

LCOM - Project Specification

Neste projeto de LCOM ambicionamos fazer um jogo do estilo de Asteroids, mas com algumas diferenças.

Nome: -Asteroids Survival

Descrição:

O jogador controla uma nave, fazendo-a mover com o teclado e fazendo-a girar e disparar com o rato. Será um jogo baseado em sobrevivência por rondas. Isto é, a cada ronda que passa tornar-se-á mais difícil de sobreviver, sendo que o jogador perde o jogo quando colide com um asteroide. Estes, inicialmente, movem-se no ecrã mais lentamente e em menor quantidade, sendo que a cada ronda que passa a velocidade dos asteroides aumenta, tal como a sua quantidade. A ronda avança quando o jogador destrói um certo número de asteroides. O jogador tem as possibilidades de se desviar dos asteroides e ainda de disparar contra estes, destruindo-os.

Tencionamos usar os seguintes dispositivos: -Timer 0; -Teclado; -Placa gráfica; -Rato;

Apresentamos as nossas ideias dos papeis e funcionalidades dos dispositivos:

Timer 0:

-Papel: Fazer animações;

-Funcionalidade: Contar intervalos de tempo para mover objetos no ecrã;

Teclado:

-Papel: Controlar a nave do utilizador e navegar as opções nos menus;

-Funcionalidade: Registrar os inputs para mover a nave: para cima (Tecla W pressionada); para baixo (Tecla D pressionada); para o lado esquerdo (Tecla A pressionada) e para o lado direito (Tecla D pressionada). E as mesmas teclas para navegar as opções nos menus, acrescentando a tecla ESC para pausar o jogo e mostrar o menu e ainda a tecla ENTER para confirmar uma opção no menu.

Placa gráfica:

-Papel: Fazer a interface entre o jogo e o utilizador

-Funcionalidade: Mostrar a nave que o utilizador controla e movendo-la ao longo do ecrã de acordo com o input. Mostrar as balas a serem disparadas da nave, os asteroides e respetivos movimentos. Também será necessário para mostrar o menu e outras informações como a ronda atual, mudança de ronda e o fim do jogo.

Rato:

-Papel: Fazer girar a nave e disparar;

-Funcionalidade: O movimento do rato será usado para girar a nave, sendo que esta aponta sempre na direção do rato. Quando o botão esquerdo é pressionado a animação de disparo da nave deve ocorrer.

Desenvolvimento do jogo:

Modulos:

Game

-Struct com a informacao do jogo: os seus objetos, como a nave e asteroides; estado atual do jogo; etc.

-State machine reponsável pela identificação do estado atual do jogo. Chama diferentes funções consoante o estado atual, permitindo interpretações diferentes do input em alturas diferentes do jogo. Contém os estados correspondentes ao menu e ao destaque da opção selecionada.

-Um estado, `RUNNING_GAME`, será o responsável pela identificação dos movimentos da nave e do rato (rotações da nave) e pela ocorrencia de disparos, chamando as funções responsáveis do modulo Game que alteram as propriedades da nave e do rato, nomeadamente as suas posições e movimentos.

Visuals

- Função updateFrame que, consoante o estado atual do jogo, mostra no ecrã o necessário. Por exemplo, para o estado MENU deve mostrar o menu, mas para o estado RUNNING_GAME deve atualizar a posição e a rotação da nave e mover os asteroides.
- Mostrar os asteroides e fazê-los mover ao longo do ecrã;
- Funções para mover e girar a nave no ecrã consoante o input.
- Processar o disparo da nave, mostrando a sua trajetória retilínea, que, quando se cruza com um asteroide, deve desaparecer, juntamente com o asteroide que foi atingido.
- Funções para mostrar no ecrã a ronda atual, o número de asteroides destruídos e uma mensagem de game over, quando o jogador perder.

GamePlay

- Funções para processar as rondas e respetivos fatores de dificuldade como a velocidade dos asteroides e a sua quantidade no ecrã.
- Funções para calcular as posições relativas da nave e dos asteroides e analisar se ocorreu uma colisão e, portanto, se o jogador perdeu;
- Contar o número de asteroides destruídos, fazendo a ronda avançar quando um certo número é atingido;

Devices Manager

- Trata de guardar convenientemente o input do rato e do teclado e passar esta informação para variáveis que possam ser mais facilmente interpretadas por outros módulos.

Planeamento

Primeira semana:

-Começar a fazer a estrutura do trabalho, ou seja, começar pelo modulo Game, fazendo a state machine e identificar que tipo de funções têm de ser chamadas para cada estado, isto é, o que cada uma necessita fazer.

-Começar a modulo Visuals, criando as imagens que vão ser utilizadas no projeto. Fazer funções para carregar e desenhar essas imagens no ecrã;

Segunda semana:

-Completar modulo Visuals. Começar a lidar com o input do teclado, fazendo o jogador mover-se no ecrã enquanto o jogo está a decorrer e navegar as opções no menu quando o jogo se encontra neste estado;

Terceira semana:

-Lidar com o input do rato. Criar um cursor para navegar o menu e para fazer a nave rodar enquanto o jogo está no estado `RUNNING_GAME`. Começar a fazer o modulo Game Logic, ou seja, deteção de colisões através da análise das distâncias entre a nave os diferentes asteroides;