

System Requirements: DigiMergo

PUM 2014

This is a living document and will change over the course of the project!

Field (v.x) indicates version on particular requirement.

Document version: 0.1

<p>Shall: Must be included in the software released at end of project. Priority 1. Should: Will be done after all the “shall” requirements if there is time. Priority 2. If time permits (ITP): These requirements are features that are planned for later release or are experimental – build only if there is plenty of time left. Priority 3.</p>

Scenario editor

Standalone application for building scenarios and resource packages.

Shall

- S.R.1:** (v.1) Have ability to create, edit and save scenarios as well as resource packages.
- 1.1: Finished scenario will be saved to file (xml?)
 - 1.2: Finished resource package will be saved to file (xml?)
- S.R.2:** (v.1) Separate scenario creation & editing from scenario execution (playing the game), since game clients may change in the future.
- 2.1: A scenario with an accident site defined in 3D must still be playable using the 2D game visualization.
 - 2.1.1 Will require translation of 3D environment to 2D (manual or auto-generated?)
Limitation: A scenario designed in 2D does not have to be playable using 3D visualizations.
- S.R.3:** (v.0.5) Build 3D views for scenarios
- 3.1: Accident site views (multiple or single sectors), “uppsamlingsplats” site
 - 3.1.1 New view “genom-vindrutan-rapport”: view of accident site from defined location in 3D environment.
 - 3.2: Include 3D models of buildings, objects of interest (cars, trees, etc), and patients
 - 3.2.1 3D models of environment can be modified to indicate danger or damage (accident), e.g. burning building, crashed car, poison gas.
 - 3.3: Patients can be placed inside a bus, or by a burning building, etc. in order to create a realistic scene.
- S.R.4:** (v.1) Build 2D views with the ETS look and feel.
- 4.1: All views, including meta views (transport times, overviews)
 - 4.1.1 New view “genom-vindrutan-rapport”: 2D camera image, film or Google Maps street view (or similar)
 - 4.2: Include possibility for adding custom graphics (e.g. images for accident scene)
- S.R.5:** (v.1) Be possible to place scenario on a map, regardless of 2D or 3D
- 5.1: Calculate driving times using the map.
 - 5.2: Manually place resources on map (e.g. location of ambulances at exercise start, location of hospitals)
 - 5.3: Possibility to specify pre-define spot for “brytpunkt”
 - 5.4: Possibility to specify pre-define spot for “genom-vindrutan-rapport”
- S.R.6:** (v.1) Be possible to create scenario without using map
- 6.1: Driving times and distances must be entered manually
- S.R.7:** (v.0.5) Have a “script” (manus) module, where instructor defines what will happen during the exercise
- 7.1: Instructions (körschema): When, what, who, where, how
 - 7.2: Scriptable events that will occur at given time in a given order
 - 7.2.1 Event behavior: automatic, prompted, manual
 - 7.2.2 Event type suggestions:
 - Environment: fire spreads, fire contained, gas leaks, building collapse
 - Resource: national resources arrive, other organization arrive, transport breaks down/delayed
 - Patient: death, health change, leaves area (if can walk)
- Note: Interface inspired by timelines – the scenarios is like a movie.*
- 7.3: Pre-define “jump-points” in time for long scenarios. (e.g. at 18.00, patients x,y,z are still at accident site, fire burnt down, transports at point a,b,c, hospital h1 overflowing, etc)
Instructors can move game state to this point in time during game play.
- S.R.8:** (v.1) Be able to set default resource package to be used for a specific scenario
- S.R.9:** (v.1) Use a keyboard and mouse interface.

Should

S.R.10: (v.1) Script (manus) should be exportable to pdf for printing and used as a physical tool during exercise.

If time permits

S.R.11: (v.0.5) Scenario forks: have more than one pre-defined sequences of events that the instructor can choose from during play.

Game client

The actual game simulators. Existing clients utilize touchscreens to simulate ETS whiteboards.

Shall

- S.R.12:** (v.1) Visualize "genom-vindrutan-view": surveying the accident site from the ambulance.
 - 12.1: 3D version: camera view of the 3D environment – accident site
 - 12.2: 2D version: custom image, video or modified "Street view" (Google) connected to a map.

Should

- S.R.13:** (v.1) Change (or improve) flip interaction
- S.R.14:** (v.1) Change (or improve - clarify) the interaction to transport patients of (and resources) from one view to another.
- S.R.15:** (v.1) Change (or improve) the representation of "incoming" objects on a view

If time permits

- S.R.16:** (v.0.5) Build accident scene client in the 3D environment (how to interact? – touchscreen not great in three dimensions).

Game manager

Instructors' main tool for controlling a training exercise. Manages server, connected clients, scenario & game control, logging, and result visualization (after action review).

Shall

- S.R.17:** (v.1) Be a standalone application running on a separate unit – not one of the clients
Note: will be used from setup to finish of an exercise, thus the interface must be segmented to support the current instructor task. Previous segments should never be lost/removed.
- S.R.18:** (v.1) Use loose couplings between interface and functionality. In the future we would like to control (some of) the manager from remote devices (phones, tablets, etc). Distributable UI.
- S.R.19:** (v.1) Use loose couplings towards clients. Possible to use the same “back end” with other ways of playing the game (using other “clients”) that will be explored in the future.
- S.R.20:** (v.1) Control server settings and status
- 20.1: View connected clients
 - 20.2: See server information (addresses, ports, etc.)
 - 20.3: This is not main focus of game manager, and should thus be “hideable” when not used.
- S.R.21:** (v.1) Setup up and specify training exercise
- 21.1: Choose scenario-file to be used
 - 21.1.1 View information on required (recommended) number of displays for scenario.
 - 21.2: Choose resource package (scenario may suggest default)
 - 21.3: Choose game “strictness”. What participants are allowed to do – e.g. can a patient be moved while undergoing treatment. In some exercises low strictness is appropriate.
- S.R.22:** (v.0.5) Monitor ongoing exercise
- 22.1: Select and display any game client’s view in miniature.
 - 22.2: Display script (manus)
 - 22.3: Display overview of current scenario status (number of patients on accident site, arrived at hospital, resource utilization, etc)
 - 22.4: View live log data – filters required.
- S.R.23:** (v.0.5) Control ongoing exercise
- 23.1: Start, pause, resume and stop exercise
 - 23.2: Display script (manus)
 - 23.3: Manage which view is displayed on which client.
 - 23.4: Move time ahead to a new point in the exercise
 - 23.4.1 Skip ahead to pre-defined points in time in the scenario (e.g. at 18.00, patients x,y,z are still at accident site, fire burnt down, transports at point a,b,c, hospital h1 overflowing, etc)
 - 23.4.2 If no pre-defined points exist in scenario, instructors must manually move patients and resources.
 - 23.5: Control events
 - 23.5.1 Skip pre-defined events in scenario
 - 23.5.2 Respond to prompted events
 - 23.5.3 Create new manual events (inspel)
- S.R.24:** (v.1) Display after action review (see result visualizer)
- 24.1: Export log data to excel
- S.R.25:** (v.1) Use mouse and keyboard interaction

Should

- S.R.26:** (v.0.5) Be able to preview scenarios
- S.R.27:** (v.0.5) Include a control for quality indicators during gameplay monitoring/control.

If time permits

S.R.28: (v.0.5) Build tablet controller (android or windows)

28.1: Use tablet device to check quality indicators and log these checks

28.2: Take live notes -> logged

Result visualizer

View within game manager which presents results as graphs and numbers for immediate after-action review.

Shall

- S.R.29:** (v.1) Have graph displaying number of patients on accident scene over time
- S.R.30:** (v.1) Have graph displaying number of triaged patients on scene (how many red, yellow, green) over time
 - 30.1: Visualize where they were thought the exercise
- S.R.31:** (v.0.5) Data which displays number of possibly avoidable complications & casualties
- S.R.32:** (v.1) Be easily projected or displayed on large screen for group discussion.

Should

- S.R.33:** (v.1) Display quality indicators and whether they were fulfilled

Is time permits

- S.R.34:** (v.1) Graph visualization selection (line, bar, pie, etc.)

Overarching guidelines

Guidelines are not explicit requirements, but should be used as guiding lights when developing.

Language and localization

Use English in interfaces and code

If time permits: use localization files for easy translation.

Code

Implement in WPF C#, using Visual Studio

Use existing GIT-versioning; create new main branch for the project (PUM 2014)

Update

Update logger whenever new functionality added (e.g. “genom-vindrutan-rapport”)

Design

Simplicity is key - simple rather than “cool”

Flexibility – for instance, scenario editor must not depend on the any particular game client.

DigiMergo should aim for using a back-end (server, manager, editor) that is not dependent on the front-end (the game client – the actual thing which personnel train with).