

MATHEMATIQUE

FLASH

PRINCIPE

La syntaxe de l'ActionScript orientée pour la Mathématique permet d'opérer **des calculs mathématiques un peu plus avancés**.

Leur usage permet d'intégrer des notions très intéressantes en particulier pour le Jeux-Vidéos et l'Animation.

DESCRIPTION

La plupart des notions Mathématiques en ActionScript sont appelées sous forme de Fonction par l'objet nommé "**Math**", parfois en y ajoutant des arguments.

Certaines de ces méthodes, comme "**Math.random()**" permettant d'obtenir une valeur aléatoire entre **1** et **0**, ont un effet immédiat.

D'autres de ces méthodes, comme "**Math.cos(X)**" permettant d'obtenir le cosinus d'une valeur, nécessitent des mises en applications spécifiques (*Décrites plus bas*) pour en obtenir des résultats appréciables.

UTILISATION & SYNTAXE

Calculs Standard

Math.round(X) → Arrondi la valeur **X** à la valeur entière la plus proche

Math.ceil(X) → Arrondi la valeur **X** au supérieur

Math.floor(X) → Arrondi la valeur **X** à l'inférieur

Math.abs(X) → Met la valeur **X** en absolu (*N'est plus négatif*)

Math.sqrt(X) → Racine carrée de **X**

Math.cos(X) → Cosinus de **X**

Math.sin(X) → Sinus de **X**

Math.atan2(X) → Tangente de **X**

Math.PI → Récupère PI

Aléatoire

Math.random() → Retourne une valeur aléatoire entre 0 et 1

Math.random() * X → Retourne une valeur aléatoire entre 0 et **X**

Math.random() * (X + Y) - Y → Retourne une valeur aléatoire entre **Y** et **X**

Astuces de Calculs

Etat += (NouvelEtat - Etat) / ContreVitesse;

Exemple : myClip.x += (100 - myClip.x) / 10;

Utilisé en boucle, permet de diriger une valeur vers une autre de manière fluide. La valeur va ralentir dans sa progression au fur et à mesure qu'elle se rapproche de la valeur à atteindre.

Math.atan2((CibleB.y - CibleA.y) , (CibleB.x - CibleA.x)) * 180 / Math.PI;

Exemple : myClip.rotation = Math.atan2(mySecondClip.y - myClip.y , mySecondClip.x - myClip.x) * 180 / Math.PI;

Retourne l'angle de rotation entre deux clips.

Vitesse *= Frein / 100;

Exemple : myClip.vitesse *= 90 / 100;

Utilisé en boucle, permet de faire progressivement tendre une valeur vers zéro.

Plus la valeur de "Frein" est faible et plus la Vitesse attendra zéro rapidement.

Si Frein dépasse 100, la Vitesse augmentera au lieu de réduire.

**Math.sqrt(Math.abs(CibleA.x - CibleB.x) * Math.abs(CibleA.x - CibleB.x)
+ Math.abs(CibleA.y - CibleB.y) * Math.abs(CibleA.y - CibleB.y));**

Exemple : Math.sqrt(Math.abs(myClip.x - mySecondClip.x) * Math.abs(myClip.x - mySecondClip.x)
+ Math.abs(myClip.y - mySecondClip.y) * Math.abs(myClip.y - mySecondClip.y));

Retourne la distance totale entre deux clips.

Cible.x += Math.cos(Angle / 180 * Math.PI) * Force;

Cible.y += Math.sin(Angle / 180 * Math.PI) * Force;

Exemple : myClip.x += Math.cos(myClip.rotation / 180 * Math.PI) * 3;

myClip.y += Math.sin(myClip.rotation / 180 * Math.PI) * 3;

Utilisé en boucle, fait se diriger un clip dans la direction définie par la valeur de l'Angle (Qui peut être sa propre rotation).