

P-00-ng

Sistema FIEB



PLATAFORMA DE JOGO EM PROCESSING COM ARDUINO INTEGRADO

Ludmila Nascimento, Rafael Mello, Kauan Dantas e Gabriel Lopes

Orientador: Marco A. dos Reis

Robótica e Sistemas Autônomos, Senai Cimatec

Julho de 2022

Introdução

- Pong é o primeiro jogo eletrônico de sucesso da história, desenvolvido pela Atari Inc. e lançado no ano de 1972, iniciando uma febre de fliperamas.
- O jogo consiste na simulação de uma partida de tênis de mesa vista de cima, composta por uma bola, duas barras e o placar.
- O objetivo do jogo é fazer com que o oponente não consiga rebater a bola, somando pontos a medida que isso acontece.



Introdução ao projeto

- O projeto então tem em mente recriar o jogo Pong utilizando de tecnologias mais recentes.
- Para isso, será utilizada uma integração entre a linguagem de programação Processing e um microcontrolador



Processing

- Processing é uma linguagem de programação Open-Source e IDE (Ambiente de Desenvolvimento Integrado) construído para projetos visuais principalmente, e para servir como base de cadernos eletrônicos.
- É uma linguagem com grande assimilação às linguagens "C" e "Java", grandes inspirações da linguagem, resultando em algo que remete à programação em C com aspectos da programação orientada a objetos do Java.



Arduino

- Arduino é uma plataforma de prototipagem e hardware livre, projetado como um microcontrolador.
- Utilizando a linguagem C/C++ como base, o Arduino tornou-se bastante popular por apresentar baixo custo, flexibilidade e facilidade de manuseio em desenvolvimento de sistemas interativos.



Etapas do projeto

- Prototipagem no TinkerCAD do Hardware;
- Estabelecimento do envio de informação serial pelo Arduino;
- Modelagem dos controles do jogo;
- Primeiros passos da programação do jogo;
- Testes do jogo utilizando o teclado
- Recepção da informação serial no Processing;
- Integração dos eletrônicos nos controles;
- Finalização do código em Processing;
- Troubleshooting;

Materiais Necessários

Hardware

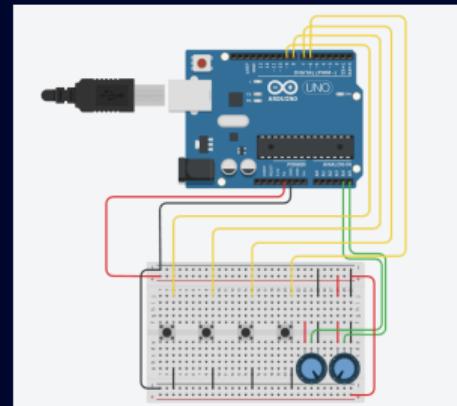
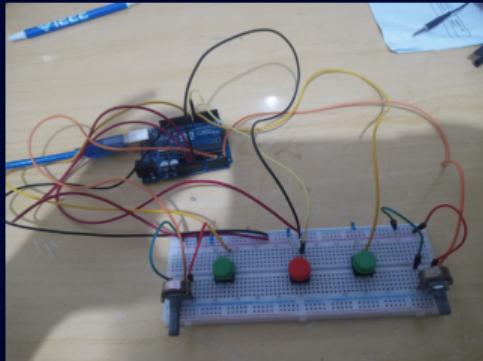
- Fios conectores;
- 3 Resistores de $1K\Omega$;
- 2 Botões com capa branca;
- 2 Potenciômetros;
- 2 cases impressas em 3D;
- 1 Arduino UNO R3;
- 1 Computador;

Software

- Visual Studio Code;
- GitHub;
- TinkerCad(C);
- Processing;

Hardware

- Circuito utilizado como os controles do jogo
- Os potenciômetros controlam as barras (paddles)
- Os botões fazem a transição de telas e pausam o jogo.



Hardware - Controles

- Modelos Impressos 3D para acomodar os botões e potenciômetros do circuito.
- Estes também são mais cômodos para os jogadores



Os modelos impressos.



A solda cos componentes no controle.

Hardware - Arduino

```
const int pot1 = A0, pot2 = A1;
const int b01 = 8, b02 = 7;
int v_pot1 = 0, v_pot2 = 0;
bool v_b01, v_b02;
int arr[10];

void setup() {
    Serial.begin(9600);
    pinMode(pot1, INPUT);
    pinMode(pot2, INPUT);
    for(int i = 7; i <= 8; i++){
        pinMode(i, INPUT_PULLUP);
    }
}
```

```
void loop() {
    v_pot1 = map(analogRead(pot1), 0, 1023, 0, 255);
    Serial.print(String(v_pot1) + "-");

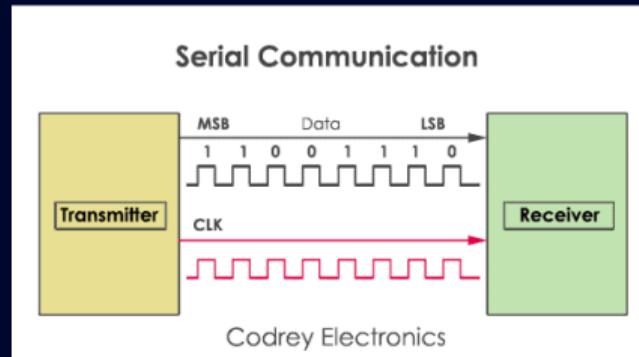
    v_pot2 = map(analogRead(pot2), 0, 1023, 0, 255);
    Serial.print(String(v_pot2) + "-");

    v_b01 = !digitalRead(b01);
    v_b02 = !digitalRead(b02);

    Serial.print(String(v_b01) + "-");
    Serial.print(String(v_b02) + "\n");
    delay(50);
}
```

Hardware - Arduino

- Como pode ser visto, uma string de modelo [x-x-x-x] é enviada ao Processing serialmente.
- No Processing, essa string será separada em caracteres e cada um sera atribuido a sua respectiva classe.



Software

Afinal, do que consiste o jogo "P-OO-ng"?

- Desenvolvimento do game em Processing;
- Menu principal, instruções e tela pause;
- Física da bola;
- Colisão com as barras;
- Interpretação do que o Arduino envia;
- Todo o desenvolvimento estético (formas, cores, etc).

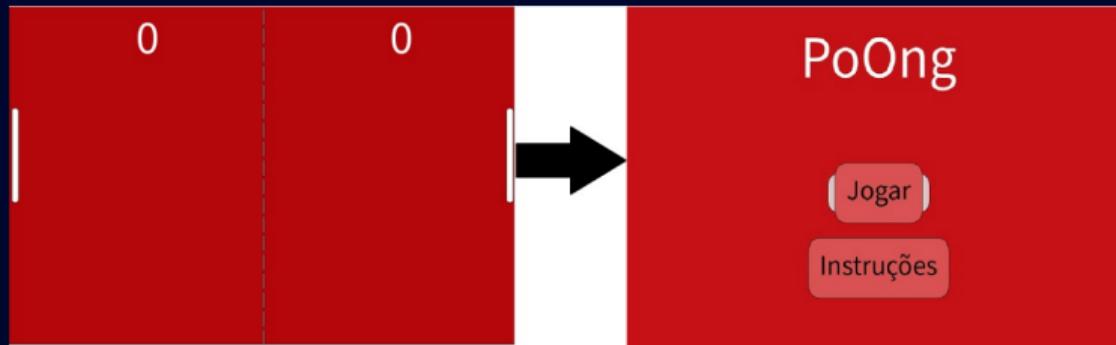
Software - Telas

- Definidas através de um "switch-case".
- Representam a interface em que o jogador está.
- Com a seleção de opções, a transição é efetuada.



Tela de início do jogo.

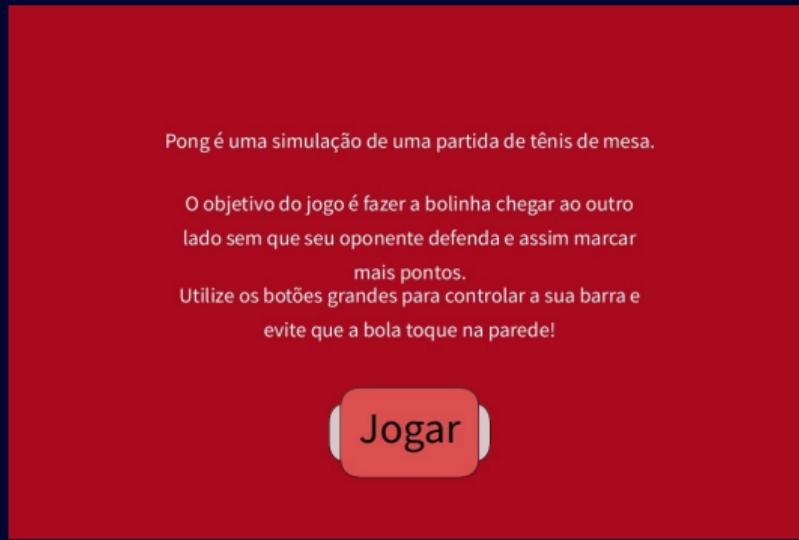
Software - Evolução da tela de início



Antes, quando era apenas o jogo X A tela de início criada.

Software - Tela de Instruções

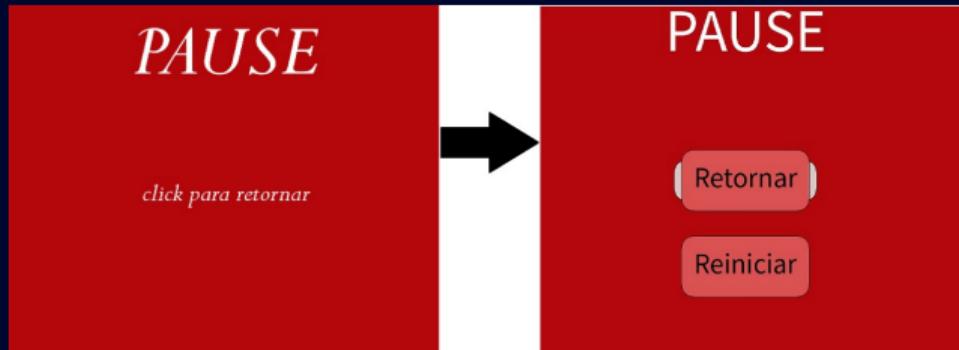
- Ensina brevemente como se jogar o jogo!



Tela de instruções do jogo.

Software - Tela Pause

- Quando o botão do controle é apertado enquanto o jogo estiver sendo jogado, a tela de pause aparece, zerando a velocidade da bolinha, deixando-a imóvel até que uma das opções seja selecionada novamente.



Tela de pause do jogo (antiga X nova).

Software - Lógica das telas

Exemplo da lógica para a tela inicial:

```
switch (ordem) { // Ordena as cenas do jogo
case 0:
tela_inicial();

if (click == 0 && (botao1.indexOf('1') != -1 || botao2.indexOf('1') != -1)) click = 1;

if (click == 1 && (botao1.indexOf('1') == -1) && (botao2.indexOf('1') == -1)) {

    if (int(strBarra1) > 127 || int(strBarra2) > 127) {
        click = 4;
        ordem = 2;
    } else {
        click += 1;
        ordem = 1;
    }
}

break;
```

Software - Componentes do jogo

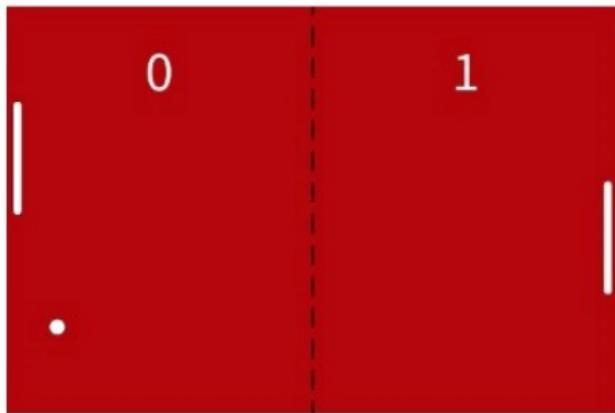


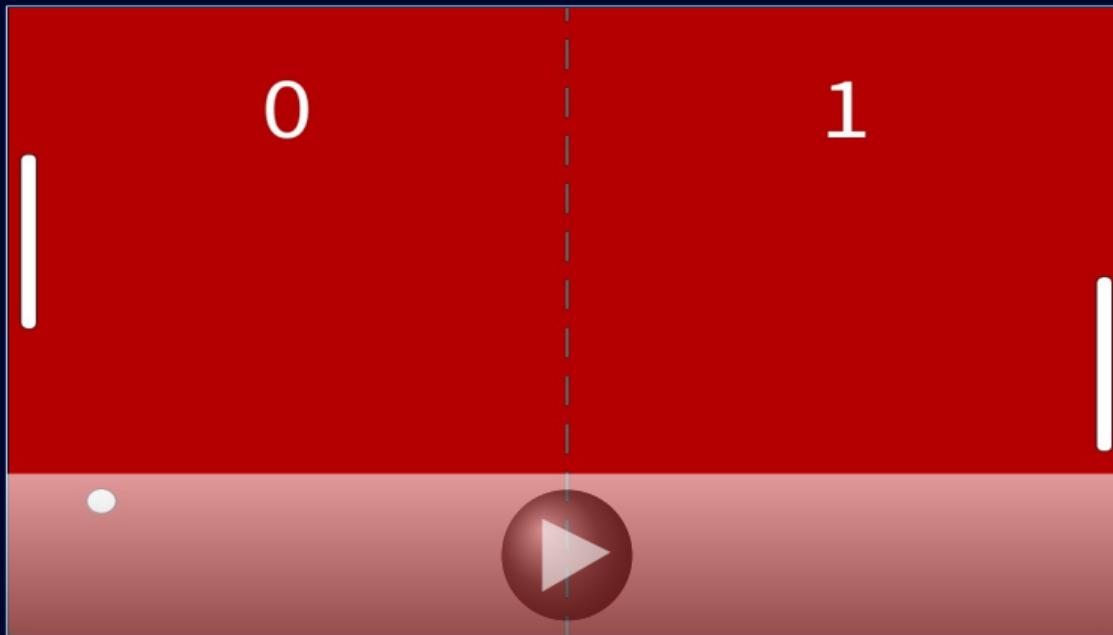
Figura 9 – Jogo rodando.

A dois objetos principais para fazer o jogo PONG funcionar, que é as duas barras laterais e a bola, ambas tem uma física inerente para funcinar, bem como seus desenhos no game. Logo abaixo segue a codificação desses objetos que são utilizados como classes dentro do jogo:

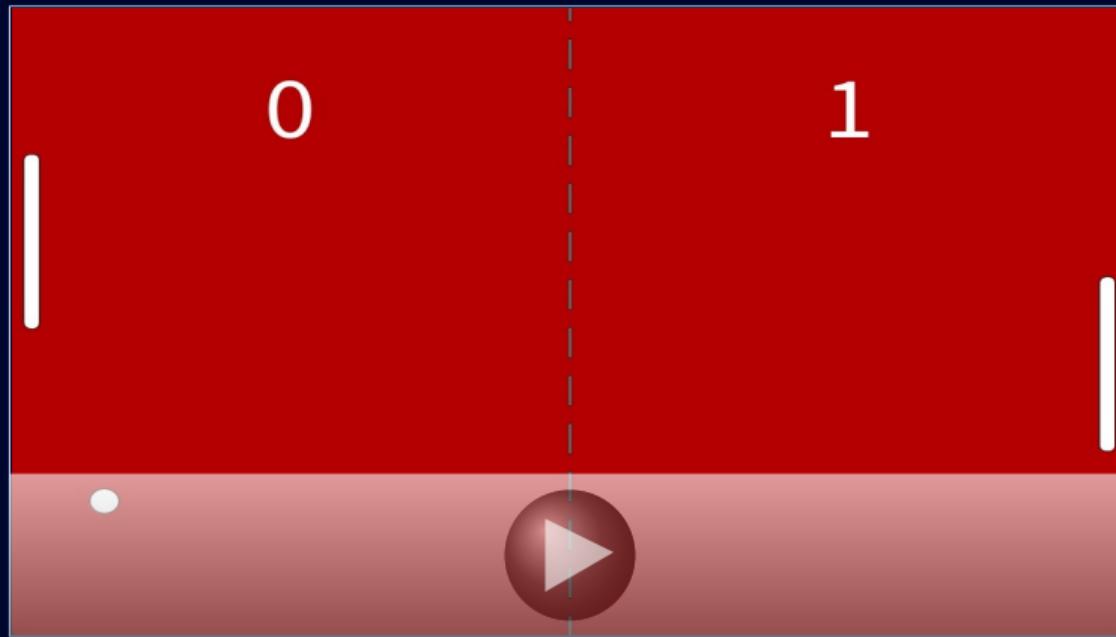
Software - Componentes do jogo

Vamos ver com mais detalhes no artigo! [▶ Link](#)

Demosntração 1



Demosntração 2



Algumas Referências (1)

- [1] COLUBRI, Andres; ZANANIRI, Elie; POTTINGER, Samuel. Processing. 2001. Disponível em: processing.org. Acesso em: 16 jun. 2022.
- [2] ARDUINO. Disponível em: arduino.cc. Acesso em: 29 maio 2022.
- [3] TANTAU, Till. Figures in Beamer: a detailed tutorial. A detailed tutorial. 2021. Disponível em: <https://latex-beamer.com/tutorials/beamer-figure/>. Acesso em: 01 jul. 2022.
- [4] ARNOLD, Ken; GOSLING, James; HOLMES, David. The Java programming language. Addison Wesley Professional, 2005.



Questions?

Obrigado pela atenção! :)

Temporary page!

\LaTeX was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page will go away, because \LaTeX now knows how many pages to expect for this document.