

ForgeFlex uma Extensão do Forge16V: Incorporando uma Biblioteca de Componentes Gráficos para Jogos

Allan D. S. Lima¹ Eduardo W. M Almeida² Nancy L. Lino³ Victor C. Rodrigues⁴

Danielle R. D. Silva⁵ Patricia R. Tedesco⁵ Geber L. Ramalho⁵

Universidade Federal de Pernambuco, Centro de Informática, Brasil

Resumo

Para acompanhar o crescimento da indústria, os estúdios de jogos vêm focando suas atenções à automatização do processo de desenvolvimento. Dentro deste contexto, são desenvolvidos *frameworks* e motores (ou *engines*) que aceleram a formulação e projeto dos jogos, incluindo também componentes gráficos que podem ser personalizados conforme a identidade visual de cada jogo. Porém, boa parte desse ferramental de apoio ao desenvolvimento de jogos é de domínio privado ou comercial, o que em muitos casos, inviabiliza o uso dessas ferramentas dentro do ambiente acadêmico, principalmente, para auxiliar no processo ensino-aprendizagem no desenvolvimento de jogos. Dessa forma, o presente artigo tem como objetivo descrever a experiência no desenvolvimento de uma biblioteca de componentes gráficos de alto nível direcionado para o desenvolvimento de jogos. Esta biblioteca foi integrada ao Forge16V, um *framework* de desenvolvimento de jogos isométricos, de código aberto, gratuito, e que tem como foco principal dar suporte ao processo de ensino-aprendizagem de jogos no âmbito acadêmico.

Palavras-chave: biblioteca gráfica, componente gráfico, motor de jogo.

Contato dos autores:

{adsl, ewma, nll, vcr2, drds, pcart, glr}@cin.ufpe.br

1. Introdução

É de conhecimento comum que o projeto de interface de um jogo difere em alguns aspectos de aplicações de automatização como, por exemplo, processadores de texto, planilhas eletrônicas, e similares. Geralmente, este tipo de aplicação segue o padrão de interface e componentização gráfica dado pelo sistema operacional ao qual irá executar, havendo pouca diferença de estilo gráfico de um aplicativo para o outro. Nestes casos o objetivo é incorporar lições de usabilidade já aprendidas e padrões já internacionalizados pelos usuários para facilitar a vida dos mesmos. Em um jogo, esse caminho é um pouco diferenciado.

Cada jogo desenvolvido tenta fornecer uma identidade visual dada pelo tema e história que o jogo transporta.

Em outras palavras, a interface de um jogo deve incorporar todos os requisitos de usabilidade porém com o cuidado de não perder a identidade temática. Isto requer um maior esforço tanto para os desenvolvedores como para os projetistas gráficos ou artistas e não impede o uso de bibliotecas comuns de componentes gráficos disponibilizadas para o desenvolvimento de aplicações comuns.

Para sanar este tipo de problema, os estúdios de jogos têm construídos suas próprias bibliotecas gráficas visando facilitar e agilizar o processo de desenvolvimento principalmente no que tange ao desenvolvimento das interfaces dos seus diversos jogos, disponibilizando componentes gráficos que podem ser personalizados conforme o tema ou identidade do jogo. Porém, boa parte destas bibliotecas é de domínio privado ou comercial.

Existem poucas bibliotecas gráficas disponíveis gratuitamente que forneçam componentes gráficos de interface facilmente personalizados. O que é mais comumente encontrado são bibliotecas gráficas como a Allegro¹ ou SDL²(*Simple DirectMedia Layer*) que ainda trabalham em um nível baixo de abstração.

No processo de ensino-aprendizagem na área de desenvolvimento de jogos, a ausência de bibliotecas de componentes gráficos dificulta a prototipação rápida de jogos, principalmente quando trabalhando com linguagens de programação como C++³, em que a curva de aprendizagem não é tão suave.

Neste contexto, este artigo descreve a extensão de um *framework*, implementado em C++, de desenvolvimento de jogos isométricos denominado Forge16V⁴, para incorporar bibliotecas de componentes gráficos de alto nível como botões, painéis, tabelas, gráficos, campos de texto, menus e etc. Este *framework* é de código aberto e já vêm sendo amplamente utilizado no desenvolvimento de jogos pelo Centro de Informática – UFPE.

1 Biblioteca multi-plataforma(<http://alleg.sourceforge.net>)

2 Acrônimo de SimpleDirectMedia Layer (<http://www.libsdl.org>)

3 Linguagem de Programação (<http://www.cplusplus.com>)

4 Framework para construção de jogos brevemente no Sourceforge (<http://www.sourceforge.net>)

O presente artigo está dividido em seis (6) seções. A seção 2 descreve alguns trabalhos relacionados ao tema; a seção 3 descreve o *framework* Forge16V e as extensões necessárias para efetivá-lo como uma ferramenta eficiente no desenvolvimento de jogos; a seção 4 descreve a integração da biblioteca de componentes gráficos para o Forge16V; e por fim, a seção 6 discute as conclusões e aponta oportunidades de trabalhos futuros

2. Trabalhos Relacionados

Atualmente existem poucas bibliotecas de componentes gráficos disponíveis gratuitamente. Geralmente, o que se encontra disponível são as de uso geral, responsáveis pela manipulação do processo de renderização das imagens, como por exemplo, componentes para facilitar a utilização dos dispositivos gráficos da plataforma alvo de execução do jogo.

Podem-se citar como exemplo de bibliotecas gráficas de uso geral, as bibliotecas multimídia Allegro e SDL. Estas são bastante utilizadas no desenvolvimento de jogos.

Allegro disponibiliza componentes que fornecem funcionalidades como, por exemplo: manipulação de desenhos ou imagens vetoriais, manipulação de *sprites*, paletas de cores e textos. Além destas, também disponibiliza componentes para manipulação de sons e funções matemáticas e existem vários pacotes de extensão disponíveis gratuitamente na Internet, inclusive, de suporte a GUI. Porém esses pacotes são de difícil compreensão e uso por serem escritos em C, o que impossibilita a utilização de heranças e polimorfismo. Isso praticamente inviabiliza o seu uso em projetos mais complexos.

A SDL fornece praticamente todas as funcionalidades disponibilizadas pela Allegro tais como, renderização 2D através de Bitmaps, suporte a efeitos sonoros e música de fundo, e também a dispositivos de entrada e saída. No entanto, ela oferece extensões mínimas voltadas para o desenvolvimento de interfaces gráficas com componentes de alta ordem. As implementações de pacotes de GUI para SDL são melhores dos que os encontrados para Allegro. No entanto a sua utilização ainda é deficitária pois a maioria desses pacotes não possuem componentes mais complexos, como tabelas e gráficos.

3. O *framework* Forge16V

O principal objetivo do Forge16V foi suprir a necessidade de um motor de jogos tanto para os alunos do CIn-UFPE quanto para estudantes de outras universidades locais, oferecendo diversas funcionalidades e não apenas as direcionadas para a manipulação da parte gráfica do jogo.

O *engine* Forge16V possui os seguintes módulos principais: o gerenciador principal que constitui a base da implementação do motor, sendo responsável pela instanciação e destruição dos demais módulos de recursos consumidos pelo motor, bem com a temporização além de tratar os eventos oriundos do sistema operacional; o gerenciador gráfico responsável pelo gerenciamento do uso da memória de vídeo; o gerenciador de entrada que gerencia os dispositivos de entrada entre o usuário e a aplicação; o gerenciador de som que permite a leitura e reprodução de arquivos de áudio.

Durante o processo de desenvolvimento do Forge16V diversos requisitos foram elicitados para integrar ao motor, entretanto, por fatores como prazos curtos para implementação, nem todos estes puderam ser contemplados. Estas extensões necessárias são: Gerenciador de multiusuários, Gerenciador de modelagem física, Gerenciador de IA, Editor de cenários e uma Biblioteca de componentes gráficos.

Conforme foi visto, apesar da implementação original do Forge16V possuir diversos módulos que são fundamentais para a construção de jogos. Porém, ainda existem muitas outras extensões a serem feitas para incrementar a qualidade do motor.

4. A Biblioteca de Componentes Gráficos

Um dos requisitos indispensáveis para o desenvolvimento de jogos é a disponibilidade de componentes para representar graficamente o jogo de maneira mais rápida e eficiente, requisito este, não disponível no Forge16V.

Assim, foi projetada uma nova reestruturação do *framework* a fim de incluir tal requisito. Esta, por sua vez, foi baseada em alguns padrões de projetos tais como o padrão Abstract Factory e Composite, que permitem, dentre outras coisas, que a biblioteca possa facilmente acoplar *look and feels* e possibilitar uma estruturação da arquitetura que define uma hierarquia entre os elementos gráficos sem replicar código.

Dessa forma, foi possível desenvolver um rico conjunto de componentes, onde, ao criar suas aplicações, os desenvolvedores podem construir facilmente um ambiente interativo e personalizado, atribuindo às funcionalidades que melhor lhe convier.

Para a criação destes componentes foram identificadas as principais necessidades em componentes gráficos na criação de interfaces de jogos, principalmente, considerando o estudo de caso, o protótipo de jogo Virtual Team. E assim, foi priorizado o desenvolvimento da lista de componentes descritos a seguir.

Button	
Permite 4 tipos de imagens como background: normal, selecionado, clicado, clicado e selecionado	Permite 5 tipos de fontes: normal, selecionado, desabilitado clicado, clicado e selecionado

Label	
Permite auto-ajuste do texto e da imagem	Suporta tabulação e quebra de linha automática

TextField	
Suporte à mascara do texto <ul style="list-style-type: none"> tamanho máximo de letras formato do texto caracteres especiais (#, @ e ?) 	

Progress Bar	
Permite incremento nas quatro direções	Permite iterar cores e figuras

Panel	
Funciona como um container básico e permite manipulação de componentes, como: <ul style="list-style-type: none"> adicionar, podendo também informar a ordem encontrar através determinada posição ou índice obter número total de componentes remover alterar ordem 	

Table	
Permite borda customizável	Características (fonte, linhas, colunas, etc.) da tabela são adaptáveis ao modelo instanciado

Chart		
Suporte a multigráficos	Permite alteração de escalas (horizontal e vertical) por gráfico	Características (cor, valores, etc) de cada gráfico são adaptáveis ao modelo instanciado

Desta forma, todo *Widget* tem associado a ele um nome, um texto, um ícone, uma cor e uma figura de fundo. Ainda é possível determinar sua visibilidade, variando inclusive o nível de transparência. Também é permitido configurar tipos, tamanhos e cores de bordas e fontes, bem como o alinhamento e orientação dos textos e margens, além da descrição associada ao texto do *tooltip*.

5. Estudo de Caso – Virtual Team

O Virtual Team é um protótipo de um jogo de negócio baseado em simulação, que visa capacitar gerentes de projetos menos experientes nos processos focados em gestão de pessoas, via a utilização de atores sintéticos. A idéia desta ferramenta é permitir ao gerente uma maior experimentação dos processos organizacionais, metodológicos, culturais e pessoais inerentes a uma equipe de desenvolvimento de software, que é uma atividade crítica hoje no Brasil.

O protótipo tem como motor de jogo o Forge Flex, e todo o projeto de interface do Virtual Team está implementado utilizando os recursos da biblioteca de componentes gráficos disponibilizada.

Os componentes foram facilmente instanciados para a identidade visual projetada para o jogo, devido à flexibilidade proporcionada pela biblioteca em relação à customização da aparência dos componentes. Algumas das interfaces do protótipo são mostradas abaixo (Figura 1 e Figura 2).

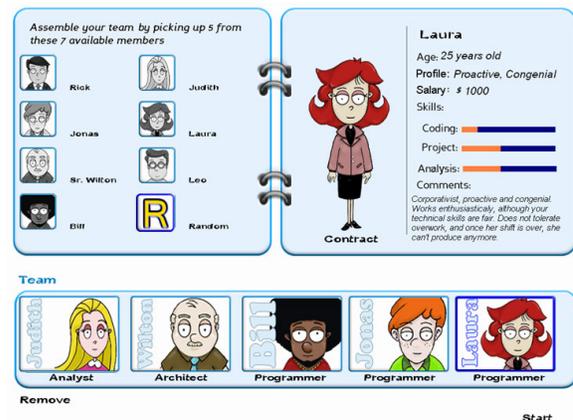


Figura 1. Tela de seleção de equipe.

Além destes, também foram desenvolvidos *togglebutton*, *checkboxitem*, *menu*, *buttongroup*, *listbox*, *tabbedpane* e *tooltip*. Embora cada componente desenvolvido tenha suas particularidades, todos compartilham propriedades básicas que determinam seu funcionamento, e podem ser representados pela menor unidade funcional de um componente de interface, conhecida como *Widget*.



Figura 2. Tela de atribuição de tarefa.

6. Conclusões

Com o crescimento da indústria de jogos, ferramentas gráficas vêm sendo construídas no sentido de facilitar e agilizar o desenvolvimento de interfaces, promovendo uma abstração à equipe de produção.

Ao fazer uso de motores de jogos, um dos módulos de maior complexidade e que deve prover grandes benefícios de personalização, respeitando a identidade de cada jogo, é o da biblioteca de componentes gráficos.

Porém, há uma escassez desse tipo de biblioteca disponibilizada gratuitamente e que possam servir de apoio ao desenvolvimento de protótipos e pequenos jogos dentro do meio acadêmico. Logo, visando a atender esta carência e, especificamente, a demanda do projeto SmartSim no desenvolvimento do protótipo de jogo Virtual Team, foi estendido o *framework* Forge16V de jogos isométricos para incorporar uma biblioteca deste tipo. Esta biblioteca de componentes gráficos deveria ser extensível, flexível e fácil de manter.

De acordo com tais metas, o principal objetivo foi alcançado, agregando inclusive as características de extensibilidade, facilidade de uso e estabilidade. Com isso o Forge16V foi estendido passando, agora, a chamar-se de ForgeFlex.

Vale salientar, porém, que ainda continuam sendo necessários alguns ajustes ao *framework*, principalmente, em relação ao desempenho. Estes ajustes estão sendo trabalhados nas próximas extensões do ForgeFlex.

Na tentativa de realizar mais um esforço de consolidação do *framework* como uma ferramenta de apoio ao desenvolvimento de jogos, o ForgeFlex está sendo preparado para ser adicionado em um repositório de *download* de código aberto tipo SourceForge.

Agradecimentos

Gostaríamos de agradecer as pessoas que fazem o projeto SmartSim por terem nos dado a oportunidade de desenvolver esta biblioteca, cujo processo nos engrandeceu tecnicamente, especialmente a Danielle Rousy pela paciência e dedicação.

Referências

ALLEGRO, A GAME PROGRAMMING LIBRARY. Disponível on-line na Url: [Http://Alleg.Sourceforge.Net/Readme.Html](http://Alleg.Sourceforge.Net/Readme.Html).

GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J.; *Design patterns: elements of reusable object-oriented software*, 1995.

KYRA, AN SPRITE ENGINE. Disponível on-line na Url: <http://www.grinninglizard.com/kyra/>.

LATHAM, ALISON, *The Oxford Companion to Music*, OUP, 2002.

MADEIRA, C. A. G. AND RAMALHO G. L., 2001. *Forge V8: Um framework para o desenvolvimento de jogos de computador e aplicações multimídia*. Dissertação de Mestrado em Ciência da Computação, Centro de Informática, Universidade Federal de Pernambuco, Recife.

RUSSELL, S. AND NORVIG, N, 2002. *Artificial Intelligence: A Modern Approach*.

OGRE. Disponível on-line na Url: <http://www.ogre3d.org/>.

ROCHA, E. J. T. S., RAMALHO G. L. AND SANTOS A. L. M. *Forge 16V: Um Framework para Desenvolvimento de Jogos Isométricos*. Dissertação de Mestrado em Ciência da Computação. Centro de Informática, Universidade Federal de Pernambuco, Dezembro de 2003, Recife – PE.

SDL. Disponível on-line na Url: <http://www.libsdl.org/index.php>.

SMARTSIM. Disponível on-line na Url: <http://www.cin.ufpe.br/~smartsim>.