

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

Leonardo Pereira Macedo

Orientador: Prof. Dr. Marco Dimas Gubitoso

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

22 de novembro de 2017

- Evolução e sofisticação de jogos eletrônicos (*games*)
- Surgimento das ***game engines***: *frameworks* voltados para facilitar o desenvolvimento total ou parcial de jogos
 - Exemplos: *Unreal Engine*, *Unity*, *Godot*

- Evolução e sofisticação de jogos eletrônicos (*games*)
- Surgimento das **game engines**: *frameworks* voltados para facilitar o desenvolvimento total ou parcial de jogos
 - Exemplos: *Unreal Engine*, *Unity*, *Godot*
- **Reconhecimento de voz** vem ficando cada vez mais integrado em nosso dia a dia
 - Autenticação de usuário, realização de buscas na Internet, etc.

- Evolução e sofisticação de jogos eletrônicos (*games*)
- Surgimento das ***game engines***: *frameworks* voltados para facilitar o desenvolvimento total ou parcial de jogos
 - Exemplos: *Unreal Engine*, *Unity*, *Godot*
- **Reconhecimento de voz** vem ficando cada vez mais integrado em nosso dia a dia
 - Autenticação de usuário, realização de buscas na Internet, etc.

Por que não fazer um trabalho que junte ambos os temas?

Reconhecimento de Voz

Definição e componentes

Definição

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)

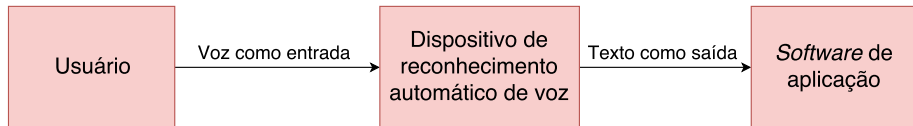
Reconhecimento de Voz

Definição e componentes

Definição

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)

- Um sistema genérico STT possui três componentes:



[National Research Council, 1984]

Reconhecimento de Voz

Principais termos

- **Fluência:** Forma de comunicação com o sistema
 - Palavras isoladas
 - Palavras conectadas
 - Fala contínua

Reconhecimento de Voz

Principais termos

- **Fluência:** Forma de comunicação com o sistema
 - Palavras isoladas
 - Palavras conectadas
 - Fala contínua
- **Dependência do usuário:** Há treinamento?
 - Sistemas dependentes
 - Sistemas independentes

Reconhecimento de Voz

Principais termos

- **Fluência:** Forma de comunicação com o sistema
 - Palavras isoladas
 - Palavras conectadas
 - Fala contínua
- **Dependência do usuário:** Há treinamento?
 - Sistemas dependentes
 - Sistemas independentes
- **Vocabulário:** Palavras reconhecidas pelo sistema

- Desenvolvida pela *Carnegie Mellon University* (projeto *CMUSphinx*)
- É de código aberto e escrita em **C**

- Desenvolvida pela *Carnegie Mellon University* (projeto *CMUSphinx*)
- É de código aberto e escrita em **C**
- Define que palavras são formadas por **fonemas**

- Desenvolvida pela *Carnegie Mellon University* (projeto *CMUSphinx*)
- É de código aberto e escrita em **C**
- Define que palavras são formadas por **fonemas**
- Utiliza o **Modelo Oculto de Markov** na interpretação
 - Trata a fala gravada como uma sequência de estados, que transitam entre si com certa **probabilidade**

- Desenvolvida pela *Carnegie Mellon University* (projeto *CMUSphinx*)
- É de código aberto e escrita em **C**
- Define que palavras são formadas por **fonemas**
- Utiliza o **Modelo Oculto de Markov** na interpretação
 - Trata a fala gravada como uma sequência de estados, que transitam entre si com certa **probabilidade**

Estados mais prováveis → Melhor interpretação da voz

- **Modelo acústico:** Arquivos que configuram detectores de fonemas

- **Modelo acústico:** Arquivos que configuram detectores de fonemas
- **Dicionário fonético:** Mapeamento {palavras → fonemas}

yellow Y EH L OW

- **Modelo acústico:** Arquivos que configuram detectores de fonemas
- **Dicionário fonético:** Mapeamento {palavras → fonemas}

yellow Y EH L OW

- **Palavras-chave:** Palavras a serem detectadas, de acordo com limiar

yellow /1e-6/



[Linietsky and Manzur, 2017]

- Criação de Juan Linietsky e Ariel Manzur em 2007
- Código aberto ao público em 2014



[Linietsky and Manzur, 2017]

- Criação de Juan Linietsky e Ariel Manzur em 2007
- Código aberto ao público em 2014
- Código fonte escrito em **C++**



[Linietsky and Manzur, 2017]

- Criação de Juan Linietsky e Ariel Manzur em 2007
- Código aberto ao público em 2014
- Código fonte escrito em **C++**
- Programação de jogos simplificada por meio de ***GDScript***

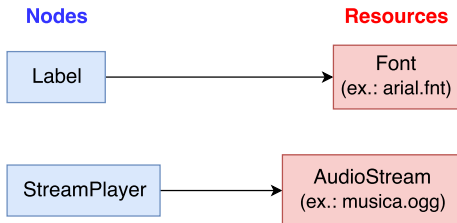
- Object: Classe base para todos os tipos não embutidos em *Godot*

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

- Object: Classe base para todos os tipos não embutidos em *Godot*
- Reference: Gerenciamento automático de memória
- Resource: Armazenamento de dados

- Object: Classe base para todos os tipos não embutidos em *Godot*
- Reference: Gerenciamento automático de memória
- Resource: Armazenamento de dados
- Node: Define um comportamento

- Object: Classe base para todos os tipos não embutidos em *Godot*
- Reference: Gerenciamento automático de memória
- Resource: Armazenamento de dados
- Node: Define um comportamento



[Godot, 2017]

- Em relação a reconhecimento de voz:
 - **Fluência:** Palavras conectadas

- Em relação a reconhecimento de voz:
 - **Fluência:** Palavras conectadas
 - **Dependência do usuário:** Sistema independente

- Em relação a reconhecimento de voz:
 - **Fluência:** Palavras conectadas
 - **Dependência do usuário:** Sistema independente
 - **Vocabulário:** Tipicamente pequeno

Módulo *Speech to Text*

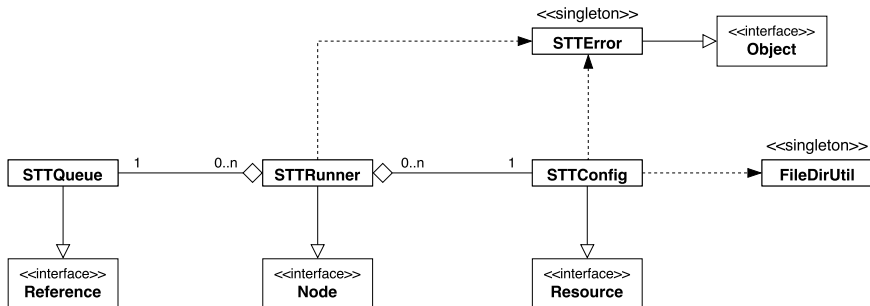
Requisitos

- Em relação a reconhecimento de voz:
 - **Fluência:** Palavras conectadas
 - **Dependência do usuário:** Sistema independente
 - **Vocabulário:** Tipicamente pequeno
- Execução do módulo em **paralelo** com o restante do jogo
 - Uso de uma *thread*

- Em relação a reconhecimento de voz:
 - **Fluência**: Palavras conectadas
 - **Dependência do usuário**: Sistema independente
 - **Vocabulário**: Tipicamente pequeno
- Execução do módulo em **paralelo** com o restante do jogo
 - Uso de uma *thread*
- Uso de um *buffer* para armazenar palavras conforme são reconhecidas

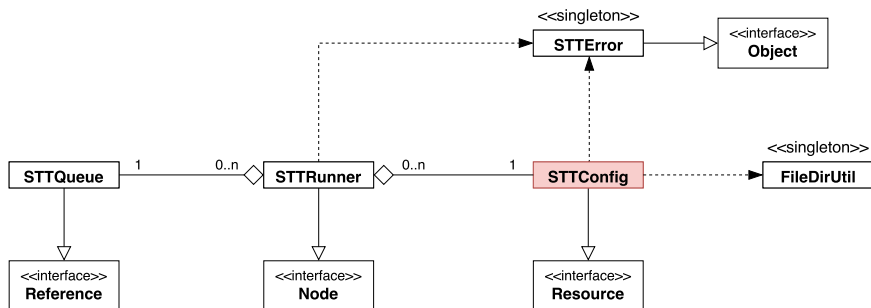
Módulo *Speech to Text*

Implementação



Módulo *Speech to Text*

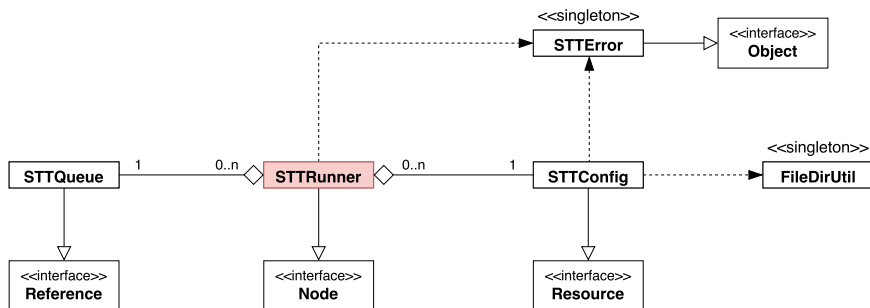
Implementação



- **STTConfig**: Controle dos arquivos de configuração *Pocketsphinx*

Módulo *Speech to Text*

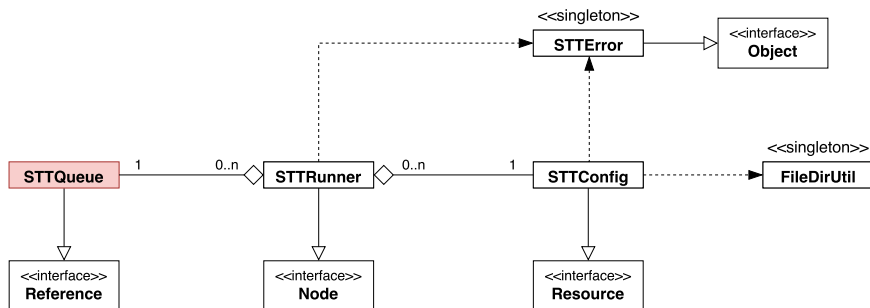
Implementação



- **STTConfig**: Controle dos arquivos de configuração *Pocketsphinx*
- **STTRunner**: *Thread* para realizar reconhecimento de voz

Módulo *Speech to Text*

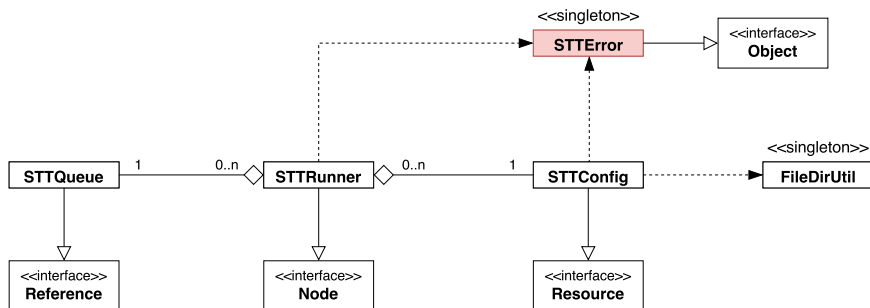
Implementação



- **STTConfig**: Controle dos arquivos de configuração *Pocketsphinx*
- **STTRunner**: *Thread* para realizar reconhecimento de voz
- **STTQueue**: Fila para guardar palavras reconhecidas pelo **STTRunner**

Módulo *Speech to Text*

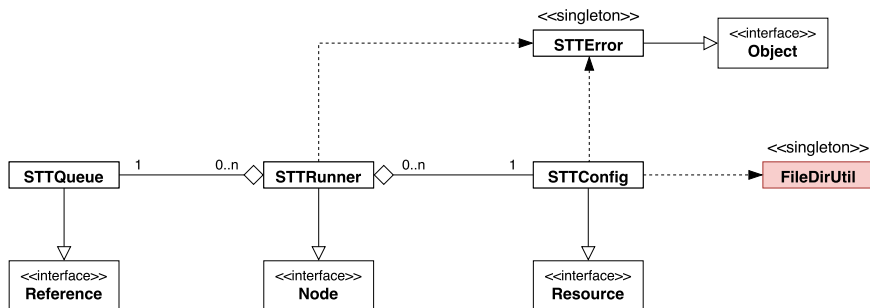
Implementação



- **STTConfig**: Controle dos arquivos de configuração *Pocketsphinx*
- **STTRunner**: *Thread* para realizar reconhecimento de voz
- **STTQueue**: Fila para guardar palavras reconhecidas pelo **STTRunner**
- **STTError**: Definição de constantes numéricas para possíveis erros

Módulo *Speech to Text*

Implementação



- **STTConfig**: Controle dos arquivos de configuração *Pocketsphinx*
- **STTRunner**: *Thread* para realizar reconhecimento de voz
- **STTQueue**: Fila para guardar palavras reconhecidas pelo **STTRunner**
- **STTError**: Definição de constantes numéricas para possíveis erros
- **FileDirUtil**: Classe auxiliar para manipular arquivos e diretórios

Color Clutter

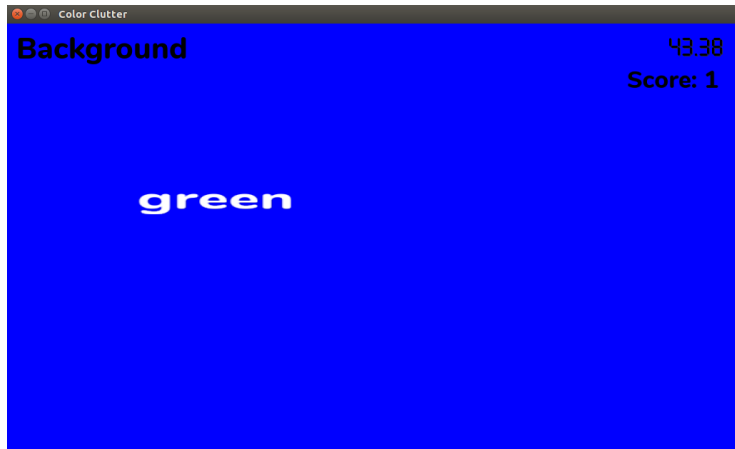
(Jogo demonstrativo)

- Jogo criado em *Godot* para demonstrar o módulo *Speech to Text*
- Todo o controle é feito por voz

Color Clutter

(Jogo demonstrativo)

- Jogo criado em *Godot* para demonstrar o módulo *Speech to Text*
- Todo o controle é feito por voz



Conclusão

Resultados e considerações finais

- Testes realizados em *Color Clutter* com 10 usuários diferentes

Conclusão

Resultados e considerações finais

- Testes realizados em *Color Clutter* com 10 usuários diferentes
 - Rápido, mas rígido quanto à pronúncia de algumas cores

Conclusão

Resultados e considerações finais

- Testes realizados em *Color Clutter* com 10 usuários diferentes
 - Rápido, mas rígido quanto à pronúncia de algumas cores
- *Speech to Text* e *Color Clutter* publicados em dois fóruns de *Godot*
 - Algumas (poucas) aprovações

Conclusão

Resultados e considerações finais

- Testes realizados em *Color Clutter* com 10 usuários diferentes
 - Rápido, mas rígido quanto à pronúncia de algumas cores
- *Speech to Text* e *Color Clutter* publicados em dois fóruns de *Godot*
 - Algumas (poucas) aprovações
- Desejo de continuar o módulo
 - Suporte para outros sistemas operacionais (*Android*, *MacOS*)

Referências I



CMUSphinx (2015).

About the CMUSphinx.

<http://cmusphinx.sourceforge.net/wiki/about>.



Godot (2017).

Godot Docs.

<http://docs.godotengine.org>.



Linietsky, J. and Manzur, A. (2017).

Godot Engine.

<https://godotengine.org>.



Macedo, L. P. (2017a).

Color Clutter.

<https://github.com/SamuraiSigma/color-clutter>.



Macedo, L. P. (2017b).

Speech to Text.

<https://github.com/SamuraiSigma/speech-to-text>.



Macedo, L. P. (2017c).

Speech to Text module for Godot 2.1.3/2.1.4.

<https://godotengine.org/qa/18322/speech-to-text-module-for-godot-2-1-3-2-1-4>.

-  Macedo, L. P. (2017d).
Speech to Text module for Godot 2.1.3/2.1.4.
[https://godotdevelopers.org/forum/discussion/18659/
speech-to-text-module-for-godot-2-1-](https://godotdevelopers.org/forum/discussion/18659/speech-to-text-module-for-godot-2-1-).
-  National Research Council (1984).
Automatic Speech Recognition in Severe Environments.
[The National Academies Press](#).

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

Leonardo Pereira Macedo

Orientador: Prof. Dr. Marco Dimas Gubitoso

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

22 de novembro de 2017