|  |
| --- |
| Jeugdtheater De Toverlantaarn |
| www.detoverlantaarn.be |
| DragonEye Software |

|  |
| --- |
| Matthias Bauw |

Inhoud

[1. Inleiding 1](#_Toc215230917)

[2. Visuele vormgeving 1](#_Toc215230918)

[2.1. Het template 1](#_Toc215230919)

[2.1.1. Slices 2](#_Toc215230920)

[2.2. De HTML-code 3](#_Toc215230921)

# Inleiding

De nieuwe site voor De Toverlantaarn wordt het eerste project dat gebruik maakt van Controlcenter2. Aanvankelijk zal er gebruik gemaakt worden van de versie die in ontwikkeling is met de bedoeling om dat uiteindelijk de afgewerkte versies te implementeren.

Deze site wordt in verschillende fases ontwikkeld:

* Basiswebsite: Vooral met de bedoeling om publiciteit te maken naar de buitenwereld toe
* Ledensite: De bedoeling van de ledensite is om een platform te creëren waar leden van De Toverlantaarn in contact kunnen komen en in een beveiligde omgeving gegevens kunnen uitwisselen.

De bedoeling is in eerste instantie om gewoon de website zoals die er nu is een nieuwe look te geven en ondertussen Controlcenter2 te implementeren.

# Visuele vormgeving

## Het template

Voor het ontwerpen van het template wordt er gebruik gemaakt van Photoshop CS3 Extended. Het basistemplate ziet er als volgt uit:



De bedoeling is om de centrale foto in de navigatie om de 5 seconden te wijzigen. Op deze manier krijgt een bezoeker een overzicht van de verschillende producties uit het verleden.

### Slices

Om een omzetting van de grafische versie naar XHTML mogelijk te maken wordt het ontwerp als volgt gesliced:



Er werd bij het definiëren van de slides met verschillende dingen rekening gehouden:

* De Slice Slideshow zal door een Flashmovie vervangen worden (zie verder)
* De Navigatie werd in 2 stukken opgedeeld omdat de CSS uitlijning van de navigatielinks op die manier veel vlotter kan verlopen. Zo kunnen de items in de navigatie toch links worden uitgelijnd

Ook moet worden opgemerkt dat de grafics van de Inhoud slice vervangen worden door CSS code: de achtergrond werd ingesteld op #EBEBEB. Het heeft dus geen zin om de gebruiker een jpeg te laten inladen als er een egale kleur kan worden ingesteld.

De borders worden opnieuw gegenereerd door CSS. Dit zorgde voor problemen aan de rechterzijde van de inhoud slice. Het probleem werd opgelost door de breedte van deze div met 2 pixels te verkleinen in de CSS-code.

## De HTML-code

Door het efficiënt gebruik maken van CSS en van DIV’s in plaats van tables kan de HTML-code tot een minimum worden herleid:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>toverlantaarn\_versie4</title>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<link href="css/positioning.css" rel="stylesheet" type="text/css" />

</head>

<body>

<div id="Bodyshell">

<div id="Schaduw-Links\_">

<img id="Schaduw\_Links" src="images/Schaduw\_Links.gif" width="20" height="800" alt="" />

</div>

<div id="Bovenbalk\_">

<img id="Bovenbalk" src="images/Bovenbalk.jpg" width="757" height="29" alt="" />

</div>

<div id="Schaduw-Rechts\_">

<img id="Schaduw\_Rechts" src="images/Schaduw\_Rechts.gif" width="4" height="800" alt="" />

</div>

<div id="Slideshow\_">

<img id="Slideshow" src="images/Slideshow.jpg" width="757" height="200" alt="" />

</div>

<div id="Navigatie-Linkerdeel\_">

<img id="Navigatie\_Linkerdeel" src="images/Navigatie\_Linkerdeel.jpg" width="270" height="56" alt="" />

</div>

<div id="Navigatie-Rechterdeel\_">

<UL>

<LI>Hoofdpagina</LI>

<LI>Producties</LI>

<LI>Fred Delfgauw</LI>

<LI>Nevenactiviteiten</LI>

<LI>Contact</LI>

</UL>

</div>

<div id="Inhoud\_">

<!--PageContent-->

</div>

</div>

</body>

</html>

## De Slideshow

Het belangrijkste visuele element is de Flash slideshow die foto’s moet tonen van de verschillende producties. Voor het bekomen van dit resultaat wordt gebruikt gemaakt van een combinatie tussen Actionscript en XML.

De syntax van de XML-file is als volgt:

<slideshow>

<slide>

<productie>Klokkenluider van de Notredame</productie>

<jaar>2008</jaar>

<foto>test.jpg</foto>

</slide>

</slideshow>

De bedoeling is dat de Flash-movie de xml file ophaalt, de foto’s importeert en deze in willekeurige volgorde weergeeft. Deze manier van werken zorgt ervoor dat de inhoud van de slideshow gemakkelijk kan worden aangepast.

De eerste versie van de ActionScript code is als volgt:

//Definitie van variabelen

var xmlFilename:String = 'slideshow.xml';

var xmlSlideshow:XML;

var Slidenodes:XMLList;

var slideTimer:Timer = new Timer(5000);

var movieclipSelector:Boolean = false;

slideTimer.repeatCount=0;

//De XML File wordt ingeladen

var xmlRequest:URLRequest = new URLRequest(xmlFilename);

var xmlLoader:URLLoader = new URLLoader;

xmlLoader.load(xmlRequest);

xmlLoader.addEventListener("complete",startSlideshow);

function startSlideshow(loadEvent:Event):void

{

//De XML-file is ingeladen. Nu kan ze verwerkt worden.

xmlSlideshow = new XML(xmlLoader.data);

Slidenodes = xmlSlideshow.children();

//De eerste afbeelding moet worden ingeladen

//Eerst wordt een willekeurig nummer gekozen van de slide

var slidenummer:int = selectSlide(Slidenodes.length());

//Blijkbaar neemt de childrenfunctie een teller die begint van nul en voor iedere tag

//1 bijtelt

var elementnummer = (slidenummer \*3) + 2;

var slidepath:String= Slidenodes.children()[elementnummer];

showSlide(slidepath,this.Slideshow.afbeelding2);

//Ook de 2e afbeelding moet onmiddelijk worden ingeladen

slidenummer = selectSlide(Slidenodes.length());

elementnummer = (slidenummer \*3)+2;

slidepath = Slidenodes.children()[elementnummer];

showSlide(slidepath,this.Slideshow.afbeelding1);

movieclipSelector= false;

//Nu moet de timer gestart worden die om de zoveel tijd de afbeelding wijzigt

slideTimer.addEventListener("timer",nextSlide);

slideTimer.start()

}

function nextSlide(e:TimerEvent)

{

//De volgende dia is al ingeladen dus nu moet deze zichtbaar worden.

this.Slideshow.play();

//Nu moet de daaropvolgende dia ingeladen worden

//Eerst wordt een willekeurig nummer gekozen van de slide

var slidenummer:int = selectSlide(Slidenodes.length());

//Blijkbaar neemt de childrenfunctie een teller die begint van nul en voor iedere tag

//1 bijtelt

var elementnummer = (slidenummer \*3) + 2;

var slidepath:String= Slidenodes.children()[elementnummer];

if(movieclipSelector == true)

{

showSlide(slidepath,this.Slideshow.afbeelding1);

movieclipSelector= false;

}

else

{

showSlide(slidepath,this.Slideshow.afbeelding2);

movieclipSelector = true;

}

trace(slidenummer);

}

function selectSlide(max:int):int

{

//Deze functie selecteert willekeurig een afbeelding die de volgende slide

//wordt

var random = Math.random();

max = max -1;

//random heeft op dit punt een willekeurige waarde tussen 0 en 1

var slideNummer:int = Math.round(random \* max);

return slideNummer;

}

function showSlide(path:String,target:MovieClip)

{

var rect:Shape = new Shape();

rect.graphics.beginFill(0x000000);

rect.graphics.drawRect(0,0,757,200);

rect.graphics.endFill();

var ldr:Loader = new Loader();

ldr.mask = rect;

var urlReq:URLRequest = new URLRequest(path);

ldr.load(urlReq);

target.addChild(ldr);

}