AN ACTOR ARCHITECTURE TO DEVELOP GAMES FOR BLIND CHILDREN

Cyrille Bertelle
Cyrille.Bertelle@univ-lehavre.fr

Antoine Dutot
Antoine.Dutot@univ-lehavre.fr

Sylvain Lerebourg
Sylvain.Lerebourg@univ-lehavre.fr

Damien Olivier
Damien.Olivier@univ-lehavre.fr

Guillaume Prévost
Guillaume.Prevost@univ-lehavre.fr

Université du Havre
25 rue Philippe Lebon, BP 540
76058 Le Havre Cedex, France

KEYWORDS

Actors, Multimodality, Visual Disability, Specific Peripherals.

ABSTRACT

In this article, we are interested in providing a framework to develop games for visually impaired or blind children. In this context, we propose a tool set to ease the creation of such games based on an *actor* metaphor. As the possible outputs are multimodals (mostly not graphical) we provide an abstract library. The architecture is composed of an engine and a I/O layer. These concepts have led to the achievement of several games. This work is part of the european project TiM (Tactile Interactive Multimedia).

GOALS

Computers are tools successfully used even by young people with blindness or a severe degree of visual impairment. Games are a powerful way to learn for children (Svensson and al., 2002). Furthermore, a growing number of homes are computer-equipped. Despite this fact, very few games designed for visually impaired and blind children do exist. In addition, already existing games are dedicated to specific peripherals and most of them are quite expensive. Then users have to buy both the game and the device.

The aim of TiM project is to provide multimodal games (Archambault and Burger, 2000) which are adapted to the available hardware. TiM also furnishes a generic development library and a methodology suitable for game creators (whatever be their computer skills). In this article we propose a framework that fulfill these goals and requirements. The architecture is made of an I/O layer whose role is to render games to the player, an engine aiming at running the games and a language that allows to specify games to the engine. Games are designed in a abstract way using the TiM Language (TL) (Archambault et al., 2002), and run by the engine using actors (Terna, 1998). Then the I/O layer renders the running game in a way suiting the needs, disabilities and computer environment of the player.

ARCHITECTURE

We use the language described in (Bertelle et al., 2002). As shown in the figure 1, *Game* links several *scenes*, several *actors* or *classes*.

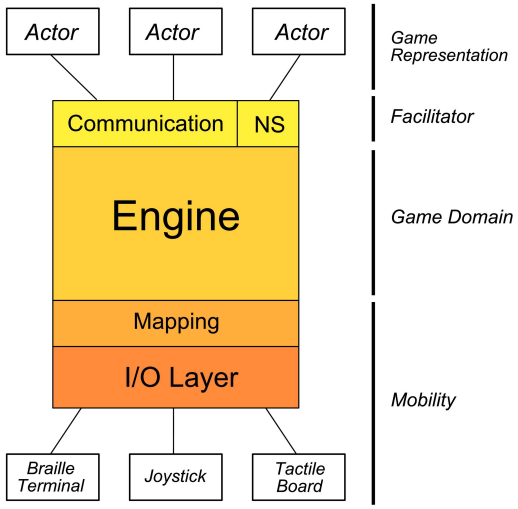

Figure 1: Architecture

There are two main types of objects in the engine: actors and passive objects. The actors have a behavior (Mataric, 1992) and a life cycle (Perception-Reaction), meaning it is perceiving its local environment and then reacting according to its goals. The passive objects only respond to external stimuli in a fixed way. As an example, actors can be non player characters, players or

an active environment. Passive objects can be a door, a wall...

Every object has a model and a representation. The model contains the characteristics of the objects in the engine (for example: coordinates, size...). The behavior defines the way it is reacting to stimuli. For active objects, the stimuli comes from the environment or from its perception. For passive objects, stimuli are only external.

The environment is a special actor containing all other objects as well as the global model representing the layout of a scene. It also has to define the rules of the game by restraining objects acts (for example, actor movement in a scene). A scene contains an environment as well as a transition rules to other scenes. The set of scenes describe the scenario.

Every interaction between objects is based on communications and they can be synchronized or not. Observers convert the model of every object into a representable form for the I/O layer.

The architecture allows two types of communications: a synchronous and an asynchronous one. The asynchronous one lays on a letterbox defined in the naming service. The synchronous one relies on a direct communication between objects.

GAMES

The main objectives of the TiM project are both to design new games and to adapt existing games. The second part is more difficult since it forces us to convert visually based games into games suited for blind children.

Tim Journey

Many games have already been designed for distinct age categories. *TiM Journey* concerns 7 years old children. In this game, the player is lost in a mysterious island (figure 2) and have to find and collect sounds spread out in the landscape and gather them in a secret place. The island is divided in many parts, each one having a specific audio environment.

During its quest the player encounters several objects and actors. For example the player finds one of a *Rune stones* which tells him a part of the history of the seven sounds and gives a clue to solve the final puzzle. We create an audio layout using spatially localized sounds that allows the player to know where he is. The footsteps sounds allow the player to recognize the surface he is crossing (sand, forest, etc.).

As an additional help, the player must find a sonar that will allow him to highlight the very important sounds. An external narrator gives hints to the player when this one doesn't know what to do.

[1]Tool build by Philippe Balin in 1978

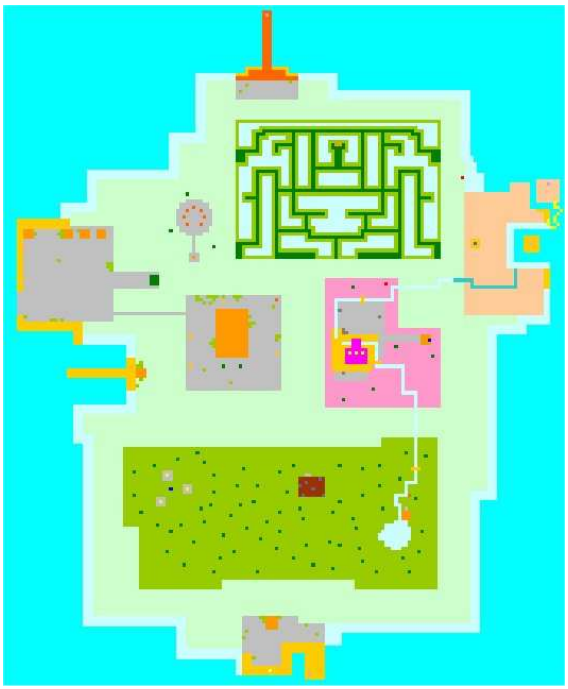

Figure 2: Map of the Tim Journey island

In the *Tim Journey* game, the player controls an avatar (representation of the player in the game) implemented as an actor (representation of the player in the engine). *Runes Stones*, doors ... are passive objects playing sounds when they are in your area of perception. They have no behavior, they just react to the player's needs.

The model is a 2D grid. A cell can contain a passive object or/and an avatar. A scene maps to an area of the island (and thus to an specific environment). A scene can also correspond to a main menu, an option menu or a cinematic scene (opening, transition, ending).

A menu contains many items with text. The blind children can use a braille terminal [1] (figure 3) in order to know what is written on the menu items. The words are displayed on the braille keys.

Figure 3: Map of a braille terminal

Reader Rabbit

The *Reader Rabbit* game is an adaptation of a french multimedia game ("Lapin Malin" [2]) for children from 2 to 4 years old. The child discovers four educational games helped by animals and by music. This game is designed for using another specific and common peripheral: the tactile board (figure 4).

Figure 4: Map of the tactile sheet

One of the four educational games is the *Bingo Basket Babies*. The child must associate a baby animal with its parent. In the original version, the child uses the mouse to select one of the parents on the screen. In the adapted version, the child uses the tabulation key at the bottom-left of the tactile board. For the tactile board, he can also use a specific touch (one of the four up keys of the tactile board) to hear the sound corresponding to the selected item. Here the texture tries to translate the mental representation of the animal. Then, he validates using the enter key at the bottom-right of the tactile board.

Other games

Some other games have been designed for the TiM project:

- *Hide and Seek* is a version of TiM Journey for 3/4 year old players.
- *MudSplat* is game where you have to kill evil aliens (much like space invader).
- *X-Tune* is a game in order to create its own sounds.
- *Magic Dictation* contains many activities like find a letter, spell a word, etc...

CONCLUSION

Many games have been developed using the framework described above. It has been showed that already developed games suit well to the visually impaired children needs. Moreover, the architecture happens to be very efficient for designing games. We provided games that use common peripherals in addition of dedicated devices. However, there is a blatant need to research a new way to adapt peripherals for the blind children. We still need to provide an authoring tool that would allow people without programming skills to create games easily.

ACKNOWLEDGEMENTS

The TiM project is funded by the European Commission, on the program IST 2000 (FP5/ IST/ Systems and Services for the Citizen/Persons with special needs) under the reference IST-2000-25298. We also thank our TiM partners for their various remarks. More information about the TiM project can be found at:
`http://www.snv.jussieu.fr/inova/tim`.

REFERENCES

Archambault, D. and Burger, D. (2000). Tim (tactile interactive multimedia): Development and adaptation of computer games for young blind children. *ERCIM WG UI4ALL & i3 Sping Days 2000 Joint workshop, Interactive Learning Environments for Children (Athens, Greece)*.

Archambault, D., Dutot, A., and Olivier, D. (2002). Tl a language to develop games for visually impaired children. *Computer Helping People With Special Needs, ICCHP Linz (Austria)*, pages 193–195.

Bertelle, C., Dutot, A., Olivier, D., and Prévost, G. (2002). Active objects to develop computer games for blind children. *GAME-ON, London, United Kingdom*.

Mataric, M. J. (1992). Behavior-based systems: Main properties and implications. *IEEE International Conference on Robotics and Automation, Workshop on Architectures for Intelligent Control Systems, Nice, France*, pages 46–54.

Svensson, H. and al. (2002). Tim: Tactile interactive multimedia computer games for visually impaired children. *Delivrable N° D05 - Users Needs - Final Report, IST Information Society Technology*.

Terna, P. (1998). Building agent based models with swarm. *Journal of Artificial Societies and Social Simulations*.

[2] Lapin Malin: Maternelle 1 - (TM) TLC Edusoft