

Plano de Aula – Avanço e Retrocesso

Tema

Programação básica de movimento com o BaseBot VEX IQ

Duração da Aula

1 aula (40 a 50 minutos)

Componentes Curriculares Envolvidos

Tecnologia, Programação, Matemática, Engenharia

Turmas Indicadas

Ensino Fundamental I e II (3º ao 6º ano)

Objetivos da Aula

- Programar o robô para mover-se para frente e para trás.
 - Explorar o uso de parâmetros de direção e distância no VEXcode IQ.
 - Desenvolver lógica de sequência de comandos.
 - Trabalhar testes, ajustes e depuração de código.
-

Competências e Habilidades da BNCC

- **EF05CI04:** Observar o uso de programação para automatizar tarefas.
 - **EF05MA20:** Relacionar medidas e movimentos com números.
 - **Competência Geral 5:** Compreender e criar soluções usando tecnologias digitais.
 - **Competência Geral 2:** Exercitar raciocínio lógico e pensamento crítico.
-



Materiais Necessários

- Kit VEX IQ com BaseBot.
 - VEXcode IQ (tablet ou computador).
 - Área livre para testes (com no mínimo 60 cm).
-

□ Etapas e Desenvolvimento da Aula (Passo a Passo)

1 Introdução (10 min)

- Apresentar o desafio: programar o BaseBot para avançar e depois recuar.
- Conversar sobre o uso de comandos sequenciais em programação.

2 Programação Inicial (15 min)

- Criar o projeto com um bloco [Drive for] configurado para 12 polegadas (30 cm).
- Executar o código e observar o movimento para frente.

3 Adicionar Retrocesso (15 min)

- Adicionar outro bloco [Drive for] configurado para “reverse”.
- Testar o projeto e observar como a mudança no parâmetro altera o comportamento.

4 Ajustes e Discussão (10 min)

- Alterar valores de distância e ordem dos blocos para explorar diferentes movimentos.
 - Discutir como sequências afetam o comportamento do robô.
-



Subindo de Nível

- **Vá além!:** Aumentar distâncias para explorar diferentes percursos.
 - **Remix:** Alterar a ordem e observar como muda a sequência de movimentos.
-



Conteúdos Trabalhados

- Sequências de programação.
- Movimento controlado por parâmetros.
- Testes e ajustes em projetos automatizados.

Dicas para o Professor

- Mostrar como o bloco [Definir velocidade de acionamento] pode variar o movimento.
- Reforçar a importância de medir distâncias para precisão no teste.
- Incentivar previsões antes de rodar o código para reforçar raciocínio lógico.

Discussões e Conclusões

- Como a alteração dos parâmetros mudou o movimento?
- Por que a ordem dos blocos é importante em programação?
- O que aprendemos sobre controle e precisão de movimentos?

Interdisciplinaridade

- **Tecnologia:** Programação e automação.
- **Matemática:** Medidas de distância e lógica de sequência.
- **Engenharia:** Ajustes de design para movimento eficiente.

Avaliação Formativa

- Participação na criação e testes do projeto.
- Capacidade de ajustar parâmetros para alcançar objetivos.
- Compreensão da relação entre código e movimento físico.

Dicas Pedagógicas

- Para turmas mais novas, usar apenas o movimento para frente e depois adicionar o retrocesso.
 - Para turmas avançadas, propor movimentos mais complexos combinando ângulos e loops.
-

Resultados Esperados

- Alunos compreendendo programação básica de movimento.
- Desenvolvimento de lógica sequencial e ajustes.
- Engajamento prático com robótica aplicada.