4 Overview on Approaches to Multimedia Programming

4.1 History of Multimedia Programming

4.2 Squeak and Smalltalk: An Alternative Vision

4.3 Director and Lingo: Advanced Multimedia Authoring

An introductory example

Lingo scripting
Director behaviours

4.4 Frameworks for Multimedia Programming

Literature:
http://www.lingoworkshop.com/
(In German:) T. Hauser, C. Wenz, Director MX 2004
Einstiegerseminar, bhv-Verlag 2004
(In German:) M. Eberl, J. Jacobsen: Director MX and Lingo,
Markt+Technik 2003

Director: History

- 1984: *Macromind* (Jamie Fenton, Marc Carter, Mark Pierce)
- 1985: *VideoWorks* for “Guided Tour” to Apple Macintosh OS
- 1987: *VideoWorks II* (colour)
- 1988: *VideoWorks interactive*, renamed to *Director*
  - John Thompson, Erik Neumann: Object-oriented scripting language *Lingo*
  - Used to create the displays for “Star Trek TNG”
- until 1997: Mainly used for multimedia CD-ROMS (games, infotainment)
- 1998: Director as part of the “Shockwave Internet Studio”
- 2000: Macromedia puts strong effort into the Flash platform
- 2004: Director MX 2004 supports JavaScript syntax as alternative to Lingo
- 2006: Adobe still selling Director MX 2004
Pictures, Sprites and Cast

• Cast members may be of various kinds (media elements):
  – Bitmap, vector graphics, sound, …
• A sprite is an instance of a cast member
• A picture contains a number of sprites (those currently on stage)
• A score (Drehbuch) is like the timeline in Flash
  – Lifetime of sprites can be easily adjusted by mouse movements
Tweening

- Tweening essentially works like in Flash
  - Key frames need to be inserted to modify sprite
  - Alternatively whole sprite can be converted to single pictures
- Details of sprite tweening:

Effect Channels

- **Effect channels** apply specific effects to the whole stage
  - Shown above the sprite channels, have to be made visible explicitly
- **Transition channel (Übergangskanal)**
  - Applies transitions like in movie editing
- **Tempo channel (Tempokanal):**
  - Pause, waiting for a cue point in some continuous media, waiting for user input, ...
- **Colour palette channel (Farbpalettenkanal):**
  - Defining a colour palette, creating colour effects
- **Sound channels (Tonkanäle):**
  - Two channels only
- **Scripting channel (Skriptkanal):**
  - See below
Motion Tweening

- Very similar to Flash but easier
  - Each sprite has a default registration point for a motion path
  - Drawing motion paths is straightforward
  - Key frames used to reshape motion path
Lingo Interpreter

• There is a “message” window which allows a direct dialogue with the Lingo programming/scripting language

The Lingo Paradigm

• Lingo is very much inspired by “HyperTalk” (Apple)
• All programming is programming event handlers
• There is no main program
  – Effectively the event handler of “prepareMovie” is kind of a main program
• Program code is only meaningful together with project file of the authoring system
  – No stand-alone programs
• All code is scattered over the project
Event Handler

- Running animations can be influenced by event handler:
  Lingo: "on mouseUp go to frame X end"

Types of Lingo Scripts

- Hierarchical search for scripts (in this order):
  - Score scripts (behaviours)
    » Executed when respective sprite is active or when a specific frame is shown
    » (object scripts) contain keyword "me"
  - Cast scripts
    » Once defined for all instances
    » (class scripts)
  - Movie scripts
    » Global for the whole movie

- If no script is found, execution continues
- If a script is found, more general scripts are ignored
- Practical hint for working with scripts in Director:
  - Use Script window and adjust script type
Local and Global Variables in Lingo

- By default, all variables are local in procedure
- Variables can be declared global
  - but then are global for whole movie

- Example for global variables:

```
global myCounter
on mouseUp
  set myCounter = myCounter + 1
  put myCounter
end
```

Object-Orientation in Director:
“Parent-Child Programming” (1)

- “Parent script” (class):

```
property pVorname, pNachname

on new me
  return me
end

on fill me, vorname, nachname
  pVorname = vorname
  pNachname = nachname
end
```
Object-Orientation in Director: “Parent-Child Programming” (2)

- Global script (film script):

  ```fps
  global lUsers
  on prepareMovie
    lUsers = []
  end

  on fillOut
    temp = new(script "parent script")
    fill(temp, member("vorname").text, member("nachname").text)
    append(lUsers, temp)
    clearFields
  end

  on clearFields
    member("vorname").text = ""
    member("nachname").text = ""
  end
  ```

  vorname, nachname are text input fields

Object-Orientation in Director: “Parent-Child Programming” (3)

- Local script for enter button:

  ```fps
  on mouseUp
    fillOut()
  end
  ```
4 Overview on Approaches to Multimedia Programming

4.1 History of Multimedia Programming

4.2 Squeak and Smalltalk: An Alternative Vision

4.3 Director and Lingo: Advanced Multimedia Authoring
   An introductory example
   Lingo scripting
   Director behaviours

4.4 Frameworks for Multimedia Programming

Literature:
http://www.lingoworkshop.com/
(In German:) T. Hauser, C. Wenz, Director MX 2004 Einsteigerseminar, bhv-Verlag 2004
(In German:) M. Eberl, J. Jacobsen: Director MX and Lingo, Markt+Technik 2003

Behaviours in Director

- Lingo programming is too tedious for many developers
- Pre-fabricated “behaviour” blocks (similar to software components)
- See Window->Library, several categories
- Application developer works by drag&drop
  - Select behaviour
  - Drag it onto object creating the firing event
- Example: Sound playing behaviour
  - Create an empty film
  - Create a button (“music”)
  - Import some music
  - Look for “play sound” behaviour in library
  - Drag it onto button
  - This is equivalent to a lot of Lingo programming!
Director & Lingo: Summary

- Lingo language was very innovative at its time but is outdated now
  - Syntax was modernized
    » "the xy of abc" replaced by "abc.xy"
  - JavaScript alternative being introduced
- Conceptually very similar to Flash
  - Fusion of both programs? (Ask the Adobe managers...)
- Nice ideas:
  - Effect channels
  - Drag & drop behaviours
- Interactivity only at runtime
  - Very restrictive compared e.g. to Squeak’s mixture of development time/runtime interaction
- Professional support for wide range of media and large built-in library
- An alternative for nowadays’s development tasks but not a model for the future...

Lessons Learnt from Comparison Flash/Director

- Varying degrees of automation
  - High degree of automation does not guarantee success when hidden in a complex overall system structure
- Exotic programming languages
  - Programmers are conservative: Prefer known structures
- Authoring usability
  - Authors can live with historically grown usage patterns