

Fundamenten van de Mens Machine Interactie (G0Q55a)

Projectverslag

LAURENS CLEEMPUT
SANDER CLEYMANS
MATHIAS DEKEMPENEER
ROBIN HAVENEERS

2017-2018

KU LEUVEN

Introductie

De applicatie die wij ontwikkeld hebben voor het vak **Fundamenten van de Mens-Machine Interactie** kreeg de naam ***QuickMaths***. In dit rapport wordt de tweede iteratie van het digitaal prototype, het *“in-the-wild”*-experiment en de evaluaties ervan besproken. In Appendix A en Appendix B kunnen respectievelijk de evaluaties van de eerste en van de tweede iteratie geraadpleegd worden. In Appendix C en Appendix D kan de code die gebruikt werd om data-verwerking en analyse uit te voeren, teruggevonden worden.

De applicatie is te downloaden op de Google Play Store via <https://play.google.com/store/apps/details?id=cs.hciproject.quickmaths&hl=nl> en op de App Store via <https://itunes.apple.com/WebObjects/MZStore.woa/wa/viewSoftware?id=1326512406&mt=8>.

1 Derde iteratie: tweede digitale prototype

1.1 Design keuzes

Uit de *user study* van het eerste digitale prototype bleek dat het initiële design reeds intuïtief was maar dat de start van het spel nog steeds een probleem bleek te zijn. Het scherm met de opgave en de countdown bleek ook verwarrend te zijn, zeker omdat dit scherm snel opgevolgd wordt door het volledige speelveld. De rest van de gebruikersinterface bleek vanaf de start van het spel duidelijk. Om de start van het spel duidelijker te maken, hebben we enkele aanpassingen doorgevoerd, zowel aan het scherm met de countdown als aan het speelveld zelf.

Aanpassingen aan het scherm met de opgave en countdown zijn:

- De animatie met virtuele hand is verwijderd.
- De opgave zelf is groter afgebeeld en staat centraal in het scherm, zonder verdere afleidingen.
- De countdown is beter zichtbaar (meer contrast met de achtergrond).
- Tijdens de countdown speelt er een geluid (3-2-1-Start) om de tekst te begeleiden.

Aanpassingen aan het scherm met het speelveld en de speelbalk zijn:

- De speelbalk (waar de invoer moet gebeuren) is één rij naar boven toe verschoven. Dit heeft als voordeel dat de balk meer in het oog springt door het contrast met de rest van het speelveld. Hierdoor wordt het ook beter duidelijk dat het speelveld van boven naar onder verschuift.
- Onder de speelbalk verschijnt de vorige lijn. De antwoorden worden ingekleurd: een juist antwoord is groen, een fout antwoord is rood.
- De lijn onderaan die elke vijf seconden gevuld wordt is vanaf nu geel gekleurd in plaats van het minder contrasterende blauw. Deze lijn komt nu beter in het zicht. Spelers die reeds een antwoord hebben ingevuld en wachten op de volgende lijn weten dat het spel niet vasthangt, maar dat ze gewoon moeten wachten.

In deze iteratie introduceren we geluid: er zijn geluiden voor de countdown, voor het voltooien van een level en voor het maken van een fout. Hierdoor wordt het algemene verloop van de app duidelijker.

Tijdens het spelen van een level speelt er nu ook achtergrondmuziek. Deze muziek draagt bij aan de *arcade-like feel* van de game. Het is een progressief nummer: naarmate men verder geraakt in een level, wordt de muziek intenser. Enerzijds geeft dit een impliciete indicatie van hoe lang het level nog zal duren en anderzijds draagt het bij tot een opbouwende spanning (“Je bent er bijna!”). Bij het starten van het volgende level begint het nummer weer van het begin te spelen. Hoe hoger het level, hoe langer de muziek speelt. Voor sommige spelers kan dit een motivatie zijn om zo ver mogelijk te willen spelen om te ontdekken wat de volgende sequentie in het nummer is.

1.2 Deelnemers

Aan de evaluatie van deze derde iteratie hebben zes leerlingen en één leerkracht wiskunde van het Heilig Hartinstituut te Heverlee deelgenomen. In Tabel 1 kunnen de demografische gegevens over de deelnemers van deze gebruikersstudie van de tweede iteratie geraadpleegd worden.

	Leeftijd	Studierichting
Leerling 1	13	Latijn
Leerling 2	13	Latijn
Leerling 3	14	Latijn
Leerling 4	12	Moderne
Leerling 5	12	Moderne
Leerling 6	12	Moderne
Leerkracht	43	/

Tabel 1: Demografie deelnemers derde iteratie

1.3 Evaluatie methodes

Voor de evaluatie van de tweede iteratie van het digitale prototype hebben we opnieuw geopteerd voor een think-aloud study. Vermits we reeds face-to-face staan met de deelnemers lijkt het ons meer waardevol om een kwalitatieve analyse van deze iteratie te kunnen doen. Deze iteratie wilden we vooral te weten komen of de veranderingen die we hebben doorgevoerd aan de countdown en de inrichting van het speelveld ervoor zorgen dat spelers vanaf het eerste spel weten wat ze moeten doen om een level te voltooien.

De spelers krijgen deze iteratie dezelfde taak als de vorige iteratie¹: voltooi de eerste twee levels. Ook bij deze test kregen de spelers in eerste instantie de tutorial niet te zien.

¹zie Appendix B

1.4 Resultaten

1.4.1 Gebruikersstudie bij de leerlingen

Tijdens de gebruikersstudie van de derde iteratie werd er bijgehouden hoeveel pogingen de leerlingen nodig hadden om het level te voltooien. In Figuur 1 staan de resultaten van deze gebruikersstudie. Zoals te zien in deze figuur, slaagt slechts 43%² van de leerlingen er in om het eerste level van de eerste maal succesvol te beëindigen. Wanneer er de mogelijkheid aangeboden wordt om het eerste level voor een tweede maal te proberen, slaagt 71%³ van de leerlingen er in om naar het volgende level te gaan. Wanneer de eerste poging van het eerste level niet succesvol werd voltooid, werd er aan de leerling een tutorial van de applicatie getoond. Dit kan de stijging van het slaagpercentage in het eerste level verklaren daar er nu gedemonstreerd wordt waar er geklikt moet worden.

De twee leerlingen die er niet in slaagden om het eerste level te voltooien, konden niet starten met het volgende level. Van de leerlingen die het tweede level bereikten, slaagde 40% er in om het level meteen te voltooien. Bij een tweede poging zal 60% van de resterende leerlingen aan het derde level kunnen beginnen.

De resultaten van deze gebruikersstudie kunnen geraadpleegd worden de samenvattende Figuur 1.

Tijdens en na de think-aloud study gaven de proefpersonen nog informele feedback:

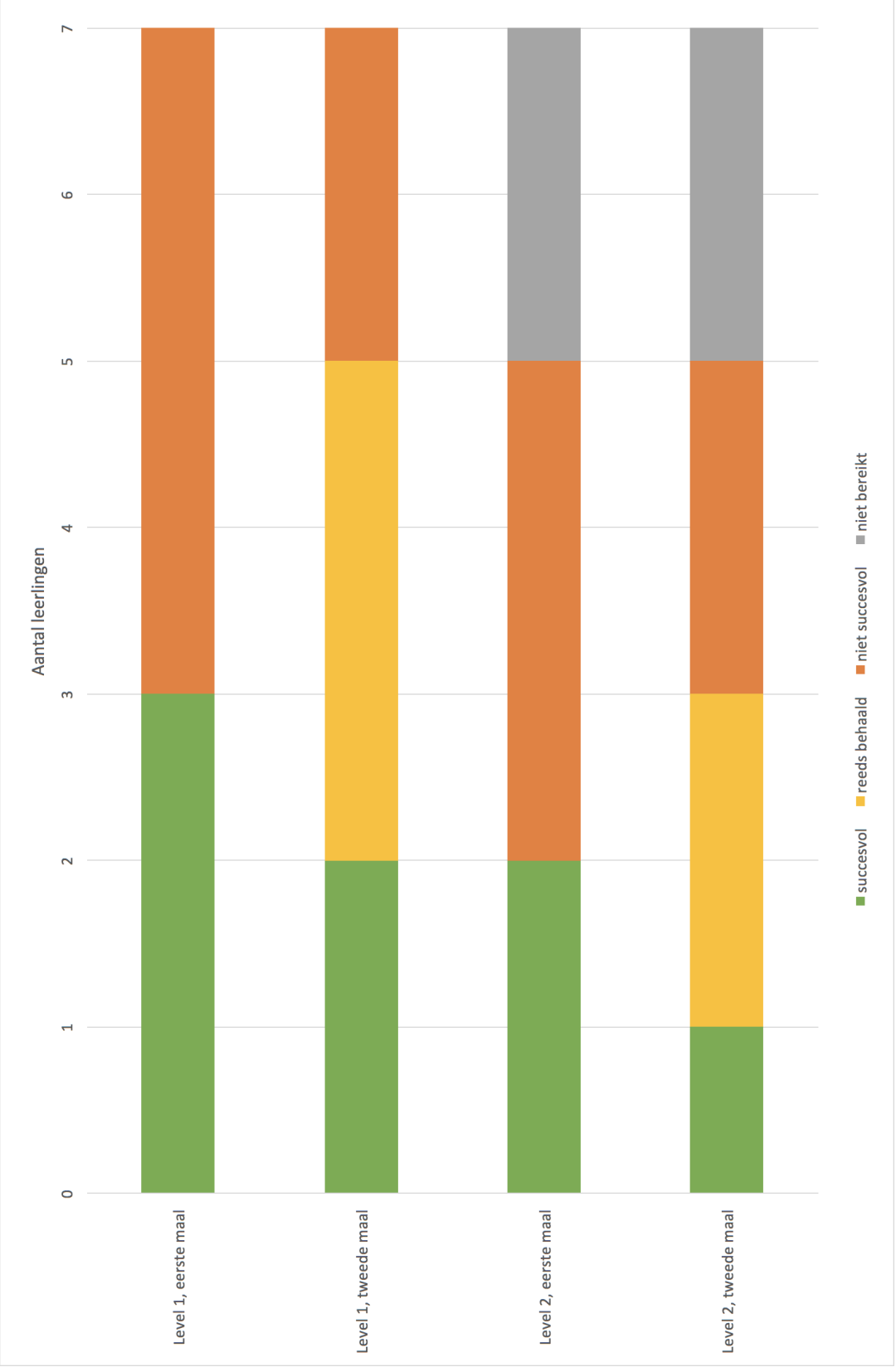
- Vijf leerlingen maakten de fout om een antwoord aan te duiden op een lijn waarbij er geen juist antwoord was. Dit is niet bepaald verrassend vermits ze de tutorial niet op voorhand te zien kregen. De aanwezigheid van lijnen zonder juist antwoord was een bewuste designkeuze: Indien elke lijn wel een juist antwoord zou hebben en de eerste twee opgaven in een lijn blijken beide fout zijn, dan is de derde opgave automatisch correct. Dit leek ons minder waardevol te zijn vanuit een educatief perspectief. Voor de leerkracht was de inclusie van deze lijnen eveneens geen probleem.
- Eén leerling vond de onderste “feedbackrij” geen waardevolle toevoeging. Hij gaf aan dat de rij zijn aandacht vroeg en daardoor afleidend werkte. Vermits deze klacht slechts één keer vermeld werd, hebben we hiervoor geen veranderingen doorgevoerd in het design.
- Eén leerling had bij de eerste poging een fout antwoord aangeduid en was zich daar meteen bewust van. Hij wist echter niet dat hij zijn antwoord nog kon verbeteren alvorens de tijd op was.
- Voor één leerling was de timer (vijf seconden) duidelijk te snel ingesteld. De eerste levels bleken voor haar reeds te moeilijk. Ze gaf zelf aan dat wiskunde niet haar beste vak was en ze was mogelijk wat onder de indruk van onze proefopstelling.

1.4.2 Enquête bij de leerlingen

Tijdens de gebruikersstudies in de tweede en de derde iteratie werd er aan de leerlingen gevraagd om een aantal spelletjes te spelen van *QuickMaths*. Nadat ze deze spelletjes gespeeld hadden, werd

²3 van de 7 leerlingen

³5 van de 7 leerlingen



Figuur 1: Resultaten gebruikersstudie derde iteratie

er hen gevraagd een korte enquête in te vullen. De resultaten van deze enquête en de evolutie tussen de twee iteraties heen worden in deze subsectie besproken.

De eerste vraag in de enquête ging over het gevoel van de leerling na het gebruik van de app. In Figuur 2 worden de resultaten van deze vraag voorgesteld. In de tweede iteratie voelt 80% van de deelnemers zich ‘goed’ na het gebruik van de app. In de iteratie die erop volgt, voelt 67% van de deelnemers zich ‘redelijk goed’ tot ‘goed’.

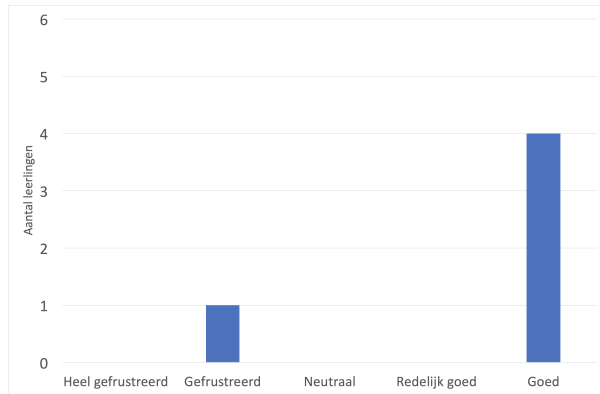
De volgende vraag in de enquête polst naar de duidelijkheid van het spel dat in *QuickMaths* gespeeld moet worden. Zoals in Figuur 3 vastgesteld kan worden, vindt de meerderheid van de leerlingen het doel van het spel ‘duidelijk’ tot ‘heel duidelijk’. Enkel in de derde iteratie zijn er twee leerlingen die het doel van het spel ‘onduidelijk’ vonden.

Eén van onze doelstellingen bij het ontwikkelen van *QuickMaths* is dat je het spel kan spelen zonder veel instructies op voorhand te moeten krijgen. Daarvoor stelden we aan de leerlingen de vraag of de manier waarop je het spel moet spelen duidelijk was van in het begin. In Figuur 4 kunnen we vaststellen dat het voor de leerlingen toch niet altijd duidelijk was hoe ze het spel moesten spelen. Dit probleem werd opgelost door het toevoegen van een kleine tutorial die aan de speler toont waar je precies moet klikken in de applicatie.

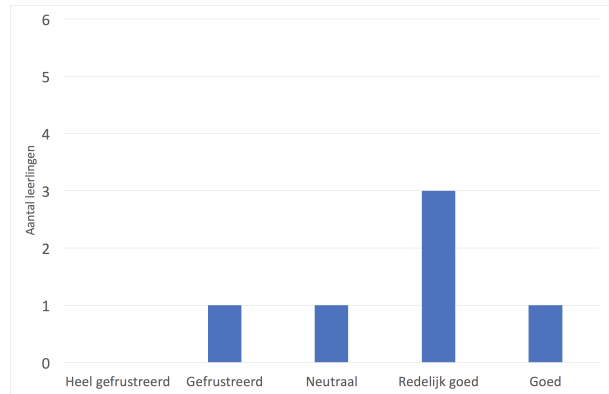
Met *QuickMaths* willen we bereiken dat de leerlingen van de eerste graad middelbare school op regelmatige basis het hoofdrekenen in wiskunde oefenen. Daarom stelden wij de vraag hoe vaak de leerlingen de app zouden willen gebruiken. Zoals we in Figuur 5 kunnen vaststellen, zijn de leerlingen gemotiveerd om de app van meerdere keren per maand tot meerdere keren per week te gebruiken.

Een volgende vraag polste naar de moeilijkheid van het *QuickMaths*-spel. De leerlingen kregen de opdracht om de app een score te geven tussen 1 en 10, wat respectievelijk ‘heel gemakkelijk’ en ‘heel moeilijk’ voorstelt. De resultaten van deze vraag staan voorgesteld in Figuur 6. In de tweede iteratie gaven de leerlingen gemiddeld een score van 4.8 voor de moeilijkheid van het spel. De mediaan is er 5.0. In de derde iteratie gaven de leerlingen gemiddeld een score van 3.5 voor de moeilijkheid van het spel. De mediaan is er 3.0. We kunnen hier vaststellen dat de leerlingen tussen de twee iteraties door het spel gemakkelijker vonden dankzij de aanpassingen die doorgevoerd werden in de applicatie. Zo werd onder andere de balk waar de antwoorden aangeduid moeten worden een rij naar boven geschoven. Dit maakte het spel eenvoudiger voor de leerlingen.

In de derde iteratie voegden we achtergrond- en spelgeluiden toe aan *QuickMaths*. Ook hier wilden wij van de leerlingen weten wat hun gevoel was bij deze geluiden. Bij de achtergrondmuziek (Figuur 7a) kunnen we concluderen dat het merendeel van de studenten deze muziek ‘motiverend’ vinden. Slechts één leerling van de zes vond deze muziek ‘storend’. Bij de spelgeluiden (Figuur 7b) kunnen we geen conclusies maken. Het merendeel van de studenten heeft geen mening over de spelgeluiden, de andere opties zijn evenredig verdeeld.

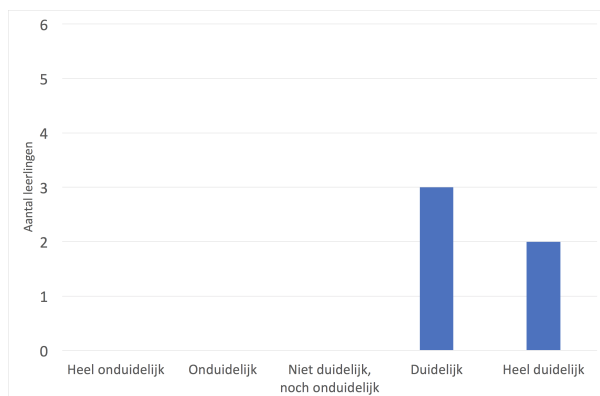


(a) tweede iteratie

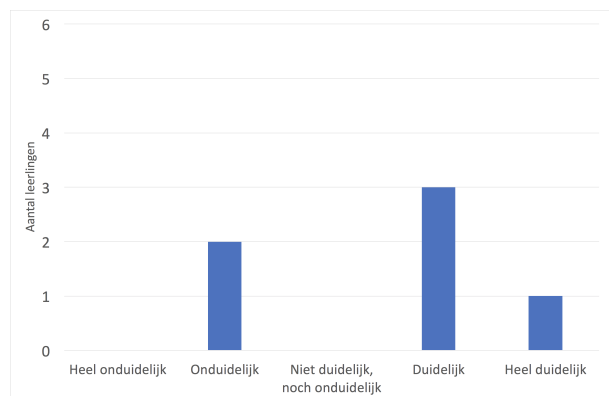


(b) derde iteratie

Figuur 2: Hoe voel je jezelf na het gebruik van de app?

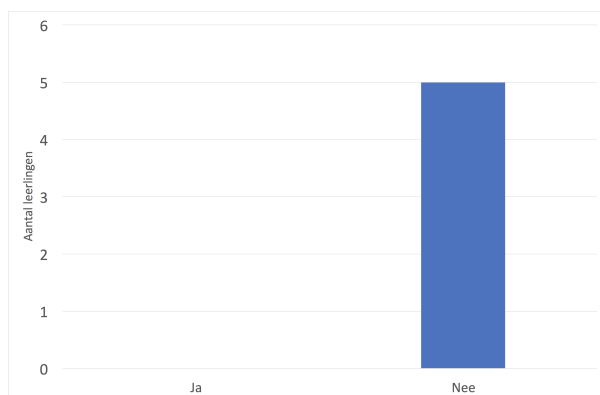


(a) tweede iteratie

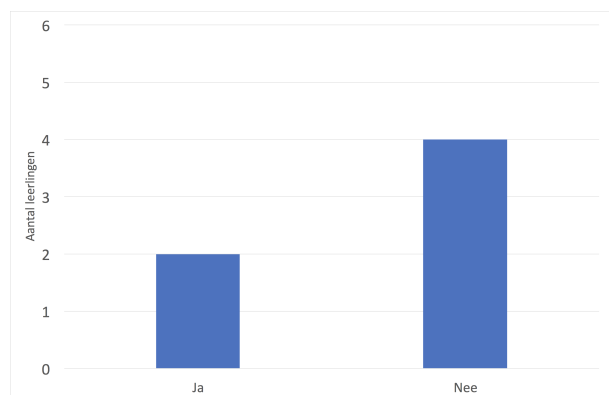


(b) derde iteratie

Figuur 3: Hoe duidelijk was het doel van het spel?

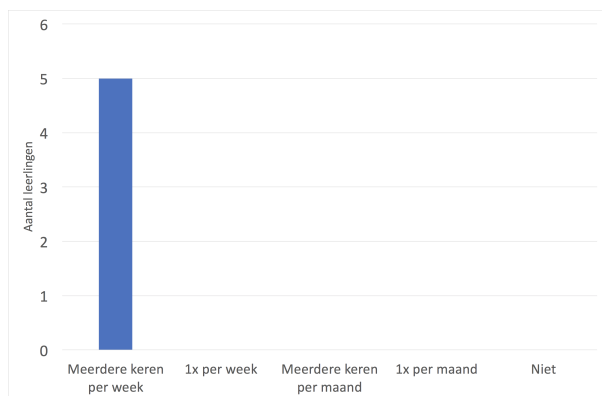


(a) tweede iteratie

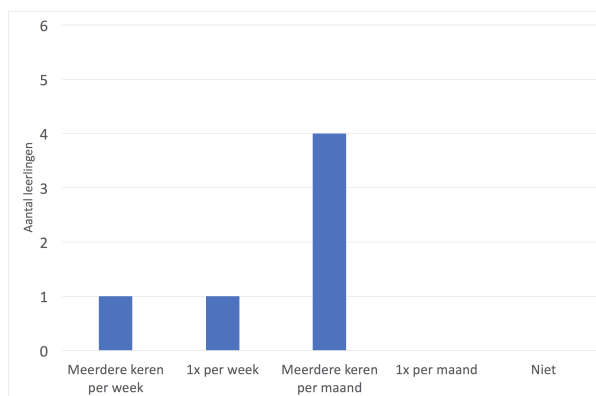


(b) derde iteratie

Figuur 4: Was de manier waarop je moet spelen vanaf het begin duidelijk?

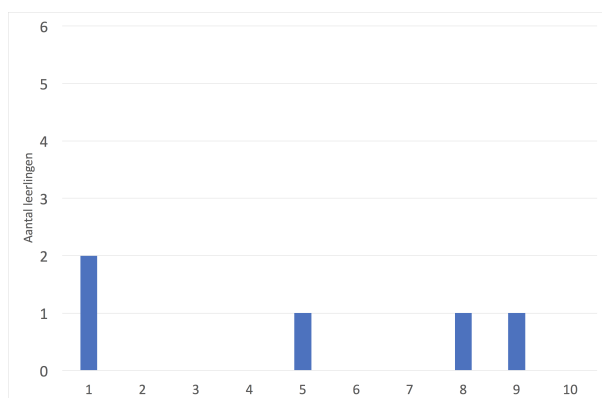


(a) tweede iteratie

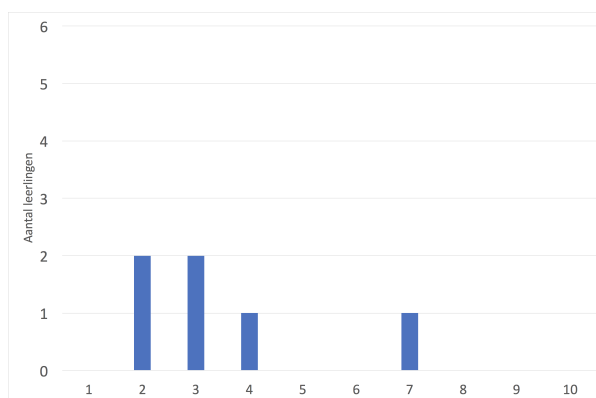


(b) derde iteratie

Figuur 5: Hoe vaak zou je het spel willen spelen?

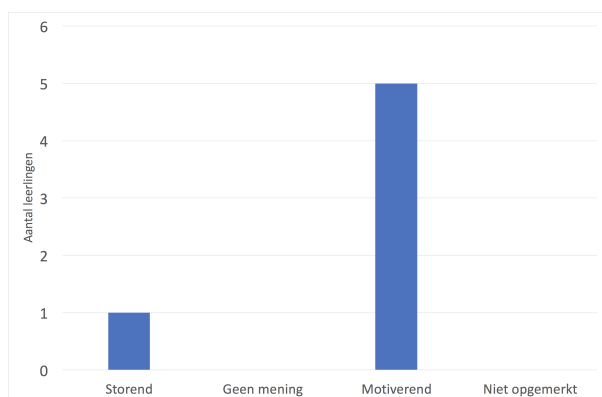


(a) tweede iteratie

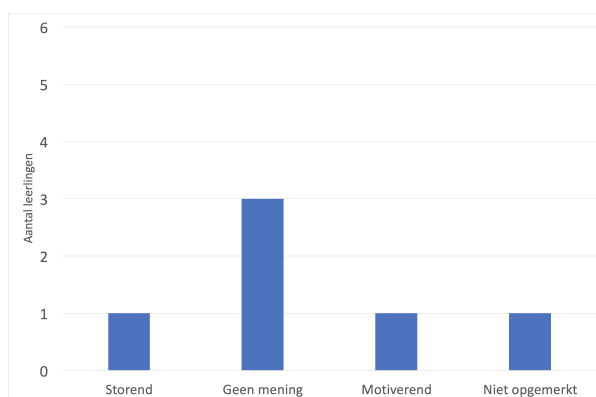


(b) derde iteratie

Figuur 6: Wat vond je van de moeilijkheid van het spel?



(a) achtergrondmuziek



(b) spelgeluiden

Figuur 7: Wat vond je van de ...?

2 Laatste iteratie: publicatie op Google Play Store & App Store

QuickMaths kan op Android geïnstalleerd worden via <https://play.google.com/store/apps/details?id=cs.hciproject.quickmaths> en op iOS via <https://itunes.apple.com/WebObjects/MZStore.woa/wa/viewSoftware?id=1326512406&mt=8>.

2.1 Design keuzes

Tijdens een afsluitend gesprek met de leerkracht tijdens het testen van het tweede digitale prototype stelden we de vraag of het relevant zou zijn om feedback te geven aan de speler na het maken van een fout. Ze stelde voor om deze functionaliteit toe te voegen, maar zonder al te veel didactische uitleg. Bij het maken van een fout krijgt de speler te zien wat zijn fout was en wat de gepaste actie zou geweest zijn.

Voor het *“in the wild”* experiment hebben we twee verschillende gamemodi ontworpen die we met elkaar willen vergelijken. Bij het openen van de app wordt er achter de schermen bepaald welke gamemode er gebruikt wordt. De verschillende modi zijn:

- **Classic mode:** De lengte van de levels neemt logaritmisch toe en convergeert naar een maximale lengte van 20 rijen. De getallen in het speelveld worden groter volgens het level maal factor 6.
- **Endurance mode:** De lengte van de levels neemt linearitmisch toe. De getallen in het speelveld worden wel minder snel groter: de toename is lineair volgens het level maal factor 4.

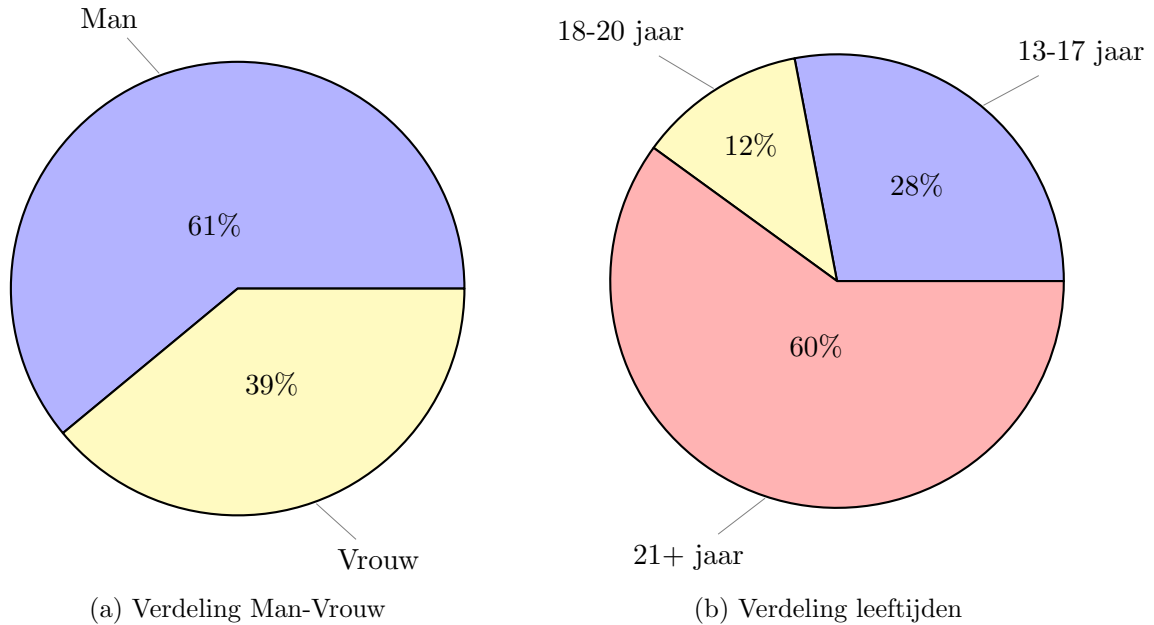
Om de resultaten van deze verschillende modi te kunnen loggen maken we gebruik van een Firebase database. Om de unieke gebruikers te kunnen identificeren maken we gebruik van een Facebook login. Spelers krijgen nog steeds de optie om te spelen zonder Facebook login, maar van deze gebruikers krijgen we geen spelstatistieken. De data die we bijhouden via logging omvat onder meer het aantal gespeelde spellen, het resultaat van het level (“gewonnen” of “verloren”), indien verloren welke fout gemaakt werd, welke modus gespeeld is ...

Bij deze finale versie introduceren we naast levels met sommen, verschillen en deelbaarheid ook levels met vermenigvuldigingen.

2.2 Deelnemers

Aan de evaluatie van deze laatste iteratie hebben in totaal 57 personen deelgenomen. In Figuur 8a kan de verdeling van het aantal mannen ten opzichte van het aantal vrouwen vastgesteld worden.

Figuur 8b toont de verdeling van de verschillende leeftijdscategorieën die wij dankzij de login van Facebook kregen. Zoals vast te stellen in deze figuur, kwamen 28% van de deelnemers uit onze doelgroep. Dit is vooral te wijten aan het feit dat deze doelgroep op het moment van deze evaluatie in een examenperiode zit.



Figuur 8: Demografische gegevens van de deelnemers

2.3 Evaluatie methodes

Bij deze kwantitatieve analyse willen we beide gamemodi vergelijken met als centrale onderzoeksvraag:

Wat is de dominante factor die de moeilijkheidsgraad van een level bepaalt: de lengte van het level of de grootte van de getallen?

We baseren ons hiervoor op de verhouding van het aantal voltooide levels tot het totaal aantal gespeelde levels.

Op basis van de gelogde gegevens kunnen we eveneens te weten komen welke modus de hoogste engagement heeft. Dit zouden we willen testen aan de hand van de hoeveelheid herkansingen van een level alvorens er wordt opgegeven (terugkeren naar hoofdmenu).

2.4 Resultaten

We hebben voor drie eigenschappen een descriptieve en inferentiële analyse gemaakt met de twee gamemodes als onafhankelijke veranderlijke:

- Failrate: de verhouding van het aantal mislukte levels tot het totaal aantal gespeelde levels
- Hoogst behaalde level: hoogste level per speler
- Totale speelduur: de totale speelduur per speler

Voor elke van deze eigenschappen beschrijven we een tabel met de gemiddelden en standaardafwijkingen (respectievelijk Tabel 2, Tabel 4 en Tabel 6) en een boxplot die de verdeling duidelijk

maakt (respectievelijk Figuur 9, Figuur 10 en Figuur 11). Voor de inferentiële analyse beschrijven we voor elke eigenschap een ANOVA-tabel.

2.4.1 Fail Rates

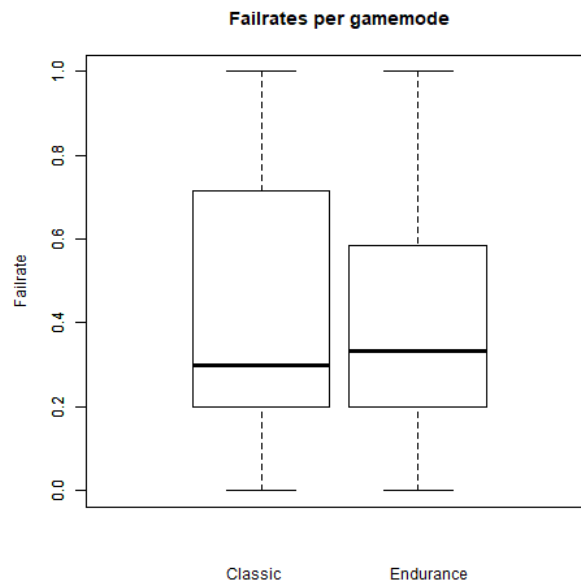
Uit de ANOVA-analyse in Tabel 3 blijkt dat de gebruikte gamemode geen significante invloed heeft op de failrate ($p > 0.05$).

gamemode	MeanFail	SDFail
Classic	42.67%	28.21%
Endurance	41.25%	29.01%

Tabel 2: Descriptieve analyse: failrates

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
gamemode	1	0.003	0.00309	0.038	0.846
Residuals	60	4.904	0.08173		

Tabel 3: ANOVA-analyse: failrates



Figuur 9: Boxplot met de failrates per gamemode per deelnemer.

2.4.2 Hoogst behaalde level

Uit de ANOVA-analyse in Tabel 5 blijkt dat de gebruikte gamemode geen significante invloed heeft op het hoogst behaalde level ($p > 0.05$).

gamemode	MeanLevel	SDLevel
Classic	5.03	3.46
Endurance	4.96	2.75

Tabel 4: Descriptieve analyse: hoogst behaalde level

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
gamemode	1	0.1	0.072	0.007	0.933
Residuals	54	539.9	9.999		

Tabel 5: ANOVA-analyse: hoogst behaalde levels

2.4.3 Totale speelduur

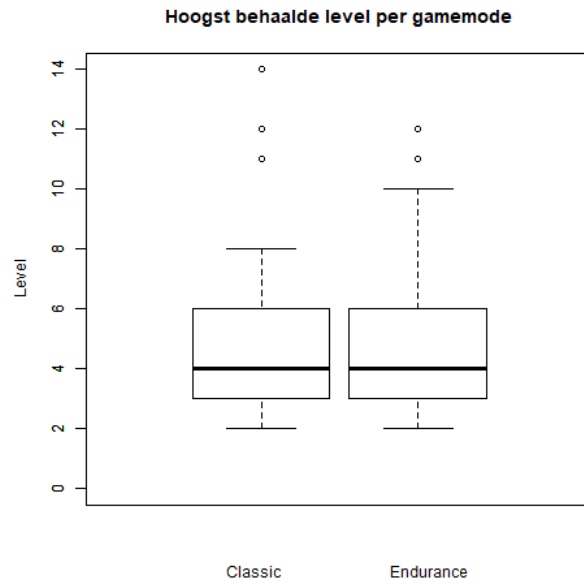
Uit de ANOVA-analyse in Tabel 7 blijkt dat de gebruikte gamemode geen significante invloed heeft op de totaal gespeelde tijd ($p > 0.05$). Hiermee wordt ook uitgesloten dat de verschillende gamemodes significant verschillende engagement rates zouden hebben.

gamemode	MeanTimes	SDTimes
Classic	370.34	487.43
Endurance	250.53	314.65

Tabel 6: Descriptieve analyse: totale speelduur

2.4.4 Bespreking van de resultaten

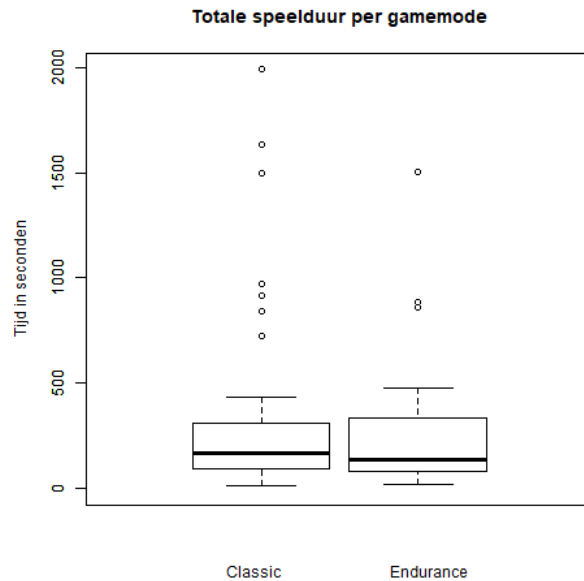
Uit bovenstaande resultaten blijkt duidelijk dat er geen significante verschillen zijn tussen de twee gamemodes. We kunnen dus geen uitspraak doen over de onderzoeksvraag. Een kanttekening die we hierbij willen maken is dat de hoeveelheid data uit de doelgroep zelf beperkt was. Bovendien worden de verschillen tussen beide gamemodes pas duidelijk bij het bereiken van hogere levels (vanaf level 6). In praktijk hebben de meeste spelers ongeveer dezelfde gamemode gespeeld (hoogst behaalde level is kleiner dan 6), het is bijgevolg normaal dat er geen significante verschillen werden opgemeten.



Figuur 10: Boxplot met het hoogst behaalde level per gamemode per deelnemer.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
gamemode	1	0.1	0.072	0.007	0.933
Residuals	54	539.9	9.999		

Tabel 7: ANOVA-analyse: totale speelduur



Figuur 11: Boxplot met de totale speelduur per gamemode per deelnemer.

3 Eindresultaat

Doorheen de ontwikkeling van de app hebben we gemerkt dat het niet evident is om een app te ontwikkelen die voor iedereen duidelijk is. Doorheen zowat alle iteraties werden ettelijke keren volledig tegenovergestelde meningen met ons gedeeld, waaruit we dan telkens het “nuttige” van het “nutteloze” moesten proberen te onderscheiden. Vanaf de eerste ontwerpbeslissingen werd dan ook elke finale beslissing meerdere keren gewikt en gewogen. Het uiteindelijke resultaat heeft ons het nut van meerdere iteraties, prototypes en een user-centered design bijgebracht. Als laatste punt bij onze app en het vak hebben we nog enkele bevindingen:

- Gebruikers gaan vaak te snel door tutorials en missen zo essentiële informatie.
- Ontwerpkeuzes die triviaal leken, waren voor sommige gebruikers volledig onbegrepen.
- Moeilijkheid is een heel moeilijk te balanceren gegeven.
- Naast gebruikers, is feedback met experts ook van essentieel belang.

Appendix

A Eerste iteratie: papieren prototype

A.1 Design keuzes

Het eerste prototype is volledig gemaakt uit papier en karton. Het dynamische, tijdsgebonden design van onze app maakt het niet makkelijk om dit op papier al te laten werken. In het uiteindelijke ontwerp hebben we gekozen voor een kartonnen gsm, met een doorschuifstelsel van papier voor ons speelveld⁴. Hierdoor kunnen we direct de verloop van onze app volledig testen van beginscherm tot “Game Over” of “Level completion”.

Het ontwerp van de schermen is heel simpel gehouden. Het beginscherm laat slechts enkele simpele knoppen zien, met vooral pictogrammen.

De verschillende knoppen zijn onder andere:

- Start knop, om het spel te starten,
- Highscores, om de beste scores weer te geven,
- Profiel, om de gegevens van de huidige speler te bekijken,
- Instellingen, om de spelinstellingen aan te passen.

Het profielscherm laat de gebruiker relevante informatie zien over:

- Zichzelf
- “Player status”
- Zijn achievements (met een knop om ze te bekijken)
- Top 3 beste scores
- Daarnaast is er de mogelijkheid om van gebruiker te wisselen.

Het speelscherm was het ingewikkeldste deel van het ontwerp. We laten het speelveld bewegen door de “computer”-persoon van onze studie. Met ons bewegend speelveld konden we zo goed als het volledige spelverloop al testen. Het ontwerp hiervan is heel simpel. Eerst laten we de vraag zien waarover de speler moet gaan nadenken. Na een countdown tonen we het effectieve speelveld.

Afhankelijk van hoe goed de speler het doet, komt er een “Level klaar”- of “Game over”-scherm. Het “Level klaar”-scherm laat de gebruiker toe om verder te gaan naar het volgende (moeilijkere) level, of om terug te keren naar het beginscherm. “Game over” laat toe om het level opnieuw te proberen of om terug te keren naar het beginscherm.

⁴Foto's en afbeeldingen van het ontwerp zijn te vinden op <https://quickmathsblog.wordpress.com/2017/10/24/paper-prototype/>

A.2 Deelnemers

De doelgroep van onze applicatie zijn de leerlingen van de eerste graad middelbare school. Hiervoor hebben wij contact opgenomen met het Heilig Hartinstituut te Heverlee⁵. Aangezien het ons wat moeite gekost heeft om deze te bereiken en afspraken te maken, hebben we voor de eerste iteratie een drietal studenten op het departement Computerwetenschappen gevraagd om deel te nemen aan deze gebruikersstudie.

A.3 Evaluatie methodes

Voor de eerste evaluatie van dit papieren prototype hebben we gebruik gemaakt van een think-aloud study. We hebben met deze eerste evaluatie al vroeg in het ontwerpproces enkele tekortkomingen geïdentificeerd. Dit was waardevol vermits deze tekortkomingen reeds duidelijk werden op basis van het papieren prototype. Er werd rekening gehouden met de opmerkingen bij het ontwerp van het eerste digitale prototype. De deelnemers werden gevraagd om een level te spelen tot voltooiing, en om vervolgens hun profielpagina te bekijken.

A.4 Resultaten

De voornaamste bevindingen worden beschreven in Tabel 8.

	Begin van spel	Waar antwoord kiezen	Overblijvende tijd
Speler A	Pas duidelijk na uitleg	Pas duidelijk na uitleg	Geen problemen
Speler B	Meteen duidelijk	Duidelijk	Verwarrend
Speler C	Meteen duidelijk	Duidelijk	Verwarrend

Tabel 8: Usability issues van papieren prototype

Speler A start het spel, de opgave en countdown volgen elkaar op en het speelveld verschijnt. Speler A overschouwt het speelveld en begint cellen aan te duiden, de speler raakt verward en geeft na een tiental seconden op. De observer geeft een tip en verwijst naar de roodgekleurde speelbalk. De speler was vergeten dat 0 deelbaar is door 7. Na deze tips is speler A erin geslaagd om het level te voltooien. Vervolgens vindt de speler zijn weg naar de profielpagina via het home scherm, en vindt hij zijn status: “Speed Junky”. Om de verwarring bij de start van het spel op te lossen, zal er in het vervolg bij de start van het spel een lege lijn in de speelbalk verschijnen die meteen doorgeschoven wordt met de eerste “echte lijn”.

Speler B weet meteen waar de inputs geregistreerd worden (roze balk). De speler doorloopt de eerste lijnen, maar maakt vervolgens een fout door een antwoord aan te duiden op een lijn waar geen juiste antwoorden zijn. Deze verwarring zou verholpen kunnen worden door deze mechanic in de korte tutorial uit te leggen. Het game-over scherm wordt getoond en de speler vindt zijn weg naar de profielpagina. De speler had wel nog een opmerking dat het tijdens het spel niet duidelijk is hoeveel tijd er is om een antwoord te geven. In het digitale prototype zullen we de pauze knop (onderaan het scherm) gebruiken om de overblijvende tijd per lijn te tonen als een balk die gevuld wordt van links naar rechts.

⁵<https://www.hhh.be/home>

Speler C begint aan het spel en merkt de speelbalk meteen op. Halfweg het level slaagt de speler er niet in om op tijd een antwoord te geven, en eindigt het spel. De speler verwijst net als speler B naar een nood om de overblijvende tijd te kunnen zien. De speler vindt eveneens makkelijk de weg naar de profielpagina om de player status te bekijken.

Tijdens de study kwamen we samen tot inzicht dat er op alle schermen buiten het home screen er nog een “Keer terug”-knop ontbreekt. We brengen deze feedback in rekening bij de ontwikkeling van het digitale prototype.

B Tweede iteratie: eerste digitale prototype

B.1 Design keuzes

Het digitale prototype is qua ontwerp zeer gelijkend op het papieren prototype⁶.

Bij het opstarten van het spel wordt de speler begroet door het “beginscherm”. Na te drukken op de startknop, krijgt de gebruiker een korte tutorial. In deze tutorial krijgt de gebruiker in een viertal afbeeldingen uitgelegd hoe het spel werkt en wat hij precies moet doen. Hierna kan hij verder gaan naar het menu.

Het menu geeft de gebruiker vier toetsen om te selecteren: “Play”, “Highscores”, “Profile” en “Settings”. Twee van de knoppen staan centraal en zijn anders gekleurd, hiermee trekken we de aandacht naar de belangrijkste features van de app.

Bij de start van een level worden de opgave, een countdown en een voorbeeld van hoe het spel werkt getoond. Na de countdown begint het spel. Tijdens het spel weergeven we enkele belangrijke elementen op het scherm:

- Links bovenaan: de opgave,
- Rechts bovenaan: het resterend aantal lijnen en het huidige level,
- Midden op scherm: de volgende rijen. Deze zijn lichtjes geblurred om de aandacht meer te vestigen op de speelbalk, onderaan.
- Onderaan: de speelbalk met de deadline onderaan. Deze deadline geeft de resterende tijd voor deze lijn in de speelbalk aan.

In de loop van het spel worden er steeds nieuwe lijnen aan de gebruiker voorgeschoteld. De knoppen worden gehighlight aan de hand van de input. Indien de gebruiker het juiste antwoord heeft aangeduid (of niets, als er geen juiste oplossing was), dan gaat het spel verder en krijgt hij een volgende lijn te zien. Zolang de deadline van deze lijn niet verstreken is, kan hij zijn antwoord nog aanpassen. Indien het fout ging, tonen we het “Game Over” scherm.

Het “Game Over” scherm laat toe om terug te keren naar het menu of het huidige level terug te proberen met een andere opgave. Is de gebruiker geslaagd voor alle lijnen van het level, dan gaat hij naar het “Level voltooid” scherm. Hierin krijgt hij de keuze om verder te gaan naar het volgende level of terug te keren naar menu. Bij het terugkeren wordt de gebruiker gewaarschuwd dat hij zijn huidige vooruitgang zal verliezen.

In het digitale prototype zijn er nog geen andere features ondersteund. Het profiel bestaat uit een dummy scherm dat een mogelijke indeling geeft voor de latere implementaties.

⁶Foto's en afbeeldingen van deze versie te vinden op: <https://quickmathsblog.wordpress.com/2017/11/13/digitale-prototype-eerste-versie/>

B.2 Deelnemers

Aan de evaluatie van deze iteratie hebben vier leerlingen en één leerkracht wiskunde van het Heilig Hartinstituut te Heverlee deelgenomen. In Tabel 9 kunnen de demografische gegevens over de deelnemers van deze gebruikersstudie van de tweede iteratie teruggevonden worden.

	leeftijd	studierichting
leerling 1	12	Latijn
leerling 2	12	Latijn
leerling 3	11	Latijn
leerling 4	12	Latijn
leerkracht	43	

Tabel 9: Demografie deelnemers tweede iteratie

B.3 Evaluatie methodes

Voor de evaluatie van dit eerste digitale prototype doen we een think-aloud study met deelnemers uit onze doelgroep. De deelnemers worden gevraagd om de eerste twee levels te voltooien, en om vervolgens hun profielpagina te bekijken. Het think-aloud aspect van deze studie is vooral interessant wanneer de gebruiker een fout maakt of wanneer de gebruiker op verkenning gaat in de app. Tijdens het spelen van het spel zelf, ondervinden ze dat het erg moeilijk is om tegelijkertijd luidop te denken en de oefeningen op te lossen.

B.4 Resultaten

We vroegen aan elke deelnemer om level 1 te voltooien. Zodra dit lukte, werden ze gevraagd om verder te spelen totdat ze level 3 voltooid hadden. Mits enkele tussenkomsten van de moderator, slaagde iedereen er in om deze taken binnen aanvaardbare tijd te voltooien.

Tijdens het uitvoeren van deze taken, stelden zich enkele problemen. Deze problemen worden beschreven en geëvalueerd, inclusief een mogelijke oplossing. Hiervoor verwijzen we naar onderstaande tabel. In de volgende iteratie(s) zullen deze problemen geadresseerd worden. We proberen daarna opnieuw testen te doen met onze doelgroep.

Als laatste punt geven we nog enkele opmerkelijke resultaten mee van onze enquête:

- Eén van de testpersonen bleek licht gefrustreerd na onze testen. Opmerkelijk was dat deze persoon wel wilde meewerken in verdere testen.
- De moeilijkheid van ons spel bleek voor iedereen ook anders: twee personen vonden het makkelijk tot gemiddeld, twee personen vonden het heel moeilijk.
- De leerlingen wilden het spel gebruiken om vlotter te worden in basis hoofdrekenen.
- Alle deelnemers willen de app meerdere keren per week spelen.

	Leeftijd	Geslacht	Start van het spel	Lege vakjes	Rijen zonder oplossing
Persoon 1	12	V	Ik snapte niet direct waar dat ik op moest klikken.	Verwarring over selecteren: juist of fout?	Lichte verwarring, maar geen probleem
Persoon 2	12	M	De tijd en meerdere keuze vragen	Geen probleem	Wilde steeds iets selecteren, zelfs foute oplossingen
Persoon 3	11	M	Ik wist niet hoe ik het moest doen.	Twijfelde lichtjes, maar maakte geen fout	Geen probleem
Persoon 4	12	M	Ik wist niet waar ik moest drukken	Geen probleem	Geen probleem
Persoon 5	43	V	Verwarring, geen idee wat te doen na countdown.	Geen probleem	Geen probleem
Mogelijke oplossingen:			<ul style="list-style-type: none"> - Intro moet minder druk worden. - Countdown moet opvallender. - Meer contrast tussen speelveld en -balk. - Statische tutorial maakt al veel duidelijk. 	Geen lege vakjes. Vermelden in tutorial.	Maakt inherent deel uit van de uitdaging. Valstrik, extra uitdaging.
Algemene opmerking:			Er werd geen tutorial getoond aan de proefpersonen.		

Tabel 10: Resultaten think-aloud study paper prototype 1

C Dataparser - Code

```
import json
import numpy as np
import csv
import urllib
import numpy as np
import pylab

def main():
    url = "https://quickmaths-baf21.firebaseio.com/.json?auth=b0gfYH0MZCMzHEw2MtRs98eWCOWKwa6f5zG6hSyy"
    response = urllib.urlopen(url)
    data = json.loads(response.read())
    users = data["users"]

    total_games_played = 0
    endurance_counter = 0
    classic_counter = 0
    younglings = []
    midlings = []
    elderlings = []
    endurance_game_times = np.zeros((100,), dtype=np.int)
    classic_game_times = np.zeros((100,), dtype=np.int)
    younglings_games_played = 0
    midlings_games_played = 0
    elderlings_games_played = 0
    solo_game_players = 0
    males = 0
    male_games_played = 0
    females = 0
    female_games_played = 0
    classic_games_won = 0
    endurance_games_won = 0
    most_games_played = 0
    highest_endurance_level_male = 0
    highest_endurance_level_female = 0

    highest_classic_level_male = 0
    highest_classic_level_female = 0

    endurance_fails = 0
    classic_fails = 0
    player_count = 0

    user_max_level_classic = 0
    user_max_level_endurance = 0

    user_fails_classic = 0
    user_played_classic = 0
    user_fails_endurance = 0
    user_played_endurance = 0

    user_gamemode_highest_level = csv.writer(open('highestLevels.csv', 'w'),
        delimiter=',', quotechar = '|', quoting = csv.QUOTE_MINIMAL)
```

```

user_gamemode_highest_level.writerow(['userID', 'gamemode', 'level'])

user_fail_rate = csv.writer(open('failRate.csv', 'w'), delimiter=',',
    quotechar = '|', quoting = csv.QUOTE_MINIMAL)
user_fail_rate.writerow(['userID', 'gamemode', 'failrate'])

user_total_time = csv.writer(open('totalTime.csv', 'w'), delimiter=',',
    quotechar = '|', quoting = csv.QUOTE_MINIMAL)
user_total_time.writerow(['userID', 'gamemode', 'totaltime'])

ios_users = 0
android_users = 0

interruptions = 0

for user_str in users:

    user_max_level_classic = 0
    user_max_level_endurance = 0

    user_fails_classic = 0
    user_played_classic = 0
    user_fails_endurance = 0
    user_played_endurance = 0

    player_count += 1
    total_games_played += users[user_str]["nb_games_played"]

    # check for solo-game players
    if users[user_str]["nb_games_played"] == 1:
        solo_game_players += 1

    if users[user_str]["nb_games_played"] > most_games_played:
        most_games_played = users[user_str]["nb_games_played"]

    #check for age
    if users[user_str]["info"]["age"] < 18 :
        younglings.append(user_str)
    elif users[user_str]["info"]["age"] < 21:
        midlings.append(user_str)
    else:
        elderlings.append(user_str)

    #check for gender
    if users[user_str]["info"]["gender"] == "male":
        males += 1
    else:
        females += 1

    if users[user_str]["info"]["device_info"]["os"] == "ios":
        ios_users += 1

    if users[user_str]["info"]["device_info"]["os"] == "android":
        android_users += 1

```

```

# GET GAME TIME PER MODE
for game_tag in users[user_str]["games"]:
    if users[user_str]["games"][game_tag]["interrupted"] == True:
        interruptions += 1
        total_games_played -= 1

    if users[user_str]["games"][game_tag]["interrupted"] == False:
        if users[user_str]["info"]["age"] < 18:
            younglings_games_played += 1
        elif users[user_str]["info"]["age"] < 21:
            midlings_games_played += 1
        else:
            elderlings_games_played += 1

        if users[user_str]["info"]["gender"] == "male":
            male_games_played += 1
        else:
            female_games_played += 1

        game_type = users[user_str]["games"][game_tag]["type"]

        if game_type == "Endurance":
            user_played_endurance += 1
            endurance_counter += 1
            start_time = get_sec(users[user_str]["games"][game_tag]["start_time"][:8])

            if users[user_str]["games"][game_tag]["end_time"] != 0:
                end_time = get_sec(users[user_str]["games"][game_tag]["end_time"][:8])
                total_time = end_time - start_time
                if abs(total_time) < 1000:
                    endurance_game_times[player_count] += total_time

            if users[user_str]["games"][game_tag]["result"] == "Won":
                endurance_games_won += 1
            if users[user_str]["games"][game_tag]["result"] == "GameOver":
                user_fails_endurance += 1

            if (users[user_str]["info"]["gender"] == "male" and
                users[user_str]["games"][game_tag]["level"] >
                highest_endurance_level_male):
                highest_endurance_level_male = users[user_str]["games"][game_tag]["level"]

            if (users[user_str]["info"]["gender"] == "female" and
                users[user_str]["games"][game_tag]["level"] >
                highest_endurance_level_female):
                highest_endurance_level_female = users[user_str]["games"][game_tag]["level"]

        if users[user_str]["games"][game_tag]["level"] >
            user_max_level_endurance:
            user_max_level_endurance = users[user_str]["games"][game_tag]["level"]

```

```

else:
    user_played_classic += 1
    classic_counter += 1
    start_time = get_sec(users[user_str]["games"][game_tag]["start_time"][:8])
    if users[user_str]["games"][game_tag]["end_time"] != 0:
        end_time = get_sec(users[user_str]["games"][game_tag]["end_time"][:8])
        total_time = end_time - start_time
        if abs(total_time) < 1000:
            classic_game_times[player_count] += total_time

    if users[user_str]["games"][game_tag]["result"] == "Won":
        classic_games_won += 1
    if users[user_str]["games"][game_tag]["result"] == "GameOver":
        user_fails_classic += 1

    if (users[user_str]["info"]["gender"] == "male" and
        users[user_str]["games"][game_tag]["level"] >
        highest_classic_level_male):
        highest_classic_level_male = users[user_str]["games"][game_tag]["level"]

    if (users[user_str]["info"]["gender"] == "female" and
        users[user_str]["games"][game_tag]["level"] >
        highest_classic_level_female):
        highest_classic_level_female = users[user_str]["games"][game_tag]["level"]
    if users[user_str]["games"][game_tag]["level"] >
        user_max_level_classic:
        user_max_level_classic = users[user_str]["games"][game_tag]["level"]

if user_max_level_classic > 1:
    user_gamemode_highest_level.writerow([user_str, 'Classic',
        user_max_level_classic])

if user_max_level_endurance > 1:
    user_gamemode_highest_level.writerow([user_str, 'Endurance',
        user_max_level_endurance])

if user_played_classic > 1:
    user_fail_rate_classic = float(user_fails_classic) / float(
        user_played_classic)
    user_fail_rate.writerow([user_str, 'Classic', user_fail_rate_classic])

if user_played_endurance > 1:
    user_fail_rate_endurance = float(user_fails_endurance) / float(
        user_played_endurance)
    user_fail_rate.writerow([user_str, 'Endurance',
        user_fail_rate_endurance])

if classic_game_times[player_count] > 0:
    user_total_time.writerow([user_str, 'Classic', classic_game_times[
        player_count]])

```



```

        if endurance_game_times[player_count] > 0:
            user_total_time.writerow([user_str, 'Endurance', endurance_game_times[
                player_count]])

youngling_verhouding = float(younglings_games_played) / float(
    total_games_played)
midling_verhouding = float(midlings_games_played) / float(total_games_played)
elderling_verhouding = float(elderlings_games_played) / float(
    total_games_played)

total_players = len(younglings)+len(midlings)+len(elderlings)

print("Verhouding spelers 13-17 jaar:\t" + str((float(len(younglings))/float(
    total_players))*100)+" %")
print("Verhouding spelers 18-20 jaar:\t" + str((float(len(midlings))/float(
    total_players))*100)+" %")
print("Verhouding spelers 21+ jaar:\t" + str((float(len(elderlings))/float(
    total_players))*100)+" %")
print("")

print("Verhouding spelers 13-17 jaar gespeeld: " + str(youngling_verhouding
    *100)+" %")
print("Verhouding spelers 18-20 jaar gespeeld: " + str(midling_verhouding*100)
    +" %")
print("Verhouding spelers 21+ jaar gespeeld:\t" + str(elderling_verhouding
    *100)+" %")
print("")

print("Endurance levels: " + str(endurance_counter) + " gespeeld en " + str(
    endurance_games_won) + " werden gewonnen. \t(" + str((float(
    endurance_games_won)/float(endurance_counter))*100) + " %)")
print("Hoogst behaalde Endurance level: " + str(highest_endurance_level_male)
    + " (male)")
print("Hoogst behaalde Endurance level: " + str(highest_endurance_level_female
    ) + " (female)")
print("")

print("Classic levels: " + str(classic_counter) + " gespeeld en " + str(
    classic_games_won) + " werden gewonnen. \t(" + str((float(classic_games_won
    )/float(classic_counter))*100) + " %)")
print("Hoogste behaalde Classic level: " + str(highest_classic_level_male) + "
    (male)")
print("Hoogste behaalde Classic level: " + str(highest_classic_level_female) +
    " (female)")
print("")

print(str(solo_game_players) + " mensen hebben slechts 1 level gespeeld.")
print("Het meeste levels dat iemand speelde is: " + str(most_games_played) + "
    .")
print("")

print(str(males+females) + " spelers hebben " + str(total_games_played) + "
    levels gespeeld,\tdus " + str((float(total_games_played))/(float(males+
    females)))) + " per persoon")
print(str(males) + " mannen hebben " + str(male_games_played) + " levels
    gespeeld,\tdus " + str(float(male_games_played)/float(males)) + " per

```

```

        persoon.")
    print(str(females) + " vrouwen hebben " + str(female_games_played) + " levels
        gespeeld,\tdus " + str(float(female_games_played)/float(females)) + " per
        persoon.")
    print("")

    print("iOS gebruikers:\t\t" + str(ios_users))
    print("Android gebruikers:\t" + str(android_users))
    print("")

    print("Interruptions:\t" + str(interruptions))

def get_sec(time_str):
    h, m, s = time_str.split(':')
    return int(h) * 3600 + int(m) * 60 + int(s)

if __name__ == "__main__":
    main()

```

D Analyse - Code

input/QMMarkdown.Rmd

```
---
title: "QM"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown

```{r}
library(dplyr)
library(ez)

setwd("D:/Laurens/Documents/(School/CW 2/1e Semester/FMMI/R Project)")

Loading file
failData <- read.csv("failRate.csv")
levelData <- read.csv("highestLevels.csv")
timeData <- read.csv("totalTime.csv")

Group data by gamemode
failGrouped <- group_by(failData, gamemode)

levelGrouped <- group_by(levelData, gamemode)

timeGrouped <- group_by(timeData, gamemode)

Summary
summarise(failGrouped, MeanFail=mean(failrate), SDFail=sd(failrate))

summarise(levelGrouped, MeanLevel=mean(level), SDLevel=sd(level))

summarise(timeGrouped, MeanLevel=mean(totaltime), SDLevel=sd(totaltime))

Boxplots

Failrates
png(filename="D:/Laurens/Documents/(School/CW 2/1e Semester/FMMI/R Project/Graphs
/Failrates.png")
failClassic <- filter(failData, gamemode == "Classic")
failBox1 <- select(failClassic, failrate)
boxplot(failBox1, at=1, xlim=c(0,3), boxwex=1.75, ylim=c(0,1), ylab="Failrate",
 main="Failrates per gamemode", xlab="Classic
 ")

failEndurance <- filter(failData, gamemode == "Endurance")
failBox2 <- select(failEndurance, failrate)
boxplot(failBox2, at=2, add=TRUE, boxwex=1.75, ylim=c(0,1), xlab="
 Endurance")

dev.off()
```

```

Levels
png(filename="D:/Laurens/Documents/(School/CW 2/1e Semester/FMMI/R Project/Graphs
/Levels.png")
levelClassic <- filter(levelData, gamemode == "Classic")
levelBox1 <- select(levelClassic, level)
boxplot(levelBox1, at=1, xlim=c(0,3), boxwex=1.75, ylim=c(0,max(levelBox1,
levelBox2)), ylab="Level", main="Hoogst behaalde level per gamemode", xlab="
Classic
")

levelEndurance <- filter(levelData, gamemode == "Endurance")
levelBox2 <- select(levelEndurance, level)
boxplot(levelBox2, at=2, add=TRUE, boxwex=1.75, ylim=c(0,max(levelBox1, levelBox2)
), xlab="
Endurance")
dev.off()

Total time played
png(filename="D:/Laurens/Documents/(School/CW 2/1e Semester/FMMI/R Project/Graphs
/TimePlayed.png")
timeClassic <- filter(timeData, gamemode == "Classic")
timeBox1 <- select(timeClassic, totaltime)
boxplot(timeBox1, at=1, xlim=c(0,3), boxwex=1.75, ylim=c(0,max(timeBox1, timeBox2)
), ylab="Tijd in seconden", main="Totale speelduur per gamemode", xlab="
Classic
")

timeEndurance <- filter(timeData, gamemode == "Endurance")
timeBox2 <- select(timeEndurance, totaltime)
boxplot(timeBox2, at=2, add=TRUE, boxwex=1.75, ylim=c(0,max(timeBox1, timeBox2)),
xlab="
Endurance")
dev.off()

ANOVA analysis
failModel <- aov(failrate ~ gamemode, data=failData)
summary(failModel)

levelModel <- aov(level ~ gamemode, data=levelData)
summary(levelModel)

timeModel <- aov(totaltime ~ gamemode, data=timeData)
summary(timeModel)

'''

```