Route simulation in Qt simulator

Qt Simulator offers simulation of various events that can be handled with Mobility APIs. This includes support for location simulation. Many of the events can be scripted and this can be used to enable route simulation. To be precise simulator already has a built in support for simple route simulation. It is scripted in a location.qs file. The default script moves starts from the predefined location and moves a bit during 20seconds.

Route simulation script

To enable route simulation we are creating our own script file based on location.qs.

1. Open Scripting part from the Qt Simulator menu.
2. Click open folder
3. Create a text file called route_simulation.qs
4. Copy the text below in to the file an save it.

Now you can run your route script by selecting it from the menu and clicking run.

To stop script execution press abort. You can see if the script works, by opening the location simulation part and observing that coordinates are changing.

route_simulation.qs

Invalid language.

You need to specify a language like this: `<source lang="html4strict">...</source>`

Supported languages for syntax highlighting:

4cs, 6502acme, 6502kickass, 6502tasm, 68000devpac, abap, actionscript, actionscript3, ada, algol68, apache, applescript, apt_sources, asm, asp, autoconf, autohotkey, autoit, avisynth, awk, bascomavr, bash, basic4gl, bf, bibtex, blitzbasic, bnf, boo, c, c_loadrunner, c_mac, caddcl, cadlisp, cfdg, cfm, chaiscript, cil, clojure, cmake, cobol, coffeescript, cpp, cpp-qt, csharp, css, cuesheet, d, dcs, delphi, diff, div, dos, dot, e, ecmascript, eiffel, email, epc, erlang, euphoria, f1, falcon, fo, fortran, freebasic, fsharp, gambas, gdb, genero, genie, gettext, glsl, gml, gnuplot, go, groovy, gwbasic, haskell, hicest, hq9plus, html4strict, html5, icon, idl, ini, inno, intercal, io, j, java, java5, javascript, jquery, kixtart, klonec, klonecpp, latex, lb, lisp, llvm, locobasic, logtalk, lolcode, lotusformulas, lotusscript, lscript, lsl2, lua, m68k, magiksf, make, mapbasic, matlab, mirc, mmix, modula2, modula3, mpasm, mxml, mysql, newlisp, nsis, oberon2, objc, objeck, ocaml, ocaml-brief, oobas, oracle11, oracle8, oxygene, oz, pascal, pcre, per, perl, perl6, pf, php, php-brief, pic16, pike, pixelbender, pli, plsql, postgresql, povray, powerbuilder, powershell, proftpd, progress, prolog, properties, providex, purebasic, pycon, python, q, qbasic, rails, rebol, reg, robots, rpmspec, rsplus, ruby, sas, scala, scheme, scilab, sdlbasic, smalltalk, smarty, sql, systemverilog, tcl, teraterm, text, thinbasic, tsql, typoscript, unicon, uscript, vala, vb, vbnet, verilog, vhdl, vim, visualfoxpro, visualprolog, whitespace, whois, winbatch, xbasic, xml, xorg_conf, xpp, yaml, z80, zxbasic

```
var loctionInformation =[{
   "longitude":0.02892424,
   "latitude":51.508074,
   "altitude":836.5
  },
  {
   "longitude":0.02892424,
   "latitude":52.508074,
   "altitude":836.5
  },
  {
   "longitude":0.02892424,
   "latitude":53.508074,
   "altitude":836.5
  },
  {
   "longitude":0.02892424,
   "latitude":54.508074,
   "altitude":836.5
  },
  {
   "longitude":0.02892424,
   "latitude":55.508074,
   "altitude":836.5
  },
                          {
   "longitude":0.02892424,
   "latitude":56.508074,
   "altitude":836.5
  }
];

var i =0;

while (true){
    location.useCurrentTimestamp = true;
    location.latitude = loctionInformation[i].latitude;
    location.longitude = loctionInformation[i].longitude;
    location.altitude  = loctionInformation[i].altitude;

    i = (++i) % loctionInformation.length;

    yield(1000);
}
```


(C) Copyright Nokia 2013. All rights reserved.

References

Location scripting interface http://doc.qt.nokia.com/qtsimulator-1.1/locationscriptinterface.html

Simulator scripting http://doc.qt.nokia.com/qtsimulator-1.1/simulator-scripting.html

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.