Fallin': Um Jogo de Queda

Maria Eduarda Contu

[1]Programação de Baixo Nível
Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)

1. Introdução

O presente estudo tem como objetivo explorar e resolver a proposta de trabalho avaliativo final da matéria de Programação de Baixo Nível. De tal modo, o objetivo do trabalho é desenvolver um jogo eletrônico que respeite as seguintes restrições:

1. Restrição 1: não poderá ser desenvolvido um jogo de tabuleiro;
2. Restrição 2: o seguinte conjunto de recursos deve ser utilizado:
    - Uso de botões, keypad ou joystick como dispositivo de entrada;
    - Uso do LCD como dispositivo de saída (usar rotinas de desenho de texto, linhas e círculos);
    - Uso de pelo menos um timer;
    - Uso de pelos menos uma interrupção (interna ou externa).

Além disso, existem **funcionalidades obrigatórias**, que são:

1. O jogo deve ter uma "tela inicial", que aguarda o pressionamento de um botão/etc para iniciar.
2. Essa tela serve para, por exemplo, gerar um valor aleatório para sorteios durante o jogo; O jogo pode ter fim, dando vitória ao jogador ou ser do tipo que vai dificultando ao passar do tempo;
3. O jogador tem que ser capaz de PERDER o jogo por algum critério (tempo, movimento incorreto, etc).

Então, assim foi decidido implementar um jogo de queda de uma bolinha, esta que deve desviar dos obstaculos.

2. Regras do Jogo

1. Você é a bolinha e a bolinha é você. Você só tem uma vida, cuide bem dela.
2. Utilize apenas as teclas A e D para movimentar e, quando necessário, utilize a tecla "espaço"para confirmar ou selecionar uma opção.
3. Caso a bolinha bata em algum obstáculo, o jogo acabou, armazenando o tempo recorde da sessão.
4. Não existe uma maneira de vencer o jogo.
5. A bolinha vai caindo na tela (ou os obstáculos vão subindo, depende de como você enxerga a situação) e você deve desviar dos obstáculos o maior tempo que conseguir, lembre que seu recorde ficará salvo na sessão!

3. Uso de Componentes

3.1. Botões

Na tarefa, para realizar os movimentos da bolinha e a escolha de opções nas telas de inicio e fim, foram utilizados quatro botões para que o funcionamento fosse executado perfeitamente. Eles estão presentes na figura 1. No código fonte existem diversos chamamentos relacionados a ativação desses módulos, como, por exemplo, os referentes ao movimento da bolinha, que está em destaque no código abaixo.

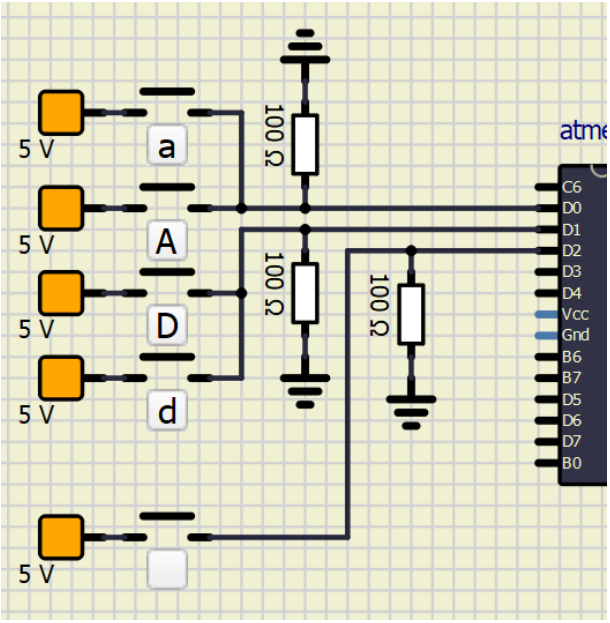

Figura 1. Botões de A/a,D/d e espaço

```
void caminha()      {
    if (PIND & (1<<EsqA))
    {
        if (xBola>=9)
        {
            xBola=xBola-5;
        }

        while (PIND & (1<<EsqA)) //debounce
        {
            _delay_ms(1);
        }
        _delay_ms(1);


    }
    if (PIND & (1<<DirD))
    {
        if (xBola<=72)
        {
            xBola=xBola+5;
        }
        while (PIND & (1<<DirD)) //debounce
        {
            _delay_ms(1);
        }
```

```
        _delay_ms(1);

    }
}
```

3.2. LCD

Na tela LCD criada, utilizamos um drive fornecido pelo professor para aplicar com menos dificuldades as imagens a serem projetadas. Com o código mostrado abaixo, referente a tela de início do programa, utilizamos retângulos e linhas para desenhar o botão de play, que ao ser selecionado da inicio ao jogo. A representação gráfica do código está na figura 2.

```
void menuTela()      {
nokia_lcd_clear();
nokia_lcd_set_cursor(5,0);
nokia_lcd_write_string("Fallin'",2);      //titulo

//botao de iniciar
nokia_lcd_drawrect(30,17,50,37);      //quadrado
//desenhar triangulo de play
nokia_lcd_drawline(37,20,37,35);
nokia_lcd_drawline(37,35,45,27);
nokia_lcd_drawline(37,21,45,28);

nokia_lcd_set_cursor(0,40);
nokia_lcd_write_string("Regras",1);
nokia_lcd_set_cursor(48,40);
nokia_lcd_write_string("Fechar", 1);

//destaca opcao selecionada
if (opcao==1)//regras
{
    nokia_lcd_set_cursor(10,33);
    nokia_lcd_write_string("*",1);
}
if (opcao==2)//iniciar
{
    nokia_lcd_drawline(37,22,44,27);
    nokia_lcd_drawline(37,23,44,27);
    nokia_lcd_drawline(37,24,44,27);
    nokia_lcd_drawline(37,25,44,27);
    nokia_lcd_drawline(37,26,44,27);
    nokia_lcd_drawline(37,27,44,27);
    nokia_lcd_drawline(37,28,44,27);
    nokia_lcd_drawline(37,29,44,27);
    nokia_lcd_drawline(37,30,44,27);
    nokia_lcd_drawline(37,31,44,27);
```

```
            nokia_lcd_drawline(37,32,44,27);
            nokia_lcd_drawline(37,33,44,27);
            nokia_lcd_drawline(37,34,44,27);

        }
        if (opcao==3)//fechar
        {
            nokia_lcd_set_cursor(58,33);
            nokia_lcd_write_string("*",1);
        }

        nokia_lcd_render();
}
```

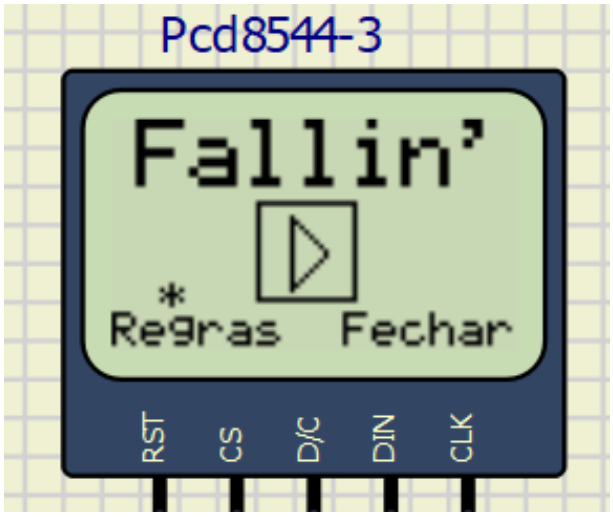

Figura 2. Menu

Além disso, utilizamos, durante o jogo, o código para criação de uma bolinha, que representa o jogador. O código com a criação dessa está abaixo, e sua representação gráfica na figura 3.

```
        ...

        //desenha circulo
        nokia_lcd_drawcircle(xBola,yBola,rBola);

        //imprime espinho
        nokia_lcd_drawcircle(xEsp,yEsp,rEsp);
        ...
```

3.3. Timer

A principio, utilizariamos dois tipos de timers, um que faria a contagem regressiva para o inicio do jogo, e outro que incrementasse todo segundo para armazenar, ao fim, o tempo

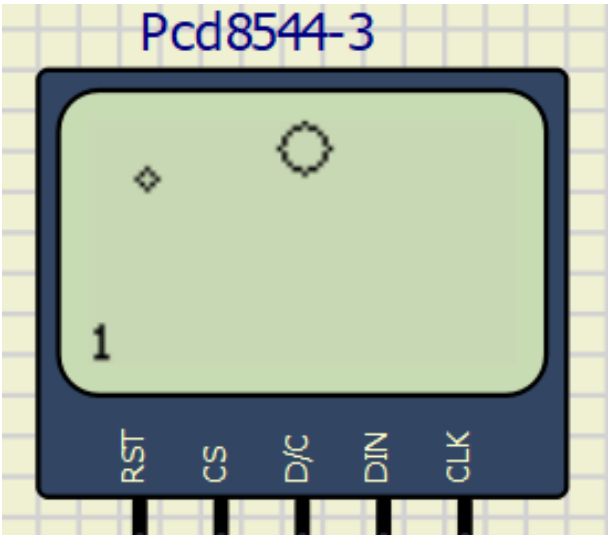

Figura 3. Bola e espinho

recorde da sessão. Porém tive diversas dificuldades para realizar esse segundo modelo de timer, então o montei utilizando apenas expressões de delay.

O timer utilizado, referente a contagem regressiva, possui o código abaixo para sua execução.

```
void setTimer() {
    TCCR1B |= (1 << WGM12);       // Modo CTC
    // Valor para contar at  10 segundos (prescaler de 256)
    OCR1A = 15624;
    // Habilita interrup  o do Timer1 comparando com OCR1A
    TIMSK1 |= (1 << OCIE1A);
    TCCR1B |= (1 << CS12) | (1 << CS10);
// Prescaler de 1024
}

ISR(TIMER1_COMPA_vect) {
    if (timer > 0) {
        timer--; // Decrementa o tempo
    }
}

//funcao que utiliza o timer
void contador()      {
    setTimer();
    sei();

    timer=3;
    nokia_lcd_clear();
    while (timer>0)
    {
```

```
            nokia_lcd_clear();
            sprintf(strTimer,"%d",timer);
            nokia_lcd_set_cursor(37,15);
            nokia_lcd_write_string(strTimer,2);
            nokia_lcd_render();
        }
        cli();
        nokia_lcd_render();
}
```

3.4. Interrupção

A interrupção utilizada no projeto é uma interrupção interna, também referente ao timer que controla a contagem regressiva. No código mostrado acima, podemos perceber sua utilização.

4. Solução e Dificuldades

De tal modo, foi possível concluir o jogo, porém não perfeitamente. Tive grande dificuldade para tentar organizar o segundo timer, que foi citado acima, mas não obtive sucesso em qualquer formato testado. Além disso, o sistema que checa a colisão entre a bolinha e o espinha possui um certo atraso referente as linhas mais baixas do círculo, demorando poucos milesimos para encaminhar a tela de game over (figura 4).

Figura 4. Tela de Fim

Mesmo com dificuldades para desenvolver o programa, com pouca organização foi possível definir como e a ordem em que cada etapa deveria ser feita. De tal modo, creio que houve um bom aproveitamento do semestre, de modo que foi possível resolver o trabalho proposto, claro que com algumas dificuldades, porém de modo que o programa final é funcional e possui uma base clara em seu código.