

Angry Pigs: Het verhaal achter de varkens.

David van Oorsouw, Matthijs Hendriks, Jesse Liebrand

29 juni 2011

Abstract

1 Abstract

In dit document vindt u de resultaten van een semester hard werk door twee studenten informatica en een student informatiekunde. Het bevat het eindverslag van een project van het vak Research & Development 1, waarin studenten leren in een groep een project te runnen. Het project omvatte het maken van een applicatie voor Android-telefoons. Hieronder volgen de specificatie en evaluatie van de uiteindelijke applicatie, en een reflectie op het project door de projectgroep.

Contents

1	Abstract	2
2	Documentatie	4
2.1	Inleiding	4
2.2	Specificaties	4
2.2.1	use-case-diagrammen	4
2.2.2	Class diagram	10
2.3	Structuur	11
2.4	Ontwerpverantwoording	11
3	Onderzoek	12
3.1	Inleiding	12
3.2	Onderzoeksvraag	13
3.2.1	Hoofdvraag	13
3.2.2	Deelvragen	13
3.3	Hypothese	14
3.4	Materialen & werkwijze	15
3.4.1	Observeren	15
3.4.2	Loggen	15
3.4.3	Evalueren	15
3.4.4	Vragenlijst	16
3.4.5	Omzetten naar resultaten	18
3.5	Resultaten	19
3.6	Conclusie	20
4	Reflectie	22
4.1	Product	22
4.2	Proces	22

2 Documentatie

2.1 Inleiding

Peckman is een game ontwikkeld voor Android-apparaten. Het doel van de applicatie is dan ook om de gebruiker te vermaken. De game draait om een vogel die zijn eieren terug moet pakken van een stel boosaardige varkens. De belangrijkste feature van de game is de speldynamiek, die ervoor zorgt dat de speler vermaakt wordt en blijft. Met speldynamiek bedoelen we het doel van het spel (de eieren verzamelen) en de interactie met de spelwereld (zowel het ontwijken van de tegenstanders als het oppakken van de powerups). Een ander zeer belangrijk onderdeel is de mogelijkheid verschillende besturingsmethoden te gebruiken. Door de speler de keuze te geven, maken we het spel speelbaar voor iedereen. Daarnaast is dit ook wat onze app smartphone-specifiek maakt.

2.2 Specificaties

De specificatie van de applicatie wordt geleverd in de vorm van use-case diagrammen en klassediagrammen.

2.2.1 use-case-diagrammen

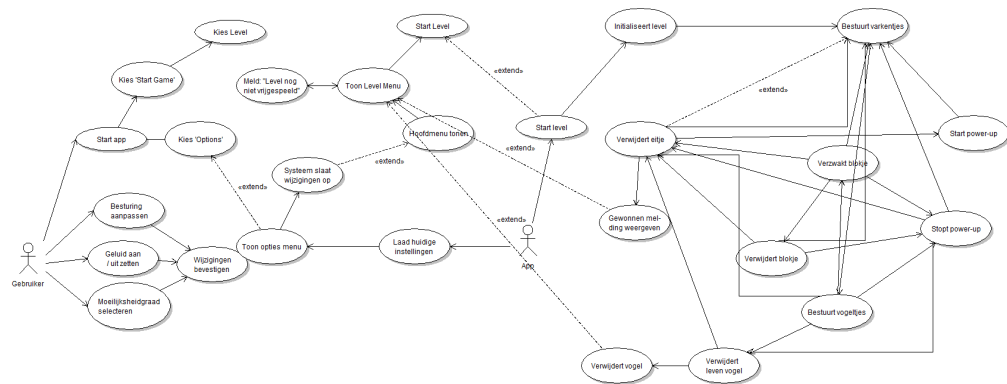


Figure 1: use-case survey

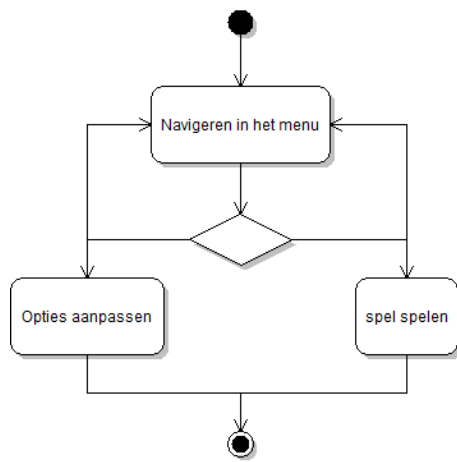


Figure 2: Sterk versimpelde activity-diagram van het totaal

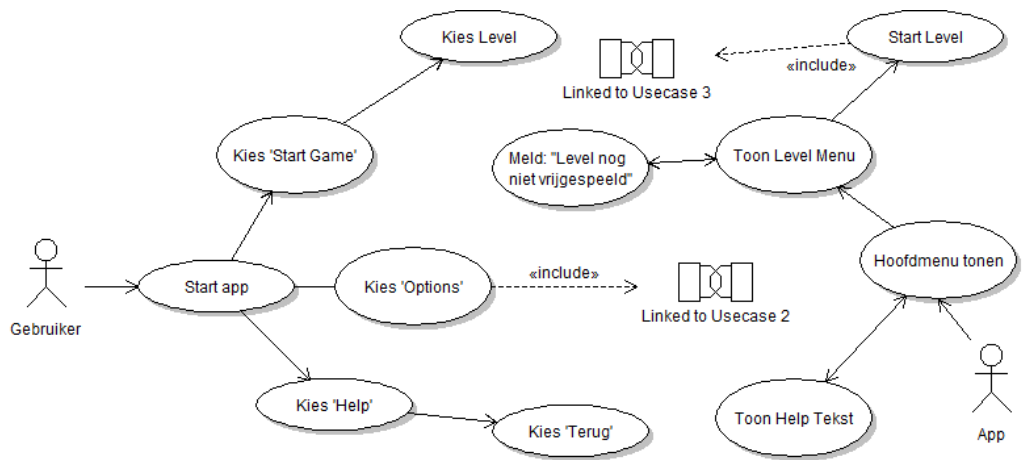


Figure 3: use-case 1

UC# :	1
Naam:	Menunavigatie
Primaire Actor:	Gebruiker van de app (speler van de game).
Trigger:	Opstarten van de app.
Basic course of events	
stap	actie
1	De app toont het hoofdmnu (beschreven in hid).
1.1	De speler selecteert "start game".
1.1b	De speler selecteert opties → uc 2.3.
1.1c	De speler selecteert help.
2	De app toont het levelmenu (beschreven in de hid).
2.1	De speler selecteert level 1.
3	De app start de wereld van level 1 op.
Alternative paths	
2.1.1	De speler selecteert level 2
3a	De app start de wereld van level 2 op
3b	De app geeft de melding dat level 2 nog niet beschikbaar is
Precondities	De app staat geïnstalleerd op de android-telefoon van de gebruiker. De app is (succesvol) opgestart en bevindt zich in het hoofdmnu.
Postcondities	Het level is opgestart
Business Rules	-
Supplementary Requirements	-
Opmerkingen	-
Openstaande punten	-

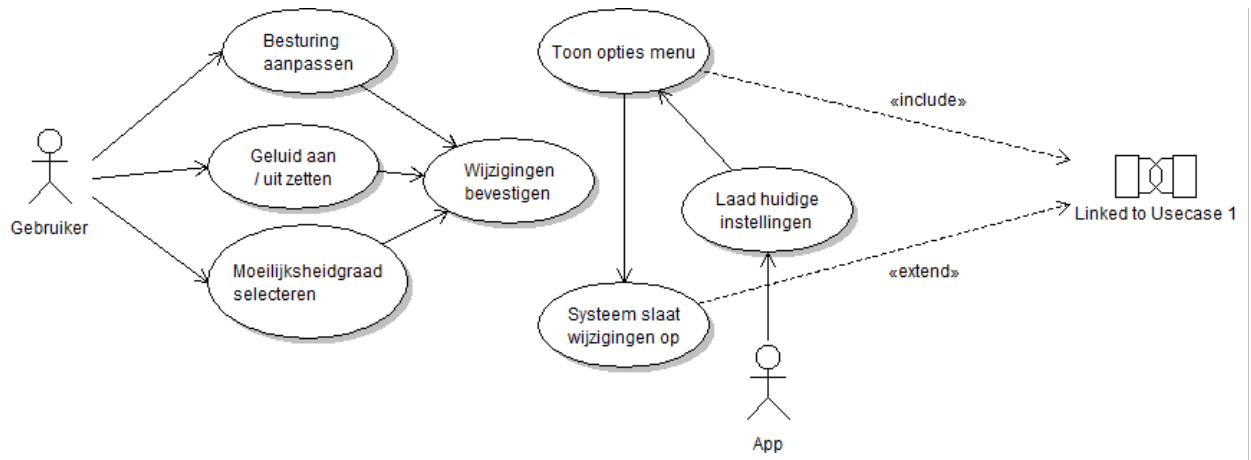


Figure 4: use-case 2

UC# :	2
Naam:	Opties aanpassen
Primaire Actor:	Speler.
Trigger:	Kiezen van "opties" in het hoofdmenu.
Basic course of events	
stap	actie
1a	De app laadt alle ingestelde opties in
1b	De app toont het opties-menu (beschreven in hid)
2a	Speler past besturing aan (scherm)
3a	Systeem selecteert de optie
2b	Speler zet geluid uit (standaard aan)
3b	Systeem selecteert de optie
2c	Speler selecteert moeilijkheidsgraad (standaard op normaal)
3c	Systeem selecteert de optie
4	Speler selecteert ok
5	Systeem past alle gekozen opties aan
6	Systeem laadt hoofdmenu in → uc1.1
Alternative paths	
Precondities	
Postcondities	De app is aangepast a.d.h.v. de door de gebruiker gewijzigde opties
Business Rules	-
Supplementary Requirements	-
Opmerkingen	-
Openstaande punten	-

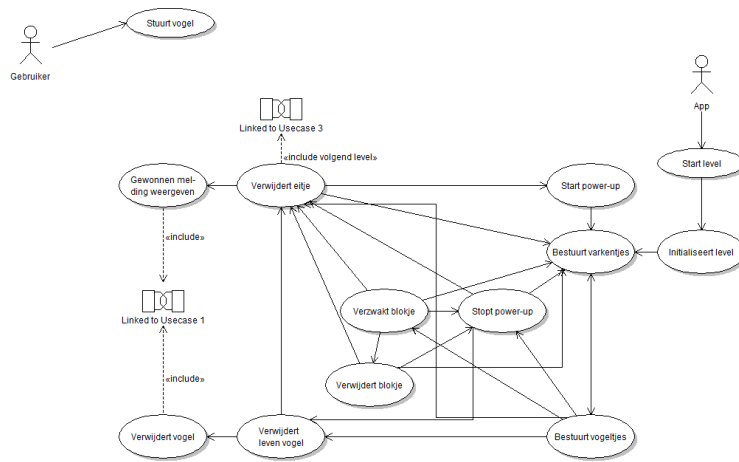


Figure 5: use-case 3

UC# :	3
Naam:	Spelen van level
Primaire Actor:	Speler.
Trigger:	Level gestart via menu (uc 1.3).
Basic course of events	
stap	actie
1.1	Speler stuurt (vogel)
2	Systeem laat vogel bewegen naar de sturing van speler
3a	Speler laat vogeltje over een ei heen lopen
3a1	Systeem haalt eitje weg
3b	Speler loopt over powerup
3b1	Systeem laat vogel veranderen in vogel bedoeld door powerup
3b2	Systeem haalt powerup (eitje) weg
3c	Speler loopt tegen big aan
3ca	Systeem verwijdert leven → uc3
3ca1	Systeem stopt power ups
3cb	Systeem heeft geen levens meer → uc1.2
3d	Speler loopt tegen muur
3d1a	Systeem zorgt dat muur verdwijnt
3d1b	Systeem zorgt dat muur verzwakt
3d1c	Systeem zorgt dat muur blijft staan
4	Speler heeft alle eitjes gepakt
5	Systeem start volgend level
Alternative paths	
5.1	Systeem geeft melding dat speler heeft gewonnen
5.2	Systeem geeft hoofdmenu weer → uc 1.1
Precondities	
Postcondities	Speler gaat dood Speler speelt level uit
Business Rules	-De vogel mag 3 keer dood gaan, deze heeft dus 3 levens (geraakt worden door het varkentje) newline -Als alle eitjes gepakt zijn en de speler is niet dood dan is het level gewonnen -Afhankelijk van het soort vogel kun je door bepaalde blokjes wel of niet heen
Supplementary Requirements	-
Opmerkingen	Voor alternative path geldt dat als er geen levels meer zijn dat stap 5.1 gestart wordt. De game is dan als het ware uitgespeeld. Daarna belandt men weer in het hoofdmenu.
Openstaande punten	-

2.2.2 Class diagram

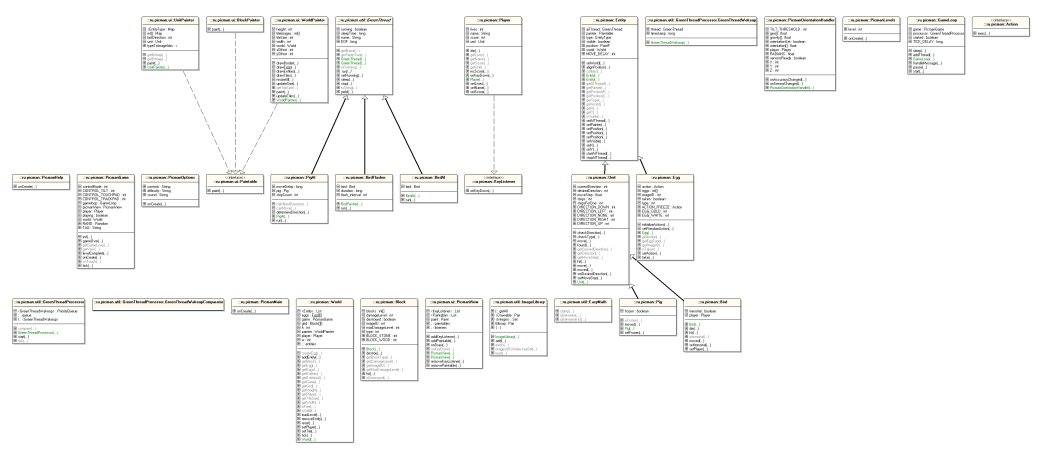


Figure 6: Volledige class diagram

Om deze afbeelding op volledige grootte te bekijken, klik hier.

2.3 Structuur

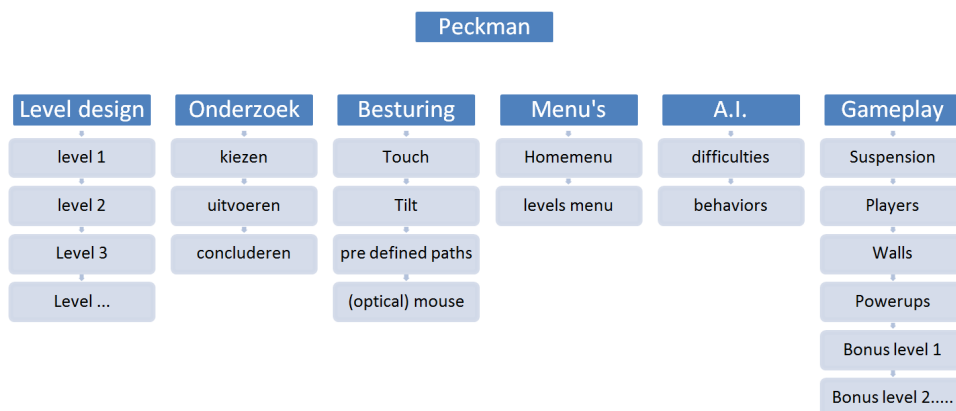


Figure 7: Product Breakdown Structure

De gedetailleerde structuur van het systeem is te vinden in onze JavaDoc.

2.4 Ontwerpverantwoording

Het ontwerp dat we hebben gekozen heeft een aantal grote voordelen. In het volgende stuk vindt u de verantwoording voor het gekozen ontwerp. Allereerst hebben we de indeling in verschillende onderdelen gekozen zodat meerdere mensen gemakkelijk aan afzonderlijke onderdelen konden werken zonder op een ander onderdeel te hoeven wachten. Dit hebben we gedaan door onderdelen zo min mogelijk afhankelijk te maken van elkaar. Zo kan er aan een bepaalde invoermethode gewerkt worden door één iemand, terwijl een ander iemand bijvoorbeeld aan het Player-object werkt.

Daarnaast is het ontwerp robuust. Het is gemakkelijk een nieuw spelelement te introduceren zonder dat er al te veel aan de bestaande klassen gesleuteld hoeft te worden. Er kan simpelweg een nieuwe subklasse van de Entity- of Unit-klasse gemaakt worden, en deze kan aan de World-klasse worden toegevoegd en getekend worden. Ook een andere invoermethode kan gerealiseerd worden door de invoermethode in een klasse te definiëren, en het te besturen object aan deze klasse door te geven.

3 Onderzoek

3.1 Inleiding

In onze presentatie hebben wij al aangekondigd om te gaan onderzoeken of onze game leuk is. Dit lijkt een beetje kinderachtig maar eigenlijk moeten we eerst weten wat leuk voor een game betekend. Dit kunnen we vangen in een aantal kenmerken en subkenmerken van games.

Attractiviteit

- Lay-out/artwork/graphics
- Concept
- Speelduur tot uitspelen
- Diversiteit/mogelijkheden in de game

Speelbaarheid

- Niveau/moeilijkheidsgraad
- Besturing
- Werking van het geheel

Deze factoren zullen allemaal met elkaar in verband staan en elkaar beïnvloeden. Samen bepalen ze de funfactor van onze game. We hebben er voor gekozen om ons onderzoek te focussen op de speelbaarheid, en te vernauwen tot besturing. Het naar ons idee het zwaarst wegende punt in de speelbaarheid. We hebben hiervoor drie verschillende besturingen ontwikkeld.

Touch:

Het scherm is in kwadranten onderverdeeld die elk met een richting corresponderen. Door het scherm aan te raken in een bepaald kwadrant, zal de vogel van richting veranderen.

Tilt:

Door de telefoon licht te kantelen in de richting waarin de vogel zich moet verplaatsen.

Optical/Vierpuntsdruktoetsen:

Helaas beschikt tegenwoordig niet iedere telefoon meer over vierpuntsdruktoetsen omdat het aantal knoppen op een smartphone steeds meer wordt beperkt. Een nieuwe toevoeging van de laatste jaren is de optische muis (een leesknopje dat de richting van een beweging met de vinger over het knopje registreert).

3.2 Onderzoeksvraag

3.2.1 Hoofdvraag

Hoe groot is de invloed van de besturing op de funfactor van de game?

3.2.2 Deelvragen

1. Wat is de beste besturing voor onze game?
2. Is de beste besturing ook de handigste besturing?

3.3 Hypothese

De effectiviteit van een besturing valt en staat met de implementatie ervan. Een deel hiervan ligt bij de fabrikant van het toestel. Hoe gevoelig zijn de gebruikte componenten en hoe wordt hier mee omgegaan? Het andere deel bij de game Peckman. Wat gebeurt er met de inkomende signalen en hoe is dit geïmplementeerd? De besturing waarmee de hoogste scores worden gehaald zal het best werken. Echter, ook zal de besturing per mobiel toestel anders zijn, wat voor een andere ervaring zal zorgen.

3.4 Materialen & werkwijze

- 2(+) Android-toestellen (het liefst met optica mousel/vierpuntstoets) met app.
- 3(+) Vragenlijst
- 3(+) Testpersonen
- 2(+) Toestellogs

3.4.1 Observeren

Door te observeren willen we te weten komen hoeveel fouten de speler maakt in het besturen van de game. Dit doen we als volgt. De testpersoon mag eerst even oefenen met het spelletje om de werking te achterhalen. Vervolgens laten we de testpersoon het spel opstarten, en het spel één keer met elke besturingsmethode spelen. Alle spellen zullen op dezelfde moeilijkheidsgraad gespeeld worden. We zorgen hierbij dat de speler het hardop vertelt wat hij of zij doet.

We zijn hierbij vooral geïnteresseerd in missers.

Bijvoorbeeld: de proefpersoon gaat naar links maar wil naar boven, of de proefpersoon gaat te laat naar in de gewenste richting. We turven de volgende missers tot de speler game over gaat of het level uitspeelt:

- Verkeerde richting
- Reageert te laat
- Reageert niet

3.4.2 Loggen

Bij de vorige proef staan ook loggegevens opgeslagen. Bij elke game-over heeft de speler een aantal punten behaald en heeft de speler een bepaalde tijd over het level gedaan. Deze tijden en punten worden in de game bijgehouden en kunnen dan ook worden opgeslagen.

3.4.3 Evalueren

We laten de speler na de twee voorgaande onderzoeken de vragenlijst invullen:

3.4.4 Vragenlijst

Wat vond je van de het spelletje?

- leuk
- niet leuk, niet saai
- saai

Attractiviteit (door afnemer in te vullen)
1 2 3 4 5 6 7 8 9 10
lay-out / Artwork / Graphics
1 2 3 4 5 6 7 8 9 10
Concept (het idee achter het spel)
1 2 3 4 5 6 7 8 9 10
Speelduur tot uitspelen
1 2 3 4 5 6 7 8 9 10
Diversiteit / Mogelijkheden in de game
1 2 3 4 5 6 7 8 9 10
Speelbaarheid (door de afnemer in te vullen)
1 2 3 4 5 6 7 8 9 10
Niveau / Moeilijkheidsgraad
1 2 3 4 5 6 7 8 9 10
Besturing (ga uit van de best werkende besturing)
1 2 3 4 5 6 7 8 9 10
Werking van het geheel (navigatie vanaf opstarten tot afsluiten)
1 2 3 4 5 6 7 8 9 10

Wat is voor jou belangrijk in een spel?

Verdeel 15 punten in de onderstaande categorieën

- Lay-out/artwork/graphics :
- Concept :
- Speelduur tot uitspelen :
- Diversiteit/mogelijkheden in de game :
- Niveau/moeilijkheidsgraad :
- Besturing :
- Werking van het geheel :

Waren er verschillen tussen dezelfde soort besturing op de verschillende toestellen? Kruis ja of nee aan bij de soorten besturing.

Touch

- Ja
- Nee

Tilt

- Ja
- Nee

Optical/vierpunts

- Ja
- Nee

Je hebt de hoogste score (= .) gehaald met de .. besturing op de , vond je deze besturing ook het beste werken?

- Ja
- Nee

Zo niet, waarom niet? Waarom was die andere besturing beter?

.....
.....
.....

3.4.5 Omzetten naar resultaten

Voor het omzetten naar resultaten zijn de volgende tabellen gebruikt:

Testpersoon x	tilt Toes- tel x	tilt Toes- tel Y	touch Toestel x	touch Toestel Y	optical/vpt Toestel X	optical/vpt Toestel Y
Tijd (sec)						
Punten						
Verkeerde richting						
Te laat						
Reageert niet						
Totaal						
Fouten per minuut						

Wat vindt de speler belangrijk in een game?				
Speler	1	2	3	totaal
lay-out				
Concept				
Speelduur				
Diversiteit				
Niveau				
Besturing				
Werking				
Berekende attractiviteit				
Berekende speelbaarheid				

Reflectieonderzoek				
Testpersoon	1	2	3	gemiddeld
Attractiviteit				
Speelbaarheid				
lay-out				
Concept				
Speelduur				
Diversiteit				
Niveau				
Besturing				
Werking				

3.5 Resultaten

Dit onderzoek is gedaan met twee Android-toestellen. De htc desire Z en de Samsung galaxy S. Er zijn drie testpersonen die hebben meegewerkt aan het onderzoek. Helaas heeft de Samsung geen optische muis en ook geen vierpunts-druktoets waardoor het testen van deze besturingsmethode op dit toestel niet mogelijk was.

Voor de gedetailleerde testresultaten klikt u hier.

3.6 Conclusie

We hebben nu gezien wat de testpersonen in het algemeen belangrijk vinden in een game. Als we kijken naar de verhouding tussen attractiviteit en speelbaarheid zien we dat het belangrijker is om te komen met een creatief idee dan met een goed werkende game. Gebruikers zullen de kleine foutjes eerder voor lief nemen als het spel er goed uitziet. Dit is goed te zien in het volgende diagram, welke de verhouding toont tussen attractiviteit en speelbaarheid. Hier is duidelijk in te zien dat de speelbaarheid ondergeschikt is aan de attractiviteit. Onze game heeft zowel op attractiviteit en speelbaarheid afgerond dezelfde score

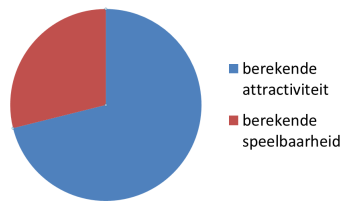


Figure 8: Wat gebruikers zoeken in een game

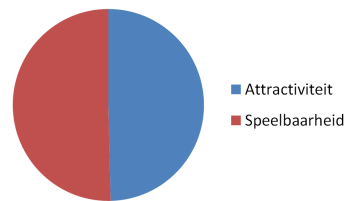
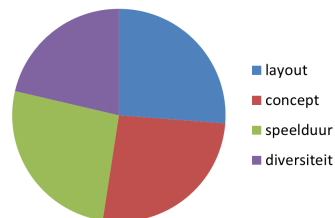
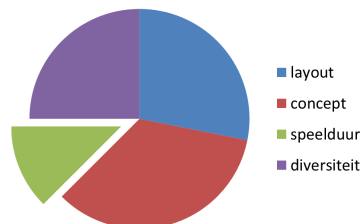


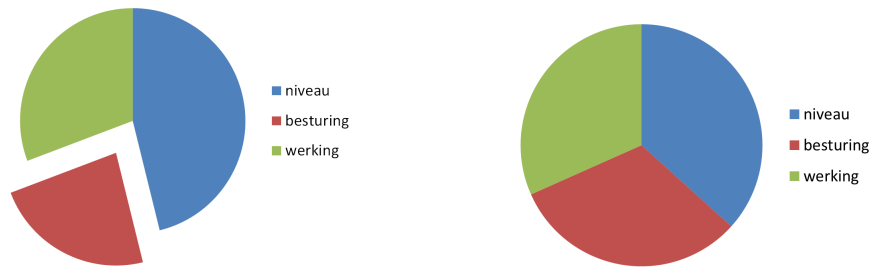
Figure 9: Verhouding tussen scores van onze game

behaald (7 gemiddeld). In zijn geheel werd hij gemiddeld beoordeeld in vergelijking met andere Android-spellen. Een reden hiervoor zou kunnen zijn dat de attractiviteit van de game niet erg hoog is, door onder andere weinig variatie en mogelijkheden in de game.

Onze testpersonen vonden het concept en de lay-out het belangrijkste in games. Zoals hieronder te zien. En als we kijken naar de punten die gegeven zijn aan onze game zie je dat deze ook maar gemiddeld een beetje hetzelfde zijn. Het hoogste cijfer is in onze games voor de speelduur terwijl dat eigenlijk volgens de testpersonen het minst belangrijk is. Op lay-out en concept scoren we dan wel weer goed.

De besturing is het minst belangrijk voor de speelbaarheid. Ook is te zien dat de cijfers wederom dicht bij elkaar liggen en er geen uitschieters zijn. We hebben hoog (7,3) gescoord op niveau en dat was voor onze testpersonen erg belangrijk. Ook werkte de game goed genoeg en worden de paar bugs die er in zaten gecompenseerd door de rest.





We hebben natuurlijk ook gekeken naar welke besturing het beste werkt, ofwel welke besturing het minste aantal fouten per minuut had. Dit is overduidelijk de touch-besturing. De optical deed het iets slechter en daarna kwam pas de tilt-besturing waarbij het aantal fouten verdrievoudigde.

Op de toestellen is te zien dat de besturing toch kan verschillen. Volgens onze testpersonen werkt de touch-besturing exact hetzelfde, maar werkt deze onhandiger op een groter scherm.

De verschillen in de tilt-functie waren minder groot en waarschijnlijk verklaarbaar door het verschil in het uitlijnen/ijken van de sensoren.

Helaas kunnen we over de optische muis/vierpuntsdruktoets niet veel zeggen aangezien de samsung niet over dergelijke invoermogelijkheden beschikt.

De gemakkelijkste besturing is ook de touch-besturing geworden. Er werden de minsten fouten mee gemaakt, en er werden de hoogste scores mee behaald. Ook kon je in het observeren terugzien dat de spelers deze invoermethode het snelst onde de knie hadden.

In Peckman is het zo dat de best werkende besturing goed aansluit op de moeilijkheidsgraad. Daarom wordt de game dan ook gemiddeld als leuk ervaren. Dit wordt ondersteund door de hogere attractiviteit van de game, want het idee spreekt mensen toch wel aan. Aan de cijfers te zien is het ook grafisch voldoende voor een mobiele game.

Voor Peckman werkt de touch-besturing het best. Soms tikte de proefpersonen wel mis, maar hier stond tegenover dat er nooit een verkeerde richting op werd gegaan. De leukste besturing was de besturing waar de meeste punten mee werden gehaald. Tot slot kunnen we stellen dat de besturing een game niet kan breken tenzij deze gewoonweg niet werkt, wat in Peckman niet het geval is. In het begin werd er al gesteld dat al deze begrippen verband met elkaar houden, en een slechte besturing heeft natuurlijk invloed op de moeilijkheidsgraad van een spel. Dus als een besturing niet werkt moet de moeilijkheidsgraad hier ook op worden aangepast. De besturing bepaald maar een klein deel van het eindoordeel van de gebruiker. De gebruiker heeft liever een leuk idee in een mooi jasje.

4 Reflectie

4.1 Product

Over het product zijn we al met al zeer tevreden. We hebben (bijna) alles wat we wilden implementeren ook geïmplementeerd, en het uiteindelijke product werkt naar behoren.

We zijn vooral tevreden met de structuur van de code, omdat het overzichtelijk is, en het makkelijk is er een element aan toe te voegen. Het laat de ruimte voor uitbreiding en verbetering. Ook het documenteren van functies en klassen is goed gebeurt, dit is gereflecteerd in de JavaDoc die is meegeleverd.

Wat beter kon is de collisions met de Pigs. Deze is net iets te ruim, en zorgt soms dat de speler een leven verliest waar dit niet had moeten gebeuren. We hadden helaas geen tijd meer om dit te verbeteren.

Ook de tilt-besturing laat ruimte over voor verbetering. Momenteel worden breekbare blokken in één keer gebroken omdat een tilt-event elke keer als er beweging is wordt gepost, en elke keer de richting van de vogel wordt aangepast. Als de vogel stilstaat na botsing met een blok, en de hoek van het toestel nog genoeg is om de vogel in de richting van het blok te sturen, zal de richting vrijwel onmiddellijk weer worden aangepast. Zo barst de vogel in een fractie van een seconde door een blok heen, waar met andere invoermethoden de speler eerst weg moet bewegen van het blok, om er met een aanloop nogmaals tegenaan te botsen voordat het breekt.

Het uiteindelijke product verschilt niet veel van hoe we het ons hadden voorgenomen. We hadden wel plannen om nog meer verschillende powerups te implementeren, en eventueel zelfs geluiden. Dit is helaas niet gelukt vanwege tijdgebrek.

4.2 Proces

Het proces verliep soms ietwat hobbelig, maar uiteindelijk hebben we het gevoel dat het over het algemeen goed ging. Er waren af en toe wat communicatiefouten, zoals onduidelijkheden over wie wat zou doen. Hierdoor bleef er soms wat werk liggen, en moest er gehaast worden om deadlines te halen.

Ook was er soms een algeheel gebrek aan communicatie. Dit werkt niet bevorderlijk. Er werden hier en daar wat te weinig afspraken over taken gemaakt.

Gelukkig ging het vaak wel erg goed, we hadden weinig moeite met het verdelen van de taken, iedereen wist (meestal) wat hem te doen stond, en wat gedaan moest worden werd ook gedaan.

Het verliep niet veel anders dan we verwachtten, maar er zijn wel een hele hoop dingen die beter kunnen in toekomstige projecten. Bijvoorbeeld een vaste dag afspreken om aan het project te werken, of eens in de week een korte evaluatiesessie houden. Ook concretere afspraken maken over deadlines is een goed idee, misschien zelfs een projectleider aanwijzen die taken verdeelt.