

PROJETO ASSISTIVA

Processo 580470/2008-3

Edital MCT/SETEC/CNPq N º 67/2008 - RHAE - Pesquisador na Empresa

Relatório Final



Resumo:

O projeto ASSISTIVA - *Assistente Inteligente Vocal para Maga Vittà* visa o desenvolvimento computacional pela empresa CONEXUM de um agente inteligente para assistência às crianças usuárias do jogo educacional e simulador de cidades Città, desenvolvido pelo LELIC (Laboratório de Estudos em Linguagem, Interação e Cognição da Faculdade de Educação da UFRGS). A Maga Vittà será uma assistente que dialogará com o usuário, relatando questões sobre a maquete virtual em desenvolvimento. Com a tecnologia desenvolvida através deste projeto, pretende-se a futura criação de outros assistentes virtuais para dispositivos móveis e outros sistemas de jogo ou monitoramento.

PROJETO ASSISTIVA

Relatório Final

Introdução

O Projeto Assistiva - *Assistente Inteligente Vocal para Maga Vitta*, foi aprovado em fevereiro de 2009 pelo edital MCT/SETEC/CNPq N ° 67/2008 - RHAE - *Pesquisador na Empresa*, no intuito de gerar uma assistente inteligente vocal (Maga Vitta) para o jogo educacional Città, desenvolvido para o projeto CIVITAS - Cidades Virtuais: tecnologias de aprendizagem e simulação, do Laboratório de Estudos em Linguagem, Interação e Cognição (LELIC), da Faculdade de Educação/UFRGS. O objetivo da assistente é alertar e questionar o usuário estudante quanto a procedimentos que levem ao prejuízo da cidade que está sendo construída por ele mesmo.

O desenvolvimento de todo o projeto foi realizado na sede da empresa CONEXUM Ltda., sala 106 do Centro de Empreendimentos em Informática, situado no Instituto de Informática da UFRGS - <http://www.conexum.com.br>

De junho de 2009 a novembro de 2011 foi prosseguida a programação do jogo Città e desenvolvida a assistente inteligente, a qual deveria ser resultante dos seguintes sistemas computacionais, segundo o projeto inicial (dezembro/2008):

- captura de fala com geração de características do léxico, prosódia e espectro vocal;
- análise sintática por sistemas estatísticos;
- análise semântica com acesso a ontologias;
- gerenciador de diálogos sensível ao contexto do sistema e do usuário;
- síntese de fala com prosódia.

A partir de nossos estudos para construção da assistente inteligente, percebemos que as etapas de análise sintática e semântica não seriam relevantes para o objetivo de apoio ao usuário estudante. Assim, demos maior ênfase ao gerenciador de diálogos.

Como a assistente inteligente, através do gerenciador de diálogos, necessitaria da percepção do ambiente (sensores) para elaborar uma questão ao estudante, percebeu-se a necessidade do mapeamento dos sensores junto ao mapa. A estes sensores demos o nome de “variáveis de ambiente” e, de comum acordo com a equipe do LELIC, elegemos a análise ambiental como primeiro fator de diálogo da assistente.

Os bolsistas **Leonivan Dias Moraes** (SET) e **Otto Lopes Braitback de Oliveira** (ITI) trabalharam, a partir do início de suas bolsas em junho de 2009 na definição e implementação destas variáveis, que ficaram assim estabelecidas como produzíveis e consumíveis na cidade virtual. Os objetos criados para composição da cidade virtual foram: casa, prédio, escola, hospital, igreja, fábrica, prefeitura, mercado, fazenda, árvore, ETA (Estação de Tratamento de Água), usina eólica, aterro sanitário, ETE (Estação de Tratamento de Esgoto), UTLH (Unidade de Tratamento

de Lixo Hospitalar). Foram pesquisados valores para atualização das variáveis no jogo. A atualização se dá a cada inserção ou retirada de objeto e por contabilidade de tempo (em dias). Por exemplo, uma casa construída começa a contar diariamente o consumo de alimento e energia, além da produção de lixo e esgoto por 4 pessoas. Assim, a cada dia as variáveis *alimento*, *energia*, *lixo* e *esgoto* são atualizadas para a casa construída. O mesmo acontece para os demais objetos existentes na cidade.

Em paralelo a esse desenvolvimento, o bolsista **Douglas Collar da Cunha** (ITI) vinha realizando levantamento bibliográfico e testes de sistemas de reconhecimento de voz em software livre.

Em setembro de 2009 o bolsista **Alexsandro Rosa dos Anjos** (DTI) iniciou seu trabalho dentro do contexto do gerenciamento de diálogo, finalizando as indicações das bolsas concedidas ao projeto. Uma vez que Leonivan havia começado um esquema mental para a agente inteligente Maga Vitta, viu-se a necessidade da separação do modelo de tomada de decisão do gerenciamento do diálogo. A partir daí distinguiram-se as tecnologias que seriam utilizadas para implementação da assistente inteligente, no que se referia aos sensores (variáveis de ambiente) e aos efetores (diálogo).

Após a leitura dos efetores, projetamos a tomada de decisões por raciocínio bayesiano. Em 2010 iniciamos os testes dentro desta perspectiva, obtendo nas simulações o resultado esperado acerca das ações que encadeariam os questionamentos ao estudante usuário do Città.

Uma vez definida a forma de tomada de decisões, determinamos que o gerenciamento mais adequado para os diálogos seria o Raciocínio Baseado em Casos. Tratando-se cada intervenção da assistente como um caso, tivemos um maior controle acerca das ações de diálogo e recorrência de eventos.

Alexsandro e Douglas desenvolveram um portal para o projeto, o qual foi hospedado no link <http://assistiva.conexum.com.br/>

Em março de 2010, Otto foi contratado como desenvolvedor por uma empresa de games e em seu lugar passamos a contar com **Taiuã Pires**. Como Otto, Taiuã ficou responsável pelo auxílio no desenvolvimento do jogo Città. Seu enfoque inicial foi a implementação dos bancos de dados para registro dos eventos do jogo.

Em junho de 2010 foi a vez de Douglas ser substituído por **Lisardo Kist**, que se aprofundou nos testes do sistema de reconhecimento de voz *Julius* (http://julius.sourceforge.jp/en_index.php), o qual já havíamos iniciados ensaios bem sucedidos. Douglas passou a estagiar em microeletrônica, área de foco de seu curso (Engenharia de Computação).

No segundo semestre de 2010 fizemos visitas a duas escolas do Projeto Civitas, onde colhemos dados sobre usabilidade do Città, além de gravações em vídeo e áudio que foram utilizadas para melhoria da interação com o jogo, projeto de interface e testes de modelos acústico e lingüístico.

Vale ressaltar que mantivemos contato e reuniões com membros do LELIC, os quais acompanhavam nossos trabalhos e faziam contatos com professoras e estudantes. Desses membros, podemos citar Paloma Dias Silveira (doutoranda UFRGS), Carla Inez Lima Freitas (professora municipal), Andrea Aparecida Konzen da Silva (doutoranda UFRGS e professora UNISC), Joelma Adriana Abrão Remião (doutoranda UFRGS), Márcio André Rodrigues Martins (professor UNIPAMPA) e Margarete Axt (professora UFRGS e coordenadora do LELIC).

Em 2011 tivemos a aprovação de uma bolsa no edital MEC/CAPES e MCT/CNPq/FINEP Nº 28/2010 – Programa Nacional de Pós-Doutorado - PNPD 2010, com o projeto *Città Cosmopolita - Simulador de redes de cidades*, que busca a interconexão das cidades construídas. Para a bolsa, foi selecionada a professora da UNEB Tânia Maria Hetkowski, membro do Grupo de Geoprocessamento daquela Universidade. Neste sentido, houve o início de uma parceria entre a equipe de Tânia, para apoio nos recursos multijogador do Città e leitura de mapas geográficos para terreno das cidades, além da aplicação da metodologia do Projeto Civitas em Salvador, em escolas atingidas pela UNEB. Também foi cogitado o intercâmbio com o Projeto Civitas Pró-África, que ocorre em Maputo, capital de Moçambique.

Como nosso projeto deveria ser encerrado em abril de 2011, foi solicitada uma prorrogação até novembro, a qual foi aceita. Houve uma sobra de bolsa, decorrente de não encontrarmos doutor disponível para a bolsa SET utilizada no projeto (Leonivan é Mestre em Computação). Segundo o relatório de andamento, nesta prorrogação deveriam ser desenvolvidos os seguintes itens:

- Testes de visualização e bate-papo da Maga Vitta na interface do Città.
- Implementação em Java do raciocínio bayesiano da Maga Vitta.
- Testes de modelo de reconhecimento de voz contínua.
- Testes de diálogo, para aperfeiçoamento do modelo de análise.
- Novas interações e testes nas escolas.

Destes itens, o primeiro a ser concluído foi o raciocínio bayesiano, utilizando-se a plataforma *JavaBayes* (<http://sourceforge.net/projects/jbnc>), a qual foi incorporada ao projeto Java do Città, obtendo-se sucesso no cálculo das variáveis de ambiente e obtenção das decisões necessárias às intervenções da Maga Vitta.

O reconhecimento de voz foi continuado com o aperfeiçoamento do modelo lingüístico e teste de palavras para reconhecimento, sempre dentro do modelo dos sistemas *HTK* (treinamento estocástico – <http://htk.eng.cam.ac.uk>) e *Julius* (reconhecimento de voz). Fizemos interações com bolsistas do Projeto Comunica (<http://comunica.conexum.com.br>), do Instituto de Informática da UFRGS, e com membros do Laboratório de Processamento de Sinais (LAPS) da UFPA, no sentido de otimizar o reconhecimento.

Quanto à síntese de fala, utilizamos o sistema *MaryTTS* (<http://mary.dfki.de>), também com apoio da equipe do Projeto Comunica, que desenvolveu um modelo acústico com voz feminina, adequada à Maga Vitta.

Em maio de 2011, Lisardo recebeu a notícia da aceitação de seu pedido de graduação sanduíche na *Technische Universität Berlin*. Em seu lugar entrou **Vinícius Brum Bertoldo**, que teve como

primeira tarefa otimizar os métodos Java do Città, uma vez que a necessidade de processamento dos diversos novos módulos estava tornando o sistema muito lento.

Em junho de 2011 foi a vez de Taiuã ser contratado como desenvolvedor de software, ficando em seu lugar **Mércio Luciano Teixeira Soares**. Luciano (como prefere ser chamado) auxiliou na substituição dos elementos gráficos, tornando a interface do Città mais amigável e o sistema mais rápido, dadas as imagens menos complexas e mais otimizadas.

O maior desafio no final do projeto foi adequar uma implementação de Raciocínio Baseado em Casos (RBC) dentro da necessidade da gerência de diálogo. Após diversos estudos e testes, convergimos para a plataforma jColibri, uma implementação Java para RBC.

Mesmo com a otimização de módulos e gráficos, o resultado da composição entre reconhecimento de voz, raciocínio bayesiano, RBC e síntese de voz foi um sistema muito lento. Desta forma, optou-se por disponibilizar o Città ainda sem a Maga Vitta para download no SourceForge (<http://sourceforge.net/projects/civitas/>), embora os fontes trabalhados na Maga possam ser obtidos através do recurso SVN no link citado. A versão atual do Città pode ser executada em sistemas Windows de 64 e 32 bits.

No segundo semestre de 2011, embora tendo feitas diversas tentativas de agendamento com as escolas, não foi possível compatibilizar a disponibilidade entre membros do LELIC e liberação de professoras para realização de ensaios com o Città. Por outro lado, a notícia da aprovação do projeto *Città Cosmopolita* dentro do edital MCT/CNPq Nº 75/2010 - RHA E Pesquisador na Empresa permitiu a manutenção da atual equipe de desenvolvimento. A única mudança será a saída de Leonivan, que parte para ser consultor de projetos em Tecnologia da Informação, e em seu lugar entra **Maribel Susane Selli** (bolsa SET), para aprofundar o relacionamento com as escolas e ser um contato permanente do projeto para realimentação da concepção dos diálogos da assistente inteligente e testes do Città dentro do Projeto Civitas.

Embora com limitações de uso prático, considera-se que o Projeto Assistiva obteve sucesso em seu desenvolvimento, uma vez que permitiu a construção de um assistente inteligente que monitora as variáveis de ambiente de um jogo educacional, e proporciona questionamentos acerca da forma pela qual o usuário estudante está conduzindo a construção de sua cidade. Além da formação da equipe de ex-bolsistas que passou por nossas mãos, fizemos as seguintes publicações:

- Trabalhos apresentados em eventos:

KONZEN, A. A. ; BRAITBACK, O. L. B.; KIST, L. ; ANJOS, A. ; MORAES, L. D.; LIMA, C. I. L.; MULLER, D. N.; AXT, M. . Maga Vitta: agente conversacional aplicado ao jogo educacional Città. In: Simpósio Brasileiro de Informática na Educação, 2011, Aracajú. Simpósio Brasileiro de Informática na Educação - 22 SBIE/17WIE, 2011. ISSN: 2176-4301.

PIRES, T. ; OLIVEIRA, O. L. B. de ; KIST, L. ; ANJOS, A. R. ; MORAES, L. D. ; SILVEIRA, P. D. ; FREITAS, C. I. L. ; MÜLLER, D. N. . Maga Vitta - um assistente inteligente para jogo educacional. In: 12o. Workshop de Software Livre (WSL 2011), 2011, Porto Alegre. Anais do 12o. Workshop de Software Livre, 2011.

- Capítulos de livro:

Lynn Alves e Tânia Maria Hetkowski (orgs). TECNOLOGIAS DIGITAIS E EDUCAÇÃO: novas (re)configurações espaciais, sociais e educacionais.

- Simulador de Cidades Città. Otto Lopes Braitback de Oliveira, Taiuã Pires e Daniel Nehme Müller.
- Agentes Pedagógicos Conversacionais: de traços de personalidade a emoções e conversações. Andréa Aparecida Konzen, Margarete Axt e Patricia Augustin Jaques.
- Personagens-Bruxas como Agentes numa Cidade Imaginária. Margarete Axt, Paloma Dias Silveira e Solange Inês Dhiel Stülp.

- Trabalhos derivados:

KONZEN, Andréa A. ; JAQUES, Patricia ; AXT, M. . Modelagem de um agente pedagógico emocional levando em consideração estratégias afetivas e cognitivas. In: Workshop Affective Computing - Simpósio Brasileiro de Informática na Educação, 2011, Aracajú. Simpósio Brasileiro de Informática na Educação - 22SBIE/17WIE, 2011.

OLIVEIRA, Otto Lopes Braitback de. Città: Cidades Virtuais na Educação. Trabalho de Conclusão do Curso de Jogos Digitais. Unisinos, 2011.

- Palestras:

MÜLLER, Daniel Nehme. Conversas com o computador: quem quer uma interface de voz? IV Jornada de Atualização em Computação (IV JAC). Departamento de Informática da Universidade de Santa Cruz. Santa Cruz do Sul, 3 de setembro de 2010.

ANJOS, Alexsandro R.; KIST, Lisardo; MORAES, Leonivan D.; MÜLLER, Daniel N. Assistiva: assistente inteligente vocal. Semana Acadêmica do Instituto de Informática da UFRGS. Porto Alegre, 25 de maio de 2011.

- Código fonte em software livre:

<http://sourceforge.net/projects/civitas/>

Nas páginas seguintes encontram-se os relatórios individuais de todos os bolsistas que passaram pelo projeto (Luciano pg. 7, Vinícius pg. 15, Lisardo pg. 18, Otto pg. 24, Taiuã pg. 30, Douglas pg. 34, Leonivan pg. 38, e Alexsandro pg. 52), a introdução do relatório de andamento (pg. 65), o plano de trabalho do período de prorrogação (pg. 69), o último artigo publicado nos anais do Simpósio Brasileiro de Informática na Educação 2011 (pg. 71), e o documento de projeto de jogo (pg. 81).

Relatório de Projeto:

ASSISTIVA – Assistente Inteligente Vocal para Maga Vitta

Projeto processo 580470/2008-3

Criado por: *Mércio Luciano Teixeira Soares*

Bolsista processo: 182713/2011-3

Período do relatório: *agosto/2011 a novembro/2011*

Apresentação do Città

O Assistiva é um projeto com foco na implementação pela Conexum do assistente inteligente Maga Vitta, voltado à assistência do jogo educacional Città no Projeto Civitas, desenvolvido pelo Laboratório de Estudos em Linguagem, Interação e Cognição ([LELIC](#)), da Faculdade de Educação da UFRGS.

O jogo Città é voltado a estudantes entre 8 e 10 anos de idade para a construção de cidades virtuais. O assistente tem a finalidade de interagir com os usuários provocando desafios e trocando informações na forma de diálogo.

Além da interação por texto, também estamos desenvolvendo o diálogo por voz, que possibilitará uma nova dinâmica na comunicação do assistente com os estudantes.

Objetivos do projeto

O objetivo principal é que o usuário aprenda através da construção e da organização de uma cidade. Para que o mesmo passe por diversas situações que proporcionem este aprendizado ele terá que completar tarefas fornecidas pelos educadores ou pelo próprio jogo. Nessas tarefas os alunos terão que satisfazer a população e ter uma sustentabilidade que respeite o meio ambiente.

Proposta inovadora

Existem inúmeros programas de computador com propostas semelhantes. Porém, a grande maioria é produzida por grandes corporações de *software* estrangeiras, com objetivos puramente comerciais. Estes jogos frequentemente estimulam a competitividade e, às vezes, induzem o participante a exercer um capitalismo desenfreado para “superar” e “vencer” seus oponentes.

É fácil constatar que este tipo de produto pode ser indicado para divertir seus usuários por algumas horas, mas não para formar um senso de cidadania e ecologia que dure a vida inteira. Eis a principal diferença entre o Città e seus assemelhados disponíveis comercialmente: o nosso programa é capaz de entreter com uma proposta lúdica enquanto mostra para o usuário noções sobre a sociedade em que ele vive e o meio-ambiente que ele, como cidadão, deve desde criança aprender a proteger.

Outro recurso inovador é a agente Maga Vitta. Que se comunicará por voz com o jogador. Através de um microfone o jogador poderá também falar com ela.

Meu trabalho

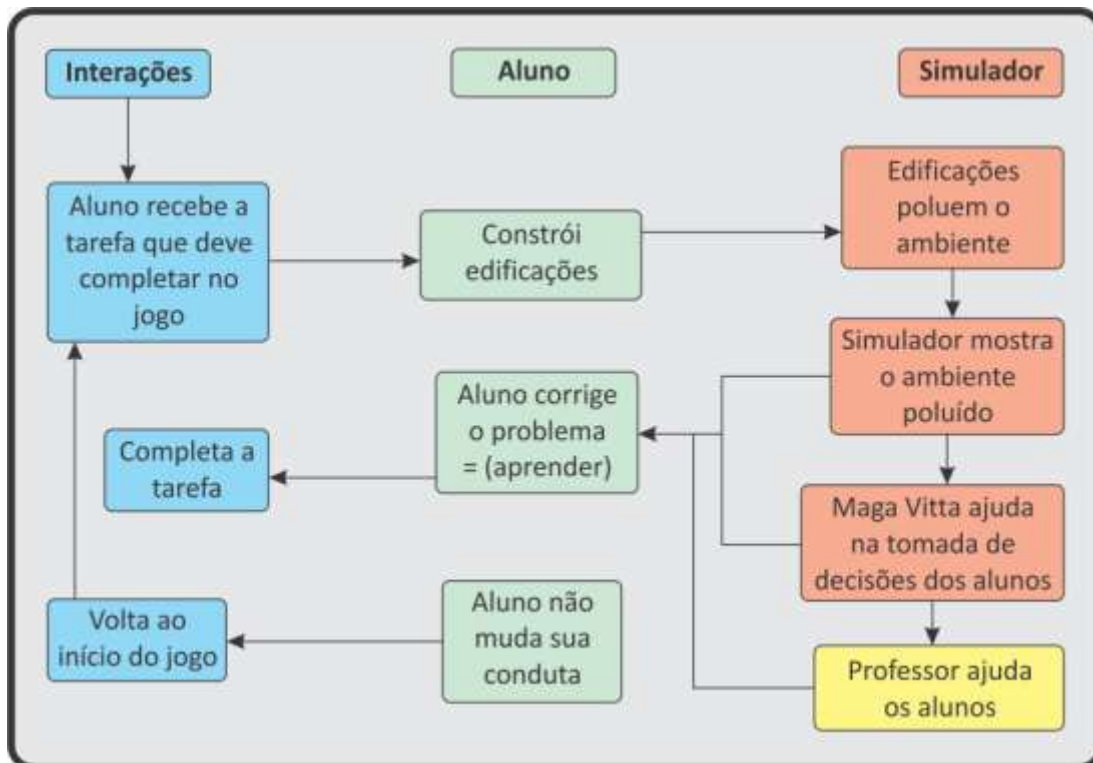
Eu fiquei incumbido de dar uma nova versão aos objetos de construção e inseri-los no menu do jogo, inicialmente. Mas aproveitei para fazer uma mudança total em toda a parte

gráfica e até o momento tenho tido êxito. Infelizmente o sistema de som ambiente ainda não está pronto para testes, mas agarrei essa ideia também e ainda estudo como aplicar melhor no jogo. Pois um jogo sem som fica incompleto.

Gameplay

O aluno receberá do instrutor ou de missões do próprio jogo, tarefas a serem completadas. Por exemplo, uma missão do jogo:

Construir 20 indústrias, pois há a necessidade crescimento da região. Logo o aluno terá que construir casas para os habitantes trabalharem nas fábricas. Habitantes precisam de alimentos, saúde, energia elétrica, transporte, educação, etc. Logo mais habitantes são necessários para preencher estes campos de trabalho, e todo este sistema polui e afeta o meio ambiente. O aluno então precisa gerenciar tais recursos para pode completar o objetivo. A figura a seguir descreve as interações entre aluno, professor e simulador.



Um pequeno esquema das interações entre aluno, Città, Maga Vitta e os professores.

Atualizações

O jogo passou por diversas atualizações, mas nesses últimos quatro meses de projeto elas foram realmente “gritantes”. Primeiramente, o código foi reescrito quase que completamente, como é um trabalho feito por muitos, é inevitável algum tipo de desorganização em dado momento, mas isso foi totalmente corrigido pelo Vinicius com uma parcela de ajuda dos outros integrantes do projeto.

Quanto à interface, me dispus a modificá-la por completo – o jogo merecia isso –, então arregacei as mangas e com muitas opiniões e ideias distintas organizei tudo e o Città está de cara nova, inclusive a Maga Vitta.

Interface remodelada!

Maga Vitta

A Maga Vitta ganhou uma atenção especial, pois precisávamos de uma figura amigável, que fosse interessante para crianças (por isso o estilo de desenho *mangá*), mas que não fosse infantil demais, porque ela é o assistente que deve auxiliar as crianças. Então foi a primeira a passar pelo processo de mudança ganhando um aspecto jovem e moderno com diferentes cores de cabelo de acordo com a reação das respostas dos alunos e/ou como está o andamento do jogo (níveis de poluição e ordem ou desordem da cidade).



Maga Vitta em suas versões de humor

Menu principal do jogo

O menu inicial tem agora a sensação de “chegada” ao Città: quem nunca desenhou uma estrada sinuosa com uma cidade ao fundo?

Esse foi o ponto de partida para a ilustração de fundo. Os botões em tons de cinza ficam coloridos e são animados quando se passa o mouse por cima, dando um divertido começo ao jogo.



Menu inicial sem os botões



Menu com os botões carregados

Objetos (construções)

CASA



PRÉDIO



ESCOLA



MERCADO



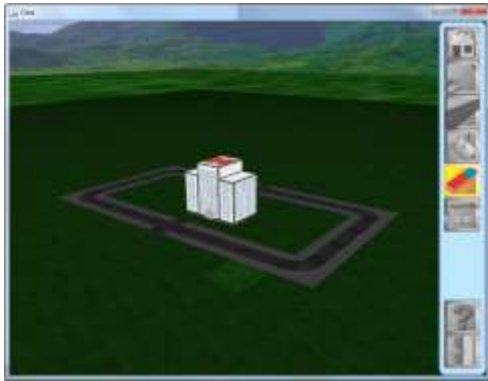
PREFEITURA



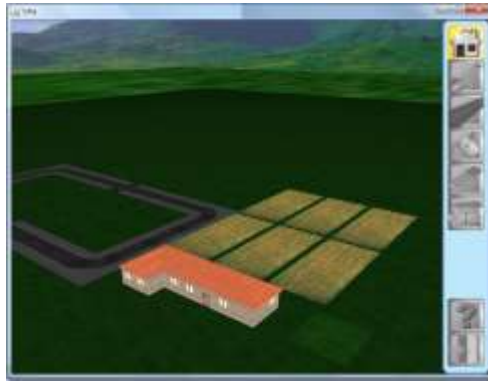
IGREJA



HOSPITAL



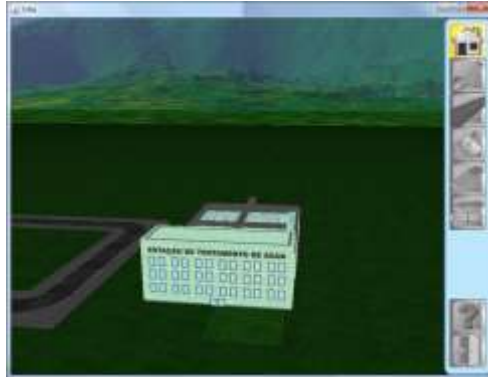
FAZENDA



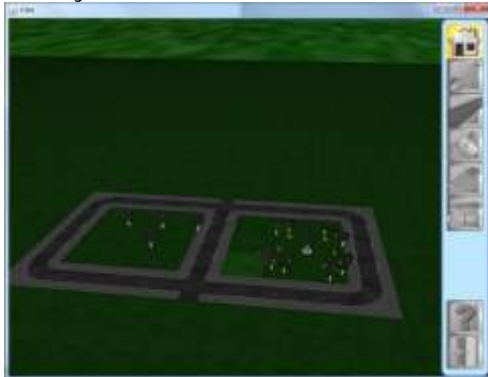
ETE



ETA



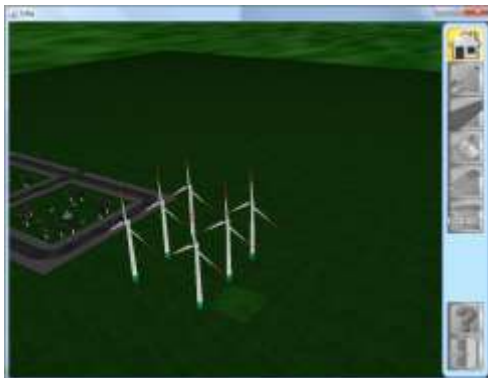
PRAÇA



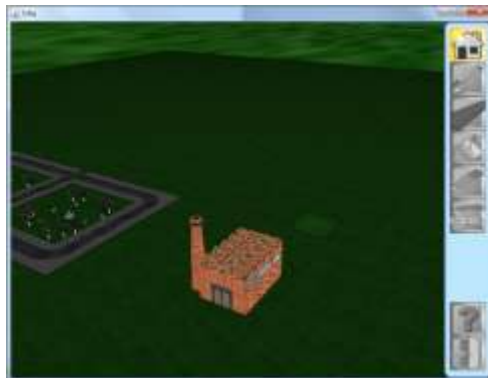
ÁRVORE



USINA EÓLICA



FÁBRICA



Todos foram modelados com 3D Studio Max por mim

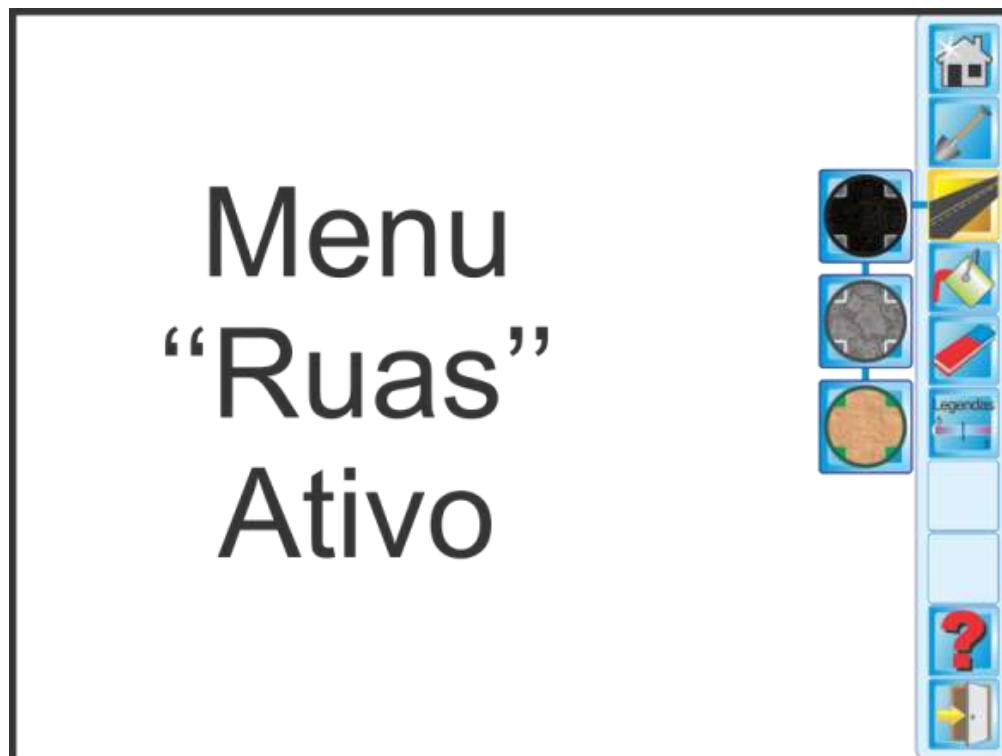
Menus do jogo



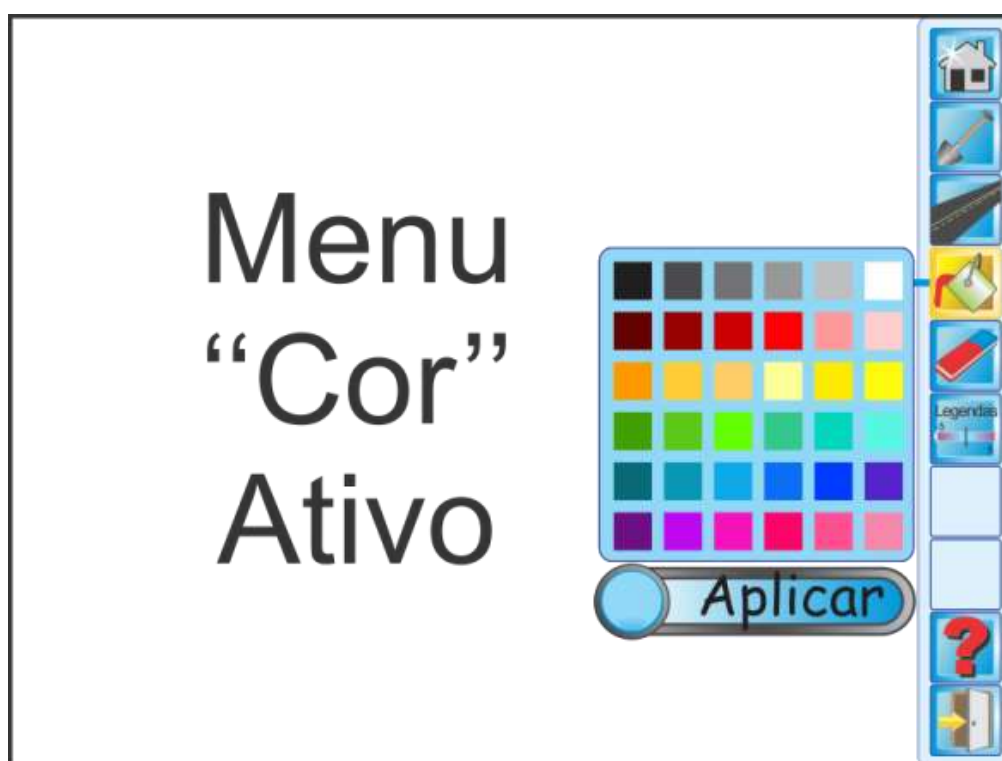
Menu de criação: onde ficam os objetos do jogo que poderão ser construídos



Menu de terreno: onde pode-se alterar propriedades do terreno como elevações e inserções de lagos, areia, grama, etc.



Menu de ruas: onde são criadas ruas e calçadas



Menu de cor: aqui são selecionadas as cores para alteração de algumas construções

Novidades na jogabilidade

Menu: agora está menor e mais intuitivo, na lateral direita da tela em transparência. Assim, ocupa menos espaço de jogo e abre em cascata de acordo com a seleção dos botões principais.

Borracha: por escolha das crianças, uma borracha tem muito mais a ver com deletar um objeto do que a antiga escavadeira que não tinha nenhum sentido para elas.

Legendas: as legendas ganharam um botão dedicado, assim o usuário poderá jogar consultando periodicamente quais as condições gerais da cidade tendo a possibilidade de resolver qualquer problema com antecedência.

Maga Vitta: ela agora está no menu, normalmente inativo, o botão muda de cor quando a Maga precisa comunicar algo, como avisar que algo não está certo ou dar alguma dica de jogo.

Salvamento automático: o jogo tem salvamento automático, já que o usuário digita seu *login* e nome da cidade antes de construí-la ou simplesmente a carrega. Praticamente descartando a possibilidade de perda de dados por falta de energia ou pane de computador, por exemplo.

Relatório de Projeto:

ASSISTIVA – Assistente Inteligente Vocal para Maga Vitta

Projeto processo 580470/2008-3

Criado por: *Vinícius Brum Bertoldo*

Bolsista processo: 182083/2011-0

Período do relatório: *junho/2011 a novembro/2011*

Inicialmente foi estabelecido um período de adaptação, durante o qual houve a familiarização com o projeto existente e os recursos utilizados. Após uma breve familiarização com a biblioteca JOGL, foi possível a correção de pequenos bugs e inconsistências na execução do jogo e sua interface, bem como outros pequenos bugs relacionados à lógica de jogo, bancos de dados e a gravação de mapas. Devido a não total familiarização com o JOGL, algumas falhas não foram corrigidas, principalmente devido a outras prioridades como a integração dos diversos módulos ao jogo.

Integração dos módulos de chat, reconhecimento de voz e inferência

Chat: A integração do chat foi feita através de um botão inserido na interface do jogo, que deve interagir junto ao módulo de inferência para chamar a atenção do jogador durante a edição das cidades. O chat por sua vez já era parcialmente integrado com o módulo de inferência, mas ainda não possuía conexão com o módulo de reconhecimento de voz.

Reconhecimento de voz: A integração do módulo de reconhecimento de voz ocorreu parcialmente através da inserção de botões de controle de gravação de voz ("Começar gravação" e "Parar gravação") na interface do chat. No entanto, o módulo de reconhecimento de voz não está funcional em conjunto com os outros módulos. Foi cogitado o uso de uma ferramenta semelhante ao grep(GNU Project) para buscar as palavras-chave capturadas pelo Julius e repassá-las ao assistente do jogo.

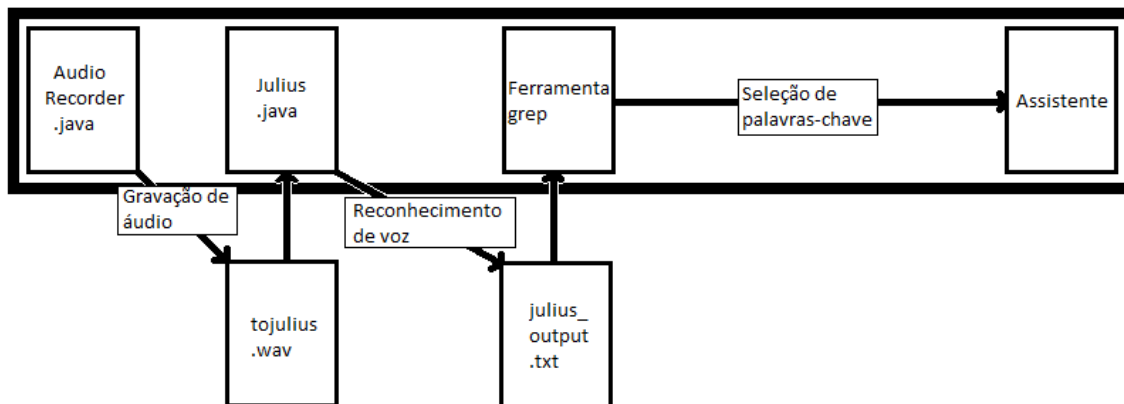


Figura 1 – Diagrama de reconhecimento de voz

Inferência: A integração do módulo de inferência foi realizada por apenas disponibilizar um método de sinalização do módulo de inferência para a interface. Ao detectar falta ou excesso de alguma variável de ambiente, bem como condição normal, a inferência ativa ou desativa a possibilidade de a Maga Vitta interagir com o jogador. A inferência já havia sido integrada com as variáveis de ambiente que são modificadas pela lógica do jogo.

Revisão geral da interface gráfica

Com base nos relatos de experiências com as crianças das escolas nas quais o Città foi apresentado, percebeu-se que havia necessidade de melhorar a interface gráfica primeiramente para facilitar a compreensão. Após diversas discussões ao longo do tempo, foi decidido que além de simplificar a interface, haveria necessidade também de uma revisão e renovação geral da interface gráfica. Foi adotada a ideia de tornar a navegação dos menus clara através de imagens apenas, já que palavras muitas vezes complicadas não eram compreendidas pelas crianças, o que restringia às crianças algumas opções de maneira inconsciente.



Figura 2 – Comparação das interfaces; À esquerda, interface antiga, à direita, interface nova

Ao implementar a nova interface, notou-se as limitações que a falta de conhecimento sobre a engine gráfica e bibliotecas não recentemente atualizadas iriam implicar em futuras atualizações mais complexas como animações. Foi considerada a utilização de uma engine

completamente diferente, como a Unity, mas essa idéia foi abandonada devido ao curto tempo de desenvolvimento restante. Para prosseguimento do projeto é extremamente preferível trocar para uma engine mais atual.

Novo modelo de banco de dados e gravação de mapas

O método de gravação de mapas utilizado previamente não se aproveitava devidamente das vantagens que um banco de dados relacional pode oferecer em termos de velocidade e espaço. Foi discutido um método mais otimizado de gravação de mapas, que também facilitaria a implementação de um histórico de eventos também mais otimizado.

Este novo método consistiria em um arquivo de banco de dados para cada mapa, contendo um registro do estado atual do mapa e um registro de eventos.

O registro atual do mapa serve para uma rápida reconstrução do mundo de jogo, independente de quantas alterações foram feitas até agora naquele mapa.

O registro de eventos grava a “história” daquele mapa, indicando a data de jogo, o tipo do evento(construção/destruição/“biografia da construção”) e um texto, descrevendo o evento em detalhes.

Devido ao término do prazo do projeto, não houve uma implementação prática do modelo, mas independente do ambiente e engine usados no projeto, essa foi a solução mais otimizada encontrada após discussão.

Foi alterado também o critério de salvamento dos mapas. Os mapas são salvos assim que o jogador decide sair do mapa atual, não há necessidade de navegar menus ou apertar outros botões.

Limpeza do código

Durante todo o processo de desenvolvimento foram realizadas diversas ações como clarificação de códigos ofuscados, adição de comentários e exclusão de trechos de código não utilizados.

Relatório de Projeto:

ASSISTIVA – Assistente Inteligente Vocal para Maga Vitta

Projeto processo 580470/2008-3

Criado por: *Lisardo Sallaberry Kist*

Bolsista processo: 181839/2010-5

Período do relatório: *julho/2010 a abril/2011*

Componente de Reconhecimento de Voz

1 Módulo

O módulo consiste em um transcritor de voz para texto, que será enviado para o assistente do jogo. Para realizar a transcrição é utilizado o software open-source [Julius](#).

O módulo funciona recebendo um *stream* de áudio (arquivo ou sinal do computador) e retorna a transcrição na forma de stream de texto. O programa presume um modelo acústico e um modelo linguístico sobre qual o reconhecimento será realizado.

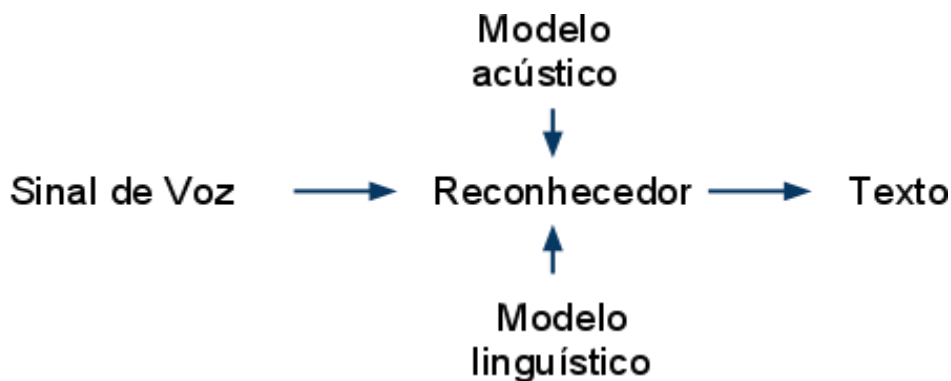


Diagrama 1: Esquema funcional do módulo de reconhecimento.

Modelo acústico trata-se de um modelo de cadeias ocultas de markov sobre qual o reconhecimento é feito. O modelo linguístico é uma gramática formal especificada para aplicação. Texto reconhecido poderá ser acessado como variável do programa e utilizado na assistência ao jogador.

2 Desenvolvimento

2.1 Projeto e API Coruja

Nos meses iniciais foi feito o projeto do módulo de reconhecimento. Há dois grandes projetos de código aberto para reconhecimento de voz hoje em dia: *Sphinx* e o *Julius*.

A decisão do projeto foi, inicialmente, usar a API brasileira Coruja, baseada no Julius. Trata-se de um projeto do [Falabrasil](#), do Laboratório de Processamento de Sinais, que permitiria o trabalhar com o reconhecimento na forma de API.

Porém, houve muitas dificuldades para conseguir compilar o projeto e obter os primeiros resultados. Tanto a versão para Linux do programa quanto os modelos acústicos eram para 32-bits, ao passo que nosso ambiente de desenvolvimento era de 64-bits.

Depois de muitas horas investidas, finalmente conseguiu-se migrar a API para Linux. Foi escrito um pequeno tutorial de como realizar a compilação depois de tudo pronto. Ele foi enviado para a lista de e-mail do Coruja para os usuários que estavam tendo problemas semelhantes.

Como o Coruja ainda estava em desenvolvimento e a documentação era parca, percebeu-se que demoraria muito para ter o reconhecimento funcionando. Acabou-se preferindo usar diretamente o Julius para realizar o reconhecimento. Para tanto, foi necessário ler o manual do Julius (Juliusbook) se inteirar dos métodos e parâmetros do reconhecimento de voz.

Apesar do reconhecimento de voz estar bastante desenvolvido para o inglês, houve dificuldade em obter-se modelos acústicos para português. Os primeiros testes foram realizados com o modelo acústico e linguístico do próprio LAPS (versão 1.3, na época), mas o reconhecimento era bastante ruim. Foi tentado, então, treinar o primeiro modelo acústico próprio para português a partir das gravações que a Conexum tinha disponível.

2.2 Treinamento do modelo acústico

Precisa-se de uma grande quantidade de amostras de voz e suas transcrições para treinamento do modelo acústico. Foi feito um curso de 20h “Introdução à Análise Acústica com Praat”, com professor Ubiratã Kickhofel Alves, e gasto aproximadamente um mês realizando transcrições acústicas.

Para o treinamento utilizou-se o software HTK. Novamente, a utilização do software não era nem um pouco trivial. Foi necessário buscar conhecimento no livro-manual do HTK (HTKbook) e no site da universidade.

No momento do treinamento, havia transcrições que estavam fora de padrão. Ademais, haviam sido feitas as transcrições dos fonemas, ao passo que o programa trabalhava com conjuntos de trifones. Foi necessário escrever um programa para agrupar em trifones e corrigir as transcrições erradas manualmente. O script de agrupamento de monofones em trifones atualmente está na wiki da empresa para uso futuro.

O treinamento do primeiro modelo acústico da Conexum terminou por meados de dezembro. Infelizmente, ele retornava apenas reconhecimentos vazios. Depois de mais pesquisa, acreditasse que o motivo do erro tenha sido a mensagem de aviso que ocorreu durante o treinamento: “WARNING [-2331] UpdateModels: o-j+s[76] copied: only 0 egs”. Ela ocorre que quando algum tipo de trifone foi mapeado muito poucas vezes nas amostras, e no nosso treinamento ele ocorreu para muitos trifones. É plausível de se pensar que esse erro seria eliminado se aumentasse as amostras em quantidade e diversidade.

2.3 Montagem de gramática para modelo linguístico

A última tentativa de melhorar o reconhecimento foi a montagem de um modelo linguístico. Foram montados 4 modelos gramaticais de exemplo, sobre os quais o reconhecimento foi testado. Houve um aumento substancial com o uso de gramáticas fechadas. Porém, imagina-se que problemas poderiam decorrer do uso de uma gramática muito fechada. Cite-se que foi feita uma visita a escolas municipais em Mato Leitão, Rio Grande do Sul, em Novembro (2010). O módulo de assistência não estava pronto na época, mas foi uma visita importante para verificar a usabilidade do programa pelos estudantes. Foi observado *in loco* como é complicada a interação quando usuários são crianças em ambiente escolar. Para o uso de uma gramática estrita seria necessário “guiar” as crianças em como buscar por ajuda, o que não é interessante quando se busca fazer um jogo para aprendizagem.

2.4 Integração com o resto do jogo.

Como Julius é escrito em C e Assistiva em Java, foi criado um pacote ReconhecimentoVoz em Java para integração dos dois softwares. O módulo encaminha a entrada de sinal acústico para o Julius e captura suas respostas como variáveis do programa em Java.

2.5 Passos futuros

O próximo passo do desenvolvimento é melhorar a taxa de acerto do reconhecimento de voz. Isso seria possível tornando modelo linguístico mais estrito. Corre-se o risco, porém, de limitar demasiadamente o escopo da interação do estudante.

Outra via para aumentar a capacidade de reconhecimento é treinar um modelo acústico própria para nossa aplicação, já que o modelo atual não foi treinado especificamente para o Civitas. Foram até mesmo coletadas amostras de voz em ambiente escolar. Contudo, era uma amostragem de gravações muito pequena para o porte necessário para um modelo acústico. Com um número maior de amostras seria possível treinar um modelo acústico para melhorar a capacidade de reconhecimento.

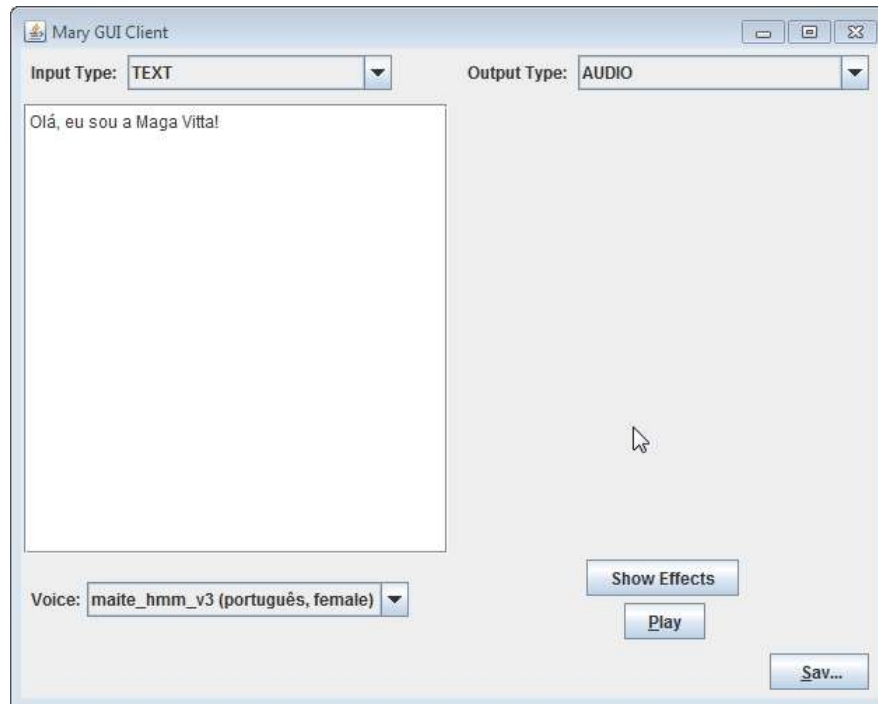
Componente de Síntese de Fala

1 Servidor de síntese

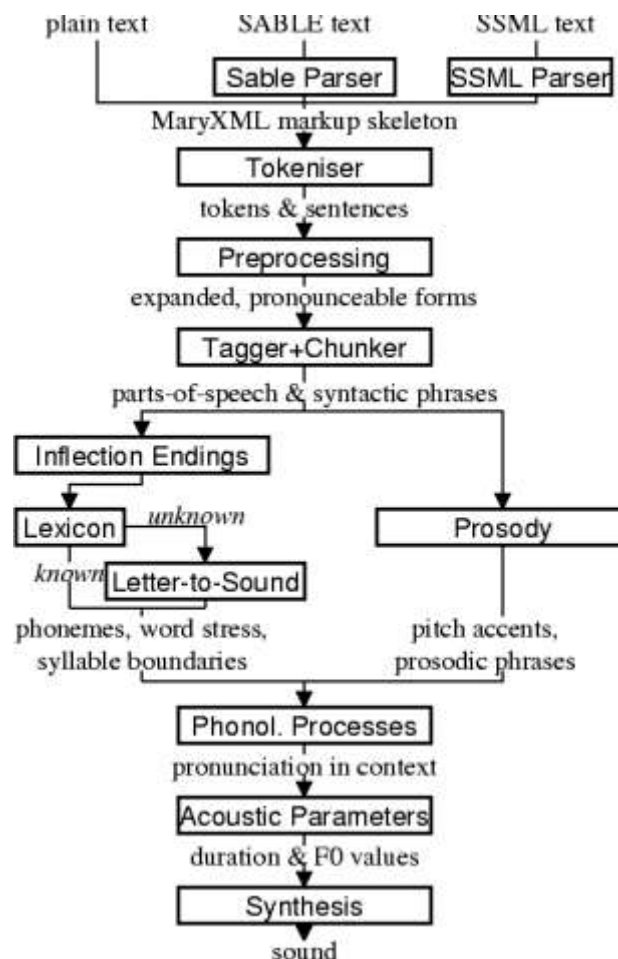
O sistema responsável atualmente pela síntese da fala da Maga Vitta é o software livre MaryTTS. Ele utiliza um padrão de serviço internet (WebService) para receber um texto e enviar um arquivo de áudio.

Para os testes com o Città, foram copiados os arquivos do MaryTTS para Windows para a mesma pasta do jogo. Ao iniciar o Città é necessário ativar o servidor (*maryserver*) e, quando necessário, apenas solicitar o arquivo de áudio para determinada frase. Para tanto, foi adicionada uma chamada de sistema para o servidor, ao início do Città, e uma classe Java (*MaryClientUser*) para a síntese, quando esta for solicitada.

Testes realizados apenas com a classe cliente (*MaryClientUser*) possibilitou a geração de uma janela para testes, como mostra a figura a seguir.



O sistema MaryTTS como um todo é uma composição de módulos, como mostra a figura a seguir, retirada da página <http://mary.dfki.de/documentation/module-architecture> :



Como se pode perceber, MaryTTS é uma solução completa e configurável para síntese, incluindo a construção do modelo acústico para voz, o léxico de fonemas e a parametrização de itens como a prosódia (entonação) e velocidade de leitura (duração).

Os modelos acústicos de síntese para o Português que vêm com o MaryTTS são primitivos, proporcionando uma voz visivelmente mecânica e de difícil audição. Neste sentido contamos com apoio da equipe da Conexum, que proporcionou a gravação de mais de 600 frases (locução feminina) e geração de um léxico de 700 palavras com sua anotação fonética para treinamento do modelo acústico. Este modelo proporcionou uma síntese com boa audição, embora ainda possa melhorar, através do ajuste de parâmetros de saída do MaryTTS.

2 Considerações práticas

Os testes realizados mostraram que a funcionalidade do uso do Città com MaryTTS é adequada aos propósitos do projeto, porém será necessária a extração apenas do código Java correspondente, uma vez que a solução como servidor torna o sistema mais lento como um todo. O fato de ter um serviço sendo executado em paralelo ao jogo deixa o sistema mais lento e consome mais recursos do computador. A execução através apenas das classes de síntese seria a saída mais adequada ao acomplamento do sistema de síntese no projeto do Città.

3 Referências

Juliusbook: <http://sourceforge.jp/projects/julius/downloads/47534/Juliusbook-4.1.5.pdf/>

Laboratório de Processamento de Sinais da UFPA: <http://www.laps.ufpa.br/>

HTK: <http://htk.eng.cam.ac.uk>

Coruja Users group: <http://groups.google.com/group/coruja-users?hl=pt-BR/>

MaryTTS: <http://mary.dfki.de/>

Relatório de Projeto:

ASSISTIVA – Assistente Inteligente Vocal para Maga Vitta

Projeto processo 580470/2008-3

Criado por: *Otto Lopes Braitback de Oliveira*

Bolsista processo: 181730/2009-0

Período do relatório: *junho/2009 a março/2010*

1. Introdução

Neste documento estão relatadas as atividades que exerci como bolsista do CNPQ no período de julho de 2008 a março de 2010. Durante este tempo, trabalhei no projeto de desenvolvimento do simulador de cidades (*Città*), parte do projeto ASSISTIVA. O projeto ASSISTIVA é realizado por bolsistas e pesquisadores do Laboratório Conexum, localizado no Centro de Empreendimentos – Instituto de Informática da Universidade Federal do Rio Grande do Sul, sob a orientação do Dr. Daniel Nehme Müller .

1.1 Visão do Projeto

Desenvolvimento de um software gráfico interativo (jogo) para apoiar pedagogicamente na construção de conhecimento das crianças de 3ª e 4ª séries do ensino fundamental, no que se refere ao desenvolvimento de capacidades que lhes permitam pensar - construir - organizar uma cidade virtual a partir da realidade infantil e seus conhecimentos incorporados através dos conteúdos oferecidos no aprendizado dos sistemas que regulam uma cidade e suas relações.

2. Desenvolvimento

O *Città* começou a ser desenvolvido no projeto CIVITAS, no qual participei e encerrou em junho de 2009. Logo o jogo já possuía um protótipo funcional, mas ainda possuía erros e melhoramentos a serem feitos. Também há a necessidade de criar interfaces para o módulo do agente inteligente e para o módulo de voz.

São utilizadas um conjunto de bibliotecas para facilitar e agilizar o desenvolvimento do jogo, estas são descritas abaixo:

Nome	Descrição
OpenGL Graphic Engine API	Projeto Open Source sobre a licença GNU, inclui suporte a OpenGL, funções matemáticas, leitor de texturas, fontes, sistema de picking, etc
SQLite	Biblioteca para gerenciar bancos de dados, como por exemplo o mapa do jogo
FMOD	Gerenciar sons do jogo
Ekit	Editor de textos, para escrever as histórias dos objetos

Tabela 1 Bibliotecas Utilizadas

Assim como também são utilizadas ferramentas para desenvolver o jogo, estas são descritas abaixo:

Nome	Descrição
Tortoise SVN	Usado para controlar versões do software
GIMP	Editor de imagens
Milkshape	Modelador 3D, utilizado para converter o modelo 3D do objeto no formato .OBJ
Eclipse	IDE utilizada no desenvolvimento
Planner	Gerenciamento de tarefas
Blender	Modelador 3D, utilizado para a criação do modelo 3D
Star UML	Criação de diagramas de Classes, para melhor entendimento do projeto

Tabela 2 Softwares utilizados

2.1 Versão 0.6.0

Após a primeira reunião do projeto ficou decidido que iria continuar as tarefas do projeto anterior (CIVITAS), e começar a fornecer meios para a criação do agente.

Número	Tarefa
1	Iluminação do terreno

2	Suavização do terreno
3	Adicionar variáveis de poluição as classes dos objetos
4	Incluir novos objetos (ETA, ETE)
5	Criação de um temporizador
6	Salvar e Carregar mapas através do SQLite
7	Adição das texturas das barras de poluição

Tabela 3 Descrição das Tarefas da versão 0.6.0

As tarefas foram definidas por prioridade e levando em consideração as dependências entre elas, abaixo segue a descrição detalhada de cada uma.

1. Sem a iluminação do terreno ficava extremamente difícil notar o relevo, para criar iluminação na OpenGL é necessário especificar as normais do plano, deste modo é possível fazer o calculo e saber o quão iluminado esta o plano, o código abaixo e a exemplifica o processo.

```
//Calcula a normal
```

```
Vector3f normal = getNormalVec(new Vector3f(terrain[x][z][0],
terrain[x][z][1], terrain[x][z][2]), new Vector3f(terrain[x + 1][z][0],
terrain[x + 1][z][1], terrain[x + 1][z][2]), new Vector3f(terrain[x][z + 1][0],
terrain[x][z + 1][1], terrain[x][z + 1][2]));
```

2. No Mappa, editor de terrenos, havia duas opções de alterar o relevo: inclinar ou declinar a altura. Deste modo não era possível construir alguma edificação em um relevo modificado. Era necessário criar uma opção de suavizar o terreno na interface gráfica e implementar uma função que calculasse a média entre 9 tiles.



Figura 1 Iluminação do terreno e a direita, terreno inclinado a esquerda com suavização

3. Após uma reunião com o Leonivan, bolsista responsável por implementar o agente inteligente, verificamos que era necessário atualizar as variáveis existentes e adicionar algumas outras. Assim como criar uma escala para o jogo, após o Leonivan realizar esta pesquisa e passar para mim estes valores adicionei a cada objeto do jogo.
4. Esta tarefa consistiu em colocar no jogo os modelos 3D feitos pelo designer, criar classes para os objetos e fazer os devidos tratamentos de construção e exclusão.
5. Para controlar o tempo de simulação da cidade era necessário criar um temporizador, que contasse os dias passados na cidade conforme os segundos. Por exemplo 1 dia na cidade é igual a 60 segundos.

```
//Pega o tempo do computador  
  
tempoMilis = String.valueOf(System.currentTimeMillis());  
  
//Pega o caráter na 9ª posição = segundos  
  
second = tempoMilis.charAt(9);  
  
//Se for diferente do segundo anterior
```

6. O arquivo do mapa do jogo, que tinha todas as informações da cidade, era criado de modo simples como um arquivo de texto. Logo ficava desorganizado para salvar em um mesmo arquivo dados de poluição, de objetos e do cenário, além de ocupar mais espaço. Deste modo optou-se em utilizar uma biblioteca de banco de dados a SQLite, que podia agrupar os 3 tipos de dados do mapa com a metade do espaço usado antes.
7. Para melhor controle de poluição do ambiente foram criadas barras de poluição como mostra a Figura 2. Deste modo o aluno pode facilmente identificar o quão poluído está a cidade.
- 8.



Figura 2 Barra de Poluição

2.2 Versão 0.6.3

Após a versão 0.6.0 estar pronta notou-se que era necessário depurar melhor as variáveis do ambiente do jogo, assim como era preciso atualizar a biblioteca gráfica fornecida pelo Jouvieje para uma versão mais nova. Conseqüentemente, implicaria em grandes mudanças no código do jogo, deste modo aproveitei para organizar o código para cada classe ter sua responsabilidade, seguindo o modelo MVC (model-viewer-controller).

Número	Tarefa
1	Calcular e adicionar valores de poluição quando construir objetos
2	Mostrar variáveis de poluição no HUD
3	Opção de “Resetar” mapa em modo de testes
4	Opção de criar Novo Mapa
5	Atualizar o motor gráfico
6	Organização de responsabilidade de Classes

Tabela 4 Descrição das Tarefas da Versão 0.6.3

1. Quando um edifício é inserido dentro do mapa do jogo, os valores de poluição deste são automaticamente adicionados à cidade.
2. Para depurar a simulação da cidade os valores de poluição são mostrados em modo texto na tela do jogo ou Head Up Display (HUD).



Figura 3 Valores de poluição

3. Esta opção faz com que as variáveis do ambiente sejam zeradas, deste modo fica mais fácil executar testes com diversos cenários.
4. Consiste em dar a opção ao jogador de iniciar uma cidade do zero, sem nenhuma construção, antes só podia carregar cidades existentes.
5. Esta tarefa foi necessária para poder utilizar os novos recursos disponíveis no motor gráfico. Algumas classes tiveram os nomes trocados “refactoring”, assim como o modo de carregar a aplicação foi alterado.
6. Aproveitando as modificações feitas na tarefa anterior, foi feito uma melhor organização do código em termo de responsabilidade de cada classe. O motor gráfico, automaticamente, cria uma thread com o contexto da OpenGL, deste modo era usado uma classe de desenho (Render) junto com a de lógica do jogo (CittaEngine). Pois qualquer requisição de desenho feito pela CittaEngine, com uma thread normal sem contexto da OpenGL, não era executada pela classe de Render. A solução encontrada foi deixar as duas threads uma para rendering e outra para a lógica do jogo, mas sem chamadas de desenho a partir da classe CittaEngine. Há somente ativações ou desativações de funções de desenho feitos pela CittaEngine.

2.3 Versão 0.7.0

Número	Tarefa
1	Excluir construções quando “R” pressionado
2	Mudar Layout de informações de poluição Poluidores ---- Despoluidores Resultados
3	Organização do projeto no SVN

Tabela 5 Tarefas da versão 0.7.0 ordenada pela prioridade.

1. Na versão anterior somente os valores eram redefinidos as construções continuavam no terreno, o que dificultava a depuração conforme fossem adicionados novos prédios.
2. Para facilitar a visualização das variáveis que poluem o ambiente e das que despoluem, assim como o resultado dos 2. O layout foi alterado como mostra a Figura 3.
3. Tarefa que consistiu em limpar e organizar os arquivos na pasta do *Source Forge*, em arquivos essenciais para o jogo e arquivos a serem adicionados ao jogo. Desta forma a cada “Update” no servidor não há problemas de rodar o jogo, pois o designer modificará apenas a pasta de arquivos a serem adicionados.

Bibliografia

Fox, B. (2005). *Game interface design*. Boston: Thomson.

Gold Standard Group. (2008). *OpenGL - The Industry Standard for High Performance Graphics*. Acesso em 10 de 08 de 2008, disponível em <http://www.opengl.org/>

Jouvieje, J. (2009). Acesso em 20 de 03 de 2009, disponível em Tutorial 25 : OpenGL components: <http://jerome.jouvie.free.fr/OpenGL/Tutorials/Tutorial25.php#Tutorial>

Jouvieje, J. (2008). *OpenGL Graphic Engine API*. Acesso em 15 de 08 de 2008, disponível em <http://jerome.jouvie.free.fr/OpenGL/Projects/GraphicEngine.php>

Jouvieje, J. (2008). *Tutorial 09 : Blending*. Acesso em 21 de 11 de 2008, disponível em <http://jerome.jouvie.free.fr/OpenGL/Tutorials/Tutorial9.php#Tutorial>

Kistler, H. (2003). Acesso em 22 de 05 de 2009, disponível em Ekit version 1.5: <http://www.hexidec.com/ekit.php>

Lighthouse 3d. (s.d.). *Picking Tutorial*. Acesso em 05 de 09 de 2008, disponível em <http://www.lighthouse3d.com/opengl/picking/>

Mark A. DeLoura, D. T. (2004). *Game programming Gems*. Charles River Media.

Microsoft Corporation. (2008). *Project Home Page*. Acesso em 02 de 09 de 2008, disponível em www.microsoft.com/Project

Microsoft Developer Network. (2008). *HLSL*. Acesso em 05 de 11 de 2008, disponível em [http://msdn.microsoft.com/en-us/library/bb509561\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/bb509561(VS.85).aspx)

NeHe Productions. (2008). *OpenGL Lesson #12:*. Acesso em 30 de 10 de 2008, disponível em <http://nehe.gamedev.net/data/lessons/lesson.asp?lesson=12>

Object Management Group. (2009). *Unified Modeling Language*. Acesso em 20 de 08 de 2008, disponível em <http://www.uml.org/>

Rollings, A., & Morris, D. (2004). *Game architecture and design : a new edition*. Berkeley: New Riders.

SourceForge, Inc. (2009). Acesso em 08 de 01 de 2009, disponível em SourceForge.net: Find and Build Open Source Software: <http://sourceforge.net/>

Sun Developer Network. (2008). *The AWT in 1.0 and 1.1*. Acesso em 11 de 12 de 2008, disponível em <http://java.sun.com/products/jdk/awt/>

Taylor, C. (2008). *Run Away Studios*. Acesso em 25 de 08 de 2008, disponível em http://www.runawaystudios.com/articles/chris_taylor_gdd.asp

Tigris.org. (2008). *TortoiseSVN*. Acesso em 22 de 10 de 2008, disponível em <http://tortoisesvn.tigris.org/>

Wikimedia Foundation. (24 de 05 de 2009). Acesso em 05 de 01 de 2009, disponível em GNU General Public License: http://pt.wikipedia.org/wiki/GNU_General_Public_License

Relatório de Projeto:

ASSISTIVA – Assistente Inteligente Vocal para Maga Vitta

Projeto processo 580470/2008-3

Criado por: *Taiuã Pires*

Bolsista processo: 180877/2010-0

Período do relatório: *abril/2010 a maio/2011*

Interface:

Novo botão no menu principal para criar um novo mapa.

Nova área de informações sobre objetos do mapa em desenvolvimento.

Banco de Dados:

Criado sistema de bancos de dados para armazenar informações de todos os eventos do jogo sobre cada mapa diferente permitindo armazenamento de textos de extensão indeterminada para cada ação feita sobre o mapa do jogo que esta sendo manipulado durante a execução.

O sistema é composto a ter um banco de dados para cada posição do mapa, o mapa é composto de cem células por cem células totalizando dez mil bancos de dados, para os textos de descrição do jogador ter tamanhos arbitrários, sem limite definido, eles precisam de um banco de dados criado durante a execução somente com o texto.

Há a necessidade de tantos bancos de dados porque as construções no jogo tem tamanhos diferentes e assim ocupando não somente posições mas áreas arbitrárias. A situação que ocorre é o espaço de uma grande construção acaba sendo ocupado por varias construções menores, algumas ocupam inclusive área onde havia outras construções.

Para resolver este problema foi-se analisado o mapa e por ele ser na lógica do jogo um tabuleiro com limites fixos e imutáveis de tamanho, cem de largura e cem de extensão, foi aplicado um histórico por célula para que o histórico, não somente fique mais preciso, mas também seja capaz de armazenar os eventos de cada área do mapa. Por isso o uso de dez mil bancos de dados.

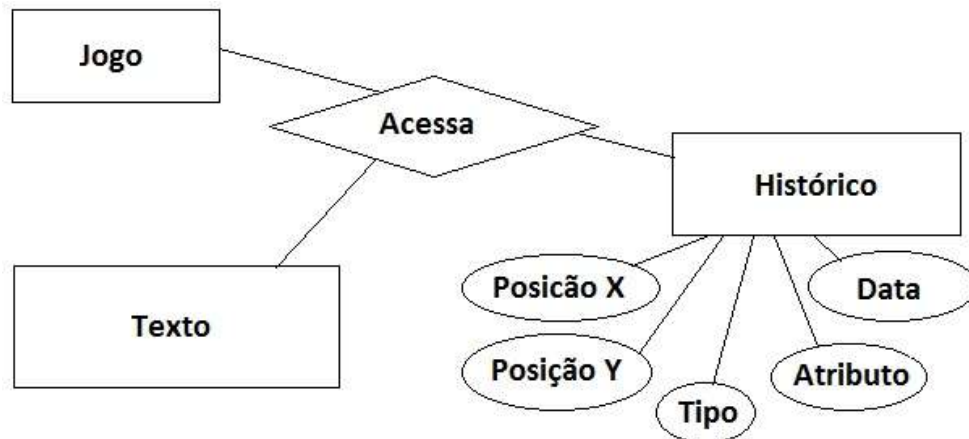
Para o acesso destes dados cada histórico tem o seu nome, que sua forma de acesso, relacionada a sua posição como por exemplo “Eventos45x05”. Retirando assim a necessidade de um módulo auxiliar para o acesso do histórico.

Toda a vez que um evento novo acontece no mapa o jogo insere no histórico uma cópia do evento na célula em que houve o evento e nas adjacentes necessárias.

O jogo acessa o histórico que contem vários atributos, o texto é acessado pelo nome e data do evento do histórico a ser acessado ficando similar a “Texto08x22555” sendo “555” o período do tempo. A velocidade do temporizador que incrementa o contador de dias não permite o jogador construir, criar um texto sobre a construção, destruir, construir novamente e inserir um novo texto, que causaria um erro na forma como está montado atualmente, mas se o sistema de dias for alterado a forma de nomear os bancos de dados de texto terão de ser modificados. A necessidade de se armazenar os textos de uma forma separada é uma limitação da ferramenta usada que não permite definir um texto sem tamanho definido. O banco de dados para o texto é criado na hora porque o único momento em que se sabe o tamanho do texto que o jogador quer escrever é depois que ele esta escrito, neste momento é criado este banco de dados.

Este sistema é muito importante para o jogo pois é através dele que a agente Maga Vitta, que instrui e corrige o jogador através da tecnologia de diálogo com voz e conceitos de inteligência artificial para personalidade, que é a grande e maior inovação tecnológica deste jogo, monitora o histórico de ações do jogador que define a própria personalidade da agente. De certa forma a forma como a Maga Vitta aborda ou responde ao jogador depende principalmente do histórico.

O modulo da Maga ainda não esta pronto para testes, cujos os quais provavelmente resultarão em novas necessidades para o sistema de histórico que já foi feito de forma adaptável para a necessidade de novos parâmetros e métodos.

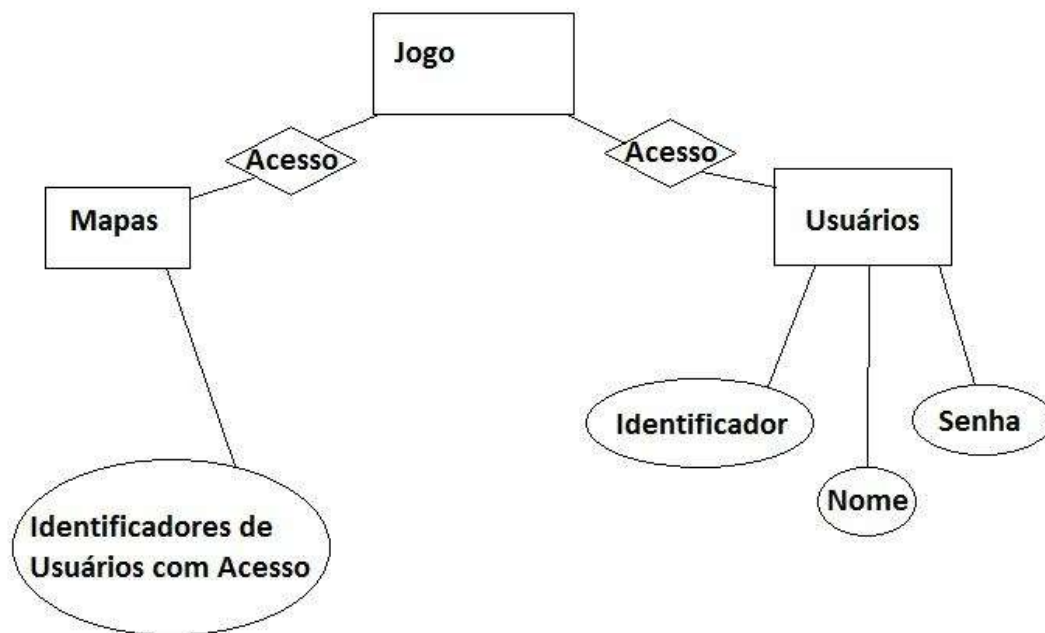


Criado sistema de bancos de dados para armazenar nomes de usuários e senhas e relacionar usuários com lista de mapas que cada usuário tem acesso.

O sistema possui um para os nomes de usuário, senhas e um identificador numérico e um subsistema que contem os nomes dos mapas e os identificadores numéricos dos usuários que tem acesso ao mapa.

O banco de dados de usuários tem uma lista de registros e cada registro tem um campo de nome, um de senha (ambos com limite de trinta caracteres que pode futuramente ser incrementado se testes mostrar necessário) e um campo de identificador que é um número.

Cada mapa possui um banco de dados cuja forma de acesso é o nome do próprio mapa (nome de arquivo) nos registros desses bancos de dados há apenas o identificador numérico dos usuários que te acesso ao mapa. Foi necessário fazer desta forma para possibilitar inserção e remoção de usuários de forma a não prejudicar o desempenho do jogo.



Os sistemas de bancos de dados foram criados com a ferramenta SQLite, a qual é uma biblioteca que, embora não tenha muitos recursos, tem um excelente suporte e é de bom desempenho quanto a processamento.

Diálogo:

Integração com versão de teste do sistema de diálogo, que inclui diálogo de teste de perguntas e respostas, concluída. Isso torna a inteligência artificial, que fará os diálogos com o jogador, já integrada no jogo embora não esteja concluída. Como a interface de invocação dos

métodos de dialogo ainda não foi implementada o modulo de dialogo só pode ser acessado através do código fonte.

Mudanças de Design:

Para criar uma necessidade constante de interferências do jogador foi decidido utilizar números primos de altíssimo valor para definir a geração e consumo de recursos do jogo (população, água, poluição,...) para que a cidade jamais possa estar estabilizada. Dessa forma o tempo do jogo irá fazer com que algo se acumule ou crie falta conseqüentemente fazendo a maga intervir no jogo para orientar o jogador. O motivo de usar números primos é evitar do jogador conseguir estabilizar os consumos e produções das variáveis de ambiente do jogo mesmo usando todo o mapa.

Modo multiplayer usará um sistema de chat que será criado para com que os jogadores possam se comunicar entre si.

Modo Multiplayer funcionara com um servidor dedicado, sem que o computador de algum dos jogadores haja como servidor. O servidor ira enviar atualizações aos clientes somente quando estes forem realizar alguma ação no mapa que esta sendo jogado em rede.

Dificuldades:

Troca do programador responsável pela parte de desenvolvimento do jogo, desenvolvedor responsável somente da parte de jogabilidade e mecânica de gameplay.

Ferramenta gráfica com poucos exemplos e instruções para uso. Estuda-se substituição por outra ferramenta mais atual, de mais baixo nível (para otimizações de desempenho) e utilizada que tenha um maior suporte.

Relatório de Projeto:

ASSISTIVA - Assistente Inteligente Vocal para Maga Vitta

Projeto processo 580470/2008-3

Criado por: *Douglas Collar da Cunha*

Bolsista processo: 180877/2010-0

Período do relatório: *junho/2009 a junho/2010*

Foi inicialmente investigado o estado da arte do processamento de voz, através do estudo das seguintes referências:

- CMU Sphinx (que estás investigando) -
<http://sourceforge.net/projects/cmuspinx/>
- Julius (do projeto em anexo) -
http://julius.sourceforge.jp/en_index.php?q=index-en.html
- Sphinx4 (pelo que entendi, é um Sphinx só em java) -
<http://sourceforge.net/projects/sphinx4/>
- Shout (um toolkit, talvez seja interessante) -
<http://sourceforge.net/projects/shout-toolkit/>
- Slapi (um projeto multilíngue) - <http://sourceforge.net/projects/slapi/>
- OpenSmile (reconhece características da fala, não só voz) -
<http://sourceforge.net/projects/opensmile/>

Destes, foi escolhido em reunião o Sphinx4 para aprofundamento. Seguindo instruções para baixar e instalar o Sphinx 4, foram executados os seguintes programas:

- JSAPI: http://freetts.sourceforge.net/docs/jsapi_setup.html
- ANT: que é instalado junto ao SPHINX 4
- Javdocs: que tbm é instalado junto ao SPHINX 4

As instruções para instalar o SPHINX 4 estão no primeiro link.

Em seguida, foi a vez de testar os exemplos, como por exemplo o programa Hello Digits Demo, que é executado da seguinte maneira:

```
jar -jar bin/"nome do programa".jar
```

como por exemplo:

```
jar -jar bin/HelloDigits.jar
```

alguns programas, como o Hello N-Gram Demo, precisam ainda de um comando `-mx312m` ou `-mx256m` para compilarem, como por exemplo o Hello World Demo, que se executa da seguinte maneira:

```
java -mx256m -jar bin/HelloWorld.jar
```

Os exemplos tiveram um resultado satisfatório, tendo algumas dificuldades com o processamento no computador, e com o reconhecimento no microfone, já que não era o tipo apropriado de microfone para a melhor captação da voz.

O vocabulário usado é o AN4, do Wall Street Journal, e pode ser encontrado aqui:

<http://www.speech.cs.cmu.edu/databases/an4/>

há ainda vários outros vocabulários que podem ser utilizados, porém esse é o mais extenso e foi o que foi usado pelos desenvolvedores da CMU para testar seus programas.

Então, após os testes, era preciso fazer um treinamento de voz, para testar, para isso foi seguido esse tutorial:

<http://www.speech.cs.cmu.edu/sphinx/tutorial.html>

então encontrou-se problemas com o seguinte comando:

```
perl scripts_pl/make_feats.pl -ctl etc/an4_train.fileids
```

então após instalar vários e vários pacotes de arquivos de PERL, a linha anterior continuou não funcionando, pois não aceitava o comando `-ctl`. (além disso, precisei instalar a `an4`, pois não estava na pasta correta.

Como alternativa, foi utilizado o comando:

```
perl scripts_pl/RunAll.pl
```

que instalou novos arquivos ao e diretórios ao diretório atual.
Porém o comando:

```
perl scripts_pl/make_feats.pl -ctl etc/an4_train.fileids
```

continuou não funcionando.

Então verificando o tutorial novamente, foi encontrado o comando:

```
perl ppm install File::Copy
```

que teoricamente instalaria corretamente o PERL, e faria com que fosse possível compilar os comandos anteriormente citados nesse relatório, porém não foi possível utilizar e tampouco instalar o pmm.

Então, foi feita uma tentativa de construir um modelo de linguagem, a partir da página:

<http://www.speech.cs.cmu.edu/cmuspinx/moinmoin/LanguageModelHowto>

A partir dele, descobriu-se o LMTools, que "constrói" um modelo de linguagem, e que foi testado em inglês e funcionou, porém, para fazer um teste em português, era necessário um vocabulário em português, o que não estava disponível no momento.

O LMTools pode ser encontrado no seguinte endereço:

<http://www.speech.cs.cmu.edu/cmuspinx/moinmoin/LanguageModelHowto>

Como as construções com o LMTools eram possíveis apenas remotamente através deste link, em reunião decidiu-se por não utilizar o Sphinx, devido a este fator limitante do desenvolvimento computacional.

Após as atividades realizadas com o Sphinx e já detalhadas anteriormente, comecei a trabalhar junto com o Peter no gravador de voz.

Posteriormente no leitor também.

O gravador/leitor funcionava tanto em Windows como em Linux.

Foram feitos teste de gravação e reprodução de áudio, assim como uma visualização das ondas geradas.

Então foram constatados alguns erros na saída das ondas, posteriormente corrigidas pelo Peter. Utilizou-se também o Audacity para verificar as ondas e fazer testes de reprodução de ondas com o leitor feito pelo Peter, que conseguia reproduzir normalmente o áudio.

Foi criada uma conta no skydrive para salvar os arquivos de voz do projeto.

Iniciei as pesquisas com os softwares Childes e Phon, porém foram descartados, pois não corresponderam às necessidades.

Começaram os testes de JSE para desenvolver o gravador online, com acesso ao banco de dados, porém, após algum tempo de pesquisa, foi constatado que isso seria muito difícil, e tomaria muito tempo, e foi então tomada a decisão de utilizar o Skydrive (<http://skydrive.live.edu>) para disponibilizar o gravador de voz para as professoras.

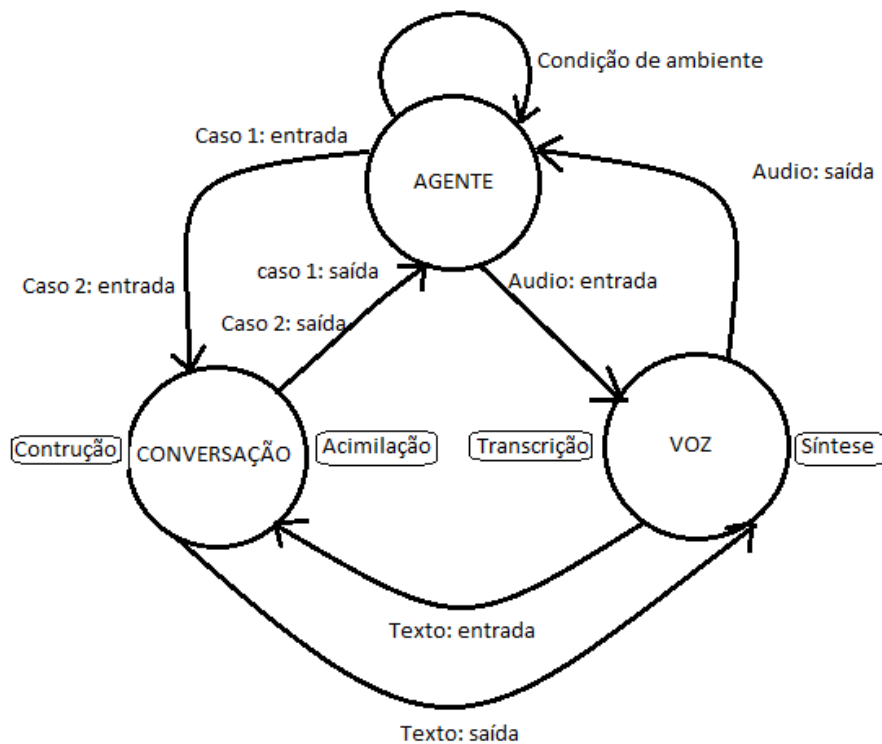
Mostrei o programa para as professoras, e a seguir fiz um tutorial explicativo para a utilização do gravador nas escolas.

O tutorial foi colocado a disposição no Skydrive, para que as professoras pudessem acessar das escolas.

Então a nova versão do gravador ficou pronta.

Desenvolvimento da DLL para reconhecedor de palavras faladas.

Em reunião, desenvolvi o seguinte esquema autômato para melhor compreensão do processamento dos nossos módulos:



Pesquisa do Projeto Falabrasil (UFPA) que utiliza JULIUS como motor de reconhecimento.

Relatório de Projeto:

ASSISTIVA – Assistente Inteligente Vocal para Maga Vitta

Projeto processo 580470/2008-3

Criado por: *Leonivan Dias Moraes*

Bolsista processo: 350208/2009-3 e 350194/2011-4

Período do relatório: *julho/2009 a março/2011 e 01/06/2011 a 30/11/2011*

Em uma reunião na Conexum, fora-me explicado os procedimentos em relação à bolsa de pesquisa e como deve ser o trabalho de um pesquisador. Também foram-me repassados os dados gerais sobre o projeto e os requisitos que deverão ser cumpridos até a conclusão. Sendo assim, tomei par das especificações técnicas para a configuração inicial da Agente Maga.

Dando continuidade ao processo de definição do agente, a equipe de trabalho realizou as seguintes deliberações:

Perfil da Maga Vitta:

- Pode ter vários perfis (cientista, filosófica, artista, mitológica, desmemoriada, criança, má, boa, indecisa, dramática, assustada, etc.) em sua forma e voz, podendo ser randômico ou selecionado pelo professor. Também pode se transformar em personagens como: duendes, E.T., e outros personagens lúdicos;
- Poderá ter personalidade dualística (boa e má) no quesito Desenvolvimento Sustentável;
- Investiga e busca soluções para a cidade;
- Instiga o usuário a buscar pelo conhecimento;
- Deve ser impactante no momento da ação;
- Pode simular/criar problemas para a cidade e realizar questionamentos;
- Nunca diz quais ações são corretas ou não, sempre instigando que o usuário pesquise e descubra;
- Deve provocar a produção de escritas para o ambiente (estórias sobre os as construções, ruas, árvores, etc.);
- Deverá, além da interação por áudio, interagir via texto e/ou imagens, uma vez esse procedimento será necessário quando os equipamentos não dispuserem de dispositivos de som;

Perfil do Ambiente do Jogo:

- O objetivo do jogo é fomentar o usuário na busca pelo conhecimento;
- O usuário deve aprender a interpretar os gráficos gerados pelo ambiente do jogo (gráficos com interfaces amigáveis);

- O próprio usuário deve calcular os fatores de poluição (elaborar forma de cálculo amigável);
- Deverá ser criado um perfil para cada objeto do jogo. Exemplo: o objeto Casa, deverá iniciar com valores definidos, tais como: número de habitantes, consumo de energia, consumo de água, geração de lixo, etc.)
- O “estado da arte” do jogo deverá ser exportável e importável;
- Haverá “caixinhas” onde o usuário poderá inserir a estória do objeto componente da cidade;
- No ambiente do jogo poderá acontecer catástrofes naturais, fomentando a busca pela reorganização ambiental no ambiente do jogo;

Informações Gerais:

- O ambiente do jogo seguirá a premissa dos 4E's: Ecologia, Economia, Energia e Ética.
- Trabalhar com fatores lúdicos, instigando o imaginário do usuário;
- Relacionar os fatores (dados do ambiente de jogo) com o ambiente real, promovendo a consciência ecológica do usuário;
- Deverá ser possível remeter o usuário para fora dos limites computacionais do ambiente do jogo, objetivando a busca por informações e consequente melhora da interação com o meio ambiente.

Posteriormente me reuni com a Profa. Maria Thereza do Departamento de Ecologia da UFRGS, na sala da Conexum, objetivando coletar informações para calcular os fatores de poluição dos objetos do jogo Citta, no qual a agente Maga Vitta será incluída. Após longa conversa e analisando o material trazido pela professora, relacionamos os coeficientes de poluição dos objetos.

Após essa reunião, surgiu a necessidade de especificação das formas de ações do agente, onde, inicialmente, foi estabelecido a gama de informações que deverão ser implementadas no ambiente de jogo para que a agente possa realizar intervenções de forma correta, contribuindo para o aprendizado do usuário.

Sendo assim, as tabelas iniciais foram especificadas, e seguem descritas na sequência:

TABELA GERAL		
Item	Valor na Escala	Valor Real (Diário)
Água	1	(Consome) 150 litros / dia / habitante
Alimentos	1	(Consome) 1,0 kg / dia / habitante
Energia	1	(Consome) 1,3 kwa / dia / habitante
Esgoto	1	(Produce) 120 litros / dia / habitante
Espaço	1	(Consome) 1 m2
Lixo	1	(Produce) 1 kg. / dia / habitante

CASA		
Qtde. Pessoas: 04		
Consome	Qtde. Na Escala	Neutralizar
Água	4	
Energia	4	5 árvores/ano
Alimentos	4	
Polui	Qtde. Na Escala	Neutralizar
Lixo	4	13 árvores/ano
Esgoto	4	

PRÉDIO		
Qtde. Pessoas: 40		
Consome	Qtde. Na Escala	Neutralizar
Água	40	
Energia	40	50 árvores/ano
Alimentos	40	
Polui	Qtde. Na Escala	Neutralizar
Lixo	40	130 árvores/ano
Esgoto	40	

ESCOLA		
Qtde. Pessoas: 40		
Consome	Qtde. Na Escala	Neutralizar
Água	40	
Energia	40	50 árvores/ano
Alimentos	40	
Polui	Qtde. Na Escala	Neutralizar
Lixo	40	130 árvores/ano
Esgoto	40	

HOSPITAL		
Qtde. Pessoas: 80		
Consome	Qtde. Na Escala	Neutralizar
Água	80	
Energia	60	76 árvores/ano
Alimentos	80	
Polui	Qtde. Na Escala	Neutralizar
Lixo	80	260 árvores/ano
Esgoto	80	

IGREJA		
Qtde. Pessoas: 5 funcionários.		
Consome	Qtde. Na Escala	Neutralizar
Água	8	
Energia	10	13 árvores/ano
Alimentos		
Polui	Qtde. Na Escala	Neutralizar
Lixo	5	17 árvores/ano
Esgoto	8	

FÁBRICA		
Qtde. Pessoas: 30 funcionários.		
Consome	Qtde. Na Escala	Neutralizar
Água	40	
Energia	100	127 árvores/ano
Alimentos		
Polui	Qtde. Na Escala	Neutralizar
Lixo	300	970 árvores/ano
Esgoto	40	

PREFEITURA		
Qtde. Pessoas: 30 funcionários.		
Consome	Qtde. Na Escala	Neutralizar
Água	12	
Energia	12	16 árvores/ano
Alimentos		
Polui	Qtde. Na Escala	Neutralizar
Lixo	30	97 árvores/ano
Esgoto	12	

MERCADO		
Qtde. Pessoas: 20 funcionários.		
Consome	Qtde. Na Escala	Neutralizar
Água	15	
Energia	18	23 árvores/ano
Alimentos		
Polui	Qtde. Na Escala	Neutralizar
Lixo	100	324 árvores/ano
Esgoto	15	

FAZENDA		
Qtde. Pessoas: 10 funcionários.		
Consome	Qtde. Na Escala	Neutralizar
Água	10	
Energia	25	32 árvores/ano
Polui	Qtde. Na Escala	Neutralizar
Lixo	50	162 árvores/ano
Esgoto	10	
Produz	Qtde. Na Escala	
Alimentos	400	

ÁRVORE P	
Qtde.	Neutraliza
300	-100 árvores

ÁRVORE M	
Qtde.	Neutraliza
300	-200 árvores

ÁRVORE G	
Qtde.	Neutraliza
300	-300 árvores

ESTAÇÃO DE TRATAMENTO DE ÁGUA	
Item	Produção
Água	200.000 litros/ dia

USINA EÓLICA	
Item	Produção
Energia	3.000 MW / dia

ESTAÇÃO DE TRATAMENTO DE ESGOTO	
Item	Capacidade de Tratamento
Esgoto	-300.000 litros / dia

ATERRO SANITÁRIO	
Item	Consome
Lixo	2.000 kg / dia

USINA TRATAMENTO DE LIXO HOSPITALAR (UTLH)	
Consome	Capacidade de tratamento
Lixo	-3.000 kg / dia

Discutindo-se acerca das definições das ações iniciais da agente no ambiente do jogo bem como a sua forma de atuação. Foi sugerido que, ao iniciar uma nova sessão de jogo, a agente aja de forma a sugerir que o usuário configure a cidade com as necessidades básicas para evitar a poluição, porém, a agente não deve intervir em outros fatores que não sejam de controle da poluição do meio ambiente.

Realizei as seguintes leituras de artigos, objetivando ampliar o entendimento do funcionamento de Redes Bayesianas e do Raciocínio Baseado em Casos (RBC), os quais estão sendo úteis para o aprimoramento cognitivo da Agente Virtual Maga Vitta:

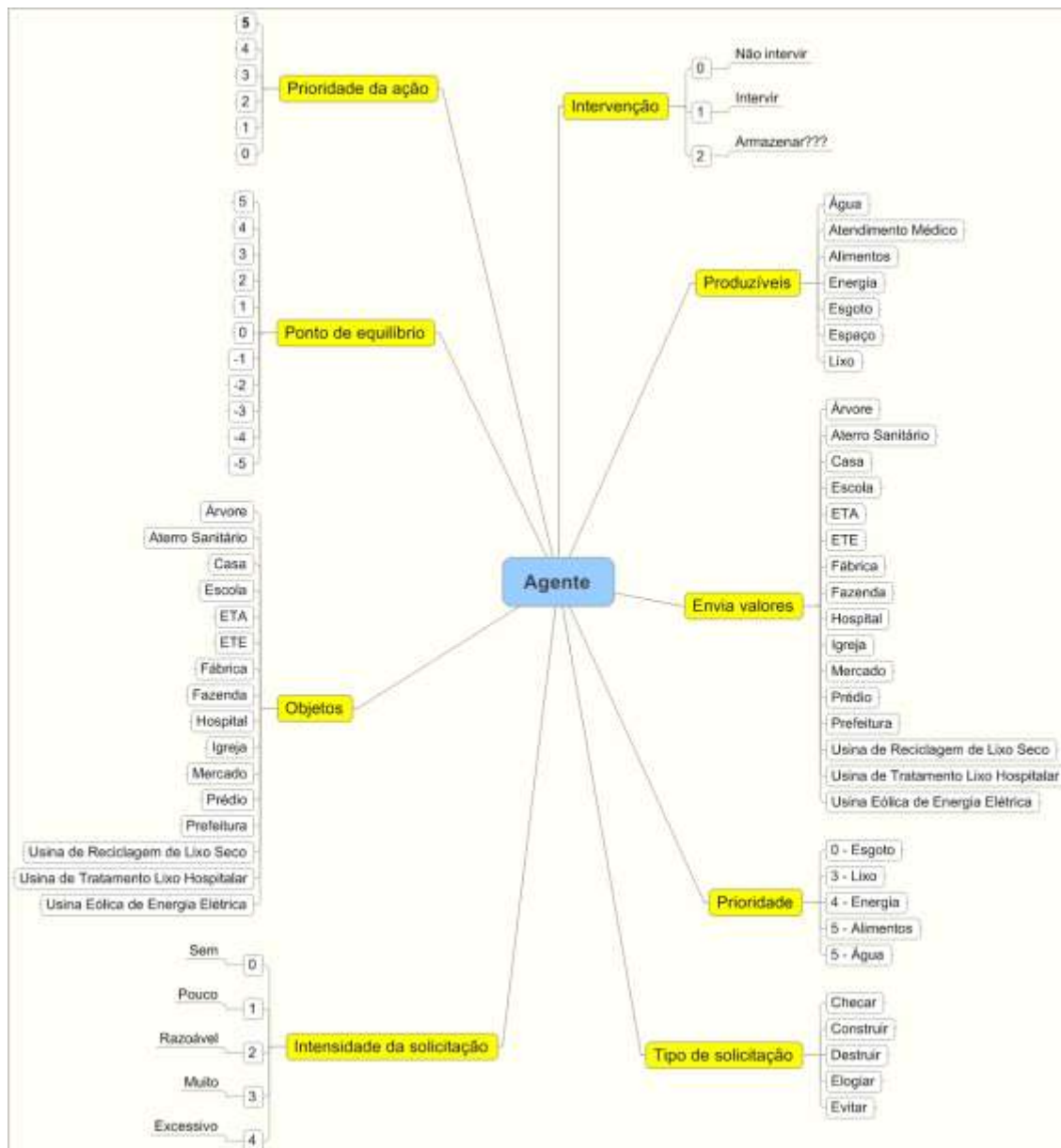
- Retrieval of Cases by Using Bayesian Network.
- Integrating Bayesian Networks into Knowledge-Intensive CBR.
- Automatic parameter tuning with a Bayesian case-based reasoning system.
- Evaluating Bayesian networks precision for detecting students learning styles.
- Cluster-based predictive modeling to improve pedagogic reasoning.
- User profiling with Case-Based Reasoning and Bayesian Networks.

Discutimos sobre como deverão ser realizadas as intervenções da Agente Maga Vitta durante o jogo e como será a interação com o usuário. Como resultado, haverá a "quebra" das frases recebidas do usuário em palavras chave, as quais serão repassadas como parâmetros para realizar o processo de interação. Também ficou acordado que, ao iniciar o jogo, haverá uma mensagem de boas-vindas e de auto-apresentação da Agente Maga Vitta.

Conversamos sobre o desenho da Maga Vitta, a sua aparência, a forma como ela poderia aparecer na tela (espaço que ocuparia), se o jogo paralisaria e ela tomaria a tela ou as crianças seguiriam visualizando o jogo enquanto ouviam/liam as dicas, conversamos também sobre a animação da Maga.

Sobre o desenho ficou acordado o seguinte: que inicialmente o desenho será em 2D e apenas com movimento da boca inicialmente (haverá futuras modificações). As dúvidas foram quanto à aparência dela, já que é uma Maga, deveria seguir os estereótipos da personagem (chapéu, estrelas e lua na roupa, varinha de condão, etc.) Como ela vai surgir para as crianças? Esse ponto será decidido e melhor pensado após realizadas algumas simulações.

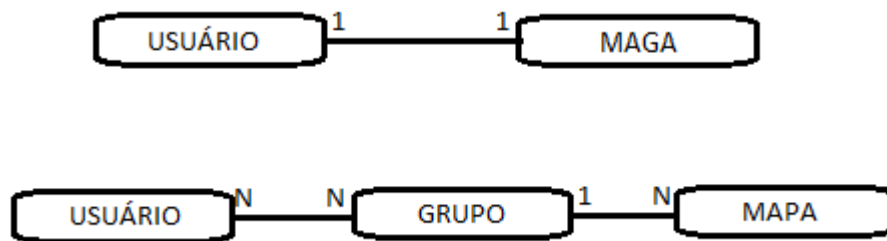
Entre o final de 2009 e início de 2010, após ampla discussão com a equipe, construí o mapa mental da Agente Maga Vitta, o qual serve de base para ilustrar o mapa cognitivo da agente, bem como os parâmetros iniciais para interação com o usuário:



Em reunião falou-se sobre a implantação do Città nas escolas e como os alunos se motivam a jogar e interagir com as tecnologias disponíveis dentro do ambiente escolar. Em relação ao ambiente de jogo acordou-se que seja armazenado o desempenho dos jogadores para poder pesquisar a evolução do aluno/jogador e seu raciocínio lógico antes e após a intervenção da agente Maga Vitta. Também ficou acertada a criação de arquivos de Log de todo o ambiente de jogo, objetivando servir como base de estudo para publicações e expôr o cenário de desenvolvimento da síntese de voz. Otto e eu apresentamos um layout do ambiente de jogo, onde as variáveis de controle ficam expostas na tela principal, objetivando realizar as simulações para calibrar o processo de construção da cidade virtual.

Dentre os comunicados, foi criado por Douglas um espaço virtual para armazenamento no Skydrive (<http://skydrive.live.edu>) dos documentos trocados no projeto (textos, imagens, áudio, etc).

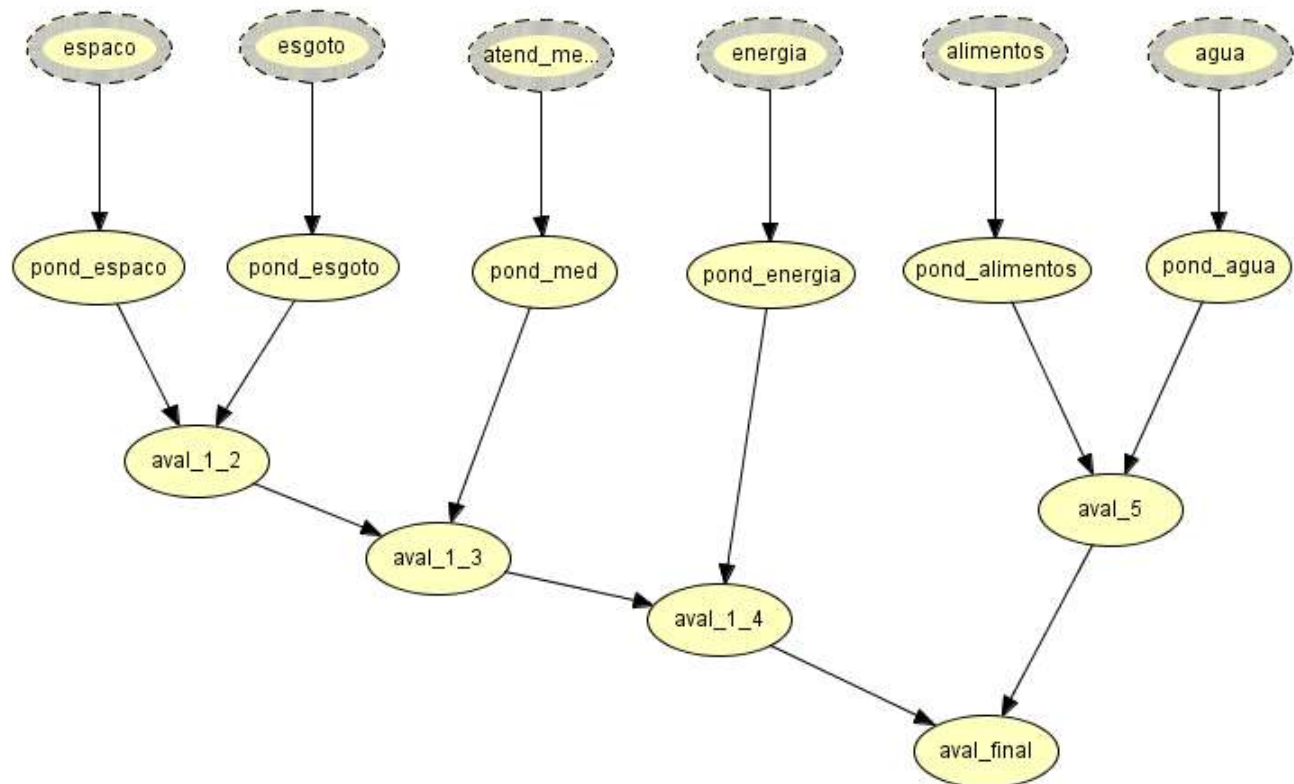
Em reunião foi definido que o controle dos bancos de dados do Città e da agente Maga Vitta se darão da seguinte forma:



Onde a Maga será única para cada usuário, com suas ações e diálogos registrados nas bases associadas ao usuário, e os mapas dos jogos associados às bases de um grupo de usuários.

De forma a prosseguir a investigação de como implementar o mapa mental da Maga Vitta, prossegui leituras de artigos, no sentido de avaliar a possibilidade de utilizar um raciocínio bayesiano juntamente com o esquema de Raciocínio Baseado em Casos previsto para o diálogo da Maga Vitta pelo Alexsandro.

Após diversas discussões em reuniões, implementei, a título de protótipo para o mapa mental da Maga, desenvolvi, no aplicativo de desenvolvimento de sistemas especialistas com redes bayesianas *Hugin Lite* (<http://www.hugin.com/productsservices/demo/hugin-lite>), o seguinte modelo bayesiano de teste:



Os nodos superiores correspondem às variáveis de ambiente definidas no modelo mental da Maga Vitta. A cada variável está associado um conjunto de ponderações, que formam a segunda linha de nodos. Elas correspondem às variáveis linguísticas *pouco*, *razoável*, *muito* e *excessivo*. Estas variáveis foram estabelecidas para simplificação do processo de tomada de decisão, o qual gera tabelas extensas para estabelecimento das avaliações, as quais formam os nodos inferiores no esquema acima.

As avaliações em cascata devem-se a sua relação com as prioridades das variáveis de ambiente, onde espaço < esgoto < atendimento médico < energia < alimentos/água.

Após finalizados os testes com o protótipo, foi iniciada a implementação em Java do modelo bayesiano, utilizando-se a plataforma em software livre *JavaBayes – Bayesian Network Classifiers in Java* (<http://sourceforge.net/projects/jbnc/>).

Implementação do Módulo de Inferência

Como a Agente Maga Vitta é a consciência do jogo Città, promovendo a regulação do sistema ecológico, sua proposta é provocar ações internas no aluno de forma que o faça manter o equilíbrio natural deste sistema. A gravidade da situação é demonstrada pelo nível de equilíbrio da variável de ambiente, pela prioridade do elemento que está sendo considerado e pelas recorrências; a soma destes fatores indica ao módulo de diálogo a importância e o nível de expressividade que o agente terá que adotar na interação.

As variáveis de ambiente são relativas aos produzíveis e consumíveis na cidade virtual, correspondendo ao quanto é produzido ou gasto por cada objeto pertencente à ao jogo. Os objetos criados para composição da cidade virtual são: casa, prédio, escola, hospital, igreja, fábrica, prefeitura, mercado, fazenda, árvore, ETA (Estação de Tratamento de Água), usina eólica, aterro sanitário, ETE (Estação de Tratamento de Esgoto), UTLH (Unidade de Tratamento de Lixo Hospitalar).

Desta forma, as tabelas construídas com os produzíveis e consumíveis foram revistas e sintetizadas na tabela abaixo, a qual foi codificada em Java no projeto do jogo Città.

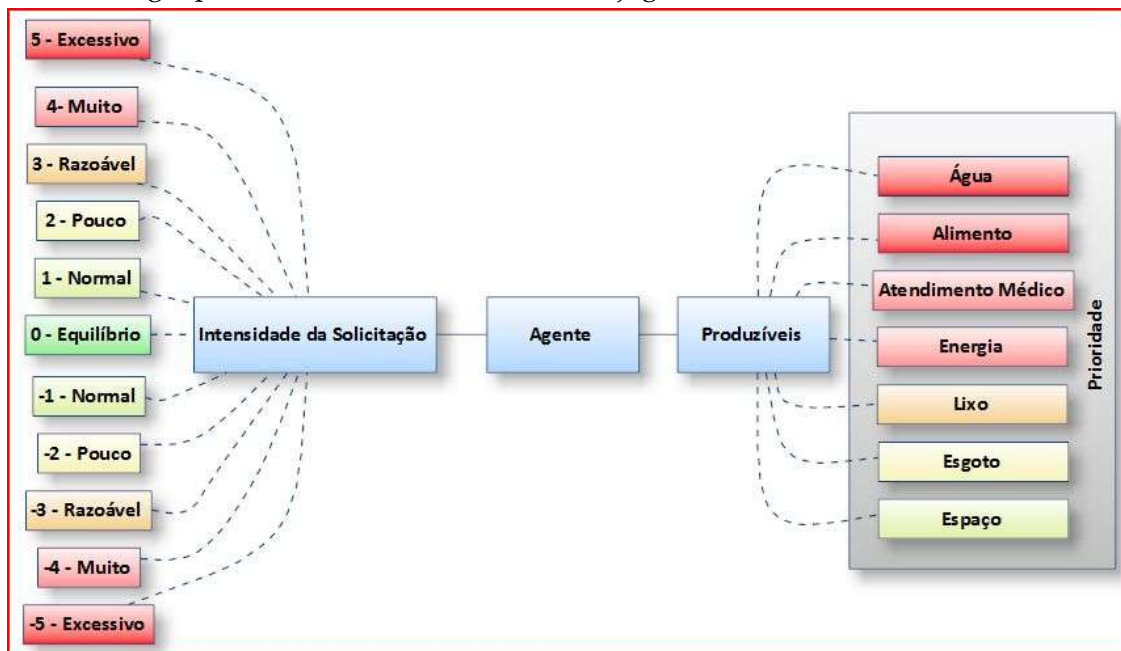
Objeto	Qtde Pessoas	Consome (em qtde)			Produz (em qtde)			Árvores p/ neutralização (ao ano)				
		Água	Energia	Alimento	Alimento	Lixo	Esgoto	Água	Energia	Alimento	Lixo	Esgoto
casa	4	4	4	4		4	4		5		13	
prédio	40	40	40	40		40	40		50		130	
escola	40	40	40	40		40	40		50		130	
hospital	80	80	60	80		80	80		76		260	
igreja	5	8	10			5	8		13		17	
fábrica	30	40	100			300	40		127		970	
prefeitura	30	12	12			30	12		16		97	
mercado	20	15	18			100	15		23		324	
fazenda	10	10	25		120	50	10		32		162	

Objeto	Qtde	Corresponde p/ neutralizar
árvore P	300	100
árvore M	300	200
árvore G	300	300
	Produz	
ETA	200.000 litros/dia	
Usina Eólica	3.000 MW/dia	
	Consome	
Aterro Sanitário	15.000 kg/dia	
	Trata	
ETE	300.000 litros/dia	
UTLH	3.000 kg/dia	

A título de exemplo, os testes para a variável de ambiente *água* tem sua implementação como no trecho de código a seguir, onde os valores da variável são testados dentro de intervalos para ponderação. Os mesmos testes são aplicados às demais variáveis, sendo utilizados os valores de teste adequados para cada uma delas.

```
// agua
if(AtualizadorAmbiente.agua > 1000000) { m_agua = 4; s_agua = true;}
else
if(AtualizadorAmbiente.agua > 500000) { m_agua = 3; s_agua = true;}
else
if(AtualizadorAmbiente.agua > 250000) { m_agua = 2; s_agua = true;}
else
if(AtualizadorAmbiente.agua > 100000) { m_agua = 1; s_agua = true;}
else
if(AtualizadorAmbiente.agua > -100000) { m_agua = 1; s_agua = false;}
else
if(AtualizadorAmbiente.agua > -250000) { m_agua = 2; s_agua = false;}
else
if(AtualizadorAmbiente.agua > -500000) { m_agua = 3; s_agua = false;}
else
if(AtualizadorAmbiente.agua <= -500000) { m_agua = 4; s_agua = false;}
```

As variáveis de ambiente definidas até o momento são espaço, esgoto, lixo, atendimento médico, energia, alimentos e água. Esta é, respectivamente, a ordem inversa de prioridade de análise, sendo espaço menos significativo, e alimentos e água os mais importantes. No trabalho de Garcia et al (2007) pode ser encontrado um exemplo de uso semelhante, onde variáveis de software educacional foram utilizadas para inferência de categorias de estudantes através de redes bayesianas. Na abordagem do ASSISTIVA, utiliza-se a inferência para determinação do caráter do diálogo que será travado com o aluno do jogo.



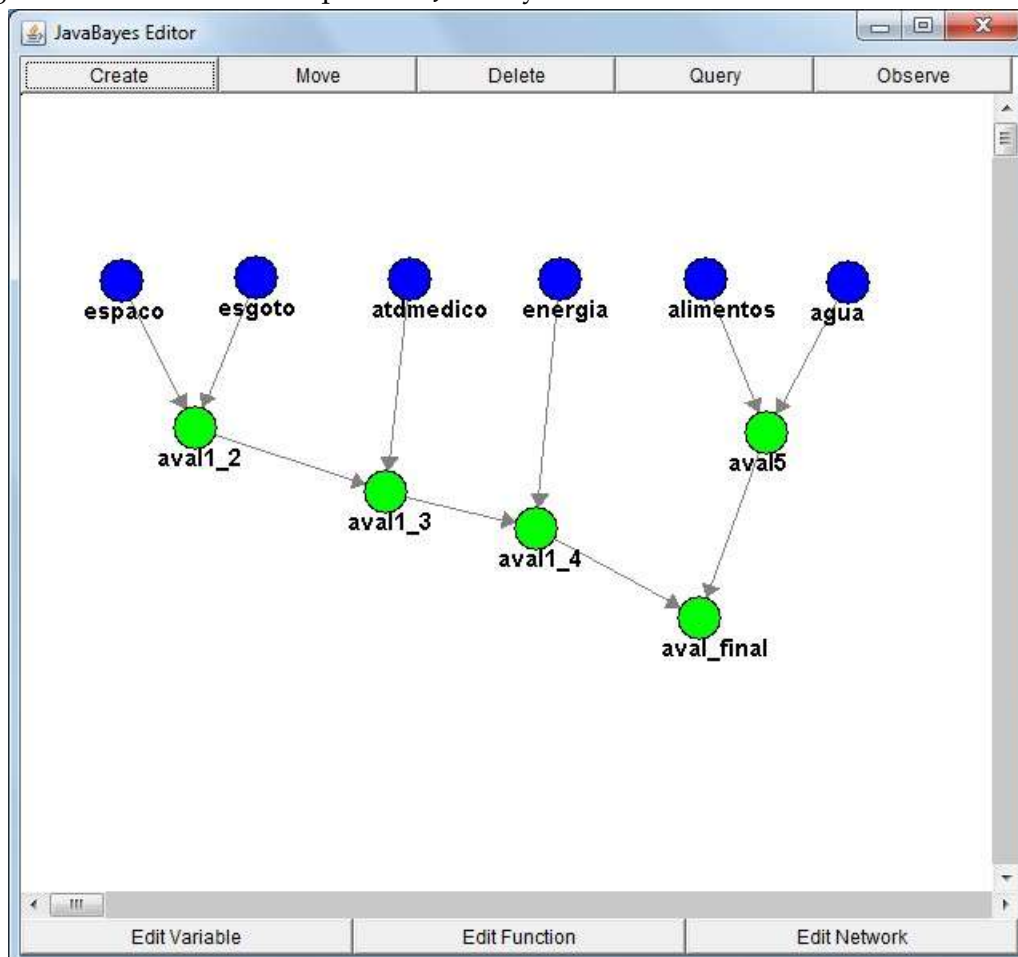
O esquema da acima redefine o primeiro mapa mental construído. Nesta nova disposição os nodos superiores correspondem às variáveis de ambiente definidas no modelo mental da Maga Vitta. A cada variável está associado um conjunto de ponderações, que formam a segunda linha de nodos. Elas correspondem às variáveis linguísticas *pouco*, *razoável*, *muito* e *excessivo*. Estas variáveis foram estabelecidas para simplificação do processo de tomada de decisão, o qual gera

tabelas extensas para estabelecimento das avaliações, as quais formam os nodos inferiores. As avaliações ocorrem em cascata devido a sua relação com as prioridades das variáveis de ambiente.

No exemplo de código a seguir, os valores lidos das variáveis de ambiente são aplicados à inferência bayesiana, a partir da qual são obtidos os valores de decisão.

```
grafoInfer = new InferenceGraph("maga3");
for (Enumeration<?> e = grafoInfer.elements(); e.hasMoreElements(); ) {
    node = (InferenceGraphNode)e.nextElement();
    if(node.is_observed()&&node.get_name().equals("espaco")){node.set_observation_value(Integer.toString(m_espaco));}
    if(node.is_observed()&&node.get_name().equals("esgoto")){node.set_observation_value(Integer.toString(m_esgoto));}
    if(node.is_observed()&&node.get_name().equals("atdmedico")){node.set_observation_value(Integer.toString(m_atdMedico));}
    if(node.is_observed()&&node.get_name().equals("energia")){node.set_observation_value(Integer.toString(m_energia));}
    if(node.is_observed()&&node.get_name().equals("alimentos")){node.set_observation_value(((Integer)m_alimentos).toString());}
    if(node.is_observed()&&node.get_name().equals("agua")){node.set_observation_value(((Integer)m_agua).toString());}
}
if ( (qbi == null) || (qbi.areClustersProduced() == false) ) qbi = new QBInference(grafoInfer.get_bayes_net(), false);
qbi.inference("aval_final");
res = qbi.get_result();
resultados = res.get_values();
```

No código apresentado acima, é utilizada a rede bayesiana projetada para tomada de decisão das variáveis de ambiente, as quais têm os valores aplicados nos nodos superiores e a resposta é obtida no nodo de nível mais baixo, conforme mostra a figura a seguir, obtida a partir da modelagem realizada com o componente JavaBayes.



Na última etapa do projeto estamos avaliando a possibilidade de utilizar um raciocínio bayesiano juntamente com o esquema de Raciocínio Baseado em Casos (RBC) elaborado para o diálogo da Maga Vitta, apresentado anteriormente. Neste sentido, pode ser citado o trabalho de Pavón et al (2009), que utiliza a inferência bayesiana como auxiliar na busca do conjunto de soluções de um RBC.

Desta forma, foram pesquisadas implementações de RBCs em software livre, para composição ao projeto do jogo Città. A plataforma RBC que convergimos para integração ao nosso projeto foi a jCOLIBRI – CBR Framework (<http://sourceforge.net/projects/jcolibri-cbr/>).

Referências

Dingsøyr, Torgeir. 1998. “Retrieval of cases by using a Bayesian network”, Proceedings of the AAAI Workshop on Case-Based Reasoning Integrations, 1998, pages 50-54.

Garcia P., Amandi A., Schiaffino S., Campo M. Evaluating Bayesian networks' precision for detecting students' learning styles. Computers and Education, 49 (3), 2007, pages 794-808.

Helge, Agnar Aamodt and Langseth, Helge. Integrating Bayesian Networks into Knowledge-Intensive CBR. In: American Association for Artificial Intelligence, Case-based reasoning integrations; Papers from the AAAI workshop. Technical Report WS-98-15. AAAI Press, 1998, pages 1-6.

Legaspi R., Sison R., Fukui K.-i., Numao M. Cluster-based predictive modeling to improve pedagogic reasoning. Computers in Human Behavior, 24 (2), 2008, pages 153-172.

Reyes Pavón, Fernando Díaz, Rosalia Laza, M. Victoria Luzón: Automatic parameter tuning with a Bayesian case-based reasoning system. A case of study. Expert Syst. Appl. 36(2), 2009, pages 3407-3420.

Schiaffino, Silvia and Amandi, Analía. User profiling with Case-Based Reasoning and Bayesian Networks. In: Open Discussion Track Proceedings of the International Joint Conference IBERAMIA-SBIA 2000, 2000, pages 12-21.

Relatório de Projeto:

ASSISTIVA – Assistente Inteligente Vocal para Maga Vitta

Projeto processo 580470/2008-3

Criado por: *Alexsandro Rosa dos Anjos*

Bolsista processo: 383954/2009-6 e 382671/2011-2

Período do relatório: *setembro/2009 a abril/2011 e junho/2011 a novembro/2011*

Competências desenvolvidas durante o projeto:

Estudo de ferramentas, linguagens e assuntos referentes a desenvolvimento dos processos de linguagem, conversação e composição dos padrões das frases para as falas da agente virtual do jogo Città.

Ferramentas utilizadas no desenvolvimento do módulo de diálogo entre o agente e o aluno dentro do projeto.

1. NetBeans – Programa utilizado como ambiente de desenvolvimento;
2. Java – Linguagem de programação
3. Sqlite – Banco de dados

Assuntos e esquemas realizados durante as reuniões do projeto

Foi estudado sobre os fundamentos de PLN (Processamento de Linguagem Natural). Esses fundamentos consistem em entender o processo para desenvolvimentos das conversações. Junto a esse estudo ficou definido que seria construído um módulo de diálogo e frases que compusessem afirmações para uso da agente virtual.

Foi decidido em reunião que os assuntos iniciais a serem tratados para o processo de criação de respostas da assistente seriam o Lixo e logo após Esgoto. Juntamente com essa decisão foi identificado a necessidade de um domínio de conceitos para o tratamentos de algumas palavras como exemplo (Recolher, Juntar, Onde – remete ao mesmo domínio “local”)

Ex:

Maga: Onde colocar o lixo? ou

A cidade está com lixo nas ruas. O que fazer?

Usuário: Recolher o lixo!

Maga: Para onde?

Esquema:

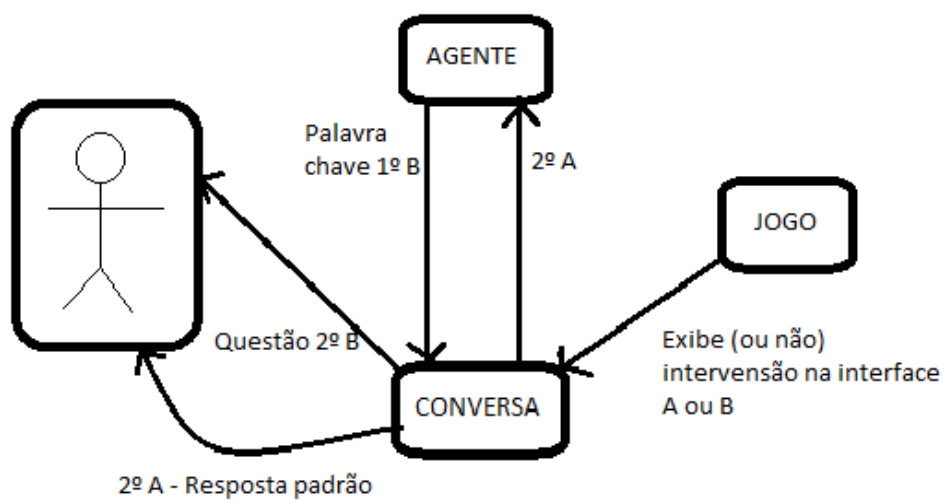


Figura 1 - Esquema de diálogo

Juntamente com esse esquema foram definidos os parâmetros para conversa entre os módulos de diálogo e o jogo, assim como, o mapa mental da agente virtual:

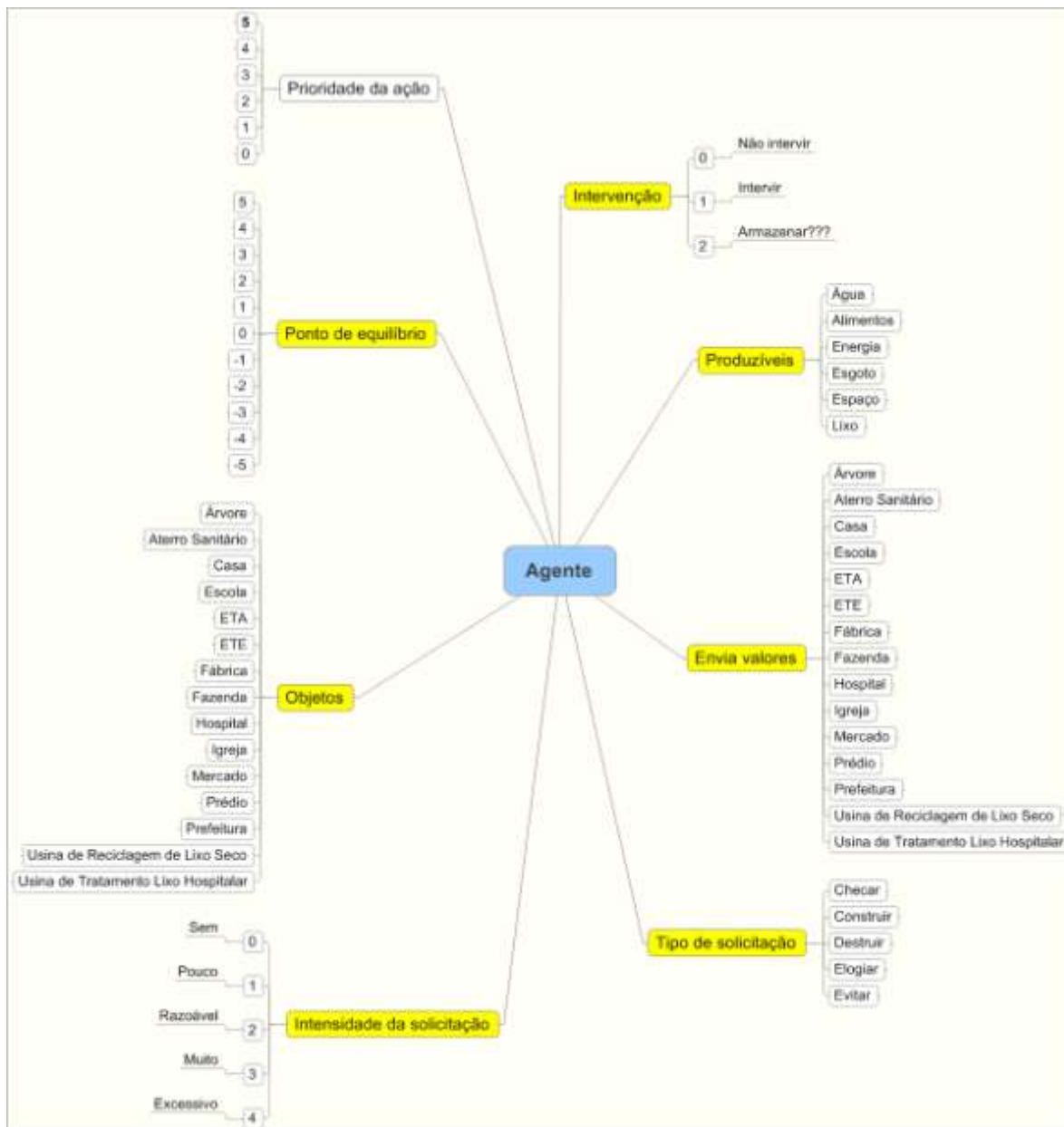


Figura 2 - Mapa mental da Agente Virtual

Palavras chave: Água, Alimentos, Energia, Esgoto, Espaço, Espaço.

Contextos: Checar, Construir, Destruir, Evitar, Elogiar.

Intensidade: 0 => Sem,

1 => Pouco,

2 => Razoável,

3 => Bastante,

4 => Excessivamente.

Exemplo:

Palavra chave: Lixo.

Contextos: Construir, Destruir, Checar, Evitar, Elogiar.

Intensidade: 0 => Sem, 1 => Pouco, 2 => Razoável, 3 => Bastante, 4 => Excessivo.

Caso Checar: {

Dica: "Com o crescimento da cidade o lixo também aumenta, cheque seu lixo."

}

Caso Construir: {

Se Intensidade = 0 {

Dica: Sua cidade já está produzindo lixo, já tem onde colocá-lo?}

Se Intensidade = 1 {

Pergunta: Sua cidade já está produzindo lixo, onde pode ser colocado?}

Se Intensidade = 2 {

Dica: A cidade aumentou a produção de lixo, se já existe um lugar para colocá-lo, o que pode ser feito?}

Se Intensidade = 3 {

Dica: Nossa! Que cidade suja! Alguma coisa tem que ser feita com esse lixo!}

Se Intensidade = 4 {

Dica: Com essa quantidade de lixo as pessoas podem ficar doentes! Alguma coisa precisa ser feita!}

}

Caso Destruir: {

Se Intensidade = 0 {

Dica: }

Se Intensidade = 1 {

Pergunta: }

Se Intensidade = 2 {

Dica: }

Se Intensidade = 3 {

Dica: Será que é necessário tanto espaço para o lixo? Esse espaço poderia ser melhor aproveitado!}

Se Intensidade = 4 {

Dica: Existe espaço de mais para o lixo, esse espaço poderia ser melhor aproveitado! Tem alguma boa ideia?}

}

Caso Elogiar: {

Se Intensidade = 0 {

Dica: }

Se Intensidade = 1 {

Dica: }

Se Intensidade = 2 {

Dica: Bom trabalho! Tá muito boa a cidade. }

Se Intensidade = 3 {

Dica: Muito bom trabalho! Espero que continue assim! }

Se Intensidade = 4 {

Dica: Mas que ótimo trabalho vem sendo feito na cidade! Muito bom mesmo! }

}

Caso Evitar: {

Se Intensidade = 0 {

Dica: }

```

Se Intensidade = 1 {
    Pergunta: }
Se Intensidade = 2 {
    Dica: }
Se Intensidade = 3 {
    Dica: }
Se Intensidade = 4 {
    Dica: }
}

```

Como proposta de melhoramento futuro no jogo, e visando a personalização da agente, sendo ela exclusiva do aluno, foi pensado num log que ajudaria a controlar o nível de evolução do diálogo através do tempo no jogo. Conjuntamente foi criado um protocolo de conversa entre módulo de diálogo e o jogo. Com esse protocolo são trocadas as informações das variáveis de ambiente do jogo para formular e auxiliar na composição das frases de repostas. E para formação dessas frases de conversa com os alunos contamos com a nossa parceria com o pessoal da educação que são os idealizadores do projeto Civitas ao qual, o Assistiva está inserido com o objetivo de criar a parte de informática do projeto.

Seguindo com reuniões e atividades durante o ano, chegamos ao entendimento e construção de um chat¹, conforme figura 3, como melhor forma de interface entre o usuário e o agente virtual. Dessa forma pode-se pausar o jogo para que o diálogo entre o agente e o aluno decore de forma mais natural e mais direcionada.

¹ Figura 3 – Primeira tela idealizada do chat

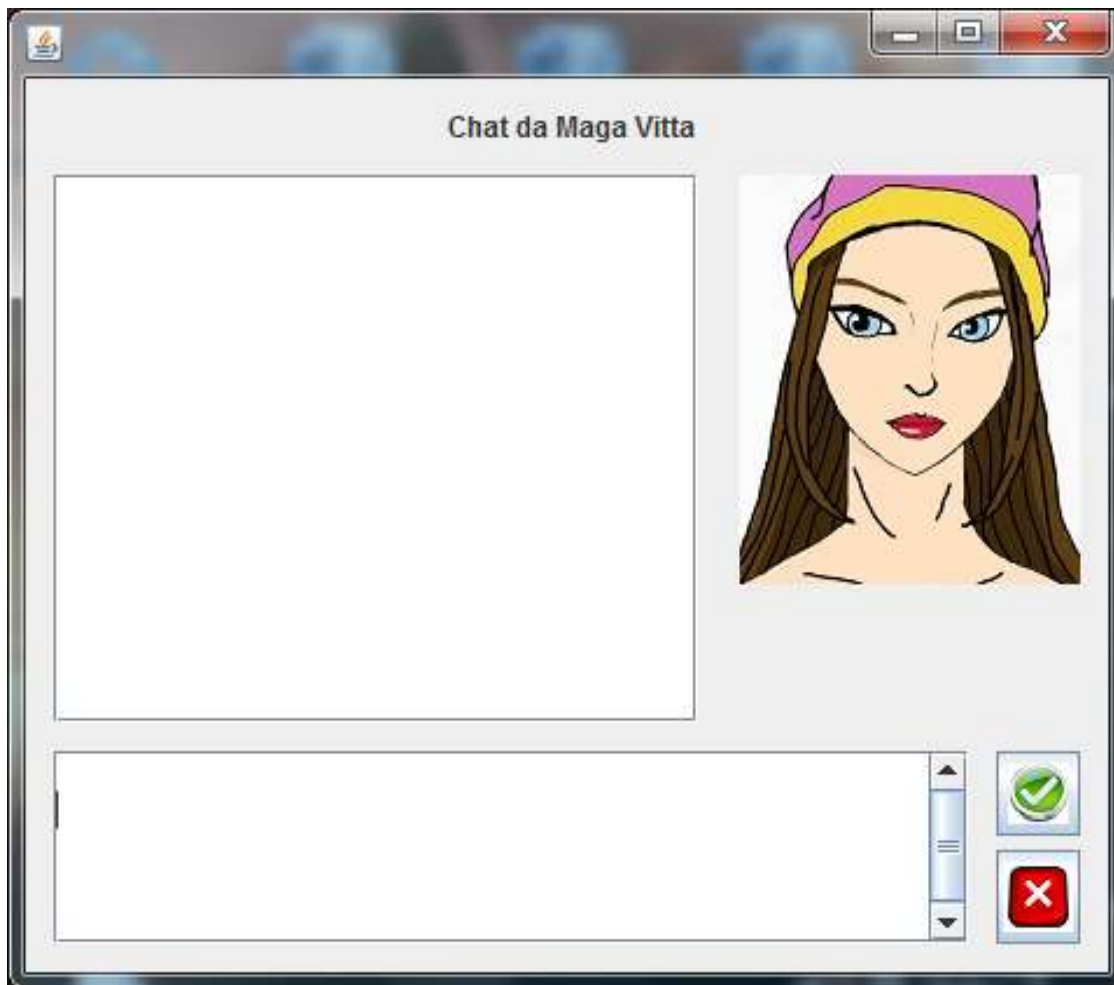


Figura 3 - Chat de interação Usuário/|Maga

Dentro do módulo de diálogo² subdivide-se em algumas etapas para construção do diálogo através do esquema na figura 4, onde a frase entra no sistema através da interface do chat e o controlador que dispara processos e rotinas para a construção do diálogo.

² Figura 4 – Diagrama dos passos para geração de diálogo com o usuário

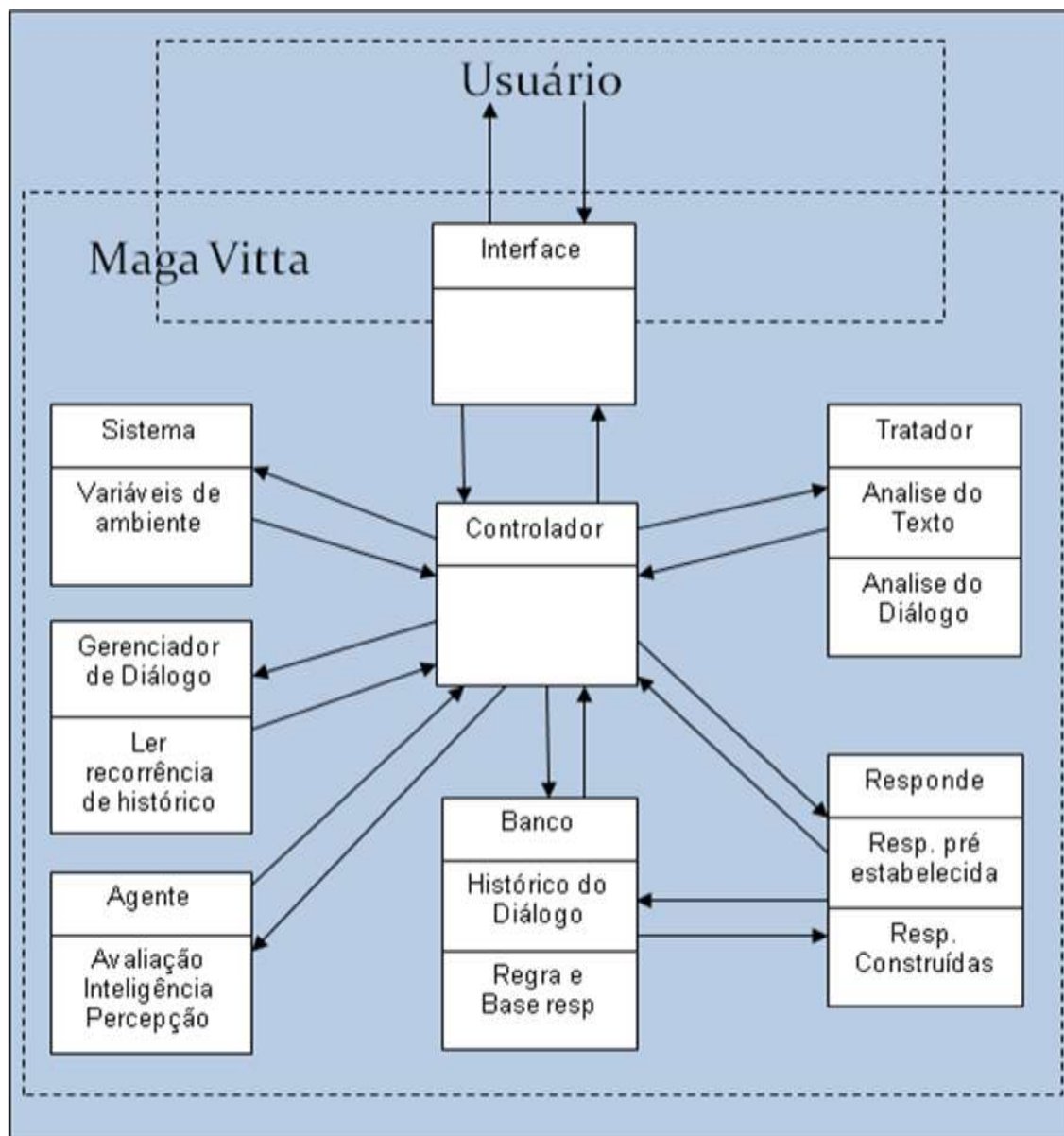


Figura 4 - Diagrama de construção da frase (Maga)

Com base no esquema da figura 4, foi criada a rotina de ponderação e tomada de decisão baseado no mapa mental (figura 2) e no protocolo de conferências das variáveis de ambiente, o desenvolvimento do processo de tratamento usado para analisar o texto e o contexto do diálogo, a rotina do gerenciador de diálogo para controlar ocorrências de diálogos paralelos (ex: outros assuntos) e também a recorrências de casos³ e históricos de acontecimentos anteriores, e a estrutura do banco de dados⁴ para armazenamento das falas conforme figura 5,

³ Figura 5 – Log de casos

⁴ Figura 6 – Estrutura de armazenamento do banco de dados.

Formato da Interação do Agente Maga Vitta com o Aluno

Para que o processo de diálogo seja iniciado, uma ou mais variáveis de ambiente deve(m) estar fora da faixa considerada equilibrada, gerando assim uma solicitação no módulo de inferência e ativando uma tela de chat/voz. Após a entrada de texto ou voz é disparado um processo para o controlador, que após avaliar as condições do ambiente, dispara um processo para o tratador que avalia o conteúdo semântico do texto, ou para ser extraído, ou para ser verificado em relação a informações do estado atual, e em uma função comunicativa e paralelamente ao tratador é disparado um processo no gerenciador de diálogo e este interagirá com o aluno, fomentando possíveis ações para que o equilíbrio seja re-estabelecido.

Desta forma, foram criados alguns dispositivos para auxiliar na interação com o aluno.

Controlador

Após a entrada de texto ou voz através da interface de chat/voz é disparado um processo para o controlador, que gerenciará as os passos necessário para a construção da resposta da seguinte forma:

- Chama o módulo Tratador
- Chama módulo de gerenciamento de diálogo
- Consulta a banco de dados
- Monta resposta.

Neste processo a IV etapa é a efetivamente é feita dentro do módulo de controle, nele serão identificado recorrências de casos e acontecimentos anteriores recentes através do RBC (Figura 5) e de passados um mais distantes do histórico (Figura 6), analisando os casos similares ocorridos, as regras para formulação da resposta, e assim complementando a resposta a ser emitida através do *chat/voz* ao aluno.

Cod Caso	Peso ponderado	Tipo Variável Ambiente			Valor Variável Ambiente	Pré frase	Cod Resp
		Variável	Prioridade	Nome			

	Cod Resp	Complemento
+ 5	1	
+ 4	2	
+ 3	3	
+ 2	4	
- 2	5	
- 3	6	
- 4	7	
- 5	8	

Figura 5 - Estruturação dos casos de diálogo

Histórico									
				Fala					
							Dialogo		
Nº_dialogo	Data	hora	Assunto	Nº_fala	Sequencia	Falador	Nº_ato_fala	Texto	Anotação
1	22/11	14:00	Lixo	1	1	Maga	1	A cidade está com muito lixo!	
1	22/11	14:00	Lixo	1	1	Maga	2	O que pode ser feito?	
1	22/11	14:00	Lixo	2	1	Usuário	1	vou construir um aterro sanitário.	

Figura 6 - Banco de Dados Histórico de Diálogos

O campo Anotação será preenchido após uma análise do diálogo por profissionais (professores, pedagogos, psicólogos, entre outros), objetivando auxiliar futuramente a evolução da personalidade do agente Maga Vitta.

Para a formulação das respostas do agente, foram necessários entendimentos sobre Processamento de Linguagem Natural (PLN), embasando o desenvolvimento das respostas para as conversações. Processamento de Linguagem e fala e Diálogo e Agentes de Conversação [Jurafsky e Martin, 2009], as quais tratam de multifuncionalidade, multidimensionalidade, anotação multidimensional, especificações do ato de diálogo, interação de voz em jogo [Bunt, 2005; Petukhova e Bunt, 2009], Fala, Conversação e Agentes [Harris, 2005].

Para a implementação em java, pesquisamos uma plataforma que suportasse nossas necessidades de estruturação de casos e escolhemos a *jCOLIBRI - CBR Framework* (<http://sourceforge.net/projects/jcolibri-cbr/>). Essa plataforma é um software livre que está sendo testado no projeto java do Città e esperamos que seja leve e estável o suficiente para mantê-la com os demais componentes.

Tratador

É o módulo responsável por buscar dentre o texto transcrito as informações para analisar o texto e o contexto do diálogo, para essa análise foram utilizadas:

Banco de dados com palavras chaves e contexto para similaridades das palavras chaves para cada produzível, como por exemplo: palavra chave: Água. Contextos: sede, molhado, lago, rio, arroio, torneira.

Sua concepção busca utilizar a Estrutura Dinâmica de Interpretação *Dynamic Interpretation Theory* (DIT, Bunt, 2011; 2005).

DIT distingue as seguintes 10 dimensões (para discussão e justificação ver Petukhova & Bunt 2009 e Bunt 2011):

1. Tarefa / Atividade: Atos de diálogo cujo desempenho contribui para executar a tarefa ou atividade subjacente ao diálogo;
2. Auto-Feedback: atos diálogo que fornecem informações sobre o processamento do locutor do enunciado anterior;
3. Allo-Feedback: diálogo atos utilizados pelo alto-falante para expressar opiniões sobre o destinatário processamento da declaração anterior, ou que solicitam informações sobre essa processamento;

4. Contact Management: Atos de diálogo para o estabelecimento e manter contato;
5. Turn Management: o diálogo atos em causa com agarrando, mantendo-se, dar ou aceitar o papel do remetente;
6. Time Management: o diálogo atos de sinalização que o falante precisa de um pouco de tempo para formular sua contribuição para o diálogo;
7. Discourse Structuring: atos de diálogo para explicitamente estruturação da conversa, por exemplo, anunciar o ato de diálogo seguinte, ou propor uma mudança de assunto;
8. Own Communication Management: os atos de diálogo em que o orador edições da contribuição para o diálogo que ele está produzindo;
9. Partner Communication Management: o agente que realiza esses atos diálogo não tem o papel de orador, e assiste ou corrige o alto-falante na formulação de uma contribuição para o diálogo;
10. Social Obligations Management: o diálogo atos que cuidar das convenções sociais, tais como cumprimentos, desculpas, agradecimentos e saudações.

Feitas as análises e as informações são devolvidas para o módulo de controle.

Gerenciador de diálogo

A estrutura para compor os diálogos é mostrada na Figura 2 e possui o registro do acionamento do tipo de diálogo, conforme a ponderação fornecida pelo módulo de inferência. Também são mantidos registros da variável de ambiente responsável pela ativação do diálogo e a codificação da resposta para o caso; a resposta codificada é complementada segundo o grau de intensidade da variável de ambiente na escala (-5 até +5) define os diferentes tipos de respostas a serem utilizados e auxilia na definição do comportamento do agente no diálogo, diferenciando a incisividade das falas e/ou textos, conforme a gravidade do problema detectado. Esse conjunto de parâmetros comparativos permitem a formulação da resposta.

Finalizando com o construtor de respostas⁵ que fará a geração das frases a partir das mais simples como cumprimentos e agradecimentos até a formulação de respostas elaboradas conforme figura 7.

⁵ Construtor de resposta – fornecerá o log de casos ocorridos

Produzíveis	Pré frase	Intensidade	Complemento
Água	Frase 1	+ 5	
		+ 4	
		+ 3	
		+ 2	
		- 2	
		- 3	
		- 4	
		- 5	
	Frase 2	+ 5	
		+ 4	
		+ 3	
		+ 2	
		- 2	
		- 3	
		- 4	
		- 5	

Figura 7 - Proposta de elaboração de respostas.

Referências

Bunt, H. (2011), Multifunctionality in dialogue. *Computer, Speech and Language*, 25, pages 225-245.

Bunt, H. (2011), The semantics of dialogue acts. In *Proceedings of the 9th International Conference on Computational Semantics (IWCS-9)*, Oxford, January 12-14, pages 1-13.

Bunt, H. (2005), A Framework for Dialogue Act Specification. In *Proceedings of 4th Joint ISO-SIGSEM Workshop on the Representation of Multimodal Semantic Information*, Tilburg, January 2005.

- Cassel, J. (2001), Embodied Conversational Agents: Representation and Intelligence in User Interface. In *AI Magazine*, 22(3), pages 67-83, AAAI Press.
- García, P.; Amandi, A.; Schiaffino, S.; Campo, M. (2007), Evaluating Bayesian networks precision for detecting student learning styles. In *Computers & Education* 49, pages 794–808, Elsevier.
- John, Oliver P. (1999), The "Big Five" factor taxonomy: Dimensions of personality in the natural language and in questionnaires. In: *Handbook of personality: Theory and research*. Edited by Lawrence A. Pervin, Guilford.
- Jurafsky, D. e Martin, J. H. (2009), *Speech and Language Processing*, Prentice-Hall, 2nd edition.
- Müller, D. N.; Oliveira, O. L. B. de; Remião, J. A. A.; Silveira, P. D.; Martins, M. A. R.; Axt, M. (2009), Virtual Cities as a Collaborative Educational Environment. In *Education and Technology for a Better World*, pages 112-120, Springer.
- Pelachaud, Catherine. (2001) Contextually Embodied Agents. In: *Deformable Avatars*. Edited by Magnenat-Thalmann, N. Thalmann, D., Kluwer Publishers.
- Petukhova, V. e Bunt, H. (2009). The independence of dimensions in multidimensional dialogue act annotation. In *Proceedings... NAACL HLT*, pages 197-200.

PROJETO ASSISTIVA

Processo 580470/2008-3

Relatório de Andamento

Maio/2009 a março/2011

Resumo:

O projeto ASSISTIVA - Assistente Inteligente Vocal para Maga Vitta visa o desenvolvimento computacional pela empresa CONEXUM de um agente inteligente para assistência às crianças usuárias do jogo educacional e simulador de cidades Città, desenvolvido pelo LELIC (Laboratório de Estudos em Linguagem, Interação e Cognição da Faculdade de Educação da UFRGS). A Maga Vitta será uma assistente que dialogará com o usuário, relatando questões sobre a maquete virtual em desenvolvimento. Com a tecnologia desenvolvida através deste projeto, pretende-se a futura criação de outros assistentes virtuais para dispositivos móveis e outros sistemas de jogo ou monitoramento.

Edital MCT/SETEC/CNPq N ° 67/2008 - RHAE - Pesquisador na Empresa

Introdução

O presente relatório é motivado pela necessidade de prorrogação do cronograma e das bolsas de projeto por mais 7 meses a contar da data de término prevista. Acredita-se que isso é viável dada a disponibilidade dos bolsistas e do saldo existente na conta do projeto.

Iniciado oficialmente em 29/04/2009, o Projeto Assistiva contou com problemas para começar com os bolsistas previstos. Havia candidatos já previstos em fevereiro de 2009, quando o projeto foi aprovado. Porém, como a autorização para contratação ocorreu em maio, foi necessário novo processo de seleção. Disso resultou que os primeiros bolsistas ITI foram contratados em junho (dois meses após a autorização inicial) e a equipe foi completada após 5 meses da autorização de implementação de bolsas. Em outras palavras, o projeto levou efetivamente 6 meses para iniciar, devido à falta de candidatos para as bolsas.

A contratação mais difícil de efetivação foi de bolsista com graduação, visto que a bolsa equivale a um terço do salário de mercado. Como se exige dedicação exclusiva, os candidatos recusavam-se a desfazer-se de outras formas de remuneração. Este foi o maior empecilho para contratação da bolsa DTI.

O mercado de trabalho foi o que motivou a substituição dos bolsistas ITI um ano depois do início do projeto. Os atraentes salários da área de informática forçaram-nos a fazer nova seleção de bolsistas, o que causou mais um mês de atraso em nosso cronograma.

Certamente consideramos positivo o fato de nossos bolsistas estarem encaminhando seu futuro profissional, porém causa-nos agora a necessidade de prorrogação em 7 meses de nosso projeto. Devido ao fato de haver saldo suficiente para a manutenção das bolsas por este período, e concordância dos atuais bolsistas na permanência no projeto, decidimos por bem solicitar a prorrogação, cujo relatório de andamento agora passamos a descrever. Os relatórios individuais dos bolsistas foram colocados ao final deste relatório geral de desenvolvimento.

Desenvolvimento do Projeto Assistiva

No primeiro mês efetivo de projeto (junho/2009), os dois bolsistas selecionados, Otto Oliveira e Douglas Cunha, foram destinados ao desenvolvimento de suas atividades, conforme previsão no projeto para atuarem, respectivamente, no módulo do jogo e da voz. Otto, por ser ex-bolsista do Projeto Civitas, já conhecia o jogo Città, no qual a agente inteligente está sendo inserida. Nesta primeira fase, Otto realizou diversas melhorias em relação à usabilidade do jogo, relativa às construções e modelagem do terreno do mapa. Douglas foi orientado a realizar um levantamento acerca dos sistemas em software livre para processamento de voz, enquanto a equipe da Conexum desenvolve um reconhecedor de fala contínua, com base no reconhecedor de comandos de voz já existente.

No segundo mês (julho/2009), selecionamos um mestre em Computação, Leonivan Moraes, uma vez que não encontrávamos um doutor para a vaga da bolsa SET. Leonivan rapidamente compreendeu os objetivos do projeto e passou de imediato a aprofundar-se nas investigações dos comportamentos da agente inteligente, Maga Vitta. Em reuniões com a equipe do LELIC (Laboratório de Estudos em Linguagem, Interação e Cognição) da UFRGS, parceira do projeto, decidiu-se por realizar toda a abordagem inicial da agente em torno da temática do meio ambiente. A partir destes aspectos foram idealizados os seguintes indicadores de ambiente: *espaço, esgoto, atendimento médico, lixo, energia, alimentos e água*. Estes indicadores foram e são as linhas mestras para o desenvolvimento dos sistemas da equipe Assistiva.

Assim, a partir dos indicadores, foram implementados simuladores para cada um deles no jogo Città. Os diálogos escritos e de voz foram e são projetados em torno do contexto destas temáticas. O próprio comportamento da agente Maga Vitta, enfim, foi concebido com base nas premissas dos indicadores de meio ambiente.

Finalmente completamos a equipe apenas em setembro/2009, quando selecionamos Alexsandro Anjos para bolsista DTI. Ele entrou no projeto para tratar dos diálogos da Maga, e seu estudo e desenvolvimento foi sempre dentro desta linha de trabalho.

A partir daí o projeto evoluiu como esperado. Em março/2010 o bolsista Otto foi chamado para trabalhar em empresa de jogos digitais e indicou para seu lugar Taiuã Pires, que deu continuidade aos recursos do Città. Taiuã trabalhou principalmente no sistema de banco de dados para armazenamento e avaliação das ações e movimentos do usuário do jogo.

Em junho/2010, Douglas também seguiu um caminho semelhante ao de Otto, saindo para trabalhar em empresa da área de microeletrônica. No seu lugar foi selecionado Lisardo Kist, que prosseguiu no desenvolvimento a partir do software livre Julius, para reconhecimento de voz.

Desde então tivemos as seguintes contribuições:

- Para o jogo Città: melhoria da usabilidade, criação de variáveis de ambiente, arquivos de histórico de construções, jogadas e texto dos usuários. Bolsistas envolvidos: Otto (anterior) e Taiuã (atual).

- Para o reconhecimento de voz: contribuição a nível nacional, tratando conjuntamente com a equipe do projeto FalaBrasil (UFPA), que também desenvolve com o sistema Julius. Até o momento foram criadas algumas gramáticas de teste do sistema. Bolsistas envolvidos: Douglas (anterior) e Lisardo (atual).
- Para a modelagem cognitiva da agente inteligente: foi realizada pesquisa no sentido de selecionar as variáveis de ambiente mais significativas, relacionando-as às necessidades de diálogo. Foi também criado um protótipo de raciocínio bayesiano para atuar como máquina de inferência da decisão da agente. Bolsista envolvido: Leonivan.
- Para o diálogo da agente: foram criados diversos recursos de banco de dados para dar suporte ao histórico de conversação, identificação dos atos de fala, gerenciamento de diálogo e análise das dimensões dos atos de fala. Bolsista envolvido: Alexsandro.

Cabe ressaltar que, apesar de termos bolsistas de referência em cada módulo, são feitas reuniões periódicas para troca de experiências, com avaliação mútua do desenvolvimento. Os sócios da Conexum supervisionam todo o processo, tendo o coordenador do projeto maior proximidade com o desenvolvimento dos bolsistas.

Também são realizadas reuniões eventuais com os demais pesquisadores do Lelic, estando mais envolvidos Carla Freitas, Andrea Konzen, Paloma Dias e Márcio Martins. Através desta equipe temos contatos com as escolas e realizamos consultas e visitas para reavaliação periódica dos conceitos e do andamento do projeto.



Nas fotos acima, reunião periódica do primeiro time do Assistiva, juntamente com Paloma Dias. Ao centro, Otto faz anotações sobre uso do Città, observando aluna. À direita, visita à Escola Municipal Castro Alves, na zona rural do município de Mato Leitão (RS), em encontro com a turma da professora Solange Stülp.



Nestas outras fotos, avaliação do Città novamente com a turma da Escola Castro Alves (à esquerda) e na Escola Municipal Santo Antônio de Pádua, também em Mato Leitão, com a turma da professora Simone Maggioni.

A partir deste momento ainda faltam alguns itens a concluir para fechamento do projeto:

- Testes de visualização e bate-papo da Maga Vitta na interface do Città.
- Implementação em Java do raciocínio bayesiano da Maga Vitta.
- Testes de modelo de reconhecimento de voz contínua.
- Testes de diálogo, para aperfeiçoamento do modelo de análise.
- Novas interações e testes nas escolas.

Para tanto, são solicitados 7 meses de prorrogação no Projeto Assistiva, sendo essencial para bom acabamento de nossas pesquisas.

PROJETO ASSISTIVA

Processo 580470/2008-3

Plano de Trabalho de Prorrogação

Maio/2011 a novembro/2011

Resumo:

O projeto ASSISTIVA - Assistente Inteligente Vocal para Maga Vitta visa o desenvolvimento computacional pela empresa CONEXUM de um agente inteligente para assistência às crianças usuárias do jogo educacional e simulador de cidades Città, desenvolvido pelo LELIC (Laboratório de Estudos em Linguagem, Interação e Cognição da Faculdade de Educação da UFRGS). A Maga Vitta será uma assistente que dialogará com o usuário, relatando questões sobre a maquete virtual em desenvolvimento. Com a tecnologia desenvolvida através deste projeto, pretende-se a futura criação de outros assistentes virtuais para dispositivos móveis e outros sistemas de jogo ou monitoramento.

Edital MCT/SETEC/CNPq N ° 67/2008 - RHAE - Pesquisador na Empresa

Proposição

Conforme relatório de atividades do projeto, foram apresentados os desenvolvimentos do projeto até o momento. A partir de abril/2011 ainda faltariam alguns itens a concluir para fechamento do Projeto Assistiva:

- Testes de visualização e bate-papo da Maga Vitta na interface do Città.
- Implementação em Java do raciocínio bayesiano da Maga Vitta.
- Testes de modelo de reconhecimento de voz contínua.
- Testes de diálogo, para aperfeiçoamento do modelo de análise.
- Novas interações e testes nas escolas.

Para tanto, são solicitados 7 meses de prorrogação no Projeto Assistiva, sendo essencial para bom acabamento de nossas pesquisas. Esta prorrogação inclui as bolsas, uma vez que existe saldo suficiente para sua manutenção.

O cronograma de trabalho para o período de prorrogação é o que segue:

1. implementação em java da modelagem cognitiva da assistente (bolsista Leonivan)
2. testes dos diálogos (bolsista Alexsandro)
3. testes de reconhecimento e síntese de voz (bolsista Lisardo)
4. beta-testes da assistente no jogo (bolsista Taiuã)
5. revisão da modelagem cognitiva (todos)
6. revisão dos diálogos (todos)
7. ajustes de síntese e reconhecimento de voz (todos)
8. visita e testes nas escolas (todos)
9. desenvolvimento de relatórios (todos)
10. publicações e seminários do projeto (todos)

Cronograma:

Atividade	maio	junho	julho	agosto	setembro	outubro	novembro
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							

Além da conclusão do desenvolvimento, o projeto está contemplando a publicação em anais de eventos e periódicos. Como exemplos de potenciais publicações, temos o Workshop de Software Livre 2011, Simpósio Brasileiro de Games, entre outros.

Maga Vitta – agente conversacional aplicado ao jogo educacional Città

Andrea Aparecida Konzen¹², Otto Lopes Braitback de Oliveira³, Lisardo Sallaberry Kist⁴,
Alexsandro Rosa dos Anjos⁵, Leonivan Moraes⁵, Carla Inez Lima Freitas⁶, Daniel Nehme
Müller⁵ e Margarete Axt⁶

¹Departamento de Informática - Universidade de Santa Cruz do Sul (UNISC)
Av. Independência, 2293- 96640-000 - Santa Cruz do Sul – RS - Brasil

²Programa de Pós-Graduação em Informática na Educação (PGIE) - Universidade Federal do Rio Grande do Sul (UFRGS)
Av. Paulo Gama, 110 - 90040-060 - Porto Alegre - RS - Brasil

³Curso de Jogos Digitais, Universidade do Vale do Rio dos Sinos (UNISINOS)
Av. Unisinos, 950 - 93.022-000 - São Leopoldo - RS - Brasil

⁴Curso de Ciência da Computação, Instituto de Informática - Instituto de Informática –
Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

⁵Conexum Ltda., Centro de Empreendimentos em Informática - Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 - 91.501-970 - Porto Alegre - RS - Brasil

⁶Laboratório de Estudos em Linguagem, Interação e Cognição - Faculdade de Educação - Pós-Graduação em Educação (UFRGS)
Av. Paulo Gama, 110 - 90040-060 - Porto Alegre - RS - Brasil

{andreakon,maaxt@ufrgs.br},{ottolb,lisardorkist,leonivanmoraes@gmail.com},
{alex,daniel@conexum.com.br},{carlalima@uol.com.br}

Abstract. *The ASSISTIVA Project aims to accomplish the free software development, by CONEXUM Company, of an intelligent agent to assist children and users of educational game cities simulator Città, developed by LELIC (Laboratory of Studies in Language, Interaction and Cognition at the Faculty Education from UFRGS). The agent Maga Vitta is an assistant that performing a dialogue with the student, reporting issues on the virtual city model in development. With the technology developed through this project, it is intended for establishment other virtual assistants and other mobile gaming or monitoring systems.*

Resumo. *O Projeto ASSISTIVA – Assistente Inteligente Vocal para Maga Vitta visa o desenvolvimento em software livre pela empresa CONEXUM de um agente inteligente para assistência às crianças usuárias do jogo educacional e simulador de cidades Città, desenvolvido pelo LELIC (Laboratório de Estudos em Linguagem, Interação e Cognição da Faculdade de Educação da UFRGS). O agente Maga Vitta é um assistente que dialoga com o aluno, relatando questões sobre a maquete virtual em desenvolvimento. Com a tecnologia desenvolvida através deste projeto, pretende-se criar outros assistentes virtuais para dispositivos móveis e outros sistemas de jogo ou monitoramento.*

Introdução

O projeto ASSISTIVA – Assistente Inteligente Vocal para Maga Vitta propõe a integração de diferentes áreas de pesquisa para o desenvolvimento de um assistente virtual de diálogo falado. O ASSISTIVA insere-se no Projeto CIVITAS – Cidades Virtuais com Tecnologias para Aprendizagem e Simulação, em andamento no LELIC (Laboratório de Estudos em Linguagem, Interação e Cognição), o qual

possui um jogo educacional chamado Città, de onde parte a concepção do agente inteligente Maga Vitta. O projeto CIVITAS é uma abordagem de ensino e pesquisa que gira em torno do desenvolvimento de maquetes de cidades projetadas por alunos das séries iniciais [Axt 2008].

Dentro deste contexto, o jogo Città insere-se como uma maquete digital que permite um diferente nível de interação entre os alunos. Città é um jogo educacional colaborativo, que visa a construção coletiva, voltada para a ação cooperativa e negociada. No jogo, o assistente virtual, Maga Vitta, foi projetado para ser um agente conversacional que trabalha como regulador ecológico. O agente interage com o aluno através de alertas, de forma a promover a problematização de situações críticas que poderiam trazer prejuízos à população da cidade virtual.

A construção deste agente deu-se pelo acréscimo de módulos ao jogo, permitindo o monitoramento e intervenção a partir das ações do aluno. Nas seções seguintes são apresentadas as concepções do agente e sua implementação e, ao final do artigo, são apresentadas algumas conclusões e perspectivas futuras.

Agentes conversacionais

[Cassel 2001] propõe um modelo conversacional, conhecido como modelo FMBT (*Functions, Modalities, Behaviors, Time*), que incorpora as principais características presentes numa conversação. O modelo FMBT é caracterizado por: (1) funções proposicionais e funções interacionais que correspondem, respectivamente, ao conteúdo da conversa (inclui fala, entonação da voz e gestos corporais) e as *deixas* que regulam o processo de conversação, as quais compreendem tanto comportamentos sonoros quanto não-sonoros; (2) modalidades de comunicação, responsáveis por construir as funções proposicionais e interacionais, podem ser verbais e não-verbais; (3) comportamentos na conversação, onde a mesma função de comunicação pode assumir vários significados, por exemplo, baixar a cabeça pode denotar concordância, saudação ou ênfase em uma palavra; (4) tempo e sincronização da conversação, na medida em que a duração excessiva de emissão da fala ou de um sinal pode, eventualmente, indicar diferentes sentimentos como ansiedade, nervosismo e tédio, e atrasos na resposta podem ser entendidos como não cooperação, desatenção, etc.

Conforme [Longhi 2004], pode-se dizer que, um agente inteligente é um agente conversacional (*conversational agent*) quando possui a capacidade de manter uma conversa “humana”, face-a-face com o usuário humano, através do reconhecimento da fala e dos gestos envolvidos durante uma conversação. O corpo pode ser um objeto animado não-humanóide, uma cabeça falante ou uma representação humana completa em 3D que incorpora atributos humanos como sentimentos e linguagem. Quando um agente conversacional faz parte de um ambiente inteligente de aprendizagem, este deve possuir *expertise* em um domínio de conhecimento e compreender as necessidades do usuário para melhor direcioná-lo nos conceitos ou nas tarefas mais complexas.

Um *Embodied Conversational Agent* (ECA), que segue a linha dos agentes conversacionais, é um agente autônomo inteligente, que tem uma representação gráfica (*embodied*) em um mundo virtual e é capaz de manter alguma forma de conversação com o usuário [Pelachaud 2001]. O agente “fala” (conversa) com o usuário e tem uma forma gráfica personificada no mundo virtual. Além das representações verbais, esses agentes possuem a capacidade de reconhecer e reproduzir uma comunicação não-verbal.

Maga Vitta

Durante a construção da cidade virtual, o aluno conta com auxílio de um agente que observa suas ações no ambiente virtual. O agente Maga Vitta não intervém na construção da cidade, mas está sempre atento para indagar, aconselhar, informar e provocar reflexões sobre os processos de

construção da cidade.

Vitta incorpora a arquitetura padrão de agente autônomo inteligente e as propriedades de um ECA. Tais propriedades são essenciais para que este agente consiga expressar o que “deseja” falar, bem como para que construa o plano de interação com o aluno, de forma a se revestir de credibilidade. A capacidade cognitiva do agente será baseada em conhecimento e decisão. A capacidade emotiva e de personalidade, que são dois tipos de estados afetivos, está baseada no modelo de emoções, o modelo OCC, focado na teoria do *appraisal* e concebido para ser implementado computacionalmente [Ortony, Clore e Collins 1988] e, no modelo de representação de traços de personalidade proposto por [John 1999], chamado de *Big Five*, que foi desenvolvido na área de psicologia com o objetivo de identificar a personalidade das pessoas, ou seja, expressar como cada traço de personalidade influencia em cada emoção. Como esta fase está em desenvolvimento inicial, outros trabalhos serão estudados e a análise destes modelos citados serão aprofundadas.

Além disso, Maga Vitta é um agente que possui como padrão interno estar eternamente preocupado com a ecologia: ao memorizar ações contra o equilíbrio ecológico, adverte sobre possíveis tragédias ambientais, fazendo o controle de quatro aspectos: natureza (poluição do ar, água e terra), água (contaminação, desperdício e abastecimento de água), energia (iluminação e uso de aparelhos elétricos) e população (monitoramento da quantidade de habitantes e sua relação com a infra-estrutura disponível). Em sua interação com o jogo Città, Maga Vitta tem um motor de inferência que trabalha em conjunto com um módulo de conversação, que pode interagir também com voz dentro do ambiente de construção da cidade virtual.

O jogo Città

Città é um jogo educacional colaborativo, que tem por objetivo a construção coletiva, voltada para a ação cooperativa e negociada. Além disso, dá apoio pedagógico na construção de conhecimento dos alunos do 4º ano do ensino fundamental, no que se refere ao desenvolvimento de capacidades que lhes permitam pensar, construir e organizar uma cidade virtual a partir da realidade infantil, incorporando seus conhecimentos através dos conteúdos oferecidos no aprendizado dos sistemas que regulam uma cidade e suas relações [Müller 2009].

O objetivo principal do Città é que o aluno aprenda através da construção e da organização de uma cidade. Para que o mesmo passe por diversas situações que proporcionem este aprendizado é necessário completar tarefas fornecidas pelos educadores ou pelo próprio jogo. Nessas tarefas, os alunos terão que satisfazer a população e ter uma sustentabilidade que respeite o meio ambiente.

Na figura 1 é apresentado um exemplo da atual interface do jogo, que está sendo remodelada após testes de usabilidade nas escolas. Além da usabilidade, também são coletados dados que realimentam o projeto em nível dos estudos teóricos e da engenharia do software.

Dentro do jogo, o projeto do agente Maga Vitta exigiu o desenvolvimento da simulação de variáveis de ambiente, além dos módulos de banco de dados, de voz, de conversação e de inferência da agente. As variáveis de ambiente são relativas ao uso dos elementos naturais e seu desgaste, devido ao consumo e poluição em função do tempo e das ações do aluno.



Figura 1 - Interface do jogo Città (esq.) e avaliação de usabilidade (dir.).

Uma vez calculadas as variáveis de ambiente, estas são lidas pela inferência do agente que dispara um diálogo com o aluno. O diálogo pode ser realizado através de texto ou de voz. Ao responder, o aluno interage com o módulo de *chat/voz*, sendo que, o texto recebido é analisado pelo módulo de diálogo, que formula uma resposta com base nas consultas ao seu banco de dados próprio e ao módulo de inferência do assistente. A partir disso, é fornecida uma resposta ao aluno. Todas as ações que partem do aluno possuem este tipo de tratamento, por exemplo, o caso do aluno chamar o agente Maga Vitta para conversar. Esse processo está ilustrado na figura 2.

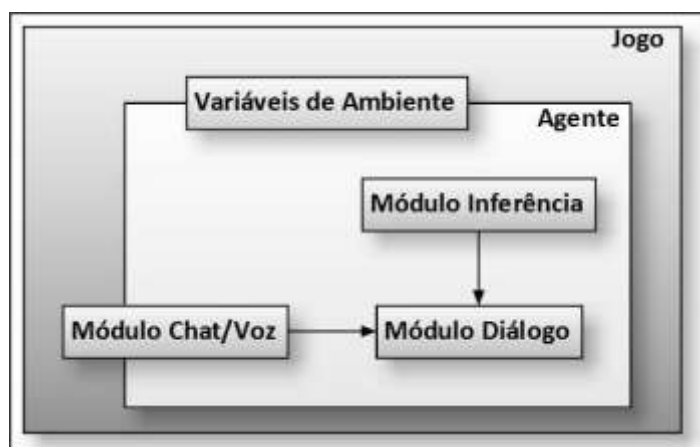


Figura 2 - Esquema de dependências dos módulos

Nas seções seguint são detalhados os módulos de diálogo, de inferência e de *chat/voz*.

Módulo de Diálogo

O módulo de diálogo tem significativa importância tanto para o jogo quanto para a integração das áreas envolvidas no projeto, pois para a construção das frases ocorre um planejamento conjunto entre a equipe do ASSISTIVA e o grupo CIVITAS, para que as respostas do agente Maga Vitta não sejam dadas diretamente, mas sim, que estimulem o raciocínio dos alunos.

O processo começa com o Controlador enviando o texto recebido através da fala ou da escrita para o Tratador, com o propósito de identificar o assunto que está sendo conversado e qual a necessidade solicitada (veja Figura 3). Em paralelo a esse processo, o Controlador dispara outro processo para o Gerenciador de Diálogo, que fará o controle informando se o diálogo está aberto, se é o mesmo assunto durante o diálogo e, se já houve ocorrência do mesmo problema. Para verificação dessas ocorrências anteriores utilizou-se a abordagem de Raciocínio Baseado em Casos (RBC).

Desta forma, foi criado um módulo de inferência para o desenvolvimento do processo de tratamento usado para analisar o texto e o contexto do diálogo, do gerenciador de diálogo que controla ocorrências de diálogos paralelos (assuntos fora de contexto ou prioridade) e ainda

recorrências de casos e históricos de acontecimentos anteriores através do RBC e da estrutura do Banco de Dados para armazenamento das falas.

A partir disso, chega-se à formulação da resposta. O módulo chamado de responde utiliza a consulta aos históricos sobre o assunto, os últimos casos similares ocorridos e as regras para formulação da resposta, resultando na resposta completa a ser dada através do *chat* para o aluno. A Figura 3 mostra as etapas descritas para construção do diálogo onde a frase entra no sistema através da interface do *chat* e o controlador que dispara processos para a construção do diálogo.

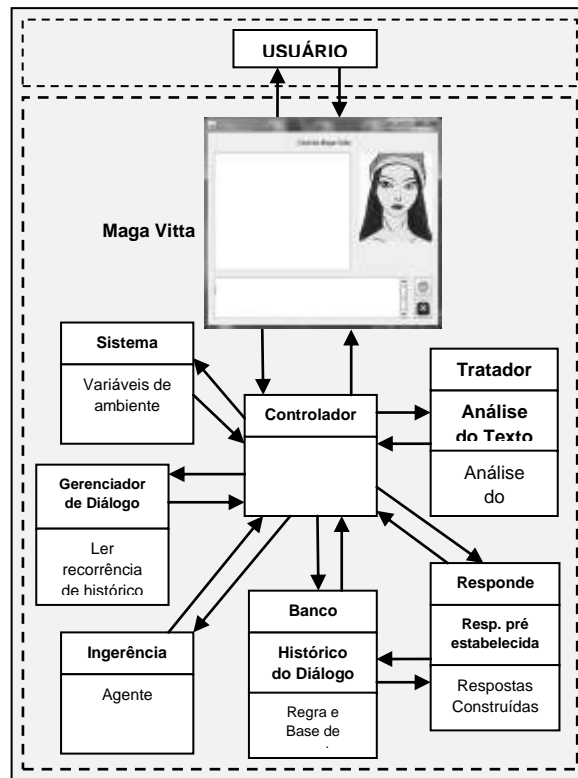


Figura 3 – Diagrama de construção da frase (Maga Vitta)

Para que esse processo de formulação da resposta obtenha resultados, foram necessários estudos em Processamento de Linguagem Natural (PLN) para entender o processo de desenvolvimento das respostas para as conversações [Jurafsky e Martin 2009], as quais tratam de multifuncionalidade, multidimensionalidade, anotação multidimensional, especificações do ato de diálogo e interação de voz em jogo [Bunt 2005; Petukhova 2009].

Construção dos Diálogos

Para criar diálogos, foi utilizado como base o estudo do Método Clínico Experimental de Jean Piaget, que foi um método investigativo, usado pelo autor como seu instrumento de diagnóstico e o que lhe proporcionou todas as descobertas da Epistemologia Genética, sendo que, com este método, investigou como as crianças percebem, agem, sentem, pensam a natureza e tudo ao seu redor. Sua essência é o objetivo investigativo do experimentador e a sua interação com o entrevistado.

Através do estudo deste método, a ideia é possibilitar que a interação do agente Maga Vitta e os alunos construtores da cidade virtual, através do Cittá, ocorra por meio de interferências criativas do agente, durante o jogo, seja pela solicitação do aluno ou pela necessidade que surgir a partir das ações do aluno sobre o jogo.

O agente virtual procurará, através do diálogo, não influenciar diretamente a tomada de decisão do aluno. A intenção é construir um agente que faz a intervenção, que provoca e, desta forma, cria situações que desacomodem estruturas e possibilitem avanços através da reflexão dos alunos

sobre sua ação no jogo.

Os alunos são crianças com nove anos de idade, aproximadamente, que frequentam o 4º ano do Ensino Fundamental das escolas participantes do Projeto CIVITAS. Segundo [Piaget 1993], no período dos nove aos onze anos, aproximadamente, pode-se observar que a criança já possui uma organização assimilativa rica e funcionando em equilíbrio com um mecanismo assimilativo. A criança encontra-se na fase das operações concretas, que é uma etapa marcada por grandes aquisições intelectuais. As formas de explicação para os fenômenos que ocorrem a sua volta deixam de ser egocêntricas e passam a ser por identificação, é um estágio intermediário entre a heteronomia moral e a autonomia plena.

Considerando, portanto, a proposta de que o diálogo do agente Maga Vitta com os alunos possibilite uma interação criativa, o estudo toma como base o Método Clínico Piagetiano e os estudos deste epistemólogo sobre a formação dos mecanismos mentais da criança.

Módulo de Inferência

O agente Maga Vitta é a consciência do jogo Città, promovendo a regulação do sistema ecológico. A sua proposta não é realizar a orientação direta, mas provocar ações internas no aluno de forma a fazê-lo a se preocupar com a situação indicada no momento do jogo. Dependendo da gravidade da situação, indicada pelo nível de equilíbrio da variável de ambiente e, ainda da prioridade do elemento que está sendo considerado, é indicado ao módulo de diálogo a importância e nível de expressividade que o agente terá que executar.

As variáveis de ambiente são relativas aos produzíveis na cidade virtual, correspondendo ao quanto é gasto por objeto construído no jogo em relação ao tempo. Os objetos podem ser casa, edifício, fazenda, hospital, igreja, escola, prefeitura, entre outros. As variáveis de ambiente definidas até o momento são espaço, esgoto, atendimento médico, energia, alimentos/água. Esta é, respectivamente, a ordem inversa de prioridade de análise, sendo o espaço menos significativo e, alimentos e água os mais importantes. No trabalho de [García 2007] pode ser encontrado um exemplo de uso semelhante, onde variáveis de software educacional foram utilizadas para inferência de categorias de estudantes através de redes bayesianas. Na abordagem do ASSISTIVA, utiliza-se a inferência para determinação do caráter do diálogo que será travado com o aluno no jogo.

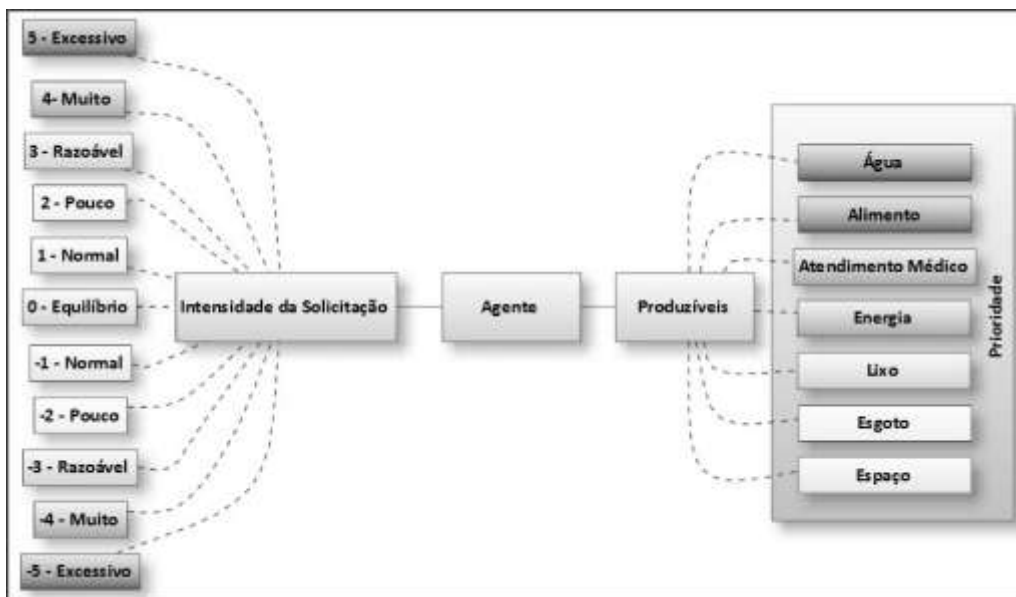


Figura 4 – Mapa mental para tomada de decisão da Maga Vitta

Na etapa atual do projeto está sendo avaliada a possibilidade de utilizar um raciocínio bayesiano juntamente com o esquema de Raciocínio Baseado em Casos (RBC) elaborado para o

diálogo da Maga Vitta, apresentado na seção 4.1. Neste sentido, pode ser citado o trabalho de [Pavón 2009], que utiliza a inferência bayesiana como auxiliar na busca do conjunto de soluções de um RBC. Desta forma, foi implementado o modelo bayesiano de teste (Figura 5), a título de protótipo para o mapa mental da Maga Vitta apresentado na Figura 4.

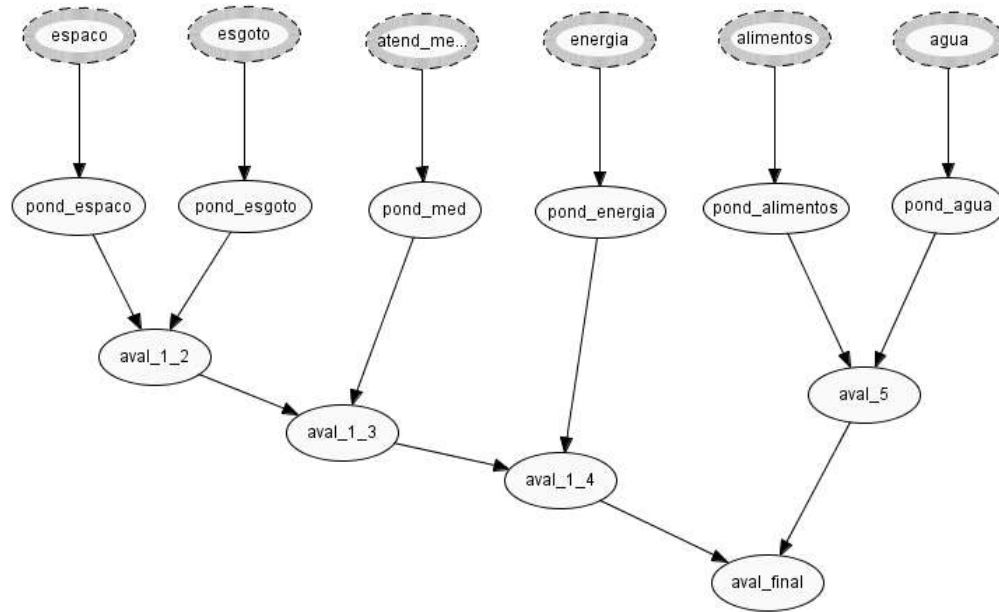


Figura 5 – Modelagem do protótipo de inferência bayesiana

No esquema representado na Figura 5, os nodos superiores correspondem às variáveis de ambiente definidas no modelo mental do agente Maga Vitta. A cada variável está associado um conjunto de ponderações que formam a segunda linha de nodos. Elas correspondem às variáveis linguísticas *pouco*, *razoável*, *muito* e *excessivo*. Estas variáveis foram estabelecidas para simplificação do processo de tomada de decisão, o qual gera tabelas extensas para estabelecimento das avaliações que formam os nodos inferiores. As avaliações ocorrem em cascata devido à sua relação com as prioridades das variáveis de ambiente.

Módulo de Chat/Voz

O módulo de voz consiste em um transcritor de voz para texto, que será enviado para o assistente do jogo. Para realizar a transcrição é utilizado o software livre Julius⁶.

Como mostra a Figura 6, o módulo de voz funciona recebendo um *stream* de áudio (arquivo ou sinal do computador) e retornando a transcrição na forma de *stream* de texto. O programa presume um modelo acústico e um modelo linguístico sobre qual o reconhecimento será realizado.

O modelo acústico trata-se de um modelo estocástico de cadeias ocultas de Markov sobre qual o reconhecimento é feito. O modelo linguístico é uma gramática formal especificada para aplicação. O texto reconhecido poderá ser acessado como variável do programa e utilizado na assistência ao aluno.

⁶ <http://julius.sourceforge.jp/>

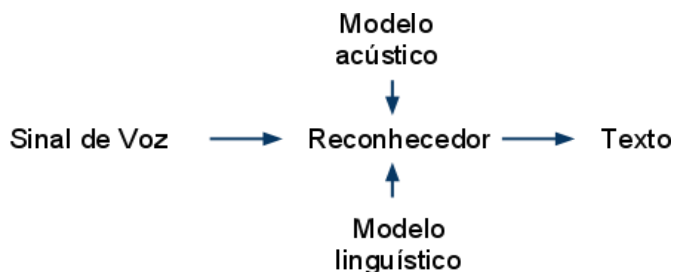


Figura 6 - Esquema funcional do módulo de reconhecimento de voz

As respostas processadas são sintetizadas de volta para o aluno. Já, no nível de síntese de voz, tem-se realizados testes com o software livre Mary Text To Speech⁷ e obtidos resultados satisfatórios.

A inclusão desses módulos ao jogo Città, enriquece a interação entre o aluno e o agente, facilitando o processo de aprendizagem e a troca de informações entre os mesmos. Além disso, torna a atuação desse agente mais próxima da realidade, tornando o jogo mais interessante do ponto de vista do aluno.

Conclusões

Como o Projeto ASSISTIVA está utilizando o jogo Città, o qual tem seu código, que é um software livre, hospedado no SourceForge⁸, na medida em que os testes dos protótipos forem consolidados, os códigos desenvolvidos para o assistente inteligente serão disponibilizados juntamente com o jogo. Maiores informações podem ser obtidas no sítio do ASSISTIVA⁹ na internet.

O desenvolvimento do agente Maga Vitta é realizado em conjunto com os bolsistas da Conexum vinculados ao projeto ASSISTIVA e os membros do LELIC, que atuam junto ao projeto CIVITAS. As reuniões são periódicas e enriquecem o processo de projeto e desenvolvimento de software.

Os módulos estão sendo testados em laboratório e a seguir serão levados para os testes de campo nas escolas, novamente realimentando e reavaliando o projeto do agente Maga Vitta. Neste sentido, encontram-se vários desafios relativos à organização dos diálogos e ao reconhecimento de voz.

Existe a necessidade de verificação da amplitude dos padrões de respostas, frente ao tipo de abordagem que os alunos criam ao travar os diálogos com a Maga Vitta. Além da linguagem, estão previstas ainda variações de humor e comportamento que serão modelados futuramente.

Quanto à voz, ainda estamos criando os modelos acústicos para trabalhar com maior acurácia dentro do universo do jogo e do contexto de discurso. Até o momento, apenas o diálogo escrito é respondido com segurança. Apesar dos investimentos nos recursos de voz, a prática aponta para maior uso da escrita, frente à eventual falta de dispositivos como *handsets* nas escolas. Mesmo assim, o recurso de voz está sendo explorado para seu posterior aperfeiçoamento em outros tipos de aparelhos.

Os testes realizados até o momento com a inferência bayesiana colocada no assistente apontam para a possibilidade de testes também com inferência fuzzy. Apesar da abordagem inicial com RBC e rede bayesiana, percebe-se que há espaço para uma investigação referente à lógica fuzzy, uma vez que tem-se regras que exigem decisões frente às escalas de incerteza. Testes futuros indicarão qual o melhor modelo para consolidação do agente.

⁷ <http://mary.dfki.de>

⁸ <http://civitas.sourceforge.net>

⁹ <http://assistiva.conexum.com.br>

A construção do assistente em vários módulos tem sido um grande desafio para a equipe e, percebe-se a falta de um elo condutor biologicamente plausível que possa integrar coerentemente as tecnologias utilizadas. Numa próxima etapa desta pesquisa, deverão ser apontados modelos compatíveis com estudos neuropsicológicos que deem suporte aos recursos, que agora são constatados como necessários ao processamento deste conjunto de módulos que colaboram entre si. Acredita-se que, uma arquitetura de sistema mais eficiente poderá ser gerada a partir desta abordagem.

O modelo de agente proposto no presente artigo é a tentativa de criação do assistente inteligente Maga Vitta para o jogo educacional Città, ambos concebidos dentro do Projeto CIVITAS, do LELIC/UFRGS. A Conexum Ltda., através do Projeto ASSISTIVA, entra como parceira deste software livre na produção do conhecimento de uma nova tecnologia de assistentes virtuais, com o intuito de promover o debate acadêmico e comercial deste novo campo que se abre na interação humem-computador. Espera-se que, com esta iniciativa de tecnologia aberta, sejam desenvolvidos trabalhos derivados, permitindo assim, a evolução da concepção de um assistente pessoal conversacional para uso nos mais diversos dispositivos eletrônicos.

Agradecimentos. O projeto Assistiva é financiado com o apoio do CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico. Agradecemos a toda equipe do Projeto CIVITAS, por sua assessoria a este projeto.

Referências

- Axt, M. e Martins, M. A. (2008) “Coexistir na diferença: De quando a formação em serviço pensa modos de habitar a sala de aula”, In: Múltiplas Alfabetizações e Alfabetismos, v. 1, pp. 133-158, editado por Iole Maria Faviero Trindade, Editora da UFRGS.
- Bunt, H. (2005) “A Framework for Dialogue Act Specification”, In: Proceedings of 4th Joint ISO-SIGSEM Workshop on the Representation of Multimodal Semantic Information, Tilburg, January 2005.
- Cassel, J. (2001) “Embodied Conversational Agents: Representation and Intelligence in User Interface”, In: AI Magazine, 22(3), pages 67-83, AAAI Press.
- García, P., Amandi, A., Schiaffino, S. e Campo, M. (2007) “Evaluating Bayesian networks precision for detecting student learning styles” In: Computers & Education 49, pages 794–808, Elsevier.
- John, Oliver P. (1999) “The ‘Big Five’ factor taxonomy: Dimensions of personality in the natural language and in questionnaires”, In: Handbook of personality: Theory and research. Edited by Lawrence A. Pervin, Guilford.
- Jurafsky, D. e Martin, J. H. (2009) “Speech and Language Processing”, Prentice-Hall, 2nd edition.
- Longhi, M. T., Nedel, L. P., Vicari, R. M. e Axt, M. (2004) “Especificação e Interpretação de Gestos Faciais em um Agente Inteligente e Comunicativo”, In: SBC – Simpósio de Realidade Virtual, São Paulo.
- Müller, D. N., Oliveira, O. L. B. , Remião, J. A. A., Silveira, P. D., Martins, M. A. e Axt, M. (2009), “Virtual Cities as a Collaborative Educational Environment”, In: Education and Technology for a Better World, pages 112-120, Springer.
- Ortony, A., Clore, G. L. e Collins, A. (1988), “The cognitive structure of emotions”, Cambridge University Press.

- Pavón, R., Díaz, F., Laza, R. e Luzón, V. (2009) “Automatic parameter tuning with a Bayesian case-based reasoning system. A case of study” In: *Expert Systems with Applications* 36, pages 3407–3420, Elsevier.
- Pelachaud, Catherine. (2001) “Contextually Embodied Agents”. In *Deformable Avatars*. Edited by Magnenat-Thalmann, N. Thalmann, D., Kluwer Publishers.
- Petukhova, V. e Bunt, H. (2009) “The independence of dimensions in multidimensional dialogue act annotation” In: *Proceedings... NAACL HLT*, pages 197–200, Boulder.
- Piaget, J. (1993) “Seis estudos de Psicologia”, Rio de Janeiro: Forense Universitária.



Documento de Projeto de Jogo:

Città Civitas

Escrito por: Otto Lopes Braitback, Anderson Santiago, Daniel Nehme Müller e Taiuã Pires

Versão # 1.0.6

Data: julho/2010

>>> *Este documento é baseado no formato original proposto por Chris Taylor/*
GameDev <<<

Índice

CITTÀ	81
CIVITAS	81
VERSÃO 0.1	85
VERSÃO 0.2	85
VERSÃO 0.5	85
VERSÃO 0.6	86
VERSÃO 0.7	87
VISÃO GERAL DO JOGO	87
DECISÕES DE PROJETO	88
Item #1	88
Item #2	88
Item #3	88
Item #4	88
Item #5	88
QUESTÕES SOBRE O PROJETO DO JOGO	88
Qual o objetivo principal?	88
O que o jogador controlará?	89
O que ele tem de diferente?	89
Perguntas referentes ao Città	89
RECURSOS DO JOGO	90
RECURSOS GERAIS	90
GAMEPLAY	90
Descrição Geral	90
Movimentação no Mundo	91
Inserção de edificações	92
Edição de histórias e maiores informações	92
REGRAS DO CITTÀ	92
EDIÇÃO DO MUNDO (MAPPA)	95
DESCRIÇÃO GERAL	95
DETALHES DO FUNCIONAMENTO	95
Funcionamento do Relevo	95
Funcionamento de construção de ruas	96
Funcionamento de edição de terreno	96
EDIÇÃO DE OBJETOS (EDITTORE)	96
DESCRIÇÃO GERAL	96
DETALHES DO FUNCIONAMENTO	96
O MUNDO DO JOGO	97
VISÃO GERAL	97
OBJETOS 3D DO MAPA	97
MAGA VITTA	99
DESCRIÇÃO GERAL	99

CICLO	99
INTERFACE DE JOGO	100
DESCRIÇÃO GERAL	100
DETALHES DA INTERFACE DO CITTÀ	101
DETALHES DA INTERFACE DO MAPPA	102
DETALHES DA INTERFACE DO EDITTORE	102
JOGO COM JOGADOR ÚNICO	103
DESCRIÇÃO GERAL	103
CARREGANDO E SALVANDO O JOGO	103
JOGO COM MÚLTIPLOS JOGADORES	103
DESCRIÇÃO GERAL	103
CARREGANDO E SALVANDO O JOGO	104
MÚSICA E EFEITOS SONOROS	104
DESCRIÇÃO GERAL	104
MÚSICAS	104
SOM	105
CRONOGRAMA	106
TESTES	106
TESTE COM VERSÃO 0.0.2.0	106
TESTE COM VERSÃO 0.0.2.6	106
TESTE COM VERSÃO 0.0.3.8	107

Versões do Projeto

Versão 0.1

A versão 0.1 foi implementada as seguintes funcionalidades:

- Interação com controles apenas pelo teclado.
- Nenhum objeto 3D modelado.
- Inserção apenas do objeto Casa, sem atributos de poluição.
- Boa Interface Gráfica.
- Carrega e salva o mapa.
- Texturas do terreno.

Versão 0.2

A versão 0.2 foi implementada a partir do conteúdo anterior e as seguintes funcionalidades foram implementadas:

- Otimização de áreas críticas. Há áreas de código que consomem muito processamento e são desnecessárias para funcionamento do jogo.
- Uso de valores percentuais e não reais para determinar o nível de poluição de cada tipo de objeto.
- Uso de uma nova interface para mostrar as informações relativas de um objeto. Como nível de poluição e número de pessoas.
- Como forma de interação com o usuário o nível de poluição total será demonstrado em modo texto, inicialmente.
- O projeto ForChat também será aplicado para este projeto sendo que isto prevê a implementação de modo *multiplayer* (multi jogador).
- Movimentação da Câmera através do mouse.
- Objetos 3D como Casa, Prédio, Igreja, Prefeitura, Fábrica, Árvore.

Versão 0.5

Na versão 0.5 as seguintes funcionalidades foram implementadas:

Número	Tarefa	Prioridade 1 - 5
1.	Estudar técnica de inclinação de vértices para o Mappa	4
2.	Estudar técnica de multi-texture para o Editore	4
3.	Desenhar texturas de rio, grama e areia poluídos	4

4.	Adicionar novos modelos (Escola, Mercado e Hospital)	3
5.	Fazer Tela de Ajuda	3
6.	Deixar o Objeto com uma cor vermelha onde não pode ser construído e verde onde pode. (auxilio de posicionamento de edificações)	3
7.	Refinar técnica de Picking	3
8.	Fazer zoom in/out com o mouse	3
9.	Criar Barras que representem o nível de poluição	2
10.	Otimização da função DisplayTerreno	2
11.	Fazer proporção de espaço no terreno para as Edificações	2
12.	Inserção do cabeçalho GPL	2

Tabela 1. Tarefas e suas prioridades no Projeto.

Versão 0.6

Abaixo segue uma tabela com a lista de tarefas a serem realizadas:

Numer o	Tarefa	Prioridad e 1 - 5
1	Integrar a UI do Editore com os Modelos	5
2	Fazer Inclinação e Declinação do terreno	5
3	Fazer UI para o Mappa	5
4	Integrar a UI do Mappa com a Inclinação e Declinação	5
5	Fazer construção de ruas e Integrar com a UI	4
6	Continuar os efeitos de poluição, (regras do Città)	4
7	Fazer caixa de texto para edição de histórias e informações do objeto	4
8	Telas com alertas de poluição	4
9	Mostrar botões selecionados pelo usuário, exemplo	4

10	Buscar sons e músicas (free)	3
11	Colocar na Interface do Città os indicadores de Poluição	3

Tabela 2. Tarefas e suas prioridades no Projeto.

Versão 0.7

Abaixo segue uma tabela com a lista de tarefas a serem realizadas:

Numer o	Tarefa	Prioridad e 1 - 5
1	Estudo de técnicas Multiplayer	5
2	Integração com o Banco de Dados	5
3	Adicionar os GLComponents	5
4	Organização do Sistema da GUI	5
5	Fazer caixa de texto para edição de histórias e informações do objeto	4
6	Suavização do Terreno	4
7	Iluminação do Terreno	4
8	Telas de Ajuda com a nova GUI	4
9	Rotacionamento de objetos	4
10	Buscar sons e músicas (free)	3

Tabela 3 Tarefas e suas prioridades no Projeto.

Visão Geral do Jogo

Desenvolvimento de um software gráfico para apoiar pedagogicamente na construção de conhecimento das crianças de 3ª e 4ª séries do ensino fundamental no que se refere ao desenvolvimento de capacidades que lhes permitam pensar-construir-organizar uma cidade virtual a partir da realidade infantil e seus conhecimentos incorporados através dos conteúdos oferecidos no aprendizado dos sistemas que regulam uma cidade e suas relações.

Decisões de Projeto

Item #1

É decisão da equipe de programação organizar o projeto atual e estudar sua estrutura, para implementar as propostas aqui demonstradas.

Item #2

Este documento será atualizado sempre que uma nova versão for concluída. Sendo que uma nova versão compreende o início de um trabalho em outra área do projeto, como por exemplo, consequências sobre nível de poluição.

Item #3

Para futura manutenção de código todas as novas funções, classes e objetos serão devidamente comentados. De forma a explicar seu funcionamento e objetividade.

Item #4

A estruturação de projeto em UML, documentação e códigos fonte serão mantidos no servidor Civitas, para mantê-lo atualizado.

Item #5

Haverá uma divisão das tarefas apresentadas acima entre os integrantes da equipe de programação. Serão definidos prazos para entrega de cada um dos itens propostos no fim do documento.

Questões sobre o Projeto do Jogo

Qual o objetivo principal?

O objetivo principal é que o usuário aprenda através da construção e da organização de uma cidade. Para que o mesmo passe por diversas situações que proporcionem este aprendizado ele terá que completar tarefas fornecidas pelos

educadores ou pelo próprio jogo. Nessas tarefas os alunos terão que satisfazer a população e ter uma sustentabilidade que respeite o meio ambiente.

O que o jogador controlará?

O jogador terá controle de uma cidade em desenvolvimento, podendo construir, destruir, editar edificações.

O que ele tem de diferente?

Existem inúmeros programas de computador com propostas semelhantes. Porém, a grande maioria é produzida por grandes corporações de *software* estrangeiras, com objetivos puramente comerciais. Estes jogos freqüentemente estimulam a competitividade e, às vezes, induzem o participante a exercer um capitalismo desenfreado para “superar” e “vencer” seus oponentes.

É fácil constatar que este tipo de produto pode ser indicado para divertir seus usuários por algumas horas, mas não para formar um senso de cidadania e ecologia que dure a vida inteira. Eis a principal diferença entre o Città e seus assemelhados disponíveis comercialmente: o nosso programa é capaz de entreter com uma proposta lúdica enquanto mostra para o usuário noções sobre a sociedade em que ele vive e o meio-ambiente que ele, como cidadão, deve desde criança aprender a proteger.

Outro recurso inovador é a agente Maga Vitta. Que se comunicará por voz com o jogador. Através de um microfone o jogador poderá também falar com ela.

Perguntas referentes ao Città

As questões tratadas a seguir são possíveis melhorias que podem ser adicionadas em versões futuras ou questões que precisam ser definidas para a continuação do projeto.

1. Aparecer ao final de cada sessão do jogo um gráfico mostrando ao aluno e ao Professor, dados relevantes como poluição, numero de habitantes, etc conforme o tempo.
2. O jogo começará em um período anterior? Por exemplo, começar o jogo em 1800, como descrito no documento de discussão da reunião de 21/06/2002. Se utilizar há a necessidade de mais modelos e texturas, assim como programação. Logo mais pesquisa deve ser feita para averiguar como eram as mesmas. Porém o jogo ganha em imersividade e aspectos educacionais.
3. Quais os efeitos ou conseqüências das decisões dos alunos que existirão. Será usado como referencia a tabela: MAGA VITTA: CONVERSATIONAL

ECOLOGICAL AGENT IN AN INTERACTIVE COLLECTIVE CONSTRUCTION ENVIRONMENT FOR BASIC EDUCATION. Por exemplo, a demasiada construção de indústrias perto dos rios, faz com que os mesmos fiquem poluídos, logo é necessário mostrar isto ao aluno.

4. Adição de uma edificação responsável pela produção de energia elétrica. Exemplos: Hidroelétrica, eólica, etc.
5. Uso de pequenos mapas na tela do usuário, contendo informações de florestas, casas, rios, etc.
6. Fazer com que os alunos criem suas músicas e adicionem ao simulador, como mencionado pela Maribel, em que as crianças criam os hinos de seus municípios alterando a letra.

Recursos do Jogo

Recursos Gerais

Processador de 1.5Ghz.

Placa de vídeo aceleradora 3D de 32MB.

Diferentes modos de resolução – 32-bit/pixel.

Espaço em disco de 13MB.

Mundo *top-view*, visão superior.

Música de fundo e ambiente.

JVM - Java Virtual Machine

Gameplay

Descrição Geral

O aluno receberá do instrutor ou de missões do próprio jogo, tarefas a serem completadas. Por exemplo, uma missão do jogo:

Construir 20 indústrias, pois há a necessidade crescimento da região. Logo o aluno terá que construir casas para os habitantes trabalharem nas fábricas. Habitantes precisam de alimentos, saúde, energia elétrica, transporte, educação, etc. Logo mais habitantes são necessários para preencher estes campos de trabalho, e todo este sistema polui e afeta o meio ambiente. O aluno então precisa gerenciar tais recursos para pode completar o objetivo. A figura a seguir descreve as interações entre aluno, professor e simulador.

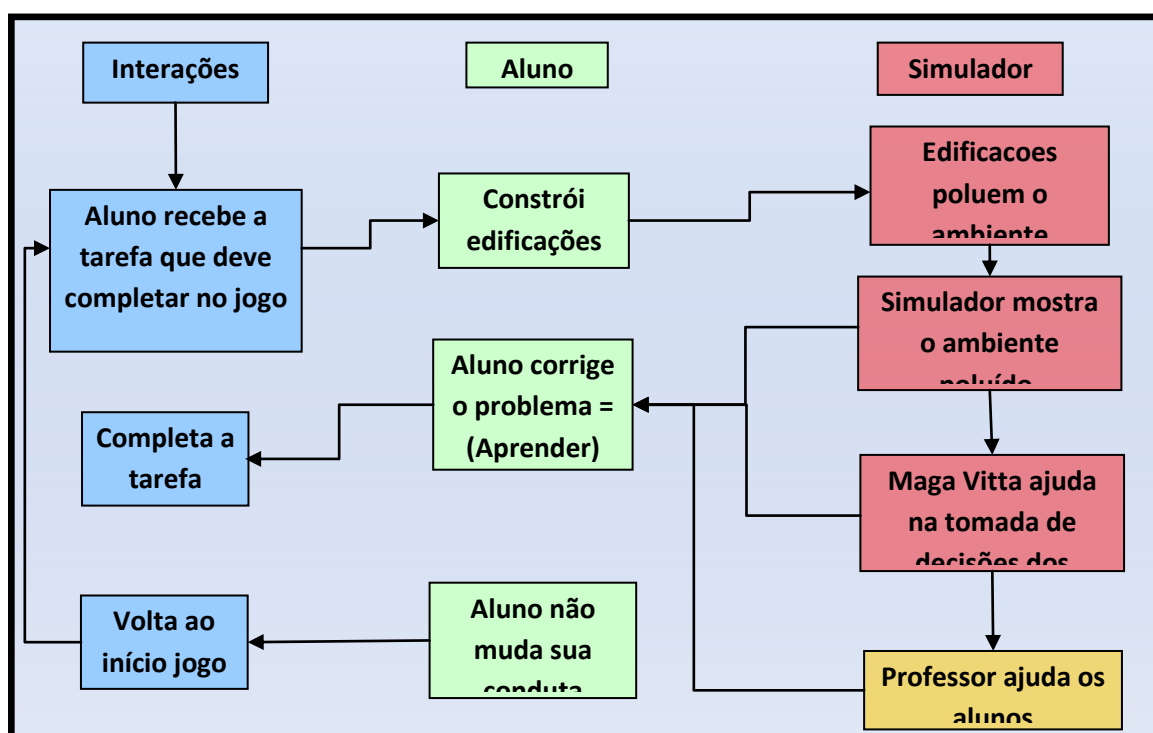


Figura 1 Interações entre aluno, Città , Maga Vitta e professores .

Movimentação no Mundo

O jogador poderá controlar a câmera do jogo através do teclado e do mouse, pelo teclado ao pressionar as setas a câmera efetua um movimento em uma das direções. Através do mouse, a movimentação da câmera é feita quando o ponteiro do mouse fica nas extremidades da tela, logo serão oito direções.

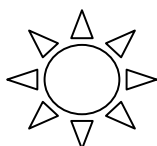


Figura 2. Oito direções de movimento

Inserção de edificações

A inserção de objetos (casa, ruas, etc), será feita através do mouse. Primeiramente o jogador escolhe entre uma edificação ou uma rua. Após, se selecionado edificação, um sub-menu contendo todos os tipos de edificações aparecerá. O usuário seleciona o tipo e escolhe no mapa aonde deve ficar a construção, para facilitar esta inserção o objeto 3D ficará com uma tonalidade diferente conforme o local onde está posicionado o objeto. Se for em um lugar próprio para a construção ficará com a cor normal senão fica em vermelho. A qualquer momento, quando já selecionar um tipo de edificação, será possível sair do modo de construção clicando com o botão direito do mouse.

Edição de histórias e maiores informações

O usuário poderá editar histórias e informações do objeto, clicando com o botão direito do mouse em cima de uma edificação abrirá a janela de informações gerais, ao clicar no botão de editar uma caixa de texto onde o usuário poderá escrever histórias e adicionar maiores informações ao objeto. Pode-se salvar este texto que ficará armazenado no servidor da ilha.

Regras do Città

A tabela a seguir mostra ações do aluno, a consequência desta ação e intervenções da Maga Vitta.

N	Ações do usuário	Consequência	Intervenção da Maga Vitta
1	Construção de indústrias perto de rios.	“Você construiu uma indústria perto de um rio, isto pode contaminar a água”.	- Mover a indústria para outro lugar. - Continuar no mesmo local.
2	Cortar árvores	“Você cortou X árvores, logo você precisa plantar X vezes 5 árvores para manter o	- Continuar a cortar. - Continuar a cortar e

		equilíbrio no ambiente”.	reflorestar. - Não cortar árvores.
3	Construção de casas perto de indústrias	“Você construiu casas perto de uma indústria. Devido à poluição produzida pela mesma, a saúde dos habitantes pode estar em perigo”.	- Não construir casas. - Continuar a construir. - Mover casa para outro local.
4	Construir perto de uma área de preservação ambiental	“Você está construindo em uma área de preservação ambiental. Esta área existe para proteger as espécies e seu habitat”.	- Não construir. - Continuar a construir. - Mover a construção para outro local.
5	Número de leitos insuficientes	“Esta cidade possui X habitantes. O hospital está apto a servir a população?”.	- Sim. - Não.
6	Consumo de energia elétrica	“Uma casa em média consome X watts por mês, uma indústria Y watts/mês, e um mercado Z watts/mês. A cidade gera N watts/mês. Logo é necessário produzir mais energia”.	- Criar uma nova estação de energia elétrica. - Expandir a estação existente. - Eliminar algumas construções. - Reduzir o consumo de energia. - Ignorar este aviso.
7	Consumo de água	“Uma casa em média consome X litros por mês, uma indústria Y litros /mês, e um mercado Z litros /mês. A cidade oferece N litros /mês. Logo é necessário construir estações de tratamento de água”.	- Criar uma nova estação de tratamento de água. - Expandir a estação existente. - Eliminar algumas construções. - Reduzir o consumo de água. - Ignorar este aviso.

Tabela 2 Relação entre a ação do usuário, consequência e intervenção da Maga Vitta.

Há certos efeitos que precisam ser mostrados ao usuário, para que haja um melhor entendimento desta consequência por parte do usuário. Logo é necessário que o Città verifique a ação do aluno e produza uma consequência visível. A seguir será explicado como serão verificadas as ações e como serão feitas as consequências.

1 - Construção de indústrias perto de rios: Verificar se foi construída uma indústria perto de 6 TILES em qualquer direção. Se houver passado determinado tempo mudar a textura antiga do rio para uma mais poluída.

2 - Cortar árvores: Se usuário cortou 7 árvores em um curto período de tempo, o ambiente acrescenta o nível de poluição que pode aumentar o nevoeiro da cidade (fog).

3 - Construção de casas perto de indústrias: Verificação igual a 1. É colocada em cima do modelo da casa uma Cruz vermelha que indica pessoas doentes dentro da casa.

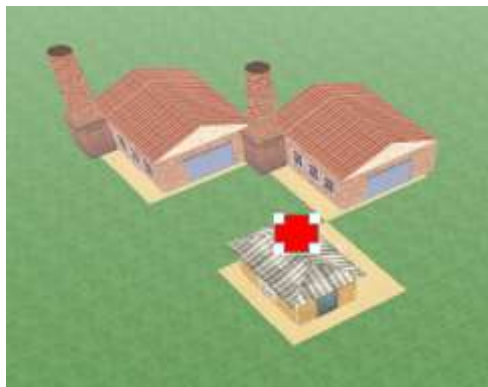


Figura 3. Exemplo do caso 3, onde há pessoas doentes na casa por estar perto de indústrias.

4 - Construir perto de uma área de preservação ambiental – Verificação igual a 1. O ambiente acrescenta o nível de poluição que pode aumentar o nevoeiro da cidade (fog).

5 - Número de leitos insuficientes – Verificar o número de hospitais na cidade e o número de habitantes. Se for insuficiente repetir caso 3 em algumas casas.

6 – Este item não terá verificação e consequência visível nesta versão 0.5.

7 – Este item não terá verificação e consequência visível nesta versão 0.5.

Edição do Mundo (Mappa)

Descrição Geral

Consiste em um editor de relevos com o objetivo de ser uma ferramenta de modelagem computacional aplicada a terrenos naturais, onde serão inclusos os demais elementos componentes de cidades. A idéia fundamental, então, consiste na construção de relevos enquanto base para a construção de cidades virtuais, estas sendo suportadas por teorias/práticas pedagógicas de aprendizagem, incentivando crianças a criar/organizar cooperativamente conceitos, mediante atividades de pesquisa e simulação, orientadas por seus respectivos professores.

Detalhes do Funcionamento

No modo de edição do terreno o usuário pode clicar em dois tipos de botão um para inclinar o relevo e outro para declinar. Após ter sido selecionado, o usuário com o mouse fica no TILE onde quer alterar e com o botão esquerdo inclina ou declina.

Funcionamento do Relevo

Quando um TILE do terreno é inclinado ou declinado seus vizinhos são afetados, o desenho abaixo mostra o funcionamento da edição de relevo.

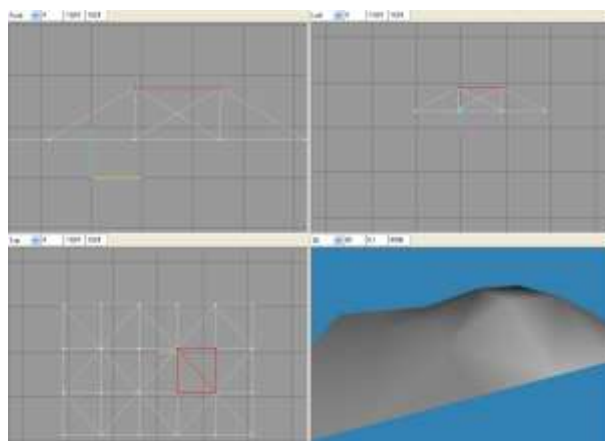


Figura 4. Exemplo de inclinação de vértices no MilkShape.

Tabela 3. Células em azul representam que haverá mudança.

Há um limite de altura tanto para inclinação quanto para declinação, isto é feito para evitar que não haja problemas com a câmera do jogo.

Funcionamento de construção de ruas

O usuário seleciona dentre 3 tipos de rua, asfalto, chão batido e paralelepípedo. Após seleciona-se com o botão esquerdo do mouse aonde se inicia e termina a rua.

Funcionamento de edição de terreno

O usuário escolhe um tipo de textura, grama, areia e água. Com o botão esquerdo do mouse posicionado no terreno é selecionado aonde começa a troca e depois com o outro clique aonde termina. Assim o usuário apenas pinta no terreno o tipo de textura, sem se preocupar com as ligações corretas entre os *tiles*.

Edição de Objetos (Edittore)

Descrição Geral

A edição de edificações será basicamente a troca de cor, visto que criar diferentes tipos de variações de modelos para cada edificação demandaria um numero maior de artistas na equipe ou permitir que o próprio usuário criasse seu edifício seria muito difícil para o mesmo. Logo será utilizada a técnica de *blending*.

Detalhes do Funcionamento

Primeiramente o usuário deve estar no contexto de edição de objetos, isto é, com o botão de edição selecionado como mostra a **figura 9**. Depois selecionar uma edificação utilizando o botão esquerdo do mouse, em seguida clicar na cor e aplica no modelo, sendo que a textura fica preservada.

O Mundo do Jogo

Visão Geral

O “mundo” do Città será uma cidade, que no começo possui um meio ambiente preservado, com árvores, florestas (Áreas de Preservação Ambiental). E conforme o usuário vai interagindo com a mesma seu ambiente pode ser afetado, como explicado nas regras do Città.

Objetos 3D do Mapa

O jogo possui objetos 3D que contem atributos relacionados às regras ou comportamento do jogo abaixo a lista dos mesmos:

Objeto	Fator de Poluição	Nº de Pessoas	Consumo de Alimentos	Produção de Lixo	Consumo de Água
Árvore	-5	-	-2	-5	-5
Casa	5	4	5	5	5
Escola	15	30	20	20	15
Fazenda	-5	4	-50	0	5
Hospital	25	20	30	40	20
Igreja	15	5	10	15	10
Indústria	70	20	30	70	50
Mercado	30	10	-25	30	30

Praça	-20	-	-	1	-
Prédio	40	40	50	55	40
Prefeitura	50	15	20	50	35

Tabela 4. Tabela de atributos dos objetos

Observações:

1 - Valores não reais.

2 - Valores com numero negativo representa que o objeto produz ao invés de consumir. Por exemplo, a fazenda que tem valor de -50 em consumo de alimentos, pois a mesma produz ao invés de consumir.

3 - Células marcadas com hífen (-), não possuem este valor, como por exemplo, uma árvore que não possui acomodações para pessoas.

Abaixo segue uma tabela relacionada ao design destes objetos.

Objeto	Nº Polígonos	Resolução Texturas (px)	Ocupação no mapa	Modelo 3D	Botão
Árvore	272	512 X 512	1x1	OK	OK
Casa	11	512 X 512	1x1	OK	OK
Escola		512 X 512	3x2	OK	OK
Fazenda		512 X 512	5x5	OK	OK
Hospital		512 X 512	2x2	OK	OK
Igreja	35	512 X 512	2x2	OK	OK
Indústria		512 X 512	2x2	OK	OK
Mercado		512 X 512	2x2	OK	OK

Prédio	6	512 X 512	2x2	OK	OK
Prefeitura		512 X 512	3x2	OK	OK

Tabela 5. Tabela com informações dos objetos.

Maga Vitta

Descrição Geral

A Maga Vitta é um agente que monitora o jogo e interage com o jogador para instruí-lo para diferentes emergências no jogo.

Durante o jogo o jogador pode criar vários tipos de incoerências e erros. A Maga Vitta ao detectar estes eventos irá intervir no jogo com um aviso, com as tecnologias de diálogo e reconhecimento de voz o jogador poderá falar com a maga sobre o que fazer, se o jogador der ouvidos a Maga, ela se torna mais calma e harmoniosa mas se o jogador não der ouvidos ela se torna mais zangada, especialmente com descasos em sequência.

Cada jogador tem uma maga própria que está relacionada ao seu nome de usuário e senha. A maga de cada jogador tem uma personalidade baseada nas ações do jogador. Tornando-a única mesmo que outros jogadores compartilhem um mapa para manipular ou jogue em outros mapas, ou seja, se a maga era calma ou agressiva com o jogador enquanto jogava em um mapa, ela continuará com a mesma personalidade quando o jogador jogar em outro mapa.

Ciclo

Interface de Jogo

Descrição Geral

No menu de ferramentas, ao lado direito da tela, constam três opções.

- Menu

Esta opção mostra um novo diálogo que oferece as seguintes funcionalidades.

- Salvar Jogo: que permite guardar as informações atualizadas em um arquivo de texto.
- Carregar Jogo: faz a importação do arquivo gerado anteriormente.
- Ajuda: uma tela com os principais comandos do Città, explica como movimentar-se pela cidade, salvar e carregar jogo, construir edificações e ruas, demolir construções e árvores.
- Mudar resolução: fornece diferentes modos de resolução dependendo com a compatibilidade do monitor e placa de vídeo.
- Sair do jogo; volta à tela de boas vindas, que permite iniciar novamente o jogo.



Figura 5. Botões de edificações e das ruas

- Construir

- Edificações: quando é selecionada esta opção são apresentadas ao usuário todas as edificações disponíveis.
- Ruas: quando selecionada os 3 tipos de ruas são mostrados.



Figura 5. Botões de edificações e das ruas.

- Editar - Editore
Esta seção será detalhada a seguir.
- Fechar;
Quando visualizado no menu *in game* apenas retorna a opção anterior. Já na interface da tela inicial, que possibilita o carregamento do jogo, fecha a aplicação.

Detalhes da Interface do Città

O usuário conta com uma interface simples, evitando a poluição visual com inúmeros botões e abas. No *HUD* serão mostradas as barras de poluição, aceitação da gestão, etc.



Figura 6. Barra da direita mostra o nível de água e da esquerda mostra o nível de poluição na cidade.

Quando o botão direito do mouse for clicado, será mostrado um balão com informações específicas do objeto selecionado.



Figura 7. Ilustração de caixa de texto para informações de edificação.

Detalhes da Interface do Mappa

Esta interface deverá ser de fácil interação com os usuários. Basicamente terá 3 botões de tipos de textura (grama, água e areia), 2 botões para modificar o terreno e outro para construir árvores, como exemplificado a seguir na figura asda .



Figura 8. Interface para edição de terreno.

Detalhes da Interface do Edittore

Esta interface tem dois diálogos um para a mudança de cor, figura assd e outro para a edição de estórias



Figura 9. Interface para edição de objetos.

Jogo com Jogador Único

Descrição Geral

Num primeiro momento vamos testar o jogo no modo *single player* (único jogador), onde é possível modificar o cenário (mapa) do jogo, inserindo construções como ruas, casas, edifícios, indústrias, escolas e hospitais. Posteriormente o modo *multiplayer* também será implementado, entretanto, está ainda em testes. O modo multi-jogador também conta com o uso de um sistema de chat (ForChat).

Carregando e Salvando o Jogo

A versão 0.1, como é denominada pela equipe de programação, atual versão do projeto Civitas, oferece a opção de salvar e carregar um estado de jogo. Este módulo foi classificado como funcional. Portanto, nas próximas versões não será desenvolvida, será preciso apenas modificá-la apenas adicionar ou remover variáveis necessárias aos novos objetos.

Jogo com Múltiplos Jogadores

Descrição Geral

O modo de múltiplos jogadores irá permitir que grupos pequenos possam jogar juntos (até quatro jogadores). Além de usarem suas magas como auxílio, os jogadores também terão uma ferramenta de chat embutida no jogo para se comunicarem uns com os outros.

Terá um computador executando um modulo servidor que ira fazer o gerenciamento do jogo, o tratamento dos comandos dos jogadores e a troca de informações entre jogadores.

Carregando e Salvando o Jogo

Ao final de uma sessão os jogadores deverão salvar o jogo como na versão de jogador único com a diferença que o salvamento do jogo será feito no servidor, os jogadores não ficarão com uma cópia do mapa. Desta forma os outros jogadores que quiserem manipular o mapa sem o resto dos outros jogadores que tem acesso ao mapa poderão faze-lo. Carregar o mapa será também na maquina do servidor para assim permitir fácil acesso a todos do grupo e evitar incoerências com versões de mapa.

Desta forma o servidor é o único que pode fazer alterações no mapa mas este só o faz a pedido dos jogadores.

Música e Efeitos Sonoros

Descrição Geral

Para dar mais percepção ao usuário de que esteja a construir uma cidade é interessante que se utilizem efeitos sonoros de cidades. E de preferência dependendo do tamanho da cidade seja um som diferente. As músicas serviram para entreter também o usuário e não ficar monótono o jogo. Será utilizada a biblioteca livre FMOD.

Músicas

Utilizar de 3 a 6 músicas no jogo que tocam de forma aleatória, as músicas serão “calmas” para não afetar o usuário, principalmente por ser um publico jovem.

Som

É interessante que seja utilizado sons para deixar o jogo mais interativo e interessante para o usuário. Porém achar ou fazer alguns destes sons seja difícil. Pode-se tentar baixá-los em páginas da Web que distribuem este conteúdo de forma gratuita, não os encontrando pode-se tentar gravá-los com um microfone de um MP3 e passar o arquivo para o PC.

Abaixo segue uma tabela com os sons e quando estes tocaram no jogo.

Número	Evento	Tipo de Som
1	Botão pressionado	Informar
2	Construir Casa	Som relacionado ao Objeto
3	Construir Escola	Som relacionado ao Objeto
4	Construir Fazenda	Som relacionado ao Objeto
5	Construir Hospital	Som relacionado ao Objeto
6	Construir Igreja	Som relacionado ao Objeto
7	Construir Indústria	Som relacionado ao Objeto
8	Construir Mercado	Som relacionado ao Objeto
9	Construir Prédio	Som relacionado ao Objeto
10	Construir Prefeitura	Som relacionado ao Objeto
11	Demolir Edificação	Som de demolição
12	Obter Informação da Edificação	Informar 2
13	Salvar e Carregar jogo	Informar 3
14	Rio Poluído	Alerta
15	Casa com pessoas doentes	Alerta

Tabela 6. Tabela relacionada com os sons e os eventos que os foram ser tocados.

Cronograma

Este cronograma foi usado até a versão 0.2, após está sendo usado o MS Project como gerenciador de Projetos ou o Planner.

Testes

A seguir serão detalhados os testes realizados em cada versão do código, que é diferente da versão deste documento, a fim de estabelecer um comparativo entre as mesmas.

Teste com versão 0.0.2.0

1. Teste realizado com um Athlon XP 2000 + 1.67 GHz com 480 MB de RAM e placa de vídeo de 32 MB *onboard*.
Città executando com todas as funções = 1.7 FPS.

Programa executando sem desenhar o terreno, mas desenhando o resto = 20FPS a 45FPS.

Programa executando desenhando o terreno, mas sem desenhar o resto = 2.0 FPS.

Conclusão do teste - Verificou-se que a função de desenhar o terreno gasta muito processamento gráfico. É necessário otimizar este código caso haja necessidade de ser executado em computadores de baixo processamento.

Teste com versão 0.0.2.6

1. Teste realizado com um processador Intel Core2 Duo 2.20 GHz com 2GB de RAM e placa de vídeo de 128 MB *onboard*.

Città executando com todas as funções = 40 FPS.

Programa executando sem desenhar o terreno, mas desenhando o resto = 200FPS.

Programa executando desenhando o terreno, mas sem desenhar o resto = 60 FPS.

Conclusão do teste - Mesma conclusão da versão anterior.

Teste com versão 0.0.3.8

1. Teste realizado com um processador Intel Core2 Duo 2.20 GHz com 2GB de RAM e placa de vídeo de 128 MB *onboard*.

Città executando com todas as funções = 80 FPS.

Programa executando sem desenhar o terreno, mas desenhando o resto = --- FPS.

Conclusão do teste - Conforme a função de desenhar o mapa ou terreno foi otimizada, além de modificações na parte da Interface Gráfica, os *frames per second* aumentaram consideravelmente.

2. Teste realizado com um processador Sempron 1.61 GHz com 1GB de RAM e placa de vídeo Ge Force 6200 de 256 MB.

Città executando com todas as funções = de 80 FPS a 45 FPS.

Programa executando sem desenhar o terreno, mas desenhando o resto = 86 FPS.

Conclusão do teste - Mesma conclusão da do teste anterior.