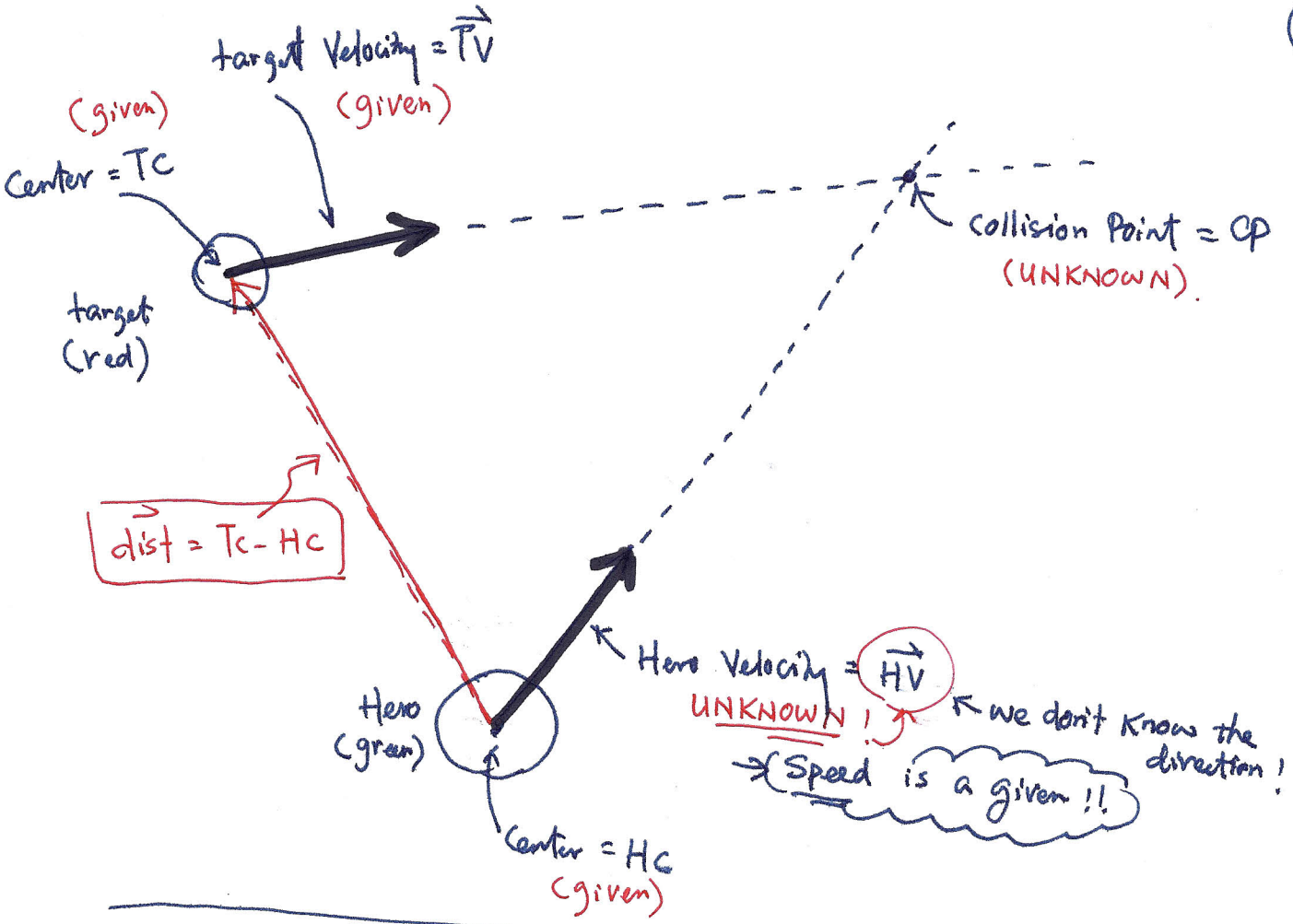
```
        //
        // we want:
        //      hv - hero velocity (hvx, hvy)
        //
        // given:
        //      hs - hero speed
        //      hp - current hero position
        //      tv - target velocity (tvx, tvy)
        //      tp - current target position
        //
        // we know:  hvx*hvx + hvy*hvy = hs*hs
        //
        // we want to find tm, hvx, hvy such that:
        //
        //    tpx + tm * tvx = hpx + tm * hvx
        //    tpy + tm * tvy = hpy + tm * hvy
        //    hvx * hvx + hvy * hvy = hs * hs
        //
        private float ComputeHeroVDirection()
        {
            Vector2 dist = mTheTarget.Center - mTheHero.Center;
            // a * mt * mt + b * mt + c = 0
            float a = mTheTarget.Velocity.LengthSquared() -
                        (mTheHero.Speed * mTheHero.Speed);
            float b = 2f * Vector2.Dot(mTheTarget.Velocity, dist);
            float c = dist.LengthSquared();

            mCollidPt.RemoveFromAutoDrawSet();
            float tm = 0f;

            float dm = (b * b) - (4 * a * c);
            if (dm > 0f)
            {
                dm = (float)(Math.Sqrt(dm));
                float t1 = (-b + dm) / (2f * a);
                float t2 = (-b - dm) / (2f * a);
                if (t1 > 0)
                {
                    if (t2 > 0)
                    {
                        tm = (t1 < t2) ? t1 : t2;
                    }
                    else
                    {
                        tm = t1;
                    }
                }
                else
                {
                    if (t2 > 0)
                        tm = t2;
                }
                if (tm > 0)
                {
                    Vector2 collidPt = mTheTarget.Center + tm * mTheTarget.Velocity;
                    Vector2 toHero = collidPt - mTheHero.Center;
                    mTheHero.Velocity = (1f / tm) * toHero;

                    mCollidPt.Center = collidPt;
                    mCollidPt.AddToAutoDrawSet();
                }
            }
            return tm;
        }
```



Let $\underline{tm}$ - time it takes for target to travel to cp

tm - ..... Hero ..... cp

$$\Rightarrow HC + tm * \vec{HV} = cp \quad -(1)$$

$$TC + tm * \vec{TV} = cp \quad -(2)$$

$$\Rightarrow \boxed{HC + \underline{tm} * \underline{\vec{HV}} = TC + tm * \vec{TV}}$$

unknown — Speed is given — Direction is unknown.

$$HV = \frac{1}{tm}\left[(TC + tm\,\vec{Tv}) - Hc \right] \quad -(3)$$

(3) is a vector equation.

$$HV = \frac{1}{tm}\left[(TC + tm\,\vec{TV}) - Hc\right] \quad \text{—(3)}$$

is actually 2 equations

$$\begin{cases} HV.x = \frac{1}{tm}\left[(TC.x + tm * TV.x) - HC.x\right] & \text{—(4)} \\ HV.y = \frac{1}{tm}\left[(TC.y + tm * TV.y) - HC.y\right] & \text{—(5)} \end{cases}$$

But we know the Speed of the Hero = (given = S)

$$S = \|\vec{HV}\| = \sqrt{HV.x^2 + HV.y^2}$$

or

$$S^2 = HV.x^2 + HV.y^2 \quad \text{—(6)}$$

Substitute (4) + (5) into (6):

$$S^2 = \left\{\frac{1}{tm}\left[Tc.x + tm * TV.x - HC.x\right]\right\}^2 + \left\{\frac{1}{tm}\left[TC.y + tm * TV.y - HC.y\right]\right\}^2$$

The only unknown is tm !!

... simplify + collect terms

$$tm^2 \underbrace{\left[(TV.x^2 + TV.y^2 - s^2)\right]}_{a} + tm * \underbrace{\left[2 * \left[(TC.x - HC.x)(TV.x) + (TC.y - HC.y)(TV.y)\right]\right]}_{b} + \underbrace{\left[(Tc.x - HC.x)^2 - (TC.y - He.y)^2\right]}_{c} = 0$$

$a \cdot tm^2 + b \cdot tm + c = 0$

$a = TV.length()^2 - S^2$

$b = (dist \cdot TV) * 2$

$c = (dist.length)^2$