

# Jogo Pong - aplicações de Aruino e Processing 4

\* Sistemas Embarcados: Prof. Marco Reis - marco.reis@ba.docente.senai.br

1<sup>st</sup> Rodrigo Freire Bastos

Engenharia Elétrica

Senai Cimatec

Salvador, Brasil

rodrigo.bastos@aln.senaicimatec.edu.br

2<sup>nd</sup> Akin Silva

Engenharia Elétrica

Senai Cimatec

Salvador, Brasil

akin.silva@aln.senaicimatec.edu.br

**Abstract**—O trabalho desenvolvido é um sistema embarcado para o jogo eletrônico pong com baixo custo, feito usando processing como interface gráfica e um arduino com dois potenciômetros como controles para o jogo.

**Index Terms**—Sistema embarcado, Pong, Arduino, Processing.

## I. INTRODUÇÃO

O jogo Pong é um jogo eletrônico do genero Arcade desenvolvido em 1972 pela Atari, Inc. O jogo consiste no conceito de ping pong introduzido numa versão 2D.



Fig. 1. Pong game

Dessa maneira, o jogo para duas pessoas PvP ou PvIA tem como objetivo rebater a bola afim de fazer-la passar ultrapassar a barra do adversário e no fim, quem fizer mais ponto vence [2].

### A. Processing

Processing é um software open Source e linguagem de programação com base em java que utiliza artificios gráficos para representar artes visuais. Hoje, o processing possui uma enorme comunidade que ajuda a desenvolver diversos programas de niveis educacionais a profissonais [3].

Identify applicable funding agency here. If none, delete this.

### B. Arduino

Arduino é uma plataforma de prototipagem, é um software open Source que abrange toda a área da microeletrônica e microcontroladores. Hoje o desenvolvimento em arduinos pode ser realizado pela própria plataforma de arduino, a Arduino IDE, que possui todo suporte para o arduino e suas placas. Além do mais, o arduino possui uma vasta comunidade muito ativa e que fornece um grande suporte aos iniciantes, tornando essa plataforma num ótimo começo na eletrônica [1].

## DESENVOLVIMENTO

### C. Objetivo geral

O objetivo desse desafio é montar um sistema embarcado refazendo o clássico jogo Pong, usando arduino e processing.

### D. Objetivo específico

- Montar o arduino com 2 potenciômetros e 2 pushbuttons;
- montar o código do pong no processing;
- Montar uma conexão serial entre o arduino e o processing;
- mover as barras do pong com os potenciômetros;
- 2 pushbuttons para realizar a função start e reset do jogo.

## II. MATERIAIS

- arduino UNO R3;
- dois(2) potenciômetros de 1KΩ;
- dois(2) pushbuttons;
- dois(2) Resistores de 127Ω;
- Cabos para conexão;
- Placa de ensaio.

### A. Métodos

A construção do desafio foi dividida em duas partes, hardware e software.

A parte de hardware conta com a conexão de potenciômetros e pushbuttons, mas para a integração desses componentes com o pong foi necessário utilizar

A divisão do processing começou criando o ambiente do jogo, tela(); e telaInicial(); e utilizando a tela em modo fullScreen pela função fullScreen();

Toda movimentação feita pela bola, inclusive as colisões com as bordas e barras foram feitas separadamente afim de

```

14
15 //definir dimensoes do background e peças
16 void tela(){
17     background(#254195);
18     stroke(#FFFFFF);
19     strokeWeight(5);
20     noFill();
21     rect(10,10,width-20,height-50);
22
23     stroke(#FFFFFF);
24     strokeWeight(5);
25     line(width/2,10,width/2,height-40);
26
27 }

```

Fig. 2. Função para criação do ambiente de jogo

```

104 void setup(){
105
106 //definir tamanho e foto para fundo da tela
107 foto = loadImage("/home/rod/Computer-Vision-IEE-RAS-desafio/desafio_VC_RAS/images/penguin.jpg");
108 fullscreen();
109
110 }

```

Fig. 3. definição de tamanho de tela e plano de fundo

facilitar a leitura, dessa forma, definiu-se o tamanho velocidade e condições para movimentação da bola.

```

25 void movementDraw()
26 {
27     cont = cont + 0.0005;
28
29     xspeed = 2.8 + cont; // Speed of the shape
30     yspeed = 1.0 + cont;
31
32 // Update the position of the shape
33
34 xpos = xpos + ( xspeed * xdirection );
35 ypos = ypos + ( yspeed * ydirection );
36
37 // Test to see if the shape exceeds the boundaries of the screen and the paddle
38 // If it does, reverse its direction by multiplying by -1 or place the ball in the middle and increase the score
39

```

Fig. 4. movimentação da bola

```

40 if (xpos < 00 && ypos < mouseY+200 && ypos > mouseY) || xpos > width-00 && ypos < mouseY+200 && ypos > mouseY) {
41     xdirection *= -1;
42 } else if (xpos > width) {
43     scoreLeft = scoreLeft + 1;
44     xpos = width/2;
45     ypos = height/2;
46     delay(500);
47     xdirection *= -1;
48     cont = 1;
49 }
50 if (xpos < 0) {
51     scoreRight = scoreRight + 1;
52     xpos = width/2;
53     ypos = height/2;
54     delay(500);
55     xdirection *= -1;
56     cont = 1;
57 }
58 if (ypos > height-rad-30 || ypos < rad+30) {
59     ydirection *= -1;
60 }
61 }

```

Fig. 5. condições para colisão da bola

segundos para executar sua função, e, por conta do delay do sistema a LED RGB também ficou com um certo delay para mudar de cor. Por outro lado, o sistema desenvolvido consegue ler corretamente as distâncias em cm, identificar em qual região ela pertence e por consequência alertar a proximidade do objeto de com uma LED RGB inserida no sistema e também informar a distância do objeto em um LCD conectado no segundo arduino por meio de uma conexão serial.

## REFERÊNCIAS

- 1) Arduino. Arduino, 2022. Disponível em: [www.arduino.cc](http://www.arduino.cc). Acesso em: 25/maio/2022;
- 2) Britannica, The Editors of Encyclopaedia. "Pong". Encyclopedia Britannica, Invalid Date, <https://www.britannica.com/topic/Pong>. Accessed 3 July 2022.
- 3) Processing. Processing.org. Disponível em: <https://processing.org/>. Acesso em: 03/07/2022.

## III. METODOLOGIA

A atividade desenvolvida foi realizada com conhecimentos prévios sobre C++ e eletrônica básica em Arduinos, porém, para sanar as dúvidas e dificuldades encontradas no desenvolvimento da atividade foram utilizadas documentações de bibliotecas no site (1), e para realizar todo experimento foi utilizada a plataforma Tinkercad, que é muito bem otimizada além de ser gratuita e ser capaz de realizar simulações.

## IV. RESULTADOS

Ao concluir as conexões e códigos percebeu-se que só era possível ler e imprimir a informação passada de um arduino para outro caso houvesse um delay de ao menos 3 segundos entre cada envio de informação, dessa maneira o sistema não ficou muito otimizado, visto que, são necessários ao menos 3