

**Desenvolvimento de robô autônomo:**

Caroline Feitosa dos Santos

David Macedo Lima

Elis Vieira Weiss

Geison de Oliveira Lemos Ferreira

Igor de Eça Almeida

Sarah Alves Borges

Wendel dos Santos Coelho

Outubro, 2024

Vitória da Conquista, Bahia

1. Objetivo do Projeto

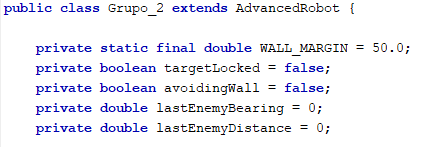
O robô desenvolvido pelo grupo 2 foi projeto para ser ágil e evitar paredes e tiros inimigos enquanto mantém o radar ativo para detectar e atacar oponentes. Ele foi programado para utilizar movimentação constante para minimizar a chance de ser atingida ao mesmo tempo que otimiza a precisão dos tiros.

1. Lógica Implementada

* Movimentação Constante: O robô deve sempre se mover, independentemente da situação. Isso diminui a probabilidade de ser atingido, pois um alvo em movimento dificulta prever a trajetória.
* Evitar Paredes: Um sistema foi implementado para garantir que o robô evite colisões por conta do dano sofrido ao colidir com as paredes.
* Radar Dinâmico: O radar do robô foi projetado para girar continuamente até o momento que detecta seu alvo. Ao localizar um inimigo, o radar modifica sua estratégia para fazer pequenas varreduras ao redor do alvo, permitindo um monitoramento contínuo.
* Tiro Ajustado: A potência do tiro é ajustada dinamicamente com base na distância do inimigo, maximizando a eficiência e o dano causado, já que a velocidade do tiro é influenciada pela potência.

1. Estrutura do Código

* Declaração de Variáveis

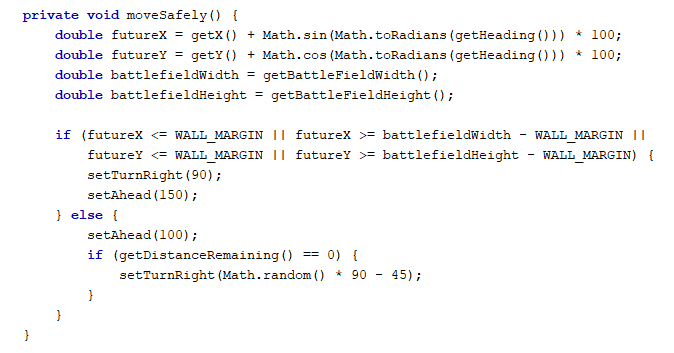


* + WALL\_MARGIN: Define uma margem de 50 pixels a partir das paredes do campo de batalha, usada para evitar colisões.
  + targetLocked: Indica se o robô tem um alvo travado. Se true, o radar se concentrará nesse inimigo.
  + avoidingWall: Indica se o robô está atualmente evitando uma parede. Se true, o robô realiza manobras específicas para evitar colisões.
  + lastEnemyBearing: Armazena o ângulo relativo do último inimigo detectado.
  + lastEnemyDistance: Armazena a distância do último inimigo detectado.
* Método run()

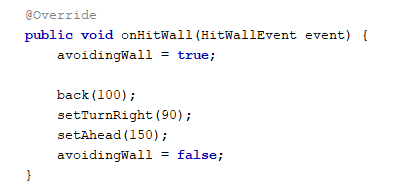
A computer code with text

Description automatically generated

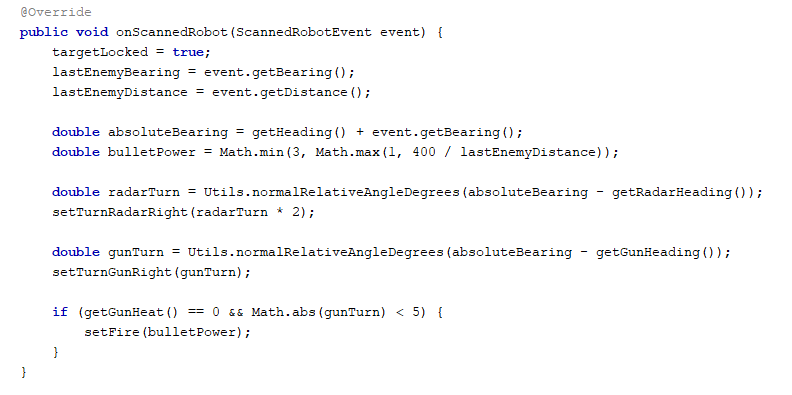
* setColors: Define as cores do robô, onde o corpo será azul, as esteiras serão cinza e o canhão será branco. O radar é configurado para ser amarelo.
* setAdjustRadarForGunTurn(true): Permite que o radar gire independentemente do canhão, aumentando a eficiência de detecção.
* setAdjustRadarForRobotTurn(true): Faz com que o radar gire separadamente do movimento do robô.
* setAdjustGunForRobotTurn(true): Permite que o canhão gire de forma independente do movimento do robô, possibilitando que o robô se mova enquanto ajusta a mira.
* while (true): Loop infinito que mantém o robô funcionando até o fim da batalha. No loop:
  + Se targetLocked for false, o radar gira 360 graus para procurar inimigos.
  + Se avoidingWall for false, o robô se move usando o método moveSafely() para evitar colisões.
  + execute(): Envia todas as ações configuradas no ciclo para o motor do robô.
* Método moveSafely()



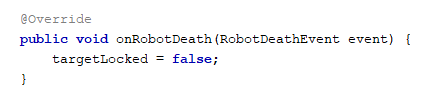
* futureX e futureY: Calculam a posição futura do robô com base na direção atual e na distância de 100 pixels.
* battlefieldWidth e battlefieldHeight: Obtenção das dimensões do campo de batalha.
* Condição if: Verifica se a posição futura do robô estará próxima das paredes. Se sim, o robô gira 90 graus para evitar a colisão e avança 150 pixels.
* else: Caso contrário, o robô continua movendo 100 pixels à frente. Se a distância restante for 0 (indicando que o movimento atual foi concluído), o robô gira em uma direção aleatória entre -45 e 45 graus.
* Evento onHitWall()



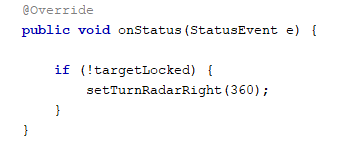
* avoidingWall = true: Define que o robô está evitando a parede.
* back(100): Recuo de 100 pixels para se afastar da parede.
* setTurnRight(90): Gira 90 graus para longe da parede.
* setAhead(150): Avança 150 pixels para garantir que o robô se afaste da parede.
* avoidingWall = false: Após a manobra, indica que o robô não está mais evitando uma parede.
* Evento onScannerRobot()



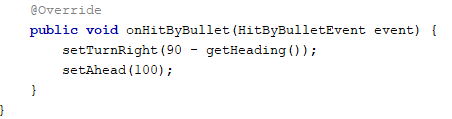
* targetLocked = true: Indica que um inimigo foi detectado.
* lastEnemyBearing e lastEnemyDistance: Armazena o ângulo e a distância do inimigo.
* absoluteBearing: Calcula o ângulo absoluto do inimigo, combinando o ângulo de rotação do robô com o ângulo relativo do inimigo.
* bulletPower: Ajusta a potência do tiro com base na distância do inimigo. Quanto mais distante, menor a potência (mínimo de 1, máximo de 3).
* radarTurn: Calcula a rotação necessária para que o radar se alinhe com o inimigo. O radar é girado mais rapidamente para se ajustar.
* gunTurn: Calcula o giro necessário para alinhar o canhão com o inimigo.
* setFire(): Atira no inimigo com a potência calculada, desde que o canhão esteja suficientemente alinhado e o resfriamento da arma esteja completo.
* Evento onRobotDeath()



* targetLocked = false: Quando o inimigo morre, o radar volta ao modo de busca, destravando o alvo.
* Evento onStatus()



* if (!targetLocked): Se nenhum inimigo estiver travado, o radar continua girando 360 graus para procurar alvos.
* Evento onHitByBullet()



* setTurnRight(90 - getHeading()): O robô gira 90 graus em relação à direção atual para mudar sua trajetória e evitar ser atingido novamente.
* setAhead(100): Move-se 100 pixels para frente após a mudança de direção.

1. Estratégias Utilizadas

* Movimentação Incessante

Manter o robô sempre em movimento reduz a chance de ser atingido por tiros inimigos. A estratégia de movimento inclui giros aleatórios para tornar os deslocamentos imprevisíveis.

* Evitação Inteligente de Paredes

O robô utiliza uma margem fixa para prever se estará perto de bater em uma parede e, assim, ajustar sua rota antes de colidir.

* Trava e Foco em Alvo

Assim que um inimigo é detectado, o radar e o canhão são ajustados rapidamente para travar no alvo, garantindo que o robô possa disparar com precisão e rapidamente.

1. Conclusão

O robô Grupo\_2 é projetado para ser ágil, evitando tanto as paredes quanto os tiros dos inimigos, enquanto mantém o radar ativo e se move constantemente. A abordagem combina evasão de obstáculos, travamento de alvo eficiente e movimentação imprevisível, aumentando suas chances de sobrevivência no campo de batalha.

Esse processo de desenvolvimento foi documentado para evidenciar as decisões estratégicas e técnicas que guiaram o projeto, visando eficiência e agilidade no combate.