdrachten uitvoert
 - b. Beschrijf in 5 tot 10 regels, optioneel met tekening, welke ethische bezwaren er zijn tegen zelf rijdende auto's
2. Naast zorgzame en educatieve toepassingen van robots, kan je robots ook inzetten als huisdier (bijv. robot hond, robot kat) met entertainment als doel. (10p)
 - a. Leg uit voor wie een robot huisdier handig kan zijn, en waarom.
 - b. Maak een visueel model van je robot huisdier, maak hiervan een schets en geef aan welke parameters er zijn en welke invloed die hebben op je model.
3. Kijk naar de volgende code die een zin analyseert: (10p)

```
Dim sentence As String
sentence = "Ki welloe g ik blijf lekker osso bij jou!"

' A.
Dim slangWordList As New Collection

slangWordList.Add 2, "mattie"
slangWordList.Add 2, "g"
slangWordList.Add 2, "spang"
slangWordList.Add 3, "loesoe"
slangWordList.Add 2, "osso"
slangWordList.Add 3, "g"
slangWordList.Add 4, "flex"
slangWordList.Add 4, "Ki"
slangWordList.Add 4, "welloe"
slangWordList.Add 5, "doekoe"
slangWordList.Add 5, "woella"

' B.
Dim inputWords() As String
inputWords = Split(sentence, " ")

' C.
Dim sentiment As Double
Dim wordCount As Integer
sentiment = 0
wordCount = 0

' D.
Dim i As Integer
For i = LBound(inputWords) To UBound(inputWords)
    If ItemExist(slangWordList, inputWords(i)) = True Then
        wordCount = wordCount + 1
        sentiment = sentiment + slangWordList.Item(inputWords(i))
    End If
Next i

' E.
If wordCount = 0 Then
    MsgBox ("No slang at all")
Else
    MsgBox ("The score is: " & (sentiment / wordCount))
End If
```

eren, wat zou de uitkomst dan zijn. Leg uit via welke stappen je op deze uitkomst gekomen bent

- b. Je kan twee soorten meldingen krijgen “No slang at all” of “The score is: X”.
 - i. Leg uit bij welke input je de eerste melding krijgt en waarom.
 - ii. Leg uit wat het laagste getal is dat op de plek van X kan staan, en wat de hoogste? Geef ook een voorbeeld van een zin om die score te krijgen
- c. Wat zou er gebeuren voor de sentiment score als we een komma aan de zin toevoegen: *Ki welloe g, ik blijf lekker osso bij jou*
- d. Je ziet in de code A, B, C, D en E staan, geef bij elk punt aan wat de regels code die eronder staan doen, en waarvoor ze dienen binnen de applicatie.

4. De volgende json is een output van een webservice verzoek naar facebook op de url:
<https://graph.facebook.com/v2.9/me?fields=picture%2Cbirthday%2Cdevices%2Ceducation%2Csports%2Cname> (10p)

a. W

```
{
  "picture": {
    "data": {
      "is_silhouette": false,
      "url": "https://scontent.xx.fbcdn.net/v/t1.0-1/c170.5XXXXX"
    }
  },
  "birthday": "02/16/1992",
  "devices": [{"os": "Android"}],
  "education": [
    {
      "school": {
        "id": "115441048469524",
        "name": "Dollard College"
      },
      "type": "High School",
      "year": {
        "name": "2010"
      },
      "id": "116315491777192"
    },
    {
      "school": {
        "id": "108176759266813",
        "name": "Utwente"
      },
      "type": "College",
      "id": "116315435110531"
    }
  ],
  "sports": [
    {
      "name": "Judo",
      "id": "115598388451256",
      "name": "Triathlons",
      "with": [
        {
          "name": "Kim Hendriksen",
          "id": "10205205322145253"
        }
      ]
    }
  ],
  "name": "Hans Jansens"
}
```


k doe we aanroepen op de webservice?

- b. Hoeveel opleidingen heeft de persoon op zijn profiel ingevuld? Geef aan hoe je dit hebt gevonden
- c. Bij sporten -> Triathlon kan je vinden met wie deze persoon de sport uitvoerde. Wat zou het id daar betekenen?
- d. Wanneer is deze persoon jarig? Geef hierbij toelichting hoe je daar bij komt, en wat je op valt.
- e. Pas de URL zo aan dat de webservice niet de sporten teruggeeft.
- f. Als je kijkt naar de URL, van welke gebruiker zullen we dan data terug gekregen hebben?

-- EINDE --

TOETS B

Deze toets bestaat in totaal uit 4 vragen en totaal 35 punten. De normering is: Cijfer = $9 * [\text{punten}] / 35 + 1$

1. Deze vraag gaat over het begrip robotica. (5p)
 - a. Geef twee argumenten waarom zelfrijdende auto's wel veilig zijn, en geef 2 argumenten waarom ze niet veilig zouden zijn.
 - b. Leg in 5 tot 10 regels met een voorbeeld uit waarom het ontbreken van gezichtsuitdrukkingen problematisch kan zijn bij digitale communicatie en hoe dit probleem opgelost kan worden.
2. Je kan van alle dingen in de wereld een model maken, zelfs van een koekoeksklok (10p)
 - a. Maak een visueel model een koekoeksklok, maak hiervan een schets en geef minimaal 4 parameters welke invloed die hebben op je model. De belangrijkste onderdelen van een koekoeksklok moeten terugkomen in je model.
3. Kijk naar de volgende code die een zin analyseert: (10p)

```
Dim sentence As String
sentence = "Ki welloe g ik blijf lekker osso bij jou!"

' A.
Dim slangWordList As New Collection

slangWordList.Add "mattie"
slangWordList.Add "g"
slangWordList.Add "spang"
slangWordList.Add "loesoe"
slangWordList.Add "osso"
slangWordList.Add "g"
slangWordList.Add "flex"
slangWordList.Add "Ki"
slangWordList.Add "welloe"
slangWordList.Add "doekoe"
slangWordList.Add "woella"

' B.
Dim inputWords() As String
inputWords = Split(sentence, " ")

Dim totalInputWords As Integer
totalInputWords = UBound(inputWords)

' C.
Dim wordCount As Integer
wordCount = 0

' D.
Dim i As Integer
For i = LBound(inputWords) To totalInputWords
    If listContains(slangWordList, inputWords(i)) = True Then
        wordCount = wordCount + 1
    End If
Next i

' E.
If totalInputWords = 0 Then
    MsgBox ("No slang at all")
Else
    MsgBox ("The score is: " & (wordCount / totalInputWords * 100) &
"%")
End If
```

- , wat zou de uitkomst dan zijn? Leg uit via welke stappen je op deze uitkomst gekomen bent
- b. Je kan twee soorten meldingen krijgen “No slang at all” of “The score is: X%”.
 - i. Leg uit bij welke input je de eerste melding krijgt en waarom.
 - ii. Leg uit wat het laagste getal is dat op de plek van X kan staan, en wat de hoogste? Geef ook een voorbeeld van een zin om die scores te krijgen
 - c. Leg uit wat de code nu doet met hoofdletters die in de zin staan, en geef aan bij welk punt (A, B, C, D of E) en wat we zouden moeten toevoegen om dit te veranderen. ** het is voldoende om uit te leggen wat de code moet doen, je hoeft niet exact werkende code als antwoord te geven*
 - d. Je ziet in de code A, B, C, D en E staan, geef bij elk punt aan wat de regels code die eronder staan doen, en waarvoor ze dienen binnen de applicatie.
4. De volgende json is een output van een webservice verzoek naar instagram op de url:
<https://api.instagram.com/v1/media/search?lat=48.858844&lng=2.294351> (10p)

```
{
  "data": [{
    "distance": 41.741369194629698,
    "type": "image",
    "users_in_photo": [],
    "filter": "Earlybird",
    "tags": [],
    "comments": {
      "count": 2
    },
    "caption": null,
    "likes": {
      "count": 1
    },
    "link": "http://instagr.am/p/BQEEq/",
    "user": {
      "username": "mahaface",
      "profile_picture": "http://distillery.s3.amazonaws.com/profiles/X.jpg",
      "id": "1329896"
    },
    "created_time": "1296251679",
    "images": {
      "low_resolution": {
        "url": "http://distillery.s3.amazonaws.com/media/2011/01/28/X.jpg",
        "width": 306,
        "height": 306
      },
      "thumbnail": {
        "url": "http://distillery.s3.amazonaws.com/media/2011/01/28/X.jpg",
        "width": 150,
        "height": 150
      },
      "standard_resolution": {
        "url": "http://distillery.s3.amazonaws.com/media/2011/01/28/X.jpg",
        "width": 612,
        "height": 612
      }
    },
    "id": "20988202",
    "location": null
  },
  {
    "distance": 41.741369194629698,
    "type": "video",
    "videos": {
      "low_resolution": {
        "url": "http://X-13.ak.instagram.com/X.mp4",
```

```

        "width": 480,
        "height": 480
    },
    "standard_resolution": {
        "url": "http://X-13.ak.instagram.com/X.mp4",
        "width": 640,
        "height": 640
    },
    "users_in_photo": null,
    "filter": "Vesper",
    "tags": [],
    "comments": {
        "count": 2
    },
    "caption": null,
    "likes": {
        "count": 1
    },
    "link": "http://instagr.am/p/D/",
    "user": {
        "username": "kevin",
        "full_name": "Kevin S",
        "profile_picture": "http://distillery.s3.amazonaws.com/profiles/X.jpg",
        "id": "3"
    },
    "created_time": "1279340983",
    "images": {
        "low_resolution": {
            "url": "http://distilleryimage2.ak.instagram.com/X_6.jpg",
            "width": 306,
            "height": 306
        },
        "thumbnail": {
            "url": "http://distilleryimage2.ak.instagram.com/X_5.jpg",
            "width": 150,
            "height": 150
        },
        "standard_resolution": {
            "url": "http://distilleryimage2.ak.instagram.com/X_7.jpg",
            "width": 612,
            "height": 612
        }
    },
    "id": "3",
    "location": null
}

```

- Wat zegt het deel ?lat=48.858844&lng=2.294351 in de url?
- Hoeveel zoekresultaten zijn er gevonden, en hoeveel likes hebben deze resultaten?
- Hoeveel en welke foto's kan je allemaal bij een resultaat vinden.
- Om in te loggen moeten we een token meesturen i.p.v. je gebruikersnaam en wachtwoord (oAuth), leg in 5 tot 10 regels uit waarom dit veiliger is.

-- EINDE --

Nakijkmodellen

Hieronder zijn voor beide toetsen nakijk modellen te vinden, omdat er ook open vragen in zitten, is bij deze vragen enkel een lijst van mogelijke antwoorden gegeven. Deze lijst is niet sluitend en het is mogelijk dat de leerlingen correcte antwoorden geven die niet in dit model staan.

Verder is het totaal te behalen punten voor de toets 35, het cijfer kan op de volgende manier berekend worden: $\text{Cijfer} = 9 * [\text{punten}] / 35 + 1$

Nakijkmodel toets A

1. Robotica (5p - 5min)

- a. **Nee**, er is geen geheel van machines gebouwd, enkel een interface die op een robot gemonteerd zou kunnen worden. (1p)
- b. Argumenten die goed zijn: (4p)
 - Software is nooit fout vrij, dus waarom zou dit wel fout vrij zijn
 - Een auto kan nooit antipieren op dingen waarvoor hij niet geprogrammeerd is (overstekend hert, overstromingen etc)
 - Een computer kan nooit goed alles herkennen (is het een bord of een bal, is het een hert of een kind)
 - Indien de auto in dubio komt tussen twee kwaden, wat kiest hij dan en wie bepaald dat. (Rij je omstanders aan, of rij je zelf in een kanaal)

2. Huisdier model (10p – 15min)

- a. Voorbeelden die genoemd kunnen zijn: (2p)
 - Eenzame ouderen
 - Allergische mensen
 - Mensen die veel werken
- b. Er moet een totaal model gemaakt zijn met minimaal 2 goede parameters (8p)
Voorbeeld parameters zijn:
 - Tong (om te likken)
 - Staart (om te kwispelen)
 - Oren (oren hangen, of staan omhoog)
 - Bek (bek open/dicht of met/zonder uithangende tong)
 - Poten (om een poot te kunnen geven)
 - Positie (staand/zittend/liggend)
 - Hoeveelheid geluid (blaffen/miauwen)
 - Ogen (kraaloogjes of juist niet)

Een voorbeeld schets van het model zou kunnen zijn:



Waarbij de toelichting is gegeven dat A voor de stand van de oren staat, C de ogen aan geeft en B de parameter is die bepaald of de tong uit de mond hangt.

3. Zin analyse: (10p – 15min)

- a. De uitkomst is “The score is: 3” de volgende stappen moeten worden aangegeven.

(1p voor uitkomst 1p voor minstens 3 van de volgende stappen)

1. Eerst wordt de zin gesplitst op een spatie
2. Vervolgens doorlopen we alle slang woorden en kijken we of elk woord in de zin voor kwam, in dit geval vind hij de woorden Ki, welloe, g en osso
3. Voor elk woord halen we sentiment waarde op en daar berekenen we de som van $(4 + 4 + 3 + 1) = 12$
4. Er zijn 4 woorden gevonden dus $12 / 4 = 3$, en dat wordt d.m.v. de message box getoont

b.

- i. Als er geen slang woorden in de zin voorkomen, is wordCount 0 en krijg je geen melding, **deze is er om delen door nul te voorkomen.** (1p)

- ii. Laagste: Het laagste was er in score kan komen is 2, namelijk als je een zin maakt met als enige slang woord osso, mattie, g of spang. (1p)

Hoogste: De hoogste score is 5, namelijk als je enkel de slang woorden doekoe of woella gebruikt (1p)

- c. Dan wordt het woord g niet meer meegenomen, omdat het programma geen rekening houdt met leestekens. De score wordt dan als volgt berekend $(4 + 4 + 1) / 3 = 3$ (2p)

- d. Hieronder per punt de minimale omschrijvingen (3p)

A. We maken een nieuwe collectie met slang woorden, de key is het woord de value de score

B. Splits de input zin op spaties en plaatst dat in een array

C. Maak nieuwe variabelen aan waarmee we straks gemiddelde kunnen berekenen

D. Doorloop alle woorden uit de input zin, en kijk of ze in de lijst met slang woorden voorkomen, indien dat het geval is pas de variabelen van C aan.

E. Omdat we niet kunnen delen door 0, moet deze apart afgevangen worden.

4. Webservice (10p – 10min)

- a. Dit is URLEncoding, alle speciale tekens worden omgezet naar een %XX formaat. Dit om verwarring te voorkomen (bijv: index.php?link=hoi&doei, is de link hier “hoi&doei” of is de link hoi, en is doi een nieuwe variabele) (2p)

- b. Drie opleidingen, er wordt drie keer wat getoond met “school”. (2p)

- c. Het id van de deelnemer + het id van Kim Hendriksen (2p)

- d. 16 februari (1p) dat zie ik aan birthday, staat wel in Amerikaanse notatie (2p)

- e. <https://graph.facebook.com/v2.9/me?fields=picture%2Cbirthday%2Cdevices%2Ceducation%2Cname> (2p)
- f. me (1/2p) Hans Jansens (1/2p)

Nakijkmodel toets B

1. Robotica (5p – 5min)

a. Voorbeelden van argumenten voor (1p)

- Een computer heeft een veel sneller reactievermogen
- Een computer kan veel meer tegelijk
- Een computer raakt niet afgeleid
- Een computer kan situaties doorrekenen

Voorbeelden van argumenten tegen (1p)

- Een computer kan moeilijker dingen herkennen (bal v.s. verkeersbord)
- Een computer kan geen emoties in beslissingen leggen
- Ongelukken komen vaak “door de ander” dus het heeft pas zin als alle auto’s zelf rijden
- Soms zijn er afwijkende situaties waar computers niet mee overweg kunnen (ontbrekende belijning, wegwerkzaamheden etc)

b. De kern van het antwoord moet aangeven dat bijklanken zoals ironie, sarcasme of juist blijf of boos niet overgebracht worden. Dit is met tekst berichten opgelost door

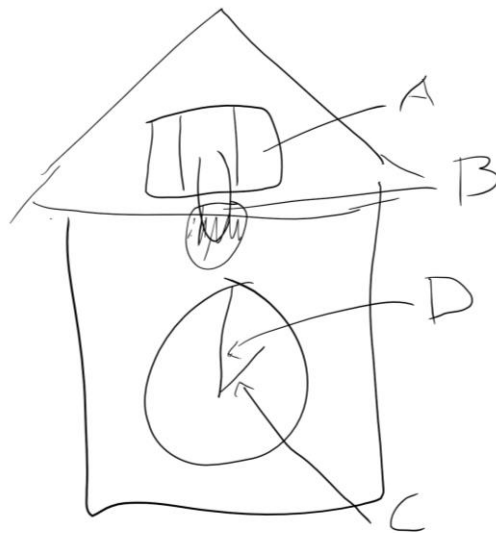
middel van emoji's. Bijvoorbeeld: Jeej 😊 versus Jeej 😞 (2p uitleg 1p voorbeeld)

2. Model (10p – 15min)

a. Er moeten minimaal 4 parameters zijn, hieronder zie je een paar mogelijke (5p)

- Hoek grote wijzer
- Hoek kleine wijzer
- Stand deurtje
- Stand vogel
- Geluid aan / uit
- Stand slinger

b. Maak een visueel model van X, maak hiervan een schets en geef aan welke parameters er zijn en welke invloed die hebben op je model. (5p)



A: parameter voor hoe ver het deurtje open staat

B: parameter voor hoe ver de vogel uit het deurtje is

C: De stand van de kleine wijzer

D: De stand van de grote wijzer

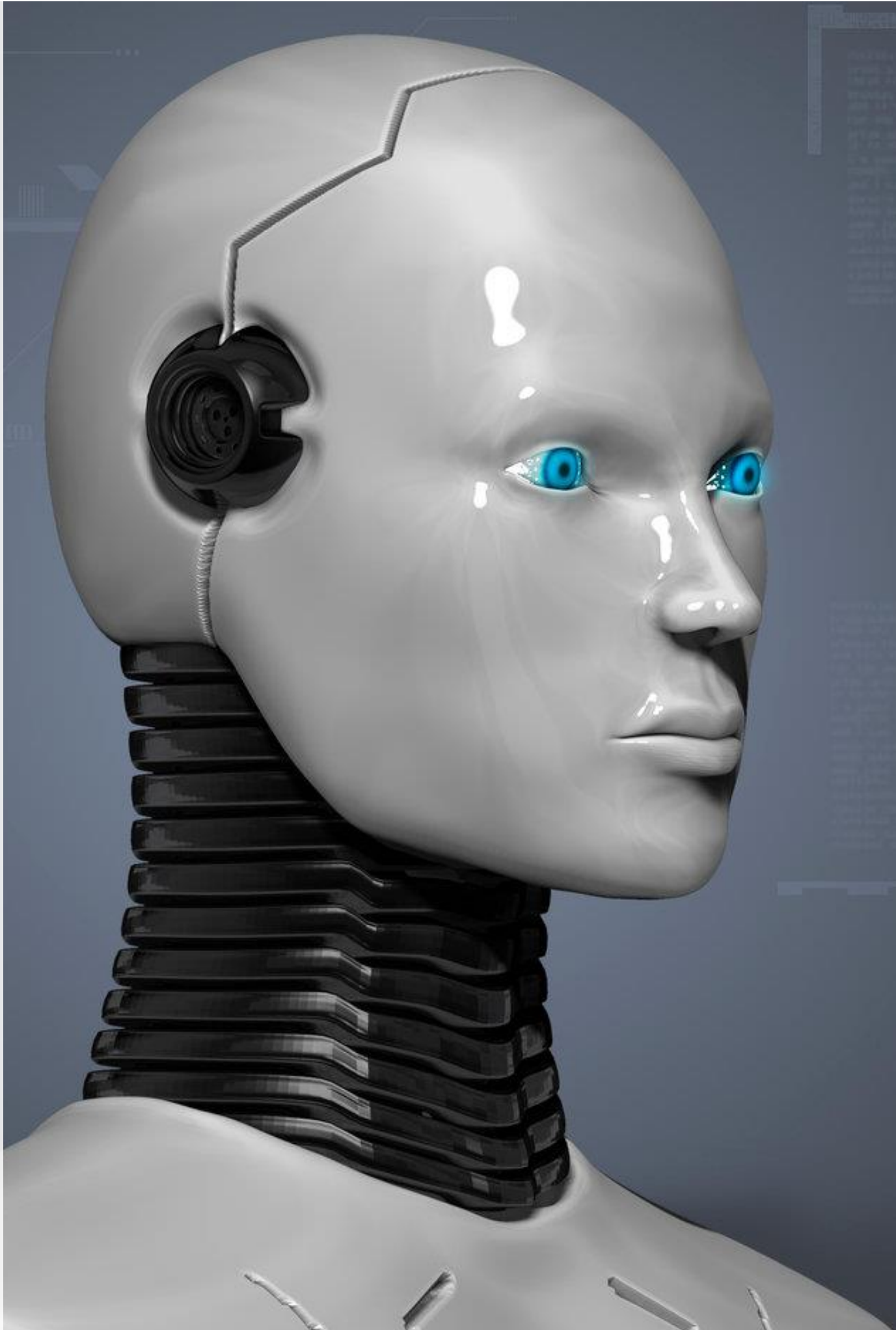
3. Taal analyse (10p – 15min)

a. De uitkomst is “The score is: 50%” de volgende stappen moeten worden aangegeven. (1p voor uitkomst 1p voor minstens 3 van de volgende stappen)

1. Eerst wordt de zin gesplitst op een spatie

2. Vervolgens bepalen we hoeveel woorden er totaal in de zin zaten
2. daarna doorlopen we alle slang woorden en kijken we of elk woord in de zin voor kwam, in dit geval vind hij de woorden Ki, welloe, g en osso
4. 4 van de 8 woorden zijn slang, dus dan is de score 50%
- b. Je kan twee soorten meldingen krijgen “No slang at all” of “The score is: X%”.
 - i. Als je invoer zin leeg is, dan is totalInputWords 0 en krijg je geen melding, **deze is er om delen door nul te voorkomen.** (1p)
 - ii. Laagste: Het laagste 0 is, als er helemaal geen slang woorden in voor komen (1p)
 Hoogste: De hoogste score is 100 als er enkel slang woorden in voor komen. (1p)
- c. De woorden zijn hoofdletter gevoelig, dus als de zin nu ki ipv Ki bevat, zal hij hem niet herkennen. Bij punt B of D kunnen we de ingevoerde zin/woorden allemaal een lange letter maken, en bij A moet dan alles met kleine letter ingevuld worden (2p)
- d. Hieronder per punt de minimale omschrijvingen (3p)
 - A. We maken een nieuwe collectie met slang woorden, we hebben hier enkel een value.
 - B. Splits de input zin op spaties en plaatst dat in een array
 - C. Bepaal het totaal aantal woorden in de zin, en maak een variabelen om slang woorden te tellen
 - D. Doorloop alle woorden uit de input zin, en kijk of ze in de lijst met slang woorden voorkomen, indien dat het geval is te 1 bij de totalCount variabele op
 - E. Omdat we niet kunnen delen door 0, moet deze apart afgevangen worden.
4. Webservice (10p – 10min)
 - a. De lat en lng geeft een locatie aan, we willen dus resultaten in de buurt van deze coördinaten. (2p)
 - b. Er zijn twee resultaten, de eerste met 3828 likes de tweede met 1 like (2p)
 - c. Een resultaat heeft 4 foto's, 1 met de avatar van de user, 1 met een thumbnail van de geuploadede foto, een lage resolutie variant en de normale. (2p)
 - d. De volgende rekenen kunnen gebruikt worden (4p)
 - Anders staat je gebruikersnaam en wachtwoord in de code, en kan iedereen die zien
 - De tokens kunnen weer ongeldig verklaard worden, waardoor je ze beter kan beheren
 - Je kan bepaalde rechten aan een token hangen, bijvoorbeeld dat je wel mag zoeken maar niet mag plaatsen.
 - De tokens zijn langer dan een normaal wachtwoord, en dus lastiger te kraken
 - Als je een bezoeker op je website aan de API wil koppelen hoeft hij nu niet zijn inloggegevens op jou website in te vullen, je krijgt enkel de tokens de zien.

Leerling materiaal



Lesmodule Sociale Robotica

Gezichtsuitdrukkingen

Leerling materiaal

De komende lessen ga je bezig met een module over gezichtsuitdrukkingen. De eerste lessen gaan over de menselijke aspecten binnen robotica en zal je een model gaan opstellen van een gezicht. De latere lessen ga je bezig dit model te implementeren en te koppelen aan een sentiment analyse en twitter feed.

Robots nemen steeds meer taken van ons over en worden belangrijker in ons leven. Het is daarom belangrijk dat je weet wat de impact hiervan is, maar ook hoe je robots kan verbeteren en inzetten voor bepaalde doeleinden.

Het einddoel is een applicatie met een gezicht die verschillende gezichtsuitdrukkingen kan tonen, welke automatisch aanpassen op basis van het sentiment (blij/boos) van tweets op een bepaald twitter kanaal. Je kan hierbij denken aan een robot die boos kijkt als er veel wordt getwitterd over regen, en die blij is als er wordt getwitterd over mooi weer; of waarbij de gezichtsuitdrukking de tweets over files of de vertragingen bij de NS weergeeft.

Aan het einde van deze lessen serie moet je twee dingen inleveren:

1. Een kort verslag
2. Een werkende applicatie

De exacte eisen van het in te leveren werk kan je aan het einde van dit document vinden. Verder wordt de lessen serie afgesloten met een korte theorie.

Inhoudsopgave

Leerling materiaal.....	1
Inhoudsopgave	2
Les 2: Gezicht modelleren.	3
Les 3: Gezicht visualiseren.....	4
Les 4 & 5 – Mode robot.....	5
Les 6.....	9
Les 7, 8 & 9	10
Les 10 & 11: Sentiment analyse	11
Les 12: Gezicht laten reageren.....	13
Les 13: Twitter API.....	14
Les 14: Twitter Machine	16
Les 15: Opdrachten koppelen	18
Les 16 & 17: Eindproduct	19
Les 18: Toets.....	20

Les 2: Gezicht modelleren.

In deze opdracht maak je een model van een gezicht met als doel om emoties en gezichtsuitdrukkingen weer te geven. Bedenk welke onderdelen in een gezicht belangrijk zijn om een duidelijke gezichtsuitdrukking te hebben en met welke parameters je deze onderdelen kan aanpassen.

1. Bedenk minstens drie parameters die onderdeel zijn van je gezichtsmodel.
2. Probeer de impact van de parameters duidelijk te maken, door de invloeden met je eigen gezicht na te doen. Maak hiervan enkele foto's.
3. Maak een duidelijke tekening/schets met alle onderdelen van je model, en hoe de bedachte parameters de onderdelen beïnvloeden.

Les 3: Gezicht visualiseren

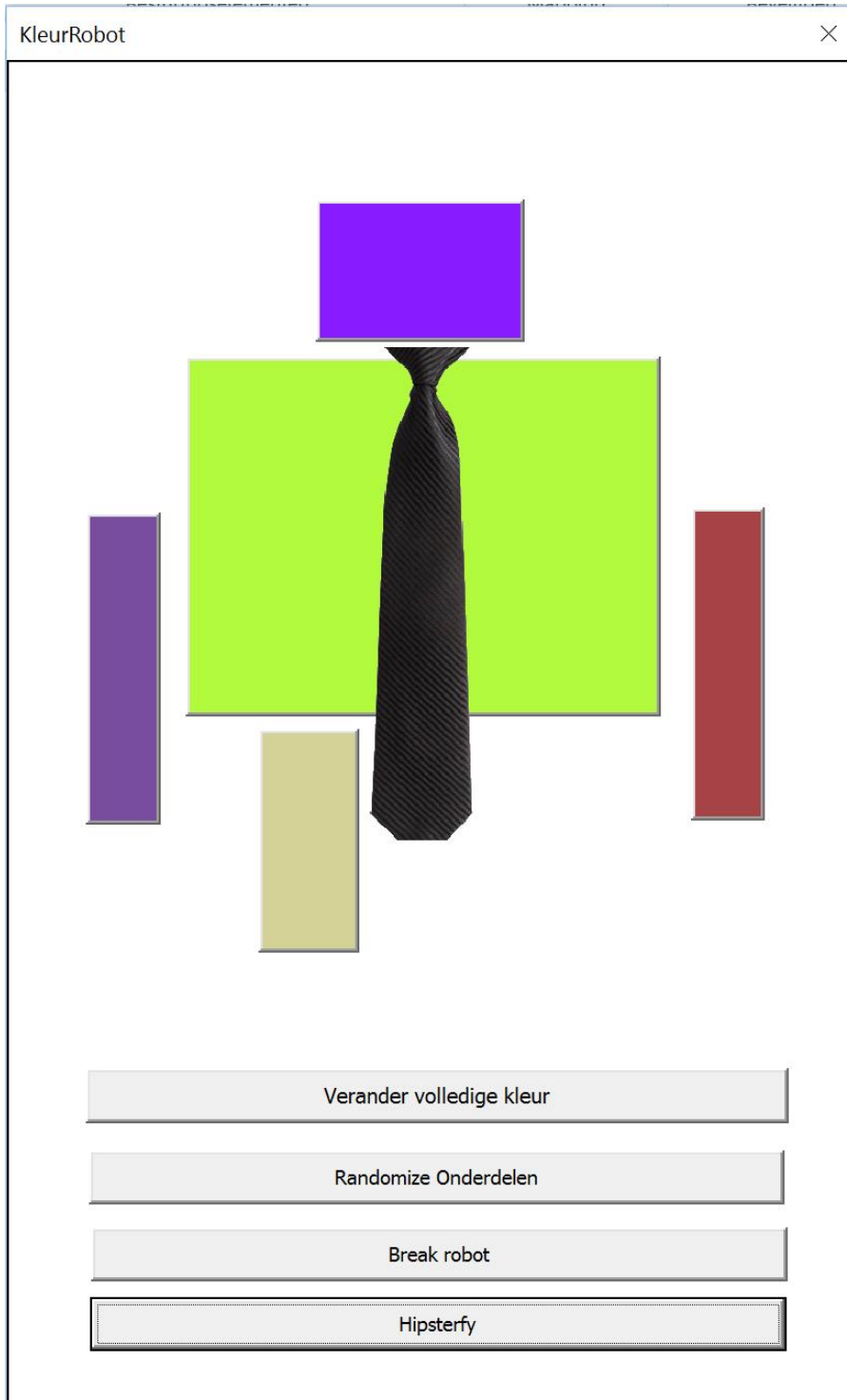
Je hebt reeds een model gemaakt en hierbij enkele parameters vastgesteld. Deze les ga je dit model omzetten naar gezichtskenmerken en hem testen. Dit gezicht wordt de basis voor je eind applicatie.

1. Zorg dat je afbeeldingen krijgt van alle losse onderdelen van je model (als ogen een onderdeel van je model is, dan heb je dus verschillende plaatjes van ogen nodig die veranderen naar aanleiding van de parameter).
 - a. Je kan deze onderdelen zelf tekenen, je mag ook foto's nemen en ze hier uit knippen of bestaande onderdelen van internet halen.
 - b. **Het is nadrukkelijk niet de bedoeling om "volledige" gezichten te gebruiken**
2. Zorg dat je ook een afbeelding krijgt van je "basis", dus een omtrek van een hoofd.
3. Print en knip deze onderdelen allemaal uit.
4. Probeer met alle losse afbeeldingen nu gezichtsuitdrukkingen te simuleren. Doe dit voor minimaal 6 verschillende gezichtsuitdrukkingen, maak hiervan foto's en geef ook de waardes aan die de parameters hebben.
5. Lever je beste 4 gezichtsuitdrukkingen in voor de volgende les

Les 4 & 5 – Mode robot

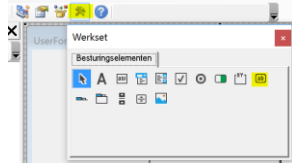
Inleiding

Deze opdracht is bedoeld om even alle basisvaardigheden van programmeren te herhalen. Het einddoel is een robot waarvan je de uiterlijke kenmerken kan aanpassen. Hieronder een voorbeeld van hoe je robot eruit kan komen te zien

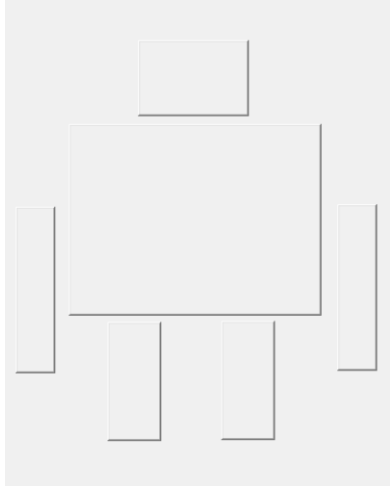


Kleur robot

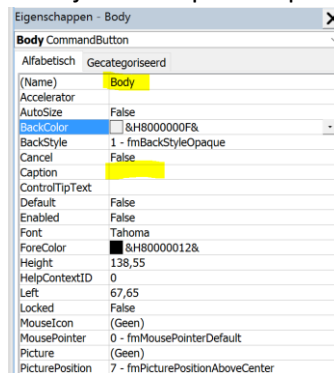
1. Start Word op en ga naar ontwikkelaars modus
2. Maak een nieuwe userform
3. Maak een robot van knoppen
 - a. Gebruik de werkset voor het maken van een nieuwe knop



- b. Plaats 6 knoppen op de volgende manier



- c. Geef de knoppen een duidelijke naam (bijv. LeftArm, RightArm, LeftLeg etc), en verwijder de caption. Optioneel kan je de robot al een begin kleur geven.



4. Plaats een grote knop onder de robot met de tekst "Verander volledige kleur". Dubbel klik op deze knop zodat je in de code weergave komt.
5. Pas de kleur aan van de robot als je op de knop drukt
 - a. Met de volgende regel kan je de kleur van een knop aanpassen:
`LeftArm.BackColor = RGB(255,0,0)`
Hiermee maken we de knop met de naam *LeftArm* rood
(<https://nl.wikipedia.org/wiki/RGB-kleursysteem>)
 - b. Herhaal deze regel voor alle knoppen zodat de complete robot rood wordt nadat je op de knop hebt gedrukt.
6. Bepaal een willekeurige kleur zodra je op de knop drukt
 - a. Met de code
`Math.Rnd()`
Kan je een willekeurig getal tussen de 0 en 1 krijgen. Als je dit vermenigvuldigd met 255 heb je een willekeurig getal tussen de 0 en 255

- b. Pas de code van punt 5 zo aan dat de robot een willekeurige kleur krijgt i.p.v. altijd rood, dit doe je door de 255, 0 en 0 te vervangen door willekeurig getallen tussen 0 en 255
7. Geef alle onderdelen van de robot dezelfde kleur
 - a. Als je voor elk onderdeel opnieuw een willekeurig getal tussen 0 en 255 pakt creëer je elke keer een nieuwe kleur. Je wil 1 keer een kleur bepalen en deze vervolgens opslaan en elke keer gebruiken. Op de volgende manier kan je de kleur bepalen en in een variabele stoppen:


```
randomColor = RGB(Math.Rnd() * 255, Math.Rnd() * 255, Math.Rnd() * 255)
```
 - b. Geef vervolgens de onderdelen de opgeslagen kleur:


```
LeftArm.BackColor = randomColor
```
8. Maak een 2e knop, die de functionaliteit krijgt dat elk onderdeel een andere kleur krijgt. Hiervoor moet je dus elke keer een nieuwe kleur bepalen en die als achtergrond instellen van de knop.

Array

9. Je hebt gezien dat je nu op 2 plekken elke keer het volgende aan het herhalen bent:


```
LeftArm.BackColor = ..
RightArm.BackColor = ..
LeftLeg.BackColor = ..
....
```

We kunnen ook een functie maken die altijd een lijst terug geeft met alle onderdelen van de Robot, als we later de robot dan extra onderdelen willen geven hoeven we dat onderdeel maar op 1 plek aan de lijst toe te voegen

 - a. Ga naar de code weergave en typ helemaal bovenaan de volgende code:


```
Private Function getParts()
    Dim parts(6) As CommandButton
    Set parts(0) = LeftArm
    Set parts(1) = RightArm

    getParts = parts
End Function
```

We geven hier aan het begin aan dat we een lijst hebben met 6 knoppen er in, en die lijst heet parts. Vervolgens voegen we de linkerarm toe op de eerste positie (getal 0), en rechter arm op tweede positie (getal 1). En aan het eind zeggen we dat de functie getParts alle robot onderdelen terug moet geven.
 - b. Vul de bovenstaande code aan dat alle robot onderdelen in de lijst zitten.
 - c. We passen nu de “kies kleur voor hele robot” knop aan, zodat hij ipv de herhaling de lijst gebruikt om de kleur aan te passen.


```
randomColor = getRandomColor()
Dim parts() As CommandButton
parts = getParts()

For i = 0 To UBound(parts) - 1
    parts(i).BackColor = randomColor
Next i
```

Allereerst bepalen we een kleur en die slaan we op in de variabelen (net als eerder). Vervolgens definiëren we een nieuwe variabele genaamd parts en geven aan dat hier een lijst van knoppen in mag komen. We roepen onze eerder gemaakte functie aan en plaatsen de lijst in de variabele. Vervolgens doorlopen we de hele lijst, en passen we de kleur van de knop aan. (UBound, telt het aantal items in een lijst)

10. Pas vervolgens de knop aan die je bij punt 8 gemaakt hebt, zodat deze gebruik maakt van de functie `getParts()`

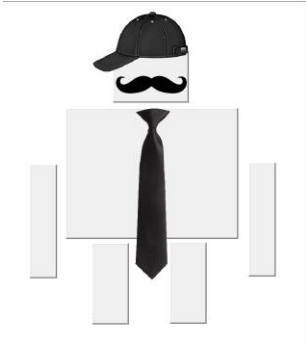
Break the robot

11. Maak een extra knop met de titel “Sloop de robot”
12. Voeg de code toe, dat zodra je op de knop drukt een willekeurig onderdeel van de Robot verdwijnt.
- Doorloop de lijst met onderdelen en zet ze allemaal op zichtbaar
 - Vervolgens kan je met de onderstaande code een willekeurig onderdeel uit de lijst selecteren en die onzichtbaar maken:

```
parts[Math.Rnd() * (UBound(parts) - 1)].Visible = True
```

Hipsterfy

13. Zoek op internet plaatjes van accessoires voor de robot, en zet deze er op bijvoorbeeld zoals hier onder. (*Note: met een gif plaatje kan je ook een transparante achtergrond hebben*)



14. Maak een nieuwe functie `getAccesoires()` die een lijst met Image objecten terug geeft van de accessoires (net als bij 9a)
15. Voeg een knop “Hipsterfy” toe, die elke keer een willekeurig accessoire uit deze lijst laat zien.
16. **EXTRA:** Probeer de knop zo aan te passen, dat hij niet 1 maar soms ook meerdere accessoires laat zien. (*Hint: bepaal in de loop voor elk accessoire of je hem wil laten zien op basis van een vergelijking met `Math.rnd()`*)

De broek aan

17. Zoek/maak een paar even grote plaatjes van broeken, die op je robot passen.
18. Maak een knop die de robot elke keer een andere broek aantrekt.
19. Je kan dit op verschillende manieren aanpakken, probeer beide manieren uit en schrijf op welke je handiger vindt.
- Zet alle plaatjes op elkaar op de juiste plek en laat d.m.v. `Visible` de juiste broek zien
 - Zet 1 leeg Image object op de plek van de broek, en plaats alle plaatjes van broeken buiten het object. Vervolgens kan je bijvoorbeeld met de volgende code de broek aanpassen:

```
PantsHolder.Picture = PantsA.Picture
```

Les 6

Deze les bestaat uit twee onderdelen in het eerste deel ga je kijken naar voorbeelden van klasgenoten en krijg je feedback op je eigen model. Kijk of je n.a.v. deze feedback je eigen gezicht nog wil/kan aanpassen of dat je extra of andere parameters pakt.

In het tweede deel ga je bezig met je model, je hebt nu parameters die invloed hebben op verschillende gezichtsonderdelen. Het is nu de bedoeling om na te denken over 1 variabele die de waarden van al je andere parameters zo aanpast dat je het gezicht in meerdere stappen van boos naar blij kan aanpassen. Bijvoorbeeld -5 is heel boos, en 5 is heel blij, dan moet het gezicht voor alle waarden daar tussen een overgangsuitdrukking hebben.

Les 7, 8 & 9

In de komende drie lessen ga je een applicatie van je gezichtsmodel bouwen. Aan het einde van deze 3 lessen moet je applicatie het volgende kunnen:

- Een gezicht weergeven die uit verschillende onderdelen staat
- Het kunnen aanpassen van alle parameters waardoor je zelf alle mogelijke gezichtsuitdrukkingen kan creëren in het model
- Het implementeren van de laatste parameter die je in les 6 bedacht hebt, om het gezicht in meerdere stappen van blij naar boos te veranderen.

Probeer bij het implementeren van je gezicht zo veel mogelijk gebruik te maken (indien nuttig) van de volgende programmeer concepten:

- Arrays/Lijsten in combinatie met for loops
- Zelf gemaakte functies
- Commentaar toevoegen wat je code doet.

Les 10 & 11: Sentiment analyse

Inleiding

In deze opdracht ga je het sentiment bepalen van een zin: is de zin boos of blij bedoeld, verdrietig of juist enthousiast. Dit doen wij door een sentimentele waarde te geven aan woorden, en vervolgens de waarde van alle woorden in een zin te bepalen en hier het gemiddelde van te nemen. Het sentiment van de woorden is een schaal van 5 tot -5, waarbij 5 extreem positief is en -5 extreem negatief.

Bijvoorbeeld:

Blij: 5

Gewoon: 3

Als je dan de zin: *Ik ben gewoon blij* analyseert, heeft blij een sentiment van 5 en gewoon een sentiment van drie dan is het totale sentiment van de zin $(5 + 3) / 2 = 4$

LETOP voor deze opdracht heb je het bestand “Leerlingen basis – sentiment.docm” nodig.

Sentiment

1. Open het bestand Leerlingen basis - sentiment.docm & ga naar de VBA-editor
2. Het programma bevat nu enkel het raamwerk en geeft altijd als sentiment waarde 1 terug. De bedoeling is om dit raamwerk verder uit te breiden toch een werkend programma. Bevestig dat je nu altijd de melding *Het sentiment is: 1* terugkrijgt, zodra je op de knop drukt.
3. Maak een implementatie van de functie *GetWordList*, deze functie geeft een lijst met woorden terug en de impact op de parameter. Het is handig als de key van de Collection het woord is en de value de sentiment waarde.

Tips:

1. Voor meer informatie over Collections bekijk de volgende pagina's:
- https://excelmacromastery.com/excel-vba-collections/#Adding_Items_Using_a_Key
- [https://msdn.microsoft.com/en-us/library/a1y8b3b3\(v=vs.100\).aspx](https://msdn.microsoft.com/en-us/library/a1y8b3b3(v=vs.100).aspx)
2. Bij een functie moet je aangeven wat de “return value” van een functie is, dat doe je door *GetWordList = ...; (http://stackoverflow.com/a/2781710/1552691)*
4. Implementeer de functie *GetSentimentValue* volgende commentaar die in de functie staat. Zodra je dit gedaan hebt zou je applicatie moeten werken en afhankelijk van je woordlist een sentiment voor de zin tussen 5 en -5 moeten teruggeven.

Tips:

1. Je kan een String splitsen met de functie *Split* ([https://msdn.microsoft.com/en-us/library/6x627e5f\(v=vs.90\).aspx](https://msdn.microsoft.com/en-us/library/6x627e5f(v=vs.90).aspx))
2. Om te weten hoe je de for loop maakt kan je kijken naar de mode-robot.
3. Je kan de reeds bestaande functie *ItemExist* gebruiken om te kijken of een woord in een Collection voor komt
4. Je hebt bij de loop twee variabelen nodig om het gemiddelde sentiment uit te rekenen: 1 voor het aantal gevonden woorden, en 1 voor de totale score.

Geheugen

5. Als je nu meerdere zinnen met tegenovergestelde sentiment invult dan vliegt je sentiment de hele tijd heen en weer. Het is de bedoeling dat je gemoedstoestand een geheugen krijgt, die vervolgens je gezicht langzaam gaat aanpassen.

- a. Maak een variabele sentimentCollection aan in module bereik (<https://support.microsoft.com/nl-nl/help/141693/scope-of-variables-in-visual-basic-for-applications>) waar je een collectie in stopt
- b. Pas je code zo aan dat hij i.p.v. het sentiment teruggeven elke keer het sentiment (indien het sentiment bepaald kan worden) aan die collectie toevoegt
- c. Je wil nu het gemiddelde berekenen van de laatste 5 getallen uit sentimentCollection, en die teruggeven.

Hint 1: Bepaal eerst hoeveel elementen je uit de collectie kan halen en plaats dat getal in een variabele: als er maar 3 dingen in zitten, moet je het gemiddelde van die drie berekenen; als er meer dan 5 dingen in zitten wil je er maximaal 5 uithalen

*Hint 2: Met een for loop ga je over dat aantal heen, en pak je de laatste elementen uit de collection. Dit kan je bereiken met
`sentimentCollection(sentimentCollection.Count - i)`*

Bonus

Indien je klaar bent kan je je sentiment analyse uitbreiden met de volgende punten:

- Maak de analyse hoofdletter ongevoelig
- Zorg dat leestekens niet meegenomen worden
- Zoek uit wat het levenshtein distance algoritme doet, en of je hiermee je analyse rekening kan laten houden met spelfouten.

Les 12: Gezicht laten reageren

6. Breid je eerder gemaakte “gezicht” applicatie nu uit met de sentiment functionaliteit. De bedoeling is dat je een zin kan invullen en afhankelijk van het sentiment van de zin de gezichtsuitdrukking veranderd. Gebruik hiervoor je “blij/boos” parameter.

Extra

We hebben nu een hele simpele sentiment analyse gedaan, er andere en zelfs zelflerende manieren om het sentiment van een zin te bepalen. De volgende site zet verschillende manier uiteen (o.a. de manier die wij hier geïmplementeerd hebben).

<http://ataspinar.com/2015/11/16/text-classification-and-sentiment-analysis/>

Een van de andere manieren die genoemd is, is de: *Naive Bayes Classifier*. De linkjes hieronder lichten deze methode verder toe:

<http://www-nlp.stanford.edu/courses/cs224n/2010/reports/suhaasp.pdf>

<http://ataspinar.com/2016/02/15/sentiment-analysis-with-the-naive-bayes-classifier/>

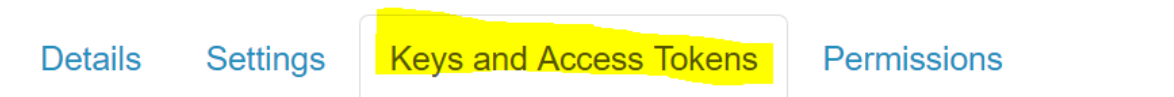
Zoek uit hoe deze manier verschil t.o.v. de manier die wij hebben geïmplementeerd, en schrijf hier een korte uitleg van voor je eindverslag.

Les 13: Twitter API

Met deze opdracht ga je kennis maken met de Twitter API.

Aanmaken API keys

1. Ga naar <https://dev.twitter.com/>
2. Klik op my apps bovenin, je komt op de Twitter Apps page.
 - a. Login met je twitter account of maak een nieuwe aan.
 - b. Klik rechtsboven op de knop "Create new App"
3. We gaan een nieuwe twitter application aanmaken
 - a. Kies een naam (bijv sentiment machine met je naam)
 - b. Vul een korte omschrijving in.
 - c. Vul als website de website van je school in.
 - d. Callback URL kan leeg blijven.
 - e. Zet het vinkje "Yes, i have read" Aan
4. Klik op het tabblad "Keys en access tokens"

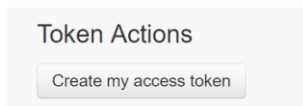


5. Kopieer je customer key en secret en sla deze op!

Consumer Key (API Key) KR1WoL A6rPjnToU9VaUpH

Consumer Secret (API Secret) uFawajSgJXh81f68tEs> .iCn9vubuzY3P9Si FMARgg3

6. Druk vervolgens op create access token:



7. Kopieer je access token en secret!

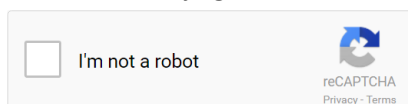
API gebruiken

1. We moeten nu requests doen naar de Twitter API, dit kan bijvoorbeeld met de applicatie Postman, maar we kunnen ook de website <http://www.hurl.it/> gebruiken.

Note: Hurl.it werkt niet goed in firefox

2. Ga naar [hurl.it](http://www.hurl.it/) en vul bij "Destination" het volgende in:
<https://api.twitter.com/1.1/search/tweets.json?q=%40ns> online
Hint: %40 staat voor een @, dit heet URLEncoding.

3. Kies bij Authentication voor OAuth 1.0a
 - a. Vul je tokens en secrets die je bij de eerste stap hebt gemaakt in.
4. Laat weten dat je geen robot bent (<https://www.youtube.com/watch?v=fsF7enQY8ul>)



5. Druk op "Launch request"
6. Scroll naar beneden en kijk welke informatie je allemaal te zien krijgt van tweets.
7. Door de url in het veld destination aan te passen, kan je ook andere zoekopdrachten doen. Bijvoorbeeld door te zoeken naar <https://api.twitter.com/1.1/search/tweets.json?q=ns> krijg je alle tweets waar NS in voor komt

API Opdrachten

Beantwoord de onderstaande vragen door middel van de API. Zet je antwoorden in een word document met: de URL die je hebt aangeroepen; het resultaat dat je kreeg; en hoe je op het antwoord bent gekomen.

Op de volgende pagina's kan je alle mogelijkheden vinden m.b.t. het zoeken van tweets met de twitter API:

- <https://dev.twitter.com/rest/reference/get/search/tweets>
 - <https://dev.twitter.com/rest/public/search>
-
1. Hoeveel tweets stuurt twitter standaard mee met de API? En kan je dit getal aanpassen?
 2. Hoe stuurt de API afbeeldingen in een tweet mee?
 3. Kan je ook zien met welk programma een tweet geplaatst is? Zo ja waar?
 4. Wat zal er in het veld geo staan, en kan je een tweet vinden waar dit gevuld is?
 5. Zal de API alle tweets laten zien? Of zie je bij het zoeken op twitter zelf andere resultaten?
 6. Probeer alle tweets te zoeken die een foto bevatten en de tekst kitten
 7. Haal 10 tweets op die de user *9gag* geplaatst heeft.

Les 14: Twitter Machine

LETOP voor deze opdracht heb je het bestand “Leerlingen basis – twitter.docm” nodig.

Voor deze opdracht krijg je een basis bestand aangeleverd, in dit basis bestand is een library die alle communicatie met Twitter voor je afhandelt. We hoeven ons dus enkel te focussen op het verwerken van de JSON output van de API.

1. Open het bestand Leerling basis - twitter.docm, en ga naar de code van de userform.
2. De code voor de koppeling met Twitter zit er al in, het deel wat je moet gaan maken is om de response van de API te verwerken.
3. Om te beginnen moeten we de Twitter API key's en secrets invullen in de functie *getApiKey()*.

```
ConsumerKey:="XX", _  
ConsumerSecret:="XX", _  
Token:="XX", _  
TokenSecret:="XX"
```

Op de plaats van de XX moet je de tokens invullen die je de vorige les hebt gebruikt om te testen met de twitter API

4. Pas de functie *QueryTwitter* aan zodat je aan twitter naast de *q* ook mee geeft dat je maximaal 5 resultaten wil
5. Je moet nu de implementatie verder bouwen van wat er gebeurt als je op de knop drukt. Hier staat in commentaar al reeds uitgelegd wat de code moet gaan doen.
De code die er al reeds is zet de JSON van twitter om in een Dictionary object, hieronder zie je hoe zo'n dictionary object werkt.
6. Zorg dat het eindresultaat is, dat je ergens op kan zoeken, en dat je de resultaten in een msgbox te zien krijgt.

Dictionary object

Let op! de onderstaande code is enkel om uit te leggen hoe een dictionary werkt en kan dus niet uitgevoerd worden.

```
1 Dim data As Dictionary  
2 Set data = [.....] ' Dictionary op basis van de onderstaande json  
3  
4 ' 1  
5 msgbox(data("results")(1)("id"))  
6 'geeft X999_999 terug  
7  
8 '2  
9 for i = 1 to data("results")(1)("accounts").Count  
10     msgbox (data("results")(1)("accounts")(i)("name"));  
11 next i  
12 ' Geeft twee message boxen, 1 met "Comment" en 1 met "Like"  
13  
14 '3  
15 msgbox(data("results")(2)("from")("name"));  
16 ' Geeft Peyton Manning terug
```

```

19 {
20   "results": [
21     {
22       "id": "X999_Y999",
23       "from": {
24         "name": "Tom Brady", "id": "X12"
25       },
26       "message": "Looking forward to 2010!",
27       "actions": [
28         {
29           "name": "Comment",
30           "link": "http://www.facebook.com/X999/posts/Y999"
31         },
32         {
33           "name": "Like",
34           "link": "http://www.facebook.com/X999/posts/Y999"
35         }
36       ],
37       "type": "status",
38       "created_time": "2010-08-02T21:27:44+0000",
39       "updated_time": "2010-08-02T21:27:44+0000"
40     },
41     {
42       "id": "X998_Y998",
43       "from": {
44         "name": "Peyton Manning", "id": "X18"
45       },
46       "message": "Where's my contract?",
47       "actions": [
48         {
49           "name": "Comment",
50           "link": "http://www.facebook.com/X998/posts/Y998"
51         },
52         {
53           "name": "Like",
54           "link": "http://www.facebook.com/X998/posts/Y998"
55         }
56       ],
57       "type": "status",
58       "created_time": "2010-08-02T21:27:44+0000",
59       "updated_time": "2010-08-02T21:27:44+0000"
60     }
61   ]
62 }

```

Les 15: Opdrachten koppelen

In deze les ga je de code van de twitter machine koppelen aan je gezicht met de sentiment analyse. Het doel is dat je applicatie zodra je op een knop drukt (of met een interval), de laatste tweet van een zoekopdracht ophaalt en hier het sentiment bepaald. Het gezicht moet dan zichzelf automatisch aanpassen naar het sentiment van de tweets.

Les 16 & 17: Eindproduct

In deze twee lessen kan je je applicatie verder afmaken en hier een kort verslag over schrijven.

Als resultaat moet je de volgende dingen inleveren:

1. Een kort verslag
2. Een robot gezicht die reageert op de sentiment van tweets

Verslag

Het verslag moet het volgende bevatten:

1. Voorblad
2. Inhoudsopgave
3. Inleiding wat je gemaakt hebt (0.5p)
4. Uitleg waarom emoties belangrijk kunnen zijn voor een robot (1p)
5. Uitleg over de samenstelling van je gezicht en welke parameters je hebt gekozen voor je gezicht model. Gebruik hierbij de foto's die je gemaakt hebt van je model. (2p incl. foto's)
6. Uitleg hoe sentiment analyse werkt. (0.5p)
7. Antwoorden op de vragen van de twitter opdracht.

Probeer in je verslag zo veel mogelijk gebruik te maken van wordt functionaliteit zoals, automatische inhoud opgaves en standaard kopjes

Note: Voeg alle informatie van de bonus/extra opdrachten die je hebt gedaan ook toe aan het verslag.

VBA Robot

De robot moet in ieder geval de volgende functionaliteit hebben:

Basis

2. Aan de hand van een instelbare parameters moet je de emotie van de gezichtsuitdrukking van de robot in alle mogelijke samenstellingen kunnen krijgen
3. Aan de hand van de instelbare boos/blij parameter, moet je de robot in stappen van boos naar blij jkunnen krijgen
4. Aan de hand van een zin die je zelf intypt en op een knop drukt, moet het gezicht aanpassen n.a.v. het sentiment van de zin
5. Je moet een twitter zoekterm kunnen invullen, als je op een knop drukt moet de robot vervolgens het sentiment van die tweet krijgen.
6. M.b.t. de sentiment waarden, moet de gezichtsuitdrukking van de robot geleidelijk veranderen.

Extra

1. Je kan een timer maken die elke 20 seconden opnieuw naar de tweets zoekt en het sentiment van het gezicht aanpast.
2. De bonus opdrachten gedurende de lessen serie.

Les 18: Toets

De lessenserie wordt afgesloten met een theorie toets, hieronder zie je enkele voorbeeldvragen die aangeven wat voor soort vragen je kan verwachten op de toets. De vragen op de toets zullen echter niet direct uit de eerdere lessen voortkomen, maar andere onderwerpen en api's gebruiken.

Voorbeeldvraag 1

Leg in 5 tot 10 regels uit wat Robotica volgens jou betekend

Voorbeeldvraag 2

Geef enkele voorbeelden van een tekst bericht, waarvan de boodschap compleet veranderd als je er emoji's aan toevoegt (smileys).

Voorbeeldvraag 3

Teken een simpel visueel model van een hoofd om de gezichtsuitdrukking weer te geven, licht je model toe en geef aan geef minstens 2 parameters die je model beïnvloeden

Voorbeeldvraag 4

Kijk naar de volgende JSON, is het tweede bericht na het aanmaken nog een keer aangepast? Let uit waarom wel/niet.

```
19 {
20   "results": [
21     {
22       "id": "X999_Y999",
23       "from": {
24         "name": "Tom Brady", "id": "X12"
25       },
26       "message": "Looking forward to 2010!",
27       "actions": [
28         {
29           "name": "Comment",
30           "link": "http://www.facebook.com/X999/posts/Y999"
31         },
32         {
33           "name": "Like",
34           "link": "http://www.facebook.com/X999/posts/Y999"
35         }
36       ],
37       "type": "status",
38       "created_time": "2010-08-02T21:27:44+0000",
39       "updated_time": "2010-08-02T21:27:44+0000"
40     },
41     {
42       "id": "X998_Y998",
43       "from": {
44         "name": "Peyton Manning", "id": "X18"
45       },
46       "message": "Where's my contract?",
47       "actions": [
48         {
49           "name": "Comment",
50           "link": "http://www.facebook.com/X998/posts/Y998"
51         },
52         {
53           "name": "Like",
54           "link": "http://www.facebook.com/X998/posts/Y998"
55         }
56       ],
57       "type": "status",
58       "created_time": "2010-08-02T21:27:44+0000",
59       "updated_time": "2010-08-02T21:27:44+0000"
60     }
61   ]
62 }
```

Voorbeeldvraag 5

- A) Kijk naar de volgende code, leg uit wat er gebeurt als je op een knop met de naam Randomize drukt:

```
Private Function getParts()  
    Dim parts(6) As CommandButton  
  
    Set parts(0) = LeftArm  
    Set parts(1) = RightArm  
    Set parts(2) = LeftLeg  
    Set parts(3) = RightLeg  
    Set parts(4) = Body  
    Set parts(5) = Head  
  
    getParts = parts  
End Function  
  
Private Sub Randomize_Click()  
    randomColor = getRandomColor()  
  
    Dim parts() As CommandButton  
    parts = getParts()  
  
    For i = 0 To UBound(parts) - 1  
        parts(i).BackColor = randomColor  
    Next i  
End Sub  
  
Private Function getRandomColor()  
    getRandomColor = RGB(Math.Rnd() * 255, Math.Rnd() * 255, Math.Rnd() * 255)  
End Function
```

- B) Wat zouden wij moeten aanpassen, als we niet elke keer een compleet willekeurige kleur willen, maar een kleur uit een vooraf gemaakte lijst? ** Je hoeft niet de exact werkende code te geven, enkel toe te lichten hoe je dit zou maken, en waar je de code dan zou plaatsen.*

c)