

## 2.1 Einführung

### 2.1.1 Adobe Animate



#### Aus Flash ...

Knapp 20 Jahre war *Flash* führend im Bereich der Web-Animationen.

Doch Flash war – im Unterschied zu den offenen Standards HTML5 und CSS3 – ein proprietäres, firmeneigenes Format. Zum Abspielen von Flash-Animationen war der Flash-Player erforderlich. Steve Jobs von Apple hatte sich jedoch geweigert, diesen Player in iOS bzw. den Browser Safari zu integrieren, so dass Flash-Filme auf iPhones und iPads nicht abspielbar waren. Hierdurch sahen sich große Portale wie YouTube gezwungen, von Flash-Videos auf HTML5-Videos umzusteigen.



#### ... wurde Animate

Mittlerweile hat Adobe reagiert und Anfang 2016 als Flash-Nachfolger *Animate* auf den Markt gebracht. Animate ist Bestandteil der Adobe Creative Cloud (CC). Wer Flash kennt, dem wird der Umstieg leicht fallen, weil die Bedienung fast identisch geblieben ist. Neu ist, dass Adobe Animate auf offene Standards setzt und Animationen z. B. in HTML5- und JavaScript-Dateien exportiert werden können.

Adobe Animate stellt eine komplette Entwicklungsumgebung für interaktive multimediale Anwendungen bereit. Mit ihr können nicht nur Animationen, sondern komplette interaktive Anwendungen realisiert werden. Hierfür wurde die Flash-eigene Programmiersprache *ActionScript 3.0* bei Animate integriert.

Typische Einsatzgebiete für Animate-Applikationen könnten sein:

- Animationen aller Art – von einfachen Bild-für-Bild-Animationen bis zu 3D-Animationen mit inverser Kinematik,
- interaktive Benutzeroberflächen für Online- und Offline-Anwendungen,

- z. B. für Webseiten oder Apps,
- audiovisuelle Anwendungen, z. B. Video-Streaming,
- Rich Internet Applications (RIAs), also Software, die ohne Installation direkt via Internet genutzt werden kann, z. B. Computerspiele, Navigations- und E-Learning-Systeme.

#### Kein Player erforderlich!

Flash ist letztlich daran gescheitert, dass zum Abspielen eine spezielle Software namens Flash-Player erforderlich war.

Mit HTML5, CSS3, JavaScript, Canvas haben sich hingegen – zur Freude der Entwickler – Technologien durchgesetzt, die quelloffen sind, und damit weltweit von jedermann frei genutzt werden können. Das Internetkonsortium W3C trägt für die erforderliche Standardisierung Sorge.

Adobe geht mit Animate folgenden Weg: Animate-Anwendungen werden in einem proprietären Dateiformat (Endung wie bei Flash: .fla) gespeichert. Zur Nutzung im Web erfolgt der Export in eine HTML5-/JavaScript-Datei. Damit ist das Abspielen auf allen Endgeräten und Betriebssystemen möglich, vorausgesetzt, dass im Browser JavaScript aus Sicherheitsgründen nicht deaktiviert wurde.

#### Ausblick

Adobe Animate ist seit 2016 auf dem Markt. Ob die Software den Stellenwert erlangen wird, den Flash hatte, ist heute noch nicht abschätzbar.

Es könnte sich aber wieder einmal als genialer Schachzug von Adobe erweisen, dass das „Look & Feel“ von Animate exakt mit Flash übereinstimmt. Der Umstieg auf Animate wird damit für frühere Flash-Entwickler einfach und könnte ein Anreiz sein, zukünftig mit dieser Software zu arbeiten.

## 2.1.2 Entwicklungsumgebung

Wenn Sie mit Adobe-Produkten wie Photoshop, Illustrator oder InDesign vertraut sind, werden Sie sich auf der Animate-Benutzeroberfläche schnell zu rechtfinden, da diese an die genannten Produkte angepasst wurde.

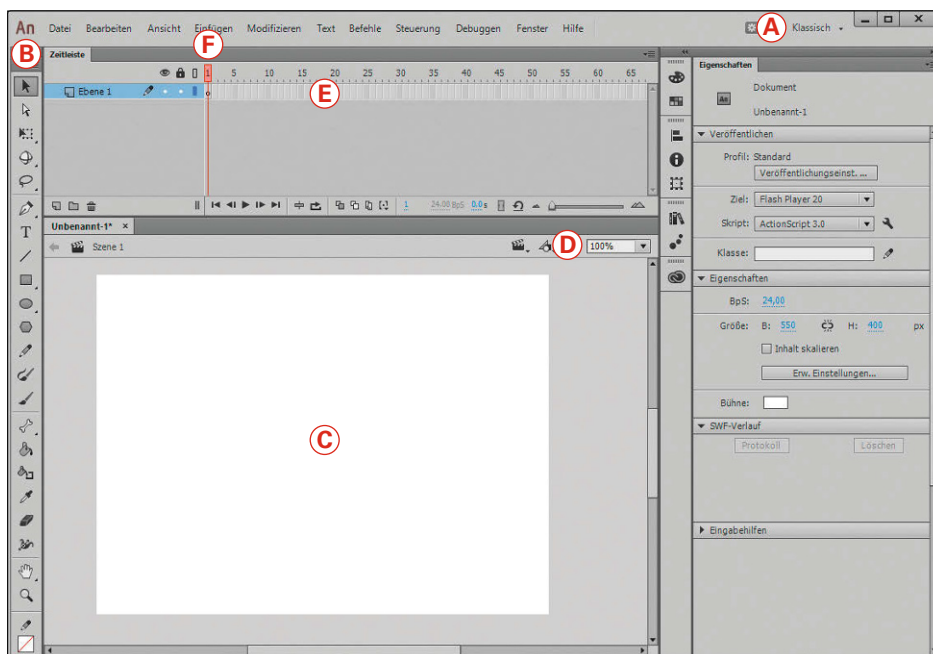
Stellen Sie als Grundlayout Ihrer Benutzeroberfläche in der oberen Menüleiste **Klassisch A** ein. Dadurch werden die Werkzeuge **B** zur Erstellung und Bearbeitung von Objekten am linken Rand platziert, wie Sie dies von den anderen Programmen gewohnt sind.

Die weiße Fläche in der Mitte heißt Bühne **C** – sie ist der eigentliche Ort des Geschehens. Alle Objekte, die sich auf der Bühne befinden, sind später sichtbar: Grafiken, Animationen, Texte, Videos, Buttons. Objekte im grauen Bereich außerhalb der Bühne sind zwar in der Entwicklungsumgebung sichtbar,

im Film jedoch nicht. Über die Zoomfunktion **D** können Sie die Darstellung der Bühne vergrößern oder verkleinern.

Die Zeitleiste **E** können Sie mit einer früheren Filmrolle vergleichen, die abgespult und durch einen Projektor bewegt wurde. Der Unterschied ist, dass bei Animate der Film feststeht und sich dafür der Abspielkopf **F** bewegt. Beim Abspielen bewegt er sich von links nach rechts über die Zeitleiste. Alle Objekte, z. B. Texte oder Bilder, die sich im Kästchen unter dem Abspielkopf in Ebene 1 befinden, werden dargestellt.

Wie bei Photoshop und Illustrator ist es auch bei Animate unerlässlich, mit sinnvoll bezeichneten Ebenen bzw. Ebenengruppen zu arbeiten. Achten Sie vor allem auf eine strikte Trennung von statischen Objekten, z. B. einem Hintergrundbild, und animierten Objekten, da sonst schnell der Überblick verloren geht.



### Die Entwicklungsumgebung von Adobe Animate

Die Benutzeroberfläche wurde an die anderen Produkte der Adobe Creative Suite bzw. Creative Cloud angepasst.

## 2.2 Animate-Projekte

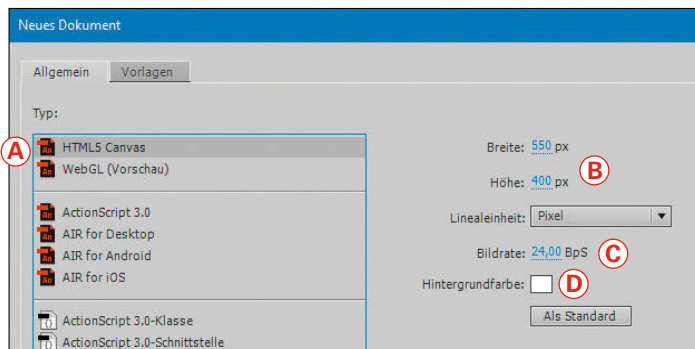
### 2.2.1 Voreinstellungen

Bereits zu Beginn Ihres Projekts sollten Sie wichtige Voreinstellungen vornehmen.

#### Making of ...

- 1 Legen Sie über *Datei > Neu...* ein neues Projekt an. Belassen Sie es bei der Grundeinstellung *HTML5 Canvas* **A**.
- 2 Legen Sie die Bühnengröße **B** fest: Breite x Höhe in Pixel. Beachten Sie, dass der Rechenaufwand mit zunehmender Größe steigt.
- 3 Die Bildrate **C**, also die Anzahl der Bilder pro Sekunde [BpS], spielt bei Animationen eine wichtige Rolle. Ist sie zu niedrig, „ruckelt“ die Animation. Ist sie zu hoch, kommt der Prozessor nicht mit und der Film stockt. Die Grundeinstellung von 24 BpS ist fürs Erste ein guter Wert.
- 4 Falls Sie keinen eigenen Hintergrund wünschen, geben Sie hier die Hintergrundfarbe der Bühne **D** vor.
- 5 Beenden Sie die Voreinstellungen mit OK.

#### Voreinstellungen eines Animate-Projekts



- 6 Wie bei Photoshop und Illustrator können Sie im Menü *Ansicht > Lineale* ein horizontales und vertikales Lineal einblenden, aus dem sich mit gedrückter Maustaste Hilfslinien auf die Bühne ziehen lassen.

- 7 Ein Gestaltungsraster können Sie im Menü *Ansicht > Raster einstellen*. Ebenfalls im Menü *Ansicht* legen Sie unter *Ausrichten* fest, wie Objekte an Hilfslinien bzw. am Raster ausgerichtet werden sollen.

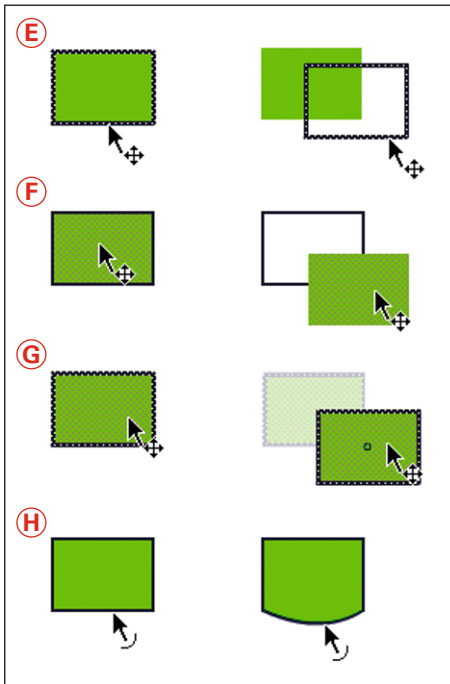
### 2.2.2 Grafiken

Zur Erstellung und Bearbeitung von Grafiken gibt es zwei Möglichkeiten:

#### Grafikmodus – Making of ...

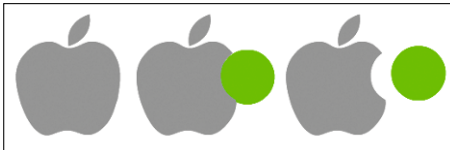
Der Grafikmodus ermöglicht eine individuelle Bearbeitung von Objekten bzw. von deren Bestandteilen. Hierin unterscheidet sich Animate von Illustrator.

- 1 Zeichnen Sie ein Objekt mit Kontur und Füllung, z. B. ein Rechteck.
- 2 Wenn Sie nur die Kontur bearbeiten wollen: Doppelklicken Sie auf die Kontur, danach können Sie diese löschen, verschieben, umfärben oder die Strichstärke ändern **E**.
- 3 Um nur die Füllung bearbeiten zu können, klicken Sie einmal auf die Füllung **F**.
- 4 Die Kombination von Kontur und Füllung erhalten Sie durch Doppelklick auf die Füllung **G**.
- 5 Animate ermöglicht die direkte Veränderung der Form eines Objekts: Berühren Sie die Kontur hierzu mit



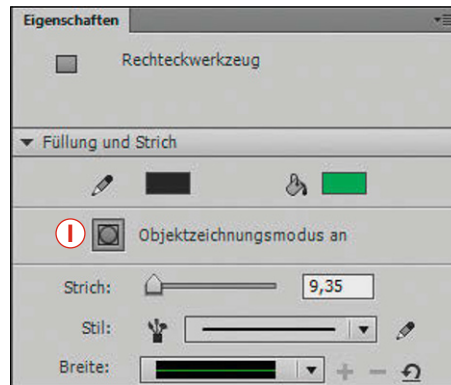
der Maus (Mauszeiger ändert sich) und ziehen Sie die Teilkontur danach mit gedrückter Maustaste **H**.

- 6 Sich überlagernde Objekte werden verbunden. Wenn Sie nachträglich ein Teilobjekt verschieben, wird der überlappende Teil aus dem unteren Objekt ausgeschnitten.

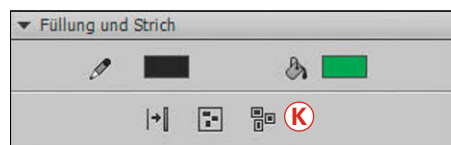


### Objektzeichnungsmodus – Making of ...

In diesem Modus werden grafische Elemente – wie Sie das von Illustrator kennen – als zusammengehörige Objekte behandelt.



- 1 Aktivieren Sie den Objektzeichnungsmodus **I**.
- 2 Zeichnen Sie ein Objekt, z. B. ein Rechteck.
- 3 Klicken Sie mit dem Auswahlwerkzeug **J** auf das Objekt: An der hellblauen Konturlinie erkennen Sie, dass es sich um ein zusammenhängendes Objekt handelt.
- 4 Um das Objekt im Grafikmodus zu bearbeiten, klicken Sie auf **Teilen** **K**.



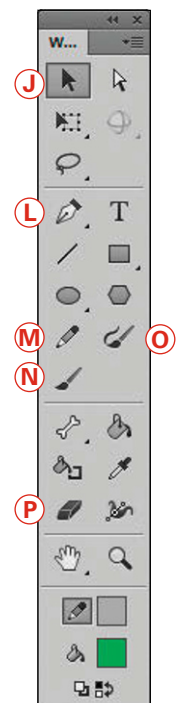
### Grafikwerkzeuge

Die Grafikwerkzeuge entsprechen in ihrer Funktion den Grafikwerkzeugen in Illustrator und Photoshop – wir gehen hier nicht näher darauf ein:

- Zeichenstift **L**
- Freihand **M**
- Einfacher Pinsel **N**
- Pinsel mit unterschiedlichen Strichstilen **O**
- Radiergummi **P**

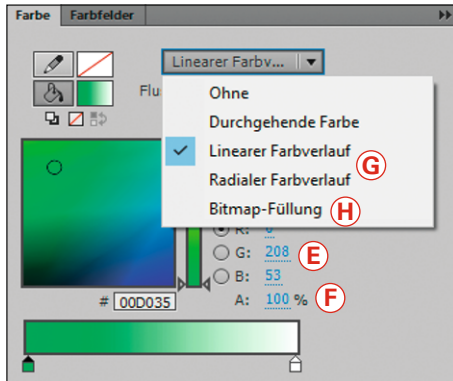
### Eigenschaften einer Form

Neben der Kontur- und Füllfarbe lassen sich die Strichstärke und Art der Kontur einstellen.



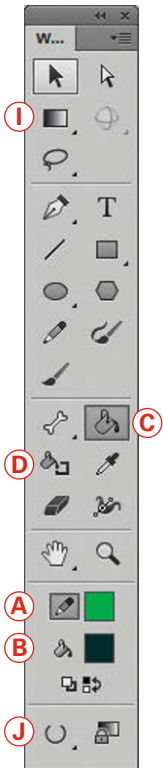
## Farbeinstellungen

In Animate können Sie RGB-Farbwerte definieren, lineare oder radiale Farbverläufe erstellen oder Texturen importieren.



## Farben und Farbverläufe – Making of...

- 1 Wählen die Konturfarbe **A** bzw. Füllfarbe **B** in der Werkzeugleiste.
- 2 Verwenden Sie den Fülleimer **C**, um die *Füllung* eines Objekts mit der gewünschten Farbe zu versehen.
- 3 Verwenden Sie das Tintenfass **D**, um die *Kontur* eines Objekts mit der gewünschten Farbe zu versehen.
- 4 Das Menü *Fenster > Farbe* bietet einige zusätzliche Möglichkeiten:
  - Farbangaben als RGB-Werte **E**
  - Transparenz über Alphakanal **F**
  - Linearer oder radialer Farbverlauf **G**
  - Verwenden von Texturen **H**
- 5 Zur Bearbeitung von Farbverläufen steht in der Werkzeugleiste hinter dem Frei-Transformieren-Werkzeug ein Farbverlaufs-Werkzeug **I** bereit. Es ermöglicht beispielsweise, lineare Farbverläufe zu drehen oder den Mittelpunkt radialer Farbverläufe zu verändern.
- 6 Ein interessantes Tools ist das „Lücke schließen“-Werkzeug **J**: Es

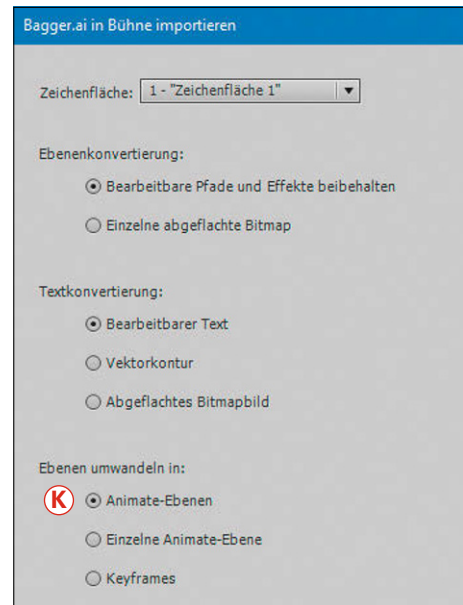


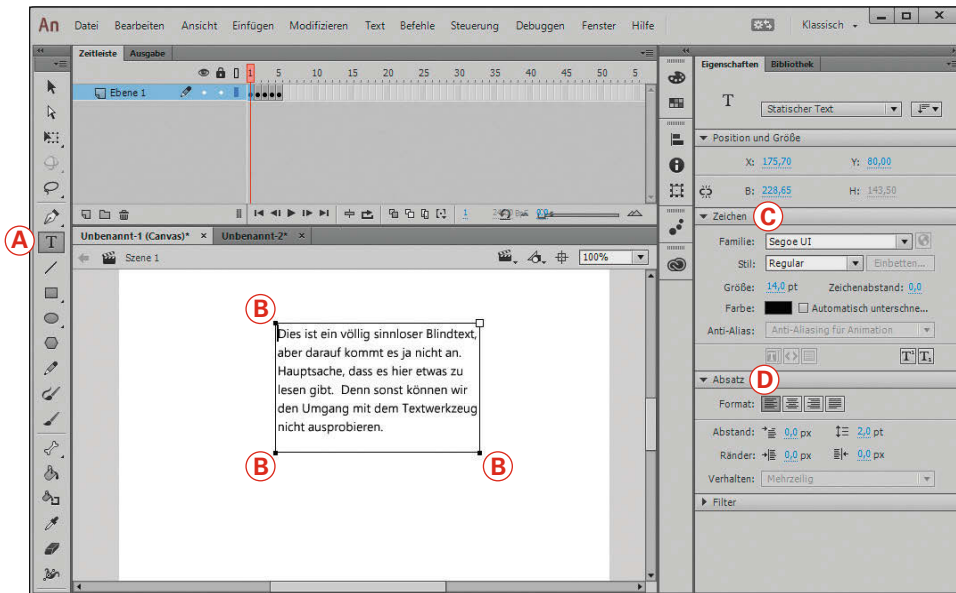
ermöglicht, Objekte mit einer Füllfarbe zu versehen, selbst wenn die Kontur des Objekts nicht komplett geschlossen ist.

## Bild-/Grafikimport – Making of...

Sind Sie ein Fan der Animate-Grafik geworden? Wenn nicht, dann können Sie Bilder bzw. Grafiken aus Photoshop bzw. Illustrator importieren:

- 1 Wählen Sie im Menü *Datei > Importieren > In Bühne importieren*. (Auf die Möglichkeit, in die *Bibliothek* zu importieren, gehen wir später ein.) Neben Illustrator-Dateien (AI) und Photoshop-Dateien (PSD) können Sie auch GIF-, PNG-, JPG- oder SVG-Dateien importieren.
- 2 Eine wichtige Option ist, dass Sie in Illustrator oder Photoshop angelegte Ebenen in Animate-Ebenen **K** umwandeln können.





### Text formatieren

Animate stellt etliche Funktionen zur Verfügung, die Sie aus Satzprogrammen wie InDesign kennen.

### 2.2.3 Text

Adobe Animate unterscheidet drei Arten von Textfeldern:

- **Statischer Text**  
Text, der in Animate erstellt wird. Er ist in der späteren Anwendung unveränderlich.
- **Dynamischer Text**  
Text, der aus einer Datei oder Datenbank in Animate eingefügt wird, wird als dynamisch bezeichnet. Dynamischer Text wird immer dann benötigt, wenn der Text veränderbar sein muss, z. B. zur Anzeige eines Börsenkurses. Mehr hierzu finden Sie auf Seite 44.
- **Eingabetext**  
Text, der in der späteren Anwendung durch den Nutzer verändert werden kann, z. B. in einem Formular. Diese Option ist nur möglich, wenn die Datei mit ActionScript 3.0 erstellt wird.

### Texterfassung – Making of ...

Alle unveränderlichen (statischen) Texte Ihres Projekts erstellen und formatieren Sie direkt in Animate.

- 1 Wählen Sie das Textwerkzeug **A** der Werkzeugleiste aus und klicken Sie an einer beliebigen Stelle auf die Bühne.
- 2 Geben Sie den Text manuell ein oder kopieren Sie ihn aus einer Textverarbeitung ins Textfeld.
- 3 Passen Sie die Größe des Textfelds durch Ziehen der kleinen Quadrate **B** wie gewünscht an.
- 4 Zur Formatierung des Textes stehen Ihnen unter den Eigenschaften rechts mit den Paletten Zeichen **C** und Absatz **D** die in der Textverarbeitung üblichen Optionen zur Verfügung.





## 2.2.4 Symbole, Instanzen, Bibliothek

Bisher haben Sie das Erstellen oder Importieren von Grafiken und Bildern sowie die Texterfassung kennengelernt. Um zu jeder Zeit in Ihrem Projekt darauf zurückgreifen zu können, ist es sinnvoll, diese an zentraler Stelle zu hinterlegen.

Adobe Animate stellt hierfür eine *Bibliothek* zur Verfügung. Um die Grafiken und Texte Ihres Projekts in der Bibliothek speichern zu können, müssen diese in sogenannte *Symbole* umgewandelt (konvertiert) werden.

Bei Webapplikationen geht es immer auch darum, durch geringe Datenmengen kurze Ladezeiten zu erhalten. Dies spielt insbesondere bei Grafiken, Sounds und Videos eine wichtige Rolle. Um dies zu erreichen, wird jedes Objekt einmalig als sogenanntes Symbol gespeichert und kann aber beliebig oft verwendet werden. Animate unterscheidet drei Typen von Symbolen:

- **Movieclip-Symbole A** benötigen Sie immer dann, wenn Sie Animationen erstellen wollen. Sie erhalten eine eigene Zeitleiste.
- **Schaltflächen-Symbole B** dienen, wie der Name sagt, zur Erstellung von Buttons oder anderer anklickbarer Elemente.
- **Grafik-Symbole C** werden für alle statischen Elemente Ihres Films verwendet, z. B. Hintergründe, Grafiken. Animationen sind zwar auch hier möglich, jedoch unflexibler.

### Symbol erstellen – Making of ...

- 1 Wählen Sie das Auswahlwerkzeug **F**.
- 2 Markieren Sie das Objekt, aus dem ein Symbol erstellt werden soll: Bei Text durch Anklicken des Text-



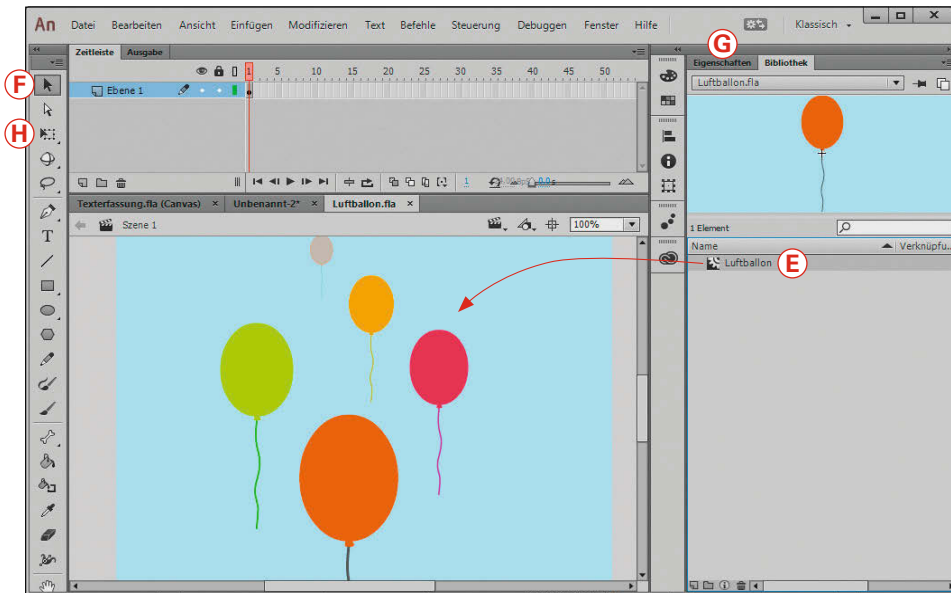
rahmens, bei einer Grafik durch Umrahmung der Grafik mit gedrückter linker Maustaste. (Alternativ können Sie das Lasso-Werkzeug verwenden.)

- 3 Wählen Sie im Menü *Modifizieren* > *In Symbol konvertieren* oder machen Sie einen Rechtsklick und wählen Sie *In Symbol konvertieren ...*
- 4 Geben Sie Ihrem Symbol einen aussagekräftigen Namen **D**.
- 5 Wählen Sie den Symboltyp, z. B. Movieclip **A**, und bestätigen Sie mit OK. Das in ein Symbol konvertierte Text- oder Grafikobjekt wird durch einen dunkelblauen Rahmen gekennzeichnet.
- 6 Wählen Sie im Menü *Fenster* > *Bibliothek*. Das neue Symbol taucht dort auf.

Jedes Symbol der Bibliothek kann im Animate-Projekt beliebig oft eingesetzt werden. Die Verwendung eines Symbols auf der Bühne wird als Symbolinstanz oder kurz *Instanz* bezeichnet.

### Instanzen erstellen – Making of ...

- 1 Ziehen Sie ein Symbol **E** mit gedrückter Maustaste auf die Bühne.
- 2 Wiederholen Sie den Vorgang, um weitere Instanzen zu erzeugen.



### Symbole und Instanzen

Um ein Objekt, z. B. Text oder Grafik, steuerbar zu machen, muss es in ein Symbol konvertiert werden. Danach steht es in der Bibliothek zur Verfügung.

Instanzen sind Kopien des Symbols auf der Bühne. Wie zu sehen ist, können sich Instanzen z. B. in Farbe und Größe vom Original unterscheiden.

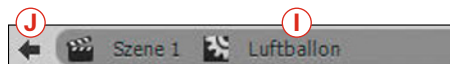
Jede Instanz kann individuell in Form und Aussehen verändert werden:

- 3 Klicken Sie hierzu die gewünschte Instanz mit dem Auswahlwerkzeug **F** an.
- 4 Wählen Sie Eigenschaften **G**, um Änderungen der Farbe, Helligkeit oder Durchsichtigkeit (Alpha) vorzunehmen.
- 5 Wählen Sie das Frei-Transformieren-Werkzeug **H**, um die Größe zu ändern oder um die Instanz zu drehen.

Die Änderungen einer Instanz wirken sich nicht auf das Symbol selbst aus. Umgekehrt gilt dies nicht: Die nachträgliche Änderung des Symbols wirkt sich auf *alle* Instanzen des Symbols aus. Hier ist also Vorsicht geboten!

### Symbol ändern – Making of ...

- 1 Um Symbole bearbeiten zu können, *doppelklicken* Sie entweder auf eine Instanz des Symbols oder auf das Symbol **E** in der Bibliothek.

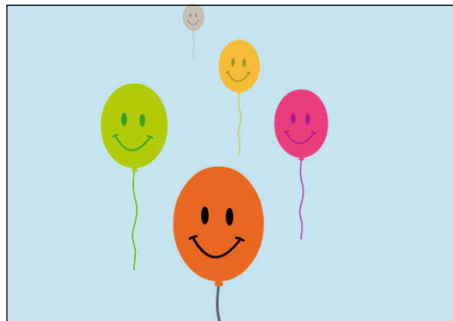


- 2 Am oberen Bühnenrand erkennen Sie, dass Sie sich nun im Bearbeitungsmodus des Symbols (hier: Luftballon) befinden **I**.
- 3 Zur Bearbeitung des Symbols stehen Ihnen sämtliche Grafikwerkzeuge zur Verfügung – siehe Abschnitt 2.2.2.
- 4 Verwirrend ist, dass Sie auch im Bearbeitungsmodus eines Symbols eine Zeitleiste sehen: Diese hat nichts mit der Hauptzeitleiste Ihres Projekts zu tun, sondern gehört zum



Symbol. Wir werden diese Symbolzeitleiste später nutzen, um innere Animationen zu erstellen. Dies könnten bei einem animierten Auto die sich drehenden Räder sein. Die äußere Animation (auf der Hauptzeitleiste) ist dann die Bewegung des Autos.

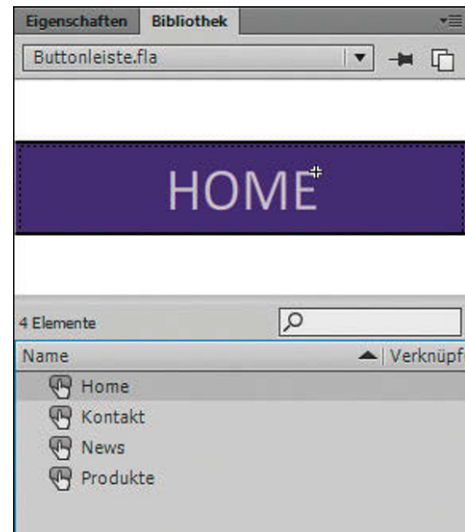
- 5 Um den Bearbeitungsmodus des Symbols wieder zu verlassen, klicken Sie entweder auf den blauen Pfeil oberhalb der Bühne (J vorige Seite) oder doppelklicken Sie auf die Bühne. Die Änderungen wirken sich auf alle Instanzen aus.



Oft kommt es vor, dass Sie Symbole in mehreren Varianten benötigen, z. B. Schaltflächen, die einen unterschiedlichen Text erfordern, aber ansonsten identisch sind. Mit Instanzen ist dies nicht möglich. Es bietet sich jedoch an, ein bereits erstelltes Symbol zu vervielfältigen und danach die Kopie zu ändern.

#### Symbol duplizieren – Making of ...

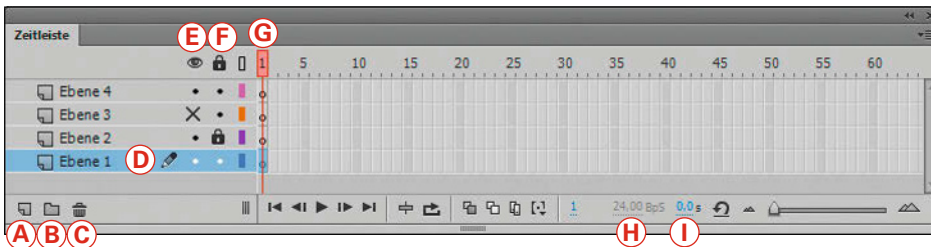
- 1 Entwerfen Sie einen Home-Button, z. B. wie im Screenshot dargestellt.
- 2 Markieren Sie die Grafik.



- 3 Konvertieren Sie die Grafik in ein Symbol. Wählen Sie in diesem Fall die Option *Schaltfläche*.
- 4 Um die Schaltfläche zu duplizieren, rechtsklicken Sie in der Bibliothek auf das Symbol und wählen *Duplizieren*.
- 5 Geben Sie der Kopie einen aussagekräftigen Namen.
- 6 Doppelklicken Sie auf das duplierte Symbol, um den Text zu verändern.
- 7 Verlassen Sie den Bearbeitungsmodus durch Anklicken des Pfeils A links oberhalb der Bühne.



- 8 Wiederholen Sie die Schritte 4 bis 7 für die weiteren Schaltflächen.



## 2.2.5 Zeitleiste

Die Zeitleiste ist die Steuerzentrale Ihres Animate-Projekts: In der Zeitleiste legen Sie fest, welche Objekte *wann* und *wie lange* sichtbar sein sollen.

Der Umgang mit der Zeitleiste erfordert einige Übung. Wir versuchen deshalb zunächst, Ihnen den Aufbau und das Prinzip der Zeitleiste zu erklären.

In senkrechter Richtung der Zeitleiste sind die Ebenen angeordnet. Die Ebenentechnik ist Ihnen aus Programmen wie Photoshop, Illustrator oder InDesign bekannt. Ebenen ermöglichen eine strukturierte Anordnung von Objekten, wobei obere Ebenen untere Ebenen überdecken. Bei Animate sollten Sie grundsätzlich mit Ebenen arbeiten und für den Hintergrund, Bilder, Texte, Animationen, Sound und Video *immer* eigene Ebenen verwenden.

### Ebenen – Making of ...

- 1 Durch Anklicken des Blatt-Icons **A** werden neue Ebenen erzeugt. Diese sollten Sie durch Doppelklick auf den Namen sinnvoll benennen.
- 2 Wie bei Photoshop und Illustrator können Sie Ebenen zu Gruppen zusammenfassen **B**.
- 3 Um eine markierte Ebene zu löschen, klicken Sie auf das Papierkorb-Icon **C**.

- 4 Die gerade aktive Ebene erkennen Sie am Stift-Icon **D**. Sie können sie mit gedrückter Maustaste nach oben oder unten verschieben.
- 5 Um die Ebene auszublenden, klicken Sie in der Ebene auf den Punkt unter dem Auge-Icon **E**.
- 6 Um ein wesentliches Verschieben oder Löschen der Elemente zu verhindern, ist es sinnvoll, fertig bearbeitete Ebenen zu sperren (Schloss-Icon **F**).

In waagrechter Richtung der Zeitleiste befindet sich die Zeitachse. Beim Abspielen der Animation läuft der rote Abspielkopf **G** von Bild 1 nach rechts, wobei die eingestellte Anzahl an Bildern pro Sekunde [BpS] **H** die Geschwindigkeit bestimmt.

Die Einstellung 24 BpS besagt, dass der Abspielkopf nach einer Sekunde in Bild 24 angelangt ist. Zwischen Bildern und Zeit **I** besteht folgender Zusammenhang:

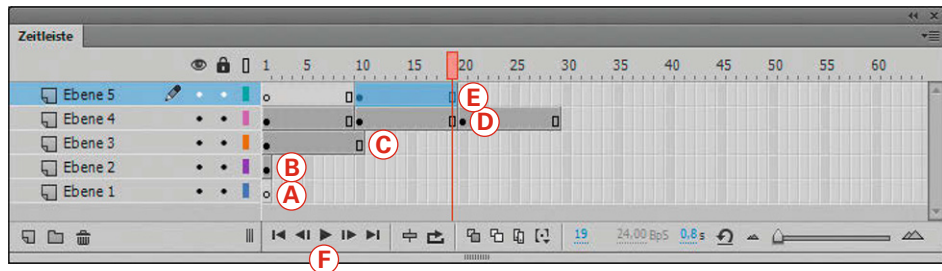
#### Abspieldauer einer Animation

$$t = \frac{B}{\text{BpS}} \text{ [s]}$$

t: Gesamtdauer in Sekunden

B: Gesamtanzahl der Bilder (frames)

BpS: Bilder pro Sekunde (Bild- oder Framerate)



Symbole können nicht direkt in der Zeitachse platziert werden. Aus einem Symbol muss immer erst eine Instanz auf der Bühne erstellt werden. Diese Instanz wird in ein leeres Schlüsselbild der Zeitachse eingefügt.

#### Zeitachse – Making of ...

##### 1 Leeres Schlüsselbild

Erzeugen Sie eine neue Ebene. Diese enthält immer ein leeres Schlüsselbild in Bild 1. Leere Schlüsselbilder werden durch einen weißen Kreis **A** symbolisiert.

##### 2 Schlüsselbild mit Inhalt versehen

Klicken Sie auf ein leeres Schlüsselbild. Ziehen Sie nun ein Symbol aus der Bibliothek auf die Bühne (nicht in die Zeitachse!) – eine Instanz wird erstellt. Das Schlüsselbild *mit* Inhalt wird durch einen schwarzen Kreis **B** symbolisiert.

##### 3 Schlüsselbild verlängern

Um ein Schlüsselbild beim Abspielen über mehrere Bilder anzuzeigen, müssen Sie es verlängern: Klicken Sie hierzu in das Kästchen, an dem das Schlüsselbild *enden* soll. Wählen Sie *Einfügen > Zeitachse > Bild* oder nach Rechtsklick *Bild einfügen*. Ein weißes Rechteck **C** kennzeichnet das Ende des Schlüsselbilds.

##### 4 Weitere Schlüsselbilder

Um eine Ebene mit weiteren Schlüsselbildern **D** zu versehen, klicken Sie ins gewünschte Kästchen und wählen *Einfügen > Zeitachse > Schlüsselbild* oder nach Rechtsklick *Schlüsselbild einfügen*.

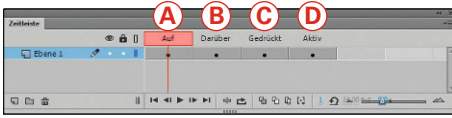
##### 5 Schlüsselbilder verschieben

Markieren Sie die Schlüsselbilder, die Sie verschieben möchten, mit gedrückter Maustaste. Markierte Bilder werden hellblau **E** dargestellt. Verschieben Sie danach die markierten Bilder.

##### 6 Nicht benötigte Bilder oder Schlüsselbilder können Sie löschen, indem Sie diese markieren und nach Rechtsklick *Bilder entfernen* wählen.

##### 7 Sie können Ihre Animation mittels Play-Button **F** oder per Return-Taste abspielen.

Ein Animate-Projekt besitzt nicht nur eine Zeitachse – jedem Symbol ist eine eigene Zeitachse zugeordnet. Sie öffnen diese Zeitachse durch Doppelklick auf das Symbol. Eine Besonderheit stellen Schaltflächen-Symbole dar, da sich deren Zeitachse von der „normalen“ Zeitachse unterscheidet. Dies hat den Grund, dass eine Schaltfläche ein interaktives Element ist, das auf Benutzerverhalten (optisch) reagieren sollte.



### Schaltflächen animieren – Making of ...

- 1 Erstellen Sie ein neues Symbol des Typs Schaltfläche oder doppelklicken Sie – falls vorhanden – auf eine bereits bestehende Schaltfläche. Die Zeitleiste bietet die Möglichkeit, vier Zustände einer Schaltfläche zu definieren.
- 2 **Grundzustand „Auf“ A**  
Der schwarze Kreis zeigt, dass sich hier bereits ein Schlüsselbild befindet. Es enthält die Schaltfläche in ihrem Grundzustand.
- 3 **Zustand „Darüber“ B**  
Der Zustand ist später aktiv, wenn sich der Mauszeiger auf der Schaltfläche befindet. Erstellen Sie ein Schlüsselbild: Menü *Einfügen* > *Zeitleiste* > *Schlüsselbild*. Ändern Sie das Aussehen der Schaltfläche wie gewünscht.
- 4 **Zustand „Gedrückt“ C**  
Der Zustand ist aktiv, wenn der Nutzer die Maustaste gedrückt hält. Erstellen Sie ein Schlüsselbild: Menü *Einfügen* > *Zeitleiste* > *Schlüsselbild*. Ändern Sie das Aussehen der Schaltfläche wie gewünscht.
- 5 **Zustand „Aktiv“ D**  
Mit diesem Schlüsselbild können Sie den anklickbaren Bereich definieren. Beispiel: Sie verwenden als Schaltfläche Text in geringer Schriftgröße. Damit der Nutzer nicht „zielen“ muss, um den Text anzuklicken, vergrößern Sie den

anklickbaren Bereich, indem Sie ein Rechteck zeichnen, das größer ist als das Textfeld. Wichtig: Der aktive Bereich ist für den Nutzer unsichtbar! Es handelt sich lediglich um eine Maske, die den anklickbaren Bereich definiert.

### 2.2.6 Sound und Video importieren

Sounds und Videos lassen sich auf einfache Weise in Ihr Animate-Projekt integrieren, indem sie in die Bibliothek importiert und im gewünschten Schlüsselbild auf der Bühne platziert werden.

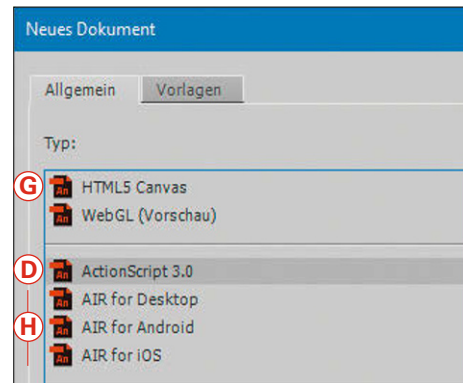
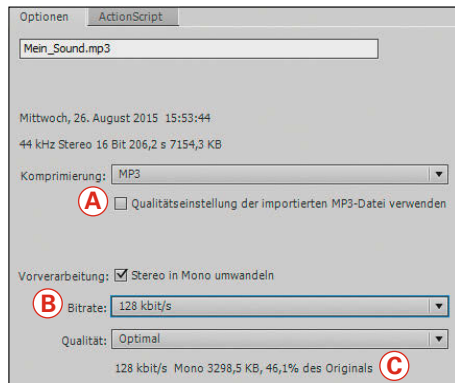
In Animate können Sie folgende Audioformate importieren:

- WAV: Sounddateien ohne Qualitätsverlust, aber mit großer Datenmenge
- MP3: Sounddateien mit Qualitätsverlust, der aber bei einer Bitrate von 128 kbit/s oder höher vernachlässigbar ist.

Grundsätzlich gilt, dass Sie Sounds bevorzugt in hoher Qualität importieren sollten. Animate bietet die Möglichkeit der Datenreduktion an, so dass Sie – in Abhängigkeit von Ihrem Projekt – einen Kompromiss zwischen möglichst hoher Qualität und möglichst niedriger Datenmenge wählen können.

### Sound – Making of ...

- 1 Wählen Sie Menü *Datei* > *Importieren* > *In Bibliothek importieren*, um einen Sound in die Bibliothek Ihres Animate-Projekts zu importieren.
- 2 Erstellen Sie in der Zeitleiste eine neue Ebene für den Sound und fügen Sie dort ein Schlüsselbild ein, wo der Sound beginnen soll.
- 3 Ziehen Sie den gewünschten Sound aus der Bibliothek auf die Bühne (nicht in die Zeitleiste).



- 4 Verlängern Sie das Schlüsselbild auf die Länge des Sounds.
- 5 Testen Sie den Sound durch Abspielen der Animation.

Um die Datenmenge und damit Ladezeit zu verringern:

- 6 Machen Sie in der Bibliothek einen Rechtsklick auf den Sound und wählen Sie *Eigenschaften*.
- 7 Entfernen Sie das Häkchen **A**.
- 8 Wählen Sie die Bitrate **B**: 128 kBit/s entspricht nahezu CD-Qualität. Im Beispiel wird hierdurch die Datenmenge mehr als halbiert **C**.

Bei Video ist es leider etwas komplizierter: Mit HTML5-Canvas-Dateien ist die Verwendung von Video (Stand: 2016) leider nicht möglich. Sie müssen hierfür auf die Flash-Technologie zurückgreifen.

#### Video – Making of ...

- 1 Legen Sie über *Datei > Neu...* ein neues Projekt an. Wählen Sie *ActionScript 3.0* **D**.

- 2 Um ein Video zu importieren, wählen Sie im Menü *Datei > Importieren > Video importieren...* Animate startet einen Assistenten zur Einbindung des Videos.

- 3 Wählen Sie die gewünschte Video-datei aus **E**. Mögliche Videoformate sind u. a.:  
FLV, F4V: Flash-Video  
MP4: MPEG-Video  
MOV: Quicktime-Video

- 4 Nur die Flash-Videoformate lassen sich tatsächlich importieren, alle anderen Formate müssen als externe Dateien **F** vorliegen und werden lediglich verknüpft. Da Videos eine große Datenmenge besitzen, ist



es ohnehin sinnvoll, sie extern zu belassen. Klicken Sie auf *Weiter*.

- 5 Suchen Sie sich eine der Komponenten zur Steuerung des Videos aus und wählen deren Farbe.
- 6 Das Video wird in die Bibliothek eingefügt und kann verwendet werden.

## 2.2.7 Veröffentlichen

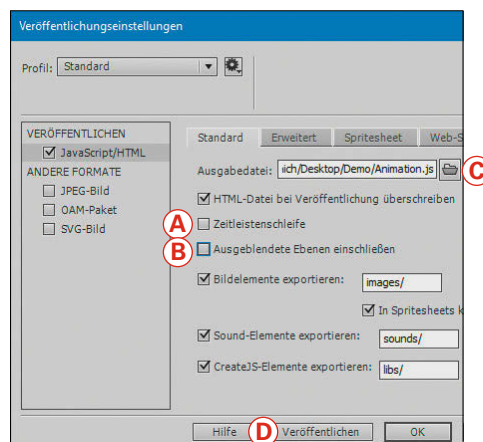
Veröffentlichen bedeutet, dass Ihr Animate-Projekt ohne Entwicklungs-umgebung im Webbrowser abspielbar wird. Folgende Varianten sind möglich:

- Bei *HTML5 Canvas* (G auf vorheriger Seite) handelt es sich um eine Technologie, die ohne Zusatzsoftware mit allen Betriebssystemen und mit allen aktuellen Browsern genutzt werden kann – der optimale Fall! Animate generiert hierzu zwei Dateien, eine HTML5-Datei und eine JS-(JavaScript-) Datei. Einzige Einschränkung ist, dass Video nicht möglich ist.
- *ActionScript 3.0* (D auf vorheriger Seite) benötigt zum Abspielen einen Flash-Player, der bekanntlich auf iOS-Geräten nicht vorhanden ist. Animate erzeugt neben einer HTML5-Datei eine SWF-(Flash-)Datei.
- Bei *AIR* (Adobe Integrated Runtime) (H auf vorheriger Seite) handelt es sich um eine sogenannte *Laufzeitumgebung*. Diese kann die Flash-Technologie einbetten und somit plattformunabhängig verfügbar machen. Mit AIR können Sie Apps für Desktop, Android und iOS generieren.

### Making of ...

Wir beschreiben hier die Veröffentlichung eines HTML5-Canvas-Projekts:

- 1 Sie können Ihr Projekt vorab im Browser testen, indem Sie im Menü *Steuerung > Testen* wählen.
- 2 Speichern Sie die finale Version Ihres Films (Dateiendung: .FLA) ab.
- 3 Legen Sie einen Projektordner an, da mehrere Dateien erzeugt werden.
- 4 Wählen Sie im Menü *Datei > Einstellungen für Veröffentlichungen...*
  - Setzen Sie das Häkchen **A**, wenn sich die Animation als Endlosschleife wiederholen soll.
  - Wählen Sie Option **B** nur, wenn auch ausgeblendete Ebenen berücksichtigt werden sollen.
  - Die übrigen Häkchen können Sie gesetzt lassen.
- 5 Wählen Sie den Projektordner für die Ausgabedateien **C**.
- 6 Beenden Sie mit *Veröffentlichen* **D**. Alternativ ist dies im Menü *Datei > Veröffentlichen* möglich.
- 7 Testen Sie das Ergebnis durch Doppelclick auf die HTML5-Datei.





## 2.3 Animationen

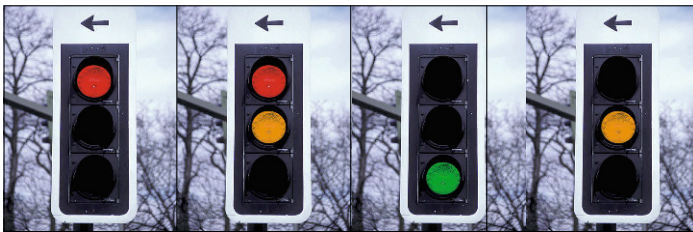
Der langjährige Erfolg von Flash – das jetzt zu Animate wurde – ist nicht zuletzt auf die umfangreichen Animationsmöglichkeiten zurückzuführen, die diese Programme bieten. Wir stellen Ihnen im Folgenden die Umsetzung der in Kapitel 1.2 eingeführten Animationstechniken vor.

### 2.3.1 Einzelbild-Animation

Wie der Name sagt, wird für eine Einzelbild- oder auch Bild-für-Bild-Animation für jede Phase der Animation ein Bild benötigt. Diese werden nacheinander angeordnet und wie bei einem Daumenkino abgespielt. Im Beispiel animieren wir eine Verkehrsampel:

#### Einzelbild-Animation

Einzelne Bilder werden wie beim „Daumenkino“ nacheinander gezeigt.

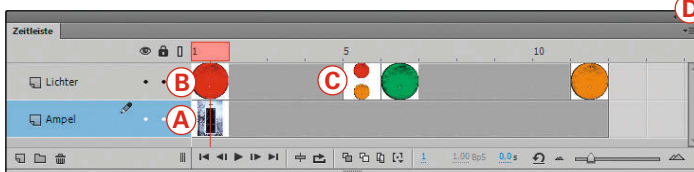


#### Making of ...

- 1 Erstellen Sie in Photoshop die Bilder für die Animation. Im Beispiel wurden vier Ebenen erstellt: Ampel, gelbes, grünes und rotes Licht.
- 2 Erstellen Sie in Animate ein neues HTML5-Canvas-Projekt. Passen Sie die Bühnengröße der Bildgröße Ihrer Photoshop-Datei an.

#### Zeitleiste

Eine Vorschau auf den Inhalt der Schlüsselbilder erhalten Sie durch Anklicken der Schaltfläche oben rechts **D**.



- 3 Importieren Sie die Photoshop-Datei in die Animate-Bibliothek: *Datei > Importieren > In Bibliothek importieren*.
- 4 Platzieren Sie den Hintergrund (hier: schwarze Ampel) im Schlüsselbild der Ebene 1 **A**.
- 5 Erstellen Sie eine neue Ebene. Platzieren Sie die erste Phase (hier: rote Ampel) im Schlüsselbild in Bild 1 **B**.
- 6 Stellen Sie im Menü *Modifizieren > Dokument* die Bild- oder Frame-rate der Animation ein (hier auf 1 BpS = 1 Bild pro Sekunde). Aus der Bildrate und der gewünschten Dauer der Animation lässt sich berechnen, wo das nächste Schlüsselbild platziert werden muss:

#### Zusammenhang Bilder und Anzeigedauer

$$B = t \times \text{BpS}$$

- t: Dauer einer Phase in Sekunden  
BpS: Bild- oder Framerate (Bilder pro Sekunde)  
B: Erforderliche Anzahl an Bildern (frames)

Beispiel: Phase 1 (rote Ampel) soll fünf Sekunden lang sichtbar sein. Die Bildrate wurde auf 1 BpS gestellt. Für fünf Sekunden werden  $B = 5\text{ s} \cdot 1\text{ BpS} = 5$  Bilder benötigt, das nächste Schlüsselbild muss also in Bild 6 (rot-gelbe Ampel) platziert werden **C**.

- 7 Platzieren Sie die weiteren Schlüsselbilder und tauschen Sie die Bildteile aus, die sich verändern sollen.
- 8 Mit der Return-Taste können Sie die Animation testen.

### 2.3.2 Schlüsselbild-Animation

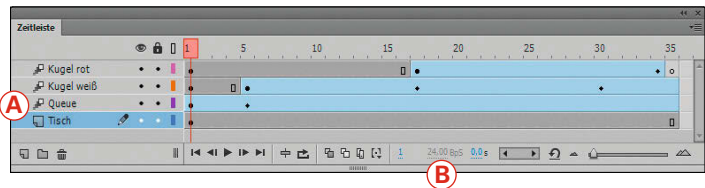
Einzelbild-Animationen sind aufwändig, da für jede Phase der Bewegung einzelne Bilder benötigt werden.

Die grundsätzliche Idee einer Schlüsselbild-Animation besteht darin, für das zu animierende Objekt lediglich die Startposition und die Endposition zu definieren. Die Animation entsteht dadurch, dass der Computer die Zwischenbilder zwischen Start- und Endposition beim Abspielen berechnet.

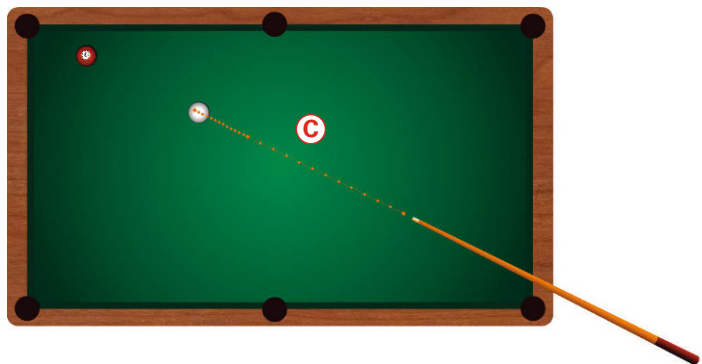
Bei Animate wird diese Animationstechnik als *Tweening* bezeichnet. Der Begriff leitet sich vom Englischen „between“ (dt.: dazwischen) ab. Im Beispiel animieren wir ein Billardspiel.

#### Making of ...

- 1 Erstellen Sie alle benötigten Objekte (hier: Tisch, Queue, weiße und rote Kugel) direkt in Animate oder importieren Sie sie als externe Dateien in die Bibliothek.
- 2 Platzieren Sie die Objekte auf der Bühne und richten Sie sie wie gewünscht aus.
- 3 Konvertieren Sie alle Objekte, die Sie animieren wollen, in Movieclip-Symbole: Menü *Modifizieren* > *In Symbol konvertieren*. Im Beispiel sind dies die beiden Billardkugeln und der Queue.
- 4 Klicken Sie auf das Symbol, das Sie zuerst animieren wollen (hier: Queue). Wählen Sie im Menü *Modifizieren* > *Bewegungs-Tween*. (Alternative: Rechtsklicken Sie auf das Symbol und wählen Sie *Bewegungs-Tween erstellen*).



- 5 Die Zeitleiste ändert sich nun folgendermaßen: Animate kopiert das zu animierende Symbol in eine neue Tween-Ebene. Sie erkennen diese am Icon **A** und sollten die Ebene sinnvoll benennen. Außerdem wird das Schlüsselbild um die in der Bildrate **B** eingestellte Anzahl Bilder verlängert und hellblau hinterlegt.
- 6 Die Dauer der Animation beträgt aktuell eine Sekunde – dies ist uns für den Stoß des Queues zu lang. Um die Animation zu verkürzen, ziehen Sie das letzte hellblaue Bild mit gedrückter Maustaste nach links (hier: auf Bild 5).
- 7 Verlängern Sie das Schlüsselbild der Hintergrund-Ebene. Im Beispiel ist der Tisch bis Bild 35 sichtbar. Klicken Sie auf Bild 35 der Ebene Tisch. Wählen Sie im Menü *Einfügen* > *Zeitleiste* > *Bild*, um das Schlüsselbild zu verlängern (alternativ: Taste F5).



Erstellen Sie Im nächsten Schritt die eigentliche Animation:

- 8 Klicken Sie auf die Ebene mit dem Animationsobjekt (hier: Queue).
- 9 Bewegen Sie den roten Abspielkopf auf das letzte hellblaue Bild (hier: auf Bild 5).
- 10 Verschieben Sie das Animationsobjekt an die gewünschte Endposition. Im Beispiel wurde der Queue bis zur weißen Kugel verschoben. Auf der Bühne wird der Animationspfad sichtbar. Jeder Punkt symbolisiert ein Bild.
- 11 Testen Sie Ihre Animation (Return-Taste). Nehmen Sie ggf. Änderungen der Start- oder Endposition des Animationsobjekts vor.
- 12 Erstellen Sie die nächste Animation – im Beispiel die weiße Kugel. Die Animation beginnt bei Bild 6 und endet bei Bild 30. Beachten Sie, dass die Kugel langsamer werden muss. Dies erreichen Sie, indem die Anzahl der Bilder gegen Ende zunimmt (vgl. Animationspfad **C** auf vorheriger Seite).
- 13 Verlängern Sie wie unter Punkt 7 beschrieben die Schlüsselbilder der übrigen Objekte (hier: Queue, Kugeln), damit sie die ganze Zeit sichtbar bleiben.
- 14 Die dritte Animation (hier: rote Kugel) verläuft von Bild 17 bis 34. Im letzten Bild (35) ist die Kugel nicht mehr sichtbar, sodass der Eindruck entsteht, dass die Kugel „eingelocht“ wurde.

### 2.3.3 Pfadanimation

Bisher haben Sie geradlinige (lineare) Animationen kennengelernt. Im nächsten Schritt erstellen wir eine Animation entlang einer beliebigen Kurve am Beispiel eines Ballwurfs.

#### Animationspfad ändern – Making of ...

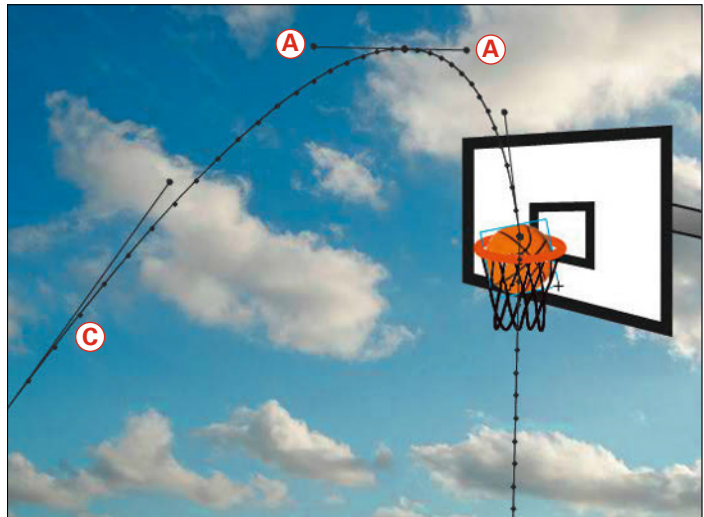
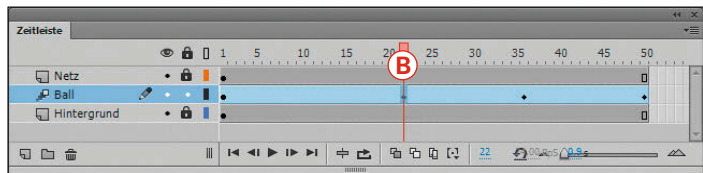
- 1 Erstellen Sie alle benötigten Objekte (hier: Ball, Korb, Hintergrund) und importieren Sie sie in Animate. Beachten Sie, dass der Korb aus zwei Teilen bestehen muss: Ein hinterer Teil mit der Rückwand und ein vorderer Teil mit dem Netz. Der Ball wird später auf einer Ebene dazwischen platziert, damit er scheinbar im Netz landet.
- 2 Konvertieren Sie den Ball in ein Symbol.
- 3 Erstellen Sie zunächst eine geradlinige Schlüsselbild-Animation von links außen zum Korb und vom Korb nach unten.

Um aus der geradlinigen Bewegung eine Kurve zu machen:

- 4 Wählen Sie das Auswahlwerkzeug (schwarzer Pfeil).
- 5 Ziehen Sie den Pfad mit gedrückter Maustaste.
- 6 Zur exakten Bearbeitung des Bewegungsablaufs können Sie die Schlüsselbilder mit dem Unterauswahl-Werkzeug (weißer Pfeil) anklicken. Danach erhalten Sie die im Screenshot gezeigten Anfasser **A**, die sich beliebig verschieben lassen.

Damit die Animation realistischer wirkt, muss der Ball beim Aufsteigen immer langsamer werden und beim Fallen beschleunigen.

- 7 Fügen Sie weitere Schlüsselbilder ein, indem Sie in der Tween-Ebene auf das gewünschte Bild klicken und im Menü *Einfügen > Zeitleiste > Schlüsselbild* wählen (alternativ: Taste F6) **B**.
- 8 Verschieben Sie das Schlüsselbild in der Tween-Ebene horizontal, um hierdurch die Anzahl an Zwischenbildern zu verändern. Je weniger Bilder sich zwischen zwei Schlüsselbildern befinden, umso schneller läuft die Bewegung ab. Sie erkennen dies am Animationspfad daran, dass der Abstand zwischen den Punkten (= Zwischenbildern) größer wird **C**.



Für das „Feintuning“ einer Animation stellt Animate zusätzlich einen – aus unserer Sicht wenig brauchbaren – Bewegungs-Editor zur Verfügung. Doppelklicken Sie auf die Tween-Ebene.

- 9 Experimentieren Sie mit den Möglichkeiten, die dieser Editor bietet – vielleicht kommen Sie damit zu guten Ergebnissen.
- 10 Ein weiterer Doppelklick auf die Tween-Ebene beendet den Editor.

Eine alternative Vorgehensweise zur Erstellung eines Animationspfads ist folgende:

- 11 Zeichnen Sie den gewünschten Pfad mit Hilfe eines Zeichenwerkzeugs in einer eigenen Ebene.

- 12 Kopieren Sie den Pfad in die Tween-Ebene und löschen Sie die Hilfsebene.

Neben dem Bewegungsverlauf entlang einer Kurve können Sie Änderungen an den Schlüsselbildern Ihrer Animation vornehmen. Im Beispiel animieren wir ein Bowlingspiel.

### Schlüsselbilder bearbeiten – Making of...

- 1 Erstellen Sie alle Objekte, die Sie für die Animation benötigen (hier: Bowlingbahn, Kugel, Kegel) und importieren Sie diese in die Bibliothek.
- 2 Platzieren Sie die Objekte auf der Bühne.

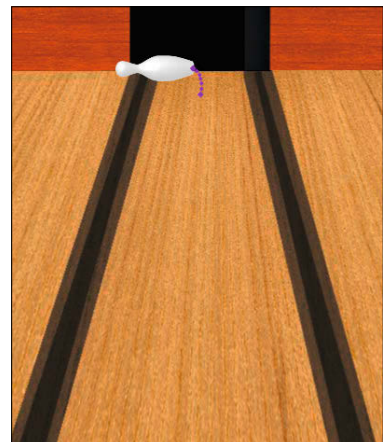
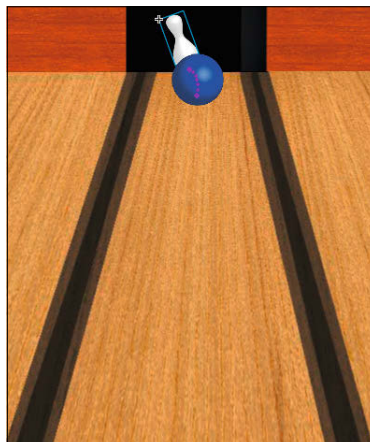
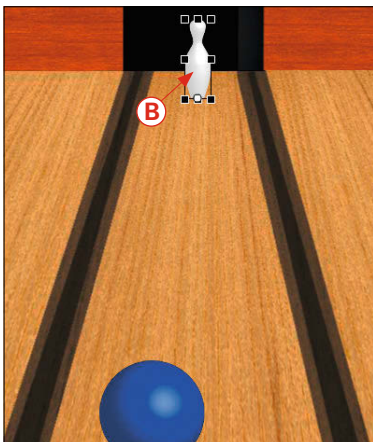


- 3 Animieren Sie zunächst den Kegel: Konvertieren Sie ihn in ein Symbol.
- 4 Wählen Sie das Frei-Transformieren-Werkzeug **A** und passen Sie die Größe des Kegels an.
- 5 Verschieben Sie den Drehpunkt **B** nach unten.
- 6 Bringen Sie einen Bewegungs-Tween am Kegel an (Rechtsklick: Bewegungs-Tween erstellen).
- 7 Klicken Sie auf das letzte Bild des hellblauen Tweens in der Zeitleiste. Drehen Sie den Kegel in eine horizontale Position.
- 8 Noch realistischer wirkt die Animation, wenn der Kegel zur Seite „springt“ und etwas kleiner wird.
- 9 Erstellen Sie ein Symbol der Bowlingkugel und bringen Sie einen Bewegungs-Tween an.
- 10 Verkleinern Sie die Kugel am Ende der Animation mit Hilfe des Frei-Transformieren-Werkzeugs.



Zusätzlich können Sie folgende Schlüsselbild-Eigenschaften anpassen:

- 11 Nach Anklicken eines Schlüsselbildes *in der Zeitleiste* können Sie das Animationsobjekt:
  - im oder gegen den Uhrzeigersinn mehrfach drehen lassen **C**,
  - beschleunigen (negative Zahlen) oder abbremsen (positive Zahlen) **D**.
- 12 Nach Anklicken eines Schlüsselbildes *auf der Bühne* können Sie
  - dessen Farbe,
  - Helligkeit oder
  - Transparenz (Alpha) ändern.

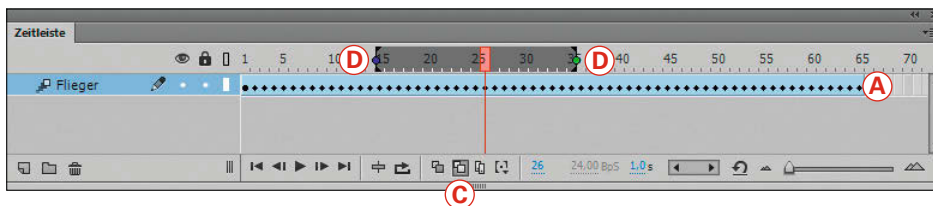
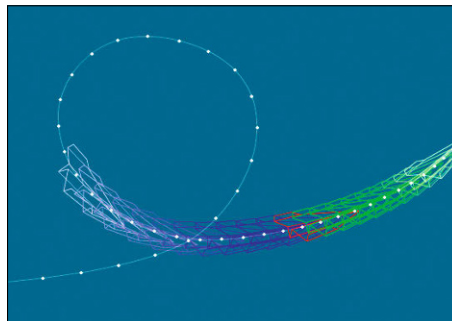
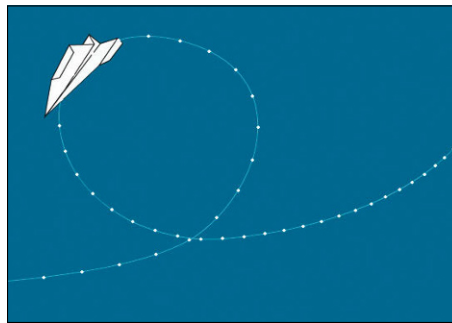
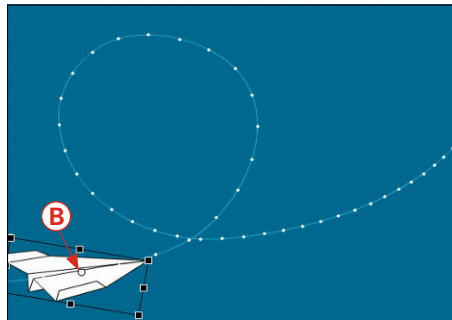




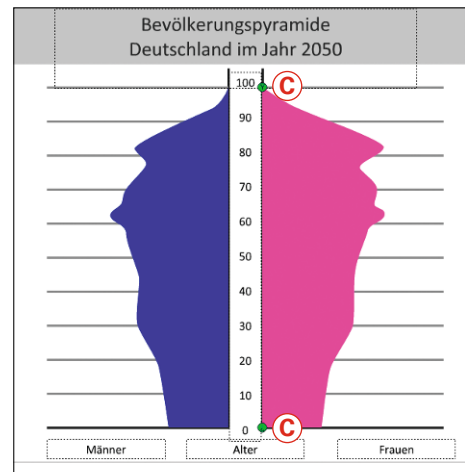
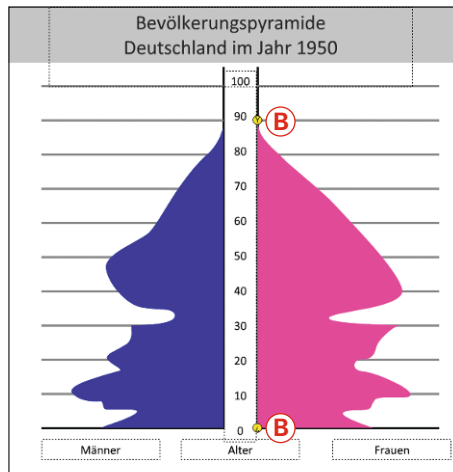
In manchen Fällen ist es erwünscht, dass ein animiertes Objekt immer in Richtung des Animationspfads zeigt:

### An Pfad ausrichten – Making of ...

- 1 Erstellen Sie das zu animierende Objekt (hier: Papierflieger) und importieren Sie es auf die Bühne.
- 2 Erstellen Sie ein Symbol. Bringen Sie einen Bewegungs-Tween an.
- 3 Erstellen Sie den Animationspfad. In diesem Fall bietet es sich an, den Pfad separat zu zeichnen und dann mit dem Tween zu verbinden (vgl. Punkt 11 und 12 auf Seite 33).
- 4 Wählen Sie nun die Option *An Pfad ausrichten* (E auf vorheriger Seite). Wie Sie sehen, wird in der Zeitleiste jedes Bild zum Schlüsselbild A.
- 5 Das animierte Objekt ist an seinem Drehpunkt B mit dem Pfad verbunden. Verschieben Sie diesen Punkt mit Hilfe des Frei-Transformieren-Werkzeugs, falls die Animation unrealistisch aussieht.
- 6 Um den Bewegungsablauf besser beurteilen zu können, lassen sich mehrere Bilder als sogenannte *Zwiebelschalen* bzw. Zwiebelschalenkonturen C einblenden. Die Anzahl an dargestellten Bildern wählen Sie mit Hilfe der Anfasser D in der Zeitleiste.







### 2.3.4 Morphing

Bei Morphing handelt es sich – wie auf Seite 10 beschrieben – um eine Animationstechnik, bei der sich ein Bild in ein anderes Bild umwandelt. Sie kennen diese Effekte aus Spielfilmen wie beispielsweise Terminator 2.

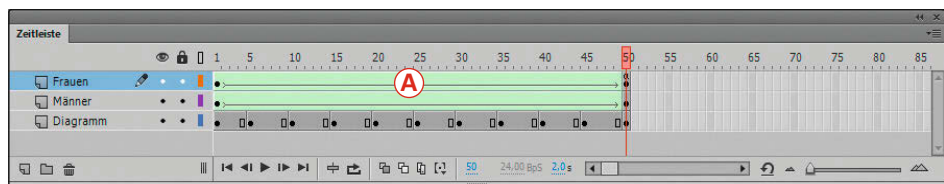
Adobe Animate ist *keine* Morphing-Software und spricht deshalb bescheidener von *Form-Tweening*. Die Umwandlung einer Form in eine andere funktioniert nur mit relativ einfachen Formen, z. B. Buchstaben oder geometrischen Objekten. Komplexe Grafiken oder gar Fotos lassen sich mit Animate nicht morphen!

#### Making of ...

- 1 Erstellen Sie ein erstes grafisches Objekt im ersten Schlüsselbild. Im

Beispiel ist dies die Bevölkerungspyramide im Jahr 1950. Konvertieren Sie das Objekt *nicht* in ein Symbol – Form-Tweens funktionieren nur mit grafischen Formen.

- 2 Fügen Sie in der Zeitleiste in einem zweiten Bild ein leeres Schlüsselbild ein. Erstellen Sie nun die Grafik, in die sich die erste Grafik umwandeln soll, im Beispiel die Bevölkerungspyramide im Jahr 2050.
- 3 Klicken Sie mit der rechten Maustaste zwischen die beiden Schlüsselbilder und wählen Sie *Form-Tween erstellen*. Ein Form-Tween wird in grüner Farbe dargestellt und ist durch einen schwarzen Pfeil gekennzeichnet **A**. Die Animation können Sie nun mit der Return-Taste testen.



Zur Verbesserung der Animation dienen *Formmarken*: Mit ihnen legen Sie fest, wie einzelne Stützpunkte von der Start- in die Zielgrafik überführt werden sollen:

- 4 Aktivieren Sie im Menü *Ansicht > Ausrichten > An Objekten ausrichten*.
- 5 Klicken Sie in Schlüsselbild 1 und wählen Sie im Menü *Modifizieren > Form > Formmarken einfügen*. Ziehen Sie die Formmarke auf die Kontur der Grafik, z. B. in die untere Ecke. Eine richtig platzierte Formmarke wird gelb dargestellt (**B** auf linker Seite).
- 6 Klicken Sie ins Schlüsselbild mit der Zielgrafik und platzieren Sie die Formmarke, z. B. ebenfalls in die untere Ecke (**C** auf linker Seite). Eine richtig platzierte Formmarke erscheint in der Zielgrafik grün.
- 7 Wiederholen Sie die Schritte 2 und 3, um weitere Formmarken einzufügen und zu platzieren. Testen Sie zwischendurch Ihre Animation.

das sichtbare Objekt verändert wird. Im Beispiel wird ein Text nach und nach aufgedeckt werden, so dass der Betrachter den Eindruck bekommt, dass der Text geschrieben wird.

### Making of...

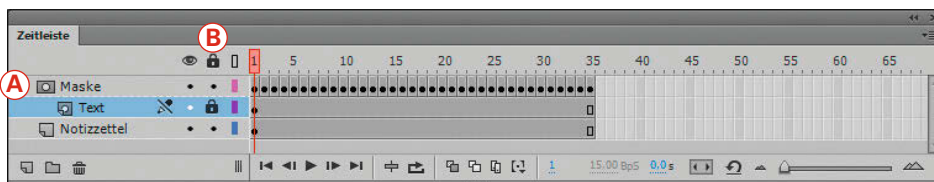
- 1 Erstellen Sie einen kurzen Text z. B. in einer Schreibschrift.
- 2 Erzeugen Sie eine neue Ebene oberhalb der Ebene mit dem Text. Rechtsklicken Sie auf diese Ebene und wählen Sie die Option *Maske*. Sie erkennen die Maskenebene am geänderten Icon **A**.

Die Maskenebene überdeckt die darunterliegende Ebene komplett. Wenn Sie beispielsweise mit dem Pinsel-Werkzeug auf dieser Ebene zeichnen, wird die Maske an dieser Stelle durchsichtig.

- 3 Entsperren Sie die Ebene durch Anklicken des Schloss-Icons **B**.
- 4 Erzeugen Sie in Bild 2 der Maskenebene ein Schlüsselbild (Menü *Einfügen > Zeitleiste > Schlüsselbild*).
- 5 Übermalen Sie mit dem Pinsel einen Teil des ersten Buchstabens, und zwar so, wie Sie ihn von Hand schreiben würden (siehe Screen-shot auf der nächsten Seite).
- 6 Fügen Sie in Bild 3 der Maskenebene das nächste Schlüsselbild ein

## 2.3.5 Masken

Animationen lassen sich auch indirekt erzeugen, indem das zu animierende Objekt mit Hilfe einer Maske ganz oder teilweise überdeckt wird. Durch Änderung der Maske, z. B. Verformung oder Bewegung, entsteht der Eindruck, dass





und übermalen Sie einen weiteren Teil des Buchstabens.

- 7 Wiederholen Sie den Vorgang für den gesamten Text.

- 8 Um die Animation zu testen, sperren Sie die Maskenebene (Schloss-Icon). Spielen Sie die Animation mit der Return-Taste ab.

- 9 Zur weiteren Bearbeitung müssen Sie die Maskenebene durch Anklicken des Schloss-Icons zunächst wieder entsperren.

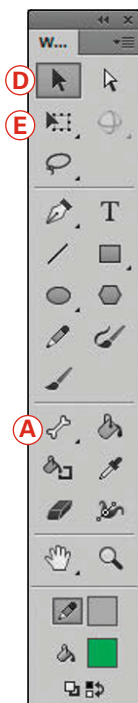
### 2.3.6 Inverse Kinematik

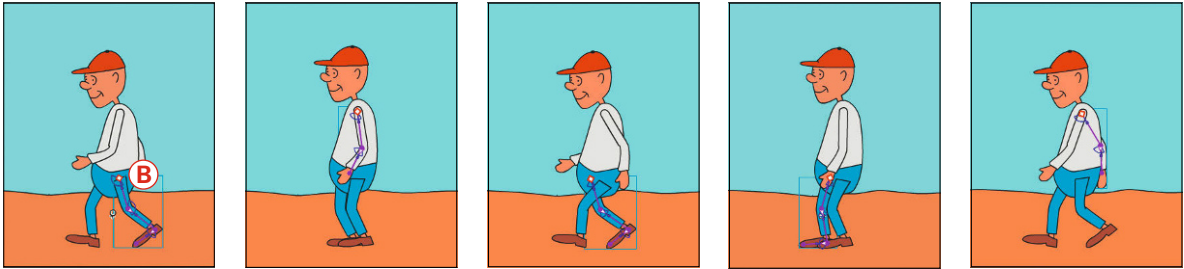
Der Begriff inverse Kinematik oder auch Rückwärtskinematik entstammt der Robotik und bedeutet, dass ein Roboter in die gewünschte Endposition gebracht wird. Aus dieser Position wird er umgekehrt (invers) zurück in die Startposition bewegt, um hierdurch die zur Durchführung der (Vorwärts-)Bewegung erforderliche Drehung und Positionsänderung der Gelenke zu ermitteln. Auch bei Animate-Objekten, wie z. B.

einer Zeichentrickfigur, lassen sich Gelenke definieren, die über *Bones* (dt.: Knochen) verbunden sind. Auf diese Weise können Sie realistisch wirkende Bewegungsabläufe erstellen.

#### Making of ...

- 1 Zeichnen oder importieren Sie das zu animierende Objekt. Zum Testen der inversen Kinematik ist es empfehlenswert, mit einfachen Objekten zu beginnen, z. B. mit zwei Stangen, die über ein Gelenk verbunden werden sollen.
- 2 Damit eine Grafik mit Bones versehen werden kann, muss sie als Form vorliegen. Importierte Objekte oder Symbole müssen im Menü *Modifizieren > Teilen* in Formen umgewandelt werden. Das Objekt erscheint dann punktiert.
- 3 Wählen Sie das Bone-Werkzeug **A**. Beginnen Sie das Zeichnen der Bones an der Stelle, die unveränderlich bleiben soll: Bei einem Arm wäre dies das Schultergelenk,





bei einem Bein das Hüftgelenk **B**. Ziehen Sie den Bone mit gedrückter Maustaste bis zum nächsten Gelenk. Wiederholen Sie den Vorgang, um weitere Gelenke einzufügen. Animate erzeugt automatisch eine Posenebene **C** und fügt das „Skelett“ in diese Ebene ein. Die ursprüngliche Ebene können Sie löschen.

- 4 Testen Sie die inverse Kinematik, indem Sie das Auswahlwerkzeug **D** (auf linker Seite) wählen. Die einzelnen Gelenke können Sie nun anklicken und bewegen.

Animate bietet zahlreiche Möglichkeiten zur Nachbearbeitung der Bones.

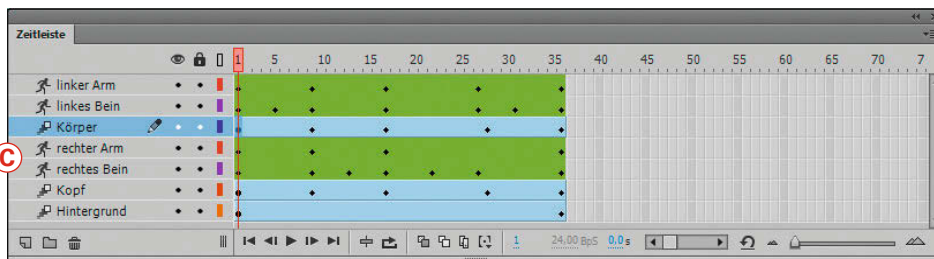
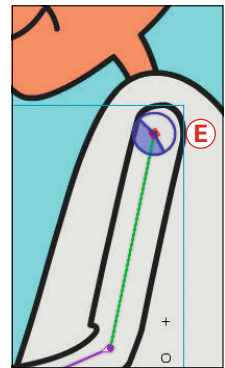
- 5 Klicken Sie auf einen Bone – er wird hellgrün dargestellt.
- 6 Klicken Sie auf das zugeordnete Gelenk – es wird als Kreis mit Doppelpfeil dargestellt.

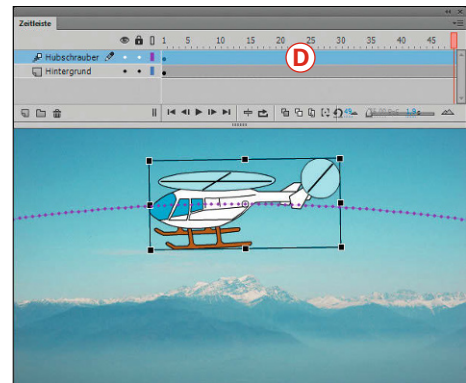
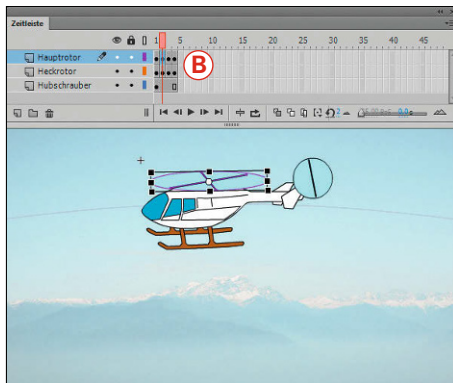
- 7 Klicken Sie auf den Kreisring, um die Drehwinkel festzulegen. Ein Arm lässt sich beispielsweise nur etwa 45° nach hinten, aber fast 180° nach vorne drehen **E**. (Alternativ können Sie diese Einstellungen auch numerisch im Eigenschaften-Fenster vornehmen.)

- 8 Wiederholen Sie die Schritte 5 bis 7 für die anderen Gelenke.

Um eine Animation zu erstellen:

- 9 Bringen Sie die Bones und Gelenke in Bild 1 der Posenebene in die gewünschte Startposition.
- 10 Rechtsklicken Sie in der Posenebene auf das Bild, bei dem die Animation enden soll und wählen Sie *Pose einfügen*.
- 11 Bringen Sie die Bones und Gelenke in die gewünschte Endposition. Testen Sie mit der Return-Taste.





### 2.3.7 Verschachtelte Animationen

Bislang haben wir unsere Animationen in der (Haupt-)Zeitleiste des Films erstellt. Dies ist für einfache Animationen auch ausreichend. Komplexere Animationen erfordern jedoch eine Verschachtelung mehrerer Bewegungen: Ein Vogel bewegt sich vorwärts, während er mit den Flügeln schlägt. Die Figur im letzten Beispiel bewegt Arme, Beine, Oberkörper und Kopf, während sie sich vorwärts bewegt.

Damit dies möglich wird, lassen sich (innere) Animationen auch direkt *in den Zeitleisten der Symbole* erstellen. Das Beispiel zeigt einen Hubschrauber, der sich von rechts nach links über den Hintergrund bewegt (äußere Animation). Die innere Animation simuliert die Drehung der Rotorblätter.

#### Making of ...

- 1 Erstellen Sie das Animationsobjekt in Animate oder importieren Sie es.
- 2 Konvertieren Sie das Objekt in ein Movieclip-Symbol: *Modifizieren > In Symbol konvertieren* (Taste F8).

Erstellen Sie die innere Animation:



- 3 Doppelklicken Sie auf das Symbol (hier: Hubschrauber). Sie befinden sich nun im Symbol **A**, das eine eigene Zeitleiste besitzt.
- 4 Nehmen Sie die gewünschte innere Animation vor. Im Beispiel wurde die Drehung der Hubschrauber-Rotoren mit Hilfe einer Bild-für-Bild-Animation **B** realisiert.
- 5 Kehren Sie durch Anklicken des Pfeils **C** zur Hauptzeitleiste zurück.

Erstellen Sie die äußere Animation:

- 6 Erstellen Sie einen Bewegungstween **D**, so dass sich das Objekt über die Bühne bewegt.
- 7 Mit der Return-Taste lässt sich nur die äußere Animation testen. Um die komplette Animation zu sehen, wählen Sie im Menü *Steuerung > Testen* oder die Tastenkombination Strg + Return-Taste.

### 2.4.1 Skriptsprachen

Wenn Sie das Animate-Kapitel bis hier durchgearbeitet haben, dann können Sie jetzt Projekte erstellen, die Text, Bilder, Sound, Video und selbstablaufende Animationen enthalten. Damit haben Sie die Grenze dessen erreicht, was *ohne* Programmierung möglich ist.

Programmierung ist immer dann erforderlich, wenn der Nutzer die Möglichkeit erhalten soll, in die Anwendung einzugreifen, um beispielsweise

- eine Anwendung über Buttons oder die Tastatur zu steuern,
- Text zu ändern oder zu erzeugen,
- auf externe Dateien zuzugreifen, z. B. auf Bilder,
- Sound zu steuern,
- Objekte mit der Maus verschiebbar zu machen.

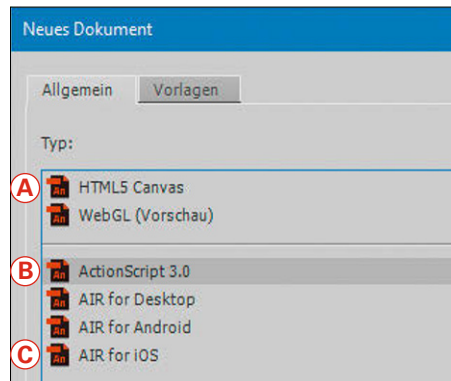
Animate bietet zwei unterschiedliche Möglichkeiten, um Anwendungen mit Interaktivität zu versehen.

#### JavaScript

Wenn Sie Ihr Projekt als *HTML5 Canvas* **A** anlegen, dann können Sie ein Animate-Projekt ohne Flash-Player in jedem Browser abspielen. Damit dies möglich wird, erzeugt Animate bei der Veröffentlichung des Projekts (siehe Seite 29) eine HTML5- und eine JavaScript-Datei. In HTML5-Canvas-Projekten können Sie aus diesem Grund direkt in JavaScript programmieren. Wir gehen in dieser Einführung nicht darauf ein.

#### ActionScript 3.0

ActionScript ist die mächtige Skriptsprache des Animate-Vorgängers Flash. In Animate können Sie diese Sprache weiterhin benutzen, wenn Sie Ihr Projekt als *ActionScript 3.0* **B** anlegen. Für die Ausführung von ActionScript ist der Flash-Player erforderlich, der in



allen aktuellen Browsern integriert ist. Einzige Ausnahme stellt das Betriebssystem iOS dar. Für Abhilfe sorgt hier AIR (Adobe Integrated Runtime) **C**, eine sogenannte Laufzeitumgebung, die das Abspielen von Animate-Projekten *ohne* Flash-Player ermöglicht.

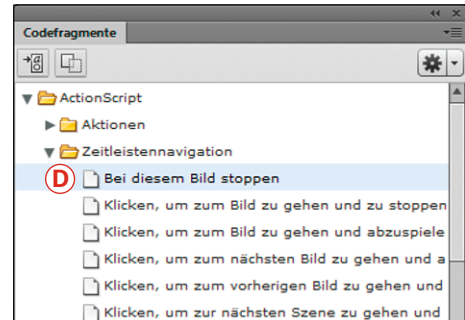
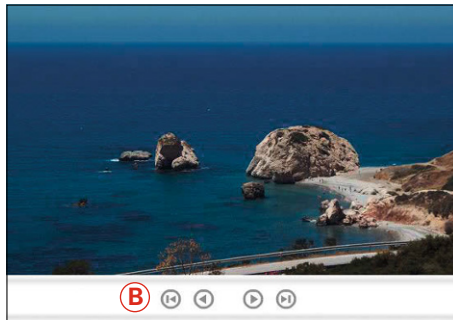
Wenn Sie bereits über Programmierkenntnisse verfügen, wird Ihnen das Programmieren in ActionScript leichtfallen. Andernfalls müssen Sie sich zunächst an die Struktur und Punktnotation dieser objektorientierten Skriptsprache gewöhnen. Animate unterstützt Sie jedoch hierbei, da Sie auf eine Bibliothek mit vorprogrammierten Funktionen zurückgreifen können.

### 2.4.2 Zeitleiste steuern

Ohne Steuerung durch ein Skript läuft ein Animate-Projekt von Anfang bis Ende ab und wiederholt sich dann, wenn nichts anderes eingestellt.

In diesem Abschnitt lernen Sie, wie Sie eine Animation stoppen und Buttons programmieren können, damit ein Hin- und Herspringen in verschiedene Bilder der Zeitleiste möglich wird. Als Beispiel realisieren wir eine kleine Diashow.





### Making of ...

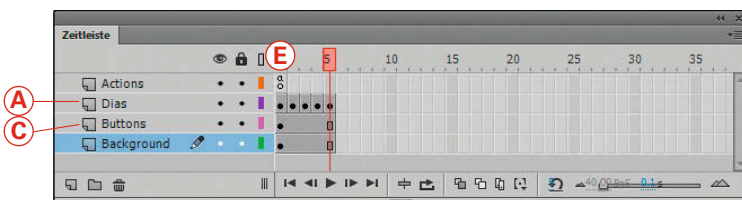
- 1 Erstellen Sie eine neue Datei. Wählen Sie *ActionScript 3.0*.
- 2 Bereiten Sie in Photoshop einige Fotos in passender Größe vor und importieren Sie diese in die Animate-Bibliothek.
- 3 Platzieren Sie die Bilder in Bild 1, 2, 3, 4, 5... der Zeitleiste **A**.
- 4 Fertigen Sie vier Buttons an (Vor, Zurück, Ende, Anfang) **B**. Konvertieren Sie diese in Schaltflächen-Symbole, wie auf Seite 22 beschrieben.
- 5 Erstellen Sie eine neue Ebene für die Buttons **C** und platzieren Sie die Buttons an der gewünschten Position auf der Bühne.

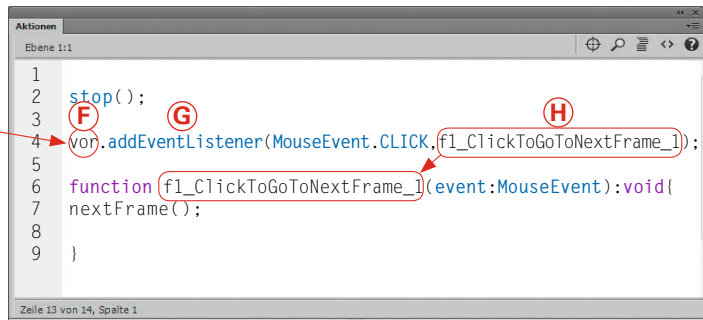
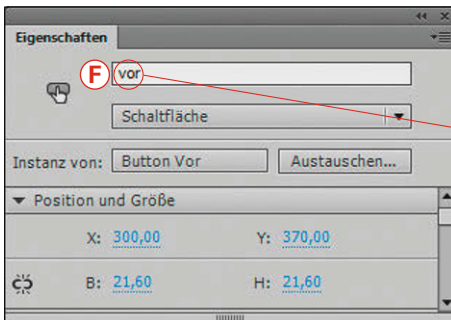
Nun erfolgt die Programmierung. Zunächst wird dafür gesorgt, dass die Animation in Bild 1 der Zeitleiste stoppt.

- 6 Platzieren Sie den Abspielkopf in Bild 1 der Zeitleiste.
- 7 Doppelklicken Sie im Menü *Fenster > Codefragmente* unter *ActionScript > Zeitleistennavigation* auf den Befehl *Bei diesem Bild stoppen* **D**. Es öffnet sich das Aktionen-Fenster und das zugehörige Skript wird angezeigt. In diesem Fall handelt es sich um den Befehl `stop()`. Die restlichen Zeilen sind Kommentare, die Sie wahlweise löschen können.
- 8 Animate hat in der Zeitleiste eine neue Ebene *Actions* erzeugt und das Skript dort in Bild 1 eingefügt, wie Sie am  $\alpha$ -Zeichen **E** erkennen.

Im nächsten Schritt erfolgt die Programmierung der vier Buttons:

- 9 Um Schaltflächen per Skript anzusprechen zu können, muss ihnen ein eindeutiger *Instanzname* gegeben werden. Wichtig: Der Name hat nichts mit dem Namen des Schaltflächen-Symbols in der Bibliothek zu tun. Klicken Sie die Schaltfläche (hier: Vorwärts-Button) auf der Bühne an und geben Sie ihr in den Eigenschaften einen Namen **F**, z. B. *vor*. Umlaute, Leer- und Sonderzeichen sind nicht gestattet!





10 Klicken Sie auf den Vorwärts-Button.

11 Doppelklicken Sie im Menü *Fenster* > *Codefragmente* auf den Befehl *Klicken, um zum nächsten Bild zu gehen und anzuhalten*. Das Skript bedarf einiger Erklärung:

#### Zeile 4:

Dem Vorwärts-Button mit Namen *vor* **F** wurde, durch einen Punkt getrennt, ein sogenannter Event-Listener **G** angefügt. Wie der Name sagt, wartet bzw. hört (to listen, dt.: hören) der Button nun auf ein Event (dt.: Ereignis). Die runde Klammer gehört zum Event-Listener dazu und liefert zwei Informationen: Der erste Parameter bewirkt, dass auf ein Maus-Ereignis (MouseEvent) gewartet wird, genauer gesagt auf ein Mausklick (CLICK). Der zweite Parameter gibt den Namen der Funktion **H** an, die durch das Maus-Ereignis aufgerufen wird. Den von Animate vergebenen Namen (hier: *f1\_ClickToGoToNextFrame\_1*) können Sie ändern, z. B. in *naechstesBild*. Umlaute und Sonderzeichen sind nicht gestattet.

#### Zeilen 6 bis 9:

Die Zeilen definieren die oben erwähnte *function*. Wenn Sie

den Namen in Zeile 4 umbenannt haben, müssen Sie ihn hier ebenfalls umbenennen. In der Klammer wird der Funktion der Parameter *MouseEvent* übergeben, *void* (dt.: Leere, Lücke) besagt, dass sie keinen Wert zurückgibt. *nextFrame()* in Zeile 7 bewirkt, dass sich der Abspielkopf um genau ein Bild weiterbewegt. Dies ist also die eigentliche Funktion des Buttons.

12 Wiederholen Sie die Schritte 9 bis 11 für den Zurück-Button. Vergeben Sie ihm einen eindeutigen Namen, z.B. *zurueck*. Doppelklicken Sie im Menü *Fenster* > *Codefragmente* auf den Befehl *Klicken, um zum vorherigen Bild zu gehen und anzuhalten*. Das Skript enthält die Funktion *prevFrame()*, wobei *prev* für *previous* (dt.: vorherig) steht.

13 Für den Zum-Anfang- bzw. Zum-Ende-Button benötigen wir das Codefragment *Klicken, um zum Bild zu gehen und zu stoppen*. In ActionScript heißt diese Funktion *gotoAndStop(...)*. Setzen Sie in der Klammer die gewünschte Bildnummer ein, also z. B. *gotoAndStop(1)*, um in Bild 1 der Zeitleiste zu navigieren.

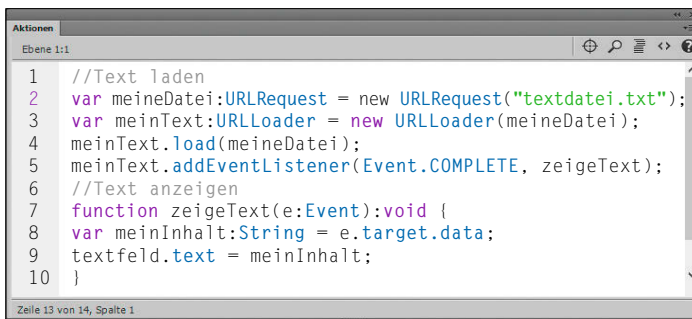
## 2.4.3 Dynamischer Text

Das manuelle Einfügen von Texten in ein Animate-Projekt ist nur bei geringer Textmenge sinnvoll. Abgesehen vom hohen Aufwand besteht der Nachteil, dass die Ausgabedatei bei jeder Textänderung neu erstellt werden muss.

Die Idee liegt nahe, Texte in externen Dateien oder einer Datenbank zu speichern und sie bei Bedarf an der gewünschten Stelle zu platzieren. Auf diese Weise können sie jederzeit bearbeitet werden, ohne dass der Nutzer etwas von Animate verstehen muss.

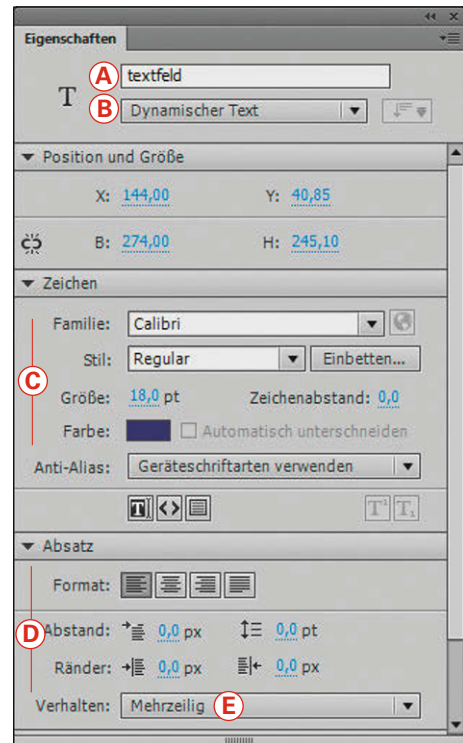
### Text laden – Making of ...

- 1 Starten Sie einen Texteditor wie *Editor* oder *TextEdit* und geben Sie einen kurzen Text ein. (Verwenden Sie kein Textverarbeitungsprogramm wie Word, weil dort viele Zusatzinformationen gespeichert werden.)
- 2 Speichern Sie den Text, z. B. unter dem Namen *textdatei.txt*, ab.
- 3 Öffnen Sie ein neues Animate-Projekt. Wählen Sie *ActionScript 3.0*.
- 4 Ziehen Sie auf der Bühne ein Textfeld auf.



```
1 //Text laden
2 var meineDatei:URLRequest = new URLRequest("textdatei.txt");
3 var meinText:URLLoader = new URLLoader(meineDatei);
4 meinText.load(meineDatei);
5 meinText.addEventListener(Event.COMPLETE, zeigeText);
6 //Text anzeigen
7 function zeigeText(e:Event):void {
8     var meinInhalt:String = e.target.data;
9     textfeld.text = meinInhalt;
10 }
```

Zeile 13 von 14, Spalte 1



- 5 Vergeben Sie der Textfeld-Instanz einen eindeutigen Namen (hier: *textfeld* **A**), damit es per ActionScript angesprochen werden kann.
- 6 Wichtig: Wählen Sie die Option *Dynamischer Text* **B**.
- 7 Wählen Sie die gewünschten Eigenschaften des Textfelds, z. B.:
  - Schriftart, -größe, -farbe **C**
  - Satzart, Abstände, Ränder **D**
  - mehrzeiliger Text **E**
- 8 Öffnen Sie – falls geschlossen – im Menü *Fenster > Aktionen*. Geben Sie das im Screenshot links dargestellte ActionScript ein – achten Sie auch auf Groß- und Kleinschreibung.

**Zeile 2:**

Diese Schreibweise ist für objekt-orientierte Sprachen typisch – für Anfänger sehr verwirrend. Mit `var` wird eine Variable mit frei wählbarem Namen (hier: `meineDatei`) definiert. Dieser Variablen wird mit `new` ein Objekt der Klasse `URLRequest` zugewiesen. Dieses Objekt fordert eine URL (für Uniform Resource Locator) an (request, dt.: Anforderung). Bei einer URL handelt es sich allgemein um eine Internetadresse, in diesem Fall um eine Textdatei (hier: `textdatei.txt`).

**Zeile 3/4:**

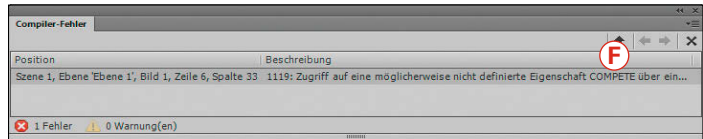
Sie benötigen eine zweite Variable (hier: `meinText`) und ein zweites Objekt, dieses Mal jedoch von der Klasse `URLLoader`. Dieses bewirkt, dass der Inhalt der angeforderten Datei auch tatsächlich geladen werden kann.

**Zeile 5:**

Event-Listener kennen Sie bereits aus dem letzten Abschnitt: Er wartet auf das Eintreten eines Ereignisses und ruft danach eine Funktion (hier: `zeigeText`) auf. Wie der Name andeutet, tritt das Ereignis ein, wenn die Datei komplett (COMPLETE) geladen ist.

**Zeilen 7 bis 10:**

In der Funktion wird eine Textvariable (String) mit Namen `meinInhalt` definiert. Ihr wird der Inhalt der Textdatei über `e.target.data` zugewiesen. Zeile 9 bewirkt schließlich, dass der Inhalt der Variablen im Textfeld angezeigt wird. Dabei ist `textfeld` der Instanzname, den Sie dem dynamischen Textfeld zugewiesen haben **A**.



- 9 Beim Testen (Menü *Steuerung > Testen*) müsste der Text im Textfeld auf der Bühne angezeigt werden. Geschafft? Wenn nicht, dann lesen Sie die Fehlermeldung des Compilers. Im Beispiel ist zu erkennen, dass in der Eigenschaft `COMPLETE` ein „L“ fehlt **F**.

Nun werden Sie zurecht sagen, dass dies furchtbar umständlich ist. Die Verwendung von vordefinierten Klassen, bietet jedoch den Vorteil, dass Sie auf die Methoden und Eigenschaften der Klasse mit Hilfe von Objekten zugreifen können. Hierdurch reduziert sich der Programmieraufwand erheblich.

Im letzten Beispiel wurde das Textfeld manuell erstellt und die Schrift von Hand formatiert. Eine elegantere Lösung ist es, auch das Textfeld mit Hilfe von ActionScript 3.0 zu erzeugen.

**Textfeld generieren – Making of ...**

- 1 Öffnen Sie ein neues Animate-Projekt. Wählen Sie *ActionScript 3.0*.
- 2 Geben Sie in Bild 1 das im Screenshot auf der nächsten Seite dargestellte ActionScript ein.

**Zeile 2:**

Hier wird einer Variablen (hier: `meinFeld`) ein Objekt der Klasse `TextField` zugewiesen.

**Zeilen 3 bis 8:**

Dem Textfeld-Objekt werden folgende Eigenschaften zugeordnet:

```

Aktionen
ActionScript:1
1 //Textfeld erzeugen
2 var meinFeld:TextField = new TextField();
3 meinFeld.x = 0;
4 meinFeld.y = 0;
5 meinFeld.width = 300;
6 meinFeld.height = 100;
7 meinFeld.wordWrap = true;
8 addChild(meinFeld);
9
10 //Text formatieren
11 var meinFormat:TextFormat = new TextFormat();
12 meinFormat.font = „Calibri“;
13 meinFormat.size = 20;
14 meinFeld.defaultTextFormat = meinFormat;
15 meinFeld.text = „Hallo Welt“;

```

Zeile 15 von 38, Spalte 1

Seine xy-Position auf der Bühne, gemessen von der linken oberen Ecke, die Breite und Höhe des Textfeldes und die Angabe, dass Zeilenumbruch (wordwrap) möglich sein soll. Zeile 8 sorgt dafür, dass das Textfeld erzeugt wird.

**Zeile 11:**  
Zur Formatierung der Schrift im Textfeld dient eine Variable (hier:

meinFormat), der ein Objekt der Klasse TextFormat zugeordnet wird.

**Zeilen 12 bis 14:**

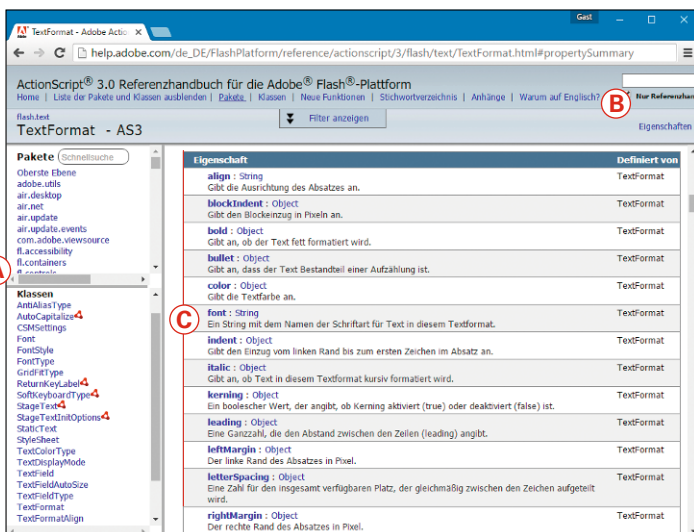
Der Variablen werden zwei Eigenschaften zugewiesen: Schriftart und -größe. In Zeile 14 werden diese Eigenschaften auf das Textfeld übertragen.

**Zeile 15:**

In dieser Zeile wird der Text „Hallo Welt“ in das Textfeld eingetragen. Alternativ können Sie hier das Skript einfügen, das Sie im vorherigen Abschnitt zum Laden einer externen Textdatei erstellt haben.

## ActionScript-Referenzhandbuch

In diesem Kapitel haben Sie Klassen kennengelernt, die zahlreiche *Eigenschaften* (engl.: properties) und *Methoden* (engl.: methods) besitzen. Natürlich ist es unmöglich, sich dies alles merken zu wollen. Abhilfe schafft hier das ActionScript-Referenzhandbuch, das Sie im Internet unter [http://help.adobe.com/de\\_DE/FlashPlatform/reference/actionscript/3/](http://help.adobe.com/de_DE/FlashPlatform/reference/actionscript/3/) aufrufen können.



## Making of ...

- 1 Wählen Sie die gesuchte Klasse **A**.
- 2 Klicken Sie auf Eigenschaften **B**.
- 3 Durchsuchen Sie die Eigenschaften dieser Klasse **C**.

Beachten Sie:

Klassennamen, z.B. TextFormat, werden großgeschrieben, Eigenschaften, z.B. color, kleingeschrieben. Weitere, angehängte Begriffe werden großgeschrieben, z. B. letterSpacing.

## 2.4.4 Dynamische Bilder

Oft ist es wünschenswert, dass Bilder aktualisiert werden können. Für diesen Zweck ist es sinnvoll, die Bilder in externen Dateien zu belassen.

In dieser Übung erstellen wir eine Bildergalerie aus Vorschaubildern (Thumbnails). Die großen Bilddateien liegen als externe Dateien vor und werden erst geladen, wenn der Nutzer das Thumbnail mit der Maus berührt.

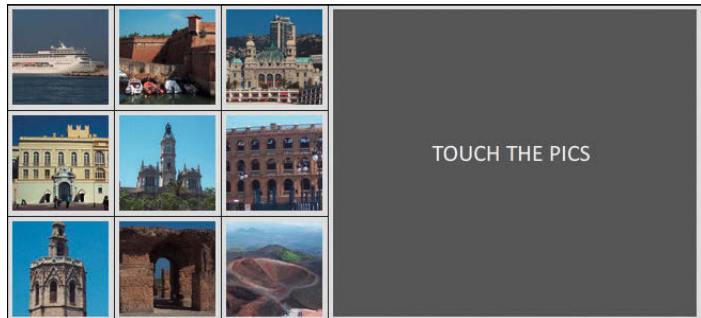
### Bilder laden – Making of ...

- 1 Öffnen Sie ein neues Animate-Projekt. Wählen Sie *ActionScript 3.0*.
- 2 Bereiten Sie Ihre Bilder in Photoshop vor und speichern Sie sie in zwei Varianten ab: als Thumbnail und in hoher Auflösung.
- 3 Importieren Sie nur die Thumbnails und platzieren Sie diese auf der Bühne.
- 4 Konvertieren Sie die Thumbnails in Schaltflächen. Geben Sie den Symbolen einen eindeutigen Instanznamen, z.B. *bild1*, *bild2* usw.

Öffnen Sie im Menü *Fenster > Aktionen* und geben Sie das rechts unten dargestellte ActionScript ein.

#### Zeile 3:

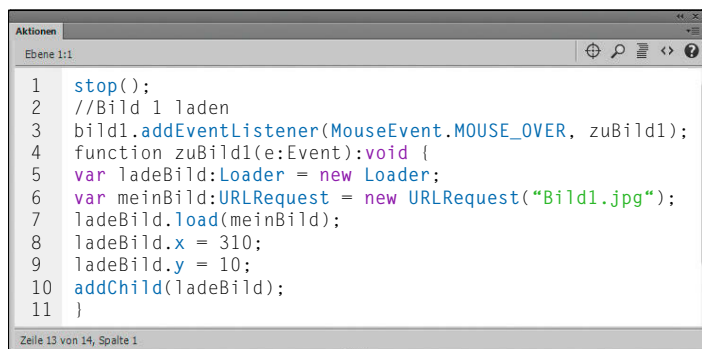
An die Instanz des ersten Bildes wird ein Event-Listener angebracht. Er wartet in diesem Fall auf das Ereignis `MOUSE_OVER`: Es tritt ein, wenn der Mauszeiger das Bild berührt – ein Mausklick (`CLICK`) ist nicht erforderlich. Die hierdurch aufgerufene Funktion heißt hier `zuBild1`.



#### Zeilen 4 bis 10:

Zunächst wird eine Variable der Klasse `URLRequest` benötigt (hier: `meinBild`), das den Dateinamen (hier: `Bild1.jpg`) erhält. Das Laden des Bildes ermöglicht ein Objekt der Loader-Klasse (hier: `ladeBild`). In Zeile 8 und 9 wird die Position der linken oberen Ecke festgelegt. Das Platzieren des Bildes auf der Bühne erfolgt schließlich in Zeile 10.

- 5 Wenn Sie das Skript für das erste Bild erstellt und getestet haben, lässt es sich für die weiteren Bilder durch „Copy & Paste“ zügig erweitern: Kopieren Sie die Zeilen 3 bis 11 und fügen Sie die Zeilen mehrfach ein. Ändern Sie danach lediglich die Nummerierung: `bild2`, `zuBild2`, `Bild2.jpg`.





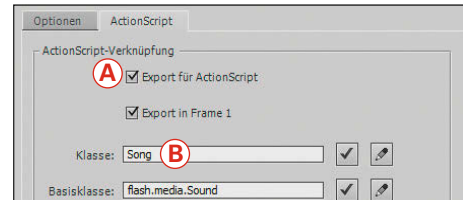
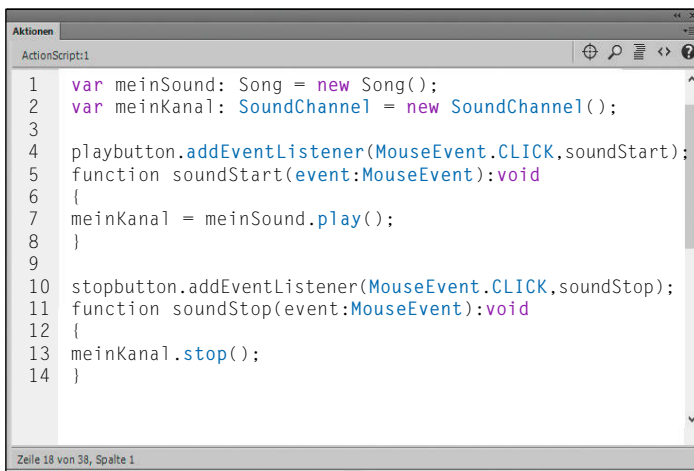
## 2.4.5 Sound steuern

Für die Verwendung von Sound haben Sie grundsätzlich zwei Möglichkeiten: Kurze Sounds können Sie in die Bibliothek Ihres Animate-Projekts importieren und direkt per ActionScript steuern.

Bei längeren Sounds empfiehlt es sich, diese in externen Dateien zu belassen und erst bei Bedarf zu laden. Hierdurch reduziert sich die Datenmenge.

### Internen Sound steuern – Making of ...

- 1 Öffnen Sie ein neues Animate-Projekt mit ActionScript 3.0.
- 2 Importieren Sie einen Sound im WAV- oder MP3-Format in die Bibliothek (siehe Seite 27).
- 3 Rechtsklicken Sie auf den Sound und setzen Sie in den *Eigenschaften* unter *ActionScript* das Häkchen bei *Export für ActionScript* **A**.
- 4 Erstellen Sie drei Schaltflächen für die Play-, Pause- und Stop-Funktion. Geben Sie den Buttons in den



Eigenschaften einen eindeutigen Namen, z. B. *playbutton*, *pausebutton*, *stopbutton*.

- 5 Erstellen Sie zunächst das Skript für den Play- und Stop-Button:

*Zeile 1:*

Zunächst wird eine Variable (hier: *meinSound*) der Klasse *Song* erzeugt, die Sie gerade neu definiert haben **B**.

*Zeile 2:*

Mit Animate können mehrere Sounds abgespielt werden. Deshalb muss jedem Sound eine Variable (hier: *meinKanal*) der Klasse *SoundChannel* zugeordnet werden.

*Zeilen 4 bis 8:*

Skript des Play-Buttons: Bei Eintreten des Ereignisses (*CLICK*) wird der dem Kanal zugeordnete Sound abgespielt.

*Zeilen 10 bis 14:*

Skript des Stop-Buttons: Bei Eintreten des Ereignisses (*CLICK*) wird der dem Kanal zugeordnete Sound beendet.

- 6 Testen Sie Ihr ActionScript – der Sound müsste sich nun starten und stoppen lassen.
- 7 Das ActionScript besitzt noch eine Schwäche: Wenn Sie mehrmals

auf den Play-Button klicken, wird der Sound mehrmals gestartet und lässt sich dann nicht mehr stoppen. Diesen Fehler beheben wir (siehe Screenshot rechts), indem wir mittels Variable prüfen, ob der Sound bereits läuft.

**Zeile 3:**

Eine Variable (hier: `spiele`) vom Typ `Boolean` wird eingeführt. Dieser Typ zeichnet sich dadurch aus, dass er nur zwei Werte, wahr (`true`) oder falsch (`false`), annehmen kann. Zu Beginn wird die Variable auf `false` gesetzt, weil der Sound noch nicht abgespielt wird.

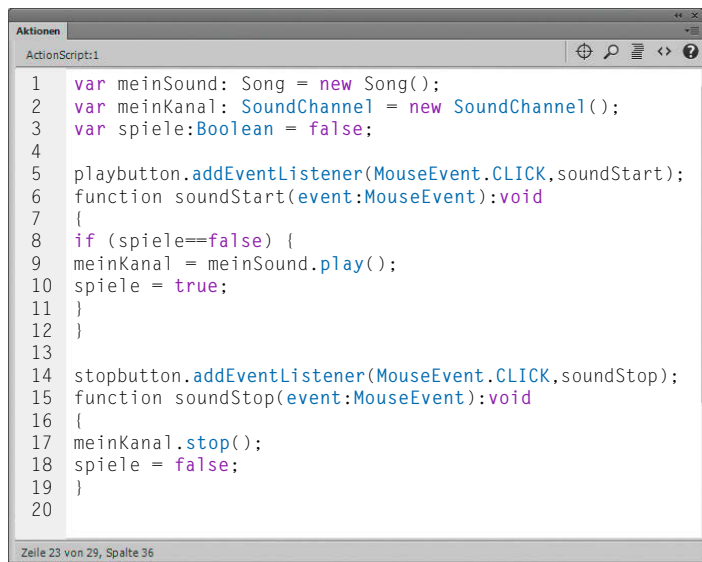
**Zeile 8 bis 11:**

Das Abspielen des Sounds wird nun an die Bedingung (`if`) geknüpft, dass die Variable `spiele` auf `false` gesetzt ist. Nachdem der Sound gestartet wurde, wird `spiele` auf `true` gesetzt. Beim erneuten Anklicken ist die Bedingung deshalb nicht mehr erfüllt und der Sound kann kein zweites Mal gestartet werden. Beachten Sie die zwei schließenden Klammern in den Zeilen 11 und 12. Sie sind erforderlich, weil in den Zeilen 7 und 8 auch zwei Klammern geöffnet werden.

**Zeile 18:**

Nach dem Stoppen des Sounds wird `spiele` wieder auf `false` gesetzt, so dass ein erneutes Starten möglich wird.

- 8 Im letzten Schritt erweitern wir das Skript um den Pause-Button. Hierzu benötigen wir eine Variable, in der die aktuelle Abspielzeit gespeichert wird.



```

1  var meinSound: Song = new Song();
2  var meinKanal: SoundChannel = new SoundChannel();
3  var spiele:Boolean = false;
4
5  playbutton.addEventListener(MouseEvent.CLICK, soundStart);
6  function soundStart(event:MouseEvent):void
7  {
8      if (spiele==false) {
9          meinKanal = meinSound.play();
10         spiele = true;
11     }
12 }
13
14 stopbutton.addEventListener(MouseEvent.CLICK, soundStop);
15 function soundStop(event:MouseEvent):void
16 {
17     meinKanal.stop();
18     spiele = false;
19 }
20

```

(Das Skript finden Sie auf der nächsten Seite.)

**Zeile 4:**

Für den Pause-Button wird eine Variable (hier: `pos`) vom Datentyp `int` definiert, mit dem sich ganze Zahlen speichern lassen. Der Startwert beträgt 0 (Sekunden).

**Zeile 10:**

Die Angabe der Variablen `pos` in der Klammer bewirkt, dass der Sound ab dem dort gespeicherten Wert gestartet wird. Wurde zuvor der Pause-Button betätigt, wird der Sound also ab der aktuellen Stelle fortgesetzt.

**Zeile 20:**

Wenn der Stop-Button gedrückt wird, muss die Variable `pos` wieder auf null gesetzt werden, damit der Sound bei Neustart wieder von vorne abgespielt wird.

```

Aktionen
ActionScript:1
1  var meinSound: Song = new Song();
2  var meinKanal: SoundChannel = new SoundChannel();
3  var spiele:Boolean = false;
4  var pos:int = 0;
5
6  playbutton.addEventListener(MouseEvent.CLICK,soundStart);
7  function soundStart(event:MouseEvent):void
8  {
9      if (spiele==false) {
10         meinKanal = meinSound.play(pos);
11         spiele = true;
12     }
13 }
14
15 stopbutton.addEventListener(MouseEvent.CLICK,soundStop);
16 function soundStop(event:MouseEvent):void
17 {
18     meinKanal.stop();
19     spiele = false;
20     pos = 0;
21 }
22 pausebutton.addEventListener(MouseEvent.CLICK,soundPause);
23 function soundPause(event:MouseEvent):void
24 {
25     pos = meinKanal.position;
26     meinKanal.stop();
27     spiele = false;
28 }

```

Zeile 5 von 29, Spalte 1

**Zeile 22 bis 28:**  
Durch Betätigung des Pause-Buttons ändert sich der Wert von `pos` auf die gerade aktuelle Abspielposition im Sound. Danach wird der Soundkanal gestoppt.

Wenn Sie mehrere Sounds verwenden, empfiehlt es sich, diese in externen

```

Aktionen
Ebene 1:1
1  var meinSong: URLRequest = new URLRequest("Song.mp3");
2  var meinSound: Sound = new Sound(meinSong);
3  var meinKanal: SoundChannel = new SoundChannel();
4  var spiele: Boolean = false;
5  var pos: int = 0;
6
7  // Buttonprogrammierung siehe oben
8
9

```

Dateien zu belassen. Der Vorteil dabei ist, dass die SWF-Datei klein bleibt und schnell geladen wird. Der oder die Sounddateien werden erst bei Bedarf (nach-)geladen.

Um das Abspielen eines externen Sounds zu ermöglichen, muss das im vorherigen Abschnitt erarbeitete Skript lediglich an einer Stelle verändert werden.

### Externen Sound steuern – Making of ...

- 1 Verwenden Sie Ihr Animate-Projekt aus dem vorherigen Abschnitt.
- 2 Löschen Sie den importierten Sound aus der Bibliothek und fügen Sie stattdessen eine Sounddatei in das Verzeichnis, in dem sich auch das Animate-Projekt befindet (hier: `Song.mp3`).
- 3 Ändern Sie die beiden obersten Zeilen des ActionScripts.

**Zeile 1:**  
Für den externen Sound benötigen Sie eine Variable (hier: `meinSong`) der Klasse `URLRequest`. Geben Sie in der Klammer in Anführungszeichen den Dateinamen des Sounds an (hier: `"Song.mp3"`).

**Zeile 2:**  
Im Unterschied zu einem importierten Sound *müssen* Sie bei externen Sounds die vordefinierte Klasse `Sound` verwenden. Dieser Klasse wird der Name der `URLRequest`-Variablen übergeben, hier `meinSong`.

Das übrige Skript unterscheidet sich nicht vom vorherigen, so dass die Steuerung bereits funktionieren sollte.

## 2.4.6 Animationen

In Kapitel 2.3 haben Sie einige Möglichkeiten kennengelernt, wie Sie Animationen ohne ActionScript erstellen können. Wozu also Animationen mit ActionScript?

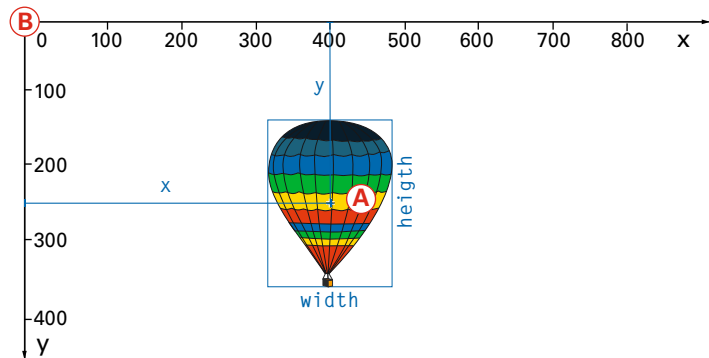
Die Vorteile der Programmierung von Animationen bestehen darin, dass

- sich Animationen exakt(er) steuern lassen, wenn Sie deren Parameter numerisch vorgeben,
- Sie letztlich Zeit und Arbeit sparen, wenn Sie Animationen mehrfach benötigen und hierfür lediglich ein Skript kopieren müssen,
- die Zeitleiste „aufgeräumt“ ist und damit übersichtlich bleibt,
- viele Funktionen, z. B. die Bewegung von Objekten mit der Maus, ohne ActionScript nicht möglich sind.

Die Animationsmöglichkeiten per ActionScript sind praktisch grenzenlos. Wir zeigen hier lediglich einen kleinen Einblick in diese Möglichkeiten. In der ersten Übung bewegt sich ein Ballon von links unten nach rechts oben über die Bühne, verkleinert sich hierbei und wird am Ende unsichtbar.

### Lineare Bewegung – Making of ...

- 1 Erstellen Sie das zu animierende Objekt (hier: Ballon) in Animate oder importieren Sie es aus einer Grafiksoftware in die Bibliothek.
- 2 Konvertieren Sie das Objekt im Menü *Modifizieren > In Symbol konvertieren* in ein Movieclip-Symbol.
- 3 Legen Sie den Bezugspunkt (Registrierung) des Objekts fest: Doppelklicken Sie hierzu auf das Symbol und verschieben Sie das Objekt, bis sich die Registrierung
- 4 Platzieren Sie die Symbolinstanz an einer beliebigen Stelle auf der Bühne. Vergeben Sie ihr in den *Eigenschaften* einen eindeutigen Namen ohne Umlaute und Sonderzeichen, z. B. *ballon* – er wird für das ActionScript benötigt.
- 5 Bevor wir uns mit dem Skript beschäftigen, werfen wir einen Blick auf das Koordinatensystem: In der Informatik ist es üblich, dass sich der Nullpunkt **B** links oben befindet und die y-Achse nach unten zeigt. In der Grafik besitzt das Objekt (genauer: seine Registrierung) also die Koordinaten  $x=400$  (Pixel) und  $y=250$  (Pixel).
- 6 Geben Sie das ActionScript in Bild 1 ein (siehe nächste Seite).



(kleines schwarzes Kreuz) **A** an der gewünschten Stelle befindet. Verlassen Sie das Symbol durch Anklicken des Pfeils links oberhalb der Bühne.

#### Zeilen 4 bis 8:

Der Ballon wird auf die gewünschten Startwerte (Position, Größe) gesetzt: Der y-Wert befindet sich außerhalb der Bühne, damit der

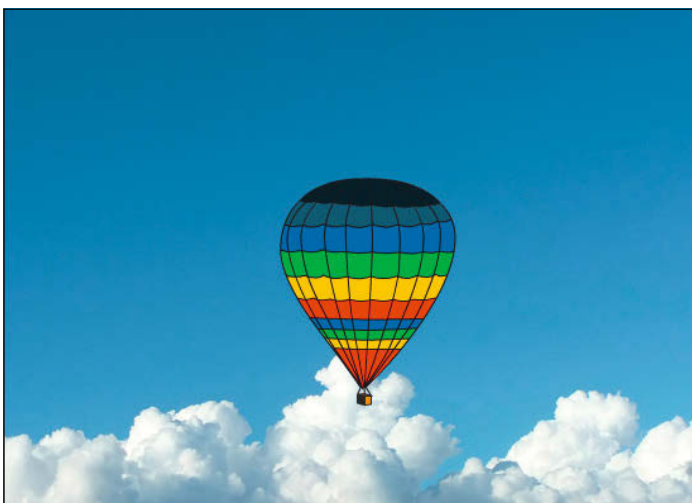
```

Aktionen
ActionScript:1
1  stop();
2
3  //Startwerte
4  ballon.x = 100;
5  ballon.y = 600;
6  ballon.width = 320;
7  ballon.height = 410;
8  ballon.alpha = 1;
9
10 //Animation
11 ballon.addEventListener(Event.ENTER_FRAME, bewegeBallon);
12 function bewegeBallon(e:Event):void {
13     e.target.x += 1;
14     e.target.y -= 2;
15     e.target.scaleX -= 0.006;
16     e.target.scaleY -= 0.006;
17     if (e.target.y < 150) e.target.alpha -= 0.005;
18 }
19
Zeile 23 von 29, Spalte 36

```

Ballon zu Beginn nicht sichtbar ist. Die Eigenschaft `alpha` definiert die Durchsichtigkeit zwischen 0 (unsichtbar) und 1 (sichtbar).

**Zeile 11:**  
Wie schon oft, benötigen wir auch hier wieder einen Event-Listener. In



diesem Fall wird das Event `ENTER_FRAME`, also „Betreten des Bildes“, geprüft. Obwohl der Abspielkopf auf Bild 1 scheinbar steht, wird dieses Ereignis ständig wiederholt, weil das Skript prüft, ob irgendwelche Ereignisse eintreten. Hierzu verlässt der Abspielkopf das Bild (`EXIT_FRAME`), um es gleich darauf wieder zu betreten (`ENTER_FRAME`).

**Zeilen 12 bis 18:**

In der Funktion bestimmen Sie, was sich bei jeder Wiederholung ändern soll. Zunächst wird die Position verändert: Die Kurzschreibweise `+=` bzw. `-=` bewirkt, dass die `x`- bzw. `y`-Koordinate um die angegebene Anzahl an Pixel erhöht bzw. verringert wird. Im Beispiel bewegt sich der Ballon pro Wiederholung 2 Pixel nach oben und 1 Pixel nach rechts. Über `scaleX` bzw. `scaleY` lässt sich das Objekt skalieren, wobei die 1 für 100% steht. Im Beispiel verkleinert sich der Ballon pro Wiederholung um 0,6%. Die letzte Anweisung ändert die Transparenz um 0,5% pro Wiederholung. Die `if`-Anweisung sorgt dafür, dass dies erst geschieht, wenn der Ballon die `y`-Koordinate 150 unterschritten hat – also gegen Ende der Bewegung.

Als zweite Möglichkeit, Objekte zu bewegen, zu drehen oder zu transformieren, bietet sich die Zeitsteuerung an. In der nächsten Übung animieren wir eine analoge Uhr.

### Zeitgesteuerte Animation – Making of ...

- 1 Erstellen Sie eine Uhr ohne Zeiger und importieren Sie die Grafik in ein neues ActionScript-Animate-Projekt.

- 2 Erstellen Sie einen Sekunden-, Minuten- und Stundenzeiger in Animate und konvertieren Sie die Zeiger in Movieclip-Symbole.
- 3 Doppelklicken Sie nacheinander auf die Symbole, um diese bearbeiten zu können. Verschieben Sie die Grafik so, dass sich die Registrierung (schwarzes Kreuz) am späteren Drehpunkt des Zeigers befindet.
- 4 Platzieren Sie die drei Zeiger auf der Uhr und vergeben Sie den Symbolinstanzen in den *Eigenschaften* eindeutige Namen.
- 5 Geben Sie in Bild 1 das ActionScript rechts oben ein.

```

Aktionen
ActionScript:1
1 //Sekundenzeiger
2 var sekunden:Timer = new Timer(1000);
3 sekunden.addEventListener(TimerEvent.TIMER, zeigeSek);
4 sekunden.start();
5 function zeigeSek(e:TimerEvent):void {
6     sekundenzeiger.rotation +=6;
7 }
8 //Minutenzeiger
9 var minuten:Timer = new Timer(60000);
10 minuten.addEventListener(TimerEvent.TIMER, zeigeMin);
11 minuten.start();
12 function zeigeMin(e:TimerEvent):void {
13     minutenzeiger.rotation +=6;
14 }
15 //Stundenzeiger
16 var stunden:Timer = new Timer(120000);
17 stunden.addEventListener(TimerEvent.TIMER, zeigeStd);
18 stunden.start();
19 function zeigeStd(e:TimerEvent):void {
20     stundenzeiger.rotation +=1;
21 }

```

Zeile 23 von 29, Spalte 36

**Zeilen 2, 9, 16:**

Für die drei Zeiger werden drei Variable der Klasse `Timer` benötigt. Die Werte in Klammer geben die Zeit in Millisekunden an: 1.000 ms = 1 s für den Sekundenzeiger; 60.000 ms = 60 s = 1 min für den Minutenzeiger; 120.000 ms = 120 s = 2 min für den Stundenzeiger (Erklärung siehe unten).

**Zeilen 3, 10, 17:**

An das `Timer`-Objekt werden Event-Listener angebracht. Diese rufen die jeweilige Funktion nach Ablauf der eingestellten Zeit auf. Danach wiederholt sich der Vorgang.

**Zeilen 6, 13, 20:**

Die Funktionen führen zur Drehung der Zeiger:

- Der *Sekundenzeiger* dreht sich pro Aufruf, also jede Sekunde, um  $6^\circ$ . Nach einer Minute hat er sich  $60 \times 6^\circ = 360^\circ$  gedreht, also eine volle Umdrehung gemacht.

- Der *Minutenzeiger* dreht sich pro Minute um  $6^\circ$ , erreicht also nach  $60 \times 6^\circ = 360^\circ$  eine volle Umdrehung.
- Der *Stundenzeiger* dreht sich alle zwei Minuten um  $1^\circ$ . Dies ergibt in einer Stunde eine Drehung um  $30 \times 1^\circ = 30^\circ$ , so dass er dann auf dem nächsten Teilstrich steht.





Zum Schluss möchten wir an einem Beispiel zeigen, dass Objekte mit Hilfe eines Skripts beweglich gemacht und durch den Nutzer bewegt werden können. Der Fachbegriff hierfür lautet *Drag & Drop* (dt.: ziehen und loslassen).

Als Beispiel realisieren Sie in dieser Übung das bekannte Logikspiel „Türme von Hanoi“ (Spielregeln siehe Wikipedia).

### Drag & Drop – Making of ...

- 1 Erstellen Sie die fünf Spielsteine direkt in Animate oder importieren Sie sie aus einer Grafiksoftware.
- 2 Konvertieren Sie die Spielsteine in Movieclip-Symbole und platzieren Sie diese (als Turm) auf der Bühne.
- 3 Vergeben Sie jeder Instanz in den *Eigenschaften* einen Namen.
- 4 Geben Sie in Bild 1 das ActionScript unten für den ersten Stein ein.

*Zeile 2:*

Der Event-Listener des braunen Steins wartet auf das Ereignis

```
Aktionen
ActionScript:1
1 //Brauner Stein
2 braun.addEventListener(MouseEvent.CLICK,startBraun);
3 function startBraun(e:MouseEvent):void {
4   braun.addEventListener(MouseEvent.CLICK,zieheBraun);
5 }
6 braun.addEventListener(MouseEvent.CLICK,stopBraun);
7 function stopBraun(e:MouseEvent):void {
8   braun.removeEventListener(MouseEvent.CLICK,zieheBraun);
9 }
10 function zieheBraun(e:MouseEvent):void {
11   braun.x = mouseX;
12   braun.y = mouseY;
13   e.updateAfterEvent();
14 }
15
```

Zeile 18 von 38, Spalte 1



MOUSE\_DOWN, also die gedrückte Maustaste. Danach wird die Funktion `startBraun` aufgerufen.

*Zeilen 3 bis 5:*

Die aufgerufene Funktion `startBraun` versieht den Stein mit einem neuen Listener, der auf `MOUSE_MOVE`, also Mausbewegung, reagiert und die Funktion `zieheBraun` aufruft.

*Zeilen 10 bis 14:*

Die Funktion `zieheBraun` platziert den Stein auf den Koordinaten der Maus. Bei Bewegung der Maus mit gedrückter Maustaste wird der Stein deshalb mitbewegt.

*Zeilen 6 bis 9:*

Der dritte Listener wartet auf das Ereignis `MOUSE_UP`, also das Loslassen der Maustaste. Die aufgerufene Prozedur entfernt (engl.: `remove`) den `MOUSE_MOVE`-Listener, so dass der Stein nicht mehr bewegt werden kann.

- 5 Für die anderen Steine brauchen Sie das Skript nur zu kopieren und die Farben zu ändern.
- 6 Jetzt ist Ihr Spiel einsatzbereit und Sie müssen das Rätsel um die *Türme von Hanoi* nur noch lösen.

### 1 Animate und Flash unterscheiden

Nennen und erklären Sie den wesentlichen Vorteil von Adobe Animate im Vergleich zu Flash.

.....

.....

.....

.....

.....

### 2 Symbol und Instanz unterscheiden

a. Unterscheiden Sie folgende Begriffe.

*Symbol:*

.....

.....

.....

*Instanz:*

.....

.....

.....

b. Erklären Sie, wie sich Änderungen, z. B. Größe, Farbe, auswirken:

*Änderung des Symbols:*

.....

.....

.....

*Änderung der Instanz:*

.....

.....

.....

### 3 Schaltflächen-Symbol kennen

Erklären Sie die Besonderheit der Zeitleiste eines Schaltflächen-Symbols.

.....

.....

.....

.....

### 4 Animationsdauer berechnen

Eine Animation soll 7 s lang dauern. Die Bildrate beträgt 24 BpS.

a. Berechnen Sie, in welchem Bild die Animation endet.

.....

.....

b. Die Animation wird nun bis Bild 210 verlängert. Berechnen Sie die notwendige Änderung der Bildrate, damit die Animation wieder 7 s lang dauert.

.....

.....

c. Erklären Sie, wie sich die Änderung der Bildrate auf die Animation auswirkt.

.....

.....

.....

---

## 5 Animationstechniken unterscheiden

- a. Unterscheiden Sie Einzelbild- von Schlüsselbild-Animationen.

.....

.....

.....

.....

- b. Nennen Sie für jede Technik einen wesentlichen Vorteil.

.....

.....

.....

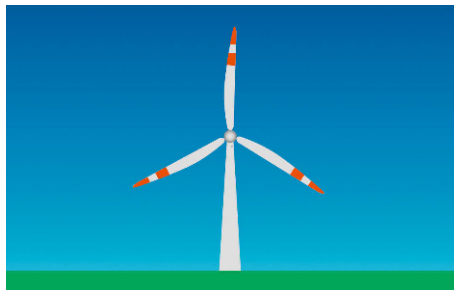
.....

---

## 6 Schlüsselbild-Animationen erstellen

Erstellen Sie folgende Schlüsselbild-Animationen:

- Lauftext von unten nach oben
- Ball, der auf und ab springt
- Windrad mit drei Flügeln, die sich drehen



- Luftballon, der aufsteigt, kleiner und unsichtbar wird
- Uhr mit sich drehenden Zeigern
- (eigene Idee)

---

## 7 Diashow erstellen

Erstellen Sie eine selbstablaufende Diashow mit mindestens zehn Bildern, bei der jedes Bild fünf Sekunden zu sehen ist.

---

## 8 Werbebanner realisieren

Realisieren Sie einen selbstablaufenden animierten Werbebanner im Format „Wide Skyscraper“ (160 x 600 Pixel) für ein Produkt oder Event Ihrer Wahl.

---

## 9 Grußkarte animieren

- a. Erstellen Sie in Illustrator eine Grußkarte, z.B. für einen Geburtstag, zu Weihnachten oder für einen sonstigen Anlass.
- b. Importieren Sie die Karte in Animate.
- c. Ergänzen Sie Animationen.
- d. Ergänzen Sie eventuell einen Hintergrund-Sound.

---

## 10 Cartoon animieren

- a. Entwerfen Sie einen Cartoon oder eine kurze Bilder Geschichte. (Alternativ können Sie auf eine Datei aus dem Internet zurückgreifen. Diese darf aus urheberrechtlichen Gründen jedoch nur für private Übungszwecke verwendet werden.)
- b. Importieren Sie die Grafik(en).
- c. Ergänzen Sie Animationen, z.B. Sprechblasen, Augen- oder Mundbewegungen bei Figuren.
- d. Ergänzen Sie eventuell Geräusche oder Sound. Im Internet finden Sie auch lizenzfreie Archive.

### 11 Animation mit inverser Kinematik erstellen

- Testen Sie die Vorgehensweise mit einfachen Objekten, z. B. mit zwei verbundene Stangen.
- Animieren Sie eine Maschine, z.B. einen Bagger (siehe Seite 12).
- Animieren Sie eine einfache Bewegung (siehe Seite 14).
- Animieren Sie eine Zeichentrickfigur (siehe Seite 39)

### 12 Verschachtelte Animation erstellen

Erstellen Sie verschachtelte Animationen. Vorschläge:

- Auto mit sich drehenden Rädern
- Vogel mit Flügelschlag
- Dampflokomotive mit Rauchwolken
- Rettungswagen mit Blaulicht
- Hubschrauber mit Rotoren
- Rakete mit animiertem Düsenantrieb
- (eigene Idee)

### 13 Animate-Projekt veröffentlichen

Animate-Projekte können auf unterschiedliche Weise veröffentlicht werden. Nennen Sie die erforderlichen Dateien, um ein Projekt wiedergeben zu können:

- im Webbrowser
- ohne Browser auf Windows-PCs (nur ActionScript 3.0)
- ohne Browser auf Apple-PCs (nur ActionScript 3.0)
- mit dem Flash-Player (nur ActionScript 3.0)

### 14 Funktion von ActionScript 3.0 kennen

- Erklären Sie die Funktion von ActionScript 3.0?

.....

.....

.....

.....

- Zählen Sie fünf Anwendungsbeispiele auf.

1. ....

2. ....

3. ....

4. ....

5. ....

### 15 ActionScript-Befehle kennen

Erklären Sie kurz die Funktion folgender ActionScript-Befehle:

- 

`stop();`

.....

- 

`nextFrame();`

.....

- 

`gotoAndStop();`

.....

## 16 Schaltflächen programmieren

Um Schaltflächen (Buttons) programmieren zu können, erhalten Sie einen sogenannten Event-Listener.

- a. Erklären Sie, wozu ein Event-Listener dient.

.....

.....

.....

- b. Nennen Sie drei typische „Events“

1.

2.

3.

## 17 Zeitleiste steuern

Gegeben sind zwei Zeilen eines Action-Scripts 3.0:

```
Aktionen
ActionScript:1
1 stop();
2 button1.addEventListener(MouseEvent.CLICK,aktion);
3
4
5
6
7
Zeile 18 von 38, Spalte 1
```

- a. Erklären Sie die Funktion der beiden Zeilen.

Zeile 1:

.....

Zeile 2:

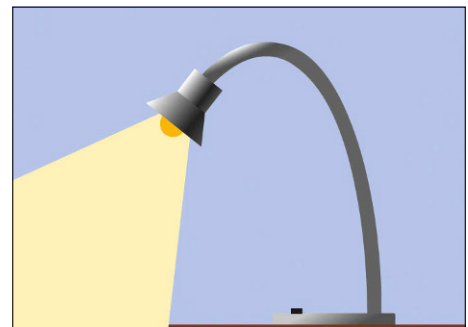
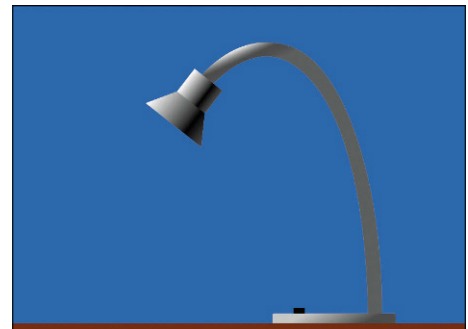
- b. Erweitern Sie das Skript, so dass die Animation bei Mausklick auf button1 zu Bild 15 springt und stoppt.

(Ergänzen Sie die Lösung links unten.)

- c. Erweitern Sie das Skript, so dass die Animation bei Mausklick auf einen zweiten Button (button2) ab Bild 30 fortgesetzt wird.

(Ergänzen Sie die Lösung links unten.)

## 18 Lampe animieren

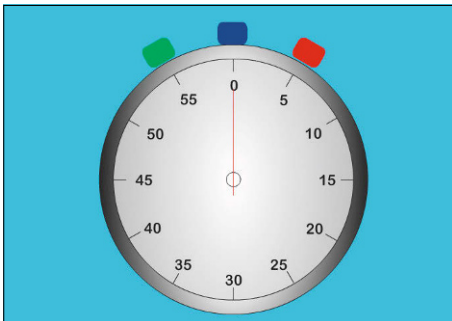


Gestalten und programmieren Sie eine Lampe, die sich mit Hilfe eines Schalters ein- und ausschalten lässt.

## 19 Stoppuhr animieren

Gestalten und programmieren Sie eine funktionsfähige Stoppuhr. Sehen Sie drei Buttons vor:

- Starten
- Stoppen
- Rücksetzen



Hinweise:

- Start-Button: Setzen Sie den Abspielkopf auf das Bild, das die Animation des Zeigers enthält. Die Animation kann wahlweise programmiert (siehe Uhr auf Seite 53) oder als innere Animation im Symbol realisiert werden.
- Stop-Button: Der `stop()`-Befehl kann mit einem Symbol verknüpft werden, z.B. `zeiger.stop()`;
- Rücksetzen-Button: Setzen Sie den Abspielkopf auf Bild 1 zurück.



Animation

Grundlagen - 2D-Animation - 3D-Animation

Bühler, P.; Schlaich, P.; Sinner, D.

2017, IX, 97 S. 150 Abb., 130 Abb. in Farbe., Softcover

ISBN: 978-3-662-53921-7