



UNIVERSIDADE
FEDERAL DO CEARÁ

Centro de Ciências
Departamento de Computação
Bacharelado em Ciência da Computação

Genshin Artifacts Optimizer

Cristian Lino Martins de Souza

Técnicas de Programação II
Professor: Lincoln Souza Rocha

Histórico de Versões

Versão	Data	Descrição	Autor(es)
0.1	06.02.2022	Criação/Adequações do Documento	Cristian Lino
0.2	06.02.2022	Criação da Seção 1, Seção 2, Seção 3, Seção 4	Cristian Lino
0.3	06.02.2022	Adequações da Seção 3	Cristian Lino

Sumário

1	Introdução	4
2	Descrição da aplicação	4
3	Desenvolvimento	4
3.1	Node	4
3.2	Jest	4
3.3	Estrutura dos arquivos	5
3.3.1	.github	5
3.3.2	node-modules	5
3.3.3	src	5
3.3.4	.gitignore	6
3.3.5	.package-lock.json	6
3.3.6	.package.json	6
3.3.7	README.md	6
3.3.8	xiao.js	6
4	Referências	7

1 | INTRODUÇÃO

Este documento contém as especificações acerca do desenvolvimento da aplicação Genshin Artifacts Optimizer.

2 | DESCRIÇÃO DA APLICAÇÃO

A aplicação tem o objetivo de informar a melhor combinação de artefatos para um personagem do jogo Genshin Impact tirar o máximo de dano médio possível. Várias variáveis são consideradas, tais como: Time escolhido, Ressonância, Rotação, Buffs de armas/sets, Resistência e level dos inimigos.

3 | DESENVOLVIMENTO

3.1 | Node

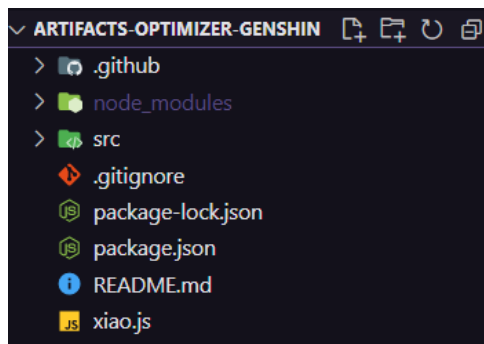
Node.js é um ambiente de servidor de código aberto e utiliza JavaScript. Atualmente no código, o node comporta toda a aplicação.

3.2 | Jest

Jest é um JavaScript Testing Framework com foco na simplicidade. Atualmente no código, o jest é utilizado para realizar os testes unitários de forma automatizada.

3.3 | Estrutura dos arquivos

Como são poucos arquivos e como não existe ainda uma divisão de Back End e Front End, os arquivos não estão um padrão específico, mas com legibilidade.



(a) Estrutura dos arquivos

3.3.1 | .github

Pasta contendo a pasta Workflows, utilizada para configurar o github actions.

3.3.2 | node-modules

Pasta contendo todos os pacotes necessários para a execução do projeto. Ela foi criada automaticamente ao executar o comando “npm install”. Ela é baseada no conteúdo do arquivo “Package.json”.

3.3.3 | src

Pasta principal do projeto. Contém a maior parte do código fonte. Dentro dessa pasta temos outras 3 pastas que são:

- *Artifacts*: Dentro dessa pasta contém apenas um arquivo que guarda todos os dados dos artefatos. Aqui é nosso banco de dados simulado.
- *CalcDamage*: Pasta contendo 2 arquivos, onde o arquivo calc.js contém apenas funções a serem exportadas. O outro arquivo é o calc.test.js, que contém os testes unitários para

cada função do `calc.js`

- *Character*: Pasta contendo 2 arquivos, onde o arquivo `Character.js` contém a classe `Character` que guarda todas as informações necessárias de um personagem. O outro arquivo é o `Character.test.js`, que contém os testes unitários para cada método do `Character.js`

3.3.4 | `.gitignore`

Contém os arquivos que o Git deve ignorar e não enviar alterações feitas localmente.

3.3.5 | `.package-lock.json`

Para obter instalações consistentes em todas as máquinas, o npm precisa de mais informações do que as dependências que você configura em seu `package.json`. O npm precisa armazenar exatamente quais versões de cada dependência foram instaladas. Para fazer isso, o npm usa um `package-lock.json`, que é arquivo na raiz do seu projeto.

3.3.6 | `.package.json`

Arquivos contendo as especificações do projeto como versão, nome, dependências e dependências de desenvolvimento. Além de alguns scripts que podemos utilizar.

3.3.7 | `README.md`

Uma forma de documentação, apenas texto simples com a explicação do projeto no github.

3.3.8 | `xiao.js`

O personagem escolhido para realizar os testes de melhor combinação de artefatos foi o Xiao. Nesse arquivo contém dados específicos do personagem e, caso queira testar outro personagem, será necessário criar um novo arquivo com os dados desse personagem. Esse arquivo é considerado nossa `main.js` ou `app.js`

4 | REFERÊNCIAS

- [1] NODE. Disponível em: <<https://nodejs.org/en/>>. Acesso em: 06 fev. 2022.
- [2] JEST. Disponível em: <<https://jestjs.io/>>. Acesso em: 06 fev. 2022.
- [3] GENSHIN-ARTIFACTS-OPTIMIZER. Disponível em: <<https://github.com/cristian-lino/artifacts-optimizer-genshin>>. Acesso em: 06 fev. 2022.
- [4] GITHUB ACTIONS. Disponível em: <<https://docs.github.com/pt/actions/quickstart>>. Acesso em: 06 fev. 2022.