



ARCADE GAME ANIMATIE

Wat is een animatie?

Een animatie is een reeks afbeeldingen die achter elkaar wordt afgespeeld zodat het lijkt alsof er iets beweegt. Je ziet animaties regelmatig in films waarbij mooie 3D-effecten van het beeld springen, maar er bestaan ook papieren animaties. Deze klassieke papieren animaties werden voor het eerst getekend op de hoek van een papiertje. Er worden dan kleine poppetjes getekend die op iedere papiertje een klein beetje van elkaar verschillen. Zodra je de papiertjes snel achter elkaar laat zien lijkt het poppetje te bewegen. Deze techniek van kleine tekeningetjes achter elkaar laten zien om een illusie van beweging te tonen, heet animeren.

In games wordt deze techniek ook toegepast, maar in plaats van te tekenen op papier wordt er getekend op een computer. Zodra je op de computer gaat animeren heten deze kleine tekeningen: *frames*. Als frames heel snel achter elkaar worden vertoond, met slechts hele subtiele verschillen, dan zien wij mensen dit als een beweging.

Dit hoofdstuk introduceert de basisbeginselen voor het maken van een animatie zodat jij een animatie kan maken voor in je eigen arcade game.

Basisbeginselen voor animatie

Om een animatie te maken moet je leren hoe je bewegingselementen kan verdelen in hun basis componenten. Dit betekent dat je al je getekende tekeningen, *frames*, moet opstellen in een logische volgorde. Voordat je dit kan doen moet je eerste twee dingen onder de knie krijgen. Ten eerste moet je de beweging van objecten goed leren observeren. Daarnaast moet je de beweging, die je een object wil laten maken, kunnen opdelen in meerdere losse basisbewegingen.

Ieder object heeft eigenaardigheden als het beweegt. Sommige van deze eigenaardigheden zijn heel subtiel, terwijl andere juist erg opvallend zijn. Er zijn een paar karakteristieken van bewegingen die je in de gaten zal moeten houden voor dat je begint met animeren. Let op dingen als:

- Bewegingslijnen
- Key-frames en in-betweens
- Gewicht en zwaartekracht
- Beweeglijkheid
- Secundaire acties
- Loops
- Framerate

Bewegingslijnen

Een bewegingslijn is een onzichtbare lijn die een object volgt als hij een serie van opeenvolgende bewegingen maakt. Een vliegende kogel heeft bijvoorbeeld een rechte lijn en een stuitende bal heeft een golvende bewegingslijn. Bewegingslijnen moeten consistent zijn met het werkelijke gedrag van een object om een mooie animatie te krijgen. Je kunt een bewegingslijn gebruiken om te bepalen of het wel of niet een realistische animatie is.

De beste manier om een bewegingslijn van een object te volgen is om het zwaartepunt van het object te bepalen. De plek van het zwaartepunt varieert per object. Om je te helpen het zwaartepunt te vinden staan er in tabel 1 een paar voorbeelden voor een aantal objecten. Met deze informatie kun je van allerlei soorten objecten het zwaartepunt vinden.

TABEL 1: Locatie van het zwaartepunt

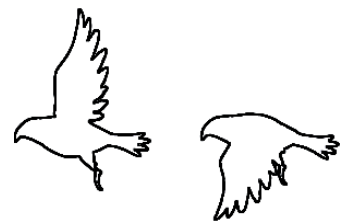
Type object	Zwaartepunt
Twee potige dieren	Hoofd
Vier potige dieren	Borst
Vliegende dieren	Romp
Mensen	Hoofd
Insecten	Romp
Ruimteschepen en vliegtuigen	(Vliegtuig)-romp
Tank en voertuigen	Geschutskoepel of carrosserie

Key-frames

De meeste mensen zijn bewust van extremen die voorkomen tijdens beweging, zoals opvallende dingen als het slaan van vleugels. De vleugel van een vogel wijst op en dan weer neer. Deze twee standen zijn de extremen in de vlieganimatie van de vogel. Tijdens het animeren horen deze extremen bij *key-frames*. Het bepalen welke frames key-frames zijn is ontzettend belangrijk.

Er is een directe relatie tussen het aantal key-frames dat gebruikt wordt en het vlotte verloop van de animatie. Des te meer key-frames er gebruikt zijn, des te vlotter de animatie loopt. Weinig key-frames zorgen juist voor een houterige animatie.

Key-frames zijn het effectiefst als ze worden geplaatst bij erg overdreven elementen, want dit zijn de meest cruciale bewegingen in een animatie. De opvallende en overdreven elementen in je animatie kunnen je helpen om het begin-, midden- en eindpunt te bepalen. In Figuur 9-3 zie je het voorbeeld van twee key-frames van de vliegende vogel. Let op hoe deze twee key-frames de twee meest extremen weergeven; de vleugel omhoog en de vleugel omlaag. Deze twee houdingen zijn elkaars uiterste en zijn daarom ideaal om te gebruiken als key-frames.



Het is belangrijk om je te realiseren dat des te meer key-frames een animatie heeft, des te meer tijd het kost om het te tekenen en des te complexer het ontwerp zal zijn. Complexe animaties hebben meer kans op

foutjes. Des te meer foutjes, des te slechter de kwaliteit van je animatie natuurlijk is. Maak het jezelf dus niet meteen te moeilijk en begin met het animeren van objecten die duidelijke begin-, midden –en eindpunten hebben in hun beweging.

In werkelijkheid is het vaak wel duidelijk hoeveel key-frames je zult moeten gebruiken. Er zijn veel situaties waar je met weinig key-frames goed weg kan komen. In de meeste arcade games kun je 2-3 key-frames per object gebruiken.

Bekijk Tabel 2 voor wat suggesties over het gebruik van key-frames in arcade games.

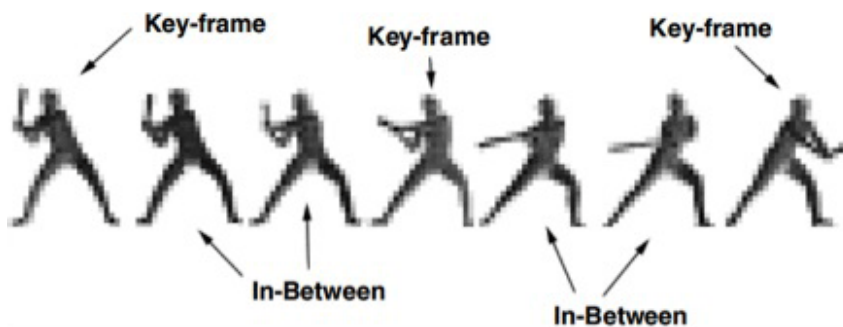
TABEL 2: Hoeveelheid key-frames per object.

Type arcade game	Aantal aangeraden key-frames
Pong	2 per geanimeerd object
Doolhofspel	2-3 per geanimeerd object
Puzzels	2-3 per geanimeerd object
Shooters	2-4 per geanimeerd object
Platform	2-6 per geanimeerd object

Nadat je de key-frames hebt vastgesteld moet je de *in-between-frames* toevoegen. In-betweens zijn frames die gebruikt worden om de tijd tussen twee key-frames glad te strijken.

In een animatie zijn key-frames belangrijk om de primaire bewegingen in een animatie aan te geven. In-betweens zijn verantwoordelijk voor het vlotte en gladde verloop van de animatie en zorgen er dus voor dat je animatie niet zo houterig loopt. Het toevoegen van frames met subtiele verschillen tussen key-frames creëert in-betweens. Dit proces heet *tweening*.

Figuur 1 geeft een voorbeeld hoe in-betweens tussen key-frames in worden gebruikt. Dit voorbeeld gaat over een man die met een knuppel slaat. Het begin van de serie is een man die klaar staat om te gaan slaan, het middelste frame laat zien dat hij de knuppel naar voren beweegt, en het laatste frame is de eindpose van de man. Dit zijn key-frames van de animatie terwijl alle tussen liggende frames in-betweens zijn.



FIGUUR 1: Voorbeeld Key-frame en In-between

Het creëren van in-betweens is één van de meest tijdrovende aspecten van het animeren van een arcade game. Sommige software zoals Flash kan de tweening (het maken van 'In-Betweens') automatisch voor je doen bij simpele animaties.

Het bepalen van het aantal benodigde frames is erg belangrijk voor het creëren van animaties en kost enige ervaring om goed onder de knie te krijgen. Om je te helpen kun je kijken in tabel 3 zodat je een beetje een idee krijgt over hoeveel frames je nodig hebt voor je animatie.

TABEL 3: Aantal frames per beweging

Beweging	Minimum aantal frames	Maximum aantal frames
Rennend 4-potig dier	4	16
Bijtend dier	2	5
Kruipen	2	12
Explosies	5	16
Vallen	3	5
Vliegen	2	12
Springen	2	10
Schoppen	2	6
Slaan	2	6
Draaien	4	16
Rennen	2	12
Zwaaien (een object)	2	8
Gooien (een object)	2	6
Vliegend voertuig	2	4
Bewegend voertuig	2	8
Lopen	2	12

Gewicht en zwaartekracht

Als je probeert realistische animaties te maken moet je nadenken over hoe gewicht en zwaartekracht je object beïnvloeden. Je moet goed kijken naar hoe de krachten van de natuur werken op de animatie die je maakt. Een van de belangrijkste dingen is gewicht, want dat beïnvloed snelheid. Neem bijvoorbeeld een groot object, des te zwaarder dit object is, des te langzamer het zal bewegen. Een olifant rent minder snel dan een cheeta.

Naast de invloed op snelheid, heeft gewicht ook invloed op hoe gemakkelijk een object beweegt. Vergelijk maar eens een snelle raceauto met een oorlogstank. De oorlogstank dendert langzaam door een veld vol hobbels heen terwijl een raceauto voorbij zoeft over een gladde snelweg. Dit komt omdat een raceauto veel minder zwaar is en zich makkelijker over glad terrein heen beweegt.

Dan is er ook nog zwaartekracht. Zwaartekracht beïnvloedt ook de beweging van een object. Alhoewel alle objecten even veel last hebben van zwaartekracht, vallen sommige objecten sneller dan andere. Dit komt door de luchtweerstand. Vergelijk bijvoorbeeld hoe snel een bom naar beneden valt met een veertje dat langzaam omlaag dwarrelt. Onthoud dat mensen niet zo snel voor de gek te houden zijn. Als iets onrealistisch beweegt zullen spelers dit door hebben en je game daardoor misschien wel minder leuk vinden. Tenzij het natuurlijk een bewuste keuze is om iets onrealistisch te doen (die dan ook consistent in je hele game doorgevoerd moet zijn).

Beweeglijkheid

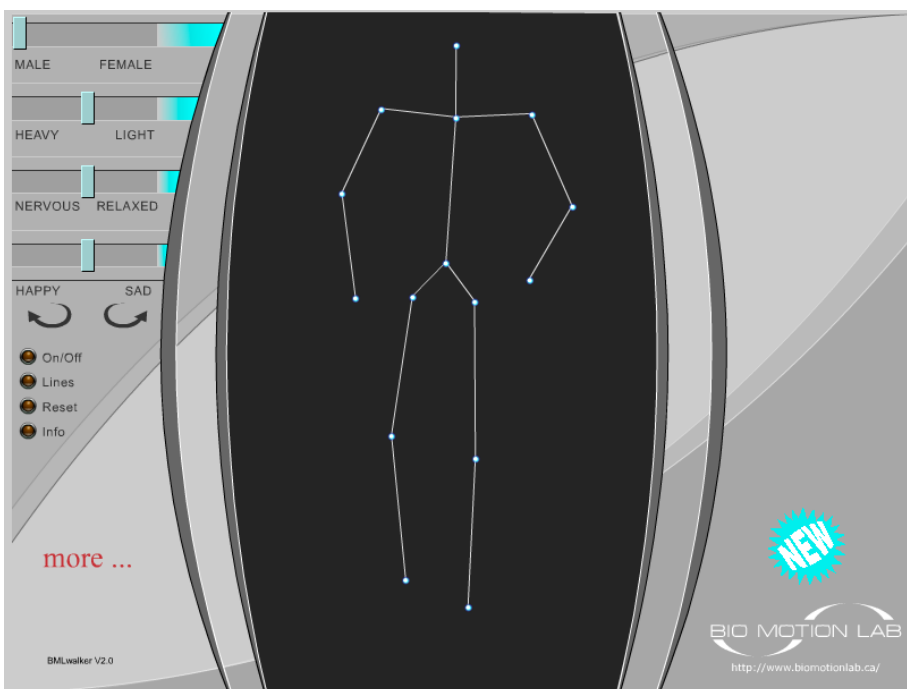
Beweeglijkheid is essentieel voor overtuigende animaties en zeker voor complexe dingen. Bijvoorbeeld objecten met een skelet zoals dieren en mensen. Animaties voor deze objecten zijn zonder beweeglijkheid al vaak stug en stijf.

Het belangrijkste bij het toevoegen van beweeglijkheid is het plannen en het observeren van hoe een object zich gedraagt als het zich beweegt. Wanneer je een skelet van iets animeert moet je kijken welk deel van het lichaam de beweging aanstuurt en welke delen de beweging volgen. Als jij je bovenbeen optilt stuurt je been de beweging en je voet volgt, aangezien je voet aan je been vast zit.

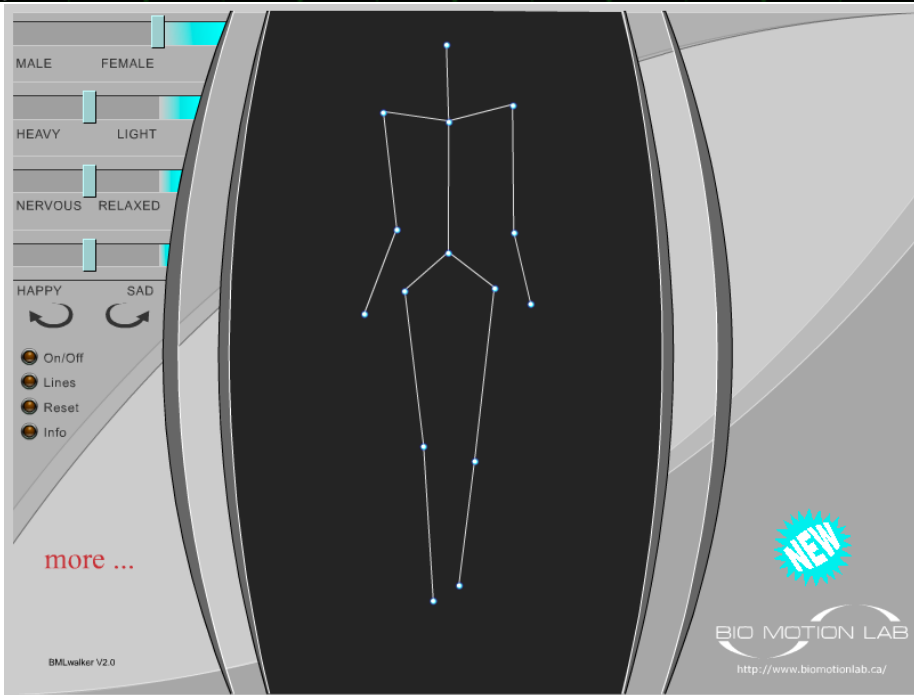
Wanneer je probeert beweeglijkheid toe te passen in je animatie moet je wel rekening houden met de limieten van de anatomie van het object. Probeer dus bijvoorbeeld nooit om een kniegewricht van een menselijk skelet de andere kant om te buigen.

De beweeglijkheid is bepalend voor het 'karakter' van je gamefiguur. Als je je gameheld stoer wilt laten zijn dan beweegt hij anders dan wanneer het gaat om een gehaast en zenuwachtig persoon. Goed kijken naar bewegingen is hier weer de sleutel. Kijk bijvoorbeeld eens naar het Biomotionlab (www.biomotionlab.ca).

Hier zie je de subtiele veranderingen in beweeglijkheid als je een mannelijke of vrouwelijke held wilt animeren. Of het verschil tussen blij en verdrietig, gespannen en ontspannen, zwaar en licht.



FIGUUR 2: Mannelijk



FIGUUR 3: Vrouwelijk

Secundaire acties

Secundaire acties zijn heel erg belangrijk in het animatieproces omdat zij iets extra's toevoegen aan hoe realistisch de animatie er uitziet. Eigenlijk zou je ieder deel dat op zichzelf beweegt een secundaire acties kunnen noemen. Dit kan van alles zijn, van haren die wapperen als iemand loopt tot ogen die knipperen. Secundaire acties zijn niet gelimiteerd tot kleine details zoals kleding en haar. Het is belangrijk dat je realiseert dat secundaire acties virtueel van alles kunnen zijn, zolang de focus er maar niet op ligt. Secundaire acties kunnen een animatie wel realistischer eruit laten zien. Kijk bijvoorbeeld eens naar het rennende meisje uit de animatie "DOT" van de Aardman studio's en let met name op de secundaire actie van het wapperende haar:



FIGUUR 5: Scene uit DOT (http://www.youtube.com/watch?v=CD7eagLI5c4&feature=player_embedded)

(NB. Het poppetje in deze animatie is slechts 9 milimeter groot en handbeschilderd)

Loops (herhalingen in animaties)

In een animatie zijn *loopjes* de herhalende frames van bewegingen die veel geanimeerde objecten maken. Zodra je beter leert te observeren kom je er al snel achter dat veel objecten in loopjes bewegen. Loopjes kunnen worden gezien als tijdbespaarders. Ze helpen ons om tijd te besparen omdat we geen nieuwe frames meer hoeven te tekenen. Zonder loopjes heb je bijvoorbeeld honderden frames nodig om een vogel over het scherm te laten vliegen. Maar met loopjes kun je de frames gewoon hergebruiken. Stel nou dat je 30 frames nodig had voor je vogel, dan hoef je nu slechts een stuk of 4 nieuwe frames te maken die de belangrijkste punten van de beweging weer geven. Je gebruikt dus de 30 frames die je al had en die wissel je af met die nieuwe 4 frames die je had getekend.

Figuur 3 laat een voorbeeld zien van hoe loopjes en loops samenwerking in animaties. Hier zijn frame 1 en 5 loopjes omdat ze het zelfde frame herhalen. Als de serie het vijfde frame vertoont springt hij terug naar frame 1 om zo voor een illusie te creëren van een voortdurende beweging.



FIGUUR 6: Voorbeeld van een loop

Je moet het jezelf niet altijd te makkelijk maken want je moet wel onthouden dat niet alle objecten een constante beweging nodig hebben. Een speler heeft al snel door dat iets zich herhaalt en je game kan

daardoor een beetje simpel lijken. Er zijn genoeg animaties die helemaal geen loops gebruiken.

Framerate

Ieder geanimeerd object wordt weergegeven met een bepaalde *framerate*. We meten tempo in *frames per seconde (FPS)*: beelden per seconden. Het menselijk oog neemt beweging, of de illusie van beweging, waar zolang er meer dan 12 frames per seconde gebruikt worden. Normaal gesproken is de animatie het mooiste als je heel veel frames per seconde gebruikt.

Tabel 4 vergelijkt de meest voorkomende animatie *framerates* voor verschillende vormen van animatie.

TABEL 4: Vergelijking van meest voorkomende frame rates.

Type animatie	Frames per seconde
Fast-action arcade game	15-30
Bioscoopfilm	24
Televisie	30

De meeste game ontwikkelaars vinden 15 FPS de minimum acceptabele framerate voor de meeste arcade games, terwijl 20-30 FPS het ideaal is. Sommige arcade game genres hebben een hogere framerate nodig dan anderen. Het ligt er namelijk helemaal aan wat de stijl van je game is. Als je iets heel realistisch wil laten zien heb je een hoge FPS nodig, terwijl iets wat zich als een robot beweegt juist weinig FPS nodig heeft. In Tabel 5 staan een aantal framerates van de meest voorkomende soorten arcade games.

TABEL 5: Meest voorkomende arcade game framerates

Type arcade game	Animatie framerates (FPS)
Pong	15-30
Doolhofspellen	20-30
Puzzels	9-15
Shooters	20-50
Platform	20-30

NB. De framerate wordt niet bepaald door uit hoeveel plaatjes je animatie bestaat maar wel uit hoe snel je computer is. Oudere computers hebben niet genoeg rekenkracht om veel verschillende animaties snel genoeg op het scherm te laten zien. Als je merkt dat je game gaat knippen, zul je dus het aantal bewegende beelden in het scherm omlaag moeten brengen of andere rekenintensieve taken moeten verminderen. Ook het speelscherm verkleinen (bv 800 x 600 pixels in plaats van 1024 x 768) helpt vaak om de framerate van je game omhoog te krijgen.

Basistechnieken voor een game animatie

Geanimeerde objecten zijn een belangrijk onderdeel van een game. De animaties zorgen ervoor dat alle objecten een eigen unieke stijl hebben. Iedere karakter beweegt anders, ieder ruimteschip vliegt anders, en iedere bom explodeert weer anders. Alle verschillende animaties zorgen er voor dat jouw game er aantrekkelijk uitziet voor een speler. Toch zijn geanimeerde objecten in een game opgebouwd uit dezelfde basiselementen. Het is belangrijk dat je doorkrijgt hoe deze elementen werken zodat je een goed inzicht krijgt in hoe een animatie is opgebouwd.

Je kan de basistechnieken opdelen in twee groepen:

- Basistechnieken voor grote acties
- Basistechnieken voor kleine acties

Basistechnieken voor grote acties

Onder deze categorie vallen de zes veel voorkomende basistechnieken. De technieken die hier beschreven worden zijn vrij simpel en daarnaast toepasbaar voor zowel karakters als mechanische objecten. Hieronder staan zes basistechnieken die iedere animator vaak nodig heeft:

- De cilindertechniek
- De rotatietechniek
- De verdwijningstechniek
- De glinstertechniek
- De kniptechniek
- De groei -en krimptechniek

De cilindertechniek

De cilindertechniek wordt gebruikt om te zorgen dat het lijkt alsof objecten ronddraaien. In arcade games kunnen dit van allerlei dingen zijn. Denk maar eens aan kogels, raketten en wielen.

De cilindertechniek is een van de makkelijkste animatie technieken om onder de knie te krijgen. Er zitten namelijk geen drastische verschillen tussen de frames, om het draaiende effect te creëren (bijvoorbeeld een rugbybal die gegooid wordt).

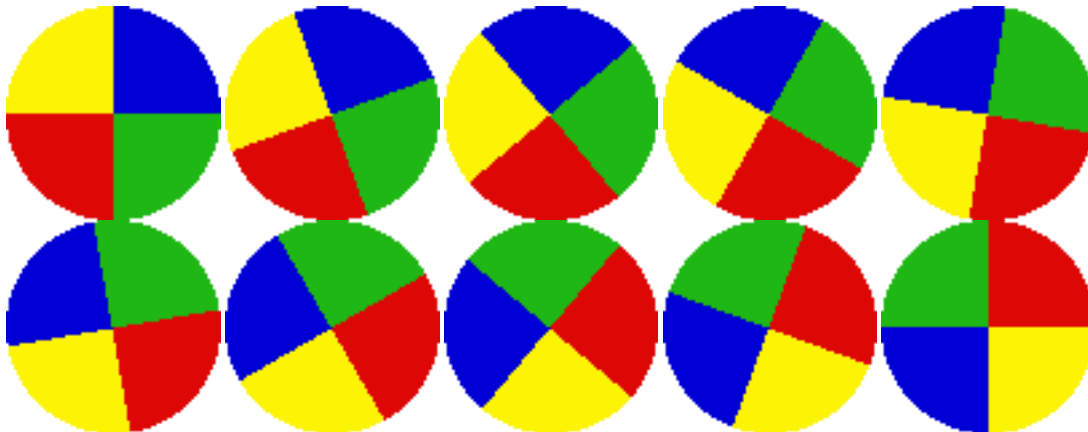


FIGUUR 7: Voorbeeld van een cilindrische animatie.

De rotatietechniek

De rotatietechniek wordt vaak beschreven als een variatie van verschillende draaiende objecten. Het wordt vaak verward met het cilindrische techniek maar het is toch echt iets anders: je kijkt vanuit een andere hoek naar je animatieobject.

In arcade games wordt een roterende animatie gebruikt voor dingen als rond draaiende tankgeweren tot rond slingerende asteroïden.



FIGUUR 8: Voorbeeld van een stukje uit een roterende animatie (Bron: GifBuilder).

Eind jaren negentig van de vorige eeuw kwam de zwitser Yves Piaget met het programma GifBuilder. Het programma is nog steeds gratis te downloaden. In de bijgeleverde documentatie zit de strandbal. Een simpele animatie bestaande uit 18 frames die een rondslingerende 4kleurige bal laten zien.

De rotatietechniek van animeren is erg populair en makkelijk. Een object maakt een volledige draai wanneer het van 0° - 360° graden om zijn eigen as is gedraaid. Je moet er dus simpelweg voor zorgen dat het object, in kleine stapjes, in totaal 360° graden draait.

Roterende animaties kunnen zowel met de klok mee, als tegen de klok in draaien. Je animatie gaat er soepel uitzien als je hem opdeelt in 360 kleine stapjes, maar meestal is het voldoende om het te doen in minder stappen. De hoeveelheid stappen die je nodig hebt om jou object mooi te laten draaien is verschillend per object. Het ene object zal minimaal maar 4 frames nodig hebben en een ander object wel 360.

De verdwijningstechniek

De verdwijningstechniek wordt natuurlijk vaak gebruikt om dingen van het scherm te laten verdwijnen. Zo zie je vaak in arcade games dat wanneer een speler dood gaat, het karakter langzaam wegvaagt van het scherm en er een nieuw spelkarakter in beeld verschijnt.

De drie meest voorkomende verdwijningmethoden zijn smelten, oplossen en het wegvagen van kleuren.



FIGUUR 9: Een voorbeeld van een explosie die oplost. Let op: een ontploffing lost op van binnenuit. Bron: www.flyinyogi.com

De Glinstertechniek

Glinsteringen worden vaak gebruikt in arcade games om te zorgen dat iets glimt of flikkert. Denk bijvoorbeeld aan glinsterende juwelen, brandende fakkels en flikkerende sterren. Glinsteringen kun je maken door kleuren, per frame van de animatie, te veranderen. Deze kleurverandering kan slechts gaan om

specifieke pixel, maar je kan ook meerdere pixels veranderen voor een nog heviger glittereffect.

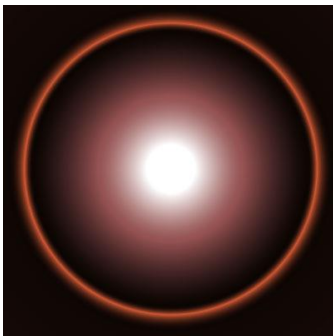
Knipperende effecten zijn makkelijk en snel te maken en geven je animatie echt iets extra's. Om een mooie glinstering te maken moet je vier dingen in de gaten houden:

1. De juiste selectie van kleuren
2. De grootte van de glinstering
3. Het aantal benodigde frames
4. De positie van de glinstering op het object

In de meeste gevallen heb je intense kleuren nodig met een hoog contrast. Zodra je hebt geanimeerd zorgen deze contrasterende kleuren er voor dat je glinstering echt van het beeld af springt en het dus duidelijk te zien is.

De grootte van de glinstering heeft een groot effect op de geloofwaardigheid van hoe je geanimeerde object er uitziet. Een glinstering die te klein is in omvang is te subtiel om goed te zien, terwijl een te grote omvang zorgt voor een ongeloofwaardig effect. Om het beste resultaat te krijgen moet je tussen deze twee extremen in zitten. In het begin zal de balans nog moeilijk te vinden zijn, maar oefening baart kunst!

Glinsteringen horen over het algemeen heel kort te duren en daarom moet je zorgen dat je slechts een paar frames per glinstering benut. Ideaal gezien zul je twee tot vijf frames nodig hebben. Zodra je er meer of minder gebruikt zul je zien dat het er onrealistisch uitziet.



FIGUUR 10: Glinstereffect gemaakt in Illustrator.

Bron:<http://www.slicktutorials.com/flare-tool-fun>

Tenslotte kan ook nog de positie van de glinstering je animatie maken of breken. Om er zeker van te zijn dat je glinstering er goed uitziet moet je hem op een logische plek plaatsen. Net zoals bij andere onderdelen van het animeren komt er bij het maken van een glinstering natuurkunde kijken. In dit geval moet je goed kijken naar de lichtval. Neem bijvoorbeeld een diamanten ring. Deze ring glinstert op de plek waar het licht valt. Dus stel je voor dat er een lamp schijnt op de linkerhoek van de ring. Je moet dan zorgen dat de glinstering dan ook in de linkerhoek geplaatst is. Plaats de glinstering dus niet in het midden van die ring of aan de rechterkant, want anders zie je meteen dat er iets niet klopt.

De kniptechniek

De kniptechniek is een van de populairste animatie technieken voor arcade games. Het wordt gebruikt voor effecten die variëren van loopanimaties tot bijt-animaties van allerlei wezens. Deze basistechniek simuleert een knippende schaar, vandaar ook de naam.

Naast het feit dat deze techniek erg populair is, is hij ook nog heel erg simpel om toe te passen. Als je de kniptechniek gebruikt begin je met een gesloten object dat langzamerhand zichzelf opent. Het werkt door gebruik van extreme veranderingen tussen de frames. Deze veranderingen houden onze ogen voor de gek en hierdoor zien wij de veranderingen als beweging. Deze techniek is ideaal voor animaties die weinig frames nodig hebben.

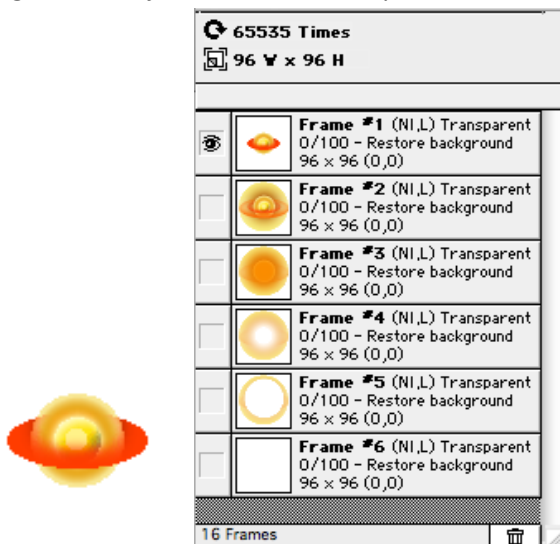


FIGUUR 11: Voorbeeld van de kniptechniek: Pac Man (gemaakt door George Williams)

De groei -en krimptechniek

De groei –en krimptechniek wordt toegepast als er een object groter of kleiner wordt. Je gebruikt het bijvoorbeeld voor een explosie, die klein begint en steeds groter wordt, of bij groeiende game karakters. Ook kun je dingen op de speler af laten vliegen. Hoe groter je een object laat worden, hoe groter de illusie is dat het object dichterbij de speler komt.

In figuur 12 zie je een voorbeeld explosie die steeds meer uitzet, gemaakt in GifBuilder



FIGUUR 12: voorbeeld van een groeiend object (in dit geval een ontploffende ster)

Basistechnieken voor kleine acties

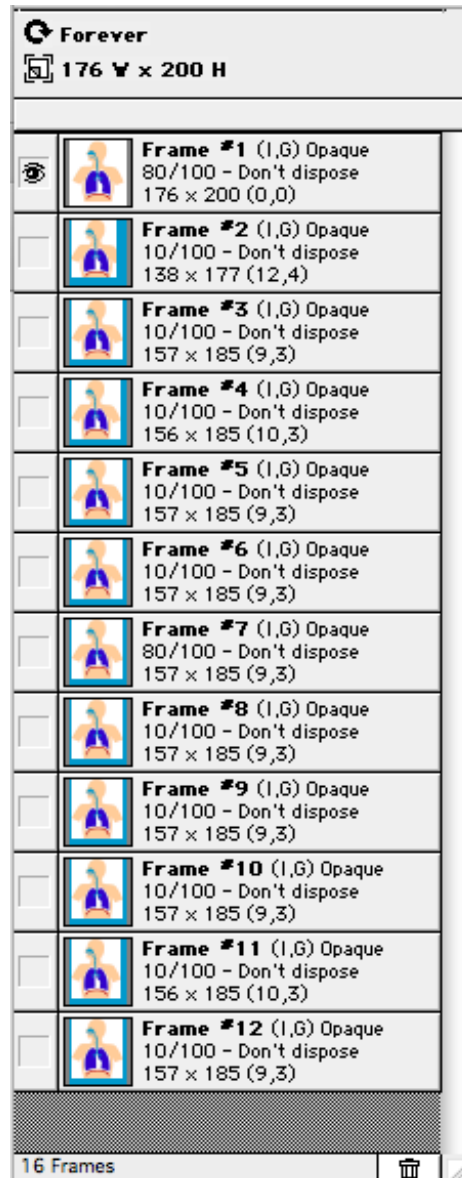
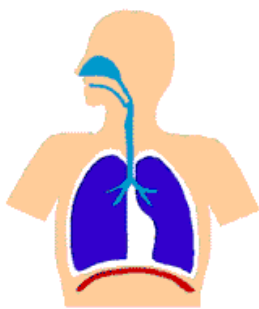
Deze set van animatietechnieken wordt in veel type arcade games gebruikt, maar komt minder vaak voor dan de technieken die hiervoor zijn beschreven. De technieken die nu worden beschreven kunnen worden toegepast op zowel karakters als mechanische objecten.

- De pomptechniek
- De knijptechniek
- De slingertechniek
- De schuiftechniek
- De open –en sluittechniek
- De stuitertechniek

- De stomptechniek

De pomptechniek

Objecten die een de pomptechniek gebruiken bewegen op en neer of van links naar rechts tijdens hun animatie. Het wordt vaak gebruikt voor machines die op en neer pompen, maar ook voor bijvoorbeeld voor een borstkas van een ademend karakter.



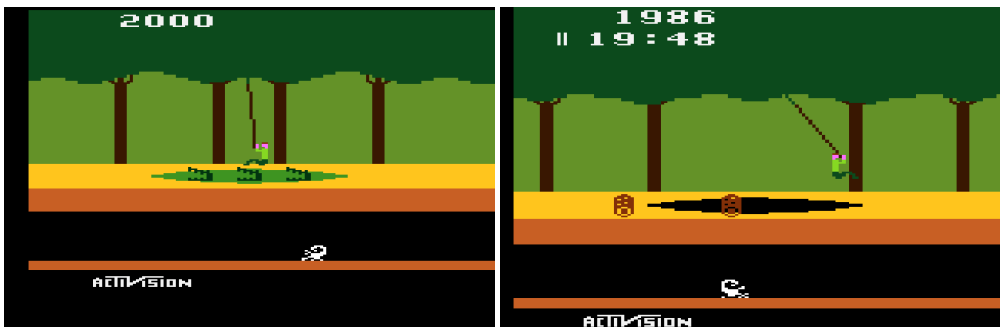
FIGUUR 13: Animatie van een borstkas, gemaakt in GifBuilder. Let op het rode lijntje onder de longen, wat pompt (middenrif).

De knijptechniek

Objecten die de knijptechniek gebruiken rekken uit en krimpen weer in. Denk bijvoorbeeld aan een accordeon die je uit trekt en weer terug duwt om muziek te maken.

De slingertechniek

Objecten die de slingertechniek gebruiken zwaaien net als een klepel van een klok heen en weer. De slingertechniek is eigenlijk een variatie op de kniptechniek en wordt gebruikt voor zowel organische als mechanische objecten.



FIGUUR 14: Slingertechniek

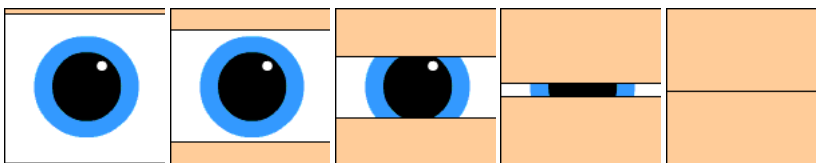
De schuiftechniek

De schuiftechniek wordt gebruikt voor het verschuiven en wegglijden van objecten. Het wordt vaak gebruikt voor bewegingen zoals lopen en kruipen, die slechts een paar frames gebruiken.

De schuiftechniek werkt door gebruik te maken van extremen in de reeks met frames. Als benen van een karakter heen en weer schuiven denken onze hersenen dat het karakter loopt, maar in feite blijft het karakter op dezelfde plek in het scherm staan. Animeren gaat allemaal om de kunst om iemands ogen voor de gek te houden en hem te laten denken dat iets beweegt.

De open –en sluittechniek

Zoals de naam al zegt heeft een object dat de open –en sluittechniek gebruikt twee standen: open en dicht. Je kan deze techniek bijvoorbeeld gebruiken om deuren te openen en te sluiten, of om kleine details toe te voegen aan je karakter, door bijvoorbeeld zijn ogen te laten knipperen.



FIGUUR 15 Voorbeeld van een knipperend oog die gebruikt maakt van de open –en sluittechniek.

De stuitertechniek

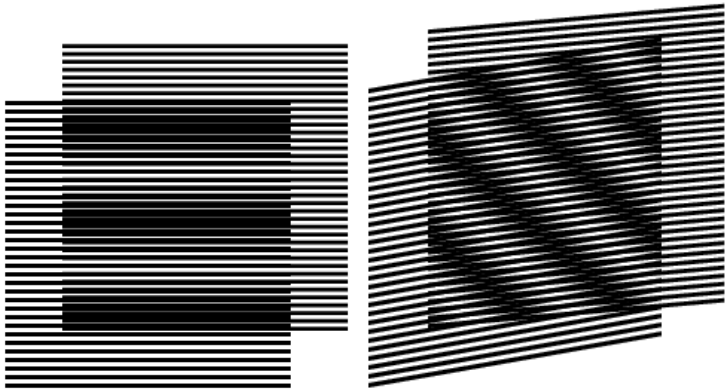
De stuitertechniek wordt vaak gebruikt in animaties om te simuleren dat een objecten tussen twee punten heen beweegt. Denk aan pinball-machine waarin een knikker tussen allerlei objecten heen stuitert.

De stomptechniek

Objecten die de stomptechniek gebruiken bewegen heen en weer. Zoals iemand die met zijn voet op de grond tikt omdat hij ongeduldig is. Omdat deze beweging slechts twee verschillende extremen heeft, op en neer, is het makkelijk om uit te beelden in een animatie. Stompanimaties zijn vaak secundaire animaties aangezien ze slechts gaan om een klein deel van het lichaam dat beweegt.

Tenslotte

Tot slot nog een techniek voor de liefhebbers: gebruik maken van optisch bedrog met het moiré effect in animaties: <http://switzernet.com/people/emin-gabrielyan/070306-optical-speedup/>



FIGUUR 15: Moiré effect

Voor meer informatie over het maken van games: <http://www.gameskool.nl>