

REGRAS - HACKATHON ESPACIAL - SETEMBRO/2020

Agência Espacial Brasileira - AEB
Programa de Desenvolvimento de Competências
Hackathon Espacial AEB/OBR/OBSAT/MCTI na SBPC jovem 2020
Apoio: CVT-Espacial Augusto Severo, UFSCar, IFRN, OBR, Funcate, SBPC, UFRN, UNICAMP
<https://obsat.org.br/hackathonespacial/>

(Este manual é baseado no manual de regras da OBR 2020 - Simulação)

Objetivo do desafio

Um rover lunar tem como missão levar amostras coletadas até a base científica da AEB na Lua sem interferência humana. A missão de sua equipe é programar o robô autônomo para que ele seja capaz de seguir a linha preta que indica o caminho seguro no terreno irregular de nossa aventura. Para tanto, será preciso realizar medidas do chão e do ambiente através de sensores, e construir a lógica de programação que faça o robô acompanhar a linha, superar falhas e detritos no percurso, desviar de obstáculos e, finalmente, alcançar a base científica no alto de uma rampa. Ao alcançar a base científica, o robô deverá indicar que atingiu seu objetivo emitindo um sinal sonoro ou luminoso.

Ambiente de projeto

O ambiente utilizado para simulação, programação e responsável por fazer a gerência da competição será o simulador **sBotics**. O simulador deverá ser **necessariamente configurado** conforme explicado a seguir.

É possível fazer o download do simulador através do site <http://weduc.natalnet.br/sbotics> para os sistemas Windows (8+), Linux (64 bits) e Mac OS.

Cenário do desafio

O cenário representado no simulador possui uma superfície cinza, com textura da Lua. Esta superfície representa um terreno irregular. As linhas pretas, existirão em toda o cenário. A capacidade do robô seguir um caminho desconhecido faz parte do desafio. As linhas representam uma passagem segura, conhecida que podem estar obstruídas por obstáculos, gaps ou ter redutores de velocidade. As linhas podem ainda fazer curvas grandes, pequenas, curvas em 90°, retas, zigue-zague, círculos, entre outras formas.

A rampa de acesso à base científica deve possuir linha preta, sendo possível contar também com um gap e/ou redutores de velocidade, além de possuir anteparos para evitar a queda dos robôs. A altura da base científica deverá gerar uma inclinação na rampa de 15 a 25 graus.

A área superior, chamada de base científica, não possui linhas no chão, apenas uma faixa cinza na sua entrada para indicar o término da rampa. A Figura 1 mostra um exemplo da disposição dos elementos mencionados.

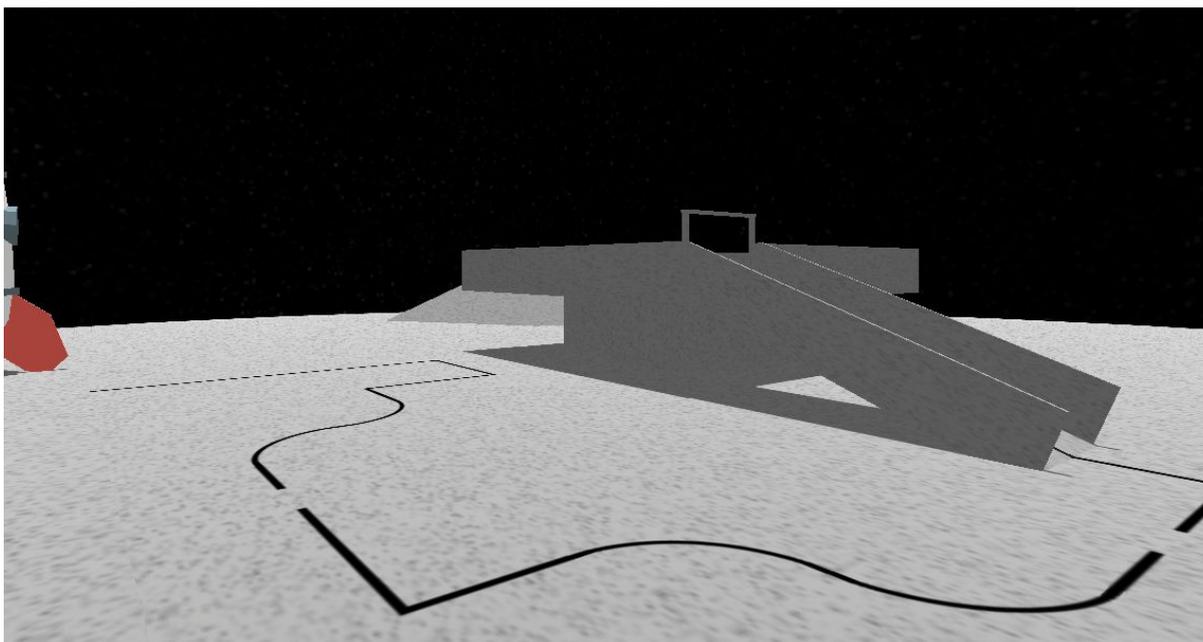


Figura 1: Ambiente da arena oficial do Hackathon.

Percurso oficial



IMPORTANTE

A organização somente aceitará envio de vídeos feitos no percurso oficial do Hackathon. Para tanto é preciso configurar o **sBotics** com o percurso definido abaixo. Dessa forma, para realizar este ajuste, siga o seguinte procedimento no **sBotics**:

1. **Acesse a opção que define a arena da competição:**

Ajustes ⇒ Arena ⇒ Importar



Figura 2: Tela que deverá ser acessada para configurar o percurso oficial do Hackathon.

2. Em *Código da arena* , copie e cole o texto abaixo, integralmente:

```
0000r0000,0010r1000,0020c0000,1020rm090,2020c1090,2010r4000,2000c0270,3000r1090,4000c0180,4010r5000,4020c0000,5020c1180,3,HoraDoDia: 11,BoolTpFim: true,BoolTSala: 1,BoolPrgso: true,BoolSlnha: true,BoolEndCd: true,VtmsVivas: 0,VtmsMrtas: 0,RobosPerm: 000001,BoolMrcdr: false,BoolShowM: false,Descricao: Hackaton Espacial,TempoMxmo: 3
```

3. Clique nas setas verdes >>

Ao realizar a importação, você deverá visualizar a seguinte arena:



Figura 3: Arena oficial do Hackathon Espacial, que deve ser usada, obrigatoriamente, por todas as equipes.

O desafio e vídeo deverão ser feitos, necessariamente, sob este percurso. Percursos diferentes não serão considerados, e a equipe não será classificada.

Treino

Caso a equipe deseje treinar diferentes situações, é possível construir arenas de forma manual ou através de um sorteio aleatório de arenas. Para tanto utilize a opção do **sBots**:

Ajustes ⇒ Arena ⇒ Gerar nova Arena

Importante lembrar, que após os treinos, os vídeos, testes e envio da documentação deverão ser feitos com base no desenvolvimento feito na arena oficial, que deve ser importada, conforme explicado na seção anterior.

Programação



PROGRAMAÇÃO

Se você tem pouca experiência em programação, ou nunca programou, fique tranquilo, o sBotics e o Hackathon espacial vão ajudar você a aprender!

A programação do robô pode ser feita em várias linguagens, e até mesmo em português, através da linguagem REDUC. Se mesmo assim você estiver em dúvida, você pode programar em blocos! É como montar LEGO®, mas você usa os blocos para montar a lógica de funcionamento do seu robô! O texto e a figura a seguir mostram exemplos de programas desenvolvidos para um robô se mover sem parar (bloco enquanto) até que o sensor de objetos detecta um obstáculo à sua frente.

```
inicio
  frente(10)
  enquanto (ultra(1) > 10) farei{
    frente(1)
  }
  parar()
fim
```

Figura 4: Programa para o robô parar na frente de um obstáculo, desenvolvido em REDUC.

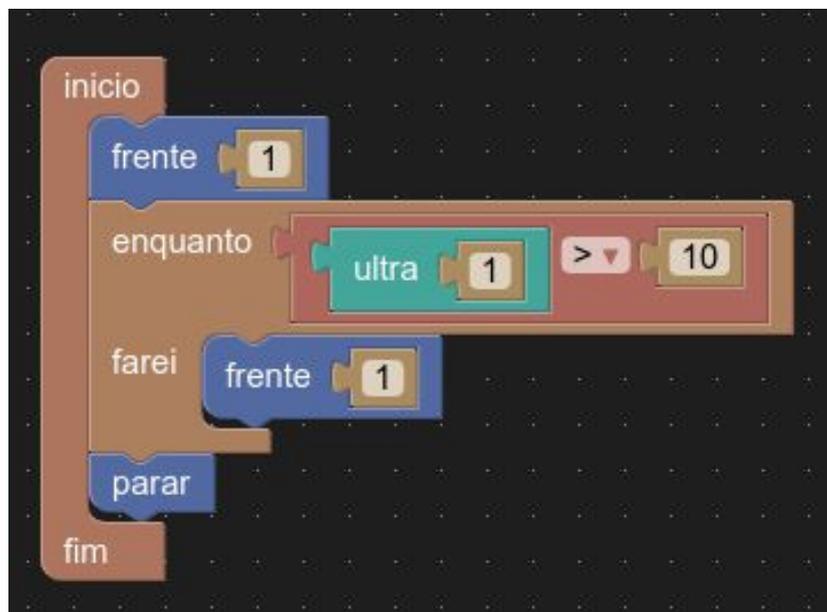


Figura 5: Programa para o robô parar na frente de um obstáculo, desenvolvido em blocos.



MANUAL / TUTORIAL

O **sBotics** tem um manual completo para te ajudar. Além disso, o manual também tem um tutorial, ensinando os conceitos básicos de uso do simulador e sua programação passo a passo! Se precisar de ajuda, acesse o manual e tutorial do **sBotics**. Caso não consiga esclarecer sua dúvida pelo manual, procure ajuda no canal do Discord do **sBotics** (<https://bit.ly/sboticsdiscord>).

Dica: Lembre sempre de salvar seu programa após qualquer alteração, compilar e enviar para o robô antes de clicar em “Testar rotina”.



DICA / EXEMPLO

Você vai perceber que parte importante do desenvolvimento do código do seu robô é tomar decisões com base na leitura de sensores. Para analisar as medidas dos sensores, você pode usar a opção escrever e visualizar os dados no console (ícone acima). O exemplo a seguir mostra um programa em blocos, e o mesmo programa em R-Educ com a seguinte função: **o robô irá andar para frente, com velocidade 20 continuamente enquanto o sensor de distância ultrasônico fizer uma medida maior que 10. Se a medida chegar a 10, o robô para.**



Figura 6: Robô parado na frente do obstáculo ao detectar a distância de 10.

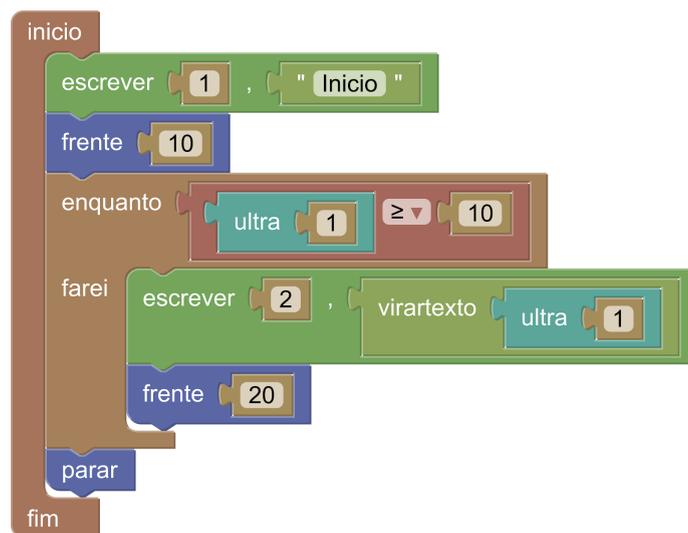


Figura 7: Programa de teste que faz o robô andar para **frente** com potência 20 e mostra a leitura do sensor na tela (**Console**) através do bloco **escrever enquanto** a distância é menor que 10.

```

inicio
  escrever(1,"Inicio")
  frente(10)
  enquanto (ultra(1) >= 10) farei{
    escrever(2,virartexto(ultra(1)))
    frente(20)
  }

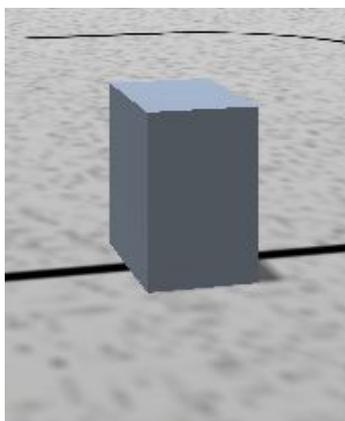
  parar()
fim

```

Figura 8: Código do programa exemplo.

Componentes do desafio

Obstáculos



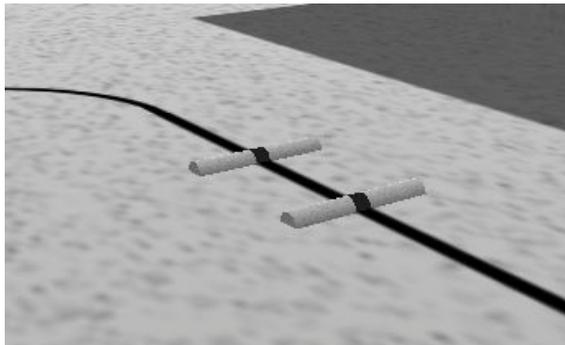
Dentro da área de percurso podem existir obstáculos. Eles são barreiras intransponíveis que forçam o robô a desviar, saindo do caminho traçado pela linha preta durante alguns instantes.

Ao desviar de um obstáculo, **o robô deve retornar para a linha logo em seguida ao obstáculo desviado em até 25 segundos para obter sucesso**. Não será permitido ao robô seguir por outra linha da arena nem a mesma linha caso ela já tenha mudado de direção após o obstáculo. Caso o robô não consiga retornar à linha no tempo máximo de 25 segundos, será considerada **FALHA DE PROGRESSO**, forçando o robô a reiniciar o seu percurso.

Os obstáculos podem variar de cor e tamanho, seguindo os modelos pré-estabelecidos no simulador. Podem estar posicionados em qualquer local da superfície inferior da arena onde se apresentem linhas retas, os obstáculos não podem ser

posicionados sobre curvas ou interseções. Os obstáculos podem estar presos, ou não, ao piso da arena. Se o robô empurrar ou deslocar algum obstáculo, em qualquer momento durante a execução da rodada, será considerada FALHA DE PROGRESSO. O obstáculo voltará para a posição correta após a FALHA DE PROGRESSO.

Redutores de Velocidade

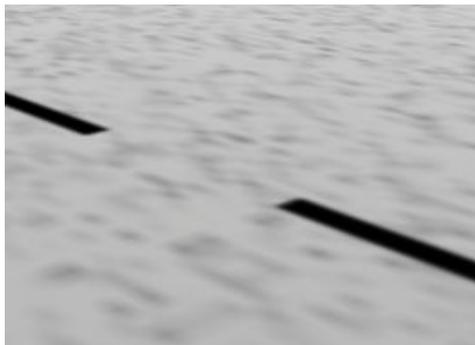


Redutores de velocidade, que simulam terrenos sinuosos, poderão estar em posição transversal à linha, sendo roliços. Têm a mesma cor do chão e estão cobertos com a fita preta que indica o percurso.

Redutores PODEM ser alocados na área de percurso e rampa, podendo formar qualquer ângulo com a linha.

Serão considerados superados quando o robô ultrapassar completamente o redutor de velocidade (todas as partes do robô).

Gap

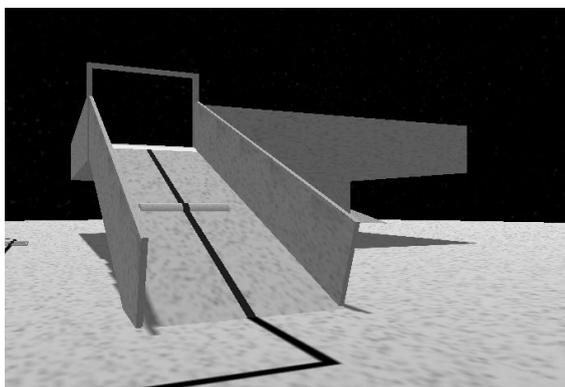


Os gaps simulam situações onde o robô não consegue distinguir o caminho a ser seguido. Isto é feito com uma descontinuidade na linha preta.

Os gaps aparecem sempre em linhas retas, podem existir na rampa e na área de percurso, e possuem um tamanho fixo.

Serão considerados superados quando o robô voltar a seguir a linha à frente e mais da metade do robô encontrar-se nesta linha após o gap.

Rampa



A rampa de acesso à base científica deve possuir linha preta, sendo possível contar também com um gap e/ou redutor de velocidade, além de possuir anteparos para evitar a queda dos robôs.

A altura da base científica deverá gerar uma inclinação na rampa de 15 a 25 graus.

A rampa é considerada superada quando o robô ultrapassa completamente a faixa cinza na entrada da área superior.

A competição

O objetivo da missão é levar o robô até a base científica, no alto da rampa, percorrendo o caminho marcado pela linha preta e superando obstáculos e dificuldades. Ao alcançar a base, o robô precisa emitir um sinal sonoro ou luminoso indicando a conclusão da missão. Lembre que o robô possui sensores em sua parte frontal, que ajudam a detectar a linha. Utilize estes sensores para que seu programa controle o robô para seguir a linha.

Cada rodada tem tempo máximo de 5 minutos (300 segundos) para cumprir a tarefa. Durante a rodada, são permitidas até 3 tentativas no percurso da linha e 3 tentativas para subir a rampa até a base científica. A cada nova tentativa no percurso, o robô é reiniciado no início do percurso. Após 3 tentativas no percurso ou a cada tentativa de subir a rampa, o robô é reiniciado ao pé da rampa. Caso o robô não sinalize sua chegada à base científica ou não alcance a base científica ao final de 5 minutos (300 segundos) de rodada, será automaticamente declarado FIM DE RODADA, salvando-se a pontuação obtida até então. Testes quanto ao sistema de pontuação podem ser realizados no simulador a qualquer momento pelos usuários.

A equipe pode treinar quantas rodadas quiser na arena de competição e deve enviar à organização a gravação de tela da rodada na qual seu robô obteve o melhor desempenho.

É imprescindível para a participação, que ao menos um membro da equipe possua um computador compatível com o simulador e acesso a Internet no dia do evento. Não havendo o acesso a esses requisitos, no dia do evento, a equipe será considerada desistente.

Pontuação

Os robôs poderão receber a seguinte pontuação ao superar elementos de dificuldade durante o percurso:

Pontuação sobre os elementos de percurso:	Pontuação
Desviar com sucesso de cada obstáculo bloqueando o caminho	15 pontos
Ultrapassar cada reductor de velocidade	5 pontos
Vencer adequadamente uma situação de gap na linha	10 pontos
Concluir o percurso de linha na primeira tentativa	60 pontos
Concluir o percurso de linha na segunda tentativa	40 pontos
Concluir o percurso de linha na terceira tentativa	20 pontos

Cada elemento de arena (gap, redutores de velocidade, obstáculos e rampa) só será pontuado uma vez por cada direção do percurso. Os pontos não são cumulativos por tentativas subsequentes durante o percurso.

Se a rampa for finalizada, a equipe recebe a seguinte pontuação:

Pontuação sobre a rampa:	Pontuação
Passagem completa pela rampa na primeira tentativa	30 pontos
Passagem completa pela rampa na segunda tentativa	20 pontos
Passagem completa pela rampa na terceira tentativa	10 pontos

A plataforma de entrada faz parte da rampa, portanto o robô deve entrar na base científica para a passagem da rampa ser pontuada.

Já na base científica, a equipe receberá a seguinte pontuação atribuída pela banca avaliadora, caso indique com sucesso sua chegada:

Pontuação para a base científica	Pontuação
Emissão de sinal sonoro	60 pontos
Emissão de sinal luminoso	60 pontos

Considera-se que o robô alcançou a base científica com sucesso apenas após a emissão do sinal sonoro ou luminoso. Após a sinalização, NÃO é necessário encerrar o programa. Nenhum ponto será atribuído a essa etapa se o robô não sinalizar sua chegada à base científica.

Falha de Progresso e Tentativas

Uma FALHA DE PROGRESSO caracteriza-se quando:

- O robô permanecer parado no mesmo lugar por 10 segundos;
- O robô subir a rampa sem seguir linha;
- O robô perder a linha preta por mais de 10 segundos;
- O robô demorar mais de 25 segundos para desviar de um obstáculo;
- O robô se perder da linha e passar a seguir uma outra linha paralela ou outra que não seja a linha da sua frente;
- O robô não conseguir contornar o obstáculo com sucesso, derrubá-lo ou empurrá-lo;
- O robô não ser capaz de passar pela entrada da base científica.

Para cada FALHA DE PROGRESSO, o robô deverá recomeçar o percurso ou rampa em que estiver atuando, considerando este reinício uma NOVA TENTATIVA. Cada NOVA TENTATIVA só pode ser solicitada a cada 20 segundos. A equipe pode ainda optar por pular o percurso antes de finalizar as 3 tentativas. Isso implicará em uma penalidade de 1 minuto que será acrescido no tempo final.

O tempo máximo da rodada, mesmo com penalidades, será de 5 minutos.

A equipe poderá solicitar o FIM DA RODADA a qualquer momento, neste caso todos os pontos conquistados pela equipe serão considerados, mas seu tempo de prova, para efeito de desempate, será o tempo máximo da prova (5 minutos).

Violações

É responsabilidade das equipes a verificação dos pré-requisitos para participação. Dessa forma, não serão oferecidos tempo extra às equipes que tiverem problemas em seus computadores ou conexão. Caso seja detectada alguma tentativa de burlar, danificar o simulador ou sistema web a equipe será DESCLASSIFICADA do Hackathon.

Não é permitido realizar um mapeamento externo do percurso, programando temporalmente no robô ações baseadas em informações do humano, de modo que seu programa funcione somente para aquela pista. Ou seja, o robô deve ser capaz de identificar as situações (linhas, obstáculos, detritos) de forma autônoma e tomar a decisão de execução com base nesse conhecimento. Assim o robô deve ser capaz de completar trajetos com outros desenhos. Caso seja detectado um comportamento de mapeamento no programa do robô, a equipe será DESCLASSIFICADA do Hackathon.

Envio do vídeo para os avaliadores

A equipe pode treinar quantas rodadas quiser na arena de competição e deve enviar à organização a captura de tela do vídeo da rodada na qual seu robô obteve o melhor desempenho. Atente-se ao prazo para envio do seu vídeo e outros documentos **(06/09/2020 - 11h (Horário de Brasília))**. Caso todos os (4) itens a seguir não sejam enviados, ou haja envio incompleto de documentos, a equipe será DESCLASSIFICADA do Hackathon, e qualquer parte de material enviado será desconsiderada para avaliação.



IMPORTANTE

Itens a serem enviados para cumprimento do Hackathon **(06/09/2020 - 11h (Horário de Brasília))**:

1. Arquivo PDF com o código do programa desenvolvido pela equipe e outras informações, explicações, imagens que a equipe considerar pertinentes;
2. Vídeo de 5 minutos, no máximo, com a gravação de todo o percurso do robô durante a simulação, garantindo a visibilidade e leitura da pontuação, tempo (na parte superior da tela) e dos movimentos do robô;
3. Vídeo de 2 minutos, no máximo, com uma apresentação da equipe, falando sobre a experiência no Hackathon, se foi interessante, dificuldades encontradas e superadas e outras informações que a equipe desejar informar;
4. Termo de uso de imagem com o nome completo e assinatura de todos os membros da equipe.

O material deverá ser enviado, necessariamente, pelo site do evento:

<https://obsat.org.br/hackathonespacial/>

Critérios de avaliação

Durante a avaliação, os seguintes critérios serão considerados:

1. Apresentação da solução
 - a. Problema
 - b. Solução
 - c. Detalhes de viabilização da ideia
 - d. Pontuação calculada automaticamente pelo simulador e tempo que o robô levou para cumprir a missão (quanto mais rápido, maior a pontuação)
2. Critérios adicionais
 - a. Criatividade e inovação
 - b. Impacto social
 - c. Viabilidade de implementação

As notas atribuídas serão:

- Solução em desenvolvimento
- Solução com potencial
- Solução funcional
- Solução de excelência

Observação importante:

Os critérios apresentados nesta seção estão mais detalhados do que no Edital de abertura de inscrições para o Hackathon. Para efeitos de classificação / cálculo da pontuação final, deve-se considerar os critérios apresentados nesta seção.

Premiação

Os vencedores terão uma matéria jornalística sobre o seu projeto publicada no website e/ou redes sociais da AEB, além de um certificado de participação e resultado da OBSAT. Outros prêmios poderão ser oferecidos / disponibilizados conforme disponibilidade de recursos da organização do evento.

Contato e suporte

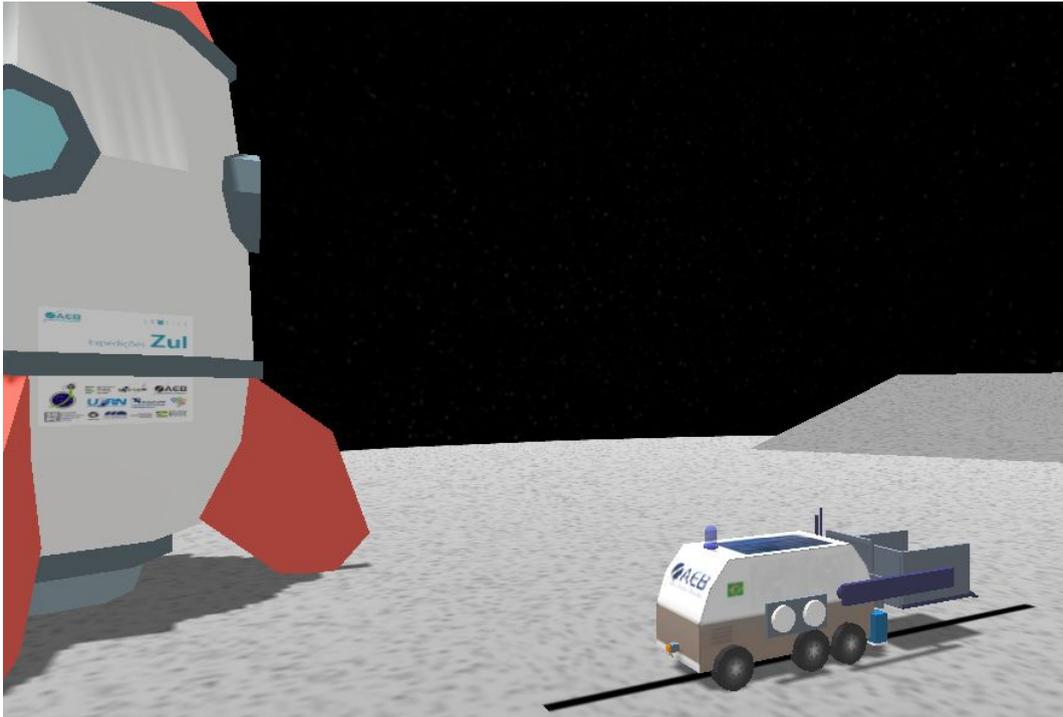
O principal canal de discussão e suporte do Hackathon será o Discord do **sBotics**. Para fazer perguntas e interagir, basta acessar este canal: <https://bit.ly/sboticsdiscord>.

Fair Play

Como a competição será realizada utilizando um mesmo simulador para todas as equipes, as pontuações de percurso serão as atribuídas diretamente pelo simulador e a pontuação referente a sinalização será atribuída pelos avaliadores, que em seguida farão uma verificação da execução geral através do vídeo enviado pelas equipes, evitando erros na identificação das pontuações.

Participe da competição de forma limpa, saudável e ética. Ajude seus colegas e outras equipes a superarem seus limites. Divirta-se durante toda a competição e colabore para que todos os demais participantes (avaliadores, amigos, familiares, etc) se divirtam também. É esperado que todas as equipes estejam imbuídas do espírito do “fair play”.

Bom Hackathon a todos! Divirtam-se!!!



Organização e Realização

