

INF216

2023/2



Projeto e Implementação de Jogos Digitais

A10: IA - Comportamentos de Navegação

Logística

Avisos

- ▶ Entrega do P4: Pac-Man terça-feira 07/11 (semana que vem)
- ▶ Teste T3: Inteligência Artificial sexta-feira 10/11 (semana que vem)

Última aula

- ▶ Pathfinding

Plano de Aula

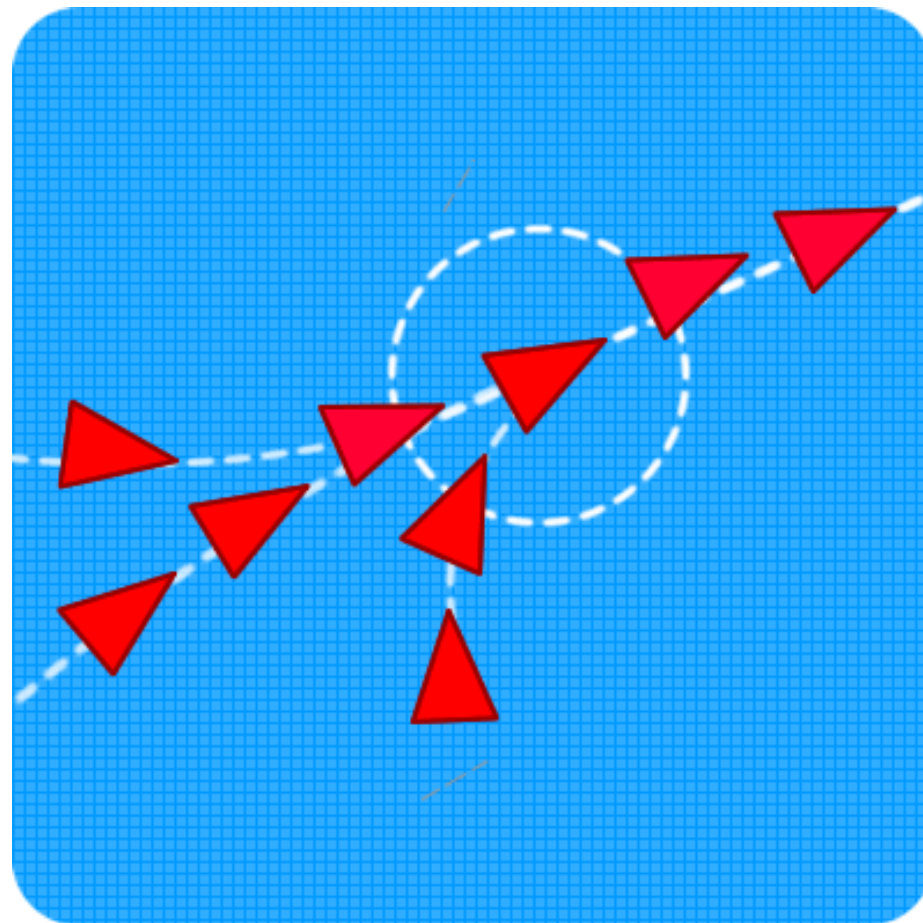
- ▶ Força de direção (*steering force*)
- ▶ Comportamentos de navegação (*steering behaviors*)
 - ▶ Procurar/Fugir
 - ▶ Passear
 - ▶ Seguir um caminho
 - ▶ Seguir um campo de fluxo
 - ▶ Movimentação em rebanho

Comportamentos de navegação



Desenvolvido por Craig Reynolds nos anos 80.

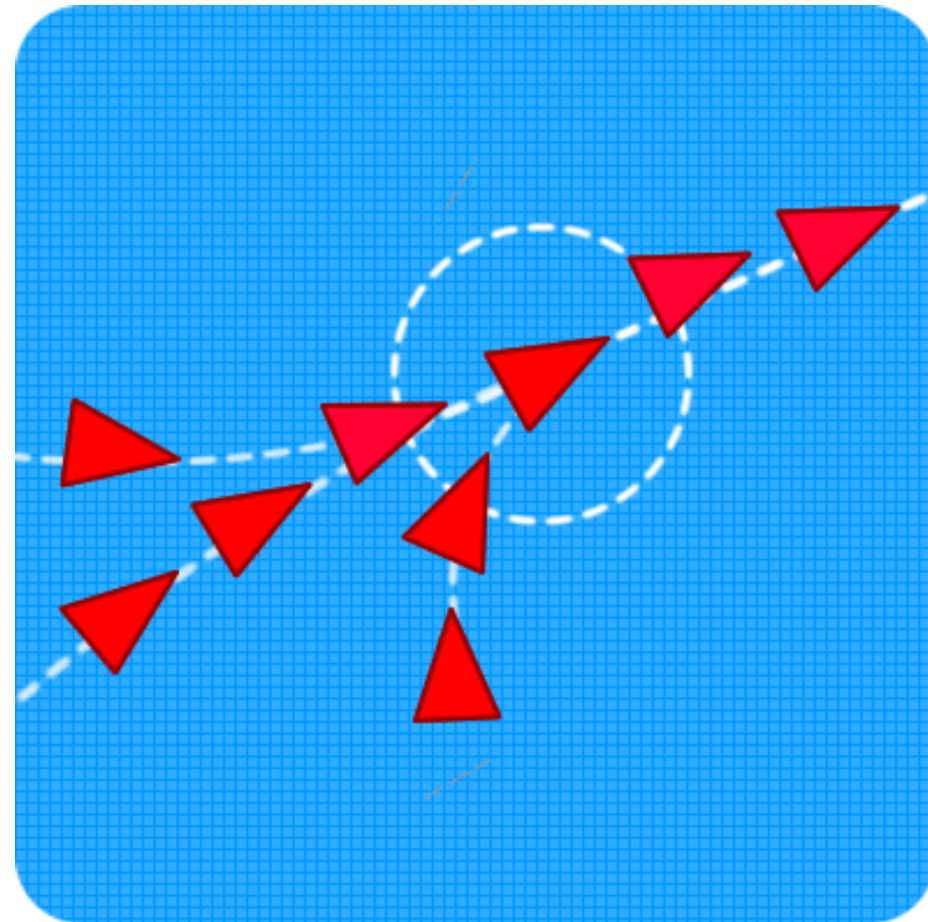
“Movimentar agentes autônomos de maneira orgânica”



Sistemas complexos de agentes:

- ▶ Cada agente se move de maneira independente
- ▶ Um agente percebe apenas suas redondezas
- ▶ Movimentos reativos, sem planejamento

Comportamentos de navegação



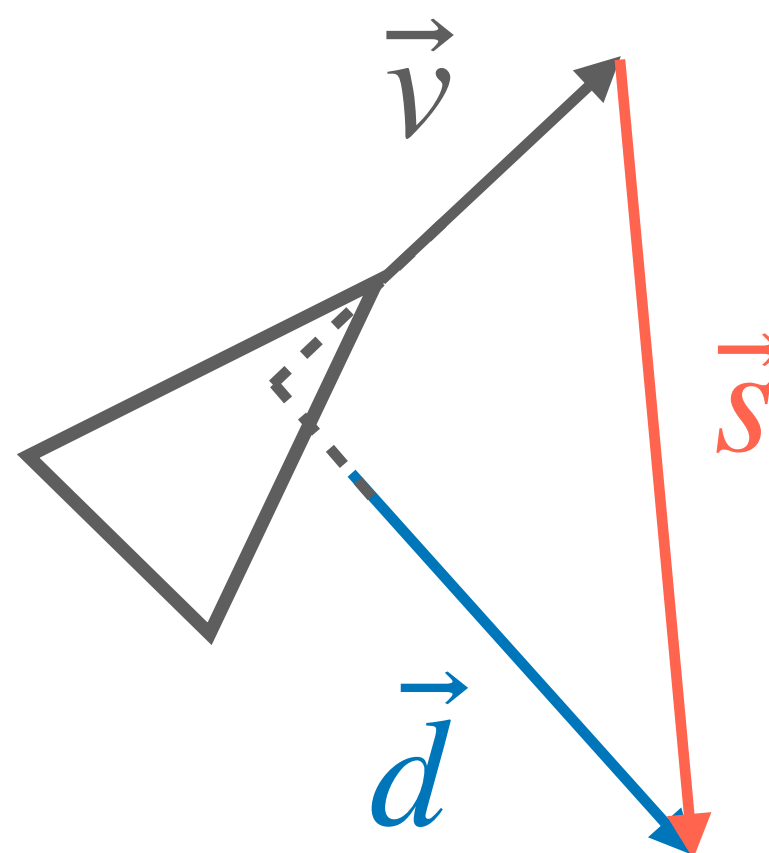
Reynolds modelou uma série de comportamentos naturais:

- ▶ Procurar/Fugir
- ▶ Passear
- ▶ Seguir um caminho
- ▶ Seguir um campo de fluxo
- ▶ Movimentação em rebanho
- ▶ ...

Força de direção

A base dos movimentos de navegação é a **força de direção** (*steering force*)

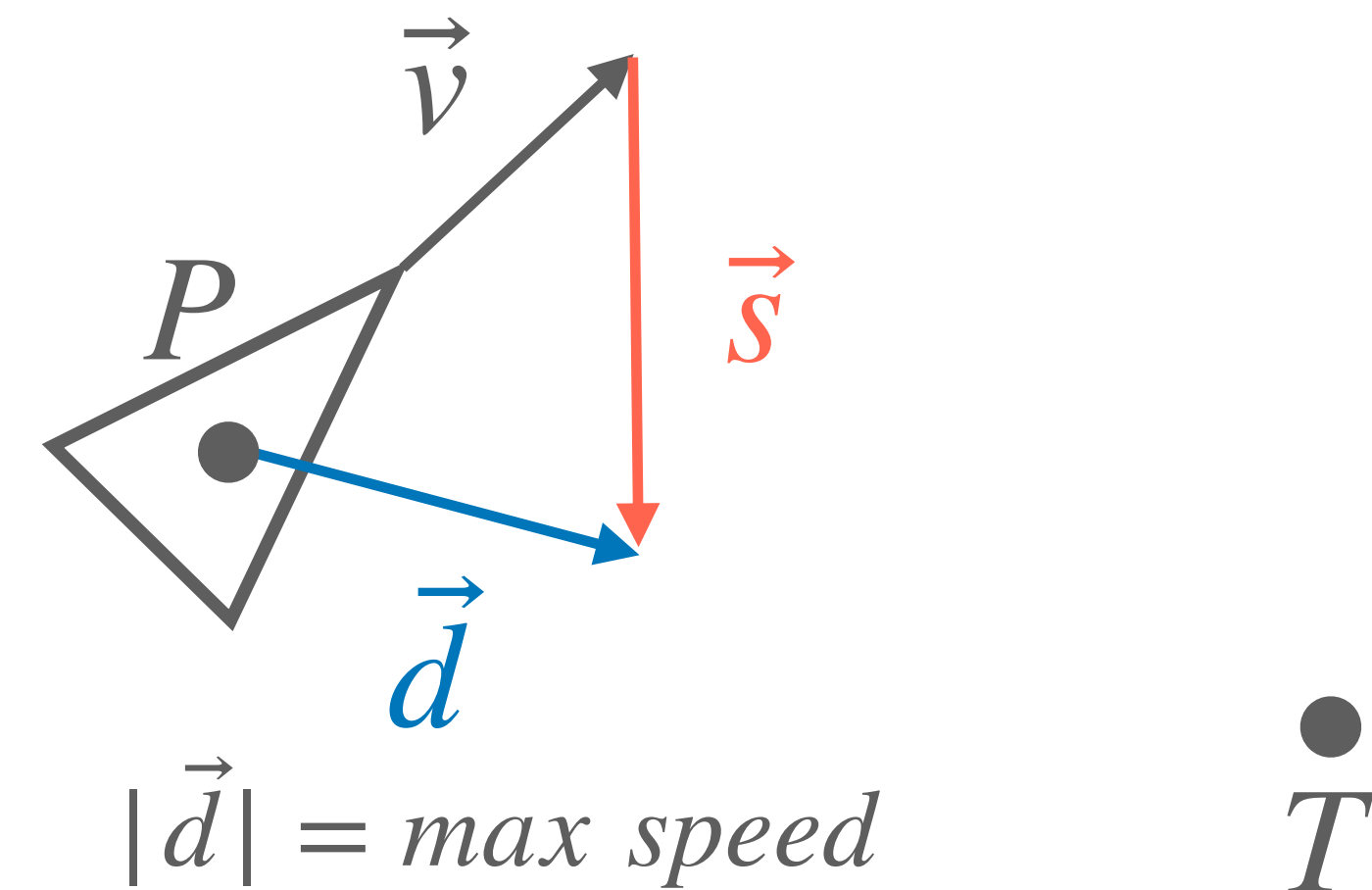
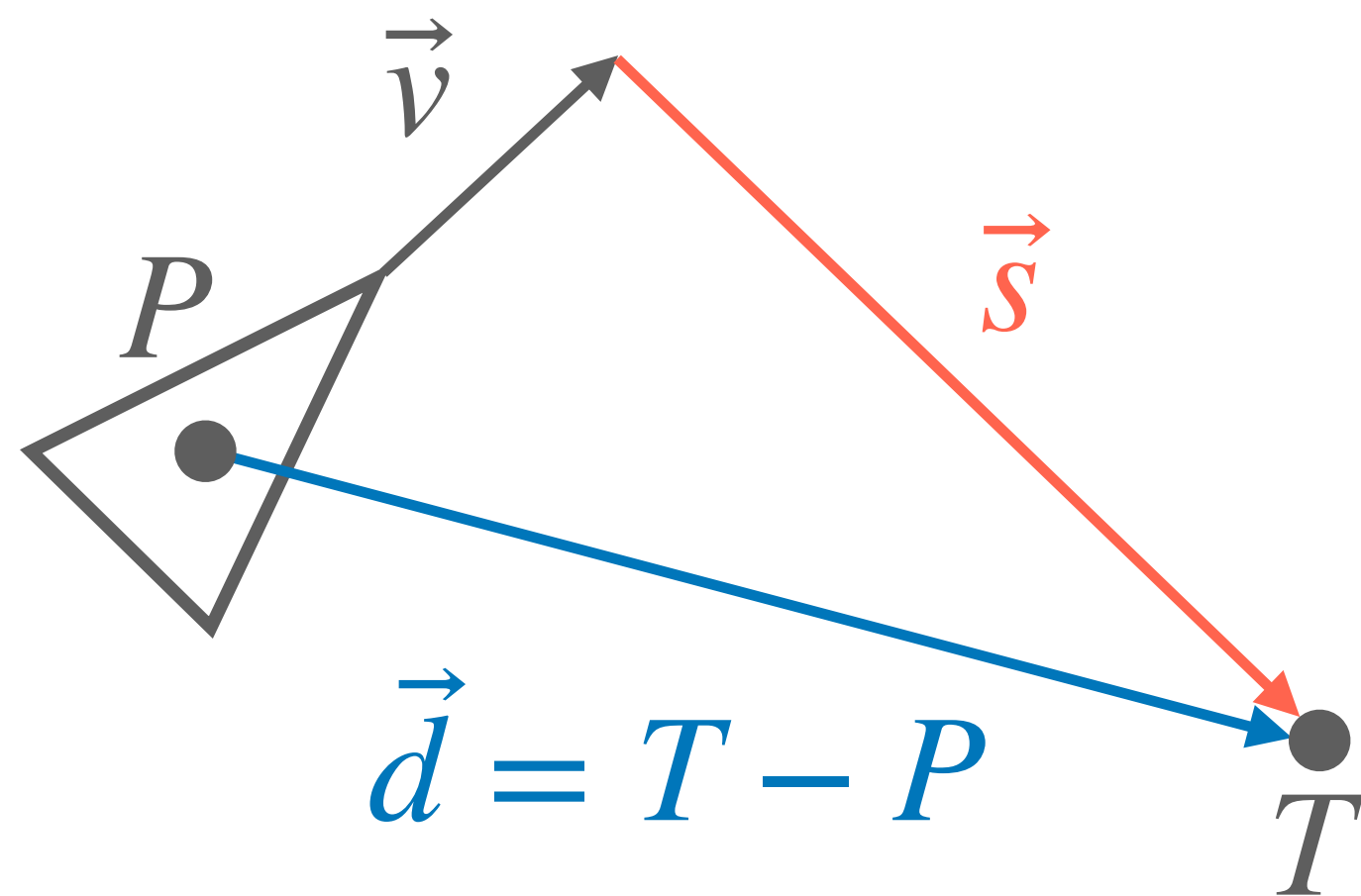
$\vec{s} = \vec{d} - \vec{v}$, onde \vec{d} é a **força desejada** (desired) e \vec{v} é a velocidade atual do objeto.



A definição de força de direção é a mesma para todos os comportamento de navegação, o que muda é a **força desejada** \vec{d} !

Procurar (seek)

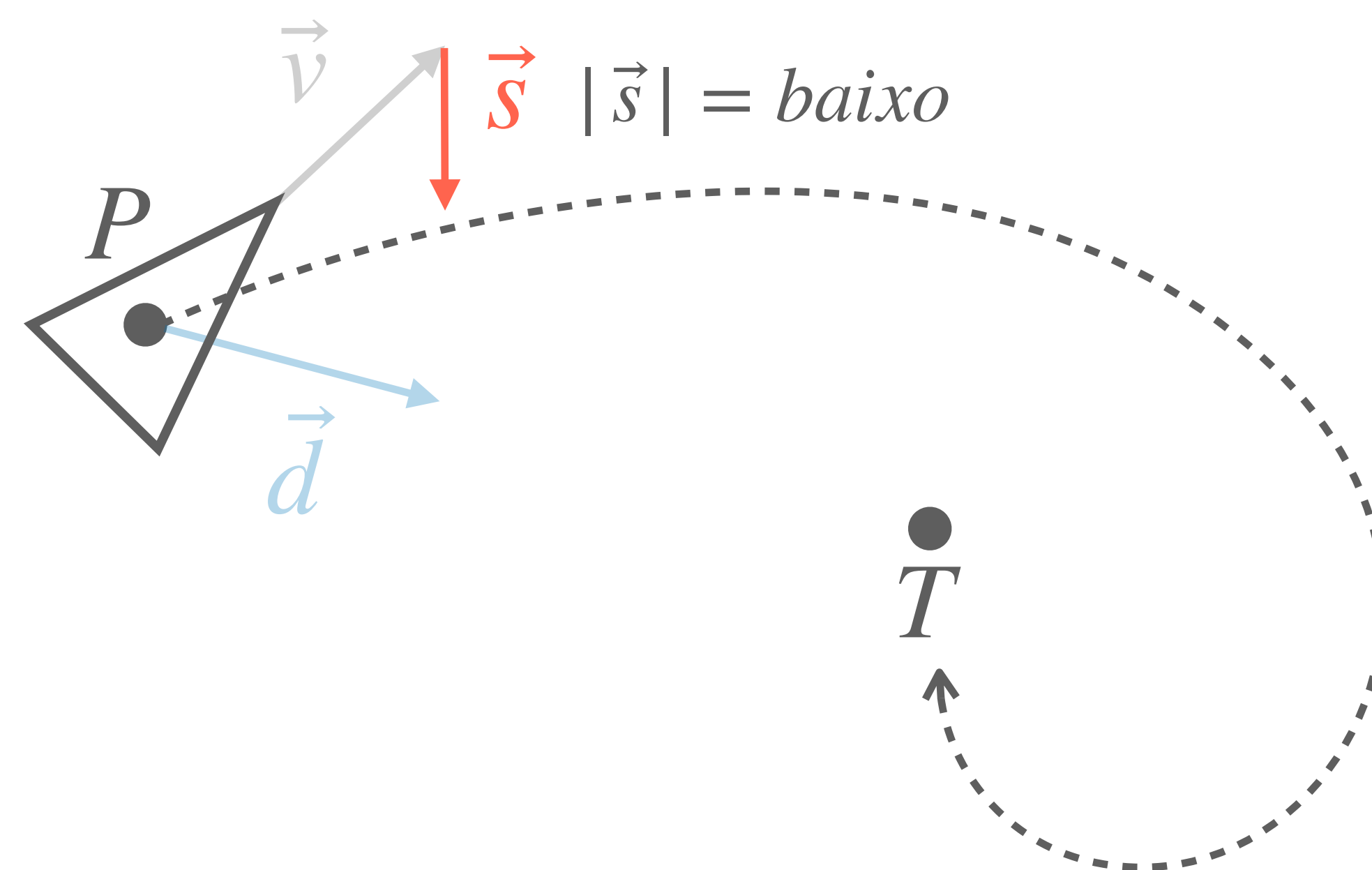
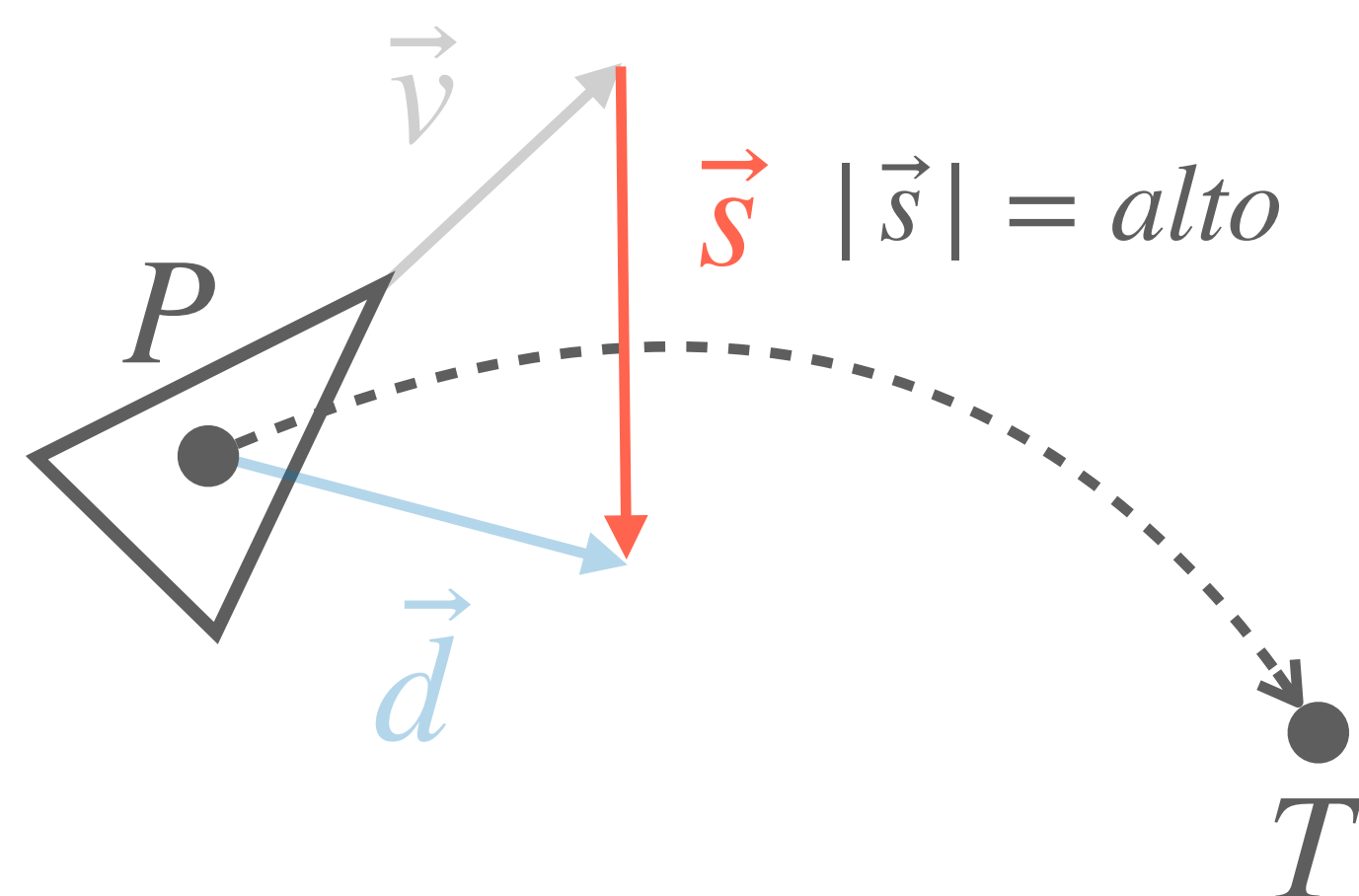
Para o comportamento de **procurar**, a **força desejada** é o vetor $\vec{d} = T - P$, onde T é a posição de um dado alvo e P é a posição do objeto.



Para que a navegação respeite as propriedades físicas do objeto, a **força desejada** deve ser reescalada para sua velocidade máxima.

Habilidade de direção

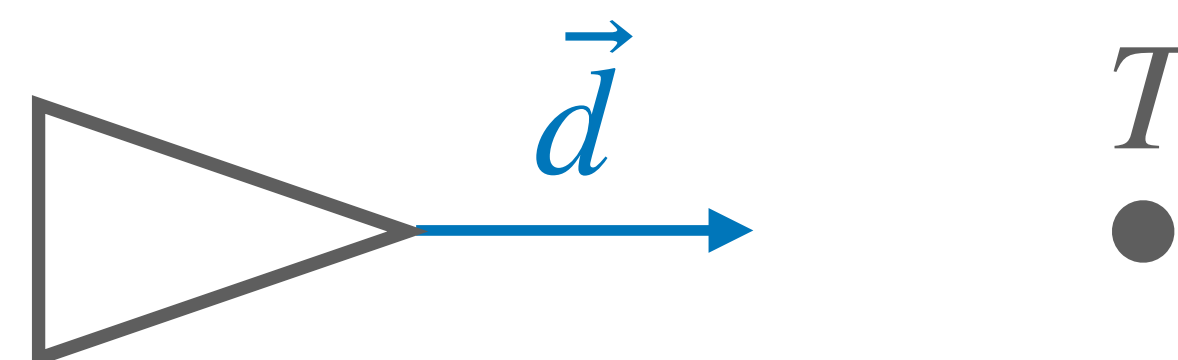
A habilidade de direção de um objeto pode ser controlada **limitando o comprimento** da **força de direção**.



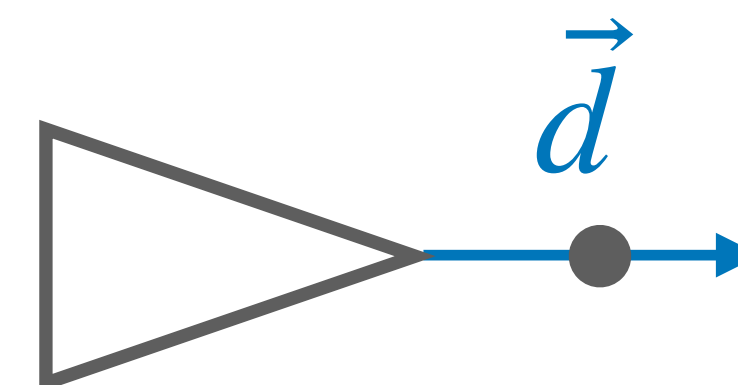
Chegar (*arrive*)

Atualmente, o comportamento de procurar reescala a **força desejada** com velocidade máxima a cada quadro:

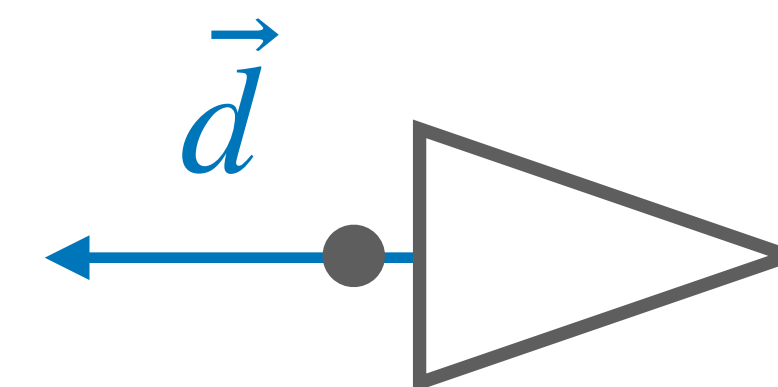
Quadro 1: mover o mais rápido possível!



Quadro 2: mover o mais rápido possível!



Quadro 3: mover o mais rápido possível!



Chegar (*arrive*)

Para **chegar** no alvo de maneira suave, é necessário reduzir o comprimento da **força desejada** \vec{d} se a distância entre o objeto P e o alvo T for menor do que um dado raio r .

Quadro 1: Eu estou longe do alvo.

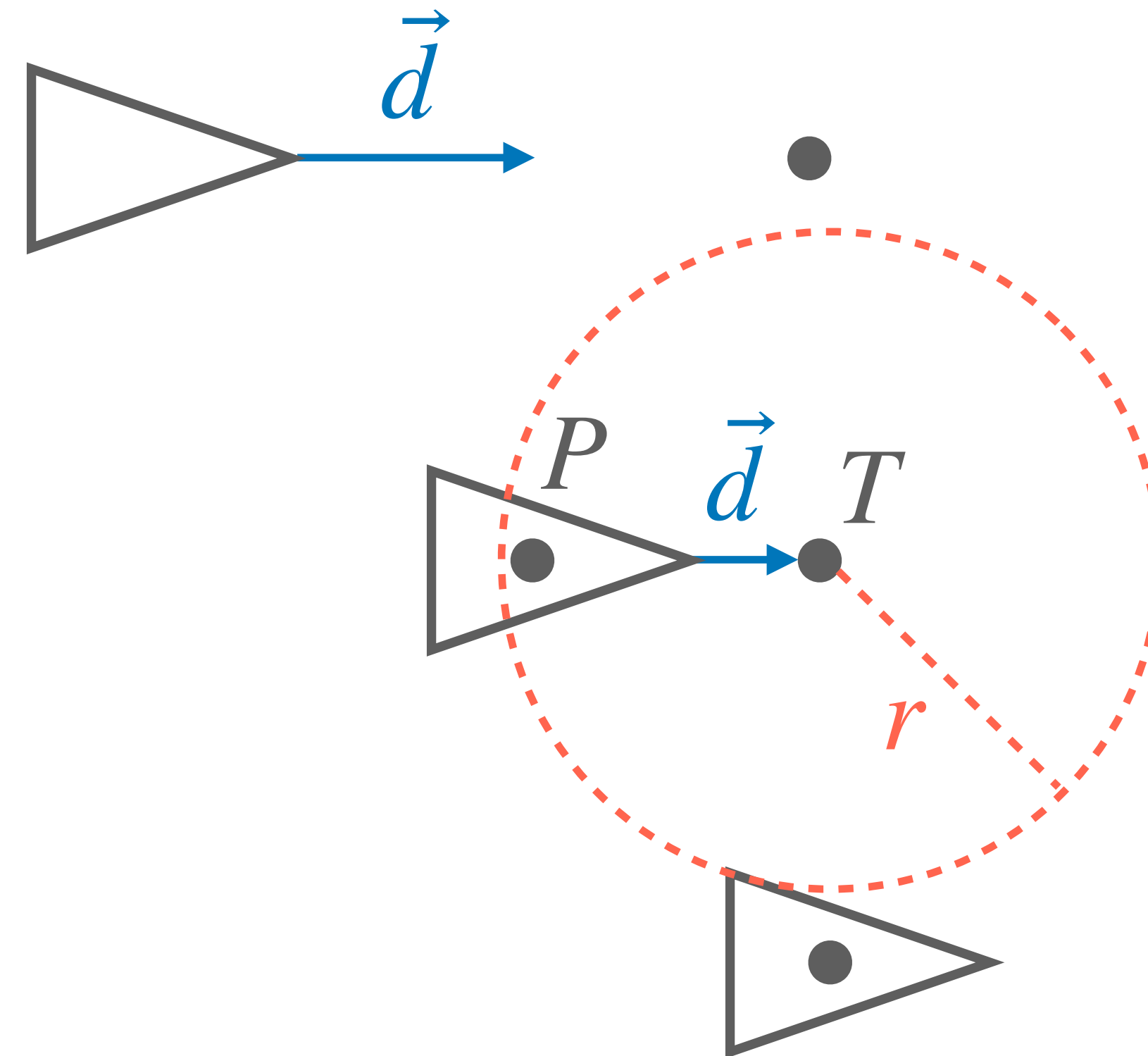
$$|d| = \text{max speed}$$

Quadro 2: Eu estou perto do alvo ($|d| < r$)

$$|d| = \frac{|d|}{r} * \text{max speed}$$

Quadro 3: Eu cheguei no alvo ($|d| < r$)

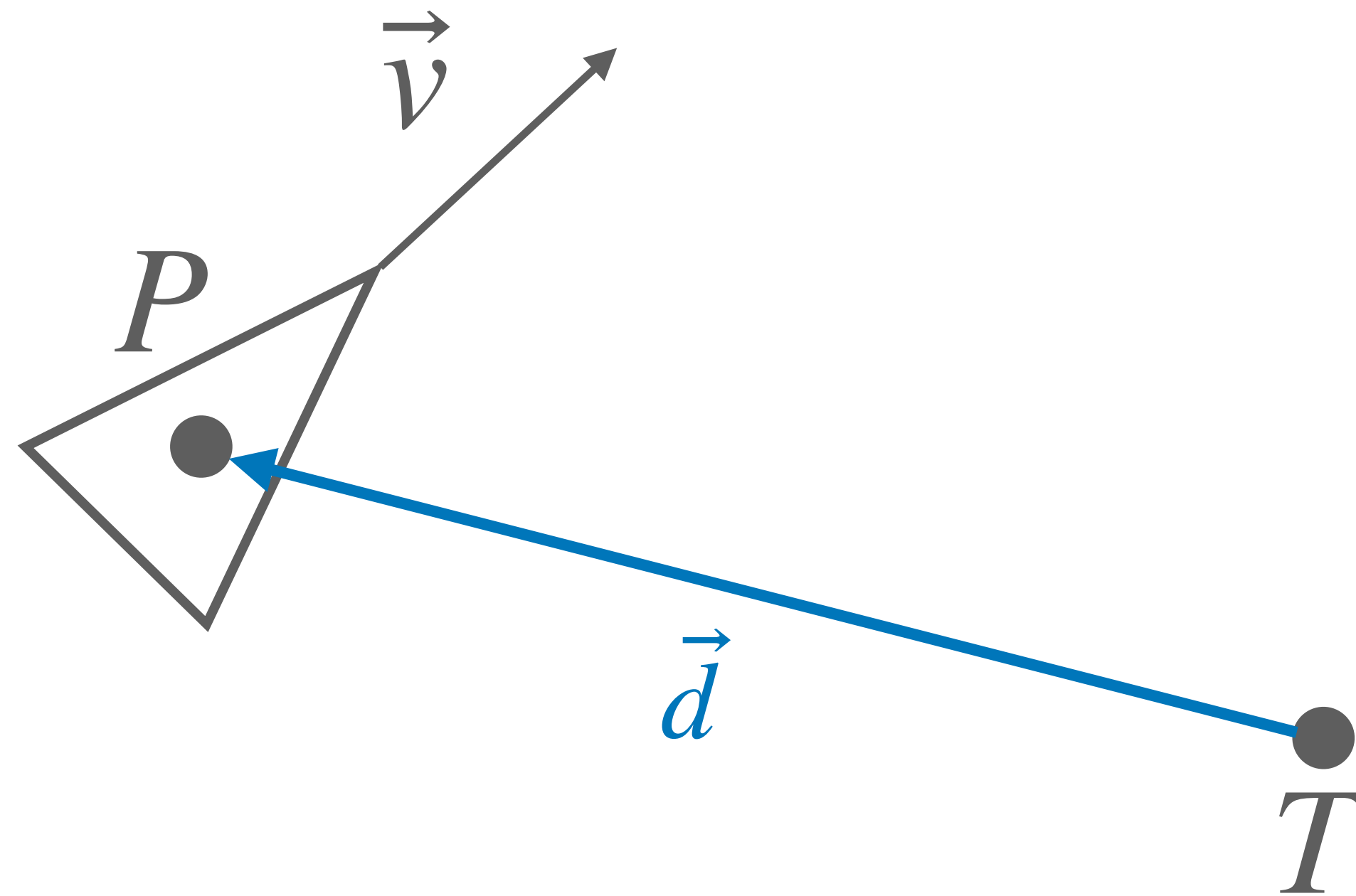
$$|d| = \frac{|d|}{r} * \text{max speed}$$



Fugir (*flee*)

Exercício 1: para o comportamento de **fugir** de um alvo T , a **força desejada** é o vetor $\vec{d} = ?$

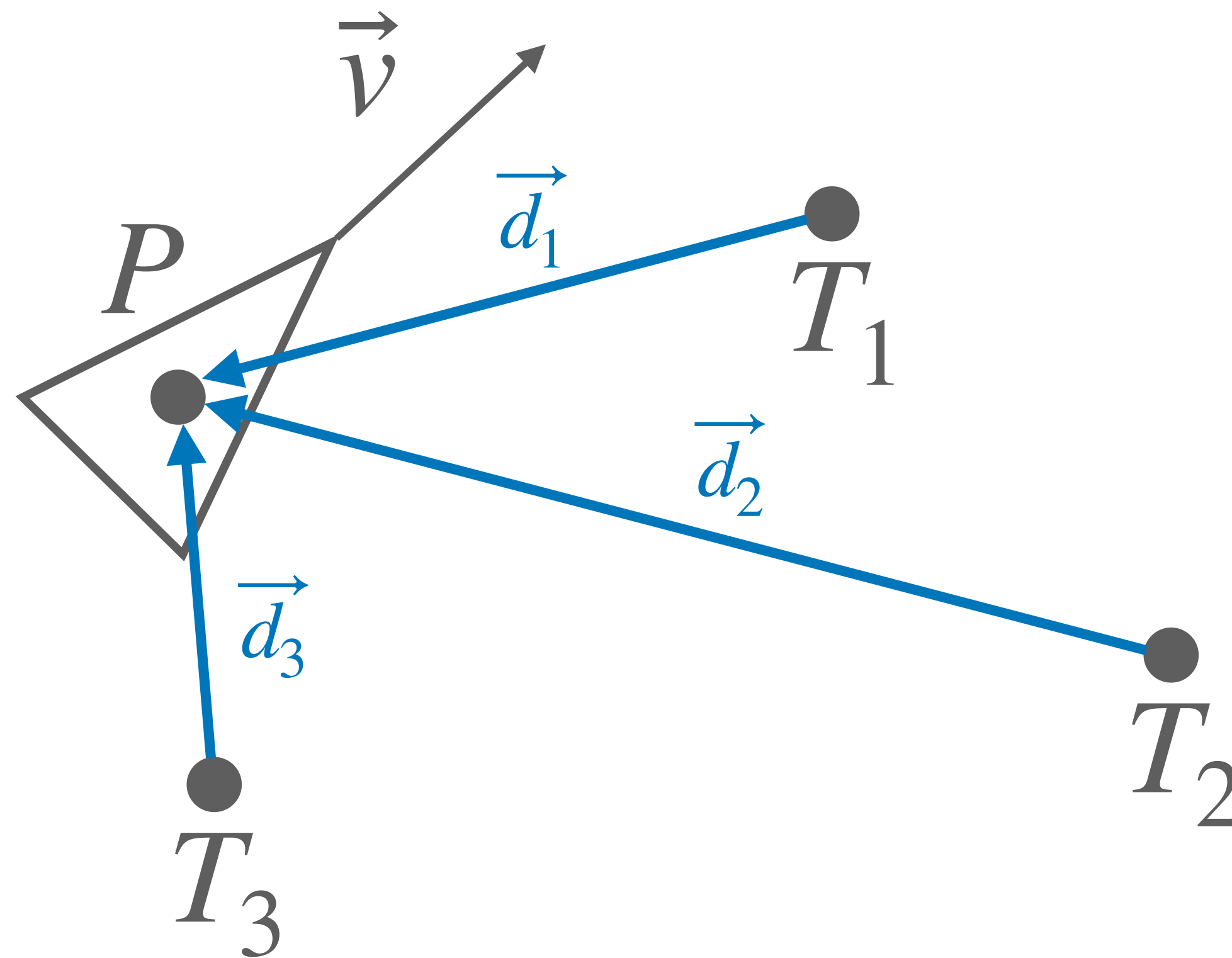
$$\vec{d} = P - T$$



Fugir (*flee*)

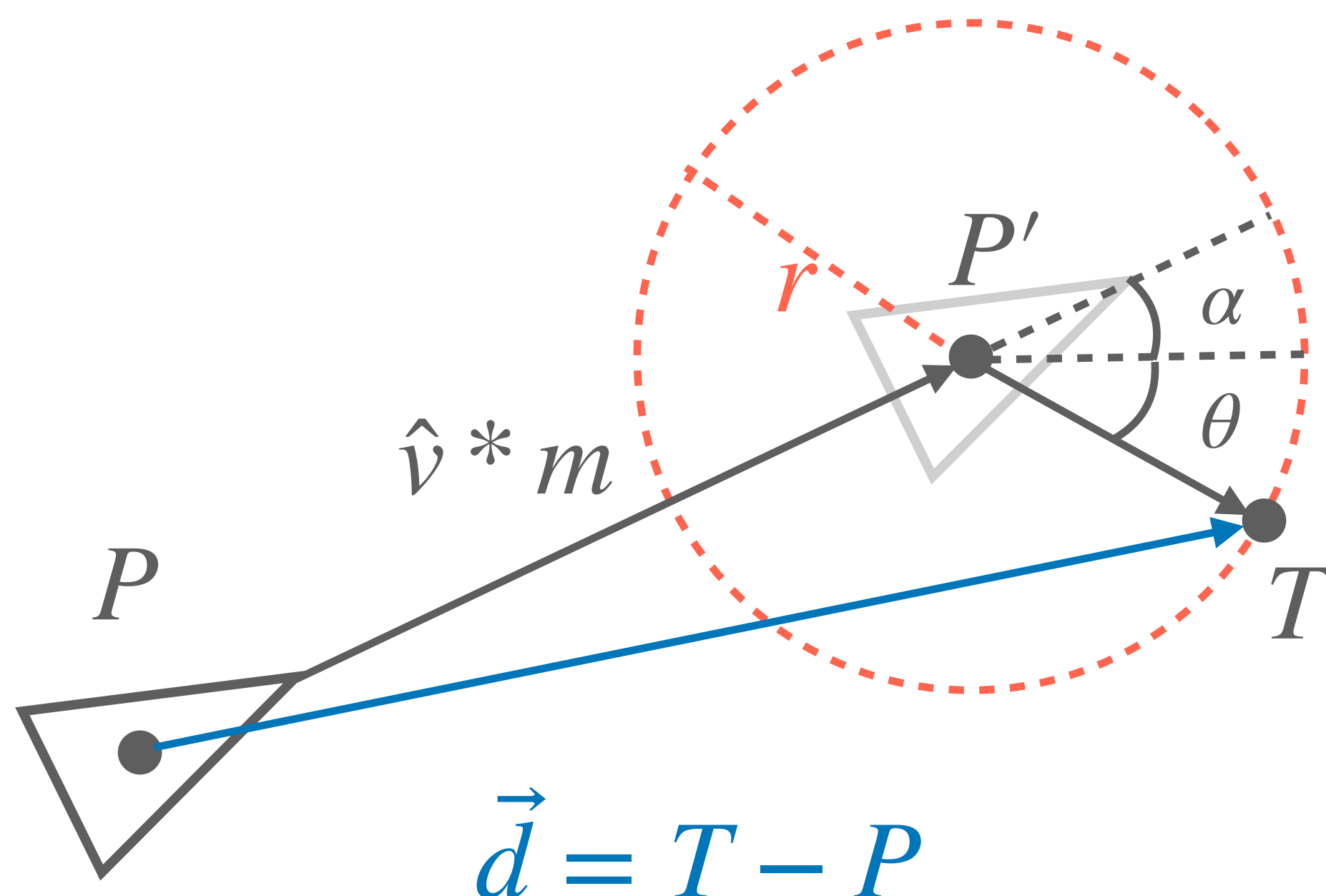
Exercício 2: para o comportamento de **fugir** de múltiplos alvos T_i , a **força desejada** é o vetor $\vec{d} = ?$

$$\vec{d} = \frac{\vec{d}_1 + \vec{d}_2 + \vec{d}_3}{3}$$



Passear (wander around)

Para o comportamento de **passear**, a **força desejada** é um ponto T aleatório em um círculo com centro na posição futura P' e raio r .



```

$$P' = P + (\vec{v} * m)$$

$$\theta += \text{random}(-c, c)$$

$$T_x = P'_x + r * \cos(\alpha + \theta)$$

$$T_y = P'_y + r * \sin(\alpha + \theta)$$

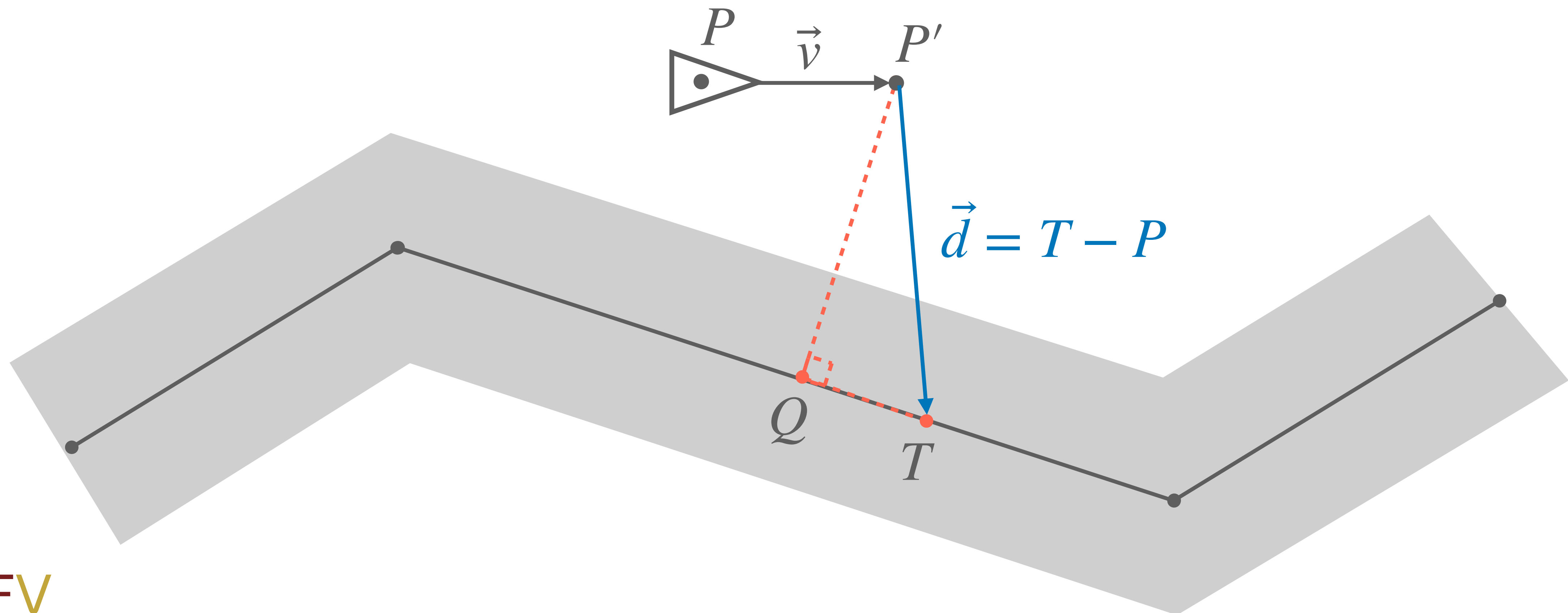
$$\text{seek}(T)$$

```

- ▶ m é um fator de projeção de velocidade
- ▶ c é uma constante de variação angular

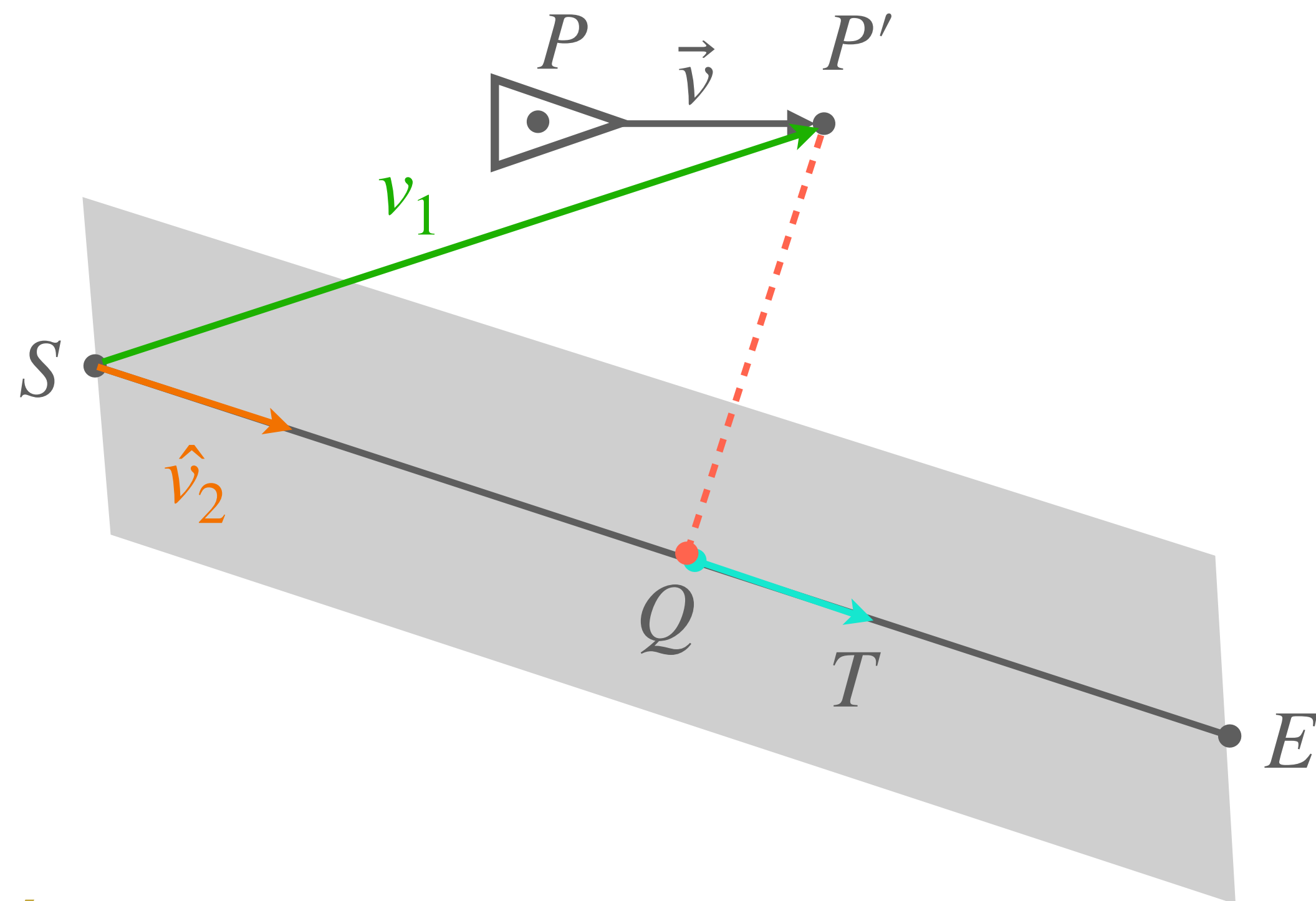
Seguir um caminho (*Path following*)

Para o comportamento de **seguir um caminho**, a **força desejada** \vec{d} é um ponto T a frente da projeção Q da posição futura P' em um dado caminho.



Seguir um caminho (*Path following*)

Vamos assumir um caminho com apenas um segmento de caminho \overrightarrow{SE} para facilitar o calculo de P' , Q e T



$$P' = P + (\vec{v} * m)$$

$$v_1 = P' - S$$

$$\hat{v}_2 = \text{norm}(E - S)$$

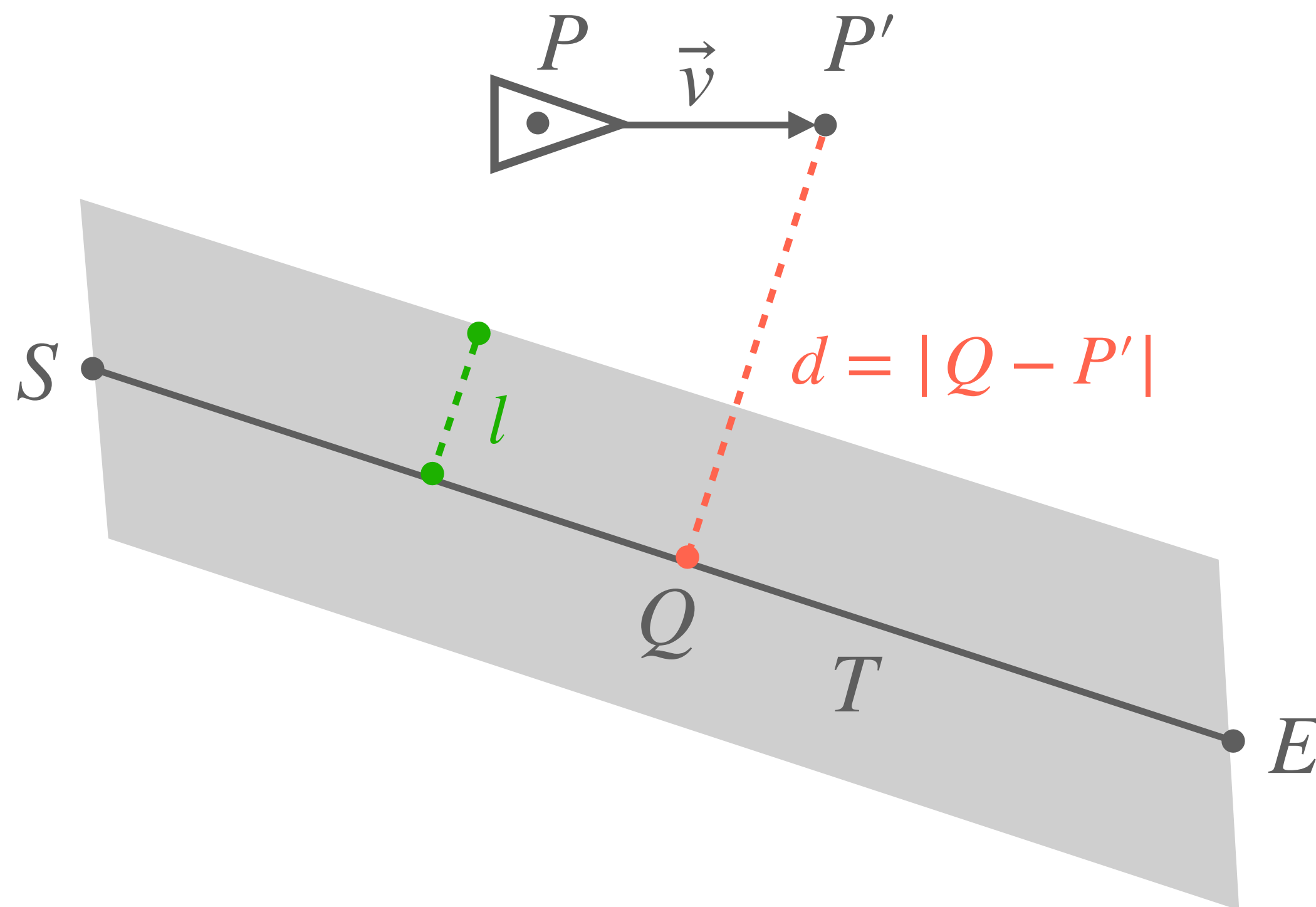
$$Q = v_1 \cdot \hat{v}_2$$

$$T = Q + \hat{V}_2 * n$$

- ▶ m é um fator de projeção de velocidade
- ▶ n é a distância desejada entre Q e T

Seguir um caminho (*Path following*)

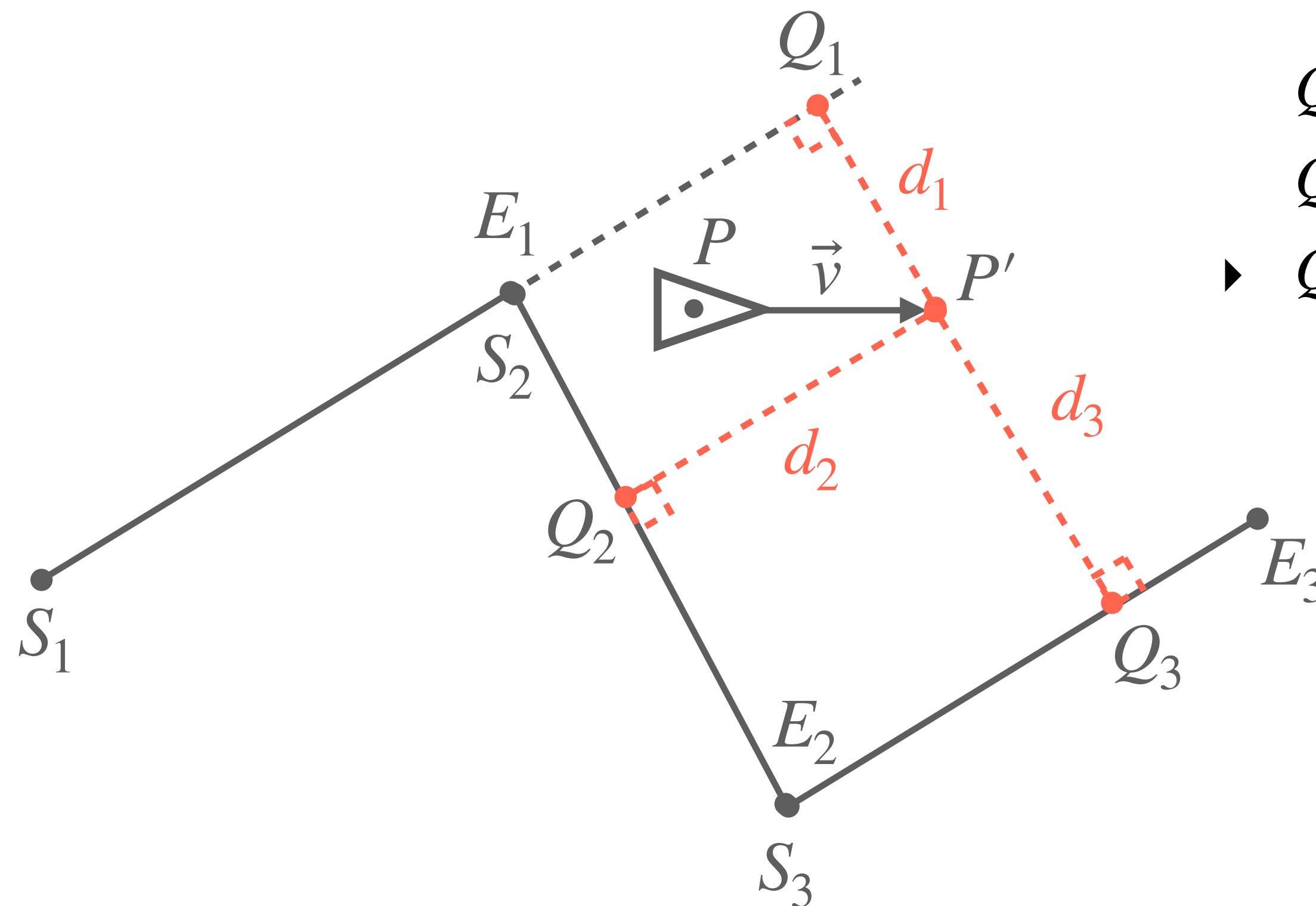
O objeto persegue T apenas se a distância d entre a posição futura P' for maior do que a largura l do caminho. Caso contrário, o objeto se move normalmente de acordo com sua velocidade \vec{v}



```
d = |Q - P'|  
if d > l:  
    seek(T)
```


Seguir um caminho (*Path following*)

Para seguir um caminho formado por mais de um segmento, calculamos a projeção Q_s para todos os segmentos s e escolhemos aquela que está contida em $\overrightarrow{SE_s}$ com menor distância $d_s = |Q_s - P'|$



- Q_1 tem a menor distância d_1 , mas não está em $\overrightarrow{SE_1}$
- Q_2 está em $\overrightarrow{SE_2}$, mas $d_2 > d_3$
- Q_3 está em $\overrightarrow{SE_3}$ e tem a menor distância! [escolhido]

Próximas aulas

A11: Interface com o usuário

Sistemas de menus, janelas de diálogo e heads-up display.

L11: The Legend of Zelda - Parte 1

Implementar o comportamento de navegação para seguir um caminho.