

abstract

inleiding

De inspiratie voor dit onderzoek komt oorspronkelijk van een van de installaties op het brusselse evenement "Nuit Blanche". Een van de docenten liet een video zien waarin een connectie wordt gemaakt tussen fysieke componenten en een projectie. In het gesprek dat volgde werd hier al snel een educatief doel aan gekoppeld. Fysieke componenten, een systeem om zogenaamde tags uit te lezen en te volgen, en een narratief. Het doel van het project: kinderen zin geven in programmeren.

Kinderen krijgen in dit project een verhaal aangeboden, in de vorm van een boek. Bepaalde pagina's zijn leeggelaten. Op deze pagina is het de bedoeling dat het kind een stukje code legt om het hoofdpersonage verder te helpen, en zo het verhaal te laten verdergaan.

Doorheen de maanden heeft dit project vele vormen aangenomen. Zo heeft vooral het projectieoppervlak vele vormen gekend, van een tafeloppervlak naar een fysiek boek. Ook zijn de codeblocks (kleine componenten die de kinderen moeten inzetten om een doel te bereiken) van een plexiglazen blokje naar een speelkaartformaat gegaan.

Ik kreeg bij dit project hulp van een aantal personen, waaronder mijn ouders, Wouter Van Den Broeck, Pieter Feys, Zach Lieberman, en Beyond.io.

--> problem --> one page --> i am going to --> bullet list of contributions --> paper provides claim

de onderzoeksvraag

onderzoeksafbakening

doelgroep!

probleemschets

Er zijn reeds een aantal applicaties met het doel kinderen te leren programmeren. Dit gebeurt vaak in de vorm van een spel of kleine opdrachten. Deze applicaties missen helaas vaak een doel, naast het ontwikkelen van een klein spel. Een narratief kan ervoor zorgen dat kinderen langer gemotiveerd zijn, en minder snel afhaken omdat opdrachten op elkaar beginnen lijken. In deze applicaties worden verschillende programeeronderdelen ook duidelijk een voor een aangeboden, in plaats van dit op een minder opvallende manier te doen. Elke opdracht is ook verschillend van de vorige. Het kind doorloopt dus een aantal opdrachten, en weet na de laatste opdracht elk onderdeel correct te gebruiken. Het doel van het narratief is deze onderdelen aan elkaar te binden, zodat het geheel een grote opdracht wordt, zonder het kind "opdrachten" op te leggen.

doel van het onderzoek

Het onderzoek dient duidelijk te maken of een narratief effectief kan bijdragen aan een leukere leerervaring. Dit via een hands-on ervaring, en een empathische band met het hoofdpersonage van het verhaal. Het resultaat van dit onderzoek zal ook een POC (proof of concept) bevatten in de vorm van een applicatie die een extra dimensie geeft aan een verhaal, en het toestaat meerdere verhalen met dezelfde applicatie te kunnen gebruiken.

onderzoeksvragen

centrale vraag

Hoe kunnen we kinderen aanzetten tot coderen in een narratieve omgeving, en aan welke voorwaarden moet de digitale omgeving voldoen om succesvol te zijn.

deelvragen

De onderzoeksvragen laten zich indelen in een aantal categorieën. De eerste categorie bevat vooral vragen omtrend het leerproces van het kind, en welke elementen bij kunnen dragen aan een betere, leukere leerervaring. Het gaat hier dan vooral over de volgende vragen:

- heeft handelend leren een positieve invloed, en hoe kan dit geïntegreerd worden
- heeft empathie effect op leren
- leren in klasverband; welke randvoorwaarden hangen hieraan vast
- op welke manier kunnen kinderen samenwerking in deze applicatie

Verder wordt het aanleren van programmeren aan kinderen in vraag gesteld.

- waarom zou je kinderen leren programmeren
- welke invloeden heeft programmeren op kinderen
- hoe kan dit technisch gerealiseerd worden

Naarmate het onderzoek vorderde, kwamen een aantal andere onderzoeksvragen aan bod. Deze gaan vooral over de functionaliteit van het programma, maar eveneens over de interactie die het programma aanbied tegenover de gebruiker

- welke software en hulpmiddelen zijn reeds beschikbaar
- hoe staat men tegenover het aanleren van programmeren aan kinderen
- zijn er motorische beperkingen bij de doelgroep
- welke stimulus kan programmeren bieden
- hoe kan een kind gemotiveerd worden een doel te bereiken

onderzoeksresultaat

Het vooropgestelde resultaat zal een verhaal bevatten, dat op bepaalde momenten het kind zal vragen een stukje te programmeren alvorens verder te gaan met het verhaal. Dit proces wordt ondersteund met een projectie en de nodige tips. Het aantal stukjes code dat hiervoor gebruikt kunnen worden zal in dit onderzoek beperkt worden. Het programma zal gebruik

maken van empathie en andere technieken om het kind te motiveren.

onderzoek

programmeren

In 2014 gaf de EU een rapport uit dat het integreren van programmeren in het lager onderwijs bespreekt. Het gaat hier vooral over landen die dit reeds doen (zoals Estland) waar eveneens over landen die dit van plan zijn in de nabije toekomst (waaronder België). Ook het nut van programmeren in een schoolomgeving komt hierin aan bod, met het oog op de toekomst van zowel de economie (meer programmeurs), maar belangrijker nog, met het oog op het kind.

Mitch Resnick (MIT, medeoprichter van Scratch) vergelijkt programmeren met schrijven; slechts weinig van de kinderen worden professioneel schrijver, maar kunnen schrijven is steeds handig. Hij vindt dat dit ook geldt voor code. Men moet niets computergerelateerd kiezen als beroep om de opgedane kennis te gebruiken. Programmeren stimuleert volgens Mitch het creatief denken, het systematisch redeneren en het samenwerken in teamverband. Verder kan het kind ideeën naar buiten brengen op deze manier.

Dit inzicht komt terug bij Derek Cabrera. Hij beschrijft in zijn TED-talk dat sommige kinderen het creatief denken verleerd hebben. Hij geeft hierbij het voorbeeld van Lego. Dit werd vroeger verkocht in dozen, zonder instructiehandleiding, terwijl deze momenteel vooral verkocht wordt in dozen waarmee men enkel de handleiding kan volgen, doordat de blokjes slechts 1 enkel model toelaten. Volgens hem verleren kinderen hierdoor het creatief denkproces (metacognitief denken). Dit soort denken houdt in dat men gemakkelijk nieuwe problemen kan oplossen, en ongestructureerde problemen kan structureren en er iets zinvol mee kan doen.

21st century skills stelt dat onze maatschappij van een industriële samenleving naar een kennis-samenleving veranderd is. Hun doel is het onderwijs aan te passen om beter aan te sluiten bij dat beeld van de 21ste eeuw. De Europese Unie is hier reeds mee bezig, samen met bv UNESCO en een aantal Amerikaanse organisaties. Nederlandse beleidsmakers zijn hier zeker reeds in betrokken in de context waarin ze werken. Deze discussie wordt helaas nog niet helemaal doorgetrokken naar het onderwijs. Deze skills zouden even belangrijk moeten zijn dan wiskunde of Nederlands. Joke Voogt is voorstander van een publiekdebat die deze discussie opentrekt waarin ook ouders en studenten hun mening kunnen uiten.

digitale samenleving

De digitalisering van de samenleving is enorm merkbaar in de tewerkstelling. Het aantal beroepen waarin productiewerk vericht wordt, daalt snel, en de beroepen waarin men creatief moet nadenken en probleemoplossend moet werken, zit in de lift. In die laatste categorie is een degelijke IT-kennis natuurlijk noodzakelijk. Vooral voor communicatie, maar eveneens om de efficiëntie te verhogen. Daarom is het belangrijk dat we dit al van

jongsafaan kunnen meegeven.

21st century skills

De term 21st century skills heeft betrekking op een geheel van vaardigheden die nodig zijn te functioneren in de huidige samenleving. Deze samenleving wordt onderverdeeld in een aantal groepen. De belangrijkste groepen zijn "digital natives" en "digital immigrants".

De digital natives zijn in dit onderzoek de belangrijkste doelgroep. Zij zijn mee opgegroeid met de technologie, en ontdekken nieuwe ontwikkelingen en toepassingen, volledig onbevangen. Deze groep heeft weinig tot geen moeilijkheden met het gebruiken van nieuwe technologie, en zijn hier snel mee weg.

Tot de groep van "digital immigrants" behoren de volwassenen die de technologie hebben zien groeien. Sommigen zijn snel weg met vernieuwing, anderen twijfelen eerder alvorens deze te omarmen. Ook deze groep gebruikers leert omgaan met de nieuwere technologieën, maar hebben soms meer moeite deze vlot te integreren.

Het grote verschil tussen deze twee groepen is het gemak waarmee men overschakeld. Dit is vooral merkbaar in de manier van interactie. Terwijl de immigrants vooral vasthouden aan mail en formelere communicatie (ook face-to-face), maakt de groep van digital natives gebruik van sociale media zoals whats-app en facebook, en schakeld zeer snel over wanneer een nieuwe vorm beschikbaar is (vb: snapchat).

21st century skills worden onderverdeeld in een aantal vaardigheden. Deze worden vaak ondersteund door een degelijk gebruik van ICT. We spreken hier bijvoorbeeld over vaardigheden zoals:

- samenwerking
- kennisconstructie
- ICT gebruik voor leren
- probleemoplossend denken en creativiteit
- planmatig werken

De nadruk in dit onderzoek zal liggen op het probleemoplossend denken en creativiteit. Dit omdat deze vaardigheid gestimuleerd wordt door programmeeractiviteiten en de denkwijze die hiermee gepaard gaat.

Algoritmisch denken

Algoritmisch denken (ook beschreven als algoritmisch denken) betekend letterlijk "denken over denken" (meta=over, cognos=denken). Algoritmisch denken wordt door de EU omschreven als "de mogelijkheid een probleem op te lossen via een stap-voor-stap oplossing (algoritme), waarin deze bestaat uit een eindige set duidelijk gedefinieerde stappen".

Dit soort denken wordt vaak gebruikt tijdens het programmeren. Het komt erop neer dat de programmeur (in dit geval ook de gebruiker) een probleem dient te verdelen in set kleinere deelproblemen, om daarna deze deelproblemen op te kunnen lossen via een stappenplan.

Men moet hierbij natuurlijk ook rekening houden wat de beginsituatie is, en het einddoel.

Het aanleren van deze vaardigheid gebeurt reeds in de alternatieve programma's. Onder andere Scratch leert dit aan door gebruik te maken van een serie uitdagingen die het kind moet doorlopen. Tijdens dit proces leert hij het probleem op te delen in kleine acties. Nadat de gebruiker alle stappen heeft doorlopen, kan deze aan de slag met alle blokjes die Scratch bevat, om zelf creaties te maken.

Scratch kan een opstap zijn naar programma's als java, visual basic of javascript, zonder de leerling al direct te confronteren met het zoeken naar bugs als komma's of aanhalingstekens. Dit kan namelijk een grote drempel vormen, omdat een programmeur veel tijd doorbrengt een code te debuggen. Het werken met voorgemaakte blokjes neemt deze drempel weg.

programming

Men kan het natuurlijk niet hebben over programmeren zonder de meest pure vorm te bespreken. Programmeren kent vele definities. de volgende is diegene die de EU gebruikt in zijn rapport:

“Het kunnen realiseren van een algoritme in machinetaal, de stappen die genomen zijn kunnen interpreteren, [...] het kunnen compileren, runnen en debuggen van een programma, alsook het kunnen identificeren en herbruiken van designpatterns”

Dit soort programmeren heeft een stijle leercurve. Het duurt even vooraleer men een eigenlijke output heeft, en het aanleren van de syntax neemt de nodige tijd in beslag. Kinderen kunnen hierdoor gedemotiveerd geraken, en haken af als gevolg. Daarom wordt in dit project gebruik gemaakt van een abstractere vorm hiervan. Een vorm waarin de syntax niet zichtbaar is voor de gebruiker, en deze ook niet aan de nodige haakjes, komma's of punt-komma's moet denken.

Programmeren laat zich onderverdelen in een aantal categorieën.

algoritmisch denken

dit omvat het programmeren zonder effectief code te schrijven. Enkel standaard acties zijn hierin opgenomen. Variabelen en andere structuren komen hierin niet aan bod. Het oplossen van een probleem (meestal het bereiken van een bepaalde plek op het speelveld) kan hier opgelost worden door een opeenvolging van stappen zoals "stap naar rechts", "stap naar boven", en meer van dit soort blokken. Zoals eerder vermeld is deze vorm zeer geschikt voor jonge kinderen. Ze kunnen hier snel mee aan de slag en zien zeer snel resultaat. Dit is echter wel een ideale manier om snel en duidelijk een eerste probleem onder te verdelen in kleine stappen, en de drempel naar het volgende niveau is zeer laag. Voor ouderen is dit echter niet heel uitdagend, omdat de gelijkheid met het volgende level vaak te groot is en het als bandwerk aanvoelt.

metaforisch programmeren

In deze stap (die meestal volgt op de vorige) komen de eerste programmeerstructuren naar

boven. Kinderen en volwassenen maken kennis met een "for loop", en een "if else" structuur. Ook variabelen komen hier aan bod. De gemakkelijkste manier om deze vorm te gebruiken is door de code onder te verdelen in kant en klare blokjes. Deze kan de gebruiker dan aaneenschakelen en een groter geheel vormen. In veel applicaties kan deze vorm voorgesteld worden als een efficiënter alternatief voor algoritmisch denken.

flow programming

Flow programming wordt voorgesteld in programmas zoals pure data. De gebruiker verbind blokjes die een actie of voorwaarde voorstellen met elkaar, om zo het programma uit te voeren. Deze vorm laat echter slechts moeilijk de implementatie van programmeerstructuren toe.

programming

Dit is de meest bekende vorm van programmeren. In deze vorm is de syntax van groot belang, en vele programmeertalen maken gebruik van deze vorm van programmeren. De nadruk ligt in deze vorm op het correct aanroepen van de juiste functies, en er moet vooraf heel hard nagedacht worden over de manier van implementatie. Deze vorm neemt ook de nodige tijd in beslag.

in conclusion

Programmeren leert het kind op een andere manier nadenken over een probleem. Het bevordert het opdelen van een groot probleem in kleinere deelproblemen, en vlotter naar een oplossing te zoeken. Het algoritmisch nadenken staat het kind ook toe bepaalde acties die herhaald moeten worden misschien efficiënter uit te voeren.

storytelling

Om volledig te kunnen begrijpen wat een narratief kan betekenen voor een gebruiker, en hoe we dit kunnen gebruiken, moeten we natuurlijk eerst definiëren hoe een verhaal in elkaar zit.

oorsprong van verhalen

Verhalen zijn doorheen de tijd vaak veranderd. Ze zijn introverter geworden (mensen lezen verhalen zelf, zonder ze aan anderen te vertellen), zijn zijn van mondelinge vorm naar een geschreven vorm gegaan. Doorheen de jaren zijn er ook meer en meer geworden.

We vertellen al verhalen sinds het begin van mensenheugenis, denk maar aan de rotstekeningen in Frankrijk. In die tijd werden verhalen vooral verteld, om geschiedenis door te geven, of fenomenen te verklaren. Met de komst van een geschreven taal zijn we deze verhalen ook beginnen neerschrijven. Het is echter pas sinds de uitvinding van de boekdrukkunst dat we deze in geschreven vorm ook beginnen verspreiden zijn. De tijd tussen deze eerste vooruitgangen spreid zich echter over duizenden jaren. Pas met de komst van de film is het echt snel beginnen gaan. Na film kwam radio en tv, waarna computers en het internet het rijtje voegden om verhalen door te geven.

Elke van deze uitvindingen brachten met zich mee dat artiesten een nieuwe manier hadden om verhalen te vertellen, maar ook hun verhalen aan te passen naar het gebruikte medium. Denk maar aan de uitvinding van de stop-motion truck in de filmwereld, waarmee het plots mogelijk werd verhalen anders vorm te geven, en andere verhalen te vertellen. Acties moesten niet meer mogelijk zijn, men kon truckeren.

In 2005 kwam youtube uit. Dit is een van de grotere milestones in het vertellen van verhalen. Het werd mogelijk voor iedereen met een camera (elk soort camera) eigen verhalen en films te verspreiden. Dit in combinatie met het goedkoper worden van de filmcamera en smartphones zorgt voor een enorme boost in het aantal en de kwaliteit van verspreide films, en dus ook verhalen.

Nieuwe technologie gaat ook vaak gepaard met een nieuwe vorm van verhaalvertelling. Denk maar aan Vines, een van de recentere vormen hiervan. een online platform waarin de gebruiker een luttel 7 seconden kreeg om zijn punt, of verhaal, in duidelijk te maken.

story en plot

David Bordwell: "een chronologische volgorde van events" -> story "volgorde en duratie van evenementen" -> plot

narrative units

conclusions and further work