

• Sobre o software

O seguinte software, ainda em fase de desenvolvimento, tem como pretensão ser um ambiente de desenvolvimento gráfico e interativo de softwares – similar a uma IDE - simples para uma linguagem funcional que será desenvolvida futuramente e será utilizado no ensino básico com a finalidade de desenvolver habilidades do pensamento computacional como o raciocínio algorítmico, a abstração de conceitos, como também a composição, decomposição e generalização. Tudo será implementado graficamente, tanto a construção do software pelo usuário, quanto o processamento/compilação do mesmo. A versão disponível só dispõe da parte gráfica, i.g, a interface de usuário, nenhum tipo de processamento ou construção de software real ainda é suportado, tudo que foi implementado é relacionado ao front-end, ou seja, a interação com o usuário final. O software foi construído em cima de um software open-source exemplo (Diagram Scene Example) para construção de diagramas disponibilizado pela própria Qt, mais detalhes estão explicitados na seção Estrutura.

• Como usar

A atual versão não é realese, e portanto precisa ser compilado e executado a partir do source code disponível, para tal é necessário o framework Qt, e a IDE Qt Creator; o framework e IDE estão disponíveis em ¹, e os requerimentos estão explícitos em ². Pode-se realizar o download do source code no seguinte repositório: <https://github.com/Matheusih/UIapp>

• Estrutura

A estrutura atual utiliza as bibliotecas disponíveis do framework Qt para construir um software modular dentro do paradigma orientado a objetos para o fácil incremento futuro do mesmo, isto é, novas funcionalidades podem ser adicionadas sem a necessidade de uma mudança estrutural muito grande.

Nele utiliza-se uma Mainwindow responsável por lidar com todos eventos ou ações do usuário (cliques, atalhos de teclado, etc), na Mainwindow também estão todos os elementos que aparecem na tela quando o software é executado, como seus botões, menus, e ícones. Na mainwindow se instanciará um objeto da classe DiagramScene (scene), responsável por dispor na tela todos os objetos gráficos através da instância da classe GraphicsView, a scene lidará com eventos dentro dela como arrastar / clicar / adicionar / deletar um objeto. Na scene pode-se adicionar objetos que serão vistos como funções, estas funções possuem uma saída (output) fixa cujo tipo poderá ser definido pelo usuário, e n entradas (inputs) que poderão ser adicionadas também pelo usuário; os tipos das entradas e saídas, no momento são: Numérico, Booleano, Imagem, String. No entanto, existe uma função principal – main – e todas funções devem ser criadas dentro dela, ela é representada como um retângulo vermelho, um clique direito nela mostrará um menu com a possibilidade de se adicionar entradas (que devem ser constantes) na função principal, a saída da mesma não está suportada na atual versão. As funções podem ser nomeadas, as suas entradas e saída, na atual versão, não. Texto também pode ser adicionado à scene, na intenção de ser utilizado como comentários. Deseja-se que um clique duplo em uma função abra uma nova aba na janela principal para que essa função possa ser implementada e composta por outras funções, dessa maneira o programa poderá ser navegado de uma maneira similar a um sistema de diretórios – em árvore.

1 <https://www.qt.io/download>

2 <https://doc-snapshots.qt.io/qt5-5.11/windows-requirements.html>

Intenta-se, com finalidade de regularidade formal, que os objetos adicionados na DiagramScene sejam todos instâncias e especializações da classe DiagramItem, esta construirá o objeto de acordo com o botão selecionado no menu da esquerda, uma vez adicionado à scene o objeto pode ser arrastado dentro da função principal; um clique direito no objeto exibirá o menu do objeto, possibilitando a renomeação dele, adicionar entradas, deletá-lo,

• Funcionalidades desejáveis

Para versões futuras será desejável:

1. Suporte para nomeação das entradas/saídas e definição dos tipos das saídas
2. Conectar a saída de uma função na entrada de outra função
3. Entradas e saídas flexíveis, que possam ser posicionadas dinamicamente pelo usuário
4. Construir a saída da função principal
5. Sistema de abas para navegação nas funções
6. Salvar o que foi criado em um arquivo, e poder carregar o mesmo
7. Gerar caminho que as entradas das funções principais percorrerão, de forma sequencial

Com estas funcionalidades implementadas o software estará funcional, e portanto apto para simular pequenas operações. A partir desse momento o back-end poderá começar a ser projetado.

• Autor

O seguinte software, ainda em construção, foi desenvolvido até o presente momento pelo bolsista de iniciação científica Matheus Hertzog pela Universidade Federal de Pelotas, sob orientação de Luciana Foss, como uma ferramenta a ser utilizada no projeto *Proposta Metodológica para a Introdução do Raciocínio Computacional no Ensino Fundamental – CDTEC*.