

# Desenvolvimento de um módulo de reconhecimento de voz para a *game engine* Godot



IME-USP

Leonardo Pereira Macedo  
Orientador: Prof. Dr. Marco Dimas Gubitoso

Bacharelado em Ciência da Computação  
Instituto de Matemática e Estatística - Universidade de São Paulo



## Introdução

*Game engines* são *frameworks* voltados especificamente para a criação de jogos, visando a facilitar o desenvolvimento e/ou algumas de suas etapas. Citamos *Unreal Engine*, *Unity* e *Godot* [3] como exemplos.

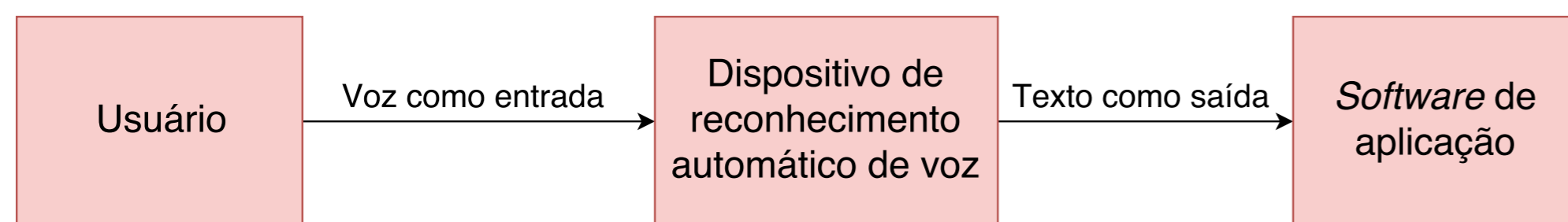
A área de **reconhecimento de voz** obteve avanços significativos desde seus primeiros sistemas na década de 50. Esta tecnologia vem ficando cada vez mais integrada em nosso dia a dia, sendo usada para autenticação de usuário, buscas na Internet, etc.

Este trabalho busca juntar ambos os temas ao desenvolver um módulo (“*plugin*”) de reconhecimento de voz para uma *game engine* em particular, *Godot*.

## Reconhecimento de voz

**Reconhecimento automático de voz** é um campo que desenvolve técnicas para computadores captarem, reconhecerem e traduzirem a linguagem falada para texto; por isso também o nome *speech to text* (STT) [1].

Um sistema genérico STT possui três componentes:



- O **usuário**: Codifica um comando através de sua voz.
- O **dispositivo de STT**: Converte a mensagem falada para um formato interpretável.
- O **software de aplicação**: Recebe a saída do dispositivo e realiza alguma ação.

Os principais termos de reconhecimento de voz incluem:

- **Fluência**: A forma de comunicação com o sistema, podendo ser por palavras isoladas, conectadas ou fala contínua.
- **Dependência do usuário**: Caracterizado pela existência ou não de treinamento feito pelo usuário para melhorar a acurácia do sistema.
- **Vocabulário**: Palavras reconhecidas pelo sistema.

## Pocketsphinx

Buscou-se uma biblioteca de reconhecimento de voz para ser usada no módulo de *Godot*. Dentre as opções encontradas, escolhemos **Pocketsphinx**. Integrante do projeto *CMUSphinx* [2], é escrita em linguagem C e desenvolvida pela *Carnegie Mellon University*.

A ferramenta considera que palavras são formadas por unidades menores chamadas **fonemas**. Usa-se o **Modelo Oculto de Markov** para melhores resultados: considera-se a fala como uma sequência de estados, que transitam entre si com certa probabilidade. Os estados mais prováveis possuem uma melhor interpretação da voz.

Destacamos três componentes de configuração:

- O **modelo acústico**, composto por diversos arquivos que configuram detectores de fonemas.
- O **dicionário fonético**, que mapeia palavras em fonemas. Por exemplo:

yellow Y EH L OW

- O **arquivo de palavras-chave**, que indica quais palavras do dicionário devem ser detectadas, de acordo com um limiar especificado. Por exemplo:

yellow /1e-6/

## Godot

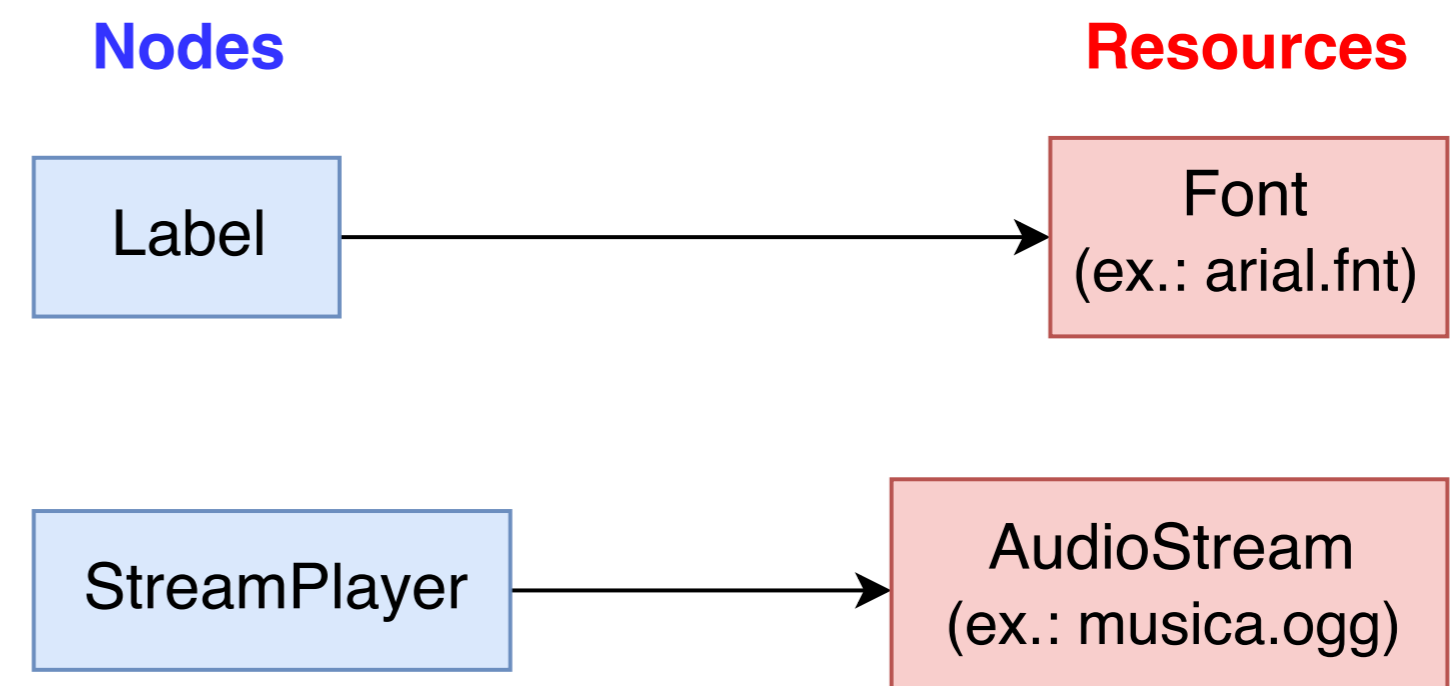


*Godot* [3] é uma *game engine* criada por Juan Linietsky e Ariel Manzur em 2007, e cujo código fonte foi aberto ao público em 2014. Seu código fonte é escrito em **C++**, mas usuários do programa utilizam a linguagem nativa **GScript**, que permite programar com maior facilidade sem se preocupar com detalhes internos de implementação.

Dentre as classes mais importantes de sua arquitetura para este trabalho, destacamos:

- **Object**: Classe base para todos os tipos não embutidos em *Godot*.
- **Reference**: Implementa gerenciamento automático de memória.

- **Resource**: Funciona como um contêiner de dados.
- **Node**: Define um comportamento a ser usado em um jogo.

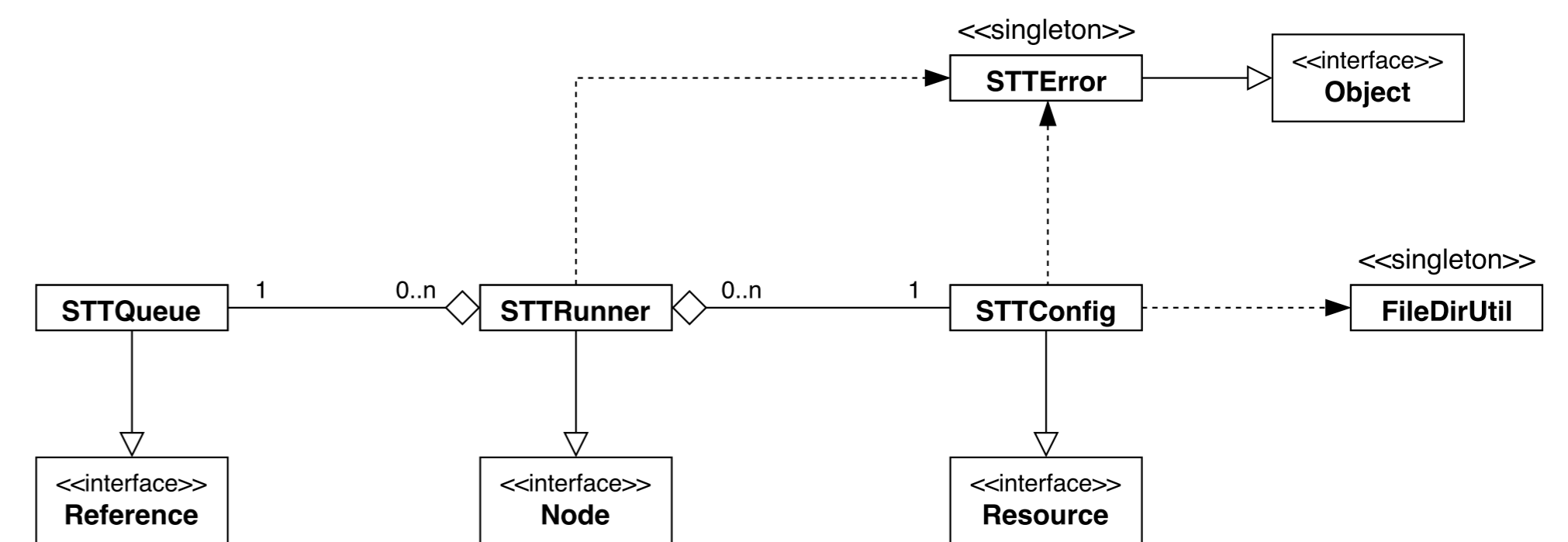


## Módulo Speech to Text

Em relação a reconhecimento de voz, o módulo possui as seguintes características:

- **Fluência**: Palavras conectadas.
- **Dependência do usuário**: Sistema independente.
- **Vocabulário**: Tipicamente pequeno.

Apresentamos a arquitetura do módulo *Speech to Text* [4] a seguir:



As cinco classes implementadas são:

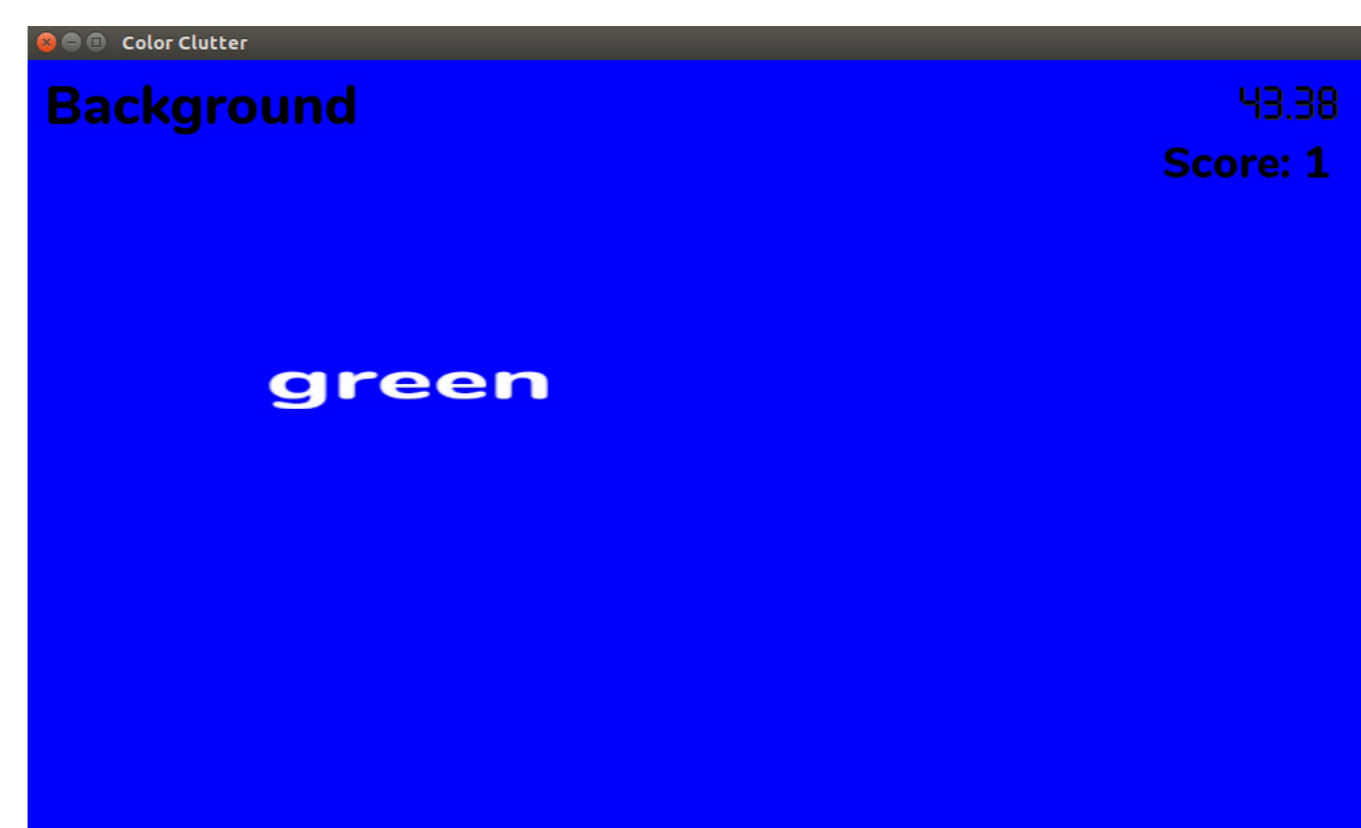
- **STTConfig**: Controla arquivos de configuração de *Pocketsphinx*.
- **STTRunner**: Realiza o reconhecimento de voz em uma *thread*.
- **STTQueue**: Contém uma fila para guardar palavras reconhecidas pelo *STTRunner*.
- **STTError**: Define constantes para possíveis erros nas demais classes.
- **FileDirUtil**: Classe auxiliar para manipular arquivos e diretórios.

## Jogo Color Clutter

Criamos um jogo simples, **Color Clutter**, para demonstrar o uso do módulo *Speech to Text*. Usamos inglês americano pela disponibilidade de arquivos.

Uma típica tela do jogo consiste em um fundo totalmente preenchido com alguma cor X. Em alguma posição da tela, uma outra cor Y aparece escrita em um tom Z. O objetivo do usuário é falar a cor correta (X, Y ou Z), de acordo com o que é solicitado em uma legenda apresentada na tela.

No exemplo a seguir, o usuário deve dizer **blue** para continuar.



## Referências

- [1] *National Research Council. Automatic Speech Recognition in Severe Environments.* The National Academies Press, 1984.
- [2] *CMUSphinx. About the CMUSphinx.* URL: <http://cmusphinx.sourceforge.net/wiki/about>.
- [3] Linietsky, J. e Manzur, A. *Godot Engine.* URL: <https://godotengine.org>.
- [4] Macedo, L. P. *Speech to Text.* URL: <https://github.com/SamuraiSigma/speech-to-text>.