

Postmortem Randovania

Randomizer de Metroid Prime 2: Echoes

henrique@gemignani.org

Discord: Darkszero#7435

randomizer.metroidprime.run

O que é um Randomizer?

Randomizers são programas que pegam um jogo e criam uma versão modificada dele, onde elementos são aleatorizados.



Jogos como Ocarina of Time e Link to the Past possuem randomizers extremamente populares.

História

Entre 2014 e 2016, uma pessoa chamada Claris fez um randomizer para Metroid Prime 2.

Esse randomizer tinha toda a engenharia reversa necessária para alterar o jogo em si, tendo um resultado jogável.



História

No entanto, esse randomizer apenas aleatorizava todos os itens do jogo sem nenhuma preocupação se o resultado era possível.

Em algum momento, a Claris montou uma base de dados com quais itens são necessários para navegar por todas as salas do jogo, com o objetivo de implementar uma lógica. Infelizmente, ela nunca chegou a implementar tal lógica.

Primeiro ciclo

Em setembro de 2017, com acesso a uma cópia dessa base de dados, comecei a implementação de um verificador.

É uma ferramenta de linha de comando que recebe um log de onde os itens estão, e responde se é possível terminar ou não.

```
C:\Users\henri\Downloads>randovania echoes validate prime2_original_log.txt  
Game is possible!  
Victory with 23% of the items.
```

```
C:\Users\henri\Downloads>randovania echoes validate echoes_log_recursion_heavy.txt  
Impossible.
```

Primeiro ciclo

Qual é a entrada? Qual item está em cada um dos 119 spots de itens do jogo.

Como é a base de dados? Lista de requisitos para cada sala do jogo.

São 5 mundos, com 284 áreas, 945 pontos de interesse e 4527 alternativas.

The screenshot shows a software interface for selecting and viewing game nodes. At the top, there are two dropdown menus for "World/Area Selection", with "Sanctuary Fortress" and "Grand Abyss" selected. Below this, the interface is split into two main sections: "Nodes" and "Node Info".

Nodes: A list of radio buttons allows selecting a specific node. The selected node is "Door to Watch Station". Other options include "Door to Vault", "Event - Grand Abyss (Vault Side)", "Event - Grand Abyss (Watch Station Side)", and "Portal from Judgment Drop".

Node Info: This section displays details for the selected node. It shows the name "Door to Watch Station" with a checked "Heals?" checkbox. Below the name is the description "Normal Door to Watch Station/Door to Grand Abyss".

Connections: A dropdown menu shows the selected connection "Event - Grand Abyss (Vault Side)". Below this, a table lists the requirements for this connection:

I: Space Jump Boots ≥ 1	I: Screw Attack ≥ 1
I: Screw Attack ≥ 1	E: Grand Abyss ≥ 1
E: Grand Abyss ≥ 1	T: Screw Attack without Space Jump ≥ 1
	Difficulty Level ≥ 2

Primeiro ciclo

Plano: identifica quais nós com itens pode chegar, escolhe um para pegar.
Repete.

Se travar em algum lugar (*softlock*), faz backtrack.

Primeiro ciclo

Plano: identifica quais nós com itens pode chegar, escolhe um para pegar.
Repete.

Se travar em algum lugar (*softlock*), faz backtrack.

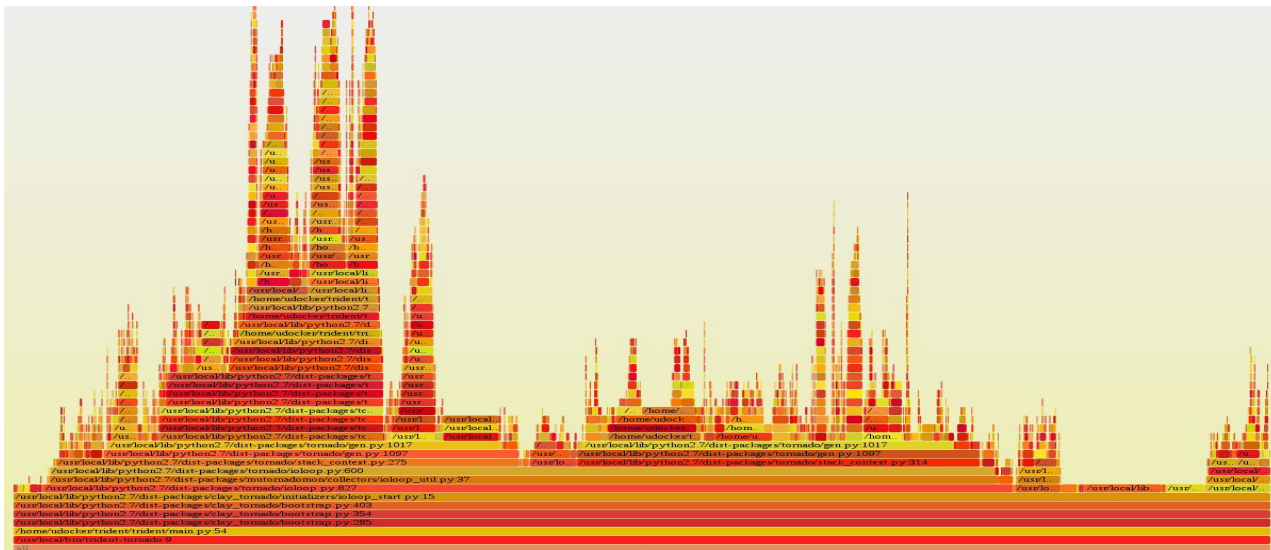
Funciona bem se a resposta é positiva.

Roda para sempre se é impossível.

Primeiro ciclo

É possível otimizar? Sim.

1. Usando profilers Python para identificar uso de CPU (pyflame, py-spy)
2. Cache de resultados



Primeiro ciclo

É possível otimizar? Sim.

1. Usando profilers Python para identificar uso de CPU
2. Cache de resultados

Não foi suficiente. Vamos pensar um pouco mais.

Tem como ganhar informação no backtrack!

Validador agora responde sim/não em menos de 5s.

Segundo ciclo

“Gerador” de seeds possíveis. Funciona através de uma busca exaustiva de seeds: gera seed, verifica que possível. Se não, tenta denovo.

Usando múltiplos processos para usar todos os núcleos da CPU.

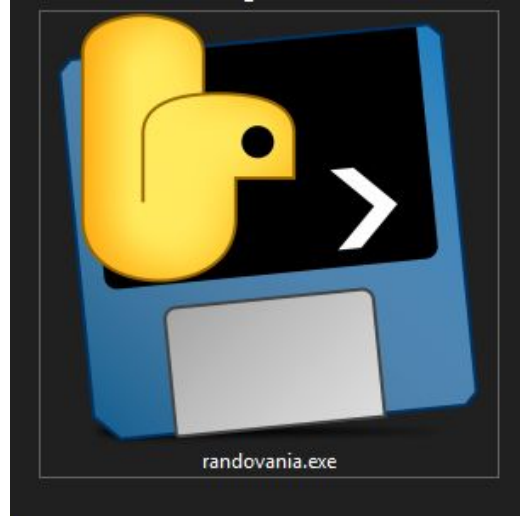
```
C:\Users\henri\programming\randovania>..\..\Downloads\randovania.exe echoes generate-seed --enable-tricks --difficulty 5 --skip-item-loss  
== Successful seed: 467341130  
Total seed count: 125
```

Gerar uma seed com Item Loss demora cerca de 20 horas de CPU.

Terceiro ciclo

Foco na usabilidade para usuário final

Exigir versão de Python certa e etc não é razoável
-> Distribuir um .exe nos releases (via PyInstaller)



Terceiro ciclo

(Pausa de ~8 meses no projeto)

A interface interativa não funcionou muito bem. O ideal seria uma interface gráfica mesmo.

Tentativa de usar Kivy:

- Ferramentas de montar telas não são muito boas
- Não gostei muito no geral



Terceiro ciclo

Qt for Python, com telas feitas no Qt Creator

The screenshot displays the Qt Creator interface for a Python application. The central area shows a window titled "Game Patching" with several sections:

- Current Game:** Text describing game file extraction, a "This label hasn't updated yet. What gives?" message, and buttons for "Load Game", "Export Game", and "Clear loaded game".
- Output Folder:** A text input field and a "Select Folder" button.
- Cosmetic Patching:** Checkboxes for "Skip Item Acquisition Popups" (with a description: "Replaces the 'Item Acquired' popup after collecting an item with an alert.") and "Faster Credits" (with a description: "Speeds up the credits to be 60 seconds long (over 4x faster).").
- Game Patching:** Checkboxes for "Create Spoiler" (with a description: "Creates a detailed spoiler file with where each modification to the game documented (item locations, elevator connections, etc) as well as a path you can take to finish the game.") and "Include Menu Mod" (with a description: "Menu Mod is a practice tool for Echoes, allowing in-game changes to which items you have and warping to all rooms.").
- Permalink:** A text input field and an "Import" button.
- ROM Output:** A green "Randomize" button and buttons for "Create only log file" and "Create from log file".

On the right, the **Object Inspector** shows a tree view of the application's objects and their classes:

Object	Class
ISOManagementWindow	QMainWindow
centralWidget	QWidget
current_game_box	QGroupBox
clear_game_button	QPushButton
description_label	QLabel
export_game_button	QPushButton
load_game_button	QPushButton
loaded_game_label	QLabel
output_folder_button	QPushButton
output_folder_edit	QLineEdit
output_folder_label	QLabel
game_patching_box	QGroupBox
create_spoiler_check	QCheckBox
create_spoiler_label	QLabel
include_m...mod_check	QCheckBox
include_m...mod_label	QLabel
seed_settin...rning_label	QLabel
warn_to_start_check	QCheckBox

Below the object tree, the **Property Inspector** shows the properties of the selected object, "ISOManagementWindow : QMainWindow":

Property	Value
cursor	Arrow
mouseTracking	<input type="checkbox"/>
tabletTracking	<input type="checkbox"/>
focusPolicy	NoFocus
contextMenuPolicy	DefaultCont
acceptDrops	<input type="checkbox"/>
▶ windowTitle	Hello, world
▶ windowIcon	
windowOpacity	1.000000
▶ toolTip	

Terceiro ciclo

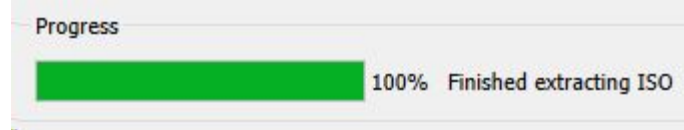
Uma ferramenta oficial do Qt for Python converte os arquivos do QtCreator em classes Python que são visualmente iguais.



Terceiro ciclo

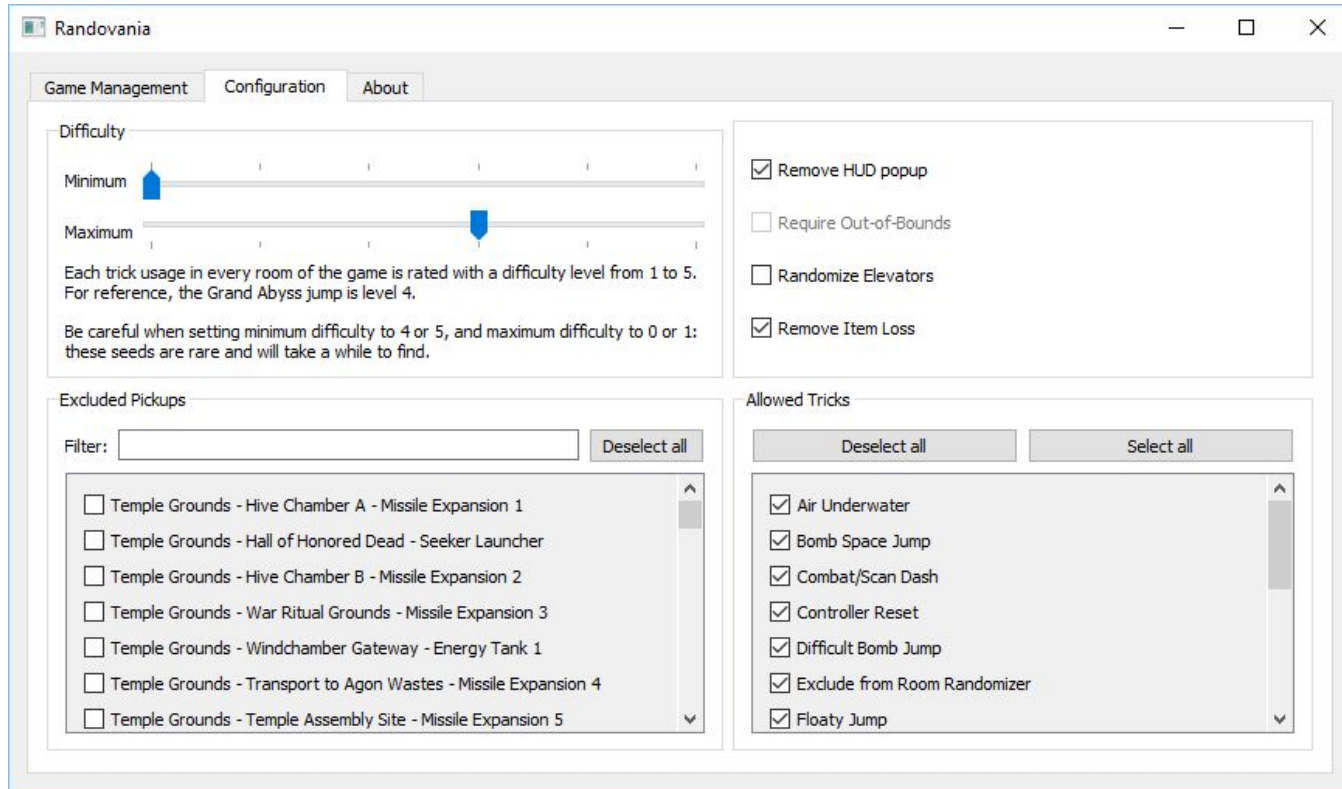
Importante manter o programa responsivo, sem travadas

-> Tarefas pesadas precisam ser executadas em outro ~~thread~~ processo



Inclusive com barras de progresso!

Terceiro ciclo



Terceiro ciclo

Para alterarmos o jogo, precisamos trabalhar diretamente com os arquivos utilizados pelo jogo. No entanto, o nosso usuário possui uma ISO do jogo, não os arquivos.

Para resolver isso, precisamos extrair a ISO e depois recriá-la.

Um par de pessoas dedicadas escreveram uma biblioteca C++ que cuida disso.

Para poder utilizá-la, foi necessário criar bindings Python para a lib. Utilizei o Cython para tal.



Quarto ciclo

Nesse momento, consegui conversar com a Claris e requisitar uma feature nova no Randomizer dela: especificar onde os itens vão.

Adaptar o algoritmo de validação para distribuição foi não-trivial:

- Resumidamente: “o N é grande”
- Queria evitar backtracking

Plano foi: jogador virtual anda pelo jogo fazendo tudo que pode. Sempre que trava, coloca algum item que é progresso onde pode chegar.

Quarto ciclo

Temos certeza que isso gera uma seed válida?

Quarto ciclo

Temos certeza que isso gera uma seed válida?

Nós temos um validador que fizemos antes. Que tal usar ele?

Quarto ciclo

Temos certeza que isso gera uma seed válida?

Nós temos um validador que fizemos antes. Que tal usar ele?





















- Encontrou muitos erros!
- Também serve para informar o usuário uma sequência de itens que vence o jogo
- Que também é útil para achar erros na lógica

Desvio de automação

Até então, releases eram feitos manualmente

Agora tem CI que roda os testes em Windows, Linux e macOS.

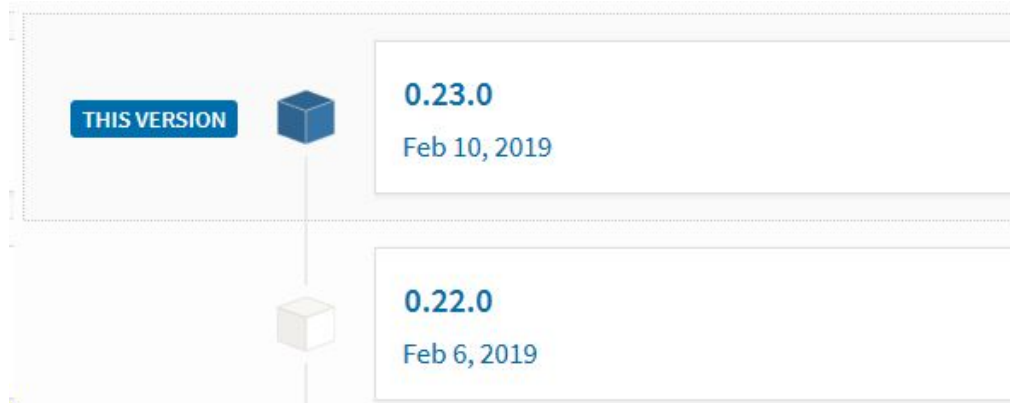
Também faz uma build para Windows e disponibiliza para download.

	 master Progress in revamp Henrique Gemignani Passos Lima	 #256 failed 265c9cb ↗	 6 min 7 sec  27 2 days ago
	 master Add RandomizerData.json Henrique Gemignani Passos Lima	 #255 failed f601384 ↗	 11 min 48 sec  27 3 days ago
	 pyup-update Add hud_text, sound_index and jing Henrique Gemignani Passos Lima	 #253 passed 9120c98 ↗	 11 min 32 sec  27 5 days ago
	 pyup-update Update twine from 1.12.1 to 1.13.0 Henrique Gemignani Passos Lima	 #252 passed 04a4714 ↗	 10 min 46 sec  27 5 days ago

Desvio de automação

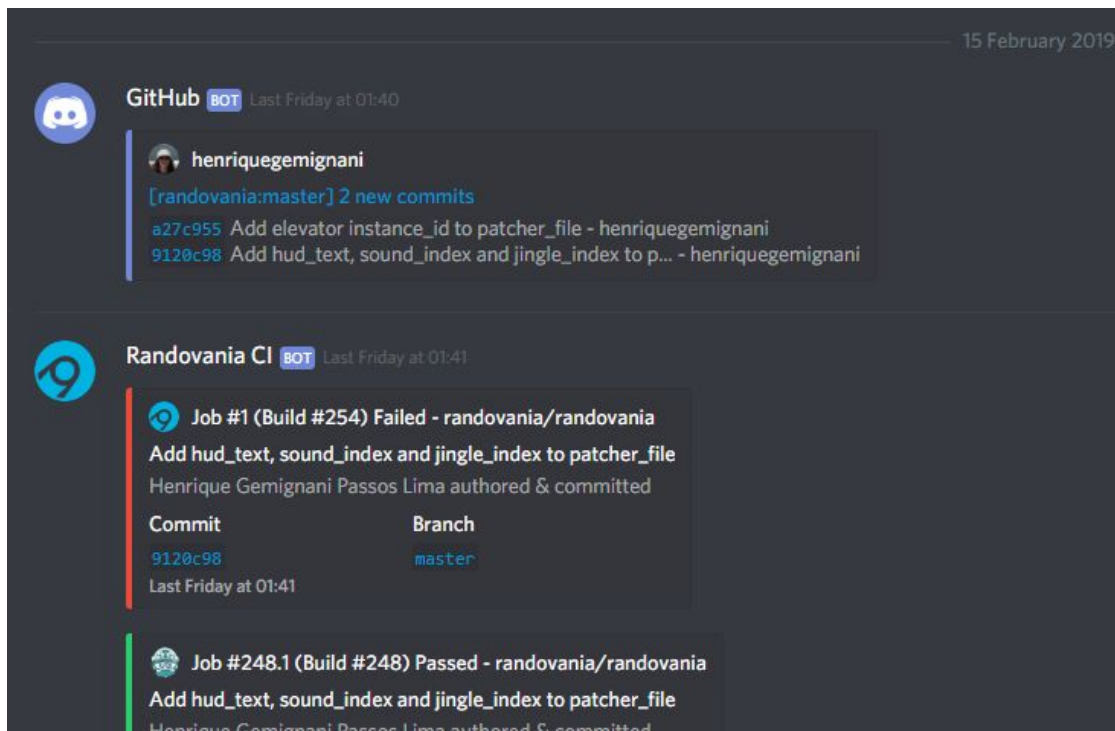
Ao dar push numa tag do git, o CI publica a build no GitHub Releases, versão nova do pacote no PyPi.

```
31 Deploying using GitHub provider
32 Creating "v0.23.0" release for repository "randovania/randovania" tag "v0.23.0" commit
   "b775d36ee3bfabe850f6eea3316fbf14ed800720"...
   OK
33 Uploading "randovania-0.23.0.zip" to release assets...OK
```



Desvio de automação

Também tem notificações no Discord do resultado das builds.



15 February 2019

GitHub BOT Last Friday at 01:40

henriquegemignani

[randovania:master] 2 new commits

- a27c955 Add elevator instance_id to patcher_file - henriquegemignani
- 9120c98 Add hud_text, sound_index and jingle_index to p... - henriquegemignani

Randovania CI BOT Last Friday at 01:41

Job #1 (Build #254) Failed - randovania/randovania

Add hud_text, sound_index and jingle_index to patcher_file

Henrique Gemignani Passos Lima authored & committed

Commit	Branch
9120c98	master

Last Friday at 01:41

Job #248.1 (Build #248) Passed - randovania/randovania

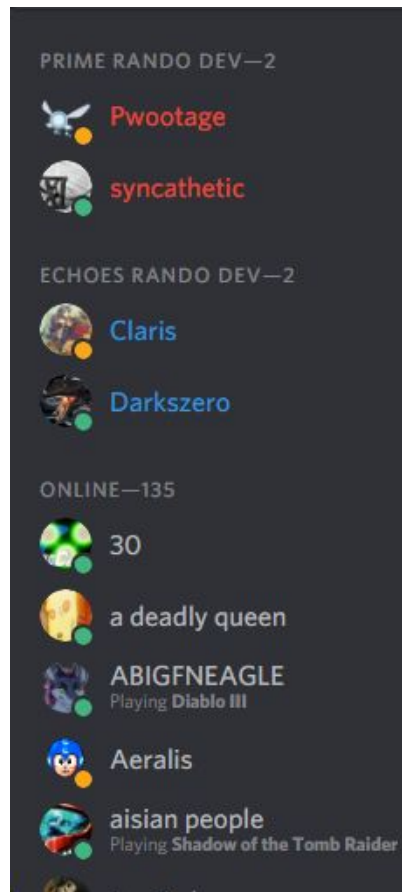
Add hud_text, sound_index and jingle_index to patcher_file

Henrique Gemignani Passos Lima authored & committed

Quinto ciclo

Agora que o programa é fácil de ser usado e não demora horas para gerar uma seed, ele começou a ser usado por mais pessoas!

Teve vários releases arrumando bugs na lógica, features novas que as pessoas pedem.



Sexto ciclo

As seeds sendo geradas estão com uma qualidade de gameplay ruim.



Mas já?

Sexto ciclo

É fácil saber que está ruim, mas o que exatamente está ruim? O que pode ser modificado para melhorar?

Gerar uma quantidade grande de seeds para fazer um relatório: para cada lugar, quais itens aparecem lá. Para cada item, em quais lugares ele aparece.

Com esse relatório, deu para identificar um erro bem feio.

```
"Dark Beam": {  
  "Storage D/Pickup (Dark Beam)": 393,  
  "Agon Temple/Pickup (Morph Ball Bomb)": 58,  
  "Doomed Entry/Pickup (Dark Agon Key 2)": 56,  
  "Mining Station Access/Pickup (Energy Tank)": 37,  
  "Agon Energy Controller/Pickup (Amber Translator)": 35,
```

Sexto ciclo

Tem vários jogos que a comunidade já fez um randomizer.

Estudada nos algoritimos usados neles

Conclusão: não funcionam dado algumas limitações que tem no Echoes

Melhorada nas buscas do grafo. Definição do tipo do problema que precisa ser resolvido: Fully dynamic all-pairs shortest paths

has a worst-case update time of $O(cn^{2+2/3} \log^{4/3} n)$

Tem um paper com um algoritimo bem simples que resolve isso

Rodada de testes

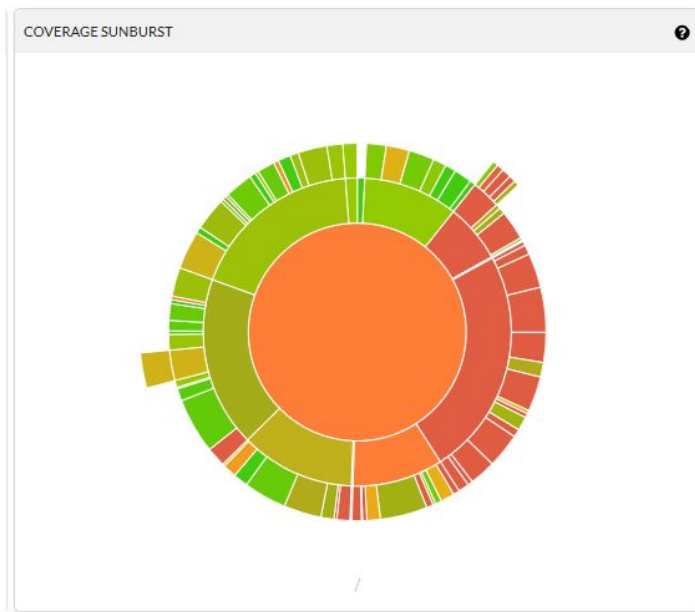
O projeto está em constante refatoração para melhor implementar features planejadas e mudanças de código

Para evitar quebrar as coisas acidentalmente, passei pelo projeto expandindo a cobertura de testes.



Rodada de testes

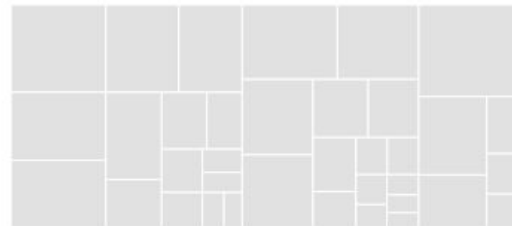
Integrei o codecov.io para poder visualizar facilmente a cobertura



codecov bot commented 5 days ago • edited

Codecov Report

Merging #66 into master will not change coverage.
The diff coverage is n/a.



@@	Coverage Diff		@@
##	master	#66	+/-
			##
-----			-----
Coverage	75.76%	75.76%	
-----			-----
Files	81	81	
Lines	5377	5377	
-----			-----
Hits	4074	4074	
Misses	1303	1303	

Hoje em dia

Features novas! Aleatorizar portas, salas, sala inicial, items novos, etc.

Algumas estão bloqueadas pelas buscas realizadas pela geração não ser muito eficiente. Gerar uma seed demora de 40s a 2min, dependendo da sorte e CPU.