

Jogo Genius

Controlado por Voz

Grupo: Bruno Machado Afonso / Felipe Claudio da Silva Santos

Professor: Fernando Gil Vianna Resende Junior

1 – Introdução

Neste trabalho, foi desenvolvido um jogo com mesmo funcionamento de um jogo Genius [1], desenvolvido pela Estrela, onde o objetivo é o jogador memorizar uma sequência de cores exibida pelo brinquedo e assim apertar os botões que representam essas cores em ordem. O objetivo deste trabalho é implementar uma alternativa deste brinquedo usando um microcontrolador e processamento de voz para controle de entrada.

2 – Descrição do Projeto

2.1 – Segmento Arduino – Réplica Genius

O brinquedo é implementado através de um módulo de microcontrolador **Arduino UNO**, que é programado para replicar a lógica em funcionamento eletrônico do brinquedo. Foram usados 3 LEDs (Azul, Branco, Vermelho) que representam as cores que o jogador pode escolher.

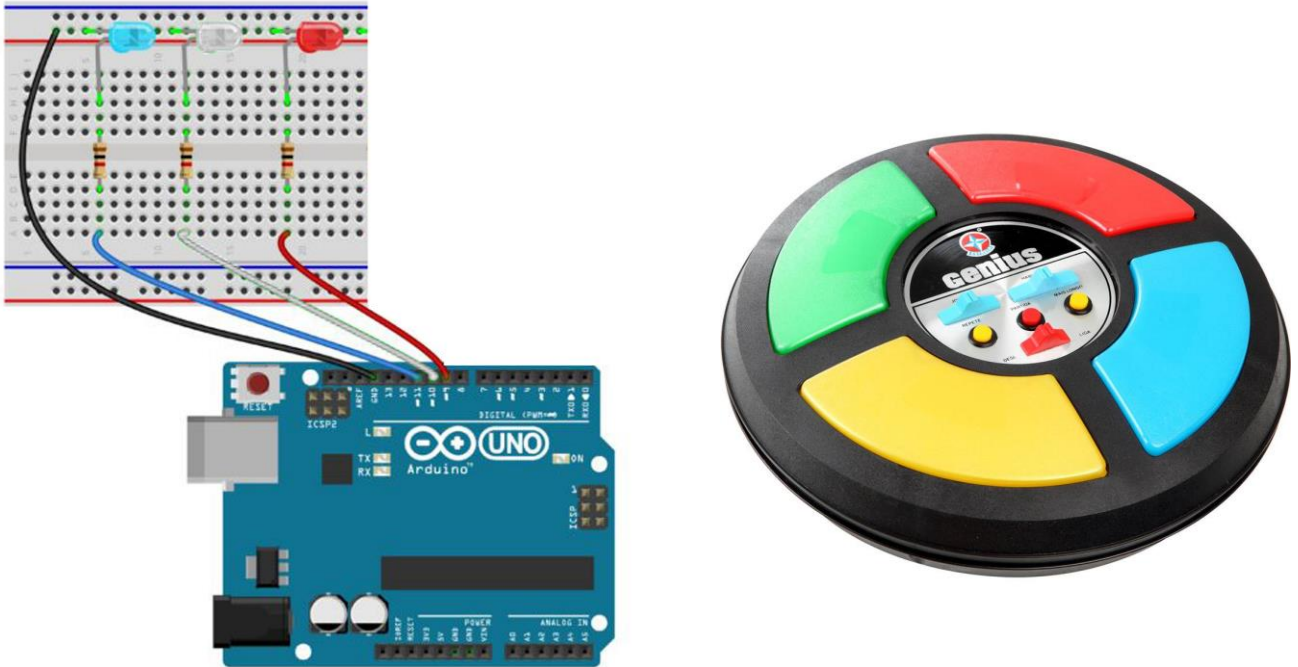


Fig. 1 – À direita, imagem do brinquedo Genius. À esquerda, imagem que representa a estrutura replicada do brinquedo com microcontrolador e LEDs

2.2 – Segmento Processamento de Voz – Controle do Usuário

A entrada de cor escolhida pelo usuário é feita por reconhecimento de voz, realizada através de um computador. O microcontrolador se comunica com o computador através do protocolo serial por uma porta USB. O computador é responsável por:

- Captar a voz do jogador através de um microfone
- Realizar o pré-processamento da voz (Cálculo dos MFCC)
- Calcular a DTW entre o áudio captado e os sons centroides das cores
- Identificar a menor distância e determinar qual a cor foi escolhida
- Enviar via Serial um caracter que identifica a cor para o Arduino

3 – Materiais utilizados

- Microcontrolador ATMEL ATmega328pu + Módulo Arduino UNO R3 – R\$ 36,00
- 3 LEDs – R\$1,00
- 3 Resistores 330Ω – R\$0,10
- Jumpers – R\$1,10
- Protoboard – R\$ 20

4 – Resultados

Para avaliar a performance do algoritmo de detecção de voz, foi feita uma base de treinamento com 30 arquivos de áudio (.wav, PCM 16bits @ 44.1KHz), com 10 para cada cor, e uma base de teste com 18 arquivos de áudio (mesma configuração), com 6 para cada cor. Cada um dos membros do grupo gravou metade da base de treinamento e teste. Após o cálculo do centroide, foi aferida a acurácia do algoritmo observando a relação entre a resposta obtida pelo algoritmo e a cor original da amostra de teste.

		INPUT		
OUTPUT	%	Branco	Vermelho	Azul
	Branco	100	50	33.3
	Vermelho	0	50	0
	Azul	0	0	66.6

Fig. 2 – Tabela que exhibe a relação entre os resultados obtidos do algoritmo sobre a base de teste.

De acordo com a tabela acima, o algoritmo foi bem sucedido para detectar a cor BRANCA, contudo ocorreu uma tendência do algoritmo a classificar a maioria das amostras nesta cor. Os resultados pressupõem que o silêncio entre as sílabas das amostras podem fazer com que a DTW identifique uma proximidade de um segmento de 20ms do som com o possível centroide da cor BRANCA. Isto leva ao algoritmo a tender à escolha do branco como a cor mais próxima.

Uma solução proposta é o uso de Modelos Ocultos de Markov, que tratam os segmentos de áudio como estados, selecionando a sequência mais provável para a amostra de teste dentro de um Codebook captado previamente.

5 – Referências

[1] – Jogo Genius, Estrela, 2018, [https://pt.wikipedia.org/wiki/Genius_\(jogo\)](https://pt.wikipedia.org/wiki/Genius_(jogo))

6 – Fontes e Códigos

Todos os códigos do projeto estão disponíveis no seguinte link:

https://github.com/Armero/ProcessamentoVoz_Genius

Dentro deste repositório existem as seguintes pastas:

1) **genius_serial**

Possui o código do jogo utilizado pelo microcontrolador

2) **voice_rec/audio_samples**

Possui a base de dados utilizada no projeto

Os códigos responsáveis por detectar a cor pronunciada são:

1) **voice_rec/dtw_rec_all.py**

Utiliza todas as amostras de treino no cálculo de distância

2) **voice_rec/dtw_rec_centroid.py**

Utiliza somente os centróides nos cálculos de distância