

Game Development for Brazilian's Digital TV: Copas

Antonio C. S. Souza Luiz H. N. Dos Anjos Pedro O. Raimundo Luiz C. M. dos Santos
Aline de O. Machado

Instituto Federal de Educação, Ciência e Tecnologia da Bahia- Brazil
LABRAGAMES- Laboratório de Realidade Aumentada, Jogos Digitais e TV Digital

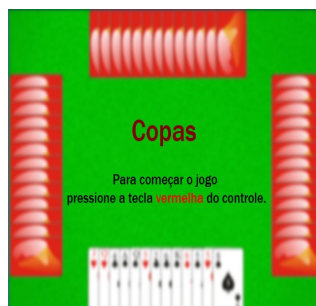


Figure 1: Frame illustrating the game produced on this project.

Abstract

This paper aims to present the process of creating a game 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 development process. The game in this text has been programmed in the NCL/Lua language. The tests to evaluate the gameplay were performed using the Virtual Set-Top-Box (STB). This software simulates the needed equipment for televisions that do not have specific receiver for digital signal.

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

Authors' contact:

{antoniocarlos,lhenriqueanjos,pedrooraimundo,luizcms,aline}@ifba.edu.br

1. Introduction

The market of electronic games has 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].

Tarquíneo Teles, founder and general director of Hoplon, says that the field in Brazil is expanding because of the large amount of qualified professionals in contrast with the ironically low amount of

companies, large or small, specialized in this kind of technological enterprise. Currently, people are less prejudiced towards this sector and obtaining investment is easier. According to him, it is a promising field, even though developing electronic games is a hard task [NPD Group 2010].

This study aims to reinforce the activities to produce software for digital TV. Later, the development of a game using GINGA middleware (distributed freely) will be shown. The version used was GINGA-NCL, developed by PUC-RJ, based on GNU/Linux system.

2. 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 that developed the game.

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 NCL language. 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 NCL language does 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 NCL language 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 (NCL/Lua).

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) and 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]. Its 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 mainstream 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. Copas

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 referenced in NCL. The main code written in NCL, with all those references will be run by GINGA-NCL, using the virtual Set-Top-Box.

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, merging and timing of the videos and the player's control.

The Lua script is where the game logic actually resides. When the script starts arrays that will be used by the game such as the deck, the "hand" of the player and his opponents and important functions – such as determining to which card the cursor is pointing – are initialized as global structures and variables, this is also when all graphical objects are loaded into memory to be eventually drawn to the screen.

In a game of Hearts, a common 52-card deck is divided among the four players, who then start playing with the task to finish the game with the fewest heart cards in his pile. Here the structures corresponding to these key elements of the game with their characteristics (x and y positions, cards in hand, etc.) and their actions (for example, choose the card to be played) were created, as shown in the code below:

```
norte = {x = 162, y = 0, mao = {},
monte = {}, escolher = escolher}
leste = {x = 531, y = 101, mao = {},
monte = {}, escolher = escolher}
oeste = {x = 1, y = 101, mao = {},
monte = {}, escolher = escolher}
jogador = { mao = {}, monte = {},
drawmao = drawmao }
```

The array *baralho* was created using 52 instances of the *Carta* class, these were created dynamically

using their value and suit attributes and the function below, it's worth noting that this function uses values and suits previously allocated in arrays (*valores* and *naipes*) to this end:

```

local function inicializar ()
  for i=1,#valores,1 do
    for j=1,#naipes,1 do
      table.insert(baralho,
        Carta.create(valores[i], naipes[j]))
    end
  end
end

```

After being initialized the deck is shuffled and distributed among the participants using the *embaralhar* and *dar* functions. After that both main and secondary redraw functions (the same functions called throughout the program's execution) are called to draw: a) the cursor, b) the cards from the opponents, c) the background and d) the player's hand as seen on Figure 2, so that then the player and his AI-Controlled opponents may start playing.

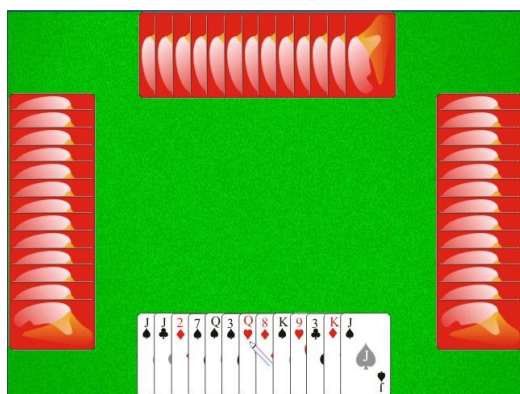


Figure 2: Game screen after initializing, shuffling, dealing and drawing the cards.

The game has four slots in the center of the screen to store the played cards for the round. These slots are also structures and have several attributes like the position they're going to be drawn on the screen and the card within. When the player selects a card it is copied to the corresponding slot and is subsequently removed from the player's hand, as shown in the following code:

```

elseif evt.key == 'o' then
  if jogador.mao[ondeestou()] then
    table.insert(slots, 1,
      jogador.mao[ondeestou()])
    table.remove(jogador.mao,
      ondeestou())
  end
end

```

In this section of code we can see the *ondeestou* function determining the card at where the player's cursor is, as well as the copy and removal of the selected card from his hand.

Similarly, after the human player makes his move, the AI opponent plays, choosing a card from his own hand and does the same copy and removal process as we can see on the code below and in Figure 3:

```

table.insert(slotw, 1,
  oeste.mao[oeste.escolher(oeste)])
table.remove(oeste.mao,
  oeste.escolher(oeste))

```

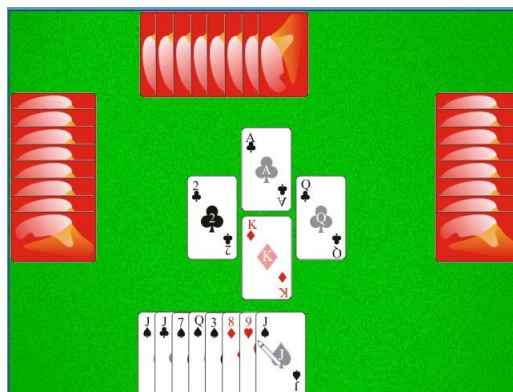


Figure 3: Game screen once all participants have played a card.

The section above shows that the *escolher* function was called. With this function the AI searches in his hand looking for the best card to play. When all the cards are on the table, the next step is to call the function that compares their values and adds the cards from the central slots to the pile belonging to the participant who played the highest card, while removes them from the central spaces. The scores are calculated by the end of every call of that function, with each card of hearts on your pile accounting for 1 point, and the queen of spades accounting for 13.

Once all the cards have been played the deck (*baralho* array) is reinitialized along with the other game variables. In this process only the score variables are kept and the game ends when one of the participants reaches a score of 100. The participant with the lowest score is the winner and the others are ranked in ascending order of score.

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.

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

6. Conclusions

The hardware of set-top-box must be affordable because its purpose is to allow many users to have the opportunity to receive the signal from digital TV. One of the characteristics of a low-cost hardware is a less processing and storage power. This is the main difficulty in developing games more robust with GINGA-NCL.

Interactivity, the concept that most stood out in the proposed deployment of SBTVD 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 applications are meant to attract users by their visual appeal, they still provide little immersion which is an important conception in games.

This work can be seen as an initial step in developing applications for Brazilian's digital TV as there is still much to explore with 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.

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].
- NPD Group. http://www.npdgroup.com/corpServlet?nextpage=corp_welcome.html, [Accessed 12 July 2010].
- 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”, 3º 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”, 3º Congresso de Pesquisa e Inovação da Rede Norte Nordeste de Educação Tecnológica.
- Schuytema, P. , 2008. Design de games: Uma abordagem prática, Brasil, Ed. Pioneira.