

EUPAT for WoW: uma ferramenta para criação de extensões em Lua

Marcelo B. Barbosa, Jair C. Leite

Universidade Federal do Rio Grande do Norte, DIMAp, Brasil

Resumo

Este trabalho visa introduzir o aplicativo *EUPAT for WoW* (do inglês, *End-User Programming Assistance Tool for World of Warcraft*), uma ferramenta desenvolvida para facilitar e auxiliar usuários não-programadores na construção de extensões (*add-ons*) para World of Warcraft.

Palavras-chave: programação por usuário-final, interação humano-computador, *add-ons*, World of Warcraft, EUPAT

1. Introdução

Nos dias de hoje, uma das maneiras encontradas para aprimorar a experiência do usuário em jogos é permitir que ele personalize sua própria vivência dentro deles. Muitos jogos do mercado atual – em sua grande maioria MMOs (acrônimo para jogos *Massive Multiplayer Online*) – já permitem que os usuários desenvolvam extensões para os mesmos. Estas extensões (ou *add-ons*) podem ser utilizadas para aprimorar a sua capacidade de jogo (conhecido como jogabilidade), através da personalização da interface de usuário, modificando a sua aparência e comportamento de forma a reinventar a experiência do usuário dentro do jogo.

Pode-se observar esta idéia sendo aplicada em grandes títulos do mercado mundial de jogos, como *Second Life* e *World of Warcraft*, um dos jogos mais populares do mundo.

Em *World of Warcraft* (WoW), os usuários não apenas criam extensões, eles reinventam as interações do jogo, personalizando completamente sua interface (tanto graficamente quanto funcionalmente) bem como realizando melhorias cruciais de usabilidade.

No entanto, a criação de *add-ons* para *World of Warcraft* é uma tarefa que requer um nível elevado de conhecimento em programação. Isto porque, no WoW, os *add-ons* são construídos como pequenos programas (*scripts*) na linguagem Lua. O problema é que a maioria dos jogadores não possui conhecimento básico de programação e criar tais extensões torna-se uma tarefa um tanto quanto complicada.

O *EUPAT for WoW* (do inglês, *End-user Programming Assistance Tool for World of Warcraft*) é uma ferramenta interativa que facilita e auxilia a criação de *add-ons* para World of Warcraft.

O objetivo do EUPAT é auxiliar jogadores de WoW que não são programadores à criarem os seus próprios *add-ons*, introduzindo alguns conceitos básicos de lógica de programação, tais como o conceito de variáveis, expressões condicionais e estruturas de controle.

O EUPAT oferece também uma ferramenta útil para aqueles que já possuem conhecimento em programação, permitindo que usuários avançados tenham acesso a estruturas mais complexas – como Tabelas e Funções – e trabalhem sobre a própria API do jogo.

2. Programação por Usuário Final

Muitas aplicações hoje em dia – não só na área de jogos – oferecem mecanismos de customização ou programação pelo usuário final, transferindo para ele parte do problema de design [Barbosa 1999].

Um bom exemplo de customização são as Barras de Ferramentas dos navegadores para Internet. Através do mecanismo de personalização da barra o usuário pode acrescentar ícones, atalhos para funções, espaços e separadores, ou mesmo criar uma nova barra de ferramenta.

Existem vários exemplos de Programação por Usuário-Final. Um deles é o Chickenfoot, uma extensão do Mozilla Firefox que permite ao usuário escrever scripts que manipulam páginas e automatizam o processo de navegação [Chickenfoot 2007]. Outro exemplo é a gravação de macros no Microsoft Office, que permite aos usuários ativarem um gravador de instruções que registra e armazena cada passo de interação dentro de uma sequência de comandos [Frye 2003].

A programação por usuário final é pouco realizada devido às dificuldades que a maioria dos usuários tem com as técnicas básicas de programação [Bonnie 1993]. Existem várias estratégias para ensinar a prática de programação a usuários comuns sem a necessidade de cursos especializados [Antunes et al. 2006]. Exemplos de ambientes de programação que permitem isso são o Scratch [Scratch 2009] e o Alice [Alice 2009].

O Scratch e o Alice são ambientes de desenvolvimento para as linguagens de programação educacionais homônimas, desenvolvidas com o propósito de auxiliar no aprendizado de conceitos matemáticos, lógicos e de computação, idealmente para pessoas que ainda estão começando a programar.

As linguagens do Scratch e Alice são mais acessíveis que a maioria das outras linguagens por utilizarem artifícios gráficos que são mais atrativos por buscar utilizar uma representação visual para os conceitos abstratos de programação. O Scratch, por exemplo, utiliza uma interface gráfica que permite que programas sejam montados como blocos - lembrando o brinquedo LEGO. Cada bloco da linguagem representa um comando em separado. Esses comandos podem ser então agrupados livremente - caso se encaixem - ou modificados através de menus.

Apesar de possuírem conceitos parecidos, o contexto desses ambientes difere quanto ao grau de aplicação: o Alice oferece uma interface de usuário com manipulação direta (*drag and drop*) voltado para a criação de animações usando modelos 3D. O Scratch, por sua vez está focado para a criação de imagens, animações e ambientes interativos em 2D.

3. Add-ons em WoW

No *World of Warcraft*, os *add-ons* possibilitam alterações em vários aspectos da interface de usuário. O escopo das alterações inclui aspectos estéticos (alterações gráficas), aspectos de interação (alterações em ações e comandos do usuário), e incremento e expansão de funções do jogo.

As possibilidades de extensão em WoW são inúmeras. Sua interface de programação (API) e a transparência fornecida pela implementação da interface do jogo permitem que o usuário passe a ter consciência e onisciência sobre a interface do programa e seus elementos, criando, expandindo e modificando livremente. Tudo isso, no entanto, requer que o usuário tenha algumas habilidades de programação.

A criação de add-ons para WoW requer tanto conhecimento sobre Lua e XML, quanto sobre a própria API do jogo. No momento, não existe nenhuma ferramenta que permita a criação de tais *add-ons* sem programar ou mesmo que reduzam o esforço em programação.



Figura 1. Interface de Usuário básica do WoW.¹

¹ Fonte: World of Warcraft, editado pelos autores.

A Figura 1 representa e descreve a interface de usuário básica do jogo, sem uso de extensões ou com qualquer alteração nas configurações básicas.



Figura 2. Interface do WoW modificada por add-ons¹

Já a Figura 2 apresenta um exemplo de interface de usuário modificada, com o acréscimo de *add-ons* que alteram os elementos já existentes e outros que incluem novas funções ao jogo.

Os *add-ons* em WoW são pequenos códigos de programas (scripts) escritos em Lua que são interpretados em tempo real. Cada script é uma série de comandos, e estruturas de controles típicas de qualquer linguagem de programação.

Para entender melhor como funciona a interface de World of Warcraft e como seria a criação de um *add-on*, vamos a um exemplo ilustrativo.



Figura 3. Destaque de um add-on criado¹.

Para construirmos o add-on responsável por mostrar a latência do jogo no canto inferior da tela (Figura 3), precisamos adotar a seguinte linha de raciocínio:

1. Criar um Frame e ajustar as suas configurações;
2. Posicionar o Frame no canto inferior da tela;
3. Popular o Frame criado com uma nova String;
4. Recuperar o valor de latência do jogo usando a função da API e atribuir à String criada anteriormente;
5. Ajustar as configurações de posição da String no Frame;
6. Ajustar o Add-on para ser exibido quando o usuário entrar no jogo;
7. Executar o Add-on.

Porém este é apenas um pequeno exemplo. *Add-ons* em World of Warcraft podem ir de simples extensões como essa - com cerca de 40 linhas de código - a projetos verdadeiramente complexos - com mais de 25.000 linhas.

Para um usuário iniciante, criar o exemplo da Figura 3 pode se tornar uma tarefa complexa. Desenvolver um código como esse envolve desde a criação de *frames* e elementos de interface ao conhecimento sobre as funções e capacidades da API do jogo, além de vários conceitos de programação, como o uso e criação de funções e a manipulação de variáveis e constantes.

4. O EUPAT for WoW

O *EUPAT for WoW* é um ambiente interativo desenvolvido em JAVA. Através dele, o usuário pode criar e salvar projetos, aprender um pouco mais sobre a linguagem Lua e técnicas de programação, ou exportar projetos de código completos no formato de Add-ons para WoW. O processo de programação no EUPAT ocorre através de manipulação direta (*drag and drop*) de objetos que representam elementos de programação.

A interface de usuário do *EUPAT for WoW* está organizada em três áreas: exibição de comandos, edição do add-on e a tela de exibição.

4.1 A Área de Exibição de Comandos

Nessa sessão do EUPAT o usuário seleciona quais comandos irá utilizar em seu *add-on*. Os comandos estão divididos em comandos de execução (como funções e operadores) e comandos de armazenamento e parâmetro.

Para os comandos de execução temos três abas: uma com as funções e comandos de Lua, outra com as funções da API de World of Warcraft e a terceira com as operações e operadores que podem ser utilizados (como o igual, menor, soma e subtração).

Já para os outros comandos, são exibidas duas abas: uma com as variáveis e uma para ajustar e armazenar algum dos parâmetros que podem ser utilizados pelo jogo (como distância, alvo, tamanho de Fontes etc).

4.2 A Área de Edição de Add-ons

Na área de edição devem ser colocados os comandos do add-on que o usuário pretende construir. A construção do código se dá através da interface *drag and drop* do aplicativo. Em outras palavras, o usuário arrasta os comandos que estão exibidos no campo descrito anteriormente para o campo Comandos do Usuário, onde os mesmos são organizados em uma lista. Essa lista representa a sequência de comandos que serão executados pelo *add-on* dentro do jogo.

4.3 A Tela de Exibição

Nesta tela o usuário tem acesso a uma miniatura do que seria a tela de jogo e ao avatar do *EUPAT for WoW*: um *Orc* (personagem de World of Warcraft) que responde algumas das dúvidas de programação do usuário bem como acompanha o mesmo na construção de Add-ons simples.

O avatar no *EUPAT for WoW* serve como intermediário para a opção *Teach me!* do aplicativo. Através desta opção, os usuários iniciantes - aqueles que não possuem conhecimento em programação - podem participar de um tutorial no formato de mini-game, onde o usuário deve realizar pequenas tarefas instruídas pelo avatar.

O objetivo dessas tarefas é introduzir alguns conceitos básicos de desenvolvimento de *add-ons* na prática, contextualizando as implementações realizadas pelo usuário com a interface de World of Warcraft.



Figura 4. Áreas de edição do *EUPAT for WoW*.

4.4 Exemplo

Para ilustrar a construção de add-ons no *EUPAT for WoW*, vamos apresentar um exemplo simples.

Em algumas circunstâncias, um jogador de *World of Warcraft* pode precisar de uma exibição rápida e direta de algumas informações importantes na tela do jogo. Por exemplo, um jogador gostaria de exibir o tipo, o valor atual e o valor máximo da *Energia* (Power) de seu personagem. A *Energia* (Power) em WoW limita a quantidade de habilidades que o personagem pode utilizar antes que tenha que recarregar a mesma. Essa é uma ferramenta muito utilizada em jogos, principalmente de RPG, e se assemelha à munição utilizada em jogos de tiro. Em WoW, a *Energia* do personagem pode ser de um (e apenas um por vez) dos seguintes tipos: *Mana*, *Rage*, *Focus*, *Energy* e *Runic Power*.

Para exibir tais informações em tela, o usuário decide criar um *add-on*. Utilizando o EUPAT, ele deve realizar os seguintes passos:

1- Criar um *Frame* para armazenar os valores.

Como o EUPAT possui um botão específico para criação de Frames, o usuário seleciona essa opção e clica em um ponto da Tela de Exibição para construir seu Frame (clicando e arrastando na Tela de Exibição o usuário define o tamanho do Frame).

2- Criar variáveis para armazenar os valores.

Criado o Frame, o usuário entra na aba *Variables* do aplicativo e seleciona a opção *Common Variable*. O aplicativo pede para que o usuário selecione um nome para sua nova variável. A nova variável é então exibida no campo de Comandos do Usuário. Como o usuário deseja três variáveis (uma para o tipo, uma para o valor atual e uma para o valor máximo de Energia), ele faz o mesmo para duas novas variáveis.

3- Pegar o tipo de Energia do personagem e armazenar na variável.

O usuário deve entrar na aba *API* do aplicativo e então selecionar e arrastar a função *unitPowerType*, que retorna o tipo de Energia do personagem, para o campo Comandos do Usuário. Como a função retorna um valor numérico para o tipo de Energia, o usuário deve converter o retorno para uma String utilizando a expressão 'IF' (um condicional do tipo "SE <condição_booleana> ENTÃO"). Exemplo:

```
IF unitPowerType == 0
  <variable1> = "Mana"
END
```

O usuário deve então repetir a operação para os outros casos.

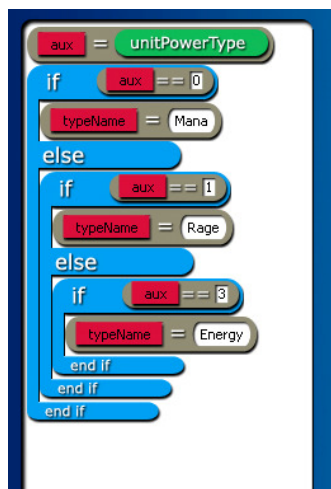


Figura 5. Exemplo de um programa sendo construído no EUPAT for WoW

4- Armazenar o valor atual e máximo de Energia do personagem.

O usuário deve entrar na aba *API* e selecionar respectivamente *unitPower*, que retorna a Energia atual do personagem, e *unitPowerMax*, que retorna o valor máximo da Energia do personagem. Através do *drag and drop*, o usuário deve arrastar cada função para o campo Comandos do Usuário. Deve ser criada então uma atribuição da função à respectiva variável que vai armazená-la. Exemplo:

```
<variable2> = unitPower
<variable3> = unitPowerMax
```

5- Adicionar as variáveis ao Frame criado.

Por fim, o usuário arrasta as variáveis para dentro do Frame criado anteriormente na Tela de Exibição do aplicativo, com isso o EUPAT entende que aquelas variáveis estão sendo armazenadas para exibição no respectivo Frame.

4. Conclusão

Com os resultados deste trabalho pretende-se apresentar uma solução para diminuir o esforço realizado por usuários não-programadores na construção de *add-ons* para *World of Warcraft* e abrir caminho para a discussão acerca de soluções para a introdução de técnicas e ferramentas que facilitem a criação de extensões para aplicativos abertos à Programação por Usuário Final.

De imediato, através da ferramenta desenvolvida, o EUPAT for WoW, pretende-se introduzir conceitos básicos de programação para usuários não-programadores, bem como fornecer uma ferramenta que auxilie no desenvolvimento de *add-ons* mais elaborados por usuários avançados.

O EUPAT está em fase de conclusão (motivo pelo qual ele está sendo submetido a esta conferência como artigo resumido) e testes ainda estão sendo realizados. A realização de testes visa avaliar os aspectos de correção, desempenho e também de usabilidade. Em relação a estes aspectos, iremos validar a eficácia do EUPAT para a construção dos *add-ons* e também as possibilidades de utilizá-lo no ensino de técnicas de programação.

Referências

- ALICE, 2009. Alice [online] Carnegie Mellon University. Disponível em: <http://www.alice.org> [Acessado 26 Julho 2010].
- ANTUNES, R., FONSECA, B., MARTINS, P. E MORGADO, L., 2006. Use of 3-D Virtual Environments to support the learning of programming. Em: Méndez-Vilas, A., et.al. (Eds.), *Current Developments in Technology-Assisted*, 2006, vol. I. Badajoz, Spain: Formatex, 689-292.
- BARBOSA, S.D.J., 1999. *Programação via Interface*. Tese de Doutorado, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil.
- CHICKENFOOT, 2007. Chickenfoot [online]. Disponível em: <http://groups.csail.mit.edu/uid/chickenfoot> [Acessado 26 Julho 2010].
- FRYE, C. 2003 MICROSOFT OFFICE EXCEL 2003 STEP BY STEP. MICROSOFT PRESS.
- NARDI, B., 1993. *A Small Matter of Programming, Perspectives on End User Computing*. MIT Press, Cambridge MA.
- SCRATCH, 2009. Scratch para Educadores [online] MIT Media Lab. Disponível em: <http://info.scratch.mit.edu/Educators> [Acessado 26 Julho 2010].