Pandemic

Gerência de Projeto e Manutenção de Software

START MENU

Guilherme Vieira, Isadora Duarte, Izabel Soares, Lylian Pacheco, Mateus Sacramento e Nayara Dornelas

Sumário

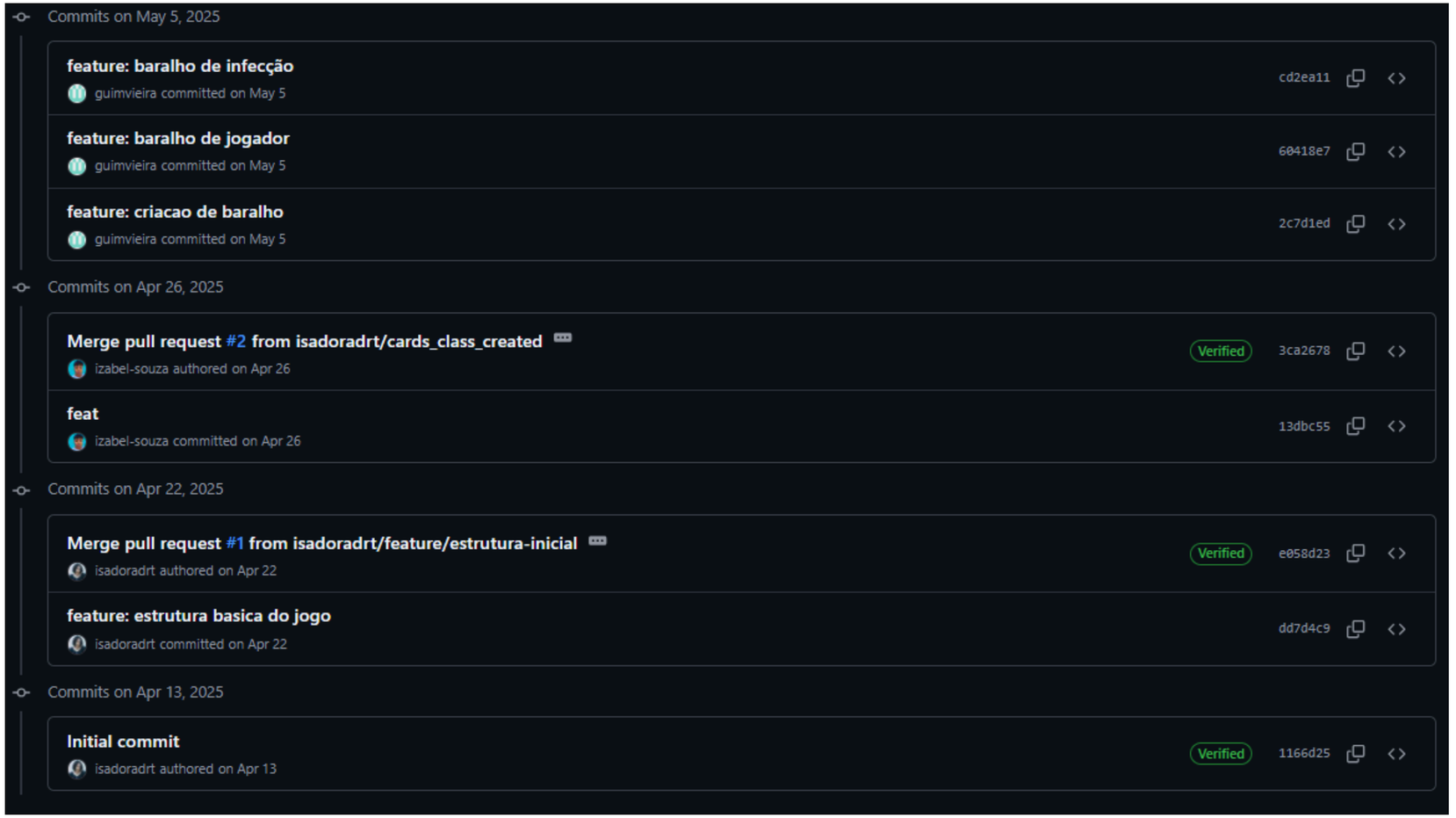
Tópicos

Controle de Versões

Commits realizados no repositório do GitHub contendo a implementação inicial de classes para o desenvolvimento do jogo, incluindo:

Exemplo:

- Classe de Baralho Infecção
- Classe de Baralho do Jogador
- Classe responsável pelo sistema de Infecção.

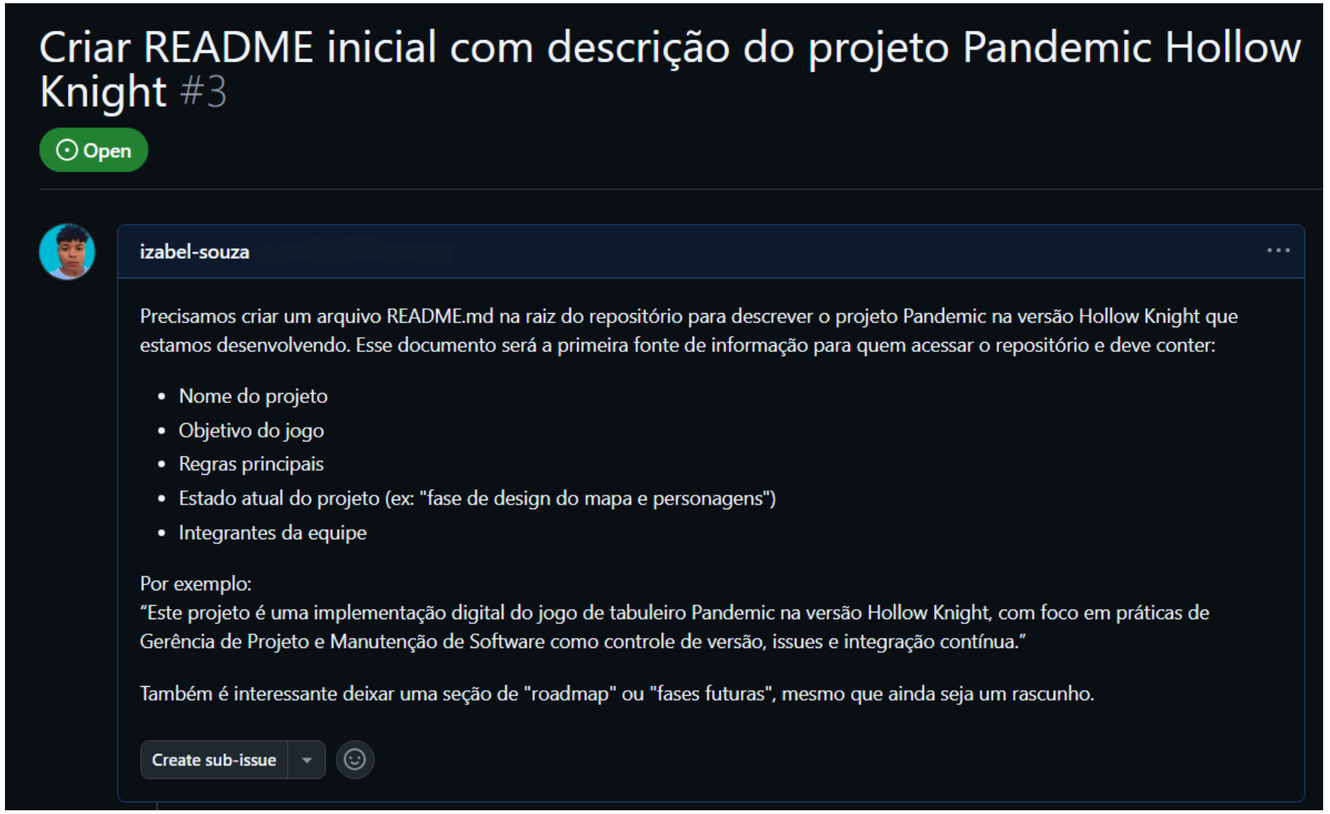


Controle de Modificação

Issue de documentação criada para controle de modificação.

Foi criada uma issue com o objetivo de organizar e rastrear a criação do arquivo README.md, centralizando as informações iniciais do projeto Pandemic (versão Hollow Knight), como:

- Nome e objetivo do projeto
- Regras principais
- Estado atual
- Integrantes da equipe

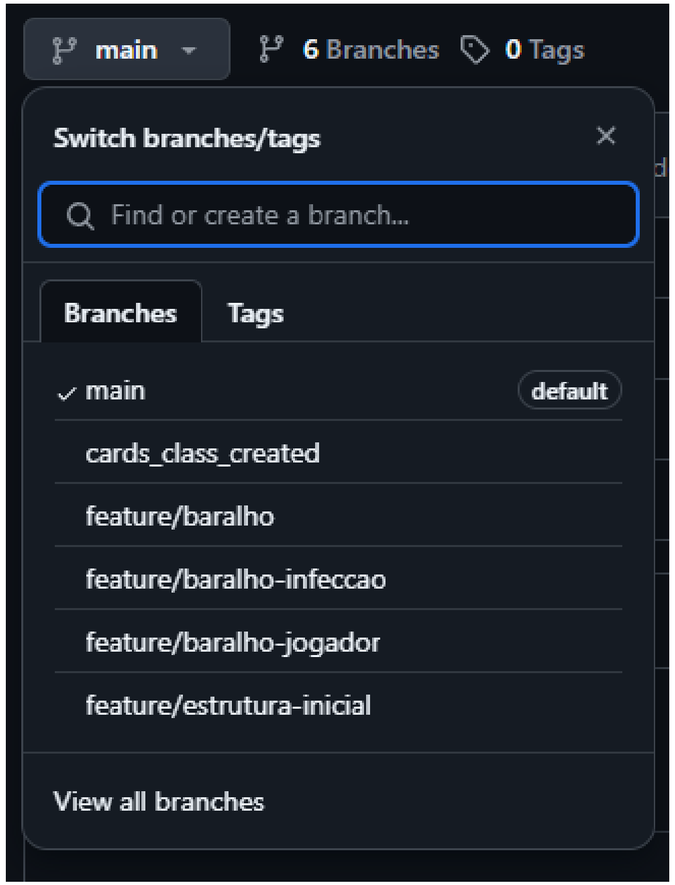


Estratégia de Ramificação

Git Flow

Adotamos a estratégia Git Flow para organizar o desenvolvimento do jogo, criando uma branch principal (main) estável e ramificações específicas (feature/*) para cada nova funcionalidade.

- facilita o controle de versões
- permite desenvolvimento paralelo
- as alterações passa por revisão via pull request antes de serem integradas



Pull Requests

Pull Requests realizados no repositório do jogo contemplam a criação e integração de componentes essenciais, como:

- A estrutura básica do jogo
- As classes responsáveis pelo baralho
- A classe que representa as cartas das cidades

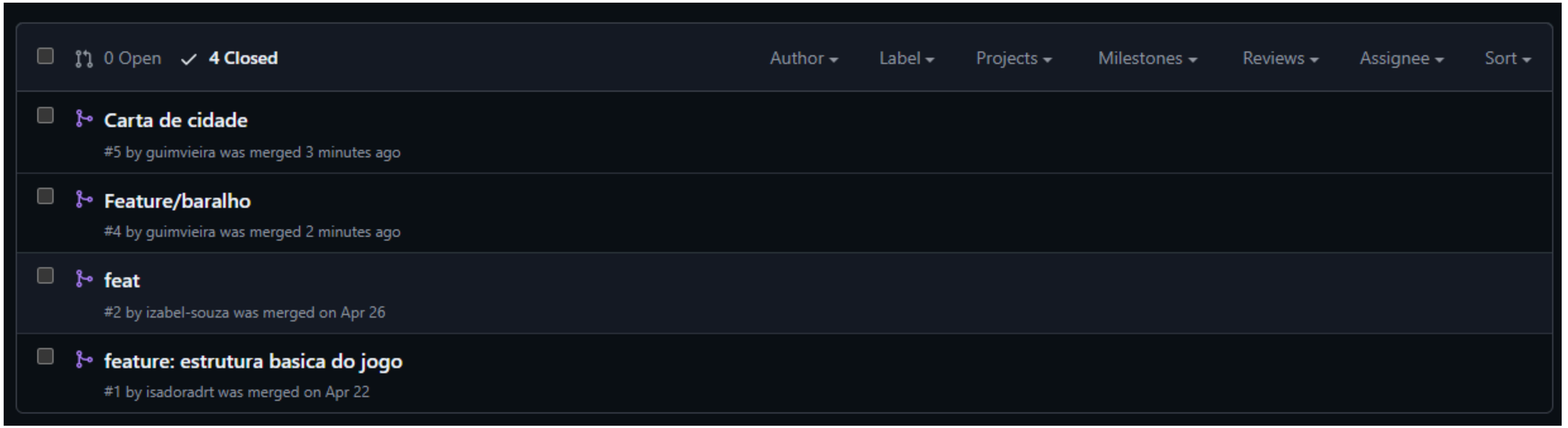


Gráfico de Burndown

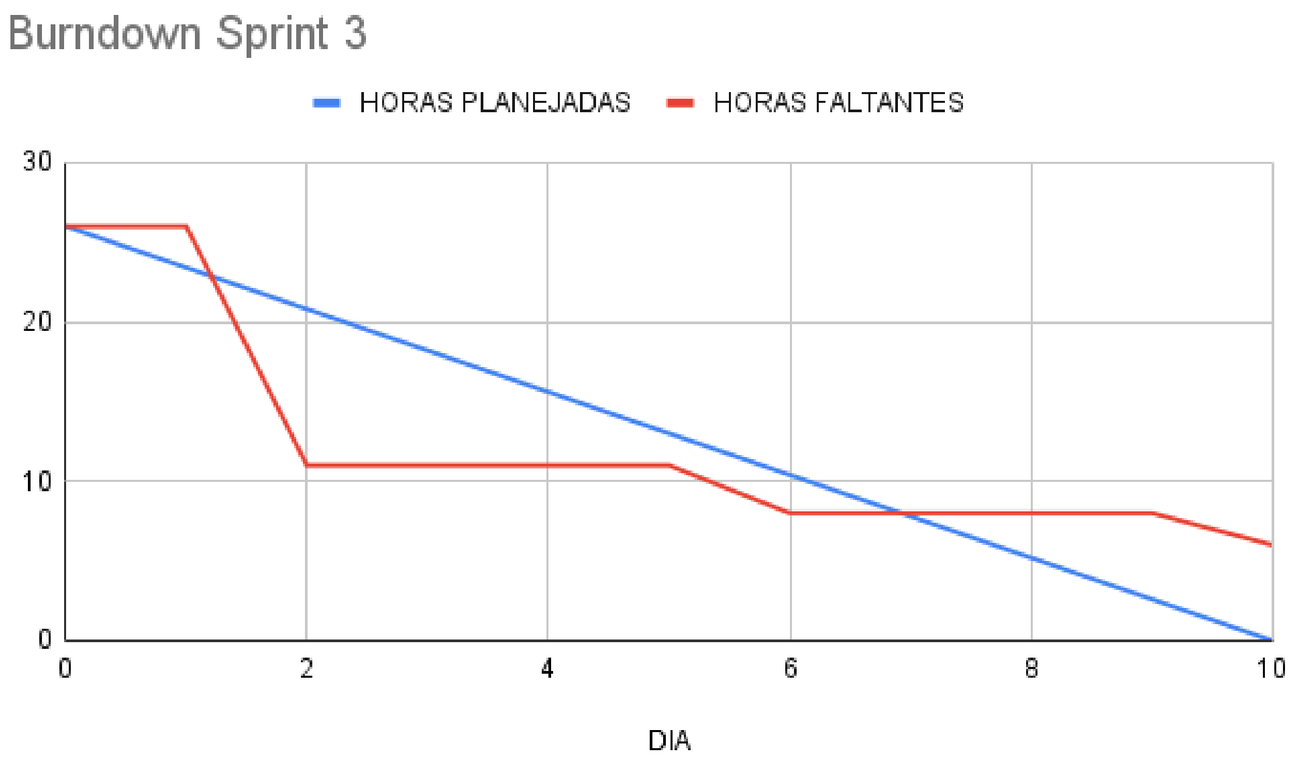
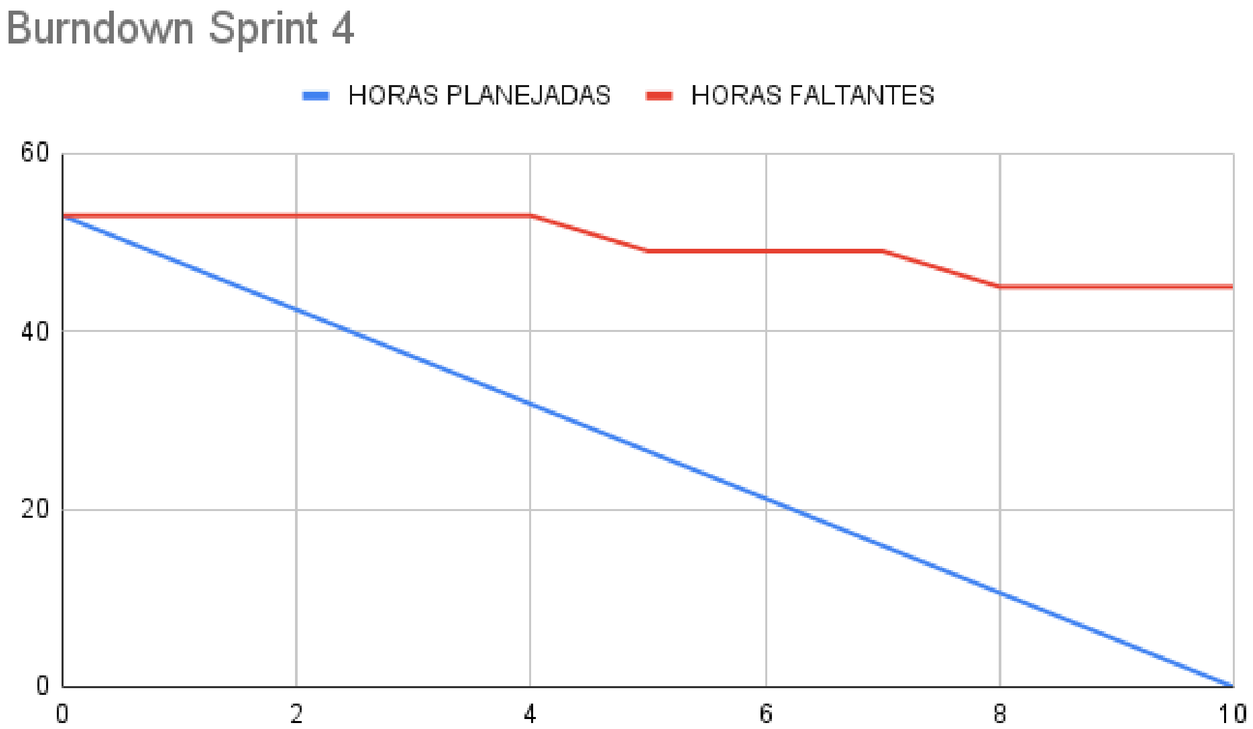


Gráfico de Burndown



Análise de Valor Agregado

Até a Sprint 4

Indicador	Valor
PV (Planejado)	146h
EV (Executado)	83h
Variação de Cronograma (SV = EV – PV)	-63
Índice de Desempenho de Prazo (SPI = EV / PV)	0,5685

Demo Parcial

Carta

```python
from enum import Enum

class TipoCarta(Enum):
    CIDADE = "Cidade"
    EVENTO = "Evento"
    EPIDEMIA = "Epidemia"

class Carta:
    def __init__(self, nome: str, tipo: TipoCarta, cor: str = None, descricao: str = None):
        self.nome = nome
        self.tipo = tipo
        self.cor = cor  # Só faz sentido para cartas de cidade
        self.descricao = descricao  # Só faz sentido para eventos ou epidemias

    def __str__(self):
        return f"{self.tipo.value}: {self.nome}" + (f" ({self.cor})" if self.cor else "")
```

Demo Parcial

Carta Cidade

```python
from cartas import Carta, TipoCarta


class CartaCidade(Carta):
    def __init__(self, nome, cor):
        super().__init__(nome, TipoCarta.CIDADE, cor)

    def __str__(self):
        return f"Cidade - {self.nome} ({self.cor})"
```

Demo Parcial

Baralho

```python
import random

# Classe de Baralho
class Baralho:
    def __init__(self, cartas=None):
        self.cartas = cartas or []

    def embaralhar(self):
        # Reembaralha o baralho
        random.shuffle(self.cartas)

    def tirar_carta(self):
        # Tira uma carta do baralho
        if self.cartas:
            return self.cartas.pop()
        return None

    def adicionar_carta(self, carta):
        # Adiciona uma carta ao baralho
        self.cartas.append(carta)
```

Demo Parcial

Baralho Jogador

```python
from baralho import Baralho

class BaralhoJogador(Baralho):
    def __init__(self, cartas=None):
        super().__init__(cartas)

    def tirar_carta(self):
        if not self.cartas:
            print("O baralho de jogador está vazio!")
            return None

        carta = super().tirar_carta()
        if carta:
            print(f"Carta do baralho de jogador tirada!")
        return carta
```

Demo Parcial

Baralho Infecção

```python
from baralho import Baralho

class BaralhoInfeccao(Baralho):
    def __init__(self, cartas=None):
        super().__init__(cartas)

    def tirar_carta(self):
        if not self.cartas:
            print("O baralho de infecção está vazio!")
            return None

        carta = super().tirar_carta()
        if carta:
            print(f"Carta de infecção tirada!")
        return carta
```

Demo Parcial

ALERTA!

Surto de infecção

ALERTA!

Surto de Esporos Fúngicos

Cura descoberta

ALERTA!

Surto de Vazio

Demo Parcial

Mensagem de Vitória/Derrota

Demo Parcial

Especialistas

KNIGHT

Impede infecção na sua cidade e nas cidades conectadas.

CORNIFER

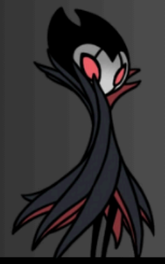
Pode dar qualquer carta para outro jogador, independentemente da cidade.

HORNET

Remove todos os cubos de uma doença com apenas 1 ação.

GRIMM

Pode mover outros jogadores como se fossem o próprio especialista, gastando cartas dele.

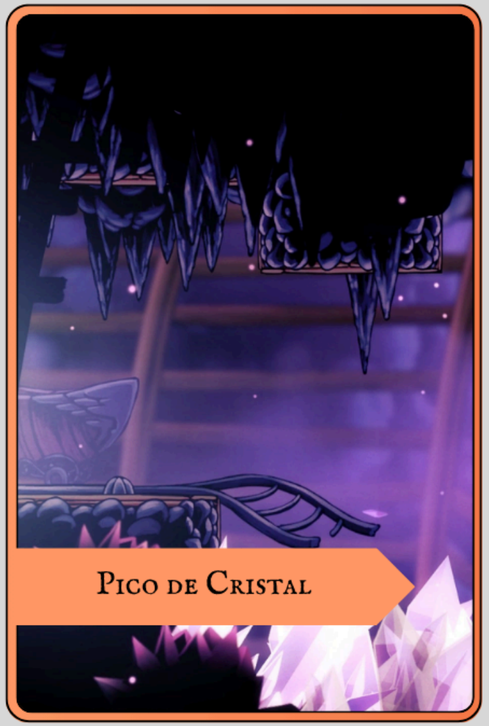


MONOMON

Descobre cura com apenas 3 cartas da mesma doença.

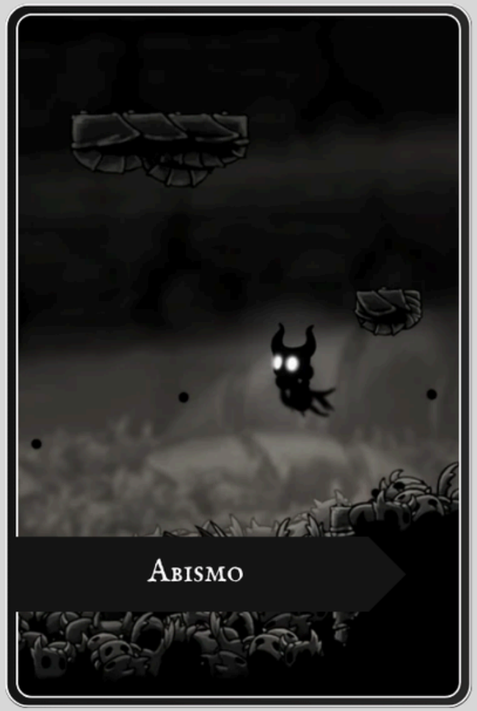
Demo Parcial

Cidades



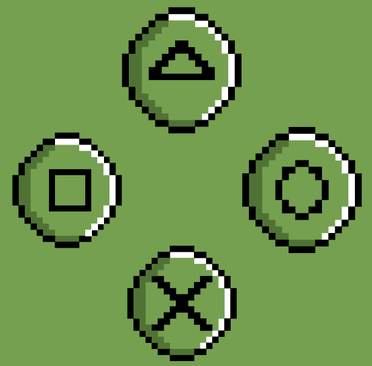
Demo Parcial

Cidades



GitHub

https://github.com/isadoradrt/Trabalho-GPMS



Obrigado!