

Um controle autônomo de um carro de corrida

Pedro Sampaio Vieira, Thiago Ribeiro de Azeredo, Antônio Alves dos Santos Neto,
Ítalo Oliveira Matias, Dalessandro Soares Vianna

Universidade Candido Mendes – Campos, Brasil

RESUMO

A utilização da Inteligência Artificial em jogos eletrônicos está crescendo a cada dia, e com isso aumenta-se o grau de exigência, principalmente em relação à performance do jogo, devido ao fato de sempre se tentar aproximar de um ambiente real. Levando-se em consideração essas afirmativas, este trabalho propõe a utilização de estratégias para o controle autônomo de um carro em um ambiente de corrida, com o foco no processo de treinamento do veículo. Entre tais estratégias estão: algoritmo genético, representação da pista por setores e checkpoints. O sistema simula a atuação de um jogador humano, atuando como se fosse um piloto automático, guiando, acelerando e freando o carro na pista.

Palavras Chave: Algoritmo genético, jogos eletrônicos, Inteligência Artificial e checkpoints.

Contato:

{pesvieira, thiagoribeiro, antonioasn}@gmail.com
{dalessandro, italo}@ucam-campos.br

1. Introdução

O fato da evolução do realismo gráfico dos jogos atuais ser bastante avançado, tem uma influência direta na aplicação das técnicas de Inteligência Artificial em um jogo. Hoje em dia, a cada novo jogo lançado, tem-se notado um grande aumento no nível de detalhamento gráfico, devido principalmente a evolução contínua das placas gráficas. Com isso, surge uma grande exigência de que a inteligência do jogo tenha que acompanhar essa evolução, porque a expectativa dos jogadores quanto ao comportamento dos NPC (*non player character*, personagens não controlados pelos jogadores) está maior, devido ao alto realismo gráfico.

A partir dessa análise, este trabalho propõe proporcionar um comportamento autônomo do veículo através da aplicação de estratégias de Inteligência Artificial no controle do mesmo. Onde tais técnicas têm a função de contribuir para um comportamento competitivo do NPC numa pista de corrida. O sistema atua realmente como se um jogador humano estivesse controlando o seu personagem no jogo.

Com isso, utiliza-se a representação da pista por setores e estruturas de checkpoints, o que torna possível a “visualização” da pista pelo NPC. Com base nessa estrutura, o algoritmo genético proposto visa encontrar os melhores pontos de referência nessas estruturas, para que se forme o traçado, permitindo o mesmo completar a volta no melhor tempo possível em relação a um jogador humano.

2. Estratégias de Controle do Carro

O problema consiste em um simulador de corrida no qual há a necessidade de controlar todos os movimentos de um veículo NPC durante a prova.

Este trabalho está focado em estratégias que permitem o carro completar uma volta na pista no menor tempo possível, tratando todo o controle de direção do veículo, de aceleração e de frenagem. Porém em [AZEREDO et al. 2006] são desenvolvidos técnicas para o tratamento das outras ações que uma aplicação desse tipo necessita.

2.1. Direção do Carro

Apesar de haver técnicas que visam apenas passar a sensação de que os oponentes estão correndo (como, por exemplo, reposicionar o carro na cena em pontos pré-determinados), há outras que buscam simular os pilotos, como se fossem pilotos automáticos.

Para o controle direcional do carro neste trabalho, partiu-se dessa idéia de criar um piloto automático. Desta forma o piloto enfrenta todos os desafios enfrentados por um jogador humano, como: acelerar, frear e conduzir o carro na pista.

2.1.1. Representação do Trajeto em um Ambiente Bidimensional

Esta etapa resume-se em representar a pista como uma estrutura de simples visualização para o piloto, fazendo com que o mesmo se localize na pista.

Assim, há a necessidade de uma representação da pista que seja consistente e que não prejudique o processamento. Para isto, foram utilizados [BIASILLO 2002a; BIASILLO 2002b] que sugerem a representação da pista por setores, em que cada setor é um paralelogramo.

Os pontos que representam os vértices dos setores no espaço são fixados sobre a pista, como mostrado na figura 1. Os setores da pista sempre compartilham dois lados com outros setores, sendo este compartilhamento denominado interface de conexão. Estas interfaces interligam o setor X com o setor $X-1$ e $X+1$.

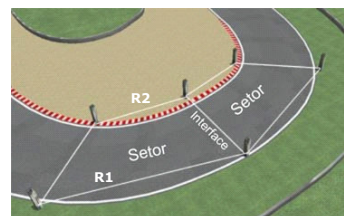


Figura 1: Demonstração dos paralelogramos dos Setores.

Cada interface é um segmento de reta, e desta forma, ela é representada por dois pontos (A e B) que definem a sua localização. Para facilitar os cálculos e reduzir o custo do processamento, a posição no eixo Y (que representa a altura) é desconsiderada, assim os pontos serão formados pelo eixo X e Z.

O critério para o posicionamento dos setores na pista é definido como tão distante da interface anterior quanto possível, de forma que se forem traçadas retas ($R1$ e $R2$) entre os pontos extremos da interface anterior e atual, estas retas estejam dentro da pista conforme figura 1, mantendo as interfaces sempre perpendiculares à lateral da pista. Nas curvas, mantê-las perpendicular à tangente.

2.1.2. Determinando em qual setor está o carro

Para determinar o setor atual do carro é necessário saber se o carro atravessou alguma interface. Esta análise é feita utilizando uma inequação do primeiro grau: $ax + by + c > 0$, onde a reta é formada pelos pontos de uma interface. A expressão retorna valores maiores ou menores que 0 (zero) relacionados ao lado da reta que o carro está, assim se há mudança de sinal no teste, sabe-se que o carro foi para outro setor.

2.1.3. Guiando para um ponto

Fazer o piloto automático guiar o carro para um ponto P requer inicialmente ter tal ponto. Tendo uma pista inteira, há a necessidade de ter vários destes pontos. Dessa maneira, a estrutura dos setores foi utilizada para armazená-los. Foi definido que os pontos guia (pontos que definem o traçado imaginário), deveriam pertencer ao segmento de reta de uma interface. Desta forma, em cada setor o carro deve guiar para o ponto guia da interface diretamente a sua frente.

Para que o carro seja guiado para um ponto foi utilizado um cálculo vetorial, a fim de determinar qual o ângulo que as rodas devem girar, conforme o algoritmo 1. Este procedimento é executado a cada passo para corrigir a direção da trajetória.

Algoritmo 1: Calcular ângulo [ADZIMA 2002].
 01 direção = alvo – posição_atual; $t = v_0 / 9.8$;
 02 $fX = \text{ProdutoInterno}(\text{direção}, X_ori)$;
 03 $fZ = \text{ProdutoInterno}(\text{direção}, Z_ori)$;
 04 retornar $\text{ArcoTangente}(fX, fZ)$;

Onde *alvo* representa a posição do ponto guia, *posição_atual* é a posição do carro, X_ori é o vetor de orientação do carro no eixo X e Z_ori é o vetor de orientação do carro no eixo Z.

Porém, em alguns momentos, como por exemplo, se houver uma perda do controle do carro e ele acabar parando de lado na pista, o piloto automático pode tentar virar a roda 90° para buscar o ponto guia, assim foi necessário criar um limite em 1 radiano (57.3°) para cada lado.

2.1.4. Suavização do traçado

Nesta parte do desenvolvimento, ao carregar os pontos guia de um arquivo, o carro já anda sozinho na pista, porém de uma forma bastante precária. Um dos

problemas neste momento é a mudança brusca do ponto guia do carro ao se trocar de setor, fazendo com que as rodas virem rapidamente, levando o carro a perder o controle.

Este problema é minimizado adicionando uma função que interpola o ponto guia atual e o próximo. Assim, o piloto automático deixa de guiar para um ponto (forma discreta) em cada setor para guiar para um ponto diferente a cada momento (forma contínua). Esta função é definida como:

$$(\text{angulo1} \times \text{dist}) + p \times (\text{angulo2} \times (1 - \text{dist})),$$

onde *angulo1* é o quanto a roda deve virar para passar no ponto guia da interface a frente (como apresentado anteriormente), *angulo2* é o quanto a roda deve virar para passar no próximo ponto guia, *dist* é o percentual (de 0 a 1) da distância percorrida pelo carro dentro do setor atual, e p é um peso (entre 0 e 1) dado para diminuir a influência do próximo ponto guia. Este resultado pode ser observado na figura 2, onde a primeira linha representa uma função genérica baseada no andar do carro passando pelos setores, a segunda linha representa a mesma função com o uso da interpolação proposta.

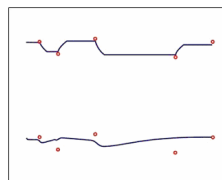


Figura 2: Representação da função de interpolação.

Com esta função, quanto mais próximo do ponto guia da interface à frente, menor sua influência e maior a influência do ponto guia seguinte. Evitando uma mudança súbita, levando o carro a manter-se estável e passando uma impressão mais humana na forma de condução do piloto automático.

2.2. Aceleração e Frenagem

Nesse momento o carro apenas acompanha um traçado ditado pelas estruturas de setores, completando a volta sem ter a preocupação de sair da pista. Com isso ele acaba atravessando as curvas passando pela grama e batendo no muro de proteção.

Esse problema ocorre pelo fato de que mesmo com uma linha guia imaginária que o carro deva seguir, a física do carro e a alta velocidade fazem com que ele escape nas curvas.

Para resolver esse tipo de problema é que surge a necessidade de trabalhar especificamente no controle de aceleração e frenagem.

Para começar a trabalhar nesse controle, optou-se por utilizar uma estrutura de *checkpoints* (pontos de checagem) para criar pontos especiais (estratégicos) com o objetivo de auxiliar o processo. Esses pontos são exatamente o início e o fim de cada curva da pista. Os *checkpoints* que marcam as entradas de curva ajudam no controle do freio e os que marcam as saídas de curva ajudam no controle da retomada da aceleração.

Agora com os pontos de referência posicionados, já é possível desenvolver um meio para que o carro possa

realizar as curvas sem escapar, uma vez que o mesmo consegue “enxergá-las” na pista.

Para melhor entendimento, pode-se tomar como base uma curva como exemplo para toda a explicação. Tudo que será mostrado para esse exemplo é aplicado às outras curvas da pista.

Porém, para cada curva na pista foi definido, de antemão, uma velocidade máxima na qual o carro tem condições de completar tal curva sem sair da pista. Logo essa curva em questão tem um valor de velocidade v definido para ela. Esses valores de v , para cada curva, foram obtidos através da média de velocidade em que o carro, ao correr na pista, fazia especificamente em cada curva, tendo como amostragem voltas feitas por algumas pessoas (jogadores humanos). A figura 3 mostra a curva com suas propriedades que fazem parte deste exemplo.

Com isso, agora é necessário saber a qual distância d do *checkpoint* de freio o carro precisa começar a frear de fato, para que ao chegar no *checkpoint*, já esteja com a velocidade necessária para realizar a curva.

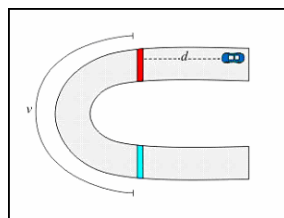


Figura 3: Esquema das variáveis do controle.

Em Beckman [1991], há dois modos de enxergar esse problema. O primeiro ponto de vista é o fato dos freios trabalharem com a transformação de energia cinética em calor. Já pelo outro modo, assume-se que todo esse sistema de frenagem funcione perfeitamente e que o processo esteja limitado não mais pelo atrito entre a pastilha e disco de freio, mas sim pela interação dos pneus com o solo. Considerando assim, o sistema de frenagem da pastilha com o disco de freio ideal.

Este trabalho optou pela segunda alternativa, pelo fato de que um jogo de entretenimento não requerer uma precisão tão alta. Isso faz com que se possa considerar que os pneus irão fornecer uma desaceleração constante de $1G$, simplificando os cálculos sem perder a precisão desejável. A expressão (1) mostra o valor aproximado de $1G$.

$$\begin{aligned} (1) \quad 1G &\equiv a = 9,8 \text{ m/s}^2 & (2) \quad t &= v_0 / a \\ (3) \quad \Delta s &= v_0 t - \frac{1}{2} a t^2 \end{aligned}$$

O tempo que o carro leva para frear a uma determinada velocidade v_0 pode ser calculado de maneira simples pela fórmula da aceleração constante (2). Com o tempo de frenagem definido, já se pode calcular a distância necessária para frear através da fórmula do deslocamento (3).

A partir dessa análise foi desenvolvida uma estratégia, com base nas fórmulas físicas apresentadas acima, para controlar a aceleração e freio do carro nas curvas. Como Beckman propõe uma técnica de

desaceleração até o veículo parar, foi necessária realizar uma adaptação para atingir o objetivo desse trabalho. O algoritmo 2 ilustra o método usado pra tal controle.

Algoritmo 2: Controle de Aceleração e Frenagem.

```

01 Se (DistanciaCpFreio == DistX) então
02      $t = v_0 / 9.8;$ 
03      $d = v_0 t - \frac{1}{2} \times 9.8 \times t^2;$ 
04 Se (DistanciaCpFreio < d) então
05     Se ( $v_0 > v$ ) então
06         Frear();
07     Senão
08         SetVelocidadeMáxima(v);
09 Senão
10     SetVelocidadeMáxima(velMax);
11 Se UltrapassouCpAcelerador() então
12     SetVelocidadeMáxima(velMax);
13 Senão
14     SetVelocidadeMáxima(v);

```

Onde as constantes $DistX$ e $velMax$ que indicam, respectivamente, o ponto (como referência a distância para o *checkpoint* de freio) em que será feito o cálculo do momento que deverão ser acionados os freios, e a velocidade máxima que o carro pode alcançar. As outras variáveis presentes no algoritmo já foram apresentadas nos parágrafos anteriores.

O passo 01 indica o momento em que deverá ser realizado o cálculo da distância de frenagem (d), e em seguida, nos passos 02 e 03, é calculada a distância d através da aplicação das fórmulas citadas anteriormente. O passo 04 verifica se o carro já atingiu a distância em que deve começar a frear. Se a condição for verdadeira, o carro começa a frear até que esteja com a velocidade máxima v necessária para realizar a curva, como é mostrado nos passos 05 a 08. Finalizando, os passos 11 a 14 controlam a re-aceleração, onde é verificado se o carro já passou pelo *checkpoint* de aceleração. Essa parte do algoritmo controla a velocidade do carro no momento em que o mesmo está fazendo a curva, e quando deve voltar à sua aceleração máxima após ultrapassar o ponto de controle.

Esse sistema permite que o carro agora respeite as curvas e consiga completar a volta dentro da pista.

2.3. Treinamento do Carro

Até o atual momento, o piloto automático já consegue guiar o carro com qualidade. O problema é encontrar bons pontos guia que levem o carro a fazer uma volta rápida e ficar o máximo possível dentro da pista. Encontrar tais pontos se torna uma tarefa complexa devido à incidência das leis da física no ambiente. Para tal, um AG foi implementado.

Como neste trabalho uma solução é formada por uma seqüência de pontos guia (formando uma volta), a população inicial foi criada a partir de diversas voltas dadas pelos jogadores humanos (capturando os pontos onde o carro passou sobre as interfaces) e pela solução montada para testes com o carro, completando 35 indivíduos. Os cromossomos (dados que montam uma

solução) são formados pelo ponto onde o carro atravessou uma interface, o tempo gasto para atravessar um setor e o tempo em que o carro ficou fora da pista naquele setor. Conseqüentemente, cada cromossomo representa um setor.

Como o objetivo deste AG é encontrar uma solução que se ajuste à forma de guiar do piloto automático e à física do carro, minimizando o tempo gasto numa volta (tempo_total) e minimizando o tempo do carro fora da pista (tempo_fora_da_pista), sua função objetivo é definida como:

$FO = \text{tempo_total} + \text{peso} \times \text{tempo_fora_da_pista}$
onde peso = 1,5.

Cada iteração consiste em:

- Selecionar entre 2 e N/2 pais aleatoriamente;
- Efetuar o cruzamento entre esses pais, onde para cada cromossomo c , um dos 30% melhores pais em c repassa seu cromossomo c para o filho;
- Com 5% de chance, gerar mutação deste filho, escolhendo três cromossomos e alterando aleatoriamente a posição do ponto guia, porém dentro de um limite. Esta mutação cria um novo indivíduo, mantendo o filho original. Caso ocorra a mutação, os próximos passos devem ser efetuados para os dois filhos;
- Fazer o carro dar uma volta na pista com o filho gerado para preencher o tempo em cada setor e o tempo fora da pista;
- Verificar se este novo filho é melhor que o pior dos pais que o gerou. Caso seja, substitui tal pai pelo novo filho.

3. Resultados Computacionais

Como pode ser visto na tabela 1, uma volta na pista pelo piloto automático, no início do treinamento, dura em média de 83 segundos, com função objetivo igual a 115. A cada iteração do algoritmo genético o piloto automático deve dar uma volta na pista. Desta forma, para realizar centenas de iterações seriam necessárias algumas horas.

Como esse processo de treinamento não visa uma solução ótima, e sim a melhor solução possível, definiu-se utilizar como critério de parada a quantidade de iterações. Sendo $Q_{de_it} = 300$ iterações.

Tabela 1: Parciais do treinamento do carro.

| Nº da volta | Tempo (s) | Função Objetivo |
|-------------|-----------|-----------------|
| 1 | 88 | 115 |
| 100 | 82,3 | 109 |
| 150 | 72,3 | 101 |
| 200 | 72,1 | 83,7 |
| 300 | 69,1 | 71,8 |

Os resultados conseguidos com a configuração descrita comprovam o poder do algoritmo genético. Ao final das 300 iterações, o tempo de uma volta foi reduzido para 69,1 segundos, sendo a função objetivo igual a 71,8. Desta forma, o tempo que o carro fica fora

da pista é de 1,8 segundos. Porém esse tempo fora da pista está diretamente ligado à geometria dos setores, pelo fato deles não preencherem a pista por completo.

Um jogador experiente e um iniciante realizam a volta em 61,1 segundos com FO igual a 62,3 e 78,4 segundos com FO igual a 85,9, respectivamente.

Como foi utilizado um conjunto pequeno de soluções para formar a população inicial, há pouca chance de um resultado que tenda para a solução ótima do problema ser encontrado. Apesar deste fato, a solução encontrada já permite um bom comportamento do veículo na pista de corrida.

4. Conclusão e Trabalhos Futuros

Após da análise dos resultados obtidos, pode-se concluir que o AG proposto é capaz de trazer bons resultados para o problema de aprendizado do traçado de um carro em uma corrida.

Tal AG tem como base a estrutura de setores, que demonstrou ser uma boa alternativa de baixo custo computacional para a representação da pista. E que juntamente com a função de interpolação dos pontos e o sistema de frenagem, tornou o comportamento do carro mais suave e estável, possibilitando um aproveitamento maior do algoritmo de treinamento. Logo, esse bom resultado alcançado só foi possível pela eficiência desses sistemas auxiliares.

Porém, assim como um jogador humano melhora seu desempenho na corrida a cada volta, o AG proposto neste trabalho também pode melhorar. Um objetivo futuro desse trabalho é estender o aprendizado a tempo real. O que faria com que o carro pudesse absorver as ações de seus adversários, adaptando seu comportamento na corrida.

Por fim, trabalhando mais no processo de frenagem, com a criação de um sistema de análise dinâmico, traria uma maior liberdade e diminuiria o tempo gasto para o carro completar uma volta.

Referências

- ADZIMA, J., 2002. Competitive AI Racing under Open Street Conditions. In: *RABIN, S. AI Game Programming Wisdom*. New York: Charles River Media. p.460-472.
- AZEREDO, T.R., VIEIRA, P.S., SANTOS NETO, A.A., MATIAS, I.O., 2006. *Inteligência Artificial aplicada à tomada de decisão em jogos eletrônicos*. 2006. 77 f. Monografia (Bacharel em Ciência da Computação) – Universidade Candido Mendes – Campos, RJ.
- BECKMAN, B., 1991. *The Physics of Racing*. Burbank.
- BIASILLO, G., 2002a. Representing a Race Track for the AI. In: *RABIN, S. AI Game Programming Wisdom*. New York: Charles River Media. p.439-443.
- BIASILLO, G., 2002b. Racing AI Logic. In: *RABIN, S. AI Game Programming Wisdom*. New York: Charles River Media. p.444-454.