

12 Enkele gevorderde technieken

In het laatste hoofdstuk van deze Basis cursus maakt u kennis met diverse gevorderde Flash-technieken. Het zijn technieken die u op uitgebreide Flash-sites regelmatig aantreft en die --mits met beleid gebruikt -- het professionele gehalte van uw movies aardig kunnen opkrikken. Technologieën die aan bod komen zijn onder meer het werken met meerdere scenes, preloading, het maken en gebruiken van movie clips, geanimeerde buttons en combinaties van op zichzelf bekende technieken om indrukwekkende effecten te bereiken.

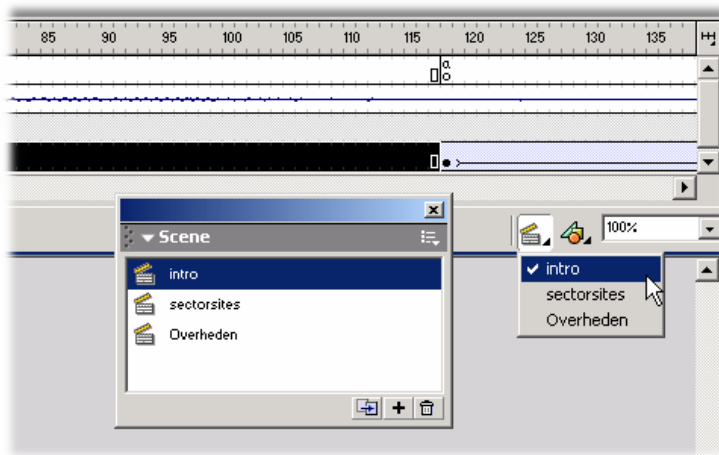
12.1 In dit hoofdstuk

12 Enkele gevorderde technieken.....	1
12.1 In dit hoofdstuk.....	1
12.2 Werken met meerdere scenes.....	1
Oefening 12.1 Verschillende scenes maken	2
12.2.1 Handmatig schakelen naar een scene	3
Oefening 12.2 Via een button schakelen tussen scenes	4
12.2.2 Werken met het palet Scene	5
12.3 Preloading toepassen	6
Oefening 12.3 Een preloader maken.....	7
12.4 Movie clips maken en gebruiken.....	10
Oefening 12.4 Een movieclip (MC) maken.....	11
12.5 Geanimeerde buttons maken.....	13
Oefening 12.5 MC's verwerken in een geanimeerde button	14
12.6 Buttons uitbreiden: bekende technieken combineren.....	15
Oefening 12.6 Voorbeeld: een tooltip maken	16
12.7 Overige tips en trucs	17
12.8 Eindoefening	17
12.9 Samenvatting	18

12.2 Werken met meerdere scenes

Indien een project groter wordt of uit meerdere logische secties gaat bestaan, kan het al snel handig zijn om de movie in meerdere scenes te verdelen. Denk bijvoorbeeld aan projecten als stapsgewijze instructie, diavoorstellingen en menugestuurde animaties. In dit laatste voorbeeld bevindt het 'hoofdmenu' zich bijvoorbeeld in de eerste scene, de buttons waarmee de keuzes gemaakt worden verwijzen naar verschillende andere scenes. In deze scenes bevinden zich op hun beurt weer buttons om terug te keren naar het hoofdmenu (de eerste scene). Wisselen tussen de verschillende scenes in een movie vindt in de praktijk meestal plaats met behulp van ActionScript opdrachten. Het werken met meerdere scenes heeft globaal gesproken de volgende kenmerken:

- Een nieuwe scene voegt u toe via **Insert, Scene**.
- Wanneer er geen Stop()-opdrachten of ander ActionScript-opdrachten in de movie voorkomen, speelt Flash de scenes automatisch één voor één af (de volgende scene begint automatisch als de voorgaande is afgelopen).
- Scenes beheert u via het palet **Scene**, dat wordt geopend via **Window, Scene** of **Shift+F2**. Het palet **Scene** maakt géén deel uit van de paletten die standaard in beeld worden getoond.



Figuur 11.1: De verschillende scenes in een project beheert u via het palet Scene en eventueel via de knop Edit Scene in de Timeline.

De kracht van het werken met scenes is voor een groot deel gelegen in het besturen van de scenes met allerlei ActionScript-opdrachten. Aan het 'eind' van de ene scene bevindt zich dan een frame met een **Stop**-opdracht en button op de stage waarop geklikt kan worden. De code die in het palet **Frame Actions** voor de betreffende button is geplaatst, luidt dan zoiets als **Go to and Stop (Scenenaam, "Labelnaam")**. Hiermee wordt de actie verplaatst naar de aangegeven scene en kan de movie vervolgd worden. Wat er dan precies gebeurt, is uiteraard afhankelijk van de opdrachten in de volgende scene.

Zo kunt u een groot project opdelen in verschillende kleinere, makkelijker te beheren deelprojecten.

In oefening 12.1 maakt u eerst een movie met verschillende scenes. In latere oefeningen voegt u buttons toe aan de scenes waarmee naar keuze gewisseld kan worden.

Oefening 12.1 Verschillende scenes maken

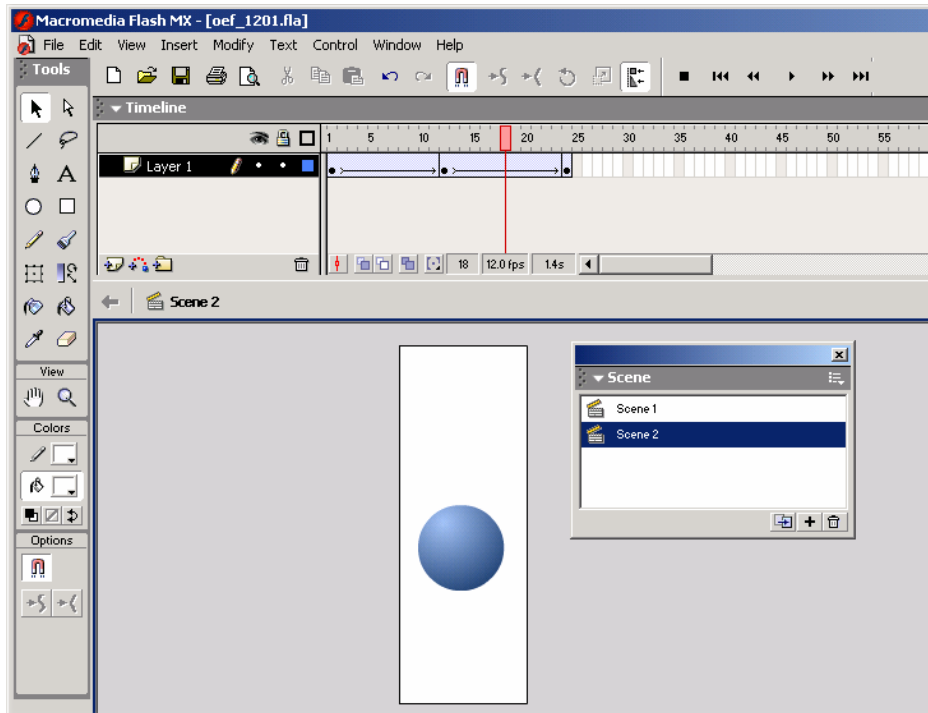
1. We maken in deze oefeningen gebruik van de movie **stuiter_klein.fl**a uit het boek, maar uiteraard kunt u ook een eigen project gebruiken. Open deze movie of download het bestand vanaf het Internet.
2. Kies **Insert, Scene**. Flash voegt een nieuwe scene aan de movie toe; de timeline van deze scene is nog leeg, alleen de aanduiding **Scene 2** (linksonder in de Timeline) is zichtbaar.
3. Met de knop **Edit Scene** in de timeline wordt geschakeld tussen de verschillende scenes. Klik op deze knop en kies **Scene 1** uit het menu dat verschijnt om terug te keren naar de eerste scene.

We kopiëren alle frames uit de eerste scene naar de tweede scene en geven vervolgens de bal in de tweede scene een andere kleur, zodat de wisseling tussen de scenes in de movie straks duidelijk zichtbaar is.

4. Selecteer alle frames in **Scene 1** en klik met de rechtermuisknop om het snelmenu op te roepen.
5. Kies **Copy Frames** en schakel vervolgens via de knop **Edit Scene** in de Timeline terug naar **Scene 2**.
6. Klik met de rechtermuisknop op het eerste frame en kies **Paste Frames** uit het snelmenu. De frames en de tweens worden in Scene 2 opgenomen. Eventuele namen van een layer worden niet gekopieerd, pas deze indien gewenst zelf aan.
7. In feite zijn er nu twee identieke scenes. We passen de kleur van de bal in Scene 2 aan, zodat hij duidelijk te onderscheiden is. Selecteer daartoe frame 1 in Scene 2.
8. Selecteer in de Property Inspector uit de lijst achter **Color** de optie **Tint**. Kies vervolgens een andere kleur en stel de waarde **Tint Amount** in. Wij kozen de RGB-kleur **66, 130, 230** met een waarde van **50%**: een blauwvariant.

9. Stel deze kleur ook in voor de frames 12 en 24, anders ondergaan de ballen tevens een kleuranimatie (ook leuk, maar nu niet de bedoeling).
10. Sla het bestand op als **stuiter_scenes fla** en kies **Control, Test Movie** (of druk op **Ctrl+Enter**) om de animatie te beoordelen binnen Flash. U kunt er ook voor kiezen om het effect met **Ctrl+F12** direct in de browser te beoordelen.

De standaardinstelling is **Loop**, de animatie wordt dus telkens herhaald, waarbij de bal van kleur verandert indien tussen de scenes wordt gewisseld. Het resultaat van deze oefening is beschikbaar in het bestand **oef_1201 fla**.



Figuur 12.2: De movie uitgebreid met een extra scene, Het palet Scene is tevens geopend

Door het kopiëren van de frames in de stappen 6 en 7 van oefening 12.1 is in feite de scene *geduplicateerd*. Het palet **Scene** biedt hiervoor ook een snelle manier, namelijk via de knop **Duplicate** (zie ook paragraaf 12.2.2). Flash maakt dan een integrale kopie van de gehele scene, met als standaardnaam **Scene 1 Copy**. Het kopiëren van losse layers en/of frames komt vooral van pas wanneer u bijvoorbeeld de achtergrond van een scene of navigatie- en tekstelementen wilt kopiëren, maar de nieuwe scene verder vanaf de grond af aan wilt opbouwen. Wanneer de te kopiëren layers/frames keyframes bevatten, bedenk dan dat elk object in het keyframe handmatig moet worden aangepast (zoals in oefening 12.1 de kleur van de bal).

12.2.1 Handmatig schakelen naar een scene

In veel gevallen zal een movie die uit meerdere scenes bestaat enige vorm van interactiviteit bieden. Vaak worden menu's aangeboden, kan de gebruiker keuzes maken, enzovoort. In werkelijkheid wordt in een Flash-project dan dus geschakeld naar een volgende scene (niet dat de gebruiker hier weet van heeft, voor de ontwerper wordt het onderhoud van en overzicht over de movie echter vergemakkelijkt).

In oefening 12.2 voegt u een button uit de meegeleverde library toe aan Scene 1, waarmee de gebruiker zelf naar de volgende scene kan klikken.

Oefening 12.2 Via een button schakelen tussen scenes

1. Open het project `stuiter_scenes fla` indien u dit na de vorige oefening gesloten had, of gebruik het bestand `oef_1201 fla` vanaf internet of de cd-rom. Controleer of **Scene 1** de actieve scene is.
2. Ten eerste voegen we een layer toe met de ActionScript-opdracht **Stop**, zodat de movie aan het einde van de animatie van Scene 1 onderbroken wordt.
 - Klik op de knop **Add Layer** in de timeline en geef deze layer de naam **Action**.
 - Voeg in frame 24 van deze layer een keyframe toe met **F6**.
3. Open het palet **Actions - Frame** door boven de Property Inspector op de titelbalk ervan te klikken.
4. Dubbelklik in de lijst **Actions, Movie Control** op de opdracht **Stop**. Wanneer u een ander frame op de timeline selecteert, zult u zien dat een kleine letter **a** is verschenen in het zojuist gemanipuleerde frame.

Het wordt aanbevolen om ActionScript-opdrachten in frames op te nemen in een eigen layer (bijvoorbeeld met de naam **Action** of **Script**), zodat de actions niet onbedoeld worden meegekopieerd naar andere layers of scenes indien u frames kopieert. Bovendien wordt het onderhoud en overzicht vergemakkelijkt indien alle 'losse' script-opdrachten in een eigen layer gegroepeerd zijn. Script-opdrachten die bij een button horen, worden uiteraard in de betreffende layer opgenomen.

5. Voeg nog een extra layer toe en noem deze layer **Button**. Voeg in deze layer een keyframe toe aan frame 24. Zorg dat dit frame geselecteerd is.
6. Open via **Window, Common Libraries, Buttons** de bibliotheek met standaardbuttons en sleep een button naar de stage. Positioneer hem bovenaan, iets links van het midden van de pagina.
 - Wij kozen de button **Circle button - Next** uit de set **Circle Buttons**, maar u kunt ook een andere button kiezen.
 - Geef de button indien nodig via het palet **Info** wat kleinere afmetingen, bijvoorbeeld **15x15** pixels.
7. Klik op de button om er zeker van te zijn dat deze is geselecteerd en controleer of het palet **Actions** nu de titel **Actions – Button** heeft.
8. Dubbelklik op de optie **goto** en zorg er voor dat de volgende instellingen van kracht zijn:
 - **Scene: Scene 2**
 - **Type: Frame Number**
 - **Frame: 1**
 - **Go to and Play: geselecteerd**
9. Minimaliseer daarna het venster door op de titelbalk te klikken.

De volledige regel ActionScript-code voor de button luidt nu `GotoAndPlay ("Scene 2", 1);`. In werkelijkheid zult u de scenes wat aansprekender namen geven, en liever een label gebruiken in plaats van een framenummer; dit is in de voorgaande hoofdstukken ook aan de orde geweest. Voor nu laten we het hier echter even bij.

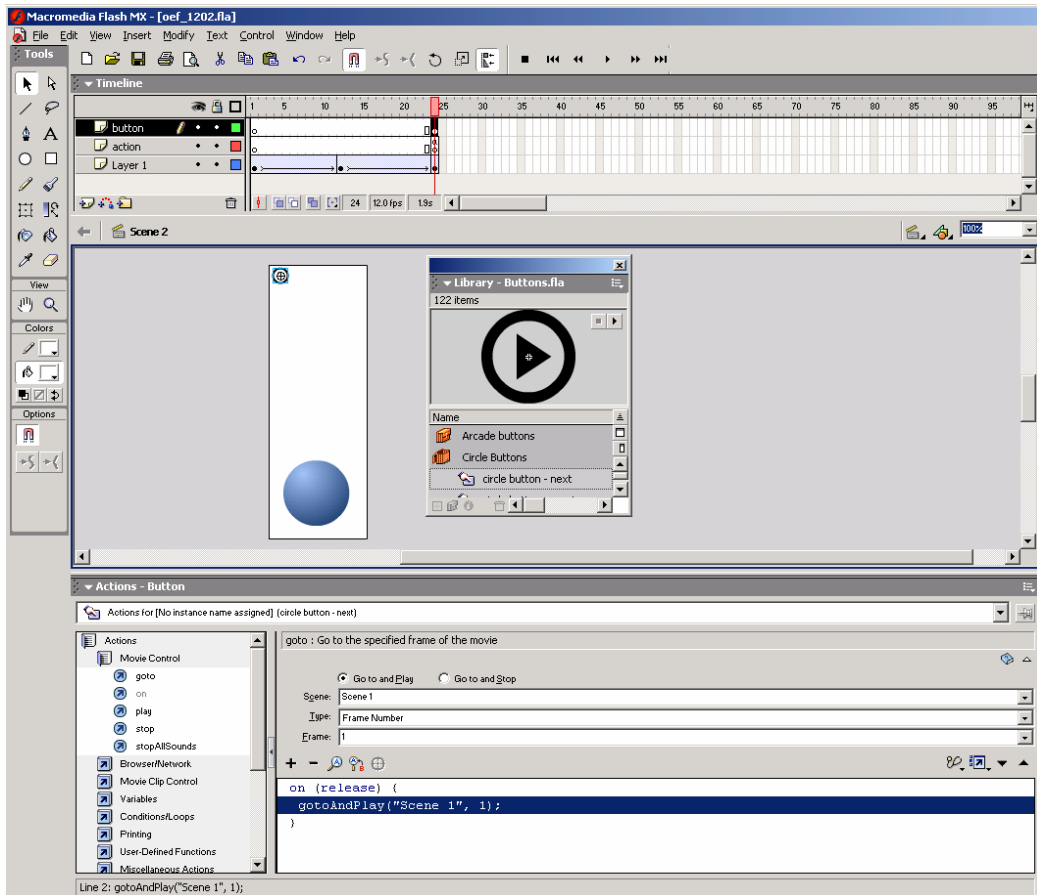
10. Sla het project op en kies **Control, Test Movie** om het resultaat binnen Flash te testen.

De animatie begint te lopen, aan het einde van Scene 1 verschijnt de button in beeld en kunt u erop klikken om door te gaan met Scene 2. Deze scene wordt geheel doorlopen, waarna Scene 1 weer begint (de standaardinstelling is immers **Loop**).

Bedenk dat de button alleen verschijnt in frame 24 van Scene 1. Als u op elk moment wilt kunnen schakelen naar de volgende scene, neem de button dan op in alle keyframes van Scene 1.

11. Voeg nu zelf op vergelijkbare wijze layers, keyframes en een button toe aan Scene 2, zodat aan het einde van deze scene alleen teruggeschakeld wordt indien de gebruiker op deze knop klikt.

Het eindresultaat van deze oefening staat op internet als `oef_1202 fla`; figuur 12.3 toont hoe de movie en de timeline eruit ziet voor Scene 2. In de eindoefening voor dit hoofdstuk vindt u enkele suggesties om de technieken voor deze oefening verder uit te breiden.

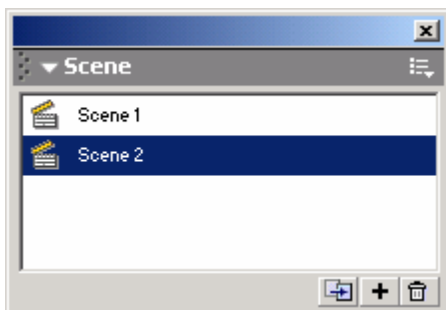


Figuur 12.3: De stage en de timeline voor Scene 2; let op de instellingen in het venster Actions – Button en de keuze uit de library

12.2.2 Werken met het palet Scene




Het palet **Scene** is evenals de andere paletten (zoals **Info** en **Transform**) een handig hulpmiddel bij het bewerken en onderhouden van de diverse scenes in een project. Met het palet **Scene** kunt u de namen van scenes aanpassen, scenes makkelijk dupliceren, de volgorde van scenes wijzigen en uiteraard nieuwe scenes maken of oude scenes verwijderen. Figuur 12.4 toont het palet **Scene** voor het project **stuijer_scenes fla** uit de voorgaande oefeningen. De wijze om het palet **Scene** te openen zal bekend zijn:

- kies **Window**, **Scene**.



Figuur 12.4: Het palet Scene voor het project uit de vorige oefeningen

Het bewerken van scenes via dit paneel zal ook niet al te veel problemen opleveren; 12.1 bespreekt kort de werking van de verschillende knoppen in het palet **Scene**.

Knop	Naam	Functie
	Add Scene	Voegt een lege scene toe aan de movie. De nieuwe scene wordt standaard ingevoegd na de op dit moment geselecteerde scene, waarmee tevens de volgorde in de movie is vastgelegd. De standaardnaam is Scene x , waarbij de x staat voor het volgnummer van de scene. De scenes worden afgespeeld in de volgorde waarin ze van boven naar beneden voorkomen in het palet Scene .
	Delete Scene	Verwijdert een scene. Deze handeling is niet ongedaan te maken. Eventuele objecten en symbolen die zich op de scene bevonden, blijven wel aanwezig in de library, maar eventueel geprogrammeerde ActionScript-code gaat verloren.
	Duplicate Scene	Kopieert een volledige scene, inclusief symbolen, tweens en ActionScript-code. De standaardnaam voor het duplicaat is Scenenaam Copy , waarbij Scenenaam uiteraard staat voor de naam van de scene.

Tabel 12.1: De mogelijkheden van de Scene Inspector

De naam van een scene wijzigen

In het palet Scene is geen knop aanwezig om rechtstreeks de naam van een scene aan te passen. Gelukkig is dat ook niet moeilijk:

- dubbelklik op de naam van de gewenste scene en typ in het kader dat verschijnt de nieuwe naam over de oude naam heen.

Wilt u de volgorde van de scenes in het project veranderen, of een nieuw toegevoegde scene verplaatsen? Sleep dan in de Scene Inspector de naam van de scene met ingedrukte linkermuisknop naar boven of naar beneden om hem anders te rangschikken ten opzichte van de overige scenes. Dit betekent ook dat de betreffende scene eerder of later wordt afgespeeld, indien u dit tenminste niet bestuurd met ActionScript.

Tot zover een inleiding tot het werken met scenes. Bedenk dat door het werken met scenes uw movie wordt opgesplitst in meerdere, logische secties. U hoeft dan niet te werken met timelines van vele honderden frames en tientallen labels. 'Verdeel en heers' is hier het credo. Het aantal scenes in een Flash-movie is onbeperkt, dus hier hoeft u geen rekening mee te houden. In de volgende paragraaf kijken we naar een veelgebruikte toepassing van scenes: het maken van een preloader die alvast getoond wordt terwijl de eigenlijke movie op de achtergrond wordt geladen.

12.3 Preloading toepassen

Preloading is een techniek die wordt gebruikt om gebruikers alvast een pagina of bewegend beeld voor te schotelen terwijl de eigenlijke (omvangrijke) movie op de achtergrond wordt binnengehaald. Het is een vriendelijke manier om mede te delen dat er wel degelijk wat gebeurt en om de gebruiker niet op te zadelen met een leeg scherm waardoor hij wellicht denkt dat er iets niet werkt -of erger nog- de site verlaat. Bij preloading wordt gebruik gemaakt van de ActionScript-opdracht **If Frame Is Loaded**. Deze action controleert of een bepaald deel van de movie geladen is, en gaat daarna pas verder met de uitvoering.

Preloading is niet alleen maar prettig voor de gebruiker die dan alvast op de hoogte wordt gesteld van het feit dat de movie wordt geladen, het kan ook noodzaak zijn voor de programmeur. Herinner u dat Flash movies *streaming* worden afgespeeld: onderdelen worden dus op het scherm gezet zodra ze binnen zijn gehaald. Dit geldt ook voor buttons en ActionScript-opdrachten. De script-opdrachten werken echter niet goed als ze verwijzen naar een deel van de movie die nog niet binnen is gehaald. De resultaten zijn dan onvoorspelbaar. Denk bijvoorbeeld terug aan de vorige paragraaf; als in een

complex project met veel bitmaps en een grote soundtrack (achtergrondgeluid) op de knop **Scene 2** wordt geklikt terwijl deze scene nog niet is binnengehaald, zal het resultaat niet naar wens zijn. Met behulp van preloading kunt u dit voorkomen.

In oefening 12.3 programmeert u een eenvoudige preloader, die getoond wordt zolang de rest van de movie nog niet is binnengehaald. Tussen de stappen van de oefening door vindt u verklarende achtergrondinformatie.

Oefening 12.3 Een preloader maken

1. Open het project **stuiter_scenes.fla** indien u dit na de vorige oefening gesloten had, of gebruik **oef_1202.fla** vanaf het internet. Dit is de uitgangssituatie.
 - Controleer of Scene 2 de actieve scene is.
 - Kies direct **File, Save As** en sla het bestand op als **stuiter_preload.fla**.

We voegen eerst een eind-label toe aan het laatste frame van Scene 2, waarop we zodadelijk kunnen testen. Zodra dit frame immers geladen is, is de hele movie binnengehaald en kan de preloader zijn werk staken.

2. Selecteer in Scene 2 frame 24 in de layer **Action**.
3. Typ in de Property Inspector in het vak **<Frame Label>** als labelnaam **einde** en druk op **Enter** om deze naam vast te leggen. Omdat frame 24 het laatste frame is, is het label niet goed zichtbaar in de timeline. Het is echter wel degelijk aanwezig.

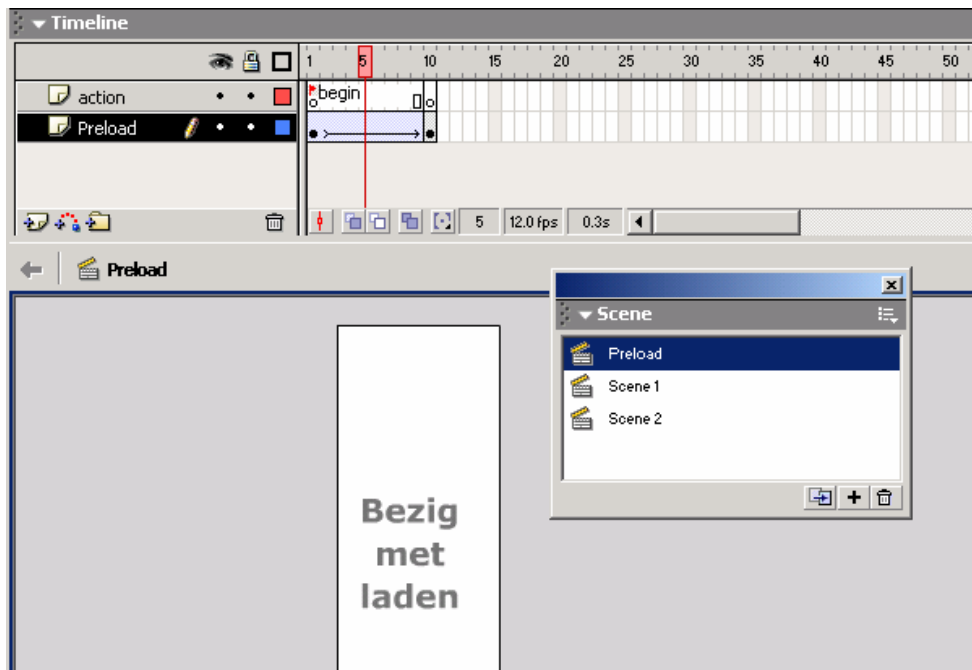
We voegen nu de eigenlijke preloader toe; uiteraard komt deze in een eigen scene te staan. Als preloader gebruiken we de tekst **Bezig met laden...**, die langzaam knippert.



4. Open het palet **Scene** via **Window, Scene** en klik op de knop **Add Scene**. Flash voegt een lege scene in, met de standaardnaam **Scene x**.
5. Dubbelklik op de naam van de scene en hernoem deze naar **Preload**.
6. Versleep de scene omhoog in het palet **Scene**, zodat hij boven Scene 1 en Scene 2 staat.
7. De scene **Preload** is direct geselecteerd op de stage:
 - voeg hier als eerste een nieuwe layer aan toe met de knop **Add Layer** in de timeline;
 - noem deze layer **Action**.
 - Hernoem de ander layer (**Layer 1**) naar **Preload**.



8. Selecteer Frame 1 in de layer **Action** en geef deze via de Property Inspector het label **Begin**.
9. Selecteer nu frame 1 in de layer **Preload** en typ hierin met de Text-tool de tekst **Bezig met laden...**. Kies zelf een geschikt lettertype, wij gebruikten **Verdana, 20** punten, **vet, zwart**. Met de knop **Align** kunt u deze tekst horizontaal en verticaal centreren op de stage.
10. Selecteer de tekst met de Arrow-tool en kies **Insert, Convert to Symbol** (of druk op **F8**). Geef het symbool de naam **grPreloader** en kies als Behavior het type **Graphic**.
11. Selecteer frame 10 in beide layers en voeg hier met **F6** een keyframe aan toe (nu wordt ook het label voor de layer **Script** zichtbaar).
12. Selecteer op het tekstsymbool op de stage en open het in de Property Inspector de lijst **Color**. Kies als effect **Alpha** en stel de waarde in op **0**.
13. Klik met de rechtermuisknop op een willekeurig frame tussen de twee keyframes in de layer **Preload** en kies **Create Motion Tween** uit het snelmenu. Het scherm kan er nu zo uitzien als in figuur 12.5.



Figuur 12.5: De scene Preload met een eigen animatie en verschillende keyframes

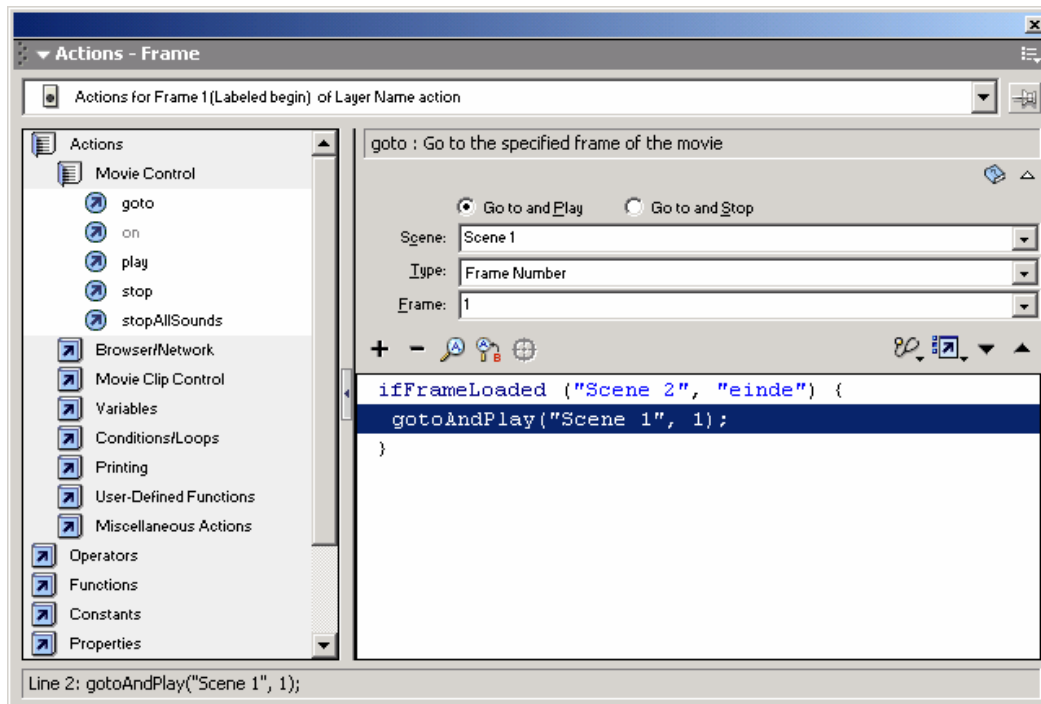
Preloader-code toevoegen

De scene en de animatie die als preloader wordt gebruikt zijn nu klaar. We kunnen in de scene **Preload** de juiste ActionScript-code toevoegen die controleert of de rest van de movie al geladen is, en zo ja, de preloading-scene afbreekt.

[hier verder]

14. Selecteer frame 1 in de layer **Action** (het frame met het label **begin**) en open het palet **Actions – Frame**.
15. Open de lijst **Deprecated** en dubbelklik op de optie **ifFrameLoaded** en maak verder de volgende keuzes:
 - **Scene: Scene 2**
 - **Type: Frame Label**
 - **Label: einde**
16. Kies uit de lijst **Actions, Movie Control, goto**. Stel hiervoor de volgende aanvullende keuzes in:
 - **Scene: Scene 1**
 - **Type: Frame Number**
 - **Frame: 1**
 - **Go to and Play: geselecteerd**

Het venster en de code zien er nu zo uit als in figuur 12.6. Controleer dit goed en sluit daarna het venster.



Figuur 12.6: ActionScript-code voor de preloader

De actie `ifFrameLoaded` staat in de lijst **Deprecated** (dit betekent: 'verouderd, liever niet meer gebruiken'), omdat in Flash MX de eigenschap `_framesloaded` beschikbaar is die meer mogelijkheden heeft. Hiervoor moet u echter meer code in ActionScript programmeren, dit laten we op dit moment even achterwege. De functie `ifFrameLoaded` wordt in Flash MX nog gewoon ondersteund.

We moeten nu nog loop-functionaliteit aan de preloader toevoegen, zodat hij wordt afgespeeld zolang het frame einde nog niet is geladen.

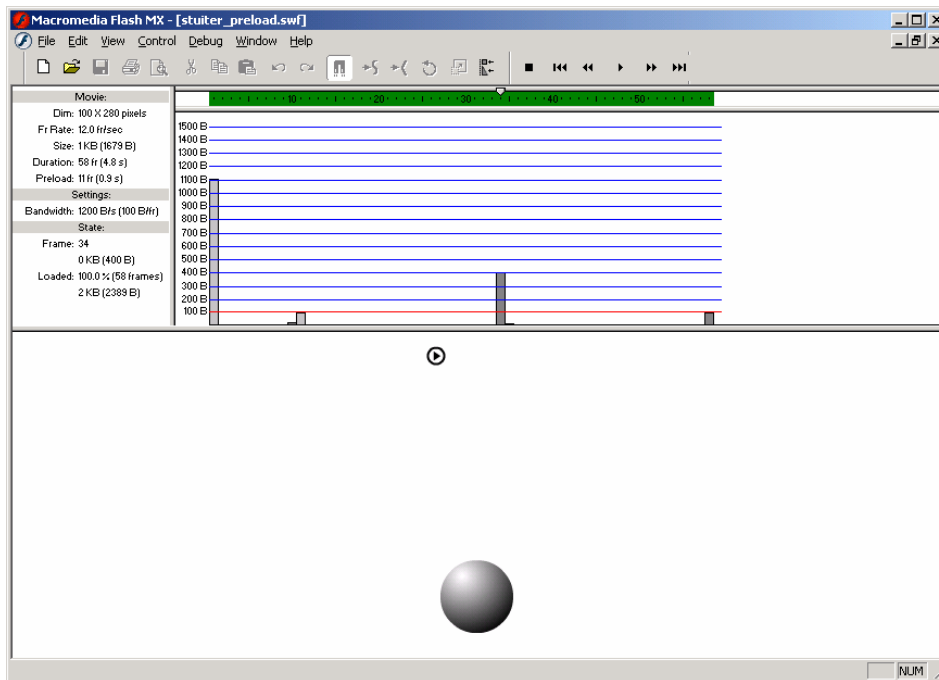
17. Selecteer frame 10 in de layer **Action**.
18. Dubbelklik op **goto** en stel de volgende aanvullende eigenschappen in:
 - **Scene: Preload** (of: `<current scene>`)
 - **Type: Frame Label**
 - **Label: begin**
 - **Go to and Play: geselecteerd**

Sluit daarna het venster en sla het bestand nogmaals op. Dit is alle code die noodzakelijk is.

De preloader testen

Om te testen of de preloader goed werkt, kunt u het beste schakelen naar de mode **Test Movie**, waarbij de optie **Show Streaming** (met een lage bandbreedte) geselecteerd wordt.

19. Kies **Control, Test Movie**. De animatie begint direct te lopen, waarschijnlijk is de preloader niet eens zichtbaar om dat het frame einde zich al in het geheugen bevond. Test eventueel met de knoppen of **Scene 1** en **Scene 2** goed werken.
20. Kies **Debug, 14.4 (1.2 KB/s)** om een lage bandbreedte te simuleren.
21. Kies **View, Show Streaming** om de movie te laden, de preloader is nu wel zichtbaar (hij knippert één à twee keer voordat de hele movie geladen is, gebruik een ingewikkelder movie om het effect van de preloader nog beter te zien).
22. Wilt u het proces wat meer in detail volgen, schakel dan de Bandwidth Profiler in via **View, Bandwidth Profiler**.



Figuur 12.7: De movie en de preloader in de mode Test Movie; de Bandwidth Profiler is zichtbaar; de optie Frame By Frame Graph is actief

Dat is alles. U hebt nu een eenvoudige preloader die wordt afgespeeld zolang het laatste frame in de animatie nog niet is geladen. U zult inmiddels wel hebben begrepen dat preloaders veel worden toegepast bij omvangrijke movies met veel beeld- en geluidsmateriaal (voor een voorbeeld, zie bijvoorbeeld www.nike.com/bowerman of www.metontour.com, kies de optie **Enter with Intro**).

Er zijn nog enkele aandachtspunten waar u bij het samenstellen van preloaders op kunt letten:

- Denk er aan in het laatste frame van de animatie eventueel ook een **Stop**-commando op te nemen, anders begint de movie na het afspelen opnieuw met de preloader (omdat **Loop** de standaardinstelling is). Dat is natuurlijk niet de bedoeling. In bovenstaande oefening hoefde dat niet apart, omdat dit al gebeurt was in eerdere oefeningen (de laatste frames van Scene 1 en Scene 2 waren al voorzien van de opdracht **Stop** in vorige oefeningen).
- Een preloader kan zelf natuurlijk ook een achtergrondmuziekje, bitmaps of ingewikkelde tweens bevatten, maar denk erom dat het doel van de preloader niet voorbij geschoten wordt. Als het secondenlang duurt voordat de preloader geladen is, zullen veel gebruikers afhaken.
- Plaats in geval van twijfel de hele movie op een Internet-server en zie hoe het complete project zich 'gedraagt' als de movie met preloader daadwerkelijk over het Internet verzonden wordt.

Met geavanceerde ActionScript-opdrachten die beschikbaar zijn in Flash 4 zijn ook meer geavanceerde preloaders te maken (zoals balkjes die 'vollopen' naarmate een groter deel van de movie geladen is). Een voorbeeld hiervan vindt u op The Flash Guide, op <http://www.turtlesshell.com/guide/general/preload/index.html>

12.4 Movie clips maken en gebruiken

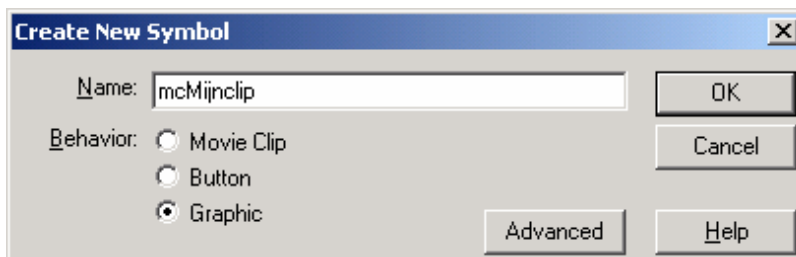
We zijn de term movie clip al enkele keren tegengekomen. Een movie clip (door ervaren Flashers vaak afgekort tot *MC*) is als het ware een mini-movie binnen de hoofd-movie. In een movie clip kunnen animaties (tweens en frame-by-frame animaties), actions, sounds, andere symbolen uit de library en zelfs andere movie clips worden opgenomen. Movie clips hebben een eigen timeline, die onafhankelijk van de hoofd-timeline verloopt. Dit kan bijzonder handig zijn voor animaties die doorlopen nadat de hoofdanimatie is gestopt, of voor animaties die binnen andere animaties moeten

worden afgespeeld. Movie clips hebben in de hoofd-timeline maar één keyframe nodig dat het afspelen van de movie clip start.

Movie clips worden bijvoorbeeld veel gebruikt om geanimeerde buttons te maken. Hiervan geven we in de volgende paragraaf een voorbeeld. Het samenstellen van een movie clip is niet erg moeilijk. Het makkelijkst is het om eerst de losse animatie te maken (eventueel in een aparte 'oefen scene') en vanuit deze animatie de movie clip te componeren. Ga hierbij als volgt te werk.

Een animatie achteraf omzetten naar een movieclip

- Selecteer elk frame en elke layer van de bestaande animatie die u deel wilt laten uitmaken van de movie clip (clips kunnen dus uit meerdere layers bestaan).
- Klik met de rechtermuisknop op de geselecteerde frames en kies **Copy Frames** uit het snelmenu (of kies **Edit, Copy Frames**).
- Kies **Insert, New Symbol** of druk op **Ctrl+F8**. Het dialoogvenster **Create New Symbol** uit figuur 12.8 verschijnt.
- Geef de clip een naam en kies als **Behavior** het type **Movie Clip**. We adviseren om zelfgemaakte movie clips te voorzien van het standaardvoorzetsel **mc**.
- Selecteer het eerste frame in de timeline van het zojuist gemaakte symbool en plak de frames hier in door te kiezen voor **Edit, Paste Frames** (of klik met de rechtermuisknop op het eerste frame en kies **Paste Frames** uit het snelmenu).
- Klik op de aanduiding **Scene 1** of kies **Edit, Edit Movie (Ctrl+E)** om terug te keren naar de hoofd-movie.
- Sla het project op. Indien gewenst kunnen de frames die de oorspronkelijke animatie bevatten nu verwijderd worden; ze zijn immers als movie clip opgeslagen in de library voor de betreffende movie. Het is dan makkelijk als u een aparte scene hebt gebruikt om de movie clip te construeren, deze kan eenvoudig via het palet **Scene** worden verwijderd.



Figuur 12.8: Een movieclip maakt u via het dialoogvenster **Create New Symbol**

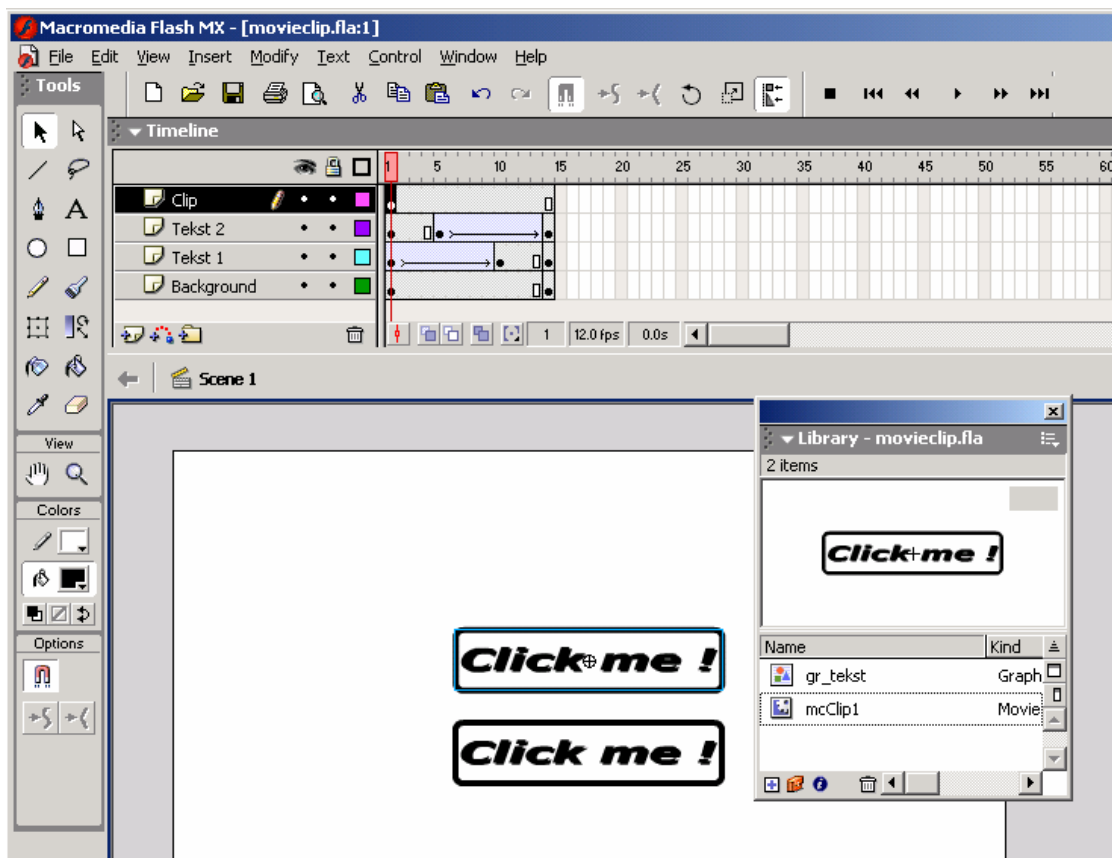
In oefening 12.4 maken we van een bestaande animatie een movie clip. U kunt zelf even snel een animatie maken (een eenvoudige tweeën op een enkele layer voldoet al) of het bestand **movieclip fla** downloaden vanaf het Internet. Dit is de uitgangssituatie voor deze oefening.

Oefening 12.4 Een movieclip (MC) maken

1. Open **movieclip fla** en bestudeer deze animatie. Het is een eenvoudige tekstanimatie, die een tekst in een rechthoekig kader doet vergroten en verdwijnen (fade-out). Hiervan maken we een movie clip en in een later stadium een geanimeerde button. Het enige symbool in de library is de tekst **Click me!**
2. Selecteer alle frames in alle layers (dit gaat het snelste door **Shift**+klikken op de layernamen) en roep met de rechtermuisknop het snelmenu op. Kies hierin de optie **Copy Frames**.
3. Kies nu **Insert, New Symbol**. Het dialoogvenster **Create New Symbol** uit figuur 12.8 wordt geopend. Geef het symbool de naam **mcClip1** en kies achter **Behavior** de optie **Movie Clip**. Sluit daarna het venster met **OK**.
4. Klik met de rechtermuisknop op frame 1 in Layer 1 (in de symboleneditor) en kies **Paste Frames** uit het snelmenu. Alle frames van de animatie worden ingevoegd.
5. Keer terug naar de hoofdmovie door op de aanduiding **Scene 1** te klikken. We voegen nu de movie clip toe aan de stage
6. Klik op de knop **Add Layer** in de timeline en noem deze layer **Clip**. Selecteer het eerste frame op deze layer.



7. Open indien nodig de library en sleep het symbool **mclip 1** naar de stage, tot boven de reeds aanwezige 'losse' animatie.
8. Kies **Control, Test Movie** om de animatie binnen Flash te testen. Omdat er nu twee dezelfde animaties aanwezig zijn op de stage, zullen ze beide tegelijk starten en synchroon verlopen. De een is echter een gewone animatie, de ander is een movie clip.
9. Druk op **Enter** om de animatie te stoppen. De 'losse' animatie wordt inderdaad onderbroken, de movie clip animeert echter gewoon door! Dit is het kenmerk van movie clips, hun timeline verloopt *onafhankelijk* van de hoofd-timeline en loopt ook door wanneer de hoofd-movie is gestopt.
10. Door nogmaals op **Enter** te drukken start u ook de onderste animatie weer, zij het dat deze nu waarschijnlijk niet meer synchroon loopt met de movie clip.
11. Sluit het venster met het swf-bestand en keer terug naar de Flash-ontwikkelomgeving. De stage ziet er ongeveer zo uit als in figuur 12.9. Het resultaat van deze oefening staat ook online, als [oef_1204 fla](#).



Figuur 12.9: Twee animaties op de stage, de bovenste is een movie clip, de onderste is een losse animatie, opgebouwd uit verschillende layers en tweens

Zoals gezegd, movie clips kunnen behalve tweens en symbolen ook geluiden, actions en zelfs andere movie clips bevatten. Zo is letterlijk elk niveau van complexiteit te bereiken.

Een goed voorbeeld van het toepassen van een movie clip zou het volgende zijn: stel dat u een geanimeerd poppetje over de stage wilt laten rennen. In plaats van een complexe frame-by-frame animatie samen te stellen kunt u de verschillende acties verdelen over movie clips. Het bewegen van de ledematen (armen en benen) kan dan door een movie clip worden verzorgd, terwijl het van links naar rechts rennen door een twee in de hoofd Timeline wordt verzorgd. Zo zijn er vele andere toepassingen te bedenken.

In de volgende paragraaf kijken we naar de wijze waarvoor movie clips veel worden ingezet: geanimeerde buttons.

Veel wat meer gevorderde Flashers plaatsen voor maximale flexibiliteit vrijwel al hun animaties in een movieclip. Deze kan dan overal in het project gebruikt worden, zonder gebonden te zijn aan een positie in de timeline. Zo kan het gebeuren dat de 'hoofdmovie' maar uit 1 frame bestaat op 1 layer! Omdat de hele scene is opgebouwd uit losse MC's welke elk een eigen timeline hebben is de functionaliteit van het Flash-programma over veel losse clips verspreid. De MC's komen dan bijvoorbeeld in actie bij een MouseOver-event, of een Click-event. MC's worden enorm veel gebruikt in combinatie met ActionScript. Ook de (slide) menu-systemen die veel met Flash worden gebouwd bestaan vrijwel uitsluitend uit MC's. Bekijk goed de voorbeelden in de honderden voorbeeld-fla's op bijvoorbeeld <http://www.flashkit.com/movies/>. U zult zien hoe belangrijk het Movie Clip-concept is in veel projecten!

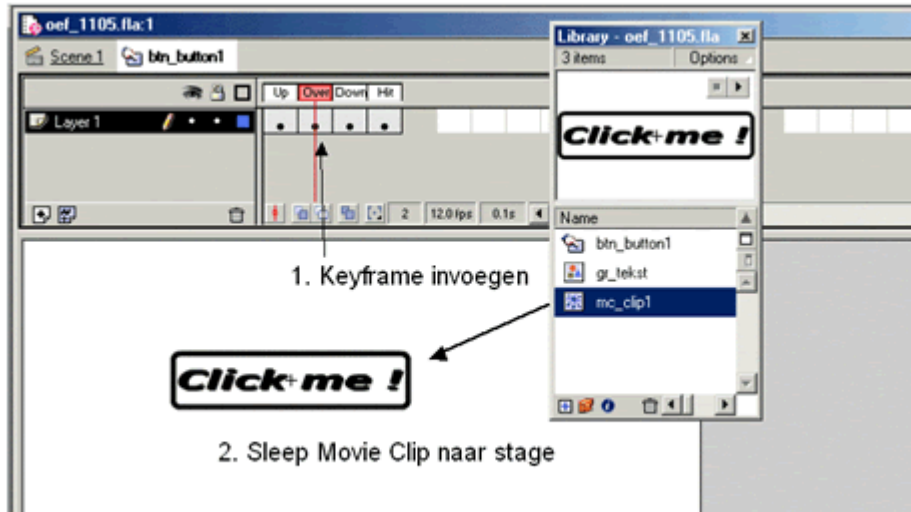
12.5 Geanimeerde buttons maken

Geanimeerde buttons kennen dezelfde functionaliteit als gewone buttons, met dat verschil dat er in een van de fasen (of in meerderde fasen) 'iets' gebeurt. Wanneer er bijvoorbeeld een movie clip wordt geplaatst in de Over-state van een button, wordt de movie clip afgespeeld zodra de muis boven de button zweeft. De movie clip stopt weer zodra de muis de button verlaat. Dit is echter maar één voorbeeld, indien gewenst kan elke state zijn eigen movie clip bevatten (en movie clips kunnen zoals u inmiddels weet weer uit vele layers, animaties en geluiden bestaan). Geanimeerde buttons komen we veel tegen op Internet-sites die zijn uitgebreid met Flash, vaak met subtiele, oogstrelende animaties.

De algemene manier om een geanimeerde button te maken is als volgt:

- Maak op de bekende wijze een nieuwe button in de library.
- Maak op de hierboven beschreven wijze een movie clip, zorg dat deze het effect bevat dat u aan de button wilt toevoegen.
- Open de button in de symbolen-editor, bijvoorbeeld door er in de library met de rechtermuisknop op te klikken en te kiezen voor **Edit**.
- Selecteer in de timeline van de button het frame waaraan u de animatie wilt toevoegen. Dit kan een van de frames **Up**, **Over** of **Down** zijn (omdat het frame **Hit** nooit zichtbaar is voor gebruikers, is het zinloos om daar een animatie aan toe te voegen).
- Als dit frame nog geen keyframe bevat, voeg dit dan toe met **F6** of via **Insert, Keyframe**.
- Keer terug naar de library (kies **Window, Library** of druk op **Ctrl+L** indien deze niet meer zichtbaar was) en sleep de movie clip naar de stage, op de plaats waar de animatie moet optreden.
- Keer terug naar de movie door op de aanduiding **Scene 1** te klikken, plaats eventueel een instantie van de button op de stage en test de geanimeerde button via **Control, Test Movie**.

Let op: geanimeerde buttons werken uitsluitend in de gepubliceerde swf-bestanden, niet wanneer de optie **Control, Enable Simple Buttons** is ingeschakeld. Test geanimeerde buttons daarom altijd via de optie **Test Movie (Ctrl+Enter)** of in de browser.

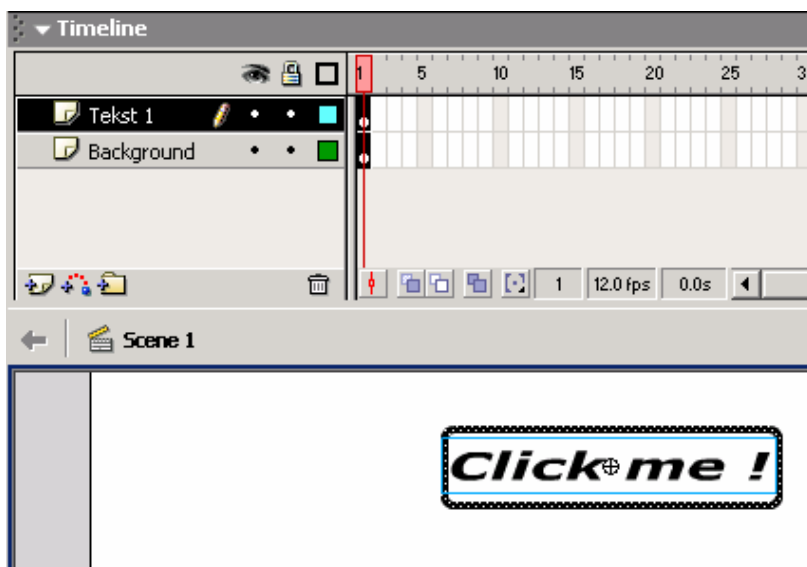


Figuur 12.10: Een movie clip toegevoegd aan een keyframe van een button zorgt voor een animatie-effect van de button

In oefening 12.5 borduurt u verder op de eerdere oefeningen en maakt u een geanimeerde button waarbij de movie clip alleen wordt afgespeeld in de Over-state van een button.

Oefening 12.5 MC's verwerken in een geanimeerde button

1. Ga verder met het bestand dat u na de vorige oefening nog geopend hebt, of open **oef_1104 fla** vanaf de cd-rom.
2. Verwijder alle elementen van de stage en alle lege frames, behalve het eerste keyframe in de layers **Background** en **Tekst 1**. Sla daarna het project op als **button_animatie fla**.
3. Selecteer de overgebleven objecten (hiervan maken we zodadelijk de Up-state van de button) en plaats ze met **Ctrl+X** of **Edit, Cut** op het Klembord (zie figuur 12.11).



Figuur 12.11: Van de overgebleven elementen maken we één button-symbool

4. Kies **Insert, New Symbol** en geef het symbool de naam **btnButton1**; als type kiest u uiteraard **Button**. Flash opent de symbolen-editor voor **btnButton1**, waarbij het Up-frame geselecteerd is.
5. Druk op **Ctrl+V** of kies **Edit, Paste** om de elementen vanaf het Klembord in te voegen. Ze worden automatisch op de pagina gecentreerd.

6. Voeg met **F6** een keyframe toe aan het frame **Over** in de Timeline en druk direct daarna op **Delete** om de elementen van de stage te verwijderen. In de fase **Over** willen we immers de movie clip gebruiken.
7. Open zonodig de library (**Ctrl+L**) en sleep het symbool **mcClip1** naar de stage. Met de knop **Align** kunt u het object exact centreren.
8. Selecteer het frame **Down** in de timeline en voeg ook hier met **F6** een keyframe aan toe. Dit frame laten we verder ongewijzigd (per saldo loopt de movie clip dus gewoon door in de fase **Down**).
9. Voeg ten slotte een keyframe toe aan de fase **Hit** en teken met de Rectangle-tool een rechthoek op de stage, die het hele oppervlak van de button bedekt. Zou u dit achterwege laten, dan treedt het **Over**- en **Down**-effect alleen op als de muis exact boven de tekst of boven de omranding zweeft, wat nu niet de bedoeling is.
10. Keer terug naar de hoofd-movie door op de aanduiding **Scene 1** te klikken. Er is nog geen button op de stage aanwezig, sleep er dus een vanuit de library naar het scherm.
11. Kies **Control**, **Test Movie** of druk op **Ctrl+Enter** om de movie als swf-bestand te beoordelen. Telkens wanneer u in de testmode de muis over de button beweegt, zal de button geanimeerd worden (met andere woorden: wordt de movie clip afgespeeld).
12. Sla het project nogmaals op (er zitten nu wel wat overbodige lege layers en frames in, maar dat kan geen kwaad). Online vindt u het resultaat van deze oefening eventueel terug als [oef_1205 fla](#).



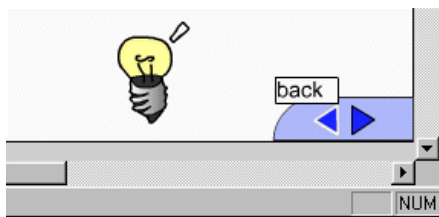
Figuur 12.12: De animatie wordt alleen afgespeeld in de fasen **Over** en **Down**

Op deze wijze zijn de meest fantastische button-effecten te realiseren, want niets let u natuurlijk om in de timeline van de button nog extra layers toe te voegen met eventuele geluidseffecten, extra movie clips, enzovoort. Zo wordt een button als het ware een project op zich. Op de sites die in appendix A van het boek genoemd worden vindt u hiervan vele voorbeelden en aanvullende technieken.

12.6 Buttons uitbreiden: bekende technieken combineren

U kent waarschijnlijk de zogeheten *tooltips*, kleine gele vakjes met extra uitleg die verschijnen zodra u met de muis boven een knop zweeft. Flash hanteert deze techniek in het programma zelf, maar ook toepassingen uit bijvoorbeeld de Microsoft Office-reeks (Word, PowerPoint, Excel) en andere Windows-programma's kennen tooltips.

Misschien hebt u enkele lesbestanden bekeken die met Flash worden meegeleverd, ook hierin verschijnen een soort tool tips wanneer de muis over een button beweegt (zie ook figuur 12.13). Deze tips zijn zelf 'geprogrammeerd' voor de betreffende buttons. De Flash-lesbestanden bereikt u eventueel via [Help](#), [Lessons](#).

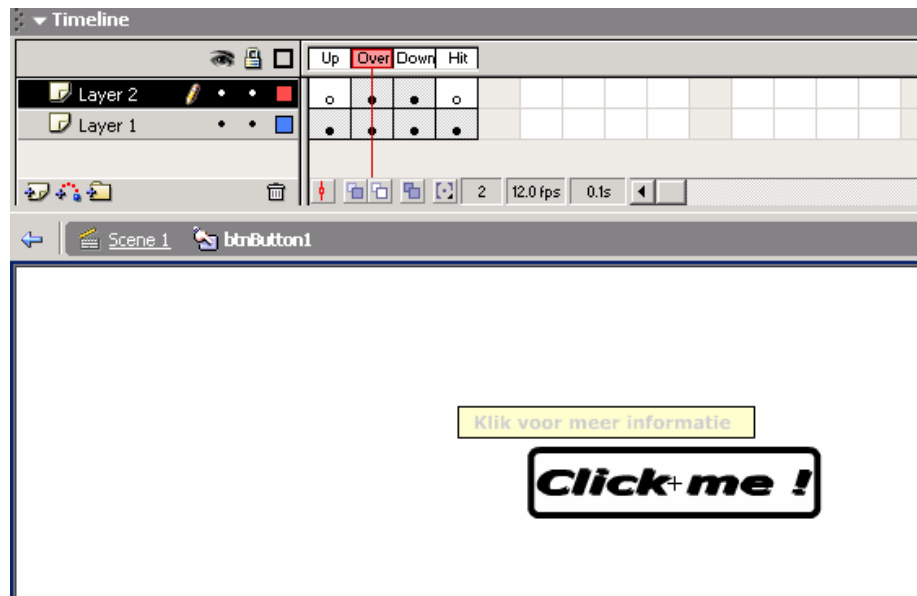


Figuur 12.13: Tool tips in Flash-movies moeten zelf binnen de button aangebracht worden

Het maken van dergelijke tool tips voor buttons is niet moeilijk en vereist geen speciale vaardigheden. Het is opnieuw een kwestie van het *combineren* van bestaande technieken. Er is namelijk gewoon een extra layer met verklarende tekst toegevoegd aan het Over-frame voor de button. In oefening 12.6 voegt u zelf een dergelijke tooltip toe aan de button die u in de vorige oefening hebt gemaakt.

Oefening 12.6 Voorbeeld: een tooltip maken

1. Ga verder met het bestand dat u na de vorige oefening nog geopend hebt, of open **oef_1205 fla** vanaf de cd-rom.
2. Open indien nodig de library en dubbelklik op het symbool **btnButton 1** om dit in de symboleneditor te bewerken.
3. Voeg een extra layer toe en voeg vervolgens een keyframe toe aan het frame **Over**.
4. Selecteer de Rectangle-tool en teken een rechthoek op de stage met een lijndikte van **0,5** en een lichtgele vulkleur. (Tooltips zijn meestal lichtgeel, maar u mag zelf natuurlijk een andere kleur kiezen.) Tekent de rechthoek op de plek waar u de tooltip wilt laten verschijnen. Meestal is dit links- of rechtsboven een button als de button onder aan de pagina staat, en links- of rechtsonder als de button boven aan de pagina staat.
5. Selecteer de Text-tool, kies een geschikt, niet te zwaar lettertype (wij kozen **Verdana, 11** punten, **zwart**) en typ een toelichtende tekst in de rechthoek, zoals **Klik voor meer informatie**.
6. Breng de tekst en de rechthoek op maat met de gereedschappen in de Modifier Tray en lijn correct uit, zoals bijvoorbeeld in figuur 12.14 wordt getoond.
7. Selecteer het frame **Hit**, voeg een keyframe toe en verwijder hier de zojuist toegevoegde tooltip. Indien u dit achterwege laat, zou het Over-effect al optreden zodra de muiswijzer boven het gebied van de tooltip komt, en dat is niet de bedoeling. Alleen de button zelf is een aanklikbaar gebied.
8. Keer terug naar de hoofd-movie en sla het bestand op. Met **Control, Test Movie** kunt u direct controleren of de tool tip goed functioneert.



Figuur 12.14: Verklarende teksten worden toegevoegd aan het Over-frame

Het resultaat van deze oefening staat online als **oef_1206 fla**. Houd er wel rekening mee dat deze techniek vooral geschikt is voor algemene buttons, met functies als **Vooruit, Terug, Home**, enzovoort. Het is niet mogelijk om *per instantie* een tooltip aan een button toe te voegen, u voegt de verklarende tekst toe aan het algemene button symbool. Het effect treedt daarom automatisch op bij alle instanties van die button.

Merk op dat het gebied dat door de button beïnvloed wordt niet beperkt is tot de initiële omvang van de button! De stage voor een button-symbool kan volledig benut worden. Zo kunt u bijvoorbeeld

honderden pixels naar rechts of naar onder iets laten gebeuren op het moment dat het **Over**-effect optreedt. Dit geeft zeker verrassende gevolgen.

12.7 Overige tips en trucs

Tot zover de beknopte behandeling van enkele veelgebruikte Flash-technieken. Voor meer informatie over allerhande technieken, tips en trucs en combinaties van dit alles verwijzen we graag naar appendix B, waar u veel Internet-adressen met extra achtergrondinformatie aantreft. Een paar korte hints kunnen u verder op weg helpen bij het samenstellen van uw Flash-projecten:

- <http://www.adversite.nl/flashles/overzicht.htm> : Een van de pagina's van Flash-les, een Nederlandstalige site waarin allerlei technieken worden besproken
- <http://www.sip.be/sji/ndl/koen/flash/index.htm> : Eenvoudige Nederlandstalige site met diverse Flash technieken uitgelegd.
- <http://www.turtleshell.com/guide/index.html> : Ook verschillende tips, met beknopte maar duidelijke uitleg. Gerubriceerd in diverse categorieën.

In de eindoefening van dit laatste hoofdstuk worden de belangrijkste technieken nogmaals herhaald, maar moedigen u daarnaast van harte aan om ook zelf de mogelijkheden van Flash verder te verkennen en ruimschoots het Internet op te gaan om te zien wat andere Flashers ervan bakken. Wij wensen u veel succes en vooral veel plezier met het gebruik van Flash!

12.8 Eindoefening

- Kunt u de movie van oefening 12.2 zodanig uitbreiden dat:
 - De button in elke scene op elk moment zichtbaar is en beschikbaar om op te klikken, zodat naar de volgende scene wordt gesprongen;
 - De huidige scene herhaald wordt totdat op de button wordt geklikt? Met andere woorden: de gebruiker ziet continu een grijze bal stuiteren totdat hij op de button **Scene 2** klikt, waarna hij continu een blauwe bal ziet stuiteren, totdat hij weer op de knop klikt om terug te keren naar de eerste scene.
 - Er op elke moment een Stop-knop zichtbaar is waarmee de animatie onderbroken kan worden?
- Neem de movie **oef_1202 fla** als basis en breid deze als volgt uit:
 - Maak een scene die als 'hoofdmenu' fungeert, hierop bevinden zich drie knoppen.
 - Het klikken op een van de knoppen laadt een scene naar keuze. Knop 1 laadt bijvoorbeeld een scene met een groene stuiterende bal, knop 2 een scene met een blauwe stuiterende bal, en knop 3 een scene met grijze stuiterende bal.
 - Programmeer in elke scene een knop om terug te keren naar het hoofdmenu.

Dit is een triviaal voorbeeld (met enkele nietszeggende stuiterende ballen), maar bedenk dat hiermee de basis is gelegd voor een volledige Flash-driven website. Veel sites bestaan immers uit een hoofdmenu met daaronder verschillende pagina's. Waar in traditionele sites elke keuze een apart HTML-bestand is, kan in een Flash-site elke menukeuze een aparte scene (of aparte movie) zijn. Wordt HTML zo overbodig?!

- Open de movie **oef_1206 fla** en voeg zelf een layer met een geschikt geluid toe aan de movie clip in deze button. Zo wordt de button nog verder uitgebreid: een animatie, een tool tip en een ondersteunend geluidseffect. Breid de button zelf nog verder uit, of ontwerp een serie bij elkaar horende navigatiebuttons voor een site (**Volgende**, **Terug**, **Home**, enzovoort).
- Bezoek de Internet-adressen die aan het einde van dit hoofdstuk en in appendix A van het boek zijn genoemd om meer informatie over Flash te vergaren en te zien waar andere gebruikers tegenaan lopen (hiervoor zijn de diverse forums buitengewoon geschikt). Zo krijgt u een goed inzicht in de wijze waarop anderen met Flash omgaan, en ontdekt u technieken en onderwerpen waar u zelf waarschijnlijk nooit aan gedacht zou hebben.

12.9 Samenvatting

In dit laatste hoofdstuk kwamen weinig nieuwe Flash-technieken aan bod, het ging meer om het slim combineren van bekende methoden en reeds aanwezige kennis. We keken onder meer naar de volgende wijzen waarop Flash ingezet kan worden.

- Als een project omvangrijk wordt, is het verstandig om de movie over meerdere scenes te verdelen. Elke scene bevat een logische sectie uit het project. U voegt een nieuwe scene toe aan een movie via **Insert, Scene**.
- Standaard speelt Flash de scenes achter elkaar af, in de volgorde waarin ze in het palet **Scene** zijn gerangschikt. U opent dit palet via **Window, Scene** of **Shift+F2**.
- Met ActionScript-opdrachten in frames of voor buttons schakelt u makkelijk naar de vorige of volgende scene. Het meest voor de hand liggende commando is hierbij **Go To and Stop** of **Go To and Play**.
- Preloading kan vooral bij grotere projecten van pas komen, wanneer het even duurt om de movie te laden. De gebruiker krijgt dan alvast wat te zien en weet dat hij niet voor niets aan het wachten is.
- Een preloader is een aparte scene in een movie die net zolang wordt herhaald tot dat een bepaald frame in de movie is geladen. Hiervoor wordt de opdracht **ifFrame Loaded** gebruikt.
- Een movieclip of kortweg MC is een apart symbool dat een eigen timeline heeft, die onafhankelijk van de timeline uit de hoofd-movie doorloopt.
- In een MC kunnen zoveel layers, tweens en symbolen worden opgenomen als nodig is om de clip samen te stellen. De aldus samengestelde movie clip kan vervolgens overal in de movie als symbool gebruikt worden.
- Movieclips worden veel ingezet om geanimeerde buttons te maken. In een van de frames van de button (**Up, Over, of Down**) wordt dan de clip opgenomen. Deze wordt afgespeeld zodra het betreffende event optreedt. Het heeft geen zin om movie clips toe te voegen aan de fase **Hit**.
- Het gebied dat een button kan bestrijken, is niet beperkt tot de omvang van het aanklikbare gebied van de button. Dit biedt interessante mogelijkheden om elders op de stage wat te laten gebeuren indien een bepaalde button-fase optreedt. In paragraaf 12.6 maakten we zo een eigen tooltip voor een button.
- We raden u van harte aan om zelf veel met Flash te experimenteren en niet te schromen om via het Internet gevonden technieken toe te passen. Met vragen kunt u terecht op de vele voortreffelijke Flash forums (zie ook appendix A in het boek).