

Plataforma para Autoria de Jogos Simples

Pedro Menezes Ribeiro de Souza, Claudio Esperança, Ricardo Marroquim
Laboratório de Computação Gráfica (LCG) - COPPE - UFRJ
Rio de Janeiro, Brasil
pedromrsouza@poli.ufrj.br, esperanc@cos.ufrj.br, marroquim@cos.ufrj.br

Resumo—O projeto tem como objetivo facilitar a criação de jogos didáticos simples por pessoas sem conhecimento na área de programação. Parte-se do princípio que o autor fornece uma especificação da lógica do jogo através de um arquivo de configuração e de uma série de imagens. Um programa de simulação lê o arquivo de configuração e os arquivos com imagens e se encarrega de apresentar a interface do jogo ao usuário, gerando animações, processando interações e, de forma geral, assegura-se que as regras do jogo são seguidas.

Keywords—Jogos simples; Máquina de estados; Ferramenta de autoria.

I. INTRODUÇÃO

Jogos educativos têm sido usados com frequência cada vez maior como ferramenta pedagógica. A construção de jogos de computador pode se realizar de forma totalmente manual ou utilizando-se ferramentas tais como as bibliotecas RubyGame [1], JMonkeyEngine [2], Allegro [3], engenhos de jogos tais como RPG Maker [4], Unity [5] ou também plataformas de desenvolvimento dedicadas, como o XNA [6]. A curva de aprendizado para o desenvolvimento de jogos utilizando tais ferramentas é normalmente bastante acentuada, por isso, com o intuito de atenuar esta curva, vêm sendo criadas plataformas tais como ClassTools [7], Sploder [8] e Adventure Maker [9] que pretendem ser ferramentas de autoria de jogos informais usando um mínimo de interações por parte do autor. Estes últimos, entretanto, sofrem de relativa baixa expressividade na especificação das regras do jogo.

O presente projeto visa a construção de uma plataforma para desenvolvimento de jogos educacionais simples que possam ser expressos usando um número limitado de elementos visuais (imagens), mas com relativa flexibilidade no que se refere à especificação de regras do jogo.

II. DESCRIÇÃO DA PLATAFORMA

Como primeira etapa do projeto, está sendo desenvolvida uma aplicação capaz de simular jogos expressos através de um conjunto de imagens e um arquivo de configuração, no qual o jogo será inteiramente desenvolvido. O arquivo de configuração, que segue a sintaxe XML (eXtended Markup Language), contém uma descrição das cenas e atores a serem representados visualmente por instâncias de imagens, bem como as regras do jogo. O protótipo de simulação, que está atualmente escrito na linguagem de programação Processing [10], similar à linguagem Java, é encarregado de processar o arquivo XML, criando um arquivo executável do jogo. Em uma segunda etapa, pretende-se construir uma aplicação que

possibilite que a autoria das cenas e de algumas regras simples possa ser realizada através de uma interface gráfica, gerando automaticamente o arquivo de configuração.

Pode-se com esta plataforma criar jogos que são baseados em interações simples como arraste e clique, como também regras envolvendo relações geométricas e lógicas entre objetos da cena, tais como sobreposição, proximidade, dentre outras ações simples, que em conjunto podem gerar uma grande quantidade de variações de regras.

III. JOGO-PROTÓTIPO

Para guiar o desenvolvimento da plataforma, pretende-se implementar todas as funcionalidades necessárias para a simulação de um jogo didático específico na área de farmacologia. O jogo se dá a partir de uma célula, seus receptores e enzimas. Após certo tempo preestabelecido surgem bactérias no ambiente extracelular e liberam lipopolissacarídeos (LPS) e lipoproteínas ao redor da célula. O jogador deve posicionar o que foi liberado por elas nos respectivos receptores celulares e posicionar as enzimas que estarão dispersas pela célula em ordem correta, respeitando o tempo limite.

IV. UM EXEMPLO

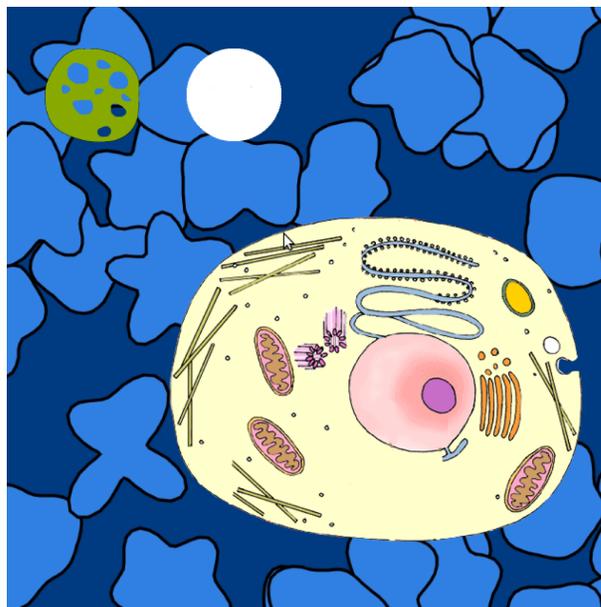


Figura 1. Célula, bactéria e anticorpo. Cena 1 do Jogo.

Para ilustrar o conceito do projeto, considere um jogo simples (Figura 1) onde tem-se uma célula, uma bactéria e um anticorpo. A meta do jogo será fazer com que a célula não morra. Para isso o jogador deve realizar uma das duas metas: arrastar o anticorpo até a bactéria ou arrastar a bactéria até a célula e após isso arrastar o anticorpo até célula. Quando uma das regras for cumprida, o jogador terá a opção de jogar novamente.

A estrutura do jogo pode ser representado por uma máquina de estados (Figura 2), na qual cada estado é uma cena e as transições entre os estados são as regras de cada cena.

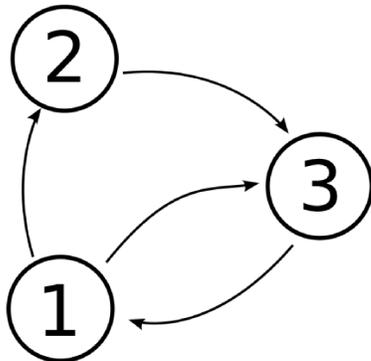


Figura 2. Máquina de estados correspondente ao jogo-exemplo. Cada estado é uma cena, cada transição de estados uma regra.

O autor do jogo inicialmente especifica o tamanho da tela do jogo. Supondo que esteja definido um tamanho de 650x650 pixels, deve introduzir no arquivo XML:

```
<screen size="650,650" />
```

Após isso o autor deve preparar imagens em fundo transparente representando cada elemento e introduzir esses elementos no arquivo XML na seção <images>:

```
<images>
<img source="bac.png" ImageName="bacteria"/>
<img source="anti.png" ImageName="antibody"/>
<img source="celula.png" ImageName="celula"/>
<img source="again_icon.png" ImageName="again"/>
<img source="background.png" ImageName="bg1"/>
</images>
```

Em seguida, o arquivo XML possui uma seção chamada <objects> onde são instanciados os elementos que serão usados em todo jogo, fazendo com que eles, assim, possuam atributos com valores globais. Por exemplo:

```
<objects>
<obj pos="100,100" id="bac1" ImageName="bacteria" static="no"/>
<obj pos="250,100" id="anti1" ImageName="antibody" static="no"/>
<obj pos="400,400" id="cell" ImageName="celula" static="yes"/>
<obj pos="400,400" id="again" ImageName="again" static="yes"/>
</objects>
```

Observa-se que elementos designam as imagens respectivas, seus identificadores, suas posições iniciais e outros atributos. Em particular, o atributo `static` indica se o elemento pode ou não ser arrastado pelo jogador.

Em seguida, o arquivo XML possui uma seção descritiva das cenas onde o jogo se desenrola. No jogo exemplo, há três

cenas, a primeira contendo a célula, bactéria, anticorpos e as duas regras.

```
<scene id="1">
<bg ImageName="bg1" />
<obj id="bac1" pos="100,100"/>
<obj id="anti1" pos="250,100"/>
<obj id="cell"/>
<near obj1="bac1" obj2="cell" distance="1" gotoScene="2"/>
<near obj1="anti1" obj2="bac1" distance="1" gotoScene="3"/>
</scene>
```

As posições e atributos dos objetos podem ser alterados também nesta seção.

A segunda cena será decorrente da ativação da primeira regra. Nesta cena têm-se apenas a célula, o anticorpo e a regra que descreve a interação entre o anticorpo e a célula:

```
<scene id="2">
<bg ImageName="bg1"/>
<obj id="anti1"/>
<obj id="cell"/>
<near obj1="anti" obj2="cell" distance="1" gotoScene="3"/>
</scene>
```

Por fim, define-se a terceira cena, onde o jogador terá a opção de jogar novamente:

```
<scene id="3">
<bg color="140,5,90"/>
<obj id="again"/>
<click obj="again" gotoScene="1"/>
</scene>
```

V. CONCLUSÕES

A plataforma para jogos simples aqui descrita pretende habilitar didatas de diversas áreas a construir jogos educativos simples sem ser necessário um treinamento específico extenso. Os jogos construídos pela plataforma permitirão que jogadores se utilizem de interações básicas usando o teclado, mouse ou outro dispositivo apontador. Tais interações disparam eventos que controlam o progresso do jogo através de um certo número de cenas até alcançar o objetivo final. Com isso, o projeto permitirá a professores das mais diversas áreas a criação de jogos didáticos, livros interativos e outras ferramentas que facilitem a transmissão de conhecimento ao aluno.

REFERÊNCIAS

- [1] "Rubygame," Jun. 2012. [Online]. Available: <http://www.rubygame.org/>
- [2] P. Martins and J. A. McCann, "ajme: making game engines autonomic," in *Proceedings of the 3rd International Conference on Fun and Games*, ser. Fun and Games '10. New York, NY, USA: ACM, 2010, pp. 48–57.
- [3] B. M. d. Sousa and B. M. Desousa, *Game Programming All in One*. Premier Press, 2002.
- [4] "Rpg maker," Jun. 2012. [Online]. Available: <http://www.rpgmakerweb.com/support/tutorial>
- [5] W. Goldstone, *Unity 3.X Game Development Essentials*, ser. Community experience distilled. Packt Publishing, 2011.
- [6] C. Reas and B. Fry, *Getting Started With Processing*, ser. Make Series. O'Reilly Media, 2010.
- [7] "Class tools," Jun. 2012. [Online]. Available: <http://classtools.net/>
- [8] "Sploder," Jun. 2012. [Online]. Available: <http://www.sploder.com/>
- [9] "Adventure maker," Jun. 2012. [Online]. Available: <http://www.adventuremaker.com/overview.htm>
- [10] C. Reas and B. Fry, *Getting Started With Processing*, ser. Make Series. O'Reilly Media, 2010.