Flood Emergency Interaction and Visualization System

Rui Nóbrega, André Sabino, Armanda Rodrigues, Nuno Correia

IMG (Interactive Multimedia Group), DI and CITI/FCT, New University of Lisbon
{rui.nobrega, amgs, arodrigues, nmc}@di.fct.unl.pt

Abstract. In this paper we describe a visualization system for an emergency simulation. We start by presenting a flooding emergency case scenario and all the elements that are involved in it. Then we describe the design decisions that were made in order to create a credible representation of the scene. This includes using a game engine to render the scenario using a three-dimensional terrain, with objects and information retrieved from geographic information systems. Additionally we describe experiments with new touch and tangible devices that support a war table like interaction with the simulation. We further describe our ideas for the emergency interface and conclude describing the directions for future work.

Keywords: Emergency, simulation, interface, interaction

1 Introduction

Visualization and interaction are key aspects in Emergency Management. Timely decisions depend on having the right tools, available in a way easily perceived by the users. In this paper we describe the user interface of the LIFE-SAVER project, and discuss our choices in the scope of the ongoing work.

Project LIFE-SAVER was created in the Portuguese context of Dam Break Emergency Management [1]. In this country, the management of flooding emergencies, resulting from problems related with Dams, is planned through the guidance of existing special emergency plans. This project aims to develop a system that can effectively validate these plans, through simulation of the flooding natural phenomena and of all the actors intervening in the situation. Another important aspect is the definition of the plan in the simulation in a visual environment.

Creating a visual interface for the simulation presents several challenges. It is important that the users interact with a visually rich environment that allows a fast perception of the emergency that is being simulated. It should also be easy to use and be targeted for people with no special skills in computer science.

Having this in mind we created a prototype application that uses a graphical game engine to render a scenario where the user can observe and interact with natural and manmade disasters. Additionally the user can edit and define the agent behavior and test several scenarios.

LIFE-SAVER includes the development of a prototype of the Validation System, tailored to work with the Alqueva Dam Emergency Plan, which also involves another smaller dam, Pedrogão [2]. The Portuguese National Civil Engineering Laboratory, responsible for the development of the Plan, is a partner in the project.

The study of new interaction devices is another important aspect of this work, it is our intention to run away from the traditional mouse and keyboard configuration.

This paper focuses mostly on the interface and interaction aspects of the project and is organized as follows. The next section presents related approaches in user interfaces for physical phenomena and novel interaction approaches. Then the study scenario and the visualization components are presented. The following section describes the user interface and the interaction options. The paper concludes with preliminary conclusions and directions for future work.

2 Related Work

There are many emergency related projects with distinct features, system support and user interfaces .Here special focus was placed on projects that provide a relevant background to our work and have a multimedia interactive interface with interesting features.

The work reported in [3] explains thoroughly the multimedia system implemented to manage the Valencia's metro emergency plans. Just like the river dams, the metro needs to have an emergency plan. Their solution was turning the emergency plan into a multimedia software system that would integrate text, audio, video, 3D models, and animation to handle emergencies in underground metropolitan transportation.

The Tangible Media Group [4] from MIT has made some interesting proposals in the disaster simulation area. The Tangible DSS (Disaster Simulation System) is a collaborative tool for planning disaster measures, based on disaster and evacuation simulation using Geographic Information Systems. They use a tool called Sensetable[5] which is a table where an image is projected enabling interaction with viewed objects.

Another important interface is the Multi-Touch board developed by Jeff Han [6]. In some examples he uses his multi-touch interface with geographic systems.

While the use of the ArcGIS [7] set of tools (such as ArcMap) has proven useful, through the years of its existence, in the manipulation of geographically-referenced information, these are not primarily designed to perform real time simulations. Their use involves difficulty in keeping the data synchronized, while being displayed in ArcMap and subject to intense update at the same time. Moreover, typical GIS user interfaces, are also not appropriate. Another approach, more similar to our own, is to use a graphical game engine as a platform for the interface.

The HazMat: Hotzone[8] from Carnegie Mellon University is a simulation that uses computer game technology to train fire fighters against chemical and hazardous emergencies. The application looks and feels just like a 3D multiplayer first-person shooter and takes advantage of a game engine to render the emergency scenario.

In the LIFE-SAVER project we are using the real time strategy (RTS) game approach, similar to games like Age of Empires or Comand&Conquer. For that we use an open source game engine called Ogre3D [9]. Before the Ogre3D was chosen, other alternatives were considered. The other main candidate was Virtools [10] a product from Dassault Systèmes'3DVIA. Virtools is a framework solution to create 3D applications visually and not programmatically. Beyond these two solutions several other game engines exist. Some open-source projects such as Crystal Space, Irrlicht or The Nebula Device 2. There are also the Doom, Doom2, Quake and Quake2 engines which have been open sourced by Id Software. The best [11] game engines available are commercial closed-source libraries used especially in the FPS game genre. These are Doom3 and Quake 3 engines from Id Software, Half-Life 2 from Valve Software and Unreal Tournament and Unreal Tournament 2004 from Epic Games.

3 Case Study Scenario

It is very difficult to evaluate one specific Dam Break Emergency Plan (DBEP) as it requires testing very specific natural conditions, associated with flexible access to updateable information and the control of specialized rescue teams. One way to solve this problem is to computationally "generate" natural or induced flooding, as well as the response to the emergency, through simulation using one spatial-temporal representation.

The simulation is supported by the available spatial data from the downstream valley of the Dam. This information is handled by the agent-based simulation engine, which automatically feeds relevant spatial information to the user interface as needed. The simulator will be able to define emergency scenarios which will include available DBEP resources, actors and roles. The system dynamics is visualized and manipulated with a graphical interface representing the emergency scenario, while parameters characterizing this dynamics are registered during the simulation period, for later analysis.

The Alqueva dam is placed in the Guadiana River, about 150 km north of its base level, at Vila Real de Santo António. It is the largest dam in Europe, holding Europe's largest artificial lake.

In a catastrophe scenario, i.e., the collapse of the dam wall, there is a set of events that can be predicted. First of all, there is a formation of a wave, starting from the dam's position. The wave is expected to reduce in strength, while it advances through the valley, eventually becoming a flood hazard. According to studies on the wave, the civil protection infrastructure can only initiate evacuation operations after the first 30 minutes of flooding. Every person placed in the territory that can be reached by the wave before that time, has to escape to pre-determined safe spots. Sirens are placed in the valley to alert the population in the case of emergency.

After this first period of time, the civil protection teams will conduct rescue operations in order to evacuate all the potentially dangerous areas.

4 Visual Features

In order to visualize the scenario case, the multimedia interface must fulfill certain requirements. It must show a full-fledged three-dimensional terrain with a complete and accurate representation of the valley. Important structures and geographic features like the dam, the river or the retained water should be represented. Additionally, there is the need to represent vehicles, people, buildings and other objects. All these elements must communicate through the network with the simulator, which defines their actions. The interface must be intuitive, easily manageable by people with no special skills in computer interaction (fig.1).

Fig. 1. Emergency Visualization Application using the Interactive White Board.

The Computer Games [11] area provides good insights on matching some of these requirements. The idea is to make the emergency visualization interface to look as close as possible as a RTS game (Real-Time Strategy game). In a RTS game the world is seen from above, the units work as agents with limited artificial intelligence and information about each unit or building can be retrieved by clicking on it. Many RTS games have sound alerts and provide several shortcuts that enable a quick response by the user to any emergency.

In the LIFE-SAVER project we are using the RTS game approach, with an open source game engine called Ogre3D [9]. This graphic C++ library supports most of the features and effects currently available in games. Additionally a plug-in to create the terrain was necessary and we used the PLSM2 (Paging Landscape Scene Manager 2) [12].

4.1 Requirements

The LIFE-SAVER Visualization application features several components, most of them inspired by RTS games or by geographic information systems.

Visualization supports multiple maps. The user should be able to see the same map or region in different ways. This includes seeing the map in false colors view, in satellite view, military charts view or black and white. The map can be scrolled in any direction, rotated, zoomed and tilted (to better observe the three-dimensional display).

Almost all RTS games have an overview window where the entire world is represented. The LIFE-SAVER will also have a RTS fashion navigation mini-map. The mini-map should enable fast moving to any map position, and fast situation highlight for any emergency situation.

Special situations should also be highlighted with sound warnings and colored arrows pointing in the direction of the events. Danger situations include, for example, a building being hit by the wave, a water-surrounded group of people or a falling bridge.

Roads are important because this is where most of the simulation will take place. Besides its visual representation, it is important to have an internal graph representation. This graph is used to determinate the actions of each agent, vehicle or person. The graph must be stored as a suitable data structure, to be able to take advantage of search-path algorithms, (e.g., Shortest-Path). Since the terrain is three-dimensional, so are the roads and all the objects in a road take in account the inclination of the terrain.

One of the features is the capability to load information layers from external systems, including shapefiles from ArcGIS [5]. This facilitates loading information like building locations, flooding areas, road blocks, bridges and all the map information that can be represented by points, lines or areas. It is also important that the user is able to switch on and off each layer.

Selecting one object shows important information in a console near the mini-map. It also shows all the enabled actions for the object. Important information, like the identification of the object, is shown directly above the object. When selecting a group of objects, all the common data should be viewed in the graphic console. Agent information can also be edited, including the agent position and its range of action.

Other features include saving the simulation. In disaster simulation, it is always important to be able replay the simulation to allow for detail observation. The evolution of the simulation is recorded and it is possible to move back and forward with it and save it in a file for later viewing.

Most of these features are already implemented and will be described in detail in the next chapters.

5 Simulation Component

The visual environment is supported by a simulation component, which is controlled by the interface and controls all the agents and their behaviour.

The agent-based simulation [13] system integrates GIS data with an Object Oriented, Individual-Based Modelling approach, providing a successful simulation platform for the emergency plan validation process. The simulation model is mainly concerned with the location-based and location-motivated actions of the involved agents, providing data describing the likely effects of a specific emergency situation response, upon which the acceptance of the plan can be supported or changes suggested.

The design of the simulation engine follows closely the Dam Break Emergency Plan (DBEP) requirements. The definition of the DBEP requires prior identification of the entities involved. These entities may be divided into two categories: geographically bound entities, whose role is the execution of a set of actions in the scenario; and organizational entities, responsible for the coordination of emergency response actions. The entities responsible for the coordination of the emergency response operate outside the scenario, and are not physically interacting with it. The entities responsible for the execution of the plan’s actions are directly interacting with the scenario, and their mobility characteristics (moving by foot or using a specific vehicle) are one of the two main defining aspects from an emergency response point of view, the other being their expertise characteristics (medical personnel, firefighting personnel, etc.).

Given the special relevance of the entities locomotion characteristics, for the emergency plan, a previously determined type of entity is directly linked to its corresponding agent. Therefore, agents will be defined based on possible travel characteristics.

The geography of the scenario is a major factor for its characterization (as seen with the definition of the agents). The simulation model must take into account those characteristics, and be parameterized by it. The geographic information involved in the project is currently managed in a Geographic Information System (GIS). The simulator will be fed GIS data and this information will be used as a constraint to the various agents contemplated by the DBEP. A geographic data stack (Feature Space) is used to manage each geographic feature (e.g., roads, water streams, buildings, etc.), allowing the confrontation of a specific feature (or set of features) to an agent as a movement constraint.

Each agent has a set of actions (either predetermined as standard response or imposed by the entities responsible for the response organization) and while the scenario evolves, the emergency response will be the result of the coordination of several agents, each with specific behaviour characteristics and movement constraints.

To accomplish the specific evolution of one agent's actions, a priority system of actions is proposed. Each agent will initiate the scenario response with a predetermined set of actions, and, if seen fit by emergency response coordinators, new actions with higher priority may be issued, or even a reorganization of the previous actions can be required to face a specific scenario configuration.

The prioritization of actions enables the coexistence of routine operations and quick response actions by the agents, and allows the definition of a generic interface to deal with the agent's actions.

Demographic information is also important, and a database with information gathered in the location, contemplating the characterization of the people involved, health conditions and location, will be available.

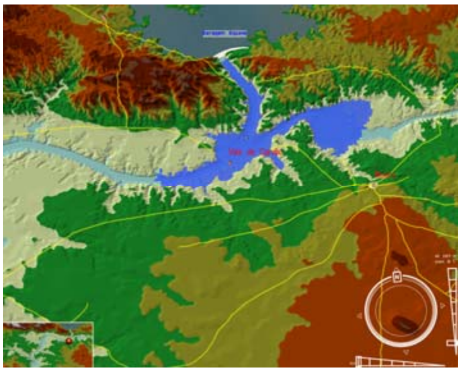

Fig. 2. Simulation of the flooding wave in the Guadiana River created by the failure of the Alqueva Dam.

6 The Visual User Interface

Presently we have created a prototype of the emergency testing application. Currently the program loads a three dimensional terrain. It loads geographic information from GIS data, namely ShapeFile data from ArcGis. There is a navigation system implemented, with sliders and a mini-map. The user can turn on and off several layers of information. The interface allows the selection of agents and the definition of actions for them. Currently it supports and presents a simulation of a flooding wave (fig. 2).

6.1 Design Architecture

The entire application was built using C++, using an object-oriented and event-based approach. The programming infrastructure is provided by Ogre3D game engine. Two C++ libraries were created (fig.3). One called BaseProject acts has a framework that takes care of all Ogre3D initializations, IO management, scene and level context and sound management. The other library is UtilityProject, an utility toolkit used to create simple custom objects, custom lines, custom materials and additional useful resources.

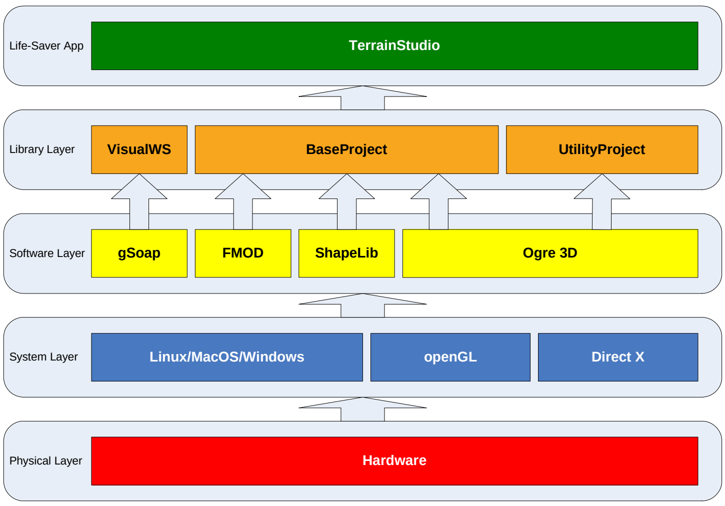

Fig. 3. Diagram of the Visualization Component. The LIFE-SAVER project is responsible for the two top layers.

The code was organized with a set of Manager Classes. These follow the “Singleton” code pattern, which means that for each class there is only one instance of the class running. This is important when, for example, it is necessary to have a global InputManager class that must be accessible from many parts of the code and has to be unique in the running program.

One of the manager classes is the StateManager. This class runs the main render loop of the program. The main advantage of this class is that it has a stack of Scenes. Each Scene is a state of the program and it can be a new level or a configuration scene. Each scene can be "pushed", "popped" or redirected to other scenes. This allows the fast creation of Scene flows.

Another important method was the use of an event based approach. When an object interacts with the mouse it is registered as a listener in the Input Manager and then the desired handlers are implemented in the object's class. This allows a decentralized approach, where each object has his input code, as opposed to having all the input treated in one place.

6.2 Graphic Implementation and Challenges

Through the partnership with LNEC, a large quantity of relevant data was acquired. This data includes maps and geographic information including digital information about roads, villages, buildings in risk, flooding zones and flood wave parameters.

To generate the 3D terrain was necessary to create a gray scale height map. This map was generated taking into account the contour lines of the terrain and the topographical height points of the river valley.

Using the PLSM plug-in of Ogre3D the map was divided in several small squares that are dynamically loaded as needed.

The representation of elements in the scenario required some developments in the computer graphics area. For example, for the roads representation it was necessary to create a line with thickness that followed the terrain's relief.

Besides Ogre3D other external C++ libraries were studied and integrated. To open ArcGis data the Shapelib [14] was used. To use sound the FMOD [15] library was tested. To create visual GUI interfaces with common window components such as buttons, editboxes or checkboxes the Crazy Eddies GUI library (CEGUI) was used.

A web service was established between the Visual component and the Simulation component. To use the web services it was necessary to integrate an open source library called GSoap [16].

6.3 Interface

The current application interface is essentially centered on the world navigation and on the simulation agent interaction options. Being a project related with emergencies planning and simulation the prototype application is designed to be used in a control center where many people interact collaboratively. Because of that we elected as our primarily input device the Interactive White Board (IWB) and the interface reflects that. All menus can be accessed using only the left click (equivalent to the screen touch).

The navigation is made using the bottom right interface. The knob rotates the world indicating the North, the vertical slider changes the zoom and the horizontal slider changes the pitch of the viewer enabling a 3D perspective. To move the world the user only has to drag the terrain. This interface has similarities with the Google Earth's configuration.

The left mini-map allows a fast recognition of where the user is. If the user clicks on it the viewing area is redirected to that area. This is a similar behavior to the RTS games.

The graphic user interface (GUI) features several menus as seen in figure 4. When the user selects an agent or a terrain feature, additional information is displayed on the Information menu. The Options displays the available options for that specific agent. The Simulation menu controls the simulation flow. An additional menu is the Layers menu that allows the user to switch on and off layers of information.

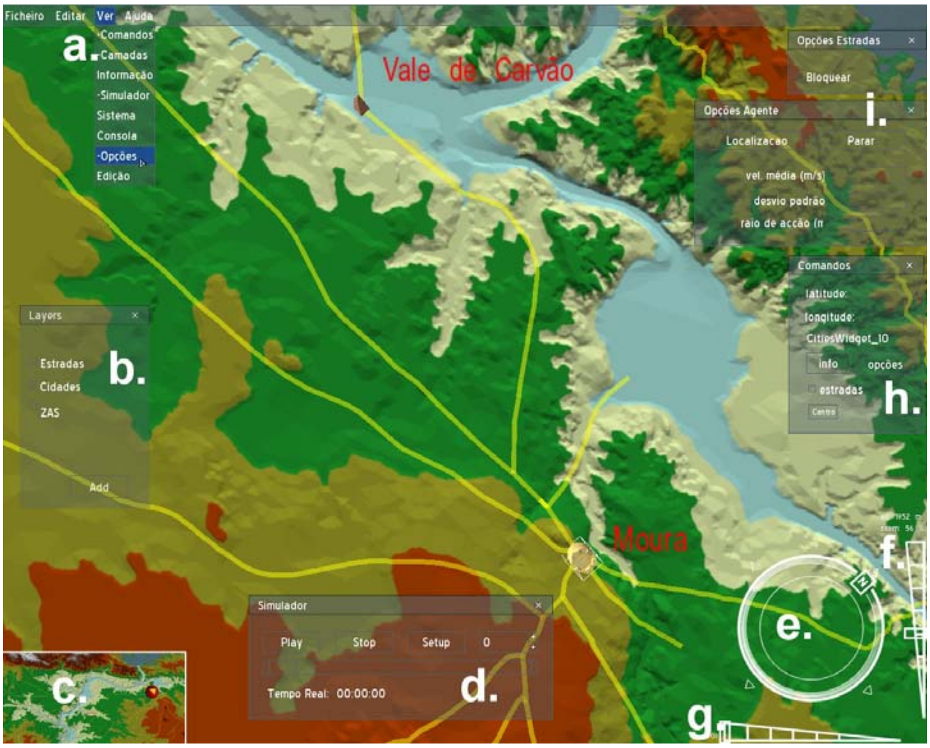

Fig. 4. Visual Interface: a) main menu, b) Layers, c) mini-map, d) simulator controller, e) map rotation, f) zoom, g) pitch, h) command console, i) agent options. To move the map the user has to drag the map.

7 Interaction

One of the main goals of the LIFE-SAVER project is to research new forms of interaction that depart from the traditional mouse and keyboard setup.

There are several devices that are being tested to be used in the LIFE-SAVER project. What is important about each device is its availability, degree of supported interaction and amount of people that can interact at each time with the application. Currently, we are supporting the mouse, digitizer pen and the touch-sensitive board. Additionally several studies are being made to use collaborative multipoint interfaces.

7.1 Interactive White Board and Tablets

The Interactive Whiteboard (IWB) is a large white interactive board where the computer image is projected and enables navigation on the screen, by touching it. It only supports one touch at any given time. Whenever the board is touched, it assumes that the touch position is where the mouse cursor is. The idea is to use the IWB as the central piece of the civil protection command room, placing it in a wall or on a table (fig.1). The IWB is a very practical hands-on device but has some limitations. It only handles absolute position, when the user touches the board, and only relative position when dragging on the board. Mainly, it only supports left clicking. A right click button is available, but it is not intuitive and easy to reach. Another handicap is that users usually interact with their fingers and fingers tend to be less precise than a pen or a mouse. There are also some synchronization problems between the position touched by the user and the cursor's position.

In this project, the IWB is used with large buttons and draggable areas, where the user can do all the world moving operations with his hands.

The same interface that works with the IWB is naturally usable in a Tablet PC. For demonstration purposes the Tablet is usually used due to its portability.

7.1 Multipoint Research

The main disadvantage from the IWB is that it only allows a single touch point. This makes the interaction rather limited. Using a multi-touch interface will lead to more invisible interfaces. All the operations will be dealt with finger gestures. We will be able to rotate and zoom with two fingers, dragging them on the board [6].

To create a multipoint interface several solutions were studied [5], including one using computer vision.

Our multipoint device is a table with a glass top (fig.5). Beneath the glass there is a common USB camera aimed right up. The glass is covered with a white polyethylene plastic that makes objects far away diffuse and is see-through with near objects. On the top of the glass we place multiple objects that have special marks on the bottom. The goal is to capture the marks with the camera and use detection algorithms to determine their positions on the table.

The users interact with the computer by moving the several marked objects (fig.6). Each object is represented on the screen with a different colored cursor. The marks and detection algorithm used are based on the Augmented Reality Toolkit (AR toolkit) [19]. Besides the marker position it is also possible to detect if the mark is touching the glass and rotation of the mark. This allows creating "click" events and rotation events that work like knobs.

The main problems of this device results essentially from lightning conditions and the low frequency of the camera (30 fps).

This research is still in an initial stage and is not ready for real users' deployment but shows promising results.

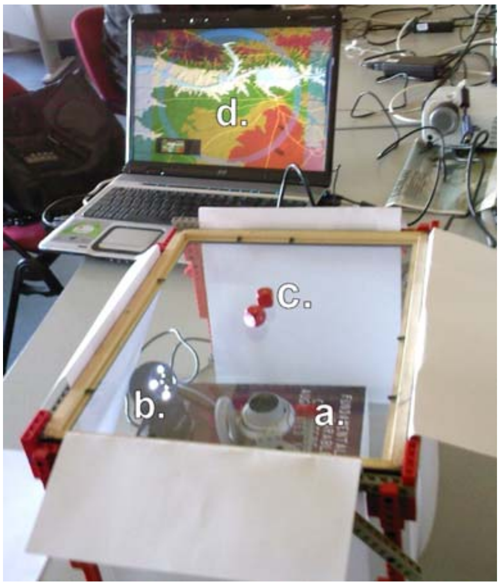

Fig. 5. Multipoint research device: a) USB web camera, b) illumination, c) two markers that sit on top of the glass, d) computer where the image is processed and the position of the marks is detected.

Fig. 6. Camera view showing the AR toolkit markers. The algorithm detects the position of the black squares.

8 Conclusions and Future Work

This paper presents a system for emergency simulation, focusing on the user interface and design decisions for this component. Using a game engine to support the interface, enables a rich set of interaction possibilities, which can be augmented as more actions are needed. Supporting multiple devices enables different usage scenarios, including decision makers, actors that are involved in the emergency, and the general public. In the system, we are also experimenting with several metaphors for navigating in the emergency area and further evaluations will be carried out. In the next months, we will be testing the different devices while adding functionality and evaluating the new features with the intended users of the system.

Acknowledgements

Project LIFE-SAVER is funded by the Portuguese Foundation for Science and Technology through grant PDCT/AMB/57247/2004. We would like to thank LNEC (Laboratório Nacional de Engenharia Civil - National Civil Engineering Lab) for all the geographic and scientific information and support. We thank all the people in IMG (Interactive Multimedia Group) at FCT-UNL.

References

1. Betâmio de Almeida, A., Matias Ramos, C., Santos, M. A., Viseu, T.: Dam Break Flood Risk Management in Portugal. LNEC, Lisbon (2003)
2. Santos, M. A., Gamboa, M.: Plano de Emergência Externo da Barragem de Alqueva (Alqueva Dam External Emergency Plan). Report nº 326/03_NTI, 169 p. LNEC, Lisbon (2003)
3. José H. Canós, J. H., Alonso, G., Jaén, J.: A Multimedia Approach to the Efficient Implementation and Use of Emergency Plans. IEEE Computer Society (2004)
4. Tangible Media Group, http://tangible.media.mit.edu/
5. Patten, J., Ishii, H., Hines, J., Pangaro, G.: Sensetable: a wireless object tracking platform for tangible user interfaces. In: SIGCHI conference on Human factors in computing systems (CHI’01), pp. 253--260, ACM Press (2001)
6. Han, J. Y.: Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection. In: Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology, ACM Press (2005)
7. ArcGis, http://www.esri.com/
8. HazMat, http://www.etc.cmu.edu/projects/hazmat_2005/
9. Ogre3D, http://www.ogre3d.org/
10. Virtools, http://www.virtools.com/
11. Kot, B., Wuensche, B., Grundy, J., Hosking, J.: Information visualisation utilising 3D computer game engines case study: a source code comprehension tool. In: Proceedings of the 6th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction: making CHI natural, ACM Press (2005)
12. PLSM2 Ogre3D Plug-in, http://tuan.kuranes.free.fr/Ogre.html
13. Sabino, A., Nóbrega, R., Correia, N., Rodrigues, A.: Life-Saver: Flood Emergency Simulator, ISCRAM, Washington D.C. (2008)
14. ShapeLib, http://shapelib.maptools.org/
15. FMOD, http://www.fmod.org/
16. Gsoap, http://www.cs.fsu.edu/~engelen/soap.html
17. Dix, A., Finlay, J., Abowd, G., Beale, R.: Human-Computer Interaction. Prentice Hall (1998)
18. Baxton’s site, http://www.billbuxton.com/multitouchOverview.html
19. AR toolkit, http://www.hitl.washington.edu/artoolkit/