

Afstudeerverslag

Mobile Game Development
Nederlanden van Nu
Reinier van der Smeede
08016399

Projectleider : Reinier van der Smeede
Opdrachtgever : Arjen Dijkstra
Datum : 22-10-2012

Voorwoord

Dit is het afstudeerverslag van Reinier van der Smeede, student aan de Haagse hogeschool, academie voor ICT en Media te Zoetermeer; Het is het eindproduct van mijn leerzame afstudeerperiode bij De Nederlanden van Nu.

Al sinds vroeg in mijn jeugd heb ik een brede interesse op het gebied van videospellen op de computer en deze hobby is inmiddels doorgegroeid naar een professionele ambitie. Vandaar dat ik vooraf aan mijn afstudeerperiode druk op zoek ben geweest naar het vinden van een stage waar ik bezig kon zijn aan het schrijven van een spel. Zo ben ik uiteindelijk terechtgekomen bij De Nederlanden van Nu, een online verzekeraar die graag een mobiel spel als reclame wil aanbieden; Een goede kans voor mij om de ervaring op het gebied te vergroten!

Vooraf aan het daadwerkelijke document wil ik graag iedereen bedanken die hierbij een als ondersteunende rol bij betrokken is geweest. Natuurlijk hoort hier Arjen Dijkstra genoemd te worden: zowel mijn opdrachtgever als de directeur van het bedrijf die mij deze opdracht aan heeft geboden. Ik heb de samenwerking met Arjen als zeer prettig ervaren.

Ook mijn docenten terug in Zoetermeer wil ik bedanken, met name de docenten direct betrokken bij mijn afstudeerperiode: Remco Ruijsenaars en Vincent Broeren, de feedback op mijn afstudeerverslag ben ik vooral dankbaar voor.

Tenslotte hoop ik dat alle geïnteresseerden het document prettig leesbaar vinden en de ervaring van de afstudeerperiode gedeeltelijk mee kunnen krijgen.



Inhoud

Inleiding	4
Sectie I : Achtergrond	5
<i>1.1 Het bedrijf</i>	<i>5</i>
<i>1.2 De opdracht</i>	<i>7</i>
<i>1.3 Competenties</i>	<i>8</i>
Sectie II : Werkwijze	9
2.1 Projectmanagement	9
2.2 Game Design	17
2.3 Technisch Ontwerp	20
2.4 Constructie	28
2.5 Testen	35
Sectie III : Evaluatie	42
3.1 Product Evaluatie	42
3.2 Proces Evaluatie	44
3.3 Persoonlijke Evaluatie	45
Sectie IV : Bijlagen	46
A. Bronnenlijst	46
B. Afstudeerplan	47
C. Plan van Aanpak	52
D. Game Design Document	57
E. Master Test Plan	82
F. SCRUM Documenten	91
G. Feedback formulier bedrijfsmentor	106



Inleiding

Een afstudeerperiode of stage is een langdurig en omvangrijk proces, vaak onder weinig of geen toezicht van een docent, waardoor een objectieve beoordeling door een derde lastig wordt. Om deze reden is het van belang dat elke student het proces van zijn of haar stage geheel documenteert.

Het doel van dit verslag is het presenteren van de werkwijze die de student gedurende de afstudeerperiode heeft gehanteerd. Na het lezen van het verslag is de lezer voldoende geïnformeerd om de professionaliteit van de student te beoordelen; Om dit te bereiken is het verslag, afgezien van de evaluatie, objectief opgesteld en verwijs ik naar mijzelf als 'de student' in plaats van 'Reinier' of 'mijzelf'.

Indeling

Het document is ingedeeld in een aantal secties, die elk onderverdeeld zijn in hoofdstukken, Sectie I gaat in op de achtergrond van het project en biedt een omschrijving van het bedrijf, de probleemstelling, de opdracht. Het behandelt ook het afstudeerplan en de competenties die de student aantoonst in dit verslag.

Voor een omschrijving van de werkwijze van de student kan de lezer terecht in sectie II, waar de werkzaamheden in een logische volgorde zijn opgedeeld in hoofdstukken. Het gehele proces van projectmanagement, het design van de applicatie, de constructie ervan en het testproces van de code komt aan de orde.

Sectie III evalueert het gehele proces en de student reflecteert op zowel het proces, het product als zichzelf.

In Sectie IV zijn alle relevante bijlagen opgenomen die de lezer optioneel kan bekijken voor dieper inzicht in de details; Hier kan de lezer een bronnenlijst en de (deel)producten verwachten die zijn opgeleverd.

Voorbeelden

In het document staan op verschillende plaatsen concrete voorbeelden van het proces in gekleurde kaders, waar er dieper wordt ingegaan hoe de stof in het hoofdstuk is toegepast in het persoonlijke project. Deze voorbeelden zijn vooral uitzonderingen op het gehele proces of verduidelijkingen van het behandelde.

Bronnen

In een aantal hoofdstukken worden verwijzingen naar bronnen gegeven in de volgende vorm van een nummer tussen gesloten haken, zoals volgt : [1].

Een complete bronnenlijst, en de volledige bronverwijzing, kan de lezer vinden in bijlage A.



Sectie I : Achtergrond

1.1 Het bedrijf

Het hoofdstuk introduceert het bedrijf waar de student werkzaam is geweest en heeft als doel om de context te schetsen; Zowel de medewerkers als een schets van de strategie komen aan de orde.

1.1.1 Generali

De student heeft als stagiair gewerkt bij De Nederlanden van Nu, een dochterbedrijf van Generali Verzekeringsgroep NV.

Generali Verzekeringsgroep NV is een traditionele verzekeraar die in Nederland schadeverzekeringen, inkomensverzekeringen, levensverzekeringen en pensioenverzekeringen aanbiedt via tussenpersonen. Generali richt zich vooral op inkomensverzekeringen en pensioenen omdat dit ingewikkelde verzekeringen zijn waarvoor klanten zich vaak voor advies naar een tussenpersoon wenden. De rechtsvoorgangers van Generali Verzekeringsgroep NV zijn sinds 1870 actief op de Nederlandse verzekeringsmarkt.

Generali is met ongeveer 1% marktaandeel een relatieve kleine verzekeraar in Nederland. Klein betekent bij verzekeraars in dit geval ongeveer 450 medewerkers en een jaaromzet van ongeveer 500 miljoen euro.

Generali maakt deel uit van de Generali Group, een groot internationaal concern met ongeveer 90.000 medewerkers in meer dan 50 landen. Het hoofdkantoor van de Generali Group staat in Trieste, Italië.



1.1.2 De Nederlanden van Nu

De Nederlanden van Nu is een zelfstandige verzekeringsdochter van Generali, die zich onderscheidt van Generali door het aanbieden van de verzekeringen zonder gebruik te maken van tussenpersonen; Het bedrijf biedt verzekeringen rechtstreeks aan consumenten aan door middel het internet. De Nederlanden van Nu bestaat sinds 2006; vanaf medio 2011 worden via internet verzekeringen aangeboden en heeft ongeveer 2000 klanten.

Team

De Nederlanden van Nu bestaat uit een kernteam van zeven personen met als belangrijkste verantwoordelijkheden :

1. Online Marketeer (Marktbewerking, website)
2. Promotie Marketeer (Marktbewerking, website)
3. Marketing Analist (Analyse / optimalisatie, website)
4. Productmanager (aanbod, crowdsourcen, website)
5. Directeur
6. Stagiair / afstudeerder 1
7. Stagiair / afstudeerder 2

Belangrijke processen, waaronder klantcontact, verzekeringsadministratie en het behandelen van schadeclaims zijn uitbesteed waardoor hier geen medewerkers voor nodig zijn. Ook mag de Nederlanden van Nu desgewenst gebruikmaken van de stafafdelingen van Generali, waaronder Juridische Zaken, Compliance, Risk Management, Actuarieel en Human Resource Management. Om deze communicatiestromen zo eenvoudig mogelijk te houden, maar wel zelfstandigheid uit te stralen zijn alle medewerkers van De Nederlanden van Nu gevestigd in een pand naast het pand waarin Generali is gevestigd.

Strategie

Een aantal elementen uit de strategie van De Nederlanden van Nu zijn:

1. Klanten echt betrekken door middel van technieken zoals crowdsourcing
2. Focus op onbetaald verkeer door ontraditionele marketing
3. Veel en snel experimenten ("Try a lot of stuff and keep what works")



1.2 De opdracht

Dit hoofdstuk behandelt de opdracht die de student heeft uitgevoerd. In de probleemstelling wordt de situatie bij aanvang van de opdracht gepresenteerd, waardoor duidelijk wordt waarom deze opdracht nodig is geweest.

Verder behandelt het hoofdstuk de doelstelling en de afbakening van de opdracht.

1.2.1 Probleemstelling

De online verzekeringsmarkt bestaat nog niet heel lang, maar is inmiddels wel behoorlijk volwassen. Naast gevestigde partijen zoals Centraal Beheer, OHRA, FBTO en Unive springen er in de afgelopen vijf jaar nieuwe partijen met fors geweld de markt op (Ditzo, InShared, Allsecur). Ook vergelijkingssites als Independer claimen een steeds belangrijke positie in de markt.

De Nederlanden van Nu heeft niet het budget om via langdurige massamediale campagnes aan naamsbekendheid te werken. Bovendien is De Nederlanden van Nu niet in staat om een sluitende business case te maken voor deze marktbenadering. De Nederlanden van Nu gelooft dat partijen zoals Ditzo, InShared en Allsecure aanzienlijke verliezen accepteren om aan marktaandeel te komen. Dit is voor De Nederlanden van Nu niet acceptabel.

De hamvraag is: Hoe kan De Nederlanden van Nu zonder aanzienlijke budgetten te steken in massamediale campagnes, toch werken aan naamsbekendheid en in aanraking komen met mogelijke nieuwe klanten?

De Nederlanden van Nu gelooft dat dit mogelijk is door ontraditionele marketing. Bovendien gelooft De Nederlanden van Nu niet in wedden op een paard maar in de portfolio-benadering van innovatie: Door snel tien dingen op de markt te brengen, wordt eerder succes bereikt dan door één ding heel lang en zorgvuldig voor te bereiden.

Het is acceptabel dat er van de tien dingen acht niet succesvol zijn onder de voorwaarde dat er van elk van de mislukkingen een goede evaluatie wordt gemaakt zodat het leereffect niet verloren gaat.

1.2.2 Doelstelling

Het ontwikkelen van een game voor een mobiel platform (iOS) is een van de experimenten binnen de strategie van veel verschillende dingen proberen. De Nederlanden van Nu hoopt door een opmerkelijke game aandacht te genereren voor haar merk, met als doel meer verkeer naar haar website.

De opdracht omvat alle fasen van game development waaronder het bedenken van het concept, de design van het game, implementatie, testen en overdracht. Uiteraard wordt verwacht dat alle goede ontwikkel-standaarden in ere worden gehouden.



1.2.3 Afbakening

Het project omvat enkel de ontwikkel- en testfase van de applicatie, met het bijbehorende voortraject; Het beheer van de game valt buiten de scope van het project. Dit is een belangrijke reden waarom de broncode goed gecommmentarieerd en uitbreidbaar achtergelaten wordt, zodat het project in de toekomst gemakkelijk op te pakken is door een extern persoon.

Eventuele PR aandacht, of het ontbreken hiervan, ligt buiten de verantwoordelijkheid van de student.

1.3 Competenties

Door het uitvoeren van de opdracht heeft de student vooraf aan het traject de onderstaande competenties beloofd aan te tonen¹. In Sectie II van het verslag wordt aangetoond dat deze competenties zijn behaald. Voor elke competentie is een specifiek hoofdstuk ingericht, enkele onderwerpen van een competentie omvatten meerdere hoofdstukken :

1. Uitvoeren Analyse door definitie van requirements (Zie hoofdstuk 2.2)
2. Ontwerpen Systeemdeel (Zie hoofdstuk 2.3)
3. Bouwen applicatie (Zie hoofdstuk 2.4)
4. Initiëren en plannen van het testproces (Zie hoofdstuk 2.5)
5. Uitvoeren van en rapporteren over het testproces (Zie hoofdstuk 2.5)

1. Bijlage B.



2.1 Projectmanagement

Het project is gemanaged via SCRUM, een AGILE projectmanagement-methodek specifiek ontwikkeld voor software-ontwikkeling. Dit hoofdstuk loopt een aantal aspecten van SCRUM langs en er wordt duidelijk gemaakt waarom de student de methodek heeft gekozen en waardoor deze op voldoende niveau is toegepast.

Van de lezer wordt verwacht dat al enige kennis van SCRUM aanwezig is;

Extra informatie over SCRUM is te vinden op de wikipedia pagina over SCRUM [1]

2.1.1 Waarom AGILE?

Waarom is er nu gekozen voor een AGILE aanpak voor het project? Dit is gebeurd door een aantal redenen, elke reden is afgesloten met een citaat uit het "Manifesto for Agile Development [2] :

1. Vooraf aan het project waren er nog geen requirements vastgesteld voor het project, van te voren een uitgebreid plan maken was daarom erg lastig. Er is gekozen voor een iteratief proces met kleine iteraties, zodat requirements gedurende het project gemakkelijk aangepast zouden kunnen worden.
"Responding to change over following a plan"
2. Er was geen projectgroep beschikbaar voor het ontwikkelproces, en het ontwerpen, coderen en testen van het systeem is allemaal door de student gedaan; Er was dus minder aandacht nodig voor documentatie. Bovendien is de klant meer tevreden met werkende software tegenover extensieve documentatie.
"Working software over comprehensive documentation"
3. In combinatie met nummer 1, het gehele idee voor de requirements is niet binnen een week te vormen; Een idee is tenslotte een aspect wat over de tijd groeit. De verschillende stakeholders betrekken in het game-design is voor dit project vanaf het begin een belangrijk punt geweest. Door elke iteratie contact met de stakeholder(s) te houden, is er mogelijk voor het idee om te groeien.
"Customer collaboration over contract negotiation"

Omdat de AGILE methodek zo goed aansloot bij dit project, en omdat deze methodek ook persoonlijk erg goed bij mij aansluit is er niet actief gezocht naar een alternatief.

Voorbeeld :

Tijdens het project is de AGILE aanpak meerdere keren goed in het licht gekomen, verandering van requirements was een constant proces.

Bij de start van de tweede sprint bleek dat het management van Generali twijfels had over de risico's die gelopen werden met het oorspronkelijke design van de game; Het zal te aggressief zijn. Veel features moesten uit het spel; Deze beslissing had echter geen consequenties voor de planning: Er werden enkel een aantal items uit de backlog gehaald, waardoor meer tijd was voor nieuwe features.

2.1.2 Rolverdeling

SCRUM onderkent een aantal rollen die betrokken zijn bij het proces. Alle betrokkenen worden langsgelopen en er wordt toegelicht wie welke rollen heeft ingevuld en waarom.

Product Owner

Arjen Dijkstra is de opdrachtgever van het project en de directeur van de Nederlanden van Nu. Voor het SCRUM proces heeft hij de rol ingevuld van 'Product Owner'. Als directeur van de Nederlanden van Nu heeft hij het contact met de stakeholders van Generali en kan hij het beste de klant representeren.

Het vormen van de backlog en de user stories is in nauwe samenwerking gebeurd met Arjen en is elke sprint meeting besproken. Ook het bepalen van de prioriteit van de user stories viel onder de verantwoordelijkheid van Arjen. Omdat Arjen zelf niet voldoende kennis had van SCRUM is dit in samenwerking gedaan met de student.

Development Team / Scrum Master

Vanzelfsprekend heeft de student de rollen ingevuld van het daadwerkelijke opleveren van de producten en het managen van het SCRUM proces.

Stakeholders

De belangrijkste stakeholder van het project was het management van Generali. Het management heeft gedurende het proces een aantal bijsturingen gedaan in verband met Risk Management. Het contact liep vooral via de product owner (Arjen Dijkstra).

Hiernaast zijn er een aantal mensen van het team van de Nederlanden van Nu die een bijvullende rol hebben bijgedragen aan het project, maar niet direct gecommitteerd aan het project waren; Met name de marketeers, die tijdens de brainstormen aanwezig waren voor ideeën over het design van de game.

2.1.3 Plan van Aanpak

Het project heeft niet vanaf dag 1 gelopen volgens de SCRUM methodiek. Er is voor gekozen om de eerste twee weken te besteden aan het maken van een traditioneel plan van aanpak, inclusief uitleg over SCRUM, en brainstormen over een game design.

De reden voor deze aanpak is simpel: de opdrachtgever was niet bekend met SCRUM; Hiernaast was er nog geen product backlog of een dergelijke lijst met requirements. Na twee weken was een product backlog te vormen en is gestart met de eerste sprint.



2.1.4 Product Backlog

De product backlog is een lijst met alle functionaliteit en taken die nog moeten gebeuren. De product backlog is in samenwerking met de product owner ontstaan, in eerste instantie door middel van brainstorm, en bestaat uit user stories zodat de product backlog gemakkelijk te begrijpen is door de opdrachtgever.

Elk van de user stories heeft een schatting in uren gekregen, deze schatting is gemaakt door de student. De prioriteit is hiervan nog niet bepaald, maar is voor elke sprint apart bepaald naar de wensen van de opdrachtgever. Het totaal aantal uren is onderaan de lijst opgesomd, om een beeld te geven van de omvang. Er zijn geen story points gebruikt, maar uren, om de lijst natuurlijker en makkelijker leesbaar te maken voor de opdrachtgever; Deze wordt tenslotte al geïntroduceerd aan genoeg nieuwe termen.

Op het moment dat een item uit de product backlog af is, is het verwijderd van de lijst zodat het voor de opdrachtgever een duidelijk beeld geeft van het werk wat nog gedaan moet worden. Dit houdt in dat voor elke sprint de product backlog is aangepast tijdens de sprint meeting (zie 2.1.5), en dus ook dat elke sprint een eigen versie van de product backlog heeft.

Om voor de opdrachtgever een duidelijk beeld te geven van de staat van zijn product, zijn in de product backlog enkel items opgenomen die betrekking hadden tot de kwaliteit van het product. Zaken zoals het afstudeerverslag zijn niet opgenomen in de product backlog.

‘Definition of Done’

De definition of done is besproken met de opdrachtgever tijdens het indelen van het SCRUM proces; Er is hierbij voor gekozen om de exit criteria voor een user story informeel te behandelen. De kleine schaal van het team maakte dit gemakkelijk en hierdoor was er meer kans op feedback van de opdrachtgever.

In eerste instantie is er gedacht aan de criteria om voor elke functionaliteit unit-tests te laten draaien en dat als exit-criteria te hanteren. Echter was het onduidelijk in hoeverre verschillende stukken functionaliteit getest zouden worden en of er wel tijd zal zijn om overall unit tests voor te schrijven.

Uiteindelijk is er besloten om met gezond verstand te bepalen of een functionaliteit / user story ‘af’ was. Tijdens de scrum meeting aan het begin van een sprint heeft de opdrachtgever de kans gekregen om de functionaliteit/user story te accepteren als ‘af’.

Ideal Hours

Alle uren in zowel de product als de sprint backlog zijn gemeten in zogenaamde ‘ideal hours’, uren die de student ongestoord aan de item kan besteden. Zodat er rekening wordt gehouden met uren die niet effectief zijn, zoals lunch, werken aan het afstudeerverslag etc.



Product Backlog bij aanvang van Sprint II

Ter illustratie van deze paragraaf kan de lezer op de volgende pagina de Product Backlog van de tweede sprint vinden (Figuur 1), met een aantal prioriteiten die door middel van de sprint meeting zijn bepaald. Aan het begin van de sprint zijn hieruit een aantal items geselecteerd voor de sprint backlog. De overige product backlogs bij aanvang van de andere sprints zijn te vinden in bijlage F.

Checklist			
ID	User Story	Est. Hr	Priority
☐ 1	Als een gebruiker wil ik een grafische weergave van spel-elementen op het scherm kunnen zien	8	1
☐ 2	Deze functionaliteit moet goed getest zijn.	8	
☐			
☐ 3	Als een gebruiker wil ik dat spel-elementen volgens natuurkundige wetten werken	8	2
☐ 4	Deze functionaliteit moet goed getest zijn.	8	
☐			
☐ 5	Als een gebruiker wil ik bommen kunnen wegschieten	12	
☐ 6	Deze functionaliteit moet goed getest zijn.	12	
☐			
☐ 7	Als een gebruiker wil ik de camera van het spel kunnen bewegen met mijn vinger	4	
☐ 8	Deze functionaliteit moet goed getest zijn.	4	
☐			
☐ 9	Als een gebruiker wil ik dat ik kan zien hoeveel bommen er nog over zijn	2	
☐ 10	Deze functionaliteit moet goed getest zijn.	2	
☐			
☐ 11	Als een gebruiker wil ik een score kunnen behalen en deze zien op het scherm	7	
☐ 12	Deze functionaliteit moet goed getest zijn.	7	
☐			
☐ 13	Als een gebruiker wil ik een eindweergave van het behaalde resultaat kunnen posten op facebook	20	
☐ 14	Deze functionaliteit moet goed getest zijn.	20	
☐			
☐ 15	Als een gebruiker wil ik mijn spel kunnen herstarten	6	
☐ 16	Deze functionaliteit moet goed getest zijn.	6	
☐			
☐ 17	Als een gebruiker wil ik geluiden horen	20	
☐ 18	Deze functionaliteit moet goed getest zijn.	10	
☐			
☐ 19	Testomgeving opzetten	9	3
☐			
☐ 20	Test Documentatie schrijven (Master test plan / Risicoanalyse)	12	4
☐			
☐ 21	Als een gebruiker wil ik mooie grafische elementen in het spel	26	5
☐			
☐ 22	Als een gebruiker wil ik mooie grafische elementen in het laatste scherm	6	
☐			
☐ 23	Ontwerpen Systeem (UML)	25	
☐			
☐	TOTAL Backlog Items		242

Figuur 1. Product Backlog bij aanvang van Sprint II

2.1.5 Sprint meeting

Het project is ontwikkeld door middel van sprints van twee weken. Aan het begin van elke sprint is een sprint meeting ingepland met de product owner en de student. Deze meeting combineerde de taken van de Sprint Planning Meeting, Sprint Review meeting en Sprint Retrospective van traditioneel SCRUM. Omdat zowel de projectgroep als het project zelf relatief klein is, was het te verwachten dat de meeting kort zal zijn; Hierdoor was het efficiënter om de meetings te combineren. Elke Sprint Meeting was dan ook tussen 30 minuten en 60 minuten te realiseren.

Tijdens de sprint meeting zijn de volgende punten besproken :

1. Allereerst is een volledig werkende versie van de applicatie gepresenteerd aan de opdrachtgever; Dit had twee doelen, allereerst kan de applicatie ter acceptatie worden gekeurd door de product owner en kan de product backlog mogelijk aangepast worden als gevolg. Ten tweede zodat de opdrachtgever altijd een werkende release in handen heeft, voor het geval dat er zodanig gevaarlijke bugs opspringen dat een release uitgesteld moet worden. Het presenteren van de applicatie werkte meteen als een user acceptance test.
2. Er wordt besproken wat er de afgelopen weken is gebeurd aan beide kanten, zijn er bijvoorbeeld contactpunten met de overige stakeholders geweest?
3. De burndown chart van de vorige sprint wordt gepresenteerd zodat de opdrachtgever een duidelijk beeld heeft van de progressie van de vorige sprint. De grafiek wordt kort geanalyseerd waardoor de volgende sprint gemakkelijk gepland kan worden met betrekking tot ideal hours.
4. Zijn er veranderingen in de requirements waardoor de product backlog aangepast moet worden?
5. Tenslotte wordt tijdens deze meeting een aantal backlog items uit de product backlog gekozen, hieruit vloeit de sprint backlog voor de komende twee weken.

Bij alle punten hierboven heeft de student de opdrachtgever zoveel mogelijk betrokken bij het proces en is een adviserende rol aangenomen voor zaken zoals het bepalen van de sprint backlog.

Voorbeeld :

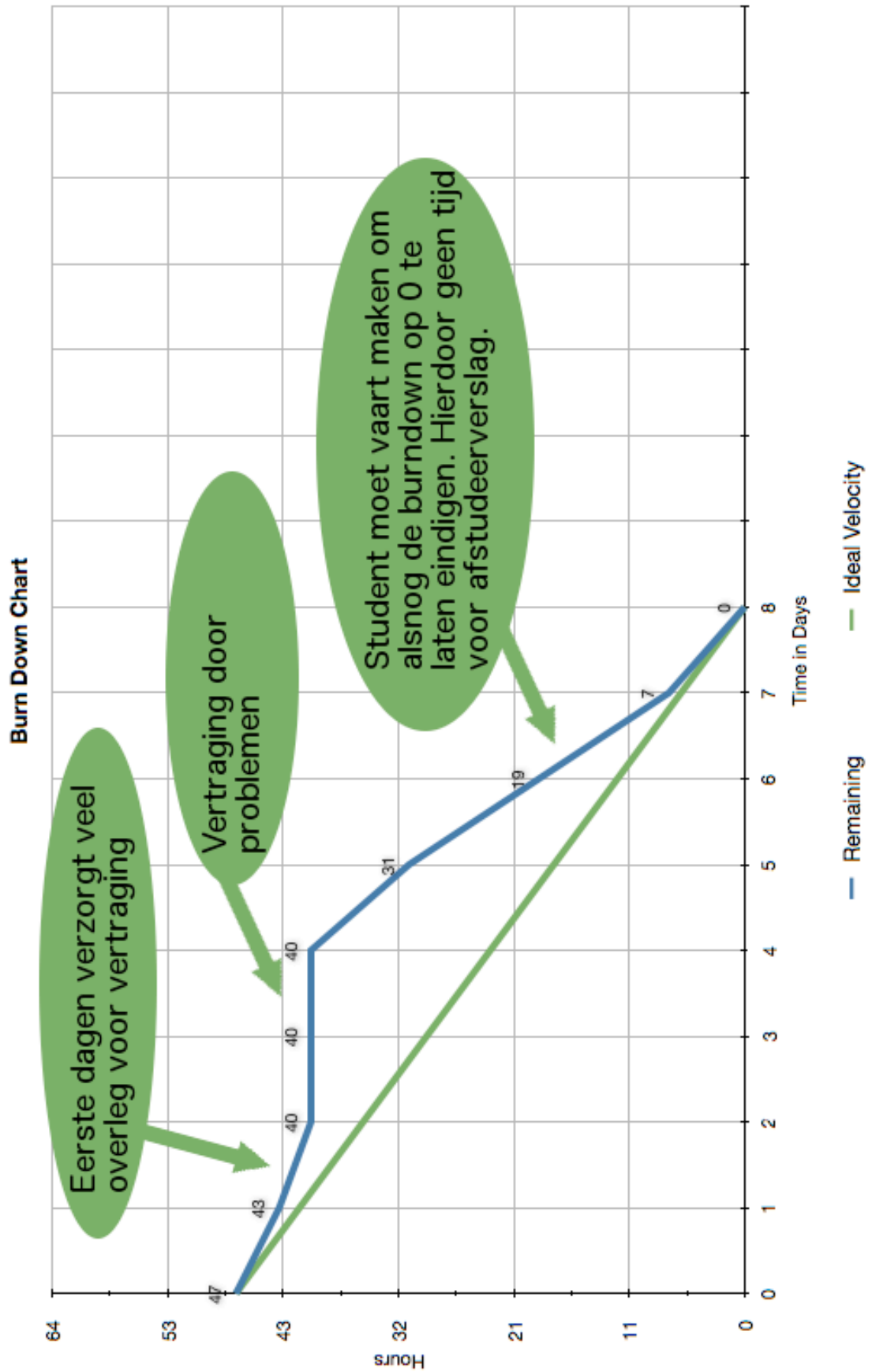
Na afloop van de allereerste sprint bleek dat er teveel uren waren ingepland na het presenteren van de burndown chart (Figuur 2);

De y-as van de grafiek toont het totaal aantal resterende uren, de x-as het aantal resterende dagen. De blauwe lijn representeert het aantal uren wat nog over is, en de groene lijn representeert de ideale snelheid.
(Zie volgende pagina)

De burndown chart laat zien dat alle backlog items af zijn, en dus is er een werkende release opgeleverd aan de opdrachtgever. Echter moest de student aardig de vaart erin houden, hierdoor was er niet genoeg tijd meer voor het afstudeerverslag deze sprint.

Hieruit viel te concluderen dat hierop-volgende sprint backlogs minder uren moesten bevatten, zodat de student meer tijd had om aan het afstudeerverslag te werken.





Figuur 2. Burn Down Chart : Sprint I

2.1.6 Sprint Backlogs

Tijdens elke sprint meeting is een sprint backlog opgesteld: een lijst van backlog items, die op prioriteit zijn gesorteerd. Deze sprint backlog was mijn belofte naar de opdrachtgever toe.

De items in de sprint backlog staan de gehele sprint vast en hier werd niks aan veranderd.

Van de verschillende items heeft de student zo klein mogelijke taken gemaakt zodat de burndown chart zo accuraat mogelijk kan worden weergegeven. Hierbij is gestreefd naar zo klein mogelijke taken;

In de sprint backlogs zijn ook taken meegenomen die niet altijd direct functionaliteit toevoegen aan het product, maar wel de kwaliteit verhogen, zoals een testomgeving opzetten. Het schrijven van het afstudeerverslag is uit de backlogs gehouden aangezien dit geen toegevoegde waarde heeft voor het product. Hiervoor is wel elke sprint rekening mee gehouden.

Figuur 3 toont een voorbeeld : de sprint Backlog van Sprint I. De overige sprint backlogs kan de lezer vinden in bijlage F.

	User Story	Tasks	Est. Hr
☐	Testomgeving opzetten		9
☐		Unit Test Omgeving	9
☐			
☐	38 Test Documentatie Schrijven		12
☐		Master Test Plan	7
☐		Risico Analyse	5
☐			
☐	39 Als een gebruiker wil ik mooie grafische elementen in het spel		26
☐		Sprites code	9
☐		Block.jpg	1
☐		Ground.jpg	3
☐		Background.jpg	3
☐		Bullet.jpg	1
☐		Explosion.jpg	2
☐		Cannon.jpg	2
☐		Hill.jpg	2
☐		Arjen.jpg	2
☐		Cloud.jpg	1
☐			

Figuur 3. Sprint Backlog : Sprint I



2.1.7 Presentatie

Tegen het eind van de vijfde sprint is de student gevraagd om een presentatie over SCRUM te geven aan het team; De opdrachtgever was namelijk onder de indruk van de effectiviteit van het projectmanagement.

Na de presentatie ontstond een discussie binnen het team over de mate waarin SCRUM bruikbaar zal zijn in het (non-ICT) team van De Nederlanden van Nu, met tegenstanders en voorstanders van het idee.

Inmiddels is er voor de voortgang op het gebied van ICT een SCRUM board aanwezig op het kantoor en worden er dagelijks een standup meeting gehouden om het team scherp op het doel te houden.



2.2 Game Design

2.2.1 Game Design Document

De functionele kant van de applicatie (het spel) is vastgelegd in een Game Design Document. Het Game Design Document is een levend document wat een blauwdruk voor het spel weergeeft en heeft ten doel het functioneel ontwerp vorm te geven. Het verslag beschrijft het spel in zijn geheel en zoveel mogelijk details zijn hierin opgenomen.

Het document is gedurende het hele proces van ontwikkeling constant aangepast en aangevuld, om het wel leesbaar te houden voor de developer is er geprobeerd de verschillende punten echter wel zo beknopt mogelijk te omschrijven.

Onderwerpen

Het Game Design Document beschreef als hoofdpunten de volgende onderwerpen :

- Achterliggende Filosofie
- Features
- Doelgroep
- Look and feel
- Gameplay
- Game Wereld
- Mechaniek
- Beschrijving Levels
- Doel Levels
- Kritieke pad Levels
- Eindcriteria Levels
- Screen Flow
- Verhaal
- Visuele Interface
- Concept Art
- Game Art
- Style Guide
- Technische Specificaties
- Technisch Ontwerp

Naast het omschrijven van de verschillende onderwerpen doelde het document ook naar het archiveren van de verschillende 'game assets'; Zo is er een hoofdstuk wat elk verschillende texture van het spel bevat, met elk een omschrijving van :

- Waarvoor is de texture bedoeld?
- Wat stelt de texture voor?

De informatie hoe een dergelijk document is samengesteld komt uit zowel het boek van Robert Bates [3] als een template van Mark Baldwin, een regelmatig geciteerde template, gevonden via google. [4]

Het volledige game design document kan de lezer vinden in bijlage D.



Game Design

De onderwerpen zijn door de student belicht en beschreven, en vervolgens bij de eerstvolgende sprint meeting besproken en eventueel aangepast. De ervaring heeft geleerd dat het document ongeveer elke sprint een iteratie doorgaan is; Het heeft in totaal zo'n 1 tot 2 uur per sprint gekost aan tijd.

Het idee voor de algemene game-design is ontstaan door meerdere brainstorms met zowel de opdrachtgever als de marketeers van het team van De Nederlanden van Nu.

Centraal punt

Door zoveel mogelijk details van het spel te beschrijven is er zodanig genoeg informatie op een centrale plek te vinden dat een lezer een beeld kan vormen van het spel, zonder dat het daadwerkelijke spel gepresenteerd hoeft te worden; Hierdoor kan een enkel document worden gestuurd naar een stakeholder zodat beslissingen over het design gemaakt kunnen worden.

Voorbeeld :

Het GGD bevat een concept art tekening van het laatste scherm, waar de speler een foto kan uploaden naar facebook.



Als reactie op deze concept art is er door het management uiteindelijk besloten dat er een feature van het spel uit het ontwerp moest; Namelijk de mogelijkheid om een foto van te voren te uploaden naar het spel, zodat de speler vervolgens een gebouw met het logo van de concurrent kan bekogelen. Het document is de eerstvolgende sprint aangepast.

De uitleg waarom deze style van design is gekozen kan de lezer van het Game Design Document vinden in de style guide, waar het volgende staat omgeschreven :
'Bij het ontwerp van de elementen wordt er uitgegaan van de no-nonsense gedachtegang van De Nederlanden van Nu. Er wordt gezocht naar een open en ruimtelijk ontwerp met minimale elementen.'

Hier was de opdrachtgever wel mee eens, en dit is dan ook meegenomen naar de implementatie van het (nieuwe) design.

2.2.2 De Game

Voor een volledige beschrijving van de game kan de lezer terecht in bijlage D : het Game Design Document. Dit hoofdstuk beschrijft het spel in het algemeen om context te scheppen voor de komende hoofdstukken.

Idee

Het idee van het spel komt voort uit het feit dat de Nederlanden van Nu bereddingskosten vergoedt in hun inboedelverzekering; Dit houdt in dat als er een brand ontstaat in een verzekerd huis, dat de schade die ontstaat door het blussen hiervan wordt vergoed.

Doel

Het doel van het spel is om meer naamsbekendheid te genereren door een (simpel) spel op de markt te brengen, iets wat opmerkelijk is voor een verzekeraar; Of het spel daadwerkelijk naambekendheid creëert valt buiten de scope van de opdracht voor de student.

Gameplay

Het spel zet de speler in een situatie waarin een gebouw in brand staat; Gedurende een vaste periode ontstaan er steeds meer brand op verschillende locaties in het gebouw. De speler moet via een waterkanon de brand zo snel mogelijk blussen om schade aan het gebouw te voorkomen.

Om enige realisme na te bootsen, en om het spel enige herspeelbaarheid te geven, gebruikt het spel physics (natuurkundige wetten) om de objecten in het spel te verplaatsen; Omdat dit te omvangrijk is om in een korte periode te programmeren wordt hiervoor het Box2D framework gebruikt.

Ook wordt er een framework (Cocos2D) gebruikt voor het renderen van de sprites (het tekenen van de textures op het scherm). Meer informatie over de keuze van frameworks is te vinden in hoofdstuk 2.4.3.

Omdat het uiteindelijke doel van het spel is om naamsbekendheid voor De Nederlanden van Nu te genereren, moet de speler een eindresultaat via de app kunnen posten op facebook.



2.3 Technisch Ontwerp

2.3.1 Unified Modeling Language

Het design aan de software-matige kant is gedaan door middel van UML, de Unified Modeling Language. De software is vormgegeven met usecases, inclusief beschrijvingen, een klassendiagram en een sequentie diagram; UML is voor het project gekozen omdat het simpelweg de enige geschikte taal is, die de student voldoende kent, die de benodigde diagrammen omvat.

Tijdens het modelleren is vooral aandacht besteed aan het flexibel houden van het systeem. In hoofdstuk drie is te lezen over de verschillende frameworks die het systeem gebruikt, het doel van de student was om de applicatie onafhankelijk te maken van deze frameworks. Hierover is meer te lezen in paragraaf 2.2.4.

Alle informatie met betrekking tot de diagrammen die zijn toegepast komt uit het boek 'Praktisch UML' van J. Warmer & Aneke Kleppe. [5]

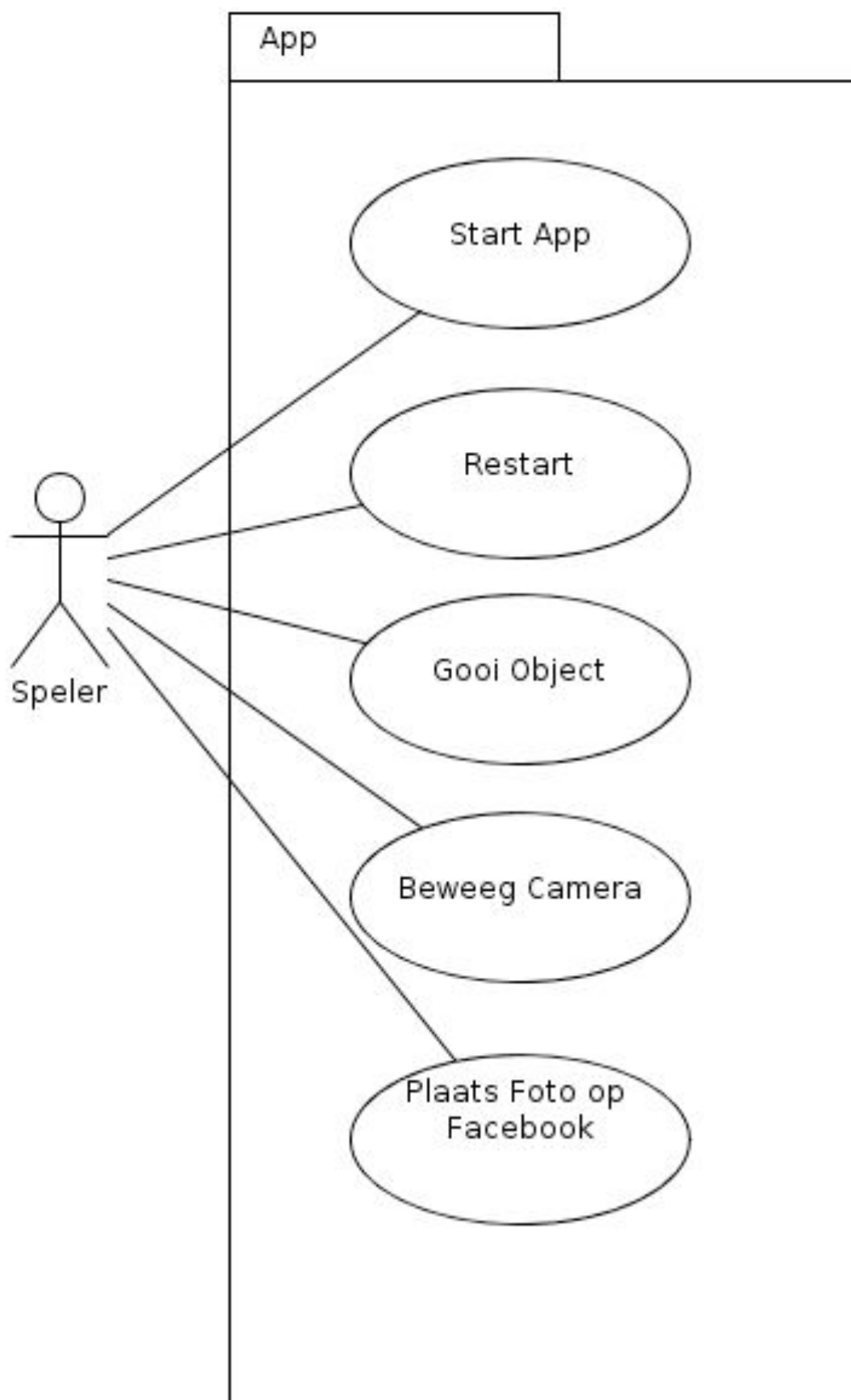
2.3.2 Usecases

De requirements vanuit de gebruiker gezien zijn in eerste instantie opgenomen in de product backlog in de vorm van user stories. Een aantal iets meer complexe user stories zijn vormgegeven in een use case diagram met beschrijvingen om bepaalde stories duidelijker te definiëren. (Zie Figuur 4 op pagina 21)

Het diagram toont de verschillende acties die een gebruiker in het spel kan uitvoeren en welke externe actoren hiervoor nodig zijn. Aangezien het een eenvoudig spel omvat die met een enkele speler wordt doorlopen, worden de meeste acties enkel door de gebruiker uitgevoerd.

De usecases zijn in een sprint meeting gepresenteerd, uitgelegd en besproken met de opdrachtgever en die is hiermee akkoord gegaan.





Figuur 4. Usecase Diagram

Elke usecase is in een usecase beschrijving verder uitgewerkt in de volgende vorm :

Naam	Gooi Object
Samenvatting	De speler kan in het spel een kanon aanzetten die objecten afvuurt
Actoren	Speler
Aannamen	Het spel is gestart en de gebruiker is in het spel-scherm
Beschrijving	1) De speler tikt op het scherm bij het wapen 2) Het wapen begint met objecten afvuren
Uitzondering	[De gebruiker sleept zijn vinger over het scherm tijdens stap 1] De app gaat over op de usecase 'Beweeg Camera' [De gebruiker sleept zijn vinger over het kanon tijdens stap 1] De app verandert de richting van het kanon [Het wapen is al aan het vuren tijdens stap 2] Het wapen stopt met vuren ipv stap 2 [De afgevuurde objecten botsen na stap 2] Objecten worden verwijderd [Het afgevuurde object raakt een vlam na stap 2] Het object wordt verwijderd en de vlam krijgt schade [Het afgevuurde object raakt niks na stap 2] Het object wordt verwijderd
Resultaat	Het wapen vuurt objecten af

2.3.3 Klassendiagram

De software architectuur is vormgegeven in een klassendiagram. Het klassendiagram is ontstaan in de tweede sprint, waar er meer aandacht was ingepland om het design in orde te krijgen, zodat in komende sprints sneller en effectiever functionaliteit ingebouwd kon worden. Het klassendiagram is te zien in figuur 5.

Flexibiliteit

De focus van het klassendiagram lag bij het flexibel houden van het systeem. De applicatie haalt de functionaliteit van twee belangrijke elementen uit externe frameworks: Het renderen van de grafische elementen, en het simuleren van natuurkundige wetten.

Om de applicatie zo min mogelijk afhankelijk te maken van deze frameworks (zodat een framework makkelijk vervangbaar is) zijn voor deze twee frameworks elk een aparte klasse bedacht, die de interne werking van de frameworks encapsuleren voor de rest van de applicatie. In een situatie dat een framework moet worden vervangen, hoeft enkel deze klasse aangepast worden, en de rest van de applicatie werkt nog, om ervoor te zorgen dat de rest van de applicatie geen problemen krijgt, zijn er interfaces geïntroduceerd.

Scope

Niet alle variabelen worden gemeld in het initiële klassendiagram, mede door de onzekere aard van de logica van het spel zelf; Hierbij is gedacht aan de gedachtegang van de AGILE methodiek : in plaats van alles tevergeefs van te voren in te plannen, wordt het systeem flexibel opgebouwd. Het volledige klassendiagram is wel tijdens het programmeren gedocumenteerd, en kan geworden worden in bijlage D, het game design Document.

Planning

Het klassendiagram is in de tweede sprint opgesteld. Er was dus al code waar twee weken lang aan is gewerkt tijdens de eerste sprint, die gedeeltelijk moest worden aangepast.

Er is voor gekozen om de eerste sprint te besteden aan het schrijven van een werkend prototype die vervolgens aan het management hogerop zal kunnen worden gepresenteerd ter acceptatie. Hier is weinig gedacht aan software-design om zo snel mogelijk een functioneel prototype te kunnen tonen. Tijdens de tweede sprint is de code omgebouwd naar het nieuwe design zodat komende functionaliteit goed wordt ingebouwd.

Voorbeeld :

Het prototype van de eerste sprint heeft ervoor gezorgd dat het management van Generali besloot om het huidige game-design niet te accepteren. Door het prototype in de eerste week te maken is uiteindelijk veel tijd bespaard. Tijdens de derde sprint kon al een nieuwe game-design worden opgesteld.



Klassen

De termen in het klassendiagram zijn vrij specifiek voor de game industrie en zullen misschien niet iedereen bekend voor komen. Dit paragraaf ligt kort een aantal klassen van het diagram in figuur 5 toe om context te scheppen.

Scene

De Scene klasse zorgt voor de logica die bij een level hoort. Hierbij kan gedacht worden aan het scheppen van een wereld en het aansturen van de objecten die bij deze wereld horen.

De Scene klasse is bedoeld als superklasse voor specifieke levels, er is namelijk algemene logica die elk level moet hebben, en een heleboel logica die specifiek voor een bepaald level geldt.

Object

De Object klasse is de representatie van een game-object in de wereld. Het houdt data vast zoals de locatie, rotatie en omvang van een object. Een object hoeft geen grafische representatie te hebben op het scherm, hiervoor moet het een Sprite - klasse vast houden. Het Object is enkel verantwoordelijk voor de functionele data van een object.

Sprite

De Sprite klasse is een texture (plaatje) die bij het object past. Zodra het geregistreerd wordt bij een Layer Renderer kan het worden getekend op het scherm. Het houdt data vast zoals de bestandsnaam van het plaatje en de doorzichtbaarheid ervan.

PhysicsSimulator

De PhysicsSimulator is verantwoordelijk voor het natuurkundig simuleren van de objecten in de wereld. Deze berekeningen doet de PhysicsSimulator niet zelf maar worden uitgevoerd in een extern framework. De PhysicsSimulator is dus eigenlijk een vertaling tussen de data in de app en de data in het framework.

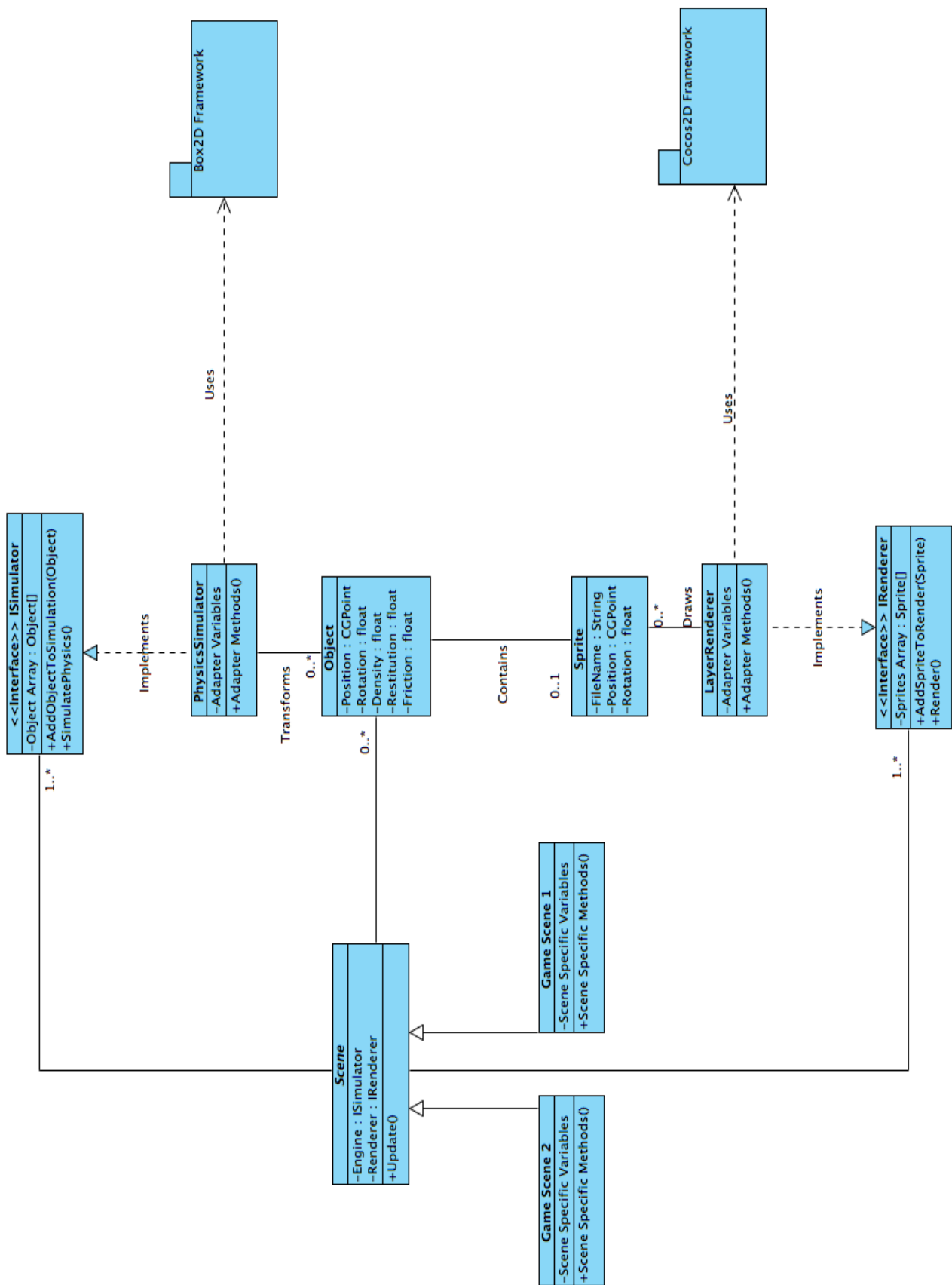
Het box2D framework is in dit geval het framework wat de natuurkundige wetten simuleert. Omdat het goed mogelijk is dat de PhysicsSimulator in de toekomst een ander framework vertaalt, implementeert de klasse een interface.

Layer Renderer

De Layer Renderer is verantwoordelijk voor het tekenen van grafische bestanden op het scherm. Deze functionaliteit haalt de klasse uit een extern framework (in dit geval Cocos2D). De Layer Renderer is dus eigenlijk een vertaling tussen de data in de app en de data in het framework.

Omdat het goed mogelijk is dat de Layer Renderer in de toekomst een ander framework vertaalt, implementeert de klasse een interface.





Figuur 5. Klassendiagram

2.3.4 Sequentie Diagram

De interactie tussen de verschillende klassen in het klassendiagram is gemodelleerd door middel van een sequentie diagram. Het sequentie diagram richt zich op het inrichten van een level.

Ook de 'main loop' is weergegeven in het diagram, deze loop wordt elke frame aangeroepen, en stuurt alle logica van het renderen en de natuurkundige calculaties.

De student heeft ervoor gekozen om enkel de 'main sequence' te modelleren en niet het totale overzicht van alle berichten tussen alle klassen. Het sequentie diagram toont de algemene logica die (praktisch) elk level zal doorlopen. Hiernaast heeft elk level zijn eigen logica; Deze logica is echter simpel genoeg om zonder sequentie diagram te implementeren.

Sequentie

In figuur 6 op de volgende pagina is het diagram te zien.

De applicatie creëert tijdens de start een instantie van een Scene. Deze Scene stuurt vervolgens de andere klassen.

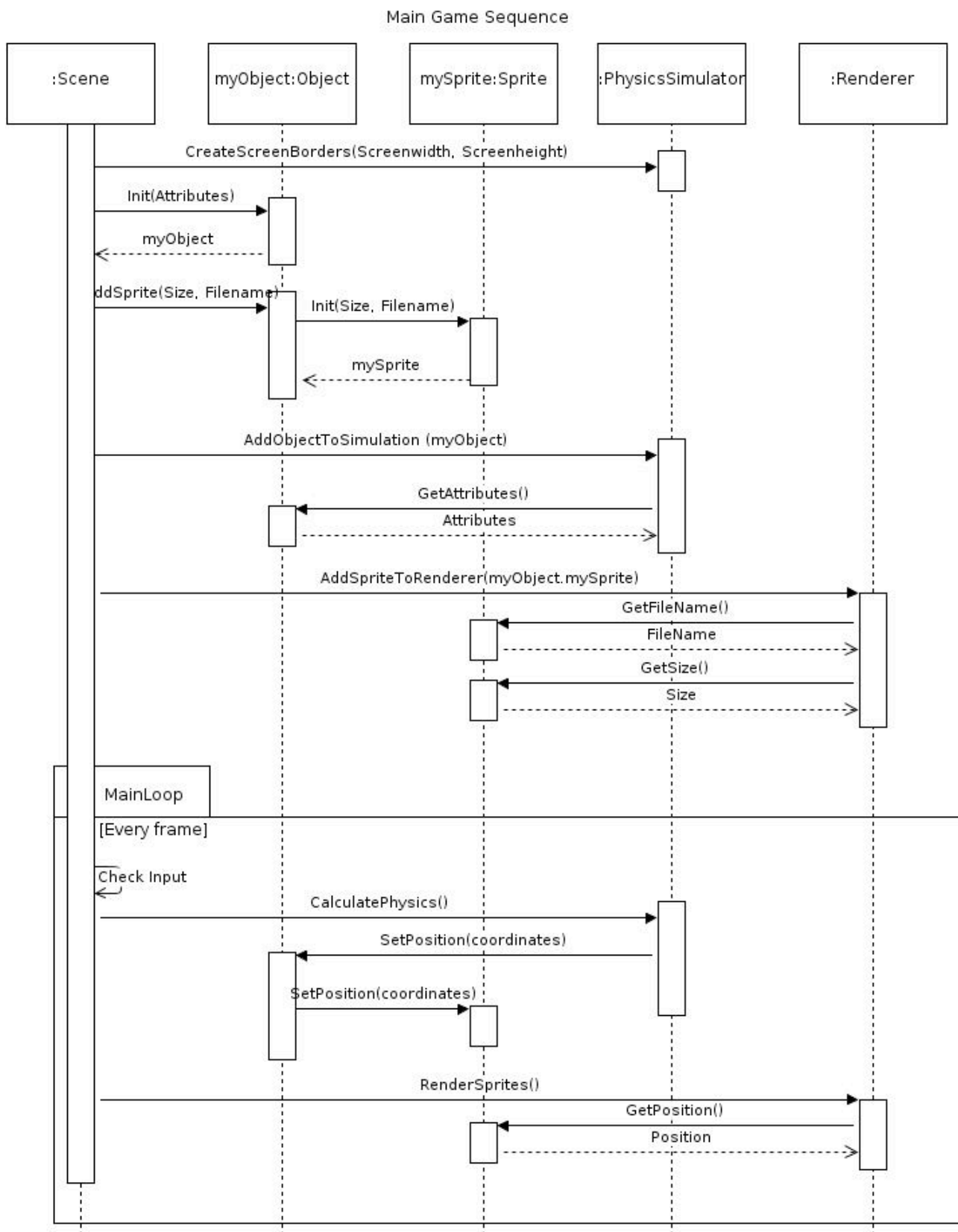
De CreateScreenBorders methode vraagt aan de PhysicsSimulator om collision detection te creëren voor de uiterste waarden van het level, om te voorkomen dat objecten van het level afvallen.

Vervolgens wordt een aantal objecten aangemaakt en deze krijgen (optioneel) een sprite zodat een visuele representatie van het object op het scherm getekend kan worden. Hiervoor wordt een naam van een bestand meegegeven.

Elk individueel object dat gesimuleerd moet worden in de PhysicsSimulator moet zichzelf registreren bij de PhysicsSimulator; Hetzelfde geldt voor de sprites die zichzelf willen laten tekenen door de renderer. Zoals in het klassendiagram te zien is, zorgen de interfaces ervoor dat deze methodes altijd zijn aan te roepen, onafhankelijk van de implementatie ervan.

De mainloop wordt elke frame aangeroepen. Hier vraagt de scene aan de PhysicsSimulator of de objecten moeten worden verplaatst en vraagt de scene aan de Renderer of de Sprites gerendered kunnen worden.





Figur 6. Sequence Diagram

2.4 Constructie

2.4.1 Ontwikkelomgeving

Voor de constructie van de applicatie is XCode 4.4 gebruikt als ontwikkelomgeving. Xcode is de IDE van Apple en wordt bij een developer licentie meegeleverd, inclusief tutorials en extra tools. Dit hoofdstuk geeft de lezer een indruk van de omgeving en illustreert dat de student in een professioneel programma heeft gewerkt. Figuur 7 op de volgende pagina geeft een grafische indruk van de omgeving.

Xcode is gericht op het schrijven van mac en/of IOS applicaties en is standaard geïntegreerd met de Cocoa en Cocoa Touch Frameworks, die de basis leveren voor de functionaliteit van apps, zoals het gebruiken van het touchscreen op de Iphone.

Feedback

Xcode zorgt voor feedback tijdens het typen van code en probeert direct eventuele fouten te corrigeren. Hiernaast biedt de omgeving een IOS simulator voor het direct testen van IOS-apps. De unit-test omgeving is volledig geïntegreerd in de omgeving en de tests kunnen met een enkele druk op de knop worden uitgevoerd.

(Voor meer informatie over het testproces kan de lezer terecht in 2.5)

Fysiek Toestel

Via de developer licentie is een fysieke iPhone aan de ontwikkelomgeving te koppelen waardoor de app direct op een fysiek toestel kan worden gebuild en getest.

Tools

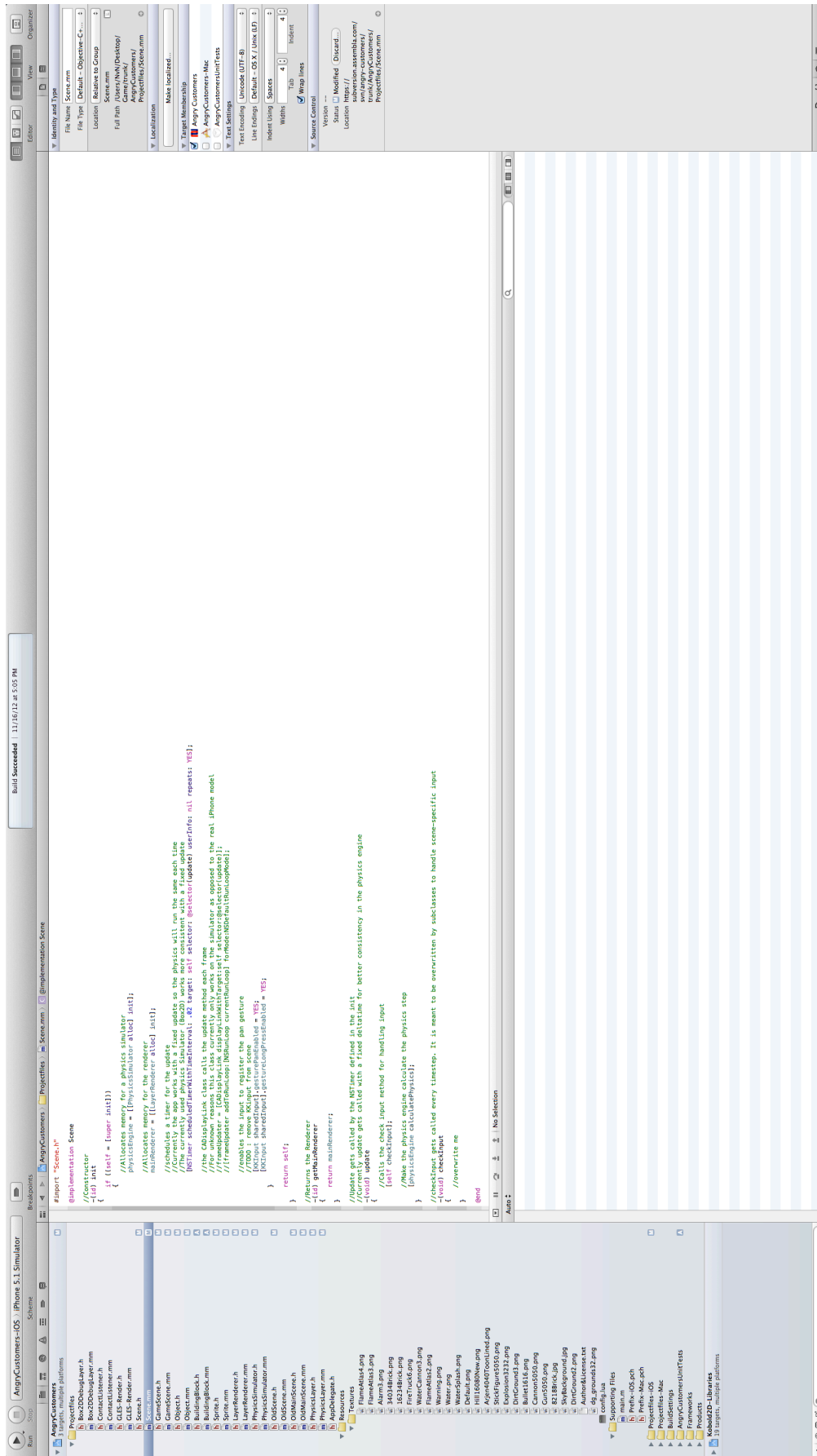
Zodat de applicatie met goede performance functioneert zijn er tools aanwezig die runtime de disk, memory en CPU usage kunnen meten, op zowel de simulator of via een kabel op een echte Iphone. Deze tools kunnen grafieken en tabels tonen waardoor het mogelijk is om bijvoorbeeld memory leaks vroeg te signaleren.

Voorbeeld :

Bij de meeste (kleine) veranderingen is snelle feedback handig: Gebeurt er wat ik wil wat er gebeurt? In deze situatie kan de IOS Simulator snel worden opgestart om de functionaliteit kort te testen.

De IOS Simulator kan echter niet de volledige ervaring van een fysiek toestel geven, features zoals het ingebouwde kompas kunnen niet werken op een iMac aangezien hier geen kompas in zit. Regelmatig is daarom de functionaliteit getest op het fysieke toestel.

In 1 geval was er een observer uit het Cocoa Framework gebruikt om een message te sturen zodra er een nieuw frame getekend is. Door een bug in het framework werkte deze klasse tijdelijk niet op fysieke toestellen, maar wel op de simulator. De klasse moest dus vervangen worden.

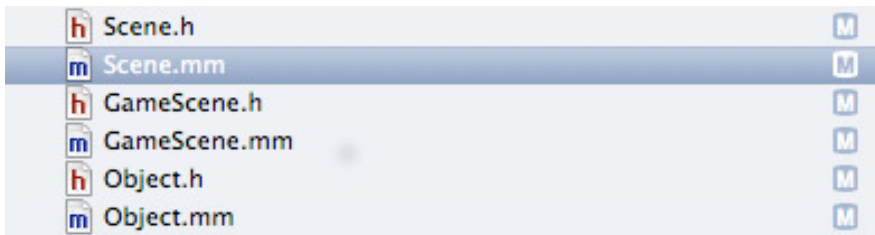


Figuur 7. XCode Ontwikkelomgeving

2.4.2 Subversion

Xcode, de ontwikkelomgeving, biedt de mogelijkheid om versiebeheer direct te koppelen aan de code. Hier is dan ook gebruik van gemaakt.

Voor het opslaan van de code is een repository aangevraagd bij www.assembla.com. Assembla is hiervoor gekozen omdat het na kort onderzoek bleek de meeste gratis ruimte te bieden in een closed-source omgeving. [6] Vervolgens is het adres bekend gemaakt aan Xcode. Tijdens het schrijven van code geeft Xcode aan welke bestanden zijn toegevoegd of gewijzigd ten opzichte van de repository. (Zie figuur 8)



Figuur 8. XCode SVN Integratie

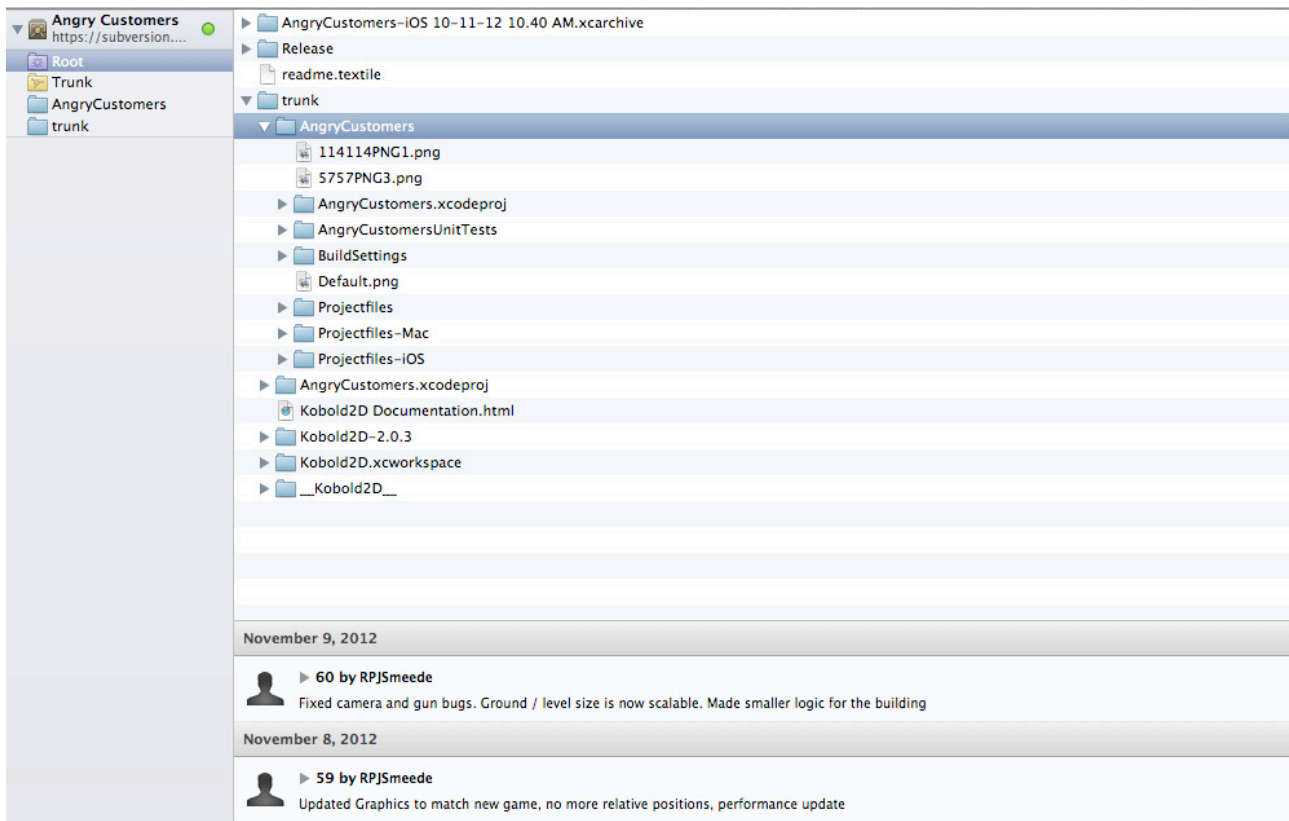
Bestanden die gewijzigd zijn krijgen een 'M' (voor modified) rechts van de naam, waardoor het duidelijk is of de svn-repository de veranderingen nog moet opslaan. Deze bestanden kunnen vervolgens met een druk op de knop worden commit naar de repository.

Repository View

Hiernaast maakt Xcode automatisch connectie met Assembla zodra het project wordt opgestart. In de repository view van Xcode is zijn de bestanden in de repository te bekijken en zijn de veranderingen per revisie te overzien. (Zie figuur 9)

De trunk is per conventie de map waar de huidige development code wordt geplaatst. Er is geen gebruik gemaakt van branches en tags mappen, aangezien deze niet van toepassing waren; Nieuwe releases werden opgeleverd als apart bestand, maar werden niet extra in svn gezet om ruimte te besparen.

Voor elke nieuwe commit is logisch commentaar toegevoegd zodat er per versie gemakkelijk te zien is wat de veranderingen zijn. Informatie over wat er per regel is veranderd is in de repository view ook te zien. Als de aparte bestanden in een release worden geïnspecteerd worden de veranderingen ten opzichte van de release daarvoor onderstreept door XCode. Het commentaar was daarom ook algemeen, kort en bondig.



Figuur 9. XCode Repository View

2.4.3 Frameworks

De applicatie haalt een gedeelte van de functionaliteit uit verschillende frameworks, deels noodzakelijk, en deels optioneel gekozen door de student. Dit hoofdstuk loopt de verschillende frameworks langs en schept inzicht waarvoor de frameworks nodig zijn, en waarom ze zijn gebruikt.

Cocoa / Cocoa Touch

Het Cocoa framework is ontwikkeld door apple en is de API voor het mac OS X. Het onderliggende Cocoa Touch is de zuster van Cocoa en voegt functionaliteit toe waardoor het kan dienen als de API voor iOS applicaties op de iPhone.

Het framework is zelf weer gebouwd op de onderliggende Foundation Kit, Application Kit en Core Data frameworks.

Het Cocoa Touch framework is de basis voor de functionaliteit van zaken zoals het herkennen van vingerbewegingen op het touchscreen en is noodzakelijk voor een iOS applicatie. Omdat het framework noodzakelijk is, heeft de applicatie een sterke koppeling met het framework.

Alternatieven : Geen, het framework is nodig voor een iOS app.

Physics

Box2D

Het Box2D framework is verantwoordelijk voor de functionaliteit van de natuurkundige wetten in het spel. Het zorgt ervoor dat objecten in het spel een massa hebben en tegen elkaar aan kunnen botsen en natuurkundige krachten kunnen ontvangen.

Het framework is onder andere gebruikt voor populaire spellen zoals Angry Birds en Tiny Wings en is in ontwikkeling sinds 2007; Het heeft zichzelf bewezen als een framework van kwaliteit.

Het framework is vooral gekozen door de goede reputatie en performance, onder andere door bovengenoemde titels;

Er is op internet een kort onderzoek gedaan naar physics frameworks voor iOS.

Alternatieven :

Voor het regelen van de physics zijn er meerdere frameworks onderzocht via internet. De eventuele alternatieven voor box2d die ik heb overwogen zijn de twee onderstaande. Een korte motivatie op de volgende pagina licht toe waarom deze frameworks niet zijn gebruikt.



Chipmunk

Chipmunk is een 2D physics framework gebouwd vanuit de basis van Box2D zelf. De twee frameworks lijken hierdoor erg veel op elkaar en meningen op het internet zijn erg verdeeld over welk framework beter is. [7, 8]

Box2D heeft echter meer mogelijkheden door functionaliteit zoals sleeping en betere performance bij objecten van verschillende grootte. Ook is er betere documentatie, en een taal waar ik persoonlijk meer ervaring mee heb (C++ ten opzichte van C) [7, 9]

PhysX

Het PhysX framework van Nvidia is van origine bedoeld voor 3D applicaties, maar wordt door de Unity SDK en UDK SDK ook gebruikt voor 2D iOS applicaties; Dit zijn twee SDK's waar ik mee heb gewerkt en goede ervaringen mee heb; Vandaar het onderzoek naar dit framework.

Ik had door de 3D basis echter mijn twijfels over de performance. Een 3D framework heeft al snel functionaliteit die overbodig is voor 2D. Op basis van onderzoek, gevonden via google, bleek dat de performance dan ook substantieel lager ligt dan box2D. [10]

Rendering

Cocos2D

Cocos2D is verantwoordelijk voor het renderen van de 2D plaatjes. Het communiceert elke frame met de OpenGL bibliotheek om te zorgen dat alle sprites getekend kunnen worden.

Het framework is deels gekozen door de goede reputatie van het framework, net als bij het box2D framework wijzen op het internet veel meningen op de goede kwaliteit van het framework. Naast de goede reputatie heeft Cocos2D goede integratie met Box2D.

Voor het cocos2D framework zijn geen alternatieven gezocht, het is standaard geleverd met box2D, dus na het uitzoeken van een physics engine was dit een snelle keuze.



2.4.4 Documentatie van de code

Er is veel aandacht besteed om de code in goede staat achter te laten, zodat eventuele toekomstige ontwikkelaars de code gemakkelijk kunnen oppakken en wijzigen. Een belangrijk onderdeel hiervan is het documenteren van de code door middel van commentaar.

De doelstelling voor het leveren van commentaar was om voor elke regel code een regel commentaar toe te voegen, enige vrijheid bij het interpreteren van deze doelstelling is logischerwijs genomen: bij regels code die zeer vanzelfsprekend zijn is een regel overgeslagen.

Vooraf aan een klasse is een samenvatting gecommenterieerd van de functionaliteit en eventuele opmerkelijke eigenschappen. Hetzelfde is gedaan vooraf aan een methode, met in een aantal gevallen verdere uitleg over de parameters.

Een voorbeeld van deze stijl van documentatie is te zien in figuur 10.

```
//The renderSprite method adds the supplied sprite to future rendering draws.
//The layer renderer creates a reflecting Cocos2D specific sprite to reflect the game-sprite
//Sprites are drawn through draw batches according to file in order to maximize performance on drawing
-(void) renderSprite:(Sprite*) sprite
{
    //Get the image part that needs to be drawn
    CGRect tileRect = sprite.getRect;

    //Create a new Cocos2D sprite reflecting the original
    CCSprite* drawSprite = [CCSprite spriteWithFile:sprite.getFile rect:tileRect];

    //reflect the position
    drawSprite.position = sprite.getPosition;

    //Create a reference to the original sprite, so the two can synchronize
    drawSprite.userObject = sprite;

    //Add the Cocos2D sprite to the sprite array
    [spriteArray addObject:drawSprite];

    //add the sprite to a batch according to its file
    NSUInteger i = 0;

    //The filename array is searched to check if the file is already is being batch drawn
    for(i = 0; i < [fileNameArray count]; i++)
    {
        NSString* name = [fileNameArray objectAtIndex:i];

        //Is the file being drawn in a batch?
        if(name == sprite.getFile)
        {
            //a batch file is already added for the file
            CCNode* batch = [self getChildByTag:i];

            //Add the current sprite to it's correct batch
            [batch addChild:drawSprite];

            //Stop the loop to prevent another batch being created further down the method
            return;
        }
    }

    //The function has not returned yet, which means the filename is not known to the renderer
    //A new batch renderer should be added for the filename
    CCSpriteBatchNode* batch = [CCSpriteBatchNode batchNodeWithFile:sprite.getFile];

    //Add the new filename to the filename-array, so future sprites using the same file will get drawn in the same batch
    [fileNameArray addObject:sprite.getFile];

    //Add the batch to the layer renderer so it gets a cocos2D draw call when the layer does
    [self addChild:batch z:sprite.getZ tag:i];

    //Add the sprite to the new batch
    [batch addChild:drawSprite];
}
```

Figuur 10. Commentaar in de code



2.5 Testen

2.5.1 Master Test Plan

Het testen is gebeurd via de TMap aanpak vanwege eerdere ervaring van de student met de aanpak; Ook kan de aanpak gemakkelijk gecombineerd worden met SCRUM / AGILE [11]; De opdrachtgever is dan ook zoveel mogelijk betrokken bij het testproces, communicatie is tenslotte belangrijker dan een plan. Zo is elke sprint meeting het testen besproken met name over welke functionaliteit getest moet worden en in welke mate.

Om sturing te geven aan het gehele test proces is een master test plan geschreven. Dit document kan de lezer vinden als bijlage E. Het document omschrijft de teststrategie van het project door middel van het beschrijven van de soorten testen die uitgevoerd zullen worden en in welke mate.

Het master test plan en de bijbehorende strategie zijn gebaseerd op risicodenken : omdat de beschikbare tijd om te testen beperkt is, moeten er keuzes worden gemaakt. Niet alles kan tenslotte even zwaar worden getest.

Risico's

Om deze risico's in kaart te brengen is een risicoanalyse uitgevoerd; Voor elke user story uit de Product Backlog (zie 2.1.4) is de kans op falen en de schade bij falen bepaald;

Hierbij is de kans op falen gebaseerd op zowel de foutkans als de frequentie van gebruik, waarbij de foutkans is ingeschat door :

- De omvang van de code (hoe meer code, hoe meer kans op een fout)
- De complexiteit van de code (hoe moeilijker de code, hoe meer kans op een fout).

De frequentie van gebruik spreekt voor zich : Hoe meer de functionaliteit gebruikt wordt, hoe groter de kans dat de fouten voor gaan komen.

Er is een tabel samengesteld die deze twee factoren gebruikt om een risicoclassificatie te bepalen. Hierbij is figuur 11 als tabel gebruikt; Deze tabel is a-symmetrisch, er is besloten dat een risico met een hoge schade bij falen belangrijker is dan een risico met een lage impact.

	Schade bij falen		Kans op falen	
		Hoog	Midden	Laag
	Hoog	A	B	B
	Midden	B	B	C
	Laag	C	C	C

Figuur 11. Risico tabel



Risico classificatie

Zoals op de vorige pagina is beschreven, is van elk stuk functionaliteit de risicoclassificatie bepaald door middel van de tabel (figuur 11). Deze risico's zijn weergegeven in figuur 12.

Kenmerk	User Story	Schade	Game Scherm	Eind Scherm
Faalkans :			Hoog	Middelmatig
Functionaliteit	1. Als een gebruiker wil ik een grafische weergave van de spel elementen zien	Hoog	A	-
Functionaliteit	2. Als een gebruiker wil ik dat de spel elementen via natuurkundige wetten werken.	Hoog	A	-
Functionaliteit	3. Als een gebruiker wil ik bommen kunnen wegschieten	Hoog	A	-
Functionaliteit	4. Als een gebruiker wil ik de camera kunnen bewegen.	Laag	C	-
Functionaliteit	5. Als een gebruiker wil ik het een weergave van het verloop van het spel op het scherm zien.	Laag	C	-
Functionaliteit	6. Als een gebruiker wil ik een eindweergave kunnen posten op facebook	Middelmatig	-	B
Functionaliteit	7. Als een gebruiker wil ik geluiden kunnen horen	Laag	C	C
Usability		Middelmatig	B	B
Performance		Middelmatig	B	B

A = Hoog risico

B = Middelmatig risico

C = Laag risico

Figuur 12. Risico classificatie

Test Strategie

Door de risico-analyse zijn de risico's bepaald van elke user story. Deze risico's zijn de basis voor het bepalen van de zwaarte van de tests. Om een overzichtelijk beeld te geven van de (relatieve) zwaarte van de tests is wederom een tabel samengesteld (figuur 13).

Kenmerk	Risicoklasse	Unit Testing	Functionele Acceptatie Test
Functionaliteit			
User Story 1	A	O O O	O
User Story 2	A	O O O	O
User Story 3	A	O O O	O
User Story 4	C	O	O
User Story 5	C	O	O
User Story 6	B	O O	O
User Story 7 - Game Scene	C	O	O
User Story 7 - Facebook	C	O	O
Usability	B	-	O
Performance	B	-	O

A = Hoge Risicoklasse
B = Middelmatige Risicoklasse
C = Lage Risicoklasse

O = Beperkte Test
O O = Gemiddelde Test
O O O = Zware Test

- = Geen aandacht

Figuur 13. Test Strategie Tabel

Voor elke user story zijn unit tests uitgevoerd waarbij de zwaarte van de tests samenhangen met de risicoclassificatie van de user story; Bij een hoog risico, wordt een zware test uitgevoerd en bij een laag risico wordt een beperkte test uitgevoerd. Voor alle user stories is besloten om een beperkte User Acceptance Test uit te voeren, zodat de story geaccepteerd kan worden door de gebruiker (in dit geval de Product Owner).



Test Aanpak

Het master test plan beschrijft tenslotte de concrete aanpak van de verschillende soorten tests. In dit gedeelte van het document wordt er dieper ingegaan op verschillende soorten tests en hun doel.

De student heeft voor dit testtraject de volgende testsoorten onderkend :

- *Unit-Testing : Functionaliteit*

Volgens het master test plan moet voor elke user story unit tests geschreven worden die de functionaliteit met betrekking tot de user story testen. Afhankelijk van de zwaarte van de test is er onderscheidt gemaakt tussen Function Coverage, Statement Coverage en Decision Coverage.

Deze testsoort is vooral gekozen door het automatiseren van de tests. Het is mogelijk dat in de toekomst een nieuw persoon/stagiair het werk verder oppakt; Als dit gebeurt moet eventuele regressie in de bestaande code automatisch moeten worden gedetecteerd;

- *User Acceptance Test : Functionaliteit, Usability, Performance*

Elke user story is na afloop van elke sprint geaccepteerd door de gebruiker. Hiervoor was geen complexiteit nodig, de product owner bekeek de functionaliteit en beoordeelde hiervan de Usability en performance; Hiernaast krijgt elke collega van het team de kans om de applicatie door te lopen en hierdoor 'fouten' in usability op te sporen.

Deze testsoort is min of meer noodzakelijk in een project waar goede communicatie belangrijk is. Door snel en vaak feedback op de black-box functionaliteit te krijgen, kunnen problemen die ontstaan zijn door foutieve communicatie met de opdrachtgever snel hersteld worden.

Er is voor gekozen om tijdens het testproces enkel met deze twee testsoorten te werken, met name in verband met de tijdsdruk in het project.

Detail Test Plan

Doordat de projectgroep, en in zekere mate het project zelf, vrij kleinschalig is was ook het testproces relatief klein. Zowel het master test als de eventuele detail test plannen waren vrij kort en overbodig om op te splitsen.

Er is daardoor geen detail test plan opgesteld voor de testsoorten, de informatie hiervan is verwerkt in het master test plan.



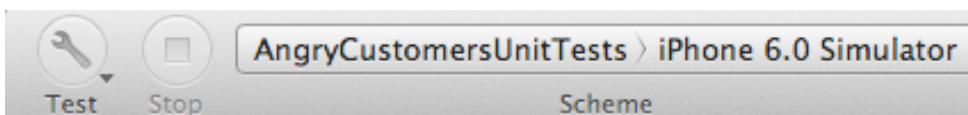
2.5.3 Unit Testing

De unit tests zijn functioneel ingestoken, dat wil zeggen, de interne werking van functies worden getest door middel van de input en output van een methode te controleren. Dit zorgt voor flexibele unit tests die niet hoeven te worden herschreven als een methode wordt herschreven; Door de tijdsdruk in het project was dit een belangrijke eis voor de unit tests.

Er is uitgegaan van verschillende niveau's van coverage. Voor zowel lichte als zware tests is uitgegaan van minimaal function en statement coverage, waar alle regels van de klasse een keer worden doorlopen. Bij zwaardere tests was ook condition coverage een doel, waar, in het geval van meerdere condities in een 'if', alle mogelijke combinaties van condities in een beslissing zijn getest. Ook de abnormale waardes zoals 0, negatieve getallen etc zijn meegenomen in het testen van de functionaliteit. De mate van coverage per functionaliteit is gedocumenteerd in het master test plan (bijlage E)

XCode

Voor het uitvoeren en integreren van de unit tests is de test-functionaliteit van de XCode development omgeving gebruikt. Hiermee is het mogelijk om met 1 druk om te knop alle testcases uit te voeren. (Figuur 14).



Figuur 14. Test Knop in XCode

Om iets specifieker te zijn over de functionaliteit van deze knop: Van alle klassen die overerven van 'SenTestCase' worden alle methodes aangeroepen waarvan de naam begint met 'test'. Hierdoor was het gemakkelijk om na elke verandering de verschillende Unit Tests te doorlopen en eventuele regressie vroegtijdig op te sporen.

Een voorbeeld van een unit test methode kan de lezen vinden in figuur 15 op de volgende pagina. Deze methode komt uit de unit test klasse van de Physics Simulator. Er wordt getest of een 'dood' object correct wordt opgeruimd van memory.

De Physics Simulator houdt referenties bij van objecten, zodra een object als 'dood' wordt gemarkeerd hoort de Physics Simulator deze referentie weg te halen, zodat het geheugen kan worden opgeruimd. (Memory wordt in Objective C vrijgegeven zodra er geen referenties meer zijn naar een memory adres)

Tijdens een nieuwe versie van de applicatie was ik functionaliteit aan het bouwen voor het blussen van vuur en maakte ik een kleine verandering in de Physics Simulator. Hierdoor gingen de bellen rinkelen bij de unit test dat memory niet goed werkt opgeruimd kreeg ik last van performance. Doordat de Unit Test deze fout direct opving heb ik veel tijd bespaard met het opsporen van de issue.


```

//The test checks if a dead object is cleared from memory after a calculate physics step
-(void) testRemoveDeadPhysics
{
    [self setUp];

    //Create an object
    CGPoint pos = CGPointMake(50, 50);
    Object* obj = [[Object alloc] initWith:pos:unmovable];
    [obj addCollision:20:30:0.1f:1.0f:0.0f:0x0002];

    //Add a reference to the object in the engine
    [engine addObjectToSimulation:obj];

    //Set the object as dead
    [obj die];

    //Keep a weak reference so we can test if the memory will be cleared
    Object* __weak weakObjPointer = obj;
    obj = nil;

    //The object should still be in memory, the physics engine should be holding a reference
    STAssertNotNil(weakObjPointer, @"The object will be removed from memory before a physics step! The engine should still hold a reference!");

    //The method should remove the reference in the simulator during a calculate physics step
    [engine calculatePhysics];

    //The object should be released from memory
    STAssertNil(weakObjPointer, @"The object has not been removed from memory!");

    [self tearDown];
}

```

Figuur 15. Unit Test : Removing dead reference



2.5.4 User Acceptance Tests

Door het uitvoeren van de Unit Tests is aangetoond dat de functionaliteit van de applicatie van voldoende kwaliteit is. Echter is hiermee nog niet bekend in hoeverre de applicatie voldoet aan de eisen van de opdrachtgever / Product Owner. Om dit te achterhalen is na afloop van een sprint, tijdens de sprint meeting, een 'User Acceptance Test' uitgevoerd.

Hiermee wordt de acceptatie van de gebruiker bedoeld; Zowel de opdrachtgever zelf als de overige collega's van de student bekijken op dit moment de applicatie en de nieuwe functionaliteit. Deze test had ik deze vorm de volgende doelen :

- Het accepteren van de nieuwe functionaliteit; Is de functionaliteit opgepakt zoals bedoeld en voldoet deze aan de eisen die de opdrachtgever verwacht?
- Door het simuleren van een gebruiker die op een natuurlijke wijze de applicatie gebruikt (black-box) kan usability, performance en applicatie-flow worden geobserveerd. Gebruikt de gebruiker de applicatie zoals de programmeur / student beoogt heeft om te gebruiken?

De resultaten van deze test zijn direct genotuleerd en verwerkt in de planning van de volgende sprint.

Voorbeeld :

Aan het eind van de derde sprint kon de gebruiker voor het eerst de spel wereld beïnvloeden door voorwerpen af te vuren. Toen de collega's deze functionaliteit probeerden bleek dat het controleren van de input niet natuurlijk werkte, maar ongeveer 50% van de collega's kon goed omgaan met het controleren van het waterkanon.

In de eerstvolgende sprint zijn de controls aangepast om sneller te reageren op de vingerbewegingen van de gebruiker en zijn de gebieden op het scherm waar de gebruiker moet klikken vergroot.

Zo'n dergelijke fout in usability had niet kunnen worden opgespoort door midden van Unit Tests.



3.1 Product Evaluatie

Dit hoofdstuk zal een evaluatie bieden hoe ik (de student) het product beoordeel.

Het product bestond in feite uit 3 verschillende delen :

- Het design van de game, in de vorm van een game design document
- De nieuwbouw, de code en applicatie
- Het test gedeelte, de unit test code en het master test plan

Het Design

Over het design van de game kan ik in mijn optiek redelijk tevreden zijn. Er is met het team van De Nederlanden van Nu goed nagedacht over het design in de vorm van meerdere brainstorms. Zowel ikzelf als mijn opdrachtgever was meer enthousiast over het eerste idee als de tweede, en dit sentiment is helaas niet gedeeld met het management van Generali; Dit is jammer.

Echter is de uitwerking ervan er niet minder op geworden, ik denk dat het design van de applicatie (zowel technisch als functioneel) in goede smaak valt bij de opdrachtgever. Wat betreft het technische aspect denk ik dat ik een flexibel en correct design heb gemaakt waar rekening met aanpassingen is gehouden. Ik heb veel tijd besteed aan het onderzoeken van verschillende mogelijkheden met betrekking tot design patterns en ik ben tevreden met het uiteindelijke resultaat.

De code

De daadwerkelijke code van de applicatie is trouw aan het technisch ontwerp en reflecteert dan ook de flexibiliteit van het design; Hierbij hoort ook het nodige documenteren, wat ik snel als valkuil ontdek bij mijzelf.

De code beoordeel ik als van goede kwaliteit. De code is correct, goed gedocumenteerd en hiernaast schaalbaar en gemakkelijk te onderhouden. Tijdens het schrijven is er gedacht aan deze waardes, maar ook aan de performance van de code, waar veel verbeteringen in zijn gebracht.

De applicatie

Wat betreft de uiteindelijke applicatie is het doel van de appstore niet bereikt en daar ben ik vanzelfsprekend minder tevreden over. Door de daarmee vervelende veranderingen in requirements is er simpelweg teveel tijdsdruk gekomen op het project waardoor de kwaliteit als gevolg achteruit is gegaan. Zowel mijn opdrachtgever als ikzelf hebben de applicatie in zijn huidige staat bestempelt als onwaardig voor publicatie, we zijn daarom ook van plan om na mijn afstudeerperiode de verloren tijd in te halen zodat er alsnog een waardig product kan worden overhandigd aan De Nederlanden van Nu. Beide partijen zien hier toekomst in. In dat opzicht kan ik tevreden zijn over de evolutie van het product, er is door middel van de SCRUM methodiek voldoende vertrouwen geschept in de potentie van het product : het is niet waardeloos, enkel incompleet.

Het testen

Het testen ben ik niet erg tevreden over. Door de enorme tijdsdruk die is ontstaan door het veranderen van requirements is het testen vanzelfsprekend opzij geschoven; Er is tenslotte niks te testen als er geen code is.

Omdat het oorspronkelijke doel van het project was om een applicatie in de app-store van Apple te publiceren is het schrijven van unit tests voor de user stories meerdere malen op een lagere prioriteit gezet als het schrijven van nieuwe functionaliteit om de applicatie speelbaar te maken. Het uiteindelijke aanbod van unit tests is dan ook veel lager dan ik zal willen; Enkel de hoogste risico's zijn afgedekt.

De unit tests die er wel zijn waren erg handig tijdens het ontwikkelen en hebben ook een aantal keer fouten eerder opgespoord dan ik persoonlijk zal kunnen. Hierdoor is uiteindelijk aardig wat tijd bespaard in het opsporen van vervelende bugs. Daarom denk ik ook dat de unit tests wel van goede kwaliteit zijn en hier ben ik blij om.

Conclusie

Ik ben ervan overtuigd dat het product goed is ontwikkeld en van voldoende kwaliteit is. Echter is het product, door de tijdsdruk, in zijn huidige staat incompleet. Er is wel voldoende vertrouwen geschept in het product en zowel de opdrachtgever als de student is van plan om het product verder te ontwikkelen zodat het bedrijf in de toekomst een waardig product kan leveren aan haar klanten.



3.2 Proces Evaluatie

Het proces is mij goed bevallen. Het project was erg geschikt om aan te pakken met behulp van SCRUM en het is ook zeker in goede smaak gevallen bij de opdrachtgever.

De keuze om niet vanaf dag 1 te beginnen met SCRUM, maar eerst te beginnen met een plan van aanpak en het uitleggen van de methodiek aan de opdrachtgever is een goede keuze geweest en zal ik in een volgend project wederom zo aanpakken. Het werken via SCRUM is echter wel snel in gang gezet zodat de samenwerking met de opdrachtgever vroeg in het project werd opgestart.

SCRUM

SCRUM is voor mij een erg fijne manier om een project te voeren; Het geeft elke keer weer een goed gevoel als een doel van een sprint is behaalt en als de burn down chart netjes op 0 terecht komt. Daarnaast is gedurende een sprint de progressie erg duidelijk en kan aan het begin van een dag duidelijk een doel worden gesteld waardoor ik die dag toch net iets harder aan de slag ga. De methodiek zorgt voor een goed tempo waardoor mijn valkuil van uitstelgedrag wordt onderdrukt.

Door de korte periodes (sprints) van SCRUM is het mogelijk geweest om snel in te springen op veranderingen. Wat in dit project erg veel naar voren kwam. Hierdoor is in veel gevallen de schade beperkt.

Samenwerking

De opdrachtgever heeft meerdere malen aangegeven SCRUM erg fijn te vinden, het zorgt voor goede inzicht in het functioneren en progressie van het project. Het team van De Nederlanden van Nu gebruikt inmiddels een aantal van de elementen van SCRUM in hun eigen proces en het team heeft na mijn presentatie overwogen om mogelijk de gehele methodiek te implementeren.

Ik heb hierdoor een goed gevoel wat betreft de communicatie naar de opdrachtgever. Beide deelden we het gevoel van een goede, hechte samenwerking en ik heb het idee dat we allebei een betrokken gevoel hebben bij het product; iets wat de kwaliteit van het project enkel ten goede kan zijn.

Stakeholders

Iets wat ik volgend project beter zou willen doen is het betrekken van alle stakeholders. Een groot probleem in het proces was de verandering halverwege het project, waar de requirements zodanig veranderde dat het project als het ware opnieuw moest beginnen; Als de stakeholder eerder was betrokken bij het project had dit risico meer beperkt kunnen blijven. Dit is helaas een inschattingsfout die zowel ikzelf als de opdrachtgever hebben gemaakt, de verwachting was dat het design wel te verkopen was aan het management.

De stakeholder betrekken bij de 2-wekelijkse sprint meeting zal ervoor zorgen dat zowel de opdrachtgever als de stakeholder sturing zouden kunnen geven aan het project. Ook het betrekken van de stakeholder bij de brainstorms was een goed idee geweest.



3.3 Persoonlijke Evaluatie

Ik ben persoonlijk erg veel gegroeid in mijn afstudeerperiode. Het zelfstandige werken is voor mij een goede manier om mijzelf uit te dagen en de kwaliteit van mijn werk naar een hoger punt te brengen.

Tijdsdruk

Voor mij is het erg jammer dat er in de tweede helft van mijn periode erg veel tijdsdruk op het project lag. Ik zie veel optimalisatie mogelijkheden in mijn product waar ik helaas geen tijd in kan steken. Hetzelfde geldt voor het testgedeelte, een gedeelte waar ik mijzelf minder inschat vergeleken met de andere competenties; Hier had ik me graag meer in willen verdiepen en in groeien, maar door tijdsgebrek kon ik me weinig verdiepen in de vaardigheid.

Groei

Waar ik in het project veel moeite in heb gestoken is het regelen van de planning, het proces en de samenwerking met de opdrachtgever. Het was mij van te voren bekend dat mijn grootste valkuil het plannen van een project is. SCRUM is daarom een goede methodiek voor mij, ik heb veel tijd gestoken in het indelen hiervan en denk dat ik hier veel in ben gegroeid; Ik ben tevreden dat ik ondanks mijn introverte persoonlijkheid goed heb gecommuniceerd met mijn opdrachtgever (en mijn collega's). Dit is een punt waar ik veel moeite voor moet doen en het doet mij goed dat mijn opdrachtgever meerdere malen heeft aangegeven erg tevreden te zijn met de samenwerking en communicatie. Ik ben van mening dat de problemen mbt. de planning in dit geval niet aan mij toe te wijzen zijn.

Aan de technische kant ben ik gegroeid in het programmeren voor iOS, het mobiele platform van Apple. Ik had voor dit project nog nooit gewerkt in Objective C en ik begrijp de taal een stuk beter. Ik ben in aanraking gekomen met concepten zoals ARC (Automatic Reference Counting), voor mij een unieke manier van garbage collection, waardoor ik meer begrijp over memory management; Wederom een zwak punt van mij.

Uiteindelijk wil ik mijn baan vinden in de richting van game development. Ik ben blij dat ik door middel van dit project in dit genre van development heb kunnen groeien. Zelf ben ik veel bezig met 3D, dus de 2D insteek van dit project heeft mij een andere kant laten zien van het gaming spectrum, waar hele andere variabelen om de hoek komen kijken.

In het handelen in maatschappelijke context ben ik nog tamelijk onhandig, mede door mijn introverte persoonlijkheid vind ik het moeilijk om oppervlakkig contact te maken met mijn collega's. Soms heb ik het idee dat ik mijzelf lichtelijk buitenspel zet in de sociale interactie. Dit is iets waar ik in wil groeien; Ik geloof namelijk erin dat deze sociale interactie naast technische vaardigheden belangrijk is voor het groeien in professionele zin.

Conclusie

In conclusie ben ik tevreden met mijn persoonlijke groei en mijn prestaties met betrekking tot mijn afstuderen; Ik ben van mening dat ik op veel punten ben gegroeid en professioneel heb gehandeld. Mijn opdrachtgever was erg tevreden met mijn functioneren en dat is tenslotte waar het allemaal voor gedaan wordt.



A. Bronnenlijst

- [1] Wikipedia : SCRUM (Development)
19 Okt 2012
[http://en.wikipedia.org/wiki/Scrum_\(development\)](http://en.wikipedia.org/wiki/Scrum_(development))
- [2] Manifesto for Agile Software Development
14 Jun 2010
<http://agilemanifesto.org>
- [3] Robert Bates
Game Design: The Art and Business of Creating Games
Prima Tech. 2001
- [4] Mark Baldwin
Game Design Document Outline
Baldwin Consulting, 2005
- [5] J. Warmer & Anneke Kleppe
Praktisch UML (3de editie)
Pearson Education, 2004
- [6] Subversion Hosting Comparison Review Chart
30 July 2012
<http://www.svnhostingcomparison.com>
- [7] Chipmunk Physics and Box2D comparison
29 March 2011
<http://forums.tigsource.com/index.php?topic=9318.0>
- [8] Chipmunk v.s Box2D
2009
<http://www.cocos2d-iphone.org/forum/topic/762>
- [9] Which Physics Engine is the best? - Kobold2D Documentation
2011
<http://www.kobold2d.com/pages/viewpage.action?pageId=918433>
- [10] Battle of the iOS Physics Engines
29 Augustus 2012
<http://flyclops.com/battle-of-the-ios-physics-engines-197>
- [11] Testing in Agile Software Development Enviroments with TMap NEXT
Sogeti, 2011
<http://www.tmap.net/tmap/downloads/testing-agile-software-development-environments-tmap-next>



B. Afstudeerplan

Afstudeerplan

Informatie afstudeerder en gastbedrijf (*structuur niet wijzigen*)

Afstudeerblok: 2012-2.1 (start uiterlijk 3 september 2012)

Startdatum uitvoering afstudeeropdracht: 3 september 2012

Inleverdatum afstudeerdossier volgens jaarrooster: 11 januari 2013

Studentnummer: 08016399

Achternaam: dhr van der Smeede

(*) *weghalen niet van toepassing*

Voorletters: R.P.J.

Roepnaam: Reinier

Adres: Pinaskade 64

Postcode: 2725ER

Woonplaats: Zoetermeer

Telefoonnummer: 079-3419187

Mobiel nummer: 06-53994814

Privé emailadres: r.p.j.vandersmeede@student.hhs.nl

Opleiding: Informatica

Locatie: Zoetermeer

(*) *weghalen niet van toepassing*

Variant: voltijd

Naam studieloopbaanbegeleider: V.E. Broeren

Naam begeleidend examiner: R. Ruijsenaars

Naam tweede examiner: V.E. Broeren

Naam bedrijf: Generali verzekeringsgroep nv
Afdeling bedrijf: De Nederlanden van Nu (verzekeren via internet)
Bezoekadres bedrijf: Diemerhof 36
Postcode bezoekadres: 1112 XN
Postbusnummer:
Postcode postbusnummer:
Plaats: Diemen
Telefoon bedrijf: 020 660 12 85
Telefax bedrijf:
Internetsite bedrijf: www.generalinl.nl en www.nederlandenvannu.nl

Achternaam opdrachtgever: dhr. Dijkstra
Voorletters opdrachtgever: Arjen
Titulatuur opdrachtgever: ir
Functie opdrachtgever: Directeur
Doorkiesnummer opdrachtgever: 12 85
Email opdrachtgever: arjendijkstra@nederlandenvannu.nl

Achternaam bedrijfsmentor: dhr. Dijkstra
Voorletters bedrijfsmentor: Arjen
Titulatuur bedrijfsmentor: ir
Functie bedrijfsmentor: Directeur
Doorkiesnummer bedrijfsmentor: 12 85
Email bedrijfsmentor: arjendijkstra@nederlandenvannu.nl

Doorkiesnummer afstudeerder: nog niet bekend
Functie afstudeerder (deeltijd/duaal): Mobile Game Developer
Titel afstudeeropdracht: Ontwikkeling van een mobiele game waarmee online verzekeraar De Nederlanden van Nu extra bezoek naar haar website trekt.4



Opdrachtschrijving

1. Bedrijf

Generali verzekeringsgroep nv is een traditionele verzekeraar die in Nederland schadeverzekeringen, inkomensverzekeringen, levensverzekeringen en pensioenverzekeringen aanbiedt via tussenpersonen. Generali richt zich vooral op inkomensverzekeringen en pensioenen, omdat dit ingewikkelde verzekeringen zijn waarvoor klanten zich vaak voor advies naar een tussenpersoon wenden. De rechtsvoorgangers van Generali verzekeringsgroep nv zijn sinds 1870 actief op de Nederlandse verzekeringsmarkt.

Generali is in Nederland met ongeveer 1% marktaandeel een kleine verzekeraar. Klein betekent bij verzekeraars in dit geval ongeveer 450 medewerkers en een jaaromzet van zo'n 500 miljoen euro.

Generali verzekeringsgroep nv maakt deel uit van de Generali Group. Dit is een groot internationaal concern met zo'n 90.000 medewerkers in meer dan 50 landen. Het hoofdkantoor staat in Trieste, Italië.

De Nederlanden van Nu is de zelfstandige verzekeringsdochter van Generali die richt op het via internet aanbieden van verzekeringen, rechtstreeks aan consumenten. De Nederlanden van Nu bestaat sinds 2006; vanaf medio 2011 worden via internet verzekeringen aangeboden.

De Nederlanden van Nu bestaat uit een kernteam van 7 personen met als belangrijkste verantwoordelijkheden:

1. Online Marketeer (marktbewerking; website)
2. Promotie Marketeer (marktbewerking; website)
3. Productmanager (aanbod; crowdsourcen; website)
4. Marketing Analyst (analyse/optimalisatie; website)
5. Directeur (overall verantwoordelijk)
6. Stagiair/afstudeerder 1
7. Stagiair/afstudeerder 2

De Nederlanden van Nu heeft maar 7 personen in dienst doordat belangrijke processen zijn uitbesteed, waaronder:

- Klantcontact (email, telefoon)
- Verzekeringsadministratie
- Behandelen schadeclaims

De Nederlanden van Nu mag bovendien desgewenst gebruikmaken van de stafleden van Generali waaronder: Juridische Zaken, Compliance, Risk Management, Actuarieel, HRM.

Alle 7 medewerkers zitten in één ruimte in het pand naast het pand waarin Generali is gevestigd.

Een aantal elementen uit de **strategie** van De Nederlanden van Nu zijn:

1. Klanten echt betrekken (o.a. door crowdsourcing)
2. Focus op onbetaald verkeer (o.a. door ontraditionele marketing)
3. Veel en snel experimenteren (motto "try a lot of stuff and keep what works")



2. Probleemstelling

De online verzekeringsmarkt bestaat nog niet zo heel lang, maar is al wel behoorlijk volwassen. Naast gevestigde partijen als Centraal Beheer, OHRA, FBTO en Univé zijn er in de afgelopen 5 jaar nieuwe partijen die met fors marketinggeweld de markt betreden (Ditzo, InShared, Allsecur). Ook vergelijkingssites als Independer claimen een steeds belangrijker positie in de markt.

De Nederlanden van Nu heeft niet het budget om via langdurige massamediale campagnes aan naamsbekendheid te werken. Bovendien is De Nederlanden van Nu niet in staat om een sluitende business case te maken voor deze marktbenadering: De Nederlanden van Nu gelooft dat Ditzo, InShared en Allsecur aanzienlijke verliezen accepteren om marktaandeel te komen. Dit is voor De Nederlanden van Nu niet acceptabel.

De hamvraag is: Hoe kan De Nederlanden van Nu zonder aanzienlijke budgetten te steken in massamediale campagnes, toch werken aan naamsbekendheid en in aanraking komen met mogelijke nieuwe klanten?

De Nederlanden van Nu gelooft dat dit mogelijk is door ontraditionele marketing. Bovendien gelooft De Nederlanden van Nu niet in wedden op één paard. De Nederlanden van Nu gelooft in de portfoliobenadering van innovatie: Door snel 10 dingen op de markt te brengen, wordt eerder succes bereikt dan door 1 ding heel lang en zorgvuldig voor te bereiden.

Het is acceptabel dat er van de 10 waarschijnlijk 8 niet succesvol zijn, eentje een beetje en eentje enorm succesvol onder de voorwaarde dat van de mislukkingen in elk geval een goede evaluatie wordt gemaakt zodat het leereffect niet verloren gaat.

Het ontwikkelen van een **game voor een mobiel platform** is één van de experimenten binnen de strategie van “try a lot of stuff and keep what works”. Doelstelling is dat de game PR-potentie moet hebben. De Nederlanden van Nu hoopt door een opmerkelijke game aandacht te kunnen genereren voor haar merk, met als resultaat meer verkeer naar haar website.

PR-waardigheid / kan bezoek naar de website opleveren zijn dus belangrijke criteria voor de game.

3. Doelstelling van de afstudeeropdracht

Ontwikkelen van een mobiele game waarmee De Nederlanden van Nu aandacht kan genereren. De diepgang van de game zal zowel in samenspraak met de opdrachtgever nader worden bepaald; Dit zal vroeg in het project gebeuren. Er kan uit worden gegaan van een 2D game van middelmatige complexiteit.

De opdracht omvat alle fasen van game development waaronder het bedenken. Dus gameconcept opstellen (samen met het team van De Nederlanden van Nu, er zijn al ruwe ideeën), gamedesign, implementatie, testen en het ervoor zorgen dat de game in de betreffende appstore staat, overdracht. Uiteraard wordt verwacht dat goede ontwikkelstandaards in ere worden gehouden (documenteren, plannen en rapporteren over voortgang etc.)

4. Resultaat

- Het resultaat is een werkende mobiele game app, te downloaden is vanuit een app-store, die on-time en on-budget is gerealiseerd conform specs.
- Gerealiseerde media aandacht valt buiten de verantwoordelijkheid van de opdracht.



5. Uit te voeren werkzaamheden, inclusief een globale fasering, mijlpalen en bijbehorende activiteiten

De globale fasering van het project zal als volgt vormgegeven worden:

Het project zal gemanaged worden door middel van SCRUM met sprints van 2 weken. Bij elke sprint hoort het schrijven van nieuwe code, het aanmaken van nieuwe content (art / sound) en het testen van het product; Dit wordt allemaal gedaan door de student. Het uiteindelijke doel van elke sprint is het opleveren van een product die werkt en getest is.

Zodra de applicatie redelijkerwijs geschikt is voor de app-store zal er direct een 'approval' proces in gang worden gezet bij apple om de app zo snel mogelijk in de app-store te krijgen. Het doel is om dit proces te starten rond sprint 3.

- Activiteit : Plan van Aanpak schrijven
 - Duur : 3 Dagen
 - Product : Plan van Aanpak
- Activiteit : Game Design / Requirements op stellen
 - Duur : 13 Dagen
 - Product : Game Design Document
 - Product : Product Backlog
- Activiteit : Training Objective C (zelfstudie)
 - Duur : 4 Dagen
 - Product : Kennis van het programmeren in Objective C
- Sprint 1
 - Duur : 10 Dagen
 - Product : Release 1
- Sprint 2
 - Duur : 10 Dagen
 - Product : Release 2
- Sprint 3
 - Duur : 10 Dagen
 - Product : Release 3
- Sprint 4
 - Duur : 10 Dagen
 - Product : Release 4
- Sprint 5
 - Duur : 10 Dagen
 - Product : Release 5
- Activiteit : Opbouwen Afstudeerdossier
 - Duur : 15 Dagen
 - Product : Afstudeerdossier



6. Op te leveren (tussen)producten

- Plan van aanpak*
- Game Design Document*
- Design Documentatie*
- Volledige mobile game op de appstore*

7. Te demonstreren competenties en wijze waarop

4 Uitvoeren Analyse door definitie van requirements

De requirements van de applicatie zijn nog niet bepaald en moeten door de student zelfstandig worden gedefinieerd na het inventariseren, analyseren en beschrijven van de gewenste informatievoorziening ten doelen een functioneel ontwerp op te stellen.

Er zullen verschillende stakeholders zijn, met elk hun eigen wensen en eisen t.o.v. de applicatie, ook is er aan te nemen dat er een groot aantal requirements zal zijn voor de applicatie.

2 Ontwerpen systeemdeel

De volledige applicatie zal door de student zelfstandig moeten worden ontworpen, het gaat hier om een objectgeoriënteerde applicatie, en er zal rekening worden gehouden met testbaarheid en herbruikbaarheid.

3 Bouwen applicatie

Het geheel zal door de student zelfstandig worden geschreven in Objective C language. Het bouwen van de objectgeoriënteerde applicatie zal gebeuren in een ontwikkelomgeving en er zal rekening worden gehouden met toekomstige wijzigingen, testbaarheid, herbruikbaarheid. De code zal onder versiebeheer staan. Objective C wordt ondersteund door het Cocoa Touch framework.

4 Initiëren en plannen van het testproces

Het volledige testproces zal door de student zelfstandig moeten worden uitgevoerd en hierbij hoort het initiëren en het plannen ervan. Hiervoor zal gebruik worden gemaakt van Tmap, er zal een testrisicoanalyse worden uitgevoerd en er zal zowel een detailtestplan als een mastertestplan worden gemaakt.

5 Uitvoeren van en rapporteren over het testproces

Het testproces zal zelfstandig via Tmap worden uitgevoerd en er zal aandacht zijn voor de herhaalbaarheid van de tests; Hiernaast zal er over het volledige proces worden gerapporteerd.



C. Plan van Aanpak



Mobile Game Development
Nederlanden van Nu

Projectleider : Reinier van der Smeede
Opdrachtgever : Arjen Dijkstra
Datum : 04-09-2012



1. Inleiding

Dit plan van aanpak beschrijft de aanpak van het project met betrekking tot mobile game development binnen de Nederlanden van Nu. Het document gaat in op de aanleiding en achtergrond van het project; Hiernaast wordt een duidelijke doelstelling geformuleerd. De kern van het document behandelt de organisatie van het project en bevat een planning.

2. Aanleiding

De online verzekeringsmarkt bestaat nog niet zo heel lang, maar is al wel behoorlijk volwassen. Naast gevestigde partijen als Centraal Beheer, OHRA, FBTO en Univé zijn er in de afgelopen 5 jaar nieuwe partijen die met fors marketinggeweld de markt betreden (Ditzo, InShared, Allsecur). Ook vergelijkingssites als Independer claimen een steeds belangrijker positie in de markt.

De Nederlanden van Nu heeft niet het budget om via langdurige massamediale campagnes aan naamsbekendheid te werken. Bovendien is De Nederlanden van Nu niet in staat om een sluitende business case te maken voor deze marktbenadering; De Nederlanden van Nu gelooft dat Ditzo, InShared en Allsecur aanzienlijke verliezen accepteren om marktaandeel te kopen. Dit is voor De Nederlanden van Nu niet acceptabel.

De hamvraag is: Hoe kan De Nederlanden van Nu zonder aanzienlijke budgetten te steken in massamediale campagnes, toch werken aan naamsbekendheid en in aanraking komen met mogelijke nieuwe klanten?

De Nederlanden van Nu gelooft dat dit mogelijk is door ontraditionele marketing. Bovendien gelooft De Nederlanden van Nu niet in wedden op één paard. De Nederlanden van Nu gelooft in de portfoliobenadering van innovatie: Door snel 10 dingen op de markt te brengen, wordt eerder succes bereikt dan door 1 ding heel lang en zorgvuldig voor te bereiden. “Try a lot of stuff and keep what works.”

3. Doelstelling

Het ontwikkelen van een game voor een mobiel platform is één van de experimenten binnen de strategie van “Try a lot of stuff and keep what works”. De Nederlanden van Nu hoopt door een opmerkelijke game aandacht te kunnen genereren voor haar merk, met als doel meer verkeer naar haar website.



4. Resultaat

De volgende producten worden opgeleverd aan De Nederlanden van Nu als resultaat van het project :

- *Een mobile game op de app-store van Apple*
- *De volledige broncode van de applicatie*
- *Een Game Design Document, waar alle inhoudelijke informatie over de game staat beschreven.*
- *Eventuele overige documentatie m.b.t. het design van de code (Klassendiagrammen, use-case beschrijvingen etc.)*

5. Afbakening

Het project betreft de ontwikkel- en testfase van de applicatie; Het beheer van de game valt buiten de scope. De broncode wordt ruim gedocumenteerd en uitbreidbaar achtergelaten, zodat het project in de toekomst gemakkelijk te beheren is door een extern persoon.

De verantwoordelijkheid voor de eventuele afwijzing van de applicatie voor de appstore ligt niet bij de projectgroep. Redelijkerwijs wordt hier echter wel moeite voor gedaan.

De projectgroep is niet verantwoordelijk voor eventuele PR aandacht of ontbreken hiervan.

6. Randvoorwaarden

Het project omvat de volgende randvoorwaarden voor een succesvolle uitkomst :

- *Voor ontwikkeling is de beschikbaarheid van een Intel gebaseerde Imac met minimaal OS X versie 10.6 noodzakelijk.*
- *Er moet een apple developer licentie aangeschaft worden.*
- *Het product moet uiterlijk 28 December opgeleverd worden aan Apple.*

7. Projectbeheersing

7.1 Tijd

Het project is georganiseerd door middel van Scrum, een software- ontwikkel methodiek. Het ontwikkeltraject van de applicatie bestaat uit 6 periodes van twee weken; Aan het einde van elk van deze periodes zal een deel van de functionaliteit zijn toegevoegd en wordt een werkende versie opgeleverd.

Alle functionaliteit wordt vooraf aan het ontwikkeltraject geprioritiseerd en eventuele vertragingen worden opgevangen door de functionaliteit door te schuiven naar de volgende periode. Dit zorgt ervoor dat de meest kritieke functionaliteit zo snel mogelijk in ontwikkeling gaat.



7.2 Kwaliteit

Aangezien het beheer van de applicatie buiten de scope valt is de kwaliteit van de code erg belangrijk. Een goed testproces is dus belangrijk. Omdat er goed getest moet worden in weinig tijd, worden er automatische tests geschreven voor alle kritieke functionaliteiten, die elke versie regressief kunnen controleren.

7.3 Informatie

De projectleider rapporteert wekelijks aan de opdrachtgever over de voortgang om transparantie te bieden over het proces; Elke twee weken zal de projectleider gezamenlijk met de opdrachtgever de prioriteit van de verschillende functionaliteiten die nog niet af zijn bepalen. De opdrachtgever bepaalt tenslotte wat het product nodig heeft!

8. Risico's

Risico	Maatregel	Verantwoordelijkheid
Apple stelt (te) hoge eisen om de applicatie te publiceren in de app-store.	Het traject van goedkeuring voor de app-store moet tijdig worden gestart.	Projectleider
Er is geen werkende versie aan het einde van het project.	De laatste werkende versie gebruiken. (max. 2 weken oud)	Projectleider
De Nederlanden van Nu is (te) traag met besluitvorming.	Vooraf met de opdrachtgever afspraken maken over hoe te handelen in situaties waarbij 'uitblijven van besluitvorming' de voortgang belemmert.	Projectleider
Geen commitment voor het game design.	Stakeholders van het bedrijf betrekken bij game design; Voldoende tijd inplannen voor besluitvorming.	Projectleider / Opdrachtgever

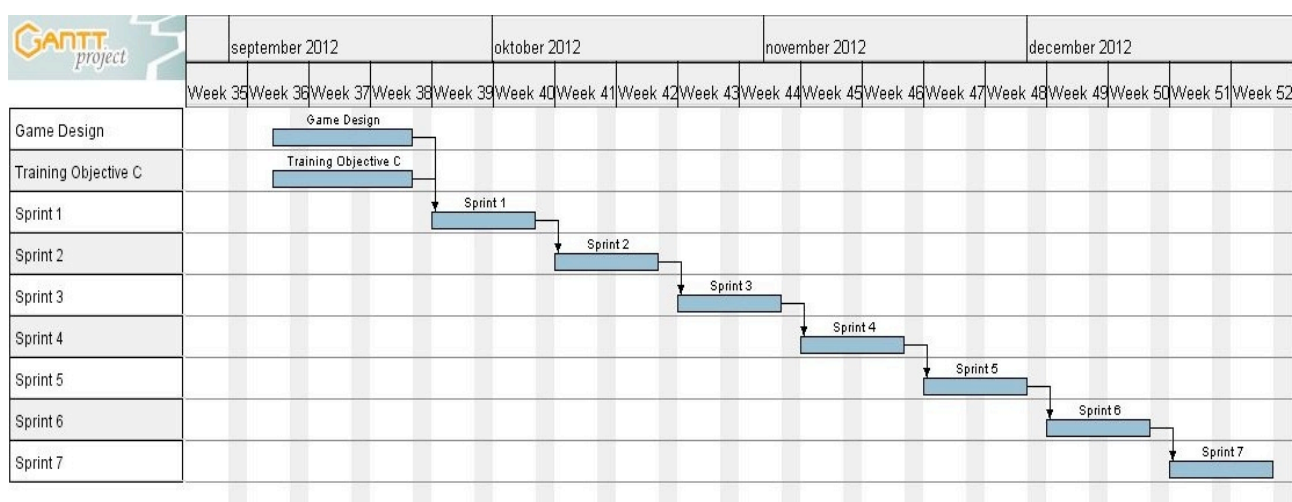
9. Planning

Door de onzekere natuur van softwareontwikkeling, en het ontbreken van requirements is het niet realistisch de planning vooraf te bepalen; Er is daarom gekozen om het project te organiseren vanuit de gedachtewijze van de softwareontwikkeling methodiek Scrum : In plaats van alles (tevergeefs) van te voren in te plannen worden er in totaal 7 'sprints' ingepland die elk 2 weken duren.

Aan het begin van het project wordt een concept ontwikkeld en daar zal een 'product backlog' uitvloeien, een groeiend set aan requirements die in de applicatie worden gebouwd.

Vooraf aan elke sprint wordt een kleine set van requirements uit deze 'product backlog' gehaald en die zullen tijdens de sprint worden gerealiseerd. De requirements worden tijdens de 2 weken gecodeerd, getest en opgeleverd in een werkend pakket. Hierdoor worden veranderingen snel opgepakt zonder dat de planning in de war gegooid wordt, en is er voor de opdrachtgever altijd een product dat 'af' is.

Het kritieke pad is in onderstaand figuur grafisch weergegeven.



D. Game Design Document 'Beredding'



DE NEDERLANDEN VAN NU

Student : Reinier van der Smeede

Opdrachtgever : Arjen Dijkstra

Versie 1.0

07-11-2012

Reiniervandersmeede@nederlandenvannu.nl

06-53994814



Versie Geschiedenis

Versie 0.1

Het document beschrijft in zijn huidige staat de game vooraf aan de daadwerkelijk ontwikkeling. Het is mogelijk dat niet alle features die staan beschreven realiseerbaar zijn in het ontwikkel-traject. Veel van de details zijn speculatief beschreven en dienen als een richtlijn.

Versie 0.2

De optie om een foto van een logo te uploaden is weggehaald in verband met de wensen van het management. Een groot gedeelte van de game art is toegevoegd.

Versie 0.3

De diagrammen van het technisch ontwerp zijn toegevoegd; Waaronder de usecases, usecase beschrijvingen, het klassendiagram en een sequentie diagram.

Versie 0.4

Veel van de requirements zijn veranderd in verband met een geheel nieuw design. Het spel is veranderd van een 'schiet het gebouw stuk' naar een 'red het gebouw van vlammen' spel. Een klein aantal hoofdstukken zijn ongewijzigd, waaronder het technisch ontwerp.

Versie 1.0

Alle informatie met betrekking tot de game in zijn eerste release versie. Rookmelder functionaliteit is niet meer haalbaar.



Inleiding

Het Game Design is een levend document wat als blauwdruk voor een spel functioneert. Het doel van het verslag is dan ook om het spel in zijn geheel te omschrijven en hierdoor is het van groot belang dat elk detail in het document is beschreven.

Het Game Design Document is een referentie voor de ontwikkelaar; Het zal door het gehele ontwikkel proces blijven groeien en aanpassen en zal uiteindelijk de game in zijn finale staat volledig beschrijven.

Hierdoor bestaat het risico dat het Game Design Document een gigantisch document wordt die enkel wordt bekeken om aanpassingen te maken in specifieke hoofdstukken; Hierdoor is er geprobeerd om elk punt zo beknopt mogelijk te beschrijven.

Het document zal ten eerste ingaan op de algemene achtergrond en filosofie van de game. Daarna zal er verder in worden gegaan op de daadwerkelijke gameplay elementen, met onder andere een beschrijving van de fysieke staat van de wereld.

Verder in het document zal verder in de diepte worden gegaan; Zo komen de gedetailleerde gedachten over de user-interface, doel en doorloop van de levels aan de orde.

Tenslotte is er een sectie toegewijd aan het kunstzinnige gedeelte van het spel: de visuele en auditieve elementen.



Sectie I : Game Overzicht

1.1 Filosofie

De Nederlanden van Nu vind dat schade zoveel mogelijk moet worden beperkt.

Bovendien stimuleert de Nederlanden van Nu het gebruik van een rookmelder om schade te voorkomen.

Het spel speelt in op de twee bovenstaande feiten, de game moet een situatie schetsen die duidelijk maakt dat het voorkomen en oplossen van dreigingen een belangrijke taak is voor iedereen.

Belangrijk is dat de applicatie wel naam genereert voor de Nederlanden van Nu; Er moet daarom mogelijk zijn om iets van een resultaat te delen op social media.

1.2 Features

De belangrijkste features in het spel zijn :

- Het voorkomen van schade door middel van vuur blussen
- Het kunnen delen van de reactie op social media

1.3 Genre

Het spel stimuleert vaardigheid en reactiesnelheid; Het is dus het beste te omschrijven als een 2D actie spel.

1.4 Doelgroep

De doelgroep is in eerste instantie alle klanten van de Nederlanden van Nu; Ook de kinderen van deze klanten moeten het spel kunnen spelen.

1.5 Game Flow Samenvatting

De applicatie zal beginnen met het logo van de Nederlanden van Nu, en zal hierna gaan naar een aantal schermen waar het spel kort wordt uitgelegd.

In het tweede scherm kan de speler via behendigheid een gebouw redden van vuur door een waterkanon te richten op het vuur.

Zodra de level is afgelopen kan de speler zijn of haar score uploaden op facebook.

1.6 Look and Feel

Het spel heeft een lichte, niet-realistische look and feel nodig en is daarom in een cartoony style vormgegeven.



De omgevingen in het spel zijn dus ook zo kleurrijk mogelijk ontworpen, en de geluiden zijn geïnspireerd door oude jaren 90 cartoons. De GUI bestaat vooral uit de kleuren van Nederlanden van Nu.

1.7 Scope

1.7.1 Aantal Schermen

Het spel heeft 4 schermen.

- Een laad scherm met het logo van de Nederlanden van Nu.
- Een ‘tutorial’ scherm, waar het spel kort wordt uitgelegd
- Het scherm voor het daadwerkelijke spel
- Het scherm voor het bericht voor de social media

1.7.2 Aantal Levels

Het spel bevat in eerste instantie een enkele level; Meerdere levels zijn het doel.

1.7.3 Aantal Voorwerpen

De volgende hoofd-voorwerpen zijn te onderscheiden :

- Het gebouw wat de speler moet redden, deze zal bestaan uit verschillende stukken
- Het waterkanon dat de speler kan controleren
- Water wat de gebruiker kan richten op het gebouw
- Vuur op het gebouw.
- De grond



Sectie II : Gameplay

2.1 Gameplay

2.1.1 Speelduur

De speelduur van het spel ligt op +-5 minuten.

2.1.2 Uitdaging

De moeilijkheidsgraad moet zodanig makkelijk zijn dat het grotendeels van de doelgroep het spel uit kan spelen. De uitdaging ligt vooral in het overtreffen van highscores van jezelf of anderen.

2.1.3 Doel

Het doel van het spel is om een gebouw te redden van vuur en hiermee zo min mogelijk water te gebruiken om de schade te beperken.

2.1.4 Game Flow

Het spel begint met het tonen van een gebouw. Dan krijgt de speler kort de tijd om rookmelders te plaatsen in het gebouw.

Vervolgens ontstaat er in groeiende mate vuur op verschillende plaatsen in het gebouw, waardoor het gebouw langzamerhand instort. De speler kan een waterkanon richten op het vuur waardoor het geblust wordt. Door deze twee gebeurtenissen ontstaan er schadekosten waardoor de waarde van het gebouw zal dalen.

Na een van te voren bepaalde tijd stopt de level en heeft de speler een score. Deze kan vervolgens worden gepost op facebook.

2.2 Mechaniek

2.2.1 Natuurkundige wetten

De wereld volgt de natuurkundige wetten van het framework van Box2D. Hierdoor zullen spelers eerder bekend zijn met de natuurkundige wetten aangezien deze vergelijkbaar zijn met de wetten van andere spellen zoals het bekende Angry Birds.



2.2.2 Beweging

Een vlam zorgt langzamerhand voor schade aan het gebouw, na een tijdje zal een gedeelte van het gebouw instorten;

De speler kan via het waterkanon water richting het gebouw schieten. Dit stopt met bewegen als het in aanraking komt met vuur of als het tegen blokkades aankomt.

De valsnelheid van objecten is niet realistisch weergegeven zodat de speler ruim tijd heeft om de verschillende bewegingen te observeren.

2.2.3 Objecten

Er zijn vijf objecten te onderscheiden :

- Het water wat de speler afschiet.
- Een stuk van het gebouw, waaruit het gehele gebouw is opgebouwd; Deze stukken kunnen kapot en los van elkaar bewegen om het gebouw in te laten storten.
- Vlammen die schade veroorzaken.
- Het waterkanon wat de speler controleert
- De grond

2.2.4 Acties

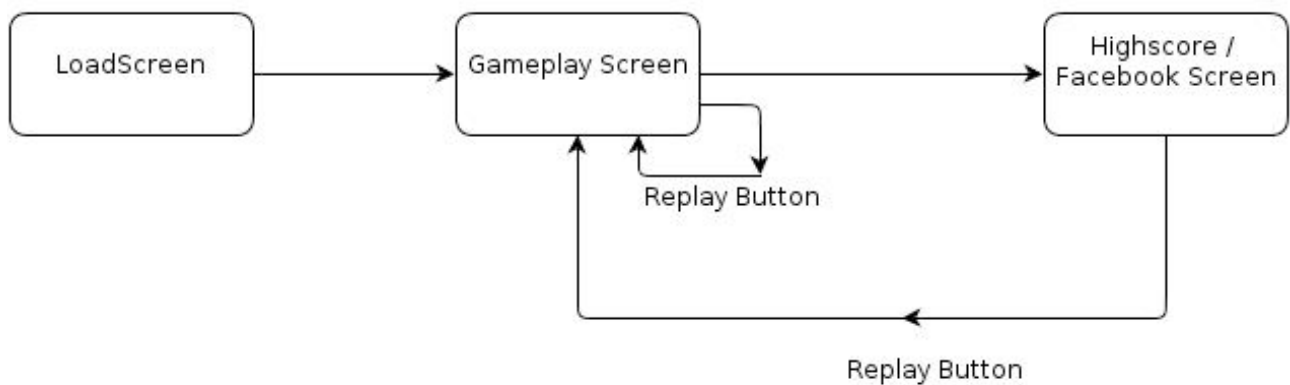
De speler kan de volgende acties uitvoeren :

- Het controleren van het kanon, waaronder beweging en afvuren van water
- Het posten van de 'eindfoto' op social media
- Het spel herstarten
- De camera bewegen



2.3 Screen Flow

2.3.1 Screen Flow Chart



2.3.2 Screen I

Het eerste scherm van de applicatie geeft het logo van De Nederlanden van Nu weer, gedurende deze tijd kan het daadwerkelijke spel geladen worden.

2.3.3 Screen II

Het tweede scherm laat het daadwerkelijke spel zien.

Rechts in het scherm is het doel (het gebouw) te zien; Met daarboven de huidige waarde van het gebouw.

Het waterkanon zijn links in het scherm te zien. Hier zal de gebruiker moeten drukken.

Onderin het scherm is een back-button met de optie om naar het vorige scherm te gaan en het spel opnieuw te starten.

2.3.3 Screen III

Het derde scherm laat bovenin de score behaalt door de speler zien.

Midden in het scherm is een foto te zien met het gebouw.

Aan de linkerkant van het scherm is een knop om het spel te herstarten.

Aan de rechterkant van het scherm is een knop om de foto op facebook te posten.

2.4 Herspeelbaarheid

Door de kleine omvang van het spel zal het matig herspeelbaar zijn. Hiervoor is de score bedacht; Zo is er een manier om jezelf en anderen uit te dagen om het spel nog een keer te spelen.

De natuurkundige wetten van het spel zorgen ervoor dat het spel elke keer net iets anders doorlopen wordt. Dit zorgt er ook voor dat de herspeelbaarheid verhoogd wordt. Ook de vlammen komen op onregelmatige plekken tevoorschijn.



Sectie III : Achtergrond

3.1 Verhaal

Het spel doorloopt een fictief verhaal van een gebouw waar vuur uitbreekt. Gelukkig is de brandweer aanwezig met een groot waterkanon! Het gebouw kan door middel van water gered worden waardoor schade wordt beperkt.

3.2 Game Wereld

3.2.1 Schaal

De spelwereld bestaat uit 2dimensionale figuren en is niet op schaal gebaseerd. Zo kan bijvoorbeeld een gebouw 10 cm zijn terwijl een persoonlijk figuur 3cm is.

3.2.3 De spelwereld

De spelwereld bestaat uit een grond, het gebouw en de heuvel waarop het waterkanon zich bevindt.

3.2.4 De speler

De speler wordt vormgegeven door een stick-figure in de stijl van een cartoon, met een brandweerauto en waterkanon.



Sectie IV : Levels

4.1 Level I

4.1.1 Beschrijving

De eerste level bestaat uit een grond, de speler en zijn waterkanon, en het gebouw. De gebruiker kan met zijn vingers het kanon besturen en gedurende 3min het gebouw proberen te beschermen tegen vlammen; Hierdoor gaat het gebouw instorten.

Afhankelijk van hoeveel water er wordt weggeschoten gaat de waarde van het gebouw omlaag.

De level eindigt na 3min, of als het gehele gebouw is ingestort.

4.1.2 Doel

Het doel is om zoveel mogelijk schade aan het gebouw te beperken en de waarde zo hoog mogelijk te houden.

4.1.3 Kritieke Pad

Hoe langer de speler bezig is met het level, hoe meer vlammen er ontstaan. De speler kan ook rookmelders plaatsen in het gebouw.

4.1.4 Einde

Zodra de 3 minuten over zijn, of als het gebouw geheel is ingestort, eindigt het level en kan de speler een foto van het resultaat plaatsen op facebook.



Sectie V : Interface

5.1 Visuele Interface

5.1.1 HUD

De Heads Up Display van het spel bevat tijdens het tweede scherm de volgende informatie :

- De waarde van het gebouw
- De resterende tijd
- Een terug knop, onderin in het midden

5.1.2 Rendering

Het renderen van de 2D omgeving zal gebeuren door middel van het Cocos2D framework, ondersteund door het Quartz-2D framework; Er is geen 3d-engine nodig.

5.1.3 Camera

De camera kan bewogen worden binnen de scope van het level.

5.1.4 Belichting

Er wordt geen gebruik gemaakt van belichting in verband met de 2D stijl en de cartoon uitstraling.

5.2 Controls

De controls zijn zoals verwacht op een iphone: geheel via de touchscreen.

Menu items zijn simpelweg aan te tappen.

Het schieten van water gebeurt door het waterkanon aan het zetten. Het kan ook gedraait worden door middel van het slepen van een vinger.

De camera wordt bewogen door een vinger over het scherm te slepen.

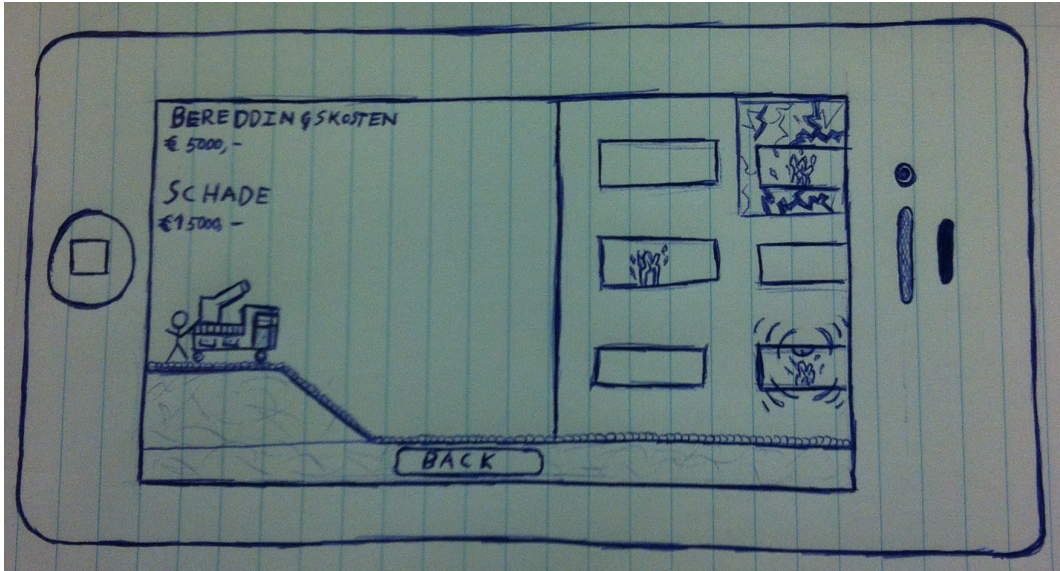


Sectie VI : Game Art

6.1 Concept Art

6.1.1 Screen Concept Art

6.1.1.2 Screen II



6.1.1.3 Screen III



6.2 Style Guide

Voor de visuele en tekstuele kant van het spel is de bestaande style guide van de Nederlanden van Nu het model.

6.2.1 Kleurgebruik

De Nederlanden van Nu hanteren de volgende hoofdkleuren :

Blauw : #002878

Grijsblauw : #647087

Rood : #e70033

Grijs : #333333

Er wordt geprobeerd om maximaal 1 rood element in beeld te hebben; Deze kleur wordt gebruikt om aandacht te trekken. Rood wordt daarom ook niet gebruikt voor grote stukken tekst.

6.2.2 Minimalistisch

Bij het ontwerp van de elementen wordt er uitgegaan van de no-nonsense gedachtegang van De Nederlanden van Nu. Er wordt gezocht naar een open en ruimtelijk ontwerp met minimale elementen.

6.2.3 Tekst

De gebruiker wordt altijd formeel aangesproken in directe en heldere taal. Het doel van de tekst is om oprecht en persoonlijk over te komen. Er wordt weg gebleven van populair en hip taalgebruik. Waar het relevant is, wordt er ruimte gegeven voor feedback en/of vragen.

Voor een betere leesbaarheid wordt de titel 'De Nederlanden van Nu' met mate gebruikt. Na de eerste vermelding kan het worden vervangen door 'wij'.



6.3 Omgevingen

6.3.1 Skybox

De skybox wordt als achtergrond gebruikt en simuleert een blauwe lucht.



6.3.2 Grond

De verschillende stukken grond vormen de voorgrond van het spel.



6.4 Karakters

6.4.1 Avatar

De speler wordt gepresenteerd als een stickfigure



6.5 GUI

Coming Soon!



6.6 Voorwerpen

6.6.1 Blokken

Hieruit wordt het gebouw opgemaakt.



6.6.2 Wapen

Het wapen wordt bovenop de brandweerauto getoont, zodat deze onafhankelijk van de brandweerauto kan bewegen.



6.6.3 Water

Het water wordt tot nu toe weergegeven met een sprite. De 'splash' onderin de atlas is bedoeld voor de death-animatie.



6.6.4 Vuur

De vuur atlas toont alle verschillende vormen van een vlam, waardoor het mogelijk is deze te animeren.



6.5 Audio

Het spel gebruikt de volgende audio files (royalty free) :

Faulty_Mechanics-Stephan_Schutze-279511784.caf	-	Beweging van kanon
Fire_Burning_JaBa-810606813.caf	-	Vuur ambient
midnight-ride.caf	-	Music
Robot-blip-Marianne_Gagnon-120342607.caf	-	Bonus blip
Rockslide_small-Sound_Explorer-1935053448.caf	-	Instorting gebouw
Short_triumpbhal_fanfare-Johm_Stracke-815794903	-	Winst music
Splash.caf	-	Water hit
spring-weather-1.caf	-	Ambient Bird Sounds
Water-Lisa_Redfern-1888623835.caf	-	Water ambient

Sectie VII : Technische specificaties

7.1 Hardware doelgroep

De applicatie is getest en ontworpen voor gebruik op een Iphone4 met IOS versie 5.1.1 (9B206).

Lagere hardware / versies zullen hoogstwaarschijnlijk geen problemen ondervinden maar zijn niet ondersteund.

7.2 Development Hardware

Om de applicatie te ontwikkelen en testen is de volgende machine gebruikt:

- Imac.

Operating System: OS X versie 10.8.1

Processor : 2.7 GHz Intel Core i5

Memory : 4 GB 1333 MHz DDR3

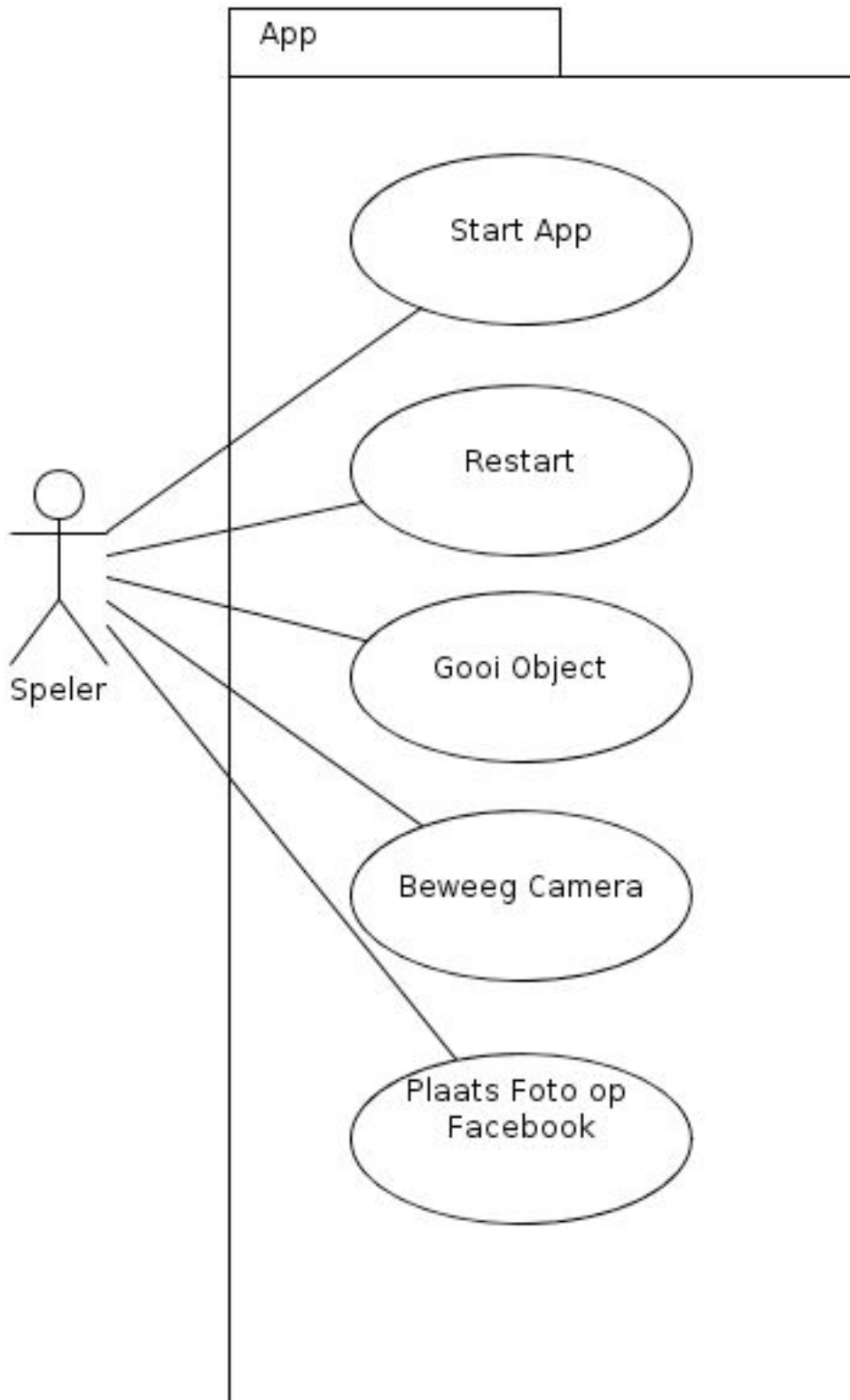
7.3 Development Software

- Xcode Versie 4.4.1 (4F1003)



Sectie VIII : Technisch ontwerp

8.1 Usecases



8.2 Usecase Beschrijvingen

8.2.1 Start App

Naam	Start App
Samenvatting	De gebruiker start de app en het spel start zich op
Actoren	Speler
Aannamen	Het spel is nog niet gestart
Beschrijving	1) De speler drukt op de knop van de app, in het IOS startscherm 2) De app wordt geopend op het scherm 3) De speler komt tijdelijk terecht op een laadscherm 4) De app laadt zichzelf 5) De speler komt terecht op het 'tutorial' scherm 6) De app wacht op feedback van de speler 7) De speler komt terecht op het eerste spel scherm
Uitzondering	[De app is al gestart] Als de app al is gestart komt de speler terecht op het scherm waar hij/zij het laatst is gebleven
Resultaat	De gebruiker ziet een scherm van het spel voor zich

8.2.2 Restart

Naam	Restart
Samenvatting	De gebruiker kan het daadwerkelijke spel restarten met een druk op de knop
Actoren	Speler
Aannamen	Het spel is gestart en in het spel-scherm en/of foto-scherm
Beschrijving	1) De speler drukt op de restart knop 2) Het spel gaat terug naar de initiele staat
Uitzondering	Geen
Resultaat	De gebruiker ziet het eerste spelscherm

8.2.3 Gooi Object

Naam	Gooi Object
Samenvatting	De speler kan in het spel een kanon aanzetten die objecten afvuurt
Actoren	Speler
Aannamen	Het spel is gestart en de gebruiker is in het spel-scherm
Beschrijving	1) De speler tikt op het scherm bij het wapen 2) Het wapen begint met objecten afvuren
Uitzondering	[De gebruiker sleept zijn vinger over het scherm tijdens stap 1] De app gaat over op de usecase 'Beweeg Camera' [De gebruiker sleept zijn vinger over het kanon tijdens stap 1] De app verandert de richting van het kanon [Het wapen is al aan het vuren tijdens stap 2] Het wapen stopt met vuren ipv stap 2 [De afgevuurde objecten botsen na stap 2] Objecten worden verwijderd [Het afgevuurde object raakt een vlam na stap 2] Het object wordt verwijderd en de vlam krijgt schade [Het afgevuurde object raakt niks na stap 2] Het object wordt verwijderd
Resultaat	Het wapen vuurt objecten af



8.2.4 Beweeg Camera

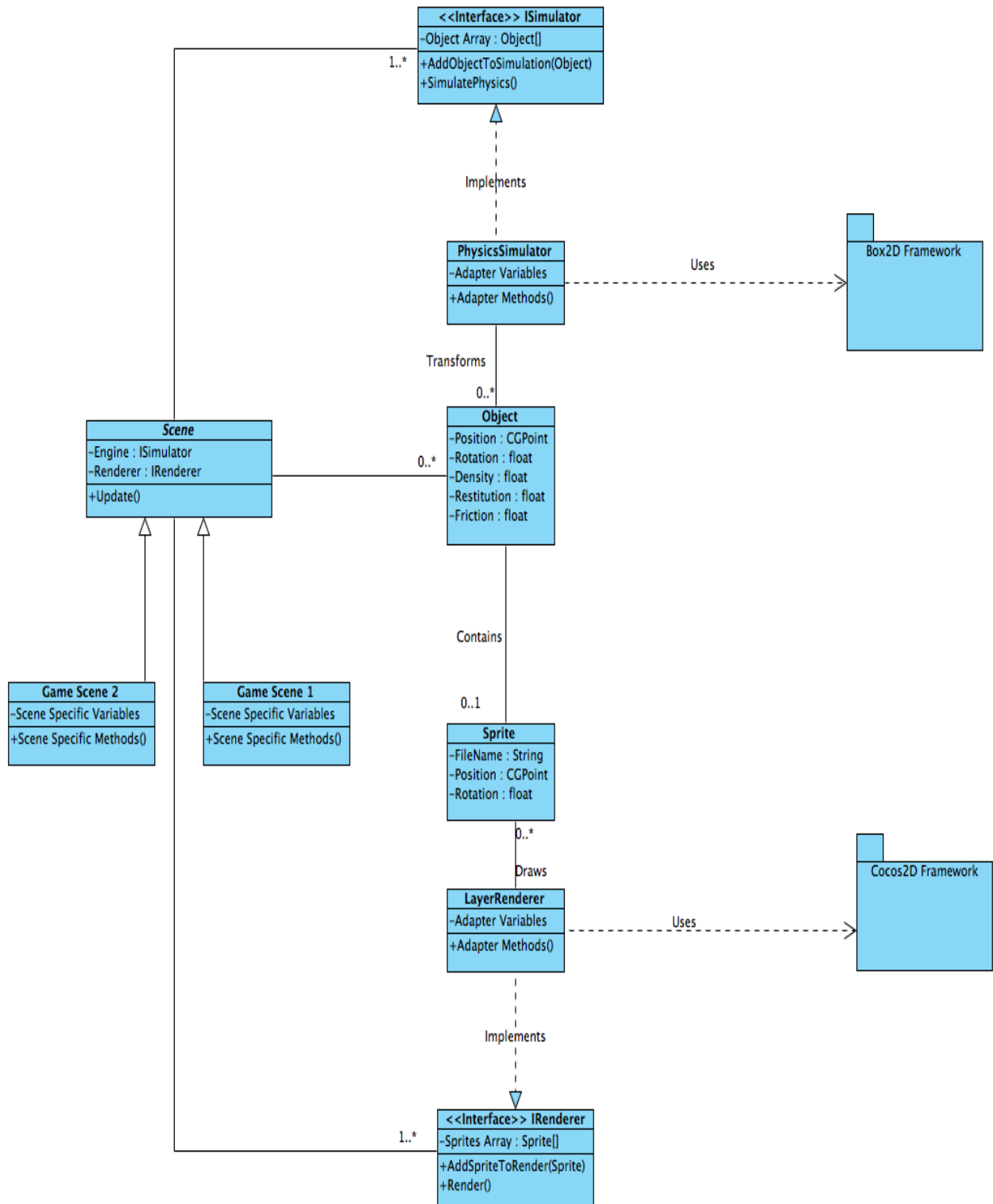
Naam	Beweeg Camera
Samenvatting	De speler kan door middel van het slepen van zijn/haar vinger de camera van het spel bewegen.
Actoren	Speler
Aannamen	Het spel is gestart en de gebruiker is in het spel-scherm
Beschrijving	1) De speler sleept met zijn vinger over het scherm 2) De camera van het spel beweegt in tegengestelde richting van de vinger.
Uitzondering	[De speler sleept een extreem korte afstand] De app registreert geen beweging en doet niks [De speler sleept zijn vinger over het kanon] De app gaat over op de usecase 'Gooi Object'
Resultaat	De camera toont een andere blik op het spelscherm

8.2.5 Plaats Foto op Facebook

Naam	Plaats Foto op Facebook
Samenvatting	De speler kan in het laatste scherm een foto delen op facebook
Actoren	Speler
Aannamen	De app bevindt zich in het laatste scherm
Beschrijving	1) Het applicatie toont een foto aan de gebruiker 2) De speler drukt op de knop 'delen op facebook' 3) De applicatie opent een link met facebook
Uitzondering	[De speler heeft geen facebook app] De applicatie toont een foutmelding [Er kan geen connectie worden gemaakt met facebook] De applicatie toont een foutmelding
Resultaat	De gebruiker bevindt zich in de facebook app met de foto in een facebook-bericht.

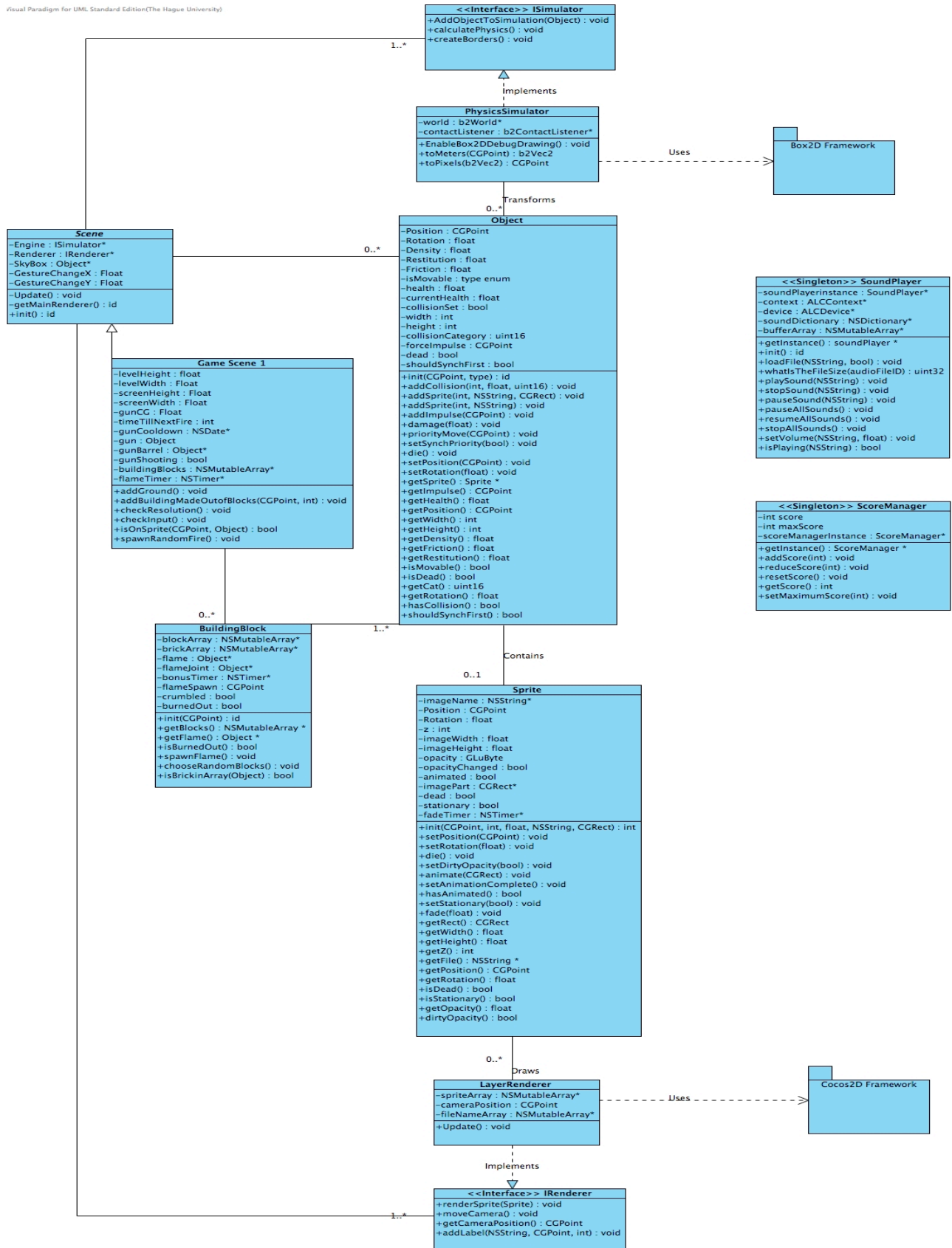
8.3 Klassendiagram

Abstract :

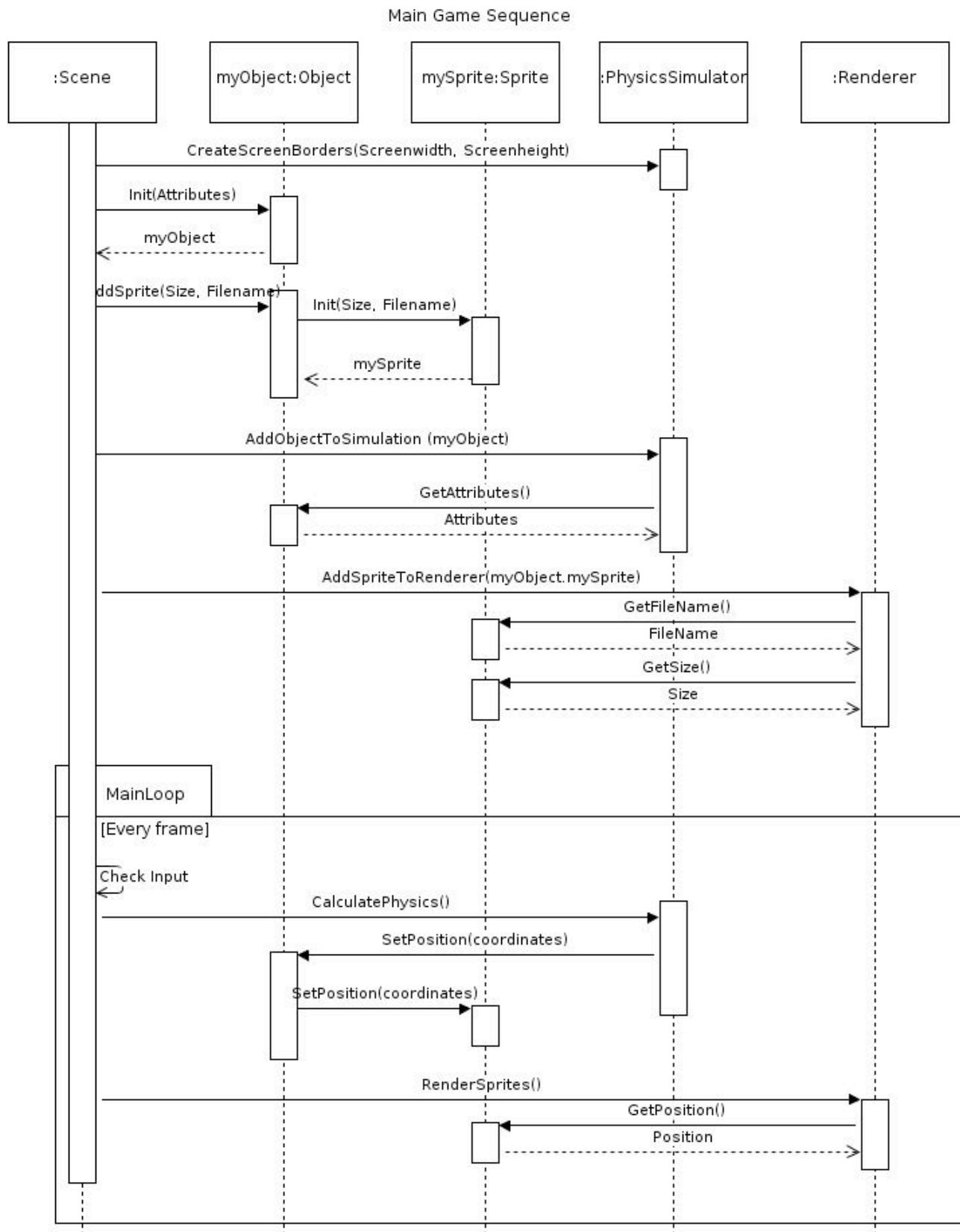


volledig :

Visual Paradigm for UML Standard Edition(The Hague University)



8.4 Sequentie Diagram





Master Test Plan

Nederlanden van Nu
Mobile Game Development

Testcoördinator : Reinier van der Smeede
Opdrachtgever : Arjen Dijkstra
Datum : 4 okt 2012



1. Inleiding

1.1 Project en projectdoelstelling

De online verzekeringsmarkt bestaat nog niet zo heel lang, maar is al wel behoorlijk volwassen. Naast gevestigde partijen als Centraal Beheer, OHRA, FBTO en Univé zijn er in de afgelopen 5 jaar nieuwe partijen die met fors marketinggeweld de markt betreden (Ditzo, InShared, Allsecur). Ook vergelijkingssites als Independer claimen een steeds belangrijker positie in de markt.

De Nederlanden van Nu heeft niet het budget om via langdurige massamediale campagnes aan naamsbekendheid te werken. Bovendien is De Nederlanden van Nu niet in staat om een sluitende business case te maken voor deze marktbenadering; De Nederlanden van Nu gelooft dat Ditzo, InShared en Allsecur aanzienlijke verliezen accepteren om marktaandeel te kopen. Dit is voor De Nederlanden van Nu niet acceptabel.

De hamvraag is: Hoe kan De Nederlanden van Nu zonder aanzienlijke budgetten te steken in massamediale campagnes, toch werken aan naamsbekendheid en in aanraking komen met mogelijke nieuwe klanten?

De Nederlanden van Nu gelooft dat dit mogelijk is door ontraditionele marketing. Bovendien gelooft De Nederlanden van Nu niet in wedden op één paard. De Nederlanden van Nu gelooft in de portfoliobenadering van innovatie: Door snel 10 dingen op de markt te brengen, wordt eerder succes bereikt dan door 1 ding heel lang en zorgvuldig voor te bereiden. “Try a lot of stuff and keep what works.”

Het ontwikkelen van een game voor een mobiel platform is één van de experimenten binnen de strategie van “Try a lot of stuff and keep what works”. De Nederlanden van Nu hoopt door een opmerkelijke game aandacht te kunnen genereren voor haar merk, met als doel meer verkeer naar haar website.

1.2 Doel van het mastertestplan

Het doel van dit mastertestplan is de betrokkenen van het project te informeren over de aanpak, de activiteiten en de op te leveren producten met betrekking tot het testtraject voor de mobiele game van de Nederlanden van Nu.



2. Opdracht

2.1 Doel en resultaat

Waarom moet er getest worden? Een goed testtraject heeft een aantal voordelen die het project een hogere kwaliteit geven; Zo kunnen hoge herstelkosten en gevolgschade in de productieomgeving voorkomen worden, doordat fouten tijdens het testen worden gevonden en binnen het ontwikkelproces kunnen worden opgelost.

Het hoeft niet voor te komen dat er veelvoudig fouten worden gevonden, in het geval dat alle bekende fouten zijn opgelost kan er vertrouwen worden geschept in het product en de kwaliteit ervan.

Hoewel het testen zelf veel tijd kost, wordt in de lange termijn tijd (en geld) bespaard en de kwaliteit van het algehele product gaat omhoog.

2.2 Scope

Niet alle factoren worden meegenomen in het testproces, testen kost namelijk veel tijd en een aantal factoren zijn te lastig om te voorspellen of te reproduceren. Dit hoofdstuk behandelt deze grenzen.

De applicatie is gericht op de iPhone4 met IOS 5.1.1 (9B206); Door het gebrek aan hardware in de ontwikkel- en testomgeving is het niet mogelijk om de applicatie te testen op andere versies van de iPhone en/of IOS. Ook aan de software kant wordt er rekening gehouden met de versie van Objective C (de programmeertaal) waarin gewerkt is.

Algemene ontwikkeltests (white box) worden tijdens het ontwikkelproces impliciet uitgevoerd en zullen niet meegenomen worden in dit document. Dit omvat onder andere het verwijderen van memory leaks in de code. Het product zal pas in de testfase komen zodra het geen memory leaks bevat, en zodra de functionaliteit op grote lijnen werkt; Het testproces is gericht op het vinden van onbekende bugs en regressie.

2.3 Unit Testing

Tijdens systeem releases zal rekening worden gehouden met regressie, er zullen automatische unit tests worden geschreven om deze regressie te detecteren en zo snel mogelijk op te kunnen lossen.

2.4 Acceptatie

Een black-box functionele acceptatie test zal uit worden gevoerd bij elke release, met de opdrachtgever als acceptant. Dit heeft als doel om te bekijken of de ingebouwde requirements bij de gebruikers voldoende en gewenst over komen of dat ze aangepast moeten worden.



3. Teststrategie

De beschikbare tijd om te testen is beperkt; niet alles kan even zwaar worden getest. Dus moeten er keuzes worden gemaakt. Daarbij hoort de testcapaciteit zo effectief mogelijk over het testtraject verdeeld te worden. Dit hoofdstuk beschrijft deze gang van zaken.

De teststrategie is gebaseerd op risicodenken : een systeem moet zodanig voldoen in de praktijk dat er geen onacceptabele risico's voor de organisatie moeten voor komen. Daar waar de oplevering van een systeemdeel veel risico's met zich mee brengt, is uitgebreid testen belangrijk; Het omgekeerde is ook waar, zolang er enkel kleine risico's aanwezig zijn, is het qua tijd en moeite niet te verantwoorden om dit systeemdeel zwaar te testen.

Deze risico's worden verder uitgewerkt in paragraaf 3.1 door middel van een risicoanalyse.

Deze risicoanalyse vormt de basis voor de teststrategie die in 3.2 wordt uitgelicht.



3.1 Risicoanalyse

Er is nooit sprake van een onbeperkte hoeveelheid testmiddelen en tijd. Daarom is het belangrijk om de testinspanning te relateren aan de verwachte risico's: grondig testen bij grote risico's, en licht of niet testen bij kleine risico's. De onderstaande tabel laat de ingeschatte risico's zien aan de hand van de van te voren bepaalde user stories.

Hierbij wordt de volgende formule gebruikt om het risico in te schatten :

$\text{Risico} = \text{Faalkans} * \text{Schade}$

De faalkans is hierbij bepaald door de foutkans en de frequentie van gebruik. De foutkans is de kans dat een onderdeel een fout bevat, hierbij wordt vooral gedacht aan de omvang van de code; Hoe meer code, hoe meer kans op een fout. Als het onderdeel vaak wordt gebruikt, is de kans groter dat deze fouten ook voor gaan komen.

Zowel de faalkans als de schade worden aangeduid door H (Hoog), M (Middelmatig) of L (Laag). Het risico wordt aangeduid door A (Groot risico), B (Middelmatig risico) of C (Laag risico). Hierbij wordt de volgende tabel gebruikt om de classificatie te bepalen :

	Schade bij falen		Kans op falen	
		Hoog	Midden	Laag
	Hoog	A	B	B
	Midden	B	B	C
	Laag	C	C	C

De risicoklassen zijn niet symmetrisch verdeeld. Het is immers belangrijker om een risico te beheersen met een hoge impact van schade en een lage kans op falen dan een risico met een hoge kans op falen waar de impact erg laag ligt.

De tabel op de volgende pagina geeft aan waar de classificatie van de risico's op is gebaseerd; Voor elk scherm is een faal-kans bepaald, en voor elk (deel)kenmerk is een schade geschat. Vervolgens is de risicoclassificatie bepaald door middel van de bovenstaande tabel.

Kenmerk	User Story	Schade	Game Scherm	Eind Scherm
Faalkans :			Hoog	Middelmatig
Functionaliteit	1. Als een gebruiker wil ik een grafische weergave van de spel elementen zien	Hoog	A	-
Functionaliteit	2. Als een gebruiker wil ik dat de spel elementen via natuurkundige wetten werken.	Hoog	A	-
Functionaliteit	3. Als een gebruiker wil ik water kunnen richten	Hoog	A	-
Functionaliteit	4. Als een gebruiker wil ik de camera kunnen bewegen.	Laag	C	-
Functionaliteit	5. Als een gebruiker wil ik dat er gevaar ontstaat zodat ik een doel heb.	Laag	C	-
Functionaliteit	6. Als een gebruiker wil ik een eindweergave kunnen posten op facebook	Middelmatig	-	B
Functionaliteit	7. Als een gebruiker wil ik geluiden kunnen horen	Laag	C	C
Usability		Middelmatig	B	B
Performance		Middelmatig	B	B

A = Hoog risico
 B = Middelmatig risico
 C = Laag risico



3.2 Teststrategie

Voor elk risico uit de risicoanalyse is de risicoclassificatie bepalend voor de zwaarte van de test. Hierbij is de classificatie A het hoogste, en C de laagste. De teststrategie is er vervolgens op gericht de hoge risico's zwaarder te testen vergeleken met de lage risico's.

De tabel laat een relatief testniveau zien, een zware test betekent hier niet dat de test uit gevoerd zal worden op de zwaarst mogelijke manier. De daadwerkelijke diepte van de test zal verder worden uitgewerkt in een detail-testplan.

Kenmerk	Risicoklasse	Unit Testing	Functionele Acceptatie Test
Functionaliteit			
User Story 1	A	○ ○ ○	○
User Story 2	A	○ ○ ○	○
User Story 3	A	○ ○ ○	○
User Story 4	C	○	○
User Story 5	C	○	○
User Story 6	B	○ ○	○
User Story 7 - Game Scene	C	○	○
User Story 7 - Facebook	C	○	○
Usability	B	-	○
Performance	B	-	○

A = Hoge Risicoklasse
B = Middelmatige Risicoklasse
C = Lage Risicoklasse

○ = Beperkte Test
○ ○ = Gemiddelde Test
○ ○ ○ = Zware Test

- = Geen aandacht



4. Aanpak

In dit hoofdstuk wordt per testsoort de teststrategie vertaald naar een concrete testaanpak. Na het lezen van dit hoofdstuk moet dan ook duidelijk zijn hoe de risico's van hoofdstuk 4 afgedekt worden. Hierbij is Reinier van der Smeede verantwoordelijk voor het gehele testproces.

4.1 Testsoorten

Voor dit testtraject worden de volgende testsoorten onderkend :

Unit-testing	: Functionaliteit
Functionele Acceptatie Test	: Functionaliteit, Usability, Performance

4.2 Unit Testing

4.2.1 Doel

Het unit testen heeft als doel de functionele requirements van de applicatie te testen op kwaliteit en zijn herhaalbaar, zodat regressie kan worden gedetecteerd en tegen gegaan worden.

4.2.2 Korte beschrijving

Voor elk deelsysteem worden unit testen geschreven die functioneren als automatische tests. Hiervoor wordt binnen de ontwikkelomgeving een testomgeving opgesteld die tijdens elke build een test kan uitvoeren om regressie te detecteren. Deze fouten zullen meestal per direct opgelost kunnen worden. De dekking van de unit tests zal afhangen van het risico wat het deelproces omvat.

4.2.3 Coverage

De diepte van een unit test wordt gemeten aan de hand van de coverage die de test zal hebben; In andere woorden, welke paden in het programma worden in de test doorlopen en geëvalueerd?

De zwaarte van deze coverage zal worden bepaald door de risicoclassificatie (tabel 1), en wordt dus per deel-functionaliteit getest, deze deel functionaliteit wordt vanaf dit punt aangegeven als unit.

De verschillende categorieën van coverage worden onderscheiden :

Function Coverage - Is elke functie van de unit doorgelopen?

Statement Coverage - Is elke regel in de code van de unit een keer uitgevoerd?

Condition Coverage - Is elke mogelijke combinatie van condities in een beslissing getest? (In het geval van meerdere condities in een statement)



In hoofdstuk 2 is bepaald of een unit een beperkte, middelmatige of zware test krijgt; Er is namelijk niet onbeperkte tijd voor testen en er moeten keuzes gemaakt worden aan de hand van de risico's. De coverage van de unit test wordt dan ook bepaald door de test-zwaarte van tabel 1 :

Beperkte Test : Function Coverage, Statement Coverage

Gemiddelde Test : Function Coverage, Statement Coverage

Zware Test : Function Coverage, Statement Coverage, Condition Coverage

4.3 User Acceptatie Test

4.3.1 Doel

De user acceptatie test heeft als doel de ingebouwde requirements te toetsen of ze samenhangen met de user stories.

4.3.2 Korte Beschrijving

Na elke release zal een user acceptatie test gebeuren met de acceptant; Hier zullen de ingebouwde user stories worden getoetst, zonder enige voorkennis van de werking (black-box). Als de applicatie werkt zoals verwacht met betrekking tot de user story is de test geslaagd.



F. SCRUM Documenten

Notulen Scrum Meeting I

Mbt GGD :

- De doelgroep voor het spel is in principe iedereen boven de 18 jaar
- Wie de speler bombardeert mag iets generieker. Het logo hoeft niet enkel van een verzekeraar te zijn maar kan ook zijn van bijv. de KPN / Voetbal maatschappij etc.
- Frameworks zijn outdated

Mbt Sprint I :

- Sprint Backlog :
 - ID 37 : Testomgeving opzetten
 - ID 38 : Testdocumentatie schrijven (Master test plan / risicoanalyse)
 - ID 39 : Als een gebruiker wil ik mooie grafische elementen in het spel
 - Totaal 47 Ideal Hours.

Notulen Scrum Meeting II

Backlog is overzichtelijker gemaakt, en het foto maken gedeelte is eruit.

Algemeen :

- Het Master test plan wordt naar Arjen opgestuurd als referentie
- Het stageverslag kan per sprint opgestuurd worden naar Arjen zodat feedback mogelijk is.
- Het uploaden van een foto is niet meer mogelijk in verband met wensen van het management

mbt Sprint I :

- Het grafische gedeelte (previously ID 39) is van voldoende kwaliteit
- Er moeten minder ideal hours ingeplanned worden in een sprint zodat er meer tijd besteed kan worden aan het stageverslag

mbt Sprint II :

- Aanstaande Donderdag heeft Arjen een vergadering met Jaap met betrekking tot het game concept.
- Deze sprint heeft minder functionele backlog items ivm het bovenstaande
- Sprint Backlog :
 - ID 2 : De functionaliteit van het tonen van de grafische elementen moet getest worden
 - ID 4 : De functionaliteit van de natuurkundige wetten moet getest worden
 - ID 23 : Ontwerpen systeem in UML
 - Totaal 40 Ideal Hours.



Notulen Sprint Meeting III

Algemeen :

- Na een meeting met Jaap is gebleken dat het idee/design niet mogelijk is door concerns van het management; er moet een geheel nieuw idee komen.

mbt Sprint II :

- Het design is gepresenteerd en toegelicht aan Arjen
- Het design is zodanig veranderd mbt sprint I en de code moest aangepast worden, hierdoor is de backlog item uitgelopen. Voor ID 2 en 4 is geen tijd gevonden en deze moeten doorschuiven naar de volgende sprint.

mbt Sprint III :

- Het bedenken van een nieuw design is topprioriteit, een brainstorm inplannen wordt zsm gedaan
- ID 2 en 4 van vorige sprint schuiven door naar deze sprint
- Sprint Backlog :
 - ID 2 : Renderer testen
 - ID 4 : Simulator Testen
 - ID 20 : Game Design
 - Totaal 36 Ideal Hours

Notulen Sprint Meeting IV

Algemeen :

- Het management heeft besloten om te gaan voor het 'brand-blus' idee

mbt Sprint III :

- Door het uitblijven van een besluit is afgelopen sprint vooral besteed aan het schrijven en verbeteren van het afstudeerverslag. Hiernaast is een facebook developer-account aangevraagd en is de app toegevoegd aan de licentie.
- Er is nog geen game design document

mbt Sprint IV :

- Er is nog geen product backlog voor het nieuwe spel, zodra dit er is worden er eventueel meer backlog items toegevoegd aan de sprint.
- Sprint Backlog :
 - Game Design Document - Schatting 14 ideal hours
 - Grafisch Design - Schatting 13 ideal hours
 - Totaal 27 Ideal Hours (voorlopig)



Notulen Sprint Meeting V

Algemeen :

- Vorige Sprint is Reinier naar het 60% feedback gesprek geweest op school. Naast wat opmerkingen was het verslag als voldoende beoordeeld
- Uit het 60% gesprek is uitgekomen dat er unit tests moeten komen, anders wordt de competentie als onvoldoende beoordeeld
- Vorige week ziek, dus deze sprint is 1 week later begonnen

mbt Sprint IV:

- Game Design Document is af, aantal opmerkingen van Arjen
- Bereddingskosten klopt niet met het spel, eerder preventie kosten. Voorlopig enkel schade kosten in het game design.
- Graphics design is prima; Alleen het hoofd van Arjen uit het spel

mbt Sprint V :

- Iets minder uren ingeplanned omdat er veel moet gebeuren aan het afstudeerverslag
- Sprint Backlog :
 - ID 4 : Physics Simulator Testen
 - ID 7 : Als een gebruiker wil ik water kunnen richten die vlammen blust
 - ID 11 : Als een gebruiker wil ik dat er gevaar ontstaat door vlammen die ik kan blussen
 - ID 15 : Als een gebruiker wil ik dat het level steeds moeilijker wordt en na 3 minuten concludeerd
 - Totaal 28 uur

Notulen Sprint Meeting VI

Algemeen :

- De afstudeerperiode is bijna afgelopen. Er is besloten om Reinier nog een aantal weken in te huren om het product af te maken. Beide kanten zijn er mee eens dat het product nog niet appstore-waardig is
- Deze sprint ligt de focus op het halen van de afstudeerperiode.

mbt Sprint V :

- Alles af, prima
- Unit test van Renderer heeft problemen ivm limitaties in het framework

mbt Sprint VI :

- Weinig uren ingeplanned zodat er veel tijd kan worden besteed aan het afstudeerverslag
- Sprint Backlog :
 - Een willekeurige unit test
 - Usability improven



Product Backlog

ID	User Story	Est. Hr	Priorit y
1	Als een gebruiker wil ik de camera van het spel kunnen bewegen met mijn vinger	0	
2	Deze functionaliteit moet goed getest zijn.	6	
3	Als een gebruiker wil ik water kunnen richten die vlammen kunnen blussen	0	
4	Deze functionaliteit moet goed getest zijn.	8	
5	Als een gebruiker wil ik dat het water natuurkundig gesimuleert is	12	
6	Deze functionaliteit moet goed getest zijn.	12	
7	Als een gebruiker wil ik dat er gevaar ontstaat door vlammen die ik kan blussen	0	
8	Deze functionaliteit moet goed getest zijn.	6	
9	Als een gebruiker wil ik rookmelders kunnen plaatsen	8	
10	Deze functionaliteit moet goed getest zijn.	8	
11	Als een gebruiker wil ik dat het level steeds moeilijker wordt en na 3 minuten concludeerd	0	
12	Deze functionaliteit moet goed getest zijn.	6	
13	Als een gebruiker wil ik een score kunnen behalen en deze zien op het scherm	0	
14	Deze functionaliteit moet goed getest zijn.	8	
15	Als een gebruiker wil ik een eindweergave van het behaalde resultaat kunnen posten op facebook	24	
16	Deze functionaliteit moet goed getest zijn.	16	
17	Als een gebruiker wil ik mijn spel kunnen herstarten	4	
18	Deze functionaliteit moet goed getest zijn.	4	
19	Als een gebruiker wil ik dat ik uitleg krijg over hoe het spel werkt	8	
20	Deze functionaliteit moet goed getest zijn.	8	
21	Als een gebruiker wil ik geluiden horen	0	
22	Deze functionaliteit moet goed getest zijn.	10	



ID	User Story	Est. Hr	Priority
23	Als een gebruiker wil ik mooie grafische elementen in het laatste scherm (GUI)	6	
	TOTAL Backlog Items		154

Sprint backlogs

Sprint I

User Story	Tasks	Est. Hr
Testomgeving opzetten		9
	Unit Test Omgeving	9
38 Test Documentatie Schrijven		12
	Master Test Plan	7
	Risico Analyse	5
39 Als een gebruiker wil ik mooie grafische elementen in het spel		26
	Sprites code	9
	Block.jpg	1
	Ground.jpg	3
	Background.jpg	3
	Bullet.jpg	1
	Explosion.jpg	2
	Cannon.jpg	2
	Hill.jpg	2
	Arjen.jpg	2
	Cloud.jpg	1



Sprint II

User Story	Tasks	Est. Hr
2 De functionaliteit van de grafische weergave testen		8
	Unit Testing	6
	Memory Leaks checken	1
	Documentatie	2
4. De functionaleit van de natuurkundige wetten testen		8
	Unit Testing	6
	Memory Leaks checken	1
	Documentatie	2
23 Ontwerpen Systeem (UML)		25
	Klassendiagram	9
	Use Cases	2
	Seq. Diagram	3
	Code aanpassen aan design	11



Sprint III

User Story	Tasks	Est. Hr
20 Game Design		20
	Brainstorm	4
	Game Design Document aanpassen	12
	UML Aanpassen	4

Sprint IV

User Story	Tasks	Est. Hr
Game Design Document		14
	Game Design Document	14
39 Als een gebruiker wil ik mooie grafische elementen in het spel		15
	Building	4
	Water.jpg	2
	FireAtlas.png	4
	FireTruck.png	3
	WaterSplash.png	1
	Warning.png	1



Sprint V

User Story	Tasks	Est. Hr
4 Testen Physics		8
Unit Testing		6
Memory Leaks		2
7 Als een gebruiker wil ik water kunnen richten die vlammen kunnen blussen		8
Water Richten		2
Water Aanzetten		2
Water Collision		2
Water kan vuur blussen		2
11 Als een gebruiker wil ik dat er gevaar ontstaat door vlammen die ik kan blussen		6
Vuur doet damage		1
Vuur vernietigd gebouw		4
Vuur spawned		1
15 Als een gebruiker wil ik dat het level steeds moeilijker wordt		6



Sprint VI

User Story	Tasks	Est. Hr
4 Testen Renderer		8
Unit Testing		6
Memory Leaks		2
7 Als een gebruiker wil ik geluiden kunnen horen		20
Geluiden Code		6
WaterSplash.wav		2
Fire.wav		2
Crumble.wav		2
Score.wav		2
Music.mp3		2
GunMovement.wav		2
WaterFiring.wav		2
11 Als een gebruiker wil ik een score kunnen behalen		8
Score on screen		4
Fire reduces Score		2
Water reduces Score		2

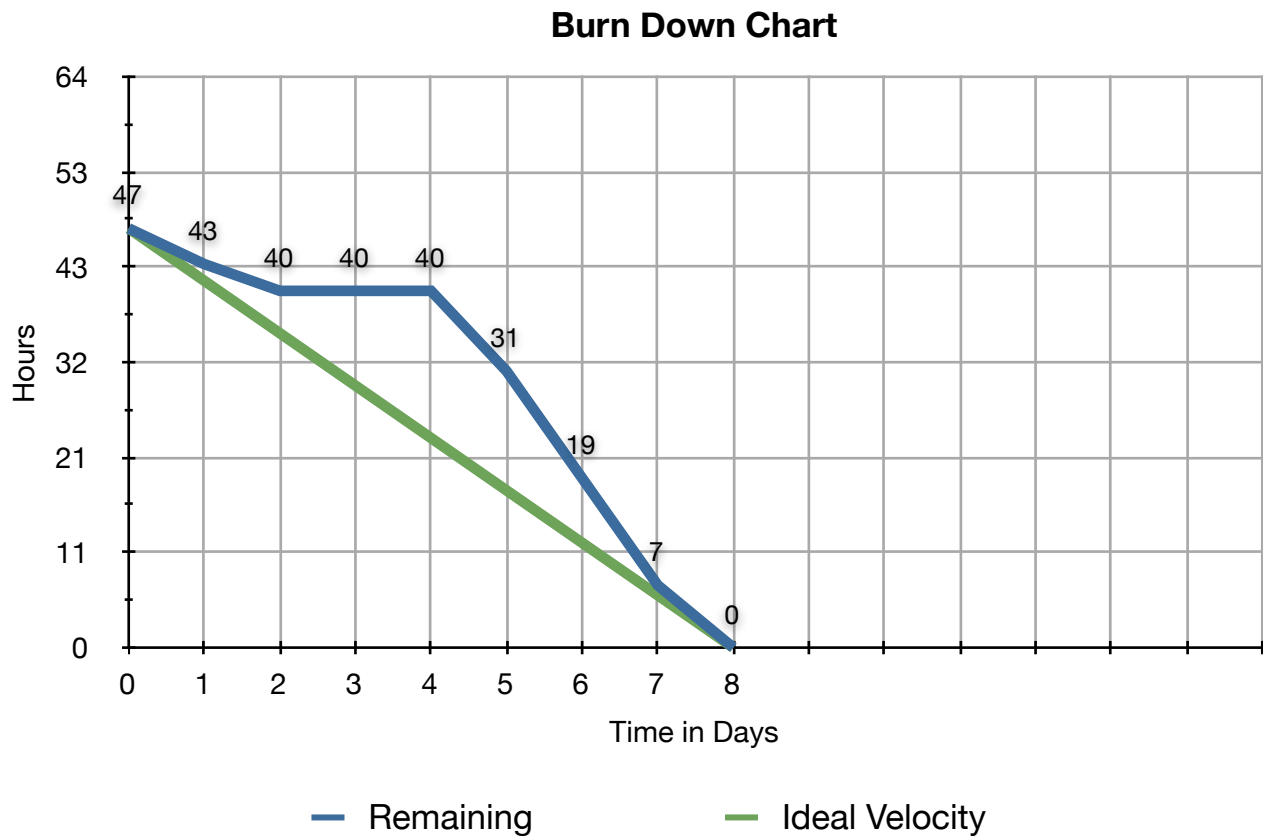
Sprint VII

User Story	Tasks	Est. Hr
4 Testen Renderer		8
Unit Testing		6
Memory Leaks Checken		2
X Usability Improvements		10
Controls		5
Level End Feedback		2
Code opruimen		3



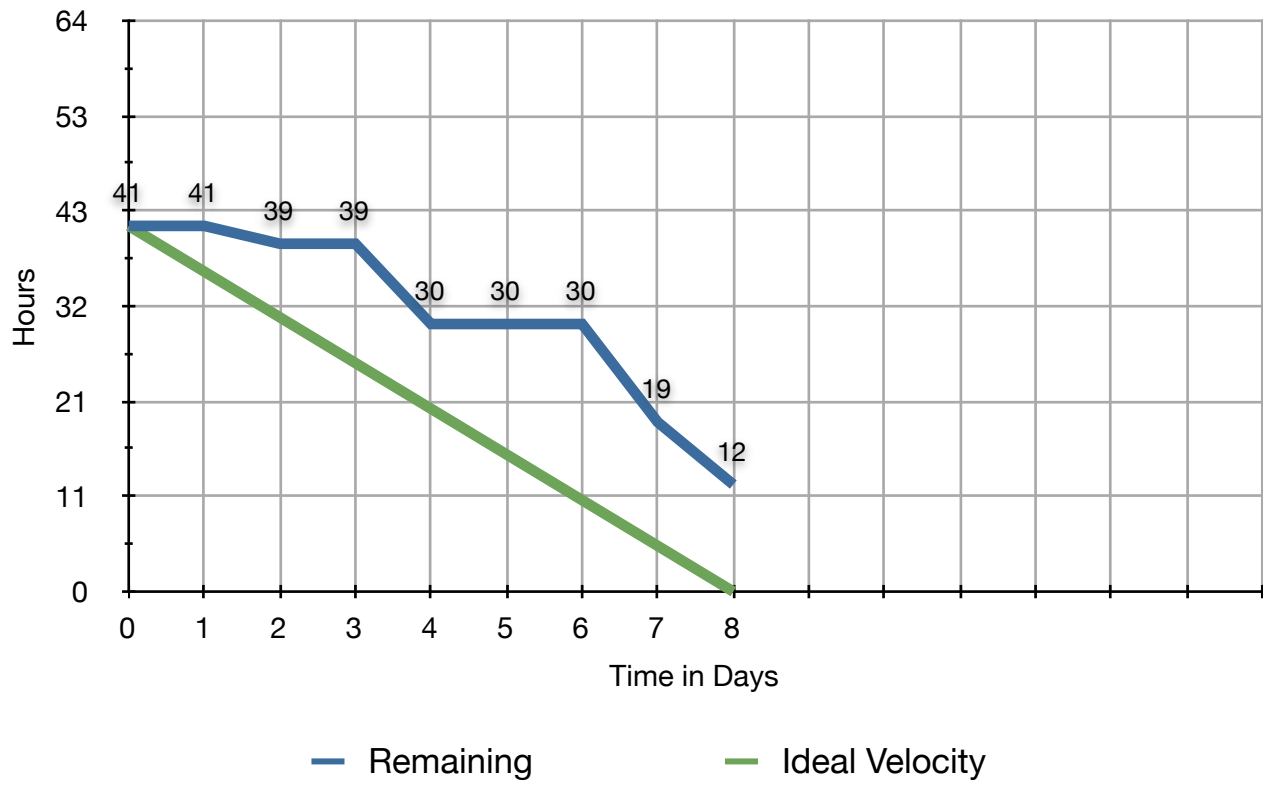
Burndown Charts

Sprint I



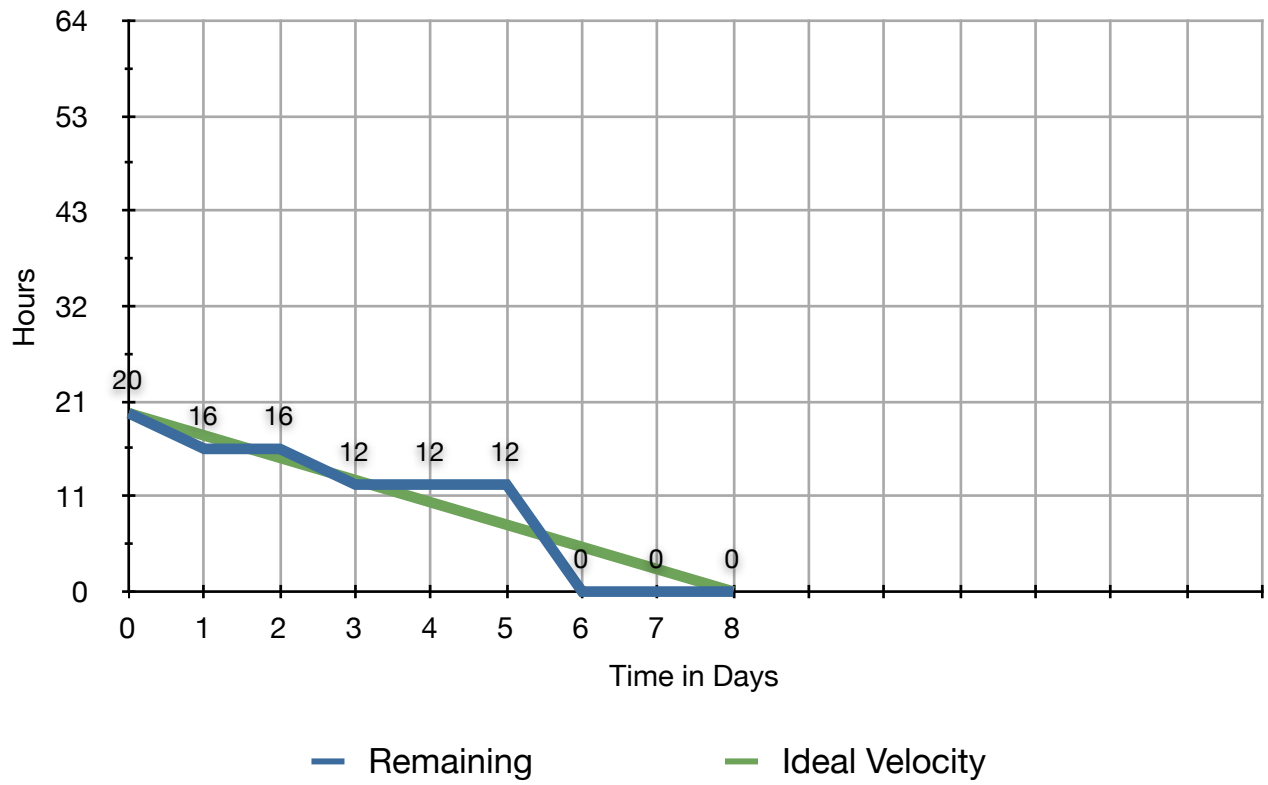
Sprint II

Burn Down Chart



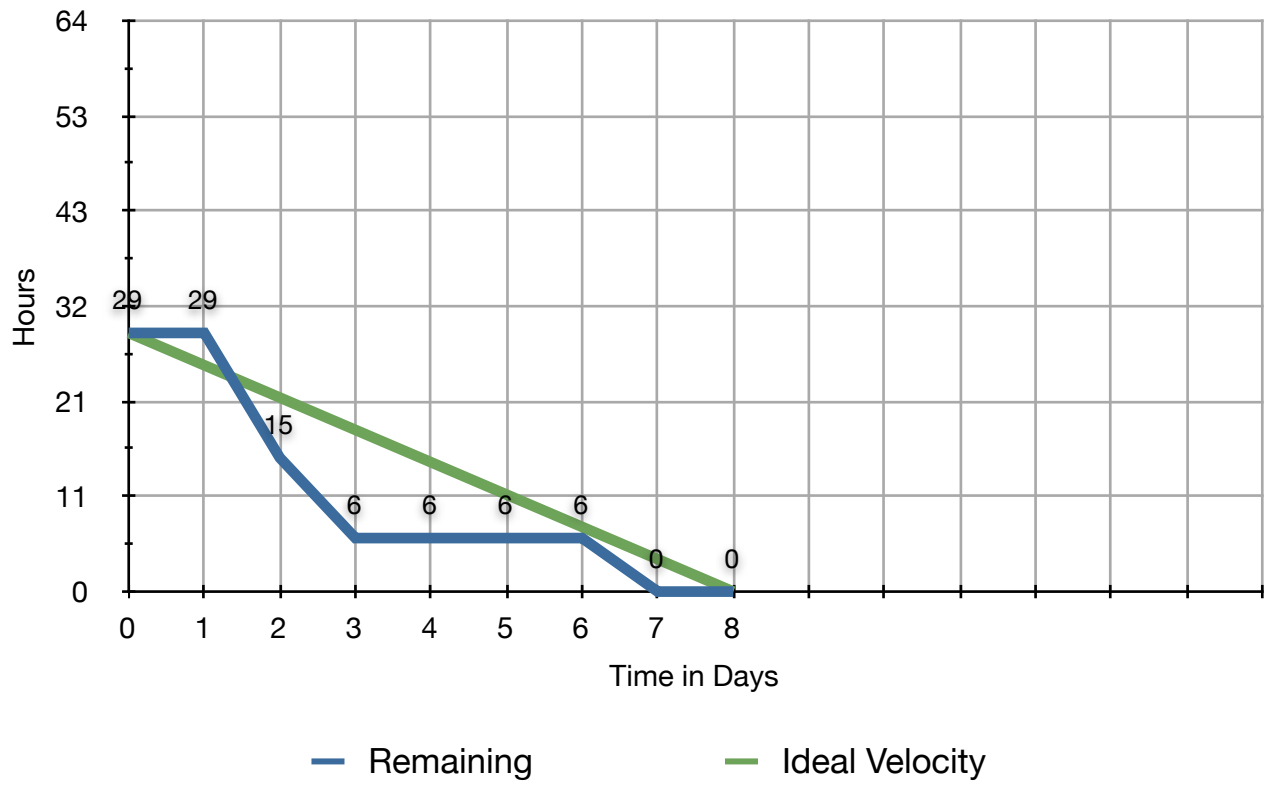
Sprint III

Burn Down Chart



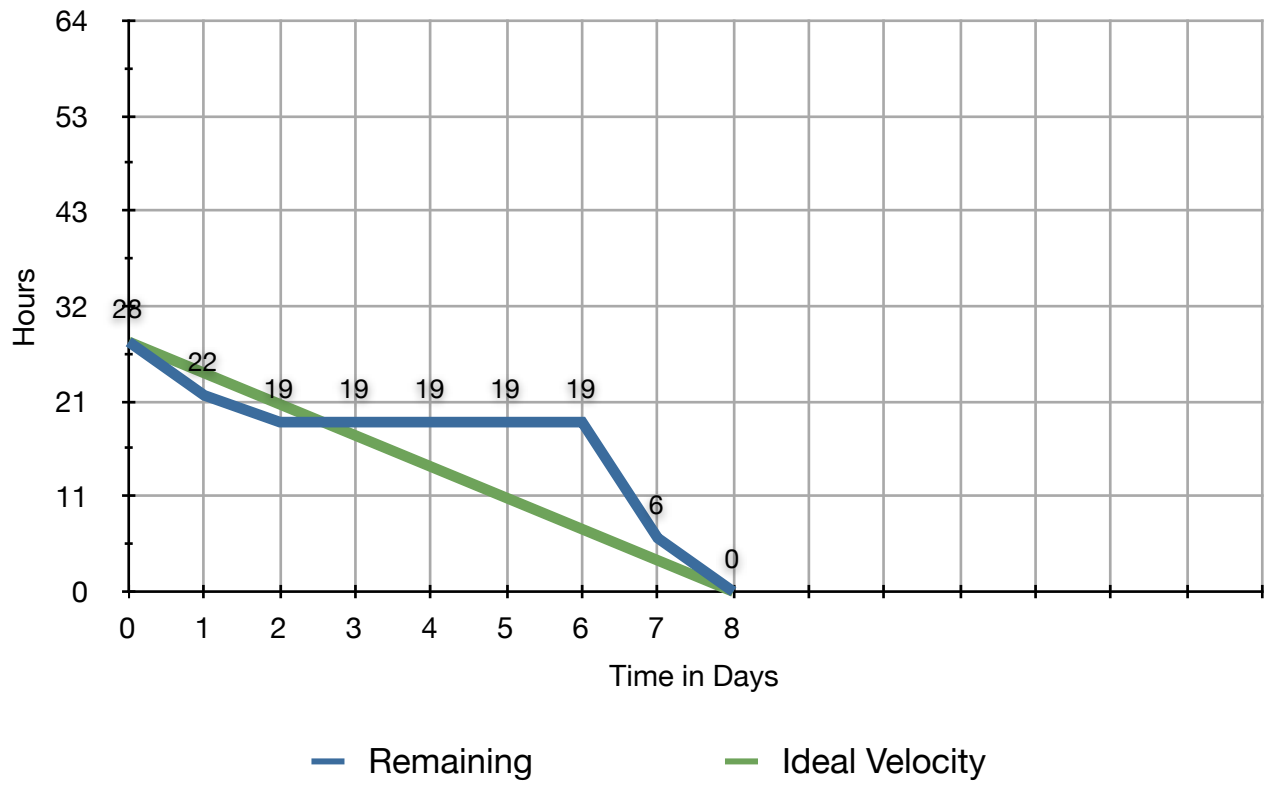
Sprint IV

Burn Down Chart



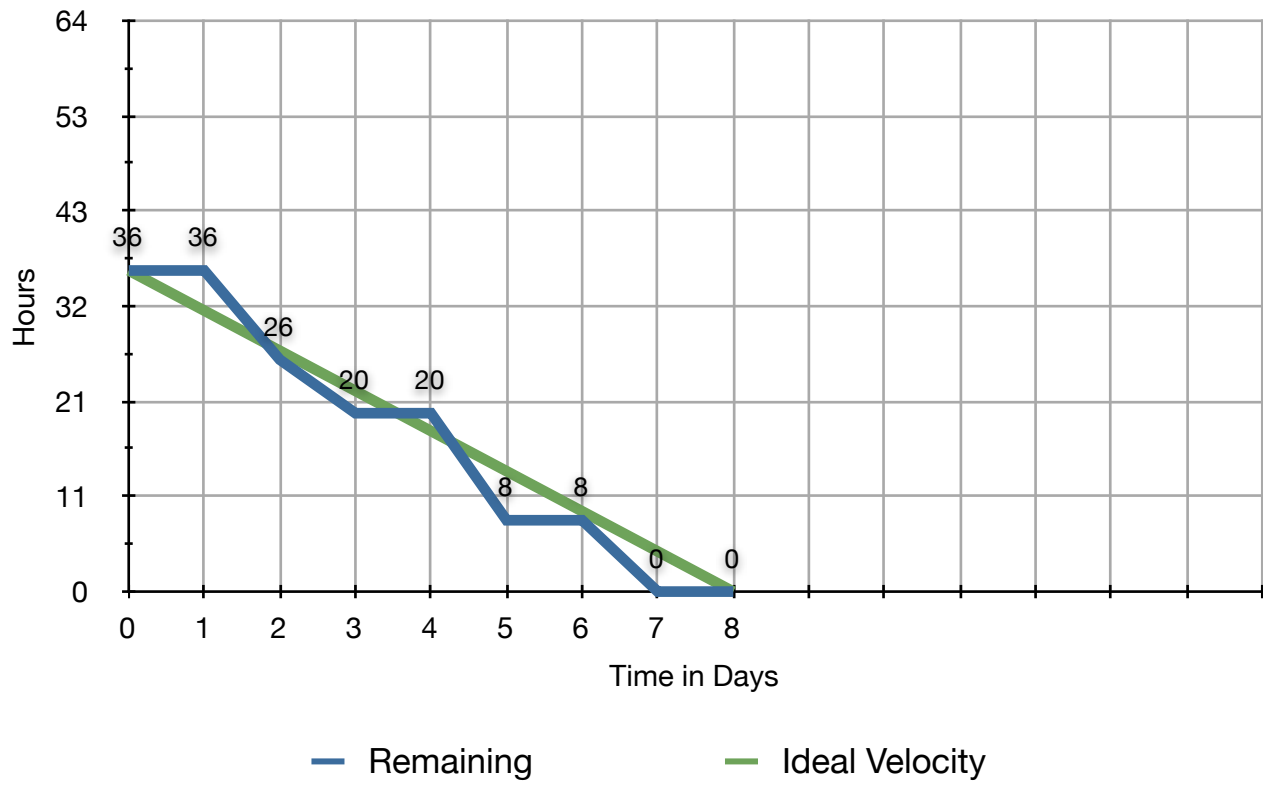
Sprint V

Burn Down Chart



Sprint VI

Burn Down Chart





Evaluatieformulier afstuderen

In te vullen door opdrachtgever c.q. bedrijfsmentor(en)

Student: Reinier van der Smeede
Periode: 3 september 2012 – 14 januari 2013
Bedrijf c.q. instelling: Generali verzekeringsgroep / De Nederlanden van Nu
Bedrijfsmentor: Arjen Dijkstra
Plaats: Diemen
Datum: 2 januari 2013

1. Heeft de student zich zelf snel en goed ingewerkt in het bedrijf en de uit te voeren afstudeeropdracht?

Ja, Reinier heeft zich snel en goed ingewerkt en was al snel effectief.

2. Hoe beoordeelt u de communicatieve vaardigheden van de student (in de samenwerking met collega's, in contacten met de opdrachtgever, bij mondelinge presentaties, schriftelijke rapportages)?

Ik vind dat Reinier prima communiceert. Dit geldt zowel in bilateraal overleg met zijn mentor als ook tijdens presentaties aan het gehele team. Bovengemiddeld ten opzichte van de informatica studenten die ik ken!

3. Hoe heeft de student tijdens het uitvoeren van de opdracht gefunctioneerd?

- | | |
|---|--|
| • Qua verantwoordelijkheid | goed / voldoende / matig / onvoldoende |
| • Qua zelfstandigheid | goed / voldoende / matig / onvoldoende |
| • Qua planmatig werken | goed / voldoende / matig / onvoldoende |
| • Qua creativiteit | goed / voldoende / matig / onvoldoende |
| • Qua productiviteit | goed / voldoende / matig / onvoldoende |
| • Qua samenwerken met collega's | goed / voldoende / matig / onvoldoende |
| • Qua draagvlakontwikkeling | goed / voldoende / matig / onvoldoende |
| • Qua inspelen op bedrijfscultuur | goed / voldoende / matig / onvoldoende |
| • Qua rekening houden met de specifieke context van het bedrijf | goed / voldoende / matig / onvoldoende |
| • Qua het op gang brengen van de nodige veranderingen | goed / voldoende / matig / onvoldoende |

4. Hoe beoordeelt u de kennis en kunde van de student in verhouding tot wat u verwacht van een bijna afgestudeerde?

Ik beoordeel de kennis en kunde van de student als “conform verwachting”, waarbij de elementen “communicatie” en “zelfstandigheid/verantwoordelijkheid” er in mijn beleving extra bovenuit springen. Dat heb ik ook wel eens anders meegemaakt!

5. Hoe beoordeelt u de kwaliteit van de opgeleverde (tussen)producten?

Goed.



6. Bent u tevreden over het opgeleverde (eind)product?

- **In hoeverre heeft u gekregen wat is afgesproken?**

Ik heb gekregen wat is afgesproken tijdens de 2-wekelijkse sprint-meetings.

- **In hoeverre voldoet het (eind)product aan uw verwachtingen?**

Als gevolg van volledig veranderende requirements gedurende de beginfase van Reinier zijn afstudeerperiode, is er onvoldoende tijd overgebleven voor een game die volwaardig is om in de App store te kunnen worden opgenomen. Conform verwachting bleek er te weinig tijd.

- **Wat is de bruikbaarheid en onderhoudbaarheid hiervan?**

Gegeven de aandacht voor testen, is mijn vermoeden dat deze goed is. Tijdens de sprint meetings hebben wij hier geregeld over gesproken. Ik heb persoonlijk echter de code niet geïnspecteerd, dus 100% zekerheid kan ik hierover niet geven.

- **Wat gebeurt er met het opgeleverde (eind)product?**

Intentie is om Reinier na zijn afstuderen nog enkele weken in te huren om de game af te ronden en te zorgen dat deze in de App-store wordt opgenomen. Bedoeling is om hier na zijn afstuderen afspraken over te maken; beide partijen hebben de intentie om hier uit te komen.

- **Kunt u direct met het opgeleverde product aan de slag?**

Zie eerdere opmerkingen.



7. Zijn er nog aspecten voor u van belang die nog niet aan de orde zijn geweest?

Reinier is een prettige vent om in het team erbij te hebben. He gets the job done.

Zijn zelfstandigheid is erg prettig en geeft veel vertrouwen.

8. Bent u bereid een volgende keer weer uw medewerking te verlenen aan het beschikbaar stellen van een afstudeerplaats (graag met toelichting)?

Mijn ervaringen zijn positief. Waarschijnlijk gaan wij niet op korte termijn een mobile game ontwikkelen. Of er een afstudeerplek mogelijk is, hangt af van de opdrachten die wij op dat moment beschikbaar hebben.

Op dit moment heeft De Nederlanden van Nu mogelijkheden voor een (afstudeer)stage webdesign en een (afstudeer)stage op gebied van het ontwikkelen van een eenvoudig datawarehouse.