

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
DEPARTAMENTO DE INFORMÁTICA APLICADA

INF01108 - Arquitetura E Organização De Computadores I - TURMA B
PROFESSOR: Marcelo Johann
SEMESTRE: 2024-1

Trabalho T3 - Intel

O Trabalho consiste em implementar o jogo Vale da Morte, de forma semelhante ao pequeno exemplo cujo código fonte em C e executável encontram-se ao final da página. O jogo consiste em controlar uma nave que desce em alta velocidade pelo escuro Vale da Morte, sem que ela colida com os paredões que o cercam. O piloto apenas pode ver alguns metros à sua frente, e por isso guia-se pelos paredões que se aproximam e afastam para evitar o seu fim, que é inevitável...

Para implementar o jogo você deve imprimir sucessivamente as linhas que compõe o cenário, e deixar que a rolagem automática da tela se encarregue de criar o cenário e o movimento necessários. O programa deve usar um número aleatório para definir a variação da posição do vale e testar se o usuário pressiona alguma tecla, atualizando a posição da nave. Devem ser testados também os limites laterais para a variação do vale, e a colisão da nave com as paredes.

À medida que o piloto avança, o vale torna-se mais estreito e sinuoso. Isso deve ser implementado pela redução gradual da largura do espaço entre as paredes, e pelo aumento da variação da posição das paredes, e é por essa razão que a colisão é certa: o piloto apenas tentará prolongar mais o seu angustiante tempo de vida, e são considerados notáveis aqueles que atingem os ambientes mais profundos do labirinto, contados em número de linhas percorridas.

Nas primeiras simulações, você identificará o final do episódio quando a nave tocar na primeira parede, o que acarretará grande explosão! Mas na verdade a nave é resistente o suficiente para atravessar a cobertura da floresta nas margens do vale e resistir a algumas colisões, embora sofrendo graves avarias que a condenam em pouco tempo. Quanto mais adiante o piloto consegue conduzir, mais forte fica o seu revestimento devido à resistência do aço de seu casco que aumenta com o calor por se tornar menos quebradiço, e por isso ela pode agüentar algumas avarias mais severas.

Você simulará esse comportamento utilizando um outro contador de linhas adicional ao contador de profundidade, mas que será subtraído de um valor razoável a cada colisão com a parede, e controlando o fim do jogo nessa versão mais exata quando esse contador de resistência chegar a zero (ou assumir um valor negativo).

Mas para ajudar um pouco os destemidos pilotos, você poderá implementar ao final um dispositivo digital de recuperação e filtro de imagem que permite ao piloto enxergar 4 linhas à sua frente. Para isso, você posicionará a nave na quarta linha debaixo para cima da janela, e deverá memorizar as quatro linhas para testar a colisão da nave nesta posição, embora a rolagem da tela continue sendo automática.

A função aleatória pode ser feita com $(x = x*9+9) \% 251 + 250) \% 3$ para três alternativas: fica no lugar, desloca para a direita ou desloca para a esquerda. Essa função foi fornecida por Fábio Cecin, e os créditos da idéia do jogo são atribuídos a Eduardo Hennemann Pitrez.

A avaliação será feita segundo os seguintes critérios e pontuação:

Parte básica - Laço de impressão: 1 ponto

Movimento das teclas 1 ponto

Teste de limites laterais 1 ponto

variação aleatória 1 ponto

Fim de Jogo - Teste de colisão 1 ponto

Informar a profundidade 1 ponto

Dificuldade - Largura vai diminuindo 1 ponto

Variação vai aumentando 1 ponto

Sofisticações - Avarias e fim por resistência zero 1 ponto

Look ahead de 4 linhas 1 ponto

Cenário e extras 1 ponto

Aconselha-se implementar uma versão sem as sofisticações, e depois tentar implementá-las em nova versão, garantindo que os pontos básicos, de fim de jogo e dificuldade sejam obtidos sem problemas causados por elas.

Código de Referência Básico:

Fonte em C para DOS - [VALLEY.C](#)

Executável para DOS - [VALLEY.EXE](#)

// END OF PAGE