

## *Trabalho Prático – “Ardugotchi”*

Trabalho realizado por:

Henrique Diz, Turma PL3

Bernardo Mateus, Turma PL3

No âmbito da Unidade Curricular Tecnologia da Informática

Docente: Tiago Cruz

### Índice

Introdução.....	1
Implementação das Funcionalidades.....	1
Conclusão .....	3

### Introdução

Neste relatório, descrevemos a implementação do projeto Arduino “Ardugotchi”.

Neste projeto recriamos o brinquedo ‘Tamagotchi’ usando o Arduino Uno. O Ardugotchi, tal como o Tamagotchi original, é como um animal de estimação, e como tal, o Ardugotchi tem certas necessidades, sendo elas, comer, brincar e dormir. Tal como o brinquedo original, o Ardugotchi também tem saúde: se os seus pedidos não forem atendidos a tempo, este pode ter o infortúnio de falecer.

Com este relatório documentamos e descrevemos as diversas funcionalidades do projeto, bem como estas foram implementadas.

Como exemplificação da implementação do projeto, foi realizado um vídeo em que os intervalos foram todos divididos por 3. A versão do código usada para o vídeo também foi enviada.

### Implementação das Funcionalidades

#### Lógica para Comer e Brincar

Abordemos primeiro a atividade de Comer:

Quando o Ardugotchi tem fome, o LED verde, que está ligado ao PIN 11, acende. O Ardugotchi tem fome de 4 minutos em 4 minutos, no entanto apesar da primeira vez que o Ardugotchi tem

fome ser aos 4 minutos, ele apenas voltará a ter fome 4 minutos depois de o botão para satisfazer esta necessidade, ser premido.

A primeira tentativa de resolução deste problema passou por fazer a divisão de resto inteiro do tempo por 4 minutos ('%') quando esta fosse igual a 0. No entanto, o sucedido era que o Ardugotchi pedia apenas para comer quando a função millis() era divisível por 4 minutos, ou seja, o Led acendia aos 4, 8, 12, 16 minutos e assim sucessivamente.

Assim, a maneira optada para resolver este problema foi a criação de um Array chamado 'Ultimo' de comprimento 5 (Comer, Brincar, Dormir, LDR e Saúde, respetivamente). Este Array fica encarregue de guardar a última vez que o Ardugotchi faz uma atividade. Agora, o Led da necessidade de comer é aceso, quando o tempo do jogo ('millis()') menos a última vez que o Ardugotchi comeu (guardado na variável 'Ultimo[0]') for maior que 4 minutos ('Intervalos[0]').

No final, devido à utilização de Arrays em que a ordem destes era a mesma ('Ultimo[0]' e 'Intervalos[0]' ambos correspondem a tempos associados à necessidade de comer), foi possível criar uma função chamada 'botao' para os botões de comer e brincar, uma vez que elas fazem exatamente a mesma coisa, o que muda apenas, são os intervalos associados a cada uma.

#### Lógica para Dormir

Quando o Ardugotchi tem sono, o LED vermelho, que está ligado ao PIN 12, acende. Para o Ardugotchi ter vontade de dormir, uma de 2 condições tem de ser atingida: ou a média das leituras do LDR é maior que 2/3 da gama do LDR ou a cada 10 minutos acordado. Se uma destas condições for atingida o Ardugotchi vai dormir por 5 minutos.

Para implementar a média das últimas medições do LDR, foi utilizada e adaptada uma ferramenta da referência do Arduino chamada 'Smoothing'. Este código da referência do Arduino calcula a média constante do LDR a cada Loop, substituindo a amostra mais velha por um mais recente a cada minuto. Como nós apenas queremos a média a cada minuto, usamos a mesma lógica usada para ativar o led: usando a variável 'Ultimo[3]' guardamos a última vez que fizemos a média e quando o tempo atual menos essa variável for maior ou igual ao intervalo fazemos a média nesse momento além de ser adicionado um minuto à variável 'Ultimo[3]'.

Para pôr o Ardugotchi a dormir foi implementado um 'while', e que quando este é ativado a função 'desliga\_leds' é chamada. A cada pressionamento do botão é impressa uma mensagem a dizer que vai dormir e atualizamos 'Ultimo[2]' para guardar quando o Ardugotchi começa a dormir. Quando esse tempo mais, o intervalo de dormida a dividir por 2 (ou seja, 5 minutos) for maior que millis() é impressa uma nova mensagem a dizer que acordou e todas as variáveis são atualizadas para o tempo atual ('Ultimo[i] = millis()') recomeçando a rotina como se tivesse acabado de nascer

#### Estado de saúde do Ardugotchi

Antes de podermos usar o Ardugotchi, temos de saber se este está vivo. Para isso é definida a variável 'saude' que inicialmente é 0. Se esta variável for maior que 25 então o Ardugotchi está morto. Desta maneira garantimos o cessar de todas as atividades do programa, quando 'saude' >= 25, porque todas as atividades ligadas ao jogo apenas são executadas dentro do primeiro 'if' (em que 'saude' < 25).

### Logica da soma e subtração de penalizações

As penalizações funcionam somando 5 ou subtraindo 5 à variável 'saude'.

A soma acontece quando o botão é pressionado apenas 1 min depois de alguma das ações ser pedida, ou seja, se o Ardugotchi estiver com fome, e se apenas se carregar no botão para o alimentar 1 min depois de este ter ficado com fome, 5 pontos de penalização serão adicionados à variável 'saude'. Assim, para implementar esta funcionalidade, ela foi escrita fora do pressionamento do botão: se o tempo atual menos a última vez que comeu for maior que o intervalo mais 1 minuto (feito através da variável 'adicionar'), é adicionado 5 a 'saude' e 1 minuto à variável 'adicionar', para garantir apenas a adição de +5 uma vez.

Por outro lado, a subtração acontece se se for rápido a responder à necessidades do Ardugotchi: se o tempo atual menos millis() e menos o intervalo da atividade, for menor ou igual a 15 segundos, então ser-se-á recompensado subtraindo 5 pontos à variável 'saude'. Sendo assim, esta parte é implementada dentro de cada botão, fazendo primeiro este 'if' e só depois atualizamos a variável 'Ultimo' para o tempo atual. Nesta parte é usado um booleano chamado 'deduzido' para garantir que apenas é subtraído, uma vez por cada pressionamento do botão. No final foi criada uma função 'bonificacao' que é chamada quando os botões são premidos.

### Conclusão

Com o decorrer deste projeto, fomos capazes de desenvolver a versão Arduino, do brinquedo infantil "Tamagotchi". Este projeto, abrangeu várias áreas da matéria lecionada nas aulas práticas e teóricas, tais como o uso de 'Debounce' nos botões, a utilização do sensor LDR (bem como fazer a média das amostras deste), a utilização eficaz da função millis(), o uso de LEDs, de funções e principalmente de 'Arrays', que foi crucial para a otimização do código.

Para concluir, podemos afirmar que tomamos este trabalho como bem-sucedido, sendo uma boa oportunidade de aplicar os conhecimentos lecionados.

Disponibilizamos ainda o link do Tinkercad usado para desenvolver o nosso projeto:

<https://www.tinkercad.com/things/4fm505CfAcp-ardugotchi?sharecode=ZKFGoS8sG1BVygvIqWMKOHZJwr38efcF8Tpy41wOU>

Coimbra, 9 dezembro de 2023

Henrique Diz e Bernardo Mateus