

Documentation of “Hora Académica”

Trabalho feito no âmbito da cadeira:

Tecnologias de Interface

2024/2025



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

Trabalho realizado por:

Francisca Rodrigues de Oliveira |2020218810

Luísa Sá Morgado Brito |2020224165

Índice:

| | |
|--|-----------|
| Índice: | 2 |
| Introdução | 3 |
| Metodologia do design | 3 |
| Implementação | 6 |
| Processing | 6 |
| Arduíno | 10 |
| Integração Arduino e Processing | 12 |
| Explicação do Código final | 12 |
| Local de implantação da instalação | 16 |
| Problemas | 17 |
| Análise de Viabilidade e Desafios | 17 |
| Conclusão | 18 |

Introdução

O nosso projeto é focado no meio da media art, com isto, o nosso objetivo desde o início era criar um objeto media art que contasse alguma história ou trouxesse algum significado consigo.

Após alguns dias de refinamento de ideias, e de alguma ajuda dos nossos docentes, concluímos que gostaríamos de trabalhar no meio acadêmico e a partir daí conseguimos explorar algumas ideias, sendo duas as principais:

- **Parede de post-its com motores:** A ideia aqui seria criar uma parede com diversas placas de metal que estariam ligadas a pequenos motores que seriam responsáveis pelo movimento destas sempre que alguém passasse perto da parede. Nestas placas de metal, estaria colados post-its com anotações de entregas, exames e muito mais tarefas do meio acadêmico, a mensagem que queríamos passar com este trabalho, é de como muitas vezes estes trabalhos e tarefas sufocam os estudantes seguindo os mesmo para todo o lado, mesmo zonas de lazer, e chamando constantemente a sua atenção.

Apesar de gostarmos muito desta ideia, a mesma teve de ser descartada devido a uma complexidade muito elevada para o tempo e materiais que tínhamos em mãos.

- **Hora académica:** Neste projeto, o nosso principal objetivo é mostrar os movimentos ao longo de um dia nos edifícios académicos. Esta acabou por ser a ideia que levaríamos até ao fim e por isso será mais desenvolvida ao longo deste documento.

Acabamos por selecionar a última ideia, uma vez que era mais viável do ponto de vista de implementação, e a sua simplicidade permitia-nos adicionar mais algumas camadas de complexidade posteriormente.

Metodologia do design

Como dito anteriormente, o nosso projeto final acabaria por ser “Hora Académica”. Falando mais detalhadamente sobre o mesmo, o nosso objetivo é mostrar os movimentos no meio académico, mais particularmente entre as salas de aula e as áreas de lazer (com o bar) e as horas a que estes fluxos acontecem.

Para que este fluxo pudesse ser evidenciado ficou claro desde o princípio que esta instalação devia estar num corredor ou local de passagem entre zonas de lazer e trabalho, neste caso bar e salas de aula.

Para que os fluxos sejam representados, começamos por definir uma forma de fazer esta representação. Para que as pessoas se interessassem pelo o projeto achamos importante que este possuísse alguma componente visual apelativa, e que demonstrasse já de alguma forma informação. Para isso pensamos em utilizar uma câmara para captar a silhueta das pessoas que passam, e colorir a mesma com uma cor a depender do sentido do movimento e da hora de passagem ([Figura 1](#)), representando as mesmas num ecrã, evidentemente estas vão se sobrepondo ao longo do dia. Para que fosse possível detectar o sentido do movimento das pessoas, precisamos de dois sensores de proximidade que dão sinais sobre a ordem segundo a qual dispararam. Concluimos também que uma parede lisa, de preferência branca ou preta para que contraste o máximo possível com pessoas, deve também ser o mais simples possível já que texturas e fundos com objetos interferem muito com a captação das silhuetas.

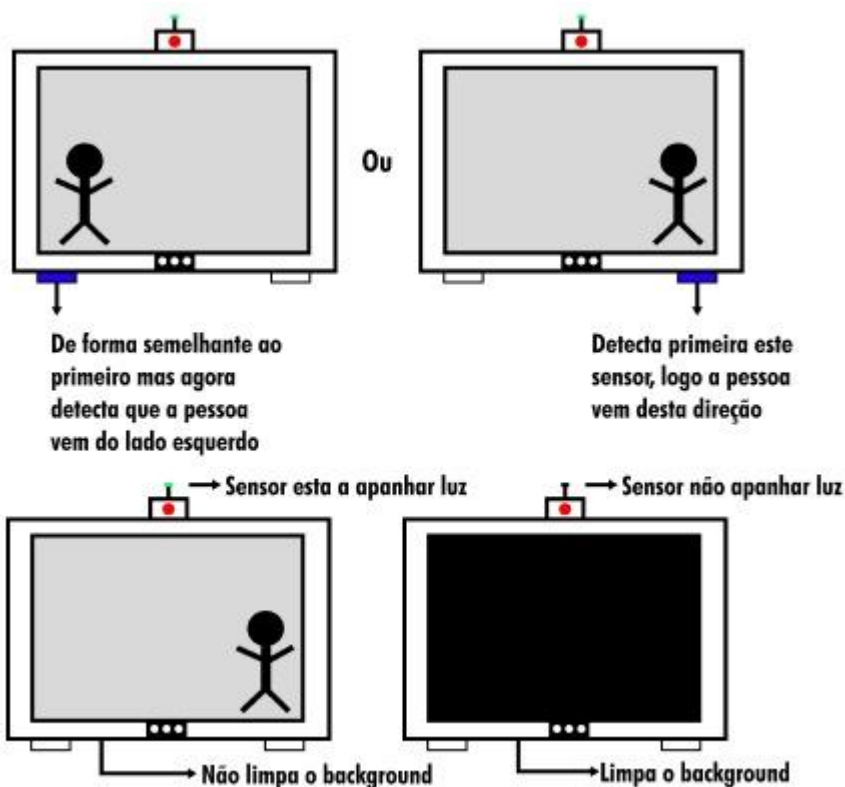


Figura 1.

Optamos por dar ao projeto um sensor de luz que lhe permite desativar-se quando já não existem motivos para estar ligado, neste caso à noite quando o departamento já se encontra vazio. Este sensor também é responsável por a certo ponto resetar o background que regressa ao estado inicial (limpo).

Para acrescentar feedback da parte da instalação foi também colocado um LED que acende para indicar que o primeiro sensor disparou e que o segundo está à espera de ser ativado.

Por fim, queremos que as pessoas consigam ter uma visão mais clara dos dados recolhidos ao longo do dia, uma vez que apenas as cores e as silhuetas não seriam o suficiente para ter uma visão geral, já que vão inevitavelmente sobrepor-se umas às outras. Assim sendo, acrescentamos três botões à interface, que permitem ao utilizador aceder a um outro modo do programa que permite visualizar a informação através de um gráfico de barras e da opção de navegar pelos diversos períodos do dia e ver que silhuetas aparecem nesse período.

Concluimos também que, idealmente, a zona da instalação deve estar bem iluminada com uma luz branca para que as captações sejam o mais limpas possível.

A lista de Materiais com que terminamos foi a seguinte:

- Um ecrã
- Uma câmara
- Dois sensores de movimento
- Uma parede lisa
- Um sensor fotossensível
- Três botões
- Luz Branca

A instalação pode ser vista na [Figura 2](#).

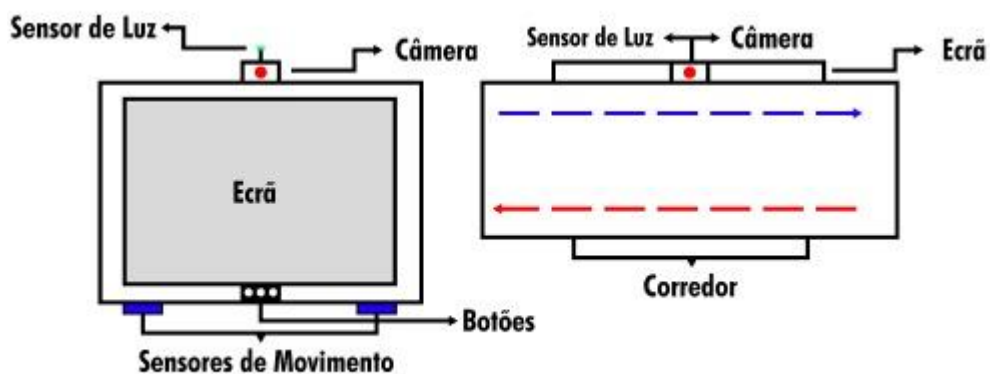


Figura 2.

Partindo deste princípio, podemos começar a implementação, adotamos para isso um sistema de prioridades na ordem pela qual as coisas foram implementadas. O mais importante, sendo a base do projeto, era estabelecer a comunicação entre o Arduino e o processing e fazer as representações das silhuetas de forma correta de acordo com o sentido do movimento, tendo sido esta a parte da implementação que exigiu maior trabalho. Apenas mais tarde implementamos o sensor de luz e o LED. E por fim foram feitos ajustes na representação das silhuetas e implementada a parte da visualização de dados.

Implementação

A nossa implementação foi com bases tanto em processing quanto em arduino, sem contar é claro o circuito arduino que também usamos para implementar os sensores e os outros outputs do programa.

Processing

No que toca ao processing, esta parte influencia maioritariamente a componente visual do nosso projeto, ou seja aquilo que aparece no ecrã e será visto pelas pessoas.

Para alcançar os objetivos estéticos do projeto, precisamos de captar imagens com a câmara e separar as silhuetas das pessoas que passavam ao longo do dia do background, permitindo a representação da informação pretendida.

Numa parte mais inicial do programa, concentramo-nos em conseguir obter a silhueta de uma pessoa a ser captada pela câmara, neste caso a branco sobre um fundo preto (Figura 3).

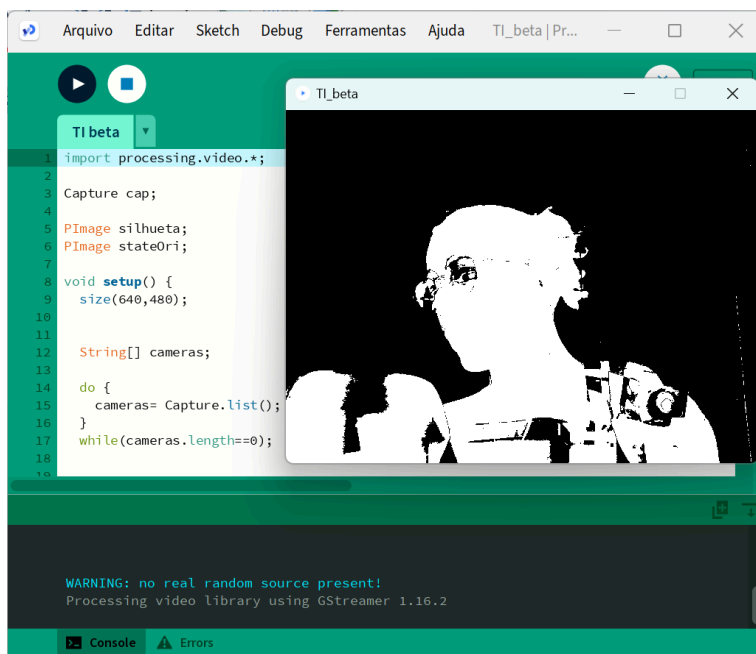


Figura 3.

Terminando esta fase, começamos a trabalhar na ideia de gravar alguns frames destes momentos e representá-los permanentemente no ecrã. Nestas versões já está a ser criada uma imagem da silhueta com um fundo transparente, de forma a poder sobrepor as várias imagens, que estão por questões de teste a ser geradas em intervalos de 2 segundos.

Nestas versões o background não era limpo nenhuma vez e a cor é escolhida de forma aleatória(Figura 4). Na Figura 5, foram feitos testes com a cor, aplicando transparência à silhueta e a cor agora varia de forma regular ao longo de tempo (com o incremento do valor da Matiz da cor a cada silhueta desenhada) de forma a começar a simular melhor os resultados pretendidos na versão final.

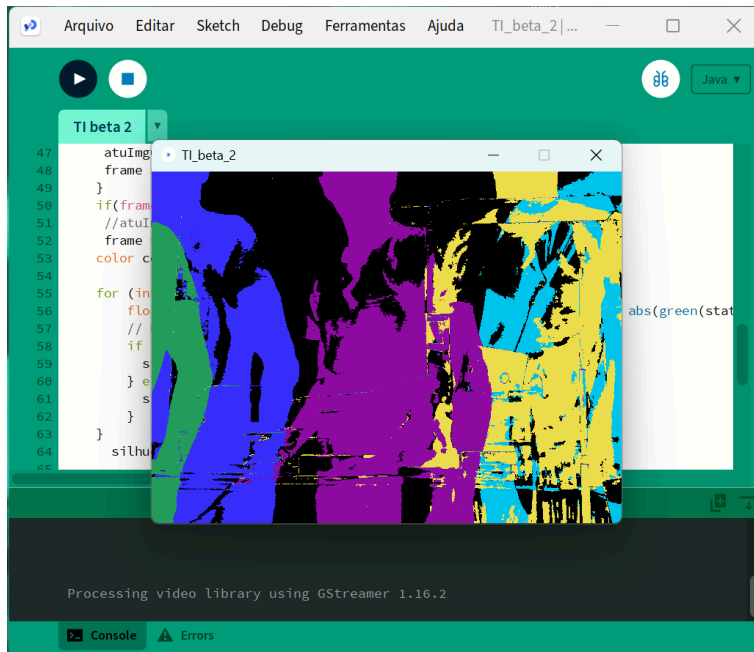


Figura 4.

Após a fase de exploração inicial, começamos a pensar melhor sobre que estratégia poderíamos adotar para o programa começar a funcionar como de fato é pretendido. Para isso o programa em processing tinha que:

- Receber uma instrução, esquerda ou direita, e com base nisso fazer representações diferentes;
- Fazer variar o valor da matiz da cor de acordo com o horário.
- O background tinha que ser continuamente limpo, uma vez que pretendemos não só que uma silhueta num dado instante fique gravada no ecrã como uma outra silhueta a branco seja desenhada continuamente para representar a posição atual do utilizador.

Para que isto fosse executável sem que o programa ficasse demasiado pesado a solução encontrada, a este ponto, foi gravar um frame da janela do processing no momento em que o programa recebe uma instrução, esquerda ou direita, imagem essa que é gravada com o nome background e que é a cada frame redesenhada em cima do fundo preto, desenhando desta forma todas as silhuetas anteriores em conjunto com a que está a ser desenhada a branco com a posição da pessoa em tempo real. Um problema desta solução é que, até ao momento, isso não nos permitiu desenharmos as silhuetas com valor de transparência.

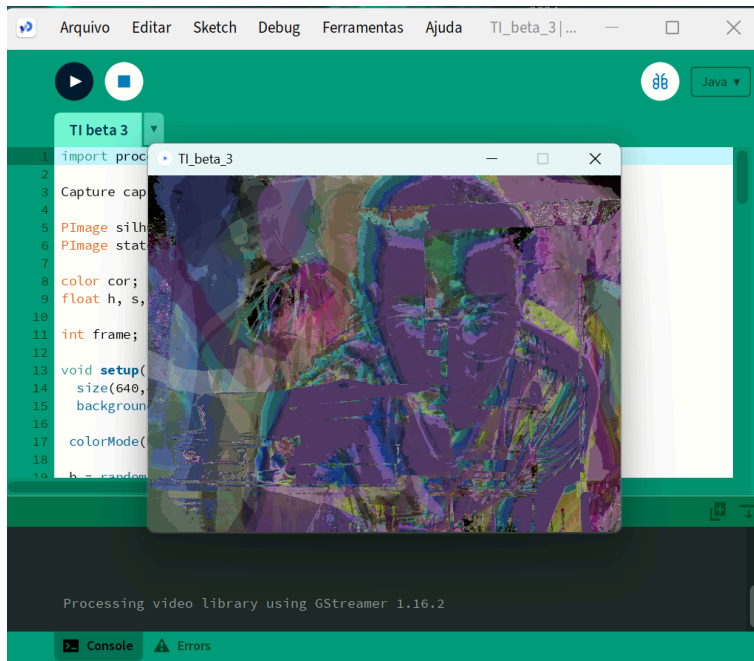


Figura 5.

Os resultados podem ser observados na [Figura 6](#), o sketch utiliza comandos do teclado para simular as instruções (esquerda e direita) que serão futuramente fornecidas pelo arduíno.

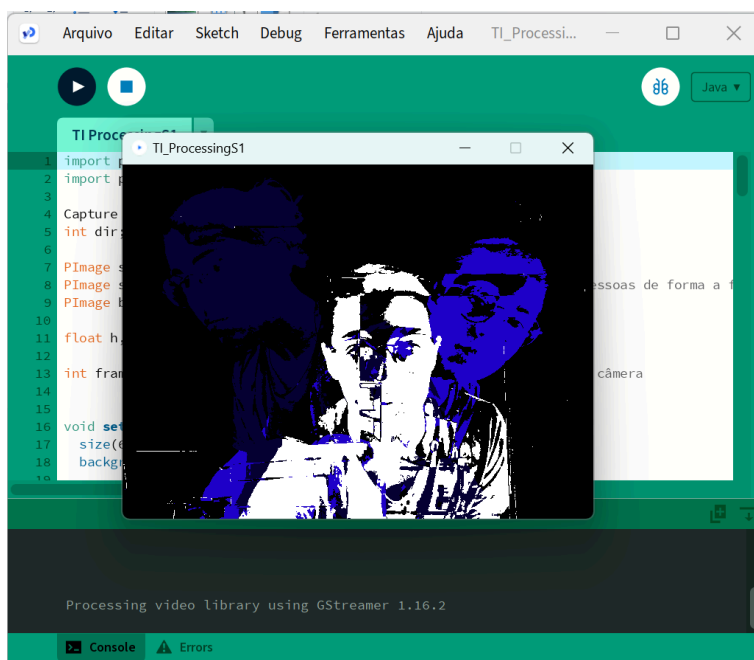


Figura 6.

Por fim existe uma versão levemente diferente desta, cuja a única diferença é que capta a cada frame uma imagem nova para servir de comparação com a atual, contrariamente à versão anterior em que a imagem utilizada como amostra de controlo nunca era atualizada (tencionávamos utilizá-la apenas em alguns momentos específicos de modo a combater as eventuais variações de luz que decorrem ao longo do dia mesmo com a instalação devidamente iluminada). Desta forma o que acontece é que a pessoa só é captada e representada no ecrã caso esteja em movimento (Figura 7). Neste momento pretendíamos prosseguir seguindo a lógica desta versão, mas após uma fase de testagens optamos por regressar à versão em que a silhueta é desenhada de forma preenchida.

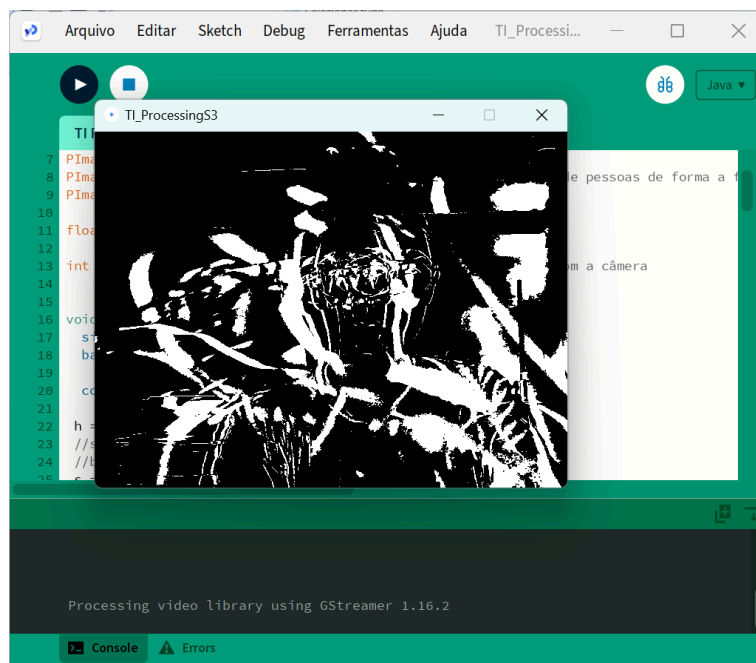


Figura 7.

As silhuetas passaram a ser desenhadas a branco, o motivo ainda se encontra por apurar, mas acreditamos que esteja relacionado com a ordem segundo a qual as instruções executadas no draw e no mousePressed são executadas (este problema é resolvido nas versões posteriores).

Arduíno

O circuito consiste essencialmente de dois sensores de proximidade que pretendemos colocar um ao lado do outro a uma distância que seja suficientes para que uma pessoa

não ative acidentalmente os dois sensores simultaneamente. Além disso, integra também três botões, um sensor de luz e um led.

Programação

A função dos dois sensores de proximidade é também bastante simples, estes devem passar para o programa em Processing uma instrução que indica em que direção a pessoa que passou à frente da instalação se moveu (se foi da direita para a esquerda ou da esquerda para a direita), bem como o momento em que o primeiro sensor dispara e quando desiste de esperar por uma segunda detecção, no caso de um utilizador voltar para trás sem ativar o segundo.

Para este efeito estamos a utilizar uma biblioteca "NewPing.h", esta facilita em questões de código calculando automaticamente as distâncias a que deteta objetos.

Após algumas tentativas chegamos ao resultado pretendido, tendo o arduíno a enviar uma string consoante a pessoa se moveu da direita para a esquerda("DE") ou da esquerda para a direita ("ED"), uma outra instrução ("detecao") assim que o primeiro sensor é ativado e outra ("cancel") caso o tempo de esperar por uma segunda detecção expire.

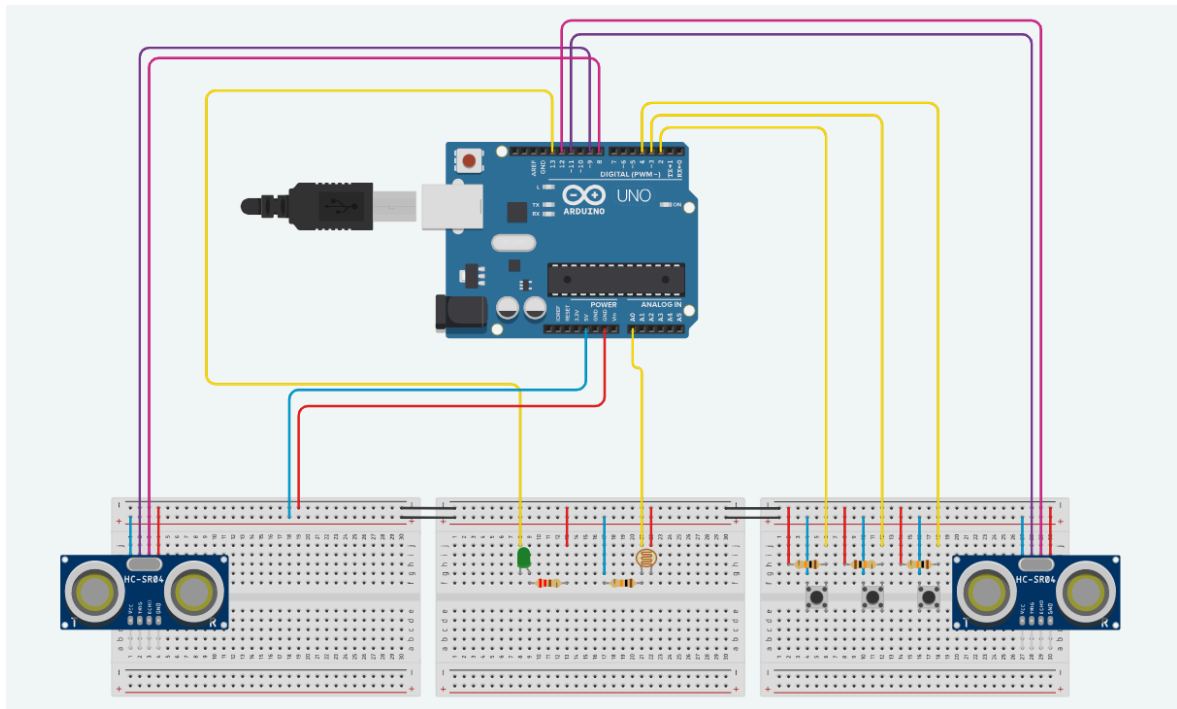
Além disso, o circuito também conta com um LED, que acende quando o programa já detetou movimento num dos sensores e está a aguardar por uma detecção no segundo.

Integrado, tem também um sensor de fotorresistência que detecta quando já é de noite e as luzes estão desligadas, retornando o programa em processing ao estado inicial (sem silhuetas representadas no background).

Lista de Componentes

- 1 Arduíno Uno R3
- 2 Sensores de distância ultrassónicos (4 pinos)
- 1 LED
- 1 Fotorresistor
- 1 Resistencia de 0.22 kiloohms
- 4 Resistência de 10 kiloohms
- 3 Botões

Montagem



Integração Arduino e Processing

A última etapa da realização deste protótipo foi integrar o programa escrito em processing com o arduino, integramos então a biblioteca “processing.serial” do lado do processing que lhe permite aceder à porta série.

Uma vez concluído este passo, os comandos que antes eram dados através do teclado passaram a chegar em forma de strings que chegam a partir da porta série segundo os inputs recebidos pelos sensores do arduino.

Explicação do Código final

A parte do arduino é simples, ele tem dois sensores de proximidade, um é o da esquerda e outro o da direita. Basicamente o que o programa faz é esperar que os sensores

detectem a proximidade de alguém abaixo de um determinado valor (que será menor do que a distância até à parede no local onde a instalação estiver montada). Idealmente estes vão detectar a presença de alguém de forma não simultânea. Além disso, quando um dos sensores detecta a presença de alguém, um led é aceso enquanto o programa espera pela passagem da pessoa no outro sensor.

Do lado do processing o que está a acontecer é que ele liga a câmara, e a partir desse momento começa a comparar as frames que são capturadas pela câmara com uma imagem de amostra que recolheu inicialmente e sempre que detectar que existem silhuetas estáticas a serem representadas constantemente. Ao mesmo tempo é criada uma imagem, “silhueta”. Quando a câmara detecta uma diferença significativa na cor dos pixels correspondentes entre a frame atual e a amostra, esses pixels são pintados a branco na “silhueta” cujo fundo é sempre transparente. Em todos os frames é desenhado um background a preto, depois uma imagem que é constituída pela sobreposição das silhuetas que foram detectadas até agora e por fim o desenho da silhueta atual a branco (caso alguém esteja presente), os resultados podem ser vistos na [Figura 8](#). No início do programa é também criada uma pasta cujo nome é igual à data atual na qual serão guardados resultados do programa.



Figura 8.

Quando o arduíno detecta alguém num dos seus sensores ele entra num estado em que espera a passagem de alguém no segundo sensor e envia uma mensagem para o processing que indica que houve uma detecção no primeiro sensor. Se alguém atravessar o segundo sensor ele especula a direção em que a pessoa se moveu (se o primeiro sensor a ser ativado for o da direita e o segundo o da esquerda, ele conclui que alguém se moveu da direita para a esquerda). Assim sendo, ele envia essa informação para a porta série na forma de uma string. Caso passem 5 segundos e o segundo sensor não for ativado o programa regressa ao estado inicial em que espera a detecção em qualquer um dos sensores e envia uma mensagem para a porta série a indicar que já não se espera uma segunda detecção.

Quando o Processing recebe uma String a indicar que um dos sensores detectou movimento, ele começa a gravar um array de imagens das silhuetas que aparecem nesse momento no ecrã, caso acabe por receber uma String a dizer “DE” ou “ED” ele escolhe aleatoriamente uma das imagens de silhueta que captou para a representação desse movimento (se modo a evitar que existissem silhuetas apenas nas extremidades do ecrã onde se encontram os sensores). A representação terá uma cor cujo valor da Matiz vai variar consoante o horário, consoante a [Figura 9](#), e cujo valor de brilho vai variar de acordo com informação que recebeu sobre o sentido do movimento da pessoa (os valores são 20 para uma direção e 80 para a outra) e a saturação máxima. Esta imagem da silhueta é gravada dentro da pasta, cujo nome corresponde à data atual, com o nome correspondente ao horário atual (em segundos). Além disso, a imagem do background é atualizada para incluir a última silhueta captada. Porém, caso a silhueta possua um número de pixels alterados superior a 50% da imagem, ele considera que a imagem está demasiado poluída, podendo haver algum problema com a imagem de amostra atual, e não capta nenhuma nova imagem.

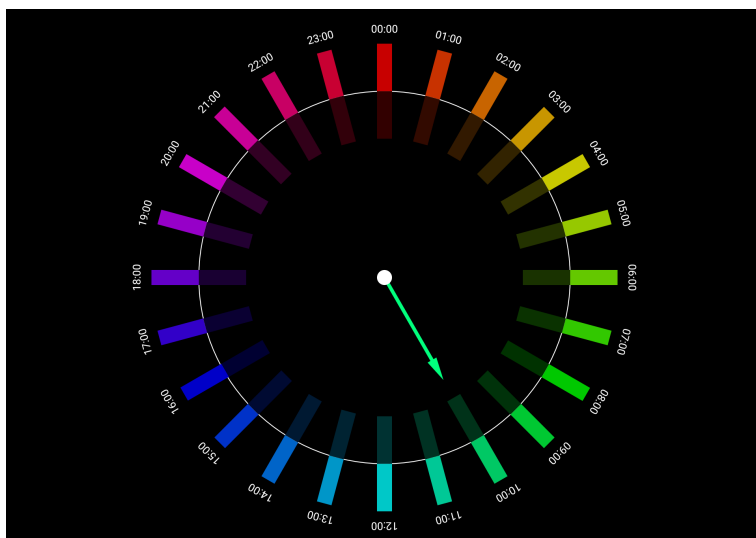


Figura 9.

O programa inclui ainda um modo de visualização da informação recolhida (Figura 10), neste caso, o número de saídas e entradas dentro de intervalos de 15 minutos e as silhuetas desenhadas nesse mesmo intervalo. Quando o processing recebe uma string (“BM”), ele alterna para este modo, carrega todos os ficheiros em formato de imagem e enquadra-os nos respectivos intervalos de tempo (de 15 minutos) a que pertencem. Para isto foi criada uma classe “dados” que contém as funções para desenhar as barras do gráfico e o conjunto de silhuetas correspondente bem como avaliar os dados quanto ao número de saídas e entradas. Depois ele espera que o utilizador pressione um dos outros dois botões (“BE” e “BD”), caso o utilizador pressione o esquerdo ele seleciona o primeiro intervalo de tempo em que houveram entradas ou saídas durante aquele dia caso seja o esquerdo, ele seleciona o último. A partir deste ponto os dois botões servem para navegar entre os intervalos de dados que estejam preenchidos com algum conteúdo, permitindo ver todas as imagens recolhidas nesse dia. O programa permanece neste modo até que o botão do centro seja novamente premido, momento em que o programa volta ao modo original em que nova informação é captada.

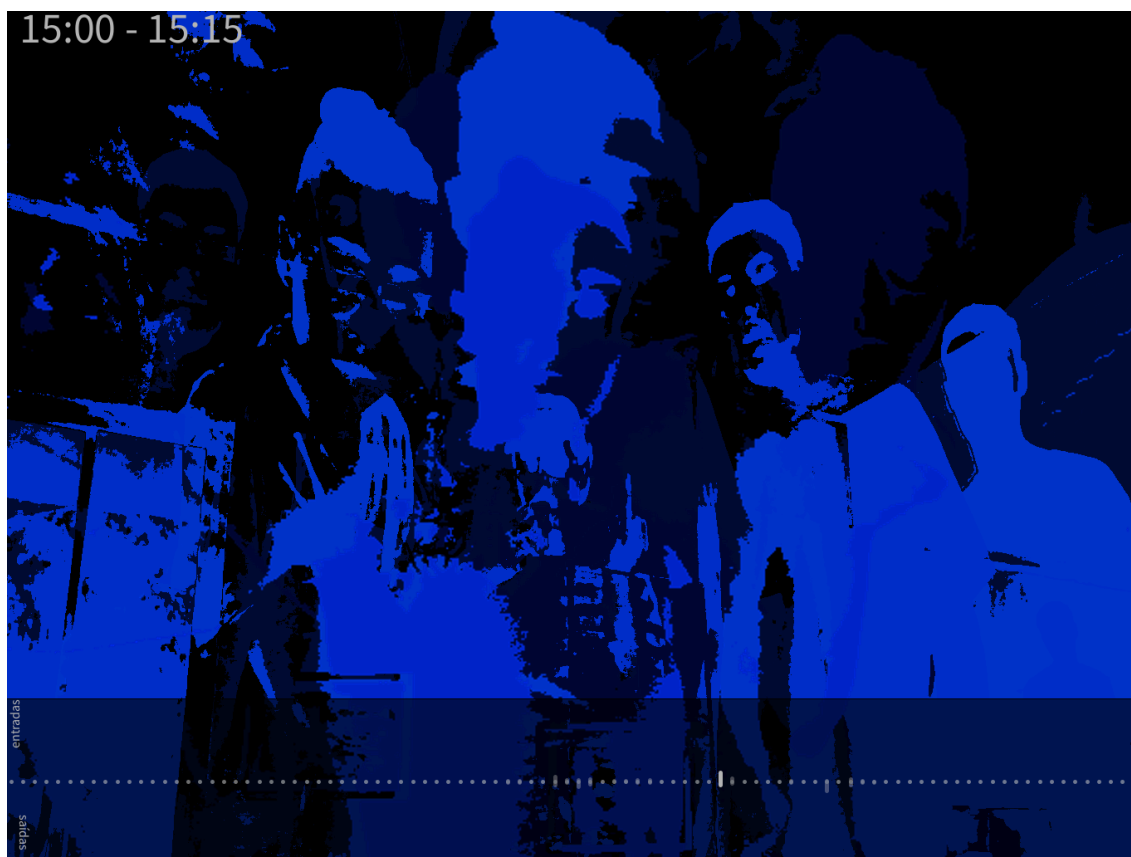


Figura 10.

No final do dia, consideramos que o ambiente da instalação fique escuro e quando isto acontece, um sensor de luminosidade detecta que está escuro e a imagem do background é limpa fazendo com que o programa regresse ao estado inicial, e permaneça em modo repouso até que luz seja novamente detectada.

Para manter a imagem de amostra convenientemente atualizada combatendo as alterações da iluminação, foi ainda adicionado uma função que detecta quando existe uma quantidade de pixels superior a 5% dos pixels da captura que estão estáticos, isto equivale a ter uma silhueta permanentemente alterada e estática na imagem, é considerado que o background se modificou, sendo então capturada uma nova imagem de amostra.

Local de implantação da instalação

O local onde será instalado deve cumprir alguns requisitos para que a instalação funcione devidamente. Idealmente, o objeto estaria instalado numa parede branca com boas condições de iluminação, parte que não está necessariamente sob o nosso controle, por isso poderá eventualmente dar alguns problemas, como:

- Má leitura da câmara;
- Pouco espaço entre o objeto e as pessoas;
- Os resultados não vão ser coerentes caso passem pessoas em direções diferentes ao mesmo tempo.

Além disso, para que o sistema de reset do projeto funcione, consideramos que este espaço fique escuro durante a noite.

Problemas

Alguns problemas ficaram por resolver na versão final. Um dos problemas para o qual não foi encontrada nenhuma solução foi a questão de que os dados não serão totalmente fidedignos uma vez que se 2 pessoas passarem ao mesmo tempo em frente à instalação ficaram registados como tendo ido apenas em um dos sentidos.

No mesmo estilo do problema anterior, para que os dados fossem completamente fidedignos apenas poderia passar uma pessoa de cada vez em frente à exposição, o que não é algo que esperamos que aconteça, em vez disso vamos assumir que grupos de pessoas contaram como apenas uma saída ou entrada.

O último problema de que vamos falar aqui diz respeito à optimização. Infelizmente o modo como estamos a guardar e a analisar os dados, atualmente, torna-se rapidamente muito pesado para o programa e esse problema só irá agravar quanto maior for o número de dados recolhidos, tornando inviável o sistema atual de visualização de dados na instalação.

Análise de Viabilidade e Desafios

Pensamos que o nosso projeto seja bastante apelativo, apesar de toda a sua simplicidade.

Relembrando alguns projetos do gênero que já vimos ao longo da nossa vida, inclusive no departamento de engenharia informática, como no Feed and Play, conseguimos ver que, regra geral, as pessoas tendem a ficar bastante interessadas em projetos interativos mesmo que estes estejam presentes de forma simples no seu dia a dia.

O simples facto de as pessoas estarem a passar por algum lugar e verem de relance algo a acontecer perto de si, traz curiosidade e faz com as mesmas procurem se aproximar deste objeto e interagir mais com este objeto.

Fora do campo da implementação, pensamos que o projeto se mostra bem simples e sem grandes desafios para interação. A nossa principal interação com as pessoas é feita de forma quase inconsciente, enquanto cada uma faz a sua rotina normal, não existe uma seleção de tarefas as quais o nosso utilizador tem de realizar para que a interação seja completa, desta forma a metodologia de design e interação não se aplica especificamente ao nosso projeto.

O nosso maior desafio, no entanto, pode estar em que teremos uma certa dificuldade em controlar, quais seriam as condições de luminosidade do local onde estará a nossa instalação. Como dito anteriormente, para resolver este problema pensamos em recorrer a iluminação controlada para estabilizar estas condições.

Conclusão

Concluindo, com o nosso projeto queremos mostrar de forma visual como é uma rotina nos ambientes académicos e em quais alturas esta se mostra mais vincada.

Criamos um objeto agradável para aqueles que frequentam este espaço diariamente e criar um ponto de encontro que dê um certo alívio ao lugar.

As nossas principais preocupações com este trabalho foram criar uma codificação da informação que pretendemos representar e em garantir uma boa localização para a instalação do nosso objeto. Sendo que ficaram por resolver alguns problemas, nomeadamente a optimização do programa.

O ponto fulcral deste objeto é ilustrar de forma visualmente interessante os fluxos de movimento no departamento de engenharia informática ao longo de um dia.