# Multimedia-Programmierung

Heinrich Hußmann
Ludwig-Maximilians-Universität München
Sommersemester 2009

# Eine Warnung vorweg!

Diese Vorlesung ist sehr stark überarbeitet
im Vergleich zu den Vorgänger-Veranstaltungen
(zuletzt Sommer 2007).

Es werden andere (neuere) Technologien behandelt.
*Flash spielt nur noch eine Randrolle!*

Es wird (noch) stärkerer Wert auf Konzepte gelegt, die
von aktuellen Technologien abstrahieren!

# Um- und Ausbau des Lehrangebots zum Thema

- Vorlesung Multimediaprogrammierung (2 SWS, 4 ECTS)
  - Diese Veranstaltung, Sommersemester
  - Überblick über Techniken der Multimedia-Programmierung
- Übung Multimediaprogrammierung (3 SWS, 2 ECTS)
  - Begleitende Übungen zu dieser Vorlesung, Sommersemester
  - Übungen zur Programmierung mit Python und anderen Programmiersprachen in Kombination mit Multimedia-Frameworks
- Blockpraktikum Multimediaprogrammierung (4 SWS, 6 ECTS)
  - 2-wöchige Blockveranstaltung
  - Intensives Programmierprojekt, voraussichtlich mit Adobe Flash oder Flex
  - Zwischen Sommer- und Wintersemester, voraussichtlich Anfang September
- Für Diplom-Studierende:
  - Bis zu 9 SWS mit diesem Thema möglich (davon aber 7 Übung/Praktikum)
  - Siehe nächste Folie
- Für Bachelor-Studierende:
  - Hauptfach Medieninformatik: Bis zu zwei mal „Vertiefendes Thema“ (jeweils 6 ECTS)
  - Hauptfach Kunst und Multimedia: Teil des Pflichtmoduls „Medienpraxis“
    - » Achtung: Bei Problemen mit den Programmiergrundlagen bitte melden!

# Einbringung in Diplom-Studiengänge

Einbringung der erbrachten Leistung im Diplomstudium Informatik oder Medieninformatik:

- Ohne Schein:
  - 2 SWS Prüfungsstoff
- Mit Schein zu den begleitenden Übungen:
  - 5 SWS als Prüfungsstoff (wenn Schein erlangt), davon 3 SWS Übung
  - Alternativ Pflichtschein für MM-Säule (Medieninformatik) + 2 SWS Prüfungsstoff
- Mit Schein zum Blockpraktikum Multimediaprogrammierung:
  - Zusätzlich 4 SWS Prüfungsstoff (Übung/Praktikum)
  - D.h. maximal 9 SWS, davon 7 Übung/Praktikum
  - Für 12-SWS-Prüfung: 2 weitere Vorlesungen (zu je 2 SWS) nötig
  - Für 18-SWS-Prüfung: 3 weitere Vorlesungen (zu je 2 SWS) nötig

# Deutsch und Englisch

- 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

- Multimedia Programming:
  - Creating programs which make use of “rich media” (images, sound, animation, video)
- Key issue in multimedia programming: mixture of skills
  - Programmers are not interested in creative design
  - Designers are intimidated by programming
- Mainstream solution in industry:
  - Authoring tools (e.g. Adobe Flash) with scripting language
- Questions (to be covered in this lecture):
  - Which ways exist to bridge between creative design and programming?
    - » Different platforms and tools
    - » Which tool to chose for which purpose?
  - What is the most efficient way of developing multimedia applications?
    - » Which techniques exist to make multimedia programming easier?
    - » What is an adequate development process for multimedia programs?

# (Not) Covered Topics

- 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 representations and (2D-)animation
  - Integration of sound and video into programs
  - Interaction techniques for rich media
  - Development process in teamwork using recent software technologies
- Various example development environments will be covered:
  - Script languages with multimedia frameworks (based on Python)
  - Multimedia scripting languages (JavaFX, Processing)
  - Document-based platforms (SMILE, OpenLaszlo)
  - To a limited extent:
    - » Authoring tools (Flash)
    - » Java multimedia frameworks

# Organisatorisches

Ausnahmsweise auf Deutsch:

- Die Lehrveranstaltung (2V+2Ü+1P) ist eine Mischung aus:
  - Vorlesung (12 Doppelstunden)
  - Klassische Übungen (incl. Hausaufgaben)
  - Eigene Freiarbeit
- Leistungsnachweis durch:
  - Erfolgreiche Bearbeitung der Übungsblätter:
    - » Zwei „Joker“, d.h. zwei Abgaben können gestrichen werden
    - » Schein ohne Note für Diplomstudierende
    - » Zulassungsvoraussetzung zur Klausur für Bachelorstudierende
  - Klausur am Ende des Semesters:
    - » Ausschließlich für Bachelorstudierende
    - » Bestimmt die Note
- *Keine Projektphase* wie in früheren Jahren!
  - Siehe Blockpraktikum Multimediaprogrammierung

# Outline (Preliminary)

1. Introduction to Multimedia Programming
   - 1.1 General Goals and Application Areas
   - 1.2 Historical Background
   - 1.3 Classification of Development Platforms
2. Development Platforms for Multimedia Programming (Examples)
   - 2.1 Introduction to Python
   - 2.2 Multimedia Frameworks for Python
   - 2.3 Document-Based Platforms: SMIL, OpenLaszlo
   - 2.4 Multimedia Scripting Languages: JavaFX, Processing
   - 2.5 Authoring Tools: Flash
3. Challenges in Multimedia Programming
   - 3.1 Interactive and Event-Driven Programs
   - 3.2 Media Integration
   - 3.3 Synchronization
4. Programming with Images
5. Programming with Vector Graphics and Animations
6. Programming with Sound
7. Programming with Video
8. Design Patterns for Multimedia Programs
9. Development Process for Multimedia Projects
10. Modelling of Multimedia Applications

# 1 Introduction to Multimedia Programming

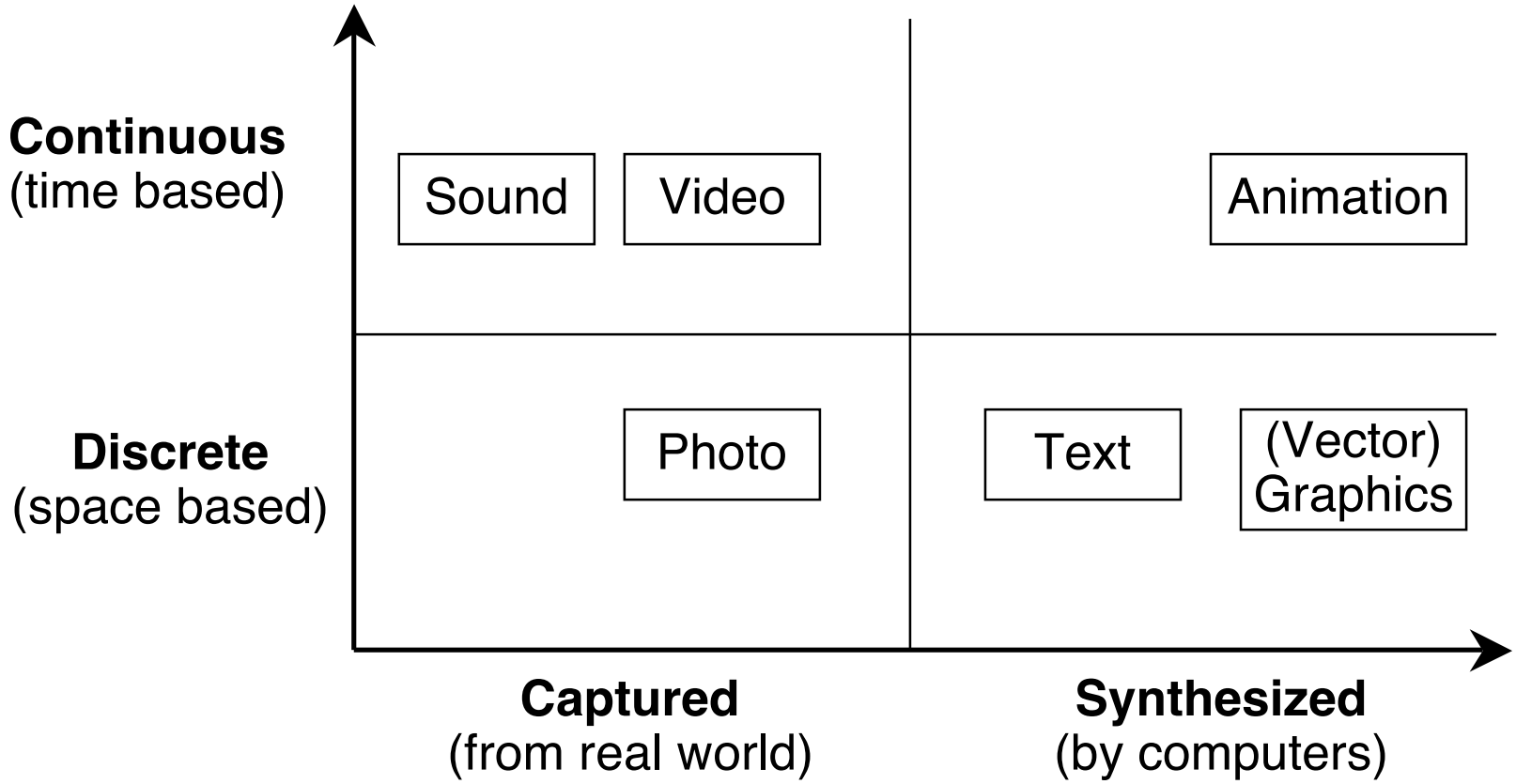
1.1 General Goals and Application Areas

1.2 Historical Background

1.3 Classification of Development Platforms

# Classification of Media Types

- Fetterman/Gupta 1993:



# Multimedia

- Literal definition: Combination of several media types
- Steinmetz/Nahrstedt 2004:
  “A multimedia document is characterized by information which is coded in at least one continuous (time-dependent) and one discrete (time-independent) medium.”
- Boll 2001:
  “A multimedia document is a media element that forms the composition of continuous and discrete media elements into a logically coherent multimedia unit.”

Observations:

- Multimedia is about *composing* and *integrating* (mono-)media.
- There is a soft borderline between *multimedia applications* and *multimedia documents*.
- A multimedia application/document always has a temporal aspect (i.e. is time-dependent).

# Multimedia vs. Multimodality

- *Modality*:
  - Channel used for communication with human user (human sense)
    - » See literature on Human-Computer Interaction
  - *Multimodal:* Using several different modalities
- Multimedia:
  - May combine media elements which belong to a single channel (i.e. visual channel: text, graphics, animation)
  - Does not require multiple channels (multimodality)
  - Combines media elements without analysing their inner structure
- Multimodality:
  - Often used for techniques which deeply *analyse* input information (e.g. speech, gestures)

# Interactivity

- Degrees of interactivity (based on T.A. Aleem 1998):
  - Passive, Reactive, Proactive, Directive
- Application to multimedia (Heller et al. 2001) - Examples:

| *Media type ↓* | *Passive* | *Reactive* | *Proactive* | *Directive* |
|---|---|---|---|---|
| *Text* | Sequential presentation | Page turner<br>Linear spacing | Browsing, Hypertext | Word processing |
| *Graphics* | Sequential presentation | Predefined changes (choice between graphics) | Change of colors, sizes, shapes, … | Drawing graphics |
| *Sound* | Sequential presentation | Predefined changes (sound clip, volume) | Selection of track, fast forward, loop | Creation of sounds |
| *Motion* | Sequential presentation | Predefined changes (path, target of motion) | Start, stop, pause, forwd, reverse | Creation of animations |

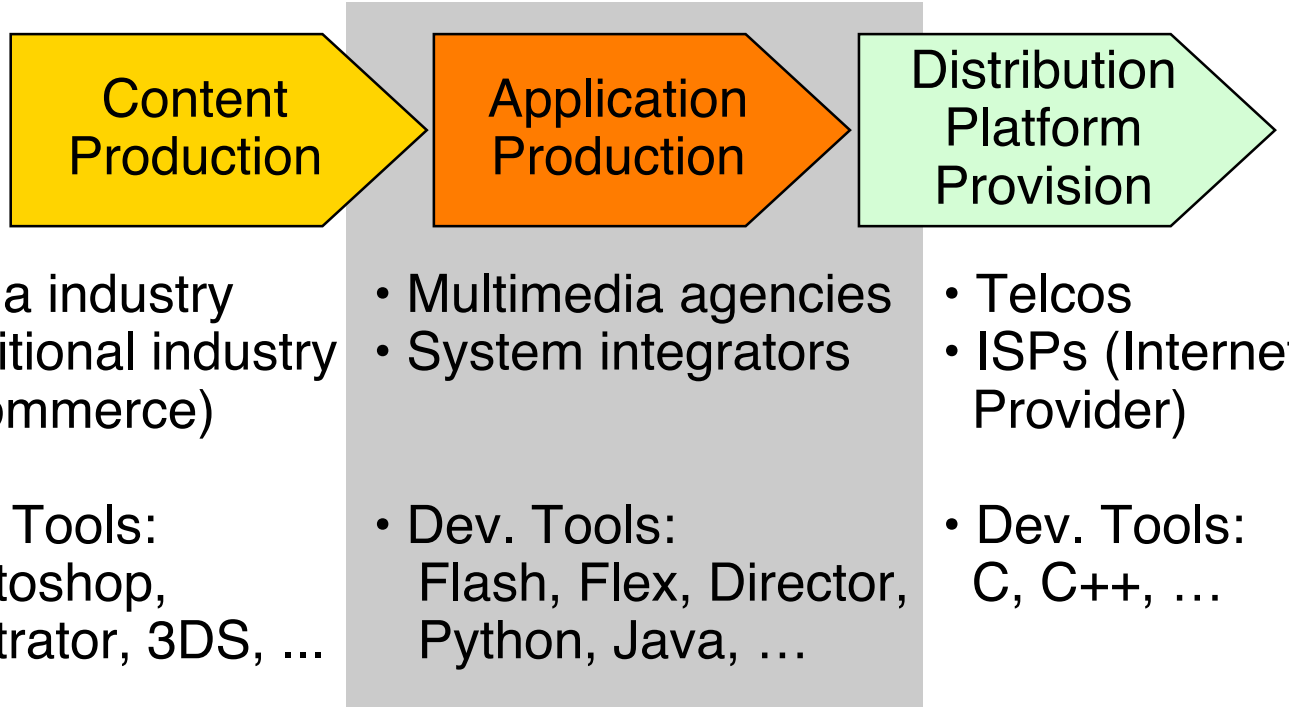
# Application Areas and Examples

| *Domain →<br>Interactivity ↓* | *Business* | *Information* | *Communi-cation* | *Edutain-ment* | *Education* |
|---|---|---|---|---|---|
| *Reactive* | Online shop | Encyclopedia | News channel | Media player | Medical course |
| *Proactive* | Car configurator | Navigation system | Video conference | Car racing game | Flight simulator |
| *Directive* | Authoring tool | Wiki | CSCW System | City building game | Electronic circuit simulator |

Based on Dissertation A. Pleuß (2009) and Projektarbeit S. Kraiker (2007)

# Multimedia Development

- Development of mostly interactive, possibly distributed, multimedia applications
- Typically carried out by “multimedia agencies” (Multimedia-Agenturen) or specialized software development companies
- Distribution channels: CD/DVD, Web, Broadcasting
- Position in the value chain:



# Origin of Media Content

- *Received:*
  - Media content is produced elsewhere
  - Multimedia application only integrates existing content
- *Designed:*
  - Media content is created specifically for the developed application
- *Generated:*
  - Media content is automatically constructed from application data (e.g. maps, data visualizations, thumbnail overviews)

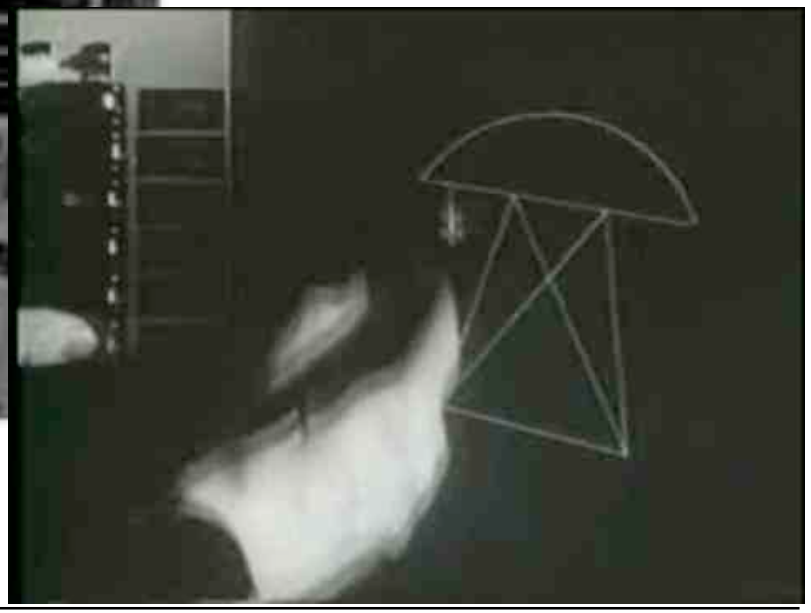
- In any case, copyright rules have to be taken care of!

# 1 Introduction to Multimedia Programming



# Ivan Sutherland's Sketchpad, 1963

- First object-oriented drawing program
- Master and instance drawings
- Rubber bands
- Simple animations

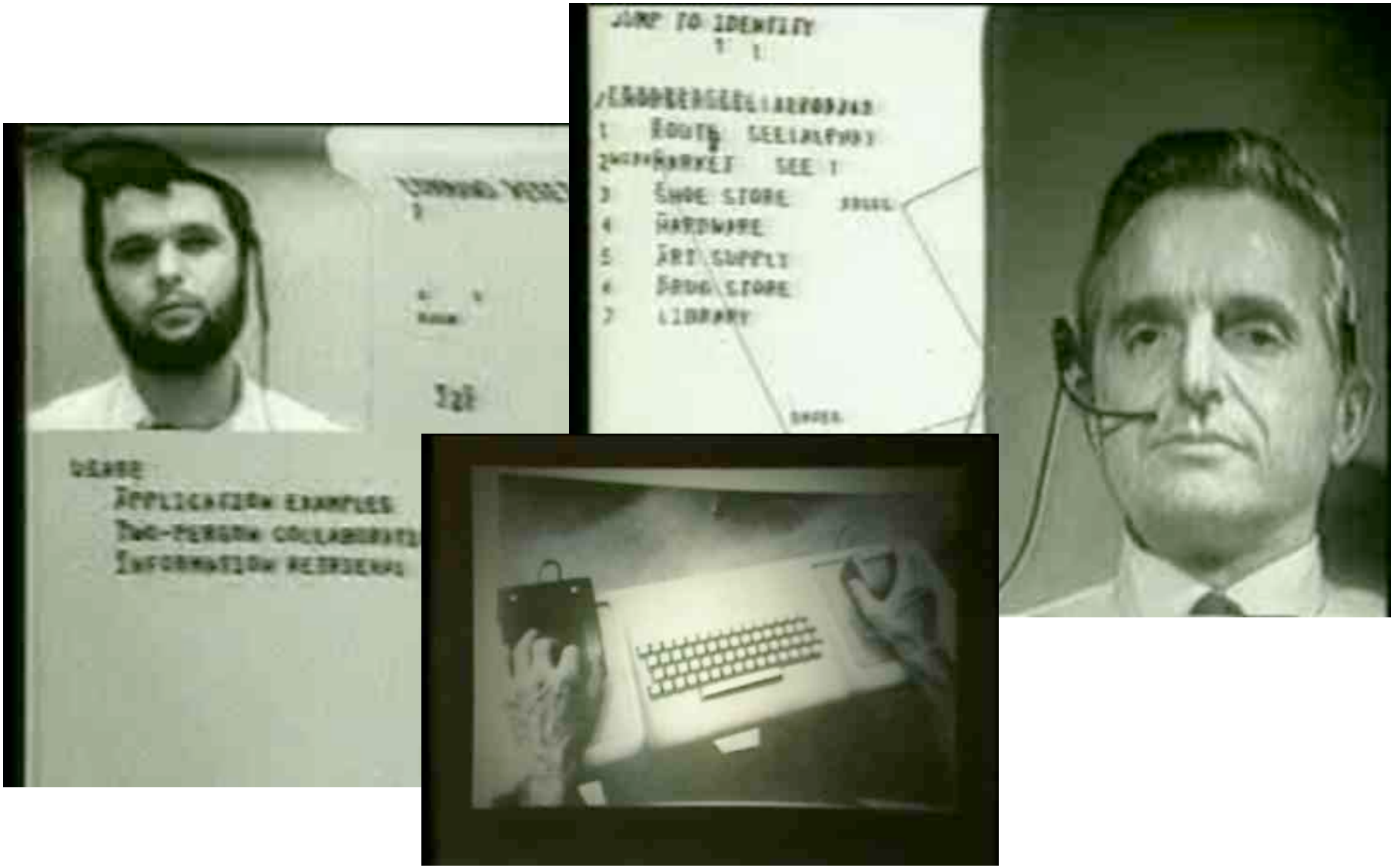


# Douglas C. Engelbart 1962

- Born 1925, Ph.D. Berkeley 1955
- Influenced by Vennevar Bush's article "As We May Think" (1945)
- 1962: Research Project at SRI (Stanford Research Institute): "Augmenting Human Intellect: A Conceptual Framework"
  - Research support triggered by the "Sputnik shock" (1957)
- Basic ideas:
  - Computer supported learning
  - Computer supported collaboration
  - Seamless integration of computer interaction into workflows
- Development of the "NLS" (oNLine System)
  - Demonstrated 1968 in Brooks Hall, San Francisco
- 1970: Patent application for "X-Y pointing device" (mouse)

http://www.bootstrap.org/augdocs/friedewald030402/augmentinghumanintellect/ahi62index.html

# NLS Demo 1968



# Alan C. Kay

- U. Utah PhD student in 1966
  - Read Sketchpad, Ported Simula
- Saw “objects” as the future of computer science
- His dissertation:
  Flex, an object-oriented *personal* computer
  - A *personal* computer was a radical idea then
  - How radical?

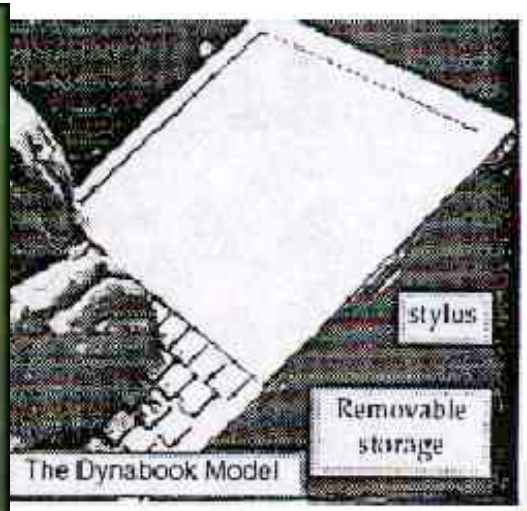
*"There is no reason anyone would want a computer in their home."*
(Ken Olsen, Digital Equipment Corp, **1977**)

Further stations of Alan Kay’s life:
– Stanford Artificial Intelligence Laboratory
– **Xerox PARC**
– Atari
– Apple
– Disney Interactive
– Viewpoints Research Institute
– Hewlett-Packard

from M. Guzdial

# The Dynabook Vision

- Small, handheld, wireless(!) device – a new *medium*
- Can be used creatively by everybody, in particular children, for learning
- Xerox PARC Learning Research Group, early 70s




Tablet PC, 2004

# Xerox PARC Learning Research Group: Smalltalk-72

- Object-oriented programming system
  - Mouse
  - Windows
  - Icons
  - Pop-up menus
- Uses simple object-oriented language “Smalltalk”
- Idea of user interface: Make computers easy to use for everybody
- Idea of language: make programming both more simple and more powerful (e.g. include *multimedia: sound*)

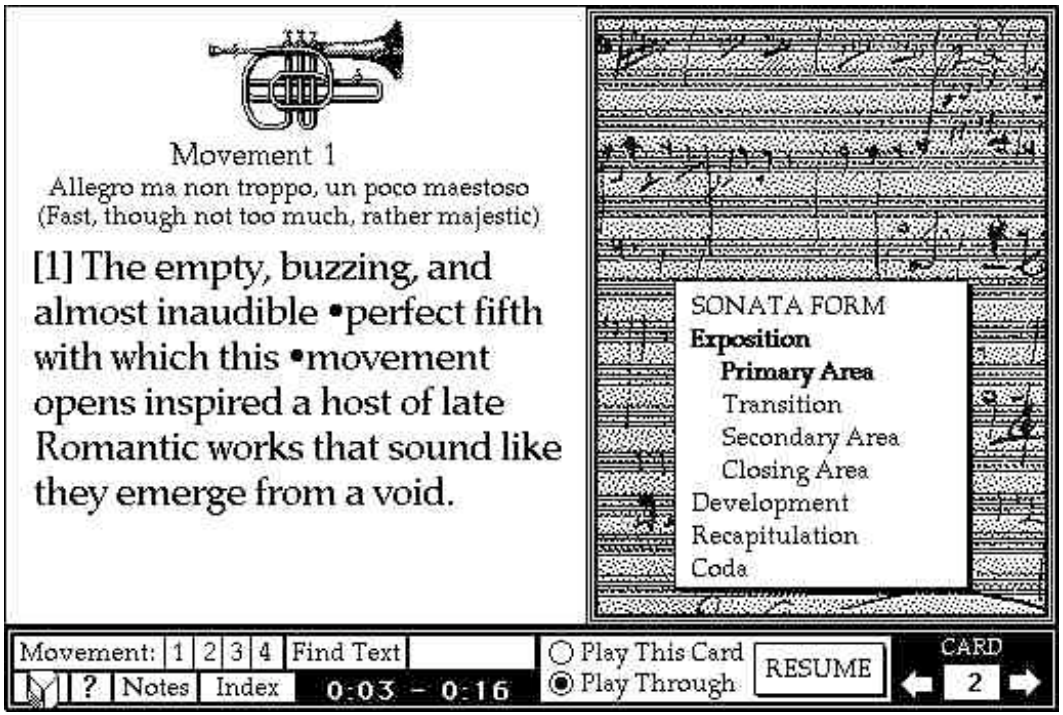
# The Alto

- The machine the prototype of which impressed Steve Jobs so much that he decided to produce the Lisa/Macintosh kind of computers for the mass market (1979)
  - Graphical user interface
  - Networked via Ethernet
  - Programming language Smalltalk
- Hardware:
  - 800 x 600 display
  - Data General 16 Bit processor
  - 400.000 instructions/second
  - 256 kByte – 512 kByte RAM
  - 2 x 2,5 MByte Festplatte

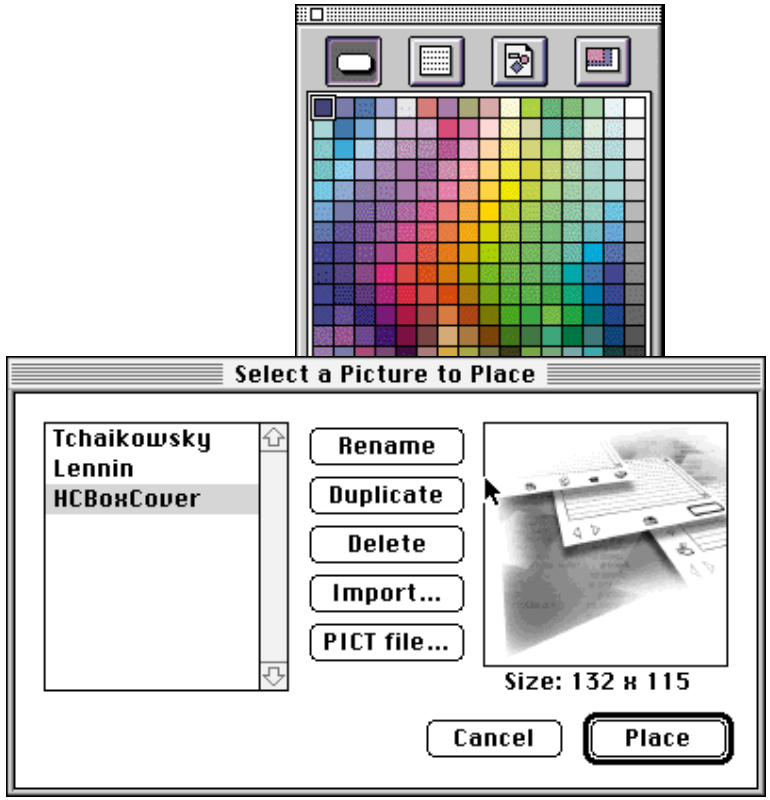
# Animation Software on the Alto

# Hypertext Authoring Tools

- Visual design of user interface, integration of media (images, sound):
  - 1982, Peter Brown (Kent): Guide authoring system
  - 1987, Bill Atkinson (Apple): HyperCard authoring system (*HyperTalk* scripting)



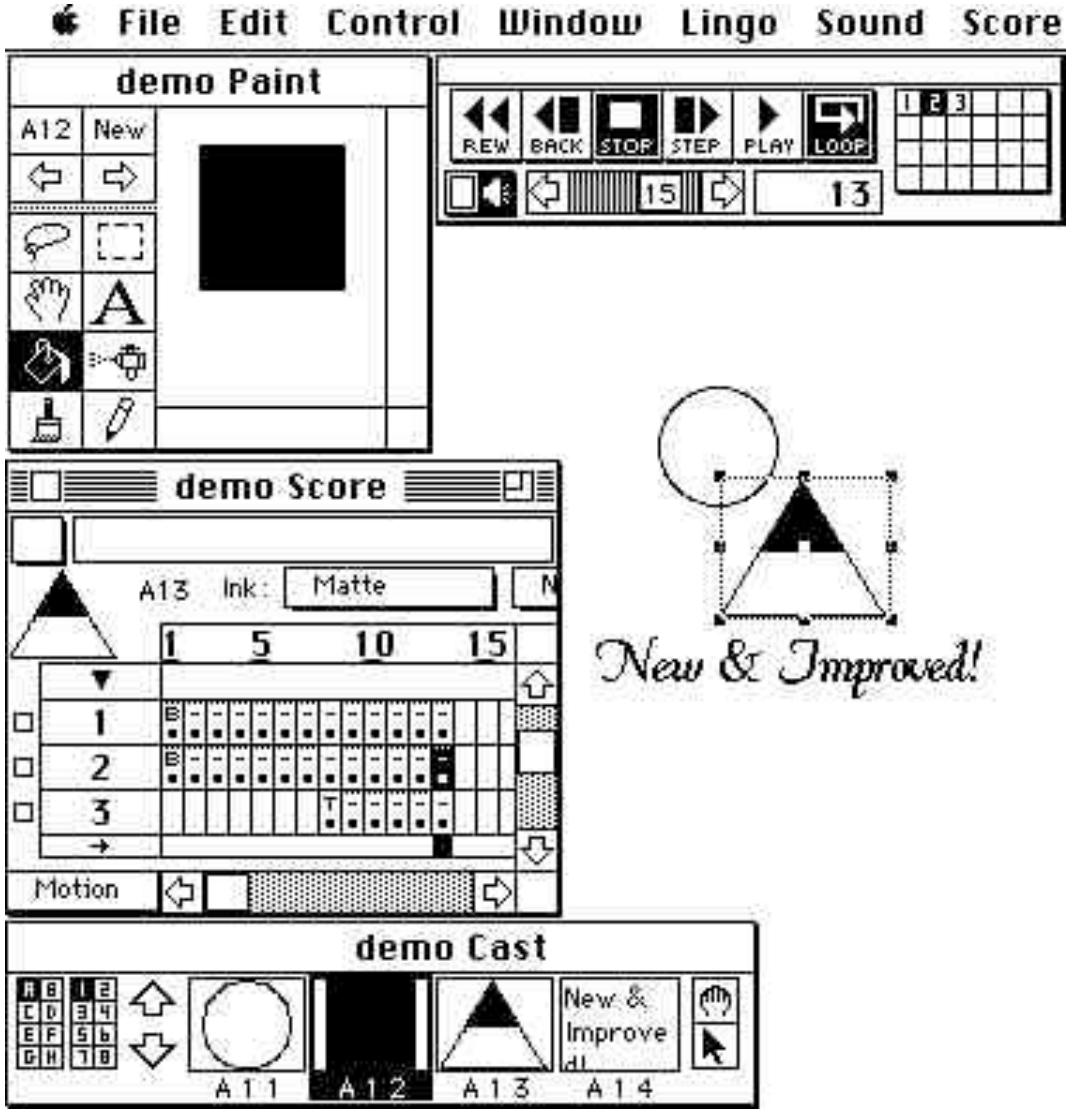

MultimediaHyperCard stack (Voyager 1989)
(Source for image: wapedia.mobi)




(Source for images: mactech.com)

# Animation Authoring: VideoWorks

- Joe Sparks
- Macromind, 1985-88
- Later renamed to *Director*
- Used (for example) for multimedia tutorials on Apple MacOS
- Specialized scripting language *Lingo*

# Timeline of Multimedia Programming History

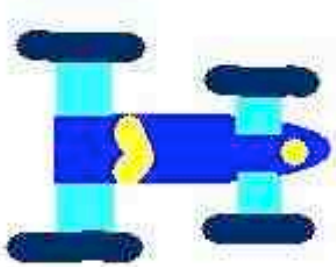
- 1963 – Sutherland: Sketchpad
- 1968 – Engelbart: NLS
- 1972 – Kay: Dynabook, Smalltalk
- 1979 – Xerox PARC: Alto
- 1982 – Brown: Guide authoring system
- 1985 – Sparks: VideoWorks
- 1987 – Atkinson: Apple HyperCard
- 1988 – Macromind Director
- 1989 – Kretz: Start of work on MHEG
- 1990s – Various multimedia education and gaming applications (CD-ROM)
- 1995 – Kay/Ingals/Kaehler: Squeak
- 1998 – W3C: SMIL
- 1997 – Macromedia Flash (ex *FutureSplash Animator* ex *SmartSketch*, by J. Gay)
- 2001 – Reas/Fry: Processing
- 2004 – ISO: MHEG-5
- 2004 – Adobe Flex
- 2004 – Bederson/Grosjean/Meyer: Piccolo framework
- 2005 – Oliver: F3 (later called JavaFX)
- 2007 – Microsoft Silverlight

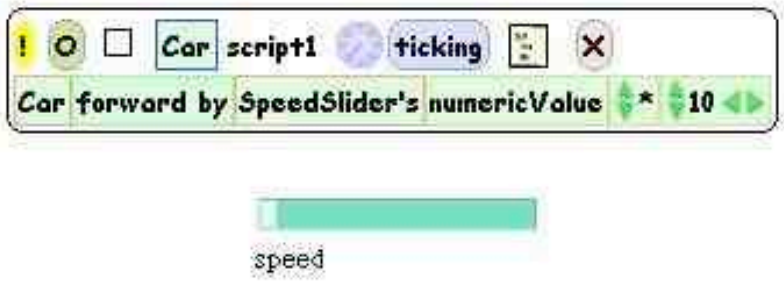
# Visual Multimedia Programming in Squeak

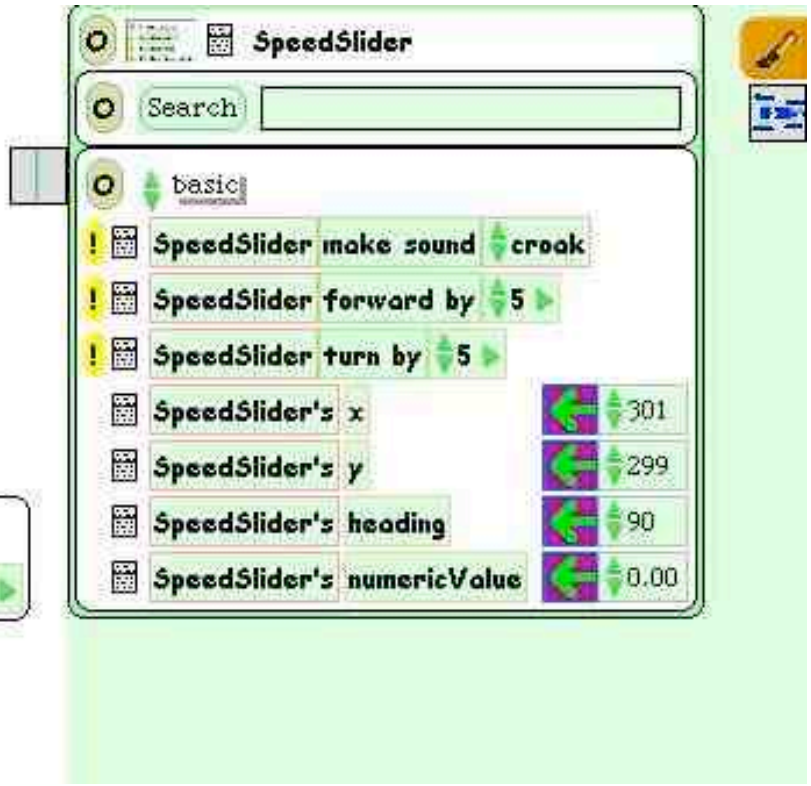
- 1995: Alan Kay, Dan Ingalls, Ted Kaehler at Apple
- Reintroducing multimedia features into Smalltalk
- Programming environment targeted at children (primary school level)


"Halo" menu

Visual scripts





# 1 Introduction to Multimedia Programming



# The Challenge for Multimedia Develoment 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 streaming, caching, …
  - Synchronization issues
    - » E.g. 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
- Development environments
  - Try to hide much of the complexity (using standard patterns and libraries)
  - Development environment accessible to non-technical people

# Categories of Development Environments

Engels and Sauer 2002:

- Frameworks and APIs (Application Programmming Interfaces)
  - Based on traditional programming and scripting languages
  - Description of control flow (execution order)
  - Development of text with editors and IDEs
- Declarative Languages
  - Specialized languages for multimedia
  - Description of multimedia documents
  - Development of text with editors and IDEs (often XML-based)
- Authoring Tools
  - Specialized development environments
  - Visual metaphors for document/application features
  - Drag & drop, property editors, drawing tools etc.

# Examples for Development Environments

- Frameworks and APIs
  - DirectX
  - Java Media Framework
  - Piccolo (for C++ and Java)
  - **Pygame (for Python)**
- Declarative languages:
  - SMIL
  - **OpenLaszlo**
  - MHEG-5
- Authoring tools:
  - Adobe Director
  - **Adobe Flash**
- Hybrid approaches:
  - Declarative language integrated with framework: **JavaFX**
  - Specialization of programming language & framework: **Processing**

# Example & Outlook: Varying Development Tools

- Example application:
  - Simple slide show, showing a sequence of bitmap pictures (photos)
- Same application behaviour and appearance
- Different development environments
- Will be further analyzed in future lectures…

# Language & Framework: Python & Pygame

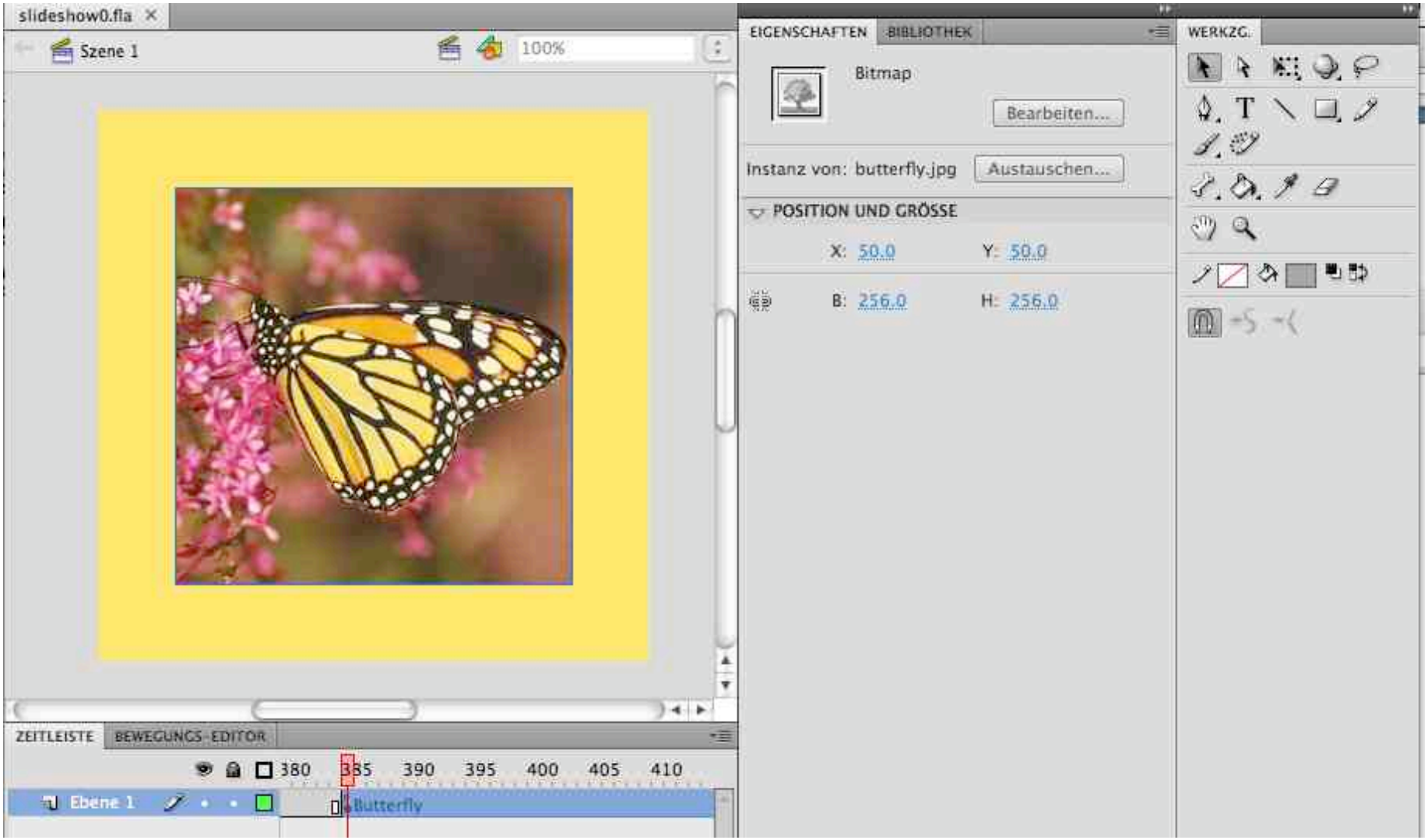
```
import pygame
from pygame.locals import *
from sys import exit

background = pygame.Color(255,228,95,0)
sc_w = 356
sc_h = 356

pygame.init()

# Create program display area
screen = pygame.display.set_mode((sc_w,sc_h),0,32)
pygame.display.set_caption("Simple Slide Show")

# Set background color by drawing a rectangle
pygame.draw.rect(screen,background,pygame.Rect(0,0,sc_w,sc_h),0)

# Load slide and show it on the screen
slide = pygame.image.load('pics/tiger.jpg').convert()
screen.blit(slide,(50,50))
pygame.display.update()
pygame.time.wait(4000)
```

...

# Declarative Language: OpenLaszlo

```
<canvas bgcolor="#FFE45F" width="356" height="356">
  <resource src="pics/tiger.jpg" name="img1"/>
  <resource src="pics/elephant.jpg" name="img2"/>
  <resource src="pics/jbeans.jpg" name="img3"/>
  <resource src="pics/peppers.jpg" name="img4"/>
  <resource src="pics/butterfly.jpg" name="img5"/>

  <view name="slide" x="50" y="50" resource="img1"
  oninit="canvas.changeSlides()"/>

  <method name="changeSlides">
      lz.Timer.addTimer(new LzDelegate(this, "change1"), 4000);
      lz.Timer.addTimer(new LzDelegate(this, "change2"), 8000);
      lz.Timer.addTimer(new LzDelegate(this, "change3"), 12000);
      lz.Timer.addTimer(new LzDelegate(this, "change4"), 16000);
  </method>

  <method name="change1">
      slide.setAttribute("resource","img2");
  </method>
  <method name="change2">
      slide.setAttribute("resource","img3");
  </method> ...

</canvas>
```

# Authoring Tool: Adobe Flash (CS4)



# References


- S. Boll, “Zyx - towards flexible multimedia documnt models for reuse and adaptation”. Dissertation Technische Universität Wien, 2001 http://medien.informatik.uni-oldenburg.de/index.php?id=31
- G. Engels, S. Sauer, “Object-oriented Modeling of Multimedia Applications”, in *Handbook of Software Engineering and Knowledge Engineering*, S.K. Chang (ed.), Singapore: World Scientific 2002, pp. 21-53.
- R.L. Fetterman, S, K. Gupta, *Mainstream Multimedia: Applying Multimedia in Business.* New York: Van Nostrand Reinhold 1993
- R.S. Heller, C.D. Martin, N. Haneef, S. Gievska-Krliu, “Using a theoretical multimedia taxonomy franework”, *J. Educ. Resour. Comput.*, vol. 1, p. 6, 2001
- R. Steinmetz, K. Nahrstedt, *Multimedia Applications*, 1st ed. Berlin: Springer 2004