




Concept Serious Game

67 - 27 = 20 

23 - 40 = 3 

8 + 2 = 10 

Time: 6 sec. 

Inhoud

1.	Inleiding	3
2.	Concept	4
	2.1 Doelgroep	4
	2.2 Achtergrondverhaal/Doelen	4
	2.3 Over het spel	4
	2.4 Uitvoering	5
3.	Technische documentatie:	6
	3.1 Lijst functionele eisen	6
	3.2 Flow	6
	3.3 Wireframes	7
4.	Gameplay & features	9
	4.1 Core mechanics	9
	4.2 Economie / balans	10
	4.3 Leercurve	10
	4.4 Stimuli & beloningen	10
	4.5 Feedback	11
	4.6 Uitspraak over het (spel)ontwerp	11
5	Verkennen van 2 extra onderwerpen	12
	5.1 Gameplay test resultaten	12
	5.2 Bediening (control)	12

1. Inleiding

Bij het bedenken van een onderwerp voor mijn serious game gaan kwamen in eerste instantie een hoop algemene onderwerpen aan bod.

Namelijk de volgende lijst:

- Global warming → energieverpilling;
- Economie → leren met geld omgaan;
- Management → hoe je tijd in de plannen;
- Het menselijk lichaam → leren over de stofwisseling;
- Biologie → evolutietheorie;
- Programmeren → visuele metaforen voor kinderen vanaf 10 jaar;
- Voeding → gezond eten;
- Sport → mensen aanzetten tot een gezonde levensstijl;
- Geschiedenis → kennis vergaren over de 2^e wereldoorlog;
- Topografie → provincies en plaatsen in Nederland leren voor de basisschool;
- Natuur- of wiskunde → basiskennis voor basisschoolkinderen;
- E.v.a, o.a. kunst, filosofie, psychologie.

De meest interessante onderwerpen leken me biologie en het menselijk lichaam. Ik heb het dit niet doorgezet omdat het me te complex leek dit voor zo'n korte periode met voldoende kwaliteit uit te werken. Vooral omdat ik een interactieve applicatie in Flash ActionScript wilde maken deze keer i.p.v. een papieren model, zoals bij de vorige opdrachten.

Een 2^e idee had ik gebaseerd op global warming. Een onderwerp wat nogal hot is tegenwoordig. Niet omdat het onderwerp me bijzonder interesseert (tenminste, niet meer dan de andere onderwerpen), maar omdat zich al een klein ideetje vormde, en je toch een keer de knoop moet doorhakken. Dit idee bestond uit een soort Sims-achtig huishouden (behoorlijk vereenvoudigd uiteraard), waarbij de bewoners heel primitief hun ding doen. Het is jouw taak om het gezin aan te sturen op een zo laag mogelijk energieverbruik. Bijvoorbeeld als ze de lichten aan laten staan in kamers waar niemand zit e.d kun je ze daar op wijzen. Minigames tijdens de main gameplay kunnen voor afwisseling zorgen. Bijvoorbeeld een flatgebouw met lampjes die at random aangaan. Het is dan jouw taak om deze weer op tijd uit te klikken. Aangezien het hoofd idee te complex is om uit te werken (in Flash), zou ik me dan vooral richten op die minigames rond het gebeuren.

Uiteindelijk kwam de mogelijkheid op om basis- of middelbare schoolkinderen vakken te leren die over het algemeen als zeer saai en vervelend beschouwd worden. Bijvoorbeeld programmeren, elektrotechniek, natuurkunde en wiskunde.

Met die laatste ben ik dan ook aan de slag gegaan. Vooral omdat het heel elementair is, en relatief makkelijk uit te werken in een korte periode.

2. Concept

2.1 Doelgroep

Als primaire doelgroep heb ik voor basisschoolkinderen gekozen van groep 4 (kinderen die eenmaal de absolute basis van het rekenen hebben geleerd) tot groep 6 (kinderen die al wat verder gevorderd zijn). Als secundaire groep eventueel nog uitschieters naar groep 7 & 8, door daar behandelde onderwerpen te integreren in de game, en de moeilijkheidsgraad nog verder te verhogen.

2.2 Achtergrond en doel van het spel

Het primaire doel is niet zozeer het leren van rekenen, maar vooral het oefenen van rekenen om de prestaties te verbeteren en het leren van nieuwe technieken in de toekomst makkelijker te maken.

Het moet vooral leuker worden om te oefenen. Dit voornamelijk door directe interactie met het programma, de mogelijkheid om fouten te maken en deze te corrigeren, directe feedback over de resultaten van een poging en, last but not least, de mogelijkheid om het programma aan te passen aan het niveau van de speler. Of deze nu 7, 10 of zelfs 12 jaar is, en weinig of veel ervaring met rekenen heeft.

De mogelijkheid om punten te halen en zo hoog mogelijk op het scorebord van de klas of school te eindigen moet kinderen nog meer motiveren om veel te blijven oefenen met het spel. Docenten of het spel zelf kunnen stimuli ('incentives'), bijvoorbeeld een beloning, aanwenden om de leerling nog verder te motiveren zo hoog mogelijk te eindigen.

2.3 Over het spel

De taak van het spel is het random genereren van bepaalde opgaven. Elke opgave heeft minimaal één oplossing. De speler moet deze oplossingen zo snel mogelijk zien te vinden. De opgaven worden gepresenteerd in verschillende rondes. Het aantal berekeningen per ronden en het aantal rondes per sessie is afhankelijk van de instelling van de moeilijkheidsgraad.

Onder andere afhankelijk van deze instelling van de moeilijkheidsgraad wordt de soort en complexiteit van de berekening bepaald. Een beginner wordt geïntroduceerd in het optellen van getallen en begint vervolgens met eenvoudige optelsommen maar waarden onder de 10. Bijvoorbeeld $3 + 5 = x$ en $7 + x = 9$. Deze speler krijgt de oplossingen 8 en 2 voorgeschoteld. Vervolgens is het aan de speler om de juiste oplossing bij de juiste opgave te vinden.

Door het oplossen van deze sommen kan de speler punten verdienen. Hoe meer antwoorden hij/zij juist heeft, en hoe sneller de tijd waarin deze oplossingen gevonden zijn, des te hoger de score. Met deze punten komt de speler op een scorebord te staan tussen zijn/haar klas- of schoolgenoten. Docenten kunnen zo ook doelen stellen aan hoeveel punten een leerling minimaal moet (kunnen) halen. Zeker als je hier wedstrijden met beloningen aan koppelt.

De punten kunnen tevens gebruikt worden om prestaties van de leerling te meten. Wanneer deze boven een bepaalde score komt, kun je andere berekeningen (met andere operatoren), meer berekeningen per ronde, of complexere berekeningen van hetzelfde soort introduceren. Eventueel zelfs combinaties van berekeningen met verschillende operatoren. Zo kun je het niveau langzaam omhoog schroeven.

Voorbeeld van een 'learning curve':

- Beginnen met optelsommen: $x + 3 = 6$ (3 opgaven en oplossingen per ronde);
- Aftreksommen introduceren: $7 - x = 5$;
- Deze combineren in een ronde: $3 + 7 = x$, $6 - x = 1$, $4 + x = 9$ met de antwoorden 10, 5 en 5;
- Grootte van de getallen opvoeren: tientallen, honderdtallen;
- Complexere berekeningen: $23 + x - 34 = 67$;
- Vermenigvuldigen en delen introduceren;
- Meer onbekende waarden gebruiken per opgave: $17 + x - x = 123$ (mogelijk omdat er maar een beperkt aantal oplossingen zijn)
- Etc.

Zo kun je de opgaven zo moeilijk of makkelijk maken als je wilt, aangepast op het niveau van de speler. Ofwel door deze in het begin in te stellen, ofwel doordat de speler genoeg punten haalt om geïntroduceerd te worden met meer complexe opgaven.

2.4 Uitvoering

Het was in eerste instantie de bedoeling op het spel heel visueel te maken vol metaforen. Bijvoorbeeld een taart met 10 stukken die je eerlijk moet verdelen over 5 mensen. Dit is naar mijn mening niet haalbaar binnen de tijdperiode, omdat het spel uit vele minigames zou bestaan die je een voor een moet ontwikkelen, tenzij ik het zou houden bij een papieren prototype.

Zoals ik al vertelde wilde ik deze keer iets meer interactiefs afleveren. Iets wat ik ook wat meer kon testen op look en feel. Daarom heb ik een heel basaal prototype gebouwd in flash die random een bepaald aantal berekeningen genereert in een bepaald aantal ronden. Het aantal ronden, aantal opgaven per ronde en in zekere zin de moeilijkheidsgraad van de opgave zijn alleen in te stellen via variabelen in de code. Maar in een uiteindelijke versie moet dat via de interface kunnen voorafgaand aan de game.

Bij dit prototype heb ik het gehouden op de kern van de game: het oplossen van berekeningen. Een sessie duurt drie ronden, waarin deze per ronde 3 berekeningen genereert met 3 getallen per berekening waarvan een waarde ontbreekt. Voor elke berekening wordt de ontbrekende waarde gegeven in de vorm van een object (speelsteen) die de gebruiker naar die juiste positie moet slepen. Deze stenen worden onder de berekeningen weergegeven en random 'geschuffled,' dus de speler kan uit de positie van de stenen niet opmaken welk antwoord juist is (wel zo handig). Het prototype implementeert alleen optel- en aftreksommen met waarden tussen de 0 en 99.



3. Technische Documentatie

Summiere technische documentatie.

3.1 Lijst functionele eisen

Prioriteiten volgens MoSCoW.

Must Haves

- Mogelijkheid om moeilijkheidsgraad in te stellen;
- Optellen, aftrekken, vermenigvuldigen en delen;
- Tussentijdse resultaten;
- Eindresultaat;
- Scorebord.

Should Haves

- Geluid en animaties als feedback;
- Gebruik van metaforen.

Could Haves

- Breuken, percentages etc. (groep 7 en 8 content).

Would Haves

- (Online) multiplayer.

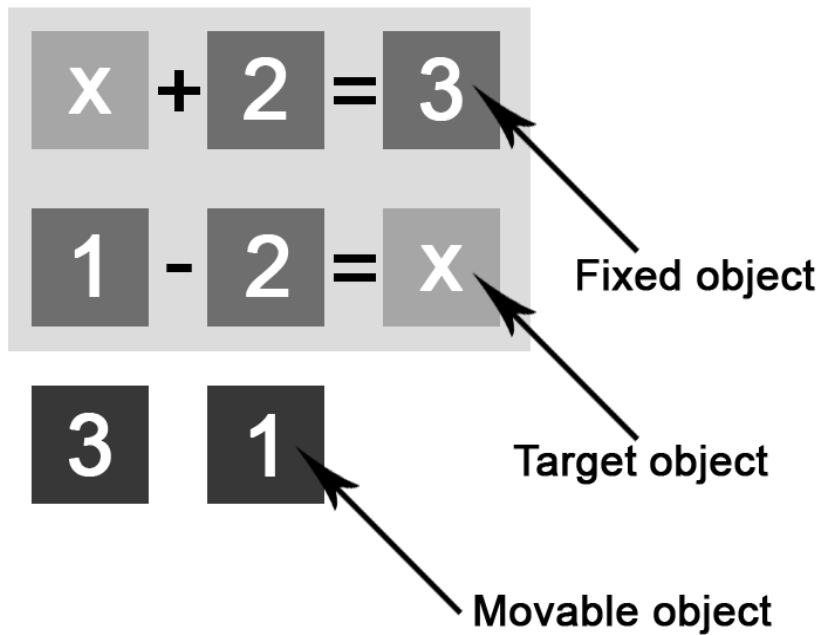
3.2. Flow

De flow van het spel nadat deze opgestart is tot het einde van een spelsessie is als volgt:

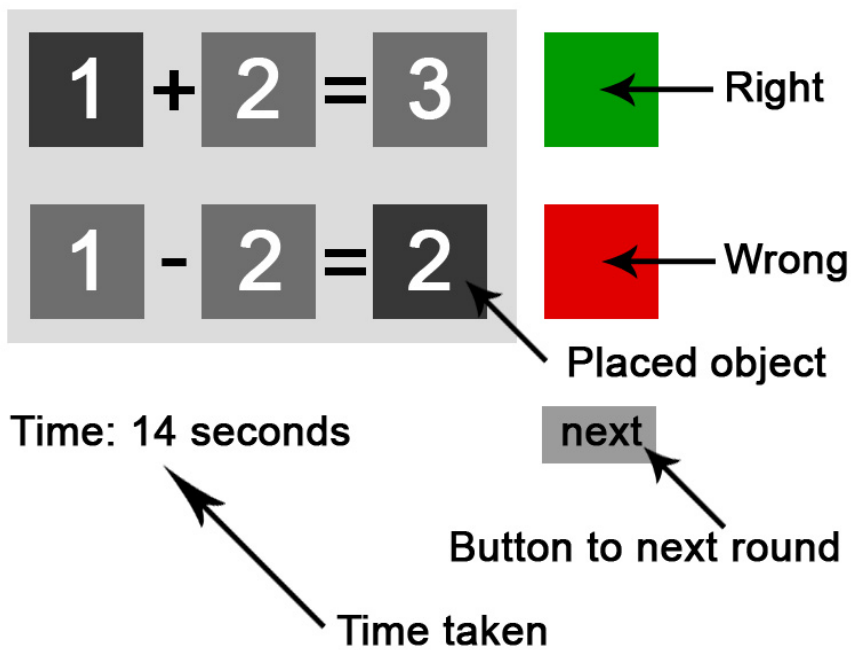
1. Startscherm;
2. Instellingen (welke operatoren, hoeveel variabelen, berekeningen, ronden, complexiteit etc);
3. Gameplay ronde;
4. Tussentijdse resultaten;
5. Iteratie tussen 3 en 4;
6. Eindscore & scoreboard.

3.3 Wireframes

Gameplay tijdens een ronde:



Tussentijdse resultaten na een ronde:



Eindscore na een volledige speelsessie:

You earned 51 points!

Total time taken: 36 seconds
6 answers right of 9 total
3 answers wrong of 9 total



6 star rating!

3	perspiciatis	8 stars	74 points
4	unde	7 stars	59 points
5	omnis	6 stars	51 points
6	iste	5 stars	46 points
7	natus	5 stars	45 points

score

again

4. Gameplay & features

4.1 Core mechanics

Core mechanic regels:

- Het aantal gegeven stenen is gelijk aan het aantal velden met ontbrekende waarden;
- De waarden van de stenen is gespiegeld aan die van de velden;
 - Elke berekening / veld heeft minimaal één oplossing.
- Elk opgelost veld levert 10 punten op;
- Hoe langer je over een ronde doet, des te meer punten worden afgetrokken van je eindscore;*
- De eerste 1.5 seconden die je nodig hebt per geplaatste steen wordt niet afgetrokken van het totale puntenaantal;*
 - Bijvoorbeeld bij 3 stenen worden eerste 4.5 seconden niet afgetrokken van het puntenaantal, dit is de tijd die je ongeveer nodig hebt om de stukken te plaatsen, op deze manier is het mogelijk om een 'perfecte score' te halen voor een ronde.*
- Stenen moeten fysiek op de lege velden geplaatst worden, niet ernaast;
 - Zo niet springt de steen terug naar zijn beginpositie.
- Stenen kunnen onderling van veld wisselen, zolang de laatste steen nog niet gelegd is;
 - Wanneer de laatste steen gelegd is worden alle stenen 'gelocked' en resultaten van die ronde getoond.
- De volgende ronde kan pas gestart worden wanneer alle stenen gelegd zijn.
 - onafhankelijk van tijd en of deze goed dan wel fout zijn (deze hebben alleen invloed op de score).

* De score wordt berekend volgens de volgende formule:

Score = totaal aantal juiste antwoorden x 10 punten

Als de totale tijd groter is dan → aantal oplossingen per ronde x aantal ronden x 1.5 seconden

Wordt de volgende berekening uitgevoerd:

Totale tijd - aantal berekeningen per ronde x aantal ronden x 1.5 seconden

Score – totale tijd ^{0.7}

Als de score kleiner is dan 0 wordt deze teruggezet naar 0.

De laatste machtsverheffing van 0.7 heeft ermee te maken dat de tijd steeds minder invloed heeft op de score naarmate deze vordert. Dit om langzame spelers nog een kans te geven. (zo niet loopt de score lineair omlaag en heeft een trage speler een betrekkelijk grote kans dat hij/zij op een score van 0 punten uitkomt).

Eigenlijk is een puntensysteem waarbij de score nooit op 0 eindigt het beste, maar die heb ik nog niet weten te implementeren.

4.2 Economie / balans

Sources and drains:

Source: steen met (een deel van) de oplossing van een berekening

Drain: een veld die de ontbrekende waarde van een berekening toont

Evenveel sources als drains, dus voor elke source een drain. De balans is dus evenwichtig en draait altijd uit op 0.

Zie feedback loop.

4.3 Leercurve

Zoals al eerder vertelt is de moeilijkheidsgraad van het spel vanaf het begin in te stellen. Tijdens het spelen van het spel loopt de moeilijkheidsgraad verder omhoog wanneer de prestaties van de speler verbeteren.

Herhaling op het voorbeeld van een 'learning curve':

- Beginnen met optelsommen: $x + 3 = 6$ (3 opgaven en oplossingen per ronde);
- Aftreksommen introduceren: $7 - x = 5$;
- Deze combineren in een ronde: $3 + 7 = x$, $6 - x = 1$, $4 + x = 9$ met de antwoorden 10, 5 en 5;
- Grootte van de getallen opvoeren: tientallen, honderdtallen;
- Complexere berekeningen: $23 + x - 34 = 67$;
- Vermenigvuldigen en delen introduceren;
- Meer onbekende waarden gebruiken per opgave: $17 + x - x = 123$ (mogelijk omdat er maar een beperkt aantal oplossingen zijn)
- Etc.

De werking van de game wordt uitgelegd in een tutorial, bereikbaar vanuit het startmenu. Een stem vertelt dat je objecten kunt slepen, en dat het de bedoeling is deze op de juiste positie te krijgen. Eventueel met een animatie. Tegelijkertijd kun je dit zelf uitproberen. Daarnaast worden het doel en de regels uitgelegd: meer oplossingen = meer punten en sneller = meer punten.

Het aftrekken van getallen zou uitgelegd kunnen worden aan de hand van animaties met metaforen (5 auto's op een parkeerplaats, 2 rijden weg). Met een koppeling naar het spelverloop.

Dit niet het primaire doel van het spel. Dit spel is met name bedoeld in het onderwijs, en eventueel op een later tijdstip thuis om te oefenen. De uitleg wordt dus primair verzorgd door de docent.

4.4 Stimuli & beloningen

In de game:

- Positie op het scoreboard (hoog willen eindigen);
- Tussentijdse resultaten (betere rondes willen draaien);
- Vooruitgang in de game, meer punten (nieuwe berekeningen);
- Uiterlijk van de game mogen customizen bij genoeg punten;
- Speler zelf laten bepalen welke soort berekeningen hij zij wil oefenen bij genoeg punten.

Extern georganiseerd door de docent:

- Wedstrijden met een beloning voor de beste prestaties.

4.5 Feedback

Twee soorten feedback:

Feedback vanuit de interface naar de gebruiker toe:

- Velden met ontbrekende waarden in een afwijkende kleur om aan te geven dat de stenen hier terecht moeten komen;
- Het automatisch op plaats snappen van stenen om aan te geven dat deze hier geplaatst mogen worden;
- Geluid als bevestiging van het succesvol plaatsen van een object;
- Het terugspringen naar de startpositie bij het onjuist plaatsen van een object;
- Next-knop weergeven om aan te geven dat de volgende ronde gestart kan worden;
- Aangeven welke berekeningen goed of fout zijn gemaakt en hoelang de speler er over gedaan heeft om de berekeningen op te lossen na het eindigen van een ronde;
- Na een volledige spelsessie de score van de speler aangeven, inclusief plaats op het scorebord. Eventueel inclusief totale tijd en het aantal goede en foute antwoorden;

Feedback loop in de game:

- Speelstenen inleveren voor vooruitgang en punten in het spel;
- Vooruitgang zorgt voor het bezit van nieuwe stenen, waar vervolgens weer meer vooruitgang en punten mee te behalen zijn/is.

(Hoe meer punten, des te meer features).

4.6 Uitspraak over het (spel)ontwerp

Het ontwerp is niet definitief. Wat ik alleen een beetje duidelijk wil maken is het gebruik van grote duidelijk knoppen en felle kleuren voor de belangrijkste elementen (vanwege de doelgroep).

Animaties, visual effects en audio ontbreekt nog. De interface is te donker.

Prestaties moeten zo visueel mogelijk weergegeven worden, met alleen puntentelling als cijfer.

I.p.v. van abstracte getallen zou je voor de kleinste gameplay kunnen implementeren gebaseerd op metaforen. (het eerlijk verdelen van een zak snoep aan een bepaald aantal personen).

5. Verkennen van 2 extra onderwerpen

5.1 Prototype uitslag

Het prototype is een flink vereenvoudigde versie in Flash ActionScript. Het prototype ondersteunt alleen optellen en aftrekken, berekeningen met 3 variabelen, heeft geen geluid, niet alle feedback werkt, geen definitief design en nog niet gebruiksvriendelijk genoeg voor kinderen. Tevens ontbreekt een tutorial-level of visuele uitleg via animaties en een vriendelijke stem.

Dit waren dan ook een aantal dingen die aan het licht kwamen tijdens het prototypen (meeste zijn dus nog niet - goed – geïmplementeerd):

- Heb de tijdslimiet weggehaald, om frustratie bij de gebruiker te voorkomen (tolerantie om fouten te maken);
 - Meer tijd zorgt simpelweg voor een lagere score.
- Geen tijden weergegeven tijdens het spelen, dit hitst de speler alleen maar op en leidt af van de gameplay;
- Probeer de uiteindelijke score zoveel mogelijk grafisch te laten zien i.p.v. met een lijst getallen;
 - Bijvoorbeeld 7 van 10 sterren + zoveel punten bij het totaal.
- Objecten al laten 'snappen' voordat je de muisknop loslaat, om goed aan te geven wat mogelijk is;
- Geluid gebruiken als feedback.;
- Score systeem die niet op 0 kan eindigen als je minimaal een aantal antwoorden goed hebt, zelfs al doe je daar heel lang over.

5.2 Bediening (controls)

Ik heb gekozen voor een interface waarbij je dingen moet verslepen omdat dit herkenbaar en natuurlijk is. Tevens is dit ruimtelijk gezien gemakkelijk te begrijpen (objecten hebben een ruimtelijke positie en kunnen van A naar B en andersom).

Het mooiste zou zijn als je dit spel verder zou ontwikkelen op een systeem met een touchscreen zoals die van een media-installatie of de Nintendo DS. Daardoor kun je de interface als het ware 'aanraken' en de objecten verslepen met je vingers. Iets wat goed overeen komt met de handeling en heel natuurlijk aanvoelt.

Om nog verder te gaan zou je ervaren spelers toetsen(combinaties) kunnen geven om objecten zonder ze te hoeven verslepen (dus zonder tijdsverlies te kunnen plaatsen). Tijd heeft dan wel direct invloed op je score (je wordt niet meer gematst met 1.5 seconde per zet). Met deze methode kun je heel direct meten wie de snelste is, met directe invloed op je score.

Op een pc, console of media-installatie zou je een knop kunnen toewijzen aan een positie. Zowel die van de stenen als de locaties waar de naartoe moeten. Op een PC zou je daar bijvoorbeeld het numpad kunnen gebruiken. Als je de 2^e steen op de 3^e positie wilt hebben druk je simpelweg 2 en vervolgens 3. Het eerste getal is de modus van het object, de tweede de modus van de locatie waar deze terecht moet komen. Deze modus kun je grafisch weergeven op het scherm en afbreken door de escape toets. Bij een console zou je posities kunnen toewijzen aan de actieknoppen.

Met oog op de toekomst zou je eventueel gebruik kunnen maken van patroonherkenning met behulp van z.g.n. 'fiducials.' Wanneer je deze voor je webcam houdt, herkent de software het ID-nummer van de betreffende fiducial, om vervolgens de positie hiervan te tracken. Zo kun je de objecten reallife plaatsen. Eventueel via een speciale tafel met videoprojector.

Voor meer info check reactIVision: <http://reactivision.sourceforge.net/>