

# Multimedia-Programmierung

Heinrich Hußmann  
Ludwig-Maximilians-Universität München  
Sommersemester 2005

# A simple but important notice

- Im Hauptstudium sind viele aktuelle Materialien nur in englischer Sprache verfügbar.
- Programmiersprachen basieren auf englischem Vokabular.
- Austausch von Materialien zwischen Lehre und Forschung scheitert oft an der deutschen Sprache.
- Konsequenz:
  - Die wichtigsten Lehrmaterialien zu dieser Vorlesung (v.a. Folien) sind in englischer Sprache gehalten!
  - Der Unterricht findet (noch) in deutscher Sprache statt.

# Multimedia Programming

- Creative Designers of multimedia content (e.g. using Macromedia Flash):
  - “Many Flash users avoid to program in Flash. They are afraid of the jump into the ‘unknown’ area of programming.”

Cover text of  
B. Dawes: Flash ActionScript for Designers: Drag Slide Fade
- Traditional programmers with interest in multimedia (like the author of these slides):
  - Multimedia programming is a special case of general programming
    - » Why are then new development environments and even programming languages developed (like Macromedia Flash)?
- Questions (to be covered in this lecture):
  - Is there a need for bridging between graphical design and programming?
  - Are systems like Flash necessary, or are they just a way to avoid hiring programmers?
  - What is the most efficient way of developing multimedia applications?

# What we will cover – and what not

- This lecture does **not** cover:
  - Treatment of multimedia data on low system levels (operating system, networks)
  - Production of media products which are consumed in a linear, non-interactive way (like movies)
- The focus of the lecture is on:
  - Graphical representation and (2D-)animation
  - Interaction, including multi-user applications
  - Integration of computer graphics with other kinds of media
  - Development process in team work using the most recent software technologies (agile development, model-driven development)
  - The application area is programming of games
- The example development environment is:
  - Macromedia Flash MX 2004 Professional, ActionScript 2.0
- Other development environments will be covered as well (e.g. Squeak, Director)

# Organisatorisches

Ausnahmsweise auf Deutsch:

- Die Lehrveranstaltung (2V+4Ü) ist eine Mischung aus:
  - Vorlesung (12 Doppelstunden)
  - Klassische Übungen (7 Doppelstunden)
  - Projektarbeit in kleinen Gruppen (20 Doppelstunden)
  - ... plus eigene Freiarbeit
- Im Zeitraum 10.5.–12.7. findet die Projektphase statt:
  - Einteilung in kleinere Projektgruppen (je 6 Teilnehmer)
  - Ergebnispräsentation am Ende der Vorlesung (vor den Ferien)
- Leistungsnachweis durch erfolgreiche Teilnahme an der Projektphase
- Einbringung in mündliche Prüfung des Fachgebiets MM für Medieninformatik-Studierende (A bei Informatik)
  - Achtung MM-interne Regelung (nicht in der Prüfungsordnung): Mindestens 50% der eingebrachten Stunden müssen bei MM aus Vorlesungen stammen!
  - 4 SWS Übungen einzubringen v.a. bei 18-SWS-Prüfung empfehlenswert

# Outline (Preliminary)

1. Example technology: Macromedia Flash & ActionScript
  - 1.1 Multimedia authoring tools - Example Macromedia Flash
  - 1.2 Elementary concepts of ActionScript
  - 1.3 Interaction in ActionScript
  - 1.4 Media classes in ActionScript
2. Development process for multimedia projects
  - 2.1 Classical models of the software development process
  - 2.2 Special aspects of multimedia development projects
  - 2.3 Example: The SMART process
  - 2.4 Agile development/Extreme Programming for multimedia projects
  - 2.5 Modeling of multimedia applications
3. Introduction to computer game programming
  - 3.1 Logic-based games
  - 3.2 Simulation-based games
4. Overview on approaches to multimedia programming
  - 4.1 History of multimedia programming
  - 4.2 Squeak and Smalltalk: An alternative vision
  - 4.3 Advanced multimedia authoring with Director
  - 4.4 Frameworks for multimedia programming
5. Development trends and outlook

# 1 Example Technology: Macromedia Flash & ActionScript

## 1.1 Multimedia authoring tools - Example Macromedia Flash

Background: History, Context

Flash Authoring Tool: A Quick Tour

SWF Format

## 1.2 Elementary concepts of ActionScript

## 1.3 Interaction in ActionScript

## 1.4 Media classes in ActionScript

# The Expectations

“I like to think that if Rembrandt or Monet were alive today, they would be using Macromedia Flash MX and would be amazed by the level of creative expression they could achieve. Flash is a paintbrush that advances exponentially every year [...]”

Gary Grossman, Director of Engineering, Macromedia Inc.



# The Purpose of Multimedia Authoring Tools

- Multimedia programs are complex:
  - Usage of special libraries
    - » (2D) graphics primitives
    - » Converters for media formats
    - » Playback components
  - High data volume
    - » Requires special techniques like client/server, caching, ...
  - Synchronization issues
    - » Some streams in stepwise synchronicity (e.g. audio track for video)
  - Interaction techniques
    - » Flexible reaction to user actions
- Multimedia content is often created by non-technical people
- Authoring tools
  - Try to hide much of the complexity (using standard patterns and libraries)
  - Development environment accessible to non-technical people

# Macromedia Inc.: History

- 1984: *Macromind* (Jamie Fenton, Marc Carter, Mark Pierce)
  - *VideoWorks* (Joe Sparks)
    - » first timeline metaphor?
  - 1988: *VideoWorks* renamed to *Director*
- 1991:
  - Merger between *Macromind* and *Paracomp*
    - » 3D modeling tool *Swivel 3D*
  - Merger between *Macromind-Paracomp* and *Authorware*
    - » Courseware authoring tool *Authorware*
- 1996: New CEO *Rob Burgess* changes focus to Web publishing
  - HTML authoring tool *DreamWeaver*
- 1997:
  - Macromedia acquires *FutureWave Software*
    - » Key product *FutureSplash* renamed *Macromedia Flash*
- 2001:
  - Merger between Macromedia and *Allaire Systems* (*ColdFusion* server)



Not to be confused with the Munich-based training company “Macromedia” !



# Flash: History

- Jonathan Gay:
  - Software developer for *Silicon Beach Software* (starting in high school...)
  - Involved in various ground-breaking Macintosh applications: Airborne!, DarkCastle (1987), SuperPaint II, IntelliDraw (drawings with behaviour)
- 1993: Foundation of *FutureWave Software*
  - Goal: Develop sketching software (*SmartSketch*) for the new “pen computer” and the PenPoint operating system from the company GO
  - GO (and later EO) computers failed
- 1995-96: *SmartSketch* becomes *FutureSplash Animator*
  - Ported to Macintosh and Windows
  - Extended with 2D animation features
  - From the beginning targeted at delivery over the Web
  - Well accepted by important customers (e.g. Microsoft, Disney)
- 1996: FutureWave bought by Macromedia
  - FutureWave Splash becomes *Macromedia Flash 1.0*



# Flash vs. Director

- Director:
  - 10 years older than Flash
  - Designed for development of interactive CD-ROMs
  - Integrated programming language *Lingo*
  - Oriented towards bitmap graphics
  - Starting from Version 7: integration of Flash content
- Flash:
  - Designed for content delivery over the Internet (*streaming*)
  - Oriented towards vector graphics
  - Early versions (up to version 3) extremely simple in their interaction possibilities, later versions with increasing support for scripting

# Shockwave Plugins

- Shockwave:
  - General name for Web plugins playing Macromedia content
- *Shockwave for Director*:
  - Often simply called *Shockwave* plugin!
  - Plays content created with Director (Shockwave Movies)
  - File types: .dcr, .dir, .dxr
  - MIME type:  
    application/x-director
- *Shockwave Flash*
  - Often called *Flash* plugin, different from Shockwave plugin!
  - Plays content in SWF (Shockwave Flash) format
  - File types: .swf, .spl (from FutureSplash)
  - MIME types:  
    application/x-shockwave-flash  
    application/futuresplash

# Shockwave Flash (SWF)

- SWF is often pronounced as “swiff”
- File format for execution-ready presentations
  - Originally the proprietary compiled format of Flash presentations
  - Can be produced by various programs, not only Macromedia Flash
    - » E.g. open-source, multi-platform scripting language SSWF  
<http://sswf.sourceforge.net/>
- Specification of SWF publicly available:
  - <http://www.OpenSWF.org/>
- Flash browser penetration over 95%
- Players exist for many platforms:
  - PDAs
  - Mobile phones (in particular the *i-Mode* system from NTT DoCoMo)
  - Java
  - ...

# 1 Example Technology: Macromedia Flash & ActionScript

## 1.1 Multimedia authoring tools - Example Macromedia Flash

Background: History, Context

Flash Authoring Tool: A Quick Tour

SWF Format

## 1.2 Elementary concepts of ActionScript

## 1.3 Interaction in ActionScript

## 1.4 Media classes in ActionScript



# Timeline and Stage

current picture consists of 3 layers

Playback head

3 layers in parallel

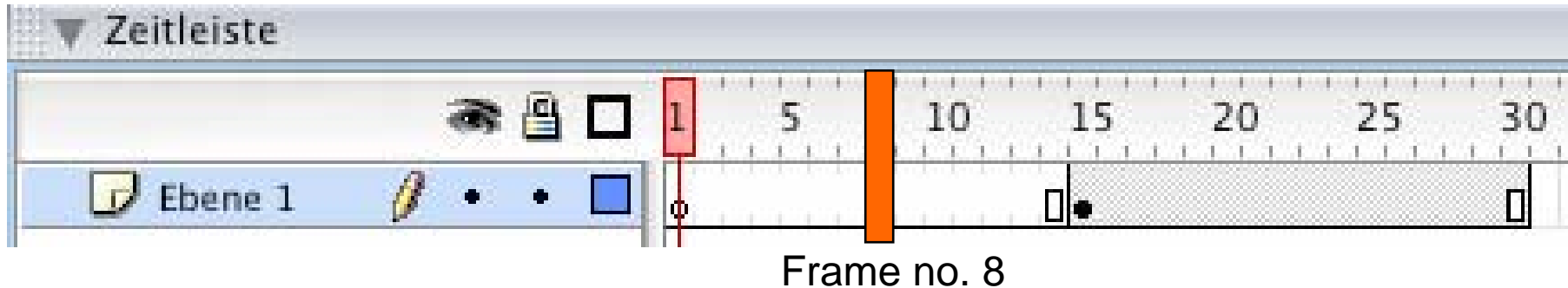
Frame change

Stage shows current frame

3 dimensions (2 plane dimensions plus time) mapped to 2D screen:

- 2D-frame (stage), no time
  - time plus layers, no frame content
- } combined

# Timeline Symbols



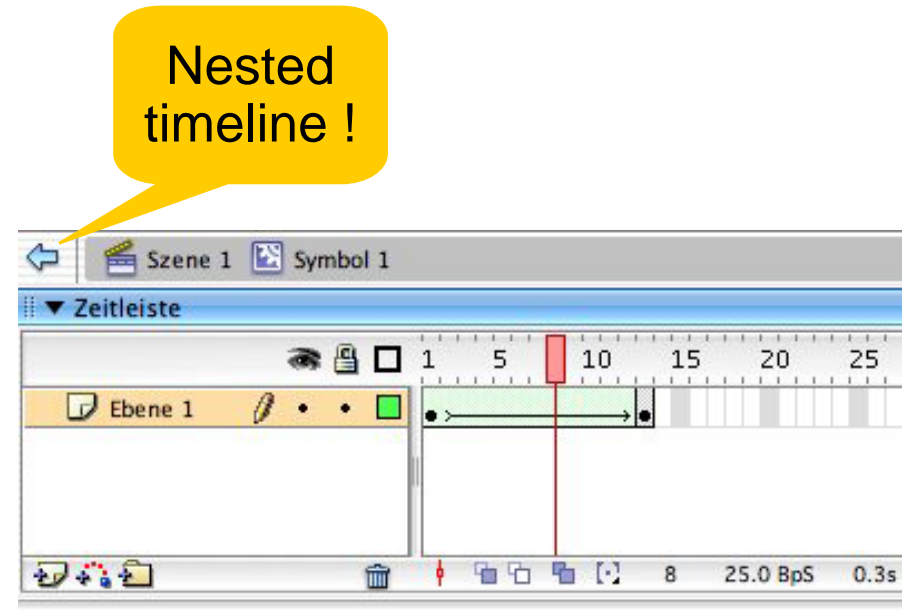
- The timeline contains *frames* (Bilder)
- *Key frames* (*Schlüsselbilder*) are defined explicitly (drawn by hand)
  - Representation in Flash:
    - hollow dot = empty key frame
    - black dot = key frame with content
- Default treatment of frame sequences: repeat last frame
  - Grey bar: Sequence of identical frames
  - Square: Last frame of a sequence
  - Changes in key frame affect all subsequent frames till next key frame!

# Animation: Change of Pictures over Time

- The suggestion of continuous change or movement:
  - Created by small changes from picture to picture
  - At least 12 frames per second, better more (25 and more)
- Single picture animation:
  - One graphic picture (drawn by hand) per frame
  - In Flash: Sequence of key frames
- Interpolation (*tweening*):
  - Sequence of frames defined by first and last frame
  - “In-between frames” generated automatically (interpolated)
  - “Tweening” possible with respect to several properties
    - » Size, location, orientation, colour of individual object (*motion tweening*)
    - » Shape of object (*shape/form tweening*)

# Hierarchical Timelines

- Each object can bring its own timeline
  - Instances of library symbols bring a copy of the timeline defined for the symbol
- Main timeline may be structured hierarchically into a tree of timelines
- Each instance of a symbol can move individually through its timeline
  - ActionScript code (see next lecture) can be added to navigate within timeline



# 1 Example Technology: Macromedia Flash & ActionScript

## 1.1 Multimedia authoring tools - Example Macromedia Flash

Background: History, Context

Flash Authoring Tool: A Quick Tour

SWF Format

## 1.2 Elementary concepts of ActionScript

## 1.3 Interaction in ActionScript

## 1.4 Media classes in ActionScript

# SWF

- The Macromedia Flash file format (SWF) (pronounced “swiff”) delivers vector graphics and animation over the Internet to the Macromedia Flash Player.
  - Pure delivery format
- Design goals:
  - On-screen display
    - » Designed for rendering
  - Extensibility
    - » Tagged format
  - Network delivery
    - » Compact binary format
  - Simplicity
  - Scalability regarding power of hardware
  - Scriptability
    - » Stack machine code compatible to “ActionScript” language

# Structure of a SWF File

- Tagged structure of objects
- Two categories of tags:
  - Definitions of content (shapes, bitmaps, sounds and so on):
    - » Each definition tag assigns a unique *character id*
    - » Definitions stored in *dictionary*
  - Control:
    - » Control the flow of the movie (timelines, actions)
    - » Create and manipulate instances of characters
- Built-in features (selection):
  - Layered display
  - Shapes, fill, line, edges, gradients, shape morphing
  - Bitmaps
  - Fonts and text
  - Buttons
  - Sound, video

# Example SWF (1)

- Tools for viewing SWF in readable form

E.g. *KineticFusion*: Conversion to XML-based language RVML (proprietary)  
(Rich Vector Markup Language)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!-- <!DOCTYPE Movie SYSTEM 'dtd/RVML.dtd' --> -->
<Movie version='6' width='550' height='400' rate='12'
  backgroundColor='white' compressed='Yes'
  xmlns='http://www.kineticfusion.org/RVML/1.0'>
  <Definitions>
    <Shape id='Main.Shape_1' bounds='bounds(197.0, 263.3, 298.95, 310.45)'>
      <FillStyles />
      <LineStyle>
        <LineStyle index='1' width='1.0' color='rgb(208,208,208)' />
        <LineStyle index='2' width='1.0' color='rgb(208,208,208)' />
      </LineStyle>
      <Edges>
        <Move x='197.5' y='309.95' />
        <SetStyle line='2' />
        <Line x='298.45' y='309.95' />
        <Line x='260.95' y='263.8' />
        <SetStyle line='1' />
        <Line x='197.5' y='309.95' />
        <ColorFill color='rgb(0,51,204)' />
        <SetStyle line='0' mainFill='0' rightFill='0' />
        <Move x='298.45' y='309.95' />
        <SetStyle mainFill='1' />
        <Line x='260.95' y='263.8' />
        <Line x='197.5' y='309.95' />
        <Line x='298.45' y='309.95' />
      </Edges>
    </Shape>
  ...
```



# Example SWF (2)

```
<Timeline frameCount='27'>
  <Frame frameNo='1'>
    <Place name='Main.Shape_1' depth='2' />
    <Place name='Main.Shape_2' depth='3'>
      <Transform scaleX='2.3459' scaleY='0.724'
        translateX='-626.35' translateY='11.5' />
    </Place>
    <Place name='Main.Shape_3' depth='4' />
  </Frame>
  <EmptyFrame blankFrames='8' />
  <Frame frameNo='10'>
    <Replace name='Main.Shape_4' depth='4' />
  </Frame>
  <EmptyFrame blankFrames='4' />
  <Frame frameNo='15'>
    <Remove name='Main.Shape_1' depth='2' />
    <Place name='Main.Shape_5' depth='1' />
  </Frame>
  <EmptyFrame blankFrames='3' />
  <Frame frameNo='19'>
    <Remove name='Main.Shape_2' depth='3' />
    <Place name='Main.Shape_2' depth='2' />
  </Frame>
  <EmptyFrame blankFrames='8' />
</Timeline>
</Movie>
```

# 1 Example Technology: Macromedia Flash & ActionScript

## 1.1 Multimedia authoring tools - Example Macromedia Flash

Background: History, Context

Flash Authoring Tool: A Quick Tour

SWF Format

## 1.2 Elementary concepts of ActionScript

Scripting in General + „History“ of ActionScript

Objects and Types in ActionScript

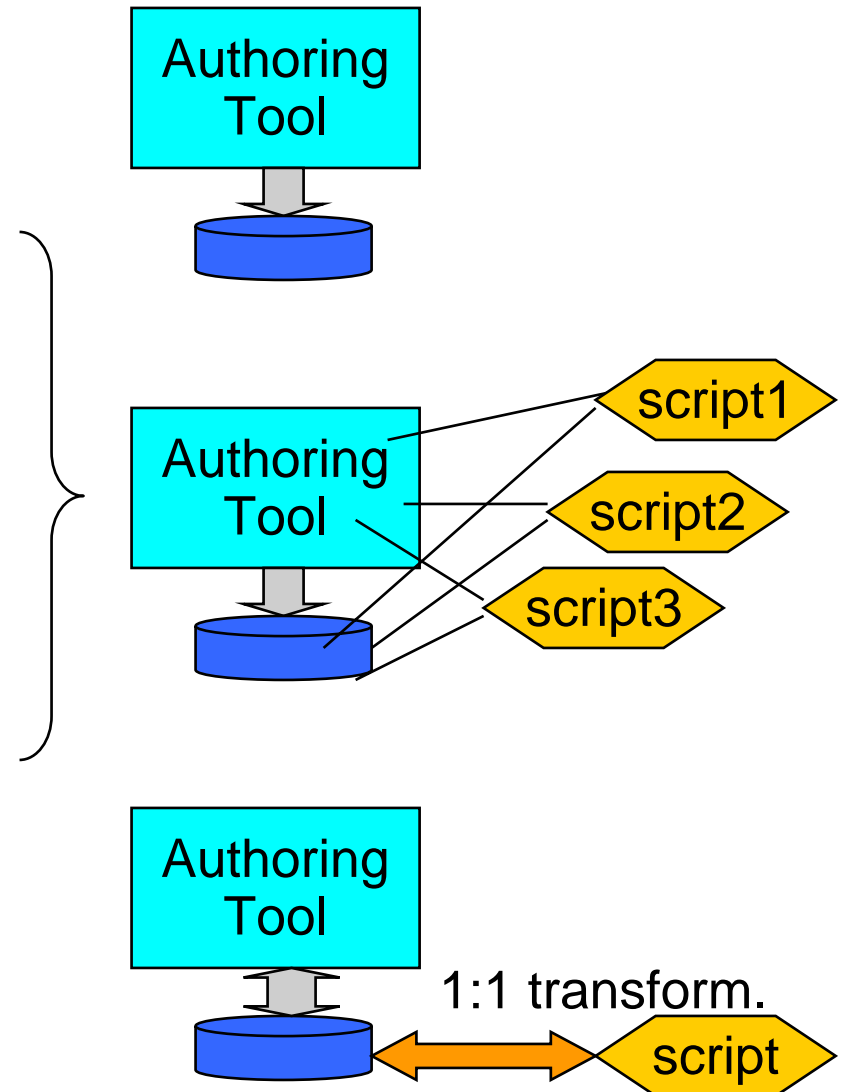
Animation with ActionScript

## 1.3 Interaction in ActionScript

## 1.4 Media classes in ActionScript

# Scripting Languages for Authoring Tools

- *Script-less authoring:*  
Just an authoring tool  
Scripts/programming avoided
- *Local scripting:*  
Scripts added at various places  
in the authoring environment  
to enhance expressiveness;  
scripts are *context-dependent*
- *Global scripting:*  
Separate script files in addition to the  
file produced with the authoring tool;  
scripts are *self-contained*
- *Script-based development:*  
Authoring tool as a comfortable view  
onto a program (script);  
Whole application can be written as  
a script in a formal language

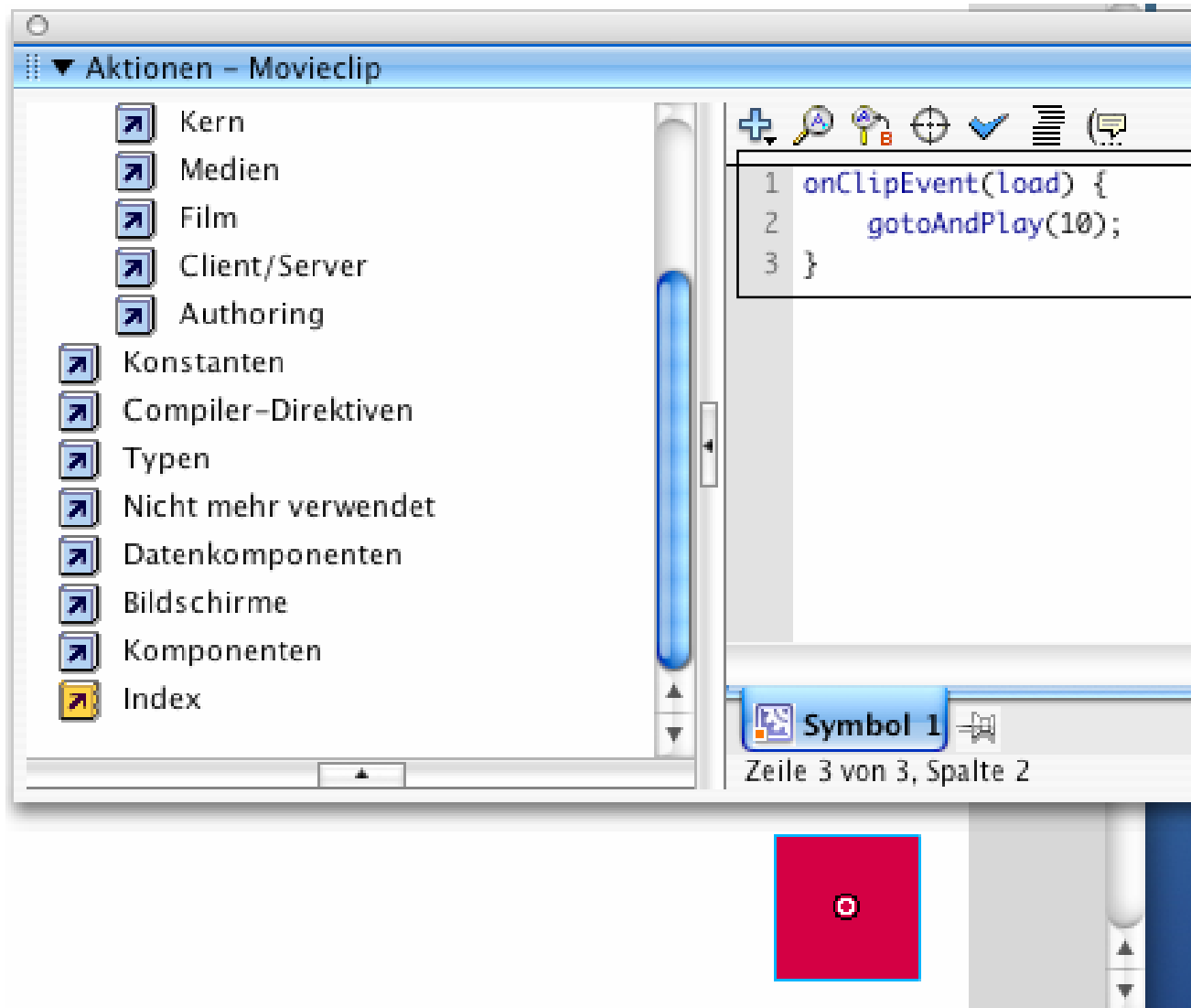


# Flash Software Versions

- Flash 1 to 3:
  - Only very limited interaction
  - No scripting at all (script-less)
- Flash 4:
  - Beginnings of ActionScript (local scripting)
- Flash 5:
  - ActionScript 1.0 (local + simple global scripting)
  - Execution very slow
- Flash 6 = Flash MX:
  - Improved execution speed
  - Custom object classes (object-based programming)
  - Prototype objects, inheritance, no classes yet
- Flash 7 = Flash MX 2004
  - ActionScript 2.0
  - Global scripting
  - Java-like syntax, full class concept (object-oriented programming)

Script-based  
development for Flash:  
E.g. with  
KineticsFusion/RVML

# Example of Local Scripting in Flash



Local event handler  
as script

# Example of Global Scripting (ActionScript 2.0)

## Object-Oriented Programming

```
class Account {  
    private var saldo:Number = 0;  
    private var num:Number;  
    public function Account(accnum:Number) {  
        num = accnum;  
    }  
    public function debit(n:Number) {  
        saldo -= n;  
    }  
    public function credit(n:Number) {  
        saldo += n;  
    }  
    public function getNumber():Number {  
        return (num);  
    }  
    public function getSaldo():Number {  
        return (saldo);  
    }  
}
```

# ActionScript 1.0 and ActionScript 2.0

- ActionScript 1.0 (AS1)
  - Simple scripting language
  - Not built for large-scale programming
  - Implicit typing (inferred from variable name and value)
  - Object-based
- ActionScript 2.0 (AS2)
  - Only from Flash MX 2004 and Flash player 7 upwards!
  - Based on the ECMAScript standard (proposal 4)
  - Very similar to Java (Object-oriented)
  - Multiple classes, each defined in its own source file
  - Strict explicit typing
  - Case sensitive

“ActionScript 2.0 can be called an object-oriented programming language, whereas previous versions were more modestly referred to as an object-*based* programming language, and that was only with the Flash MX version.” William B. Sanders

# Example of Global Scripting (ActionScript 1.0) Object-Based Programming

```
var Account = function(accnum) {  
    this.saldo = 0;  
    this.accNumber = accnum;  
}
```

```
Account.prototype.debit = function(n) {  
    this.saldo -= n;  
}
```

```
Account.prototype.credit = function(n) {  
    this.saldo += n;  
}
```

```
Account.prototype.getNumber = function() {  
    return (this.accNumber);  
}
```

```
Account.prototype.getSaldo = function() {  
    return (this.saldo);  
}
```



# Concepts of Object-Based Programming

- No classes:
  - Special objects (*prototypes*) serve as blueprints for newly created objects
  - No concept of a class!
  - Advantage: Dynamic changes to prototypes easily possible (e.g. adding a method - applies to all objects derived from prototype)
- ActionScript 1.0:
  - Variables can store everything, including functions and constructor functions
    - » This also applies to fields of objects
    - » No difference between attributes and methods
  - Constructor function as replacement for class
  - Constructor function associated with prototype (i.e. extensible by features)
- Flexibility vs. structuring:
  - Class-based, object-oriented languages (Java, AS 2) are easier to understand
  - Object-based languages (Smalltalk, AS 1) are more flexible and powerful
- Flash allows to some extent a mixture between the two styles!