



# Lógica da Computação: Shooter Language

Feito por: Rafael Evangelista Monteiro

# Introdução

Nesse projeto, foi desenvolvido um compilador para uma nova linguagem de programação chamada Shooter Language. Ela foi criada com base no Python e nas características que eu gosto de uma linguagem. Além disso, a linguagem é em Português para possibilitar que os falantes do idioma tenham mais acessibilidade à programação. Por fim, ela tem funcionalidades (Atirar e Posicionar alvos) que possibilitam o usuário fazer seu minigame e tornar a escrita de códigos mais divertida.

# Motivação



## Jogos de tiro e estratégia

Adicionar às linguagens de programação tradicionais a possibilidade de jogar minigames ou desenvolver estratégias.



## Linguagem de programação

União das características que eu gosto em uma linguagem, além de seu uso ser abrangente.



## Português como padrão

Remover a barreira da língua para o aprendizado da programação

# Funcionalidades exclusivas

## Comando "atirar"

Um comando específico para atirar em um alvo. Esse alvo estará em uma Coordenada (Uma letra seguida por um número). O usuário não pode atirar em um lugar que não tenha inimigo

## Posição do inimigo

Com o comando "posicao", o programador pode indicar as coordenadas do inimigo, além de poder mudar ao longo do programa.

## Português

- If → caso
- Else → seNao
- While → enquanto
- Def/Func → funcao
- print → mostra
- Chamada de função → chama

# Exemplos de código

## Comando de "Atirar"

O comando "Atirar" pode ser usado para simplificar funções de lógica de jogos eletrônicos:

```
posicao Inimigo1 : B5
vida_inimigo = 10
atirar B5
caso (B5 == Inimigo1) {
  vida_inimigo = vida_inimigo - 1
  mostra("Acertou")
}
```

1

## Posição do inimigo

A indicação da posição do inimigo facilitará a programação de jogos eletrônicos. Veja um exemplo:

```
posicao Inimigo1 : B5
posicao Inimigo2 : a5
```

2

3

## Português

O uso do português para a definição de comandos facilita a escrita dos programas por aqueles que não dominam o inglês:

```
caso (B5 == Inimigo1) {
  mostra("Acertou")
} seNao {
  mostra("Errou")
}

funcao esconder (Inimigo1, Inimigo2) {
  posicao Inimigo1 : Z100
  posicao Inimigo2 : Z200
}

chama esconder(Inimigo1, Inimigo2)
```



# Exemplo completo e melhorias

- Para futuras versões seria interessante que as funções atirar e posicao fossem diferentes.
- Posicao: Ao declarar uma posicao seria ideal declarar a vida do inimigo. Além disso, o compilador deveria ter CREATE na Symbol Table, não somente SET.
- Atirar: A linguagem poderia permitir atirar em qualquer posição e já tirar a vida do inimigo.
- Implementar outras funcionalidades do Python como: FOR, ARRAY, CLASSES e etc.

```
posicao Inimigo1 : B5  
posicao Inimigo2 : a5  
vida_inimigo = 3  
vida_inimigo2 = 3
```

```
atirar B5  
caso (B5 == Inimigo1) {  
    vida_inimigo = vida_inimigo -1  
    mostra("Acertou")  
} seNao {  
    mostra("Errou")  
}
```

```
posicao Inimigo1 : B4
```

```
enquanto (vida_inimigo2 > 0) {  
    atirar a5  
    vida_inimigo2 = vida_inimigo2 - 1  
}  
mostra(vida_inimigo, vida_inimigo2)
```

```
funcao esconder (Inimigo1, Inimigo2) {  
    posicao Inimigo1 : Z100  
    posicao Inimigo2 : Z200  
}
```

```
chama esconder(Inimigo1, Inimigo2)
```

