



UNIVERSIDADE DE COIMBRA

Mestrado em Design e Multimédia
Tecnologia da Interface

Tecnologias de Interfaces em New Media Art

Instalação interativa com sistema de partículas

Cristina Verush (2023163523)

Laura Silva (2023186784)

Luisa Santos (202318751)

Coimbra, 2024

Sumário

1. Introdução.....	3
1.1. Enquadramento.....	3
1.2. Objetivos.....	4
1.3. Motivações.....	4
1.4. Calendarização.....	5
2. Conceito.....	6
2.1. Tema e Proposta de Projeto.....	6
2.2. Estudos de Casos Relevantes.....	6
3. Metodologia de Design de Interação.....	7
3.1. Abordagem de Design Centrado no Utilizador.....	8
3.2. Equipamentos Necessários.....	9
3.3. Tecnologia e Implementação Técnica.....	10
3.3.1. Arduino.....	11
3.3.2. Processing.....	12
3.3.3. MQTT.....	15
3.4. Testes de Usabilidade.....	16
4. Prototipagem.....	16
4.1. Desenvolvimento de Protótipos Iniciais.....	17
4.2. Refinamento do Protótipo.....	18
5. Resultado final.....	19
5.1. Implementação da Versão Final.....	20
5.1.1. Montagem do Hardware.....	20
5.1.2. Desenvolvimento do Software.....	20
5.1.3. Calibração e Ajustes Finais.....	20
5.2. Testes finais e validação.....	20
5.2. Demonstração dos Outputs.....	21
6. Conclusão.....	22
Referências.....	22

1. Introdução

O presente relatório documenta o projeto realizado no âmbito da unidade curricular de Tecnologias de Interface relativo ao plano de estudos de Mestrado em Design de Multimédia (MDM), na Universidade de Coimbra (UC).

O projeto foi dividido em três fases: 1) State of the Art and Theoretical discussion; 2) Conceptualisation and Prototyping; e 3) Proof of Concept. Na primeira fase foi nos dado vários temas com base no que era possível realizar, com o objetivo de podermos demonstrar o conhecimento adquirido nas aulas, então realizámos uma pesquisa mais aprofundada de cada tema e discutimos sobre qual gostaríamos de trabalhar. No fim da pesquisa e já escolhido o tema da New Media Art, iniciamos a segunda fase do projeto, que consistiu na definição de uma ideia e prototipagem inicial. Terminado as duas primeiras fases do projeto, entramos na terceira fase e mais complexa delas todas, a implementação do projeto em si, durante esta fase tivemos alguns obstáculos que tivemos de ultrapassar para chegarmos ao resultado pretendido, por fim o projeto teve a duração de três meses.

Nos próximos capítulos descreveremos, com mais detalhes, todas as fases existentes no decorrer do projeto até ao produto final. Este relatório relata também as tecnologias utilizadas para a realização do projeto, bem como a metodologia utilizada.

1.1. Enquadramento

A arte é uma expressão da humanidade em constante mudança, explorando infinitas habilidades criativas. Uma área interessante desse desenvolvimento está dentro da New Media Art, onde utiliza a interatividade e a tecnologia para criar experiências artísticas únicas para o utilizador. Neste trabalho, examinaremos as tecnologias de interface no âmbito da New Media Art, considerando a história, impacto e previsões futuras.

Quanto à proposta de projeto, este centrar-se-á na criação de uma obra de arte interativa. Neste contexto, a instalação artística terá de responder dinamicamente ao movimento do espectador, criando estruturas e partículas imaginárias que mudarão consoante a velocidade e direção do movimento. Para implementar esta proposta serão utilizados sensores de movimento conectados à placa Arduino, de modo que possa registar as mudanças na posição e velocidade do observador. Esses dados serão processados e utilizados para modificar o desenvolvimento das partículas em tempo real. O resultado esperado será uma experiência sensorial única e participativa, na qual o público se torna o cocriador da obra, influenciando diretamente na beleza visual e na temática do arranjo da obra. A interação entre o público e a obra visa explorar novas formas de expressão artística, quebrando barreiras tradicionais entre o espetador e as obras de arte.

Neste relatório, o nosso objetivo é explorar os fundamentos teóricos que sustentam a prática dos New Media, fornecer uma análise crítica das abordagens existentes e, em particular, destacar a relação entre teoria e prática neste campo. Acreditamos que esta compreensão teórica é essencial para apoiar qualquer projeto prático que desenvolvemos. O principal objetivo da proposta do projeto é demonstrar as amplas possibilidades criativas e técnicas oferecidas pelo uso de sensores. Queremos dar a oportunidade ao público de uma

experiência sensorial única e estimulante, onde não será apenas um espetador, mas que se sinta como um participante ativo na criação de uma obra de arte, interagindo com ela de forma única e significativa.

1.2. Objetivos

O objetivo central deste projeto é explorar as potencialidades das tecnologias de interface na criação de experiências artísticas interativas na New Media Art.

Um dos principais objetivos foi aprofundar a nossa compreensão dos fundamentos teóricos que sustentam a prática dos New Media. Isso implica em realizar uma investigação detalhada das abordagens existentes, oferecendo uma análise crítica que destaca a relação entre teoria e prática neste campo em constante evolução. Através dessa compreensão teórica, procuramos contextualizar a nossa prática artística dentro de um conceito robusto e atualizado.

Outro objetivo central é demonstrar as vastas possibilidades criativas e técnicas oferecidas pelo uso de sensores adaptados para o Arduino. Estes dispositivos proporcionam uma plataforma flexível e acessível para a criação de obras de arte interativas. Procuramos não apenas explorar as capacidades técnicas desses sensores, mas também destacar a sua aplicação na criação de uma experiência sensorial única e estimulante para o público.

Um aspecto fundamental deste projeto foi promover uma interação significativa entre o público e a obra de arte. Queremos criar um ambiente onde os espetadores não sejam meros observadores, mas sim participantes ativos na criação de significado e na construção da narrativa visual. Através dessa interação, queremos desafiar as convenções tradicionais da arte contemporânea, criando um espaço de co-criação entre artistas e espectadores.

Outro objetivo foi explorar as novas formas de expressão artística que desafiam as convenções estabelecidas. Procuramos criar um ambiente onde a experimentação e a inovação sejam incentivadas.

Por fim, pretendemos utilizar o feedback dos utilizadores para informar e aprimorar continuamente a experiência do público. Valorizamos a participação ativa dos espetadores e reconhecemos a importância da sua contribuição para o desenvolvimento do projeto. Desta forma, queremos garantir que a instalação final seja otimizada para envolver e cativar os espetadores de forma significativa, proporcionando uma experiência artística memorável e impactante.

1.3. Motivações

Sendo a nossa primeira experiência na área das tecnologias de interface mais propriamente ciências informáticas, o nosso principal objetivo passava por entender como funciona e conseguir incluir essa aprendizagem no meio que até então era desconhecido. Foi igualmente importante para nós alcançar um resultado positivo do trabalho desenvolvido.

Um dos maiores motivos pelo qual decidimos escolher esta ideia prendeu-se com a vontade de aprender novos conhecimentos. Assim sendo, um dos principais objetivos era poder aprender e conhecer novas ferramentas, evoluir para aplicar os conhecimentos adquiridos até então. Nesse aspeto, os docentes Tiago Cruz e Ana Cláudia estiveram sempre dispostos a partilhar o seu conhecimento e feedback, independentemente do grau de conhecimento e experiência. Para além disso, outro objetivo era podermos acabar o projeto e dizer que se fecha um ciclo muito importante e portanto tínhamos de dar o nosso melhor e ser exigentes com nós mesmas. Por fim, mas não menos importante, este projeto teve como objetivo estimular o nosso trabalho em equipa.

1.4. Calendarização

Como referido anteriormente, o projeto teve duração de três meses e durante esse período havia tarefas estabelecidas pelos docentes. Os pontos seguintes mostram as datas de início e fim de cada etapa, seguido do Diagrama de Gantt.

Milestone 0 – Definição do grupo e do tema (**23 de fevereiro - 7 de março**)

Milestone 1 – Estado da Arte e Discussão Teórica (**7 de março - 18 de abril**)

Milestone 2 – Conceptualização e Prototipagem (**18 de abril - 30 de maio**)

Milestone 3 – Prova de Conceito (**30 de maio - 9 de junho**)

Milestone 4 – Documentação (**9 de junho**)

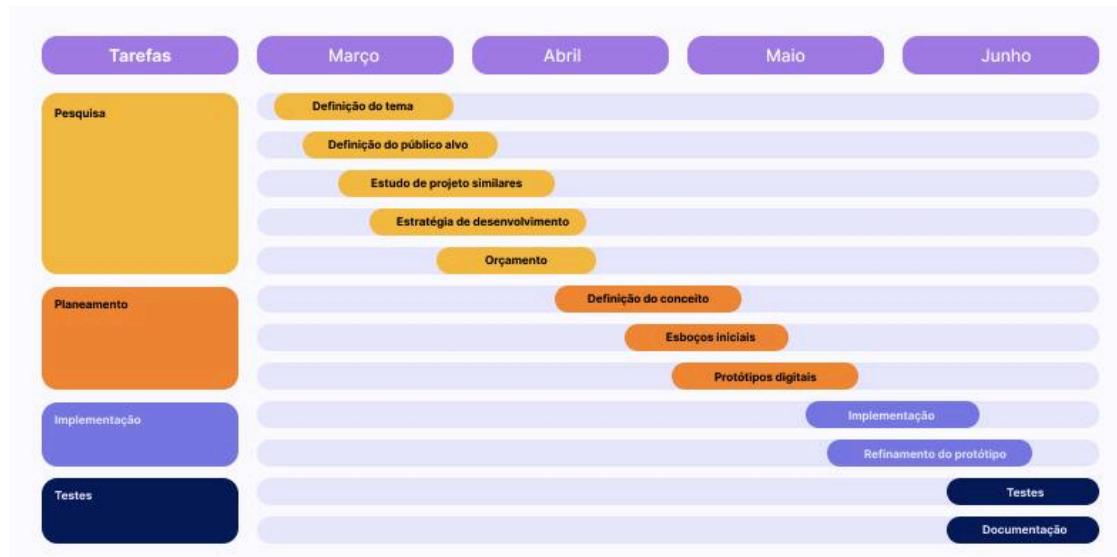


Figura 1 - Diagrama de Gantt

2. Conceito

Este projeto pretende a apresentação para além de uma obra de arte, visando criar um ambiente imersivo e participativo onde os espetadores tornam-se co-criadores ativos da experiência estética. A inspiração nos principais fundadores da New Media Art não é apenas uma escolha estética, mas sim uma afirmação sobre a natureza da arte contemporânea e do seu papel na sociedade digital. Ao explorar e expandir as fronteiras da relação entre arte, tecnologia e participação pública, a instalação procura não apenas entreter, mas também procura a interação direta entre utilizador e “máquina”.

2.1. Tema e Proposta de Projeto

A interatividade foi o ponto central desta proposta artística, permitindo que cada movimento dos participantes seja imediatamente traduzido em elementos visuais dinâmicos que transformam o ambiente em tempo real. Neste contexto, a experiência artística deixa de ser passiva e observacional para tornar-se uma jornada personalizada e participativa, onde cada gesto desempenha um papel crucial na construção da narrativa visual. Ao capacitar os participantes a assumirem o controle do ambiente artístico que os rodeia, esta instalação desafia as convenções tradicionais da arte e convida o público a colocar-se ativamente na criação de significado e na construção de novas formas de expressão.

A detecção de movimento emerge como uma ferramenta fundamental neste contexto, oferecendo uma forma imediata e altamente personalizada de interação entre os espectadores e a obra de arte. Cada gesto e movimento dos participantes tornam-se dados que alimentam o sistema interativo, permitindo que cada ação tenha um impacto visual imediato e tangível nas projeções artísticas. Esta dinâmica transforma a experiência geral da instalação em um diálogo contínuo entre arte e público, desafiando as noções tradicionais de autoridade e receção na arte contemporânea.

2.2. Estudos de Casos Relevantes

Para a conceptualização do projeto, usamos como inspiração principal trabalhos referidos na Milestone anterior, entre os quais, dois destacaram-se por trazerem resultados muito semelhantes aos que pretendemos. Primeiro, o trabalho realizado por Michelle YAO, nominado como #EXPRESS [1], visto que este trabalho utiliza um sensor de movimento, no caso um Kinect, e que o resultado final apresenta um jogo de partículas dinâmicas, com base nas emoções do utilizador.



Figura 2- #EXPRESS

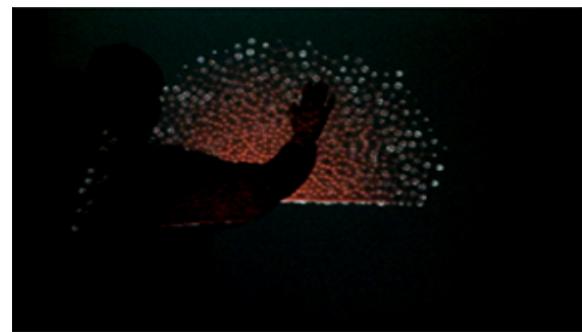


Figura 3- #EXPRESS

Um outro projeto que nos inspirou para o projeto, destacou-se o projeto *Interactive Art Project* [2], da Arduino Education, neste caso usa o HC-SR04 mostrando uma exemplificação de como o sensor funciona e integra o gesto físico de avançar e recuar a mão para controlar a velocidade de uma espiral.



Figura 4- Interactive Art Project

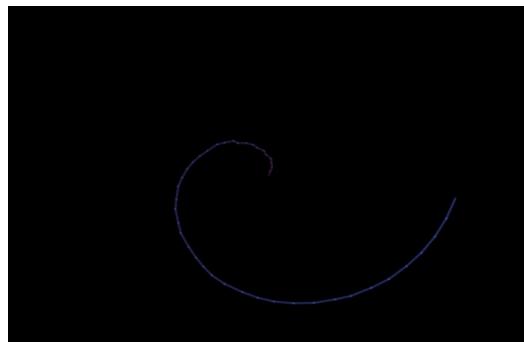


Figura 5- Interactive Art Project

Também para esta fase inicial, encontramos o conceito de *Swarm Intelligence*, este conceito compreende onde grupos de animais ou indivíduos trabalham juntos para alcançar um objetivo comum, adaptando-se e se auto-organizando conforme necessário. Sendo também aplicado na inteligência artificial, para desenvolver sistemas capazes de resolver problemas complexos de forma eficiente, adaptando-se e aprendendo com o ambiente.

3. Metodologia de Design de Interação

A pesquisa iniciou-se com uma discussão coletiva sobre diversos temas, optando por explorar a New Media Art. A ideia não estava definida, então o grupo procurou inspiração em exemplos [3] e referências. Artistas [4] como Nam June Paik [5] e Mario Klingemann [6], que nos deram o contexto para explorações. O grupo criou uma MiroBoard para organizar e visualizar, coletivamente, os exemplos encontrados. A pesquisa incluiu diversos trabalhos que enriqueceram a compreensão do grupo sobre a proposta, tais como *VOID* [7], *Interactive art demonstration* [8] e *Interactive art installation with sensors* [9], como casos de estudo que achamos relevantes para a exploração.

Com base na pesquisa, decidiu-se criar uma projeção interativa onde os visuais seriam influenciados pelo movimento [10], utilizando o conceito de Swarm Intelligence [11] [12] [13] num sistema de partículas que imitassem estrelas. Identificamos que o sensor ultrasónico (HC-SR04) [14] [15] [16] [17] era amplamente utilizado em projetos semelhantes. Fundamentado pelos projetos, pesquisou-se as características, vantagens e desvantagens do componente para garantir a adequação ao projeto.

Na primeira apresentação (Milestone 1), os docentes sugeriram que o projeto fosse "wireless" para permitir a projeção em uma sala inteira, recomendando o uso do ESP32 [18] [19] [20]. O grupo então pesquisou sobre a nova tecnologia e como utilizá-la com um servidor MQTT [21]. No entanto, enfrentamos desafios técnicos com a conexão wireless do ESP32 via servidor MQTT, além de dificuldades na captura de movimento com o sensor.

Para a Milestone 2, apresentamos modelo da caixa em 3D, adaptado de um projeto de caixa para abrigar um HC-SR04 [22], para guardar e proteger os componentes. Um protótipo conectando o ESP32 via USB ao computador, devido aos problemas de conexão com o servidor, por motivos de demonstração da funcionalidade do código com o sensor ultrassônico. Recebendo feedback, para a adição de mais inputs visuais ou sonoros. O grupo optou por incluir dois inputs, um input sonoro, desejado através de palmas, utilizando o sensor KY-037 [23] e um input visual, utilizando a webcam do computador para captar movimento modificando a função do HC-SR04 para ajustar o tamanho das partículas.

Com a ajuda de um dos docentes, resolvemos o problema da conexão com o MQTT e começou-se a impressão 3D da caixa que protegeria o circuito. Enquanto a impressão estava em andamento, o grupo melhorou o código para o produto final, resolvendo problemas visuais e de funcionamento dos componentes. Após a impressão, montamos todas as componentes na caixa impressa em 3D, resultando num artefacto funcional que integrava os elementos de movimento, som e visual, conforme planeado.

3.1. Abordagem de Design Centrado no Utilizador

A abordagem de design centrado no utilizador (DCU) foi um pilar fundamental na criação da nossa instalação de arte interativa com sensores Arduino. Esta abordagem garante que o projeto não apenas incorpora a tecnologia de forma eficiente, mas também proporciona uma experiência intuitiva e envolvente para todos os utilizadores. O processo DCU seguiu várias etapas essenciais, cada uma focada em entender e integrar as necessidades dos utilizadores finais no desenvolvimento da instalação.

O primeiro passo foi a realização de uma pesquisa inicial para compreender o nosso público-alvo. Realizámos entrevistas com potenciais utilizadores, incluindo estudantes universitários, artistas e visitantes ocasionais de exposições de arte. Estas entrevistas forneceram insights valiosos sobre como diferentes grupos percebem e interagem com instalações de arte interativa. Descobrimos que muitos utilizadores tinham pouca familiaridade com tecnologias como o Arduino, mas estavam entusiasmados em explorar novas formas de arte interativa.

Com base nos dados recolhidos, criámos várias personas que representavam os diferentes perfis dos utilizadores. Exemplos de personas incluíam um estudante curioso de artes

visuais, um profissional de tecnologia interessado em novas aplicações do Arduino e um turista que visita exposições de arte para relaxar e se inspirar. Para cada persona, desenvolvemos cenários de utilização detalhados, mapeando como cada tipo de utilizador interage com a instalação ativa com os sensores e a visualização das partículas geradas.

3.2. Equipamentos Necessários

Para a realização do nosso projeto, selecionamos uma série de equipamentos acessíveis, garantindo flexibilidade e eficácia na implementação. O Sensor Ultrassônico HC-SR04 será empregado para medição precisa de distâncias. Complementando essa funcionalidade, o Sensor de Ruído KY-037 permitirá a deteção e monitorização de níveis sonoros. Além disso, uma webcam, será utilizada para captura de vídeo. Estes componentes, devido à sua versatilidade e facilidade de integração com microcontroladores como o Arduino, formam a base tecnológica e econômica necessária para o sucesso do nosso projeto.

Sensor HC-SR04

O Sensor HC-SR04 foi uma opção altamente acessível e amplamente empregada em conjunto com placas Arduino e outros dispositivos para medição de distâncias por meio de ondas sonoras, notadamente em projetos de robótica para evitar objetos. Diferentemente da primeira geração do HC-SR04, que estava restrita a operar somente entre 4,8V e 5V, esta versão mais recente possui uma faixa de voltagem de entrada mais ampla, possibilitando sua utilização com controladores operando em 3,3V .



Figura 6- HC-SR04

A precisão de medição alcança até 3 mm, e o ângulo efetivo é inferior a 15°. Este sensor proporciona um método econômico e simples de medir distâncias, utilizando ondas sonoras. Um pulso ultrassônico, emitido pela unidade a uma frequência acima da audição humana, cerca de 40kHz, é transmitido, e a distância até o alvo é determinada pela medição do tempo necessário para o retorno do eco. Apresenta excelente precisão de alcance e leituras estáveis em um formato fácil de utilizar.

Além disso, por já possuir pinos, o sensor pode ser facilmente conectado a uma breadboard sem a necessidade de solda, simplificando o processo de montagem. A escolha desse sensor também se deve ao fato de ser amplamente utilizado, além de ser uma opção econômica, com preços variando de aproximadamente 1,30€ a 4€ .

Sensor KY-037

O Sensor de ruído foi uma opção altamente acessível e amplamente empregada em conjunto com placas Arduino e outros dispositivos para deteção e medição de níveis de som, notadamente em projetos de automação e monitorização ambiental. Um exemplo popular é o módulo KY-037, que permite a deteção de ruídos ambientais em uma faixa de voltagem de operação que varia tipicamente entre 3,3V e 5V, possibilitando sua utilização com uma ampla gama de microcontroladores.

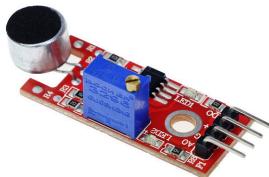


Figura 7 - KY-037

Por já possuir pinos compatíveis, o sensor pode ser facilmente conectado a uma breadboard sem a necessidade de solda, simplificando o processo de montagem e experimentação. A escolha deste sensor deve-se ao fato de ser amplamente utilizado e documentado na comunidade maker, além de ser uma opção econômica, com preços variando de aproximadamente 1,50€ a 3€.

Webcam

As webcams são dispositivos de captura de vídeo utilizados em projetos que requerem monitorização visual ou interação por vídeo.



Figura 8 - Webcam

3.3. Tecnologia e Implementação Técnica

Os sensores escolhidos desempenharam funções específicas no sistema. Por exemplo, o sensor HC-SR04 foi responsável por medir distâncias, interpretadas como o tamanho da aglomeração de estrelas na visualização. A deteção de movimento na projeção foi possível graças à câmera do computador, que funcionou como sensor de movimento, capturando imagens em tempo real do ambiente. O sensor de ruído adicionou um efeito à visualização, refletindo nas mudanças sonoras repentinas. Além disso, um sensor de Wi-Fi integrado ao ESP32 estabelece a comunicação sem fio, conectando o Arduino ao servidor MQTT.

3.3.1. Arduino

Com relação ao Arduino, várias bibliotecas (Figura 9) foram incluídas para fornecer funcionalidades específicas, como conectividade WiFi, comunicação MQTT e uso de sensores de ultrassom. As bibliotecas utilizadas para essas funções são: WiFi, PubSubClient e NewPing, respectivamente. Além disso, foram definidas variáveis para armazenar as credenciais do WiFi, os detalhes do broker, os tópicos e a porta para o MQTT.

```
const char *ssid = "IoT-Test";
const char *password = "Denohd0dkooz80ir";

const char *mqtt_broker = "10.6.0.57";
const char *topic = "Test";
const char *topic_ultrasonic = "ultrasonic";
const char *topic_noise = "noise";
const int mqtt_port = 1883;
WiFiClient espClient;
PubSubClient client(espClient);
```

Figura 9- Bibliotecas usadas

Figura 10- Variantes para MQTT

O código configura o sensor de ultrassom (Figura 11), definindo os pinos de trigger e echo, além da distância máxima de detecção, e inicializa um objeto NewPing com esses parâmetros. Também configura um pino de entrada para um sensor de ruído (Figura 12).

```
#define TRIGGER_PIN 13
#define ECHO_PIN 12
#define MAX_DISTANCE 200

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
```

Figura 11- Configuração do sensor de ultrassom

```
const int noisePin = 34;
int val = 0;
```

Figura 12- Configuração do pino de ruído

Na função setup, a comunicação serial é configurada, a conexão WiFi é iniciada e o programa tenta conectar ao broker MQTT, subscreve aos tópicos definidos e publica uma mensagem inicial (de testagem). A função callback é chamada quando uma mensagem é recebida em um tópico subscrito, imprimindo a mensagem recebida no console serial.

```
void setup() {
  Serial.begin(115200);
  pinMode(noisePin, INPUT);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("Connecting to WiFi..");
  }
  Serial.println("Connected to the Wi-Fi network");
  client.setServer(mqtt_broker, mqtt_port);
  client.setCallback(callback);

  while (!client.connected()) {
    if (client.connect("esp32-client")) {
      Serial.println("Connected to MQTT broker");
    } else {
      Serial.print("Failed with state ");
      Serial.print(client.state());
      delay(2000);
    }
  }
  client.publish(topic, "Hi, I'm ESP32 <3");
  client.subscribe(topic);
}
```

Figura 13- Função Setup

```
void callback(char *topic, byte *payload, unsigned int length) {
  Serial.print("Message arrived in topic: ");
  Serial.println(topic);
  Serial.print("Message:");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();
  Serial.println("-----");
```

Figura 14- Função callback

Na função loop, o código lê a distância do sensor de ultrassom e a intensidade do ruído do sensor de som, publicando esses dados nos tópicos MQTT correspondentes, mantendo a conexão com o broker MQTT ativa.

```
// Publish ultrasonic data
char ultrasonicData[50];
sprintf(ultrasonicData, "{\"size\":\"%c\", \"distance\":%d}", size, distance);
client.publish(topic_ultrasonic, ultrasonicData);

// Publish noise data
char noiseData[20];
sprintf(noiseData, "{\"noise\":\"%c\"}", ruido);
client.publish(topic_noise, noiseData);

client.loop();
}
```

Figura 14- Função loop

3.3.2. Processing

O programa utiliza bibliotecas para comunicação serial, captura de vídeo e MQTT para conectar-se e interagir com os sensores. A configuração inicial envolve a inicialização das variáveis e dos parâmetros necessários para a captura de vídeo, cliente MQTT e outros elementos essenciais para o funcionamento do sistema.

```
MQTTClient client;           int cellSize = 20;
Serial myPort;               float ballX = -1;
Capture video;               float ballY = -1;
int numPixels;               float targetX = -1;
int[] previousFrame;         float targetY = -1;
int[] differenceFrame;       float ballSpeed = 20;
int numCols;                 float smoothing;
import processing.serial.*;   int numRows;
```

Figura 15- Bibliotecas

Figura 16- Configuração Inicial

A classe ‘Adapter’ Implementa a interface MQTT Listener para gerenciar a conexão MQTT e processamento de mensagens recebidas. Conecta-se aos tópicos "ultrasonic" e "noise" e processa as mensagens recebidas, ajustando as partículas de acordo com os dados dos sensores.

Todas as partículas movem-se rapidamente em direções aleatórias antes de retornarem ao comportamento normal se o valor recebido for maior a 40, simulando uma explosão. Em relação ao tamanho das partículas, este é ajustado com base na distância: se a distância for menor que 20 cm, o tamanho aumenta; se for maior, diminui.

```

Galaxy galaxy;

class Adapter implements MQTTListener {
    void clientConnected() {
        println("Client connected");
        client.subscribe("ultrasonic");
        client.subscribe("noise");
    }

    void messageReceived(String topic, byte[] payload) {
        String message = new String(payload);
        println("New message: " + topic + " - " + message);

        if (topic.equals("noise")) {
            JSONObject json2 = parseJSONObject(message);
            if (json2 != null) {
                int val = json2.getInt("val");
                if (val > 40) {
                    for (Particle p : galaxy.particles) {
                        p.explode();
                    }
                }
            }
        }
    }
}

```

Figura 17- Class Adapter

A função Setup Configura a tela em modo fullscreen, inicializa o adaptador MQTT, conecta ao broker MQTT e inicia a captura de vídeo. Inicializa a galáxia com um número específico de partículas. Enquanto o Draw atualiza e exibe a galáxia. Lê os frames da webcam e calcula a diferença entre frames consecutivos para detectar movimento. E ajusta a posição das partículas e a bola de acordo com o movimento detectado.

```

void draw() {
    background(0);
    galaxy.update();
    galaxy.display();

Adapter adapter;

void setup() {
    fullScreen();
    adapter = new Adapter();
    client = new MQTTClient(this, adapter);
    client.connect("tcp://192.168.150.121:1883",

galaxy = new Galaxy(3500); // QUANTIDADE
ballX = width / 2;
ballY = height / 2;
video = new Capture(this, width, height);
video.start();
numPixels = video.width * video.height;
previousFrame = new int[numPixels];
differenceFrame = new int[numPixels];
numCols = width / cellSize;
numRows = height / cellSize;
}
}

// Movimento Webcam
if (video.available()) {
video.read();
video.loadPixels();
for (int i = 0; i < numPixels; i++) {
    color currColor = video.pixels[i];
    color prevColor = previousFrame[i];
    int currR = (currColor >> 16) & 0xFF;
    int currG = (currColor >> 8) & 0xFF;
    int currB = currColor & 0xFF;
    int prevR = (prevColor >> 16) & 0xFF;
    int prevG = (prevColor >> 8) & 0xFF;
    int prevB = prevColor & 0xFF;
    int diffR = abs(currR - prevR);
    int diffG = abs(currG - prevG);
    int diffB = abs(currB - prevB);
    differenceFrame[i] = color(diffR, diffG, diffB);
    previousFrame[i] = currColor;
}
}
}

```

Figura 18- Setup

Figura 19- Draw

Para o movimento a webcam captura vídeo e detecta movimento comparando cada frame com o anterior. A posição da bola na tela é ajustada para o ponto com maior mudança detectada. As partículas na visualização se ajustam em brilho e posição em relação à posição da bola, criando uma interação visual baseada no movimento detectado pela câmera.

```

// Detección de movimiento
if (frameCount > 30) {
    float maxDiff = 0;
    for (int col = 0; col < numCols; col++) {
        for (int row = 0; row < numRows; row++) {
            int x = col * cellSize;
            int y = row * cellSize;
            int pixelIndex = y * video.width + video.width - x - 1;
            float diffBrightness = brightness(differenceFrame[pixel]);
            float diffNorm = diffBrightness / 255;
            if (diffNorm > maxDiff) {
                maxDiff = diffNorm;
                targetX = x + cellSize / 2;
                targetY = y + cellSize / 2;
            }
        }
    }
    // Suavizar
    targetX = lerp(targetX, ballX, 0.1);
    targetY = lerp(targetY, ballY, 0.1);
}

// Actualización de la posición de la bola
if (ballX == -1 && ballY == -1) {
    ballX = width / 2;
    ballY = height / 2;
}

// Movimiento de la bola hacia el punto de destino
if (targetX != -1 && targetY != -1) {
    float dx = targetX - ballX;
    float dy = targetY - ballY;
    float distance = dist(ballX, ballY, targetX, targetY);
    smoothing = map(distance, 0, width, 0.02, 0.02); // velocidad
    ballX += dx * smoothing;
    ballY += dy * smoothing;
}

// Lógica para controlar el movimiento de las partículas y
galaxy.adjustParticles(ballX, ballY);

for (Particle p : galaxy.particles) {
    p.adjustPosition(ballX, ballY);
    p.adjustBrightness(ballX, ballY);
}

```

Figura 20- Deteção de movimento

A classe ‘Galaxy’ é responsável por gerenciar um conjunto de partículas que compõem a galáxia. No construtor, ela inicia uma lista de partículas com base no número fornecido. Ela possui métodos para atualizar e exibir as partículas da galáxia. Através do método ‘adjustParticles’, este ajusta a posição das partículas com base em um ponto de destino específico, criando uma dinâmica de movimento na visualização.

```

class Galaxy {
    ArrayList<Particle> particles;

    Galaxy(int numParticles) {
        particles = new ArrayList<Particle>();
        for (int i = 0; i < numParticles; i++) {
            float startX = random(width);
            float startY = random(height);
            particles.add(new Particle(startX, startY));
        }
    }

    void update() {
        for (Particle p : particles) {
            p.update();
        }
    }

    void display() {
        for (Particle p : particles) {
            p.display();
        }
    }

    void adjustParticles(float targetX, float targetY) {
        for (Particle p : particles) {
            p.adjustPosition(targetX, targetY);

            p.x += random(-1, 1);
            p.y += random(-1, 1);
        }
    }
}

```

Figura 21- Class Galaxy

Por outro lado, a classe ‘Particle’ representa cada partícula na galáxia. Estas possuem propriedades como posição, velocidade, tamanho e cor. Seus métodos incluem ‘update’ para atualizar a posição da partícula, ‘display’ para exibir a partícula na tela, ‘adjustBrightness’ para ajustar o brilho da partícula com base em sua posição em relação a um ponto de referência e ‘adjustPosition’ para ajustar a posição da partícula de acordo com um ponto de destino. Além disso, possui um método ‘explode’ que simula um comportamento explosivo ao alterar a velocidade da partícula.

```

class Particle {
    float x, y, vx, vy, size, targetSize;
    color baseColor;
    float maxBrightness = 255;
    float minBrightness = 10;
    boolean isExploding = false;

    Particle(float startX, float startY) {
        x = startX;
        y = startY;
        vx = random(-1, 1);
        vy = random(-1, 1);
        baseColor = color(255);
        size = random(0.5, 3);
        targetSize = size;
    }

    void adjustBrightness(float targetX, float targetY) {
        float d = dist(x, y, targetX, targetY);
        float brightness = map(d, 200, width - 600 /*Distancia
        baseColor = color(brightness);
    }

    void adjustPosition(float targetX, float targetY) {
        float d = dist(x, y, targetX, targetY);
        float attractionRadius = 90; // Diámetro de atracción
        if (d < attractionRadius) {
            float newX = targetX + random(-50, 50);
            float newY = targetY + random(-50, 50);
            x = lerp(x, newX, 0.1);
            y = lerp(y, newY, 0.1);
        }
    }

    void explode() {
        isExploding = true;
        vx = random(-5, 5);
        vy = random(-5, 5);
    }
}

```

Figura 22- Class Particle

3.3.3. MQTT

Para configurar o servidor Mosquitto, foi essencial para garantir a comunicação adequada entre os dispositivos. O endereço IP e a porta do Mosquitto foram verificados usando o comando "netstat -ano | findstr :1883".

Para a integração do Arduino e do Processing usando MQTT via Mosquitto, foi feito o download e instalado o Mosquitto do site oficial ou do gestor de pacotes do sistema operacional. O estado do Mosquitto foi verificado usando o comando "mosquitto -v". O Arduino foi configurado para se comunicar com o Mosquitto via MQTT usando a biblioteca "PubSubClient", garantindo que ambos estivessem conectados à mesma rede. O endereço IP e a porta do servidor Mosquitto foram estabelecidos no código do Arduino, configurando os tópicos MQTT necessários.

Foi instalada uma biblioteca de cliente MQTT para Processing e o código foi configurado para se conectar ao servidor Mosquitto usando a biblioteca MQTT. As mensagens recebidas no Arduino foram processadas e enviadas conforme necessário. No Processing, foi implementada a lógica para processar e enviar mensagens conforme necessário. A conexão MQTT estabelecida entre o Arduino e o Processing foi usada para enviar dados do Arduino para o Processing para visualização ou controle de dispositivos.

Os códigos Arduino e de Processing foram essenciais para o funcionamento do sistema. O código Arduino leu os dados dos sensores e os enviou para o servidor MQTT (Mosquitto), representando os dados do sensor HC-SR04 e do sensor de ruído nos tópicos MQTT "ultrasonic" e "noise", respectivamente. No Processing, o código conectou-se ao servidor MQTT para receber as mensagens enviadas pelo Arduino. Quando uma mensagem foi recebida, o código ajustou a visualização de acordo com os dados recebidos.

Estes códigos juntos formaram um sistema de comunicação entre o Arduino e o Processing usando MQTT via Mosquitto, permitindo que o Arduino enviasse dados dos sensores para o Processing, que os processava para controle e visualização. Durante o desenvolvimento do sistema, o MQTT Explorer foi uma ferramenta valiosa para monitorizar e depurar a comunicação MQTT.

3.4. Testes de Usabilidade

A fase de prototipagem rápida foi crucial para validar nossas ideias iniciais. Construímos protótipos funcionais simples usando sensores de movimento e ruído conectados ao Arduino, que foram testados em sessões de feedback com utilizadores reais. Estes testes revelaram que os utilizadores inicialmente não percebiam que os seus movimentos e sons estavam a ser captados. Em resposta, introduzimos indicadores visuais e auditivos para tornar a interação mais clara e imediata.

Conduzimos vários testes de usabilidade ao longo do desenvolvimento do projeto. Nestes testes, observámos como os utilizadores interagem com a instalação, focando em aspectos como a facilidade de iniciar a interação, a compreensão dos efeitos visuais resultantes e a satisfação geral. Utilizamos métodos como a observação direta e questionários pós-uso para recolher feedback detalhado. As observações mostraram que a altura e o posicionamento dos sensores eram cruciais para uma interação fluida, levando-nos a ajustar a configuração física da instalação.

Desde o início, a inclusão e acessibilidade foram prioridades. Ajustamos a altura dos sensores para garantir que utilizadores em cadeiras de rodas pudessem interagir confortavelmente. Adicionalmente, simplificamos a interface visual, utilizando cores contrastantes e feedback tátil para garantir que a experiência fosse acessível a pessoas com diferentes capacidades visuais e auditivas. Esta consideração permitiu uma interação mais ampla, garantindo que todos os utilizadores pudessem participar plenamente.

Durante todo o desenvolvimento do projeto, mantemos um canal aberto para feedback contínuo. Participantes dos testes iniciais foram convidados a revisitar a instalação e a fornecer feedback adicional à medida que o projeto evolui. Implementámos uma metodologia ágil, permitindo que ajustaremos rapidamente o design com base no feedback recebido. Esta prática assegurou que estávamos continuamente alinhados com as expectativas e necessidades dos utilizadores.

4. Prototipagem

A fase de prototipagem desempenha um papel crucial no desenvolvimento de projetos de arte interativa, proporcionando uma oportunidade para explorar e refinar conceitos, testar funcionalidades e validar a viabilidade técnica e estética da obra de arte final. Neste projeto específico, a fase de prototipagem foi marcada por uma abordagem ampla que combinou modelagem 3D, simulações e testes práticos para garantir não apenas a funcionalidade, mas também a integração estética na instalação final.

4.1. Desenvolvimento de Protótipos Iniciais

Durante esta etapa, outra iniciativa importante foi a descoberta e adaptação de um modelo 3D para abrigar os componentes essenciais do projeto, como os sensores Arduino HC-SR04 e o ESP32 Wrover-B. Essa adaptação não se limitou apenas à funcionalidade técnica dos componentes, mas também levou em consideração a estética e a harmonia com o conceito artístico geral da instalação. Por meio de iterações cuidadosamente projetadas, foram refinados os detalhes do design para garantir uma integração perfeita dos componentes dentro do contexto visual e espacial da obra de arte.

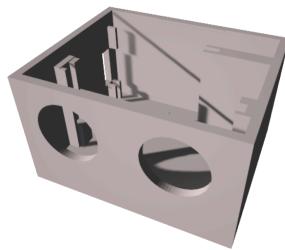


Figura 23 - Caixa para o circuito e sensor

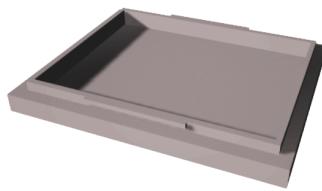


Figura 24 - Tampa da caixa

Para demonstrar e entender melhor como a interação dos participantes influencia a experiência artística, foram realizadas simulações 2D. Essas simulações permitiram uma análise da dinâmica da interação, destacando não apenas os aspectos técnicos, mas também os elementos visuais e interativos que compõem a experiência geral. Por meio dessas simulações, foi possível refinar a interação entre os participantes e o artefato, ajustando e otimizando os elementos para criar uma experiência imersiva e única.

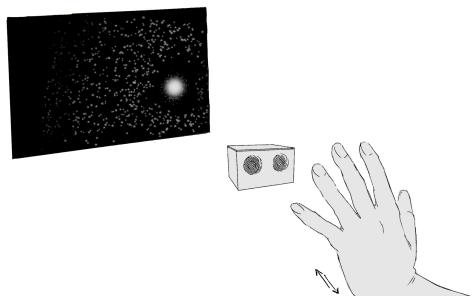


Figura 25- Movimentação com os sensores para a projeção

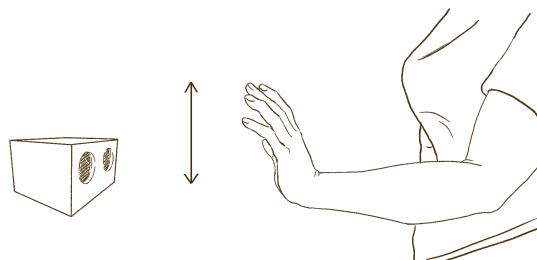


Figura 26- Movimentação com os sensores para a projeção

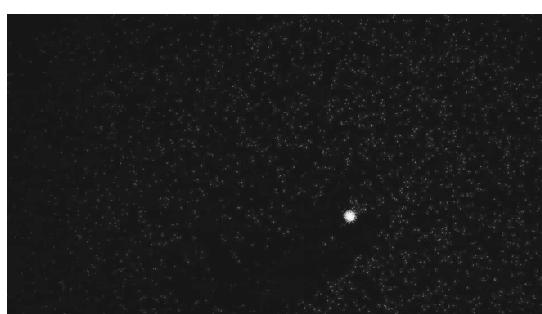


Figura 27 - Screenshot da interação no Processing

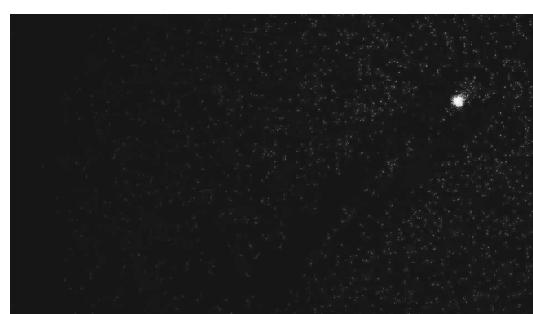


Figura 28 - Screenshot da interação no Processing

As figuras apresentadas durante esta fase de prototipagem desempenham um papel essencial na documentação e na comunicação dos resultados obtidos. Elas não apenas ilustram os detalhes técnicos do projeto, como a disposição dos componentes e o fluxo de interação, mas também capturam a essência estética da experiência artística.

4.2. Refinamento do Protótipo

Durante o refinamento do protótipo, várias iniciativas importantes foram tomadas para garantir a integração harmoniosa de todos os componentes essenciais do projeto. Uma dessas iniciativas foi a descoberta e adaptação de um modelo 3D para abrigar os sensores Arduino HC-SR04, o ESP32 Wrover-B, e posteriormente, o sensor de ruído. A adaptação do modelo 3D não se limitou apenas à funcionalidade técnica dos componentes, mas também levou em consideração a estética e a harmonia com o conceito artístico geral da instalação. Por meio de iterações cuidadosamente projetadas, refinamos os detalhes do design para garantir uma integração perfeita dos componentes dentro do contexto visual e espacial da obra de arte.

A adição do sensor de ruído exigiu modificações no modelo 3D original. Foi necessário criar um espaço específico para acomodar este novo sensor e a breadboard, garantindo que ele estivesse posicionado de forma a capturar com precisão os níveis de som ambiente. As alterações incluíram ajustes nas dimensões internas do modelo e a criação de aberturas adequadas para a captura eficiente de som pelo sensor. Este processo envolveu várias iterações e protótipos para assegurar que o sensor de ruído estivesse corretamente integrado e funcional dentro do conjunto.

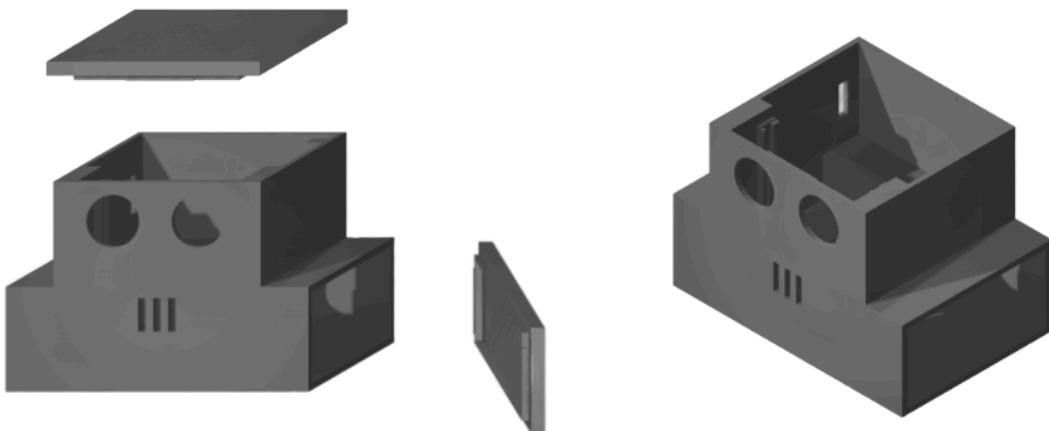


Figura 29- Modelo 3D da caixa para o circuito

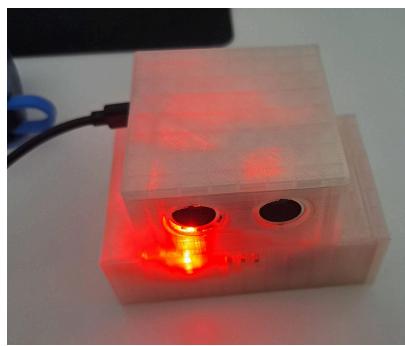


Figura 30- Caixa para o circuito

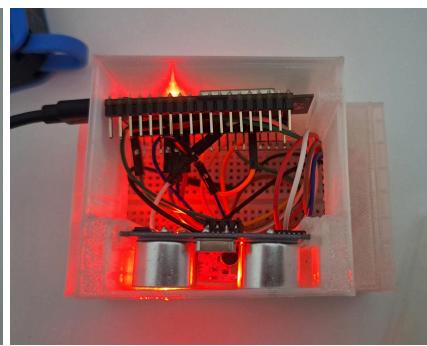


Figura 31- Caixa para o circuito

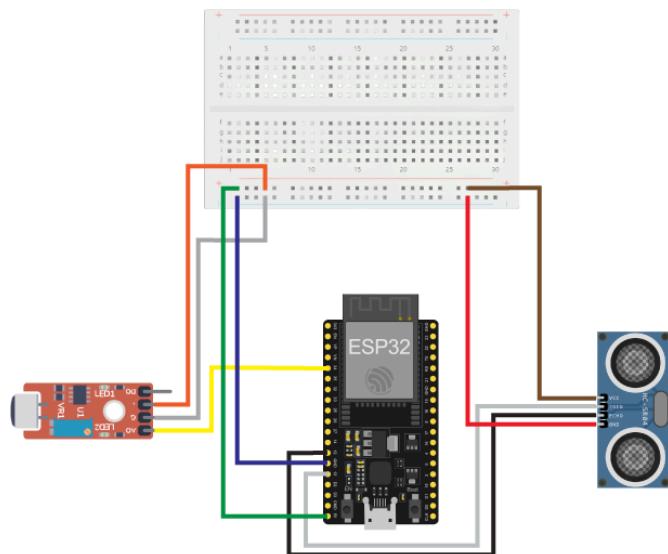


Figura 32- Circuito

As figuras apresentadas mostram o diagrama do circuito e o modelo 3D impresso, permitindo-nos testar e validar a integração de todos os componentes físicos. A precisão da impressão 3D foi crucial para garantir que cada sensor estivesse firmemente encaixado e operando corretamente dentro do modelo.

A inclusão do sensor de ruído trouxe uma nova dimensão à nossa instalação artística, adicionando uma camada adicional de interatividade e sensorialidade. Este refinamento foi fundamental para garantir que a experiência final fosse rica, envolvente e responsiva às interações do público.

5.Resultado final

O desenvolvimento final do projeto representou a culminação de meses de trabalho e refinamento, integrando diversas tecnologias e conceitos para criar uma instalação de arte interativa imersiva e dinâmica. A adição do sensor de ruído foi um dos componentes centrais neste projeto, trazendo uma nova dimensão de interatividade e resposta sensorial à instalação.

5.1. Implementação da Versão Final

A implementação da versão final envolveu a integração completa de todos os componentes do projeto em um sistema coeso e funcional. Esta fase incluiu várias etapas cruciais:

5.1.1. Montagem do Hardware

Todos os sensores, incluindo os sensores de movimento HC-SR04 e o sensor de ruído KY-038, foram montados e conectados ao Arduino e ao ESP32 Wrover-B. As conexões foram cuidadosamente verificadas para garantir a integridade dos dados e a funcionalidade dos sensores. A caixa de 3D projetada anteriormente para abrigar os componentes foi utilizada para organizar e proteger o hardware, garantindo uma apresentação estética e funcional.

5.1.2. Desenvolvimento do Software

O código do Arduino foi finalizado para ler os dados dos sensores e enviá-los via MQTT ao servidor Mosquitto. Os tópicos MQTT foram definidos claramente para cada tipo de dado (movimento e som) para facilitar a comunicação com o Processing. No lado do Processing, o código foi ajustado para receber os dados do MQTT e gerar respostas visuais dinâmicas. A lógica de visualização foi refinada para criar uma experiência fluida e responsiva, onde os movimentos e sons do público influenciavam diretamente a exibição artística.

5.1.3. Calibração e Ajustes Finais

Calibraramos os sensores para garantir a sensibilidade e a precisão do sistema. Para o sensor de ruído, foram realizados testes em diferentes ambientes com diversos níveis de som. Embora tivemos alguma dificuldade em relação à captura do som, ajustamos o máximo possível a sensibilidade para garantir que apenas sons significativos gerassem uma resposta visual, tentando evitar reações a ruídos de fundo. Similarmente, os sensores de movimento foram calibrados para captar movimentos a distâncias variadas.

Finalmente, a instalação física foi preparada com o material e componentes necessários previamente mencionados.

5.2. Testes finais e validação

Os testes finais e a validação foram essenciais para assegurar que o sistema funciona conforme esperado e proporciona a experiência desejada. Esta fase envolveu várias etapas:

Realizamos testes para verificar a funcionalidade de cada componente do sistema. Os sensores foram testados individualmente e em conjunto para garantir a precisão e a consistência dos dados. A comunicação via MQTT foi monitorada para assegurar que os dados eram transmitidos e recebidos corretamente.

Participantes foram convidados para interagir com a instalação, fornecendo feedback sobre a experiência. Esses testes de usabilidade ajudaram a identificar qualquer problema ou área de melhoria, permitindo ajustes finais no software e na calibração dos sensores.

A resposta visual do sistema foi avaliada para garantir que fosse atraente e coerente com o conceito artístico do projeto. Ajustes finais foram feitos na visualização para melhorar a fluidez e a resposta às interações do público.

A instalação foi validada tanto do ponto de vista técnico quanto artístico. Do ponto de vista técnico, verificou-se que todos os componentes funcionam conforme esperado. Do ponto de vista artístico, a instalação foi validada para garantir que a experiência proporciona a imersão e a interatividade desejadas.

5.2. Demonstração dos Outputs

Quanto ao resultado final, temos então uma integração de diferentes sensores para alcançar os objetivos desejados. A webcam é responsável pelo controle do movimento. O sensor ultrassônico deteta a distância a que se encontra o espetador, modificando assim o tamanho das partículas. Por fim, o sensor de ruído é utilizado para simular a explosão.



Figura 33- Movimento com Webcam



Figura 34- Modificação do tamanho com HC-SR04



Figura 35 - Simulação da explosão com KY-037

6. Conclusão

O presente projeto representa uma jornada emocionante e desafiadora na interseção entre arte, tecnologia e interatividade. Ao longo deste trabalho, exploramos as potencialidades das tecnologias de interface na criação de experiências artísticas interativas na New Media Art, com foco específico na deteção de movimento e sensor de ruído utilizando o Arduino.

Iniciámos o nosso percurso com uma profunda imersão no estado da arte e na discussão teórica, onde contextualizamos o nosso trabalho dentro do panorama da New Media Art e identificámos referências importantes que nos inspiraram. Através de uma abordagem centrada no utilizador, procuramos não apenas criar uma obra de arte interativa, mas também promover uma experiência imersiva e participativa para o público.

Ao longo das fases de concepção e prototipagem, refinamos a nossa ideia inicial e exploramos diferentes abordagens técnicas para alcançar o nosso objetivo. A adaptação de um modelo 3D para abrigar os componentes essenciais do projeto foi um passo crucial, permitindo-nos integrar os sensores de movimento e ruído de forma harmoniosa dentro da instalação artística.

Durante o desenvolvimento final e os testes subsequentes, enfrentamos desafios técnicos e estéticos que nos exigiram criatividade e resiliência para superar. A integração dos sensores Arduino HC-SR04 e do sensor de ruído, juntamente com a comunicação entre hardware e software, exigeu uma abordagem cuidadosa e iterativa para garantir o funcionamento adequado da instalação.

No entanto, apesar dos desafios, o resultado final é uma obra de arte interativa que transcende as fronteiras da expressão convencional. Ao permitir que os espetadores se tornem co-criadores ativos da experiência estética, a nossa instalação desafia as noções tradicionais e receção na arte contemporânea, promovendo uma nova forma de interação entre artista e público.

À medida que concluímos este projeto, reconhecemos que a nossa jornada está longe de terminar. A New Media Art é um campo em constante evolução, e esperamos que o nosso trabalho sirva como um ponto de partida para futuras explorações e experimentações nesta área emocionante.

Em última análise, o nosso objetivo é criar não apenas uma obra de arte, mas sim uma experiência transformadora que inspire, provoque reflexão e promova um diálogo significativo entre os espetadores e o mundo que os rodeia. Este projeto é apenas o começo de uma jornada mais ampla em busca da fusão entre arte, tecnologia e humanidade, e mal podemos esperar para ver para onde ela nos levará.

Referências

- [1] Behance. (n.d.). *#EXPRESS*. Retrieved from
<https://www.behance.net/gallery/46677951/EXPRESS>
- [2] Arduino. (n.d.). *Interactive Art Project*. Retrieved from
<https://www.arduino.cc/education/interactive-art-project/>
- [3] Euroinnova. (n.d.). *Arte con Arduino: Mira 10 ejemplos de obras de arte elaboradas con Arduino*. Retrieved from
<https://www.euroinnova.ec/blog/arte-con-arduino#mira-10-ejemplos-de-obra-de-arte-ela-boradas-con-arduino>
- [4] Electric Artefacts. (n.d.). *10 famous new media artists you should know*. Retrieved from
<https://www.electricartefacts.art/news/10-famous-new-media-artists-you-should-know>
- [5] Paik Studios. (n.d.). Retrieved from <https://www.paikstudios.com/>
- [6] Sotheby's. (n.d.). *Mario Klingemann*. Retrieved from
<https://www.sothbys.com/en/artists/mario-klingemann>
- [7] ArchDaily. (n.d.). *Void: Una instalación de arte interactivo que utiliza la luz, el espacio y el movimiento para producir una experiencia sensorial*. Retrieved from
<https://www.archdaily.cl/cl/799590/void-una-instalacion-de-arte-interactivo-que-utiliza-la-luz-el-espacio-y-el-movimiento-para-producir-una-experiencia-sensorial>
- [8] Facebook. (n.d.). *Interactive Art Installation*. [Video]. Retrieved from
<https://www.facebook.com/watch/?v=711896715976856&ref=sharing>
- [9] HeavyM. (n.d.). *Interactive art installation with sensors*. Retrieved from
<https://www.heavym.net/pt/interactive-art-installation-with-sensors/>
- [10] 90+10. (2019, March 11). *El movimiento de los espectadores para crear una obra de arte*. Retrieved from
<https://90mas10.com/2019/03/11/el-movimiento-de-los-espectadores-para-crear-una-obra-de-arte/>
- [11] Scholarpedia. (n.d.). *Swarm intelligence*. Retrieved from
http://www.scholarpedia.org/article/Swarm_intelligence
- [12] Sisinna, G. (n.d.). *How does swarm intelligence work & what is its potential?*. LinkedIn. Retrieved from
<https://www.linkedin.com/pulse/how-does-swarm-intelligence-work-what-its-potential-giovanni-sisinna/>
- [13] FasterCapital. (n.d.). *Inteligencia de enxame: O poder do coletivo - Swarm Intelligence in AI*. Retrieved from
<https://fastercapital.com/pt/contente/Inteligencia-de-enxame--o-poder-do-coletivo--Swarm-Intelligence-in-Ai.html>
- [14] Programar Facil. (n.d.). *Sensor ultrasonico Arduino: Medir distancia*. Retrieved from
<https://programarfácil.com/blog/arduino-blog/sensor-ultrasonico-arduino-medir-distancia/>

[15] Electrónica Made. (n.d.). *Sensor ultrasonico*. Retrieved from
<http://electronicamade.com/sensor-ultrasonico/>

[16] BJUltrasonic. (n.d.). *Ultrasonic transducers and ultrasonic sensors*. Retrieved from
<https://www.bjultrasonic.com/es/ultrasonic-transducers-and-ultrasonic-sensors/>

[17] Amazon. (n.d.). Retrieved from
https://www.amazon.es/dp/B06W9JD4X2/ref=sr_1_1?sr=8-1-spons&sp_csd=d2lkZ2VoTmFtZT1zcF9hdGY&pse=1

[18] Espressif. (n.d.). *ESP32-DevKitC*. Retrieved from
<https://www.espressif.com/en/products/devkits/esp32-devkitc>

[19] Nabto. (n.d.). *Guide to IoT ESP-32*. Retrieved from
<https://www.nabto.com/guide-to-iot-esp-32/>

[20] Electronics Hub. (n.d.). *Getting started with ESP32*. Retrieved from
<https://www.electronicshub.org/getting-started-with-esp32/>

[21] EMQX. (n.d.). *ESP32 connects to the free public MQTT broker*. Retrieved from
<https://www.emqx.com/en/blog/esp32-connects-to-the-free-public-mqtt-broker>

[22] Thingiverse. (n.d.). *KY-037 high sensitivity sound detection module*. Retrieved from
<https://www.thingiverse.com/thing:3776067>

[23] ArduinoModules. (n.d.). *KY-037 high sensitivity sound detection module*. Retrieved from
<https://arduinomodules.info/ky-037-high-sensitivity-sound-detection-module/>