

## DESENVOLVIMENTO DE JOGOS PARA TV DIGITAL

**ANTONIO CARLOS SOUZA (1); LUIZ CLÁUDIO MACHADO DOS SANTOS (2);  
ROMILSON LOPES SAMPAIO (3); PEDRO OLIVEIRA RAIMUNDO (4)**

(1) CEFET-BA, Rua Emídio dos Santos, Barbalho, Salvador-BA, 71-2102-9525, e-mail: [antoniocarlos@cefetba.br](mailto:antoniocarlos@cefetba.br)

(2) ESAD/LASID/UFBA, e-mail: [lmachado@fapex.org.br](mailto:lmachado@fapex.org.br)

(3) CEFET-BA, e-mail: [romilson@cefetba.br](mailto:romilson@cefetba.br)

(4) CEFET-BA, e-mail: [pedrooraimundo@gmail.com](mailto:pedrooraimundo@gmail.com)

### RESUMO

O presente trabalho tem o objetivo de relatar o processo de desenvolvimento de jogos para o Sistema Brasileiro de Televisão Digital (SBTVD), que vai desde a investigação das possibilidades tecnológicas dessa arquitetura, da codificação do jogo na linguagem LUA até a execução no emulador GINGA-NCL Player. Tal emulador simula o SET-TOP-BOX (STB), equipamento necessário para conversão do sinal de transmissão digital para o analógico, com o middleware instalado, que no caso do Brasil é o GINGA (ambiente desenvolvido pela PUC-RIO e a UFPB). O processo de desenvolvimento dos games<sup>1</sup> baseou-se em uma narrativa simples com ambientes bidimensionais (desenhados) com conteúdos voltados para o entretenimento e para o exercício do raciocínio lógico, de forma que fosse possível avaliar a possibilidade do GINGA e do STB serem utilizados para execução de jogos. Os jogos apresentados nesse artigo estão sendo desenvolvidos por uma equipe interdisciplinar que envolve estudantes e profissionais das áreas de programação, roteiro, arte conceitual e design (modelagem, texturas e animações) de cenários e avatares.

**Palavras-chave:** TV digital, Jogos, GINGA, LUA, SET-TOP-BOX (STB).

---

<sup>1</sup> Nesse trabalho utilizaremos a palavra Game para designar jogos.

## 1. INTRODUÇÃO

Nos últimos trinta anos, os games têm marcado presença na sociedade contemporânea principalmente pelo viés do entretenimento, sendo responsáveis por uma grande movimentação financeira, e de acordo com a ABRAGAMES (Associação Brasileira de Desenvolvedoras de Jogos Eletrônicos) “o segmento de jogos eletrônicos mundial ultrapassou o faturamento do cinema em 2003” (2004, p. 4). Fator esse que vem despertando o interesse de diversos países e gerou a abertura de empresas na área de desenvolvimento de games, principalmente os casuais, como os jogos de celulares que demandam menos tempo e recursos para o seu desenvolvimento.

De acordo com a ABRAGAMES (2004) foram identificadas cinquenta e cinco empresas neste segmento e este número tende a crescer. Estes dados apenas retratam o grande interesse e entusiasmo que os games produzem em diferentes públicos - crianças, adolescentes e adultos. No Brasil, tem sido crescente o número de LAN Houses abertas, sendo que em alguns bairros das grandes cidades é tão comum encontrar estes estabelecimentos, como encontrar uma padaria ou uma farmácia. Também é crescente o número de títulos de revistas sobre games nas bancas de jornal, superando revistas com outros temas. Através de uma simples pesquisa, utilizando a internet, é possível verificar que o número de sites dedicados ao assunto é incontável.

Ainda segundo a ABRAGAMES (2004) países como a Austrália e a Coreia do Sul se tornaram exemplos de como políticas públicas adequadamente implementadas podem mudar o cenário de um país ou região. O mercado da Coreia do Sul atingiu 3,3 bilhões de dólares e atualmente emprega mais de 50.000 profissionais. A ABRAGAMES complementa indicando que o desenvolvimento de jogos cresceu nove por cento em países que não tem uma cultura nessa área, fato esse que ocorreu basicamente em dois cenários: I) Os desenvolvedores tornam-se competitivos no mercado internacional, devido aos baixos custos de desenvolvimento. II) Uma adequada implantação de políticas públicas de incentivo que fortaleceu o mercado interno tornando os desenvolvedores de jogos competitivos internacionalmente.

A análise da experiência desses dois cenários nos permite fornecer uma visão de como o mercado brasileiro poderá estar futuramente. O primeiro cenário apresenta o mercado prejudicado pela pirataria, que não atua no segmento de games para consoles (o maior do mundo) e é focado em exportação. Nesse cenário os desenvolvedores não produzem jogos com temáticas nacionais, pois não atuam no mercado interno e acabam, além de economicamente enfraquecidos, culturalmente prejudicados. Já os países do segundo cenário atuam no mercado interno além de desenvolver games para consoles. Outro dado significativamente importante é o tempo de maturidade das empresas nos dois cenários, que de acordo com a ABRAGAMES (2004), as empresas dos países com incentivo governamental (Coreia do Sul e Austrália) apresentam uma média de sete anos de existência enquanto aqueles sem incentivo a exemplo, dos localizados no Leste Europeu (República Tcheca, Rússia, Croácia, Hungria e Ucrânia) indicam uma média de quinze anos.

Alguns fatores externos foram identificados pela ABRAGAMES (2004), como sendo favoráveis a produzir impactos nas condições de crescimento da indústria de desenvolvimento de jogos no Brasil. Um dos fatores apontados foi a implementação da TV digital no país e o seu impacto em termos de convergência tecnológica com a internet, comunicação sem fios etc. Há um consenso de que interatividade será a palavra-chave para entender esse novo cenário de conteúdo digital. E interatividade é a “praia” da indústria de jogos.

No que se refere ao desenvolvimento de jogos para TV Digital o panorama é bastante incipiente, não trazendo dados significativos tanto a nível local como internacional. Sinalizando a necessidade de fomentar este tipo de investimento, considerando que a TV Digital já é uma realidade nos Estados Unidos, Japão e Europa desde final da década de 90 e o Brasil iniciou sua implantação em 2006 com o Decreto Presidencial 5820 (BRASIL, 2008).

Desta maneira, este trabalho busca apresentar a metodologia de desenvolvimento de jogos para arquitetura da TV Digital e mostrar alguns exemplos desenvolvidos pelo Grupo de Tecnologia da Informação – Linha de Pesquisa sobre TV Digital do Centro Federal de Educação Tecnológica da Bahia (CEFET-BA), para demonstrar as vantagens oriundas do uso dessa tecnologia.

Assim, a seção 2 apresenta uma fundamentação teórica sobre a TV Digital e suas características. A seção 3 descreve a metodologia de desenvolvimento, mostrando as linguagens de programação que são utilizadas para desenvolvimento de aplicações para TV digital e o emulador GINGA-NCL. Na seção 4, alguns jogos

desenvolvidos nessa arquitetura são mostrados e finalmente, na seção 5, são exibidas as considerações finais do trabalho.

## **2. FUNDAMENTAÇÃO TEÓRICA**

Segundo Moreno (2006), o sistema de TV digital, de forma resumida, é um conjunto de especificações que determinam as técnicas de codificação digital para transmitir o conteúdo de áudio, vídeo e dados, das emissoras (ou provedores de conteúdo) aos terminais de acesso dos telespectadores. Assim, a televisão digital é um conjunto de novas tecnologias de geração e transmissão de conteúdo que promete revolucionar a maneira como as informações chegam até as casas das pessoas e como estas interagem com elas. Essa tecnologia apresenta as seguintes características: a transmissão sem interferência, melhor qualidade de imagem e som, maior variedade de canais, inclusive aproveitando melhor o mesmo espectro de transmissão, possibilidades de usar recursos interativos e a compatibilidade com computadores e a internet, dando suporte inclusive ao desenvolvimento de aplicações interativas.

Na proposta da TV digital, o usuário terá a possibilidade de interagir com multicâmeras, extras vinculados ao programa, portal de informação, novos formatos de publicidade e Guia Eletrônico de Programação - GEP. O espectador muda de lugar na medida em que poderá atuar individualmente, participando de cursos, realizando compras, montando sua grade de programação, enviando e recebendo mensagens, interagindo com jogos on-line, participando de programas com respostas personalizadas e tendo acesso ao TV-gov que consiste na solicitação de serviços como: declarações, prontuários, agendamento de serviços, publicidade dirigida com respostas (CPqD, 2008).

Para isso, foram desenvolvidos três padrões mundiais de Televisão Digital dominantes até o momento: o ATSC (Advanced Television Systems Committee) nos EUA, o DVB (Digital Vídeo Broadcasting) na Europa e o ISDB (Integrated Services Digital Broadcasting) no Japão. O padrão ISDB é o mais flexível de todos por responder melhor a necessidade de mobilidade e portabilidade, além de ter uma excelente qualidade na transmissão do conteúdo, inclusive esse foi o padrão escolhido pelo Brasil em 2006, após os testes técnicos comparativos conduzidos por um grupo de trabalho da Sociedade Brasileira de Engenharia de Televisão (SET) e da Associação Brasileira das Emissoras de Rádio e Televisão (ABERT), com o aval da Fundação Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPqD) (MENDES E FASOLO, 2002).

A partir dessa definição vêm sendo desenvolvidas pesquisas pela Universidade Federal da Paraíba (UFPb) e Pontifícia Universidade Católica do rio de Janeiro (PUC-Rio) para possibilitar a incorporação de um grupo de modificações que permitirão trabalhar com a realidade brasileira. O principal fator para a escolha do sistema ISBD foi a versatilidade, pois além de enviar os sinais da televisão digital, tal sistema pode ser empregado em diversas atividades, como: transmissão de dados, comunicação com sistemas computacionais e telefonia celular, acesso aos sites dos programas de televisão, serviços de atualização do receptor por download, sistema multimídia para fins educacionais.

Para atender as essas novas possibilidades disponíveis no sistema de TV Digital e considerando que boa parte dos telespectadores ainda não pode ter acesso ao aparelho de TV totalmente digital, os usuários terão que conectar a TV analógica (comum) a um equipamento denominado Set-top-Box, que consiste no recebimento e decodificação dos novos sinais totalmente digitais transmitidos pelas emissoras para o sinal analógico entendido pelos aparelhos existentes nas habitações da maior parte da população brasileira. Outra característica deste dispositivo é a existência de um canal de retorno que pode ser ligado à rede telefônica ou de transmissão de dados em alta velocidade, permitindo que o telespectador forneça algum tipo de informação ao programa assistido, originando níveis diferenciados de interatividade.

Os módulos principais de hardware do Set-top-Box são: sintonizador, demodulador, acesso condicional, demultiplexador, decodificador de áudio, processamento de dados(processador e memórias), interfaces físicas (modem, porta paralela, USB, FireWire, Ethernet, Serial RS232, Controle Remoto) (PICCOLO, 2008). O Set-top-Box típico contém um ou mais microprocessadores para executar o sistema operacional, memória Flash e SDRAM, variando entre 4 e 256 Mbytes, para poder armazenar temporariamente alguns dados, um decodificador de vídeos em formato MPEG e processadores específicos para trabalhar com os diversos canais de áudio.

Além da parte física da arquitetura da TV digital, existe uma camada de abstração, denominada middleware, que esconde das aplicações a complexidade dos mecanismos definidos pelos padrões, protocolos de

comunicação e até mesmo sistema operacional dos terminais de acesso. Simplificadamente, as implementações de middleware devem oferecer as bibliotecas necessárias às aplicações através de uma API (Application Program Interface) bem definida.

Atualmente existem quatro middlewares utilizados mundialmente, são eles: MHP, MHEG-5, ACAP e ARIB. A PUC-Rio e a UFPb estão desenvolvendo o middleware GINGA, para ser o padrão brasileiro. Desde dezembro de 2007, foi disponibilizado no site <http://www.ginga.org.br/>, o emulador para testar as aplicações desenvolvidas para esse middleware.

Segundo Soares (2007), o sistema brasileiro de TV digital procurou as alternativas tecnológicas mais recentes e entre elas estava a concepção de um middleware onde a convivência dos ambientes declarativo e procedural fosse a mais eficiente possível, em termos de custo e desempenho, além de dar suporte a aplicações declarativas mais eficientemente, tendo como foco o sincronismo de mídia na sua forma mais ampla, a interatividade do usuário como caso particular, a adaptabilidade do conteúdo a ser apresentado e o suporte a múltiplos dispositivos de interação e exibição.

Com isso, a proposta do middleware GINGA é incorporar o ambiente procedural e o ambiente declarativo baseado na linguagem NCL-Lua. Como o GINGA-J, que é o ambiente procedural que possibilita a utilização do JAVA TV ainda está na versão beta (OPENGINGA, 2008), os jogos que serão mostrados nesse artigo foram desenvolvidos para o ambiente GINGA-NCL com a linguagem NCL-Lua, utilizando o Emulador GINGA-NCL, mostrado na Figura 1, desenvolvido em JAVA.

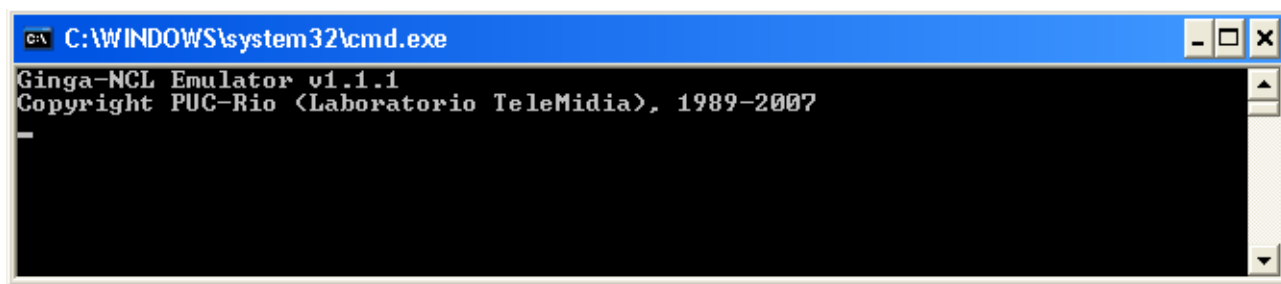


**Figura 1 – Tela principal do Emulador GINGA-NCL.**

O Emulador GINGA-NCL acompanha um controle remoto virtual, apresentado na Figura 2, e o prompt de execução, Figura 3, indicando a situação do programa e possíveis erros. O controle remoto é a principal interface para a interação do usuário com o jogo. É possível a inclusão de outras interfaces nas portas USB do STB.



**Figura 2 – Controle Remoto Virtual do Emulador GINGA-NCL.**



**Figura 3 – Prompt de Execução do Emulador GINGA-NCL**

### **3. METODOLOGIA DE DESENVOLVIMENTO DE JOGOS PARA TV DIGITAL**

De acordo com Schuytema (2008) o ciclo de desenvolvimento de um game pode ser dividido em três etapas. Na primeira, nomeada de pré-produção, toda a equipe de desenvolvimento cria um conceito para um game. Esse é um momento de discussões, brainstorms e avaliações dos games concorrentes. Nesse período, os designers também escrevem diferentes documentos de design que serão submetidos à aprovação posteriormente. Documento de Design (DD) é uma “planta baixa” de um game e seu objetivo é ilustrar como se deve jogá-lo, apresentando uma descrição detalhada de todos os aspectos.

O DD é um artefato utilizado na área de desenvolvimento de jogos como um documento que caracteriza o game a ser produzido e que define, entre outras coisas, a história do jogo, seu público alvo, seu gameplay e fases. Ao acompanhar a história, o jogador manterá seu interesse e terá uma sensação mais intensa da finalidade do jogo. A história de um jogo não é tão relevante em um jogo de raciocínio lógico.

A produção do game é a etapa seguinte. Nesse período, os artistas conceituais criam modelos de personagens e níveis; os programadores escrevem e revisam o código-fonte. Os designers fazem o roteiro do gameplay, avaliando-o em termos de diversão e trabalhando de perto com os setores de arte conceitual e programação para garantir que a funcionalidade do game esteja coerente com os documentos de design. Isso é fundamental, porque a equipe de testes deverá usar esses documentos como padrão durante o processo de testes.

A terceira etapa é a de pós-produção, que começa logo em seguida ao lançamento do game. Inclui-se aí a disponibilidade de correções do jogo, chamados de patches, ou avaliações da receptividade do game para futuros lançamentos.

Já o processo de desenvolvimento de software para TV Digital tem algumas peculiaridades em relação à outras arquiteturas como PCs. Por isso, a seguir, são enumeradas as etapas da metodologia definidas pelo Grupo de Tecnologia da Informação na Linha de Pesquisa sobre TV Digital do Centro Federal de Educação Tecnológica da Bahia (CEFET-BA) em conjunto com o Núcleo de Modelagem Computacional do SENAI/CIMATEC para o projeto de desenvolvimento de jogos para TV Digital.

1 - Análise das funções disponibilizadas pelo middleware, GINGA, objetivando explorar os recursos de imagem, som, armazenamento e dispositivos de entrada e saída.

2 - Especificações dos jogos a serem produzidos - o desenvolvimento do projeto se iniciará com uma etapa de brainstorm, onde serão definidas as especificações gerais dos jogos. Essas definições constituirão o Design Document (D.D.).

3- Definição das ferramentas a serem utilizadas - estudo, avaliação e seleção das ferramentas que serão utilizadas.

Após a definição do jogo, os designers especificarão os objetos gráficos a serem utilizados no jogo e utilizarão a ferramenta 3DMax para gerar os frames de cada objeto. Para programação das rotinas do jogo, a linguagem NCL-LUA foi escolhida.

4- Implementação - será conduzida em torno dos GDs (Grupos de Desenvolvimento), e terá início após a definição da identidade visual dos jogos. Na sequência, será elaborada a interface principal dos jogos, pelo grupo de artes, a programação das funcionalidades especificadas no D.D., pelo grupo de programação, e a definição do roteiro das fases dos jogos, pelo grupo de game design. Essas atividades serão executadas em

paralelo, e a partir daí se iniciará um processo cíclico e incremental, onde cada ciclo implementará uma fase ou missão do jogo.

#### 4.1- Equipe de Artes

O roteiro dos jogos subsidiará a etapa de modelagem dos objetos, por parte da equipe de artes. Essa etapa será constituída das seguintes sub-etapas: decupagem (divisão de uma cena em planos e a previsão de como estes planos vão se ligar uns aos outros através de cortes), arte conceitual (etapa que constrói o conceito ou idéia do jogo), elaboração do storyboard (roteiro em quadrinhos ou a divisão de ações em quadros) e modelagem.

#### 4.2- Equipe de som

Assim como na equipe de artes, o trabalho do grupo de som será subsidiado pelo roteiro da fase em implementação. As etapas de um ciclo de produção do grupo de som estão indicadas a seguir: decupagem, busca/catalogação de músicas e efeitos sonoros, edição/composição/ajuste, composição e edição de dublagem:

#### 4.3 - Equipe de programação

Assim como as demais equipes, o grupo de programação será guiado pelo roteiro da fase em implementação. As atividades deste grupo são: decupagem, planejamento do modelo solução, implementação, level design onde será feita a integração dos produtos gerados na fase anterior, posicionando os personagens e objetos, as fontes de som, adicionando toda a lógica de programação necessária ao funcionamento correto desses objetos.

A etapa de testes e ajustes encerra o ciclo de desenvolvimento, e será auxiliada pelo uso de um software de código aberto de rastreamento e correção de erros, o Mantis. Se for um jogo com várias fases, inicia-se então um novo ciclo, que construirá uma nova missão do jogo.

O primeiro ciclo de desenvolvimento será utilizado para concepção da versão de demonstração dos games. Essa versão será utilizada para testes de usabilidade do game, utilizando-se para tal, jogadores externos à equipe do projeto. As sugestões e críticas desses jogadores registradas em um instrumento de avaliação serão analisadas, e as mais importantes serão incluídas como correções para o ciclo posterior.

### 5- Acompanhamento

Para auxílio no acompanhamento das atividades e na gestão das equipes de desenvolvimento é utilizado o Microsoft Project, software da área de gestão de projetos de Tecnologia da Informação. Este software possibilita, entre outras coisas, a distribuição de atividades entre integrantes de uma equipe, a atualização da percentagem de conclusão dessas atividades e a geração de relatórios de andamento do projeto.

### 6- Teste-Piloto (Beta) com usuários de games e ajustes finais.

Este teste será realizado com um grupo de jogadores, analisando o nível de motivação dos sujeitos envolvidos, bem como a contribuição do jogo para o entretenimento. Os problemas constatados serão corrigidos, encerrando-se assim o processo de desenvolvimento do jogo.

### 7- Finalização

A etapa de finalização do projeto inicia-se com o fim da etapa de ajustes finais. Na finalização será feita uma avaliação geral do jogo por parte de toda a equipe do projeto. Serão discutidos e avaliados tanto o jogo produzido, quanto o processo de desenvolvimento adotado. Cada GD apresentará o seu relatório geral de desenvolvimento, onde constarão informações como dificuldades encontradas, soluções para problemas enfrentados e metodologias adotadas. Os frutos da reunião de finalização irão compor um documento de avaliação do projeto, que irá reunir o aprendizado da equipe e sugestões de possíveis melhorias no game desenvolvido. A segunda e a terceira etapa envolvem o processo de distribuição e acompanhamento do jogo.

Como a maioria das etapas é comum a vários processos de desenvolvimento de games, detalharemos apenas a definição das ferramentas e a implementação na subseção seguinte.

### 3.1. Definição das Ferramentas e Implementação

Como descrito anteriormente, o processo de desenvolvimento de software para TV Digital tem algumas diferenças em relação ao desenvolvimento para PCs. Nessa arquitetura, assim como para celulares, são utilizados os emuladores do middleware no qual o aplicativo será executado para que não seja necessária a instalação no Set-top-Box toda vez que forem realizados procedimentos de testes. Os jogos desenvolvidos e que serão apresentados nesse artigo foram executados no Emulador GINGA-NCL Player, que pode ser encontrado no Portal do Software Público Brasileiro<sup>2</sup> para Windows, Linux e MAC OS. Para tanto, tem-se como pré-requisito a instalação do JRE – Java Runtime Environment – versão 6 ou superior.

O Ginga-NCL foi criado pela PUC-Rio para oferecer uma infra-estrutura de apresentação para aplicações multimídia/hipermídia sob o paradigma declarativo, escritas em linguagem NCL. Diferente do HTML, ou XHTML, a linguagem NCL não mistura a definição do conteúdo de um documento com sua estruturação, oferecendo um controle não invasivo, tanto do leiaute do documento (apresentação espacial), quanto da sua apresentação temporal. Como tal, NCL não define nenhum objeto de mídia, mas apenas referencia esses objetos semanticamente juntos em uma apresentação multimídia.

Mas, o uso da linguagem NCL não é suficiente para o desenvolvimento de jogos, pois esses aplicativos exigem um controle do fluxo em tempo de execução. Para tanto, o grupo da PUC-Rio disponibilizou uma integração na forma de objeto com a linguagem LUA (NCL-LUA). Lua é uma linguagem de programação leve, mas poderosa, que foi projetada para estender aplicações, essa linguagem combina sintaxe simples para programação procedural com poderosas construções para descrição de dados, baseadas em tabelas associativas e semânticas extensíveis, permitindo, assim, certo controle ao designer de games sem a enorme curva de aprendizagem, normalmente associada à programação (SCHUYTEMA, 2008).

A linguagem Lua é tipada dinamicamente, é interpretada a partir de bytecodes para uma máquina virtual(engine) acoplada ao formatador NCL, e tem gerenciamento automático de memória com coleta de lixo incremental. Essas características fazem de Lua uma linguagem ideal para configuração, automação (scripting) e prototipagem rápida (SOARES, 2007).

A máquina Lua (código livre e aberto) é disponibilizada como uma pequena biblioteca de funções C, escritas em ANSI C, que compila sem modificações em todas as plataformas conhecidas. Tal linguagem tem simplicidade, eficiência, portabilidade e baixo impacto de inclusão em aplicações, sendo hoje uma das linguagens mais utilizadas no mundo na área de entretenimento (LucasArts, BioWare, Microsoft, Relic Entertainment, Absolute Studios, Monkeystone Games, etc.). Naturalmente, NCL-Lua se tornou o casamento ideal para o ambiente declarativo do sistema brasileiro de TV digital (SOARES, 2007). Afinal, tanto o NCL quanto a LUA foram desenvolvidas pelo mesmo laboratório.

### 3.2. Limitações para o Desenvolvimento de Jogos no SBTVD

Com linguagens procedurais e a manipulação de imagens é possível o desenvolvimento de jogos no SBTVD. Porém existem quatro limitações que estão listadas a seguir:

**- Os Shapes (objetos) gerados pelo 3D Studio Max não podem ser exportados diretamente para a linguagem Lua** - Diferente do que acontece nos Engines de jogos como o TORQUE<sup>3</sup> (GARAGEGAMES, 2008) que a exportação é feita diretamente. Para resolver isso, o Shape é convertido em um Sprite Sheet<sup>4</sup>, que pode ser referenciado no código fonte escrito em linguagem LUA. Outra opção é gerar um mapa de bits para ser tratado diretamente pelo pacote da linguagem LUA.

**- O tamanho das imagens e das figuras utilizáveis no emulador GINGA-NCL é pequeno** – O emulador GINGA-NCL Player só possibilita o uso de figuras e imagens com até 350 Kbytes com resolução de 800x600. De forma que, se essa quantidade de bytes for ultrapassada, o emulador informa a falta de memória.

---

<sup>2</sup> Disponível em [http://www.softwarepublico.gov.br/ver-comunidade?community\\_id=1101545](http://www.softwarepublico.gov.br/ver-comunidade?community_id=1101545). Acesso em 18 de Julho de 2008.

<sup>3</sup> Disponível em: <http://www.garagegames.com/docs/tge/general/>. Acesso em 18 de Julho de 2008.

<sup>4</sup> Sprite Sheet é uma imagem com vários quadros de posições diferentes de uma animação do mesmo objeto.



- **O processamento gráfico ocorre na CPU** – Os STB comuns não possuem Unidade de Processamento Gráfico (GPU) e assim todo o processamento gráfico ocorre na CPU. Isso inviabiliza as “renderizações” de objetos 3D para uma representação 2D.

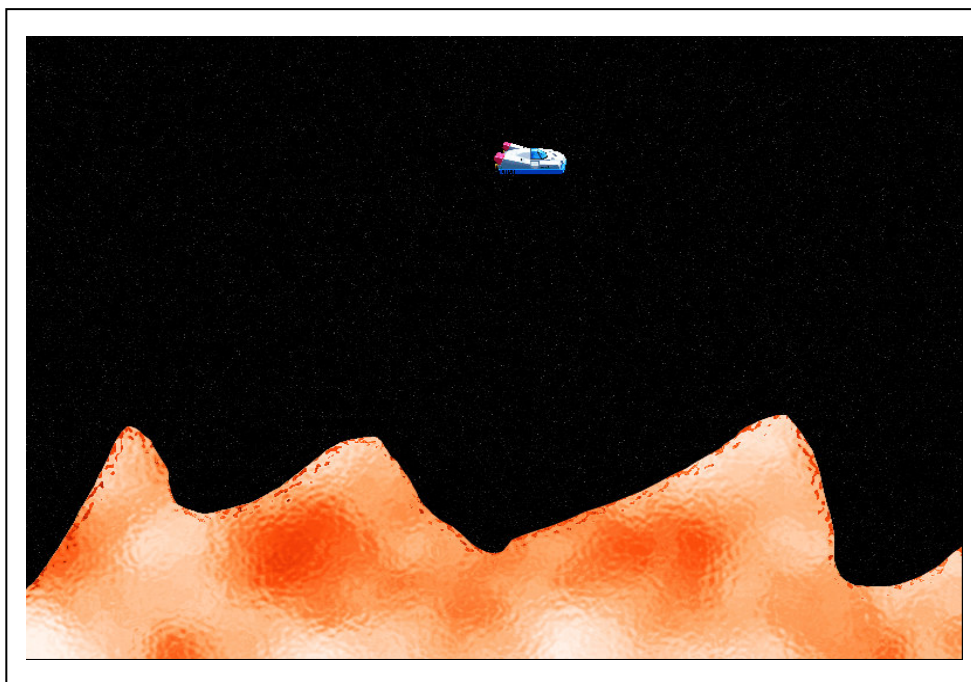
- **O GINGA não permite a execução de Threads** – Quando um jogo é dividido em Threads, o jogo não precisa esperar a execução de um módulo para começar o outro. Cada módulo pode ser uma Thread, que pode ser executada “quase simultaneamente”. Se o STB tiver mais de um processador, é possível essa execução, desde que existam mudanças também no middleware GINGA.

#### 4. JOGOS DESENVOLVIDOS

Após o estudo teórico sobre o desenvolvimento de aplicações para TV Digital, iniciou-se a elaboração de dois jogos, mostrados nas Figuras 4 e 6, que pudessem ser utilizados como uma avaliação inicial do potencial gráfico da arquitetura GINGA-NCL.

O primeiro demo desenvolvido, mostrado na Figura 4, consiste em uma aventura com uma narrativa bem simples, aos moldes dos antigos jogos para console como Gradius e R-type. Só que nessa versão com uma melhoria na qualidade gráfica. No jogo, a nave é guiada pelo jogador, que segue por diferentes caminhos defrontando-se com inúmeras barreiras e inimigos diversos que tentarão impedir sua passagem e destruí-la. Para alcançar o objetivo de cada missão do game, o jogador deverá transpor todas essas dificuldades.

Para adicionar um elemento de jogabilidade haverá um sistema de armas elaborado para que o jogador possa acompanhar o gradativo aumento de dificuldade. Esse sistema de “*power-ups*” procura ser uma versão melhorada do sistema visto em R-type ou do sistema visto em Gradius.



**Figura 4 – Tela do demo de aventura com uma nave espacial**

Na Figura 5, é apresentada uma parte do código do demo da NAVE no LuaEdit versão 5.0, que é um editor para linguagem LUA desenvolvido por Jean-François Goulet. O desenvolvimento do demo consistiu basicamente em definir os objetos que guardam as informações da nave, do cenário do fundo, e da montanha. Os objetos guardam as informações posição x, posição y e imagem. A função de redesenho (*redraw*) é chamada a cada vez que se pressiona uma tecla, atualizando a posição x e y do personagem e redesenhando o cenário, usando para tal a função *compose*.



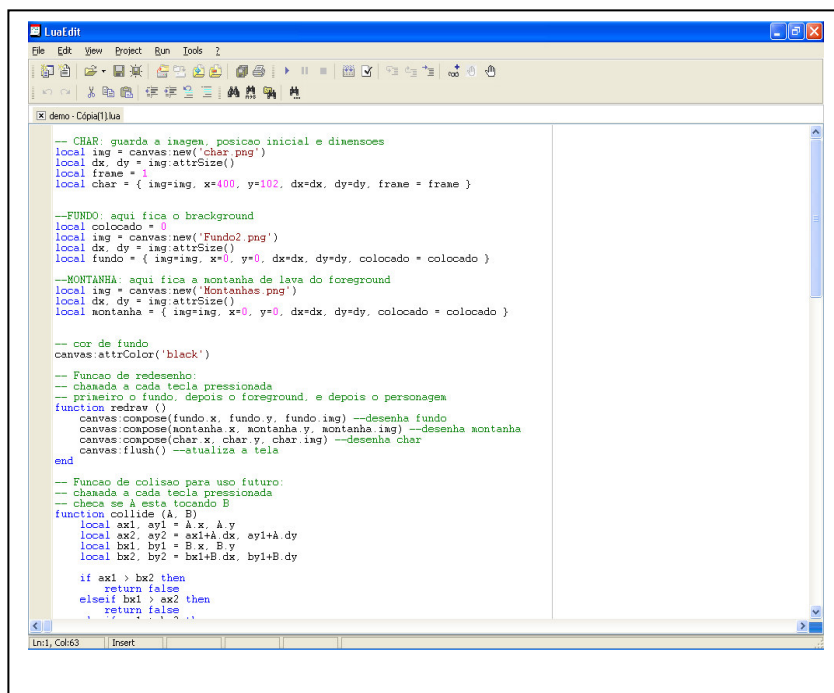


Figura 5 – Código Fonte Lua para o Jogo da NAVE.

O segundo jogo, mostrado na Figura 6, é um Sudoku, que é um tipo de quebra-cabeça baseado na colocação lógica de números. O objetivo do jogo é a colocação de números de 1 a 9 em cada uma das células vazias numa grade de 9×9, constituída por 3×3 subgrades chamadas regiões. O quebra-cabeça contém algumas pistas iniciais. Cada coluna, linha e região só pode ter um número de cada um dos 1 a 9. Para resolução do problema é necessário raciocínio lógico e tempo.

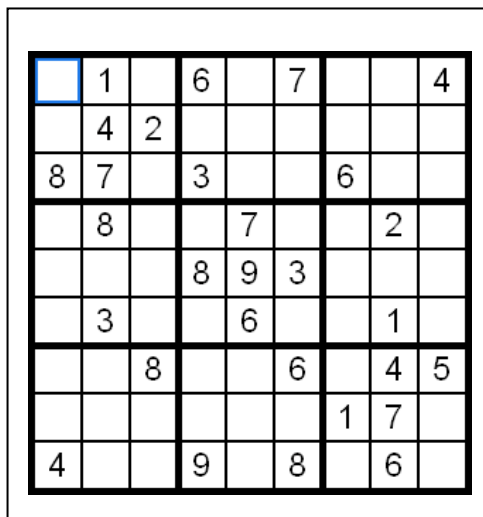


Figura 6 – Tela do jogo SUDOKU

## 5. CONSIDERAÇÕES FINAIS

Esse trabalho apresentou o processo de desenvolvimento de jogos para o Sistema Brasileiro de Televisão Digital (SBTVD), indo desde a investigação das possibilidades tecnológicas dessa arquitetura, da codificação do jogo na linguagem LUA até a execução no Emulador GINGA-NCL. Mostrou-se que mesmo com poucas ferramentas disponíveis e com um potencial gráfico limitado, pois não há uma unidade gráfica de processamento e não é possível a criação de Threads, para execução das instruções em paralelo, é possível desenvolver jogos no SBTVD.

Apesar da realização de testes de módulo, integração, aceitação e de sistemas em conjunto com inspeção dos jogos no próprio SET-TOP-BOX ser um dos objetivos desse trabalho, optou-se por não arriscar a compra desse equipamento sem o middleware GINGA já instalado e consolidado. Portanto, foram feitos apenas os testes de forma limitada no próprio emulador.

Além da inspeção e dos testes de módulos, de sistemas, de usabilidade e de desempenho realizadas diretamente no STB, um trabalho futuro será o desenvolvimento de jogos multiplayer para avaliar o módulo de comunicação entre STB e arquitetura de redes como um todo.

## REFERÊNCIAS

ABRAGAMES. **Plano Diretor de Promoção da Indústria de Desenvolvimento de Jogos Eletrônicos no Brasil** – Diretrizes Básicas. 2004. Disponível em [http://www.abragames.org/docs/pd\\_diretrizesbasicas.pdf](http://www.abragames.org/docs/pd_diretrizesbasicas.pdf) Acesso em: 16 de Julho de 2008.

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. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/\\_Ato2004-2006/2006/Decreto/D5820.htm](http://www.planalto.gov.br/ccivil_03/_Ato2004-2006/2006/Decreto/D5820.htm)>. Acesso em: 10 de julho de 2008.

CPqD. Centro de Pesquisa e Desenvolvimento em Telecomunicações - Site SBTVD. **Modelo de Referência Sistema Brasileiro de Televisão Digital Terrestre**. Disponível em [http://sbtvd.cpqd.com.br/cmp\\_tvdigital/divulgacao/anexos/76\\_146\\_Modelo\\_Ref\\_PD301236A0002A\\_RT\\_08\\_A.pdf](http://sbtvd.cpqd.com.br/cmp_tvdigital/divulgacao/anexos/76_146_Modelo_Ref_PD301236A0002A_RT_08_A.pdf) Acesso em 18 de Julho de 2008.

FERNANDES, Jorge; LEMOS, Guido; SILVEIRA, Gledson. **Introdução à Televisão Digital Interativa: Arquitetura, Protocolos, Padrões e Práticas**. In: JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA DO CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, JAI-SBC, em Salvador – BA – Agosto de 2004. Anais do JAI—SBC, 2004.

MENDES, Luciano L; FASOLO, Sandro A. **Introdução a TV Digital**. Anais do Congresso da Semana Internacional das Telecomunicações. Santa Rita do Sapucaia. Brasil, 2002.

MORENO, Marcio Ferreira. Um Middleware Declarativo para Sistemas de TV Digital Interativa. Orientador: Luiz Fernando Gomes Soares – Rio de Janeiro: PUC, Departamento de Informática, 2006.

OPENGINGA. Projeto OpenGINGA. Disponível em <http://www.openginga.org/index.html>. Acesso em 18 de Julho de 2008.

PICCOLO, Lara S. Godoy. **Arquitetura do Set-top Box para TV Digital Interativa**. Disponível em [www.cin.ufpe.br/~gds/TAI/GDS\\_CEMR-APLIC-06.pdf](http://www.cin.ufpe.br/~gds/TAI/GDS_CEMR-APLIC-06.pdf). Acesso em 18 de Julho de 2008.

RHODES, Glen. **Desenvolvimento de Games com Macromedia Flash Professional 8**. Cengage Learning, 2008.

SOARES, Luiz Fernando Gomes. **Ambiente para Desenvolvimento de Aplicações Declarativas para a TV Digital Brasileira**. In: TV Digital: Qualidade e Interatividade. Instituto Euvaldo Lodi. Brasília, 2007.

SCHUYTEMA, Paul. **Design de games: Uma abordagem prática**. 1ª Ed. São Paulo. Cengage Learning, 2008.

## AGRADECIMENTOS

Agradecemos à FAPESB – Fundação de Amparo à Pesquisa do Estado da Bahia – pelas bolsas concedidas e pelo apoio financeiro.