

Algorithm to create trackable challenge

(1) The first step is avoid very short line. Assume we don't want any line shorter than 100 pixel. For each pair of adjacent points (p_i, p_j) , calculated the distance between p_i and p_j :

$$d = \sqrt{(p_i.x - p_j.x)^2 + (p_i.y - p_j.y)^2}.$$

if $d \leq 100$, then the current challenge is not usable.

(2) The second step is avoid any two adjacent lines too close to each other. If so, there must be two adjacent lines have similar slopes. To split adjacent lines, any two lines share one node can't slope difference smaller than 30 degree. For any point p_i , calculate slope s_- of (p_{i-1}, p_i) and slope s_+ of (p_i, p_{i+1}) . If $|s_- - s_+| \leq 30$, then the current challenge is not usable.

(3) To scussfully spread points over whole screen, I divide the overall screen range into 4 screen parts shown in figure1. For any four points path, we need to make sure there is only one point in each screen part. Path could start from any screen part and end at any screen part. The sequence of screen part is random generated.