

SteriaQuiz
teknisk dokumentasjon

Henning Klevjer og Esther Bloemendaal

20. Juli 2012

Innhold

1 SteriaQuiz	3
2 Deploye på server	3
3 Quiz-fila	4
4 Mobilsiden	4
4.1 Kjøring	4
4.2 Paneltyper	6
5 Deltakerlisten	7
6 Svakheter	7
7 Kodekommentarer	8
8 Mappestruktur i JavaZone-prosjektet	12

1 SteriaQuiz

Steriaquizen er laget for profilering av Steria på konferanser (JavaZone, etc.), bedriftspresentasjoner og liknende. Den er i hovedsak skrevet i javascript, og bruker Sencha Touch 2-APIet¹ til alt grafisk.

Programmet kjører på en hjemmelaget server skrevet i Java. Dette har vi gjort for å unngå å bruke PHP og Apache (unødvendig mye kluss), men ikke minst for å få spesialtilpasset tilgangskontroll til filene serveren hoster, da spesielt muligheten til å lese korrekte svar. Quizen fungerer altså med Apache+PHP², men da må quizsvarene separeres i egen fil for å ikke være offentlig lesbare.

Presentasjon av deltakere er gjort i XML med stylesheets i XSL og CSS, i tillegg til lett javascripting for å trekke en vinner.

Dette dokumentet er skrevet mest som et oppslagsverk. All informasjonen er ikke nødvendig for å endre koden. De fysiske delene (“programmene”) i quizprosjektet blir forklart under.

Støttede nettlesere

Siden quizen bruker Sencha er den begrenset til WebKit-nettlesere³. For øyeblikket er disse de viktigste WebKit-baserte nettleserne:

Android Browser (følger med operativsystemet)

Safari

Safari Mobile (følger med operativsystemet)

Google Chrome

Google Chrome for Android

(Mobilnettlesere markert i **bold**)

Dette betyr altså at Opera og Internet Explorer ikke støttes, hverken på mobiltelefon eller ellers.

2 Deploye på server

SteriaQuizen har vært testet lokalt med Apache 2.2 og PHP 5, og bør fungere på en hvilken som helst server som støtter PHP. I tillegg, som nevnt over, er det skrevet en quiz-server **SteriaQuizServer** som kan sanere quizfila før sending til mobiltelefonen. Sistnevnte er altså enklest og sikrest (men kanskje ikke fri for bugs og idiotiske *løsninger*).

Brukere bør komme inn i quizen via en link til **app.html**, gjerne hjulpet av url-forkorter eller QR-kode⁴.

På den hjemmelagde jahaserveren redirectes PHP-forespørsler til en handler i serverkoden **SteriaQuiz.validateParams()**. Den største forskjellen fra Apache-kjøringen er at parametrene

¹<http://sencha.com>

²Det *døende* serversidespråket

³<http://webkit.org>

⁴Selv om nesten ingen på jorda liker, bruker eller trenger QR-koder kan det være at den ene promillen som har QR-scanner og lyst til å bruke den dukker opp på JavaZone. Tenk så fornøyde *de* blir når de kan scanne QR-kodene!

til `reg.php` sendes Base64-kodet, for å støtte spesialtegn i navn.

Viktig

Skal man bruke den hjemmelagde serveren må man sette variablen `APACHE_AND_PHP` til `false` i `app.js`. Dette for at klienten må velge hvorvidt parametrene skal kodes om eller ikke. Skal man kjøre både Apache+PHP og den hjemmelagde serveren må man ha to utgaver av `app.js`, eller en smartere måte å velge koding eller ikke på. Evt. implementere Base64 i `reg.php` og splitte parametre etter dekodning.

3 Quiz-fila

Quiz-fila definerer oppgavene, løsninger og informasjon om gevinst og trekning. Den er skrevet i XML-format og endres med en fritt valgt tekstbehandler. Hver quiz-fil støtter ett `<quiz>`-element. Under `<quiz>` er det en tittel, `<title>` og to informasjonselementer, `<info>` som beskriver quizen, og `<lastPageInfo>` som blir vist til brukeren etter at svarene er registrert. Ett `<quiz>`-element kan inneholde evig mange oppgaver, `<question>`. Hvert `<question>` består alltid av kun en `<type>`, en `<title>` og `<correct>`, der `<type>` er en av paneltypene beskrevet under, i punkt 4.2, `<title>` er tittelen (oppgaveteksten) og `<correct>` beskriver svaret man skal fram til i oppgaven. Programmet regner 0 som det første svaret.

I javascriptet (`app.js`) er adressen til quizfila hardkodet inn, så om hver quiz har forskjellig navn må dette endres i koden.

4 Mobilsiden

Mobilsiden – altså selve quizen – er en samling Sencha-paneler, hver definert i sin variabel. Disse kan fjernes og legges til i quizen ved at man lager flere instanser av samme type panel. Siden noe av GUIet som genereres med Sencha trenger spørsmåldata for å lastes, gjøres initialiseringen i to faser, `initStep1()` og `initStep2()`. Steg to settes i gang når spørsmålene er lastet. Metoden `switchTo()` bytter mellom vinduer.

4.1 Kjøring

Førstesiden inneholder en startknapp og første informasjonsfelt fra quiz-fila. Når man etter å ha trykket på startknappen kommer til første oppgave åpnes en “karusell”⁵, en sencha-container som inneholder alle oppgavene. Karusellen gjør det mulig å “swipe” mellom spørsmålene. Hver “swipe” trigger en meldingsboks med oppgaveteksten der dette ikke er skrevet på siden.

På alle sider er det også mulig å trykke på enten en knapp eller pil for å komme videre til neste eller forrige spørsmål.

⁵Ext.carousel.Carousel i Sencha API-et

```

<?xml version="1.0" encoding="UTF-8"?>
<quiz>
  <title>Sterias JavaZone-quiz</title>
  <info>Dette er siste test i JavaZone-quiz-betaen. Ende med.,</info>
  <lastPageInfo>Siden dette er en beta får du dessverre ingen premie</lastPageInfo>
  <question>
    <type>slider</type>
    <title>Hvor høyt er bygget hvor Steria har sitt norske hovedkontor?</title>
    <correct>26</correct>
  </question>
  <question>
    <type>imgsel</type>
    <title>I hvilke av disse landene ble Steria grunnlagt?</title>
    <correct>1</correct>
    <img_src>country_de.png</img_src>
    <img_src>country_fr.png</img_src>
    <img_src>country_se.png</img_src>
    <img_src>country_no.png</img_src>
  </question>
  <question>
    <type>ordinary</type>
    <title>Hvor mange ansatte har Steria i Norge?</title>
    <correct>0</correct>
    <answer>ca. 700</answer>
    <answer>ca. 1000</answer>
    <answer>ca. 300</answer>
    <answer>ca. 1100</answer>
  </question>
  <question>
    <type>ordinary</type>
    <title>Hvilket uttrykk er lovlig i Java (1.7)?</title>
    <correct>3</correct>
    <answer>long int x;</answer>
    <answer>Object i = (byte) * (int) 3;</answer>
    <answer>... catch ( EOFException || Exception e ) {}</answer>
  </question>

```

Figur 1: Quiz-fila

Når man har svart på alle spørsmål har man muligheten til å levere inn svarene. Da blir disse svarene lagret i nettleseren via `localStorage`⁶. Dermed kan man sjekke spørsmålene sine senere om man må svare på utrolig viktige telefonsamtaler som dukker opp midt i quizen, for eksempel. Lagringen er nå satt til å gjøres ved registrering, men kan enkelt settes til å lagre på hvert enkelt svar. Et argument mot denne mellomlagringen er at det kan bli enklere å sammenligne poengsummer og svar og spekulere seg fram til hva som er riktig. Men hvor farlig dette er kan også diskuteres.

Når man registrerer navn, epost og telefonnummer i registreringsskjemaet som følger, er man nødt til å bruke en epostadresse og et telefonnummer som ikke er registrert fra før – dette for å unngå at man øker vinner sjansene med flere besvarelser. Informasjonen fra brukeren lagres i to xml-filer, `scoreboard.xml` og `scoreboard-private.xml`. Mer om disse under punkt 5.

Når alt er gjort får man en beskjed om når trekningen foregår og lignende.

4.2 Paneltyper

For å være ekstra presis er det ikke panelene som er forskjellige, men innholdet i dem. Vi har fire forskjellige typer paneler, som presenteres under. Om man ikke liker navnet `ordinary` på flervalgsspørsmålene kan dette naturligvis endres i koden.

Map

`<type>` i `.quiz-fil`: `map`

Dette panelet er ikke i bruk. Meningen var at brukeren skulle kunne få oppgaver som f.eks “Hvor ligger Sterias hovedkontor?”, hvor brukeren zoomer seg inn på kartet og finner (innenfor en satt margin (“threshold”)) stedet med koordinatene angitt i svarvariabelen. Grunnen til at det ikke er i bruk er at oppgavene ble for tungvinne for brukeren og at kartet (Google Maps API) i seg selv bruker for mye ressurser på telefonen, noe som resulterer i en rimelig kjedelig brukeropplevelse, siden oppgaven tar for lang tid.

Slider

`<type>` i `.quiz-fil`: `slider`

Slider-panelet brukes for at brukeren skal velge et område i et bilde som samsvarer med en fasitverdi fra quiz-fila. Vi har brukt dette panelet til spørsmålet “Hvor høyt er bygget hvor Steria har sitt norske hovedkontor”. Etasjen bestemmes av området brukeren trykker på, med en event hook på et `<div>`-element som trekker ut hvor på siden (i y-aksen) man har trykket. Dette tallet brukes til å regne ut hvilken etasje det er man gjetter på i bildet. Denne spesielle utregningen er hardkodet.

Bildevelger

`<type>` i `.quiz-fil`: `imgsel`

I denne panelet er meningen at man skal velge riktig av fire bilder. Det kan brukes i oppgaver

⁶<http://www.w3.org/TR/webstorage/>

hvor man f.eks. skal finne ett bilde som ikke passer inn blant de andre. De fire bildene er vanlige ``-tags, og klikkene blir registrert på panelet de ligger i.

Radio buttons

`<type>` i .quiz-fil: `ordinary`

Dette er de klassiske oppgavene hvor svarer på ett spørsmål med ett riktig av flere svar. Disse feltene kan sannsynligvis enkelt utvides til å bli “Check box”-er, hvor man altså har flere riktige svar.

5 Deltakerlisten

Deltakerlisten er todelt. Den blir presentert til brukeren som `scoreboard.xml`, den *offentlige* listen, og til “quizmastere” som `scoreboard-private.xml`. I førstnevnte noteres bare tidspunkt for levering, navn og poengsum. Her kan deltakerne se hvordan de og andre deltakere ligger an, og denne siden kan også brukes til trekning av vinnere. Den private siden skal kun brukes av quizmastere for å finne kontaktinformasjonen til vinneren. `scoreboard.xml` kan godt publiseres på nettet og linkes til fra avslutningssiden i quizen, mens `scoreboard-private.xml` (hvor deltakernes telefonnummer og epostadresser er lagret) må holdes privat, men skrivbar. Altså rettigheter `rw-x-w-w-`.

Deltakerlistene er skrevet i XML for enkelhets skyld – de kan endres manuelt om det trengs. Begge er stilsatt med CSS for å matche sterias profil (`scoreboardstyle.css`). XSL brukes for å sette opp listene korrekt ut fra innholdet i XML-fila. Skal man slette en eller flere deltakere må man passe på at man bare fjerner hele `<entry>`-elementer og at toppelementet `<scoreboard>` fortsatt er intakt.

6 Svakheter

Skjermretning

Hvis mobiltelefonen holdes i “landscape-modus” (sidelengs) blir det svært mye scrolling.

Liten skjerm

Bruker man en mobiltelefon med liten skjerm kan bildene i bildevelgeroppgaven bli veldig små. Er det i tillegg tekst i bildene (som i eksempelet) kan dette bli vanskelig å få med seg.

Tekstkoding

Det er viktig at tekstfilene med innhold som skal vises til brukeren er lagret som Unicode-fil! Hvis ikke må man bruke escape-tegn for spesialtegn. Skal man skrive linjeskift i XML-fila (f.eks i et spørsmål) må man bruke `<` og `>` istedenfor hhv. `<` og `>`.

For å garantere at filen er lagret i Unicode kan man lagre den i gode gamle Notepad og velge Unicode under *encoding*.

Levert	Navn	Telefon	Epost	Svar	Poengsum
13:56:27	Ja Ja				100
15:06:37	Per Pettersen	99889988	per@pettersen.no	28;1;0;1;1;0;	7
08:54:49	per pettersen	77445544	per@pet.no	21;1;1;1;1;0;	3
08:55:37	siri sol	88552233	siri@sol.no	51;1;2;0;0;2;	3

Personinformasjon lagres hos Steria fram til trekningen.

start

Figur 2: Privat scoreboard – dette brukes for å finne kontaktinformasjon om vinneren

Levert	Navn	Poengsum
13:56:27	J. Ja	100
15:06:37	P. Pettersen	7
08:54:49	p. pettersen	3
08:55:37	s. sol	3

Personinformasjon lagres hos Steria fram til trekningen.

Uavgjort?

Trekk en vinner

start

Figur 3: Offentlig scoreboard – dette kan presenteres til brukerne og brukes til trekning

7 Kodekommentarer

Globale variable

APACHE_AND_PHP Er true hvis man kjører Apache, false hvis man bruker Java-serveren.

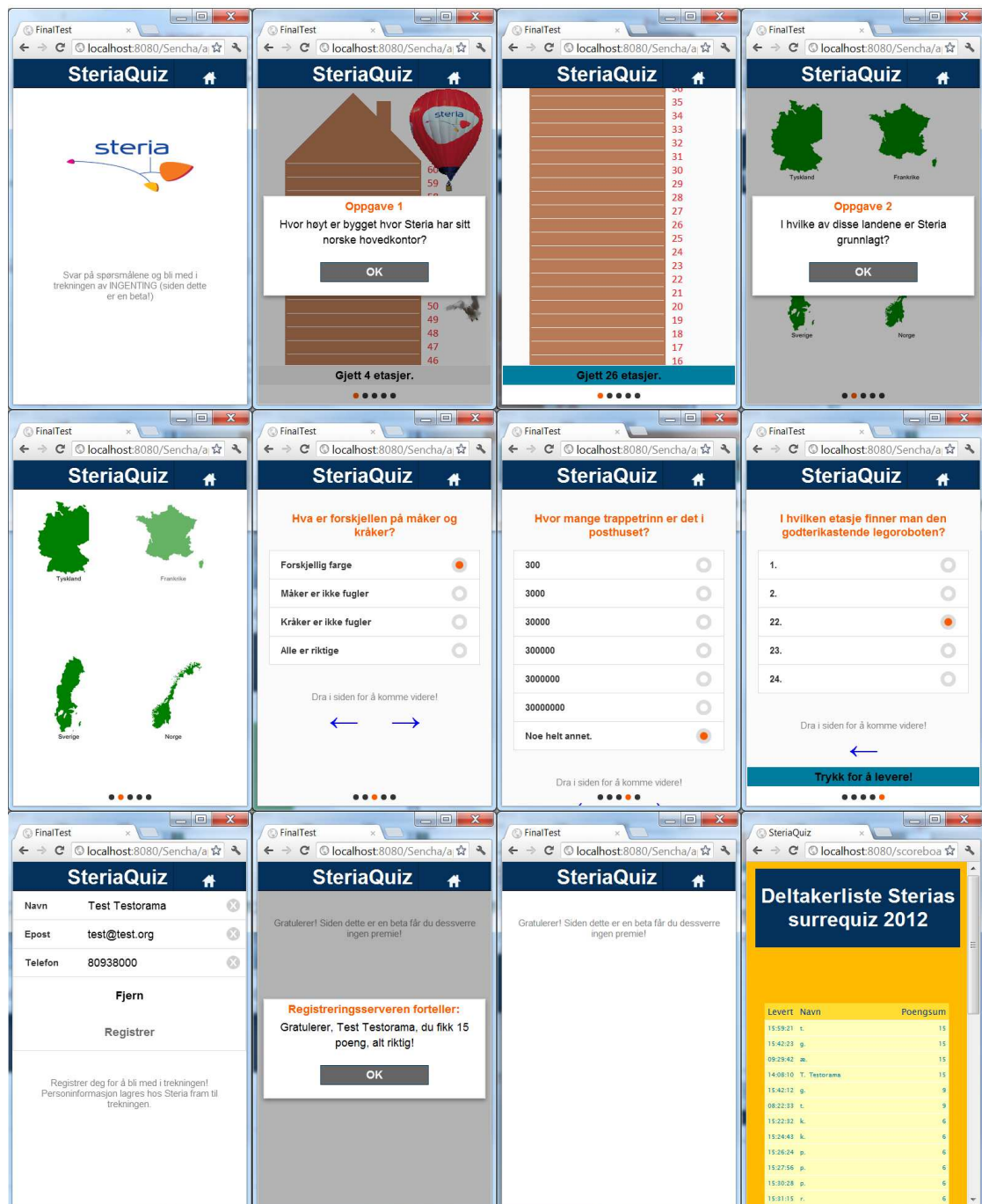
titlebar Titteltekst med hjem-knapp

currentPage Inneholder hvilken side man befinner seg i

firstPage Førstesiden med startknapp

carousel Karusellen (brukes til å velge oppgave)

regPage Registreringssiden



Figur 4: Forskjellige skjermbilder fra SteriaQuiz

lastPage Siste side (informasjonsside)
sliderQ "Hus"-oppgaven
imgQ Fire bilder-oppgaven
radioQ Radio button-spørsmålene
answers Brukerens svar
regPath Adressen til registreringssiden
globAnsCnt Brukes for å holde styr på antall oppgaver
totQuestions Totalt antall oppgaver
quizInfo Informasjon for første side.
lastPageInfo Informasjonsholder for siste side.
lastPageText Teksten på siste side.
firstClick Hjelp til å håndtere klikking på Slider-oppgaver
QUIZ_PATH Adressen til quizfila.

funksjoner

Slider() Konstruktør til slider-"klassen".
RadioBtnQuestion() Konstruktør for radiobuttonspørsmålsider. Lager et panel med radio buttons. Parameter-format: [tittel][q1][q2]...[qn]
ImageSelector() Konstruktør for bildevalgoppgaven
initStep1() Første initialiseringssteg
initStep2() Andre initialiseringssteg Settes i gang av getQuestions()
switchTo() Bytter skjerm bilde
createTitlebar() Genererer tittelfeltet
createFirstPage() Genererer førstesiden
createLastPage() Genererer sistesiden
createRegPage() Genererer registreringssiden
createRegPath() Legger parametrene til registreringsserveradressen
createCarousel() Genererer spørsmålskarusellen

`getQuestions()` Henter spørsmålene fra quizserveren

`loadQuestions()` Henter spørsmålene fra `localStorage`

`isAValidPhoneNumber()` Hjelpesfunksjon til telefonnummervalidering. Vurderer om en variabel er et tall.

Fotnoter

Koden til SteriaQuizen ligger på: <https://github.com/xhenk/JavaZone.git>

Brukernavn: `xhenk`, passord: `1234foo` om noe skal endres

Dokumentasjon til Sencha Touch 2: <http://docs.sencha.com/touch/2-0/>

Apache : <http://www.apache.org/>

PHP : <http://www.php.net/>



8 Mappestruktur i JavaZone-prosjektet

Listing 1: De viktigste filene i prosjektet

\index.php	Hovedsiden til prosjektet
\reg.php	Registreringssiden i PHP
\scoreboard-priv-style.xsl	Stil til privat scorebaord
\scoreboard-private.xml	Privat scoreboard
\scoreboard-pub-style.xsl	Stil til offentlig scoreboard
\scoreboard-style.css	CSS-stylesheet til scoreboards
\scoreboard.js	JS-del av offentlig scoreboard
\scoreboard.xml	Offentlig scoreboard
\sterialogo.gif	Sterias logo
\tips.html	Tipsside for hovedsiden
\tips.php	Tipsregistrering for hovedsiden
\Dokumentasjon\	All dokumentasjon til prosjektet
\ikketeknisk.pdf	Ikketeknisk – SteriaQuiz
\ikketeknisk.tex	Ikketeknisk – SteriaQuiz (source)
\legoroboten.pdf	Teknisk – legorobot
\legoroboten.tex	Teknisk – legorobot (source)
\SteriaQuiz-teknisk.pdf	Teknisk – SteriaQuiz
\SteriaQuiz-teknisk.tex	Teknisk – SteriaQuiz (source)
\LeJOS	Programmet som kjører på roboten
\bin	LeJOS-klasser
\Lego\$1.class	
\Lego\$2.class	
\Lego.class	
\Lego.nxj	LeJOS NXJ executable for NXT
\javadocs	Dokumentasjon av prosjektet
\src	Kildekoden til robotprogrammet
\Lego.java	

Listing 2: De viktigste filene i prosjektet, forts.

\Quiz\	Tekstinnhold til quizen
\JavaZone2012.quiz	XML-fil med all quizinfo
\Sench	Presentasjonen av quizen
\app.html	Siden som inneholder quizen
\app.js	Hovedprogrammet for quizen
\country_*.png	Bildefiler for bildevalgoppgave
\steria-bg.png	Bildefil for slider-oppgave
\sterialogo.gif	Startknapp for quizen
\stil.css	Stilinnhold til quizen
\SteriaQuizServer	
\src	
Server.java	Serverprogrammet. Kjører på port 80
\bin	(...)