

START

HERE

GAMEDESIGN VOOR BEGINNERS

VOORWOORD

LET THE GAME BEGIN

Even serieus lezer: ga toch spelen.

Als mensen spelen, oefenen ze zich in handigheid, inzicht, reactievermogen en sociaal gedrag. Niet op leven en dood, want de spelomgeving is altijd niet-echt, een niet-werkelijkheid. Het spel wordt daarom wel eens als triviaal gezien, niet zo echt als 'echt'. Onzin natuurlijk. Wie niet oefent, baart geen kunst. Een spel traint je in het leven zelf, of zoals Johan Huizinga schreef in zijn Homo Ludens: er moet iets 'lukken', met zekere inspanning.

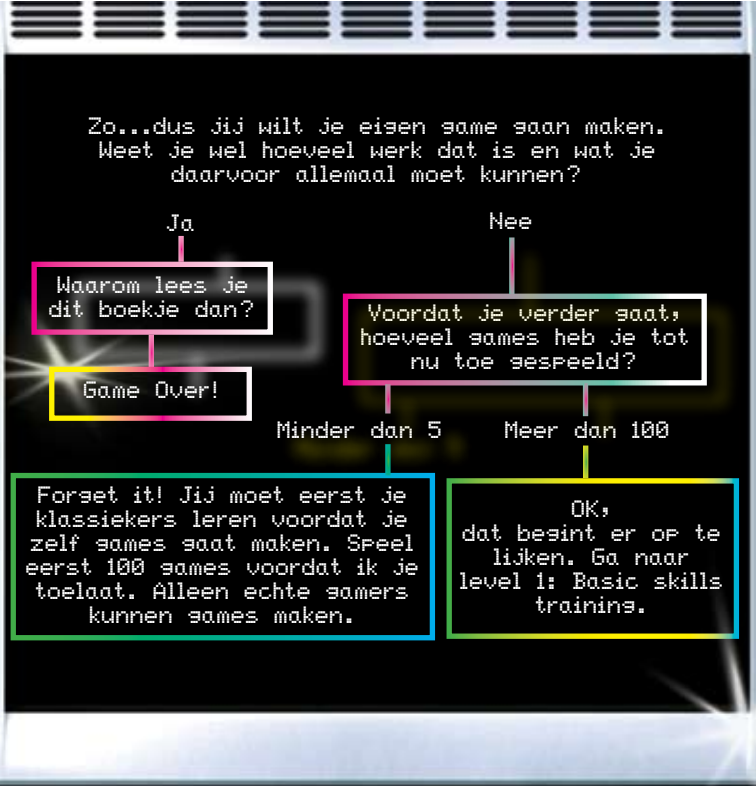
Computergames zijn dominante spelvormen van nu. Gamedesign is een multidisciplinaire activiteit, die een beroep doet op logisch denken, fantasie, ontwerp skills, samenwerking en communicatie.

Veel niet-spelers worstelen met de vraag of games slecht zijn voor de geest, leiden tot sociaal isolement en zo. Het antwoord is simpel: de beste manier om daar invloed op uit te oefenen, is het zelf te doen. Doe mee, speel mee, maak samen games, en discussieer over de ideale game. Leer spelenderwijs, spelend en wijs.

Dit boekje is een lekker licht maar bruikbaar vertrekpunt om daarmee te beginnen. De principes van gamedesign, de achtergrond, de technieken. Alle tools binnen handbereik.

Ga verder naar Start.
Start Here.

Ben Kamphuis
consulent digitaal



PS. Alle genoemde links zijn clickable op www.amateurkunst.net/starthere



Photoshop



Paint Shop



The Gimp

NOTEN

1...alhoewel, speciaal voor blinden worden er ook games gemaakt die helemaal op geluiden zijn gebaseerd! Zie o.a. <http://www.accessibility.nl/algemeen/nieuws?id=51>

2. Zoals Gordon Ramsay zegt in Ramsay's Kitchen Nightmares: "there are too F#@ \$ing many items on the menu."

3. Tenminste dat beweert Robert Langdon in de Davinci Code.

4. Volgens de verkoopster van de Prénatal.

00:4

LEVEL 1: BASIC SKILLS TRAINING GRAPHICS GEREEDSCHAPPEN

Goed, je hebt besloten om een eigen game te gaan maken. Eens kijken of je al genoeg vaardigheden hebt... Voor een videogame moet je de volgende elementen beheersen:

- ★ Graphics
- ★ Sound
- ★ Concept
- ★ Programmacode

LEVEL 2: GRAPHICS

Zonder graphics geen videogame....¹ Alle elementen in je game hebben een uiterlijk nodig: helden, monsters, ballen, geweren, bomen, gebouwen, ruimteschepen, enzovoort, enzovoort. Maar wat dacht je van de achtergronden (wolken, sterrenhemels, grasgronden, enzovoort)? Ook die moeten gemaakt worden.

Daarnaast heb je de bewegende graphics in je game: roterende sterren, stuiterende botsauto's, groeiende bloemen, ontploffingen en dergelijke. Deze zogenaamde animaties zul je moeten maken. Hoe?

[Go to Level 3](#) : Animatie

Oh ja, dan heb ik het nog niet over 3d gehad! Je wilt een 3d game maken....hmmm ambitieus....goed!

[Go to Level 8](#) : 3d, maar misschien is het verstandig om eerst alle basic skills te leren.

Er moeten dus een hoop plaatjes geproduceerd worden voordat je game er een beetje uitziet. De graphics zijn de bouwstenen die je later met programmacode 'tot leven' gaat wekken. Voor het maken van graphics zijn er talloze softwarepakketten beschikbaar. Van gratis pakketten tot hele dure professionele pakketten.

Een bekend open source (gratis) pakket is: the Gimp. Bekende commerciële grafische pakketten zijn: Photoshop, Illustrator, Paint Shop pro en Fireworks. En er is nog veel meer beschikbaar. Het valt een beetje buiten het kader van dit boekje om al deze pakketten en hun voor- en nadelen te bespreken. Kijk goed op de forums van het internet wat mensen zeggen over bepaalde pakketten of vraag aan vrienden wat zij gebruiken.

Er zijn talloze cursussen, boeken, online tutorials om de pakketten te leren beheersen. Kies een van bovenstaande pakketten, leer er mee werken en ga aan de slag.

GRAPHICS STIJLEN

Als je een game gaat maken, is het van belang na te denken over welke vormgeving je de game wilt geven. Wellicht word jij de bedenker van een nieuwe stijl, maar tot die tijd kan je je game baseren op een van de bekende basis game stijlen:

- ★ realistische stijl
- ★ cartoonstijl
- ★ vectorstijl
- ★ retro stijl

Belangrijk is dat je de stijl consequent toepast. Als je de stijlen gaat mixen op je scherm, ziet dat er niet goed uit.

Stijl en vormgeving zijn een kwestie van smaak en daarover valt - zoals je weet - niet te twisten. Toch nog een paar tips van experts dan:

☆ Minder is beter: Als je game er niet goed uitziet is dat vaak omdat het design te druk is. Er zijn teveel objecten in het scherm, teveel verschillende vormen en/of teveel kleuren gebruikt.²

☆ Zorg dat je schermindeling voldoet aan de gulden snede: De gulden snede zijn verhoudingen die vaak in de natuur terugkomen. De reeks loopt als volgt: 1,2,3,5,8,13,21,... Met andere woorden als jij de vlakken op je scherm de verhouding 1:2 of 2:3 of 5:8 laat hebben zien ze er "mooi" uit.³

☆ Jongens houden van blauw en meisjes van roze⁴: Dit is natuurlijk onzin, maar het is wel zo dat de voorkeur voor kleuren afhankelijk is van leeftijd, sexe en culturele achtergrond. Hou dus rekening met de doelgroep van je game.

☆ Beter goed gejat dan slecht bedacht: Dit geldt niet alleen voor graphics maar ook voor sounds en ideeën. Echter, let op! Een idee mag je altijd jatten, maar een graphic of een sound niet. **Go to Level 18** : Je game publiceren voor meer informatie daarover.

Het zijn zo maar een paar tips. Er is nog veel meer over design te zeggen.⁵ Het beste is om gewoon te beginnen en de gamekarakters te schetsen. Of je dat nu op papier doet en dan later inscant of dat je direct in een programma als the Gimp of Photoshop gaat schetsen. Als je echt niet kan of wilt tekenen, kan je op het internet ook heel veel gamefiguren vinden.⁶ Let wel goed op of je ze mag gebruiken.

Go to Level 11 : Copyrights. Bij twijfel altijd schriftelijk toestemming vragen aan de maker.

Voor het laten bewegen van de objecten in je game heb je animatietechnieken nodig.

Go to Level 3 : Animatie.

LEVEL 3: ANIMATIE

Animaties ontstaan door een paar 'stilstaande' plaatjes snel achter elkaar af te spelen. Als je een filmrol bekijkt, dan zie je een hele hoop dia's na elkaar. Als je die snel afspeelt dan lijkt het of de karakters bewegen. Je moet dit wel sneller doen dan je oog kan waarnemen. Om te voorkomen dat je animatie knippert, wordt wel een snelheid van 30 plaatjes per seconde aangehouden.



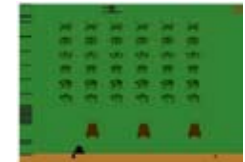
Realistische stijl (beetje fotografisch)



Cartoonstijl (lijkt op stripboeken)



Vectorstijl (oude SF films)



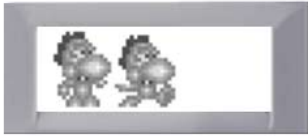
Retrostijl
(als vroeger op de Atari2600)

5. Zie o.a. Designing Arcade Computer Game Graphics, door Ari Feldman
Te downloaden van <http://www.gamemaker.nl/feldman.html>

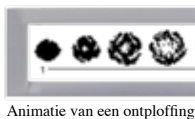
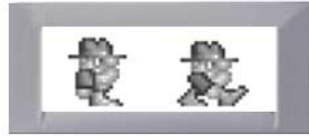
6. B.v. op <http://www.panelmonkey.org/>



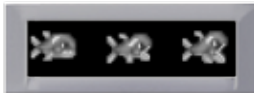
Beweging



Beweging



Animatie van een ontploffing



Animatie van een happende vis

Dat betekent dus dat je voor een lopend mannetje in je game die gedurende 1 seconde loopt, 30 plaatjes nodig hebt!

Gelukkig kan je de reeks stilstaande plaatjes wel oneindig herhalen. In bovenstaande reeks is het zo dat bij het laatste mannetje de cyclus weer van voren af aan kan beginnen. Voor bovenstaande animatie van een lopend mannetje heb je dus 12 losse plaatjes nodig.

Je kan je animaties ook met minder individuele plaatjes maken. Hiernaast loopt een mannetje door middel van maar 2 plaatjes. Het resultaat is natuurlijk veel minder realistisch, maar als je game toch een cartoonstijl aanhoudt is dat geen enkel probleem.

Alles wat beweegt in je game is opgebouwd uit een serie van stilstaande plaatjes. Zie hiernaast bijvoorbeeld een animatie van een ontploffing en van een happende vis.

LEVEL 4: SOUND

Sound oftewel geluidproductie is de tweede vaardigheid die je nodig hebt voor het maken van je game. Het is een van de onderbelichte maar erg belangrijke elementen van games. In een goede game wordt je als speler als het ware ondergedompeld. Je voelt je even als in een eigen wereld. Geluid is daar heel belangrijk bij. Kijk maar eens naar een erg emotioneel filmfragment en zet daarbij het geluid uit. Raakt je een stuk minder, niet?

Geluid in de computerwereld kent twee stromingen, namelijk midi en audio. Midi is ontstaan in de jaren '80 van de vorige eeuw toen muzikanten een protocol bedachten om synthesizers met elkaar te kunnen laten communiceren. Midi was en is dus een communicatie protocol voor muziek. Later bleek dat erg goed bruikbaar voor toepassing bij computers. Je kan door gebruik te maken van midi je computer allerlei composities laten spelen. Daarvoor heb je een zogenaamde software synthesizer nodig zoals bijvoorbeeld ingebouwd zit in Quicktime of Windows Media Player. Als je wel eens een midi file hebt afgespeeld in Quicktime of in Windows Media Player dan hoor je ook gelijk het nadeel van midi: de geluiden zijn wel erg synthesizer-achtig. Prima voor een techno-beat maar voor het produceren van meer realistische geluiden of gevoelige instrumentele muziek niet goed geschikt.

Bekende programma's die je kan gebruiken om midi muziek mee te maken zijn: Cubase, Logic, Cakewalk, Music Maker, Garageband. Heel aardig is ook het gratis Anvil studio en er is natuurlijk nog veel meer.

Wat te doen als je niet muzikaal bent en/of geen noten kan lezen? Je kan je natuurlijk aanmelden voor een cursus bij de muziekschool, maar je kan ook gewoon wat proberen. Veel van de midiprogramma's zijn prima geschikt voor een beginnende muzikant. Bovendien zitten er allerlei swingende grooves bij de meeste programma's zodat het al snel wat lijkt.

Audio is de andere tak in de computer-geluidswereld. Audio gaat niet uit van het midi-protocol maar van de geluidsgolven zelf. Als je dus voor je game een sprekende stem nodig hebt of het geluid van een overvliegend vliegtuig of van het kapot vallen van een vaas of de melodie van een fijngevoelige viool, dan heb je een audio programma nodig (en iemand die bereid is zijn kostbare vaas kapot te gooien zodat jij het kan opnemen).

Het programmaatje waarmee je met de ingebouwde microfoon iets kan opnemen op je PC is een eenvoudige audio programma. Er zijn veel uitgebreidere programma's beschikbaar waar je niet alleen mee kan opnemen, maar waarmee je ook aan het audio geluid kan sleutelen. Om van een mannenstem een vrouwenstem te maken, of om galm en echo toe te voegen. Stel dat je een adventure maakt waarin je hoofdspeler op een gegeven moment in een grot loopt, dan moet

het wel echt klinken als een grot met galm en druppelende stalactieten erbij.

Bekende audio programma's zijn: Protools (een complete professionele studio voor het opnemen van bands), Adobe Audition of de gratis programma's Audacity en Ardour.

MINITUTORIAL: HET GELUID VAN DE MAGNETISCHE POORT

Stel dat je een RPG fantasy game aan het maken bent in de stijl van Heroes of Might and Magic. Daarin heb je een magnetische poort waarmee je van de ene wereld naar een andere wereld kan reizen. Je hebt natuurlijk een geluid nodig voor als je held in de buurt komt van die poort. Hoe maak je zo'n "magnetisch" geluid? Het geluid in de volgende tutorial klinkt aardig "magnetisch".

Allereerst heb je een soundeditor (audio) programma nodig. Audacity is zo'n programma dat je gratis kan downloaden van www.audacity.de. Installeer en start het programma. Je scherm ziet er op PC iets anders uit dan op een Mac, maar het scheelt niet veel. Hoe je daarmee een Magnetische Poort Geluid maakt, zie je op de volgende bladzijde.



Cubase



Logic Pro

LEVEL 5: EEN GOED IDEE

Veel beginnende gamemakers beginnen met het nabouwen van hun favoriete game. Daar is niets op tegen, het is zelfs een goede oefening om de vaardigheden van gamedesign onder de knie te krijgen. Maar het wordt pas echt interessant als je echt iets nieuws probeert te maken.

Hoe kom je op een goed idee? Tja...gebruik je nachtmeries als inspiratie, mediteer op een berg, ga in de kroeg zitten, maak het uit met je partner en gebruik je gebroken hart om tot bevlogen ideeën te komen, vraag advies aan je kat of aan je stofzuiger, brainstorm met je vrienden, haal inspiratie uit de natuur, jat een idee en verbeter het...kortom er is geen standaardoplossing voor een goed idee. Ideeën komen of ze komen niet.

Er is ook geen standaard recept voor een goede popsong, een goed schilderij of een geslaagde roman. Dus ook niet voor een goede game. Wel is het zo dat je je game idee kan toetsen.

Checkout level 9: Aantrekkelijke games maken.

Het is een goede gewoonte om je idee op papier te schetsen voordat je aan de slag gaat. Laat het uitgewerkte idee aan een paar mensen zien om het idee te testen. Vaak komen ze met goede opmerkingen of aanvullingen.

MINI-TUTORIAL: HET GELUID VAN EEN MAGNETISCHE POORT



1: De editor is nog leeg. Kortom, we hebben een basistoon nodig om het magnetische geluid te maken.

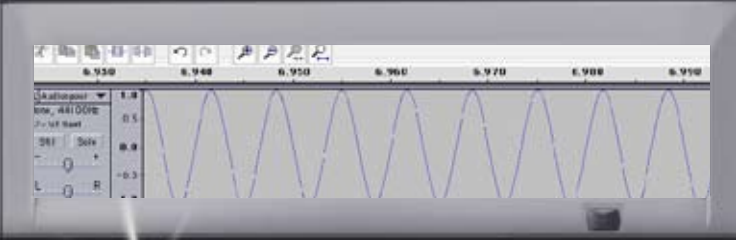
2: Kies onder "Genereren" voor "Toon.."



3: Zoals je weet zijn geluiden opgebouwd uit golven. In de pop up die verschijnt, kan je kiezen uit een aantal soorten golfvormen. Experimenteer later met zaagtand en vierkante tonen. Voor nu kan je de golfvorm op sinus laten staan. Kies een frequentie. Hoe lager het getal, hoe lager de toon zal zijn. Hier is er voor gekozen om de frequentie op 150.00 Hz te zetten. De amplitude (de hoogte van de golven oftewel het volume van het geluid) kan je laten staan op 1.



4: De computer moet nu wat rekenen en voila, je hebt een toon gegenereerd. In je editor staat nu een balk.



5 (lb):

Maar waar is nu die typische sinusvorm gebleven? Klik met je muis een stuk of 10 x op het vergrootglas linksboven en de sinus wordt zichtbaar.

6 (rb).

Als je op de play knop drukt dan hoor je welk geluid je hebt. Zet je speakers niet te hard want het sinusgeluid is niet echt prettig om naar te luisteren. Druk op stop. Klik ergens in de editor klik op appeltje+a (Mac) of CTRL+a (PC) om de geluidsgolf te selecteren. Tijd om het geluid "magnetisch" te maken. Ga naar "Effect".

Effect menu options:

- BASSBOOST...
- Compressor...
- Echo...
- Fade-in
- Fade-out
- FFT-filter...
- Herhalen...
- Normaliseren...
- Nyquist prompt...
- Omkeren
- Phaser...
- Ruis Verwijderen...
- Snelheid veranderen...
- Spiegelen
- Tempo veranderen...
- Toonhoogte veranderen...
- Toonregeling...
- Versterken...
- Wahwah...
- Cross Fade In
- Cross Fade Out
- CVerb...
- High Pass Filter...
- Low Pass Filter...

7 (lo).

Bij alle audioprogramma's zitten meer of minder effecten ingebouwd. Vergelijkbaar met de treble of bassboost om je versterker, alleen uitgebreider. Kies uit deze lijst met effecten de Tremolo. Experimenteer wat met de mogelijke instellingen en druk op ok. De computer rekent het tremolo effect uit op de sinusgolf en als het goed is gegaan, ziet je editor er ongeveer zo uit:

8 (ro).

Druk op play om te controleren of je het geluid "magnetisch" goed genoeg vindt. Via "export als wav" kan je het geluid save om later te gebruiken in je game. Experimenteer met de andere effecten. Er is van alles mogelijk. Je stem achterstevoren afspelen, lasergeluiden maken, het geluid van een straaljager, je eigen gitaarspel opnemen (en de foutjes eruit knippen), enzovoort.



LEVEL 6: GAMES PROGRAMMEREN

Je hebt je graphics, je hebt je sounds en je hebt een goed idee. Tijd om alles in elkaar te gaan zetten. Maar hoe? Games zijn computerprogramma's, net als Word of Excel, maar dan anders. Kortom, je zult moeten (leren) programmeren.

Ho, stop, HAAK NIET AF...(althans nog niet hier). Velen denken dat programmeren heel moeilijk is en dat het iets met wiskunde is enzo en dat alleen echte nerds dat leuk vinden. Welnu, JA DAT KLOPT!!! Programmeren kan behoorlijk lastig zijn, het heeft zeker iets met wiskunde te maken en je moet wellicht wel een beetje sociaal afwijkend zijn om dit moderne monnikenwerk leuk te vinden...

Wat nu? Je kan je natuurlijk specialiseren in graphics of sound design of de ideeën man/vrouw zijn en anderen het werk laten doen. **Checkout level 12**: Samen of alleen. Maar misschien is het wel aardig om eens te verkennen hoe dat nu zit met dat programmeren...

Computers zijn opgebouwd uit een hele hoop schakelaars (maar dan ook echt heel veel) die aan of uit staan. Vandaar dat je in de computertaal veel binaire getallen tegenkomt. De 1 staat voor een schakelaar die aanstaat en de 0 staat voor een schakelaar die uitstaat. Als je software gaat programmeren, ga je spelen met die schakelaars. Als je bijvoorbeeld wilt dat er een rode stip op het scherm moet komen dan programmeer je de computer om de schakelaar van de

rode pixel op je scherm aan te zetten (op 1 te zetten). Er gaat dan een stroompje naar je beeldbuis en zie een rode stip! Als de rode stip weer wegmoet, zet je de schakelaar op 0, het stroompje naar de beeldbuis stopt en weg is de stip.

Gelukkig hoeft je met moderne computers niet meer met 0-en en 1-en te werken. Er zijn computertalen gemaakt die een beetje op Engels lijken die de vertaling naar 0-en en 1-en voor je doen. Voorbeelden van dergelijke talen zijn Basic, Lingo, C++, Java, Python, HTML, PHP, enzovoort.

Omdat de schakelaars in een computer maar twee standen kennen, een 0 of een 1, moet je wel heel precies zijn in het definiëren van de taken voor een computer. Je kan de schakelaar niet op een half zetten. Je kan niet tegen een computer zeggen: "doe mij maar een rode stip ergens op het scherm". Je zult de computer heel precies moeten vertellen hoe groot die stip moet zijn, hoeveel pixels, waar hij op het scherm moet komen en welke tint rood het moet zijn.

Bijvoorbeeld een rode stip van 4 bij 4 pixels, in de RGB (rood-groen-blauw) waarde (255,0,0) op positie 100, 100 in het scherm.

Help, Help, echte cijfers en getallen! Ben je er nog of ben je al weg naar een ander level? Zo niet dan is het tijd voor een kleine tutorial over game programmeren:

Je hebt een game idee waarin een cowboy over het scherm loopt. Je kan de cowboy naar links en rechts bewegen met de spatiebalk en de entertoets.

Goede gewoonte is om je game eerst stap voor stap te beschrijven in pseudocode voordat je hem in een echte programmeertaal gaat schrijven. Voor bovenstaande game zou de pseudocode bijvoorbeeld kunnen zijn:

Als de game start:

- ☆ Plaats een plaatje van een cowboy in het midden van het scherm. Het scherm is 400 bij 400 pixels groot.

Als de game loopt:

- ☆ Kijk of er een toets is ingedrukt en zo ja:
- ☆ Is dit de ENTER toets?
 - o ja: verplaats de cowboy 5 pixels naar rechts
 - o nee: ga verder naar de volgende stap
- ☆ Is dit de SPATIE toets?
 - o ja: verplaats de cowboy 5 pixels naar links
 - o nee: ga verder naar de volgende stap
- ☆ Kijk weer opnieuw of er een toets is ingedrukt (ga weer naar boven)

Bovenstaande pseudocode is 80% van het denkwerk dat er moet gebeuren. Begrijp je wat hierboven staat, dan snap je wat programmeren is! Stap voor stap nauwkeurig beschrijven wat er moet gebeuren. Het

vertalen naar Basic, Java, of welke programmeertaal dan ook is dan niet meer zo moeilijk:

In Lingo, de taal van Macromedia Director zou bovenstaande er zo uit komen te zien:

```
On startmovie
Sprite(1).locH = 200
Sprite(1).LocV = 200
End startmovie
```

Uitleg van de code:

In Director worden de programma's "movies" genoemd. On startmovie betekent zoiets als: "wanneer de movie (game) start, doe dan het volgende." In Director krijgen sprites (plaatjes) een nummer. Het plaatje van de cowboy is je eerste (en in deze game enige) sprite en die sprite krijgt nummer 1. Dus Sprite(1) is je cowboy. Als je later ook een indiaan gaat toevoegen zou die bijvoorbeeld Sprite(2) kunnen zijn. LocH en LocV zijn afkortingen voor de horizontale en verticale locatie op het scherm. Met andere woorden, als we de cowboy in het midden willen plaatsen, dan zetten we beide waarden op 200 (de helft van het scherm van 400 bij 400 pixels).

Play a game?

Dan gaat de game lopen. De code in Lingo daarvoor zou bijvoorbeeld zijn:

```
On keydown
If the key = RETURN then
Sprite(1).locH = Sprite(1).locH +5
End if
if the key = SPACE then
Sprite(1).locH = Sprite(1).locH -5
End if
End keydown
```

Uitleg van de code:

On keydown (als er een toets wordt ingedrukt)

If the key = Return then (als de toets de return toets is dan ...)

Sprite(1).locH = Sprite(1).locH +5 (de horizontale positie van de cowboy wordt opgehoogd met 5 pixels. Hij staat dus niet meer op 200, maar nu op 205, een stukje naar rechts.)

Idem voor de spatie (space) toets, 5 pixels naar links. De End If en End Keydown woorden zijn om de boel weer af te sluiten, zodat de computer weet wat bij elkaar hoort.

Laten we voor de grap eens kijken naar een andere programmeertaal. Bijvoorbeeld GML (GameMakerLanguage). In GML zou het programmeren van de cowboy er zo uitzien:

Als de game begint:

```
Instance_create(200,200,Cowboy);
```

Uitleg: Gamemaker werkt met objecten. Alle elementen in je game zijn een object. Je cowboy is dus ook een object. De functie Instance_create(200,200,Cowboy); zet een kopie van dat object op de positie 200, 200 in je scherm. Zo'n kopie wordt een instance (letterlijk vertaald betekent instance in het engels aftreksel) genoemd. Er wordt een kopie gemaakt, zodat je later gemakkelijk nog meer instances, meer cowboys op het scherm kan plaatsen.

Als je game loopt:

In de RETURN toets events:

```
Cowboy.x =Cowboy.x+5
```

In de SPACE toets events:

```
Cowboy.x= Cowboy.x-5
```

Uitleg van de code:

Gamemaker werkt met zogenaamde "events". Events zijn gebeurtenissen in je game, zoals het indrukken van een toets, het bewegen van je muis, het afgaan van een alarm, het botsen van sprites enzovoort. De events triggeren de acties in je game.

In dit geval als de RETURN toets ingedrukt wordt, wordt de x waarde (je weet wel de X-as in je wiskundeles) met 5 pixels opgehoogd. De cowboy

7. B.v. The Game Maker Apprentice, door Jacob Habgood en Mark Overmars of Advanced Lingo for Games, door Gary Rosenzweig.

8. Zie b.v. <http://www.gamemaker.nl/tutorial.html>

9. Zie www.gamemaker.nl

gaat naar rechts. Idem 5 pixels naar links als je op SPACE duwt.

Nou was dat moeilijk? Nee..., mooi! Boven staan een paar simpele voorbeelden van het programmeren van een game. Natuurlijk moet er nog veel meer bij gemaakt worden voordat je een complete game hebt. Komen er nog meer cowboys en indianen bij? Gaan ze op elkaar schieten? Hoe vaak mag je geraakt worden voordat je dood bent? Wat gebeurt er als de cowboy van het scherm afloopt? Enzovoort. Er is dus nog een hoop meer code nodig voordat je game af is.

Misschien vond je bovenstaande voorbeelden toch wel lastig te begrijpen. OK, ik (de schrijver van dit boekje) neem de schuld op me. Een paar pagina's is veel te kort om te leren programmeren en dan ook nog twee talen door elkaar! Misschien is het beter een meer uitgebreid boek te lezen over (games-)programmeren.⁷ Programmeren leer je het beste door het zelf te doen. Stap voor stap een game programmeren. Op het web zijn er genoeg tutorials te vinden die je uitleggen hoe je games kan maken⁸ en anders zijn er ook veel forums waar je hulp kunt krijgen als je er niet uitkomt.

IK WIL ECHT
NIET PROGRAM-
MEREN!

reclameblokje

Gamemaker⁹ is een programma waarmee je games kan maken zonder dat je hoeft te programmeren. Door middel van de weldoordachte drag&drop interface kan je met Gamemaker een game in elkaar zetten zonder coderen. Je plaatst de sprites eenvoudig op het scherm en door middel van het aanwijzen van events en acties doet de sprite alles wat je wilt. Events zijn bijvoorbeeld het indrukken van een toets, het reageren op de muis, het afgaan van een alarmklokje of het botsen van twee sprites. Acties zijn bijvoorbeeld verplaatsen, ontploffen, verhogen of verlagen van de score. Alles wat je nodig hebt om een volwaardige game te maken. Er zijn meer gamebouwpakketten die dit concept volgen, maar Gamemaker is zeker een van de best gelukte. En het is nog gratis ook! Voor de gevorderde gamebouwer die uitgekeken raakt op de drag&drop interface is er GML (Game Maker Language) de ingebouwde scripttaal, waarmee je meer mogelijkheden hebt.

Ik wil het niet, ik kan het niet, programmeren is echt niets voor mij...maar toch wil ik een eigen game maken. Kan dat? Oh meester red mij! ...



**Ja het kan.
Een reclameblokje dan...**



Gamemaker



Flash actionscripting

In welke taal moet ik mijn game programmeren?

De reden dat hiervoor twee programmeertalen werden getoond, is dat ik wilde laten zien dat programmeertalen erg op elkaar lijken.

In Director zeg je:

```
Sprite(1).locH = Sprite(1).locH +5
```

In GML is het: `Cowboy.x =Cowboy.x+5.`

om je cowboy 5 pixels naar rechts te verplaatsen.

Je zult zien dat het in alle programmeertalen iets dergelijks is, met enkel wat verschillen in de syntax. Als je eenmaal een programmeertaal beheerst, is het niet moeilijk om een volgende erbij te leren. Makkelijker dan een mensentaal erbij leren!

Professionele games zoals de games voor de consoles worden in C++ geschreven. C++ is een krachtige taal omdat je je computer er op alle lagen in de machine mee kan programmeren. Je kan bijvoorbeeld rechtstreeks het geheugen of de grafische kaart benaderen vanuit C++. Dat maakt de taal wel complexer om te leren.

Een ander nadeel van C++ is dat je veel code (veel tekens) nodig hebt om iets te bereiken. De hoeveelheid code om een bepaalde actie gedaan te krijgen in je game is snel een paar pagina's lang. Om dat te vereenvoudigen en te versnellen zijn er zogenaamde scripttalen uitgevonden. Scripttalen zijn minder

krachtig dan C++ maar je hebt veel sneller resultaat. Voorbeelden van bekende scripttalen zijn: Python, Actionscript (Flash en Director), Lingo (Director) en Basic (DarkBasic, Blitzbasic, Realbasic).

Alleen als je superfanatiek bent, is het aan te raden om te beginnen met het leren van C++. Een scripttaal leer je sneller en je hebt er ook sneller resultaat mee. Als je tegen de begrenzings van je scripttaal aanloopt is het een mooi moment om C++ te gaan leren. Naast de programmeertaal moet je ook je programmeeromgeving kiezen waarin je werkt. Stel dat je wilt werken met een BASIC dialect om je game te programmeren. Er zijn dan veel mogelijkheden.

Bijvoorbeeld de algemene programmeeromgevingen als Realbasic of VisualBasic.net. Daarnaast zijn er speciale zogenaamde gamekits die in feite programmeeromgevingen zijn speciaal voor het maken van games. Gamekits die met een BASIC dialect werken zijn onder andere: DarkBasic, BlitzBasic en TNT-Basic. Omdat we games willen maken, hebben de gamekits natuurlijk de voorkeur boven de algemene programmeeromgevingen.

Waar op te letten bij het kiezen van een programmeertaal en een programmeeromgeving (gamekit):

Welk niveau van programmeren ambieer je? Wil je gewoon wat proberen en niet al te ingewikkelde games maken, dan zijn pakketten als Gamemaker of

TNTbasic waarschijnlijk meer dan geschikt, zeker om mee te beginnen. Je ziet dan later wel of je wilt overstappen naar (semi-)professionele pakketten als Director of Torque.

Wil je de programmeertaal ook voor andere doelen gebruiken? Als je voor je opleiding bijvoorbeeld Java gaat leren, is het misschien wel slim om een gamekit of omgeving te zoeken die op Java is gebaseerd, zoals Octlight bijvoorbeeld.

De kwaliteit en de ondersteuning van de gamekit. Sommige gamekits zijn niet zo stabiel.¹⁰ Kijk op de forums of er (veel) over geklaagd wordt. Zijn er veel gebruikers van het pakket, is er een grote community die elkaar helpt en zijn er veel (hand-)boeken en tutorials beschikbaar of cursussen? Wordt de gamekit regelmatig verbeterd door middel van upgrades?

Welke speciale mogelijkheden wil je? Alleen 2d (bv. Gamemaker, Torque Game Builder) of ook 3d (bv. Blender Gamekit of 3Dstudio)? Stand alone games of ook webgames (Flash, Java, Director)? Op PC, Mac, Linux (of crossplatform)?

Hoeveel wil je betalen voor de programmeeromgeving? Macromedia Director en Microsoft Visual Studio.net zijn prima omgevingen om games in te maken, maar wellicht wat prijzig voor een liefhebber.

Als je studeert in Nederland, kan je de pakketten via Surfspot¹¹ vaak voor maar enkele eurotientjes kopen, waardoor de situatie wat anders ligt.

En tenslotte, bekijk de games die gemaakt zijn met de gamekit om te kijken wat het eindresultaat kan zijn. Toegegeven: een slechte designer maakt met mooi gereedschap nog steeds een baggergame en een slimme designer maakt zelfs met een eenvoudige gamekit toch een leuk spelletje. Maar toch, zoals de Engelsen zeggen: "The proof is in the pudding".

Kijk op www.gamesmaken.startpagina.nl voor een lijst van gamekits en voor games geschikte programmeertalen.

LEVEL 7: LEVEL EDITING EN GAME MODDING

Een andere manier om een eigen game te maken is door middel van level editing of game modding. Level editing is het maken van je eigen level in je favoriete game. Game modding gaat verder, hierbij wordt de game grondig veranderd, maar wordt nog wel de game engine gebruikt van de oorspronkelijke game.



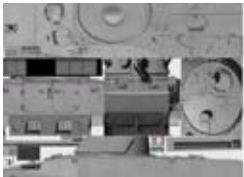
www.gamesmaken.startpagina.nl

10. Het is niet te doen om alle gamekits uitvoerig te testen. Kijk daarom goed op de forums wat gebruikers schrijven over de gamekits.

11. Zie: <http://www.surfspot.nl/>



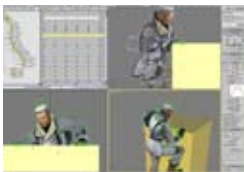
Draadframe van een tank



De skins in 2d



Het resultaat met een skin erop



3d animeren van een klimmende man. Linksboven de tijdlijnen, de overige vierkanten zijn de verschillende perspectieven van de gamefiguur.

LEVEL EDITING

Bij de games die je in de winkel koopt zitten vaak level editors. Je kan daarmee je eigen level of scenario maken van Starcraft, Heroes of Might and Magic, Doom, of welke game dan ook. Vaak kan je dan ook je eigen graphics importeren en zo je eigen helden creëren. Je kan ook zelf de held zijn. Door middel van face mapping projecteer je een pasfoto van jezelf op het 3d hoofd van de gameheld en loop je daarna door je eigen game!¹²

De zelfgemaakte levels worden veelvuldig uitgewisseld en gespeeld als multiplayer games op het internet. Level editen is niet zo moeilijk en het is fun. Een Doom level maken van je schoolgebouw, een simcity level maken van je eigen stad. Heel veel is mogelijk, al ben je wel gebonden door de beperkingen van het oorspronkelijke spel. Wil je daarbuiten gaan, dan moet je gaan Game modden.

GAME MODDING

Game modding gaat verder dan je eigen levels of graphics toevoegen aan een bestaand spel. Als er een nieuw spel wordt geproduceerd, wordt er tegenwoordig meestal eerst een game engine gemaakt. In die game engine wordt de basis van het spel geprogrammeerd. Bijvoorbeeld hoe de zwaartekracht wordt afgehandeld in de computer. Als dat namelijk gedaan is hoeft je niet meer voor elk object in de game de zwaartekracht te programmeren, maar kan je als het

ware verwijzen naar de code in de game engine.

Game modders gebruiken die game engine en maken daarmee hun eigen game. Die kan er heel anders uit komen te zien dan de oorspronkelijke game. Zo zijn er van Battlefield race spellen gemaakt, een Star Wars spel maar ook een Muppets mod. Een van de bekendste en succesvolste mods is Counter Strike dat gemaakt werd door het modden van Halflife.

Er zijn verschillende websites van game modders communities.¹³ Daar kan je meer informatie vinden over de technieken. Vaak zijn er ook projecten bezig waarin mensen samen werken aan hun eigen game mod.

LEVEL 8: 3D

De meeste commerciële games die nu in de winkel verschijnen zijn 3d games en in de laatste paar jaar zijn die games steeds mooier geworden. Met de komst van een nieuwe generatie consoles, zal er weer een stap gemaakt worden in de verbetering van de grafische kwaliteit van games.

Bij 3d games zijn de spelkarakters en werelden in 3d vormgegeven. Om een 3d karakter of wereld te maken heb je een 3d modellingsprogramma nodig. Bekende programma's zijn bijvoorbeeld: 3dMax, Maya, Cinema4d (commerciële pakketten) en Blender (gratis en open source). 3d Max is het dominante

pakket in de professionele gamedesign wereld. Maya wordt vooral gebruikt in de filmwereld, maar wordt ook wel in de gamewereld gebruikt. Als je aspiraties hebt om in een van deze werelden je werk te gaan vinden, is het aan te bevelen om deze pakketten door en door te leren.¹⁴ Als je die ambitie niet hebt is een gratis pakket als Blender uitstekend (en vooral veel goedkoper dan 3dMax of Maya). Uiteraard is er nog veel meer keus uit 3d software¹⁵, dus onderzoek wat je prettig vindt werken.

3d modellen is alsof je met ijzerdraadjes in de ruimte een vorm (draadframe) maakt. Als de vorm goed is gebruik je een zogenaamde skin om die over het draadframe te projecteren. Die skins worden gemaakt in een 2d grafisch pakket als Photoshop of the Gimp. Om een realistische skin te krijgen, worden vaak de texturen van echte foto's gebruikt.

Het animeren van 3d objecten is een stuk ingewikkelder dan het animeren van 2d sprites. Zie de afbeelding. Om een man te laten klimmen moet je alle groene blokken in een tijdlijn animeren. Niet alleen romp en benen bewegen, maar alle ledematen. En misschien trekt hij er ook nog een moeilijk gezicht bij!

Vergis je niet in de hoeveelheid werk die het maken van een 3d game met zich meebrengt. Het maken van een model als bovenstaande tank is voor een ervaren

3d modeller ongeveer 2 weken werk. Reken maar uit hoeveel werk het is als je een game hebt met 6 helden, 30 monsters, 10 voertuigen, 5 werelden met elk 4 levels en nog eens een tiental kleinere objecten!

In de professionele game industrie is 3d een specialisme. Eigenlijk zie je zelden dat een 3d modeller ook nog audio programmeert of conceptdesign doet. Binnen de 3d modellering zijn er weer aparte specialiteiten: 3d level design, face modelling, 3d animatie, skin designers. Als 3d je ligt is het heel leuk om te doen. Maar realiseer je dat het niet reëel is om in je eentje een uitgebreide 3d game te bedenken, te modelleren, te programmeren en van geluid te voorzien. Tenzij je genoegen wilt nemen met een kleine 3d game is het misschien beter te werken in een team?

Checkout level 12 : Samen of alleen

LEVEL 9:

AANTREKKELIJKE GAMES MAKEN

Genoeg over technieken! Het is allemaal wel leuk als je mooie graphics kunt maken of stoere 3d monsters, maar maakt dat een game interessant? Wat maakt een game eigenlijk interessant? Deze vraag is voor gamedesigners de heilige graal en (misschien wel gelukkig) zullen we die nooit vinden. Toch zijn er in de jonge geschiedenis van gamedesign wel een paar wetmatigheden ontdekt.¹⁶



Face modellering in 3d. De streepjes vormen het onderliggende wireframe waarmee de vorm van het gezicht en het hoofd is gemaakt.

12. Zie: The official blender gamekit, Ton Roosendaal en Carsten Wartman, hoofdstuk 1.1 voor een uitleg van face-mapping.

13. Zie www.gamesmaken.startpagina.nl voor een lijst van gamemod communities.

14. Al kan het best zo zijn dat over een paar jaar heel andere 3d pakketten de markt domineren.

15. Kijk bijvoorbeeld op www.3d.pagina.nl

16. Grote kans dat we over 10 jaar weer heel anders denken over wat een game nu interessant maakt.



Besteed dus meer aandacht aan de binnenkant van je spel dan aan de buitenkant. Een spel met goede spelregels en matige graphics is leuk, een spel met mooie graphics maar slechte spelregels is snel uitgespeeld. Bovendien zijn er veel succesvolle spellen geweest die er grafisch gezien niet bijzonder uitsprongen: Tetris, Bomberman, Snake, Worms, en vele andere. Misschien bedenkt jij wel het volgende succes...¹⁸

WAT MAAKT EEN GAME INTERESSANT?¹⁷

1. De gamescene is niet belangrijk
2. Goeie spelregels zijn het belangrijkste
3. Het spel moet uitdagend zijn
4. Gevoel van controle
5. Verschillende keuzes zijn mogelijk
6. Easy to begin, hard to master
7. Balans in je spel
8. Goede interactie

1. De gamescene is niet belangrijk. Het maakt voor het plezier niet uit of je een game hebt in een space, fantasy, oorlog of western wereld. Uiteraard zijn er mensen die meer hebben met space dan met de western wereld en andersom, maar toch maakt dat voor het plezier van de game weinig tot niets uit. Warcraft en Starcraft zijn beide uitstekende games, maar dat is omdat het goed bedachte spellen zijn, niet omdat de ene in de fantasy wereld speelt en de andere in de space wereld. Dit brengt ons bij de tweede wet:

2. Goede spelregels maken het spel. Warcraft en Starcraft zijn beide goede spellen omdat de onderliggende spelregels (real time strategy) goed in elkaar zitten. Sterker nog, het zijn dezelfde spelregels voor beide games. Zou er een CowboyCraft komen, een Warcraft in de western wereld, dan zou dat weer een goed spel

zijn (alleen wellicht een beetje te veel van hetzelfde). Ook grafische vormgeving is van ondergeschikt belang. Zie hiernaast het klassieke schaakspel. Het onderste spel is net zo leuk als het bovenste.

3. Het spel moet uitdagend zijn. Een goed spel is uitdagend. Je moet als speler iets zien te bereiken, er is een doel. Dat doel bereik je door te lopen, jumpen, bouwen, vechten, handelen, zoeken, jagen, verkenen, communiceren, enzovoort. Het is moeilijk om het doel te bereiken maar niet te moeilijk. Als je vastloopt in het spel moet je als speler het gevoel hebben dat het aan jou lag en dat je beter kunt (dan gooi je namelijk weer een euro in die gamekast!). Als je spel zo moeilijk is dat de speler het gevoel krijgt dat hij er geen controle over heeft, haakt hij af.

4. Gevoel van controle is belangrijk voor een speler. Een goed spel heeft een doel en als speler moet je dat doel proberen te bereiken. Of dat nu het redden van een prinses is, het bereiken van werelddominantie of het behalen van een hoge ranking op een laddergame. De speler moet daarbij controle hebben over zijn spelfiguur. Als hij niet het gevoel heeft dat hij zijn spelfiguur kan beheersen, is het spel te moeilijk of is er teveel toeval in het spel. Een fruitautomaat is zo'n spel waar je als speler geen controle over hebt. Tenzij je gelooft dat je hogere krachten bezit en daarmee de fruitautomaat kan beheersen is daar weinig aan.

5. Een goed spel kent meerdere winnende strategieën. Een puzzel kent slechts één winnende oplossing. Een goede game kent verschillende winnende strategieën. Je kan Heroes of Might and Magic winnen door heel veel toverkracht te krijgen, door grote legers te bouwen, door rijk te worden aan grondstoffen, door het vinden van een artefact, door veel helden te hebben en waarschijnlijk op nog een hoop andere manieren. Dit houdt het spel verrassend. Een slimme tactiek om je game interessant te houden is door hem multiplayer te maken. Mensen hebben allemaal hun eigen manier van spelen en dat vraagt om (steeds) andere strategieën om te winnen. Een goed geprogrammeerde single player game reageert ook steeds anders op de spelers door middel van de ingebouwde AI¹⁹, toch kan dit nog steeds niet op tegen het vernuft van de menselijke tegenspeler.²⁰

6. Easy to begin, Hard to master. Een goede game is niet moeilijk te leren. Wat is het toch vervelend als je bij een nieuwe game eerst de handleiding moet doorspitten en daarna een uitgebreide training moet volgen voordat je echt kan beginnen! Gemakkelijk beginnen dus. Naarmate je de game als speler langer speelt wordt je steeds beter in het spel. Daarom moet het spel ook steeds een stapje moeilijker worden. Zoveel moeilijker dat het uitdagend blijft maar weer niet te moeilijk dat spelers afhaken. Dit is in de praktijk nog lastig te bereiken want iedere gamer heeft

een individueel leertraject. Veel door testers laten spelen dus je game en kijk waar ze te gemakkelijk doorheen gaan of waar ze vastlopen. Zorg er voor dat je na een aantal levels een nieuw spelelement inbrengt. Hou het verrassend.

7. Balans in je spel. Je spel moet constant in balans zijn. Dus niet te moeilijk in het begin en niet te gemakkelijk halverwege en aan het einde van het spel. Dit geldt ook voor de balans tussen spelers (of dat nu een computer is als tegenspeler of een menselijke tegenspeler). Als je door het vinden van een superwapen alle tegenstanders overwint, is het spel snel niet meer leuk. Dus elk voordeel moet een nadeel hebben. Elk superwapen moet een zwakke plek hebben. Tvenaars kunnen goed toveren tegen dwergen, maar zijn zwak tegen de sterke barbaren. De barbaren zijn op hun beurt weer zwak tegen de dwergen. Op die manier is geen enkel volk het sterkst in je RPG game.

8. Goede interactie maakt het spel lekker speelbaar. Uiteraard moet je game “lekker lopen”. Niet te traag en niet te snel. Als je op een knop drukt, moet de computer gelijk reageren en het moet duidelijk zijn op welke knoppen je überhaupt moet drukken. Ook moet bijvoorbeeld de navigatie in het spel duidelijk zijn.²¹ Dit lijkt vanzelfsprekend, maar toch gaat dit nog wel eens mis bij te snel in elkaar gezette spellen.

17. Lees onder andere: A Theory of Fun for Gamedesign, Raph Koster; What is a good game?, Mark Overmars

18. Een aardig boek over het ontwerpen van spelregels is Chris Crawford on Game design.

19. AI = kunstmatige intelligentie. Door middel van het programmeren van je computer, probeer je hem een intelligente en waardige tegenstander te laten zijn van de menselijke gamer.

20. Behalve bij schaken, waarbij de computer zelfs de beste schaker van de wereld verslaat.

21. Voor een wetenschappelijk verhaal over interactiedesign: Baaui, E., Bekker, M. M., and Barendregt, W. (2005): A Structured Expert Evaluation Method for the Evaluation of Children's Computer Games. In: Proceedings of Human-Computer Interaction INTERACT 2005, 14-16 September 2005 Rome Italy, Springer Verlag.

Het toverwoord ook hier is testen, testen en nog eens testen. Zorg ervoor dat je ook op oudere machines test (een pentium III of Mac G3 bijvoorbeeld) omdat die toch nog veel gebruikt worden.²²

We weten niet echt wat een game een goede game maakt. De regels die hierboven staan, zijn maar een paar richtlijnen die je kunt gebruiken om je game te evalueren. Maar veel belangrijker nog is dat je je game laat testen en testen en testen en testen en.....

LEVEL 10: JE GAME PUBLICEREN

Na een paar weken of maanden zwoegen is je game eindelijk af. Misschien ben je bescheiden over het resultaat. Je bent immers maar een amateur...

Amateur...? Bedenk dat alle gamebouwers amateurs zijn! Pas nu, zo'n dertig jaar na het ontstaan van de videogames ontstaan er officiële gamedesign en game development scholen. En de eerste officiële gamedesigners en game developers met een officieel diploma moeten nog afstuderen. Met andere woorden: alle gamedesigners en –developers die het tot nu toe gemaakt hebben in de gamewereld zijn AMATEURS!

Dus het zou zonde zijn als de game alleen door vrienden en familie gespeeld wordt. Tijd om hem te gaan

verspreiden. Hoe kan je dat aanpakken?

Vroeger werden games alleen verspreid via fysieke dragers (CD roms, floppies en zelfs cassettebandjes). Om jouw game dan over de wereld te krijgen, had je de hulp nodig van een uitgever die de games voor jou distribueerde. Als je denkt dat jouw game een (commercieel) succes zal zijn, kan je nog steeds naar de uitgevers stappen om te vragen of zij je game in de markt willen zetten. Een hoop gedoe, waarbij je dan zo'n uitgever moet overtuigen dat de game echt een kassucces zal zijn. En wat als jouw game een hele leuke remake is van Donkey Kong? Leuk om te spelen, maar waarschijnlijk niet interessant om te kopen.

Dan is er een andere mogelijkheid: Je kan nu je game via het web verspreiden. Of je dit nu gratis doet, als shareware of als betaalde game. Lang leve het internet! Verschillende sites bieden plekken aan om je game in te zetten, compleet met feedback forum en zoekfuncties.²³ Je kan natuurlijk ook je eigen site maken voor de game.

Denk na of je je game gratis, als shareware, commercieel of open source wilt verspreiden. Gratis spreekt voor zich, bij shareware vraag je de speler of hij een klein symbolisch bedrag wil overmaken als hij de game goed vindt. Je moet dan wel iets regelen dat mensen die euro's kunnen overmaken. Bijvoorbeeld via Paypal.

Als je denkt dat mensen je game goed genoeg vinden, zou je hem commercieel kunnen uitbrengen. Je zou

22. Zoals bij bepaalde doelgroepen als jonge kinderen of lagere scholen.

23. Zie bijvoorbeeld: <http://www.gamemakergames.com>

24. Zie: <http://www.gnu.org/copyleft/gpl.html> en <http://www.creativecommons.nl>

25. Alhoewel Zmoeffies nu ook niet meer kunnen, omdat ik dat hier in dit boekje heb bedacht. Beware mijn advocaten!

de game ook open source kunnen uitbrengen. Dan is de game niet alleen gratis voor gebruik, maar biedt je ook de broncode aan, zodat anderen verder kunnen werken aan jouw game. Dit heeft als voordeel dat je niet al het werk alleen hoeft te doen. Voordat je dat doet is het aan te raden om je te verdiepen in een aantal licentiemodellen waarmee je de rechten van je game beschermt, bijvoorbeeld de GPL of de creative commons licenties.²⁴

LEVEL 11: COPYRIGHTS

Hoe zit het eigenlijk met de rechten van materiaal van anderen dat je gebruikt in je game? Stel dat je een game wilt maken maar je kan niet zo goed grafisch designen. Op het internet heb je echter een paar prachtige plaatjes gevonden die je zo als gamesprite kunt gebruiken. Mag dat?

NEE.

Maar als je die plaatjes nu in photoshop inscant en de kleuren een beetje aanpast. Mag het dan?

NEE.

En als je de plaatjes nu uitprint en overtrekt en dan weer inscant, mag het dan?

NEE.

Mag je het plaatje dan natekenen in dezelfde stijl? NEE.

Het mag niet, tenzij je toestemming (schriftelijk) hebt van de oorspronkelijke maker. De regel is dat je niets mag kopiëren, namaken, overnemen dat eerder door iemand anders is gemaakt. Geen teksten, geen plaatjes, geen geluiden, geen programmacode.

Een game maken met Smurfen mag dus niet, ook niet als je die smurfen zelf hebt getekend. Ook niet als je ze Zmurfen noemt, dat lijkt te veel op het origineel. Maar een game maken met kleine paarse mannetjes die je Zmoeffies²⁵ noemt mag wel. En die Zmoeffies moet je zelf getekend hebben, niet een Smurfplaatje inscannen en paars maken.

Het lijkt misschien wat overdreven om zo'n punt te maken van de copyrights. Maar als je je game wilt publiceren zul je ervoor moeten zorgen dat alles eigen werk is. Of je je game nu gratis uitgeeft of er geld voor vraagt maakt niet uit. Hetzelfde geldt voor het gebruiken van software. Stel dat jij een 3d game maakt van software die, laten we zeggen, van "de vrachtwagen is gevallen". Zolang je die game niet publiceert, valt het niet op. Maar als je de game op het web gaat zetten, kom je in de picture en kan je problemen verwachten. Want hoe heb je die game kunnen maken zonder betaalde licentie? Het is ook zo onnodig, er is heel veel open source software beschikbaar om mee te werken.



'NEE'



"Hallo ik ben een Zmoeffies"



TIME REMAINING
14:57:23:12

Het werk van anderen kopiëren mag dus niet, ideeën jatten echter mag wel. In de hele ontwikkeling van videogames zie je dat er voortdurend ideeën gejat worden, waar iemand anders dan weer iets aan toevoegt.

EVOLUTIE VAN IDEEËN EN GAMES:²⁶

1. Space invaders: verdedig de aarde
2. Galaxian: de aanvallers komen nu ook recht naar beneden op je af
3. Malaga: de aanvallers vliegen nu allerlei patronen
4. Gorf: introductie van de "Boss": het vijandige moederschip
5. Defender: het scherm gedraaid, speler vliegt over vijandig terrein
6. Xevious: scherm teruggedraaid, zelfde concept als Defender
7. River raid: andere variatie, je vliegt nu in rivierbeddingen
8. Zaxxon: vertaling van het idee naar een scherm in perspectief
9. Paperboy: Zaxxon zonder geweld
10. Wolfenstein: Eerste 3d game vanuit first person perspectief

LEVEL 12: SAMEN OF ALLEEN

Voordat je in je eentje gaat beginnen: bedenk dat er een team van tientallen mensen jarenlang aan de release van nieuwe 3D games werken. Tenzij je geen slaap nodig

hebt, verder niets te doen hebt en 5 keer zo snel werkt als de gemiddelde game maker, kan je een spel maken op dat niveau dus wel vergeten. Natuurlijk zijn er talloze simpele maar toch leuke spelletjes te maken die je in een paar weken tot maanden in je eentje kan maken. Het leuke van alleen werken is dat je alles zelf kan doen: het idee bedenken, programmeren, designen, sounds maken, animaties maken, teksten schrijven. Het voordeel van werken in een groep is natuurlijk dat je veel meer kan bereiken. Daarnaast is het inspirerend om met mensen van verschillende achtergrond te werken.²⁷ Hieronder volgen twee interviews met ervaren gamebouwers. Als eerste Arno van der Hulst, die game designer en developer is sinds zijn vijftiende, daarna met Matthijs Beelen die het Project-Zero²⁸ leidt waarin vele liefhebbers een professionele game ontwikkelen.



Space Invaders

Quit game? Save game?

26. Voor een overzicht van de ontwikkeling van videogames zie: High Score door Rusel Demaria & Johnny L. Wilson

27. Op http://sourceforge.net/softwaremap/trove_list.php?form_cat=80 staan enkele duizenden game projecten waar je aan deel kan nemen. Als designer, programmeur, schrijver, vertaler enzovoort.

28. Project-Zero Development, <http://www.project-zero.info>, contact@project-zero.info



ZX-Spectrum 16K

INTERVIEW MET ARNO VAN DER HULST²⁹:

Hoe lang maak je al games en hoeveel heb je er gemaakt?

Ik hou me al sinds 1983 bezig met het ontwikkelen van gameconcepten. Ik was toen 15 jaar.

De concepten stuurde ik naar Philips. Ik was op dat moment de trotse eigenaar van een Philips VideoPac (Odyssey 2). Nadat Philips de stekker uit de VideoPac trok besloot ik maar om zelf mijn ideeën te gaan uitwerken.

In 1984 heb ik een ZX-Spectrum 16K aangeschaft. Ik begon met het ontwikkelen van games in Basic. Later ben ik gebruik gaan maken van software die deze Basic games omzette naar machinetaal. Omdat dit toch te veel beperkingen met zich meebracht heb ik mezelf machinetaal (Z80) aangeleerd. In 1987 heb ik mijn eerste 'echte' game geschreven. Dit was Fruitmachine. Nadat ik deze had afgerond, ben ik direct begonnen met het ontwikkelen van een verbeterde versie met voor die tijd een groot aantal nieuwe game noviteiten. Het resultaat was the Top Fruitmachine: een multi-level fruitkast met veel extra spelletjes.

Los van de door Codemasters uitgegeven shooter Tornado ECR heb ik mijn grootste succes op de ZX-Spectrum bereikt met het spel Block-Dizzy. Ondanks dat dit spel alleen als cover game is uitge-

geven heeft het wel de 86ste plaats behaald in de officiële Your Sinclair Reader's Top 100 allertijden³⁰.

In 1988 ben ik bij het multimedia bedrijf Bit-IC gaan werken. Voor dit bedrijf heb ik voor de zakelijke markt een groot aantal games in Macromedia Director ontwikkeld. Momenteel werk ik bij Capgemini als Multimedia adviseur en ontwikkel ik in mijn vrije tijd.

Softology

Voor de ZX-Spectrum (zie <http://www.worldofspectrum.org>)

- ★ Fruitmachine (1987)
- ★ Top Fruitmachine (1988) cover game op the Your Sinclair (spectrum magazine)
- ★ Brain Sport (1989) cover game op the Your Sinclair (spectrum magazine)
- ★ Tornado ECR (1991) uitgegeven door Codemasters
- ★ Block-Dizzy (1992) cover game op the Your Sinclair (spectrum magazine)

Voor de PC

- ★ Block-Dizzy (1998)

Macromedia Director

- ★ Moving Memory (1995-2005), o.a. Planet Internet, Z@pp, OCW, ABN AMRO
- ★ De Oliequizz (1998) voor Shell

29. ahulst@capgemini.com

30. <http://www.worldofspectrum.org/ys-top100.html>

- ★ Kerstspel (1998) voor Capgemini
- ★ Games voor de website “De Virtuele Kazerne” (1999) voor de Landmacht
- ★ Games voor de cd-rom “De Rommelzolder” (2002) voor Artsen voor Kinderen
- ★ Block-Dizzy (2005/6)

Hoeveel werk is het om een game te maken?

Dit is afhankelijk van de ontwikkeltaal en de complexiteit en omvang van het spel. Daarnaast maakt het nogal uit of je alleen ontwikkelt of dat je dit met een compleet team doet.

Hoe maakte je vroeger je games en hoe doe je dat nu? Waar ben je nu mee bezig?

Vroeger deed ik alles in mijn uppie. De ontwikkeltools waren toen zeer beperkt. Ik tekende bijvoorbeeld eerst mijn figuurtje op ruitjespapier en dan moest je het puntje voor puntje inprogrammeren. Nu heb je een scanner en veel betere grafische pakketten. Het ontwikkelen van een game in Director gaat daarnaast veel sneller dan als je iets in machinetaal ontwikkelt. Ik heb nu ongeveer 2 tot 8 weken nodig om iets goeds in elkaar te zetten.

Momenteel hou ik me bezig met de ontwikkeling van casual games. Dat zijn games die je online kunt spelen. Ik heb nu bijna de internetversie van Block-Dizzy klaar. Ik kan dus alweer gaan nadenken over het volgende spel. Belangrijk uitgangspunt bij mijn

game concepten is dat ook vrouwen het spel leuk en uitdagend moeten vinden. Ik vind namelijk dat de meeste gamebouwers te weinig aandacht aan deze doelgroep besteden.

Wat is je mooiste game en is die ook echt officieel uitgebracht? Wat is er bijzonder aan (vind er je zelf bijzonder aan)?

Dit is een lastige vraag. Mijn mooiste game vind ik misschien wel de Top Fruitmachine. De reden hiervoor is de playability. Toen ik dit spel schreef waren er al tientallen versies van de traditionele gokkast in gamevorm uit. Alleen leken al deze games sterk op elkaar. Ze bestonden allemaal uit één scherm. Het spel was vaak niet meer dan doelloos drukken op de start knop en hopen dat je een prijs wint. Mijn versie was gebaseerd op een bestaande gokkast met meerdere niveaus. Het speelveld had ik daarom uit vier schermen opgebouwd met op ieder scherm weer allerlei subgames. Ik heb in Nederland met dit spel aan een computerwedstrijd meegedaan. Tijdens de demonstratiedagen waren de mensen niet bij mijn computer weg te slaan. Een betere waardering voor je werk kun je niet krijgen.

Verder ben ik trots op mijn noviteit om in games een “uptodate high score” bij te houden. Dit houdt in dat de speler tijdens het spelen ziet op welke positie hij of zij in de high score staat. Dit zorgt voor extra

stimulans. Ik vind het bijzonder dat niemand dit geintje toepast want het is zeer effectief.

Wat is het belangrijkste aan een succesvolle game?

Ik ben zelf begonnen in de tijd dat vormgeving minimaal was. Ik vind vormgeving belangrijk maar het mag nooit de boventoon voeren. Uiteindelijk gaat het om het spelconcept en de uitvoering. Doel is om de speler continu te blijven prikkelen. Het is daarom ook belangrijk om goed te weten voor welke doelgroep je een spel ontwikkelt.

Welk advies heb je voor startende gamebouwers?

Ga niet als je een idee hebt direct aan de slag met de vormgeving en het schrijven van code. Werk eerst het idee verder uit. Toets hierna het concept op de volgende aspecten:

- ★ Is het doel van het spel duidelijk?
- ★ Zijn de regels van het spel simpel en duidelijk? Wanneer het spel te complex wordt, zal een groot aantal spelers het spel als niet leuk ervaren. Ga uit van het KIS model (Keep-It-Simple).
- ★ Biedt het spel voldoende variatie en uitdaging?
- ★ Is het spel te realiseren? Leg de lat niet te hoog. Zeker niet als het je eerste spel is. Een gamebouwer moet over doorzettingsvermogen beschikken want het begin is allemaal erg leuk en nieuw, maar het einde van het traject is meestal zwaar.

Waarom ben je games gaan maken en wat is er leuk aan?

In het maken van games kan ik al mijn creativiteit kwijt. Ik vind het leuk om met beeld en geluid te werken.

INTERVIEW MET MATTHIJS BEELLEN, PROJECTLEIDER VAN PROJECT-ZERO:

Söldner Wars is de game waar jullie aan werken. Söldner is begonnen als commerciële game, maar de fabrikant ging failliet. Jullie hebben het toen overgenomen. Hoe ging dat in zijn werk?

Ik zelf was al enige tijd bezig met het gratis ontwikkelen van mods, toen ik hoorde dat er een nieuw spel op de markt kwam genaamd Söldner Wars. Söldner had hele goeie mogelijkheden om te modden dus begon ik met het modden van deze game.

Na ongeveer 2 jaar van het maken van kleine mods kreeg ik te horen dat de ontwikkelaars ermee stopten. Op dat moment heb ik met de uitgever van de game contact gezocht en ze uitgelegd dat het zonde zou zijn om het spel zomaar dood te laten gaan. Na veel gepraat en tijd erin gestoken te hebben, kregen we de sourcecode van Söldner. Daarna is het steeds beter gegaan. Op dit moment zijn er zelfs al serieuze gesprekken om een nieuw spel te maken voor de uitgever. Daar hebben wij dus ja op gezegd. Want het mooiste wat

je kunt meemaken is een hobby uitbouwen tot een part time/full time job.

Als je zo'n kans krijgt die wij gekregen hebben, geef ik je wel als advies dat je het echt moet willen en kunnen. Anders is de gameindustrie wel een harde wereld. Bevalt het ze niet, dan is het geen gemaar: je bent weg.

Wat voor mensen (welke rollen, activiteiten) heb je in je team? Hoe groot is je team?

De mensen die in mijn team zitten zijn mensen uit verschillende beroepen en van verschillende niveaus en achtergrond. Ons team is opgedeeld in development (het maken van het spel) en promotion & PR (het aan de man brengen). Het allerbelangrijkste is dat deze 2 teams heel goed samenwerken. Een goede samenwerking voor elkaar krijgen, kostte in het begin veel tijd. Inmiddels doe ik zelf niks meer in development maar ben ik alleen nog met leidinggeven bezig. Veel tijd en arbeid maar zeker de moeite waard.

Het team is voor development als volgt opgebouwd:

- ★ Managing Director (dat ben ik)
- ★ Development directors: aan hen geef ik de opdracht en zij werken het uit
- ★ Verschillende lead developers: de mensen die het echte werk doen, en zich het meeste bezig houden met het spel zelf. C++, XML en Python programmeurs, 3d modellers, 2d artists, echt de vakmensen.

Ik moet er trouwens wel bij zeggen dat ook de leidinggevenden van alle technische dingen iets afweten. Soms helpen zij mee met de uitvoering en springen ze bij problemen bij.

★ QA Quality Assurance: deze mensen zijn voor mij toch een zeer belangrijk team. Zij testen de nieuwe releases. Daarnaast komen ze met verbeteringsuggesties aan. Zonder zulke mensen krijg je je spel niet goed.

Veel spelmakers zeggen, ik doe dit, en dat is mijn werk. Dit vind ik zelf een hele stomme opmerking. Zonder je team en de mensen kun je niks. Jij kunt misschien goede tekeningen maken, maar iemand anders moet het actualiseren in het spel. Hou daar goed rekening mee.

COMMUNITY AND PR

Dit is de afdeling van mijn collega. Zijn werk is het spel verkopen, lan parties regelen, nieuwe ideeën aanbrengen voor promotie. Ze hebben heel veel contact met game magazines. Hoe meer informatie we in die magazines krijgen hoe beter.

Ons team bestaat uit 25 man in development, 10 man in de community die de game in de lucht houden, en dan nog eens 50 man testers.

Het zijn allemaal vrijwilligers, maar wel professionele mensen. Wat doe je/doen ze in het dagelijkse



leven? Komen ze allemaal uit de IT wereld?

Ik heb 3 mensen die komen ook echt uit de IT wereld, de rest is... tja heel verschillend. Ik heb een lasser erbij zitten, constructiemedewerkers, mensen van overheid instanties, leger, bankwezen, studenten. Ja echt een bijelkaar geraapte groep. Het zijn wel mensen die weten wat ze doen en hoe ze het moeten doen.

Nu we dan bijna “officieel” nieuwe spellen gaan maken dat is iets wat het team echt oppept.

Wij hebben één regel gehad: samen maken we het en niet alleen. Dat geldt dus ook voor de inkomsten die we krijgen: het wordt verdeeld.

Hoe heb je deze mensen gevonden en hoe houd je contact met ze?

Mijn mensen heb ik bijna allemaal via de forums van het spel binnengehaald of via andere forums. Je moet zoeken op sites waar mensen met ideeën en vragen zitten, die willen ook graag meedoen met projecten. Contact houden doen we online, en tegenwoordig ook aan de telefoon. Online middelen die wij gebruiken zijn: icq, msn, yahoo, teamspeak, ventrilo, forums.

Zoek je nog nieuwe mensen? En zo ja, wat moeten die kunnen?

Dit is voor mij een vraag met een simpel antwoord: ja. Mensen zoek je altijd en zeker als ze gratis werk verrichten. Het belangrijkste criterium vind ik de

kennis die ze hebben. Daarnaast is het belangrijk dat ze dingen willen leren en niet te eigengereid zijn.

In het team zitten op dit moment twee mensen die wij opleiden om te kunnen programmeren. Toen ze begonnen, wisten ze er niks van, maar ze zijn zeer ijverig. Nu hebben ze mijn uitvoerende taken voor een groot gedeelte overgenomen.

Maar om concreet te zijn, mensen die wij altijd te kort komen, zijn:

- ★ c++ programmeurs
- ★ 3d modellers
- ★ Photoshoppers (voor de skins)
- ★ Xml en python programmeurs

Wat voor advies heb je voor beginnende game bouwers? Welke software zouden ze moeten beheersen?

Advies voor beginner, denk groot maar begin klein. Ga nooit de fout in door je game meteen heel groot aan te pakken. Je komt dan geheid jezelf tegen. Begin met kleine dingen. Maak het af en breng het naar buiten. Kijk hoe het publiek er op reageert en leer daarvan. Je leert ook veel door mods van andere makers te bekijken en te zien hoe zij het doen.

Programma's die echt belangrijk zijn voor de grote spellen zijn: 3ds max, Photoshop en Microsoft visual studio.net. Ga je een gameproject starten dan zijn programma's om je onderlinge communicatie te optimaliseren zeer belangrijk. Wij gebruiken

zelf verscheidene software voor dit doel: serverbox, mantis, forums, ftp, svn, teamspeak/ventrilo.

Hoeveel tijd gaat er in het maken van een game zitten (niet per se Söldner, maar in zijn algemeenheid)?

Ja dit is heel moeilijk te zeggen, om eerlijk te zijn is een spel nooit af. Maar een 3D klein schietspel is toch alles bij elkaar zeker een drie tot vier jaar werk. Dan heb ik het wel over een commercieel spel. Praten we over een mod, dan heb je in één jaar toch al een leuke modificatie van een spel.

Begin je met een nieuw spel dan is het opstarten en het daadwerkelijk beginnen het meeste werk. Voor het nieuwe spel ben ik nu al een jaar bezig met het schrijven van het game design document en het uitwerken van game strategieën. Daarna begint het bouwen pas.

Ken je nog andere teams die voor hun hobby aan games werken?

Ja ik ken er heel veel. De meeste zijn kleine projecten. Meestal hebben ze nog nooit een source gezien van het spel, en maken ze uitsluitend kleine mods.

Een echte grote onder deze groep is het team van Unreal Tournament: Red Orchestra.³¹ Dat is een top team met zeer goede mensen en echt mensen die het ver hebben geschopt. Zij zijn ook commercieel gegaan, net zoals wij doen. Maar het kost jaren voordat je

genoeg ervaring hebt dat je dat kan gaan doen.

Wat gaan jullie doen na Söldner Wars?

Momenteel zijn we bezig met een nieuwe patch (red: verbeteren van het spel). Het patchen is inmiddels bijna ten einde, en daarna gaan we zeker nog even door met het supporten van Söldner todat het nieuwe spel klaar is. Dan stoppen we met Söldner en gaan we verder met het nieuwe spel. Door deze tactiek te gebruiken, houden we de huidige spelers bij ons, die gaan zeker mee naar het nieuwe spel.



31. <http://www.redorchestra.clanservers.com/>

**STICHTING BEELDENDE
AMATEURKUNST**

De Stichting Beeldende Amateurkunst is het landelijk kenniscentrum en ondersteuningsinstituut voor de beeldende, audiovisuele en digitale amateurkunst. De SBA initieert en ondersteunt projecten waarbij de kwaliteit van de amateurkunstbeoefening bevorderd wordt. Zij richt zich daarbij vooral op het verbeteren van inhoudelijke en organisatorische deskundigheid, het versterken van de regionale infrastructuur en het verschaffen van informatie, documentatie en faciliteiten.

De SBA werkt nauw samen met amateurkunstverenigingen en hun overkoepelende organisaties, met de overheid, met landelijke en provinciale ondersteunende instellingen, met de plaatselijke centra voor de kunsten, met docenten, met professionele kunstenaars en kunstinstellingen.

SBA

Boothstraat 3
3512 BT Utrecht
telefoon (030) 234 22 11
fax (030) 234 23 82

e-mail: bureau@sbakunst.nl
www.amateurkunst.net

EERDER VERSCHENEN KATERNEN

Dit is de 21ste uitgave in de reeks SBA-katernen

1 Hetzelfde zien, maar dan anders

Fotografie is een inspirerende en fascinerende beeldtaal die door iedereen aan te leren is. Uitgegeven bij gelijknamige videoband. (september 1996)

2 Film en auteursrecht (uitverkocht)

Uiteenzetting over auteursrechten en naburige rechten, speciaal toegespitst op de amateurfilmer. (oktober 1996)

3 Goed gezien goed bekeken

Kanttekeningen bij het maken van tentoonstellingen door beeldende amateurs. (april 1997)

4 De fotowedstrijd, een handleiding

Over het organiseren van fotowedstrijden. Waar dien je rekening mee te houden, welke problemen kun je tegenkomen? (mei 1997)

5 10-slagen-analyse

Een analysemethode voor filmmakers. (september 1997)

6 Onder het mes

Zes amateurfilms in analyse. (november 1997)

7 Verhalen van textiel

Over verschillende vormen van textielkunst. Een verhaal over passie, inspiratie en onderzoek. Uitgegeven bij gelijknamige videoband. (september 1998)

8 Jureren en selecteren van foto's, een analyse

Overwegingen bij het jureren van fotowedstrijden, advies aan juryleden. (december 1997)

9 Eerst het verhaal

Schrijfimprovisaties voor filmamateurs. (november 1998)

10 Kunst in vrije tijd

Een overzicht van landelijke amateurorganisaties op het gebied van de beeldende, textiele en audiovisuele vormgeving. (december 1998)

11 10x Computer...anders

Tien creatieve computergebruikers aan het woord. (april 1999)

12 Onder het mes deel II

Vier amateurfilms in analyse. (juni 2000)

13 De museumcollectie als inspiratiebron

De verborgen textielschatten uit het Catharijneconvent. (juli 2000)

14 Onder het mes deel III

Dansfilms in analyse. (april 2001)

15 Meesterlijk gekopieerd... en dan?

Amateurschilders kopiëren kunst uit musea.
(mei 2001)

16 Theaterfotografie voor amateurs

Een introductie (januari 2002)

17 Verhaal in beeld

Handleiding voor documentaire fotografie.
(februari 2003)

18 Souvenir van de Zaanstreek

Een intercultureel handwerkproject. (maart 2003)

19 Van Bisdom en Baronie

De museumcollectie van Breda's museum als
inspiratiebron voor textiel. (maart 2005)

20 Sportclub in beeld

Fotografie in sportverenigingen.
Handleiding voor fotograaf en sportclub. (juni 2005)

21 Start Here

Gamedesign voor beginners. (mei 2006)

COLOFON

Dit boekje is een uitgave van de
Stichting Beeldende Amateurkunst.

Graphics op blz 3 met permissie gebruikt uit "De-
signing Arcade Computer Game Graphics" van Ari
Feldman (www.flyingyogi.com)
De beelden op blz 16 en 17 zijn met toestemming ge-
bruikt uit Söldner wars: (www.project-zero.info)

Tekst: **Wouter Baars**

Meer informatie:

www.gamesmaken.startpagina.nl,

www.wouterbaars.net

Redactie: **Ben Kamphuis, Leo van der Meer en
Paulien Franken**

Productie: **Paulien Franken**

Vormgeving: www.overburen.nl

Druk: **Ter Burg, Alkmaar**

Mei 2006

ISBN-10: 90-77488-06-5

ISBN-13: 978-90-77488-06-5

Deze digitale versie van Start Here is een heruitgave door www.gameskool.nl



Start Here van Gameskool is in licentie gegeven volgens een [Creative Commons Naamsvermelding-NietCommercieel-GeenAfgeleideWerken 3.0 Unported](https://creativecommons.org/licenses/by-nc-nd/3.0/) licentie.