Construção de um plataforma de jogo em Processing com integração ao Arduino.

Ludmila Nascimento Dos Anjos

Rafael Ferreira Viana de Mello

Kauan Dantas Brito da Silva

Gabriel Lopes Guimarães

20/06/2022

Resumo

O presente projeto teve como finalidade a criação de uma plataforma de jogo que inclua o jogo "Pong", criado pela Atari Inc. em 1972, adaptando-o para tecnologias masis recentes. Nesse caso, foi utilizado a linguagem Processing para criar o jogo em si, e este foi integrado com um microcontrolador Arduino UNO, que coletará informações de potenciômetros e botões, acoplados a controles que foram impressos 3D, e enviará serialmente para o jogo, que processará as informações para executar ações, como mover as barras ou pausar. Em síntese, o projeto é uma composição mista de Hardware e Software que visa execução do jogo "Pong"em um computador.

Palavras-chave: Arduino. Processing. Hardware. Jogo Eletrônico.

Abstract

The current project aimed to create a game plataform that includes the game "Pong", developed by Atari Inc. in 1972, adapting it to newer tecnologies. For it's development, the Processing programming language was utilized to create the game itself, and it is integrated with an Arduino UNO microcontroller, which collects information from potentiometers ans buttons, coupled to controls that were 3D printed, and sending it serially to the game, which will process the gathered information to perform actions in game such as moving the paddles or pausing. In summary, the project is a mixed composition of Hardware and Software that aims to run the game "Pong" on a computer.

Keywords: Arduino. Processing. Hardware. Electronic Game.

1 Introdução

Em 1972, na garagem de um grupo de engenheiros que criariam uma empresa de jogos futuramente, a Atari, surgiu, de um pequeno exercício de simulação o que muitos consideram como o primeiro jogo da história: o "Pong". Na tentativa de simular uma partida de tênis de mesa ou "Ping-Pong"como é conhecido, o jogo eletrônico foi incrementado pelo grupo que tornou o jogo mais divertido e apropriado para o público.

Apesar de ter sido recusado por um cliente, alegando que prefereria um jogo de carros, um dos criadores não desistiu. "Bushnell convenceu um bar, chamado Andy Capp's, em Sunnyvale, na Califórnia, a instalar o Pong em uma máquina de fliperama —daquelas que funcionam com a inserção de moedas"(UOL, 2022).

Esse foi o salto inicial que, não só impulsionou o sucesso do jogo "Pong", mas também do mundo dos fliperamas, que teve sua alta na década de 80.

Com isso em mente, o presente trabalho tem como finalidade de recriar o jogo "Pong"utilizando de tecnologias mais modernas. Em suma, através da integração de um Arduino UNO, que irá receber a informação dos botões e joysticks(potênciometros) e enviará por comunicação serial ao jogo, desenvolvido utilizando a linguagem de programação Open Source "Processing", utilizada para programação dentro do contexto de artes visuais[2].

2 Desenvolvimento

Conforme o supracitado, o presente estuda visa o desenvolvimento de uma plataforma de jogo, adaptando o jogo "Pong"para tecnologias mais recentes. Inicialmente, para que hajam menos complicações eventuais, é primordial entender o funcionamento do jogo e suas características.

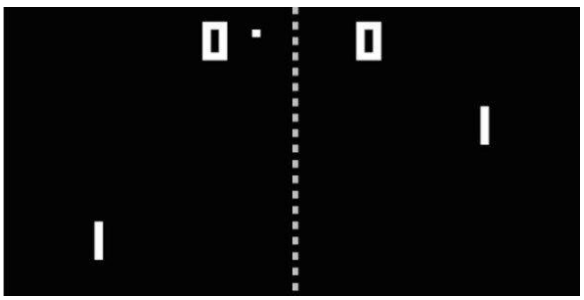

Figura 1 – Exemplo de como é o jogo "Pong".

O "Pong"é uma simulação de uma partida de tênis de mesa vista de cima, com alguns objetos que compõem a tela: duas barras, sendo uma para cada jogador, uma bola que ficará andando pela tela e o placar, com a pontuação de cada jogador respectivamente. O objetivo do jogo é que o jogador, controlando uma das barras, impeça a bola de bater no seu lado, somando pontos fazendo a bola chegar do outro lado sem que o oponente consiga "defender". As barras tem movimento unidimensional que pode ser controlado por um pequeno joystick composto por um reostato ou potenciômetro.

Como o projeto não envolve apenas o jogo, mas os controles deste, que utilizarão botões e potenciômetros, o primeiro passo será efetuar a integração desses com o Arduino UNO através do simulador virtual TinkerCAD, visando uma segurança maior e a garantia de que nenhum componente falhe durante a construção do Hardware.

Com isso, estabelecer uma comunicação entre o Processing e o Arduino é fundamental, já que as informações coletadas dos botões e potenciômetros devem passar de um para o outro. Essa comunicação será possível através do modelo serial, já que ambos podem enviar e receber informações por esse modelo com suas respectivas bibliotecas específicas e para que isso ocorra, basta utilizar o cabo USB do próprio microcontrolador e associar a porta a qual este está conectado ao Processing, garantindo que essas informações sejam enviadas. Isso ocorre de maneira mais simples por se tratar de uma comunicação Arduino-computador ao invés de dois microcontroladores diferentes, sem necessidade de utilizar métodos como o "Protocolo UART"nessa situação.

Em se tratando do jogo, que será programado na liguagem Processing, que é notadamente baseada em noções de "C"e "Java". Além do próprio "Pong", o jogo contará com uma tela de início, um menu com as instruções e uma tela de pause. Por ser uma linguagem de ambiente de desenvolvimento integrado (IDE), Processing se mostra apropriado para desenvolvimento de um jogo. A utilização de "classes"tangentes a da linguagem "Java"por exemplo, é uma maneira muito eficaz de se criar objetos dinâmicos que integram o jogo, como as barras e a bola. Por definição, as classes são um conjunto de atributos e métodos que podem ser instanciados por objetos, abstraindo-os.

3 Materiais e Métodos

Para o desenvolvimento do presente projeto, os componentes serão separados em duas partes: Hardware e Software. A parte referente ao hardware tangencia o físico do projeto, ou seja, os componentes eletrônicos e os controles. Já o Software refere-se à "mente do dispositivo", o que estará disposto na forma de programação, que será o próprio jogo. Assim sendo, os componentes são divididos da seguinte forma:

3.1 Hardware

- Fios conectores;
- 3 Resistores de 1KΩ, para proteger os botões.
- 2 Botões com capa branca;
- 2 Potenciômetros, que funcionarão como os joysticks;
- 2 cases impressas em 3D, que funcionará como os controles do jogo;
- 1 Arduino UNO R3, que coletará as informações dos botões e potênciometros;
- 1 Computador;

3.2 Software

- Visual Studio Code;
- GitHub, como plataforma para hospedar o desenvolvimento;
- TinkerCad(C), para simular o hardware do jogo;

- Processing;

O projeto visa um caráter aplicado, gerando conhecimento ao redor de uma aplicação prática e imediata. Isso será apoiado por uma pesquisa exploratória de modo à realização de estudos prévios para familiarização com o projeto em demanda.

Para sua execução, primordialmente, será efetuada uma simulação pela plataforma TinkerCAD, visando estabelecer maior garantia de que quando for construída fisicamente, a quantidade de possíveis falhas ou erros de projeto seja mínima. Em paralelo a essa etapa, o desenvolvimento da programação inserida no Arduino será realizado, adaptando-o se necessário para comunicar com o Processing sem demais problemas.

Em sequência, o jogo programado em Processing será iniciado, e ao longo de seu desenvolvimento, a integração com a comunicação serial será feita. Por fim, demais detalhes independentes da comunicação podem ser adicionados para tornar o jogo mais completo, como as próprias instruções do jogo, além de uma etapa selecionada para correção de erros presentes nos códigos e no hardware.

4 Resultados

4.1 Hardware

Como afirmado anteriormente, o primeiro passo do projeto é a prototipagem do Hardware do sistema, que inclui o Arduino, os botões e potenciômetros, através da plataforma online TinkerCAD, que inclui em si um monitor serial para visualizar o que está sendo enviado serialmente pelo microcontrolador.

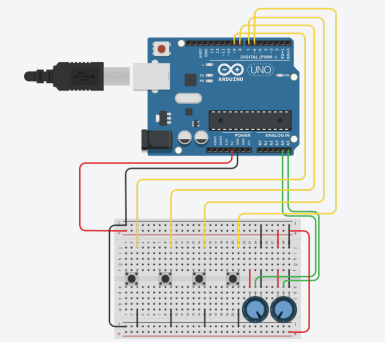

Figura 2 – Modelo prototipado no TinkerCAD.

Como pode ser visto na figura acima, o modelo foi criado para abrigar dois potenciômetros e uma quantidade de botões, que foram 4 no momento da simulação. Cada um destes associados a um pino digital respectivo. Assim, a função do Arduino será pegar as informações desses componentes eletrônicos e enviar serialmente para o computador serialmente. O que tornará isso possível é a utilização da biblioteca "softwareserial.h"e suas funções, que possibilitam a comunicação.

Para enviar todas as informações de uma única vez, em um pacote só, elas serão inseridas em uma string, e sua separação será realizada no código do Processing. Assim sendo, o código inserido no Arduino compreende a seguinte lógica:

```
1 const int pot1 = A5, pot2 = A3; //Potenciometros
```

```
const int b01 = 8, b02 = 7; // Botoes
int v_pot1 = 0, v_pot2 = 0; //Valores dos potenciometros
bool v_b01,v_b02; //Valores dos Botoes
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);
}
```

Além da declaração de variáveis, alguns recursos foram utilizados para saber por exemplo, se um botão estava sendo pressionado. Como ele está pinado em uma porta digital, é possivel utilizar uma propriedade chamada "INPUT_PULLUP"para fazer essa distinção. Isso ocorre já que o botão está conectado a um resistor de "Pull-Up", que está inbutido no Arduino UNO, e quando vemos a expressão abaixo, significa que essa resistência irá impactar naquele input quando acionado, atuando como um inversor, para assim se obter um nível lógico alto quando o botão é pressionado:

```
pinMode(i, INPUT_PULLUP);
```

Coletadas as informaçãos, elas são adicionadas a uma string de modelo: [x-x-x-x], e esta será separada por uma lógica criada na programação em Processing, transformando-as em chars, e associando-as às suas respectivas classes/variáveis.

Por fim, para acomodar os potenciômetros e botões, foi impresso em 3D um modelo de controle "joystick"chamado "Paddle Game Controller"apropriado para o "Pong", visto abaixo:

Figura 3 – Controle impresso 3D para o jogo.

Figura 4 – Eletrônicos soldados no controle.

4.2 Software/Processing

Antes de dar continuidade, é necessário estabelecer o que é desejado para o jogo: Uma tela inicial, que pode levar ao jogo em si ou a um menu de instruções, e que o jogo possa ser pausado. Isso foi pensado ao longo do desenvolvimento do jogo no qual, logo no inicio não tinha-se essa maturide, como pode-se observar na imagem abaixo:

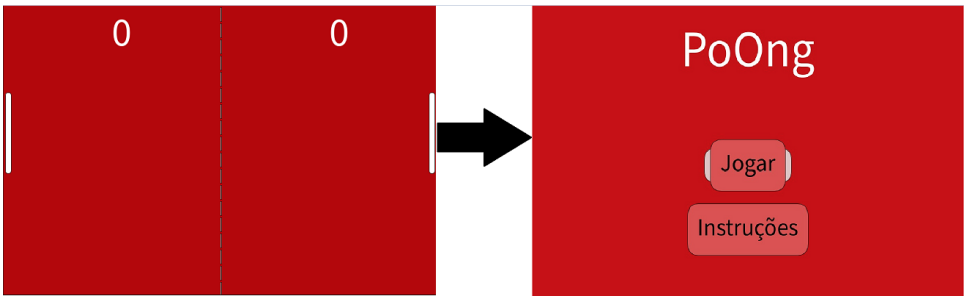

Figura 5 – Tela inicial do jogo.

Essa tela inicial foi confecionada através do sofware processing, em que essa IDE posssibilitou o visualização e programação do jogo PONG. Portanto a lógica feita no programa foi a seguinte:

```
1    void tela_inicial() { // Primeira tela
2   pont1 = 0;
3   pont2 = 0;
4
5   background(200, 0, 0);
6   textSize(height/6);
7   fill(255);
8
```

```
  textSize(pulsando);
  text("PoOng", width/2, height/3 -150); // Textos finais
  if (pulsando == height/6 ) pulsando = height/7;
  else pulsando += 1;

  if (int(strBarra1) > 127 || int(strBarra2) > 127) jogar.select_bot();
  else instrucoes.select_bot();

  jogar.escreve("Jogar");
  instrucoes.escreve("Instruções");
```

Através dessa função é possivel definir a cor de fundo do game com o "background", a cor da letra com o "fill", como também qual local o texto irá ser alocado com o "textSize"e especialmente determinar as opções da tela principal, que é "jogar"e "intruções". Com isso, selecionando a tela de instrções é possível observar a seguinte imagem:

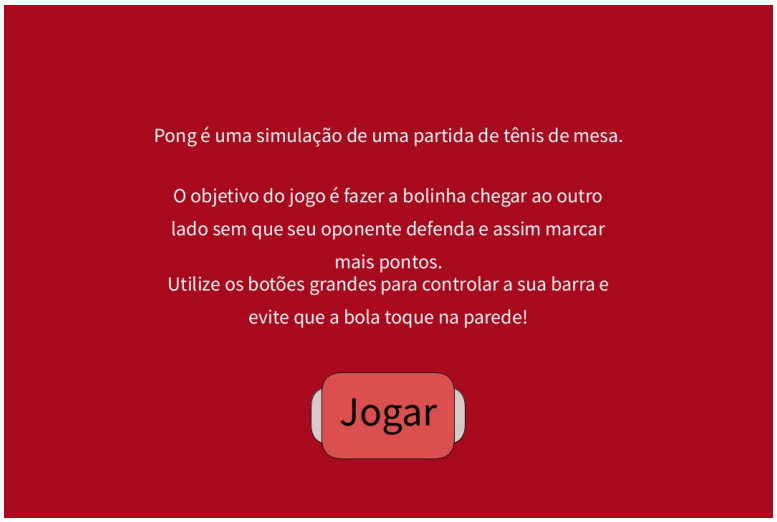

Figura 6 – Tela de instruções do jogo.

Bem como sua programação, que é determinada por meio de uma estrutura textual do processing, como já citada anteriormente denominada de"text"e caso a opção de "jogar"for selecionada o jogo inicia. Segue abaixo a lógica:

```
  void tela_instrucoes() {
     background(170, 10, 30);
     textSize(height/25);
     fill(255);

     text("Pong é uma simulação de uma partida de tênis de mesa.", width
  /2, height/4);

     String s = "O objetivo do jogo é fazer a bolinha chegar ao outro lado
   sem que seu oponente defenda e assim marcar mais pontos.";
     text(s, width/2, height/2-50, width/2+100, height/2);

     String n = "Utilize os botões grandes para controlar a sua barra e
  evite que a bola toque na parede!";
     text(n, width/2, height/2+50, width/2+100, height/2);

     b3.select_bot();
     b3.escreve("Jogar");
```

```
16      }
```

A tela de pause também teve sua evolução no decorrer do tempo, pois inicialmente não tinha-se tanta noção de como fazer e configurar um game, mas no decorrer do desenvolvimento o conhecimento foi se aprimorando e como consequência o jogo foi contnuando melhorando.

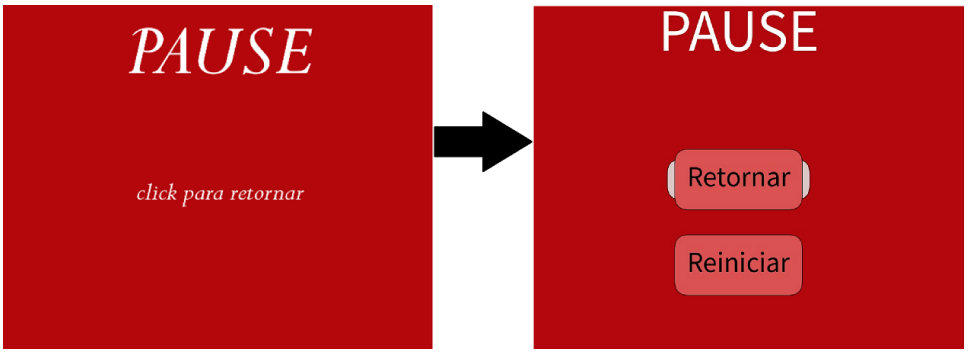

Figura 7 – Tela de pause.

A tela de pause é acionada por intermédio de dois botões, na qual ambos jogadores tem que combinar a parada do jogo juntos e selecionar se querem retornar ao game ou reinicia-lo. Conforme o código abaixo:

```
1   void tela_pause() { // Tela de pause do jogo
2
3   background(180, 0, 0);
4   textSize(height/6);
5   fill(255);
6   text("PAUSE", width/2, height/3 -200); // Textos de pause
7
8   if (int(strBarra1) < 127 || int(strBarra2) < 127) retornar.select_bot();
9   else reiniciar.select_bot();
10  retornar.escreve("Retornar");
11  reiniciar.escreve("Reiniciar");
12  //text ();
13
14  ordem = 3; //Mantém tela pause
15 }
16
```

Como também a sua chamada no "draw", no qual é o loop onde o jogo roda.

```
1   case 3:
2   tela_pause();
3   if (click == 6 && (botao1.indexOf('1') != -1 || botao2.indexOf('1') !=
     -1)) click = 7;
4
5   if (click == 7 && (botao1.indexOf('1') == -1) && (botao2.indexOf('1') ==
      -1)) {
6     if (int(strBarra1) > 127 || int(strBarra2) > 127) {
7       click = 4;
8       ordem = 4;
9     } else {
10      click = 4;
11      ordem = 2;
12    }
```

```
  }
}

```

Elaborou-se também uma tela de vencedor, no qual exibe o vencedor da partida(esquerdo ou direito), essa função também sofreu alterações com o decorrer do tempo, conforme explicitado na imagem abaixo:

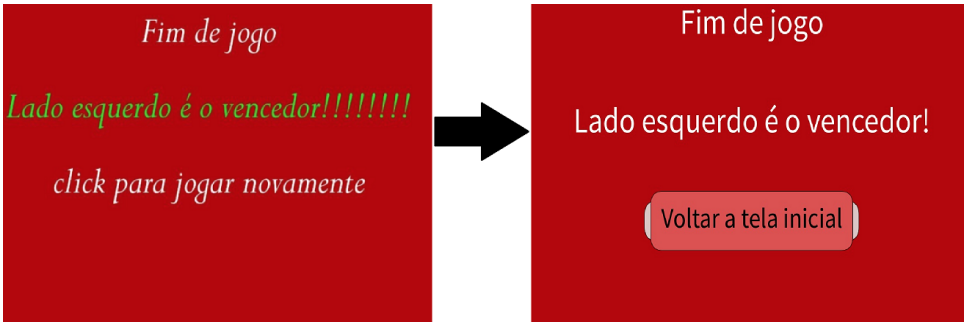

Figura 8 – Tela de vencedor.

Dessa forma foi definido um vencedor por intervenção da pontuação, ou seja, se o lado esquerdo estiver com uma maior pontução ele vence e acaba o jogo, de maneira analoga acontece com o lado direto se o mesmo obter mais pontos. Por fim o jogo oferece a opção de retornar a tela inicial, de acordo com a lógica abaixo:

```
  void tela_pause() { // Tela de pause do jogo

  background(180, 0, 0);
  void fim_jogo() { // função de fim de jogo
  if (pontc2 == vencedor) { //definindo vencedor como um lado
    ganhou = "Lado esquerdo é o vencedor!";
  } else if (pontc1 == vencedor) { //definindo vencedor como um  outro
   lado
    ganhou = "Lado direito é o vencedor!";
  }
  background(180, 0, 0);
  textSize(height/10);
  fill(255); // definindo a cor das letras como brancas
  text("Fim de jogo", width/2, height/3 -200); // Textos finais
  text(ganhou, width/2, height/3);

  voltar.select_bot();
  voltar.escreve("Voltar a tela inicial");

  ordem = 4; // Mantem a tela final ativa
}

```

Por último e mais importante, o jogo rodando, é importante salientar que o game em sua essência foi o mais modificado, não só visualmente, mas principalmente logicamente, pois ocorreram diversas alterações até chegar a versão final, no qual não pode-se quantificar em imagens ou em números.

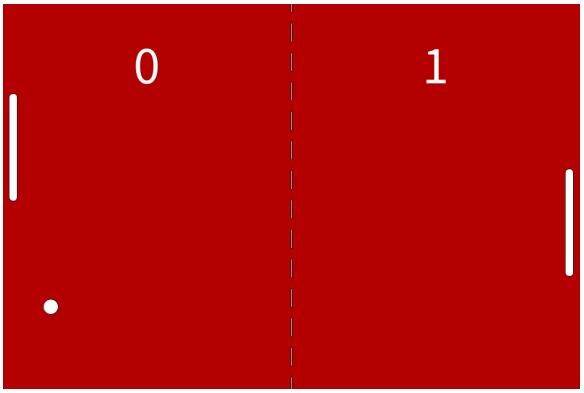

Figura 9 – Jogo rodando.

A dois obejtos 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:

classe da bola:

```
      class bol { // Classe da bola

  float x, y; // Posicao da bola
  float xspeed = 5, yspeed = 2.3; //  Velocidade da bola
  float r = 14; //raio da bola

  bol() {
    x = width/2;
    y = height /2;
  }
  void display() { // Desenha a bola
    stroke(0);
    ellipse(x, y, 28, 28);
  }

  void move() { // Move a bola
    x += xspeed;
    y += yspeed;
  }

  void checkb() { // Checa se bateu em algum lugar
    if (x > width || x < 0) {
      xspeed = xspeed * (-1);
    }
    if (y > height || y < 0) {
      yspeed = yspeed * (-1);
    }

  }

  void checkpont() { // Aumenta a pontuação dos jogadores
    if (x == 0) {
      pont1 += 2;

      x = width/2;
      y = height/2;
```

```
      xspeed = xspeed * (-1);
    }
    if (x == width) {
      pont2 += 2;

      x = width/2;
      y = height/2;
      xspeed = xspeed * (-1);
    }
  }

  void colisaobarrad(float xx, float yy, float largura, float altura) {
    if (x + r > xx - largura/2 && y - r < yy + altura/2 && y + r> yy -
    altura/2) { // condições para a bola recochetear
      xspeed *= -1; // inverter a velocidade da bola
    }
  }

  void colisaobarrae(float xx, float yy, float largura, float altura) {
    if (y - r < yy + altura/2 && y + r > yy - altura/2 && x - r < xx +
    largura/2) { // condições para a bola recochetear
      xspeed *= -1; // inverter a velocidade da bola
    }
  }
}
    }

```

classe da barra:

```
   class barra {

  float largura = 15;
  float altura = 200;
  float local_y = height/2; //posição y/2 para comerçar no centro da barra
  float local_x;

  barra(boolean esquerda) {
    if (esquerda) {
      local_x =  largura + 5; // o +5 serve para se afastar da borda
    } else {
      local_x = width - largura - 5;// o -5 serve para se afastar da borda
    }
  }

  void barra_inicio() { // configurando as barras iniciais
    fill(255);
    rect(local_x, local_y, largura, altura, 30);
  }

  void moverPot(int pos) {
    if (pos != local_y) {
      local_y = map(pos, 0, 255, 100, height - 100);  // Potenciometro
    }
  }
}

```

5 Conclusão

Tendo em vista o que foi aplicado, pode-se afirmar que o presente projeto conseguiu construir uma plataforma de jogo utilizando a linguagem de programação Processing e integrando um microcontrolador Arduino para criar controles para o jogo, com modelos para adequar os botões e potenciômetros impressos 3D para maior conforto em sua manipulação. O jogo em si possui um menu inicial, uma tela de instruções, detalhando como o jogo funciona e uma tela de pause, além do próprio "Pong".

A comunicação entre o Arduino, que coleta as informações dos controles, e o jogo em si programado em Processing fora efetuada de modo que o microcontrolador envia um pacote em formato de string ao jogo, que irá separá-los em "chars"e associa-los as suas respectivas funções dentro dele, na qual os potenciômetros controlam as barras de cada jogador e os botões efetuam seleções e/ou pausam o jogo enquanto este estiver sendo executado.

Referências

Referências

[1] PONG, o jogo que deu origem à indústria de videogames há 5 décadas. In: UOL. [S. l.], 21 fev. 2022. Disponível em: https://www1.folha.uol.com.br/tec/2022/02/pong-o-jogo-que-deu-origem-a-industria-de-videogames-ha-5-decadas.shtml. Acesso em: 16 jun. 2022.

[2] COLUBRI, Andres; ZANANIRI, Elie; POTTINGER, Samuel. Processing. 2001. Disponível em: processing.org. Acesso em: 16 jun. 2022.

[3] ARDUINO. Disponível em: arduino.cc. Acesso em: 29 maio 2022.

[4] OLIVEIRA, André Schneider de; ANDRADE, Fernando Souza de. Sistemas Embarcados: hardware e firmware na prática. 2. ed. [Salvador]: Editora Érica, 2010. 320 p.

[5] MCROBERTS, Michael. Arduino básico. Novatec Editora, 2018.

[6] ARNOLD, Ken; GOSLING, James; HOLMES, David. The Java programming language. Addison Wesley Professional, 2005.