

Projeto RobôFun

01/01/2021

|  |  |  |
| --- | --- | --- |
| Quem  somos | Onde  estamos | Onde chegaremos |

|  |  |  |
| --- | --- | --- |
| Quem somos | | |
| Um pouco da história do projeto | | |
| Surgido em 2012, o projeto RobôFun consiste em criar um robô de baixo custo, que busca atuar diretamente no ensino de programação de maneira diferente do convencional | | |
| O projeto RobôFun, iniciado em 2012, consiste no desenvolvimento de um robô de baixo custo, dotado de sensores básicos e de uma interface de programação para usuários leigos. De forma mais abrangente, tem por objetivo permitir ao grupo PET aprimorar seus conhecimentos sobre conteúdos técnicos e metodológicos relacionados ao projeto do hardware e do software de controle do robô e a uma linguagem e interface de programação de alto-nível. Também trabalha conhecimentos ligados ao ensino e à gestão de projetos aprimorando habilidades sociais (ex. trabalho em grupo e comunicação) e de planejamento e execução de tarefas (ex. metas, acompanhamento e avaliação). |  | Como atividade extensionista, o projeto ambiciona atrair estudantes do ensino médio para as áreas de computação e engenharia. Também busca amenizar o problema de evasão dos recém-ingressos nos cursos de Engenharia de Computação, Bacharelado em Sistemas de Informação e afins. |
| Edifício bonito. Novo edifício. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Estado da arte | | | |
| O que já fizemos até o presente momento | | | |
| Diante de várias transformações ao longo de 8 anos, tanto de equipe como de planejamento, diversos tópicos foram desenvolvidos. Dividimos elas em duas partes principais: Hardware e Software  O hardware | | | |
| A qualquer momento você pode alterar facilmente o texto de qualquer seção deste documento. Para isso, basta clicar e digitar. O modelo está preparado para que a formatação permaneça intacta com as novas informações que você incluir.  Para atualizar a formatação, na guia Página Inicial, você pode usar a galeria de Estilos Rápidos para os estilos internos. Você também pode formatar diretamente o texto usando outros controles na guia Página Inicial. | |  | Pintura |
| Construtores. Construções. | |  | Na guia Design, altere o tema, as cores ou fontes do documento. Com o clique de um botão, atualize todo o documento para o esquema de cores principal e o esquema de fontes que você escolher!  Para alterar qualquer uma das imagens, clique nela e escolha "Alterar imagem" na guia Ferramenta de imagem. Na imagem da primeira página, clique duas vezes no cabeçalho para acessar a imagem e atualizá-la conforme necessário. |
| Outro Cabeçalho Aqui | Acrescente um item adicional que você deseja realçar nessa área. Certifique-se de incluir as informações de contato, detalhes do evento ou outros tópicos importantes que seu público precise saber! | | |

|  |  |  |
| --- | --- | --- |
| O software | | |
| Inicialmente cabe ressaltar que existem duas partes: A parte voltada para o ensino (que contém a programação em blocos etc), e a voltada para a pesquisa, que consiste apenas no simulador, que permite a utilização de algoritmos mais complexos, com por exemplo, o LRTA\*.  Todo o projeto está disponível de maneira livre em: [*https://github.com/luancarlosklein/RoboFun-PETECO*](https://github.com/luancarlosklein/RoboFun-PETECO). Foco na Educação Uma vez que o objetivo do projeto é o ensino da programação, foi desenvolvido um sistema de programação em blocos, que permite o usuário programar o robô, e aplicar o que ele produziu tanto em um simulador adaptado como no mundo real, no próprio robô físico.  Na imagem ao aldo está um exemplo simples de como se encontra, atualmente, a programação em blocos e o simulador para a programação realizada. | Toda essa parte está dentro da pasta *ProgrammingToTeach.* Para realizar a execução do projeto, basta inicializar o *main.py*. Todo o código está documentado e comentado. Abaixo está uma imagem do diagrama de classes (simplificado) do mesmo, para facilitar a compreensão. |  |

|  |  |
| --- | --- |
| Foco na Pesquisa Além de ter como objetivo o auxilio no ensino da programação, na atração de novos alunos para os cursos de tecnologia e na diminuição da evasão, o projeto também adquiriu o objetivo de fomentar a pesquisa científica dentro do grupo PET e também na universidade. Essa última parte foi criada no segundo semestre de 2020, e ainda está sendo desenvolvida e construída.  Utilizando como base a iniciação científica em sistemas multiagente, do professor César A. Tacla, foi criado um tutorial que visa facilitar a adptação aos novos conceitos, para que os alunos possam realizar pesquisa e desenvolver resultados efetivos de maneira mais rápida e sólida possível. Esse material está disponível em: [*https://github.com/luancarlosklein/ProjetoPetecoEnsinoIA*](https://github.com/luancarlosklein/ProjetoPetecoEnsinoIA).  Utilizando como base os conhecimentos adquiridos no material desenvolvido, busca-se realizar pesquisa científica, utilizando o RoboFun como ferramenta. Para tal, foi desenvolvido um simulador (presente na pasta *SimulatorToSeach*), que permite a execução de um ambiente similiar ao do Jason, com agente baseado no modelo BDI, e a possibilidade de aplicação de algoritmos mais complexos para a execução (tanto no simulador como no robô físico). Um dos algotimos já desenvolvido que está presente nele, de maneira funcional é o Learning Real Time A\* (LRTA\*).  Além disso, outro ponto que já está implementado é a utilização de um *planner*. Na atual versão, é utilizado uma versão online e gratuita através de uma API. | O ambiente atual é em formato de grid e estático. Mesmo sendo simples atualmente, apresenta oportunidades de evolução e incremento do software. |

|  |  |  |
| --- | --- | --- |
| Onde queremos chegar | | |
| Quais são nossos planos e objetivos | | |
| Tendo em vista que 2020 foi um ano peculiar e de muitas incertezas, buscamos tentar tirar o máximo de aprendizado dele, e fazer com que 2021 seja proveitoso e de evolução, mesmo sendo no estilo home office | | |
| Dentre todos os objetivos do projeto, dividimos-os em 3 aspectos distintos nesse momento para serem concluídos nesse ano de 2021: Funcional, Educativo e Científico. Funcional Nesse âmbito, o objetivo é fazer com que o robô esteja 100% funcionando de maneira satisfatória e integrado com as plataformas de programação em blocos e simulação. Dentre alguns aspectos que devem ser observados ainda são: - Correção de possíveis bugs na programação; - Melhorar aspectos de usabilidade da programação em blocos (ajuste de imagens, correção de encaixes); - Evolução para que tanto a programação como o simulador permitam ambientes livres, e não apenas *grids*. Educativo Visto que o objetivo do RobôFun é ensinar programação e robótica, buscando motivar e atrair estudantes para a área de tecnologia, é necessário a realização de atividades presenciais para que vejam o funcionamento “ao vivo” da tecnologia. Entretanto, devido ao Covid-19, atividades presenciais se encontram restritas. |  | Devido a falta de previsão para que elas possam ser retornadas com segurança, as aulas não seram realizadas no momento. Porém, buscaremos desenvolver todo o material necessário para realizar as futuras aulas. Dentre esses materiais estão: - Planejamento das aulas, definindo seus conteúdos, slides, preparação das aulas etc; - Construção de atividades para serem realizadas pelos alunos; Científico O novo aspecto que o projeto busca abordar e o da realização de pesquisa científica. Através dela, além da própria evolução pessoal dos envolvidos, o grupo em si se beneficia, pois em caso de publicação de artigos, o grupo PET também se faz presente. Com isso, o objetivo é fazer com que a “cultura da pesquisa” se estabeleça dentro do grupo como um todo, possibilitando integração com mebros internos e externos do grupo PET para a realização de pesquisa e troca de experiências e informações. Um outro aspecto que será o objetivo será a publicação de artigos em revistas e eventos cíentificos; |
|  | | |

# MANUAL DE USO

## HARDWARE

## SOFTWARE

Para a utilização do software, é necessário ter o *Python 3* e o a biblioteca *pygame* instaladas.

Sempre para realizar a execução do programe, execute o arquivo *main.py.*

Dentro desse arquivo se encontram 3 váriaveis importantes:

***mazeRows*** 🡪 Define o número de linhas

***mazeColumns*** 🡪 Define o número de colunas

***mesh*** 🡪 Define o tipo da malha (pode ter os valores "triangle" ou “square”)

A execução acontece sempre da seguinte forma:

**Parte 1: Construção da Malha**

**1°** Abre-se uma tela com uma malha (de triangulos ou de quadrados)

**2°** Clique com o botão direito em cima de um trinagulo/quadrado, e selecione qual item deve estar ali.

**3°** Construa a malha como desejar, e ao final, aperte a tecla ENTER.

**OBS:** Só é possível colocar um agente e um objetivo. Caso não seja definido nenhum, o agente começa por padrão em (0,0) e o objetivo em (2, 2).

**Parte 2: Programação (Caso esteja realizando pesquisa, essa parte não irá existir)**

**1°** Clique e arraste os blocos para a tela branca, construindo o programa.  
**2°** Para excluir algum bloco, selecione ele, e aperte a tecla DELETE.

**3°** Ao concluir a programação, aperte o botão: Compilar (verdinho no canto esquerdo inferior);

**4°** Se tudo ocorreu bem (i.e. se sua programação está correta), irá para a perte 3, caso contrário, uma mensagem de erro será mostrada no console.

**Parte 3: Execução**

**1°** Essa parte é automática, pois o agente estará realizando o que foi programado.

**2°** Cada item é identificado por uma cor, que foi colocado na parte 1.

**3°** Ao fim da execução, feche o programa.

## Software voltado para a pesquisa

Para a utilização voltado para pesquisa, acesse a pasta *simulatorToSearch.*

## **Adicionando um novo algoritmo para os agentes**

Para adicionar um novo algortimo, cria uma nova classe, e tenha um método *do* dentro dela, que será utilizado em uma espécie de prolimorfismo pelo agente. Esse método deve retorna uma tupla com a direção do movimento (“N”, “S”, “L”, “O”) e o valor booleano se aquele foi o ultimo “passo” do algoritmo. Uitlize como inspiração o algoritmo LRTA, da classe ***lrta***. Em seguida, vá na classe *agent*, importe o novo algoritmo, e coloque ele dentro dos planos (do plano e da biblioteca.

Dessa maneira, o agente irá realizar o algoritmo criado. Caso necessário, vá na classe *model*, e lá altere método *Go* e *Do.* O *Go* é referente a ações de movimento, e o *Do* a ações de ação em algum bloco.

Para realizar ações nos blocos (como nos casos de bloco ativáveis), chame o método do agente *actionDo(),* passando como parametro o local do bloco onde a ação deve ser realizada.

## Software voltado para o ensino

Para a utilização voltada para o ensino, acesse a pasta *ProgrammingToTeach*.

## **Adicionando novos blocos a programação**

Preste atenção especial nos comentários dentro do código, isso facilitará em cada etapa. Para adicionar novos blocos para a programação (algum outro tipo de operação, ou função específica, além dos que já encontram), é necessário realizar alguns passos específicos:

1. Se necessário, crie uma nova “Classe” para aparecer dentro do menu lateral:
   1. Para tal, basta criar uma nova classe no python, utilizando a classe *Button* como classe pai (utilize como exemplo as classes já criadas, como por exemplo a *MathClass*);
   2. Em seguida, adicione essa nova classe dentro do *menu*, da mesma maneira que as demais.
2. Dentro da “Classe” que deseja que o novo bloco pertença:
   1. Crie uma nova intância do objeto *ObjectThing,* fornecendo como parametro as particularidades do novo bloco (utilize como exemplo os que já estão criados, como o bloco *constante* – self.cte – da classe MathClass). Adicione dentro da lista self.things, como os demais.
3. Se necessário, atualize os outros blocos para que permitam receber os encaixes desse novo Bloco, e que ele permita receber os outros.
4. Dentro da classe *main*:
   1. Dentro do método *generateCode*:
      1. Atualize ele para que o código gerado com esse novo bloco fique em conformidade com um código python verdadeiro. Tome especial cuidado com a identação (no próprio método já há uma maneira de realizar a identação, preste atenção a ele);
   2. Dentro do método *run*:
      1. Vá até a parte de sugestão de encaixe (que fica dentro da parte de Drag and Drop), e realize a atualização das partes necessárias para que ele sugira esse novo bloco. (necessário apenas se ele tiver alguma maneira de encaixe diferente, como blocos internos, aninhados, etc...);
   3. Se foi criado um novo tipo de encaixe, vá até a classe *ObjectUse*:
      1. Vá até o método *setPos*, realize o ajuste de como deve ser feito o drag and drop daquele encaixe (use como base os outros que já estão presentes ali).

## **Colocando o aprendizado no robo físico**

Crie uma pasta chamada *RoboFun* dentro da área de trabalho do Raspberry Pi do robo.

Em seguida Copie em um pendrive a pasta *API-connection,* conecte ele no raspberry, e cole essa pasta na pasta RoboFun (criada anteriormente).

Para executar, utilize o arquivo *mainApi.py.*

Passo a passo para a execução:

**1°** Rode o programa de programação em blocos no computador, e crie o código, tenha certeza que eesteja correto e funcionando.

**2°** Ao clicar no botão *compilar*, um novo arquivo foi gerado na pasta, chamado *codeBlock.py*

**3°** Copie esse arquivo para um pendrive, e passe para o RaspBerry Pi.

**4°** Cole o arquivo dentro da pasta *API-connection.*

**5°** Ajuste o robo como necessário, e execute o main*Api.py.*

***OBS:*** Pode ser que seja necessário fazer ajustes nos valores (nas quantidades de se mover e girar). Caso necessário, ajuste dentro desse mesmo arquivo mainApi.