

Technical Outline ElasticMapping

Time scheme:

residence at Banff April-May 2010

preparations of forehand:

find collaboration with scientists and get permission for data use

find collaboration with local programmer

organize contacts with local cameraman for photo and video documentation

Week 1 contacts/meetings with scientist and programmer

Week 2-3 concentrated work on software development and technical planning

week 4-5 continues work on software and first experiments with the wildlife data, based on which software requirements might be adjusted.

week 6-7 concentrated work on wildlife tracks editing and if possible feedback from scientist and general audiences.

week 7-8 concentrated work on exhibition presentation in collaboration with Ivar van Bekkum

Software

The functionality of ElasticMapping software will be roughly based on the GPS data software that was developed for my earlier projects, both conceptually and technically.

The software (MilkMachine; several versions) is now developed in Microsoft .NET

Framework⁵. In the residence I would like to either continue working in this framework, or build a new tool from scratch in for example Processing Programming Language⁶ with the MilkMachine software primarily as a reference. The latter would open up for an open source approach in the future. As I am not a programmer myself, I expect this decision also to be depended on possible collaboration with programmer during Banff Centre residence.

The outline of ElasticMapping software is also inspired by research after already existing other software⁷ for GPS data that is mostly developed for specific users needs -sailors, sports activities, touristically trips- and/or commercial use.

Exhibition presentation

The main goal for exhibition presentation is to engage an audience with the realization that "locative media" has an daring artistically autonomous quality, and can be something different than the well known examples of what I sometimes call: "electronic travel guides" no matter how elaborate those might be.

Whether this goal is best served by a presentation of a series of developed animation, or a (interactive) presentation of the software is self, is for me at this moment difficult to judge. As I have a very good experience collaborating with Ivar van Bekkum in making a final well designed exhibition presentations, I intent to collaborate with him for the last 10 days of the residence to finalize a splendid presentation setting. If needed we might extend this collaboration in June in the Netherlands to make sure the exhibition piece is ready by July.

⁵ The Microsoft .NET Framework is a software framework that can be installed on computers running Microsoft Windows operating systems. It includes a large library of coded solutions to common programming problems and a virtual machine that manages the execution of programs written specifically for the framework. The .NET Framework is a key Microsoft offering and is intended to be used by most new applications created for the Windows platform.

⁶ Processing is an open source project initiated by Casey Reas and Benjamin Fry, both formerly of the Aesthetics and Computation Group at the MIT Media Lab. It is "a programming language and integrated development environment (IDE) built for the electronic arts and visual design communities", which aims to teach the basics of computer programming in a visual context, and to serve as the foundation for electronic sketchbooks.

⁷ Google earth KML format, runkeeper iphone-software, several photo-geo tag software, GPSbabel, GPS action replay, everywhere iphone application,

Presentation requirements (see also estimated budget)

beamer projection
sound system
interactive setting with computer
and other 3D installation parts like lamps, screens table and chair
Series of autonomous prints
materials (tape tools, test material)

software requirements:

notification: not all described features need to be totally finalized during the residence: needed is though a serious new step in the direction of a usable editing software and enough experience to move on with theoretical reflection and technical development later.

manipulations:

The software needs to be able to load any GPS track, and edit/manipulate them:

- shrink and enlarge (x, y and z as)
- integration of elevation data
- rotation in 3D
- cut and paste,
- increase and/or decrease the speed, relative and absolute
- repeat certain parts several times (Like the refrain in a song)
- drawing tool for a fictional path or parts of paths
- erase tool

notification: all manipulations need to be also possible performed on a selected part(s) of the route

visualizations:

- visualization of the track either in line or in dots
- thickness of line depending on speed and/or other parameters (elevation)
- color of the line depending on speed and/or other parameters (elevation)
- animation of the tracks on any chosen "ground overlay" (background) also transparent
- playing of several tracks simultaneously
- controlled zooming and panning over animated route(s)
- zooming in and out on extreme level differences.
- creating scenes and combinations of scenes

player

- the possibility of building scenes, movements, so that the data visualization decisions can be saved and replayed on their own, or as a sequence of related animations/scenes with an exactly pre-edited rhythm and pace.
- dedicated "camera angle"

import and export formats

the following (export) formats need to be supported in order to make it possible to use/upload the tracks in other applications:

- G7t
- kml (visualizations included)
- gpx (for re-use the field with mobile GPS devices and or smartphones)
- motor comments for robots, and or drawing machines (think of rapid prototyping, laser cutting and so forth) dance instructions so that live dancers can execute the tracks in a theatre setting.
- exporting of scenes as animations in .mov, .avi or .flv format with an transparent or any chosen background, in order to import animations in existing video editing software