

A journey to rule the world

Pieter van Laatum en Max Tijssen

29 juni 2012

Inhoudsopgave

1	Voorwoord	3
2	Beschrijving	3
2.1	Inleiding	3
2.2	Productverantwoording	3
2.3	Specificaties	4
2.3.1	Functionele eisen	4
2.3.2	Niet-functionele eisen	4
2.3.3	Use case	5
3	Ontwerp	6
3.1	Globaal ontwerp	6
3.2	Detailontwerp	7
3.3	Ontwerpverantwoording	9
3.3.1	Gebruik van een server	9
3.3.2	Touch Array	9
4	Evaluatie	10
4.1	Testdoelen	11
4.2	Gebruikers(sub)groepen	11
4.3	Scenario's	12
4.4	Methoden voor gegevensverzameling	12
4.5	Procedure	12
4.6	Resultaten	13
4.7	Conclusies	13
5	Reflectie	14
5.1	Groep	14
5.2	Individueel(Pieter van Laatum)	15
5.3	Individueel(Max Tijssen)	15
6	Bijlage	16

1 Voorwoord

Dit document beschrijft de ontwikkeling van onze mobiele Android-app: "Backstab". We zullen proberen om inzicht te geven in hoe de ontwikkeling hiervan gegaan is, hoe het product ontworpen is en hoe onze app uiteindelijk door gebruikers ervaren wordt.

Allereerst zullen we een beschrijving geven van wat onze app nou precies is, hierbij zullen we ook een productverantwoording en de specificatie van de app geven. Hierna zullen we wat verder ingaan op het ontwerp van de app. Hoe hebben we het laten doen wat het doet? Hierbij geven we een globaal ontwerp met de componenten, en daarna per component in detail hoe we deze uitgewerk hebben. Ook zullen we een verantwoording geven voor een aantal keuzes van ons project waar mogelijk vraagetekens bij gezet kunnen worden. Dan volgt een eind-usability test, om aan te tonen dat sinds de eerste test onze app vooruit gegaan is. Afsluitend zal er een reflectie zijn over het proces van ons project, hoe het groepsverband is verlopen en dergelijke.

2 Beschrijving

2.1 Inleiding

Wat is Backstab? Backstab is een tactisch spel gebaseerd op het bordspel "Diplomacy". De speelkaart kan vegeleken worden met die van Risk, er zijn landen die in het bezit van de spelers kunnen zijn en op ieder land staat maximaal 1 unit. Het overnemen van landen is hier in tegenstelling tot Risk niet gebaseerd op kans, maar op tactiek.

Onze versie van het spel is een twee-spelerspel dat zich afspeelt in Groot Britannië, rond 1900. Het spel wordt online gespeeld tussen twee personen. Nadat er op nieuw spel gedrukt wordt geeft de server een spel terug waar vanaf dat moment op gespeeld wordt.

Het spelen met andere mensen wordt via een server geregeld, beurten worden opgestuurd en de nieuwe speelkaart wordt weer teruggestuurd. De app werkt op alle android telefoons met minstens android 2.1, en het speelscherm wordt automatisch aangepast naar de grote van het scherm van het apparaat.

2.2 Productverantwoording

Is er eigenlijk wel behoefte aan een app zoals deze? Wat is er nog meer op de markt, en kan dit daar tegenop?

Allereerst is er op de markt nog geen spel zoals dit, voor zover bij ons bekend is er geen andere app die dit borspel emuleert zoals wij dat doen. Al bestond deze dan hebben we alsnog onze eigen twist er op. Aan de andere kant is de app wel een spel, en daarvan bestaan er op de markt al meer dan te tellen zijn, dus waarom verdient onze app een plek in een markt die al uitpuilt van de spellen? Veel van de spellen die op dit moment op de markt zijn zijn bedoelt voor kort,

snel vermaak, zonder er echt bij na te hoeven denken (Angry Birds, Draw Something, Doodle Jump etc.) Dit willen veel mensen ook graag, maar er zullen ook gebruikers zijn die liever een iets tactischer spel spelen, waarin het goed nadenken over wat je doet beloont wordt.

Daarnaast is het feit dat je met andere mensen speelt een groot voordeel. Niet alleen is hierdoor ieder potje compleet anders, ook zal een overwinning zeker voelen als een echte overwinning. Je hebt namelijk bewezen een andere speler slimmer af te kunnen zijn, een beter plan te kunnen smeden.

Ook bestaan er op internet al actieve fora waar mensen dit spel spelen door het opsturen van hun beurten naar een 'gamemaster' die dit allemaal bijhoudt. Deze mensen hebben nu een manier om het spel ook onderweg te kunnen spelen, niet meer alleen thuis achter de computer, en zonder gebruik te hoeven maken van iemand die alle zetten controleert en verwerkt. Bij onze app wordt hier intern en op de server voor gezorgd. Dit zorgt voor een meer gestroomlijnde en snellere speelervaring.

2.3 Specificaties

In dit gedeelte van het verslag zullen we wat specifieker ingaan op wat onze app precies inhoudt, en wat hij doet. We zullen nog niet te diep ingaan op hoe we dit gerealiseerd hebben (dit is het volgende deel van het verslag), maar we zullen aan de hand van een use case al wel laten zien hoe het, vanuit de gebruiker gezien, werkt.

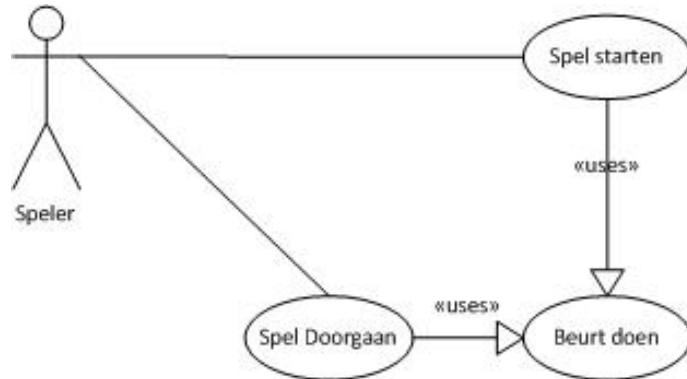
2.3.1 Functionele eisen

- De gebruiker kan een spelletje Diplomacy spelen met een ander persoon.
- Een online multiplayer spel spelen
- Het spel kan opgeslagen worden
- De beurten verlopen volgens licht aangepaste regels van het origineel.
- Een gebruiker kan zonder hulp van ons een beurt doen, en de gevolgen van deze beurt zien.

2.3.2 Niet-functionele eisen

- Het spel moet werken op alle mobiele telefoons met Android 2.1
- Er moet geen beveiligings-/privacyrisico voor de gebruiker zijn

2.3.3 Use case



UC#:	03
Naam:	Beurt doen
Omschrijving:	Deze use-case dient voor het uitvoeren van een beurt
Primaire Actor:	De persoon die de applicatie wil gebruiken, hierna de speler genoemd
Trigger	De speler heeft een nieuw spel gestart, of gaat met een spel door.
Basic Flow:	
Stap	Actie
1	Het systeem laat de kaart en GUI zien.
2	Indien de speler een veld selecteert, wordt sub-flow 3.1 uitgevoerd. Indien de speler op end turn drukt, wordt 3.2 uitgevoerd
3.1.1	Indien het geselecteerde land een unit bevat van de speler, of het winter is en het geselecteerde land is van de speler laat het systeem het land oplichten om duidelijk te maken dat het geselecteerd is. Anders wordt stap 2 weer uitgevoerd.
3.1.2	Indien het lente of herst is en de speler Move kiest: 3.1.3.1. Indien Support: 3.1.3.2. Indien Hold: 3.1.3.3. Indien Convoy: 3.1.3.4 . Indien het winter is en de speler kiest Destroy: 3.1.3.5. Indien Build: 3.1.3.6.
3.1.3.1.1	Het syteem doet de velden waar een eenheid naartoe kan oplichten.
3.1.3.1.2	Als de speler een van de opgelichte velden selecteert, wordt een move order uitgevoerd en daarna nogmaals stap 2. Anders wordt stap 2 uitgevoerd.
3.1.3.2.1	Het systeem doet alle velden oplichten waar units staan die de huidig geselecteerde unit zou kunnen supporten.
3.1.3.2.2	Als de speler een van deze velden selecteert wordt 3.1.3.2.3 uitgevoerd. Anders wordt stap 2 uitgevoerd..
3.1.3.3.1	De unit hold.

3.1.3.3.2	Voer stap 2 uit.
3.1.3.4.1	Het systeem doet alle mogelijk velden oplichten vanwaar de unit kan convoyen.
3.1.3.4.2	Als de speler niet een van de oplichtende velden selecteert, wordt stap 2 uitgevoerd.
3.1.3.4.3	Het systeem doet alle mogelijke velden oplichten waarnaartoe de unit kan convoyen.
3.1.3.4.4	Als de speler niet een van de oplichtende velden selecteert, wordt stap 2 uitgevoerd.
3.1.3.4.5	De convoy wordt uitgevoerd.
3.1.3.5	Indien er een unit op deze tile stond wordt deze ge-destroyed, de unit count verlaagt en stap 2 uitgevoerd. Anders wordt stap 2 uitgevoerd.
3.1.3.6	Indien er nog geen unit op deze tile stond wordt er een gebouwd, de unit count verhoogt en stap 2 uitgevoerd. Anders wordt stap 2 uitgevoerd.
3.2	Indien het geen winter is: systeem brengt de speler terug naar het hoofdmenu. Indien het wel winter is: als de unit count gelijk is aan het aantal supply points: het systeem brengt de speler terug naar het hoofdmenu, anders wordt er een pop up gegeven met de informatie dat er nog meer units gebouwd/destroyd moeten worden en wordt stap 2 uitgevoerd.
Alternate Flows:	
4	Afbreking van het starten. De speler kan op ieder gewenst moment terug naar het begin van zijn beurt
Precondities:	De speler is aan de beurt
Postcondities:	De gewenste zet is gedaan.
Business Rules:	n.v.t.
Supplementary Requirements:	n.v.t.
Opmerkingen:	geen
Openstaande Punten:	geen

3 Ontwerp

3.1 Globaal ontwerp

- *Startmenu's*: Deze module zal alle menu's bevatten die nodig zijn om een spel op te starten, of verder te gaan. Hierbij hoort het beginscherm met daarop de knoppen om een spel voort te zetten, instructies te lezen en een nieuw spel te starten. Ook hoort hierbij het instructiescherm.

Deze module zal vooral belangrijk zijn om het laten beginnen van een spel zo soepel mogelijk te laten verlopen, zodat hier niet teveel tijd aan verdaan hoeft te worden.

- *Speelkaart*: Deze module bevat het daadwerkelijke spel. Hierbij moet gedacht worden aan de kaart met gebieden, de spelers en hun belangrijke eigenschappen, de schermen die deze gegevens tonen en ook de individuele

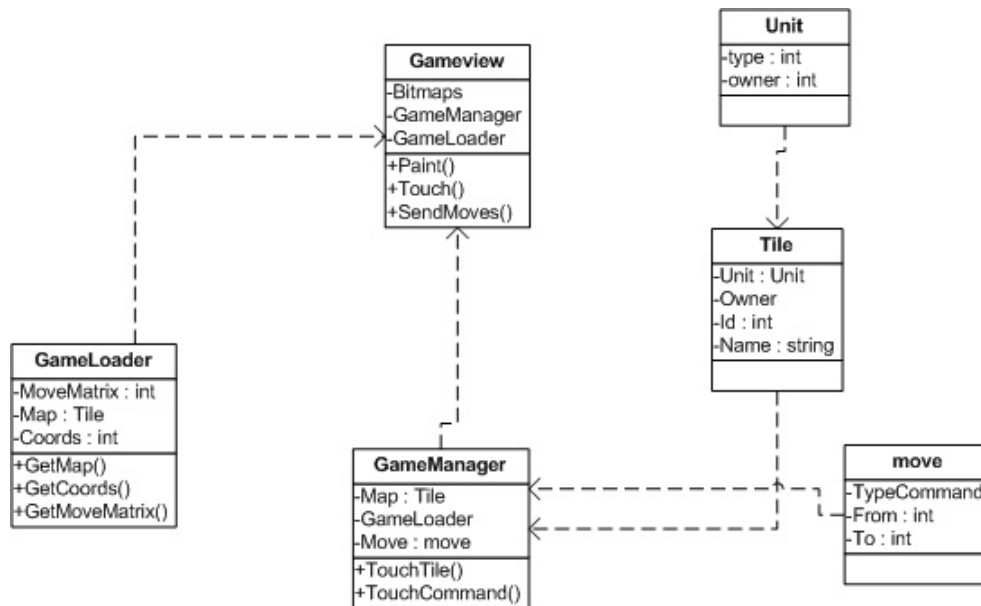
troepen van het spel. Daarnaast zal hier ook de 'spelleider' bij zitten die verantwoordelijk is voor bijvoorbeeld het bijhouden van de beurten, het bepalen wat welke unit kan doen en er voor te zorgen dat de kaart alle belangrijke gegevens bevat.

Deze module is belangrijk voor het verloop van het spel zelf, en zal dan ook het grootste deel van het werk innemen.

- *Server*: De server zal er voor zorgen dat de module speelkaart de correcte kaart heeft om te laten zien. De server zal de beurten die naar hem opgestuurd worden verwerken en dan, als hierom gevraagd wordt, de aangepaste kaart weer terugsturen. Ook kan de server errors terug sturen, bijvoorbeeld als er nog op meer spelers gewacht moet worden. Dit wordt dan verwerkt door de andere modules.
- *Samenhang*: De startmenu module dient vooral om de regels uit te leggen en om berichten naar de server te versturen. Zo kan er een bericht gestuurd worden voor een nieuw spel waarop de server een nieuwe kaart en een ID terugstuurt, ook kan een spel vervolgt worden met een eerder verkregen ID. De kaart die terug gestuurd wordt door de server zal dan door de speelkaart module gebruikt worden om het spel te laten zien, en hier weer nieuwe moves op te generen om naar de server te sturen.

3.2 Detailontwerp

De door ons uitgewerkte use case geeft in feite de module Speelkaart weer. Het draait ten eerste om het laten zien van de speelkaart en de elementen die hierbij horen, waarbij de plaatjes ingeladen moeten worden en op de correcte positie geplaatst worden. Ook om het laten doen van een beurt, de correcte plaatjes moeten oplichten om duidelijk te maken wat er gekozen kan worden en de beurten moeten worden opgeslagen om opgestuurd te worden.



Het hierboven gegeven klassediagram geeft grotendeels weer hoe dit in Java-klassen gebeurt. Gameview is de grote overkoepelende klasse die alles laat zien, en de touches verwerkt. Gamemanager is de onderliggende klasse die (aan de hand van de movematrix) bepaalt waar verschillende moves mogelijk zijn. De gameloader neemt de gamefile die de gameview bij aanmaken meegekregen heeft en maakt hier nuttige informatie van, zoals wie welk land bezit en waar verschillende units staan. Deze informatie gebruikt de gameview om de map te tonen en de gamemanager gebruikt het om de moves te bepalen. De positie van de tiles wordt bepaald door middel van een coördinatensysteem dat meegegeven wordt per kaart en door de gameLoader verwerkt wordt, deze bepaalt (na geschaalt te zijn naar de device) de linkerbovenhoek van iedere tile. Als laatste neemt de gameLoader ook de moveMatrix file en maakt hier fromaat van dat ook doo rde rest van de klassen te gebruiken is.

Bij het aanmaken van de gameview wordt ook de touch array aangemaakt; deze array geeft per x en y waarde van het scherm aan bij welke tile dit hoort, deze array wordt daarna gebruikt om de touch mee te regelen. Hoe dit gebeurt is als volgt: De gameview krijgt een touch binnen en indien deze op een tile is, en de tile is geschikt (bijvoorbeeld: het is lente en er staat een unit op) stuurt hij deze aanraking door de gamemanager. Gamemanager zal deze tile dan als gloeiend aanvinken, wat bij het drawen in de gameview daarna te zien zal zijn. Nadat er dan op een commando gedrukt is zal de oorspronkelijke tile niet meer gloeien, maar n u gloeien alle tiles waar naartoe dit commando uitgevoerd kan worden. Nadat hier een keuze in gemaakt is wordt de move binnen gameManager opgeslagen. Natuurlijk kan een move ook op ieder moment afgebroken worden, door nogmaals op hetzelfde tile of commando te drukken. Zodra de beurt beindigt wordt, vraagt gameView van gameManager alle moves op, en

worden deze opgestuurt naar de server om daar verder verwerkt te worden.

3.3 Ontwerpverantwoording

In dit onderdeel zullen wij twee keuzes die wij bij ons project hebben gemaakt verantwoorden.

3.3.1 Gebruik van een server

Voor het verwerken van beurten hebben wij gekozen om een server te gebruiken, met daarop een MySQL database die met PHP bestuurd wordt. Welke andere opties waren er geweest?

De eerste andere optie die we bekeken was peer to peer; iedere telefoon is gelijk en stuurt naar iedere andere telefoon zijn moves op. Nadat een telefoon de moves van iedere andere telefoon binnen heeft berekent hij intern wat voor gevolgen dit heeft en update hij de map zoals nodig.

Waarom kozen we hier niet voor?

Ten eerste zou dit de app vertragen, er zouden $2n^2$ (n =aantal telefoons) verbindingen per beurt nodig zijn, in plaats van de $2n$ die we nu nodig hebben. Daarnaast speelt er het probleem dat als de telefoon waar je de informatie heen probeert te versturen geen ontvangst heeft of uit staat, het bericht verloren kan gaan. Conclusie: deze optie was trager, minder betrouwbaar en bied geen directe voordelen boven ons huidige systeem.

Een andere optie die we bekeken was om gebruik te maken van een 'host' telefoon, de persoon die het spel start is direct ook verantwoordelijk voor het verwerken van beurten. Iedere andere speler stuurt zijn beurten op naar deze mobiel, en hier worden de beurten dan verwerkt.

Waarom kozen we hier niet voor?

Een voordeel van deze optie tegenover de vorige is dat het evenveel verbindingen nodig heeft als onze huidige oplossing, dus niet voor vertraging zorgt. Het brengt wel een erg groot nadeel met zich mee. Vergelijkbaar met de vorige optie bestaat het probleem dat het totaal niet te garanderen is dat de 'host' altijd bereikbaar is, terwijl dit bij een server wel zo is. Conclusie: deze optie is niet sneller maar wel minder betrouwbaar

Concluderend hebben we besloten om gebruik te maken van een server, omdat hier het minste aantal verbindingen gemaakt moet worden, en hij vrijwel onfeilbaar is, en daardoor 24/7 bereikbaar is.

3.3.2 Touch Array

Zodra onze app zich eenmaal in de gameview bevindt werkt de 'touch' door het gebruik van een array met daarin per x en y coördinaat opgeslagen bij welke tile

dit hoort (of dat hij bij geen enkele tile hoort). Waarom hebben we hier voor gekozen? Kon het anders?

De eerste andere optie die we gebruikte (dit was onze oorspronkelijke implementatie) was dat ieder plaatje in feite het hele scherm bedekte, maar doorzichtige pixels had op alle plekken waar het land/de regio niet was. Als alle plaatjes boven op elkaar lagen, vormden ze netjes de kaart. Hierdoor konden we om te bepalen of een land aangeraakt was simpel de kleur van de pixel per plaatje oproepen die bij de x en y coördinaat hoorde. Als deze niet transparant was dan was de tile aangeraakt.

Waarom kozen we hier niet voor?

Het probleem met deze oplossing is dat het onmogelijk is dit in een normale android telefoon te implementeren, het kost namelijk veel te veel geheugen om 40 van dit soort plaatjes op te slaan. Dit was dus meer dan reden genoeg om deze optie niet te gebruiken.

De andere optie was om de plaatjes te houden zoals we nu doen (zo min mogelijk transparante pixels), en samen met het coördinaten stelsel bepalen of het mogelijk is dat een aanraking in een tile was. Weer vroegen we dan hier de x en y coördinaat kleur van op en konden we zo controleren of het land aangeraakt was.

Waarom kozen we hier niet voor?

Deze optie lost het geheugenprobleem van de vorige implementatie op, maar het zorgt er wel voor dat de app een stuk trager wordt. Na testen bleek dat het opvragen van een pixel een vrij traag proces is en dit moest dan alsnog voor iedere tile per aanraking gebeuren.

Onze huidige implementatie is niet perfect, maar wel de beste optie die we konden bedenken. Het werkt door bij het aanmaken van de kaart per pixel van het scherm alvast te bepalen bij welke tile het hoort. Dit wordt in een array gezet en dan wordt bij een latere aanraking via een heel simpele actie opgehaald welke tile bij deze positie hoorde. Het nadeel hiervan is een laadtijd bij het openen van de kaart, maar dit valt in het niet bij de laadtijd van andere opties. Daarnaast is de aanraking per pixel perfect, en is er na het laden geen enkele wachttijd meer. Al met al vonden we dit dan ook verreweg de beste optie.

4 Evaluatie

Dit onderdeel van het verslag zullen we gebruiken om aan de ene kant de vooruitgang van onze app (ten opzichte van de eerste usability test) aan te tonen en om te laten zien dat onze app zijn specificaties waarmaakt. Hiervoor gebruiken wij een usability test, die hieronder beschreven is.

4.1 Testdoelen

Met het uitvoeren van deze test hopen wij de volgende doelen te behalen:
Sommige van deze doelen zijn gebaseert op resultaten van onze vorige test, het hierbij behorende verslag is terug te vinden op onze projectpagina

- Onderzoeken of de gebruikers de laadtijd als te lang of storend ervaren.
- Onderzoeken of gebruikers makkelijk het hulpscherm kunnen vinden en deze voldoende informatief vinden.
- Onderzoeken of het voor gebruikers duidelijk is welke tiles 'gloeien' en dus een optie zijn om aan te raken.
- Onderzoeken of gebruikers de terugkoppeling die ze van de app krijgen begrijpen en voldoende vinden.
- Onderzoeken of gebruikers makkelijk een spel dat ze eerder gestart hebben kunnen voortzetten.
- Onderzoeken hoe natuurlijk het doen van een beurt voor gebruikers overkomt, en of ze dit uit zichzelf begrijpen.

De eerste drie doelen zijn eigenlijk een vervolg op onze vorige usability test, aangezien dit toen de slechtere punten waren, en nu willen we zien of deze verbeterd zijn. De andere drie doelen komen eigenlijk overeen met onze specificaties (eerder in het document gegeven), om te testen of deze behaald zijn.

4.2 Gebruikers(sub)groepen

Wij delen onze gebruikers in in 2 groepen:

- **Smartphonegebruikers**
Mensen die het gewend zijn om een smartphone, en de apps die hierbij horen te gebruiken. Deze groep verwachten wij sneller in te zien hoe het spel te besturen is, aangezien dit overeen zal komen met hun eerdere ervaringen.
- **Niet smartphone gebruikers**
Mensen die over het algemeen geen smartphone gebruiken, deze mensen hebben mogelijk meer moeite met het begrijpen van de besturing van de app.

Met deze gebruikersgroepen hopen we te onderzoeken hoe de ervaring verschilt tussen deze twee groepen, en of wij hiervoor misschien verbeteringen in kunnen voeren.

4.3 Scenario's

Wij zullen gebruikers vragen de volgende scenario's uit te voeren:

- "Look and feel", gebruikers eerst een beetje met de app laten spelen en zien wat ze er van vinden.
- Start een nieuw spel, doe hier een beurt, en ga weer verder met hetzelfde spel.
- Verplaats een van je units met behulp van move.
- Voer een beurt in het seizoen winter uit.

4.4 Methodes voor gegevensverzameling

Voor het verzamelen van de gegevens van deze test zullen we gebruik maken van een vragenlijst die we achteraf afnemen. Daarnaast zullen we natuurlijk ook andere nuttige uitspraken die gebruikers doen noteren. De vragenlijst is als volgt:

- Zou u zichzelf als een smartphonegebruiker of als niet smartphone-gebruiker omschrijven?
- Op een schaal van 1 tot 5 (1 langzaam, 5 snel) vond u de app snel of traag laden?
- Op een schaal van 1 tot 5 (1 langzaam, 5 snel) vond u dat de app snel of traag reageerde op uw invoer?
- Op een schaal van 1 tot 5 (1 totaal niet, 5 heel erg) hoe duidelijk vond u aangegeven welke tiles geselecteert konden worden?
- Op een schaal van 1 tot 5 (1 totaal niet, 5 heel erg) hoe duidelijk vond u de spelregels na het lezen van het instructiescherm?
- Op een schaal van 1 tot 5 (1 totaal niet, 5 heel erg) hoe nuttig vond u de feedback van de app?
- Op een schaal van 1 tot 5 (1 totaal niet, 5 heel erg) hoe duidelijk vond u het hoe een spel vervolgt kon worden?
- Op een schaal van 1 tot 5 (1 totaal niet, 5 heel erg) hoe duidelijk vond u het doen en afsluiten van een beurt?

4.5 Procedure

In totaal hebben wij vier gebruikers getest, dat hebben we op de volgende manier gedaan: Eerst werd de gebruiker de ruimte gelaten om een beetje te experimenteren met de app, om te zien of ze onverwachte acties deden. Daarna vroegen we ze de laatste 3 scenario's uit te voeren. Afsluitend namen we de hierboven geschreven vragenlijst af, waarvan hieronder de resultaten beschreven staan.

4.6 Resultaten

Bij deze resultaten zijn de eerste 2 antwoorden steeds van niet smartphone gebruikers en de antwoorden daarna van wel smartphone gebruikers.

- Zou u zichzelf als een smartphonegebruiker of als niet smartphone-gebruiker omschrijven?
Niet, Niet, Wel, Wel.
- Op een schaal van 1 tot 5 (1 langzaam, 5 snel) vond u de app snel of traag laden?
3, 3, 4, 4. **Gemiddeld: 3.5**
- Op een schaal van 1 tot 5 (1 langzaam, 5 snel) vond u dat de app snel of traag reageerde op uw invoer?
5, 5, 5, 5. **Gemiddeld: 5**
- Op een schaal van 1 tot 5 (1 totaal niet, 5 heel erg) hoe duidelijk vond u aangegeven welke tiles geselecteert konden worden?
3, 4, 4, 4. **Gemiddeld: 3.8**
- Op een schaal van 1 tot 5 (1 totaal niet, 5 heel erg) hoe duidelijk vond u de spelregels na het lezen van het instructiescherm?
4, 3, 4, 4. **Gemiddeld: 3.8**
- Op een schaal van 1 tot 5 (1 totaal niet, 5 heel erg) hoe nuttig vond u de feedback van de app?
3, 5, 4, 5. **Gemiddeld: 4.3**
- Op een schaal van 1 tot 5 (1 totaal niet, 5 heel erg) hoe duidelijk vond u het hoe een spel vervolgt kon worden?
4, 5, 5, 5. **Gemiddeld: 4.8**
- Op een schaal van 1 tot 5 (1 totaal niet, 5 heel erg) hoe duidelijk vond u het doen en afsluiten van een beurt?
3, 3, 4, 3. **Gemiddeld: 3.25**

4.7 Conclusies

Over het algemeen zijn we zeer tevreden met de resultaten van deze test, alle gemiddelden zijn boven de 3, waaronder ook een paar zeer hoge cijfers. De volgende conclusies trekken wij uit deze resultaten:

- Eigenlijk zijn alle verbeterpunten die we in de vorige usability-test vonden opgelost; er is een instructiescherm wat over het algemeen als duidelijk ervaren wordt, de tiles lichten duidelijker op en de laadtijd is enorm verlaagd, zodat deze nu nog maar een klein probleem is.
- Daarnaast hebben we aangetoond dat de reactietijd van de app nog steeds zeer snel (nagenoeg direct) is, ondanks de nieuwe manier van hoe we dit regelen. Daar zijn we dus ook zeker blij mee.

- De nieuwe popups die de app geeft worden als helpvol ervaren en we zijn dan ook blij met deze toevoeging.
- Het doen van een beurt werkt nog steeds niet helemaal intuïtief, mensen verwachten nog steeds dat de unit direct van plaats zou moeten veranderen. Mogelijk zouden we hier een pijltje kunnen plaatjes waar de unit heen zal gaan bewegen, zodat gebruikers weten dat hun move wel binnengekomen is.

5 Reflectie

5.1 Groep

In deze sectie zullen we een negatief en een positief punt dat er voor ons tijdens het werken aan dit project uitsprong wat verder toelichten.

Positief:

Hoe we uiteindelijk het painten en de touch besturing van de plaatjes hebben geregeld. Dit is niet eens zo zeer omdat we dit nu op een heel aparte of moeilijke manier hebben gedaan, en er zal vast een beter alternatief bestaan, maar toch is het iets waar we erg trots op zijn. Dit komt vooral door de hoeveel tijd we er in hebben moeten steken om het werkend te krijgen op een manier die weinig laadtijd heeft, direct reageert op input en minimaal geheugen verbruik. Het feit dat hier veel tijd in is gaan zitten komt omdat we heel lang een verkeerd pad gevolgt hebben, totdat we uiteindelijk alles wat we hadden weg gooide en het opnieuw probeerde. Daarnaast is het ook leuk dat dit een punt is waar ieder van ons werk in gestoken heeft en over meegedacht heeft (Simon was nog bij het bedenken maar niet meer bij de implementie van de oplossing die we nu gebruiken, zie volgend punt) waardoor het ook voelt als iets waar we beide trots op mogen zijn.

Negatief:

Het kwijtraken van ons derde groepslid. Compleet onverwacht is ons laatste groepslid Simon weggevallen bij de cursus en het lijkt erop dat hij met de complete opleiding gestopt is. We staan nog steeds voor een raadsel, want we hebben geen idee wat er precies aan de hand is. We hem nog steeds niet gesproken hebben, maar we gaan ervan uit dat hij simpelweg gestopt is met deze opleiding. Het vetrek van Simon heeft een boel problemen met zich meegebracht. Ten eerste was een groot deel van de architectuur van de code en opzet van de app zijn idee, maar heeft hij ons nooit helemaal verteld wat zijn uiteindelijke visie was. Hierdoor hebben we veel code gevonden die we niet kunnen verklaren of van kunnen begrijpen waar het voor gebruikt zou moeten worden. Dit sluit aan op het volgende probleem, het opeens moeten voorzien van zijn code. Aangezien ieder van ons aan een apart deel van het project gewerkt heeft, dit om efficiënt vooruit te kunnen. Hierdoor moesten we dus opeens code die al dan niet getest of gedocumenteert was zien te gebruiken bij de rest van het project. Nog een ander klein probleem was dat Simon eigenlijk de enige was die het bordspel waar

onze app op gebaseerd is ooit gespeeld heeft. Na zijn vertrek hadden we alleen internetbronnen om te controleren of wat wij deden wel in overeenstemming was met wat de bedoeling was. Uiteindelijk hebben we onszelf en elkaar over al deze problemen heen kunnen helpen en heeft onze app veel kunnen groeien. Daar zijn we dan ook erg tevreden mee.

5.2 Individueel(Pieter van Laatum)

Het moet gezegd worden; dit was voor mij zo ongeveer de nuttigste cursus van het gehele eerste jaar. Ik heb erg veel geleerd over het ontwikkelen van iets in het algemeen, maar ook in het bijzonder het maken van android apps. Daarnaast was het voor het eerst in een project dat ik weinig inzicht had in het werkelijke programmeren. Per slot van rekening was ik de enige informatiekundige in de groep, en daardoor ook de enige die maar weinig ervaring had met object georiënteerd programmeren in Java. Dit was op sommige momenten wel wat spannend, en soms was het zelfs frustrerend dat ik maar weinig kon helpen bij een aantal problemen. Gelukkig heb ik mijn programmeer-tegoed ingehaald door de enige te zijn met kennis en ervaring in PHP. Dat heeft ervoor gezorgd dat ik de aangewezen persoon was om de server op te zetten.

Verder heb ik geleerd dat groepsgenoten nooit te vertrouwen zijn, ze kunnen op ieder moment besluiten dat ze iets anders gaan doen en je niet meer op hun hulp kunt rekenen.

5.3 Individueel(Max Tijssen)

R&D is voor mij een zeer leerzame cursus geweest. Het heeft me erg veel technische vaardigheden geleerd, zoals programmeren in java voor android en hoe je contact maakt met een server om hier informatie van te krijgen. Daarnaast leerde het me ook veel technieken die in een later beroep erg handig zullen zijn, zoals hoe je een presentatie opzet, of een verslag als deze schrijft. Het was voor mij de eerste keer om een project dat zo groot is uit te voeren, en het heeft de laatste weken dan ook vrijwel al mijn vrije tijd opgeslokt, maar uiteindelijk denk ik dat het het waard was. De lessen uit deze cursus over werken in groepsverband, het verantwoorden van je acties en nog vele andere zullen later in veel beroepen ook zeker nuttig zijn.

Daarnaast heb ik geleerd over gewoon stug door te blijven werken als het tegen lijkt te zitten. Aan het begin van de ontwikkeling kon ik niet geloven dat we ooit iets zouden krijgen dat ook maar dichtbij de app zou komen die we wilden, maar door gewoon stukje bij beetje te blijven proberen en coderen is het ons uiteindelijk toch gelukt.

Deze cursus heeft me ook wat geleerd over hoe het zou zijn om een groep te leiden, omdat ik een aantal keren binnen de groep mensen wat heb moeten bijsturen en op planning heb moeten letten.

Al met al vond ik dit dus een goede cursus die me veel geleerd heeft, maar ben ik toch ook wel blij om hierna weer wat vrije tijd te hebben.

6 Bijlage

De door ons geproduceerde code is terug te vinden op onze project pagina (de hoofdpagina) als een link naar een zip bestand met daarin de java bestanden, en benodigde resource files.