Appendix C

Board-Event Encoding

This appendix provide a summary of the E-Chalk board event format and its mapping to MPEG-4 BIFS.

C.1 The E-Chalk Board Format

The E-Chalk board-event format is thoroughly described in [Knipping, 2005], Section 4.11. This section only provides a short overview. The events are described by a simple line-based, human-editable ASCII format. After a few header lines, that provide version information, specify the resolution of the board, the lecture title, and the background color, every event is stored in a separate line in the following syntax:

```
<timestamp>"$"<event>["$"<arguments>*]
```

The `timestamp` is the hexadecimal-coded amount of milliseconds that have gone by since starting the lecture. The dollar character serves as token delimiter. After the mandatory name of the event, a number of parameters can be passed to the event, again delimited by the dollar sign. E-Chalk knows the following events:

- `Nop` This event can be used to add comments. The event as well as any string passed as argument is ignored.
- `RemoveAll` This event clears the entire board and sets the board position back to the beginning. It takes no arguments.
- `Undo` This event triggers the undo manager to undo the last set of strokes. The event is inserted when the user pushes the respective button in the board toolbox.
- `Redo` This event is the inverse of `undo`.
- `Terminate` The command is only used in live transmissions to signal termination of the transmission.
- `Scrollbar` This event takes one integer parameter that specifies the new vertical offset of the board's top position.

- `Form` This event marks that something is actually drawn on the board. The next parameter specifies what exactly:
  - `Line` This event takes the arguments $\$x_0\$y_0\$x_1\$y_1\$r\$c$ and triggers the drawing of a line segment from point (x_0, y_0) to point (x_1, y_1) with stroke radius r and color c.
  - `Image` This event gets an *id* and two coordinates and inserts image number *id* at the specified position. A forth argument specifies whether the inserted image may be updated. Images are tagged as updatable if they actually show screenshots from a Java Applet inserted into the board.
  - `Text` This event takes a MIME-encoded [Freed and Borenstein, 1996a] string, two coordinates, a color, and a font size. After this event, one of several possible events can follow:
    1. `Text$End` This event signals that the text can be put directly on the board. The events follows directly after a `Form$Text` event if the text was printed out automatically, for example the response of a CGI-script.
    2. `Text$Char` The event takes a character as first argument. A set of these events is used between a `Form$Text` event and a `Text$End` event if the text is typed by the user. When the user presses a certain key on the keyboard the event is inserted. Special characters, like `backspace` or `delete` have their own subevents.
    3. `Text$SetTxt` and `Text$Str` Both events take strings as first arguments and are used when the user sets an entire text line at once using the text history (cursor up and down) or pastes text at the current cursor position.
    4. `Text$Cursor` Takes a position as first argument and is used for cursor movement during string input.
- `Image$Update` This event can happen any time after `Form$Image`. It takes two arguments: id_1 and id_2. Applets that are inserted into the board are played back as consecutive screenshots. The command triggers a replacement of the image with id_1 with the image having id_2.

C.2 Mapping E-Chalk Events to MPEG-4 BIFS

This section shows two examples of mapping the E-Chalk board format to the BIFS text format as described in Section 5.4. A more detailed description can be found in [Jankovic et al., 2006].

Scene Definition

After the obligatory `InitialObjectDescriptor` (not shown here), the initial scene containing the board, audio, and video is defined as follows:

```
# Root of the scene tree
DEF Root OrderedGroup {
       children [
```

```
                Background2D {
                        backColor 0.0 0.0 0.0
                }
                # Define hook for audio node
                Sound2D {
                        source AudioSource {
                                # Object descriptor with ID 3 is defined below
                                url [od:3]
                        }
                }
                # empty board
                DEF BOARD Transform2D {
                        translation 0 0
                        children [
                        ]
                }
                # Define hook for video node (for overlaid replay)
                Transform2D {
                        translation 0 20 # Video offset to better match instructor
                        children [
                                Shape {
                                        appearance Appearance {
                                                # Transparency settings for the video
                                                material DEF M1 MaterialKey {
                                                        keyColor 0 0 0
                                                        lowThreshold 0.1
                                                        transparency 0.1
                                                }
                                                texture MovieTexture {
                                                # Object descriptor with ID 4 is
                                                # defined below
                                                        url [od:4]
                                                }
                                        }
                                        # Video isn't scaled now but could be.
                                        geometry Bitmap{
                                                scale 1.0 1.0
                                        }
                                }
                        ]
                }
        ]
}
```

Stroke Encoding

The first line segment of a stroke (i. e., when a line segment's starting point is not connected to the ending point of the last line segment)

```
3e8f$Form$Line$17c$aa$17c$ab$ff00ff00$3
```

is encoded as BIFS Text as follows (the coordinate systems of E-Chalk and MPEG-4 differ, so the coordinates have to be translated):

```
AT 16015 {
# at timestamp 16015 (after 16 seconds)
       APPEND TO BOARD.children DEF STR1 OrderedGroup {
                children [
                        # Beginning circle
                        # (optical correction to simulate pen down)
                        Transform2D {
                                translation -81 129
                                children [
                                        Shape {
                                                appearance DEF APP1 Appearance {
                                                        material Material2D {
                                                                emissiveColor 0.0 1.0 0.0
                                                                filled TRUE
                                                        }
                                                }
                                                geometry Circle {
```

```
                                        radius 1
                                }
                        }
                ]
        }
        # Line segment
        # Beginning of stroke starts with first line segments
        Shape {
                appearance Appearance {
                        material Material2D {
                                lineProps LineProperties {
                                        lineColor 0.0 1.0 0.0
                                        width 3
                                }
                        }
                }
                geometry DEF STROKE1 IndexedLineSet2D {
                        coord DEF STROKEP1 Coordinate2D {
                                point [-81 129 -81 129]
                        }
                }
        }
        # (provisional) ending circle (simulate pen up)
        DEF ENDCIRCLE1 Transform2D {
                translation -81 129
                children [
                        Shape {
                                appearance USE APP1
                                geometry Circle {
                                        radius 1
                                }
                        }
                ]
        }
    ]
  }
}
```

Further line segments that belong to the stroke are appended consecutively. For example, the line segment

```
3e9f$Form$Line$17c$ab$17c$ad$ff00ff00$3
```

is appended to the last line segment as follows

```
AT 16031 {
       REPLACE ENDCIRCLE1.translation BY -81 127
       APPEND TO STROKEP1.point -81 127
}
```

until the next line segment does not belong to the same stroke (i. e., it is not connected).

Scroll events

Scroll events are easily implemented by translating the board scene. For example the event:

```
2717f$Scrollbar$b
```

is encoded as:

```
AT 160127 {
       REPLACE BOARD.translation BY 0 11
}
```