Desafio da
Semana Acadêmica de TSI 2015

Introdução

O desafio desse ano será um jogo de corrida: **TSI Racing**. A Figura 1 é uma imagem do jogo.

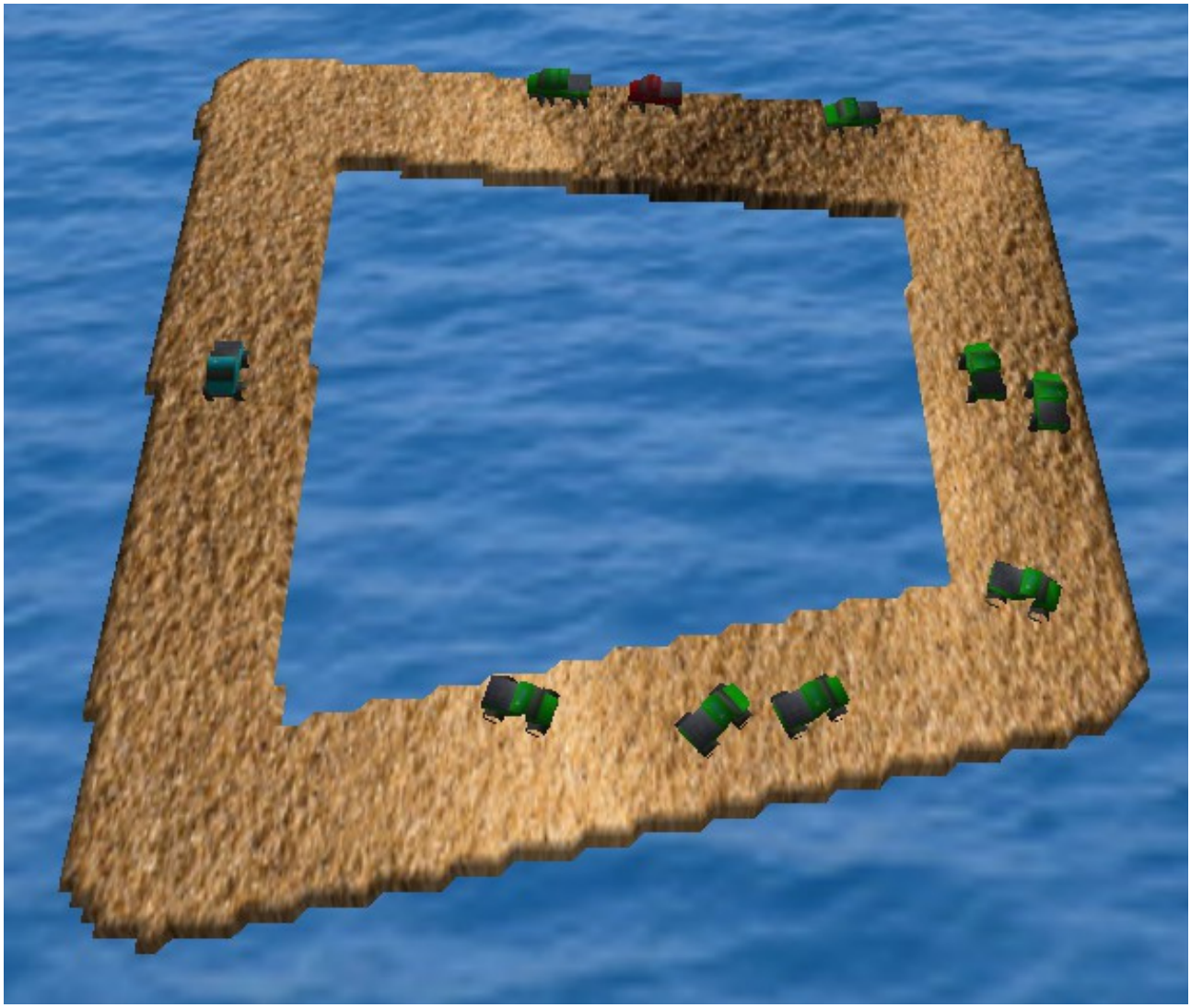

Figura 1. TSI Racing.

Cada aluno poderá desenvolver um robô para controlar uma caminhonete de corrida. Além disso, os alunos também poderão alterar a textura da sua caminhonete.

Os alunos terão cerca de um mês para desenvolver o robô. Os robôs deverão ser enviados pelo site do desafio da semana acadêmica. O site será atualizado uma vez por dia e mostrará a classificação dos alunos.

Na semana acadêmica, será mostrada a corrida com os 8 melhores robôs. O aluno que vencer a corrida receberá um prêmio surpresa!

Regras

Participação:

- O desafio é individual.
- É recomendado que o aluno já tenha terminado o 1º período, pois é necessário conhecimento de programação.
- Os calouros também podem participar, porém eles terão um pouco mais de dificuldade.

Desenvolvimento:

- Os conhecimentos de algoritmos necessários para implementar o robô são: variáveis e estruturas condicionais.
- O robô deverá ser implementado na linguagem Javascript.
- A implementação será somente um objeto em Javascript. Ele deverá ter os atributos “pedal” e “volante”. Ele deverá ter o método “jogar” com 1 argumento.
- O aluno deverá implementar a lógica utilizando somente os dados contidos no objeto do robô e os dados que serão passados por parâmetro para o método “jogar”.
- Não será possível utilizar bibliotecas externas em Javascript. O aluno terá à disposição somente a biblioteca padrão do Javascript.
- É proibido manipular threads no robô.
- É proibido qualquer tipo de acesso externo ao robô. É proibido acessar qualquer variável global.
- É proibido enviar o robô padrão como se fosse o robô que o aluno implementou. O robô padrão já vem com o jogo.
- Se o aluno desejar, ele poderá alterar a textura padrão da caminhonete e enviar pelo site. O formato da textura deverá ser JPG.
- O tamanho máximo do arquivo do robô é de 50 kb. O tamanho máximo do arquivo da textura é de 300 kb.

O Jogo

O jogo foi implementado em Java. Requisitos para o jogo funcionar:

- Windows, Linux ou Mac.
- Java 8 instalado (https://www.java.com/pt_BR/download/).

Como executar:

- Baixe e descompacte o arquivo ZIP do site da semana acadêmica de TSI.
- A pasta “cars” é onde devem estar os arquivos dos carros: javascript (**JS**) e texturas (**JPG**).

Para o jogo carregar a textura, os arquivos JS e JPG deverão ter o mesmo nome. O arquivo de textura é opcional. Caso não haja uma textura, será utilizada a textura padrão.

- O arquivo **tsiracing.jar** é o jogo em Java. Para executar basta utilizar um dos procedimentos a seguir:
  - clicar 2 vezes; ou
  - abrir o terminal, entrar na pasta do jogo e executar o comando **`java -jar tsiracing.jar`**.
- O melhor modo de executar o jogo é pelo terminal, pois é possível passar por parâmetro qual será a pista. O jogo já vem com duas pistas: pista1 e pista2. Se não for passada a pista, será escolhida a pista1. Exemplo de como executar com a pista2: **`java -jar tsiracing.jar pista2`**.
- Também é possível escolher a velocidade do jogo. Basta passar por parâmetro: velocidadeX. Ela pode ser de 1 a 4. Por padrão, a velocidade é 1. Exemplo de como executar com a velocidade 4: **`java -jar tsiracing.jar velocidade4`**.

O jogo já vem com um robô padrão implementado. Este robô possui o mínimo para que o carro consiga terminar a corrida. Ele servirá como base para o aluno desenvolver o próprio robô. Também é desejável que o aluno implemente um robô melhor que o padrão.

A corrida termina quando todos os carros terminarem a corrida ou quando passar 10 minutos de corrida. Se o aluno desejar terminar a corrida antes, é melhor clicar no “x” da janela do que encerrar o processo pelo terminal.

Após o término da corrida, será gerado um arquivo “**resultado.txt**”. Neste arquivo, cada linha é um carro. A ordem dos carros é a ordem de classificação. Só aparecerão no arquivo, os carros que tiverem terminado a corrida. Se o jogo for fechado no botão “x”, também será gerado o arquivo de resultado somente com os carros que terminaram a corrida.

Desenvolvimento do Robô

O aluno poderá escolher seu editor preferido para desenvolver o robô. Sugestões de editores: Vim, Notepad++, gedit, Sublime Text, Eclipse, NetBeans, etc.

O robô deverá ser um único objeto JS, utilizando a sintaxe da Figura 2, que mostra o carro de exemplo que já vem com o jogo.

```
{
    /*
    -------------------------
    -- O ALUNO PODE ALTERAR --

    this.pedal = 'acelerar', 'frear', ''

    this.volante = 'direita', 'esquerda', ''

    ----------------------
    -- O ALUNO DEVE LER --

    info.velocidade = 0 até 200

    info.curvaDistancia = 0 até 99999

    info.curvaDirecao = 'esquerda', 'direita'

    info.centro = -20 até 20, é a distância até o centro da pista,
                negativo significa que está à esquerda do centro.
    */
    pedal: '',
    volante: '',

    jogar: function(info) {
        this.pedal = 'acelerar';

        if (info.centro > 0) this.volante = 'esquerda';
        else if (info.centro < 0) this.volante = 'direita';
        else this.volante = '';
    }
}
```

Figura 2. Exemplo de robô em javascript.

Considerações importantes:

- Se a primeira linha for alterada, pode ser que ocorra um erro de sintaxe. Portanto, não altere a primeira linha.
- Caso exista algum erro de sintaxe no robô, o carro ficará parado e será emitida uma mensagem de erro no terminal.
- Os atributos “pedal” e “volante” são obrigatórios. O aluno poderá alterar o valor inicial dos atributos. Eles deverão conter valores do tipo texto.
- O método “jogar” é obrigatório e ele dever ter um argumento. O aluno deverá alterar o corpo do método, que é a lógica do robô. O argumento “info” é um objeto com os atributos: “velocidade”, “curvaDistancia”, “curvaDirecao” e “centro”.
- O fluxo de execução do jogo é o seguinte: o jogo chama o método “jogar” do robô. Depois que o método termina, o jogo lê os valores dos atributos “pedal” e “volante”.

- O aluno pode criar outros atributos e métodos.
- Os valores válidos dos atributos “pedal” e “volante” estão na Figura 2. Os tipos de valores que estarão no objeto “info” estão na Figura 2.
- O carro não poderá ficar muito longe do centro da pista. Se ele for para longe, então ele baterá na “parede invisível”. Neste caso, a velocidade do carro cai para zero.
- Nunca esqueça do **this** para acessar os atributos do objeto. Nunca esqueça do **info** para acessar os atributos que vêm por parâmetro.