PROJETO RECYCLER – AGENTES REATIVOS[[1]](#footnote-1)

Gregory Miola da Silva[[2]](#footnote-2) <gregory@mukiranas.com>

Rodrigo Perozzo Noll[[3]](#footnote-3) <rodrigo.noll@ulbra.edu.br> – Orientador

Universidade Luterana do Brasil (Ulbra) – Curso de Ciência da Computação – Câmpus Gravataí

Av. Itacolomi, 3.600 – Bairro São Vicente – CEP 94170-240 – Gravataí - RS

27 de setembro de 2012

Resumo

Este artigo descreve algumas das dificuldades encontradas no desenvolvimento e elaboração de trabalho sobre agentes reativos para a disciplina de Inteligência Artificial Curso área de informática da Universidade Luterana do Brasil (Ulbra), utilizando ferramentas que já foram estudadas em outras disciplinas do curso.

**Palavras-chave:** Inteligência Artificial; IA; Agentes Reativos; jQuery; Html; CSS.

Abstract

**Title:** “Project Recycler – Reactive Agents”

This article describes some of the difficulties encountered in the development and preparation of work about Reactive Agents for the discipline of Artificial Intelligence Course Area Computer Lutheran University of Brazil (Ulbra), using tools that have already been studied in other disciplines of the course.

**Key-words:** Artificial Intelligence, AI, Reactive Agents, jQuery, Html; CSS.

1. Introdução

Este artigo trará uma abordagem parcial de como foi o processo de desenvolvimento do projeto sobre agentes reativos e as ferramentas utilizadas na solução, além das principais dificuldades encontradas durante sua execução.

A Seção 2 descreve algumas definições das tecnologias utilizadas. A seguir, a Seção 3 apresenta sua implementação e a Seção 4 descreve as principais dificuldades encontradas. Por fim, são apresentadas as conclusões.

1. Tecnologias utilizadas
   1. HTML

A linguagem HTML foi desenvolvida por *Tim Bernes Lee*, ganhou popularidade após a criação do *browser* *Mosaic* na década de 1990. Entre 1992 e 1995 ganhou as versões 2.0 e 3.0. Em 1997 foi lançada a versão 3.2 com os padrões W3C em 1999 a versão 4.0. Foi criado o HTML5 para atender algumas necessidades como as APIs que formarão a base da arquitetura *web* (DOM Level0), fornece ferramentas para CSSe *JavaScript*, mais iteratividade sem necessidade de instalação de *plug-ins*, retro compatibilidade.

* 1. jQuery

“jQuery is a new kind of JavaScript Library. jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. jQuery is designed to change the way that you write JavaScript.” [Site oficial do jQuery](http://www.jquery.com/)

Ou seja, o jQuery simplifica diversas coisas para um desenvolvimento mais rápido. jQuery é projetado para mudar a maneira que você escreve JavaScript. Mas ele não dita qual a estrutura nem qual o fluxo em que sua aplicação deve ser montada, consequentemente, por esses e outros motivos, não deve ser considerado um framework.

* 1. Cascade Style Sheet - CSS

CSS é uma linguagem de estilo, utilizada para formatação e descrever a apresentação de um documento escrito em uma linguagem de marcação como HTML, XML ou XHTML. Essas definições são aplicadas para alterar a forma como as informações são apresentadas.

As formatações normalmente são aplicadas as tags do HTML como TABLE, P, H1 ou BODY. No CSS são chamados de seletores. Nos seletores se determina a propriedade de formatação que cada um deva receber, atribuindo as propriedades e valores a cada seletor.

1. Implementação

O aplicativo Recycler foi desenvolvido utilizando HTML simples e representando seus objetos por imagens via CSS, porém toda a lógica e manimulação dos elementos foi feito em jQuery.

O primeiro passo foi definir uma linguagem que proporcionaria maior facilidade na manipulação de objetos interagindo em uma interface gráfica. Feito isso, os passos que se suced em são desenvolver uma lógica e organizar elas em classes para que as instâncias tenham maior flexibilidade nas suas chamadas.

* 1. Descrição do problema

O trabalho consiste basicamente em simular agentes recicladores de lixo autônomos. Isto deve ser feito a partir da construção de um ambiente parametrizado, definido por uma matriz n x n onde este será o ambiente ao qual estes agentes irão interagir.

Os agentes irão interagir neste ambiente mediante a ciclos, sendo que cada ciclo irá corresponder a uma atualização completa de todo o ambiente uma vez (abordagem sequencial) ou também podendo ser definido um tempo t para que ocorra tal ação (abordagem paralela). Co-existem nesse ambiente:

* Agente Reciclador de Lixo;
* Lixeira de Lixo Orgânico;
* Lixeira de Lixo Seco;
* Lixo Orgânico;
* Lixo Seco.
  1. Classes utilizadas

As classes a seguir compõem o ambiente e nelas são armazenadas todas as informações referentes a criação do ambiente, população dos mesmos com os objetos já referidos anteriormente, manipulação dos mesmos e as validações necessárias para aplicar a lógica determinada no escopo deste trabalho.

1. Reciclador.js

Classe responsável por criar o ambiente e controlar os movimentos do agente. Utiliza a classe ambiente.js para que efetue a criação dos objetos e controla com eventos de click os movimentos de ciclo unico ou ciclos seguidos.

1. Ambiente.js

Classe responsável por criar e popular a matriz do ambiente com os objetos que serão utilizados, além de armazenar as informações de cada agente durante a sua manipulação. Na criação dos objetos são feitas chamadas para cada um deles passando funções dessa própria classe como delegate para execução de funções a partir destas instâncias criadas.

1. Matriz.js

Classe que armazena informações e cria elementos html na página para ser manipulada pelo ambiente com todos os objetos já inseridos.

1. Localizacao.js

Classe que valida, cria e manipula as localizações de cada elemento da matriz, utilizada por todos os objetos.

1. Agente.js

Classe que representa o objeto do agente, recebendo as informações que nele devem ser manipuladas e validadas como localização, sacos e quantidade de lixos, além da lógica para descartar e começar uma nova varredura no ambiente.

1. Saco.js

Classe que cria um saco de lixo do tipo 'random' para o agente. Trás informações básicas para sua validação pelo agente.

1. Lixos.js

Classe responsável por criação dos objetos de lixos e lixeiras do tipo 'random'. Trás informações básicas para sua validação pelo agente e/ou pelo ambiente.

1. Util.js

Classe com unica finalidade de criar a imagem de cada objeto.

* 1. Interface

A interface desenvolvida foi bem simples, demonstra a matriz n x n, juntamente com todos os elementos nela contidos conforme especificação do trabalho. O arquivo index.html carrega todos os fontes javascript que trazem a logica desenvolvida utilizando jQuery. A figura 1 apresenta um exemplo gráfico do ambiente em que é executado o código.



**Figura 1 – Apresentação do ambiente**

1. Dificuldades encontradas

A maior dificuldade encontrada durante o andamento do projeto com certeza foi a leitura e a continuidade do processo de desenvolvimento do código. Conforme os arquivos fontes foram crescendo com os condicionais de se/então/se não, fica mais complexo fazer os testes e realizar a validação de determinada ação dos agentes.

Também logo no inicio do desenvolvimento foi encontrado problemas para trabalhar com o escopo de variáveis do jQuery para as classes, quando tentamos simular elas com functions. Problema este solucionado utilizando uma variável recebendo a instância direta na declaração e criando um construtor para receber por um delegate a instrução que será executada (no caso a respectiva function).

1. Conclusão

Este artigo apresentou as etapas de desenvolvimento do projeto de agentes reativos e as suas principais dificuldades para o seu desenvolvimento. Lembrando que nesse tipo de aplicação, onde os agentes reagem a regras pré-determinadas pelo seu desenvolvedor, na prática não passa de testes condicionais de IF ELSE, o que não é algo muito simples de colocar em prática devido as várias alternativas e regras que compõem as especificações do trabalho.

Agradecimento(s)

Agradeço a contribuição do amigo Bruno Souza que ajudou a pensar na melhor forma de organizar e trabalhar com jQuery e Html para o desenvolvimento do trabalho.

Referências

BATISTA, Diogo C. T. O impacto do HTML5 no desenvolvimento para a internet. Disponível em: <http://www.diogocezar.com/files/html5/artigo\_html5.pdf> Acessado em: 03 Julho 2012.

TIAGO, Luis jQuery: biblioteca ou framework?. Disponível em: < http://jquerybrasil.org/jquery-biblioteca-ou-framework/> Acessado em: 24 de Setembro 2012.

AMARAL, Luis Gustavo CSS Cascatding Style Sheets. Disponível em: <http://www.olivreiro.com.br/pdf/livros/cultura/1809834.pdf> Acessado em: 03 Julho 2012.

1. Artigo elaborado para descrever o desenvolvimento do trabalho de G1 para IA I sobre Agentes Reativos. [↑](#footnote-ref-1)
2. Aluno da disciplina de Inteligência Artificial do curso de Ciência da Computação, na Ulbra Gravataí. [↑](#footnote-ref-2)
3. Professor da disciplina de Inteligência Artificial do curso de Ciência da Computação, na Ulbra Gravataí. [↑](#footnote-ref-3)