

HOGESCHOOL VAN AMSTERDAM

PCG Quests

*On a journey to discover the possibilities of a system procedurally
generating quests for MMORPG's*

Rosa Corstjens, 500702627

Januari, 2016

2e jaar, Game Development, Informatica

Vak: Scientific Research Skills

Docent: Irene Overtoom

LA: Alexander Mulder

Samenvatting

Dit onderzoek richt zich op de mogelijkheden die alreeds bestaan voor een procedureel content genererend systeem om *quests* te creëren voor *MMORPG's*. *MMORPG's* zijn gelimiteerd in de kwantitatieve en kwalitatieve ervaringen die ze bieden. Het procedureel genereren van *quests* is een potentiële oplossing voor dit probleem. Naast dat er (nog) geen algemene methode bestaat voor het genereren van *quests*, kamt dit vakgebied momenteel met een aantal eigen problemen, die eerst opgelost moeten worden voordat het een beduidende rol kan innemen in de game industrie.

Om een stap te zetten in de richting van het mogelijk maken van procedurele *quests* in *MMORPG's*, analyseert dit onderzoek de bestaande mogelijkheden voor het genereren van *quests* in *MMORPG's*. Hierbij zijn verschillende aspecten van belang, zoals het ontstaan van een *quest* en de representatie ervan. Door deze aspecten vast te stellen, kan de methode en aanpak voor het genereren worden bepaald.

Vervolgens moet deze methode deel uit kunnen maken van het systeem van de game. Hiervoor moet een systeem worden opgezet, wat een managende rol vervult. De verantwoordelijkheden van dit systeem lopen erg uiteen en zijn afhankelijk van de gebruikte methode. Er is getracht een voorstel te doen voor de minimale verantwoordelijkheden, waaronder het onderhouden van een *quest library* en het communiceren van de actuele state van de *game world*.

Ondanks de vele alreeds gepresenteerde methodes en classificaties zijn de moeilijkheden rondom dit onderwerp nog niet opgelost. De grootste moeilijkheid is het stellen van een algemene werkwijze, zodat toekomstige onderzoeken in dezelfde richting kunnen zoeken naar verdere oplossingen. Een aantal andere problemen zijn niet of nauwelijks aangekaart tot dus ver. Hieronder vallen bijvoorbeeld het genereren van dialoog en een online en genetwerkte *MMO* implementatie.

Inhoudsopgave

Samenvatting	2
1. Inleiding	4
2. Methodes	5
3. Terminologie	6
3.1 PCG in het kort.....	6
3.2 Veel voorkomende termen	6
3.3 Quest nader toegelicht	6
4. Voordelen procedurele quests	7
5. Mogelijkheden voor het genereren van quests	8
5.1 Classificaties.....	8
5.2 Ontstaan van quests	8
5.3 Representeren van quests	9
5.4 Genereren van quests.....	10
6. Verantwoordelijkheden van het systeem.....	12
6.1 Te genereren content	12
6.2 Het plannen, monitoren en managen	13
7. Actuele problemen	14
8. Opzet fieldresearch.....	15
Conclusie en discussie.....	18
Literatuurlijst	20
Bijlage 1, Enquête Quests	22

1. Inleiding

Dit onderzoek is gericht op de mogelijkheden die reeds bestaan voor een procedureel content genererend (PCG) systeem om *quests* te creëren voor *MMORPG*'s. Een dergelijk systeem kan zelfstandig aan de hand van voorgeprogrammeerde algoritmes content creëren, die oneindig veel combinaties kent. Van al deze mogelijke content wordt in dit onderzoek specifiek gekeken naar het genereren van *quests*. Het genereren van *quests* voor *MMORPG*'s biedt interessante mogelijkheden voor actuele problemen, zoals het te kort aan content, de hoge kostenposten en meer.

De hoofdvraag van dit onderzoek luidt daarom:

Wat zijn de bestaande mogelijkheden voor een PCG systeem dat quests voor MMORPG's in een aanhoudende en vooraf gecreëerde game world kan genereren?

Met deze hoofdvraag wordt getracht te achterhalen wat bekend is over het genereren van *quests* en hoe ver gevorderd de technieken zijn. Daardoor kunnen de problemen, die het implementeren van procedurele *quests* in games momenteel onmogelijk maken, ontdekt worden. In verdere onderzoeken kan bekeken worden welk van deze methodes geschikt is om in gebruik te nemen en hoe de problemen overwonnen kunnen worden.

Lezers die niet bekend zijn met de term PCG worden doorverwezen naar hoofdstuk 3.1 'PCG in het kort' waarin een algemene uitleg wordt gegeven over PCG.

De specifieke eis om *quests* te genereren voor *MMORPG*'s geeft een aantal vereisten waaraan de game moet voldoen en waarmee rekening dient te worden gehouden tijdens het realiseren van het PCG systeem. De *game world* met (een deel van) de content is vooraf gecreëerd. Hierin bestaan locaties, gebouwen, wezens en relaties, zoals in iedere andere *game world* van een *MMORPG*. Tevens is van belang om deze *game world* aanhoudend te houden. Dat wil zeggen dat grote wijzigingen in de *game world*, die bijvoorbeeld voortkomen uit een gegenereerde *quest*, niet kunnen bestaan.

Verschillende deelvragen kunnen onderscheiden worden:

- Wat is het voordeel van het procedureel genereren van *quests*?
- Wat zijn de mogelijkheden voor het procedureel genereren van *quests*?
- Wat zijn de verantwoordelijkheden die het systeem zou moeten hebben?
- Wat zijn de centrale problemen die momenteel spelen rondom het procedureel genereren van *quests*?

Het onderzoek is opgedeeld in een aantal delen die gericht zijn op het behandelen van de deelvragen. In hoofdstuk 2 'Methodes' worden de gebruikte onderzoeksmethodes toegelicht. In hoofdstuk 3 'Terminologie' wordt een introductie gegeven op de term PCG, om een duidelijk beeld te schetsen, het onderwerp af te bakenen en om de onervaren lezer in te leiden. Tevens worden een aantal veel gebruikte termen kort toegelicht en wordt dieper ingegaan op de term *quest*. In hoofdstuk 4 'Voordelen procedurele *quests*' wordt ingegaan op de voordelen die procedurele *quests* kunnen bieden en welke problemen rondom *MMORPG*'s ermee opgelost kunnen worden. Nadat deze zaken vastgesteld zijn, gaat hoofdstuk 5 'Mogelijkheden voor het genereren van *quests*' in op opgesteld classificaties en methodes om de *quests* te kunnen genereren. Daarna bespreekt hoofdstuk 6 'Verantwoordelijkheden van het systeem' eisen aan en verantwoordelijkheden van het systeem dat de integratie van procedurele *quests*

in games mogelijk maakt. In hoofdstuk 7 ‘Actuele problemen’ worden de moeilijkheden rondom het procedureel genereren van *quests* aan het licht gebracht. In hoofdstuk 8 ‘Opzet fieldresearch’ is de opzet voor een fieldresearch gemaakt waarin getracht wordt een oplossing te vinden op het probleem van het stellen van een algemene werkwijze en richting, verder besproken in hoofdstuk 7. Het onderzoek sluit af met ‘Conclusie en discussie’ en de literatuurlijst.

2. Methodes

Voor dit onderzoek is gebruik gemaakt van een literatuuronderzoek. Het onderzoek bestaat uit een literatuurstudie naar bestaande wetenschappelijke artikelen. Daarnaast is een opzet gemaakt voor het houden van een enquête. In dit hoofdstuk wordt beschreven hoe verschillende bronnen zijn verkregen, hoe de informatie is verwerkt en hoe een opzet is gemaakt voor het houden van een enquête.

In eerste instantie is gezocht via Google Scholar. In deze zoekmachine zijn termen gerelateerd aan ‘*procedural world generation*’ en ‘*procedural quest generation*’ ingevoerd. Andere termen waarop gezocht is, zijn nauw verbonden met deze termen, maar anders geformuleerd, zoals ‘*PCG quests*’ of ‘*dynamic storylines*’, om zo andere bronnen naar boven te halen. Aangezien later in het onderzoek de focus is verplaatst naar enkel *quest generation*, wordt verder geen informatie gegeven over het zoekproces voor *world generation*. Regelmatig was het niet mogelijk een specifieke bron te verkrijgen via Google Scholar en is de titel van de bron ingevoerd in de AMC databank via de databanken catalogus van de Hogeschool van Amsterdam, waardoor de bron toch toegankelijk was. In tweede instantie is contact opgenomen met Anders Bouwer, docent aan de Hogeschool van Amsterdam, die de bron ‘*A Prototype Quest Generator Based on a Structural Analysis of Quests from Four MMORPGs*’ aanbod. Tot slot is de waardevolste methode voor het vinden van bronnen het gebruik van bronnenlijsten van de alreeds gevonden bronnen geweest. Vooral de bronnenlijsten van de artikelen van Doran en Parberry waren bruikbaar.

Uit alle gevonden bronnen is een selectie gemaakt van een aantal bronnen, die aansluiten op de probleemstelling van dit onderzoek. Deze artikelen gaan o.a. in op de problemen die er spelen in actuele *MMORPG*’s, methodes om *quests* mee te genereren en classificaties van *quests*. Daaropvolgend zijn de bronnen gelezen, geordend en samengevat.

Tijdens het lezen en samenvatten van deze collectie artikelen is regelmatig gezocht naar aanvullende kennis over het onderwerp. Deze kennis was vereist om de benodigde basiskennis op te bouwen om de oorspronkelijk geselecteerde bronnen goed te kunnen begrijpen. In Google zijn de volgende zoektermen ingevoerd: ‘*Petri net*’, ‘*Planning algorithmes*’, ‘*World of Warcraft*’, ‘*Everquest*’, ‘*interactive storytelling*’ en ‘*singleton*’. Op deze termen is gezocht, omdat tijdens het lezen bleek dat de benodigde basiskennis niet toereikend was. Hiermee zijn verschillende artikelen, sites en boeken gevonden. Na het verzamelen van voldoende informatie is de informatie doorgelezen. De verbeterde basis kennis heeft geholpen de oorspronkelijke verzameling van bronnen beter te begrijpen en te verwerken.

Voor het fieldresearch is gezocht naar gegevens over de populatie die onderzocht gaat worden. Hiervoor zijn verschillende internet bronnen geraadpleegd, die via Google verkregen zijn. De zoektermen die ingevoerd zijn in de zoekmachine zijn ‘*game industrie Nederland*’,

‘werknemers game industrie’, ‘MMORPG gamers Nederland’ en ‘gamers Nederland’. Het is als erg lastig ervaren om duidelijke en eenduidige getallen over deze gegevens te vinden, omdat er weinig onderzoeken zijn geweest gericht op deze erg specifieke informatie. Daarom zijn, o.b.v. de gevonden getallen, een aantal berekeningen en gissingen gemaakt om tot een uiteindelijke populatie te komen.

3. Terminologie

Dit hoofdstuk bevat een vluchtige introductie over het onderwerp PCG om een duidelijk beeld te schetsen, het onderwerp af te bakenen en om de onervaren lezer in te leiden. Tevens worden een aantal veel gebruikte termen kort toegelicht en wordt dieper ingegaan op de term *quest*.

3.1 PCG in het kort

PCG staat voor procedurele content generatie of in de veelgebruikte Engelse term: *procedural content generation*. Wanneer ieder woord los wordt bekeken, is de betekenis heel logisch. Procedureel wijst op het feit dat het stap voor stap wordt uitgevoerd, oftewel volgens een procedure. Generatie wijst erop dat iets wordt gegenereerd. Content is dat ‘iets’ wat gegenereerd wordt. Hoe de computer deze content moet genereren staat beschreven in algoritmes, maar daarin staat niet precies beschreven wat de computer moet genereren. De algoritmes begeleiden de generatie als het ware door eisen te stellen aan de uiteindelijk content. Dat is de kracht van PCG: de computer kan een oneindig aantal mogelijkheden voorschotelen. Hiervoor gebruikt hij een semi-willekeurig proces en designers of spelers kunnen eventueel het proces beïnvloeden met behulp van parameters [Doull, 2008]. Er zijn uiteraard verschillende methodes en technieken om content te genereren, vaak ontworpen voor een specifiek soort content. Niet al deze technieken zijn optimaal in gebruik, andere zijn nog niet genoeg onderzocht en ontwikkeld.

3.2 Veel voorkomende termen

Een *Massively Multiplayer Online Role-Playing Game (MMORPG)* is een game waarin veel spelers samen spelen in één interactieve en aanhoudende *game world* en waarin de spelers de rol van een karakter aannemen [Pita, 2007]. De *game world* is interactief, omdat de spelers acties kunnen uitvoeren met anderen en met objecten. Hij is aanhoudend, omdat een *MMORPG* gecreëerd is rondom een bepaald design. Het is niet mogelijk om de *game world* sterk te laten veranderen door acties van een speler, omdat door de vele spelers en interacties de game dan niet in tact kan worden gehouden.

Iedere speler in dit soort games wordt vertegenwoordigd door een *character*. Daarnaast zijn er *Non-Player Characters (NPC's)*, *creatures* en *items* in de *game world*. Het onderscheid tussen *characters* en *creatures* wordt gemaakt op basis van het niveau van interactie dat een speler met een ander kan hebben [Doran, 2010].

3.3 Quest nader toegelicht

Een *quest* is [Doran, 2010; Ashmore, 2007] een mechanisme om de speler interactief verhalen te vertellen. Ze moedigen de speler aan om interacties te hebben met de *game world* en *NPC's*. In *MMORPG's* uiten *quests* zich vrijwel altijd al persoonlijk groei en zelden als ruimtelijke expansie, wat veel gebruikelijker is voor *single player RPG's*. De *quests* zijn vaak simpele vormen van verhaal zonder veel diepgang.

Quests bevatten echter, als het geheel van *quests* bekeken wordt, wél diepgang. Volgens Propps analyse van volksverhalen [Propps, 1968] begint een groot avontuur, oftewel *quest*, wanneer een misdaad wordt gepleegd of een gebrek wordt herkend, waarna de held vertrekt op een reis met als hoogtepunt een strijd tegen een soort tegenstander, maar op de reis komen vaak andere avonturen voor. Dit wijst op de vele *subquests* die één *quest* kan bevatten, die onderlinge samenhang vertonen. Verschillende onderzoeken zien de samenhang tussen *quests* en gebruiken dit in hun methodes [Pita, 2007; Doran, 2011; Soares de Lima, 2014].

De drijfveer van *MMORPG's* zijn *quests*. Een *quest* is een missie die de speler toegewezen krijgt waarin hij wordt uitgedaagd om doelen te behalen in ruil voor beloningen [Doran, 2011]. Vanuit de designer gezien is een *quest* een element wat uitdaging en concrete doelen biedt in de game [Tyschen, 2006]. Vanuit de speler gezien is een *quest* een verhalend element wat hem verder in de game brengt door te leren over de *game world* en sterker te worden. Dankzij *quests* worden spelers meegenomen in het verhaal, verkennen ze de wereld en hebben interacties.

4. Voordelen procedurele quests

MMORPG's bieden de speler een wereld waarin hij kan ontdekken en interacties kan uitvoeren. Ze geven de speler het idee dat hij in een andere wereld is, waarin hij op kan gaan in de verhaallijnen en aangegrepen kan worden door gebeurtenissen. Ondanks de rijkdommen die *MMORPG's* lijken te bieden, wordt in vrijwel ieder geval slechts een verzameling aan voorafgaand bedachte en vastgesteld content voorgeschoteld. Dit houdt de speler tegen in het ervaren van een unieke en persoonlijke *gameplay*. *MMORPG's* zijn dus gelimiteerd op het aspect van ervaringen en daarmee kwaliteit [Lee, 2012].

Kijkend naar de herkomst van *MMORPG's* is er een direct verband met *table-top RPG's*, zoals *Dungeons & Dragons*. In *table-top RPG's* zijn de *combat* systemen gebouwd rondom regels en berekeningen – welke simpel te implementeren waren in computer games – en verzorgt de *dungeon master (DM)* het *non-combat* aspect van de game, zoals de verhaallijn en *character* ontwikkelingen [Tosca, 2003]. De vastgestelde ervaringen zouden mede veroorzaakt worden door het ontbreken van de *DM*.

MMORPG's zijn daarnaast ook gelimiteerd op het aspect kwantiteit [Lee, 2012]. Spelers consumeren momenteel snel bergen content en eisen dat deze content divers, interessant en van hoge kwaliteit is. Het is daardoor niet mogelijk om nog veel langer door te gaan met het handmatig blijven creëren van deze content. Het handmatig creëren van content is een kostbare klus.

Met behulp van procedurele *quest* generatie kunnen de bovengenoemde problemen worden aangekaart [Togelius, 2011]. Er kunnen bijvoorbeeld unieke ervaringen gecreëerd worden voor iedere speler, omdat de content niet vastgesteld staat, maar veranderlijk is. Door het gebruik van *seeds* in het generatie proces is het echter nog steeds mogelijk om dezelfde *quests* te reproduceren.

Behalve unieke ervaringen kunnen procedurele *quests* ook een heel ander soort ervaring bieden dan de *MMORPG's* van nu. *Quests* die niet *combat* gerelateerd zijn en dus de *quests* die de speler meenemen in de game zoals de *DM* dat doet, worden als moeilijker ervaren om te produceren en kosten daardoor vaak meer tijd en geld. Een PCG systeem zou hier echter

geen problemen moeten ondervinden, waardoor een grotere variatie in soorten *quests* kan ontstaan [Sullivan, 2009].

Daarnaast wordt het kwantitatieve limiet opgeheven, vanwege de mogelijkheid van PCG om een eindeloos aantal variaties te creëren. Het opheffen van het kwantitatieve limiet zou in de momentele game industrie een grote groei in de kostenpost zijn, maar PCG kan dit zonder een immense groei, zelfs met een waarschijnlijke daling.

5. Mogelijkheden voor het genereren van quests

Quests zijn geschikt om op procedurele wijze te genereren, omdat ze een duidelijke structuur hebben waaruit eisen op kunnen worden gesteld [Doran, 2010]. Er zijn meerdere onderzoeken gedaan naar het onderverdelen van verschillende soorten *quests*. Velen zien terugkomende eigenschappen in *quests*, zoals: een doelstelling, taak en succes voorwaarden. Zo is het mogelijk om a.d.h.v. categorieën eisen op te stellen en *quests* te genereren.

5.1 Classificaties

Er zijn een aantal classificatie studies gedaan die hebben gepoogd een onderverdeling te maken voor verschillende soorten *quests*.

Walker [Walker, 2007] maakt onderscheid tussen *exploration* en *combat quests*, wat een duidelijk onderscheid is, maar te weinig detail heeft om specifieke eisen voor de *quests* op te stellen. Sullivan [Sullivan, 2009] sluit hierbij aan met haar classificatie van *quests* van World of Warcraft. Ze stelt vast dat het grootste deel op vechten gebaseerd is en een kleiner deel op het verzamelen van *items*.

Doran en Parberry hebben een uitgebreidere classificatie opgesteld [Doran, 2010]. Zij onderscheiden acties, motivaties en strategieën, die als kenmerkend naar voren zijn gekomen in hun analyse van 3000 *quests* van *MMORPG's*. De motivaties corresponderen naar één of meerdere mogelijke strategieën, die de basis van de *quest* definiëren. Een strategie bestaat op zijn beurt uit een aantal acties. Met behulp van deze uitgebreide onderverdeling, waartussen vele relaties bestaan, vonden zij een manier om eisen en beperkingen aan *quests* te stellen, die in een kwalitatief goede output resulteren. De wijze, waarop deze output wordt gegenereerd, wordt nader toegelicht in hoofdstuk 5.3 'Genereren van quests'.

5.2 Ontstaan van quests

Voor het genereren van *quests* moet bepaald worden wanneer de *quest* gegenereerd moet worden en waaruit de *quest* ontstaat. Dit resulteert in een moment om de generatie te activeren en de eisen waaraan de *quest* moet voldoen.

Lock-and-key mechanisms kunnen gebruikt worden om een *quest* op te baseren [Ashmore, 2007]. Door deze focus ligt de benadrukking op de speler context en doelen. Dit zou een geschikte methode zijn, indien we *quests* genereren voor *single-player games*, waar de game zich afspeelt rondom de speler. In *MMORPG's* zou het focussen op *NPC* context en doelen beter aansluiten op de al bestaande *game world* en helpen deze *game world* consistent te houden [Doran, 2010].

Focussend op *NPC* context en doelen zijn er nog steeds aan aantal mogelijkheden voor het ontstaan van een *quest*. Doran en Parberry komen hier weer terug met hun classificatie waarin acties, motivaties en strategieën gedefinieerd worden. Zij bespreken de mogelijkheid dat *NPC's* behoeften hebben, die worden beïnvloed door evenementen en interacties, die plaatsvinden in

de *game world*. Wanneer een behoefte niet vervuld is, leidt dit tot één van de negen motivaties waaruit een *quest* kan ontstaan. Deze motivatie staat centraal in het genereren van een *quest*, hij bepaald welke acties relevant zijn om de motivatie te kunnen vervullen en beperkt daarmee ook de relevante mogelijkheden binnen de *quest*. Hoe relevant de content precies wordt, is afhankelijk van de acties die een strategie definiëren en de strategieën die corresponderen naar motivaties. Ook is het van belang dat de behoeften van de *NPC* hiërarchisch geordend zijn, zodat de meest relevante motivatie voortkomt uit de onvervulde behoeften. De classificatie van Doran en Parberry heeft bewezen toepasbaar te zijn op 3000 *quests* van *MMORPG*'s en kan daarom als relevant worden bestempeld. Pita [Pita, 2007] is tevens overtuigt van het belang van de focus op *NPC* context en doelen. Hij bespreekt echter het belang van relaties en herinneringen, omdat ze kunnen bepalen hoe een *NPC* reageert en kunnen worden weergegeven in een model. Dit model kan dynamisch worden aangepast gebaseerd op acties en evenementen in de wereld. Een *quest* ontstaat in dit geval uit het verzoek van een speler aan een *NPC*, waarop de *NPC* door de herinneringen van hem en een deel van die van de speler heen scant en daarop een aantal mogelijke *quests* baseert. Er is ruimte voor het maken van onderscheid tussen het korte en lange termijn geheugen, waardoor de relevantie kan toenemen. De *quest* kan verder beperkt worden om in relevantie toe te nemen door het definiëren van mogelijke acties en toegang bij de generatie tot bepaalde informatie en attributen, zoals *skill levels* van de speler, die bepalen of een speler in staat is de *quest* te voltooien.

5.3 Representeren van quests

Om een algoritme een *quest* te kunnen laten genereren moet de *quest* op een bepaalde manier gerepresenteerd worden, zodat het algoritme a.d.h.v. de representatie nieuwe *quests* kan genereren en bestaande *quests* kan analyseren.

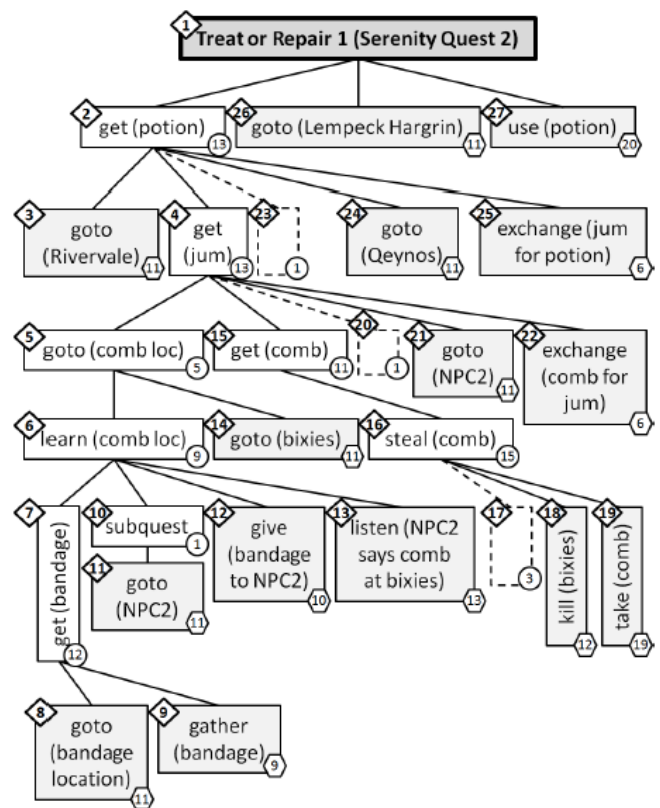
Voor het representeren van *quests* kunnen verschillende datastructuren gebruikt worden. Iedere aanpak komt echter samen op het punt dat een *quest* op zichzelf of meerdere *quests* samengenomen op een manier – op wat voor manier dan ook – hiërarchisch zijn [Soares de Lima, 2014]. Iedere *quest*, kijkend naar *quests* in bestaande *MMORPG*'s, kan worden onderverdeeld in *subquests* en/of acties.

Verschillende onderzoeken maken desondanks geen gebruik van een hiërarchische datastructuur voor het representeren van *quests*. Er kan ook gesteld worden dat de *quest* één of meerdere attributen moet bevatten [Tomai, 2010; Pita, 2007]. Attributen kunnen bijvoorbeeld een aantal doelen, randvoorwaarden waaraan de speler moet voldoen bij het beginnen van de *quest*, een *quest* geveer en nemer, dialoog of hoofdactie zijn. Pita c.s. [Pita, 2007] implementeerden herinneringen in *NPC*'s, waardoor de verzameling van losse *quests* samenhang vertoont door op eerdere gebeurtenissen gebaseerd te zijn.

Ook methodes die gebruik maken van *planning* algoritmes representeren *quests* niet hiërarchisch. De *game world* zoals hij momenteel is, is een bepaalde *state* en ook de *game world* zoals hij is na het afronden van een bepaalde *quest*, is een bepaalde *state*. Een geplande *quest* bevat een begin *state*, een aantal acties die telkens leiden tot een nieuwe *state* van de *game world* en tot slot een eind *state* [LaValle, 2006]. In deze alternatieve visie op *quests* kan gebruik worden gemaakt van een *finite state machine (FSM)*. De *FSM* representeert een opeenvolging van *game world states* en acties om de momentele *state* te veranderen en vormt te samen de complete *quest*.

Een mogelijk betere variatie op het gebruik van een *FSM* is een Petri net. Een Petri net is, net als een *FSM*, een abstracte manier om het verloop van acties weer te geven. Enerzijds is een *FSM* goed in het specificeren van een sequentieel systeem. Anderzijds is een Petri net geschikt voor het specificeren van een asynchroon systeem waarin gelijktijdigheid mogelijk is [Reisig, 2013]. Lee en Cho representeren *quests* als een sequentie van evenementen [Lee, 2012], waarin ieder evenement wordt weergegeven m.b.v. een Petri net. Een event heeft een aantal randvoorwaarden voordat hij gestart kan worden, een eindvoorwaarde, een hoofdactie, die ervoor zorgt dat de eindvoorwaarde behaald wordt, en een aantal sub acties. In *quests* wordt regelmatig van de speler gevraagd om taak A en taak B te voltooien voordat hij aan taak C begint. Het maakt echter niet uit of eerste taak A en dan taak B wordt uitgevoerd of vice versa. In een Petri net is het mogelijk deze acties naast elkaar weer te geven en de voltooiing van beide als randvoorwaarde stellen om aan taak C te beginnen.

Een *tree* heeft van zichzelf al een hiërarchische structuur, die de natuurlijke vorm van een *quest* ondersteunt. Daarom kan deze datastructuur goed gebruikt worden voor het representeren van *quests* [Doran, 2011]. De root van de tree bevat de zogenaamde *quest* strategie, bijvoorbeeld ‘*Obtain Luxuries*’. Een strategie bestaat, zoals al eerder vermeldt, uit een aantal opeenvolgende acties. Zo kan ‘*Obtain Luxuries*’ bestaan uit de acties ‘*get*’, ‘*goto*’ en ‘*give*’. Deze acties worden als *child nodes* van de *root* genoteerd. Iedere actie kan echter op zichzelf ook uit een aantal opeenvolgende acties bestaan, waardoor een uitgebreide *tree* kan ontstaan. Een actie is dus oftewel een atomaire of een recursieve actie. Door de *tree* in *pre-order traversal* te doorlopen, ontstaat een lijst van de *leaves* van de *tree* met alle atomaire acties, die uitgevoerd moeten worden om de *quest* te voltooien. Alle andere *nodes* in de *tree* bevatten acties, die taken zijn, die onderweg uitgevoerd moeten worden. Zie voor een voorbeeld Figuur 1, een *quest* als *tree* weergegeven.



Figuur 1: Analysis of “Cure for Lempeck Hargrin” from Everquest [Doran, 2011]

5.4 Genereren van quests

Het generatie proces van *quests* betreft het verkrijgen van een bepaalde *input*, het analyseren en verwerken van deze *input* en het geven van een *output* in de vorm van een *quest*. Ondanks dat er verschillende mogelijkheden zijn voor dit proces, maakt het merendeel gebruik van klassieke *planning* algoritmes.

AI technieken, zoals *planning* algoritmes, zijn de robuuste basis voor interactieve *quest* systemen [Ghallab, 2004], omdat ze in staat zijn dynamisch een opeenvolging van

evenementen te genereren. Ze nemen als *input* een willekeurige initiële *state* van de *game world*, een willekeurige eind *state* en een verzameling van mogelijke acties [Lee, 2012]. Acties die helpen dichterbij de eind *state* te komen worden net zo lang toegevoegd aan de lijst van acties totdat de eind *state* bereikt is. Het gebruik van klassieke *planning* algoritmes is echter onuitvoerbaar in de praktijk door het gigantische aantal mogelijkheden wat doorzocht moet worden om tot een oplossing te komen, mits er überhaupt een oplossing bestaat. Hierdoor is het niet mogelijk om deze techniek online, tijdens *run time* van de game, te gebruiken [Doran, 2011].

Indien de bovenstaande techniek wordt gecombineerd met het gebruik van een Petri net, is de techniek wel geschikt voor toepassing. Lee en Cho [Lee, 2012] doen een voorstel voor een *quest generation engine* die gebruik maakt van Petri net *planning*. Hiermee genieten ze van een aantal van de eerder genoemde voordelen van een Petri net boven een FSM. Daarnaast worden ook een aantal van de problemen met klassieke *planning* algoritmes vermeden. Ten eerste kan een Petri net situaties analyseren met het gebruik van matrixvergelijkingen. A.d.h.v. deze analyse kan bepaald worden of de eind *state* op een mogelijke manier bereikbaar is vanuit de initiële *state*. Indien dit niet het geval is, stopt het algoritme met een mogelijkheid vinden, dit heeft immers geen zin. Ten tweede kan niet alleen bekeken worden of de eind *state* bereikt kan worden, maar ook naar de mate waarin hij bereikt kan worden. Hierop gebaseerd kan een keuze gemaakt worden of de acties die naar de eind *state* toe leiden niet te omslachtig en irrelevant zijn.

Een ander voorstel om de problemen van klassieke *planning* algoritmes te omzeilen is gedaan door Soares de Lima de Lima c.s. [Soares de Lima, 2014]. Zij presenteren een methode voor het genereren van dynamische *quests* gebaseerd op *planning* algoritmes met non-determinisme. Hiermee ondersteunen ze de onvoorspelbaarheid van het verloop van *quests* en interactieve en dynamische verhalen. I.p.v. analyseren of de eind *state* haalbaar is, houden zij de mogelijkheid open om de *quest* opnieuw te plannen en bij te stellen door stapsgewijs het plan uit te breiden.

In een *quest library* zijn vooraf *quests*, met een begin *state* en één of meerdere eind *states*, gedefinieerd, er is daarom voorafgaande interactie met een developer en/of designer vereist. Tijdens *run time* is het systeem in staat om a.d.h.v. evenementen en interacties binnen de game de oorspronkelijke *quests* bij te stellen. Een *quest* uit de *library* wordt gerepresenteerd als een *tree* met *subquests*. Ieder van deze *subquests* kan een *primitive* of *compound operator* zijn. Een *compound operator* kan de rest van de *quest* beïnvloeden en is daarmee een non-deterministisch element in het systeem. Iedere *operator* heeft effecten, maar bij een *compound operator* kunnen deze effecten anders uitpakken tijdens uitvoering van de *quest*. Om eerste instantie de *quest* te kunnen genereren, wordt ook voor *compound operators* uitgegaan van een vastgesteld, deterministisch effect van de *operator* in kwestie. Tijdens het uitvoeren van deze *operator* kan echter blijken dat het effect anders uitpakt dan in eerste instantie voorspelt was, waardoor een latere *subquest* niet meer relevant of mogelijk is. Dit vereist het herzien en bijstellen van het *quest* plan. Om dit mogelijk te maken, heeft iedere *subquest* een instantie van de *quest planner*, verantwoordelijk voor het herzien van een *quest* plan indien nodig, en *quest* monitor, verantwoordelijk voor het monitoren van de actuele *state* van de *game world* en in hoeverre de *quest* hierin nog mogelijk is. In hoofdstuk 6 ‘Verantwoordelijkheden van het systeem’ wordt kort ingegaan op de wijze waarmee de *quests* opnieuw gepland en gemonitord worden.

Het is uiteraard ook een mogelijkheid om de klassieke *planning* algoritmes in het algemeen achterwegen te laten en een andere techniek te benutten. Doran en Parberry stellen een generator voor die nagaat of er in de game world momenteel een *NPC* is, die een on vervulde behoefte heeft [Doran, 2011]. Hier begint het generatie proces, zoals al eerder besproken. Er wordt één van de vooraf gedefinieerde motivaties geselecteerd om de behoefte te vervullen. Vervolgens wordt a.d.h.v. deze motivatie een corresponderende strategie gekozen. Deze strategie wordt geïntantieerd als de *root* van de *tree* met als *children* een aantal acties. A.d.h.v. bijvoorbeeld het *skill level* van de speler kan een moeilijkheidsgraad worden gedefinieerd met een aantal punten. Dit aantal punten wordt willekeurig verspreid over de *tree*. Herhaaldelijk worden acties van de *tree* vervangen door *subquests*, bestaande uit een soortgelijke set van acties. Zo kan de acties *<get>* bestaan uit *<goto>* *<get>*. De regels waarop het herhaaldelijk vervangen van acties gebaseerd is, zijn vastgesteld na een uitgebreide analyse van *quests* in *MMORPG*'s en kan mogelijk uitgebreid worden. De overgebleven punten worden willekeurig verspreid over de nieuwe acties van de *subquests*. Dit proces herhaalt zichzelf totdat er geen punten meer over zijn. Het resultaat van dit proces is de abstracte *quest*.

De abstracte *quest* wordt omgezet in een concrete *quest* door het toewijzen van *assets*, zoals *characters* en *items*, het genereren van dialoog en het creëren van benodigde *assess*s.

6. Verantwoordelijkheden van het systeem

In voorgaande onderzoeken zijn verschillende systemen gepresenteerd, die meer of minder gemeen hebben met elkaar. Het is daardoor lastig om de eisen vast te stellen waaraan het systeem moet voldoen. Desondanks kan gesteld worden dat iedere methodiek een *quest manager* vereist, om zo een interface te bieden aan de game, bestaande *quests*, entiteiten in de *game world*, etc. Ook kan gesteld worden dat het systeem in staat is autonoom *quests* te genereren [Doran, 2011], sinds niet gezocht wordt naar een systeem als *design tool*. Hiervoor geldt dat het systeem zonder menselijk interactie kan besluiten wanneer een *quest* gegenereerd kan worden en hoe hij moet worden gegenereerd. Tot slot kan gesteld worden dat het systeem een onderliggende, drijvende kracht moet zijn, die de *quests* en interacties verwerkt [Pita, 2007].

In dit hoofdstuk wordt gepoogd een aantal van de belangrijkste eisen scherp te stellen en mogelijke eisen worden besproken. Enkele van deze eisen zijn voorstellen waarvan niet bewezen is dat ze correct zijn en waarmee niet geëxperimenteerd is.

6.1 Te genereren content

Van alle content die zich bevindt in de *game world*, wordt slechts een deel gegenereerd door het *quest* systeem. De content waarvoor het systeem in ieder geval verantwoordelijk is, zijn de *quests*.

Daarnaast bestaat de mogelijkheid om het systeem verantwoordelijk te maken voor het genereren van benodigde *assets* en de dialoog voor een gegenereerde *quest*. Eventueel gaat het systeem alleen na welke content vereist is en delegeert de taak daarna naar een ander systeem wat onderdeel uitmaakt van de game. Het is ook een mogelijkheid om het systeem bij het genereren van *quests* te beperken tot *assess*s die zich op dat moment in de *game world* bevinden.

Tot slot is het gebruikelijk dat spelers een assistent hebben bij het doorlopen van *quests* [Soares de Lima, 2014]. Deze assistent is meestal in tekstuele vorm gepresenteerd en legt de speler zijn actuele doelen voor binnen de *quests* waar hij mee bezig is.

6.2 Het plannen, monitoren en managen

Zoals in de inleiding van dit hoofdstuk al genoemd is, is het vereist om een *manager* voor de game te hebben. Deze *manager* functioneert als interface tussen alle componenten van het systeem en is daarmee de vereiste drijvende kracht.

Een component van het systeem moet gewijd zijn aan het genereren van de *quests*. Zijn taak betreft het plannen. De vormgeving van dit component is divers en hangt samen met de algemene aanpak voor het genereren. Voor het genereren is op zijn minst vereist dat een algoritme gebruikt kan worden om een *quest* te genereren – hoe dit algoritme dan ook te werk gaat – en een controle voor de behoefte aan nieuwe *quests*. Om dit laatste te controleren kan bijvoorbeeld naar de behoefte van *NPC*'s worden gekeken of simpelweg naar het moment dat een speler in een conversatie met een *NPC* om een *quest* vraagt.

Sommige systemen vereisen dat *quests* bijgesteld moeten kunnen worden tijdens de uitvoering ervan om de *quests* dynamisch te maken. Aangezien in een *MMORPG* gigantische aantallen spelers zich tegelijkertijd bezig houden met *quests*, is het niet mogelijk één centrale planner alle actieve *quests* bij te laten sturen indien nodig. Daarom zou iedere *quest* een eigen instantie kunnen hebben van een *planner* [Soares de Lima, 2014]. Deze *planner* is in staat het verloop van de *quest* te herzien. Om dit mogelijk te maken moet de actuele *state* van de *game world* worden bijgehouden en moet de *planner* ingelicht worden indien een *quest* herzien moet worden. Deze verantwoordelijkheden liggen echter niet bij de *planner* en worden later in dit hoofdstuk besproken.

Eén van de taken binnen het monitoren, die uitgevoerd moet worden is het bijhouden van de actuele *state* van de *game world*. Het is vereist dat de *game world* wordt gerepresenteerd op een wijze dat andere entiteiten er informatie van kunnen verwerken. Een goede optie voor deze representatie maakt gebruik van proposities, die alle *characters*, *items*, locaties en situaties beschrijven [Soares de Lima, 2014].

Een centrale *manager* is geschikt om deze taak uit te voeren, omdat hij, als interface zijnde tussen alle andere componenten, deze informatie kan communiceren naar de verschillende entiteiten. O.b.v. deze actuele *state* van de *game world* kunnen nieuwe *quests* gegenereerd worden en kunnen entiteiten hun informatie en kennis aanpassen.

Een andere taak binnen het monitoren betreft het bijhouden van het verloop van actieve *quests*. Hieronder kan bijvoorbeeld vallen dat er gecontroleerd moet worden of het verdere verloop moet worden bijgesteld, of de assistent van de speler geüpdatet moet worden en of er eventueel items geïnitieerd moeten worden. Het monitoren van alle *quests* ondervindt echter dezelfde problemen als het plannen van aanpassingen in bestaande *quests*. Door de grote aantallen spelers en daarmee de vele actieve *quests*, is het ook hier geen overbodige luxe om een monitor voor iedere individuele *quest* te instantiëren [Soares de Lima, 2014].

De laatste taak, die zich bezig houdt met monitoren, is een optionele taak. Indien de manier van genereren resulteert in *quests* met gangbare acties, acties die herhaald moeten worden en acties die optioneel zijn, kan het gewenst zijn om bijvoorbeeld genummerde *triggers*, die normaal, herhaalbaar of optioneel zijn, te koppelen aan acties van de *quest* en een, boven de

quest staande, *manager* de verantwoordelijkheid geven om deze *triggers* te kunnen vuren [Pita, 2007]. Nadat de *trigger* het voorafgaand gedefinieerde aantal maal heeft gevuurd, kan de *quest* worden voortgezet.

Voor de management taken is opnieuw de *quest manager* een geschikte kandidaat. Onder het managen wordt verstaan dat de informatie, die tijdens het monitoren verzameld is, wordt gecommuniceerd naar entiteiten in de wereld, die behoefte hebben aan deze informatie. Daarom moeten deze entiteiten worden bijgehouden. De kennis over welke entiteiten wel en niet bijgehouden moeten worden, kan bij de entiteiten liggen of bij de *manager* zelf, dit kan verschillen per implementatie. Het meest flexibel is een implementatie waarbij de entiteiten zelf weten welke informatie ze hoe en wanneer moeten krijgen.

Naast het communiceren van informatie kunnen onder het managen nog een aantal taken vallen. Lokale modificaties in de *game world* kunnen bijvoorbeeld moeten worden bijgehouden, waarmee *selective visibility* mogelijk wordt gemaakt [Doran, 2015]. Met *selective visibility* wordt bedoeld dat alleen de speler of spelers, die verbonden zijn met een specifieke *quest*, deze specifieke content kunnen zien en dat deze content wordt verwijderd wanneer dit gedeelte van de *quest* is afgerond.

Andere voorbeelden van mogelijke taken zijn het bijhouden van herinneringen van *NPC's*, het bewaken van gebruikte *processing* kracht, het beperken van *memory usage* en het *spawnen* van *items* etc. Al deze taken zijn echter alleen vereist voor bepaalde methodes, voor andere zijn ze irrelevant. Daarom is het lastig vast te stellen wat de complete set is van taken die worden uitgevoerd door een managend component van het systeem.

Het systeem moet tevens gebruik kunnen maken van een database [Soares de Lima, 2014]. Deze database is de complete verzameling van *quests*, die zich momenteel in het wereld bevinden. Het is van belang dat de *quest manager*, of een ander component die verantwoordelijk gesteld wordt, zonder complicaties nieuwe *quests* kan toevoegen en bestaande *quests* kan verwijderen uit de database. De informatie, die opgeslagen moet worden in de database, is sterk afhankelijk van de specifieke representatie van *quests*. Attributen die naar waarschijnlijkheid gevonden zullen worden in deze database zijn een *quest* titel, betrokken *NPC's* – al dan niet met een gedefinieerde rol –, benodigde *items* en start en eind criteria.

7. Actuele problemen

Uit de voorgaande hoofdstukken is gebleken dat er momenteel al een heel aantal methodes zijn waarmee een aantal problemen van dit vakgebied worden geadresseerd. Desondanks is er voor een deel van deze problemen nog niet een algemene aanpak gedefinieerd en zijn de gebruikte methodes erg ongelijksoortig. Daarnaast zijn een groot deel van de moeilijkheden überhaupt nog niet geëntameerd. Dit hoofdstuk bespreekt eerst de problemen waaraan al aandacht is besteed, maar waarvoor nog verder onderzoek vereist is, en daarna de problemen die in het geheel nog niet of nauwelijks onderzocht zijn.

Voor iedere reeds voorgestelde methodiek voor het genereren van de *quests* zelf uit de verschillende gelezen onderzoeken geldt dat het goede elementen bezit en een aantal problemen open laat voor latere invulling. Vrijwel ieder onderzoek is echter met een andere methodiek gekomen met erg diverse aspecten. Daarom is het belangrijkste probleem voor het

genereren van *quests* het stellen van een algemene werkwijze, zodat toekomstige onderzoeken in dezelfde richting kunnen zoeken naar verdere oplossingen. In een poging om een ‘beste’ methode aan het licht te brengen, is een voorstel geschreven voor het houden van een enquête. Zie hiervoor hoofdstuk 8 ‘Opzet fieldresearch’.

Voordat het daadwerkelijk integreren en gebruiken van *quests* in bestaande MMORPG’s mogelijk is, moeten eerst een aantal van belang zijnde aspecten opgelost worden. Ten eerste zijn de technieken voor het procedureel genereren van aansluitende dialogen nog niet genoeg ontwikkeld [Doran, 2014; Tomai, 2012].

Ten tweede zijn er verschillende onderzoeken van mening dat het van belang is om een vorm van een traditionele spanningsboog te implementeren [Doran, 2014]. Dit is echter niet vereist voor *quests* in MMORPG’s en het is nog de vraag of dit noodzakelijk is. De meeste *quests* zijn simpelweg een opeenvolging van acties, die naar een centrale actie opbouwen en eventueel een aantal subacties. Indien uit later onderzoek blijkt dat er toch behoefte is vanuit spelers aan een vorm van een spanningsboog, zal dit probleem opnieuw bekeken moeten worden.

Ten derde is pas een eerste stap gezet naar een online en genetwerkte MMO implementatie door Doran en Parberry [Doran, 2014]. Aangezien de meeste methodes nog niet toe zijn aan deze implementatie is er momenteel nog weinig aandacht aan besteed, maar andere onderzoeken zien ook het belang hiervan [Pita, 2007].

8. Opzet fieldresearch

Om verdere antwoorden te vinden op de mogelijkheden voor het procedureel genereren van *quests* is deze opzet voor een fieldresearch geschreven. Met behulp van internet enquêtes wordt getracht een overzicht te krijgen van de meningen van professionals uit de game industrie en MMORPG spelers. De gezochte meningen beschrijven de eisen en kenmerken van *quests*, die de respondenten belangrijk vinden. Dit aspect is belangrijk voor het genereren van *quests*, omdat het vaststelt hoe de *quests* gegenereerd moeten worden en aan de hand van welke aspecten. Op het resulterende overzicht kan vervolgens een analyse worden uitgevoerd om te bepalen welke meningen door de meeste respondenten gedeeld worden en wellicht kan gesteld worden wat de meest geschikte methode is. Door een consensus te verkrijgen op dit onderwerp wordt een grote stap naar de oplossing gezet.

De keuze voor het houden van een enquête is genomen, omdat tijdens het deskresearch van dit onderzoek sterk is gebleven dat er opmerkelijk veel meningen zijn over de eisen en kenmerken van *quests*. Door deze grote verscheidenheid in meningen is het vereist om van een grote groep de mening te meten om zo te bestuderen wat de meest voorkomende meningen zijn. Een enquête maakt het mogelijk om een groot aantal mensen te ondervragen. Tevens bevordert een enquête de mogelijkheid om de vergaarde informatie achteraf te generaliseren en te analyseren.

De groep respondenten zal uit ongeveer 250 professionals uit de game industrie en ongeveer 400 MMORPG spelers bestaan [Checkmarket, z.d.]. Er is gekozen voor deze twee verschillende groepen, omdat de ene groep, de professionals, een goed beeld heeft van de mogelijkheden, terwijl de andere groep, de spelers, een beter beeld hebben van de

daadwerkelijke ervaring. Voor het uitrekenen van de benodigde aantallen respondenten is de volgende rekensom gemaakt:

Er wordt uitgegaan van enkel Nederlandse respondenten, omdat naar verwachting de mening per land of regio niet of weinig zal afwijken en Nederland op zichzelf al een omvangrijke groep geschikte respondenten bevat. De eerste reden hiervoor is dat het niet het doel is van dit onderzoek om een stelling te kunnen doen over de doelgroep, maar het verzamelen van meningen om een schets te maken van deze meningen op grotere schaal. De tweede reden is dat het betreffende onderwerp en de populatie allebei vooral op globaal niveau handelen en niet op landelijk niveau.

Voor het berekenen van deze getallen zijn een aantal gissingen gemaakt, zoals het aantal designers en developers, o.b.v. persoonlijke kennis. Deze gissingen zijn gemaakt vanwege gebrek aan cijfers.

Ruim 2.000 mensen zijn werkzaam in de game industrie in Nederland. Van alle bedrijven die zich in de game industrie bevinden, houdt 60% zich bezig met het ontwikkelen van games [Oosteren, 2010]. Ervan uitgaande dat ongeveer 50% van de overgebleven 1230 werknemers een functie heeft als developer of designer, resteert een totaal van 615 werknemers als relevante populatie voor de gezochte professionals voor de steekproef.

Ongeveer 28% van de Nederlanders speelt wekelijks games. De andere 72% is niet relevant voor dit onderzoek, omdat, zeker voor *MMORPG's*, spelers die vaker spelen geëngageerde spelers zijn en daarom meer inzicht hebben in het spel. Van deze 28% speelt 11% *multiplayer online games* [Hendrixen, 2013]. Er zijn geen precieze cijfers gevonden over het aantal *MMORPG* spelers in Nederland, daarom wordt uitgegaan van deze 11%. 11% van 28% van de Nederlanders resulteert in een totaal van 523.600 *MMO* spelers.

Met het gebruik van de bron Checkmarket [Checkmarket, z.d.] is vervolgens berekend hoeveel respondenten de enquête moeten invullen om de uiteindelijke resultaten te kunnen globaliseren.

De werving van deze respondenten kan op meerdere manieren verlopen. Het voorstel wat hier gedaan wordt, betreft een mogelijke wervingsmethode voor beide groepen en een aantal alternatieven indien er met de primaire methode niet voldoende respondenten zijn geworven.

Ten eerste wordt contact opgenomen met Dutch Game Garden, waar meer dan 40 indie game bedrijven gevestigd zijn. Doordat de bedrijven relatief klein zijn, wordt er ook tussen bedrijven intensief contact gehouden en zal met het gebruik van hun netwerk naar waarschijnlijkheid een groep van 250 professionals kunnen worden geworven.

Ten tweede wordt op forums van bekende *MMORPG's* een bericht achtergelaten met de vraag of spelers mee willen doen aan dit onderzoek. Door de grote aantallen actieve spelers kan naar verwachting zonder enig probleem op deze wijze het aantal respondenten worden geworven. Indien dit bericht in het Nederlands wordt geplaatst, zullen enkel Nederlands sprekende respondenten reageren. De voorgestelde *MMORPG's* zijn *World of Warcraft*, *Elder Scrolls Online* en *Runescape*.

Indien deze methodes niet toereikend zijn, wordt het volgende voorgesteld. Voor de werving van professionals uit de game industrie kan gebruik worden gemaakt van LinkedIn, Facebook, netwerken van personen en forums voor game development. Voor de werving van *MMORPG* gamers kan op forums van andere *MMORPG's* berichten achtergelaten worden. De

voorgestelde *MMORPG*'s kunnen dan worden aangevuld met Star Trek Online, Skyforge, Star Wars: The Old Republic, Guild Wars 2 en Rift.

Na de werving van de respondenten wordt de reeds opgestelde enquête naar de geselecteerde respondenten gestuurd. Voor het opstellen van de enquête is gebruik gemaakt van Google Forms. De vragen van de enquête betreffen onderwerpen besproken in hoofdstuk 5 'Mogelijkheden voor het genereren van quests'. De enquête kan gevonden worden met de volgende link: <http://goo.gl/forms/szKo0z0hsI> en is tevens opgenomen in de bijlage. Google Forms ondersteunt de generalisering van de verzamelde informatie met de mogelijkheid de reacties te exporteren als Excel bestand. In Excel kunnen vervolgens met het gebruik van de optie *conditional formatting* ordeningen worden gemaakt op volgorde van meest gegeven antwoorden of kunnen kleurcodes worden aangebracht. De geordende informatie moet worden geanalyseerd en er moet een top 3 meest gegeven antwoorden per vraag worden opgesteld. Voor alle vragen waar de optie 'anders ...' is ingevuld, moeten de antwoorden worden geanalyseerd en gecategoriseerd. Indien bepaalde antwoorden opmerkelijk vaak naar voren komen, moet in overweging worden genomen om deze antwoorden mee te nemen in de analyse van de oorspronkelijk opgestelde antwoorden. Het uiteindelijke meest gegeven antwoord is een aanbeveling voor verder onderzoek op deze specifieke aspecten.

Naar verwachting zullen de meest gekozen antwoorden aansluiten op de methode van Doran en Parberry, omdat hun techniek natuurlijk en ongedwongen over komt, en dat de antwoorden laten blijken dat procedurele *quests* een waardevolle toevoeging zijn voor *MMORPG*'s volgens de respondenten.

De respondenten zullen waarschijnlijk *quests* als leuker ervaren wanneer de speler invloed heeft op het verloop. Spelers van games genieten van de illusie van vrijheid in games en zullen van daadwerkelijke vrijheid en invloed meer genieten.

Of *quests* wel of niet een spanningsboog bevatten zal waarschijnlijk verschillend zijn per respondent. Dit zal ook afhankelijk zijn van de games die de respondent kent en heeft gespeeld, aangezien dit aspect meer of minder terugkomt in verschillende games. Wel kan op basis van deze antwoorden gesteld worden of het belangrijk is om onderzoek te focussen op het verwerken van een spanningsboog in de structuur van *quests*.

Vermoedelijk zullen respondenten een hiërarchie terug zien in *quests* en in het geheel van *quests*, omdat games vaak de verbanden hiertussen laten zien. Daarmee zullen ze ook van mening zijn dat *subquests* een belangrijke bijdrage leveren.

Zoals eerder gezegd zal waarschijnlijk de methode van Doran en Parberry als meest gekozen naar voren komen, wat inhoudt dat de respondenten veelal zullen antwoorden dat het begin van een *quest* verbonden is met de behoefte van een *NPC* en dat *quests* onderverdeeld kunnen worden in verschillende categorieën op basis van de acties die de speler moet uitvoeren, ook al kan het doel van de *quest* hier waarschijnlijk ook naar voren komen. Ondanks dat de methode van Doran en Parberry o.a. het level van de speler gebruikt als *input* voor de moeilijkheidsgraad, zullen de respondenten waarschijnlijk meer voelen voor het antwoord dat het afhankelijk is van de *quests* die de speler al heeft voltooid of de beloning, wat gebruikelijker is voor *MMORPG*'s vanwege de vastgestelde content die zich in de game bevindt. Mogelijk zijn deze elementen voor het vaststellen van de moeilijkheidsgraad interessant voor verder onderzoek.

Conclusie en discussie

Dit onderzoek zette een aantal mogelijkheden voor het genereren van *quests* voor *MMORPG*'s tegen elkaar uit om zo het antwoord op de hoofdvraag 'Wat zijn de mogelijkheden voor een PCG systeem dat *quests* voor *MMORPG*'s in een aanhoudende en vooraf gecreëerde game world kan genereren?' te vinden.

De conclusie die kan worden getrokken is dat de potentie van procedurele *quests* duidelijk is: het zou een waardevolle aanwinst zijn voor *MMORPG*'s. Voordat dit echter mogelijk is, is verder onderzoek vereist om de problemen, die voortgekomen zijn uit de laatste deel vraag.

Het gebruik van procedurele *quests* in *MMORPG*'s biedt veel mogelijkheden. *MMORPG*'s van nu kampen met een aantal problemen en de procedurele *quests* hebben de potentie om deze problemen aan te pakken en mogelijk op te lossen. Zo zijn *MMORPG*'s gelimiteerd in de kwantitatieve en kwalitatieve ervaringen die ze bieden, omdat het niet mogelijk is nieuwe ervaringen, of *quests*, eindeloos te kunnen blijven creëren, vanwege hoge kosten van werknemers, en dat deze ervaringen nooit uniek zijn voor iedere speler [Lee, 2012]. Procedureel gegenereerde content is nooit gelimiteerd op kwantiteit, omdat het algoritme altijd nieuwe, originele *quests* kan samenstellen. Hiermee wordt direct het aspect kwaliteit opgelost, door de originaliteit van de nieuw gegenereerde *quests*.

Verschillende onderzoeken bekijken de mogelijkheden om *quests* te genereren. De drie aspecten die hierin een grote rol spelen zijn het ontstaan van een *quest*, het representeren van een *quest* en het daadwerkelijk genereren van een *quest*. De voorgestelde methodes, die in staat zijn zelfstandig nieuwe *quests* te genereren, i.p.v. bestaande *quests* aanpassen, zijn het er veelal over eens dat *quests* ontstaan vanuit een *NPC*. Doran en Parberry [Doran, 2011] stellen het gebruik van motivaties voor *NPC*'s voor, die veranderen o.b.v. interacties en evenementen in de wereld. Pita [Pita, 2007] doet echter een voorstel voor het gebruik van memories. Op het gebied van representatie komende de voorgestelde aanpakken minder overeen met elkaar. Ondanks dat vele het hiërarchische element van *quests* zien, gebruikt niet iedere methode een hiërarchische representatie. Datastructuren als een FSM en een Petri net zijn naar voren gekomen als niet-hiërarchische datastructuren, waarvan het gebruik toch goed verloopt [Soares de Lima, 2014; Lee, 2012; LaValle, 2006]. Indien er gebruik wordt gemaakt van niet-hiërarchische datastructuren, wordt vrijwel altijd op een andere manier een vorm van hiërarchie of samenhang geïmplementeerd. Een goede hiërarchische datastructuur is een *tree* [Doran, 2011]. Om een conclusie te trekken over de meest potentievolle en natuurlijke randvoorwaarden van *quests*, moet het voorstel voor een fieldresearch, hoofdstuk 8 'Opzet fieldresearch' worden uitgevoerd.

AI technieken, zoals *planning* algoritmes, zijn de robuuste basis voor interactieve *quest* systemen [Ghallab, 2004]. Het is echter niet mogelijk om deze techniek online, tijdens *run time* van de game, te gebruiken [Doran, 2011]. Het grootste probleem met *planning* algoritmes is het gigantische aantal mogelijkheden wat doorzocht moet worden om tot een oplossing te komen, mits er überhaupt een oplossing bestaat. Verschillende onderzoeken presenteren methodes, die gebruik maken van *planning* algoritmes, maar de klassieke problemen omzeilen. Cho en Lee combineren de *planning* algoritmes met Petri nets, wat de mogelijkheid geeft situaties te analyseren met matrixvergelijkingen, om te kunnen besluiten of het *planning* probleem mogelijk op te lossen is [Lee, 2012]. Soares de Lima [Soares de Lima, 2014] stelt een methode voor waarbij wordt uitgegaan van het non-determinisme van

quests. I.p.v. analyseren of de eind *state* haalbaar is, houdt hij de mogelijkheid open om de *quest* opnieuw te plannen en bij te stellen door stapsgewijs het plan uit te breiden. Een andere optie is uiteraard het vermijden van *planning* algoritmes in het algemeen. Doran en Parberry [Doran, 2011] maken gebruik van strategieën, die centraal staan voor een *quest*. Iedere strategie bestaat uit een aantal opeenvolgende acties. Dit wordt gezamenlijk weergegeven in een *tree*. Door herhaaldelijk de *leaves* van de *tree*, acties van de strategie, te vervangen met een reeks opeenvolgende acties met hetzelfde resultaat, kan een uitgebreide *tree* gecreëerd worden. De uiteindelijke *tree* geeft de complete *quest* in *pre-order traversal* weer. Deze methode is uitgebreid, is in staat om vele verschillende soorten *quests* te genereren naast *combat quests* en is opgesteld na een analyse van meer dan 3000 *MMORPG quests*. Daarom heeft deze methode veel potentie om verder uitgewerkt te worden voor implementatie. Desondanks kan dit pas vastgesteld worden na uitvoering van het voorstel voor een fieldresearch, hoofdstuk 8 ‘Opzet fieldresearch’.

Het is lastig om de eisen vast te stellen waaraan het systeem moet voldoen door de grote diversiteit aan methodes. De minimale eisen die gesteld kunnen worden, gelden voor de aspecten te genereren content, plannen, monitoren en managen.

De systeem moet een aantal soorten content opleveren. Dit op zijn minst *quests*. Daarnaast kan het systeem ook verantwoordelijk zijn voor het genereren van benodigde *assests*, dialoog en een *player assistant* [Soares de Lima, 2014].

Rondom de taak plannen heeft het systeem de verantwoordelijkheid om de *quests* te genereren en eventueel bij te sturen. De mogelijkheid bestaat om een *planner* per *quest* te hebben, i.p.v. het uitvoeren van deze taak door een centrale instantie [Soares de Lima, 2014].

De verantwoordelijkheden rondom de taak monitoren zijn het bijhouden van de actuele *state* van de *game world* en het bijhouden van het verloop van actieve *quests*. Ook hier bestaat de mogelijkheid om

De management taak betreft de verantwoordelijkheden het verzamelen van informatie en deze door communiceren naar entiteiten in de wereld, zoals de actuele *state* van de *game world*, het onderhouden van lokale modificaties, om *selective visibility* [Doran, 2015] mogelijk te maken en het managen van de gebruikte *processing* kracht en *memory usage*.

De grootste moeilijkheid die moet worden aangepakt, is het stellen van een algemene werkwijze, zodat toekomstige onderzoeken in dezelfde richting kunnen zoeken naar verdere oplossingen. Door de uitvoering van de opzet voor een fieldresearch, hoofdstuk 8 ‘Opzet fieldresearch’ is het wellicht mogelijk een voorstel te doen voor de meest potentiële werkwijze en deze verder te exploreren. Hierna is verder onderzoek vereist op deze gebieden. Een aantal andere problemen zijn niet of nauwelijks aangekaart tot dus ver. Hieronder vallen het genereren van dialoog, het implementeren van een spanningsboog in het verloop van *quests* en een online en genetwerkte *MMO* implementatie.

Literatuurlijst

- Ashmore, C. & Nitsche, M. (2007). *The Quest in a Generated World*. Gedownload op http://homes.lmc.gatech.edu/~nitsche/download/AshmoreNitsche_DiGRA_07.pdf.
- Checkmarket. (z.d.) *Steekproefcalculator*. Geraadpleegd op 20 januari 2015, van <https://nl.checkmarket.com/marktonderzoek-hulpbronnen/steekproefcalculator/>.
- Doran, J. & Parberry, Ian. (2010). *Towards Procedural Quest Generation: A Structural Analysis of RPG Quests*. Gedownload op <http://larc.unt.edu/techreports/LARC-2010-02.pdf>.
- Doran, J. & Parberry, Ian. (2011). *A Prototype Quest Generator Based on a Structural Analysis of Quests from Four MMORPGs*. Gedownload op <https://larc.unt.edu/techreports/LARC-2011-02.pdf>.
- Doran, J. & Parberry, Ian. (2015). *A Server-Side Framework for the Execution of Procedurally Generated Quests in an MMORPG*. Gedownload op <http://larc.unt.edu/techreports/LARC-2015-01.pdf>.
- Doull, A. (2008). *The death of the level designer: procedural content generation in games*. Gedownload op njema.weebly.com/uploads/6/3/4/5/6345478/the_death_of_the_level_designer.pdf.
- Ghallab, M., Nau, D. & Traverso, P. (2004). *Automated Planning: Theory & Practice*. San Francisco: Elsevier.
- Hendrixen, M. (2013). *Interessante feitjes en cijfers over games, gamers en gaming*. Geraadpleegd op 22 januari 2016, van <http://hendrixen.nl/interessante-feitjes-en-cijfers-over-games-gamers-en-gaming/>.
- LaValle, S. (2006). *Planning Algorithms*. Cambridge: Cambridge University Press.
- Lee, Y. & Cho, S. (2012). *Dynamic quest plot generation using Petri net planning*. Gedownload op <http://rps.hva.nl:2413/citation.cfm?id=2425304&CFID=565976299&CFTOKEN=43410985>.
- Oosteren, C. van. (2010). *Game-industrie in beeld*. Gedownload op http://www.ois.amsterdam.nl/pdf/2010_game_industrie_in_beeld.pdf.
- Soares de Lima, E., Feijó, B. & Furtado, A. (2014). *Hierarchical generation of dynamic and nondeterministic quests in games*. Gedownload op <http://rps.hva.nl:2413/citation.cfm?id=2663833&CFID=565976299&CFTOKEN=43410985>.
- Tomai, E., Rosendo, S. & Salinas, D. (2012). *Adaptive Quests for Dynamic World Change in*

- MMORPGs*. Gedownload op <http://rps.hva.nl:2626/10.1145/2290000/2282402/p286-tomai.pdf>.
- Tosca, S. (2003). *The Quest Problem in Computer Games*. Gedownload op <http://www.it-c.dk/people/tosca/quest.htm>, 03-01-2015.
- Tyschen, A., Tosca, S. & Brolund, T. (2006). *Personalizing the Player Experience in MMORPGs*. Gedownload op <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.734&rep=rep1&type=pdf>.
- Pita, J., Magerko, B. & Brodie, S. (2007). *True story: dynamically generated, contextually linked quests in persistent systems*. Gedownload op <http://rps.hva.nl:2413/citation.cfm?id=1328228&CFID=565976299&CFTOKEN=43410985>.
- Propp, V. (1968). *Morphology of the Folktale*. Austin: University of Texas Press.
- Reisig, W. (2013). *Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies*. Berlijn: Springer.
- Sullivan, A. (2009). *Gender-Inclusive Quest Design in Massively Multiplayer Online Role-Playing Games*. Gedownload op <https://games.soe.ucsc.edu/sites/default/files/sullivan-fdg09-dc-cr2.pdf>.
- Togelius, J., Champandard, A. J., Lanzi, P.L., Mateas, M., Paiva, A., Preuss, M. & Stanley, K. O. (2013). *Procedural content generation: goals, challenges and actionable steps*. Gedownload op <http://julian.togelius.com/Togelius2013Procedural.pdf>.
- Walker, J. (2007). *A Network of Quests in World of Warcraft*. Gedownload op <http://jilltxt.net/txt/Walker-for-second-person.pdf>.

Bijlage 1, Enquête Quests

Deze enquête verzamelt informatie over de ideeën die de respondenten hebben over de eisen en kenmerken van *quests* voor *MMORPG's*. Aan de hand van deze informatie wordt getracht een beeld te schetsen van de algemene mening hierover, om zo wellicht een stap te zetten in het mogelijk maken van procedurele *quests* in *MMORPG's*.

Ik ben een ...

- ☐ Game designer
- ☐ Game developer
- ☐ Gamer

Quests zijn leuker als de speler invloed heeft op het verloop.

- ☐ Eens
- ☐ Oneens

Een *quest* heeft een spanningsboog in de verhaallijn.

- ☐ Eens
- ☐ Oneens

Een *quest* op zichzelf heeft een hiërarchische structuur.

- ☐ Eens
- ☐ Oneens
- ☐ Dit is verschillend per *quest*

Alle *quests* uit een game vertonen onderlinge samenhang.

- ☐ Eens
- ☐ Oneens, er is geen sprake van samenhang
- ☐ Oneens, er is alleen sprake van een *main storyline*
- ☐ Dit is verschillend per game

Subquests zijn een belangrijke bijdrage in de verhaallijn van een *quest*.

- ☐ Eens
- ☐ Oneens

Het begin van een *quest* is verbonden met ...

- ☐ De behoefte van een *NPC*
- ☐ Een evenement
- ☐ Een interactie van een speler
- ☐ Anders ...

Quests kunnen onderverdeeld worden in verschillen categorieën op basis van ...

- Acties die de speler moet uitvoeren
- Het doel van de *quest*
- De beloning op het eind van de *quest*
- De lengte van de *quest*
- De moeilijkheidsgraad van de *quest*
- Anders ...

De moeilijkheidsgraad van een *quest* hangt samen met ...

- Het level van de speler
- De *quests* die de speler al heeft voltooid
- De beloning van de *quest*
- De omgeving waar de *quest* start
- Anders ...

Ik vind dat procedurele *quests* een waardevolle toevoeging zijn voor *MMORPG's*.

Procedurele quests zijn quests die tijdens het spelen van de game gemaakt worden door het systeem op basis van bepaalde factoren. Ze bieden een mogelijkheid om eindeloos nieuwe en originele quests te creëren.

- Eens
- Oneens
- Dat is verschillend per game

