

IMPLEMENTAÇÃO DE UMA INTERFACE GRÁFICA PARA O CONTROLE DE AGENTES AUTÔNOMOS EM AMBIENTES DINAMICOS

Ana Carolina Cabral de Almeida – Bolsista PIBIC/CNPq
Universidade de Taubaté – UNITAU
Orientador: Carlos Ho Shih Ning
Divisão de Processamento de Imagens
Instituto Nacional de Pesquisas Espaciais – INPE

Muito se fala no uso do computador para soluções de problemas do nosso dia a dia: cadastro de pessoas, contabilidade, pagamento de contas via Internet, etc. O computador, ou os programas nele executados, acabam por representar uma personificação de um indivíduo capaz de realizar tarefas especializadas. Uma maneira de fazer com que os programas de computador sejam um pouco mais inteligentes seria o de dar uma autonomia supervisionada a esses programas tratando-os como agentes.

Muitos dos programas de computador que normalmente não são visualmente visíveis podem ser apresentados numa forma gráfica. Um exemplo disso é o Microsoft Word Help, que todos nós conhecemos, que nada mais é do que um programa típico de ajuda que reside no menu de Help em qualquer programa mas com um aspecto humano, ou simpático. Isto é, quando o Help é acionado uma figura aparece (Fig.1). Essa figura se movimenta, dá dicas e sugere informações mais apropriadas para cada situação. Esse tipo de programa assistente é mais conhecido com *wizard*, mas agente é uma nomenclatura mais geral para programas inteligentes sejam elas visíveis ou não.

Neste trabalho pude conhecer os mecanismos de um sistema de animação de múltiplos robôs autônomos que se movimentam num ambiente cheios de objetos como cadeiras, mesas, copos e balcões. Os robôs são entidades (agentes) com formas humanóides e com comportamentos autônomos (ações). Eles têm um repertório de movimentos, ou ações, que se adaptam para cada situação. Por exemplo, tomemos o cenário de um bar (Fig.2) e que um certo robô agente deseja apanhar uma bebida, neste caso: ou ele vai até o local onde está a bebida e apanha o copo contendo a bebida; ou ele pede para um agente *garçon*; ou ele pede para um agente *barman*. Ainda assim, para cada uma dessas alternativas o robô deve decidir uma das muitas alternativas disponíveis. No caso do robô decidir-se por ir pessoalmente ao local da bebida ele deve verificar onde, em que balcão, tem copos com bebidas e ir até lá. Se ele já estiver no próprio local da bebida ele não precisará caminhar, bastando estender o braço e apanhar o copo. Mesmo nesta simples ação o robô ainda deverá decidir se deverá inclinar/virar o corpo ou simplesmente estender o braço, e assim por diante.

As ações que os agentes robôs podem realizar em cada situação depende do repertório de ações implementadas. A capacidade de percepção de uma variedade de situações para acionar a ação apropriada também faz parte do repertório do robô. Isto é feito através de regras de decisão do tipo: *SE percepção da situação ENTÃO realize uma ação*. Em cada ação escolhida novas sub-decisões podem ser tomadas até que uma ação concreta final possa ser escolhida. A codificação dessas regras de decisão é feita numa linguagem de Inteligência Artificial chamada *Prolog*. Estas duas partes da regra são na realidade programas. Se o primeiro programa *percepção da situação* executar com sucesso então o programa *realize uma ação* será feito. Dessa forma novas regras poderão ser facilmente agregadas ao repertório.

Relatório Final

IMPLEMENTAÇÃO DE UMA INTERFACE GRÁFICA PARA O CONTROLE DE
AGENTES AUTÔNOMOS EM AMBIENTES DINAMICOS

Ana Carolina Cabral de Almeida – Bolsista PIBIC/CNPq
Universidade de Taubaté – UNITAU
Orientador: Carlos Ho Shih Ning
Divisão de Processamento de Imagens
Instituto Nacional de Pesquisas Espaciais – INPE

Período de Janeiro de 1998 à Junho de 1998.

Introdução

O uso do computador está cada vez mais frequente no cotidiano do nosso dia a dia, ora fazendo parte do nosso ambiente de trabalho ora servindo de terminal de compras ou de consultas. A variedade de programas disponíveis é tão grande que é praticamente impossível dedicar o pouco do nosso tempo para se ter uma razoável noção do seu funcionamento, quanto mais a solução de uma eventual falha destas. Diante de tal situação, muitas vezes nos sentimos desorientados e incapazes encontrar alguém que nos auxilie no entendimento do funcionamento de tais programas que atendem um determinado serviço. Assim, cai por terra o mito do computador como uma ferramenta útil para as tarefas repetitivas em substituição ao trabalho humano.

O foco principal deste trabalho foi o de compreender algumas das técnicas de Inteligência Artificial empregadas num sistema de animação computadorizada. Antes de entrar nessa questão em definitivo, eu gostaria ilustrar um exemplo que mostra que o uso de Inteligência Artificial está mais presente do a gente imagina.

Muitos dos programas de computador que normalmente não são visualmente visíveis podem ser apresentados numa forma gráfica. Um exemplo disso é o Microsoft Word Help, que todos nós conhecemos, que nada mais é do que um programa típico de ajuda que reside no menu de Help em qualquer programa mas com um aspecto humano, ou simpático. Isto é, quando o Help é acionado uma figura aparece (Fig.1). Essa figura se movimenta, dá dicas e sugere informações mais apropriadas para cada situação. Esse tipo de programa assistente é mais conhecido com *wizard*, mas agente é uma nomenclatura mais geral para programas inteligentes sejam elas visíveis ou não.

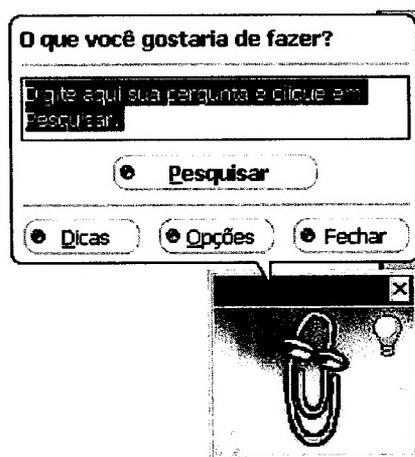


Fig.1 – Agente Help do Microsoft Word. Personificação de um programa de ajuda numa figura mais amigável.

Animação por Computador

A animação por computador é uma típica área de aplicação em que o computador é empregado para auxiliar o homem no trabalho de composição de uma série de atividades bastante monótonas. Nesta área, inicialmente o computador era considerado como sendo uma versão computadorizada da tradicional animação com papel e caneta. A seguir, com o desenvolvimento da computação gráfica e de computadores mais potentes, pode-se representar figuras tridimensionais e até simulá-los como tendo propriedades físicas.

Com o tempo foi se percebendo que simular objetos dinamicamente ou cinematicamente não era o suficiente para produzir os trabalhos mais corriqueiros. Mesmo com o uso do computador o animador tinha que definir todas as seqüências de transição que cada uma das figuras deveria sofrer. O trabalho se complicava extremamente quando mais de uma figura era animada, pois, só após toda a preparação da animação é que se percebia que algo não acontecia conforme o imaginado. Para complicar mais o problema, a animação se tornava impraticável quando as figuras eram articuladas e pior ainda quando se desejava que eles tivessem um comportamento típico do ser humano.

Surgiu-se então um novo tipo de animação conhecida como comportamental. Nessa nova abordagem foi empregada técnicas de Inteligência Artificial como forma de solucionar problemas complexos. Esse tipo de estratégia está presente nas tecnologias de software nos dias de hoje sob forma de agentes como foi mencionado anteriormente. No caso deste trabalho o sistema de animação tem duas componentes principais: a parte de controle e o uso de Inteligência Artificial e a parte de computação gráfica. Como parte do trabalho (Ning, 1996) já estava pronto e envolvia muitos conceitos complexos, o meu orientador pediu para que eu me concentrasse mais na primeira parte onde envolvia Prolog e componentes mais inteligentes.

Neste trabalho pude conhecer os mecanismos de um sistema de animação de múltiplos robôs autônomos que se movimentam num ambiente cheios de objetos como cadeiras, mesas, copos e balcões. Os robôs são entidades (agentes) com formas humanóides e com comportamentos autônomos (ações). Eles têm um repertório de movimentos, ou ações, que se adaptam para cada situação.

Metodologia

O Prolog (Bratko, 1987) é uma linguagem de programação usada em Inteligência Artificial. Com ela podemos fazer tudo que se faz com uma linguagem comum de computador e mais a representação de idéias. Podemos representar um objeto ou informação usando literais. Por exemplo, *copo*, *copo_verde*, *mesa*, *mesa_redonda*, etc.

Podemos representar também a relação entre dois objetos usando a forma de função:

objeto(copo) – entende-se que o copo é um objeto.

é_um(copo, objeto) – é uma outra forma de dizer que o copo é um objeto.

copo(condição, limpo) – a condição do copo é limpo.

condição_de(copo, limpo) - é uma outra forma de dizer que o copo está limpo.

Além disso, com o *flex* uma ferramenta baseado no Prolog, podemos armazenar informações sobre os objetos numa estrutura mais complexa na forma quadros, frames em Inglês. Dessa forma podemos buscar as informações diversas de um mesmo objeto.

```
frame entidade;
  default nome is ninguem and
  default cor is azul and
  default local is pos(0,0,0) and
  default direcao is 0 and
  default regio is mundo .

frame copo is entidade;
  default tipo is copo and
  default regio is balcao and
  default condicao is limpo and
  ...
```

Uma vez que definimos um quadro de um objeto podemos criar novas instâncias, ou copias, de objetos. Por exemplo, num programa podemos fazer:

```
copo_azul is an instance of copo;
copo_verde is an instance of copo;
```

Assim, qualquer um dos copos terá o seu conjunto de propriedades e que poderão ser modificadas quando forem necessário.

Uma forma importante de associar um conhecimento com o outro é feito através de regras, normalmente conhecidas como regra de decisão ou regras de inferência. As regras de decisão do tipo:

SE condição da situação ENTÃO realize uma ação. (em Português)
IF THEN (em Inglês)

Finalmente, o procedimento é uma forma de representar um programa usando *flex/Prolog*. Basicamente é uma sequencia de instrucoes que modificam as informacoes dos objetos ou criam novas instancias de objetos. Assim, ao modificar as informacoes, ou fatos, de um objeto afetamos a decisao de uma regra.

Num sistema como o de animação por computador onde temos muitas decisões a fazer teremos uma coleção grande de informações representadas na forma de quadros, regras e procedimentos. Com isso conseguimos realizar o que se chama de componente controle usando a Inteligência Artificial. Neste trabalho eu implementei um subconjunto dessa coleção como forma de compreender o seu funcionamento.

Componente do Controle

No cenário de animação temos diversos robôs, cada um com a autonomia de realizar atividades sejam elas predefinidas por script ou instigado pela situação. Obviamente, as ações que os agentes robôs podem realizar em cada situação depende do repertório de ações implementadas. A capacidade de percepção de uma variedade de situações para acionar a ação apropriada também faz parte do repertório do robô. Isto é feito através de regras de decisão do tipo: *SE percepção da situação ENTÃO realize uma ação*. A percepção é um conjunto de fatos acerca dos objetos e robôs do cenário. Em cada ação escolhida novas sub-decisões podem ser tomadas até que uma ação concreta final possa ser escolhida. A codificação dessas regras de decisão é feita em *Prolog*. Estas duas partes da regra são na realidade programas. Se o primeiro programa *percepção da situação* executar com sucesso então o programa *realize uma ação* será feito. Dessa forma novas regras poderão ser facilmente agregadas ao repertório. Essa idéia é facilmente visualizada na Fig. 2.

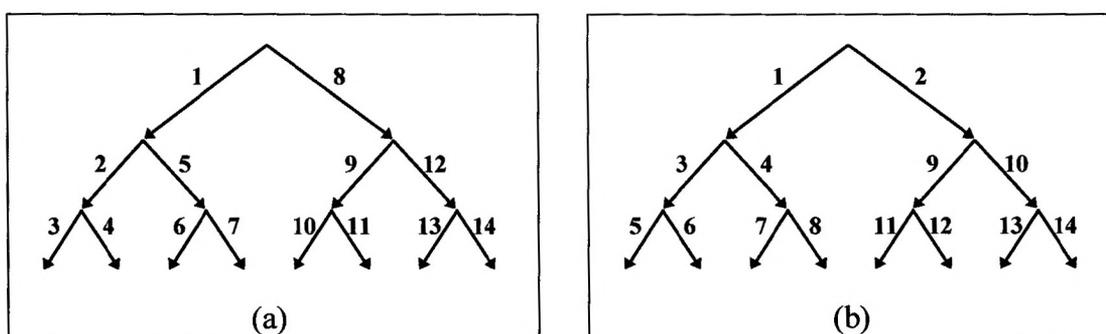


Fig. 2: (a) Busca em profundidade

(b) Busca em largura.

Na Fig. 2 cada regra está representada por uma seta. Cada ramo que parte nó são ações alternativas possíveis para as condições verificadas nos nós. Numa determinada situação podemos ter uma ou mais opções e dessa forma temos que estabelecer uma política de decisão: ou segue as primeiras condições validas e vai adiante, busca em profundidade; ou verifica as decisões possíveis e escolhe a melhor, busca em largura. Cada regra tem o seguinte formato:

```
rule < nome_do_plano >
  if plano( < nome_ação >, Instancia, Agente ) and
    < condições >
  then
    escolha_plano_ação( < nome_do_plano >, Instancia ).
```

Resultados

O sistema de animação contém uma coleção de regras que são acionadas conforme requisitadas. O acréscimo de um subconjunto de regras não altera muito o comportamento dos agentes, apenas dá mais opções na hora de escolher alternativas de ações. O cenário onde foi testado a animação é mostrada pela Fig. 3 com indicações de

visão através de diversas cameras. A Fig. 4 apresenta a visão da animação pelo topo e de frente ao cenário.

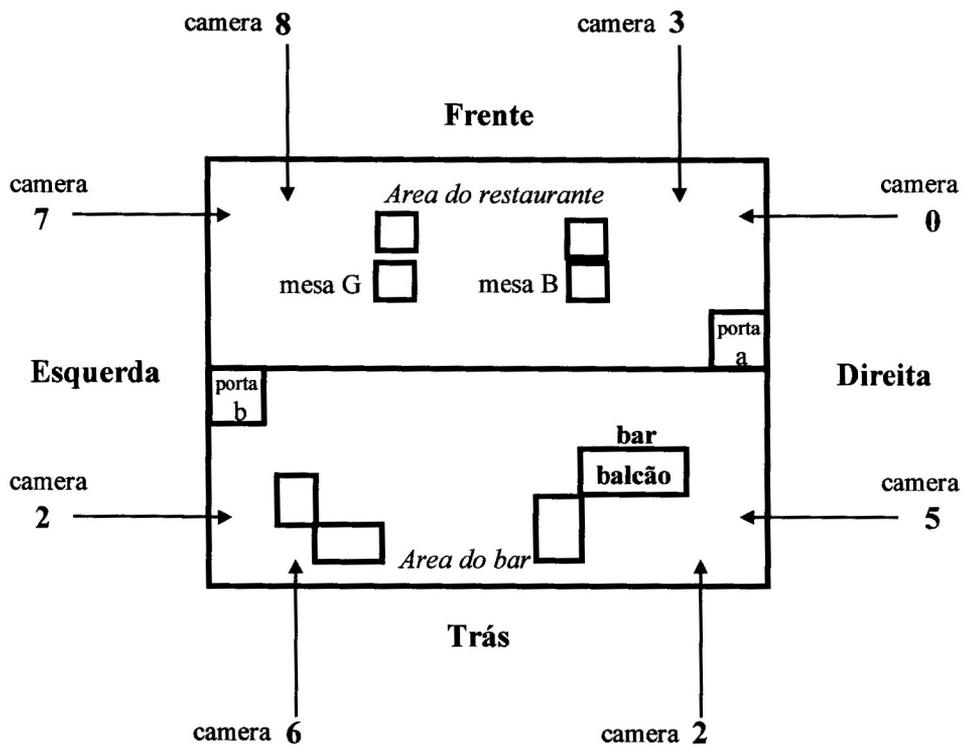


Fig. 3 – Distribuição dos objetos no cenário, vista do topo.

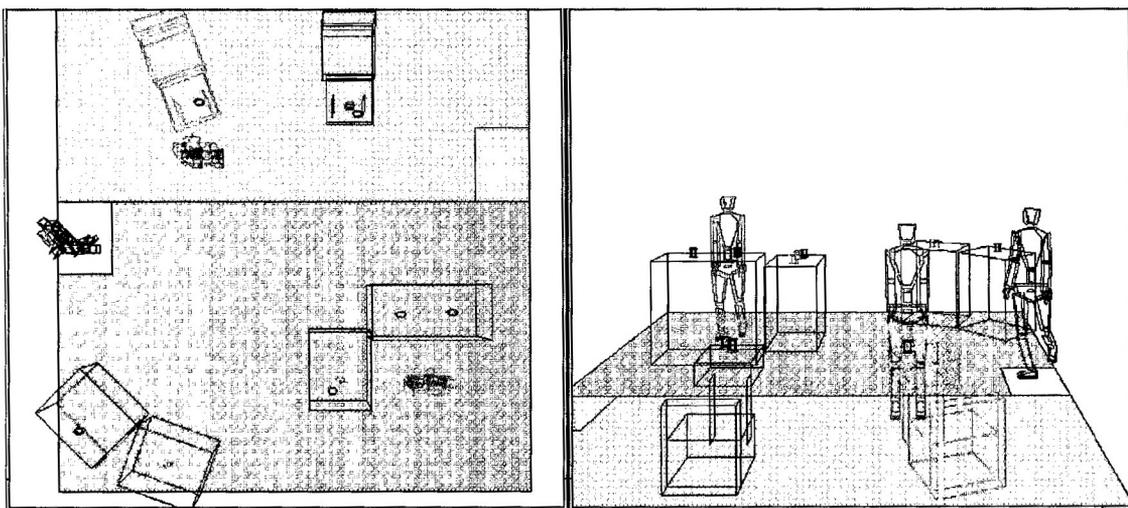


Fig. 4- (a) Visão do cenário por cima (b) Visão frontal do cenário durante a animação.

Conclusão

A visualização de agentes sob a forma gráfica permite que se “enxergue” a atividade de um programa, isto é, a associação de uma figura familiar a um programa, como é o caso do MS Word na Fig.1 e do robô neste trabalho na Fig.4, permite que se enxergue os programas como personagens que realizam atividades no computador. A inteligência desses personagens pode ser ampliada a medida que acrescentamos mais regras que identifiquem as situações com maior nível detalhe. Normalmente as figuras de agentes auxiliares podem ser simples que o da Fig. 1 ou mesmo não ter figura alguma mas só a parte do diálogo desta.

Neste trabalho de animação por computador as figuras são bastante complexas e difíceis de se lidar. Além disso, havia muitos assuntos envolvidos que não se aprende na escola o que me exigiu muito. O trabalho é bastante interessante pois as ferramentas utilizadas (flex/Prolog e Borland C++) estão sendo úteis para os meus trabalhos na escola.

Devido período limitado da bolsa, iniciado a partir de Janeiro de 1998, apenas parte do trabalho proposto será alcançado. Computação gráfica

Bibliografia

Bratko, I. "Prolog programming for artificial intelligence", Addison-Wesley, 1990.

Internet. "3-D Computer Animation". http://www.halcyon.com/asllc/cle_ani2.html

Ning, CSN "A framework for behavioural control in computer animation". Sussex University. England, 1996.

LPA-PROLOG. "Prolog Programming Guide", 1993.

Ana Carolina C. de Almeida