Development of Games for SET-TOP-BOXES with Brazilian's Middleware GINGA-NCL

Aderbal N. Silva Junior Antônio C. S. Souza Luiz C. S. Machado

Instituto Federal de Educação, Ciência e Tecnologia da Bahia, DTEE, Brazil

Figure 1: Frames illustrating the games that are being produced on this project.

Abstract

This paper aims to present the process of creating games for set-top-boxes with Brazilian's middleware GINGA embedded on it, focusing on GINGA-NCL, since game's pre-production until the end of the demonstrration (DEMO) version developed by the Research Group of Information's Technology on Digital TV of the Instituto Federal de Educação, Ciência e Tecnologia da bahia in conjunction with the Core of Computational Modeling of SENAI / CIMATEC. All games in this text have been programmed in the language Lua coupled to NCL and the tests to evaluate the gameplay were performed using the emulator Ginga-NCL Player and the Virtual Set-Top-Box (STB) Ginga-NCL. Both softwares simulate the needed equipment for televisions that do not have specific tuner for receiving digital signal.

Keywords: digital tv, ginga, set-top-box

Authors' contact:
{aderbalnunes,antoniocarlos,luizcms}
@ifba.edu.br

1. Introduction

The market of electronic games have been highlighted in the last decade on Brazilian economy, being currently responsible for handling 87.5 million reais adding specific hardware and software according to the Brazilian Association of Developers of Electronic Games (ABRAGAMES). It was also discovered in the last survey that the software industry of games in Brazil has grown about 30% per year. Worldwide's economic viability is even more visible since 2003, when the segment of video games exceeded the revenues of the film industry [ABRAGAMES 2004].

The current scenario for the deployment of Brazilian System of Digital Television (SBTVD) is an opportunity to encourage the Brazilian scientific and technological development, since the guidelines and strategies of Decree No. 4901 of November 26th, 2003 of Brazilian's government that establishes the SBTVD [Brazil 2006], are to encourage regional and local industry in the production of digital applications. Looking deeper in the guidelines and strategies of the decree establishing the SBTVD, its inclusion intends to implement three key concepts: portability, connectivity and interactivity, field in which the development of games stands out among all other types of applications. However, being a relatively new technology, the construction of more complex and dynamic applications for SBTVD is still a challenge. The middleware GINGA set for the Brazilian's digital TV system, developed by the Catholic University of Rio de Janeiro (PUC-Rio) and the Federal University of Paraíba (UFPB) has some limitations, showing that still needs to be improved, given that it does not meet all needs of interactive applications' development.

The transmission of free signal of digital TV in Brazil is on implementation's phase and for this reason the majority of Brazilian's cities are not receiving the digital signal. So the feedback from researchers and developers who work in this area and were not part of the team responsible for the creation of middleware is essential for a stronger version of GINGA to be available.

The main objective of this work is to strengthen the activities aimed at production of digital content in Bahia, including its research centers in the area of Digital TV and production of electronic games. The research and development in the Group of Research on Information Technology, Line of Digital TV are

part of the project contemplated by FAPESB, August 2007 edict, for the development of innovative solutions in the field of information and communication technology. A partnership of the Instituto Federal de Educação, Ciência e Tecnologia da Bahia, with SENAI / CIMATEC, joining a team of programmers, designers, writers, specialists in sound effects, masters and doctors in computing area for the strategic positioning of the State of Bahia in the forefront of development and improvement of the transmission of digital TV in Brazil.

2. Detailed Steps of Development

The process of developing software for Digital TV presents peculiarities compared to other architectures such as PCs, cell phones and video game consoles, although the cycle of production is based on three stages set by Schuytema [2008] with pre-and post-production well defined. Below are the detailed steps of the methodology used by the Group of Information Technology - Research Line of Digital TV from Instituto Federal de Educação, Ciência e Tecnologia da Bahia with SENAI / CIMATEC for Digital TV Games Development Project.

2.1 Analysis of Functions Provided by the Middleware

In this case, GINGA was the middleware analyzed. The analysis seeks to exploit the resources of image, sound, storage devices and input and output. Due to possible problems with patents, GINGA-Java (or Ginga-J) had no official forecast for full release when this project started and that is why the development of Java applications for digital TV was discarded. Ginga-Java adopted GEM, a software block, which is part of Multimedia Home Platform (MHP), the european standard, and OpenCable Application Platform (OCAP), the north-american standard. The GEM's adoption to make the system more compatible with the other technologies became a delay factor for GINGA-Java because GEM was patented, as it was described by B4DTV. Then, the SBTVD forum made a deal with Sun Mycrosystems to develop an open version of GEM and that is why all the work described here was designed and implemented under GINGA-NCL using its support for the programming language LUA.

2.2 Game's Specification

Some ideas of casual games were chosen to be worked remembering the platform and physical limitations such as the remote control. These limitations had to be considered because they affect directly the gameplay of certain applications, such as the use of remote controls as joysticks that prevents excessively rapid responses and a greater care in the provision of keys to be used. These specifications and settings will be the Design Document (DD) and the basis for script and analysis of games geared for TV and other platforms that have characteristics in common with the games that are part of the project. This is the first model of the DD because the process of creating games is iterative and this document is in constant improvement in each step what prevents the development team to lose focus of the project.

2.3 Tools' Definition

This step consists in study, evaluation and selection of tools that will be used. After the definition of the game, the designers specified the graphical objects to be used in the game table to table exported in 3D Studio Max (to be in 3D), Adobe Flash (for animation 2D) and Adobe Photoshop (for 2D paintings and illustrations) in order to generate the frames of each object. For programming of game's routines in languages NCL and Lua the selected tools were Lua's Composer to write the code, the emulator GINGA-NCL player to run the first tests and virtual machine Ginga-NCL Virtual STB, that simulates the environment of a STB with the middleware GINGA-NCL embedded on it.

2.4 Implementation

It is conducted around the Development Group (DG) and begins after the definition of the visual identity of the games. Usually the team of arts and design make the first characters and environments for the game, which are turned into a test environment by the programming team. Then, in parallel or in sequence, the whole interface is designed (the main screen and other screens of the game). The features' programming and phases of the game are supervised by the DG. The activities are executed in parallel, making it a cyclic and incremental process, where each cycle implements a new stage, mission or play mode. Below are described the parts of implementation, the responsible staff and their respective functions.

2.4.1 Design and Arts

The game's script and DD will subsidize the illustration and modeling of characters, objects and scenarios which is built up by the group of design and arts. This stage consists of the following sub-steps: decoupage (stage where functionalities of the particular phase are identified and documented), conceptual art, storyboard, illustration or modeling, animation and animations exporting.

2.4.2 Sound

As the design and arts team, the sound team is subsidized by the game's script in the implementation phase, so the soundtrack portrays faithfully the atmosphere suggested by each stage or mission and its peculiarities in terms of effects. The stages of production in the group are decoupage, search and

cataloging of music and sound effects, editing, composition, adjustment and composition and editing with animations.

2.4.3 Programming

Like previous teams, the group will be guided by the stage or mission's script. The activities of this group are: decoupage, model solution planning, implementation and level design - where occurs the integration of products from the previous phase placing characters, objects and the sources of sound and also adding all the programming logic required to make the correct operation of these objects.

The step of testing and adjustments close the development cycle and starts a new one, which will create a new stage or part of the game.

The first cycle of development was used to design the demonstration version of the game. This version is being used to test the usability of the game and, soon players from outside the group of research and development will be called to play it. The players' suggestions will be taken for evaluation and the most important of them will be included as a possible fix on the subsequent cycle.

2.4.4 Beta Version and Final adjustments

This test will be conducted with a group of players after the completion of the final version of the game. Level of motivation and entertainment offered by the game will be considered on this phase. Noticed problems like breaks on the flow, components that maximize the difficulty to the point of discouraging casual gamers and general unexpected errors as well as significant suggestions for improved use of the remote control will be corrected before release to consumers, thereby ending the process of developing the game.

2.4.5 Completion

This stage is the end of the last adjustment. In the completion a fully game's evaluation will be held by the whole project team. There will be discussed and evaluated the produced games and the development process adopted as well. Each DG will present their report of overall development, which contains information such as problems encountered, solutions to problems and methodologies adopted. This meeting will result in a document with the project's analysis that will synthesize the knowledge and suggestions for possible improvements in a new version or sequel for the game developed. The second and third sub-steps from this stage involve the process of distribution and monitoring of the game. As most of the steps is common to many processes of development of electronic games, here was just explored the tool's definition and its implementation which are described in the following section.

3. Definition and Implementation of Tools

As shown previously, there are some differences between the process of software development for Digital TV and the same process on common platforms. In this architecture, as well as cellular phones, emulators are used to simulate the middleware in which the application will run, making the installation of the application in the STB unnecessary, at least while you are doing test procedures of implementation.

The games developed and presented in this article were run in Emulator GINGA-NCL Player, which can be found in the Brazilian Public Portal Software for the operational systems: Windows, Linux and MAC OS. Note that the package Java Runtime Environment (JRE) is a prerequisite.

Ginga-NCL was created by PUC-Rio to provide an infrastructure for submitting applications for multimedia / hypermedia under the declarative paradigm written in language NCL. NCL provides facilities for specification of interactivity aspects, space-time synchronization between media objects, adaptability and support for multiple devices [Brazilian Public Software Portal 2008]. Unlike HTML or XML, the language NCL do not mix the definition of the document's contents with its structure, offering a noninvasive control of both the document's layout (presentation space) and the exhibition time. NCL does not define any objects of media, but only reference these objects semantically together in a multimedia presentation. As Such, as quoted in Souza [2008], the use of language NCL is not sufficient for the development of games, because these applications require a control flow at runtime. Thus, developers of PUC-Rio provided a integration with LUA language (LUA-NCL).

LUA is a light but really powerful programming language, which was designed to extend applications. It combines simple syntax for procedural programming with powerful constructs for data's description based on associative tables and extended semantics. It thus allows the game designer to have some control over the language without having the need to follow the huge learning curve, usually associated with programming, as said by Schuytema [2008]. As said by Celes [2004], Lua is a scripting language, designed to provide meta-mechanisms that enable the construction of more specific mechanisms. In particular, developers of the games can provide adequate abstractions for writer and artist, simplifying the tasks of these.

Lua is dynamically typed and is interpreted from bytecodes to a virtual machine (engine). It has automatic memory management with incremental garbage collection. These characteristics make it an ideal language for configuration, automation

(scripting) and rapid prototyping [Soares 2007]. This language simplicity, efficiency, portability and low impact of inclusion in applications, makes it one of the most used programming languages in the world of entertainment and can be seen today in games from LucasArts, BioWare, Microsoft, Relic Entertainment, Absolute Studios, Monkeystone Games, among others mainstreams producers as cited by Souza [2008].

4. Results

In the second step of the methodology six games were chosen to be produced. Of these six games, one has already a beta version being on implementation's phase and another two are under the demonstration version production. For all these games was adopted a line of creativity where the visual theme overlaps the simple aspect inherent to casual games. The interfaces are more dynamic and interactive, which easily draws the user's attention due to the easy assimilation.

4.1. Sudoku

Sudoku was the first produced game and there is, actually, a demonstration version with most of its features. This is a logical reasoning game where each new game usually lasts 10 to 40 minutes depending of the difficulty level and the player's experience. The puzzle starts with some initial clues and the player has to fulfill the remaining grids with numbers from 1 to 9, so that no digit is repeated in any block, row or column. Despite the numerical appearance the game does not need algebraic knowledge of the player, as said before it is a logic game.

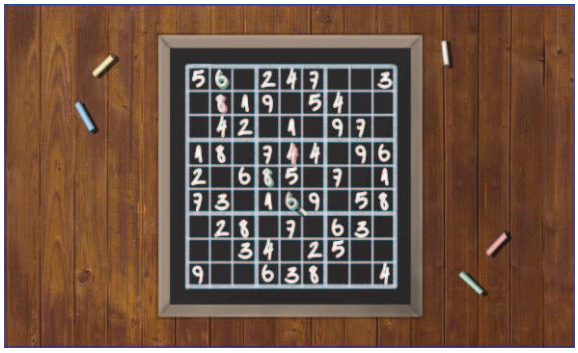


Figure 2: Sudoku's main menu.

The game was developed exploring the idea of using the colored buttons on the remote (as shown on Figure 2), the simplicity of handling, the games of chance inherent in reasoning, the limitation of using the memory emulator that simulates a platform and a design characteristic of board games as can be seen in Figure 3.

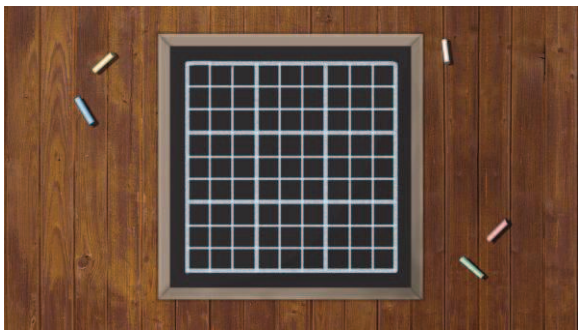


Figure 3: Sudoku's clean board.

Animated videos identified and coordinated as objects in NCL, were used to create the animated menu. The start of the presentation, the loop of animations, the interaction between the player and the screen through colored buttons and the script that describes the game itself (identified as an object of type LUA) were are referenced in NCL. The main code written in NCL, with all those references, will be run by GINGA-NCL, be it on the emulator or in the virtual machine.

Transitions are made through NCL simple connectors like "onEndStart", "onBeginStart" or "onKeySelectionStartStopAbort" that executes literally their descriptions. The trick to the flow dynamism in the NCL controlled part is on the production, merge and timing of the videos and the player's control.

Inside the LUA object called through the main code are the chunks (lines or blocks of code / commands) that describe the paths to load the entire graphical interface within the game and the variables that store the objects (chalk, board, numbers) its original position and size, in addition to the functions that govern the movement, appearance, disappearance and replacement of what is shown on screen, win or lose and life systems.

The win or lose system for board games of this type can be done through a finder function like:

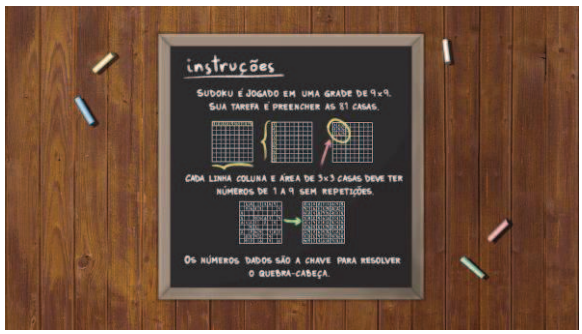
```
function whereami()
x  = math.floor((cursor.x-PIH)/EHQ) + 1
y  = math.floor((cursor.y-PIV)/EVQ) + 1
return y*TQV+(x-TQH)
end
```

Where PIH / PIV represents the initial position of the cursor in the horizontal / vertical, EHQ / EVQ are the spaces that the cursor has to walk in order to reach the next grid in horizontal / vertical and TQH / TQV is the total number of grids in horizontal / vertical. Combined with a generic function to fill the board (and also the array with the new information to be compared with the solution array) the code will be like:

```
function changeimage(target, num)

if target.fix == false and
tonumber(num) ==
rightgame[whereami()]then
target.img =
canvas:new('/imagens/'..num..'_ok.png')
target.num = num

if gameimplayn == rightgame then
event.post('out', { class='ncl',
type='presentation', area='fim',
transition='starts' })
IGNORE = true
end

elseif target.fix == false and num ~=
rightgame[whereami()] then
target.img =
canvas:new('/imagens/'..num..'_no.png')
target.num = num

playerslife= playerslife-1
if playerslifer== 0 then
event.post('out', { class='ncl',
type='presentation', area='fim',
transition='starts' })
IGNORE = true
end
else return end
end
```

As can be seen on this block, there is already attached to the function the change of images, the lose system and the correction with its penalties (reduction of player's life). Also the notification that determines the end of the game, consequently the end of LUA's script, and the return to NCL control which will be verified by the IGNORE on the handler of events that controls the input in the program. Note that the "var = var – 1" works on LUA. You just need to make sure that "var" is not NIL (null value).

Board games for GINGA-NCL as the Sudoku can use the same principle, applied here, where tables (arrays) with pre-defined positions according to their scenario can load random matches each time a player starts a new game. The loading of random files where arrays with different games are could be done through the "require" function. The possibility of loading different games from chunks of the source code allows the possibility to download new and different level games. In this game, the opening animation was made in 3D and all the gameplay (screen background, and cursor numbers) was created with 2D digital paintings and illustrations, including the sub-menus (Figure 4).



.Figure 4: Sudoku's sub-menu.

4.2. Wing Force

Wing Force was based on old shooters of vertical progression where players control an aircraft seen from above, such as River Raid of the extinct console ATARI 2600. The player must destroy enemy aircrafts, tanks, ships and interactive elements of the scenario as he travels through it collecting power-ups (improvements to the machinery of the aircraft) and keeping himself alive until the end of each stage and thus achieve the next level.

The source code of this game is severely more complex than sudoku's code, since you need to control a time flow with constant handling of the scenario, ship, enemies, etc. that should be controlled by co-routines (threads). The problem is that despite the support of co-routines by the programming language Lua, the virtual machine that simulates the STB with the Ginga-NCL does not offer this support installed. The algorithm is improved using the movement of variables to control the appearance and movement of enemies.

Today the enemies on the screen tend to bump against the player at a lower speed than the player's ship what gives the player enough time to evade. Right now the enemy's ships mechanism is being studied for better handling, be it random or fixed, to further minimize the problem of GINGA-NCL lacking of co-routines.

Some objects can explode in case of collisions. The collisions can be done by a comparator function of position in the handler of events.

Figure 4: Sudoku's sub-menu.

The scenarios are 2D paintings and enemy ships and tanks as other scenario's elements were modeled and animated in 3D and then exported frame to frame. The animations were completed in Adobe Flash, retouching the texture in Adobe Photoshop. The soundtrack is very fast, with typical effects of 8-bit consoles and a melody-based on 80s rock.

5. Related Work

Although the SBTVD's game development is something new, all the analysis about limits and possibilities of GINGA made by Souza et al.[2008] was a essential part of this work, as much as Piccolo et al.[2008] study about STBs and Digital TV architecture is particularly recommended for understanding of the operation of STBs and how

Development of games in Lua is perfectly describe in the whole Celes and Ierusalimsch work as much as development of applications in NCL is described in Soares at al.[2007] work.

6. Conclusions

The difficulty of developing more robust games for GINGA-NCL is visibly noticeable. Not just for the question of co-routines, but also by the limitation of the size of imported images and scenarios. By the close time of this article, the latest virtual machine that simulates the STB (GINGA LIVE CD V1.0) still does not support co-routines. The concept that most stood out in the proposed deployment of SBTVD, interactivity, is still not satisfactory because applications that are being produced and disseminated are still quite poor in terms of interaction between the user and the TV. While these application hopes to astonish users for their visual appeal they still provide little immersion an important conception in games.

Although the games presented here are almost finished, this work can still be seen as an initial step in developing applications for Brazilian's digital TV as there is still much to explore with GINGA-JAVA especially on development of applications aimed at interactivity and entertainment. It is presumable now that with the GINGA-JAVA's confirmation a new way to the development of games for STBs with Brazilian's middleware fostering interactivity, and conducting a more robust version of GINGA-NCL with more opportunities and less restrictions.

Acknowledgements

The authors would like to thank Fundação de Amparo à Pesquisa do Estado da Bahia and Ifba's CTPGP for the financial support and for believing in this project. Also it is important to mention B4DTV blog and DEVDTV discussion list the best source for those who want to develop absolutely anything for GINGA or just to keep in touch with everything about Digital TV on Brazil.

References

ABRAGAMES, 2004. Plano Diretor de Promoção da Indústria de Desenvolvimento de Jogos Eletrônicos no Brasil – Diretrizes Básicas, http://www.abragames.org/docs/pd_diretrizesbasicas.pdf, [Accessed 17 July 2009].

B4DTV – Blog for Digital Tv. http://b4dtv.blogspot.com/[Accessed 15 August 2009].

BRASIL. Decreto n 5.820, de 29 de Junho de 2006. Implantação do Sistema Brasileiro de Televisão Digital Terrestre - SBTVD-T. DOU de 27/11/2006. http://www.planalto.gov.br/ccivil_03/_Ato2004-2006/2006/Decreto/D5820.htm, [Accessed 20 June 2008].

Celes, W., Figueiredo, L. H. e Ierusalimschy, R., 2004. "A Linguagem Lua e suas Aplicações em Jogos", http://www.tecgraf.puc-rio.br/~lhf/ftp/doc/wjogos04.pdf, [Accessed 15 May 2008].

Openginga, 2008 ProjetoOpenGING. http://www.openginga.org, [Accessed 15 July 2008].

Piccolo, L. S. G., Melo, A. M., Baranauskas, M. C. C.[2008]. "Accessibility and Interactive TV: Design Recommendations for the Brazilian Scenario", Human Computer Interaction INTERACT 2007 11th IFIP TC 13.

Piccolo, L. S. G., "Arquitetura do Set-top Box para TV Digital Interativa". http://www.grupos.com.br/group/designcefet20051/Messages.html?action=download&year=08&month=5&id=121077537895923&attach=arquitetura%20conversor.pdf, [Accessed 15 May 2008]

Soares, L. F. G., Rodrigues, R. F., Moreno, M. F.,2007. "Ginga-NCL: the Declarative Environment of the Brazilian Digital TV System". Journal of the Brazilian Computer Society, v. 12, p. 37-46, 2007.

Souza, A. C., Machado, L. C. S., Sampaio, R. L. e Raimundo, P. O. , 2008. "Desenvolvimento de Jogos para TV Digital", 3° Congresso de Pesquisa e Inovação da Rede Norte Nordeste de Educação Tecnológica.

Souza, A. C., Machado, L. C. S., Sampaio, R. L. e Raimundo, P. O. , 2008. "TV Digital: Limites e Possibilidades Tecnológicas", 3° Congresso de Pesquisa e Inovação da Rede Norte Nordeste de Educação Tecnológica.

Schuytema, P. , 2008. Design de games: Uma abordagem prática, Brasil, Ed. Pioneira.