

Relatório do Projeto 1 de Estrutura de Dados

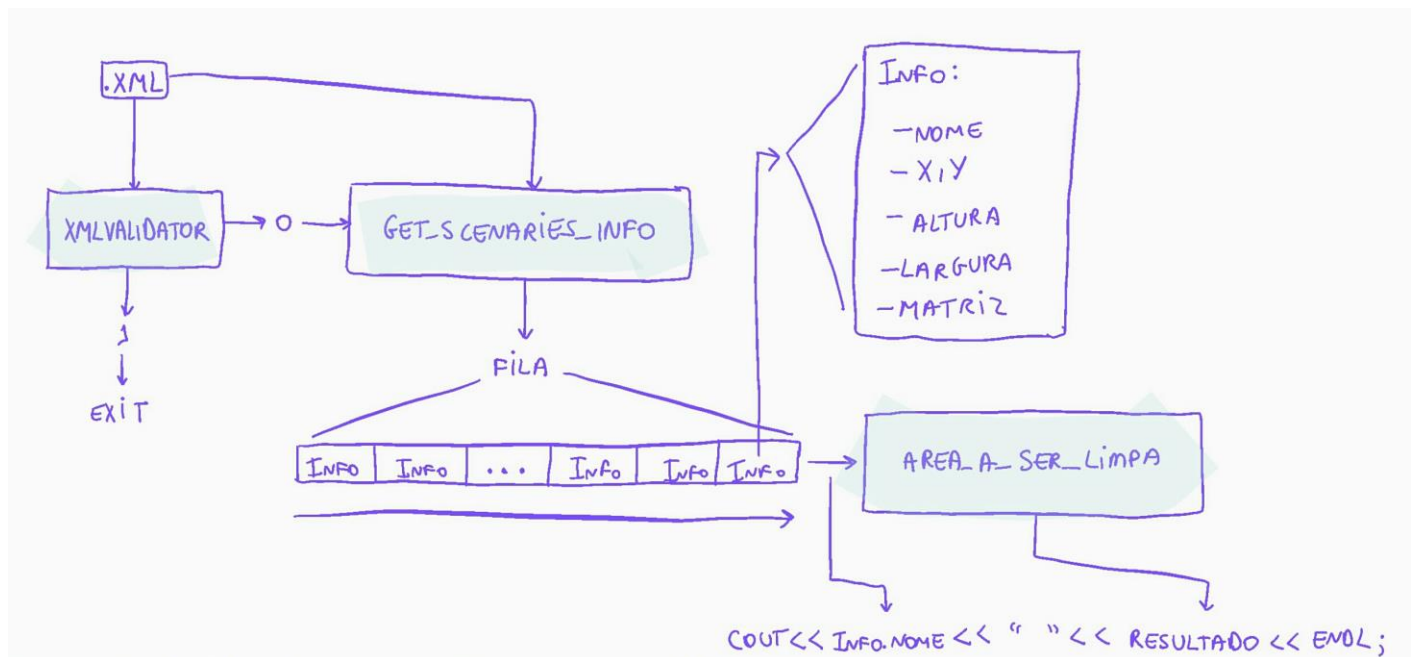
Nome: Enzo Nicolás Spotorno Bieger

Matrícula: 22100614

Data: 23/04/2023

UFSC

Desenho do funcionamento do programa:



Explicação:

O funcionamento do programa pode ser dividido em 3 partes principais:

- XMLVALIDATOR: Função que valida um arquivo XML;
- GET_SCENARIOS_INFO: Função que lê um arquivo XML e guarda as informações de cada cenário presente no arquivo em uma fila circular;
- AREA_A_SER_LIMPA: Função que dadas as informações de um cenário, retorna a área a ser limpa pelo robô;

XMLVALIDATOR:

- Recebe o nome de um arquivo XML;
- Aloca uma pilha de strings;
- Abre o arquivo XML;
- Percorre cada caractere de cada linha do arquivo;
- Quando encontra uma marcação de abertura:
 - o Descobre qual o identificador e o empilha;
- Quando encontra uma marcação de fechamento:
 - o Descobre qual o identificador e:
 - Verifica se a pilha está vazia: ERRO;
 - Verifica se o identificador do topo da fila é diferente do encontrado: ERRO;
 - Se não houver erro, desempilha;
- Ao finalizar a análise do arquivo, se a pilha não estiver vazia: ERRO;
- Fecha o arquivo;
- Caso nenhum erro tenha sido levantado, significa que o arquivo é válido.

GET_SCENARIOS_INFO:

- Recebe o nome de um arquivo XML;
- Aloca uma fila circular de Info's (estrutura que guarda as informações de um cenário);
- Abre o arquivo XML;

- Inicializa uma flag `is_matrix` que simboliza quando encontramos uma matriz (a matriz está distribuída em linhas consecutivas do arquivo);
- Inicializa uma matriz pra ir guardando a matriz lida do arquivo;
- Inicializa uma estrutura do tipo `Info` para guardar as informações do cenário;
- Percorre cada linha do arquivo;
- Se a flag `is_matrix` está ativada:
 - o Se não achou a marcação de fechamento de matriz:
 - Guarda essa linha na matriz;
 - o Se achou a marcação de fechamento de matriz:
 - Guarda a matriz na estrutura do tipo `Info`;
 - Desativa a flag `is_matrix`
- Se a flag `is_matrix` não está ativada:
 - o Se achou a marcação de abertura de matriz:
 - Aloca a matriz com altura linhas e largura colunas;
 - Ativa a flag `is_matrix`;
 - o Sempre que encontrar alguma outra marcação que não seja a de fechamento de cenário:
 - Guarda a informação no respectivo campo da estrutura `Info`;
 - o Se encontrar a marcação de fechamento de cenário:
 - Coloca a estrutura `Info` desse cenário na fila;
- Fecha o arquivo;
- Retorna a fila com estruturas `Info` de cada cenário;

AREA_A_SER_LIMPA:

- Recebe uma estrutura `Info` com as informações de um cenário;
- Cria uma fila de coordenadas (estrutura com `x` e `y`) cujo tamanho máximo é o número de 1's da matriz do cenário;
- Cria uma matriz `R` (reconstrução) preenchida de zeros com o mesmo tamanho da matriz do cenário;
- Pega a coordenada (`x`, `y`) com a posição do robô;
- Se na matriz do cenário essa coordenada corresponder à 1:
 - o Insere (`x`, `y`) na fila;
 - o Atribui 1 na coordenada (`x`, `y`) da matriz reconstrução;
- Enquanto a fila não estiver vazia:
 - o Retira uma coordenada da lista;
 - o Pega as coordenadas vizinhas desta coordenada;
 - o Pra cada vizinho:
 - Verifica se a coordenada é válida (Dentro do domínio da matriz, 1 na matriz do cenário, 0 na matriz reconstrução);
 - Se for válida coloca na fila;
 - E coloca 1 na matriz reconstrução nessa coordenada;
- Agora que a matriz reconstrução está devidamente preenchida,
- Retorna a quantidade de 1's na matriz reconstrução;

Desta maneira, na função `main` validamos o arquivo XML, se ele for válido obtém a fila circular com as informações de cada cenário do arquivo e pra cada cenário na fila printa na tela o nome do cenário seguido pela área a ser limpa pelo robô.

Conclusão:

Acredito que a parte mais complicada/trabalhosa do projeto foi a leitura e obtenção dos dados lidos dos arquivos XML, nesta etapa tive que pesquisar sobre as maneiras de ler as tags do arquivo e passar um tempo entendendo como o arquivo estava estruturado pois algumas linhas continham mais do que uma tag enquanto

as matrizes estavam distribuídas em várias linhas. Outra parte que me enrolei um pouco foi com x sendo as linhas e y as colunas (eu não havia percebido isso e então passei um tempo tentando descobrir o problema).

Acredito que o projeto foi significativamente mais difícil por não se poder debugar o código com ferramentas próprias, isso tornou mais complicado diagnosticar e resolver os problemas.

Referências:

<https://cplusplus.com/doc/tutorial/files/>

<https://cplusplus.com/reference/string/string/>