

Abordagem Tridimensional e Interactiva para Analizador Tático de *Match Racing*

Fausto José Mourato
Departamento de Sistemas e Informática
Escola Superior de Tecnologia
Instituto Politécnico de Setúbal
2910-761 Setúbal
fmourato@est.ips.pt

Manuel Próspero dos Santos
CITI, Departamento de Informática
Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa
2829-516 Caparica
ps@di.fct.unl.pt

Sumário

Tem havido, recentemente, uma ascensão tecnológica no contexto desportivo. É cada vez mais recorrente, em diversas modalidades, o uso de sistemas audiovisuais, comunicação e gestão de dados, entre outros. Neste artigo é apresentada uma aplicação no domínio da computação gráfica interactiva, para uma nova abordagem de construção e recriação de trajectos efectuados em regatas de vela, na modalidade de Match Racing. A aplicação consiste num editor de cenas tridimensionais, onde é possível inserir e posicionar diversos elementos de Match Racing, tais como embarcações, marcas e outros, definindo-se também o comportamento temporal destes. O resultado final é uma animação virtual passível de estudo pelos especialistas. A utilização prática possui carácter inovador na possibilidade de análise detalhada das situações, podendo-se recriar pontos de vista na primeira pessoa, sendo uma aplicação de elevada utilidade para equipas profissionais e árbitros, tanto em contexto de formação como de competição. No artigo é analisada uma estrutura original utilizada para representação dos diagramas táticos e a forma como esta abordagem pode ser reutilizada em projectos similares. Também é apresentado o sistema de Graphical User Interface (GUI) construído para controlo desta aplicação, sendo referidas as vantagens e desvantagens da sua estrutura para inclusão noutros projectos.

Keywords

Concepção de Sistemas Interactivos, Aplicações Gráficas Interactivas, Motores Gráficos, XML

1. INTRODUÇÃO

A sociedade tem sofrido, nas últimas décadas, fortes alterações devido ao impacto tecnológico em todos os sectores. A informática e as telecomunicações mudaram em muito a forma de viver das pessoas e facilmente fizeram esquecer como tudo funcionava anteriormente. O contexto desportivo não é excepção, e nele tem havido, nos últimos anos, um crescente uso de tecnologia, levando a mudanças técnicas, táticas, estratégicas e financeiras, entre outras. São inúmeras as aplicações de tecnologia nos desportos actuais, que vão desde a utilização do mais simples software e hardware até complexos sistemas de informação ou dispositivos altamente sofisticados. Os auriculares de comunicação utilizados por árbitros de futebol [Chap04] são um exemplo de hardware comum recentemente adaptado para este universo. Em termos de software, e como exemplo de aplicação em desporto, podem-se referir os programas para análise de movimentos no golfe [Swis05] por intermédio de análise de vídeo. Diversas implementações futuras estão também já em estudo, como a possibilidade de colocação de chips em bolas desportivas [Blau05] para uma correcta localização da mesma.

A *Match Racing* [Meni05] é uma disciplina da modalidade de vela, com a particularidade de se tratar de uma corrida de curta duração entre dois barcos, ao invés das corridas longas mais tradicionais entre um número superior de barcos. Devido à natureza de duelo, existe uma elevada necessidade de rigor técnico e tático. O único objectivo é o de chegar primeiro que o adversário, e qualquer outro resultado é negativo. A mais prestigiada competição de *Match Racing* é a *America's Cup* [ACSA07], que se realiza a cada quatro anos, e em que um conjunto de equipas tenta destronar o vencedor da prova anterior. Trata-se de um evento mediático de cobertura a nível mundial, com investimentos financeiros elevados.

Embora seja uma modalidade com muitos seguidores, foi detectada uma insuficiência na utilização de tecnologias informáticas neste meio, sendo que o software mais evoluído é somente utilizado pela comunicação social. Do contacto académico com os praticantes nacionais desta modalidade, surgiu a percepção da utilidade de uma aplicação tridimensional para representar as situações vividas nas regatas, por parte dos atletas e dos juizes.

Assim, é apresentado neste documento o protótipo desenvolvido com o objectivo referido. Primeiramente é feito um enquadramento das tecnologias e aplicações existen-

tes nesta modalidade. Em seguida são apresentados os requisitos levantados para a aplicação, bem como as diversas tecnologias utilizadas para o desenvolvimento deste projecto e as condicionantes que levaram a determinadas decisões. Depois é apresentada a estrutura lógica segundo a qual se desenvolveu a aplicação, com detalhe na abordagem da representação espaço-temporal dos diagramas e a forma de implementação escolhida para o fazer. Também é apresentada a interface gráfica com o utilizador que foi desenvolvida no âmbito desta aplicação, a qual poderá ter aplicabilidade noutros projectos. Na parte final apresenta-se, sumariamente, o protótipo da aplicação e são explicitadas algumas conclusões sobre o projecto, a sua aplicabilidade e desenvolvimentos futuros.

2. ENQUADRAMENTO DO PROJECTO

2.1 Tecnologia em *Match Racing*

A ligação à tecnologia na modalidade de *Match Racing*, no que respeita à informática, surgiu com uma aplicação de nome *Tactical Sailing Situations* [Pero08], referido usualmente no meio simplesmente como TSS. Este software permite a representação gráfica bidimensional, com vista de topo, das situações ocorridas nas regatas. As características desta aplicação serão aprofundadas um pouco mais na secção 3.1 deste documento. Em termos de hardware, esta modalidade recorre actualmente a sensores que permitem indicar aos árbitros se as duas embarcações em competição se encontram sobreladeadas, ou seja, se existe algum prolongamento lateral de uma embarcação que intercepte a outra. O sobreladeamento tem muita relevância na interpretação de certas regras.

2.2 Motivação

O treino tático em *Match Racing*, enquadrado na preparação das equipas ou no plano de formação dos juizes, desenvolve-se quase exclusivamente em torno de diagramas estáticos, criados em contexto de aula com modelos magnéticos ou impressos como diagramas 2D. Estes diagramas surgem inevitavelmente associados à aplicação TSS referida anteriormente. Esta ferramenta é, na sua essência, um editor que permite o desenho em computador dos diagramas táticos, os quais podem ser exportados como imagens e utilizados posteriormente, sob este formato, em apresentações de formação ou em palestras às equipas. Na representação de situações em diagramas estáticos e discretos, a componente temporal é pobre, não existindo a noção de continuidade. Mesmo considerando só a abordagem bidimensional, é possível conseguir-se melhoramentos em termos da criação de animações contínuas, com benefício para os utilizadores, que podem assim perceber melhor as particularidades das trajectórias descritas pelas embarcações. Por outro lado, a restrição ao domínio 2D impossibilita uma análise na primeira pessoa, mediante determinadas perspectivas. Só em 3D é possível ter noção de certos pormenores da modalidade, no que respeita à grande dimensão dos barcos e relativa proximidade entre estes. As próprias ilusões causadas pela perspectiva podem ser analisadas, tendo-se também assim em conta a subjectividade no processo de análise.

Desta forma, surgiu a necessidade de uma nova aplicação que permitisse a criação de um novo tipo de diagramas, usando uma representação tridimensional, mais realista e com animação, permitindo uma representação mais detalhada e com um maior grau de possibilidade de análise. Esta nova aplicação torna possível a representação de aspectos outrora não contemplados, como por exemplo o ponto de vista a partir de um determinado barco.

2.3 Objectivos

O objectivo deste trabalho assentou, então, na criação de uma aplicação tridimensional que permitisse representar situações de regatas de *Match Racing*. Assim, é possível inserir e posicionar diversos elementos deste domínio, tais como embarcações, marcas e outros, bem como definir o comportamento temporal destes. Aspectos particulares como o posicionamento das velas e marcas auxiliares são também contemplados. O resultado final do uso da aplicação é uma animação controlável pelo utilizador e passível de estudo e análise, tanto por parte de atletas como por juizes. Esta análise tem validade em contexto de formação, para explicação de determinadas situações, ou em competição, no final dos diversos *rounds*, aquando da discussão dos principais casos que, actualmente, são unicamente representados com recurso ao programa TSS.

3. APLICAÇÕES EXISTENTES

O objectivo deste ponto é do de apresentar os principais programas aplicáveis à modalidade de *Match Racing* e o contexto de tal aplicação. Pretende-se identificar assim os aspectos que são actualmente contemplados, percebendo-se melhor os detalhes das principais carências de funcionalidades.

3.1 *Tactical Sailing Situations*

O programa principal de referência na área chama-se TSS (*Tactical Sailing Situations*) e, como já foi referido anteriormente, permite aos utilizadores representarem graficamente situações de regatas. Trata-se de um programa com origem ainda nos paradigmas monolíticos de programação, o qual não tem sofrido grandes alterações desde o seu desenvolvimento inicial, baseando-se estas em pequenas actualizações e correcção de erros. A sua aparência é simples, tornando a aplicação eficaz (Figura 1). A reduzida qualidade gráfica não se tem apresentado como um problema, uma vez que tal aspecto resulta também em simplicidade. Devido à sua eficácia, e também por ausência de boas soluções alternativas, esta aplicação é uma referência ímpar na área da vela, sendo conhecida e utilizada por todos os praticantes da modalidade.

A interface é simples e está perfeitamente assimilada por todos os utilizadores do programa, muitos dos quais utilizam a aplicação há diversos anos, sendo esse facto um obstáculo a qualquer perspectiva de mudança. É disponibilizada uma barra de ferramentas com diversos tipos de objectos que se podem inserir no quadro representativo da zona da regata. Estes objectos podem ser facilmente reposicionados por um mecanismo tradicional de *drag-and-drop*. Para as embarcações, é possível adicionar-se um conjunto de posicionamentos que são interpre-

tados como consecutivos, obtendo-se assim uma representação sob a forma de diagrama estático. Finalmente, é possível animar o diagrama, sendo feito um *slideshow* posição a posição. Não existe, portanto, uma real animação com interpolação dos diversos pontos no trajecto, sendo este visto simplesmente como uma sequência discreta de posicionamentos.

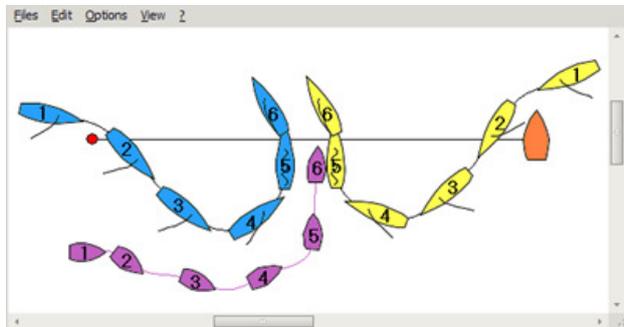


Figura 1 – Tactical Sailing Situations

3.2 Outras aplicações

As aplicações com maior carácter gráfico utilizadas em *Match Racing* têm outros objectivos que não a análise táctica. Um programa, de nome *Virtual Eye* [Anim03], faz representações tridimensionais de trajectos realizados por embarcações, à semelhança do analisador táctico apresentado neste artigo. Há dois aspectos principais em que esta aplicação é muito forte. Em primeiro lugar, a qualidade gráfica é elevada, utilizando tecnologia corrente e modelos de elevado detalhe. Em segundo lugar, esta aplicação utiliza dados em tempo real através de dispositivos de localização.

É uma aplicação focada para transmissões televisivas, estando a sua utilização inclusivamente restrita a este contexto. Não se trata, portanto, de uma aplicação que possa ser utilizada em contexto de treino e formação. A própria integração de tecnologia para o seu funcionamento não pode ainda ser considerada trivial, requerendo alguns meios não disponíveis para todos os praticantes da modalidade e sob quaisquer condições.

Há a referir ainda a existência de um jogo, de nome *Virtual Skipper* [Nad07], cujo último lançamento data de 2007, associado ao evento da *America's Cup* do mesmo ano. A quase totalidade desta aplicação possui uma vertente lúdica. Embora se trate de uma modalidade à escala mundial, o *Match Racing* é uma modalidade desconhecida por muitas pessoas. Assim, este jogo disponibiliza um modo de aprendizagem onde são explicadas algumas das técnicas e tácticas, para que quem desconheça a modalidade a possa aprender resumidamente. No entanto, a aplicação é essencialmente um videojogo, com objectivos totalmente diferentes dos de um analisador táctico.

4. LEVANTAMENTO DE REQUISITOS

Mais do que uma simples representação esquemática bidimensional para uma análise global, torna-se importante disponibilizar capacidades de análise particular. De facto, a utilização de tecnologias 3D com realismo permi-

te a transmissão de determinados detalhes impossíveis de contemplar de outra forma, como se exemplifica na secção 7. Com esta base, foi levantado um conjunto de requisitos importantes para a aplicação, a seguir apresentados.

- **Modo de funcionamento**

Edição de situações de modo muito semelhante ao TSS, com o acréscimo da navegabilidade tridimensional na cena e elevada qualidade gráfica (à semelhança de *Virtual Eye* ou *Virtual Skipper*).

- **Principais objectos a representar**

As duas embarcações em competição, as embarcações dos árbitros e árbitros auxiliares, o barco da comissão de regatas e as bóias de sinalização.

- **Posicionamento dos objectos**

Deve ser feito com base numa metáfora comum, permitindo ainda a selecção de um ou múltiplos objectos numa só operação.

- **Linhas auxiliares dos trajectos**

Correspondem aos trajectos efectuados pelas embarcações, devendo ser automaticamente actualizados, em tempo real, sempre que o utilizador altere as posições ou orientações da representação gráfica dos barcos.

- **Controlo da posição da vela**

Orientação da vela principal e representação da vela balão (*spinnaker*) quando estiver içada. A possibilidade desta representação não está contemplada no programa de referência TSS, o que se lhe aponta como uma grande falha.

- **Reprodução das animações**

Possibilidade de acesso ao modo de animação, após a construção de um diagrama táctico, para navegação e análise da situação. Para ser útil em casos reais, surge a necessidade de se representarem, de forma contínua, os trajectos das embarcações durante uma prova, para além de se poder posicionar a câmara em qualquer localização.

- **Armazenamento dos diagramas**

Escrita em disco, para posterior reutilização nesta ou noutra aplicação, dos diagramas criados.

- **Informações auxiliares**

Informações complementares que são relevantes para os utilizadores do programa e que podem ser representadas sob a forma de configuração dos objectos existentes. Por exemplo: as marcas que permitem verificar facilmente o sobreladeamento entre dois barcos; os gráficos (círculos) que, dada uma distância, assinalam o espaço de vizinhança de uma certa embarcação; as bandeiras (*Y-Flags*) que as equipas acenam para reclamarem uma infracção, bem como as bandeiras que a equipa de arbitragem usa para comunicação das suas decisões às equipas.

- **Capacidade de integração**

Requisito necessário para comunicação e partilha de informação entre as aplicações do conjunto a ser desenvolvido para apoio a treino e formação em *Match Racing*.

5. TECNOLOGIAS

Para o desenvolvimento deste projecto foi necessário tomar algumas decisões a nível tecnológico. O objectivo deste ponto é, então, o de apresentar as principais questões que foram colocadas nesse aspecto, bem como as opções tomadas e consequentes implicações.

5.1 Motor gráfico

O desenvolvimento de uma aplicação tridimensional requer a utilização de um motor gráfico que garanta um bom conjunto de funcionalidades, num nível de abstracção mais elevado do que a simples definição de polígonos e cálculos de componentes básicos de iluminação, a título de exemplo. Exige-se, de forma geral não exaustiva, que esse motor faculte a importação de modelos tridimensionais criados por aplicações dedicadas para o efeito, que possibilite uma fácil e intuitiva hierarquização do grafo da cena, e que possua um bom suporte para a utilização de materiais. A criação de um motor gráfico de raiz é uma tarefa de grande complexidade, caso se pretenda conseguir uma qualidade visual elevada, e só se justifica em empresas de dimensão significativa. São projectos que requerem grandes equipas de desenvolvimento e em que a sua rentabilidade prática só se manifesta aquando do seu uso em diversas aplicações. Deste modo, a decisão de se adoptar um motor gráfico existente surgiu naturalmente, devido ao contexto em que este projecto se insere.

Os motores gráficos livres mais populares da actualidade, tais como *Ogre 3D* [Junk06], *Crystal Space* [Jorr00] ou *Panda 3D* [Gosl04], usam a linguagem de programação C++, embora existam actualmente outras alternativas [Bull01, Wils06].

Após uma análise comparativa das funcionalidades, a qual sai fora do âmbito deste artigo, a escolha para o desenvolvimento deste trabalho recaiu sobre o *Ogre 3D*. Os principais pontos que sustentaram esta decisão foram o conjunto de ferramentas associadas e as excelentes capacidades de suporte (fórum, comunidade e documentação). No âmbito em que se enquadrava o desenvolvimento deste projecto, os referidos aspectos tornam-se especialmente relevantes.

5.2 Ficheiros de Diagramas

A necessidade de guardar os diagramas criados, para posterior utilização, foi também um objecto de estudo neste trabalho. Considerou-se, para o efeito, um conjunto de alternativas, sendo estas apresentadas em seguida.

A forma mais básica para guardar os diagramas criados seria a utilização de simples ficheiros de texto, nos quais se escreveria toda a informação usada nos diagramas. No entanto, esta simplificação tecnológica poderia acarretar dificuldades futuras, uma vez que é desejável ter uma estrutura dinâmica e flexível.

Como se referiu anteriormente, a linguagem escolhida para o desenvolvimento da aplicação foi o C++, assentando no paradigma orientado por objectos. Assim, a serialização binária de objectos seria também uma forma de guardar os diagramas criados. O grande problema de implementação desta solução prende-se com a integração

com outras aplicações. Foi assumido, como requisito, a existência de boas capacidades de integração e, neste aspecto, a serialização apresenta grandes falhas, uma vez que é muito limitativa no que respeita à expansibilidade de integração. A possibilidade de existência de novas aplicações que utilizem os diagramas criados pelo Analisador Tático 3D requer um formato independente da linguagem e a serialização é, no mínimo, um obstáculo nesse aspecto.

Uma forma alternativa de armazenar o conteúdo dos diagramas passa pela utilização de ficheiros XML. Os problemas estruturais colocados anteriormente, com a utilização de simples ficheiros de texto, são suprimidos pelas próprias características do XML. Torna-se apenas necessário que a definição do formato siga as regras da construção de um documento XML. No fundo, sem aumentar a complexidade do problema, consegue-se ter a informação estruturada de forma hierárquica e num formato padrão aceite em larga escala. O problema de integração levantado na utilização na serialização é também ultrapassado, uma vez que o XML é uma das mais famosas soluções de integração e é vasto o leque de ferramentas que funcionam no seu âmbito. Mesmo nos casos mais complicados de integração, em que não existem ferramentas desenvolvidas, há que não esquecer o facto de um ficheiro XML ser, na sua essência, um mero ficheiro de texto.

A capacidade de integração, aqui apresentada, tem o objectivo de permitir que os diagramas táticos criados na aplicação possam ser posteriormente armazenados em bases de dados ou associados a vídeos das situações representadas nesses mesmos diagramas, sendo assim possível documentar melhor uma prova.

6. ESTRUTURA DA APLICAÇÃO

6.1 Funcionamento geral

A decisão de utilização da aplicação em dois modos distintos teve em conta os requisitos apresentados nas secções anteriores.

O primeiro modo é aquele em que o utilizador constrói a representação de uma determinada situação de *Match Racing*, a qual se designa por diagrama tático (ou apenas diagrama). Cabe ao utilizador, neste modo, seleccionar as diversas entidades de *Match Racing* e definir as suas propriedades visuais e espaciais. É também aqui que o utilizador pode definir trajectos para os elementos presentes no diagrama. O segundo modo de utilização da aplicação é aquele em que se faz a animação de um determinado diagrama. Os elementos colocados no modo anterior são agora animados, mediante a configuração estabelecida.

6.2 Estratégia de representação

A criação de um diagrama tático, por parte do utilizador, implica a sua definição espacial e temporal. Porém, e pela sua natureza, é impossível obter uma noção global de um conteúdo temporal. Nestes casos a solução passa, muitas vezes, pela sua transposição para uma representação espacial, aplicando-se uma metáfora apropriada.

A interface desenvolvida para a aplicação tenta facilitar ao utilizador a aquisição de uma adequada visão global do conteúdo espacial, em conjunto com a sua evolução temporal. Assim, um barco ao longo do tempo é representado pela sua replicação em diversos locais, complementada por uma linha que designa o seu percurso, à semelhança do que está representado na figura 2.



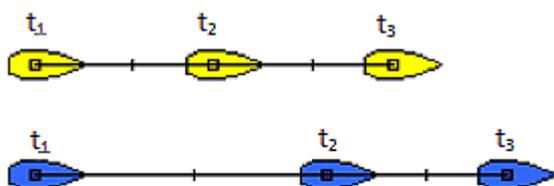
Figura 2 – Representação de um trajecto

No fundo, um trajecto contínuo é representado pela interpolação de um conjunto discreto de pontos pertencentes a ele. Múltiplas entidades podem viver num mesmo diagrama, sendo que as que partilham a mesma linha de trajecto são as diversas representações das instâncias de uma mesma entidade ao longo do tempo.

O exemplo dado para as embarcações das regatas expande-se às diversas entidade que podem constar num diagrama, sendo o mesmo princípio aplicado a estas. Entidades cujo percurso não seja definido por opção do utilizador são consideradas entidades de posição invariante no tempo. Há entidades que são sempre consideradas estáticas, como sejam as bóias ou outros objectos de sinalização.

Deste modo, um diagrama táctico é visto como um conjunto de entidades colocadas no espaço. A representação do diagrama agrega todas as instâncias dessas entidades, num determinado número de instantes, em sequência temporal crescente e com intervalos que podem não ser iguais entre si.

Considere-se o exemplo da figura 3. Se forem escolhidos, como amostragem temporal, os instantes t_1 , t_2 e t_3 , no respectivo diagrama são desenhadas as representações de 3 instâncias de cada uma das entidades não estáticas. A cada um desses instantes de tempo é atribuído um número de ordem, neste caso valores inteiros de 1 a 3. Os instantes intermédios, entre amostragens de entidades distintas, são equivalentes entre si relativamente à percentagem decorrida entre essas mesmas amostragens.



Números de ordem: $t_1 \equiv 1$ $t_2 \equiv 2$ $t_3 \equiv 3$

Figura 3 – Numeração dos instantes de tempo

A abordagem acima referida levou a uma implementação com base numa estrutura em lista para a representação espacial das diversas instâncias das entidades, complementada com referências entre os diversos elementos da lista para a representação espacial. Assim, cada entidade presente num diagrama possui um conjunto de caracterís-

ticas que a definem, como a cor e a orientação das velas, para o caso das embarcações das regatas. Várias propriedades são partilhadas pelos diversos elementos da lista, existindo também alguns atributos particulares de cada um. Finalmente, cada elemento representado possui referências para os elementos das instâncias imediatamente anterior e seguinte no tempo, caso existam (Figura 4).

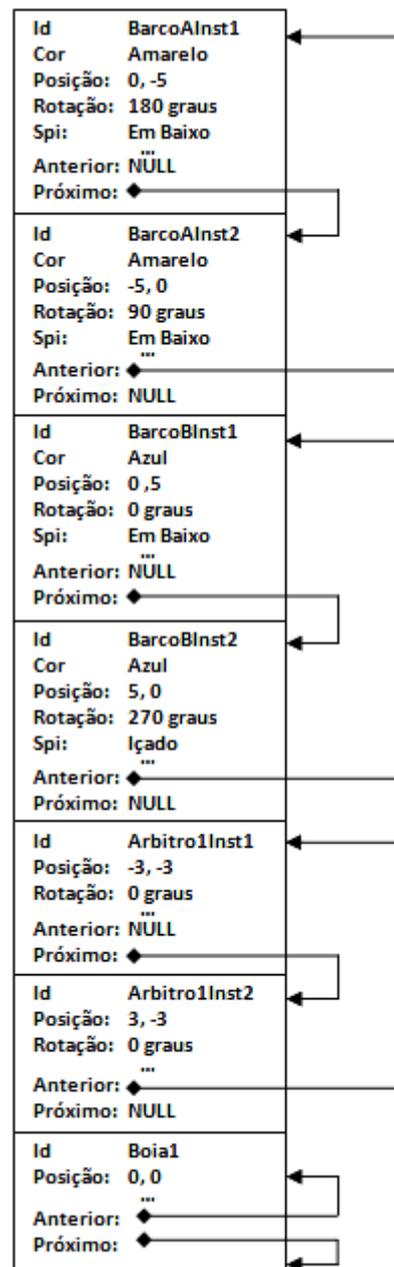


Figura 4 – Estrutura de representação de diagramas

6.3 Relação entre grafo e diagrama

Do ponto de vista de implementação, a abordagem passou por manter uma separação entre o grafo da cena e a estrutura do diagrama táctico. Assim, uma primeira estrutura de dados permite o armazenamento de cada um dos elementos lógicos do diagrama, tal como foi apresentado no ponto 6.2. O polimorfismo tem a sua relevância neste aspecto, permitindo a generalização de diversas proprie-

dades em objectos variados, mas permitindo também que cada um destes possa ter as suas características próprias e distintas. Como se pode constatar, nessa estrutura não houve quaisquer referências a entidades físicas. A abordagem na forma de lista duplamente ligada permite uma fácil navegação entre os vários instantes temporais de um objecto e simplicidade na remoção de instantes, entre outras vantagens. A estrutura do grafo de cena é mantida em separado da estrutura anterior, sendo gerida praticamente na íntegra pelo próprio motor gráfico.

O motor *Ogre 3D* possui uma hierarquia de objectos e transformações, tendo como característica relevante a separação entre as estruturas lógica e física do grafo, na forma apresentada pela figura 5 [Junk06].

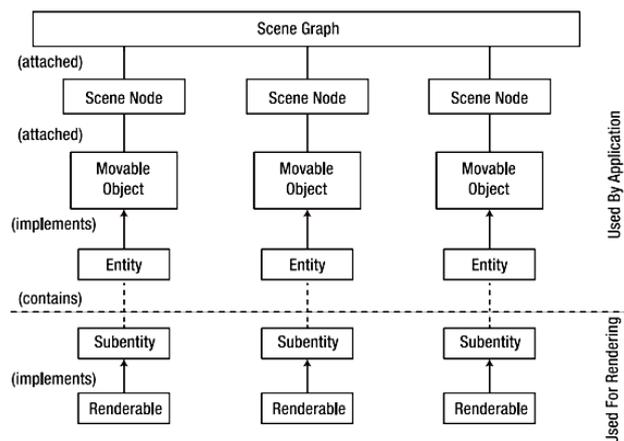


Figura 5 – Estrutura de uma cena *Ogre 3D*

Os elementos físicos têm a possibilidade de apontar para um objecto genérico, tendo este sido aproveitado para referenciar os elementos correspondentes na estrutura lógica, a partir de qualquer objecto ou sub-objecto existente no grafo. Cada um dos elementos na estrutura lógica do diagrama possui também um campo indicando o nó do grafo de cena a que corresponde. Assim, existe uma associação bidireccional entre os elementos lógicos do diagrama e a sua representação tridimensional no grafo da cena. Na figura 6 é apresentada uma esquematização gráfica da abordagem referida. Uma das grandes vantagens da estrutura apresentada é a elevada facilidade de conversão para um motor gráfico que substitua o *Ogre 3D*, desde que esse motor respeite uma estrutura hierárquica compatível com um grafo de cena, e que, preferencialmente, possua nos seus elementos uma forma de referenciar objectos de qualquer tipo. Em cada elemento da estrutura lógica, para além do apontador para o nó do grafo de cena correspondente à sua representação tridimensional, não existe qualquer outra referência que não seja independente do motor gráfico. Todos os restantes atributos estão representados de forma mapeada nos tipos básicos da linguagem de programação em causa, sendo essa a única dependência. Deste modo, a estrutura torna-se compatível com outro motor gráfico desde que a referência para o grafo de cena seja adaptada para a nomenclatura desse novo motor. No fundo, esta adaptação consistirá em redefinir um tipo de um apontador para um tipo

diferente. Por outro lado, é importante que esse motor gráfico possua um mecanismo que possibilite aos elementos visuais do grafo da cena indexarem, de certa forma, objectos de uma determinada classe. Esta característica torna possível associar, directamente, os elementos visuais às estruturas lógicas acima descritas, mantendo a relação bidireccional já referida. Naturalmente que, para identificar o elemento lógico correspondente a um determinado elemento visual, se poderia iterar sobre toda a estrutura lógica à procura da referência para o nó em que esse elemento se encontraria. Por esta razão, esta característica é simplesmente preferencial e não um requisito obrigatório.

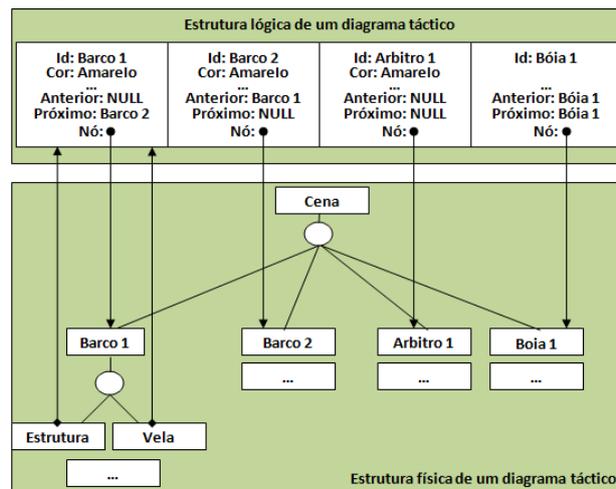


Figura 6 – Relações criadas entre Grafo e Diagrama

6.4 Cálculo dos trajectos

Um dos problemas principais a resolver na criação desta aplicação prendeu-se com os trajectos. Como já foi indicado, estabeleceu-se que o utilizador definiria tais trajectos pela colocação de barcos em posições intermédias.

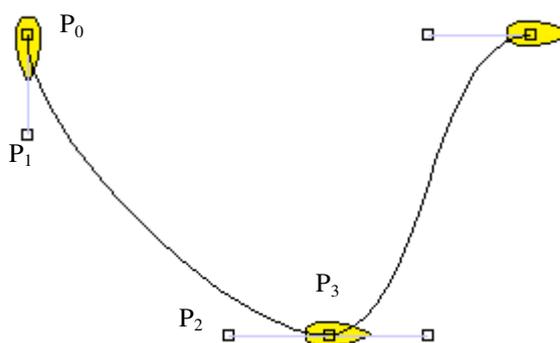


Figura 7 - Trajectos representados como curvas de Bézier

As curvas cúbicas de Bézier [Bart98] conseguem ser uma boa aproximação à solução do problema da interpolação, reduzindo a complexidade para a interface que poderia existir com a utilização de curvas de classe de continuidade superior, como sejam as curvas *B-Spline*. Com efeito, as curvas cúbicas de Bézier interpolam o ponto inicial (P_0) e o ponto terminal (P_3) de cada traço, para além de

se poder garantir continuidade paramétrica de classe C' nas junções de troços por escolha adequada da posição dos restantes pontos de controlo (P_1 e P_2) de cada troço em causa (Figura 7).

6.5 Gestão da animação

Os pontos apresentados anteriormente permitiram definir os objectos a posicionar num diagrama táctico, bem como estabelecer os trajectos de cada um. Deste modo, estão presentes os dados necessários para a construção de uma animação dos objectos, sendo os pontos de controlo intermédios calculados de forma a garantir a continuidade das primeiras derivadas, ou seja, será contínua a velocidade dos objectos na simulação.

Fazer uma animação consiste, em primeiro lugar, na replicação do grafo inicial. Este novo grafo, dito de animação, também possui, nos seus elementos, apontadores para a lista das instâncias das entidades presentes no diagrama, podendo assim obter-se quaisquer propriedades associadas. Durante o modo de animação, o grafo inicial encontra-se inactivo. Desta forma, terminar a animação e voltar ao modo de edição consistirá simplesmente em tornar novamente activo o grafo inicial, sendo o grafo da animação removido da memória.

6.6 Armazenamento dos diagramas

Justificando-se pela necessidade de integração, os diagramas são armazenados em formato XML. Foi também visto anteriormente que as representações visuais e lógicas dos diagramas tácticos são estruturas distintas, mantendo-se o grafo de cena separado da própria estrutura lógica do diagrama. A relação temporal entre os vários elementos constantes num diagrama é mantida por um sistema de apontadores referenciando as entidades anterior e seguinte.

A conversão da representação escolhida para XML, nomeadamente devido à abordagem por lista duplamente ligada utilizada para a representação da estrutura temporal, tornou-se um desafio interessante. Em primeiro lugar, o paradigma de programação utilizado levou a que a listagem de elementos consistisse num conjunto de objectos, relacionados entre si por intermédio de apontadores indicando quais os elementos anterior e seguinte na estrutura temporal. Converter os elementos para XML é um processo praticamente directo, bastando converter os atributos guardados. No entanto, repor um diagrama a partir de XML não é um processo tão directo quanto possa parecer à primeira vista, uma vez que a estrutura de lista duplamente ligada faz com que existam diversos elementos dependentes entre si. Assim, a leitura de uma só passagem originaria avanços e recuos nos apontadores até que estes ficassem totalmente repostos. Por exemplo, um elemento só pode ser construído quando o seu sucessor temporal já estiver pronto, de forma a colocar o apontador de elemento seguinte a apontar para ele. No entanto, este elemento só poderá ser construído quando o anterior também já o estiver, uma vez que ele será referenciado pelo apontador de elemento anterior. Perante esta situação de *deadlock* por dependência mútua, e de forma a tornar o

processo mais compatível com eventuais abordagens semelhantes de editores sem animação, o formato XML foi dividido em três regiões principais de estruturas independentes. Numa primeira parte são guardadas as informações atómicas sobre os diversos elementos do diagrama, tais como a cor das embarcações, posições das velas, etc. Eventualmente podem até ser criados novos diagramas para tipos de objectos diferentes, não afectando a independência. As referências temporais são guardadas numa segunda fase, sob a forma de par origem/destino, ignorando-se os campos nulos. Deste modo, a leitura de um diagrama XML parte de uma fase inicial, baseada na reposição espacial dos diversos elementos, em que estes são criados tal como os respectivos elementos no grafo de cena. As referências de elementos anteriores e seguintes são lidas numa segunda fase, a qual consiste numa passagem por todos os pares origem/destino guardados no ficheiro, actualizando-se os apontadores dos elementos a que correspondem. Finalmente, de forma a ser possível guardar também as durações referentes a cada um dos intervalos temporais definidos implicitamente, foi previsto, na estrutura dos ficheiros XML, um elemento no qual estes são descritos. A esquematização apresentada na figura 8 traduz o resultado final de representação em XML.

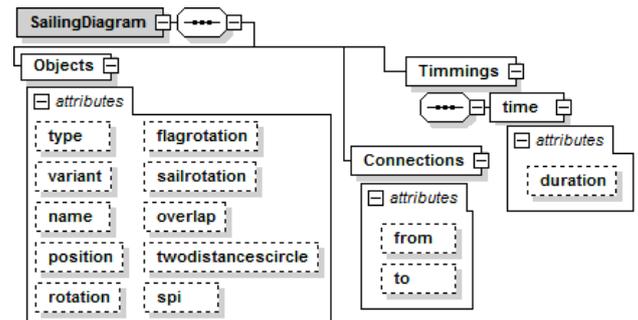


Figura 8 - Estrutura XML para diagramas tácticos

6.7 Interface Gráfica

Para incorporação na aplicação, foi ainda desenvolvido um sistema de interface gráfica com o utilizador (*Graphical User Interface* – GUI) para uso na API *Ogre 3D*. Verificou-se que os sistemas equivalentes existentes não se apresentavam adequados para utilização nesta aplicação. De facto, alguns deles encontram-se ainda em fase primária de desenvolvimento, com nítida falta de funcionalidade. É o caso da omissão de opções tão simples e necessárias como, por exemplo, a existência de caixas de diálogo ou mensagens de alerta. Por outro lado, outros sistemas GUI actualmente existentes apresentam demasiada complexidade, muita dela desnecessária para os objectivos pretendidos, como é o caso da *wxWidgets* [Smar07]. Deste modo, foi desenvolvido um sistema GUI baseado em eventos, com estruturas e controlos simples e hierárquicos. Esta estrutura é válida e aplicável directamente na aplicação desenvolvida, sendo também expansível para incorporação em qualquer aplicação *Ogre 3D*.

A estrutura baseada em eventos foi construída de modo a evitar algumas características não desejadas, como a exis-

tência de classes com associação mútua. Assim, a estrutura conta com um elemento base central, denominado *GUIElement*, a partir do qual derivam todos os elementos de interface. *GUIElement* deriva de *GUICallback* para garantir um elemento de nível superior que consiga evitar associações mutuamente dependentes. Assim, é definida uma classe *GUIListener*, representante dos objectos responsáveis por captar os eventos, estando esta associada a *GUIElement*. Os *GUIElements* que possuem um *GUIListener* não nulo passar-lhe-ão todas as informações dos eventos disparados em execução. Como o *GUIListener* pode receber informações de múltiplos *GUIElements*, os eventos por este recebido contam com informação de qual o elemento de onde o evento partiu, indicado sobre a forma de *GUICallback* para evitar o já referido problema de associação mútua. Esta abordagem pode ser clarificada pela relação de classes apresentada na figura 9.

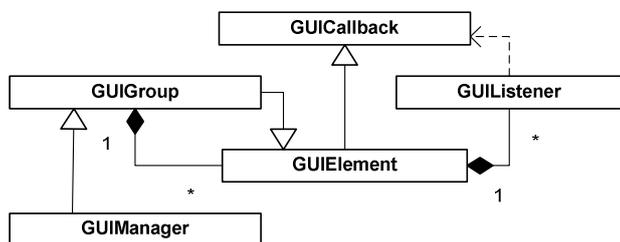


Figura 9 – Classes para gestão de eventos da GUI

Existe uma estrutura *GUIGroup* que é um *GUIElement* particular, agrupando um conjunto de outros objectos também da classe *GUIElement*. Este é ainda expansível sobre a forma de *GUIManager*, que é um *GUIGroup* particular possuindo a característica de reencaminhar, para o seu *listener*, os eventos de todos os elementos que o compõem. No caso geral, o *GUIGroup* utiliza os eventos dos elementos que o compõem exclusivamente para uso interno, não sendo feito reencaminhamento dos mesmos ou, quando tal é feito, estes vão já assinados pelo grupo e não pela componente de forma independente.

Os controlos disponibilizados pelo sistema GUI são então associados directamente a *GUIElement* nos casos mais simples ou a *GUIGroup* quando se trate de um elemento composto. Abaixo destes elementos é naturalmente possível expandir a hierarquia, para que existam elementos do sistema GUI que se particularizam na forma de outros.

A estrutura actual conta com diversos tipos de controlo, hierarquizados na forma apresentada na figura 10.

O estado actual do sistema apresenta como principal vantagem a forma simples como se estrutura, sendo possível, com poucas linhas de código, criar-se um botão na interface. Também é simples a própria base dos botões utilizados, recorrendo somente a uma imagem para cada um deles e sendo o efeito de clique representado com uma diminuição do tamanho dessa imagem, dando a sensação do botão ter sido premido. O destaque aquando do posicionamento do rato sobre um botão é conseguido só com uso de transparências, sendo o botão exibido com maior opacidade quando o cursor do rato se lhe sobrepõe. Estas características permitem que a criação de um botão con-

sista na inicialização de um objecto da classe correspondente, sendo só necessário especificar qual a imagem a utilizar, o seu posicionamento e as suas dimensões. Qualquer projecto *Ogre 3D* que necessite de sobreposições gráficas 2D de funcionamento simples e eficaz poderá recorrer ao sistema GUI desenvolvido, usufruindo, neste âmbito, de uma abstracção sobre os conceitos de criação de materiais e *overlays* presentes neste motor.

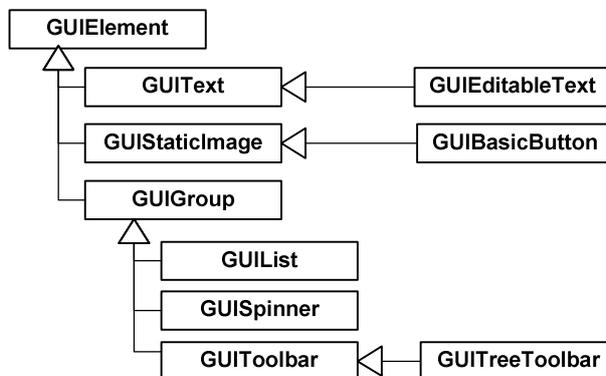


Figura 10 - Hierarquia de classes da GUI

7. APLICAÇÃO DESENVOLVIDA

O protótipo já construído revela simplicidade na utilização, tendo sido utilizadas abordagens e metáforas comuns em aplicações editoras de cenas. Estão aí presentes os tradicionais mecanismos de *drag-and-drop* e *region selection*.

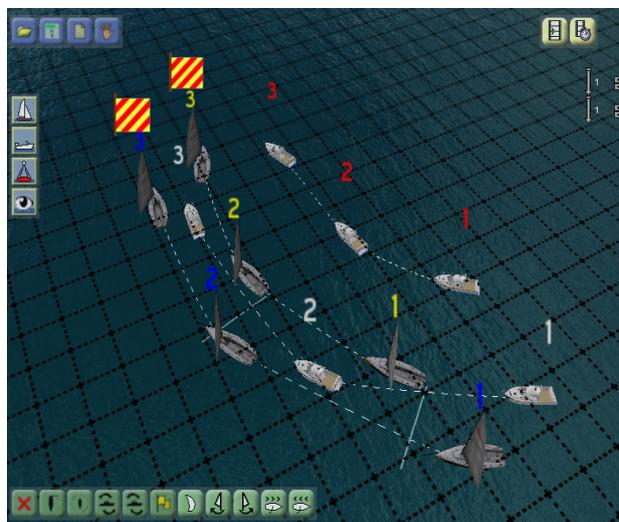


Figura 11 - Vista geral durante uma sessão

Na figura 11 é exibida uma vista geral da janela de execução da aplicação, após a criação de um diagrama. Este retrata uma situação ocorrida em 2000, aquando do evento *America's Cup* realizado em Auckland. Neste caso, ambos os barcos protestaram no instante de ordem 3, exibindo a bandeira respectiva (*Y-Flag*).

No topo da janela da aplicação, de forma a ser semelhante às abordagens tradicionais que recorrem a uma barra de menu, são disponibilizados os botões que permitem guardar, abrir ou criar novos diagramas, sendo também disponibilizada uma opção para sair do programa.

À esquerda é possível observar-se a barra de selecção de objectos. É aí que se escolhe o objecto que se pretende colocar em cena (barcos de regata, embarcações dos juizes, ou marcas auxiliares para o trajecto), ou se simplesmente se quer navegar por esta sem inserir quaisquer elementos no diagrama.

A edição dos objectos em cena é conseguida através de um mecanismo comum de *drag and drop*. Um clique em qualquer um dos objectos permite a sua selecção e há mecanismos de selecção simultânea de vários objectos.

Para cada embarcação, é possível criar novos pontos de controlo para a trajectória que esta irá descrever. Bastará que o utilizador faça *drag-and-drop*, a partir de uma posição existente de um barco para um novo ponto qualquer na água, com o botão direito do rato. Mesmo em trajectos já definidos, o processo descrito anteriormente permite acrescentar pontos intermédios com facilidade.

Na zona inferior da janela é possível observar-se a barra de edição dos elementos do diagrama. Nela é possível alterar diversas características nos elementos que estiverem seleccionados: rotação da embarcação, rotação das velas, colocação de marcas auxiliares, especificação da velocidade da embarcação, colocação de bandeiras indicando protestos por parte das embarcações, entre outras.

No lado direito são disponibilizados pequenos controlos, nos quais se pode definir a duração de cada um dos intervalos de tempo (2 intervalos para o caso da figura 11).



Figura 12 - Vista na primeira pessoa em modo reprodução

Finalmente, o canto superior direito possui um botão que permite alternar entre o modo de edição e o modo de reprodução. Ao escolher-se o modo para reprodução, os barcos começam a deslocar-se no trajecto definido. A barra inferior, que permitia a edição dos objectos em cena, é substituída por uma outra, a qual permite o controlo da animação como se fosse um vulgar *media player*, com comandos tais como *play*, *pause* e *stop*.

Para além da câmara livre, que já era utilizada no modo de edição, é agora possível ver a cena a partir do interior de qualquer uma das embarcações ou marcas em cena,

bastando, para isso, clicar no botão para esse efeito. Na figura 12 é possível ver a aplicação em modo de reprodução, com câmara de perspectiva de primeira pessoa.

Neste caso é apresentado o ponto de vista para a situação da figura 11, em que existia um protesto de ambas as embarcações. A decisão dos árbitros passou pela penalização do barco azul, porque mediante a situação de sobreladeamento existente, este deveria ter-se mantido na sua direcção, sem encurtar espaço ao barco amarelo. A vista de perspectiva permite representar a forma como a situação foi observada pelos juizes, com o barco azul à esquerda e o amarelo à direita, podendo-se perceber melhor a noção desse ponto de vista, identificando-se a necessidade que o barco amarelo teve de se afastar. Na figura 13 mostra-se que é também possível verificar a forma como os juizes auxiliares constataram o sobreladeamento. Este tipo de observação era impossível com as abordagens bidimensionais existentes anteriormente, sendo esta a principal vantagem da utilização da solução 3D proposta neste artigo.



Figura 13 – Vista na primeira pessoa dos juizes auxiliares

8. CONCLUSÕES E TRABALHO FUTURO

Em relação à aplicação desenvolvida, os aspectos principais do TSS foram mantidos, o que permite uma adaptação acessível aos potenciais utilizadores da aplicação. Adicionalmente, a diferença no aspecto gráfico aliada à abordagem 3D permite um novo conjunto de funcionalidades que se apresentam como uma mais-valia na análise táctica em *Match Racing*. A aplicação desenvolvida apresenta bons níveis de qualidade gráfica, apesar de não igualarem a do *Virtual Eye* ou do *Virtual Skipper*. Estes dois programas apresentam superioridade em questões de pormenor, como a existência dos velejadores dentro das embarcações e representação das cidades junto à zona da regata. No entanto, estes aspectos não são significativos no âmbito de funcionamento do analisador táctico, podendo até funcionar como elemento de distração.

Como o trabalho foi desenvolvido com proximidade e ligação a pessoas associadas à modalidade, houve uma boa convergência entre as funcionalidades implementadas e as necessidades existentes. A aplicação foi testada por

diversos elementos ligados à modalidade de *Match Racing*, incluindo atletas e juizes, com *feedback* positivo.

A característica importante de possibilidade de integração com outras aplicações foi conseguida como pretendido, sendo que os diagramas gravados em XML podem ser utilizados por outras aplicações que consigam aceder a este tipo de ficheiros. Por exemplo, é possível desenvolver uma folha XSLT [Clar99] que converta os diagramas para VRML [W3D08a] ou X3D [W3D08b] para disponibilização numa página Web.

A abordagem utilizada, em relação à representação tridimensional, englobou as componentes espaciais e temporais com separação entre conteúdo de diagrama e grafo de cena. Resultou numa solução prática, eficaz, integrável e expansível, pelo que poderá servir de base para aplicações similares, numa vertente semelhante para outra modalidade ou até noutra tipo de edição espaço-temporal.

A não inclusão de muitas outras técnicas do actual estado da arte para aumentar o realismo gráfico garante a fluidez da aplicação em qualquer computador recente. No entanto poderão ser incorporadas, gradualmente, novas características que melhorem o grafismo da aplicação com técnicas eficientes que vão surgindo, desde que não interfiram negativamente na capacidade de análise táctica.

O sistema GUI desenvolvido para incorporação no projecto cumpriu os seus objectivos, tendo um funcionamento simples e fácil integração em qualquer projecto *Ogre 3D*. Este facto leva a que seja interessante prosseguir o seu desenvolvimento, eventualmente como projecto distinto, de forma a serem criados novos controlos e novas estruturas na hierarquia construída, tornando-a mais completa e aplicável a uma maior gama de aplicações, constituindo-se como uma contribuição interessante para a comunidade *Ogre 3D*.

Um outro ponto importante para expansão em trabalho futuro seria uma possível incorporação de dados oriundos de um dispositivo GPS. Numa primeira fase esta incorporação poderia basear-se numa utilização *a posteriori*, em que as embarcações realizassem um determinado treino ou prova, possuindo nela um dispositivo GPS que armazenasse as coordenadas em intervalos de tempo curtos. Os ficheiros poderiam depois ser importados para a aplicação. Existe, naturalmente, o constrangimento dos formatos fechados de GPS, sendo que esta adaptação teria sempre que ser gradual. Numa segunda fase seria interessante explorar alternativas para que a comunicação de dispositivos GPS e o programa pudessem funcionar em tempo real, para se conseguir um funcionamento de reprodução remota semelhante ao *Virtual Eye*, mas agora aliada às funcionalidades actuais de edição e análise.

Em resumo, pode-se afirmar que os objectivos propostos foram cumpridos, sendo o resultado final uma aplicação com aplicabilidade prática, atingindo os requisitos inicialmente identificados. A análise táctica em *Match Racing* conta agora com uma nova ferramenta, a qual possibilita aos técnicos e atletas estudar alguns aspectos

que anteriormente não eram contemplados. Por outro lado, a comunidade da computação gráfica conta com a experiência de concepção de um exemplo prático, onde foram criadas estruturas lógicas próprias e um novo sistema GUI para o motor gráfico *Ogre 3D*.

9. REFERÊNCIAS

- [Anim03] Animation Research Ltd. Virtual Eye (America's Cup).
<<http://ar1.co.nz/americas-cup>>
- [ACSA07] AC Management SA. 33rd America's Cup Competition Regulations. 2007.
- [Bart98] R. Bartels et. al. Bezier Curves. An Introduction to Splines for Use in Computer Graphics and Geometric Modeling, C10, Morgan Kaufmann, 1998.
- [Blau05] J. Blau. FIFA boots chip ball from 2006 soccer World Cup. *Infoworld Homepage*, 2005.
- [Bull01] M. Bull et. al. Benchmarking Java against C and Fortran for Scientific Applications. *Proceedings of the 2001 joint ACM-ISCOPE conference on Java Grande*, pp. 97–105. 2001.
- [Chap07] M. Chaplin. Referees named for UEFA Euro 2008 – Communication System. UEFA Euro 2008 – Austria/Switzerland (Website), 2007.
- [Clar99] J. Clark. XLS Transformations (XSLT) Version 1.0 – W3C Recommendation. *World Wide Web Consortium - Web Standards*, 1999.
<<http://www.w3.org/TR/xslt>>
- [Gosl04] M. Goslin. The Panda3D graphics engine. *Computer*, 37(10), pp. 112–114, 2004.
- [Jorr00] J. Tyberghein. Open Source 3D Package for Game Creation:Crystal Space. *Linux Magazine*, 2000.
- [Junk06] G. Junker. Pro Ogre 3D Programming, Apress, 2006.
- [Meni05] H. Menin, J. Cuttler. Basic Match Racing, Understanding the Game, 2005.
- [Nad07] Nadeo. Virtual Skipper PC DVD-Rom, 2007.
<www.virtualskipper-game.com>
- [Pero08] H. Peronneaus. Tactical Sailing Situations. *TSS - Tactical Sailing Situations – Official Site*, 2008
<<http://tss.peronneau.net>>
- [Smar07] J. Smart, et. al. wxWidgets 2.8.7: A portable C++ and Python GUI toolkit, 2007.
- [Swis05] K. Swisher. Swinging in cyberspace - New software can analyze your every move. Does it help?. *The Wall Street Journal*, 2005.
- [W3D08a] Web3D Consortium, Inc. VRML97 and related specifications.
- [W3D08b] Web3D Consortium, Inc. Information technology - Computer graphics and image processing - Extensible 3D (X3D).
- [Wils06] K. Wilson. Why C++?. *Gamearchitect.net - Musings on game engine structure and design*, 2006.
<<http://www.gamearchitect.net/Articles/WhyC++.html>>