

Promptes IA

Prompt1 - calcul : placer mes pensées dans l'écran sans dépasser

Moi :

Je dois placer des pensées aléatoirement dans mon écran, comment je dois me prendre pour que mes éléments restent entièrement visibles à l'écran? j'ai trouvé ces deux sources. explique moi source: <https://stackoverflow.com/questions/32471199/why-is-the-center-of-a-rectangle-xwidth-2-yheight-2-in-java?>
<https://stackoverflow.com/questions/11409895/whats-the-most-elegant-way-to-cap-a-number-to-a-segment?>

AI :

Si tu places une pensée avec une position aléatoire entre 0 et la largeur de l'écran, son point de départ sera visible, mais pas forcément toute la pensée. Comme l'élément a lui-même une largeur, il faut retirer cette largeur de l'espace disponible.

Même principe pour la hauteur : si tu places l'élément trop bas, son texte peut dépasser en dessous.

Les deux sources t'aident à comprendre deux idées :

- La source sur le centre d'un rectangle explique comment on peut utiliser les dimensions d'un élément pour calculer sa position.
- La source sur le "cap" explique qu'il faut limiter une valeur dans une zone autorisée, pour éviter qu'elle devienne trop grande ou trop petite.

Donc la logique est :

1. récupérer la largeur et la hauteur de l'écran ;
2. récupérer la largeur et la hauteur de la pensée ;
3. prévoir une marge sur les bords ;
4. calculer l'espace disponible en retirant la taille de la pensée et les marges ;
5. choisir une position aléatoire uniquement dans cet espace ;
6. bloquer le résultat si jamais l'espace devient négatif.

Comme ça, la pensée peut apparaître à une position aléatoire, mais elle reste entièrement visible à l'écran

Prompt2 - calcul : placer mes pensées dans l'écran sans dépasser

Moi :

ok merci, est ce que tu peut me faire un exemple pour le calcule?

AI :

Exemple de calcul

Imaginons :

- largeur de l'écran = **1000 px**
- largeur de la pensée = **200 px**
- marge à gauche et à droite = **50 px**

Étape 1 : Calculer l'espace disponible

On retire la largeur de la pensée et les marges :

$$1000 - 200 - (50 \times 2)$$

$$1000 - 200 - 100$$

$$700 \text{ px}$$

Il reste donc **700 px** dans lesquels la pensée peut être placée.

Étape 2 : Choisir une position aléatoire

Supposons que le générateur aléatoire donne :

$$0,5$$

On calcule :

$$0,5 \times 700 = 350$$

Étape 3 : Ajouter la marge

$$350 + 50 = 400$$

La pensée sera placée à :

$x = 400 \text{ px}$

Vérification

La pensée mesure :

200 px

Donc son bord droit sera :

$400 + 200 = 600 \text{ px}$

Comme l'écran fait :

1000 px

La pensée reste entièrement visible.

Cas limite

Imaginons :

largeur écran = 300 px

largeur pensée = 400 px

marge = 50 px

Calcul :

$300 - 400 - 100 = -200$

On obtient une valeur négative.

Grâce à la logique de limitation expliquée dans la deuxième source, on remplace alors :

-200

par :

0

Cela évite de générer une position impossible et empêche les erreurs de calcul.

Prompt1 - Calcul pour repousser les pensées à l'approche de la souris

Moi :

je dois faire en sorte que a l'approche de la souris les pensées s'éloigne dans le sens inverse de la souris pour avoir un effet de repoussement ? j'ai trouver une source.

explique moi source: <https://codeguppy.com/blog/how-to-calculate-the-distance-between-two-points-using-javascript-and-p5.js/>

AI :

Oui, tu peux formuler la réponse comme ça :

Pour créer un effet de repoussement, il faut d'abord savoir si la souris est proche d'une pensée. La source sur le calcul de distance entre deux points permet justement de comprendre comment mesurer l'écart entre la position de la souris et la position d'un élément.

Si on utilise directement le coin de la pensée, le résultat est moins précis. Il est donc plus logique de partir du centre de la pensée, en tenant compte de sa largeur et de sa hauteur.

La source t'aide à comprendre deux idées :

- calculer la distance entre deux points grâce à leurs coordonnées ;
- utiliser cette distance pour déclencher une action seulement si la souris est assez proche.

Donc la logique est :

1. récupérer la position et la taille de la pensée ;
2. calculer le centre de la pensée ;
3. récupérer la position de la souris ;
4. calculer l'écart horizontal et vertical entre la souris et la pensée ;
5. calculer la distance réelle entre les deux points ;
6. si la distance est assez petite, déplacer la pensée dans le sens opposé à la souris.

Comme ça, la pensée ne réagit pas tout le temps, mais uniquement quand la souris s'approche d'elle, ce qui donne un effet de repoussement.

Prompt2 - Calcul pour repousser les pensées à l'approche de la souris

Moi :

ok merci, explique moi le calcul dans la source: `Math.sqrt((x2 - x1)*(x2 - x1) + (y2 - y1)*(y2 - y1));`

AI :

Bien sûr. Cette formule est la formule du **théorème de Pythagore** appliquée à deux points.

`Math.sqrt((x2 - x1)*(x2 - x1) + (y2 - y1)*(y2 - y1));`

La source cherche à calculer la distance entre :

- Point A : (x1 ; y1)
 - Point B : (x2 ; y2)
-

Exemple

Imaginons :

Souris : (250 ; 200)

Pensée : (400 ; 300)

Donc :

$x1 = 250$

$y1 = 200$

$x2 = 400$

$y2 = 300$

Étape 1 : Calculer l'écart horizontal

$x2 - x1$

$400 - 250 = 150$

Cela signifie que les deux points sont séparés de **150 px horizontalement**.

Étape 2 : Calculer l'écart vertical

$$y_2 - y_1$$

$$300 - 200 = 100$$

Cela signifie que les deux points sont séparés de **100 px verticalement**.

Étape 3 : Mettre les écarts au carré

La source fait :

$$(x_2 - x_1) * (x_2 - x_1)$$

donc :

$$150 \times 150 = 22500$$

et :

$$(y_2 - y_1) * (y_2 - y_1)$$

$$100 \times 100 = 10000$$

Le carré permet d'obtenir une valeur toujours positive et prépare l'application du théorème de Pythagore.

Étape 4 : Additionner

$$22500 + 10000 = 32500$$

Étape 5 : Calculer la racine carrée

La source utilise :

`Math.sqrt(...)`

qui signifie :

$$\sqrt{32500}$$

Résultat :

$$\approx 180,3$$

