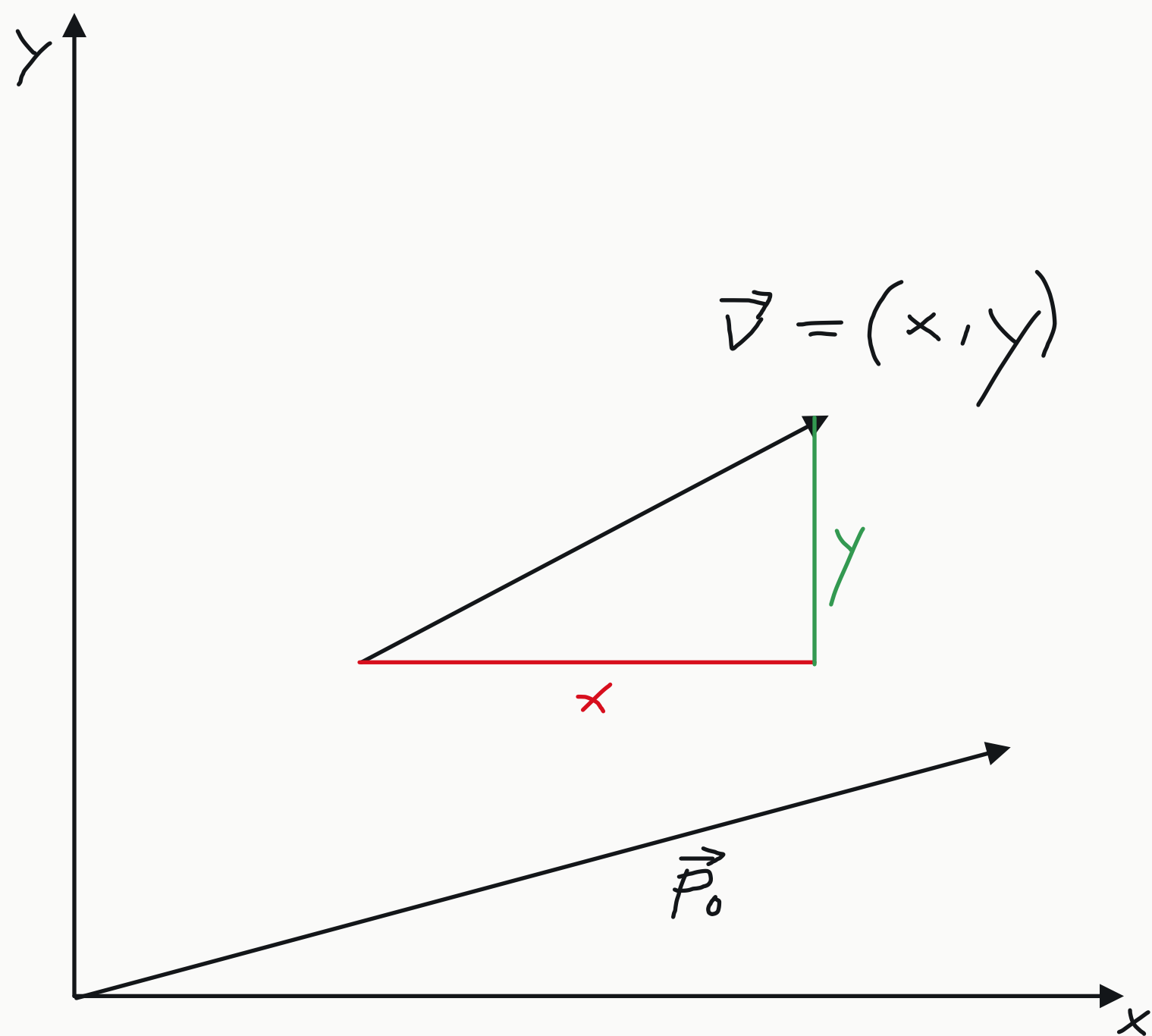
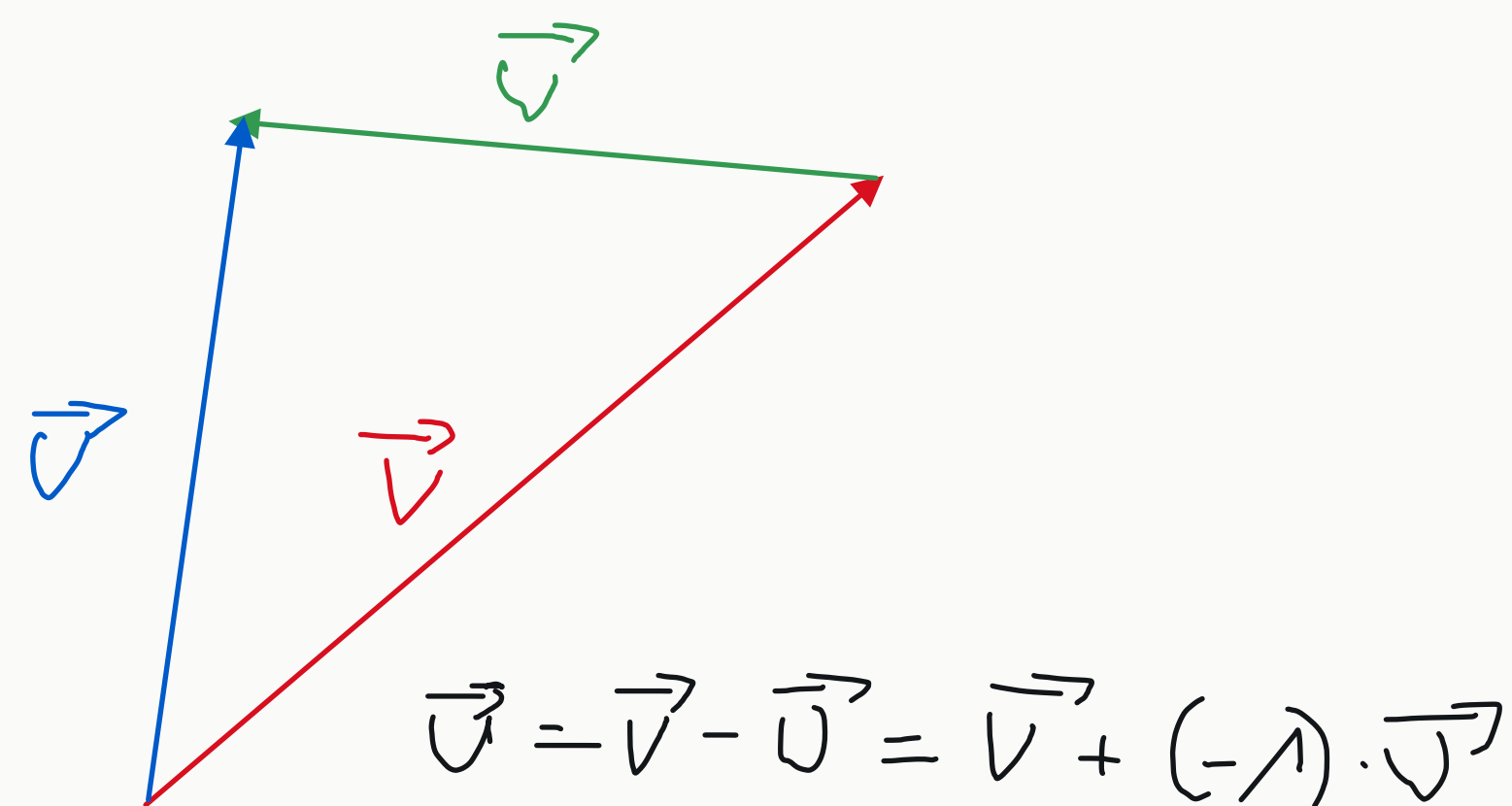
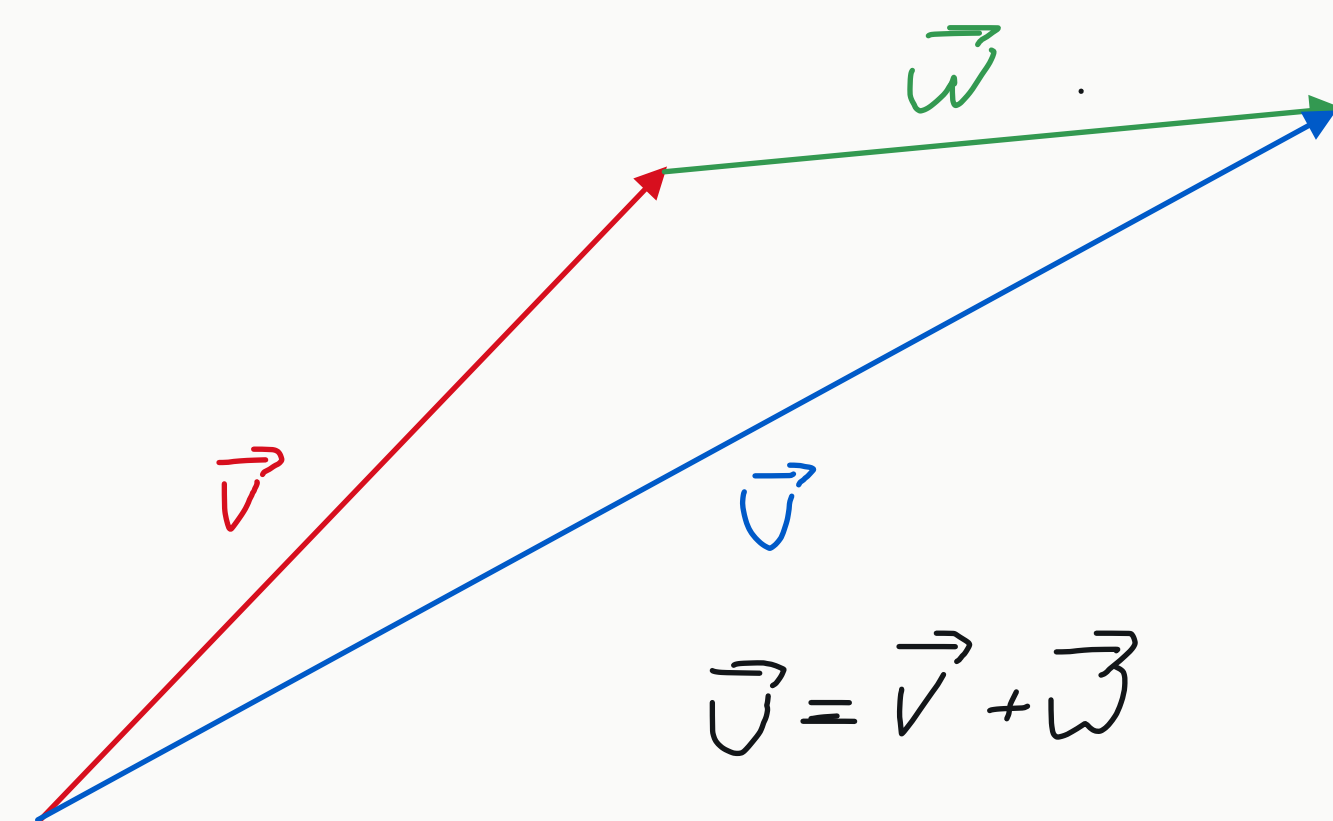
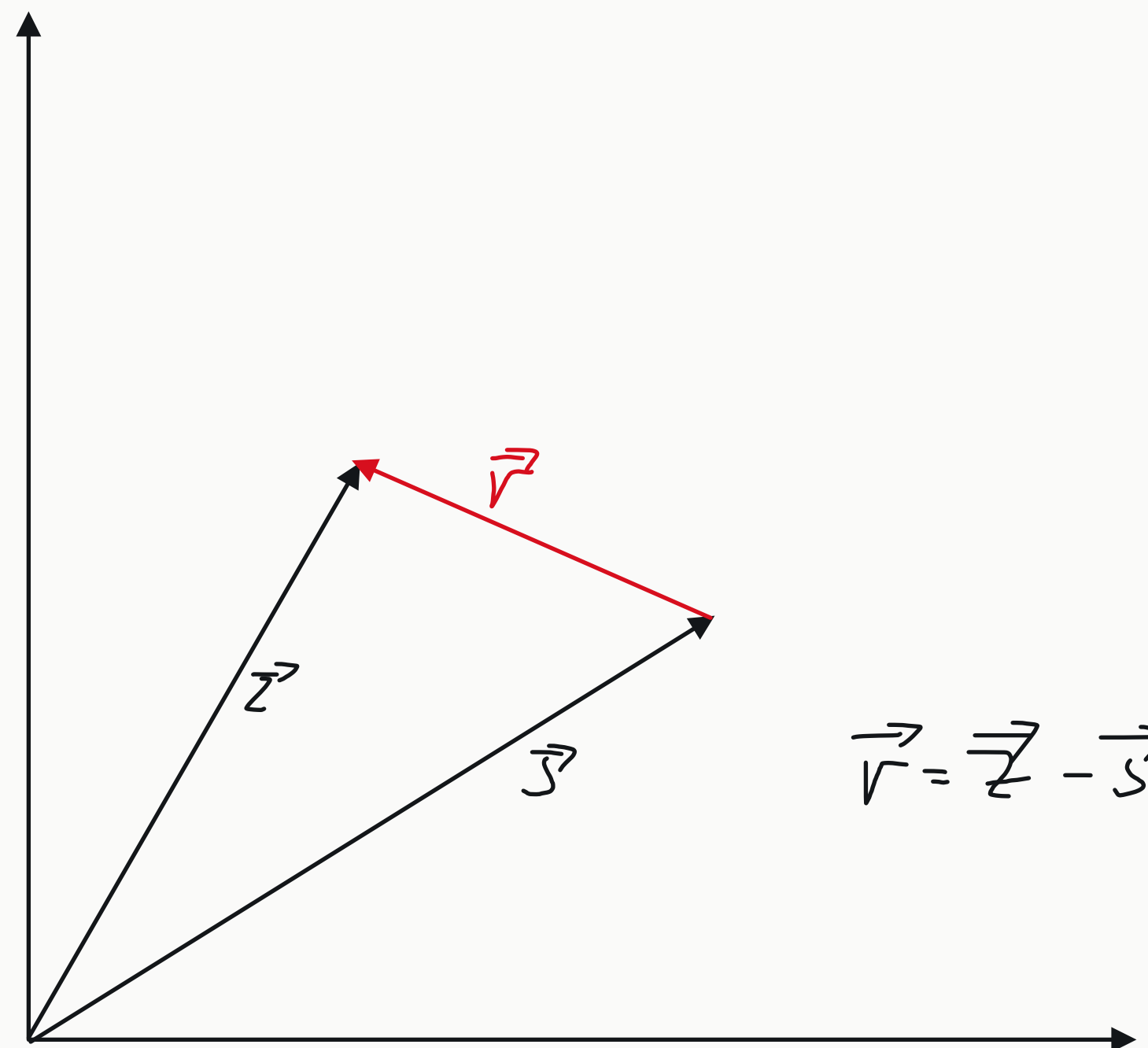


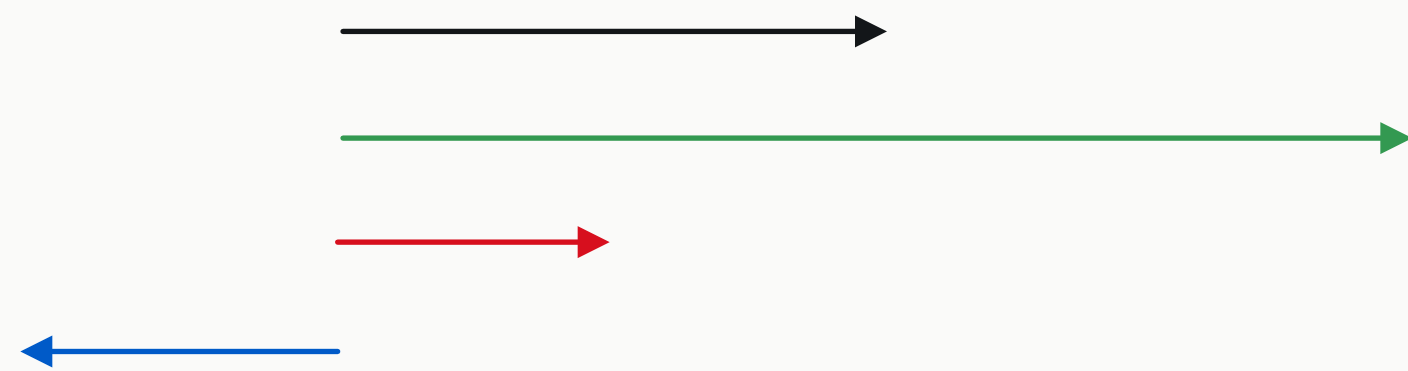


$$|\vec{v}| = \sqrt{x^2 + y^2}$$

$$\|\vec{v}\| = \frac{\vec{v}}{|\vec{v}|}$$



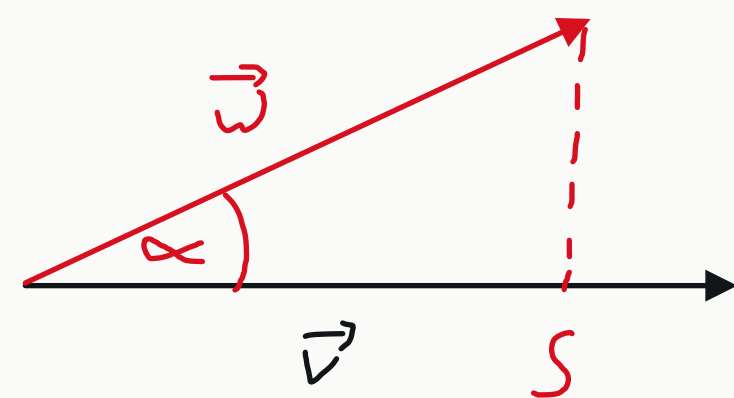
Multiplication with a Scalar



$$\vec{v} \cdot a = a \cdot \vec{v} = \begin{pmatrix} v_x \cdot a \\ v_y \cdot a \\ v_z \cdot a \end{pmatrix}$$

`transform.position += moveVector*(speed*Time.deltaTime)`

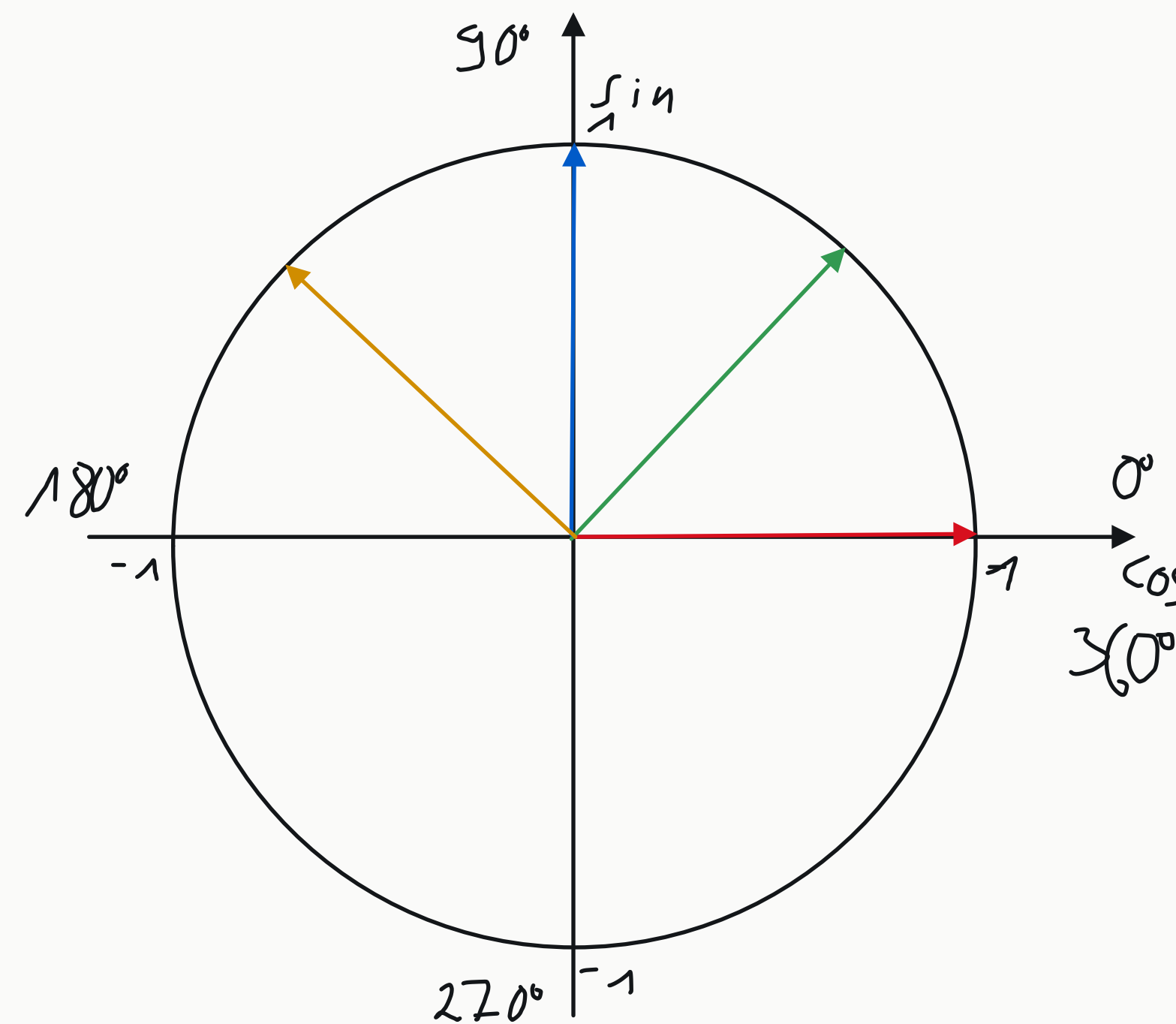
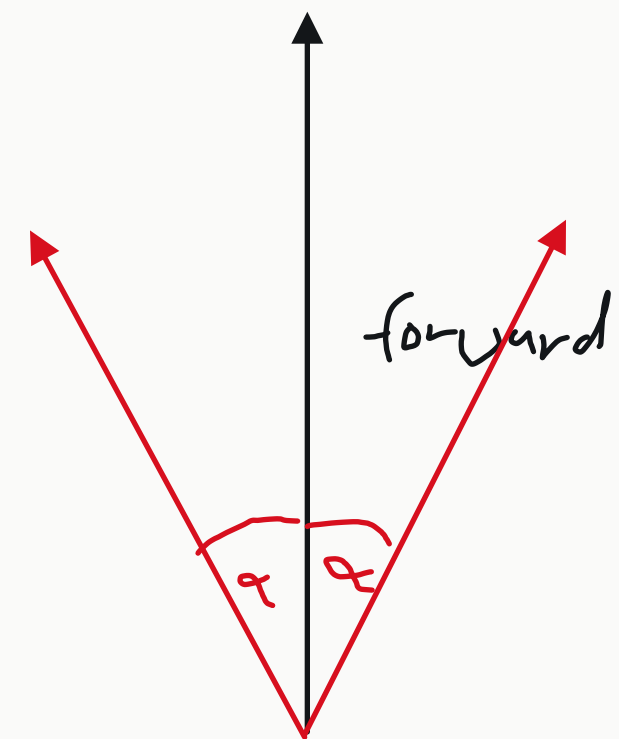
Dot Product



$$\vec{v} \cdot \vec{u} = \vec{v}_x \cdot \vec{u}_x + \vec{v}_y \cdot \vec{u}_y + \vec{v}_z \cdot \vec{u}_z$$

$$\vec{v} \cdot \vec{u} = |\vec{v}| \cdot |\vec{u}| \cdot \cos \alpha \Leftrightarrow \cos \alpha = \frac{\vec{v} \cdot \vec{u}}{|\vec{v}| \cdot |\vec{u}|} \quad \cos^{-1} \quad \arccos$$

$$\alpha < \beta \rightarrow \cos \alpha > \cos \beta$$

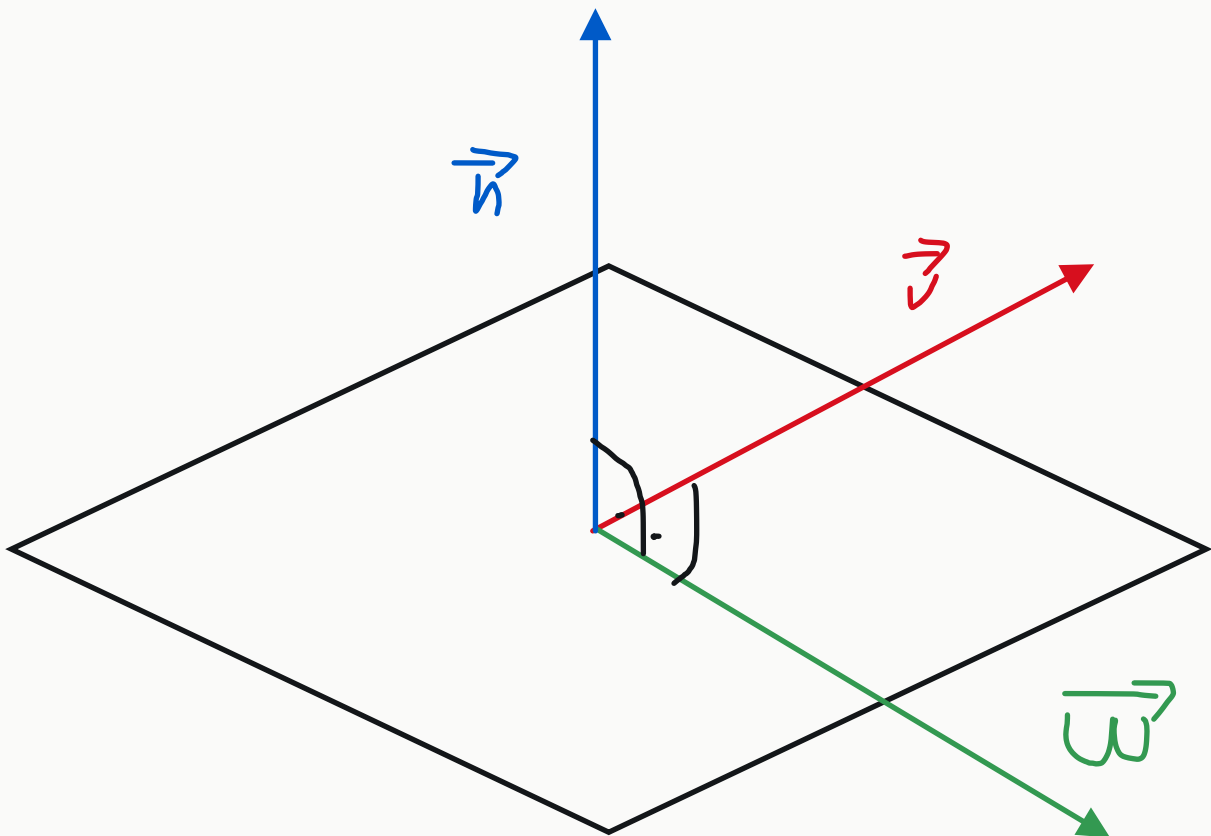


$$\frac{\alpha}{180^\circ} = \frac{x}{\pi}$$

$$\alpha = x \cdot \left(\frac{180^\circ}{\pi} \right) \quad \text{Rad2Deg}$$

$$x = \alpha \cdot \left(\frac{\pi}{180} \right) \quad \text{Deg2Rad}$$

Cross Product



$$\vec{v} \times \vec{u} \neq \vec{u} \times \vec{v}$$

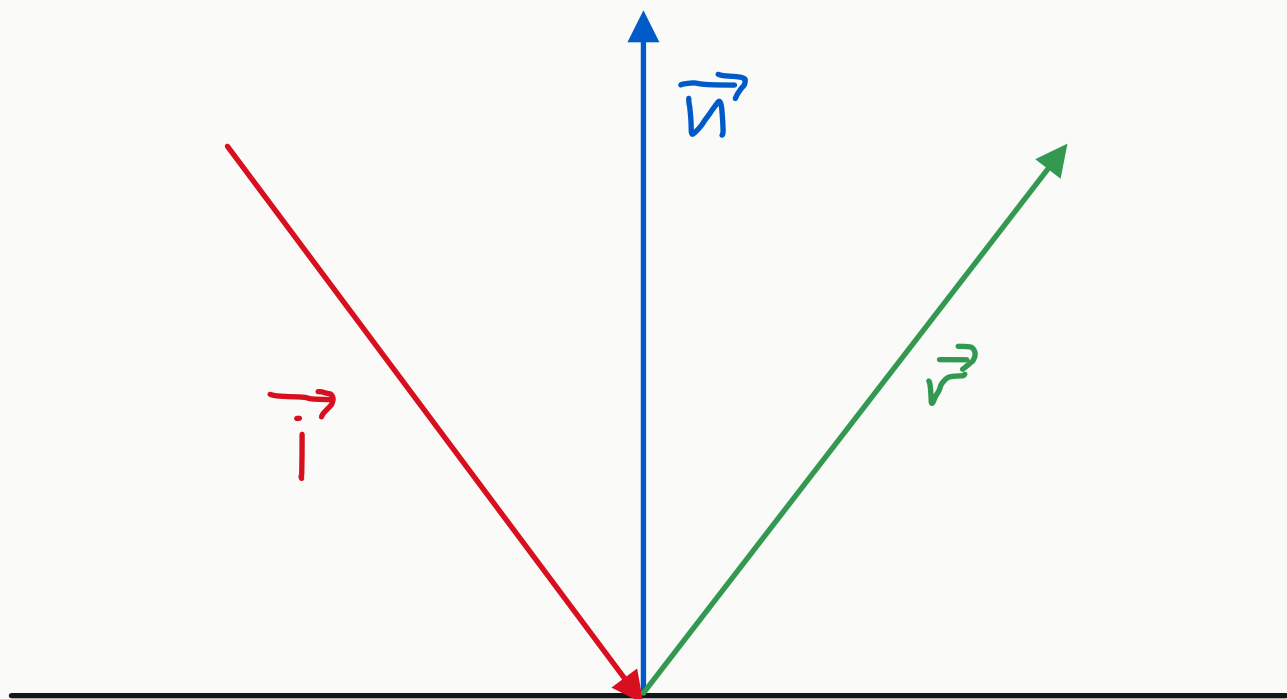
$$\vec{n} = \vec{v} \times \vec{u} \quad -\vec{n} = \vec{u} \times \vec{v}$$

$$v_x \quad u_x$$

$$\begin{array}{l} v_y \quad u_y \\ -v_z \quad u_z + \\ -v_x \quad u_x + \\ -v_y \quad u_y + \end{array}$$

$$v_z \quad u_z$$

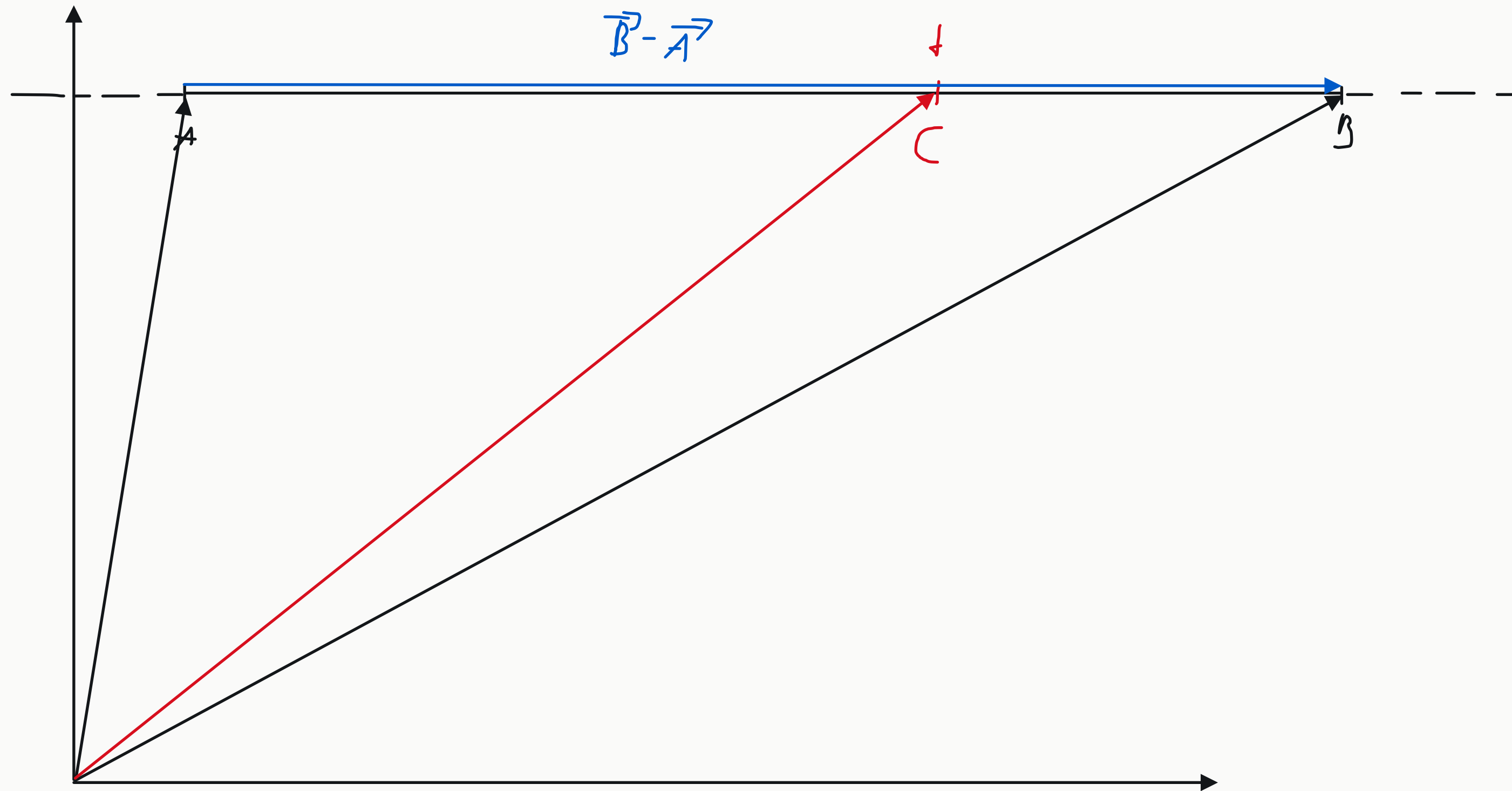
$$\vec{n} = \begin{pmatrix} v_y u_z - v_z u_y \\ v_z u_x - v_x u_z \\ v_x u_y - v_y u_x \end{pmatrix}$$



$$\vec{r} = \vec{i} - 2 \cdot (\vec{i} \cdot \vec{n}) \cdot \vec{n}$$

SIMD - Single Instruction Multiple Data

Lineare Interpolation (lerp)



$$0 \leq t \leq 1$$

$$\vec{C} = \vec{A} + t \cdot (\vec{B} - \vec{A})$$

Lineare Extrapolation (lerpUnclamped)

PingPong

Tween Library (Unity) -> PrimeTween

<https://assetstore.unity.com/packages/tools/animation/primetween-high-performance-animations-and-sequences-252960>

Bezier Curves

Linear



$$B_L(t) = P_0 + t(P_1 - P_0)$$

$$= P_0 + tP_1 - tP_0$$

$$= (1-t)P_0 + tP_1$$

https://en.wikipedia.org/wiki/Bezier_curve#Constructing_Bezier_curves

<https://assetstore.unity.com/packages/tools/utilities/dreamteck-splines-61926>

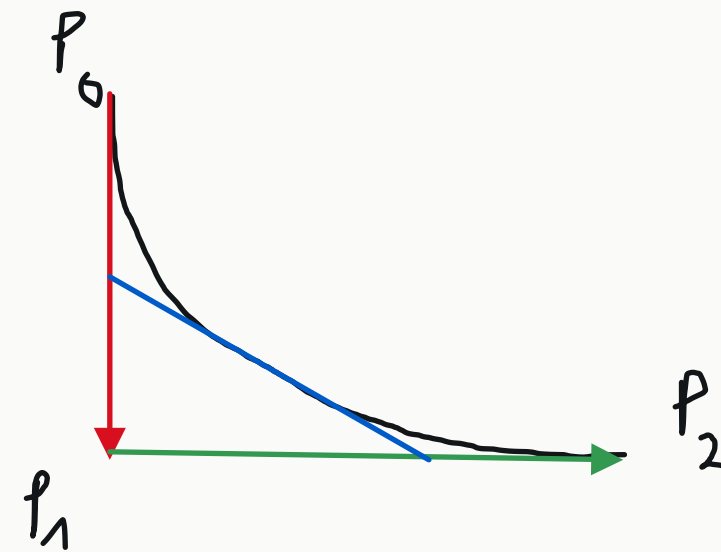
<https://assetstore.unity.com/packages/tools/utilities/curvy-splines-8-212532>

Quadratic

$$B_Q(t) = (1-t)((1-t)P_0 + tP_1) + t((1-t)P_1 + tP_2)$$

$$= (1-t)^2 P_0 + (1-t)t P_1 + (1-t)t P_1 + t^2 P_2$$

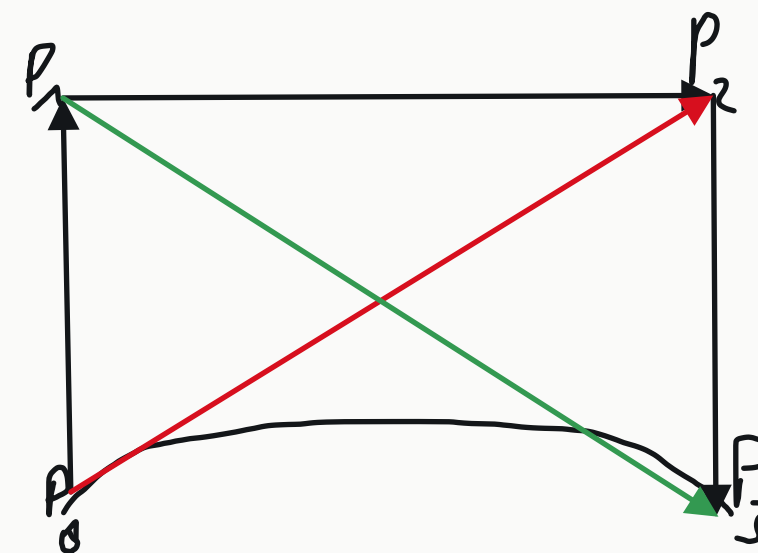
$$= (1-t)^2 P_0 + 2(1-t)t P_1 + t^2 P_2$$



Cubic

$$B_C(t) = (1-t)B_Q(t)_{P_0, P_1, P_2} + tB_Q(t)_{P_1, P_2, P_3}$$

$$= (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t)t^2 P_2 + t^3 P_3$$



S