1 Estrutura

O projeto foi estruturado de acordo com o autómato finito representado na figura 1. Este autómato é atualizado cada vez que o interrupt do temporizador é ativado. A cada estado corresponde uma função, sendo o endereço da função do estado atual guardado na variável global 'STATE'. Na rotina do interrupt do temporizador é incrementada a variável global 'TICKS', e no loop principal, sempre que essa variável deixa de ser 0, é decrementada e o autómato finito é atualizado ao fazer um jump para o endereço guardado na variável 'STATE'.

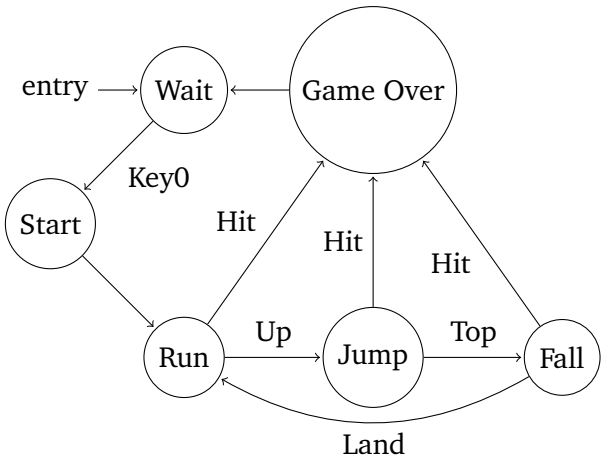

Fig. 1: Autómato finito do jogo

1.1 Autómato Finito

A mudança de estado é feita dentro das funções de cada estado. O estado 'Wait' apenas verifica se o botão 'Key0' foi premido, e se sim, altera o estado para 'Start'. O estado 'Start' inicializa o jogo e muda de estado imediatamente para 'Run'.

O estado 'Run' corresponde ao movimento normal do dinossauro, no qual apenas os cactos são movidos para a esquerda. Neste estado é também verificado se o botão 'Up' foi premido, e se sim, o estado é alterado para 'Jump'. O estado 'Jump' corresponde ao movimento ascendente do dinossauro que acontece depois de ser inicializado o salto, sendo aqui movido o jogador uma linha para cima e os cactos movidos também para a esquerda. Quando o jogador atinge a altura máxima, o estado é alterado para 'Fall'. O estado 'Fall' corresponde ao movimento descendente do jogador depois de atingir a altura máxima do salto. O jogador é movido uma linha para baixo, até chegar novamente ao nível do chão. Quando isto acontece, o estado é alterado para 'Run' novamente.

Para além disso, tanto no estado 'Run', como no 'Jump' e 'Fall' é verificado se houve alguma colisão do jogador com cactos, e se sim, o estado é alterado para 'Game Over'. O estado 'Game Over' desenha a mensagem de 'G A M E O V E R' no meio do ecrã e altera imediatamente para o estado 'Wait'.

1.2 Input

O input dos botões 'Key0' e 'Up' é feito através de flags globais ('KEY0_HANDLED' e 'UP_HANDLED'), que são postas a 0 pelos interrupts respetivos e verificadas nos locais necessários. Sempre que é lido o valor da flag e esta tem o valor 0, é lhe dado o valor 1. Além disso, a mask de interrupts é alterada de forma a excluir os interrupts não necessários em certas mudanças de estado, de forma a impedir interrupções desnecessárias. Um exemplo disto é a mudança de 'Wait' para 'Start', na qual o interrupt do botão 'Key0' é desativado, e mais tarde reativado na mudança de 'Game Over' para 'Wait'.

2 Implementação

Nesta secção segue-se uma descrição breve do propósito de cada função, para explicações mais detalhadas consultar os comentários no código.

2.1 Função de entrada, ciclo do jogo e funções dos estados

Na função 'entry' são inicializadas várias variáveis e interruptores pela primeira vez e é onde está definido o ciclo do jogo.

O propósito e funcionamento das funções de estado já foram explicados na secção 1.1.

2.2 Funções da lógica do jogo

inc_score	Incrementa a pontuação e atualiza o 7 segment display.
reset_score	Põe o score a 0 e atualiza o 7 segment display.
p_in_cactus	Verifica se o jogador está dentro de um cacto.
update_map	Antigo atualizajogo. (Ver 3.1)
gen_cactus	Antigo geracacto. (Ver 3.1)
update_game	Implementa o comportamento comum aos estados 'Run', 'Jump' e 'Fall'.
read_key0	Verifica se o interrupt KEY0 foi chamado.
read_up	Verifica se o interrupt UP foi chamado.

2.3 Funções dos gráficos do jogo

term_pos	Posiciona o cursor do terminal na posição correspondente.
term_color	Muda a cor do próximo caractere escrito no terminal.
term_write	Escreve um caractere na posição atual do terminal, e avança a posição de escrita.
term_clear	Apaga o terminal (escreve um espaço em todas as posições)
draw_many	Escreve um carácter N vezes.
draw_cactus	Desenha um cato no terminal.
draw_map	Desenha o mapa no terminal.
draw_floor	Desenha o chão.
clear_player	Apaga o jogador.
draw_player	Desenha o jogador.
draw_text	Desenha o texto no terminal.
draw_textbox	Desenha uma caixa de texto no centro de ecrã.
draw_game	Desenha os elementos comuns aos estados 'Run', 'Jump' e 'Fall'.

2.4 Rotinas de interrupção e rotinas de interrupção auxiliares

O propósito e funcionamento destas funções já foram explicados na secção 1.2.

3 Dificuldades Encontradas

3.1 Convenção de Nomes

Como optamos por 'snake_case', nomes em inglês ao longo do programa e utilizámos o nome 'update_game' para outra função, surgiu a necessidade de renomear 'atualizajogo' para 'update_map' e 'geracacto' para 'gen_cactus'.

3.2 Estabilidade do Temporizador

Depois de obtermos a primeira versão testável do jogo, descobrimos que havia uma pequena chance de ao carregar no botão 'Up' ou 'Key0', ativando os interruptores, o temporizador não voltar a ser reativado, o que fazia com que o jogo congelasse.

Não conseguimos encontrar uma solução definitiva para este problema, mas conseguimos minimizar o problema ao desativar os interrupts 'Key0' e 'Up' quando não fossem necessários. Isto é, desativar o interrupt 'Key0' depois de começar o jogo e ativar novamente quando acaba, e desativar o interruptor 'Up' quando o jogador salta e ativar novamente quando o jogador aterra.