

Liste des erreurs courantes dans Unity et solutions à envisager

Des erreurs, des avertissements et des informations

Vous devez distinguer trois sortes de messages dans la console :

- Les erreurs, avec un logo rouge. Ils vont bloquer votre programme. Il faut les corriger impérativement. Il est même possible que vous ne puissiez pas faire PLAY.
- Les avertissements, avec le logo jaune. Unity repère quelque chose d'étrange, mais il estime être capable d'exécuter le programme. Ce peut-être un bon repère pour anticiper de vraies erreurs. Elles sont aussi parfois liée à des fonctions ou variables que l'on n'utilise plus, mais qui semblent encore appelées.
- Les informations, en gris. Ce sont des messages précis que Unity peut vous envoyer. Vous lirez plus souvent ces messages de par votre propre intention. En effet, il est possible d'envoyer des messages dans la console, c'est TRES IMPORTANT pour repérer et éliminer les erreurs trop cachées.

Exploitez les informations de console. La commande suivante :

```
Debug.Log("Bonjour, je suis un lapin.");
```

Vous permet d'envoyer un message dans la console. Vous pouvez l'employer pour des cas typiques suivants :

- Vérifier que votre script arrive jusqu'à la ligne de commande indiquée.
- Vérifier qu'une instruction a bien lieu.
- ~~Envoyer des messages rigolos à vos copains.~~

Le caractère manquant

Unexpected symbol `}' in class, struct, or interface member declaration

Lorsque vous écrivez une instruction, vous avez souvent besoin d'indiquer des caractères spéciaux tel que le point-virgule (;) ou l'accolade { }. Ces caractères sont essentiels au programme pour comprendre ce que vous lui demandez. Ce message d'erreur apparaît lorsqu'une instruction semble manquer un caractère essentiel.

Traitement :

- Double cliquez sur le message d'erreur et vérifiez la ligne indiquée.
- Vérifiez s'il ne manque pas un point-virgule, une accolade, une parenthèse, etc.
- Si la ligne vous semble correcte, remontez les lignes jusqu'à trouver un éventuel oubli de caractère spécial.

La mauvaise valeur

Constant value 'true' cannot be converted to a 'float'

Chaque variable ne peut indiquer qu'un nombre précis d'informations. Par exemple, si la variable ne peut contenir que des chiffres, vous ne pourrez pas lui indiquer des lettres.

Traitement :

- Vérifiez le type de la variable s'il correspond bien à votre souhait.
- Vérifiez la valeur du type de variable dans le cas contraire.

Un type inexistant

The type or namespace name 'MonTruc' could not be found. Are you missing a using directive or an assembly reference?

En C#, comme dans tout langage, vous avez un ensemble précis et déterminé de type que vous pouvez utiliser. Vous ne pouvez pas en inventer spontanément. Il faudrait passer un sorte d'officialisation de la création de ce type. Ce message apparaît parce que vous faites référence à un type inexistant ou mal écrit.

Traitement :

- Vérifiez la bonne orthographe du type de variable à la ligne indiquée.
- Limitez-vous aux types employés.
- Si vous savez le faire, déclarez le type avant de l'exploiter.

Mettre un cube dans un trou en forme d'étoile

Argument '#1' cannot convert 'bool' expression to type 'UnityEngine.Vector3'

Parfois, par inadvertance, par égarement, par mésentente avec sa mémoire, il peut nous arriver de vouloir utiliser une variable inappropriée pour une fonction. Il vous faut bien connaître ce que peut faire la fonction avant de l'employer, sinon vous risquez d'exploiter des valeurs qui ne seront pas comprises.

Traitement :

- Retournez à la ligne de code, retirez la valeur que vous avez indiqué et remplacez la par la bonne.
- Si vous ne savez pas quoi faire, vous pouvez supprimer, la fonction et en la réécrivant, vous remarquez que MonoDevelop ou Visual Studio vous proposent dans un petit rectangle l'expression complète de votre fonction. C'est un moyen de vous repérer sur les valeurs que vous pouvez indiquer.
- Dans Help>Scripting Reference, vous trouverez votre fonction, si elle provient d'Unity, ainsi que son écriture.