**UNIVERSIDADE REGIONAL INTEGRADA DO ALTO URUGUAI E DAS MISSÕES**

**PRÓ-REITORIA DE ENSINO, PESQUISA E PÓS-GRADUAÇÃO**

**CÂMPUS DE ERECHIM**

**DEPARTAMENTO DA**

**CIÊNCIA DA COMPUTAÇÃO**

**Axcel Lucas De Mello**

**Gabriel Belitz Baldez**

**CONHECIMENTOS BÁSICOS DE ARDUNIO  
“SPACESHIP VS ALIEN”**

**ARDUINO + TELA DE LCD**

**ERECHIM - RS**

**2021**

**Sumário**

**1. RESUMO ...................................................................................................................3**

**2. INTRODUÇÃO………………..................................................................................3**

**3. MATERIAL………………........................................................................................4**

**4. DESENVOLVIMENTO.............................................................................................4**

**5. RESULTADOS E DISCUÇÃO.................................................................................8**

**6. CONCLUSÃO............................................................................................................8**

**Resumo**

Tendo em vista que optamos por trabalhar com a reação em tempo real do usuário, decidimos usar um conceito simples de jogo no estilo Arcade, relembrando e fazendo referência a alguns nomes conhecidos do passado.

Galaga – Projetado pelo estúdio Namco

Space Invaders – Activision

Asteroids – Atari

T-rex game – Google

A proposta é fazer com que o usuário desvie dos aliens que aparecem na tela para continuar vivo no jogo, podendo mudar de dificuldade caso queira.

**Introdução**

As placas Arduino são conhecidas por sua facilidade de aplicação em projetos iniciais com foco educacional, desta forma a mesma foi objeto central utilizado para construção do trabalho da disciplina de Projeto Integrador.

Percebemos que em diferentes projetos é muito comum a utilização de ferramentas que forneçam interação com usuário desde telas, botões ou outros tipos de estímulos que devolvem ao usuário a sensação de interação com o objeto construído.

Desta forma decidimos construir um projeto que pudesse ter algum grau de interação com usuário, em seu primeiro esboço decidiu-se construir alguns botões e utilizar um suporte de tela de led para interação por meio de algum jogo.

Esta primeira versão foi abandonada devido à necessidade de construção de uma arquitetura que promovesse a comunicação entre o controlador da placa de led e a placa arduino, julgamos um nível de complexidade superior ao que conseguiríamos construir no período desta disciplina.

No segundo esboço definimos a utilização de botões juntamente com uma tela LCD, considerando o fato de que existem bibliotecas consolidadas para controlar a tela LCD diretamente da placa arduino.

Definimos a partir deste esboço dois objetivos centrais para o projeto, sendo:

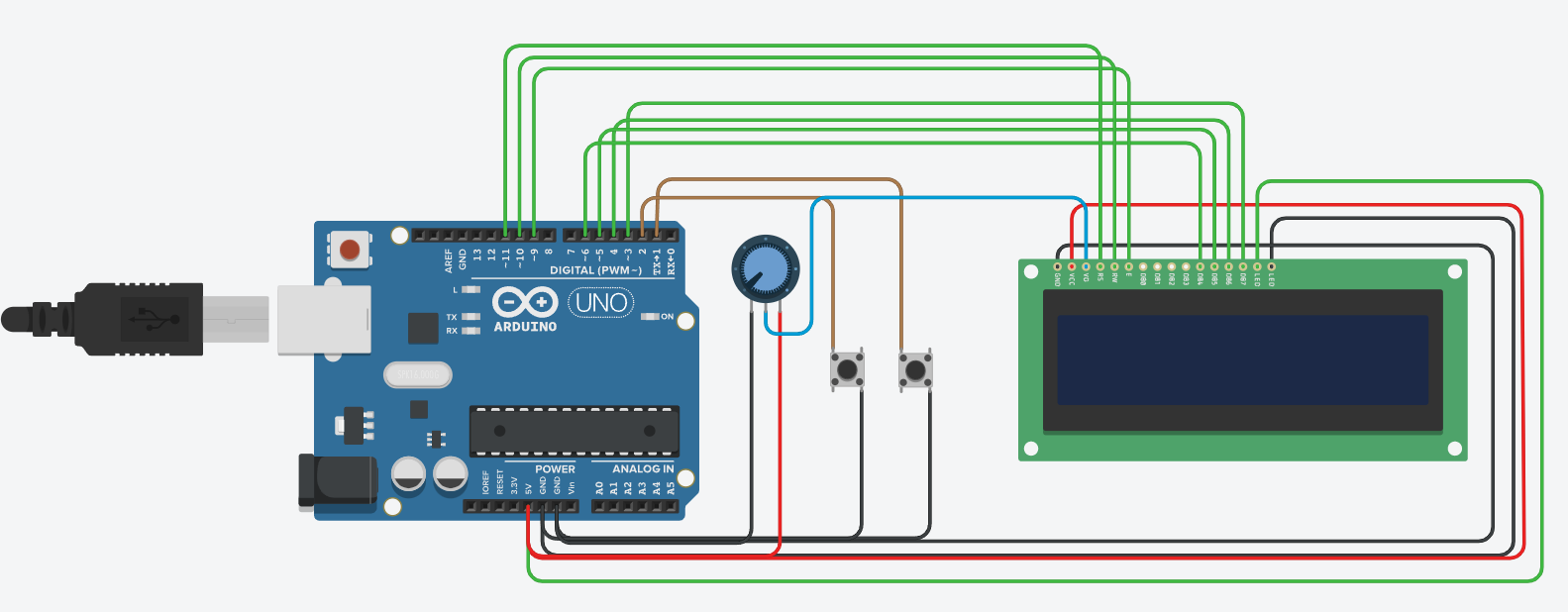
- Construir um jogo eletrônico simples;

- Construir um circuito que permita interação do usuário com o jogo.

**Material**

O material utilizado para construção do projeto foi um arduino Uno R3 conectado a tela LCD 16x2 com dois botões para interação e um potenciômetro para regulação da luminosidade da tela LCD.

Para garantir a testagem dos componentes e avaliação do funcionamento, também foi utilizado uma protoboard para intermediário da conexão entre os componentes.

A montagem do circuito e os componentes utilizados seguem conforme figura e tabela de apoio.****

| **Componente** | **Quantidade** |
| --- | --- |
| Arduino Uno R3 | 1 |
| RGB backlight positive LCD 16x2 | 1 |
| Botão | 2 |
| Kit Jumper | 1 |
| Protoboard | 1 |
| USB-A to B | 1 |
| Potenciômetro 10kΩ | 1 |

**Desenvolvimento**

O código construído tem o apoio da biblioteca LiquidCrystal, encontrada no próprio material de apoio do arduino, esta biblioteca tem o objetivo de configurar as portas do arduino para comunicação com a tela LCD, preparando quais portas seguem para quais objetivos como transferência de dados ou configuração e energização da tela.

O código fonte do projeto ficou da seguinte forma:

///||||||||||||GROUND0||||||||||||||||||||||||||||||||||||||||||

#include <LiquidCrystal.h>

byte plane\_body [] = {

B10110,

B10011,

B11111,

B11111,

B11111,

B11111,

B10011,

B10110};

byte plane\_head [] = {

B11111,

B01100,

B11110,

B11111,

B11111,

B11110,

B01100,

B11111};

byte enemy [] = {

B11000,

B01111,

B11010,

B11111,

B11111,

B11010,

B01111,

B11000};

///|||||||||||||DECLARACAO||||||||||||||||||||||||||||||||||||||

#define down\_button 10

#define up\_button 9

#define pin\_contraste 12

#define pin\_rw 10

#define vazio ' '

int tempo = 400;

///LCD(regselect, enable, data4, data5, data6, data7)

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int posicao = 1;

bool colisao = false;

int dificuldade = 0;

enum estados {

tela\_inicial,

jogando,

gameover

};

int estado = tela\_inicial;

int verifica\_estado(){

if (colisao){

return estado = gameover;

}

}

///|||||||||||||SETUP|||||||||||||||||||||||||||||||||||||||||||

void setup(){

config\_portas();

}

///|||||||||||||LOOP||||||||||||||||||||||||||||||||||||||||||||

void loop(){

switch (estado){

case tela\_inicial:

Serial.println("Tela Inicial");

telainicial();

break;

case jogando:

Serial.println("Jogando");

lcd.clear();

jogo();

break;

case gameover:

Serial.println("Game Over");

lcd.setCursor(2,1);

game\_over();

break;

}

}

///|||||||||||||||FUNCOES|||||||||||||||||||||||||||||||||||||||

int game\_over(){

lcd.clear();

lcd.setCursor(4,1);

lcd.write("YOU LOSE!");

for (int x = 1; x <= 16; x++){

lcd.setCursor(x,0);

lcd.write(byte(2));

delay(150);

update\_pixel(x,0);

}

for (int x = 16; x >= 0; x--){

lcd.setCursor(x,0);

lcd.write(byte(2));

delay(150);

}

while (estado != tela\_inicial){

if((digitalRead(up\_button) == LOW) or (digitalRead(down\_button) == LOW)){

return estado = tela\_inicial;

}

}

}

int telainicial(){

lcd.clear();

delay(500);

colisao = false;

dificuldade = 0;

lcd.setCursor(2,1);

lcd.write("PRESS START!");

int j = 1;

while (estado != jogando){

lcd.setCursor(j,0);

lcd.write(byte(1));

lcd.setCursor(j+1,0);

lcd.write(byte(0));

lcd.setCursor(j-1,0);

lcd.write(byte(vazio));

j++;

if (j == 17){

j = 1;

}

if(digitalRead(up\_button) == LOW){

estado = jogando;

}

if(digitalRead(down\_button) == LOW){

estado = jogando;

}

delay(200);

}

delay(500);

lcd.clear();

lcd.setCursor(3,0);

lcd.write("DIFFICULTY");

lcd.setCursor(0,1);

lcd.write("<- Easy");

lcd.setCursor(9,1);

lcd.write("Hard ->");

while (dificuldade == 0){

if(digitalRead(up\_button) == LOW){

dificuldade = 400;

}

if(digitalRead(down\_button) == LOW){

dificuldade = 200;

}

}

}

void update\_pixel(int x, int y){ /// Após cada desenho

lcd.setCursor(x,y); /// esta função apaga

lcd.write(vazio); /// o espaço utilizado

}

void draw\_ship(bool posicao\_ship){

lcd.setCursor(0,posicao\_ship);

lcd.write(byte(1));

lcd.setCursor(1,posicao\_ship);

lcd.write(byte(0));

update\_pixel(0,not posicao\_ship);

update\_pixel(1,not posicao\_ship);

}

void jogo(){

while(colisao == false){

int y = random(0,2);

for (x = 16; x >= 1; x--){

lcd.setCursor(x,y);

lcd.write(byte(2));

delay(dificuldade);

update\_pixel(x,y);

if (x == 2 and y == posicao){

colisao = true;

estado = gameover;

}

draw\_ship(check\_buttons());

}

}

}

int check\_buttons(){

if(digitalRead(up\_button) == LOW){

posicao = 1;

}

if(digitalRead(down\_button) == LOW){

posicao = 0;

}

return posicao;

}

void config\_portas(){

Serial.begin(9000);

lcd.begin(16,2);

lcd.clear();

pinMode(pin\_contraste, OUTPUT);

digitalWrite(pin\_contraste, LOW);

digitalWrite(up\_button, INPUT\_PULLUP);

digitalWrite(down\_button, INPUT\_PULLUP);

lcd.createChar(0, plane\_head);

lcd.createChar(1, plane\_body);

lcd.createChar(2, enemy);

}

Para a construção dos caracteres que compõem a nave e os inimigos em tela é necessário realizar uma etapa de declaração de um mapa de bytes, onde utiliza-se a construção de 1 e 0 para sinalizar quais devem ficar ligados ou desligados compondo a imagem em uma das posições da tela LCD.

Após declarados assim como as portas do arduino, também declaramos todos os caracteres criados para serem utilizados.

A forma mais simples encontrada para estabelecermos os estados de jogo foi por meio de um Switch-Case, alternando entre a tela inicial, jogo e gameover. No início do loop principal do arduino declaramos algumas variáveis que serão manipuladas ao longo do loop como a boolean de colisão, a posição inicial da nave e a dificuldade.

Na tela inicial foi construído um loop que aguarda a resposta do jogador para inicialização, juntamente com uma animação breve da nave circulando, após qualquer pressão de um dos botões o loop seguinte requisita uma das opções de dificuldade disponíveis, sendo fácil ou difícil. Após esta seleção o estado do switch-case é testado e entra na seção de jogo.

O jogo principal consiste na criação do caractere que representa um inimigo e o loop de movimento dele entre os quadrantes da tela até o último quadrante próximo a nave, é utilizado um valor randômico para seleção de qual linha será posicionado este inimigo, e no loop que o posiciona mais à frente é realizada a verificação de pressão dos botões que alteram o posicionamento da nave.

O objetivo do jogador é evitar a colisão da nave com um dos inimigos em movimento, caso o último desenho do inimigo sobreponha a posição atual da nave a variável booleana colisão é alterada para True saindo do loop principal do jogo e testando o estado novamente direcionando o código para o estado de gameover.

No estado de game over é exibida uma mensagem de encerramento para o jogador e uma breve animação do inimigo sendo desenhado na tela, nesta mesma tela de gameover é testado o retorno de pressão sobre um dos botões, e caso os mesmos sejam pressionados as variáveis globais são reinicializadas e o estado do jogo retorna para a tela inicial, permitindo o recomeço de uma nova sessão de jogo.

**Resultados e Discussão**

No começo do projeto foi implementado no código a função de brilho da tela, utilizando uma saída digital, como usamos o tinkercad.com para executar os primeiros testes o projeto rodava normalmente.

No momento que iniciamos a construção do projeto físico, não recebíamos sinal em tela, assim foi necessário fazer uma mudança na aplicação e mudar a função de brilho para uma porta analógica com um potenciômetro, assim conseguimos regular manualmente o brilho.

As montagens eletrônicas foram a maior dificuldade no decorrer do projeto, tendo em vista as alterações entre o ambiente de teste e a construção do circuito, felizmente algumas correções na utilização da porta de setup de brilho da tela e a conferência da montagem de circuitos garantiu o funcionamento do modelo final.

Percebemos vários aspectos que podem ser melhorados no projeto como a implementação de contagem de vidas e a melhora na inicialização dos inimigos aplicando outra técnica para produzir mais de um inimigo em tela.

Também é perceptível algumas limitações no material. Em relação às dificuldades possíveis no jogo foi testada uma versão que aumenta a velocidade de movimento dos inimigos a cada ciclo, entretanto em velocidades mais altas o tempo de ativação e desligamento de cada seção da tela de LCD causa uma sensação de borramento da imagem. Desta forma optamos por apenas duas dificuldades que não causasse este mesmo borramento.

**Considerações Finais**

No final da produção conseguimos construir um jogo funcional, que alterna entre diferentes níveis de dificuldade e apresenta algumas animações que interagem com o usuário.

O objetivo principal do trabalho considera-se alcançado diante disto, entretanto é importante salientar que muitas melhorias podem ser aplicadas ao projeto, mesmo utilizando-se dos mesmos equipamentos e considerando algumas de suas limitações ainda é possível produzir uma experiência de interação com responsividade maior ao jogador.

Com relação ao desenvolvimento por parte dos alunos, consideramos um projeto adequado às expectativas da disciplina e ao nível de conhecimento dos integrantes do grupo, realizando uma tarefa de solidificação de alguns conhecimentos adquiridos no decorrer do semestre de estudo.