

# Bomberman - IIA

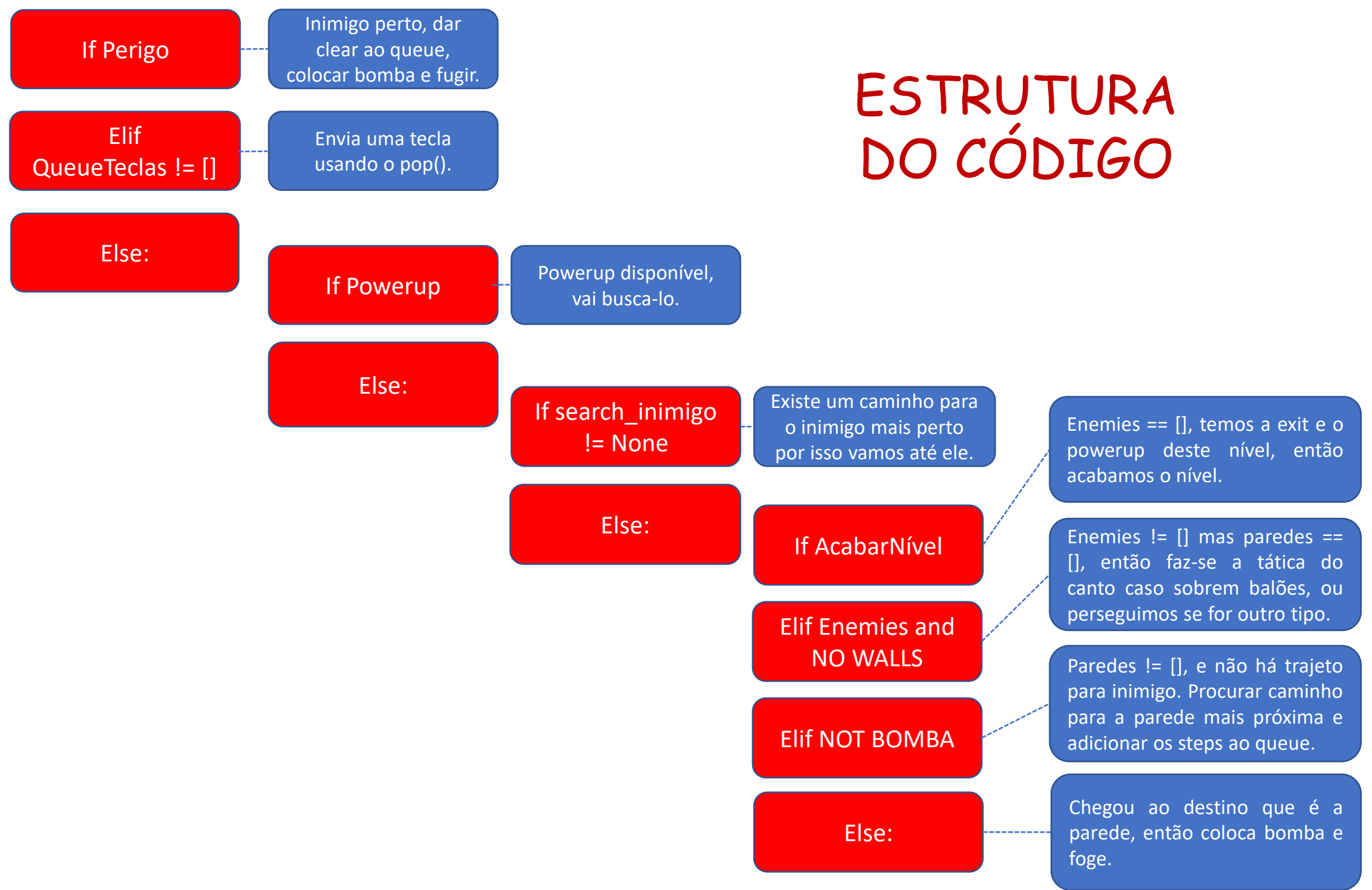
Eduardo Coelho (88867)

Joaquim Ramos (88812)

Rita Amante (89264)

<https://github.com/detiuaveiro/trabalho-de-grupo-bomberman-bomberman-ia-88867-88812-89264>

# ESTRUTURA DO CÓDIGO



Para fazer a pesquisa de um caminho usamos o tree search em modo greedy feito na aula.

Usamos lógica de submapas para criar as conexões entre as coordenadas.

Temos então um retângulo de ligações entre o bomberman e o destino (parede ou inimigo).

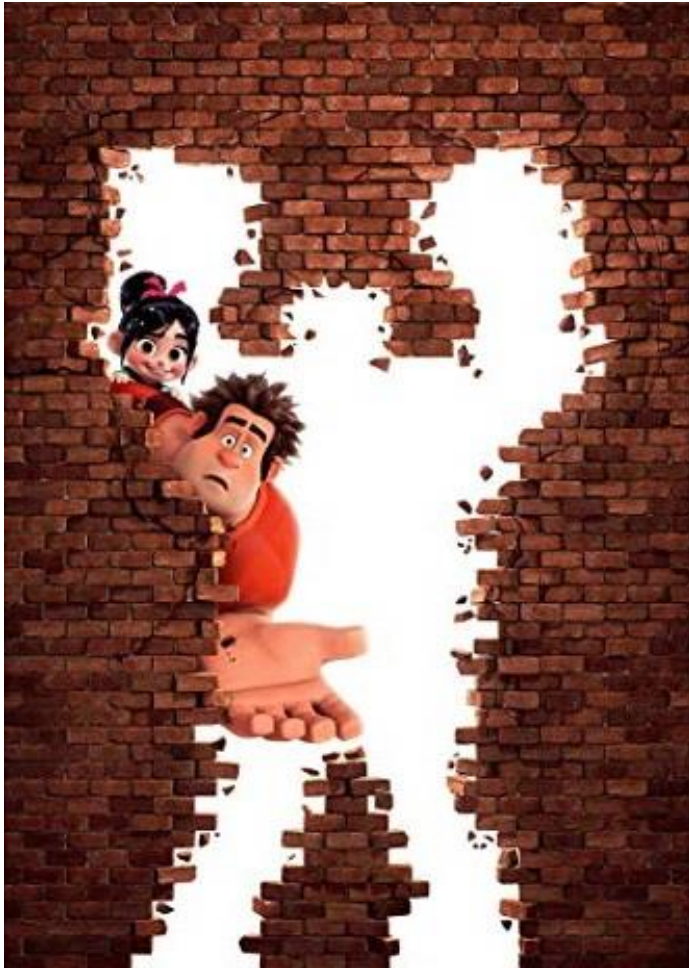
```
def criar_submap(bomberman,parede):  
    if(bomberman[0] < parede[0]):  
        xmin = bomberman[0]-2  
        xmax = parede[0]+2  
    else:  
        xmin = parede[0]-2  
        xmax = bomberman[0]+2  
  
    if(bomberman[1] < parede[1]):  
        ymin = bomberman[1]-2  
        ymax = parede[1]+2  
    else:  
        ymin = parede[1]-2  
        ymax = bomberman[1]+2  
  
    return [xmin,xmax,ymin,ymax]
```

No entanto, quando queremos fugir, criamos um submapa com raio 2 à volta do bomberman para procurar um sitio seguro para onde fugir.

O raio é enviado como argumento, o que permite à função fugir\_inimigo ir aumentando recursivamente o raio do submapa caso não encontre um sitio seguro.

```
# Criar um submap à volta do bomberman com raio K  
def criar_submap_fuga(bomberman,raio):  
    xmin = bomberman[0] - raio  
    xmax = bomberman[0] + raio  
    ymin = bomberman[1] - raio  
    ymax = bomberman[1] + raio  
  
    if xmin < 0:  
        xmin = 0  
    if ymin < 0:  
        ymin = 0  
  
    return [xmin,xmax,ymin,ymax]
```

# PAREDES



Para destruímos paredes seguimos os seguintes passos:

- Calcular através da hipotenusa a parede mais próxima;
- Calcular através da hipotenusa qual dos 4 lados dessa parede é mais perto;
- Calcular o caminho com tree search e converter para letras;
- Adicionar combinação de letras ao queue;

# INIMIGOS



✶

A cada step colocamos na variável *near\_enemy* qual é o inimigo mais próximo do bomberman, calculado através da hipotenusa.

Caso não consiga criar um caminho para esse inimigo, o bomberman vai destruir algumas paredes.

Quando encontra caminho simplesmente vai para as coordenadas do inimigo como se fosse uma parede.

Ao aproximar-se, o programa vai entrar no “if de perigo” que irá colocar uma bomba e fugir, possivelmente matando esse inimigo.

A distância a que o bomberman coloca a bomba depende da variável *rangeperigo*, que é atualizada a cada step dependendo de qual o inimigo mais próximo, pois como todos os inimigos têm características diferentes, existe um range ótimo diferente para cada um deles.