

Plano de Aula – Avanço e Retrocesso

Tema:

Controle de movimentos diretos e reversos com o robô VEX EXP utilizando comandos de programação sequencial

Duração da aula:

1 aula de 50 minutos

Componentes curriculares envolvidos:

- Matemática
 - Ciências
 - Tecnologia
 - Pensamento Computacional
 - Educação Digital
-

Turmas indicadas:

4º ao 5º ano do Ensino Fundamental

Objetivos da aula:

- Compreender os comandos de direção direta e inversa em um robô programado.
 - Programar deslocamentos simples com controle de tempo, distância e rotação.
 - Estimular o raciocínio lógico e a compreensão da relação entre comando e movimento.
 - Desenvolver noções espaciais de orientação e percurso.
-

Competências da BNCC:

- Competência 1: Conhecimento
 - Competência 2: Pensamento científico, crítico e criativo
 - Competência 5: Cultura digital
 - Competência 6: Trabalho e projeto de vida
 - Competência 7: Argumentação
-

Habilidades da BNCC:

- (EF05MA07) Resolver problemas com deslocamentos e sentido de movimento.
 - (EF04EM04) Programar comandos básicos de movimentação com precisão.
 - (EF05MA20) Identificar padrões e regularidades em sequências lógicas.
 - (EF04CI01) Observar funcionamento de sistemas com deslocamento automatizado.
-

Materiais necessários:

- Kit VEX EXP com robô BaseBot montado
 - Dispositivo com VEXcode EXP (modo Blocks ou Python)
 - Fita adesiva para demarcar áreas no chão (ponto de partida, chegada e obstáculos)
 - Objetos leves para contornar ou simular barreiras
 - Papel e lápis para anotações
-

Etapas e Desenvolvimento da Aula (Passo a Passo):

1. Introdução ao conceito de movimento programado (5 min)

- Apresente os comandos principais:
 - `drive_for(FORWARD, 300 mm)`
 - `drive_for(REVERSE, 300 mm)`
- Demonstre na prática o que acontece com cada direção no robô.

2. Desafio simples de percurso (10 min)

- Proponha um trajeto:
 - “Avançar até a fita, depois voltar ao ponto inicial.”
- Alunos preveem quanto o robô deve andar (medem com régua, estimam distância).
- Criam um código com os dois comandos (avanço e retrocesso) ajustando valores.

3. Programação e testes (15 min)

- Alunos escrevem os blocos e testam o robô no percurso delimitado.
- Ajustam distância, direção e tempo se necessário.
- Podem usar medidas diferentes (em mm, graus ou tempo, dependendo do modo usado).

4. Variedade de rotas (10 min)

- Proponha novos desafios com obstáculos:
 - Avançar, contornar, recuar e finalizar.
- Adicione `turn_for` ou `turn_to_heading` para curvas, se desejado.
- Os alunos podem desenhar o mapa do percurso antes de programar.

5. Discussão coletiva e revisão (5 min)

- Cada grupo compartilha o que funcionou ou não.
 - Discutem:
 - “O que mudou quando usamos `REVERSE`?”
 - “Por que é importante testar várias vezes?”
-



Subindo de Nível:

- Percurso com tempo cronometrado
 - Avançar e recuar com sensores (evitar obstáculos automaticamente)
 - Programar com comandos de `if` ou `while` baseados em sensores (para turmas avançadas)
-



Conteúdos trabalhados:

- Direção e deslocamento programado
 - Relação entre valor e movimento (quantidade/duração)
 - Sequência lógica de ações
 - Noção espacial e correção de rota
 - Testes e ajustes
-



Dicas para o professor:

- Use uma régua e fita métrica como ferramenta de apoio para estimar distâncias.
 - Estimule os alunos a anotar comandos e resultados para comparar tentativas.
 - Proponha desafios colaborativos: um grupo cria o trajeto, o outro programa.
-



Discussões e conclusões:

- Como o robô sabe até onde ir?
 - Qual a vantagem de usar REVERSE em vez de turn + FORWARD?
 - Por que testar várias vezes melhora a precisão?
-

Interdisciplinaridade:

- Matemática: distância, direção, estimativas
 - Tecnologia: controle programado
 - Ciências: força, movimento, deslocamento
 - Educação Digital: linguagem de programação, sequenciamento
-

Avaliação formativa:

- Precisão nos comandos de movimento
 - Clareza na lógica do código
 - Capacidade de testar e ajustar o programa
 - Cooperação durante a execução em grupo
-

Resultados esperados:

- Domínio dos comandos básicos de avanço e retrocesso
- Capacidade de planejar percursos e ajustar deslocamentos
- Raciocínio lógico vinculado a orientação espacial
- Aumento da autonomia na construção de códigos simples