

Aan de slag als Virtueel Theatermaker

...wellicht een handvat...

[illegible]

Arnaud Loonstra - DVTG 2005

*Eerst komt het temmen,
dan komt het stemmen,
komt het spelen,
dat mag eeuwig
duren, durén,
duren, durén,
duren, durén,
duren...*

Frédérique Clément

Inhoudsopgave

1	Inleiding.....	4
1.1	<i>Virtueel Theatermaker.....</i>	4
1.2	<i>Aannames.....</i>	4
2	Professie.....	6
2.1	<i>Technische Diepgang.....</i>	6
2.2	<i>Het prutsen voorbij.....</i>	7
2.3	<i>Niet vis hebben maar kunnen vissen.....</i>	8
2.4	<i>De kunstenaar en de programmeur.....</i>	9
2.5	<i>Bedenkingen.....</i>	10
2.6	<i>Kennis gebrek bij opdrachtgevers.....</i>	11
3	De trukendoos.....	14
3.1	<i>Algoritmes.....</i>	14
3.2	<i>Herhaling.....</i>	15
3.3	<i>Knijp eens met je ogen.....</i>	16
3.4	<i>Niet alles hoeft je te zien.....</i>	17
4	Bouwen.....	18
4.1	<i>Iteratieve bouw.....</i>	18
4.2	<i>Incrementele bouw.....</i>	18
4.3	<i>Uitbesteden.....</i>	19
4.4	<i>8 gouden regels van UI design.....</i>	19
5	Project Management.....	21
5.1	<i>Hoeveel tijd kost het.....</i>	21
5.2	<i>Vergaderen.....</i>	22
5.3	<i>Planning en timetracking.....</i>	23
5.4	<i>Het wiel en de community.....</i>	24
6	Conclusie.....	27
7	Bronnen.....	28

1 Inleiding

Deze scriptie heb ik geschreven tijdens en na mijn afstudeerproject aan de opleiding Design for Virtual Theater and Games van de Hogeschool voor de Kunsten Utrecht. Mijn idee bij deze scriptie was om enigszins in kaart te brengen hoe je een project in het werkveld van de virtueel theatermaker uitvoert. Ik heb deze scriptie in eerste instantie voor mijzelf geschreven. Ik wil graag een blauwdruk hebben van hoe ik een project op succesvolle wijze kan doorlopen. Ik ben echter wel van mening dat de onderwerpen van deze scriptie van waarde kunnen zijn voor elke beginnende virtueel theatermaker.

In hoeverre het mij gelukt is om een blauwdruk te verkrijgen zal de tijd moeten leren. Tijdens mijn afstudeerproject zijn veel dingen niet gegaan zoals ik het gewild had. De meesten buiten mijn macht. De methoden en ideeën die ik beschrijf zal ik in de toekomst blijven toepassen en vanzelfsprekend aanpassen waar nodig. Ik ga er niet vanuit dat ik altijd bij het juiste eind heb en sta open voor alle kritiek en ideeën van anderen.

1.1 Virtueel Theatermaker

Ik gebruik in deze scriptie bewust de term “virtueel theatermaker”. De virtueel theatermaker is een persoon die eigenlijk gespecificeerd is in 2 dingen. Hij past computers in het theater toe en hij past theater op de computer toe. Dit is eigenlijk de meest beknopte beschrijving van de virtueel theatermaker zoals ik hem of haar zie. *(Ik ga verder in de mannelijke vorm)* Concreet gezien zou de virtueel theatermaker een game kunnen bouwen. Dat kan een game zijn zoals ze nu ook geproduceerd worden maar de virtueel

theatermaker onderscheid zich mijns inziens doordat hij meer vanuit een theater achtergrond een game bouwt. Een gevolg hiervan kan zijn dat niet de speler de belangrijkste rol heeft maar meer de ervaring die de speler meemaakt. De virtueel theatermaker zou ook computers kunnen toepassen in een theatervoorstelling. Zo zou hij bijvoorbeeld virtuele acteurs in het theater kunnen projecteren.

In deze scriptie heb ik mij minder gericht op het werkveld van de virtueel theatermaker die computers toepast in het theater. Toch denk ik dat alles zeker van toepassing is op deze kant van het werkveld.

Het werkveld van de virtueel theatermaker is nauw verbonden met het werk van de games industrie. Vanzelfsprekend komt veel van de techniek rechtstreeks uit de games industrie. Eigenlijk is het virtueel theater het eigenwijze broertje of anarchistische zoontje van de games industrie. Omdat ik persoonlijk de games industrie weinig vooruitstrevend vindt wil ik ook liever de virtueel theatermaker los van de games industrie bekijken. De virtueel theatermaker maakt theater. Een game kan theater zijn maar dat hoeft niet. Een product van de virtueel theatermaker hoeft geen game te zijn. Een van mijn persoonlijke drijfveren is het streven naar kunst middels het computer medium.

De virtueel theatermaker heeft een artistieke eigenschap. Ik vindt dat vanzelfsprekend. Ik zie de virtueel theatermaker als een kunstenaar. De virtueel theatermaker vertelt zijn verhalen en uit zich middels het computer medium. De artistieke kant van de virtueel theatermaker behandel ik niet uitgebreid in deze scriptie.

1.2 Aannames

Ik ga ervan uit dat de lezer enigszins bekend is met de

games industrie of bekend is met de opleiding Design for Virtual Theater and Games. Design for Virtual Theatre and Games is een opleiding binnen de Hogeschool voor de Kunsten Utrecht. Deze opleiding leidt je op tot game designer vanuit principes die uit het theater stammen. Deze opleiding heb ik zelf gevolgd en daar komt ook de term virtueel theatermaker vandaan. Er komen termen aanbod die voor de leek wellicht onbekend zijn. De doelgroep van deze scriptie zijn collega virtueel theatermakers. Ik ga ervan uit dat zij met de gebruikte termen bekend zijn.

De scriptie richt zich met name op de kleinere productie projecten van de virtueel theatermaker. Bij grotere productie projecten, bijvoorbeeld met 10 man of meer en looptijden van meer dan een half jaar, komt meer kijken dan in dit document beschreven is. Dit document is meer een soort handboek voor de beginnende virtueel theatermaker.

Je kunt het compleet oneens zijn met hetgeen beschreven is. Dit is niet een document wat pretendeert DE methoden te beschrijven om een project te kunnen uitvoeren. Dit document is een beschrijving van hoe ik een project uitvoer en ga uitvoeren en waarom ik het op die manier doe.

^@^O<84>I^B^@^@<8B>MËÿMÎ^O¶^Q<8B>ËÖä UÄÿÿËË<83>;ØrÜ^O^·^DE^@<90>@^@#EÄ<8B><8D>
^D<81>^O¶H^AÖmÄ+Ü^OöÄ^P^t^X<83>ä^O<89>^O·@^B<89>F^LÇ^F^D^@^@é^k^B^@^@öÄ^O<85>^E
ÿÿÿ<89>N^L^O·H^B<8D>^D<88><89>éP^B^@^@<8B>ë <83>}I^@^O<84>Q^B^@^@<8B>MËÿMÎ^O¶^Q
<8B>EÖä UÄÿÿËË<83>;ØrÜ^O·^LE^@<90>@^@#MÄ^AN^L<8B>EÖmÄ+ØÇ^F^E^@^@<8B>EÖ<8B>V^L
<8B>E+I<81>é ESC^@^@;Ës^S<8B><8E> <9B>^@^@+Ë+I<8D><8C>^A`äÿÿé^D<8B>Ë+Ë<83>~^D^@
<89>Mä^O<84><90>ÿÿÿ<8B>}Ö<85>ÿ^O<85><91>^@^@<8B>¼ <9B>^@^@;Çu#<8B><8E>= <9B>^@
^@<8D><96> ESC^@^@;Ët^S<8B>Ä;Äs^G+ËI<8B>üé^B+æ<85>ÿudÿ<89><86>`<9B>^@^@ä
^B^@^@<8B><86>`<9B>^@^@<8B><8E>= <9B>^@^@;Ä<89>EÖs^G<8B>ü+æO<8B>¼ <9B>^@^@+æ<8B>
<96> <9B>^@^@;Ä<89>Uöu^_<8D><96> ESC^@^@;Ët^U<8B>Ä;Ä<89>EÖs^G+ËI<8B>üé^E<8B>}Ø+æ
<85>ÿ^O<84>d^A^@^@<8B>Mä<8A>^Q<88>^P@AO;<8E> <9B>^@^@<89>EÖ<89>Mä<89>}Öu
<8D><8E> ESC^@^@<89>MäÿN^D^O<85>:ÿÿÿéÄöÿÿ<8B>EÖ<8B>}D<85>Ä^O<85><91>^@^@<8B>
<8E> <9B>^@^@;üu#<8B><86>= <9B>^@^@<8D><96> ESC^@^@;Ät^S<8B>ü;Øs^E+ÇHé^D+I<8B>Ä
<85>Äudy<89>¼`<9B>^@^@ä8^A^@^@<8B>¼`<9B>^@^@<8B><8E>= <9B>^@^@;ü<89>}Øs^G<8B>Ä+ÇH
<8B><86> <9B>^@^@+Ç<8B><96> <9B>^@^@;ü<89>Uöu^_<8D><96> ESC^@^@;Ët^U<8B>ü;ü<89>}
Ds^G+I<8B>Äé^E<8B>Eæ+Ç<85>Ä^O<84><93>^@^@<8A><88>^OGH<89>}D<89>EÖé^Qöÿÿ<83>ü
^Gv <83>ÿEÿÿME<8B>EÖÿ<89><86>`<9B>^@^@ä±^@^@<8B><8E>`<9B>^@^@<8B><96>= <9B>
^@^@;Ë<89>MÖs^G<8B>Ä+ÄH<8B><86> <9B>^@^@+Ä;Ë<89>EÖu9<8B><86>^T^E^@^@<83><89>^Fu3
<8B>^F<83>ø^O^O<86>2öÿÿé<93>öÿÿ<8B>EÄ3ÿ<89><86>^`^E^@^@<8B><89><9E>^X^E^@^@<89>x
^Dé<98>öÿÿ3ÿéÿöÿÿ3ÿGéqöÿÿL^V@^@ ^V@^@ö^V@^@F^W@^@Ä^W@^@^X^@^@N^Y@^@¿^Y@^@x^P@^@^M
^R@^@2^R@^@é^S@^@é^?^S@^@b^U@^@é: ^P@^@I^Y@^@SV<8B>tS^LW<8B>¼`<9B>^@^@<8B><9E> <9B>
^@^@;üv^F<8B><9E>`<9B>^@^@<8B>F^L+B;Ør^B<8B>ØSWÿ+Ä<89>F^Lä<81>J^@^@^A<8B><86>`
<9B>^@^@CÜ;öu^V9<86> <9B>^@^@<8D>¼`ESC^@^@u^<89>¼, <9B>^@^@ä±<89>¼`<9B>^@^@_ [Ä
^D^@U<8B>i<81>i^@^@éSV<8B>u^LWj^P3ÄY<8D>}<90>ö«<8B><8B>Ö<8B>^A<83>Ä^D<8D>D<85>

2 Professie

Omdat het werkveld van de virtueel theatermaker een nieuw werkveld is, is er geen kwaliteitsnorm. Ik vind het persoonlijk zeer belangrijk om professioneel aan de slag te gaan. Om professioneel aan de slag te gaan dien je je vak te verstaan. Dit hoofdstuk gaat over de professionaliteit bij het werken als virtueel theatermaker.

2.1 Technische Diepgang

Voorop gesteld, met technische diepgang bedoel ik niet dat het alleen maar over perfecte beheersing van techniek gaat. Alleen perfecte beheersing van techniek, is levenloos. De uitdaging is meer dat je met de techniek de diepgang in kunt. Een goede beheersing van de techniek maakt dat het medium een extensie van jezelf kan zijn. Is die beheersing er niet dan zie je vaak dat het andersom werkt. De beoefenaar wordt dan een extensie van de techniek en het medium. De techniek dicteert de beoefenaar.

Dat dit soms goed kan werken is mij weleens opgevallen in sommige uitingen. Ik vindt dat zelf een metafoor voor de mensheid die worstelt met zijn eigen vooruitgang. Het schrikbeeld van deze metafoor is dat de vooruitgang van de mens zijn eigen ondergang wordt. Het is jammer dat ik daar geen concrete voorbeelden van kan noemen behalve dan mijn eigen eerste ervaring op de theatervloer. Een ervaring die ik nooit zal vergeten want het was een verschrikking. We moesten na ons eerste jaar als DVTG onszelf en ons werk presenteren in het theater. We moesten een stuk theater neerzetten. Er waren zelfs journalisten van Game magazines uitgenodigd, was ons verteld. Een leuke uitdaging maar 5 minuten voordat de voorstelling zou beginnen hield de eerste

computer ermee op. Vervolgens bleek het programma dat op die computer moest draaien ook de geest te hebben gegeven. We konden het publiek niet laten wachten dus na enig gepruts hebben we de gehele intro uit het programma gehaald en Maartje (oud klasgenoot) met veel moeite omgepraat om maar even een woordje te doen voor het publiek. Je moet je voorstellen dat de theatervloer bezaaid was met alle oude computers, tv's en andere rotzooi die we hadden kunnen vinden in een straal van 5 km van het theater. Maartje legde het publiek uit dat er wat technische problemen waren en dat daarom waarschijnlijk na 5 minuten voorstelling de pauze al ingelast zou worden. Ik moest de centrale computer bedienen. Deze computer toonde van alles op een groot scherm in het midden van het theater. Mijn taak was om alle "file requesters" op te vangen! Deze sprongen af en toe voor het beeld omdat het programma weer een paar bestanden niet kon vinden welke in de digitale mist waren opgegaan. Ondertussen waren een paar andere DVTG-ers op de achtergrond aan het gamen. Ook hiervan hadden we beeld en wederom moest ik een camera bedienen die op het centrale beeld toonde wat alle gamers op de achtergrond aan het uitgesproken waren in de digitale wereld. Dat ging vanzelfsprekend met veel kabaal en geweld gepaard. Ik stond op een klein podium aan de rand van de theatervloer samen met Mirjam die ook computers bediende. Midden in alle drukte had zij mij schijnbaar ingefluisterd dat ze haar computer niet werkend kreeg en dat ze de computer maar even zou herstarten. Ik had er geen oor voor aangezien ik druk bezig was om de centrale computer te besturen zodat het publiek nog wat van het digitale geweld kon mee krijgen. Ik kon op het laatste moment nog voorkomen dat Mirjam de reset knop van de centrale computer in drukte. Dat dit wel wat met wat gewelddadige gebaren en uitdrukkingen gepaard ging werd mij achteraf pas verteld. Ondertussen voorzag Machiel via een headset vanuit het regiehok mij de gehele voorstelling van de morele ondersteuning. Het is dat Machiel mij zo goed door de gehele situatie heen praatte anders was ik gillend weggerend. Wat was ik blij toen het

allemaal achter de rug was om vervolgens van Marcel Albers te horen dat hij in tijden niet zo genoten had en dat dit de beste voorstelling van het jaar was geweest.

Ik schrijf deze scriptie niet om dit gepruts verder te propageren. Het is mijn inziens absoluut de bedoeling om meester van het medium te worden. Daar is een gedegen technische kennis noodzakelijk bij. Ik mag het graag vergelijken met de schilderkunst waarbij de kunstschilder zijn gereedschappen volledig beheerst en eigen gemaakt heeft. Hij zal instaat zijn een prachtig gedetailleerd stilleven te schilderen. Wat je echter ziet is dat dat vaak niet gebeurt. Waarom? Het is niet nodig en het gaat daar helemaal niet om. Juist de diepgang doet zijn intrede als de techniek alleen nog maar gereedschap en drager is. Het doek, de kwast, de verf, de kleuren, de aandacht, het wordt allemaal een extensie van de kunstenaar. Zo ook de techniek bij de computer. Alhoewel complexer van aard.

2.2 Het prutsen voorbij

Ik kan vele redenen bedenken waarom er zoveel geprutst wordt met computers maar geen fatsoenlijke verzinnen. Prutsen is goed, ik doe zelf eigenlijk nooit anders. Het is leerzaam, het is leuk, het kan vernieuwend zijn, enzovoort. Dus vooral blijven prutsen is mijn motto. Echter we moeten er wel aan voorbij kunnen gaan. Immers we willen meesters van het medium zijn. Mijn zoektocht naar dit thema bracht me bij de verstandhouding tussen de computer en de gebruiker. De afstand. Iedereen kent het algemene idee wel dat de computer als apparaat ervaren wordt dat nooit doet wat jij wilt. Ik ben het daar ook zo ontzettend mee eens. Als er 1 regel geldt op de computer dan is het wel die van Murphy. Op mijn weg kwam het boek “Zen de kunst van Motor Onderhoud” en de daarvan afgeleide “Zen en de kunst van Computergebruik”. Daaruit kwam naar voren dat onze

relatie met computers erg dual is. Ik wil niet de spirituele reis gaan maken om één te worden met mijn computer maar ik ben wel van mening dat dit waar is. Wanneer treedt een conflict met het apparaat op? Zodra jij iets van het apparaat wilt maar het apparaat geeft het jou niet. Voeg daar nog eens onbegrip aan toe en ons conflict is compleet. Dit onbegrip is gemakkelijk weer te splijten in twee schuldigen. Allereerst begrijp jij niet waarom het apparaat het niet doet. Je weet kennelijk niet genoeg van het apparaat om te begrijpen waarom hij niet doet wat jij wilt. Ten tweede is de schuld te leggen bij de maker van het apparaat die allereerst niet heeft kunnen bereiken dat het apparaat iets doet wat jij wilt en daarbij waarschijnlijk niet uitlegt waarom het apparaat niet doet wat jij wilt. Mijns inziens is dit probleem makkelijk geanalyseerd zo. Is het zo makkelijk? In weze wel. Er komt echter een ander probleem bij. De complexiteit van de computer is moeilijk te doorgronden. Stel je een machine voor die alleen maar uit aan- en uitschakelaars bestaat. Hoe kom je in vredesnaam tot het apparaat waarmee jij je foto's bewerkt, je email controleert, het internet surft etc. Iemand die mij dat uitlegt zodat ik het begrijp, is altijd welkom op de koffie. Toch werkt het apparaat zo. Dit is een ontzettende 'te ver van mijn bed' show voor vele mensen. Dit probleem gaan we echt niet oplossen. Ach, haal alle luxe die het moderne leven met zich meebrengt weg, donder mij in een groot eenzaam bos zonder ook maar iets. Ik zou het waarschijnlijk niet overleven. Dus die kennis hebben we ook niet (meer) en toch gaat het prima, op zich. Laten we het ook maar niet met de complexiteit van het leven vergelijken want ook daar snappen we geen donder van en toch schijnt dat ook wel steeds te lukken.

Terug naar ons onderwerp. Met het prutsen voorbij bedoel ik dat we kwaliteit kunnen realiseren. Als we een productie doen dan willen wij de regie hebben. Het systeem wat we bouwen krijgt alleen de vrijheid die wij hem bewust geven. Neemt het systeem zelf de regie en komt daar iets moois uit voort, niks mis mee, dan weten we dat te herkennen en kunnen we daar wat mee. Let wel wij zijn de regisseurs, wij

zijn de kunstenaars, het medium en dus de productie is een extensie van onze geest. Ik ben van mening dat om dit beestje (het medium) te temmen wij ontzettend veel discipline moeten opbrengen. Ik denk dat daarin de oplossing zit voor onze duale verhouding met de computer. Allereerst zullen wij dit apparaat moeten doorgronden. Het is belangrijk om inzicht te hebben in de denkwijze van het apparaat. Hier zijn gewoon handleidingen voor verkrijgbaar en hier is goed les in krijgen. Echter we kunnen zo diep de computer in duiken als we maar willen. Dat gaat simpelweg niet want dan houden we geen tijd meer over om er ooit weer uit te komen. We hebben maar beperkte mogelijkheden dus we moeten een grens ergens trekken. Die grens moet een ieder voor zichzelf bepalen. Een gedegen abstracte kennis van het apparaat is belangrijker dan de specifieke. Verder zul je ervaring met de computer moeten opdoen. Je kunt psychologie gaan studeren om je vriend of vriendin goed te leren kennen maar uiteindelijk is gewoon met hem of haar leven een methode die je langer volhoudt denk ik.

De ervaring leert je wat de mikken en makken kunnen zijn. Wil je wat van het apparaat en doet het apparaat wat anders? So be it. Je weet dat dit soms gebeurt dit heb je ten alle tijde in gecalculeerd. Hoe beter jij begrijpt hoe het apparaat in elkaar steekt hoe beter jij hem kan regisseren. Hoe beter jij instaat bent het onvoorspelbare karakter te accepteren hoe prettiger jij met het apparaat uit de voeten kan. Val niet in het gat van eeuwig gepruts. Vooral als je zo'n neiging tot perfectionisme hebt als ik. Het brengt zoveel onnodige stress met zich mee. Het is zo makkelijk het apparaat tot in zijn hoogste geheugenadres te haten. Moeilijker is om hem te begrijpen en begrip op te brengen. En die kant zul je mijns inziens wel op moeten.

2.3 Niet vis hebben maar kunnen vissen

“Geef een hongerige een vis en zijn honger zal die dag gestild zijn. Leer hem vissen en hij zal nooit meer honger hoeven hebben”. Het misschien bekende gezegde is in zekere zin ook van toepassing op de virtueel theatermaker. Het computer medium is een medium wat aan constante verandering onderhevig is. Dit is een ontzettend sterke eigenschap van de computer. Tegelijkertijd is het een eigenschap wat zijn eigen succes in de weg kan staan. We gebruiken allerlei gereedschappen op de computer. Is het vandaag Director, morgen is het Virtools en overmorgen is het Java. De week daarop is DirectorNG 12.2 en Flash Professional 33.3 2005. Dit gaat zo maar door en dat zal ook zeker niet gestopt worden. Het is dus vrij moeilijk om je te specificeren met 1 gereedschap of algemener 1 kennisgebied. Dat zou ook zonde zijn want mocht je op je CV hebben staan dat je superskills hebt in het ene pakket maar van het andere geen kaas hebt gegeten. Een bedrijf wat toch de keuze aan kandidaten heeft zal degene met brede kennis of specifieke kennis in de door hen gebruikte gereedschappen kiezen. Wat mij opviel in de meeste gereedschappen waar de virtueel theatermaker mee te maken kan krijgen is dat ze allemaal achter de ouderwetse gereedschappen aan hobbelen en eigenlijk alleen maar het ouderwetse in een nieuw jasje stoppen. Stel ik neem Virtools. Een pakket wat een absolute topper is als het aankomt op het maken van interactieve content in 3D. De visuele manier van programmeren maakt de drempel van programmeren een stuk lager. Niets dan lof. Echter als je je verder in het pakket verdiept dan zie je dat het visuele programmeren eigenlijk exact hetzelfde is als welk ander programmeren ook. Je komt bij dezelfde gedachten uit om tot bepaalde oplossingen te komen. Uiteindelijk zal het visuele programmeren je zelfs gaan tegen staan aangezien het een onoverzichtelijk zootje wordt. Neem Flash, in het begin maak je een geweldige dynamische website met geanimeerde buttons en weet ik

veel. Op een gegeven moment wil je verder en duik je 'actionsript' in. Voor je het weet ben je scripts en algoritmes aan het schrijven net als elke willekeurige programmeur in elke willekeurige taal. (Helemaal willekeurig ook niet want er zijn programmeertalen die een radicaal andere aanpak kennen, maar die terzijde gelaten). Uiteindelijk kom je door je programmeer ervaring tot bepaalde kennis die te gebruiken is bij elke programmeertaal. Dat is ook niet zo verwonderlijk want dit is de kennis die je inzicht geeft in hoe de computer intern werkt. Ik noem dit de abstract technische kennis van de computer. Het is de kennis van het kunnen vissen en niet van 1 specifieke gereedschap. Deze kennis goed beheersen zal je instaat stellen om elke willekeurige programmeertaal makkelijk op te kunnen pikken en eigen te maken. Er is echter wel een klein nadeel aan het hebben van deze kennis. Je zult gaan inzien wanneer bepaalde fabrikanten een slechte implementatie hebben.

Ik heb dat zelf met mijn afstudeerproject meegemaakt. Mijn keuze was gevallen op Flash voor het maken van mijn product. De laatste versie van Flash was met lof beschreven en de stap om Actionscript professioneler te maken en te baseren op volwassen talen werd geroemd. Het viel zwaar tegen want vele logica was ver te zoeken en de bugs waren talrijk. De implementatie van Actionscript 2 was zeker niet volwassen te noemen. De ontwikkelomgeving was qua programmeren een uit de steentijd. Uiteindelijk leer je de mikken en makken kennen en kun je er wel mee uit de voeten. Maar wat een frustratie. Nu kan ik redelijk uit de voeten met een aantal programmeertalen en was actionsript eigenlijk een eitje. De problemen die ik tegen kwam kon ik snel beredeneren en uiteindelijk wel oplossen. Ik ervaar het gemakkelijker om begrip op te brengen voor bepaalde implementaties. Flash was wel wat uitzonderlijk omdat ze teveel compatible willen blijven met een obscuur verleden.

2.4 De kunstenaar en de programmeur

“There's too much art in programming and not enough science” (Rui Guerra)

De virtueel theatermaker begeeft zich in 2 werelden van uitersten. Aan de ene kant de logische, mathematische, wetenschappelijke en gestructureerde wereld van de computer. Daar naast de artistieke wereld met zijn vrije, chaotische, ongestructureerde en emotionele wereld. Ik heb me hier langer over gebogen maar moet toch concluderen dat die wereld van 2 uitersten inderdaad zo is.

Als concreet voorbeeld herinner ik mij een situatie waarin ik iemand, die ik een groot kunstenaar vind, liet prutsen met wat algoritmes. Deze algoritmes toonden wat vage grafische effecten op het beeld. Dat vond deze persoon zo geweldig dat hij daar lekker mee aan de knoei ging. Echter van enig begrip hiervan was geen sprake. Algauw had deze persoon het voorelkaar dat de algoritmes helemaal niks meer deden en dat de lol ervan dus ook snel voorbij was.

Ik vind deze persoon een groot kunstenaar. Maar ik vind hem geen virtueel theatermaker omdat ik bij hem een gebrek aan structuur en logica zie. Ik ben daardoor er ook van overtuigd dat hij niet het computer medium beheerst maar dat het computer medium eerder hem beheerst. Zoals ik wel eerder heb gemeld sluit dat niet uit dat daar geen prachtige dingen uit te voorschijn kunnen komen. In het hierboven genoemde voorbeeld zag deze persoon de potentie van een algoritme maar niet de kwetsbaarheid van zo'n algoritme. Ik ben er absoluut van overtuigd dat deze persoon wel kan leren hoe zo'n algoritme werkt. (Zie hoofdstuk 3.1) Daar is eigenlijk alleen een stukje aandacht en wiskunde voor nodig. De onlangs overleden Robert Moog, zegt in de documentaire “Moog”: “I'm an artist and a scientist. Knowing what science to use is being an artist”. Ik sluit me bij die uitspraak aan.

De artistieke kant van de virtueel theatermaker is onontbeerlijk. Iemand die ik dat moet uitleggen komt maar eens op de koffie. Ik heb daar al eerder op gewezen maar alleen maar structuur en logica vindt ik persoonlijk levenloos. Ik wil graag iets van de kunstenaar zien in zijn werk. Ik wil dat hij zijn ziel toont en ik wil me daar aan reflecteren. Ik denk dat er best wel een scheidingslijn te trekken is op momenten tussen de artistieke en gestructureerde kant van de virtueel theatermaker. Tijdens de bouw en de planning en structurering hiervan komt de wetenschappelijke kant meer aanbod. Tijdens de idee ontwikkeling en uitwerken van concepten zal de artistieke kant zegevieren.

Voor de kunstenaarszieltjes speelt bij sommigen een slopende eigenschap mee. De kunstenaars ziel is vaak een lijdende ziel. Val me alsjeblieft niet aan op deze uitspraak want het hoeft net zo goed niet te zijn als wel. Ik heb voor deze scriptie een aantal documentaires bekeken over een aantal kunstenaars, te weten Picasso, Matisse, Gaudin en van Gogh. Daarnaast kijk ik graag goed om me heen. Het valt me gewoon op dat vele kunstenaars erg zwaar op de hand leven. Als je jezelf misschien een beetje herkent hierin dan nodig ik je uit om de uitdaging van het beheersen van het computer medium aan te gaan. We hebben je nodig! We hebben al teveel programmeurs die kunstenaar proberen te zijn.

2.5 Bedenkingen

We hebben het gehad over een aantal onderwerpen en vele wezen in de richting van techniek. Tijdens “Technische Diepgang” hadden we het over dat de techniek een extensie van jezelf moet worden. Bij “Het Prutsen Voorbij” hebben we het gehad over de problemen die de techniek met zich meebrengt en die je weerhouden om het eigenlijk een extensie van jezelf te laten zijn. Vervolgens hebben we het gehad over de schizofrenie van de virtueel theatermakerij. Ik

wil nu proberen in te gaan op problemen wat dit alles met zich meebrengt.

Ik ken een flink aantal programmeurs en mag graag met ze praten over het computer medium. Ze lachen mij weleens uit als ik weer eens de verkeerde term gebruik bij het verkeerde onderwerp gebruik. We praten vaak over het bouwen van projecten en dan met name over de techniek. Wat mij opvalt is dat programmeurs vaak zeggen dat iets niet kan. De exacte reden doet er vaak niet toe want als ik doorvraag is er vaak wel omheen te werken. Ik weet niet precies wat het is maar ik betrap mezelf er ook weleens op. Het probleem is namelijk dat verdieping kan leiden tot een kokerblik. Als je je verdiept in een techniek wordt het daarna moeilijk om er voorbij te kijken. Je gaat alles vanuit de techniek bekijken en voor je het weet draai je mee in een besloten vicieuze cirkel. Je hoort vaak dat de computer nerds in hun eigen wereld leven die voor buitenstaanders vaak onbegrijpelijk is. Dus zou je dan kunnen stellen dat de verdieping in de techniek leidt tot een verloedering van creativiteit? Ik geloof daar niet zozeer in maar moet wel bekennen dat er een verleiding is. Een verleiding om vanuit de techniek allerlei dingen te doen in plaats van de techniek als middel te gebruiken. Creativiteit wordt juist vaak opgestookt als het begrensd wordt en in dit geval is de techniek één van de grenzen. Dan zou je kunnen stellen dat verdieping in techniek de grenzen ruimer maakt en dat dat de creativiteit weer laat af nemen. Misschien is het eerder zo dat de techniek al van alles mogelijk maakt maar dat we in een leegte vast zitten omdat de grenzen van de techniek buiten bereik zijn. Tja daar zit je dan in de leegte. Ga maar wat maken als je niet zoveel kan. De paden richting de grens geven weinig uitzicht over het landschap maar een eigen pad uitzetten is wel wat veel werk. Als je het in deze metafoor plaatst dan zou de virtueel theatermaker inderdaad in deze leegte kunnen zitten. Mij lijkt het dan dat we niet met z'n allen de grens gaan lopen zoeken maar gewoon een dorpje op deze plek stichten en in de omtrek een beetje gaan rondhangen met elkaar. Met andere woorden, laten we

gewoon aan de slag gaan met wat we hier hebben. Dat bouwen we vanzelf wel uit als de behoefte daar toe aandringt.

Ken Levine: I'll never forget the first story I wrote in gaming. It was for a (eventually canceled) Star Trek: Voyager game. I wrote the opening cutscene, which included this gem: THE CAMERA ZOOMS IN ON JANEWAY... WE SEE A LOOK OF TERROR IN HER EYES AS IT REFLECTS THE INCOMING MISSILE. The lead programmer pretty much laughed in my face. First of all, our characters were low-resolution bitmaps, with one fixed expression on their face. Their eyes were maybe 4x4 pixels each. The camera zooming in on that wouldn't have shown a performance; they would have shown a scattered mess of random pixels."

In dit voorbeeld schreef de kunstenaar een scène voor een game maar werd meteen door de programmeur op de vingers getikt. Volgens mij is dit het cliché voorbeeld van de botsingen tussen de programmeurs en de kunstenaars. Ik ben ten eerste van mening dat de kunstenaar het medium enigszins moet kennen zodat hij wel een beetje kan inzien wat te doen is. Aan de andere kant staat de programmeur die lacht en zegt dat het niet kan. Het voorbeeld is best wel uit te voeren en zo moeilijk is het nou ook niet. Het houdt alleen een heleboel extra werk in voor de programmeurs. Waarschijnlijk hadden ze die tijd niet gezien de grootte van het project.

Een ander probleem van het beheersen van de techniek is een van het hebben van overzicht. Stel je hebt flink wat abstract technische kennis van de computer opgedaan. Je voelt je er vertrouwd mee en je groeit erin mee. Op een mooie dag krijgt je artistieke kant een geweldige ingeving maar dat eureka moment wordt meteen overschaduwd door je technische kennis die ineens inzien hoeveel werk dat wel niet is. En hoe hou je dat complexe systeem in de hand? Laat je alsjeblieft niet ontmoedigen door deze momenten. Met de juiste discipline is elke complexiteit wel klein te krijgen. Als je zelfs een erg goed idee hebt kun je vaak best wel hulp

vinden.

2.6 Kennis gebrek bij opdrachtgevers

Het afstudeerproject waaraan ik werkte was een opdracht uitgeschreven door de Advies Commissie Beeldende Kunst Broekpolder van de gemeente Beverwijk en Heemskerk. Het computer medium waarvoor zij een opdracht uitschreven is een tamelijk nieuw medium. Het kwam dus ook niet als een verrassing dat deze commissie de kennis over dit medium ontbeerde. Zij schreven de volgende 2 opdrachten uit:

Opdracht 1

Webvormgever

Aansluitend op de officiële Broekpoldersite wil de ABKB een site opzetten, die informatie verstrekt over de achtergronden en actualiteiten van de beeldende kunst opdrachten in de Broekpolder. Behalve een actuele stand van zaken zal de website ook het begin zijn van een verslaglegging van de processen en resultaten van de kunst opdrachten in de Broekpolder. Voor het ontwerp hiervan zoeken wij een webdesigner, die vorm en inhoud op creatieve wijze op elkaar aan laat sluiten. Voor het ontwerp van de website is € 2.500,- ex BTW gereserveerd.

Opdracht 2

Webmaster

Het web speelt een belangrijke rol voor de (toekomstige) bewoners van de wijk. Al voor zij zich vestigen in de Broekpolder, leggen ze contacten en wisselen ze informatie uit via eigen websites.

Het gaat hierbij veelal om praktische zaken. De opdracht aan de webmaster is bedoeld om met de bewoners in dialoog te treden op een ander vlak: dat van de hoop, de dromen en verwachtingen waarmee ze naar de Broekpolder komen.

De webmaster krijgt hiervoor ruimte op de website over kunst in de Broekpolder, maar kan zich ook mengen in de gesprekken op andere sites.

Voor deze opdracht is een budget van € 6.000,- ex BTW gereserveerd.

Het maken van de website is duidelijk. Websites zijn een onderdeel binnen het computer medium die zich al verder ontwikkeld hebben. Een bedrijf of instelling kan tegenwoordig niet meer zonder. Mensen hebben al veel meer aanraking gehad met websites dan met bijvoorbeeld games, laatstaan kunst op de computer. Het gebruik van websites is inmiddels gemeengoed geworden.

Wij reageerden op deze opdracht met het idee om deze 2 opdrachten met elkaar te verweven. Voor ons leek het logisch dat de gene die de website van de ABKB bouwde ook onderhield. Toen we met de commissie een gesprek aangingen bleek uit het gesprek helemaal niet dat ze een webmaster wilde hebben die de website van de eerste opdracht onderhield. Of iemand die conversaties trachtte aan te gaan met bewoners op sites waar zij al met elkander communiceerden. Misschien was dat ook wel wat banaal gedacht van ons. Het project van de webmaster was een project wat onder één van de beeldende kunst opdrachten viel die in de Broekpolder uitgevoerd werden. Wij hadden dat

zo helemaal niet geïnterpreteerd maar wilden wel graag die kant op. Het mocht dus een losstaand kunstwerk op internet zijn die een dialoog trachtte aan te gaan met de bewoners. De woordkeuze van de commissie bleek enigszins een verkeerde mijns inziens. De term “webmaster” is een algemeen bekende term voor een persoon die websites en/of servers beheert. Een belangrijk onderdeel van de acquisitie van soortgelijke opdrachten is een transparante communicatie met de opdrachtgevers. Ik denk het gezond is om aan te nemen dat de opdrachtgevers weinig kennis van zaken hebben. Als dat meevalt ontdek je dat meestal snel genoeg. Hierdoor moet je jezelf dwingen om jou ideeën zo helder mogelijk uit te leggen zonder in onbegrijpelijke termen te vallen. Want termen hebben we genoeg in ons wereldje:

Hoi Nanon,

We hebben een beta ge-upload. Deze staat nu online op de onderstaande URL. Hier zitten nog wel een aantal bugs in. De preloader doet het niet aangezien het compileren van de preloader veroorzaakt dat gecachte objecten niet meer aanwezig zijn. Tevens doen een aantal object classes het niet omdat de preloader ze te laat in zijn cache zet. Hier moeten we nog even omheen werken. Het gevolg voor nu is dat het scherm vrij lang zwart blijft alvorens je kunt beginnen. Ook is de bestandsgrootte vrij groot maar dit wordt veroorzaakt door een compressie bug in de compiler. Mocht de intro het niet doen dan dien je de bestanden in C:\documents and settings\Application data\Macromedia\Flex player\shared objects\localhost.swf te verwijderen. Het kan zijn dat deze bestanden hidden zijn maar dan moet je in de opties van windows verkennen verborgen bestanden tonen aan zetten.*

Mocht er wat zijn dan hoor ik het wel.

Vanzelfsprekend kan deze mail alleen verstuurd worden naar iemand die wel wat verstand heeft van zijn computer. Een leek zou je dit niet uit kunnen leggen.

H^O^E^P^C^A^<99>+ûÁ^O¶Ä^KÈ<8D>E6P<89>Møý^UHp@^@<89>E^TP<8D>EäPÿu^Lÿ^U`r@^@ÿu^Tÿx<83>Eè^D<83>Eð^D9]è^O<8C>wÿÿÿ<83>~Pÿt}3Üh,<90>@^@SSSSj^ASSj^Ah4^B^@^@SSSSj(ÿ^ULp@^@;Ä<89>E^TtU<8B>] ^Lj^ASÇEä^P^@^@^@ÇE^@^@^@ÿ^UPp@^@ÿvPSÿ^UTp@^@ÿu^T<8B>5\p@^@SÿÖ<89>E^L<8D>Eäh^@^@Pjÿh äB^@Sÿ^Udx@^@ÿu^LSÿÖÿu^Tÿx<8D>E#Pÿÿ^Uhx@^@ ^3Ä[ÉÄ^P^@<83>=<8C>Y@^@^@Vu-3E<8B>Ä^<8B>ð<80>ä^AöÜESCÖ<81>ä <83>,iÑè3ÄNuè<89>^D<8D><8B>Y@^@Ä<81>ù^@^A^@^@|Ö<8B>T\$^P<8B>D<85>ö÷ðv#<8B>L\$^LW^O¶9<8B>ð<81>æÿ^@^@3÷Ä<8B>4µ<8B>Y@^@3EÄJuä ÷ð^Ä^L^@V<8B>té<84>^@^@^@<8B>E<8B>^MPiB^@kÄ^\\^CÄ<83>8^AtzPe<81>^@^@^@=ÿÿÿ^?ts<85>Ä)^S@^: ^@ðB^@Ää+EQèG^@^@<85>Äu^F3Ä@Fè^GH<8B>Î<8B>ð+Ä<83>| \$^L^@t8^A^E^LäB^@; öäB^@3Ej^@<85>Ä^O<94>Ä^CÈQh0u^@^@ÿ5^LäB^@ÿ^U^Pq@^@Ph^B^D^@^@ÿt\$^Xÿ^UPr@^@<85>ö^O<8D>tÿÿÿ3Ä^@ ^ÿÿÿ^?äöU<8B>i<81>iu^A^@^@;xiB^@SVW<89>EÎ; \$iB^@j^\\<89>Eøÿ<8D>EÖ3ÜP<89>]üèDD^@^@<8B>Eø<8B>MÜ<8D>Uø<8B>ð<8B>ù<89>^UÄ@^@<8B>UÖÄæÄç<83>Äp<81>E^@ðB^@<81>Ç^@ðB^@<83>üA^O<87>T^V^@^@ÿ\$<95><9D>7@^@SPèW<^@^@é<85>^O^@^@ÿ^ÿ^EiäB^@9]ø^O<84>v^O^@^@éj^O^@^@;Ä)^Q@^: ^@ðB^@Ää+EQè@F^@^@HSPè¶ÿÿé^V^V^@^@;Èt) öt^O; 8<90>@^@EÖ<92>@^@é6^U^@^@; ö<92>@^@<89>^MÖ<92>@^@E8<90>@^@é6^U^@^@SPèE^M^@^@Sè<9E>^W^@^@<83>ø^A^?^C3Ä@Pÿ^U p@^@é4^U^@^@ÿÿuøÿ^U^Tr@^@é6^U^@^@j^Aäu^W^@^@<8B>Mø<89>^D<8D><80>iB^@é<99>^U^@^@<8B>Eä<8D>4<85><80>iB^@3Ä<8B>^N; È^O<94>Ä#Mä<8B>D<85>ø<89>^Né<83>^U^@^@ÿ4<8D><80>iB^@Vé<97>^T^@^@<8B>^UðäB^@<8B>5^Xr@^@; Öt^GQRÿÖ<8B>Eø<8B>^M^DäB^@; È^O<84>F^U^@^@PQÿÖé=^U^@^@jðè ^W^@^@ÿuUPÿ^U<9C>p@^@<85>Ä^O<85>\$^U^@^@éø^R^@^@jðè^B^W^@^@<8B>øWeÄA^@^@8^ ^<8B>ðt=; öt9j\VèDÄ^@^@<8B>ðW<8A>^F<88>^^<88>E^Kè÷H^@^@; Äu^LSWÿ^U<98>p@^@<85>Äè^Cö^@^@P u^CÿEü<8A>E^K<88>^FF: ÄuÇ9]ÜWt^^jæèÑ: ^@^@Wh^@HC^@èIE^@^@Wÿ^U<94>p@^@é@^T^@^@jðé

3 De trukendoos

Lang leve de computerwereld waar alles nep mag zijn als het maar wat lijkt. Ook het computer medium zit hier vol mee. Het slepen van een venster op je super heftige Gamer Edition 4000 XP Pro 2005 NG, is allemaal nep. Wat er werkelijk gebeurd is dat er 2 tekeningetjes zijn in je computer. 1 tekeningetje is zichtbaar en de andere wordt op dat moment getekend. Zodra het tekeningetje dat getekend wordt klaar is dan wordt hij getoond en wordt het andere tekeningetje weggegooid en op zijn plaats wordt weer een nieuw tekeningetje getekend. Dit gaat zo non-stop door. Alles is nep, niks echt, het is allemaal theater. Dus ken je dit theater goed dan kun je van alles uithalen. Het wordt dan pas echt leuk.

3.1 Algoritmes

Algoritmes zijn de kunst van het calculeren. Een reeks instructies die een bepaalde taak uitvoeren. Zoals ik het zelf gebruik is een algoritme een wiskundige beschrijving van bijvoorbeeld een animatie. Een simpel concreet voorbeeld. Stel je wilt een object om een bepaalde punt laten circuleren. Ik gebruik dan de volgende reeks acties:

```
Getal_i = Getal_i + 1;  
object.Xpositie = sinus(Getal_i);  
object.Ypositie = cosinus(Getal_i);
```

Stel ik voer deze 3 acties elke seconde uit. Het object zal dan ronddraaien om het punt $x=0$, $y=0$ met een straal van 1. Als je dit eenmaal weet typ je dit sneller in dan dat je het kan animeren. Het biedt je ook nog mogelijkheid genoeg voor aanpassingen en variaties. Hoe beter je de formules achter

een beweging snapt hoe sneller je hem kunt doorgronden en kunt omzetten naar een algoritme. Het animeren van een vis is bijvoorbeeld perfect te doen met een algoritme. Je kunt natuurlijk met de hand de staart van de vis heen en weer zwiepen maar je kan ook een formule schrijven die dat voor je doet. Wil je daar nog verder in gaan dan breid je je formule uit met bijvoorbeeld het aanpassen van de snelheid van het zwiepen van de staart aan de hand van de acceleratie of deceleratie van de vis. Heb je deze formule eenmaal uitgewerkt dan hoef je voor de vis alleen nog maar het traject met de hand te animeren. Maar dat zou je natuurlijk ook weer met een algoritme kunnen doen.

Bij het bouwen van een virtueel theater project kun je dus algoritmes goed gebruiken om bijvoorbeeld de animatie van een object aan te passen aan bepaalde omstandigheden. Op die manier krijg je dynamische animaties. Bij het bovengenoemde voorbeeld is het simpel om een variatie toe te voegen waardoor het object sneller ronddraait.

```
Getal_i = Getal_i + 2;
```

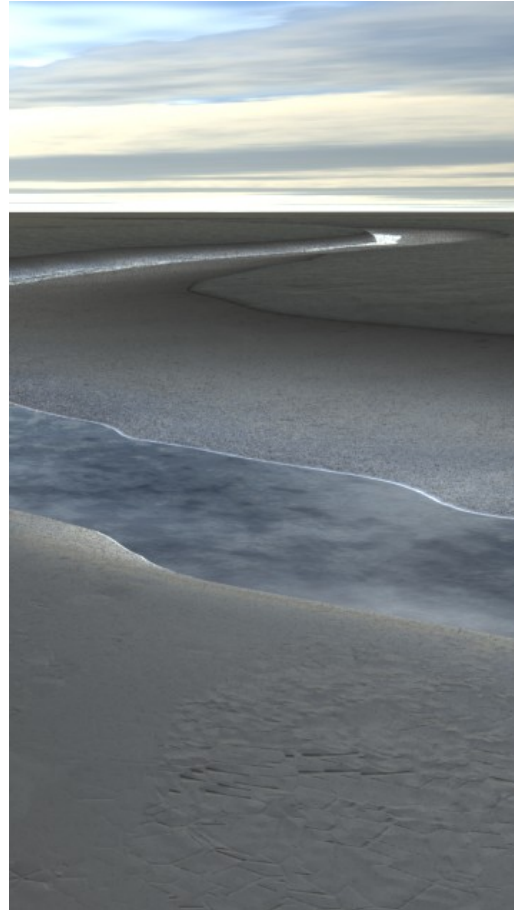
Het object zal nu 2x zo snel ronddraaien. Of een andere variatiewaardoor het object een steeds grotere straal zal krijgen.

```
Object.Xpositie = sinus(Getal_i) * Getal_i;  
Object.Ypositie = cosinus(Getal_i) * Getal_i;
```

De straal waarmee het object nu ronddraait zal steeds groter worden. Ik hou het hier ontzettend simpel om het duidelijk te kunnen maken. Een beetje ingewikkelde beweging zal natuurlijk veel complexer worden. De mensen die ooit Wiskunde B in hun pakket gehad hebben zullen zeker veel aan die kennis hebben bij algoritmes. Mensen die maar weinig kennis van wiskunde hebben zullen hier lastiger mee uit de voeten kunnen. Voor die mensen is er natuurlijk altijd het pure handwerk beschikbaar.

Een ander pracht terrein van algoritmes is bij textures in 3D pakketten. Hierbij hoef je meestal niet zelf met wiskunde

aan de slag. De mensen die enigszins bekend zijn met het werken in 3D begrijpen dat je een texture op een 3D object plaatst. Daarvoor zijn verscheidene technieken beschikbaar. Zo is er ook de techniek van procedurele textures. Eigenlijk zijn dat textures die door een algoritme gegenereerd worden. Het enorme voordeel van procedurele textures zijn dat ze geen begin of einde kennen. Je kunt er oneindig op inzoomen en ze blijven scherp. Je kunt ze blijven herhalen maar je ziet niet of nauwelijks waar de herhalingspunten zitten. Je kunt ze in de hoogte gebruiken zonder dat je ziet dat ze uitgerekt worden. Ook met procedurele textures kun je ontzettend mooie maar ook complexe resultaten boeken. In ieder geval een onmisbare kennis bij 3D werk.



Illustratie 1: voorbeeld realisatie met procedurele textures

3.2 Herhaling

Net zoals in theater is herhaling een veel gebruikte methode. Althans dat is mij vaak opgevallen. Acteurs zie je vaak dezelfde bewegingen doen en dezelfde teksten uitspreken. Vaak is het een karakter trek van de personage die ze spelen. Soms is het een terugkerend thema of handeling. In ieder geval maakt herhaling het voor het publiek herkenbaar. Ik zie mijzelf niet als een kenner in de theaterwetenschappen. Ik wil ook niet over dezelfde herhalingen hebben zoals dat in theater gebeurt. Al zie ik dat wel als een belangrijk onderdeel van de virtueel theatermaker. Dat is een heel eigen scriptie waard. Herhalingen bij het computer medium sluiten perfect aan bij het vorige onderwerp; Algoritmes. Computers zijn ontzettend goed in kopiëren. Niet voor niets drijft het computer medium de muziek industrie tot wanhoop. Herhaling binnen de computer heeft ontzettend veel voordelen. Het is efficiënter voor het geheugen gebruik, processor gebruik, video kaart belasting en bus belasting. Redenen genoeg.

De ontwikkeling van elektronische instrumenten is enigszins vergelijkbaar. Een elektronisch instrument produceert eigenlijk een elektronische golf. De golf wordt constant geproduceerd en als je je speakers op deze golf aansluit dan hoor je het geluid van die golf. Deze golven worden eigenlijk ook door algoritmes opgewekt. Door middel van een aantal variabelen te veranderen, verander je het geluid. Zo werkt een elektronische synthesizer van origine. Moderne synthesizers werken in principe nog steeds zo. Echter er is iets anders bijgekomen. Tegenwoordige synthesizers zijn ook vaak 'sample' gebaseerd. Dit houdt in dat de synthesizer een stukje opgenomen geluid gebruikt als basis om te veranderen en niet zozeer de geluids basis zelf creëert door middel van een algoritme. Je kunt op zo'n zelfde manier stellen dat bijvoorbeeld een animatie gecreëerd door een algoritme de oorspronkelijke synthesizer is en dat je nu kunt gaan zien dat er samples bij komen. Dit is te vergelijken

met een handgemaakte animatie die je kunt beïnvloeden of verrijken door een algoritme, of andersom. In ieder geval vullen beide methoden mekaar aan en is het aan het inzicht van de virtueel theatermaker om in te zien waar welke techniek het beste te gebruiken is.

Het herhalen of hergebruiken van onderdelen zit soms diep in de computer ingebakken. Object georiënteerd programmeren heeft als fundament dat het gemakkelijk te hergebruiken en te herhalen is. Ik wil niet diep op object georiënteerd programmeren in gaan. Daar zijn boeken vol over geschreven. De praktijk is, is dat je je systeem opdeelt in klassen. Het enige wat je doet is beschrijven wat de eigenschappen van klassen zijn. Vervolgens maak je objecten aan die lid zijn van een bepaalde klasse. Een beetje moderne implementatie gebruikt deze methodes. Toch zie je dat ondanks dit principe al uit 1963 stamt er nu vaak nog niet gebruik van gemaakt wordt. Flash heeft dit bijvoorbeeld sinds versie MX 2004 geïmplementeerd. Virtools kent dit al helemaal niet.

Een goed besef van de herhalende mogelijkheden van de computer kan je veel werk uit handen nemen. Samen met algoritmes kun je ontzettende dynamische dingen realiseren met maar een klein beetje programmeer werk. Tijdens 1 van onze lessen Virtools zag ik ooit voor het eerst iets tot leven komen op deze wijze.

We wilden voor een project allemaal beesten laten vliegen die wolken uit poepten. Ter oefening hebben we toen 1 zo'n beestje geprogrammeerd met een aantal eigenschappen. Dit beestje zwom steeds naar een willekeurig punt en naar het middelpunt. Als laatste hadden we het beestje een aantal functies mee gegeven zodat ze konden botsen en mekaar uit de weg konden gaan. We hadden zijn bewegingen zo geprogrammeerd dat ze aangepast werden aan zijn snelheid en richting. Als we dit projectje startten dan zagen we 1 zo'n beestje een beetje rondzwemmen. Niet zoveel aan. Dus toen hebben we geprogrammeerd dat dat ene beestje maar 200 maal gekopieerd moest worden. Dit gaf zo'n prachtig effect.

De beestjes zwommen allemaal rond en botsten op elkaar of zwommen om elkaar heen. Het was 1 grote uitverkoop stoet bij het middelpunt. Vervolgens zat de hele klas naar deze beestjes te kijken om te zien wat ze deden. Heel fascinerend waaruit duidelijk naar voren kwam hoe zoiets simpels zo'n effect kon hebben.

3.3 Knijp eens met je ogen

Omdat toch alles al nep is en meestal het beeld voortdurend beweegt kun je heel veel weglaten. Een vogel die vliegt en wappert met zijn vleugels zie je voor 99% niet.

Een paar jaar geleden hadden een paar klasgenoten een prachtig pittoreske filmpje gemaakt over een vlinder op een zolder. Het was zou eigenlijk een genot moeten zijn om naar te kijken. Echter er was 1 probleem; De vlinder. De vlinder bewoog zo onnatuurlijk dat het de sfeer van het gehele filmpje onderuit haalde. Na een aantal vragen hierover vertelden de klasgenoten dat ze ontzettend gezocht hadden naar hoe een vlinder bewoog en hoe ze dat na konden bootsen maar dat dat niet geheel gelukt was.

Het voorbeeld van de vlinder vindt ik een mooie omdat de makers niet instaat zijn geweest om de beweging van de vlinder te analyseren en geloofwaardig na te bootsen. In de vorm waarin zij de vlinder neergezet hadden was dit wel van belang. Hierin schuilt wel een stuk vakmanschap wat de virtueel theatermaker zich eigen moet maken. Het gaat er niet om dat je de vlinder perfect nabootst het gaat erom dat hij geloofwaardig is. Nogmaals het is allemaal nep en het is allemaal theater. Het gaat erom hoe het overkomt. Ik doe een poging de beweging van de vlinder op te delen. Door middel van opdelen van de bewegingen kun je hem makkelijker analyseren en zelfs omzetten naar algoritmes. Allereerst zie je als je met je ogen knijpt en naar de vlinder

kijkt een grote vlek wat eigenlijk een beetje willekeurig stuiterend door de lucht vliegt. Dus animatie 1 is een willekeur van op en neer gaan en een willekeur van rotatie binnen een bepaald bereik. De vleugels van de vlinder gaan vanzelf sprekend steeds op en neer en dus is dat animatie 2. Daarbij is de vlinder zo licht en zijn vleugels zo groot dat zijn lichaam met het bewegen van de vleugels ook op een neer gaat. Op deze manier krijg je 3 animaties die je vervolgens alleen nog in een hiërarchie moet plaatsen. Allereerst gaan de vleugels op en neer. Vervolgens gaat het lichaam op en neer en als laatste voeg je de willekeur er aan toe. Het belangrijkste wat je aan deze animatie reeks moet toevoegen is dat je je harde werken niet moet zien. Dus een stevige blur op de vlinder maakt je werk stukken minder zichtbaar maar wel geloofwaardiger. Niks is perfect. Maak niet alles strak, een blur doet wonderen zelfs bij een lage framerate. In plaats van een complexe animatie scheelt een blur veel tijd en resource. Dus houdt het motto hoog dat er meer is dat wij niet zien dan dat wij wel zien.

Dit is slechts een simpel voorbeeld van hoe je iets kunt maskeren door dingen weg te laten, juist te analyseren en vervolgens in onderdeeljes op te snijden. De kern van het verhaal is dat je gewoon goed moet observeren. Observeer en vang de kern van wat iets tot dat geen maakt. Kijk naar wat je zelf maakt en probeer daar objectief naar te kijken om te beoordelen of het overkomt. Bedenk dat het niet perfect moet maar goed genoeg. Wees je eigen publiek en leer jezelf om je eigen publiek te spelen. Kijk fris naar hetgeen je maakt en zonder het gehele proces wat erbij kwam kijken in je achterhoofd. Knijp op die manier ook met je ogen. Dit is de kunst van de dramaturgie, een kunst die als je die jezelf niet eigen maakt zult moeten uitbesteden.

3.4 Niet alles hoeft je te zien

Als gevolg op dat je veel kunt maskeren is er zelfs heel veel dat je kunt weglaten. Je kunt veel aan de fantasie van je publiek overlaten en ik denk dat het een verrijking is als je dit goed doet. Een bijkomend voordeel is, is dat het je veel werk kan schelen. Als regel kun je wel aannemen dat als het publiek iets hoort ze niet altijd dat gene hoeven te zien. je ziet dit ontzettend vaak gebeuren in theater. Dus op diezelfde manier kun je dat ook als virtueel theatermaker gebruiken.

4 Bouwen

We kunnen eindeloos prachtige ideeën verzinnen maar vaak heb je toch wel de ambitie om een aantal ideeën af en toe te realiseren. In dit hoofdstuk wil ik nader ingaan op een aantal methoden om een idee concreet te krijgen. Het bouwen is de fase nadat je je concept afgerond hebt en je dit wil concretiseren. Je bent dus niet meer bezig met de vorm van je product. Een keuze tussen 2D of 3D is niet een keuze tijdens of na de bouw fase. Me dunkt dat een gezond verstand dit wel inziet.

4.1 Iteratieve bouw

Bij een iteratieve aanpak begin je met het maken van een volledig functionerende schets. Hierbij laat je je vormgeving even achterwege. Iteratieve bouw is eerst het geheel heel grof bouwen. Vervolgens ga je steeds meer verfijning aanbrengen. Het voordeel bij deze aanpak is dat je vroeg in het begin kunt wijden aan de de programmering van je product. Hierdoor kun je vrij snel een functionerende schets krijgen van je product en vroeg inzicht krijgen waar je programmering spaak loopt. Als je dit goed aanpakt en je werkt wellicht met meerdere mensen dan houdt de verfijning vaak in dat je alleen nog maar de grafische onderdelen hoeft te vervangen. Dit zou een ander persoon kunnen doen waardoor de programmering gewoon door kan werken. Tevens kan deze bouw methode voordelig zijn als je je opdrachtgevers op de hoogte dient te houden. Vooral als je opdrachtgevers minder kennis hebben met betrekking tot het computer medium is dit voor hen gemakkelijker om een helder idee te krijgen welke kant jij op wilt.

De nadelige kanten zitten meer in de keuze van de

techniek die je gebruikt voor het bouwen van je product. Een probleem bij vele pakketten is het importeren en bij houden van externe bestanden. Er zijn pakketten die dit niet of nauwelijks doen. Vaak zijn het de professioneler en volwassenere pakketten die dit zeer goed kunnen. Deze pakketten gaan er echter wel vanuit dat de gebruiker een gedegen kennis van zaken heeft. Makkelijker gezegd zijn die pakketten minder gebruiksvriendelijk en minder toegankelijk. Voorbeelden zijn de zogenaamde IDE's (Integrated Development Environment) voor C/C++, Java, C#, python etc.) Pakketten die slecht met versie beheer overweg kunnen zijn bijvoorbeeld Virtools en Flash. Het is ook geen wonder waarom bepaalde pakketten nog altijd voor de grote producties gebruikt worden. Van Virtools wordt wel gezegd dat het door game fabrikanten gebruikt wordt om schets versies van hun game te maken. Dit omdat Virtools je instaat stelt om gemakkelijk en snel iets te bouwen. De uiteindelijke game wordt meestal in een robustere omgeving gebouwd. Het komt er eigenlijk op neer dat wil je een iteratieve bouw gaan doen dan zul je een structuur binnen je bouw proces moeten toepassen die aansluit bij je gekozen software pakket. Een investering in de kennis over hoe een pakket dit zelf aanraadt is absoluut een pre. Met name als je ook met meerdere mensen tegelijk aan het project werkt.

4.2 Incrementele bouw

Incrementele bouw houdt eigenlijk in dat je begint bij het begin en vervolgens steeds verder uitbouwt. Er valt wat voor deze methode te zeggen maar het lijkt mij dat je te gemakkelijk in valkuilen terecht komt. Ondanks dat ik geen aanhanger van deze bouwmethode ben heb ik hem wel vaak toegepast. Dat ik dit vaak toegepast heb is niet altijd met instemming gegaan. Volgens mij sluipt deze bouw methode erin zodra je het overzicht over je project kwijt raakt. Dit gebeurt vrij gemakkelijk in een ongedisciplineerde

werksfeer. Op zich kan de incrementele bouw goed werken. Echter er ontstaat gemakkelijk een gevaar als er weinig structuur toegepast wordt in het bouwproces. Het begin van je project gaat er prachtig uit zien maar als je verder in het project vordert blijven er steeds meer mankementen zitten. Dit gaat vaak gepaard met het onderschatten van de tijd die het kost om bepaalde onderdelen van je project te bouwen. Algauw kom je tijd te kort. Dan sluipen er ook nog gebreken in die je steeds ter plekke moet opvangen. Ik kan een incrementele bouw eerder aanraden als dit goed bij het creatieve proces van je project past. Bijvoorbeeld als je een project start en bewust er niet voor kiest om al te weten waar het gaat eindigen. Als je vrij wilt blijven bepalen welke richting je project op gaat. Dan kan een incrementele bouw goed van pas komen.

4.3 Uitbesteden

Deze paragraaf vraagt eigenlijk om weinig uitleg. Als virtueel theatermaker is er bij elk project veel te doen. Een goede uitweg is dan vaak om bepaalde werkzaamheden uit te besteden. Er zijn toch altijd mensen die specifieke dingen beter kunnen dan jij. Het kost je waarschijnlijk ook veel te veel tijd om bepaalde kennis te werven. Dit is vrij simpel te berekenen. Kost het jou minstens een week of 2 om bepaalde kennis jou eigen te maken dan lijkt het verstandig om er iemand bij te halen die de kennis al heeft. Bij projecten waarbij je ook met een budget werkt is het vaak vrij simpel om uit te rekenen wanneer je iets zelf kunt doen of wanneer je iets uitbesteedt. Het hoofdstuk Financiën gaat wat verder over de geldzaken van projecten. Er is ook nog een andere goede reden voor uitbesteding. Het werkveld van de virtueel theatermaker is eigenlijk maar klein. Het is nog jong en het werk moet eigenlijk zelf gecreëerd worden. Zie het daarom ook maar als jou taak om deze industrie op gang te helpen. Zie het ook maar als jouw taak om de industrie in jouw

omgeving op gang te helpen. Dat werkt op de lange termijn alleen maar in jouw voordeel.

4.4 8 gouden regels van UI design

Als virtueel theatermaker kom je soms in de buurt van de wat traditionelere software makers. Deze software makers hebben te maken met het ontwerpen van een interface die de communicatie met de gebruiker draagt. Je kunt spreken van een dialoog tussen de gebruiker en het medium. Aangezien deze dialoog theorieën al wat vaker onderzocht zijn, is hier ook meer over bekend. Ik probeer hier de 8 gouden regels van dialoog ontwerp uit een te zetten.

1. Streef naar consistentie: Dit principe wordt vaak het snelst overtreden maar is wel een van de makkelijkste om te implementeren of te repareren. Consistentie zou op zoveel mogelijk terreinen gebruikt moeten worden. Consistentie in terminologie, menu's, help schermen, besturing, participatie etc.

2. Pas shortcuts toe bij veel gebruikte handelingen: Gebruikers die veel met je systeem werken zullen het waarderen als ze handelingen die ze vaak zullen uitvoeren door middel van een shortcut kunnen doen. Niets is zo irritant als door 3 menu's heen te moeten alvorens een bepaalde actie uit te moeten voeren.

3. Geef informatieve feedback: Voor elke operatie die een gebruiker uitvoert zou er feedback vanuit het systeem moeten komen. Veel voorkomende operaties zouden een subtiele genuanceerde feedback moeten geven terwijl operaties die veel veranderingen doorvoeren duidelijke en heldere feedback moeten geven.

4. Ontwerp feedback vensters die voortgang en afronding benadrukken: Dit was een moeilijke om te vertalen maar

waar het op neer komt is dat als je vele acties sequentieel uitvoert dat je die moet groeperen en dat je feedback venster het einde, midden, en begin duidelijk moet weergeven. Misschien simpeler gezegd zou ik zeggen dat je duidelijke feedback moet geven over de voortgang van de acties en duidelijk moet weergeven wanneer dit succesvol is afgerond. De informatieve feedback die de gebruiker krijgt bij het afronden van de sequentie van acties zal de gebruiker een gevoel van tevredenheid geven. Deze regel zal wellicht wat minder van toepassing zijn bij de virtueel theatermaker. Ik kon hem echter je niet onthouden.

5. Bied simpele fout behandeling: Natuurlijk moeten fouten voorkomen worden maar leef niet in illusies. Wat fout kan gaan zal fout gaan. Mocht dit gebeuren zorg dan dat dit dan ook vriendelijk in beeld gebracht wordt. Waar mogelijk zorg dat 1 fout niet betekent dat alles opnieuw gedaan moet worden maar geef de gebruiker de mogelijkheid om alleen hetgeen dat fout gegaan is aan te passen en opnieuw te proberen. Niets is zo irritant als al je acties compleet opnieuw te doen als alleen de laatste actie fout gegaan is.

6: Bied het ongedaan maken van acties: ook al is dit voor de virtueel theatermaker vaak wat minder relevant, waar mogelijk implementeer het ongedaan maken van acties. Dit geeft de gebruiker gevoel van veiligheid. Ze hebben het idee dat als ze iets verkeerd doen ze het in ieder geval ongedaan kunnen maken.

7: Geef gebruikers het gevoel de controle te hebben: Of het nou zo is of niet. Je publiek leeft graag met het idee dat ze de controle over iets hebben. In de traditionele software is dit een wat uitgebreidere regel. Ik laat hem hierbij aangezien wij erg van theater houden.

8: Voorkom korte termijn geheugen gebruik: Het korte termijn geheugen wordt niet graag vaak gebruikt door gebruikers. Het is erg gelimiteerd. Probeer het dus zoveel mogelijk te vermijden. Kort gezegd: mensen onthouden liever niets.

Dit zijn zo ongeveer de 8 gouden regels van het ontwerpen van interfaces zoals ze ooit door B. Shneiderman opgesteld zijn. Ze zijn behoorlijk gericht op de software die als gereedschappen gebruikt worden. Als virtueel theatermaker is het toch slim om ze te kennen. Al is het alleen al voor de menu's van de het project dat je bouwt.

5 Project Management

Hoe je het ook wendt of keert er zal brood op tafel moeten komen, er moeten rekeningen betaald worden en af en toe moeten we een biertje drinken. Een hoofdstuk getracht te weiden aan het aanpakken en afronden van projecten en de financiële kant van het werk. Ik heb vele artikelen gelezen en als je bijvoorbeeld alleen al het artikel “Object-Oriented Game Development: Iterative Development techniques” van Julian Gold leest dan merk je wel dat er simpelweg heel veel structuur nodig is om een redelijk groot game project succesvol uit te voeren. Dit geldt vanzelfsprekend ook voor een project van de virtueel theatermaker. Bereid je dus voor op veel structurele aanpak.

5.1 Hoeveel tijd kost het

Tijdens mijn afstudeerjaar kwamen wij tijdens een groepsproject af en toe bij een bedrijf over de vloer. Zij grapt dat bij alles wat zij planden zij de zogenaamde “Pi” regel toepasten. Deze regel hield in dat bij het inschatten van de benodigde tijd zij deze tijd altijd met getal Pi vermenigvuldigden. Grappig genoeg werd die regel al gauw geassocieerd met hun chaotische bedrijfsvoering maar uiteindelijk moet ik wel concluderen dat die regel vaak goed toe te passen is.

Er valt niet omheen te komen maar vaak zijn de projecten van de virtueel theatermaker projecten die enorm veel werk inhouden. Hoe ambitieuzer hoe meer arbeid. Het moeilijkste in dit vakgebied is toch om een heldere inschatting van het aantal werkuren te maken. Ik zou eigenlijk niet weten hoe je goed het aantal werkuren voor het project kan inschatten. Ik weet alleen wel dat het mij nog nooit gelukt is om onder het

aantal ingeschatte uren te blijven. Meestal ga ik er ver overheen. Dit is erg vervelend want vaak is het zo dat die extra uren niet betaald worden. De “Pi” regel van hierboven is wel een regel die je ruimte biedt voor de onvolkomenheden van je schatting. Maar zeg nou zelf, je inschatting maal 3... Volgens mij mankeert er dan echt wat aan je inschatting. Toch, hanteer hem nou maar in het begin. Naar mate je meer ervaring krijgt zul je steeds verfijnder kunnen inschatten en zal de “Pi” regel niet meer van toepassing zijn.

Maar wat maakt toch dat ik de ervaring heb dat veel projecten altijd meer tijd kosten dan eigenlijk initieel gepland was. In het artikel van Timothy Ryan komt wel naar voren dat de hele industrie hier last van heeft en dat het veelal genegeerd wordt. Toch zegt hij ook dat de geldschieters dit niet meer accepteren. In mijn zoektocht hiernaar kwam ik maar tot 1 conclusie. Chaos, slechte planning, slechte communicatie, slecht management, weinig reflectie, weinig structurering, moet ik door gaan? Oh nee ik had maar 1 conclusie. Ik ben tijdens 6 jaar DVTG toch vaak op dezelfde mythe gestoten. Structuur is dodelijk voor de creativiteit. Die mythe is grotendeels niet waar. Ik heb zelf 1 project meegemaakt waarbij de gehele groep iets te gestructureerd werd. Alles werd zo ver uitgedacht dat uiteindelijk alles ontzettende bedacht en gemaakt werd. Dat ging hartstikke mis. Hierbij was het inderdaad waar dat de structuur dodelijk was voor de creativiteit. Maar het is ook niet de bedoeling dat je structuur in de creativiteit aanbrengt. Dat is inderdaad dodelijk. De enige reden waarom je structuur in je project aanbrengt is om het project in goede banen te geleiden zodat je op een prettige wijze bij een eindproduct aankomt. De structuur moet niet gaan ingrijpen in creatieve processen die je concept sturen. Helemaal niet. De structuur brengt dat je afspraken nakomt, dat je onderdelen op tijd af hebt, dat je onvoorziene omstandigheden opvangt. Dat je weet wat een ieder in je project groep aan het doen is en wat een ieder doen moet. Tijd is geld en je tijd moet je goed gebruiken net als geld. Een professionele houding houdt in dat je goed de creativiteit en structuur weet te gebruiken en te bewaken.

5.2 Vergaderen

Het uitvoeren van project vergt nogal wat taken. Omdat het werk van de virtueel theatermaker nogal wat inhoudt zal het waarschijnlijk vaak met meerdere mensen gedaan worden. Het toepassen van bedrijfsstructuren is wellicht een goed idee. Niet omdat we daar allemaal zo creatief van worden maar om te garanderen dat het werk gedaan wordt. In een wat grotere groep is een vergadering hiervoor zeer doeltreffend. In een kleinere groep is een werkbepreking misschien voldoende. Belangrijk aan het moment van reflectie en terugblik is:

1. Vast tijdstip: Dit tijdstip is bij iedereen van te voren bekend. Het makkelijkst is bijvoorbeeld gewoon elke maandag ochtend bijvoorbeeld.

2. Zakelijk: Bij voorkeur is het moment zakelijk en sober van opzet. Ik bedoel niet dat je als een stel zombies rond een tafel moet zitten. Als het gezellig is en er gelachen wordt is dat natuurlijk prima. Echter vaak kunnen de momenten van reflectie en terugblik met heftige emoties gepaard gaan. Niet iedereen kan even gemakkelijk omgaan met kritiek. Het is daarom belangrijk om de emoties niet de voortgang te laten verstrikken. Pas ook op dat het moment teveel over de negatieve dingen gaan. Neem je voor om ook de dingen die goed gegaan zijn te benadrukken. Kortom een emotionele balans.

3. Hiërarchisch: Bij een grotere groep is er vaak wel een rolverdeling. Mocht dit niet zo zijn zorg ervoor dat in ieder geval tijdens het moment er 1 iemand is die bewaakt dat iedereen gelijkwaardig aan woord komt en dat er niet eindeloos op bepaalde onderwerpen doorgegaan wordt. Deze persoon krijgt de autoriteit van alle aanwezigen om knopen door te hakken of om onderwerpen op te schuiven naar een ander moment. Hij leidt het gehele moment.

4. Gedocumenteerd: Dit is eigenlijk meer van belang bij

vergaderingen en de wat grotere project groepen. Bij een vergadering is altijd een agenda aanwezig welke minstens 1 dag van te voren voor iedereen bekend is. Op deze manier kan iedereen zich voorbereiden op hetgeen besproken moet worden en zal dit een vlotte voortgang van de vergadering begunstigen. Naast de agenda wordt van elke vergadering notulen opgesteld. Deze notulen worden zo snel mogelijk na de vergadering aan iedereen uitgedeeld. Het gaat er eigenlijk om dat er afspraken gemaakt worden en dat deze vastgelegd worden. Op deze manier staat gewoon vast wat een ieder doen moet en is dit verifieerbaar. Dit voorkomt bijvoorbeeld discussies over wat mensen doen moesten of dat men niet wist wat men doen moest. Vooral met externe opdrachtgevers is het ten eerste aan te raden om de afspraken goed vast te leggen om veel ellende bij conflicten te voorkomen. Naast de agenda en notulen is het ook nog aan te bevelen om een actielijst bij te houden. Dit is een lijst met alle acties die naar voren komen tijdens de vergadering. Deze lijst bevat in ieder geval de beschrijving van de actie, de naam van de persoon die de actie uit zal voeren en voor wanneer dit gedaan moet worden. Deze lijst kan vervolgens bij elke vergadering bijgehouden worden waardoor gemakkelijk inzichtelijk kan worden of alles op tijd gebeurt.

De vergadering is in wezen een saai zakelijk momentje maar verschrikkelijk belangrijk. Het belang van de vergadering zit niet zozeer in het accelereren van het werk maar eerder in het voorkomen van deceleratie en stranden van het project. Mijn ervaring was vaak dat binnen mijn opleiding het nut van vergaderen niet altijd serieus genomen werd en ook niet ingezien werd. Eigenlijk kwam het besef van nut van een vergadering en structurele aanpak pas vaak als het al drastisch te laat was. Ik kan me eigenlijk binnen mijn studie bijna geen vergadering herinneren waarbij iedereen van te voren zich voorbereid had met de agenda en de notulen van de vorige vergadering gelezen had. Uiteindelijk moet je het natuurlijk zelf weten hoe je je project aanpakt. Al doende leert men en ik denk wel dat je hier uiteindelijk uitkomt. Er is ook zoiets als een kwaliteit om te

kunnen herkennen wanneer je opnieuw het wiel gaat uitvinden.

Last but not least, omdat een vergadering vaak erg saai is, is het aan te raden om het ietsjes gezellig te maken. In mijn geval doet een bak koffie en koekje al wonderen. Sfeer is te maken!

5.3 Planning en timetracking

Vanzelfsprekend komt bij het uitvoeren van een project de planning om de hoek kijken. Het is meestal zo dat je maar een beperkte tijd tot je beschikking hebt dus zul je die tijd nuttig moeten indelen. Als je alleen werkt is dat best wel te doen. Je houdt een aantal deadlines goed in de gaten en je werkt zo hard mogelijk daar naar toe. Bij grotere projecten en projecten waarbij je met meerdere mensen werkt wordt dit al gauw wat complexer.

Voor ik verder over planning door ga wil ik eerst een belangrijke eigenschap benoemen waarvan ik vind dat je die jezelf moet aanleren. De eigenschap om een deadline te kunnen voelen als hij er misschien in werkelijkheid niet is. Misschien ken je het gevoel van de deadline wel. Je opdrachtgever heeft gevraagd om op die en die datum het geheel op te leveren. Een week van te voren voel je die deadline vaak wel. Helemaal als je nog verschrikkelijk veel doen moet. Echter de afspraak die je in je project groep gemaakt hebt om modelling af te hebben op Woensdag de 3e die voel je niet. Een dag van te voren kom je er eigenlijk pas achter dat het wat langer gaat duren en dat die deadline maar even opgeschoven moet worden. Moet kunnen toch? Nog 3 dagen heb je nodig. Maar omdat het weekend er tussen zit lever je dus de modelling op op de Dinsdag. Ga maar na 3 dagen extra. Dat is Donderdag, Vrijdag, Maandag en dus de oplevering op Dinsdag. Al met al een week later

dus. Alleen de mensen die de textures voor de modellen maken lopen nu ook een week achter en ook zij hebben wat meer tijd nodig dan gepland aangezien er aantal kleine foutjes zaten in de UV mappen van de modellen. Slordigheidje, sorry. Kan gebeuren. Nu zijn de textures pas 2 weken later af en loopt programmering 4 weken achter aangezien er vreemdgenoeg verschillen waren ontstaan tussen de modellen van de animatoren en de modellen van texture mensen. Eigenlijk kan dat helemaal niet maar vreemd genoeg is het wel gebeurd, heel vreemd. Het punt wat ik wil maken is dat je jezelf zou moeten aanleren om een afspraak na te komen. Een deadline die je jezelf of door de groep oplegt moet je ook als deadline behandelen. Als je die deadline niet gaat halen dan zie je dat waarschijnlijk ruim op tijd aankomen en kun je dit communiceren en opvangen wellicht.

Er zijn vele methoden om te plannen. Het gaat te ver om diep in te gaan op allerlei soorten manieren van plannen. Het hangt een beetje af van de manier van werken van jezelf en van de project groep. Voorop staat in ieder geval dat er een globale planning is waar iedereen van de project groep van op de hoogte is. In deze planning staan alle groepsactiviteiten gepland. Werkbesprekingen, vergaderingen, presentaties, repetities, deadline momenten etc. Afhankelijk van je manier van werken kun je ook in de planning de sleutelactiviteiten van de dag opnemen of bijvoorbeeld de roadmap. Week 1: Concept en Design Document, Week 2: vormgeving opzet: Week 3: Modelling. Je zou ook per ontwikkel cyclus een planning kunnen maken mits dit aansluit bij het type project je uitvoert. Bijvoorbeeld een planning voor de animaties, modelling, texturing, programmering etc. Je kunt ook een individuele planning maken. Dit is met name handig om jezelf langs een lopende band aan werkzaamheden te loodsen. Dit hangt een beetje af van je eigen discipline om het werk gedaan te krijgen.

Maak in ieder geval een helder overzicht van alle taken in de planning en verdeel taken in prioriteiten. Wat je ten allen

tijde wilt voorkomen is essentiële project taken uit te voeren op het laatste moment. Toch, zo zegt ook Julian Gold in zijn artikel, is dit helaas nog steeds gebruikelijk.

Een ander punt wat in het verlengde ligt van de individuele planning is timetracking. Naast de planning kan het handig zijn om bij te houden hoeveel tijd je aan activiteiten besteed. Dit geeft je op de lange termijn inzicht in hoeveel tijd je aan projecten of specifieke activiteiten kwijt bent. Dit is op zijn beurt weer een erg waardevolle kennis als het aankomt op inschatten hoeveel tijd je denkt te besteden ergens aan. Op die manier kun je inschatten wat de kosten van een project zijn of hoeveel je kunt doen voor een bepaald budget. Op korte termijn kun je een overzicht krijgen van waar de tijd heen slurpt en dat inzichtelijk maken. Je kunt je tijdsbestedingen natuurlijk globaal bijhouden op een papiertje of in je agenda. Er zijn ook verschillende hulpmiddelen hiervoor. Deze komen aan bod in de volgende paragraaf.

5.4 Het wiel en de community

Je bent niet de enige die soortgelijke projecten doet. In soortgelijke werkvelden, met name de zakelijke, is al veel kennis opgedaan met betrekking tot projecten uitvoeren en ten einde brengen. Er is tegenwoordig een gemeenschap van mensen die ontzettend veel kennis deelt met iedereen. In plaats van alles zelf uit te zoeken kun je veel van die gemeenschap leren. Met name de autonome virtueel theatermakers doen er mijns inziens goed aan om van deze gemeenschap gebruik te maken en er wellicht aan te contribueren. Uit deze gemeenschap zijn vele hulpmiddelen ontstaan die wij vaak goed kunnen gebruiken. Ik wil een aantal hulpmiddelen behandelen die makkelijk toegankelijk zijn en je werk uit handen kunnen nemen of ondersteunen. Tevens zijn deze gratis en legaal beschikbaar.

Website:

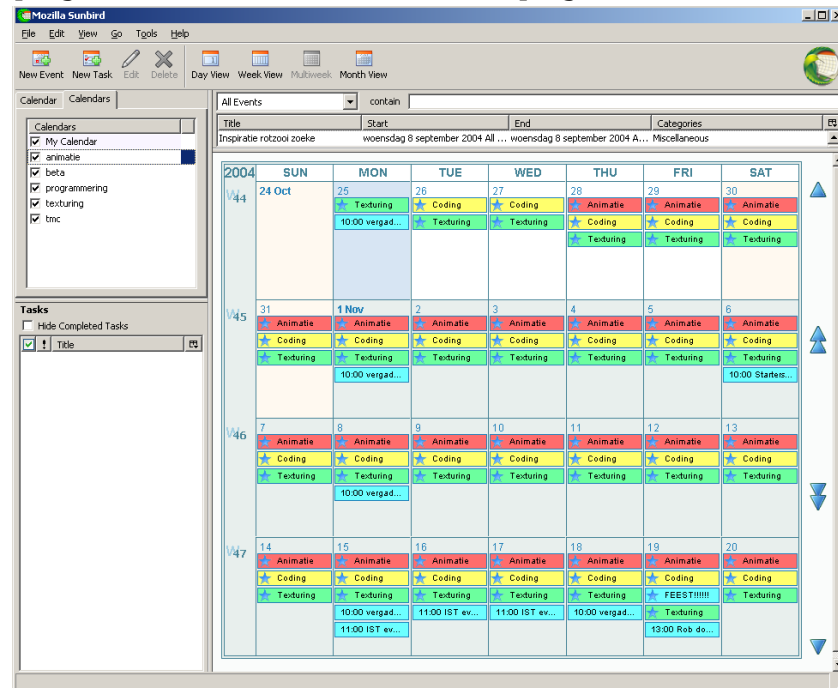
Vele project groepen zetten tegenwoordig een website voor het project op. Deze website is vaak de centrale plek met informatie over het project voor de betrokkenen en geïnteresseerden. In plaats van zelf een website te knutselen is het handig om een Content Management Systeem op de site te plaatsen. Dit biedt vele voordelen zoals een consistente vormgeving, gemakkelijk te updaten, gebruikers accounts en vele meer ingebouwde functionaliteiten. Er zijn hier zoveel keuze mogelijkheden dat het onmogelijk is om hier verschillende te behandelen. De keuze in CMS is ook sterk afhankelijk van de behoefte en het soort project wat je uitvoert. Als je alleen een informatieve website wilt voor bijvoorbeeld opdrachtgevers zodat zij de voortgang kunnen volgen kom je zeer waarschijnlijk bij een heel ander CMS uit dan wanneer je met een internationale groep mensen verspreid over de planeet een project uitvoert. Het is aan te bevelen om je hier eens in te verdiepen. Als je eenmaal een systeem kent blijf je daar vaak bij. Zo ben ik ooit aan het Mambo systeem begonnen en daar maak ik nog steeds gebruik van. Op de website <http://www.opensourcecms.com> kun je verschillende systemen proberen.

Eén CMS wil ik toch benadrukken daar deze een niet geringe omvang kent. Hij gaat rond onder de naam Wiki wat inmiddels een begrip geworden is. De bekendste wiki is wel die van wikipedia (<http://www.wikipedia.org>) Het principe van Wiki is dat iedereen de inhoud kan bewerken en inhoud kan toevoegen. Wikipedia is in dit geval een online gratis encyclopedie. De software achter de wiki's is vaak vrijelijk beschikbaar. Vele projecten passen dit principe bij de documentatie van hun product toe.

Planning & Kalender:

Een planning bijhouden is in project groepen vaak onontbeerlijk. Hiervoor zijn vele systemen ontwikkeld alleen wat minder vaak zijn deze systemen vrijelijk beschikbaar. Deze behoefte kan ook heel goed door een CMS opgevangen

worden. Echter de grootte van projecten zijn vaak niet toereikend om alles via het web te doen. Immers een zeer uitgebreid CMS vergt ook wel een webserver die wat aan kan. Ruimte op een webserver is vaak niet gratis beschikbaar. Voor een goed kalender programma zou ik het programma SunBird aanraden (of de plugin voor FireFox of



Illustratie 2: Sunbird

ThunderBird). Dit programma is gratis te downloaden vanaf <http://www.mozilla.org> echter bij schrijven nog wel in beta fase maar ondanks dat goed te gebruiken. Dit programma is tevens goed uit te breiden met gedeelde kalenders. Dit maakt het mogelijk om 1 kalender met meerdere gebruikers te delen. Hiervoor heb je wel een webaccount nodig met DAV ondersteuning. Meer informatie hierover is op de website van Sunbird te vinden.

Spreadsheets:

In onze ontzettend gestructureerde aanpak kan een simpel spreadsheet programma ons geweldig assisteren. Iedereen kent spreadsheets wel. Mocht het niet zo zijn, het is gemakkelijk te leren. Wat men vaak niet weet is dat het befaamde Excel van Microsoft officieel een flinke duit kost. Gezien ons kleine budget en onze angst voor de BSA kijken we naar een alternatieven. De meest bekende gratis Office Suite is OpenOffice. <http://www.openoffice.org>

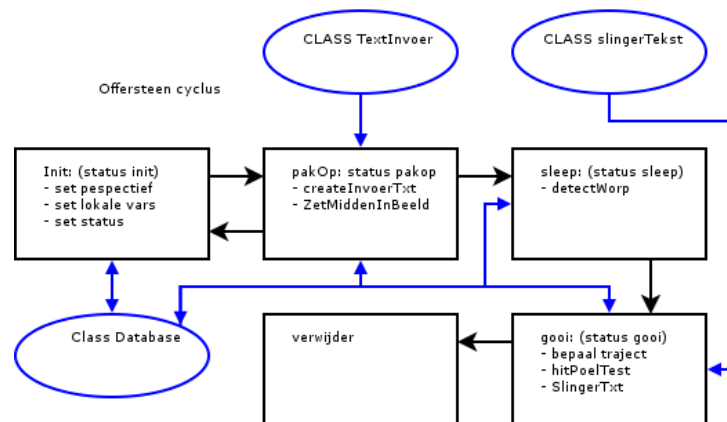
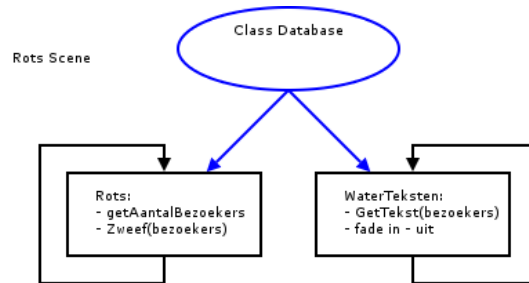
Mailinglist:

De communicatie met de mensen in de groep wordt vaak via email gedaan. In plaats van iedereen een CC-tje te mailen kun je ook een mailinglist opzetten. Het voordeel van zo'n systeem is dat alle correspondentie over de mailinglist centraal opgeslagen wordt. Er gaat dus geen mailtje verloren. Een ander gemakkelijk voordeel is dat een beantwoord mailtje teruggaat naar de mailinglist en dus naar iedereen. Het gebeurt namelijk vaak dat als een normaal verstuurd mailtje met allemaal CC's beantwoord wordt het antwoord alleen naar de verzender van het originele mailtje gaat. Er zijn op internet verschillende plaatsen waar je gratis een mailinglist kan opzetten. Een nadeel daaraan is de reclame die je er vaak gratis bij krijgt. Wil je het goed doen dan zet je natuurlijk zelf een servertje op met een mailinglist manager. Google levert vele hits op.

Diagrammen:

Bij de wat technisch ingewikkeldere projecten is het vaak aan te bevelen om eerst een flowchart te maken. Hierbij maak je een schematische weergave van hoe bijvoorbeeld je interactiviteit in je product verloopt. Het voordeel van deze diagrammen is dat je van te voren helder krijgt hoe je het moet programmeren en dat je dit kunt tonen en uitleggen aan anderen die wellicht mee programmeren. Er zijn ontzettend veel standaarden om officiële diagrammen te maken. Echter ik heb zelf nooit de noodzaak gezien om me aan deze standaarden te houden terwijl ik diagrammen wel broodnodig

vond. Wellicht zal bij grotere projecten het belang van de officiële standaarden wel mee spelen. Er zijn maar weinig applicaties die in de behoefte van diagrammen voorzien. De enige die gratis beschikbaar is is het programma Dia.
<http://www.gnome.org/projects/dia/>



Illustratie 3: Dia diagram

Bugzilla:

Ga je ooit een groots project uitvoeren. Wordt daar veel bij geprogrammeerd en kan er veel misgaan? Bugzilla biedt uitkomst als een geavanceerd bugtracking systeem. Het is

wederom een product van de Mozilla Foundation die onder andere Firefox en ThunderBird op de markt brachten. Bugzilla zul je niet snel gebruiken in een klein project maar je zult hem eerder tegenkomen als je zelf met software van anderen aan de slag gaat. <http://www.bugzilla.org>

Subversion/CVS:

Ook dit is software dat je waarschijnlijk alleen zult gebruiken bij de grotere projecten waar veel programmeer werk bij komt kijken. Subversion en CVS zijn versie controle systemen. Eigenlijk houdt dit systeem alle veranderingen van data bij. Dit wordt veel bij tekstbestanden met programmeer code gebruikt maar in principe is het voor elk type bestand te gebruiken. Alle veranderingen zijn te bekijken en terug te draaien. (<http://subversion.tigris.org> , <http://www.nongnu.org/cvs/>)

6 Conclusie

Het einde is in zicht. Wat heb ik nu eigenlijk willen zeggen met deze scriptie. Ik heb het veel over de techniek gehad en over het computer medium als extensie van de virtueel theatermaker. Ik belandde in allerlei structuren en tips. Waar het uiteindelijk op neer komt is dat je niet onder de structuur uit komt. De structuur is mijns inziens onmisbaar in projecten van de virtueel theatermaker. Er simpelweg zoveel te doen en zoveel uit te zoeken. Als je daarnaast ook nog enigszins vrij artistiek in het project bezig wilt zijn dan zul je wel een stevige dosis structuur toe moeten passen. Artistiek zijn houdt in dat je voorbij aan de techniek gaat. Voorbij aan de techniek gaan houdt in dat je de techniek beheerst. De techniek beheersen houdt aandacht en oefening in. Dit alles samen houdt in dat je tijd maar schaars is. Neem van mij aan dat een juiste structurele aanpak je creativiteit ten goede komt doordat je meer artistieke ruimte schept. Tevens is het fijn als je de techniek goed beheerst omdat het je veel frustratie kan schelen. Frustratie is vaak niet goed voor je creativiteit.

Er is nog zoveel te doen in de virtueel theatermakerij. De opzet van deze scriptie was niet om het diepe in te gaan met het computer medium. Ik denk dat er knappe koppen genoeg rond lopen die prachtig kunnen beredeneren hoe de toekomst van het computer medium eruit zal zien. Er zijn volgens mij maar weinig knappe koppen die daadwerkelijk ook die toekomst alvast bouwen. Ik zie hier een toekomst weggelegd voor de virtueel theatermaker. Er kan nog zoveel leuks gedaan worden met het computer medium. Er kan nog zoveel leuks gedaan worden met de technieken die het computer medium nu kent. De toekomst van de virtueel theatermaker speelt zich af op het artistieke terrein van het computer medium en niet op het technologische. Mijn hoop is dat deze scriptie getoond heeft dat er behoorlijke discipline nodig is als virtueel theatermaker. Discipline om het computer

medium te beheersen en eigen te maken. Discipline om de technieken te leren gebruiken. Ik hoop tevens dat deze scriptie de virtueel theatermaker over de drempel heen zet van de gestructureerde aanpak mocht die drempel er zijn. Wellicht zou deze scriptie een praktisch handvat kunnen zijn voor de beginnende virtueel theatermaker. Misschien dat mensen van buiten het werkveld van de virtueel theatermaker een kijk krijgen in de keuken van de virtueel theatermaker. Bovenal was deze scriptie voor mij een kristallisatie van de vele processen die ik tijdens de jaren bij DVTG heb door gemaakt. Dit document beschrijft voor mij op welke manier ik virtueel theater projecten zou uitvoeren en met welke redenen en gedachten.

Nu zit ik hier op de camping in Makkum. Om het idyllisch te maken zakt de zon net het IJsselmeer in net zoals mijn 6 jarig DVTG studentschap. Ik heb helemaal geen zin het verhaal te eindigen. Er kan nog zoveel toegevoegd worden. Ik heb niet het gevoel dat het klaar is. Nu je toch weet wat het is; misschien moet ik dit document maar omzetten naar een Wiki.

Ten slotte voor de mensen die op de uitnodiging van koffie drinken in willen gaan:

Arnaud Looonstra
Wolvenstraat 80
3512CH Utrecht

<http://www.sphaero.org>

7 Bronnen

- Flash MX 2004 – training from the source
Derek Franklin/Jobe Makar
 - Zen en de kunst van het Motoronderhoud
Rober M. Pirsig
 - Zen en de kunst van computergebruik
Philip Toshio Sudo
 - 3 jaar acquisitie van elektrische muziek instrumenten
Haags Gemeentemuseum
 - Graphical User Interface Design and Evaluation
David Redmon-Pyle & Alan Moore
 - Death to the Games Industry
www.escapistmagazine.com
 - Everything is possible
Greg Kasavin
www.gamespot.com/features/6120427/index.html
 - Moog (2004)
Hans Fjellestad
 - CLOSE UP – Matisse en Picasso
Philippe Kohly
 - Gesprekken met Rui Guerra
 - Wikipedia – www.wikipedia.org
- Gamasutra.com
- Manager In A Strange Land: Most projects Suck
Jamie Fristrom
 - Object-Oriented Game Development: Iterative Development Techniques
Julian Gold
 - Pedersen's Principles on Game Design and Production
Roger Pedersen
 - Moving Up in the World: How Artists Can Become Game Development Leaders
Di Davies
 - Risk Management With Development Schedules
Timothy Ryan

- Artists and Game Design Documents: From Interpretation to Implementation

Joshua D. Gordon

Met dank aan:

- Marcel Dolman
- Jorrit van de Langerijt