

Universell nutzbarer Editor zur Erstellung naturwissenschaftlich-technischer Aufgaben für die Lernplattform Moodle

entstanden im Rahmen eines Projektes der Universität Leipzig,
gefördert vom Europäischen Sozialfond und der Sächsischen Aufbaubank

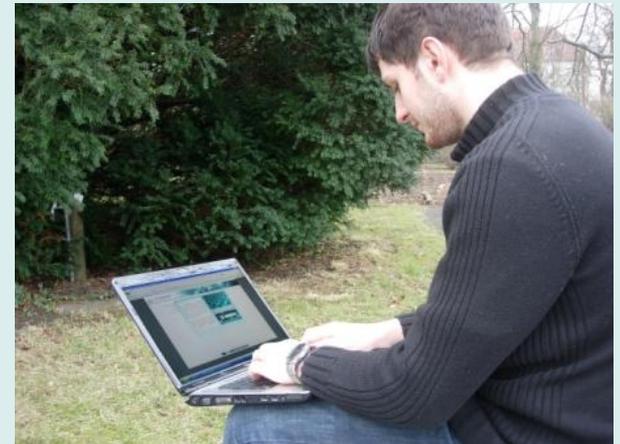
Prof. Dr. Roger Gläser, Dipl.-Ing. Ralf Moros, M.Sc. Claudia Staudte, B.Sc. Ron Gastler

Inhalt

1. Hintergrund des Projektes
2. Allgemeines
3. Editor aus Trainersicht
4. Anwendungsprogrammoberfläche für Teilnehmer
5. Moodle-Einbindung / SCORM

Hintergrund

- Universität Leipzig, Institut für Technische Chemie
- Blended-Learning-Weiterbildungskurs mit Inhalten aus den Bereichen Grundlagen der Technischen Chemie und der Verfahrenstechnik
- *Zielgruppe:*
 - Naturwissenschaftler
 - Techniker
 - Ingenieure
 - Fachfremde Akademiker



Hintergrund

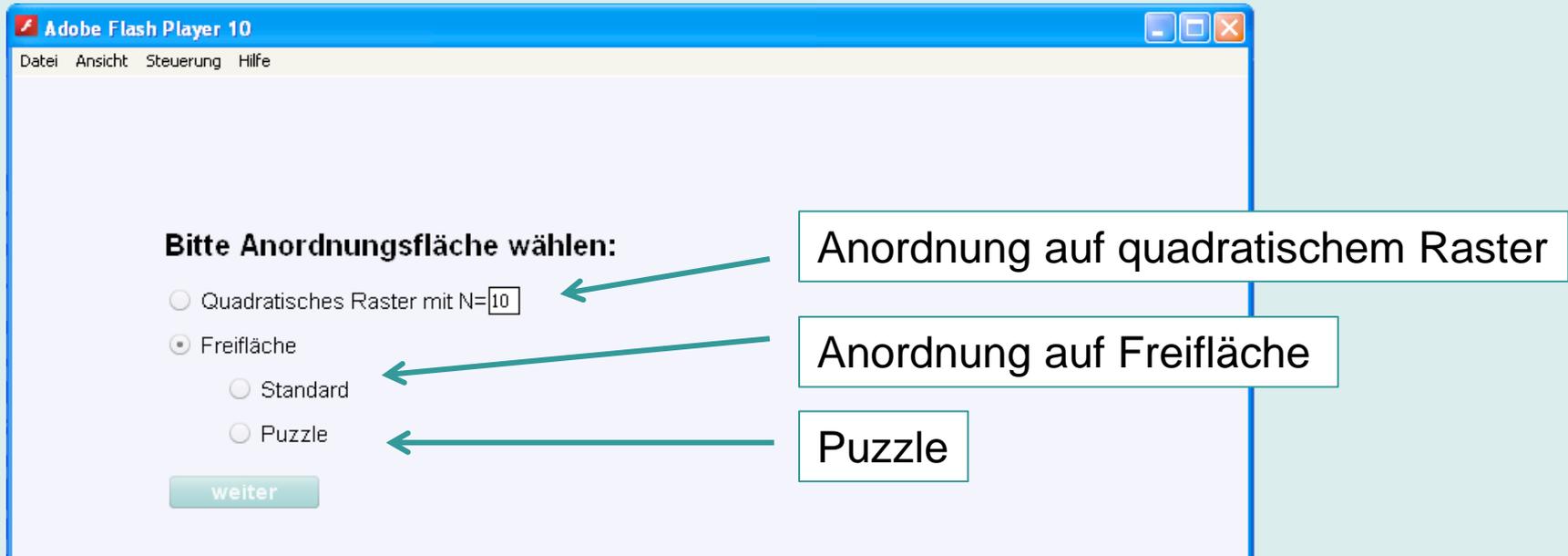
- Moodle während der e-Learning-Phase genutzt zur
 - Bereitstellung von Lernmaterialien
 - Durchführung von Online-Praktika
 - Lernerfolgskontrollen
 - Kontakt zu Betreuern
- Speziell bei Tests reichten bereits vorhandene Moodle-Module nicht aus, um alle nötigen Szenarien umzusetzen
- Lösung: im Rahmen einer Bachelorarbeit entwickelte Anwendung zur Realisierung von naturwissenschaftlich-technischen Aufgabenstellungen

Allgemeines

- Umsetzung mit Hilfe von
 - Adobe Flash CS4
 - JavaScript
 - SCORM-Editor
- Tool besteht aus 2 Programmen:
 - Editor → Erstellung von Aufgaben
Trainer
 - Anwendungsprogramm → Anwendung der erstellten Aufgaben
Teilnehmer

Editor – Hauptmenü

Trainersicht

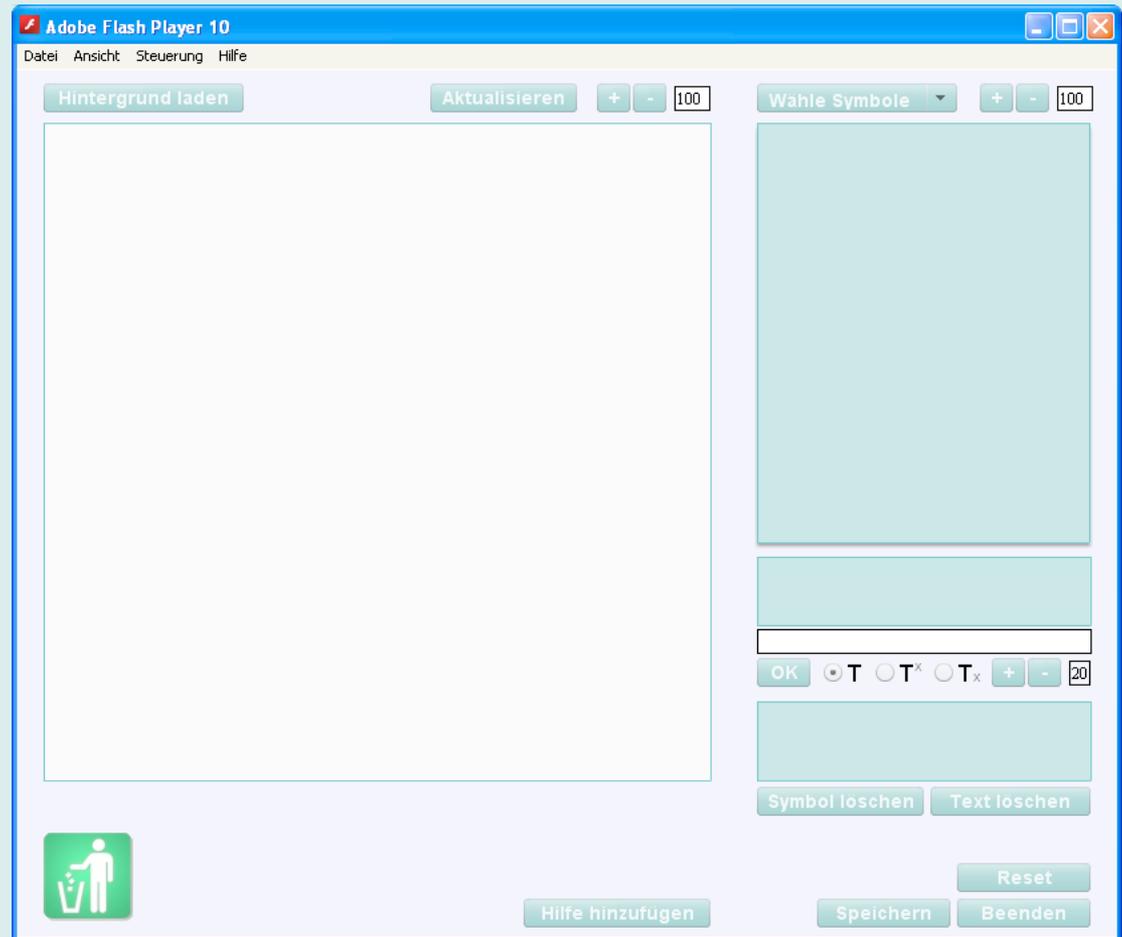


- Aufruf der exe-Datei
- Wahl zwischen den verschiedenen Aufgabentypen
- Auswahl der vorbereiteten Grafiken

Editor – Freifläche

Trainersicht

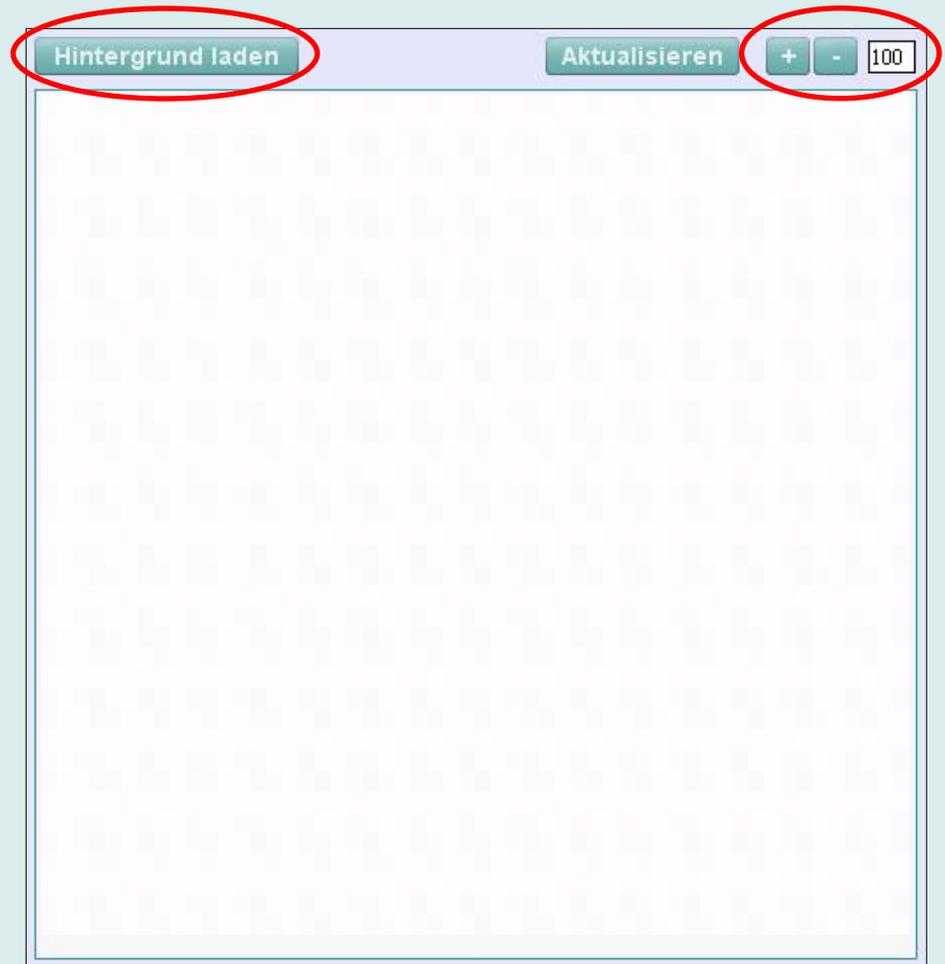
- Anordnung von Symbolen, Bildern und Texten auf einer Freifläche per Drag & Drop



Editor – Freifläche

Trainersicht

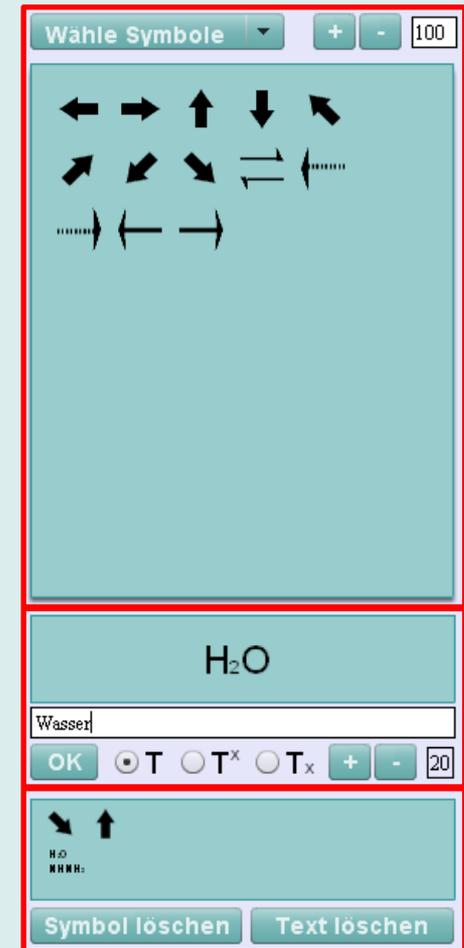
- Fläche zum Anordnen von Symbolen und Texten
- Laden der Hintergrundgrafik
- Anpassen der Hintergrundgrafik



Editor – Freifläche

Trainersicht

- Symbolauswahlcontainer zum Laden verschiedener Symbole
- Textauswahlcontainer zum Erstellen von Text
- Container zum Anordnen zusätzlicher „falscher“ Texte/Symbole



Editor – Freifläche

Trainersicht



- Papierkorb zum Löschen von Texten und Symbolen
- Hinzufügen eines Hilfetextes
- Speichern der erstellten Aufgabe
- Zurücksetzen aller Einstellungen
- Beenden des Editors

Editor – Freifläche

Trainersicht

Adobe Flash Player 10

Datei Ansicht Steuerung Hilfe

Hintergrund laden Aktualisieren + - 100

Zeichen + - 100

Diffusion 1

Diffusion 2

Diffusion 3

1 10 10² 10³ 10⁴ 10⁵

Porendurchmesser d_p in 10^{-10} m

OK T T^{*} T_x + - 20

Direction A
Direction B

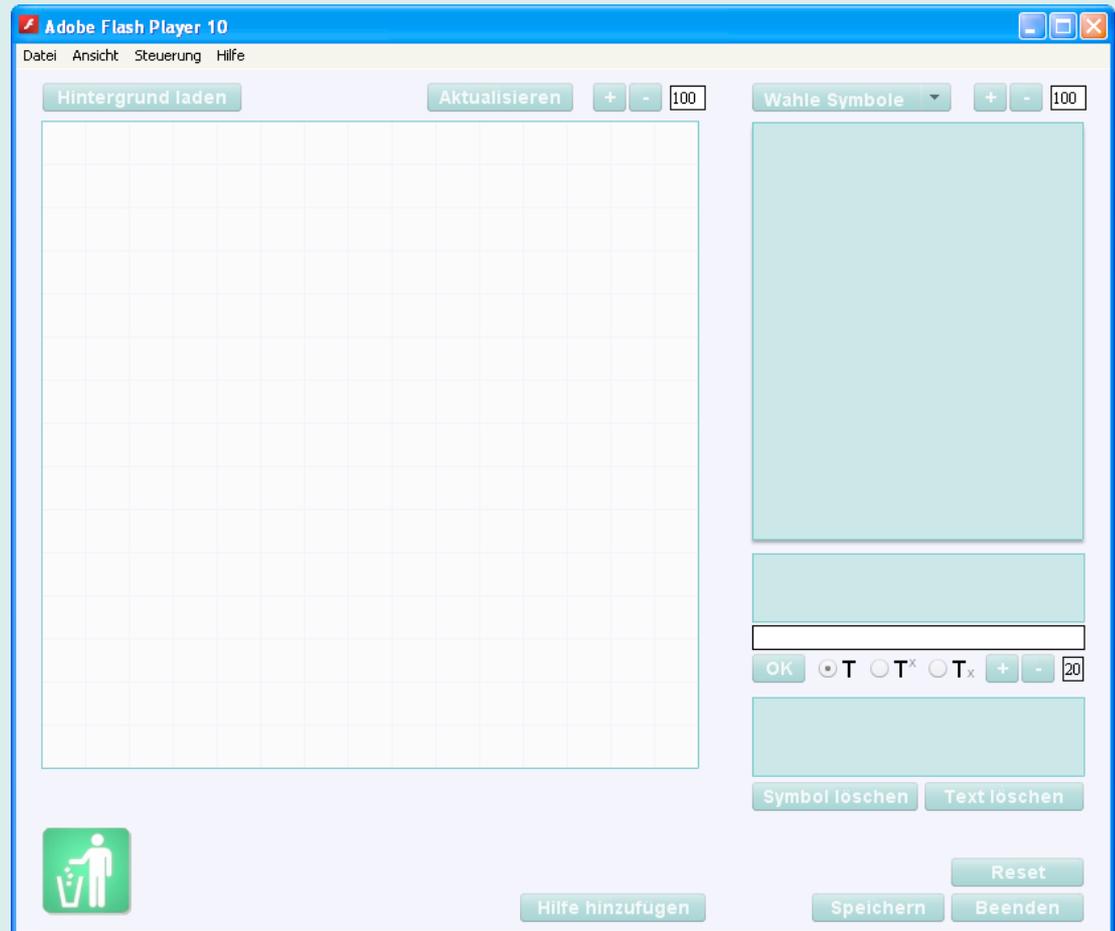
Symbol löschen Text löschen

Reset

Hilfe hinzufügen Speichern Beenden

Editor – Matrix *Trainersicht*

- Gleiche Funktionalitäten wie Freifläche
- Aber Anordnung der Symbole und Texte auf einem Raster
 - Rastergröße im Menü anpassbar



Editor – Puzzle

Trainersicht

- Einladen von Bildfragmenten (Puzzleteile) im Hauptmenü
- Anordnung der Bildfragmente auf Freifläche
- Container zur Auswahl der eingeladenen Bildfragmente

Editor – Puzzle

Trainersicht

Bitte Anordnungsfläche wählen:

Quadratisches Raster mit N=

Freifläche

Standard

Puzzle

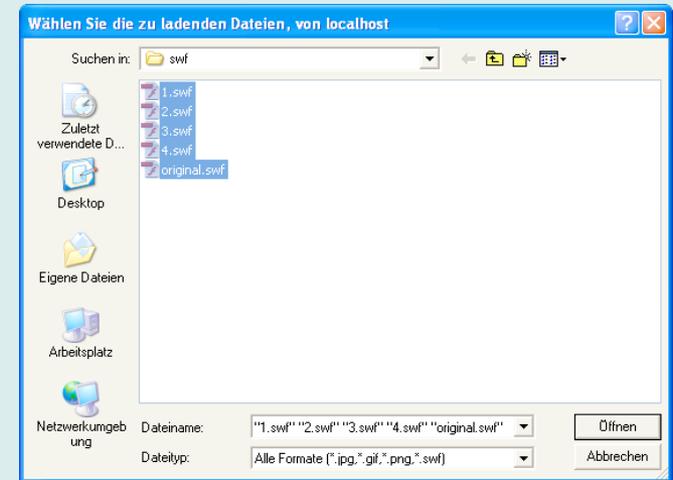
1. Geben Sie bitte einen Projektnamen ein:

2. Wählen Sie bitte die Puzzleteile sowie das Vorgabebild aus:

Anzahl der Teilbilder: 4

*Empfohlenes Format für Puzzleteile und Vorgabebild ist das swf-Format.
Das Vorgabebild muss 'original.xxx' heißen. Die ausgewählten Puzzleteile
müssen, von der 1 beginnend, durchnummeriert sein.*

Auswahl-Beispiel: [original.swf, 1.swf, 2.swf, 3.swf, 4.swf]

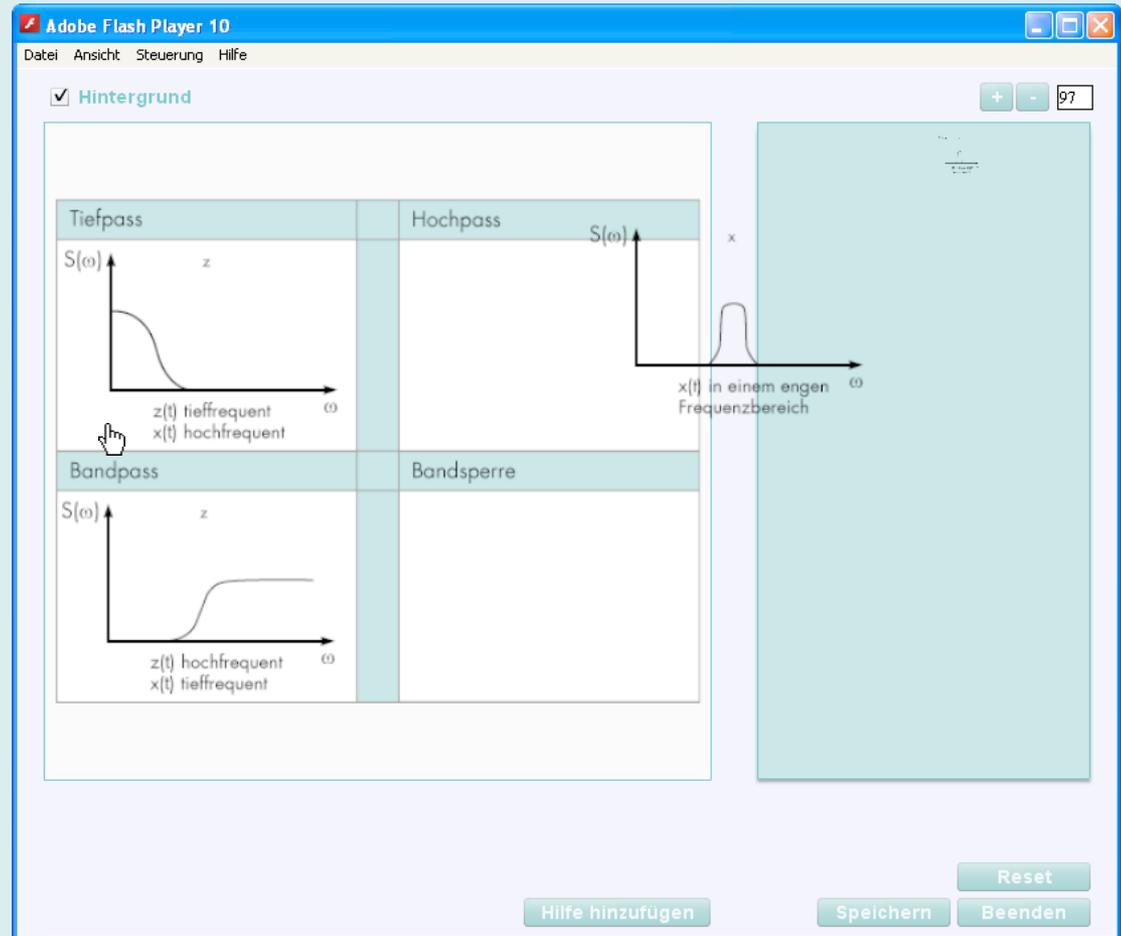


- Angabe eines Projektnamens
- Auswahl der vorgefertigten Bildfragmente vom lokalen System

Editor – Puzzle

Trainersicht

- Einblenden des kompletten Lösungsbildes als Hilfe zum Anordnen der Teilbilder
- Ausblenden des Hintergrund vor Speichervorgang



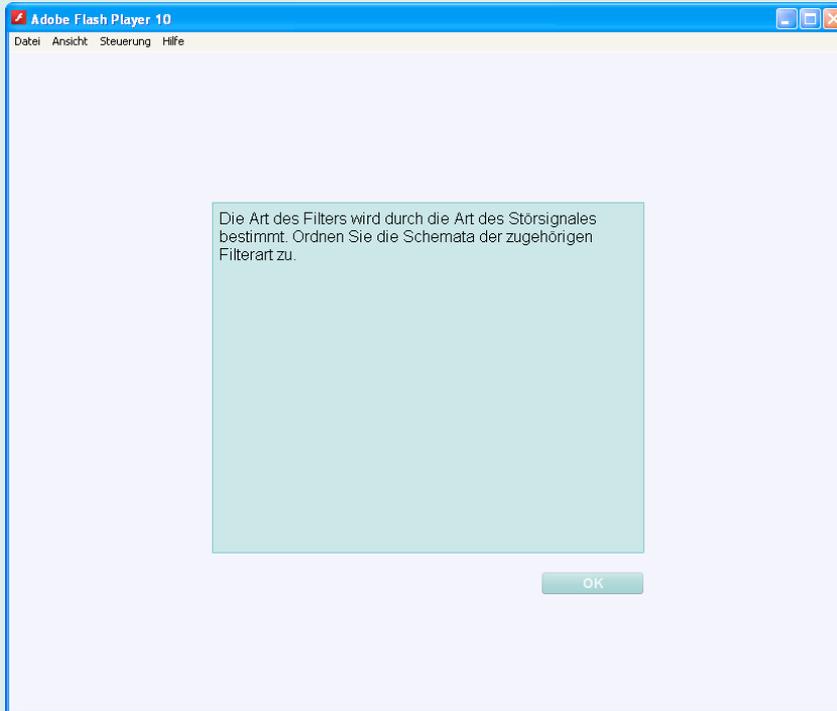
Anwendungsprogramm

Teilnehmersicht

- Anwendung der erstellten Aufgaben
- 3 Anwendungsprogramme, für Teilnehmer aber irrelevant
 - Freifläche
 - Matrix
 - Puzzle
- Hintergrundgrafiken werden eingeladen
- Lösungsbausteine (Symbole, Texte oder Puzzleteile) werden eingelesen
- Zuordnung der Lösungsbausteine auf Ankerpunkte

Anwendungsprogramm

Teilnehmersicht

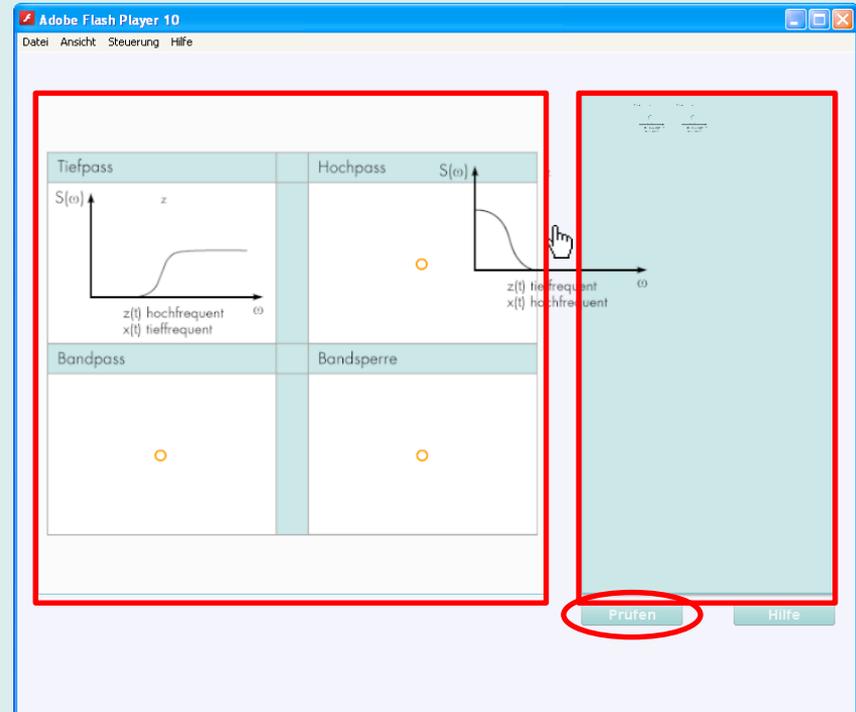


- Anwendungsprogramm startet mit Fragetext, der auch später wieder aufgerufen werden kann

Anwendungsprogramm

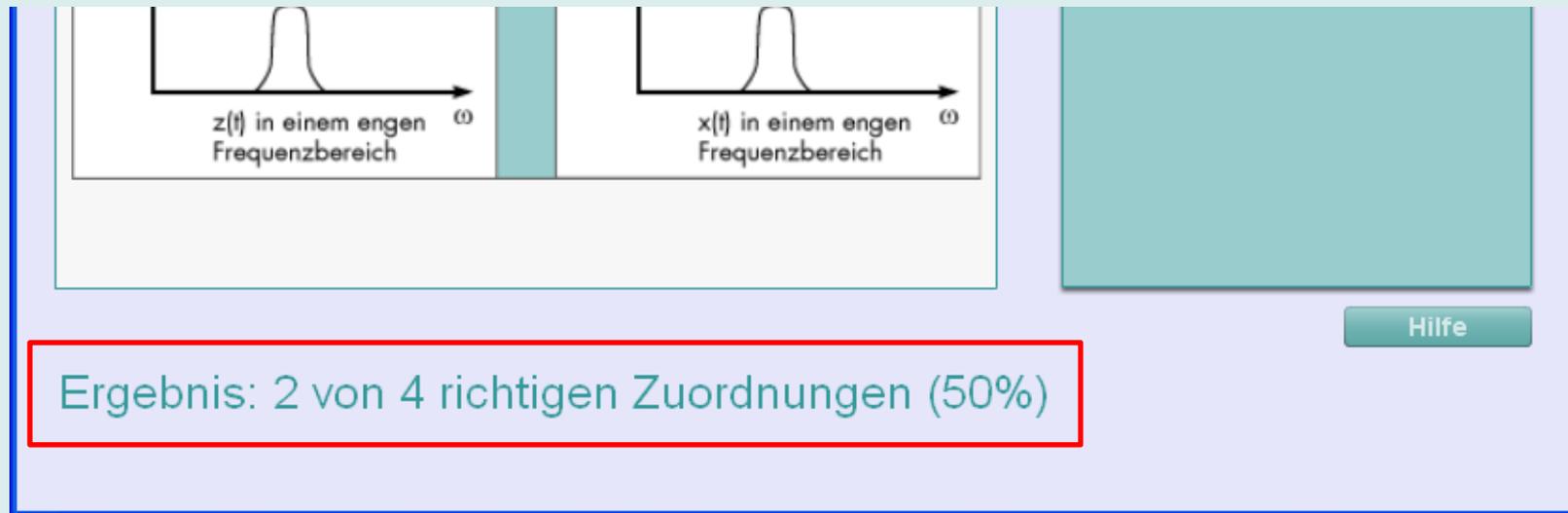
Teilnehmersicht

- Anordnungsfläche mit geladener Hintergrundgrafik und Ankerpunkten für Lösungsbausteine
- Auswahlcontainer mit richtigen und falschen Lösungsbausteinen
- Button zum Prüfen der Anordnung



Anwendungsprogramm

Teilnehmersicht



The screenshot displays a user interface for an application. On the left, there are two side-by-side plots. The left plot shows a bell-shaped curve labeled $z(t)$ in a narrow frequency range ω . The right plot shows a similar bell-shaped curve labeled $x(t)$ in a narrow frequency range ω . A vertical teal bar is positioned between the two plots. To the right of the plots is a large teal rectangular area, and below it is a button labeled "Hilfe". At the bottom left, a red-bordered box contains the text "Ergebnis: 2 von 4 richtigen Zuordnungen (50%)".

- Berechnung des Gesamtergebnisses anhand richtiger Zuordnungen nach Betätigung des Prüfen-Buttons

Moodle-Einbindung / SCORM

Integration des Anwendungsprogramms als Lernpaket

lernpaket.zip

tool.html

ruft Lerninhalt auf und beinhaltet JavaScript-Code (für SCORM-API)

tool.swf

Lerninhalt (Anwendungsprogramm)

imsmanifest.xml

beschreibt Aufbau und Inhalt des Lernpaketes

Metadaten.dtd

Metadaten.xsd

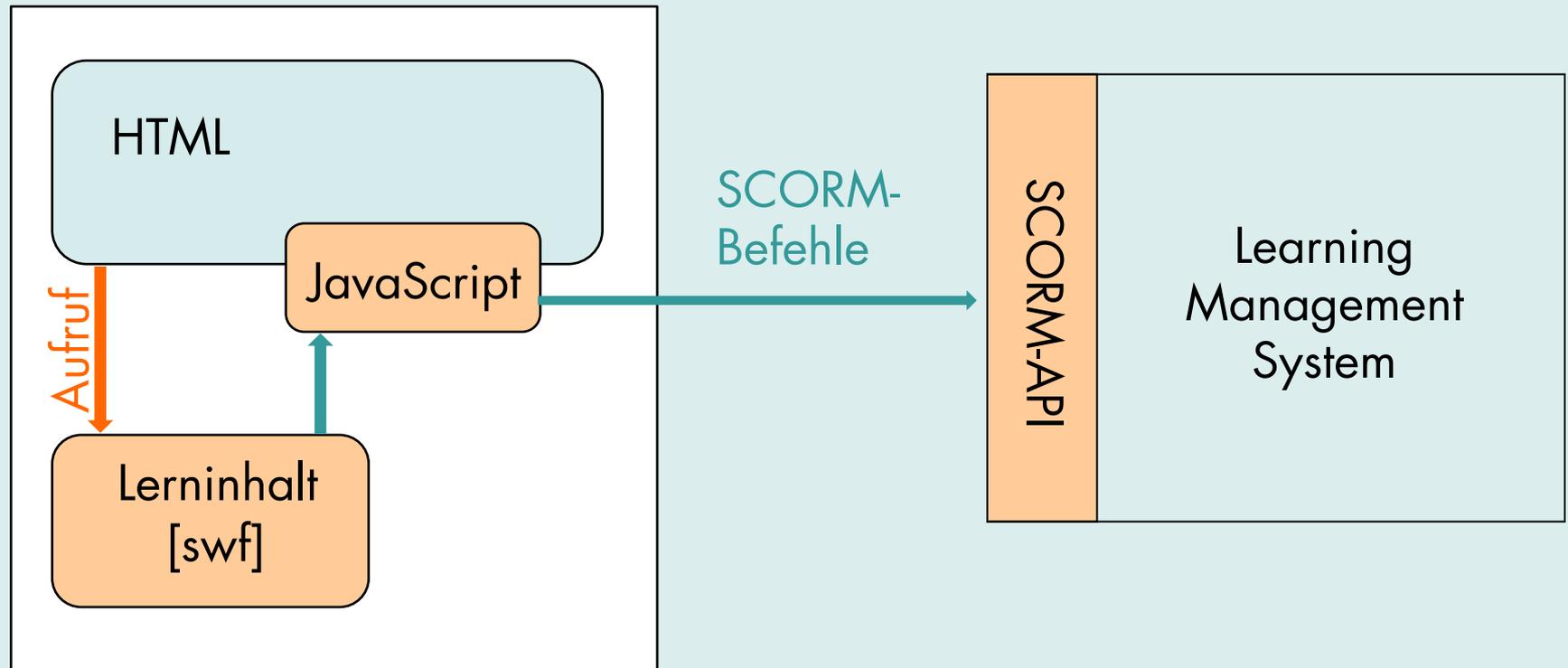
Metadaten der beteiligten Paketinhalte, automatisch erzeugbar mittels SCORM-Editor

Moodle-Einbindung / SCORM

- Lernpaketerstellung mit „Reload Editor“
- SCORM-API zur Übergabe von Werten an die Moodle-Plattform
- JavaScript-Code dient als Kommunikationsschnittstelle zwischen Lernpaket und Moodle

Moodle-Einbindung / SCORM

SCORM-Lernpaket



Moodle-Einbindung / SCORM

- JavaScript-Funktionen
 - 1. Verbindungsaufbau:

```
var API = null; // SCORM-API

function SCOInitialize() {
    initAPI(window);
    if (API != null) {
        API.LMSInitialize('');
    }
}

function initAPI(win) {
    findAPI(win);
    if ((API == null) && (win.opener != null)) {
        findAPI(win.opener);
    }
}

function findAPI(win) {
    while ((win.API == null) && (win.parent != null) && (win.parent != win)) {
        win = win.parent;
    }
    API = win.API;
}
```

Moodle-Einbindung / SCORM

- JavaScript-Funktionen
 - 2. Werteübergabe:

```
function SCOSetValue(parameter, wert) {  
    if (API != null) {  
        API.LMSSetValue(parameter, wert);  
    }  
}
```

- 3. Verbindungsabbau:

```
function SCOFinish() {  
    if (API != null) {  
        API.LMSFinish('');  
    }  
}
```

Moodle-Einbindung / SCORM

- Aufruf der JavaScript-Funktionen
 - Aus Flash:

```
3 var setValue:String = "cmi.core.score.raw,"+ergebnis;  
4  
5 fscommand("LMSSetValue", setValue);  
6 fscommand("LMSFinish");  
7
```

```
function anwendung_puzzle_DoFSCommand(command, args){  
    var Funktion = new String(command);  
    var Argumente = new String(args);  
    var Parameter, Wert;  
  
    var sep = Argumente.indexOf(",");  
    if (sep > -1){  
        Parameter = Argumente.substr(0, sep);  
        Wert = Argumente.substr(sep+1);  
    } else {  
        Parameter = Argumente  
    }  
  
    if ( Funktion == "LMSSetValue" ) {  
        SCOSetValue(Parameter, Wert);  
    } else if ( Funktion == "LMSFinish" ) {  
        SCOFinish();  
    }  
}
```

Vorführung

Ansprechpartner

www.tc-compact.de

Universität Leipzig

Fakultät für Chemie und Mineralogie

Institut für Technische Chemie

Projektleiter

Prof. Dr. Roger Gläser

E-Mail: roger.glaeser@uni-leipzig.de

Tel. 0341 9736301



Koordinator

Dipl.-Ing. Ralf Moros

E-Mail: ralf.moros@uni-leipzig.de

Tel. 0341 9736329

