Enunciado do Trabalho Prático: Mars Rover

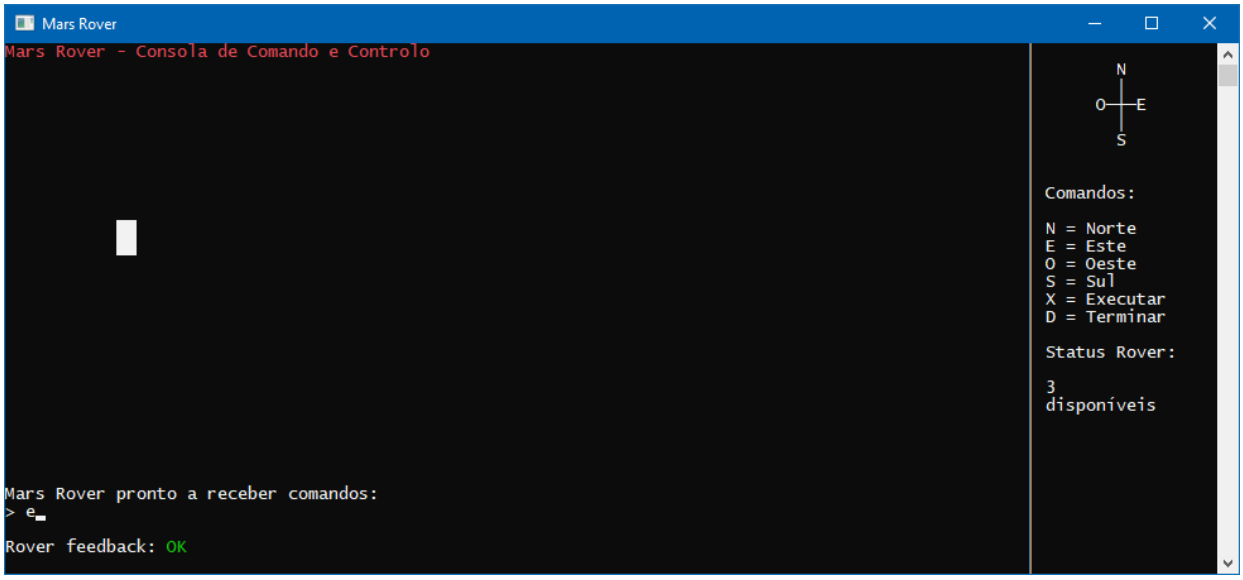
Tema: Desenvolvimento de um aplicativo de consola com recurso à estrutura de dados **Queue**

Problema

Encontra-se na superfície do planeta Marte um veículo de exploração, o **Mars Curiosity Rover**. Este veículo é controlado remotamente, a partir do planeta Terra.
Dada a quantidade de memória limitada que o veículo possui, e a grande distância a que o planeta Marte se situa da Terra, o veículo é controlado da seguinte forma:

- os controladores (no planeta Terra) enviam um conjunto de comandos para o Rover
- o Rover executa os comandos por ordem de chegada
- após a execução dos comandos, o Rover pode receber novos comandos

Com o projeto C# incompleto que o professor disponibiliza, deverás escrever o código que permita ao Rover receber e executar os comandos enviados pelos controladores da missão.



Objetivos

- Colocar em prática os conteúdos abordados na disciplina de Programação e Sistemas de Informação
- Desenvolver o espírito crítico e de seleção de informação
- Desenvolver a capacidade de planear um produto e prever os passos necessários para a sua concretização
- Estimular a autonomia e criatividade dos alunos

Desenvolvimento do Trabalho

- O trabalho é desenvolvido individualmente
- O trabalho consiste na criação de um aplicativo que simula o envio e execução de comandos para um veículo (Rover) que se encontra na superfície de outro planeta (Marte)
- Para tal o professor disponibiliza algum código na forma de um projeto Visual Studio: os alunos deverão completar e, se desejarem, melhorar o código
- Ver a última página para uma descrição do código disponibilizado
- Os comandos deverão ficar armazenados numa estrutura do tipo **Queue** (fila)
- O aplicativo termina quando o utilizador introduzir o comando **d**

- A classificação máxima a obter pelos alunos depende do esforço: quanto mais complexa for a aplicação, maior a probabilidade de obter uma classificação alta (desde que o aplicativo esteja corretamente desenvolvido e de acordo com os critérios de avaliação)
- O professor disponibiliza, na plataforma YouTube, dois vídeos exemplificativos do funcionamento pretendido bem como uma explicação do código a acrescentar
- **Os alunos são encorajados a fazer, com base nestas especificações, aplicativos criativos e com mais funcionalidades (por exemplo: impedir que o rover ultrapasse os limites do ecrã, colocar obstáculos no caminho (obrigando o rover contorná-los), dotar o rover de combustível que vai sendo gasto sempre que se movimenta, etc.)**

Recursos

- Computador / Internet
- Visual Studio
- Materiais disponíveis na plataforma de apoio ao ensino: apresentações dos conteúdos dos módulos, exemplos, fichas de trabalho, etc.
- Vídeos exemplificativos no YouTube:
  - 1ª parte: https://youtu.be/oadSN_04Lx8
  - 2ª parte: https://youtu.be/AWtGchBb_NM
- Projeto C# fornecido pelo professor

Avaliação

- A avaliação do trabalho será feita de acordo com os seguintes critérios:

Critérios	Cotação
Inserção de código (inserir o código em falta nos locais assinalados da aplicação; o código deverá estar corretamente indentado, respeitar as especificações descritas neste documento e utilizar corretamente a linguagem C#)	
• Ficheiro `Display.cs` → classe `Display` → método `DesenharTituloAplicacao()` ○ Mostrar o título da aplicação na consola	10
• Ficheiro `Rover.cs` → classe `Rover` ○ Declarar uma queue chamada `Comandos` do tipo `Comando`	20
• Método `InserirComando()` ○ Inserir um comando na queue	20
• Método `ExecutarComandos()` ○ Retirar um comando da queue	20
• Ficheiro `Program.cs` → método `Main()` ○ Desenhar o rover	10
○ Instrução `switch` ▪ Terminar a aplicação	10
▪ Completar a chamada ao método `InserirComando()`: inserir o comando Norte no parâmetro	14
▪ Completar a chamada ao método `InserirComando()`: inserir o comando Este no parâmetro	14
▪ Completar a chamada ao método `InserirComando()`: inserir o comando Oeste no parâmetro	14
▪ Completar a chamada ao método `InserirComando()`: inserir o comando Sul	14

no parâmetro ▪ Inserir o método apropriado para executar os comandos do rover	14
Criatividade e complexidade • Complexidade do produto final • Os alunos desenvolvem um produto criativo, adquirem e colocam em prática conhecimentos avançados, ainda não abordados em aula	40
Total	200

O não cumprimento dos prazos estabelecidos poderá resultar em penalização na classificação

Anexo
Descrição do código disponibilizado pelo professor

É disponibilizado um projeto com **4** ficheiros de código C#:

`Program.cs`	Na classe `Program`, o método `Main()` é responsável por todo o ciclo de vida da aplicação: inicializa a aplicação, recebe e processa o input do utilizador, termina a aplicação.
`Rover.cs`	Contém a classe `Rover` e todo o código referente ao veículo no planeta Marte. Permite enviar os comandos e receber feedback. É nesta classe que os alunos deverão acrescentar a maior parte do código.
`Display.cs`	Contém a classe estática `Display` (significa que todo o seu código e propriedades são acessíveis diretamente, sem que haja necessidade de criar uma instância da classe). Esta classe é responsável pelo display de dados, nomeadamente: • Título da aplicação (método `DesenharTituloAplicacao()`) • Pontos cardeais (bússola) (método `DesenharPontosCardeais()`) • Lista de comandos (método `DesenharComandos()`) • Indicação da quantidade de comandos disponíveis no rover (método `DesenharStatusRover()`)
`LinhaComandos.cs`	Contém a classe estática `LinhaComandos`, responsável por mostrar a linha de comandos e por mostrar o feedback do rover.

Cada classe possui responsabilidades bem definidas: