

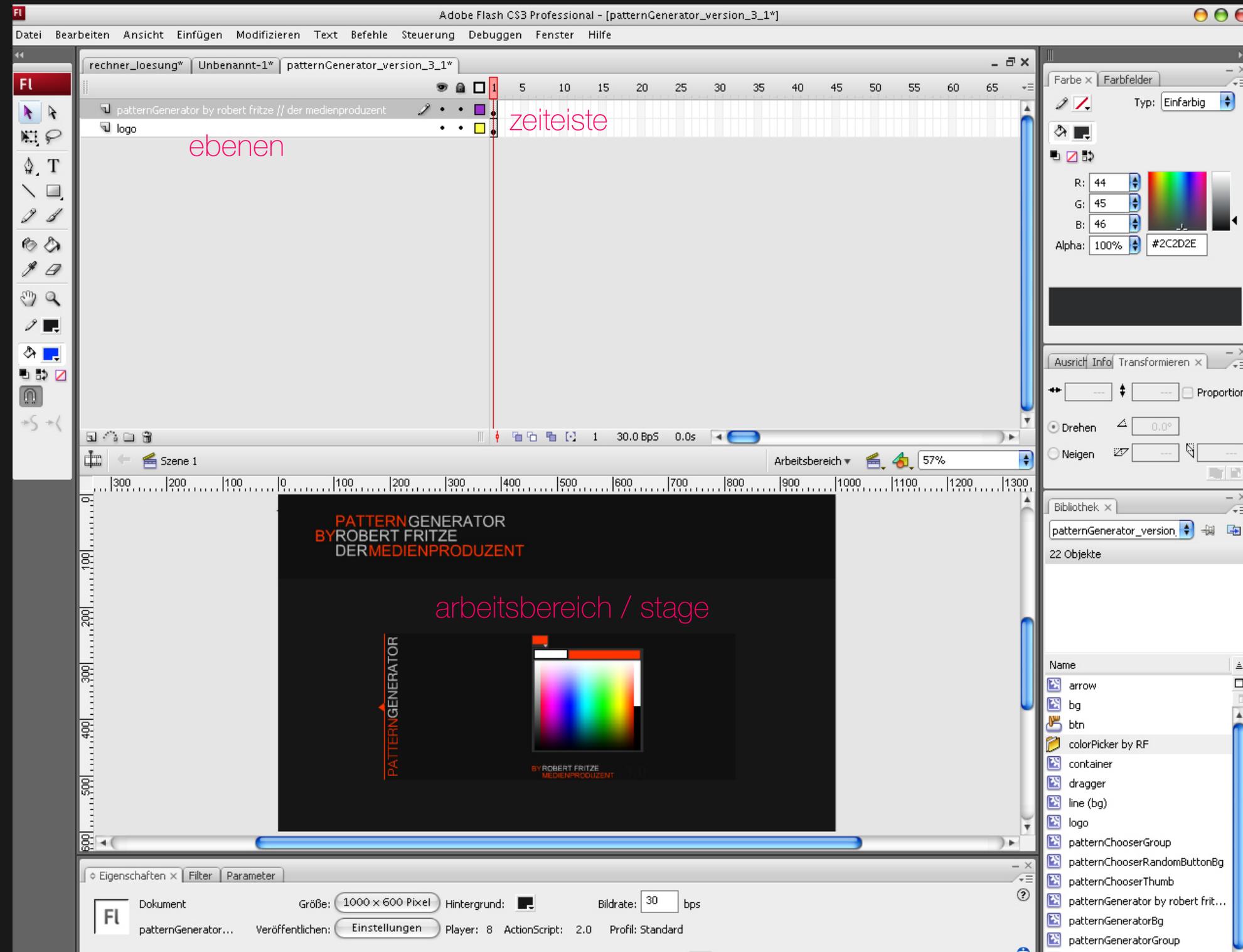
- I allgemeines
- II oberfläche und werkzeug
- III einführung as3

- entstanden 1996
- zum entwickeln von
 - a) internetanwendungen
 - b) spielen
 - c) interfacevisuals
- plattformunabhängig (win, mac)
- kleine dateien (swf = small web format)
- streamingfähig (während des ladens wird schon abegespielt)
- schnelle und komplexe entwicklung möglich
- nahezu alles kann eingebettet werden (vektoren, bilder, sound, videos)

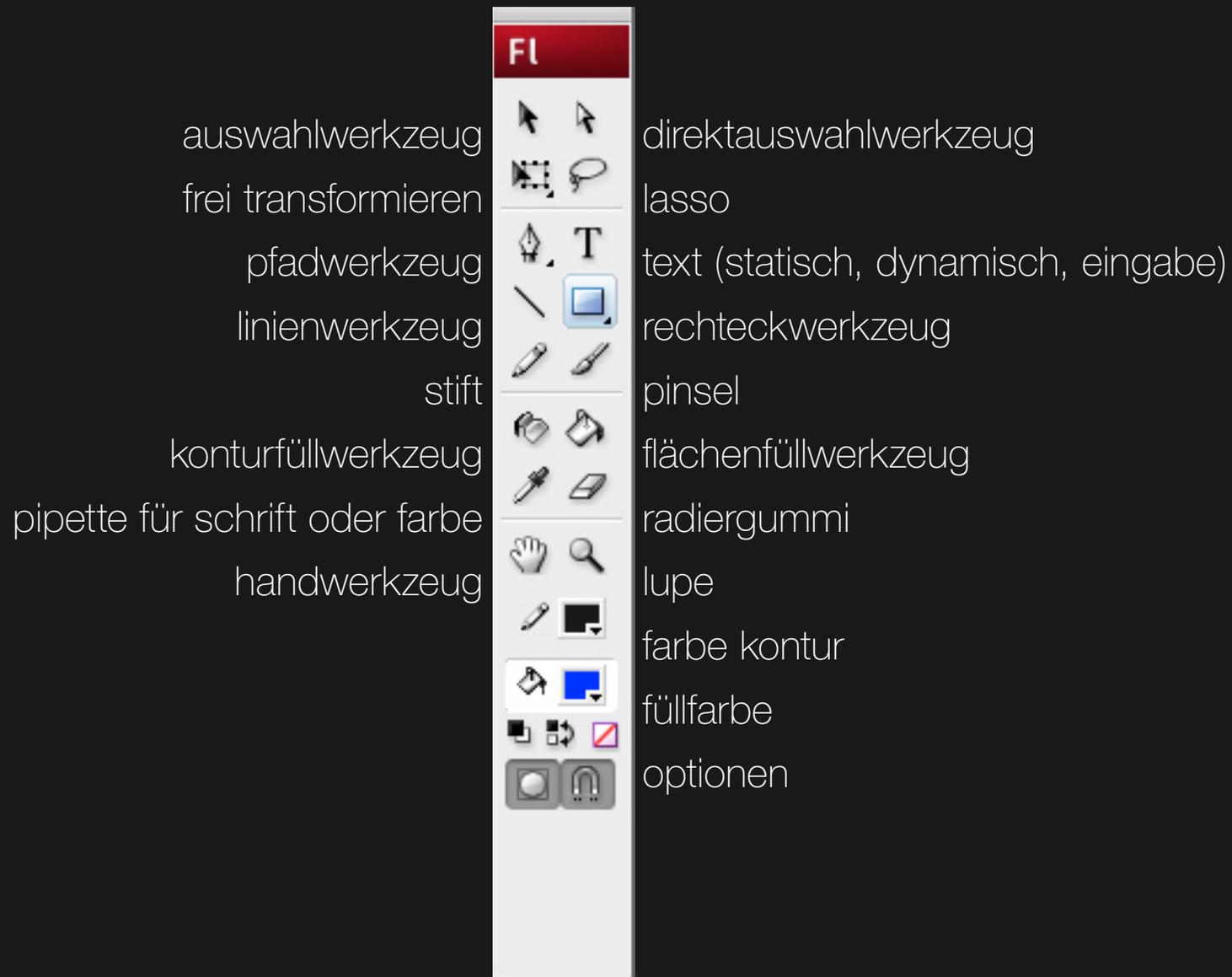
Adobe Flash Player Version Penetration

Worldwide Ubiquity of Adobe Flash Player by Version — March 2008

| | Flash Player 7 | Flash Player 8 | Flash Player 9 | Flash Player 9.0.115 ¹ |
|-----------------------------|----------------|----------------|----------------|-----------------------------------|
| Mature Markets ¹ | 98.8% | 98.5% | 97.2% | 61.8% |
| US/Canada | 98.7% | 98.5% | 97.3% | 62.1% |
| Europe ² | 98.5% | 98.1% | 96.5% | 61.9% |
| Japan | 99.8% | 99.5% | 98.0% | 61.0% |



II. Oberfläche u Tools
werkzeuge



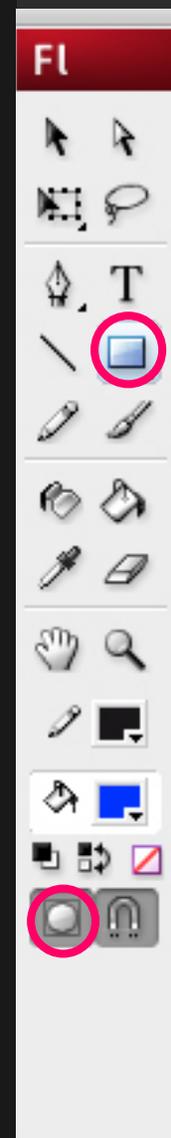
II. Oberfläche u Tools

werkzeuge

rechteck

wenn modus objektzeichnung (unterer kreis oder taste „j“) aktiviert:

- zeichnen von objekten
- überlagernde bereiche bleiben separat (eigene objekte)
- objekte bleiben somit erhalten



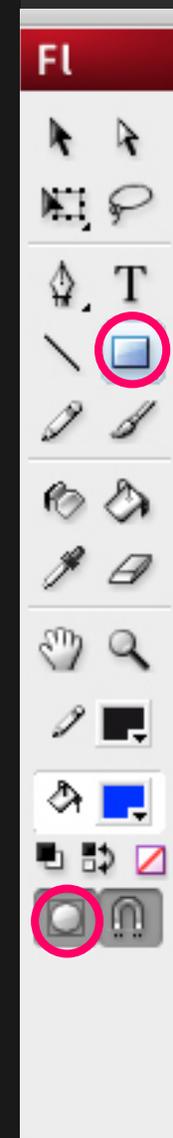
II. Oberfläche u Tools

werkzeuge

formen

wenn modus objektzeichnung (unterer kreis oder taste „j“)
nicht aktiviert:

- zeichnen von shapes
- überlagernde bereiche werden vermischt/beschnitten
- objekte können somit beschnitten und getrennt werden



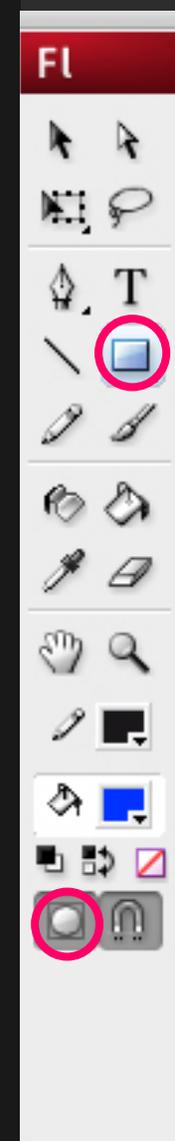
II. Oberfläche u Tools

werkzeuge

formen

wenn modus objektzeichnung (unterer kreis oder taste „j“)
nicht aktiviert:

- zeichnen von shapes
- überlagernde bereiche werden vermischt/beschnitten
- objekte können somit beschnitten und getrennt werden



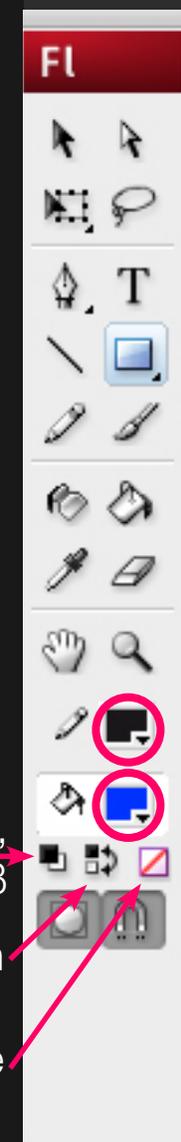
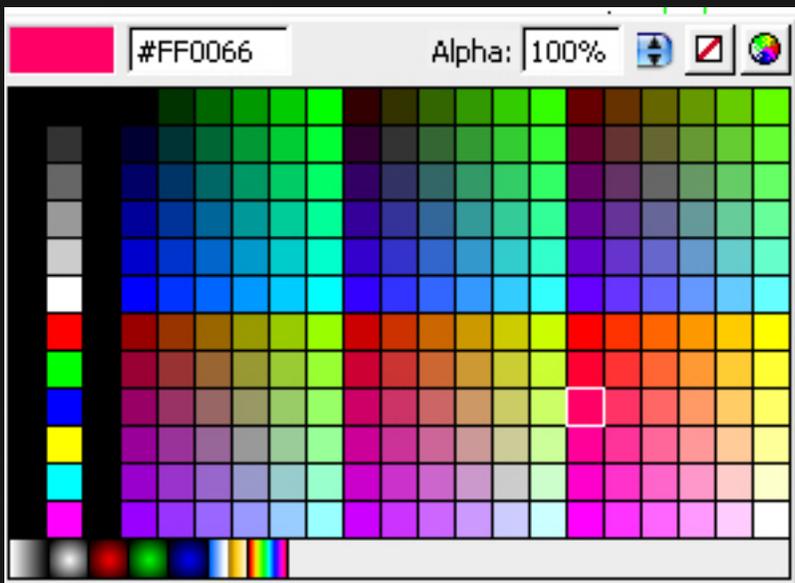
II. Oberfläche u Tools

werkzeuge

formen

farbe

- objekte und shapes haben immer kontur und füllung
- kann separat abgeschaltet werden
- separat kann der alphawert geregelt werden (farbmenü)
- farben können getauscht werden
- es werden webfarben verwendet



kontur
füllung

kontur schwarz,
füllung weiß

farben tauschen

keine farbe

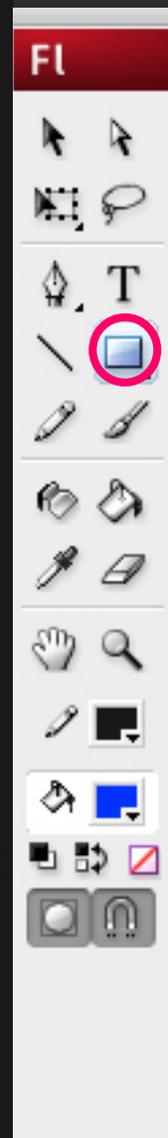
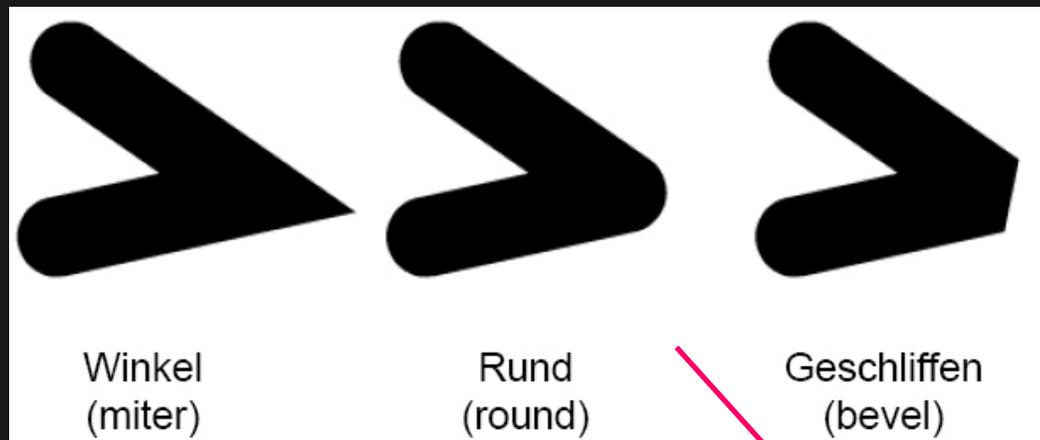
II. Oberfläche u Tools

werkzeuge

formen

zeichenmodi

- einstellung von kontur/linienfarbe
- strichstärke kontur
- art der kontur (haarlinie, normal, gestrichelt)
- rundung der ecken des objekts kann bestimmt werden
- abschluss (ende der linie, werte: rund oder quadratisch)
- linien, shapes können verschiedene verbindungen haben:



II. Oberfläche u Tools

werkzeuge

formen

zeichenmodi

rechteckwerkzeug

- mit gedrückter „shift“-taste wird quadrat erzeugt
- farbe, kontur kann ausgewählt werden
- rundung der ecken NUR VOR ZEICHNEN möglich

werkzeug für rechteckgrundform

- wie rechteckwerkzeug
- rundung der ecken kann nachträglich verändert werden

polysternwerkzeug

- hier wird ein vieleck gezeichnet
- optionen: stern oder vieleck
- anzahl der seiten muss vorher definiert werden (eigenschaften --> werkzeugoptionen)



II. Oberfläche u Tools

werkzeuge

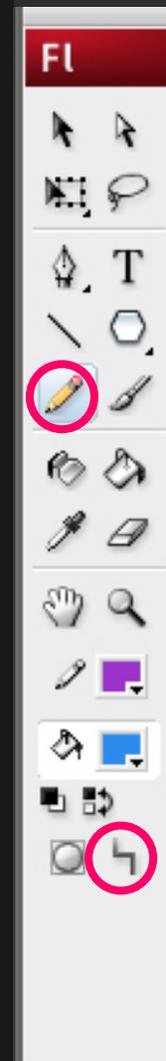
stift

stiftwerkzeug

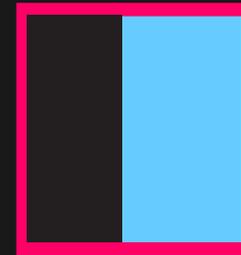
- zum freihandzeichnen

optionen für stiftwerkzeug (unterer kreis)

- begradigen (kurve wird begradigt)
- glätten (kurve wird geglättet (smooth))
- freihand (3 mal dürft ihr raten ;))



ausgehend von folgendem quadrat mit rahmen
(mit 2 farbfüllungen)



einfach klick auf blaue füllung

- nur die blaue füllung wird ausgewählt

doppelklick auf blaue füllung

- blaue füllung & rahmen (aber nur um die blaue füllung) gewählt

einfach klick auf kontur

- linie der kontur bis zum nächsten „trennpunkt“ wird gewählt
(ecke oder der punkt, wo die füllung sich ändert)

doppelklick auf kontur

- gesamte kontur gewählt

auswahl kann immer durch gedrückte „shift“-taste erweitert werden

II. Oberfläche u Tools

werkzeuge

auswahlwerkzeug



II. Oberfläche u Tools

werkzeuge

unterauswahlwerkzeug

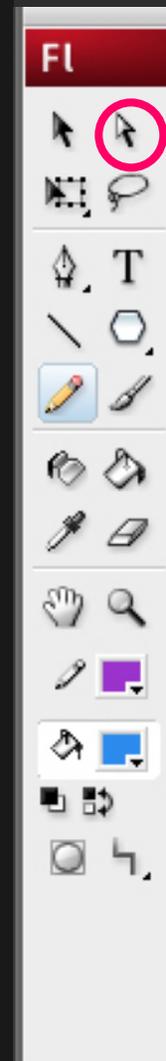
gedacht zum bearbeiten der vektoren/eckpunkte von objekten

weißes quadrat am pfeil

- einzelner oder mehrere gewählte punkte werden verschoben
- bei gedrückter „strg“ + „alt“ + „umschalt“-taste werden punkte hinzugefügt und es entstehen schnittmengen

schwarzes am pfeil

- gesamtes objekt wird verschoben



II. Oberfläche u Tools

werkzeuge

pfad, linie, pinsel

pfadwerkzeug

- zur erstellung von linien und punkten
- zum bearbeiten, hinzufügen oder löschen von ankerpunkten

linienwerkzeug

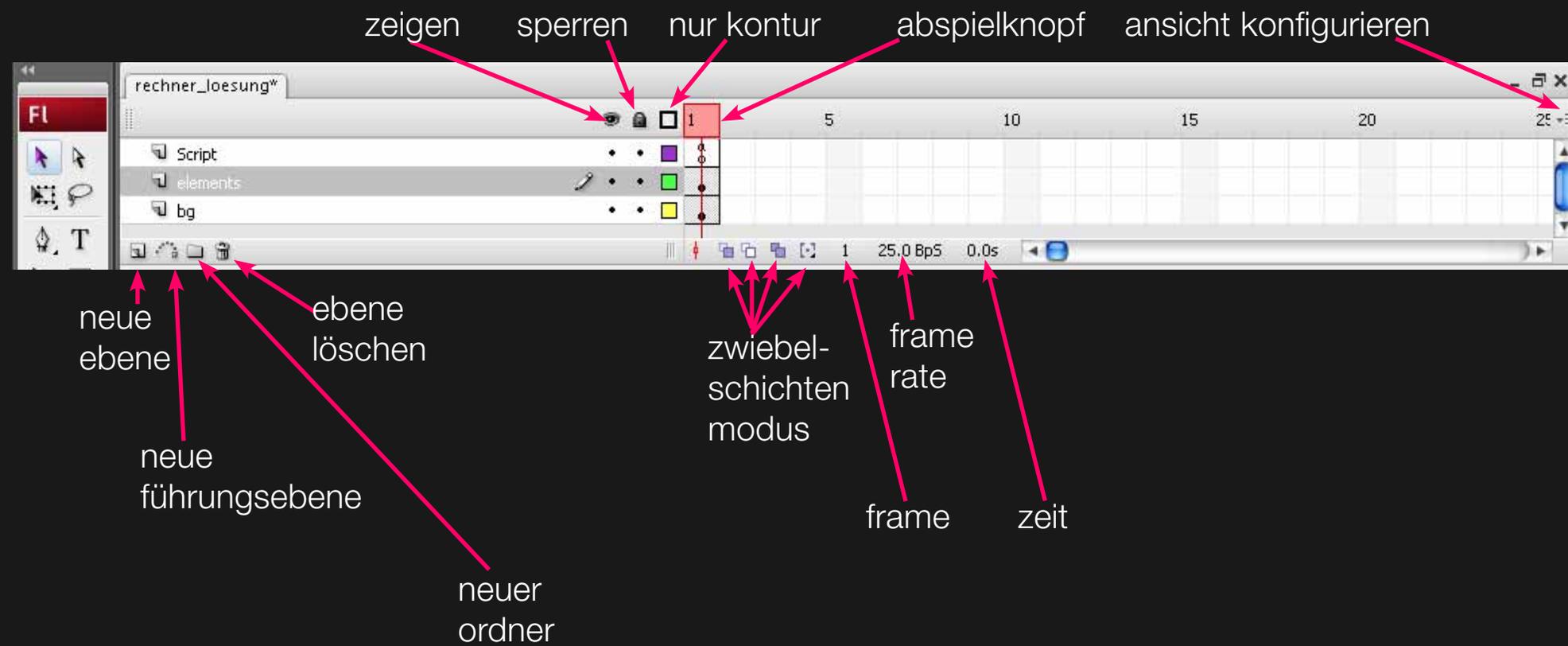
- was erstellt es wohl ?? ;)
- bei gedrückter „umschalt“-taste werden waagerechte, senkrechte oder diagonale linien gezeichnet

pinselwerkzeug

- zum freihandzeichnen
- viele optionen (pinselgröße, pinselform)
- es kann wie folgt gemalt werden:
normal, im hintergrund, füllen, auswahl füllen, innen malen
- gezeichnetes kann hinterher mit „tintenfass“ gefüllt werden



II. Oberfläche u Tools
zeitleiste



schaltflächen

- einfügen -> neues symbol (strg+f8)
 - sind eine einfache möglichkeit für interaktion
 - schnelle erstellung von rollovereffekten
- > inzwischen überholt --> mit movieclips gehts dynamischer
- objekte müssen benannt werden um sie mit action script anzusteuern
 - sollten immer wie folgt benannt werden: `buttonName_btn`

beispiel

//hier wird der eventlistener des klickens auf den button erstellt

```
unser_btn.addEventListener(MouseEvent.CLICK, weiterspringen);
```

//hier wird die funktion beschrieben

```
function weiterspringen(parameter:MouseEvent):void{  
gotoAndPlay(2);  
}
```

II. Oberfläche u Tools

movieclips

- haben eine eigene zeitleiste
- können bibliothekselemente und externe daten laden
- swf, jpg, gif und png können geladen werden
- sollten immer wie folgt benannt werden: name_mc

textfelder

- flash unterscheidet zwischen statischen-, dynamischen- und eingabetextfeldern
- bei statischem text muss text NICHT auf dem rechner, wo die swf abgespielt wird, installiert sein
- bei dynamischen text muss die schrift eingebettet werden
- sollten immer wie folgt benannt werden: name_txt

movieclips

textfelder

flash läuft normalerweise linear ab (also ein frame von oben nach unten, danach der nächste frame von oben nach unten)

action script

- unterbricht diese linearität durch: `stop()`;
- ist ereignisorientiert
- kann ab as3 nur noch in einem frame der zeitleiste oder in eine externe action-script datei geschrieben werden

listener warten quasi auf gewisse aktionen
(mouseclick, tastendruck, ende eines ladevorgangs, etc)

erstellen eines listeners

```
//auf der bühne befindet sich bereits der button beispiel_btn
```

```
stop();
```

```
beispiel_btn.addEventListener(MouseEvent.CLICK, weiterspringen);
```

```
//name button | hinzufügen des listeners | ereignis auf das reagiert werden soll | funktion die dann  
aufgerufen wird
```

funktionen sind abläufe von ereignissen die geschehen sollen.

funktionen werden durch das wort **function** eingeleitet.

danach kommt der funktionsname und in klammern dann der übergebene parameter

(hier wird das ereignis des MouseEvent.Click in die variable ereignis gespeichert)

anschließend: rückgabeparameter -> hier **void** (es wird nix zurückgegeben)

danach in den mengenklammern: die anweisung

```
function weiterspringen(ereignis:MouseEvent):void{  
gotoAndPlay(2);  
}
```

variablen

- dienen zur speicherung von informationen
- haben einen namen und einen wert
- einer variablen kann ein wert zugewiesen werden

konventionen bei der variablenverarbeitung

- variablen dürfen nur mit einem zeichen oder einem „_“ anfangen
- ab dem zweiten zeichen sind auch ziffern erlaubt, keine sonderzeichen
- flash unterscheidet groß- u. kleinbuchstaben
(„test“ und „Test“ sind 2 unterschiedliche variablen!!)
- schlüsselwörter sind nicht erlaubt (z.b. _parent, function, _x, usw)
- es gibt verschiedene typen von variablen
Number (fließkommazahlen zB 1.23)
int (ganzzahlen negativ wie positiv zB -5, -123.456, 0, 5)
uint (positive ganzzahlen)
String (zeichenketten)
Boolean (true / false (wahrheitsprüfung))
- variablen können umgewandelt werden
zb: number(irgendeiner Zahleingabe eines Textfeldes)

III as3

variablen

for schleife

III as3

- dient zur ausführung einer bestimmten aktion bis eine bedingung erfüllt ist (zB ob eine variable des typs number einen bestimmten wert erreicht hat)

for schleife

```
//allgemeine form
```

```
for(initialisierung; bedingung; aktualisierung) {  
    was soll gemacht werden;  
}
```

```
//beispiel
```

```
for(var i=0, i<=4, i++) {  
    //der mc soll auf der x position um den wert (10 * i) verschoben werden  
    beispiel_mc.x += i*10;  
}
```

thx so far :-)