AI.

A inteligência artificial do jogo foi implementada em Prolog através do WinProlog da LPA.

A comunicação entre o jogo e o PROLOG é feita através do Intelengence server apropriado para C++.

Foi usado o código de kickoff fornecido para implementar o código em para devolver os percurso do grafo mais rápido para fazer a entrega de todas as encomendas.

Existiram 2 pontos chaves para o desenvolvimento do programa kickoff:

1. A escolha e implementação da parte que encontra o melhor caminho entre dois nós;
2. A escolha da ordem dos pontos de entrega;

No primeiro caso optou-se por implementar o algoritmo Astar que permite um melhor desempenho em termos de racio entre melhor solução e tempo de execução. A implementação do Astar foi baseada nos slides das aulas de ALGAV com a diferença que adaptou-se a distancia usada nesse predicado para o tempo que demora a percorrer cada ramo (Estrada) tendo em conta a velocidade máxima do camião ou então a velocidade máxima permitida na estrada para uma dada hora.

Assim sendo como a heuristica usou-se o tempo que o camião com a velocidade máxima demoraria a percorrer a distancia em linha recta entre os nós.

A escolha da ordem dos pontos de entrega foi baseada na ideia geral do caixeiro viajante. Neste caso decidimos por ordernar as encomendas tendo em conta a distancia delas. Assim sendo permitimos ao programa procurar primeiro o melhor percurso para o ponto de entrega mais próximo, e depois assim sucessivamente para os pontos de entrega restantes para devolver no final o percurso total.

Para a implementação total destas componentes de prolog foram implementados vários predicados auxiliares dos quais enumeramos alguns dos mais importantes:

soma\_tempo\_a\_relogio: permite somar tempo em minutos a um relogio para ir actualizando o respectivo relogio ao longo do percurso;

actualiza\_hora\_actual: predicado evocado para ter em conta a evocação do gps no jogo durante o jogo tendo em conta o tempo decorrido;

relogio\_para\_minutos: transforma o relogio em minutos;

inicio\_no\_tempo\_clean\_nos: Este é o principal predicado evocado para comunicação com o jogo. Executa todo o código de maneira limpa (sem impressões intermédias) e recebe como parametro o ponto actual assim como o tempo decorrido que vêm do jogo.

ordenar\_encomendas\_porDistanciaConsecutiva: predicado que permite ir escolhendo recursivavemente o proximo nó de ponto de entrega.

procuraAstarTempo: implementação do a star modificado para ter em conta o tempo de percurso em alternativa a distancia de percurso.

obtemVelocidade: predicado que devolve a velocidade para um dado ramo e para um dado tempo decorrido de jogo tendo em conta a hora do relogio inicial.

Quick\_sort2: predicado auxiliar que permite a ordenacao das encomentas tendo em conta a distancia.

limpar\_ruas\_sem\_possibilidade\_de\_trafego: predicado que limpa os ramos que estao impossibilitados de serem circulados pelo camião dado, tendo em conta as restricoes de Peso, Largura e Altura. Para estas restrições não são consideradas horas.

Sem\_restricao\_horas: predicado se verifica se existe alguma restricao (explosivo e/ou inflamavel) para um dado ramo para uma dada hora.

Passa\_Nodos\_Ramos: predicado que permite passar uma lista de nodos para uma lista de ramos, ou viceversa.

SGRAI:

A implementação do modo de jogo foi feita em OpenGL, com auxílio da biblioteca GLUT.

O sistema gráfico encontra-se dividido em três principais modos de jogo, sendo que cada um deles visa à satisfação de uma necessidade específica.

O primeiro desses modos é visualizado assim que arranca o jogo, pois é o menu principal do jogo. Neste foi optada por usar uma projecção ortogonal 2D. Existem 4 níveis diferentes no menu correspondentes às várias opções disponíveis. Principal (onde se pode optar pela escolha de um jogo ou de um torneio), escolha de mapa (onde são carregados e visualizados os mapas existentes na pasta “mapas” existente no directório raiz da aplicação, escolha de torneios, onde se convencionou que nunca haveria mais que três torneios por onde se pudesse escolher e finalmente a visualização das classificações devolvidas pelo webService implementado para o efeito e que “responde” com o top 5 das classificações depois de recebida uma nova pontuação. A escolha entre os vários elementos existentes no menu é feita através de “picking” ou através da utilização das teclas do cursor assim como da tecla “Enter” + “Esc”.

O segundo modo corresponde ao modo grafo. Neste ainda podem ser diferenciados duas situações distintas, onde na primeira existe a hipótese de seleccionar os destinos a efectuar paragens e o segundo em que decorre a animação do modelo camião entre os vários nós escolhidos.

Na primeira situação a escolha dos destinos é feita também através de “picking” sobre os nós a escolher, sendo que os mesmos são apresentados numa pequena área própria para o efeito que se encontra à direita do ecrã. A escolha de nós tem de obedecer a validações de integridade, nomeadamente se o próximo nó a percorrer está ligado directamente ao nó anterior. Na área de visualização dos nós escolhidos estes são apresentados na ordem em que serão percorridos. É possível nesta fase utilizar o recurso de AI através do uso do GPS. Para activar o GPS, é necessário pressionar na tecla “d” o que irá escolher o melhor percurso conforme descrito acima. Uma penalização para o uso do GPS está prevista, mas ainda não está implementada. Apenas é necessário estabelecer com o cliente valores para a penalização que irá ser automaticamente adicionada ao tempo (pontuação). Para finalizar a escolha do percurso e passar para o modo de animação terá de ser premida a tecla “g”.

O terceiro modo de jogo corresponde à zona de cargas e descargas do veículo. Nesta o utilizador terá como missão carregar as encomendas no nó inicial e descarregar as mesmas em cada nó respectivo. Este modo também foi implementado com o auxilia da técnica de “picking”, onde é possível arrastar as mesmas mantendo premido o primeiro botão do rato ou efectuar uma rotação de 90º pressionando o segundo botão do rato. Caso alguma das condições necessárias para o sucesso da operação esteja em falta o utilizador é informado através de uma mensagem. Enquanto todas as encomendas respectivas não tiverem sido carregadas ou descarregadas não é permitido avançar deste modo. A validação desses mesmos procedimentos é feito também durante a movimentação das peças, sendo que a movimentação destas para locais não permitidos não é possível. Esta verificação foi implementada através da criação de uma matriz binária de posições que reflecte a posição que cada uma das peças ocupa em cada instante, assim como da área disponível na “dock” de carga. O formato das encomendas foi baseado em tétrominos, muito à semelhança das peças existentes no jogo Tetris. A disposição das peças assim como da “dock” é feita também em 2D.

Antes da entrada em qualquer um destes modos existe ainda uma vertente de desenho do grafo que foi implementada de raiz, sendo que a proposta oferecida, embora tenha servido de exemplo, não foi implementada, optando por criar uma abordagem nova. Tal deveu-se à dificuldade encontrada em transformar a solução apresentada numa que previsse arcos com ligações em sentidos oblíquos entre os vários nós. Assim sendo foram criados diferentes métodos de desenho dos vários elementos do grafo:

Desenho de estradas com rampas:

As estradas são desenhadas tendo em conta as coordenadas dos nodos que as ligam. As rampas são criadas para preencher as lacunas de espaço entre as estradas e as rotundas. Os angulos de declive e de rotação são calculados todos da mesma maneira e implementou-se uma função geral que serve para todas as rotações e declives. As estradas/arcos são variáveis em largura e essa mesma variação determina a textura que é visualizada na mesma. Para arcos com largura inferior a 1.5 é apresentado o mesmo como uma estrada em paralelo. Todos os arcos com largura maior são apresentados como uma estrada alcatroada de duas faixas. Não é aconselhável o uso de estradas com largura inferior a 0.8 já que irá fazer com que o camião entre em contacto com as paredes desse mesmo arco, já que o mesmo anda á direita do eixo de rodagem.

Desenho das paredes das estradas:

As paredes das estradas são preenchidas com uma textura. Aqui tivemos o cuidado de ter em conta a inclinação do declive das estrada para que a textura (baseada em predios) fique com os predios direitos em relacao ao eixo dos z. Este premonor foi implementado recorrendo À técnica shering onde tivemos que ajustar uma matriz de transformação adaptada a inclinação da estrada para permitir a distorção certa para dar o efeito direito.

Desenho das rotundas (Nos):

As rotundas são desenhadas atraves de poligonos e uma textura colorida para dar mais vida à animação e desenho do jogo em geral.

Desenho das paredes das rotundas:

O desenho das paredes das rotundas é semelhante ao desenho das paredes das estradas. As paredes da rotunda são baseadas na mesma textura, neste caso as rotundas não tem inclinação mas têm a curvatura que tem de ser seguida pela textura.

Para cada rotunda desenha-se um numero de paredes que corresponde ao numero de nos que essa rotunda consegue ligar. 1 no de ligação 1 parede, 2 nos de ligação 2 paredes, etc... Com excepção para o caso onde teriamos um nó sem ligação, nesse caso a parede é 1 que faz os 360 graus da rotunda.

Para o desenho correcto destas paredes têm de ser calculados todos os angulos (no plano xy) das ligacoes das estradas aos nos ligados. Além disso é também necessário considerar para cada uma das estradas ligada a largura respectiva.

Nesta fase também foram implementadas alguma correcoes de desenho, mais concretamente os buracos das ligações entre as paredes da estrada e as paredes da rotunda.

Em todos os desenhos de paredes com a textura foi forçado que a textura comece e acabe com valor inteiro. Desta maneira evitamos que as casas não fiquem cortadas no final das estradas ou paredes da rotunda.

Às texturas das casas foram aplicadas transparencias usando imagens targa (.tga) com o canal alpha e activado o GLBLEND permitindo assim que apenas o desenho das casas ficasse visível e permitindo as áreas em branco ficassem transparentes.

A seguir descreve-se o modo como é feita a movimentação do modelo no grafo.

Devido a especificações do cliente optou-se por criar uma animação em detrimento de uma movimentação controlada pelo utilizador. Dessa forma a abordagem utilizada foi diferente da que seria usada caso fosse necessária essa mesma movimentação pelo utilizador.

Assim sendo a animação é feita automáticamente. Não são detectadas colisões, apenas é permitido ao camião seguir o caminho que é sempre validado antes de permitirmos a acção da animação.

Na animação temos em conta a velocidade do camiao e as curvas de entrada, saida de rotunda, assim como a curva que percorre a rotunda até encontrar a saída para o proximo nó.

Sons de Jogo:

Implementamos a introdução de uma musica de jogo que arranca no ínicio deste. Além da musica de jogo também introduzimos um efeito sonoro de buzina. Este é accionado através da tecla b.

O som de jogo foi implementado recorrendo ao alut em anologia com o glut.