SteriaQuiz - teknisk dokumentasjon

Henning Klevjer

July 23, 2012

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-APIet[1] til alt grafisk. Registrering av svar foregår i PHP[2] mot en XML-fil.

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-nettlesere[3]. For øyeblikket er disse de viktigste WebKit-baserte nettleserne:

Android Browser (følger med operativsystemet)

Safari

Safari Mobile (følger med operativsystemet)

Google Chrome

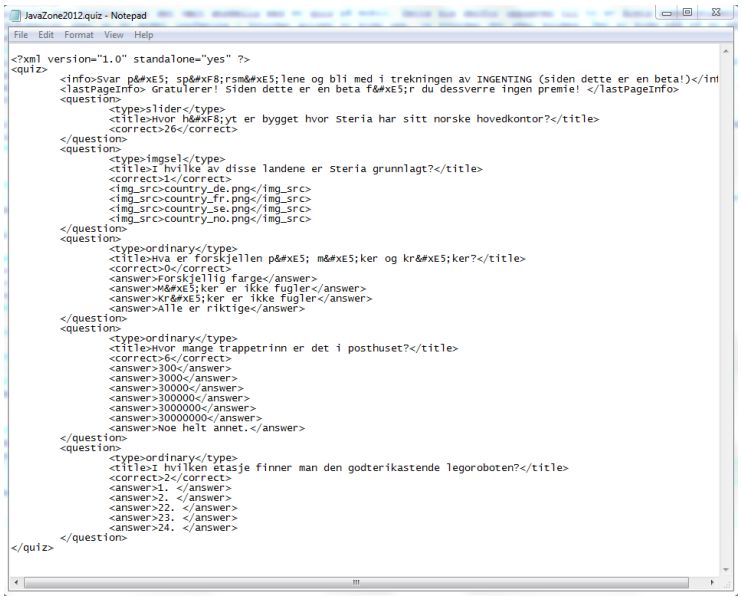
Google Chrome for Android

(Mobilnettlesere markert i **bold**)
Dette betyr altså at Opera og Internet Explorer ikke støttes, hverken på mobiltelefon eller ellers.

2 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

[1] http://sencha.com
[2] Det *døende* serversidespråket
[3] http://webkit.org



Figur 1: Quiz-fila

beskrevet under, i punkt 3, `<title>` er tittelen (oppgaveteksten) og `<correct>` beskriver svaret man skal fram til i oppgaven.

3 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.

3.1 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"[4], 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.

Når man har svart på alle spørsmål har man muligheten til å levere inn svarene. Da blir disse svarene lagret i nettleseren via localStorage[5]. 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

[4] Ext.carousel.Carousel i Sencha API-et
[5] http://www.w3.org/TR/webstorage/

satt til å gjøres ved registrering, men kan enkelt settes til å lagre på hvert enkeltsvar. 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 vinnersjansene med flere besvarelser. Informasjonen fra brukeren lagres i to xml-filer, `scoreboard.xml` og `scoreboard-private.xml`. Mer om disse under punkt 4.

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

3.2 Paneltyper

For å være ekstra presis er det ikke panelene som er forskjellige, men innholdet i dem. Vi har fire forskjellige typer paneler:

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 i seg (Google Maps API) selv bruker for mye ressurser på telefonen, noe som resulterer i en rimelig kjedelig brukeropplevelse, siden oppgaven tar for lang tid.

Slider

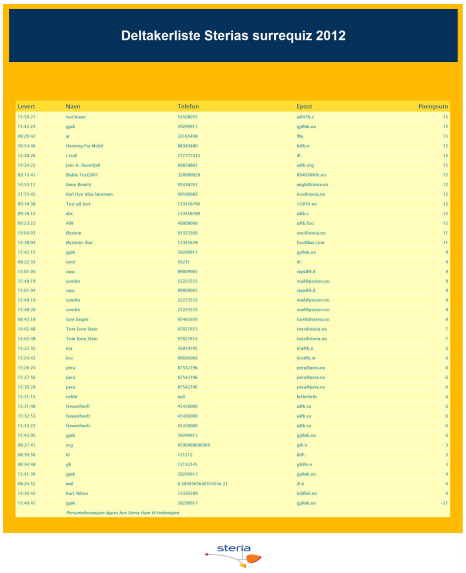
`<type>` i .quiz-fil: `slider`
Slider-panelet brukes for at brukeren skal velge et område i et bilde som samsvarer med en fasitverdi fra quiz-fila. Vi har brukt dette panelet til spørsmålet "Hvor høyt er bygget hvor Steria har sitt norske hovedkontor". Etasjen bestemmes av området brukeren trykker på, 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

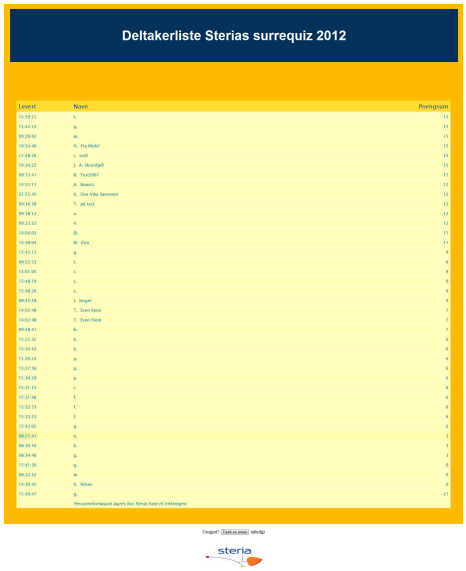
`<type>` i .quiz-fil: `imgsel`
I denne panelet er meningen at man skal velge riktig av fire bilder. Det kan brukes i oppgaver hvor man f.eks. skal finne ett bilde som ikke passer inn blant de andre. De fire bildene er vanlige `<img>`-tags, og klikkene blir registrert på panelet de ligger i.

Radio buttons

`<type>` i .quiz-fil: `ordinary`
Dette er de klassiske oppgavene hvor svarer på ett spørsmål med ett riktig av



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



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

flere svar. Disse feltene kan sannsynligvis enkelt utvides til å bli "Check box"-er, hvor man altså har flere riktige svar.

4 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.

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 (`scoreboard-style.css`). XSL brukes for å sette opp listene korrekt ut fra innholdet i XML-fila.

5 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

Vi har plages mye med å få ISO-8859-1-koding til å fungere med Sencha. Derfor har vi brukt HTML-escape-characters der det trengs spesialtegn. {æøåÆØÅ} fungerer dessverre heller ikke.

6 Kodekommentarer

Globale variable

`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
`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`	Informasjon for siste side.
`firstClick`	Hjelp til å håndtere klikking på Slider-oppgaver

Figur 4: Forskjellige skjermbilder fra SteriaQuiz

funksjoner

`Slider()`	Konstruktør til slider-"klassen".
`RadioBtnQuestion()`	Konstruktør for radiobuttonspørsmålsider. Lager et panel med radio buttons. Parameterformat: [tittel][q1][q2]...[qn]
`ImageSelector()`	Konstruktør for bildevalgoppgaven
`initStep1()`	Første initialiseringssteg
`initStep2()`	Andre initialiseringssteg Settes i gang av getQuestions()
`switchTo()`	Bytter skjermbilde
`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
`isANumber()`	Hjelpefunksjon til telefonnummervalidering. Vurderer om en variabel er et tall.

Fotnoter

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

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