



UNIVERSIDADE ESTADUAL PAULISTA
CIÊNCIA DA COMPUTAÇÃO

Documentação do projeto

Ana Beatriz Silva Zerati
William D'Abruzzo Martins

Documentação do projeto final da disciplina
Programação Orientada a Objetos do curso
de Ciência da Computação da Universidade
Estadual Paulista.

São José do Rio Preto
Maio de 2023

Sumário

1	Introdução	2
2	Objetivos	2
3	Termos e concepções do projeto	2
4	Desenvolvimento	3
4.0.1	Resumo geral	3
4.1	Interface gráfica	3
4.1.1	Tela inicial	3
4.1.2	Cadastrar	3
4.1.3	Visualizar mapa	3
4.1.4	Listar cadastrados	4
4.2	Mapa	4
4.2.1	Classes	4
4.2.2	Escolha dos ícones	4
4.3	Componentes	5
4.3.1	Interface Mapeavel	5
4.3.2	Classe abstrata Cadastrável	5
4.3.3	Rebeldes	5
4.3.4	Classe abstrata Ameaca	6
4.3.5	Patrulha	6
4.3.6	Blitz	6
4.3.7	Recebimento de carga	6
4.3.8	Combate	6
4.3.9	Marcha	6
4.3.10	Util	6
4.3.11	Main	6
4.3.12	Login	7
5	Trabalhos futuros	7

1 Introdução

Aqui é o Mestre Obi-Wan Kenobi. Sinto informar que tanto a Ordem Jedi quanto a República sucumbiram, e que as forças sombrias do Império tomaram seu lugar. Esta mensagem é um aviso e uma lembrança a qualquer Jedi sobrevivente: confiem na Força. Não retornem ao Templo, esse tempo já passou, e o nosso futuro é incerto. Seremos todos postos à prova: nossa confiança, nossa fé, nossas amizades, mas devemos perseverar. E com o tempo uma nova esperança surgirá. Que a Força esteja com você, sempre.

A República acaba de ser dissolvida e é instaurado o Império Galático, nas mãos do imperador Palpatine e seu aprendiz Darth Vader. Formam-se alguns núcleos de resistência contra a opressão do Império, mas o controle e imensidão da tropa imperial oprimem qualquer chance de revolta.

Muitas regiões não são capazes de comunicar-se umas com as outras devido ao perigo em se locomover, visto que o Império está atrás dos Jedis remanescentes e dos grupos que se opõem ao novo governo. Uma alternativa encontrada foi a de mandar hologramas, mas ainda sim há a necessidade de transição de pessoas ou droides que, caso interceptados, podem entregar informações importantes e comprometer a segurança dos rebeldes.

Com isso, faz-se necessária uma forma de comunicação à distância entre os grupos rebeldes, de modo a não comprometer a segurança dos comunicadores.

2 Objetivos

O *software* desenvolvido tem como objetivo facilitar a comunicação entre grupos rebeldes através de um mapa interativo. O mapa mostra locais onde eventos estão ocorrendo, sendo, assim, possível identificar locais com o risco de se transitar.

Além de mostrar os eventos, o mapa também apresenta os locais com os principais núcleos de encontro entre os rebeldes, facilitando a locomoção segura e promovendo acolhimento aos refugiados.

3 Termos e concepções do projeto

As atividades mais comuns do Império consistem em 5 diferentes tipos: operações de patrulha, blitz, recebimento de carga/escravos, combate e marcha. Um evento, assim, consiste na atividade, suas informações específicas e sua localização no mapa.

Os núcleos rebeldes, por sua vez, são específicos somente para acolhimento e reuniões. Dessa maneira, não há a divisão de diferentes tipos de núcleos, apenas as informações gerais e a sua localização no mapa.

4 Desenvolvimento

4.0.1 Resumo geral

O projeto começa com uma tela de Login que pede uma senha. Ao inserir a senha correta, uma tela inicial é aberta. Nela, o usuário possui quatro opções: cadastrar um novo evento, visualizar o mapa, visualizar a lista de cadastrados e sair.

Ao clicar na opção de cadastro, é necessário selecionar o tipo de cadastro a ser feito e preencher os dados, tanto gerais quanto específicos ao tipo escolhido.

Ao submeter as informações, os dados serão enviados para uma classe responsável por instanciar o objeto da entidade correspondente e guardar os dados no sistema, também cadastrando-os no mapa. Depois dessa operação no *backend*, o usuário retorna para a tela inicial.

Na tela de listagem, itera-se sobre os objetos de cada *Marker* cadastrado, imprimindo suas informações e sua descrição. Haverá um botão para o usuário retornar à tela inicial quando bem preferir.

Na tela de mapa, será mostrado o mapa com os eventos cadastrados, também dando ao usuário a opção de voltar.

Ao clicar o botão de sair, os dados são novamente salvos no arquivo (serializando os objetos em um arquivo de extensão *.ser*) e o programa é fechado.

4.1 Interface gráfica

Apresenta para o usuário as telas tanto de apresentação quanto de preenchimento de dados, salvá-los nos objetos quando necessário.

4.1.1 Tela inicial

Tela que deverá mostrar as opções de operações para realizar, como Cadastrar, Listar, Visualizar mapa e Sair. A escolha de uma operação redireciona da tela inicial para a tela específica da operação.

4.1.2 Cadastrar

Tela genérica de cadastro que, a princípio, apresenta informações básicas. Elas são coordenadas em x, coordenadas em y, descrição longa e o tipo do cadastro. A partir da seleção de um tipo, itens específicos são apresentados para preenchimento, como, por exemplo, veículo, no caso de cadastro de Blitz.

4.1.3 Visualizar mapa

Tela que apresenta o mapa, com ícones representando cada cadastro. Ao posicionar o cursor do mouse no ícone, é apresentado o nome, como Rebelde, Patrulha, entre outros, e uma descrição curta.

4.1.4 Listar cadastrados

Tela responsável por mostrar todos os eventos que estão cadastrados no mapa, em forma de lista.

Cada *Panel* apresentado é clicável, que abre uma tela secundária menor mostrando as informações completas do evento e também dando a opção de excluí-lo.

4.2 Mapa

O mapa deve ser desenvolvido de modo que a proporcionar as seguintes finalidades:

- Visualização geral do mapa
- Visualização de ameaças cadastradas
- Visualização de núcleos rebeldes
- *Toggle* para uma detalhamento curto de determinado evento.
- *Toggle* para um detalhamento curto de um núcleo rebelde.

4.2.1 Classes

Classes usadas para o desenvolvimento do mapa

- **MapPane** Tem como atributo o fundo (*BufferedImage*). Classe responsável por armazenar a base do mapa.
- **Marker** Classe genérica que representa um marcador no mapa. Como atributo, tem um objeto genérico *T* que implementa a interface *Cadastrável*. A partir da classe, recupera o caminho da imagem do ícone e sua descrição curta (atributos estáticos das classes que herdam de *Ameaca* e *Rebeldes*).
- **Tracker** Classe responsável por renderizar o mapa com seus ícones. Ela possui como atributo uma lista de *Markers* e um *MapPane*. Assim, também tem a responsabilidade de cadastrar e remover ícones do mapa, gerenciando-o.

4.2.2 Escolha dos ícones

Os ícones escolhidos servem para representar visualmente a atividade em questão, de modo que seja identificável apenas olhando para o ícone.

- **Patrulha**



- Marcha



- Blitz



- Recebimento de carga



- Combate



- Núcleo de rebelde



4.3 Componentes

Pacote com classes que representam as entidades do sistema

4.3.1 Interface Mapeavel

Tem a responsabilidade de definir métodos para tratar o relacionamento de um objeto com o mapa, como suas descrições e posição.

4.3.2 Classe abstrata Cadastrável

Classe que apresenta todos os atributos que os objetos cadastráveis no mapa devem ter. Implementa a interface Mapeável, sobrescrevendo alguns de seus métodos.

4.3.3 Rebeldes

Classe responsável por representar a entidade rebelde do sistema. Possui como atributos o caminho da imagem que o representa no mapa (String), suas coordenadas, descrições (curta e longa) e número de pessoas.

4.3.4 Classe abstrata Ameaca

Classe responsável por representar as ameaças aos rebeldes. Possui atributos compartilhados pelas ameaças, como suas coordenadas, caminho para imagem, descrição curta, descrição longa, gravidade (grave, média, pouco), e número.

4.3.5 Patrulha

Herda da classe abstrata Ameaça, sendo responsável por representar uma operação de patrulha. Tem como atributo específico seu tipo: reconhecimento, assalto ou desconhecido.

4.3.6 Blitz

Herda da classe abstrata Ameaça, sendo responsável por representar uma blitz. Tem como atributo específico o tipo dos veículos (String).

4.3.7 Recebimento de carga

Herda da classe abstrata Ameaça, sendo responsável por representar uma operação de recebimento de carga. Seu atributo específico é o tipo da carga (String): armas, escravos, desconhecido, entre outros.

4.3.8 Combate

Herda da classe abstrata Ameaça, sendo responsável por representar situações de combate. Seus atributos específicos são a quantidade de aliados e a situação geral (boa, ruim ou média).

4.3.9 Marcha

Herda da classe abstrata Ameaça, sendo responsável por representar uma marcha do império. Possui como atributo específico o tamanho da marcha (em Km).

4.3.10 Util

Classe final, responsável por apresentar métodos de leitura e escrita de arquivos serializáveis.

4.3.11 Main

Lê dados dos arquivos e chama a classe da interface inicial, que recebe o controle do programa.

4.3.12 Login

Tela na qual aparece as opções para logar e também inicia o sistema caso a senha colocada pelo usuário esteja correta.

5 Trabalhos futuros

Atualmente o *software* só funciona no planeta de Coruscant. Como expansão, seria importante dar a oportunidade do usuário pesquisar e selecionar o planeta em que ele deseja visualizar os eventos, de forma que possamos ajudar outros núcleos rebeldes espalhados pela galáxia.