



**Forum: WD 10**

**Topic: Thread**

**Subject: Re: Thread**

Posté par: Celtic

Contribution le : 18/5/2006 9:35:00

Cela ne marche pas se qui pose probleme c'est que plusieurs thread utilise la meme methode et c'est pour ca qu'il me jete.

Je veux remplacer mon Tantque par un thread dans ma classe.

voila mon code c'est pas très propre je vous l'accorde...

```
CLSuiCursor est une classe GLOBAL str_obj est une chaîne str_msg est une chaîne lstr_mot,type
est un entier Tstr_tab est un tableau dynamique FIN PROCEDURE Constructeur(obj,msg est une
chaîne) ::str_obj = obj ::str_msg = msg {::str_obj}..Valeur = ::str_msg ::lstr_mot = Taille(::str_msg)
i,oc est un entier str_tmp est chaîne oc=0 POUR i = 1 A ::lstr_mot oc++ FIN ::Tstr_tab=allouer un
tableau de oc chaîne POUR i = 1 A ::lstr_mot str_tmp="l"+i ::Tstr_tab[i]=str_tmp
ChampClone({::str_obj},str_tmp,50,50) {str_tmp}..Valeur = Milieu(::str_msg,i,1) FIN
AnimationChangeType({::str_obj},actDéfilement) AnimationLitParamètre({::str_obj},acpBoucle)
{::str_obj}..Animation = Vrai InitHasard() TimerSys("::SuisCur",10,10)
TimerSys("::Changecolor",50,11) PROCEDURE Destructeur() PROCEDURE GLOBAL SuisCur()
nPos, nPosX, nPosY, i sont des entiers nPos = CurseurPos(cpFenêtre) nPosX = PoidsFaible(nPos)
nPosY = PoidsFort(nPos) POUR i=1 A ::lstr_mot SI i = 1 ALORS {::Tstr_tab[i]}..X = nPosX
{::Tstr_tab[i]}..Y = nPosY SINON ::Mouv(i) FIN FIN PROCEDURE GLOBAL Changecolor()
mapolice est une Police R,V,B sont des entiers sur 1 octet R=Hasard(255) V=Hasard(255)
B=Hasard(255) mapolice..Couleur = RVB(R,V,B) mapolice..Taille = 14 i est un entier POUR i=1 A
::lstr_mot {::Tstr_tab[i]}..Police = mapolice FIN PROCEDURE GLOBALE Mouv(i) tmp est un entier =
Hasard(1000000) SI {::Tstr_tab[i]}..X > {::Tstr_tab[i-1]}..X + 10 ALORS TANTQUE {::Tstr_tab[i]}..X
<> {::Tstr_tab[i-1]}..X + 10 {::Tstr_tab[i]}..X = {::Tstr_tab[i]}..X-1 FIN
//ThreadExécute(i+tmp,threadNormal,::MouvXP,i,i+tmp) FIN SI {::Tstr_tab[i]}..X < {::Tstr_tab[i-1]}..X
+ 10 ALORS TANTQUE {::Tstr_tab[i]}..X <> {::Tstr_tab[i-1]}..X + 10 {::Tstr_tab[i]}..X =
{::Tstr_tab[i]}..X+1 FIN //tmp=Hasard(1000000)
//ThreadExécute(i+tmp,threadNormal,::MouvXM,i,i+tmp) FIN SI {::Tstr_tab[i]}..Y > {::Tstr_tab[i-1]}..Y
ALORS TANTQUE {::Tstr_tab[i]}..Y <> {::Tstr_tab[i-1]}..Y {::Tstr_tab[i]}..Y -- FIN FIN SI
{::Tstr_tab[i]}..Y < {::Tstr_tab[i-1]}..Y ALORS TANTQUE {::Tstr_tab[i]}..Y <> {::Tstr_tab[i-1]}..Y
{::Tstr_tab[i]}..Y ++ FIN FIN PROCEDURE GLOBALE MouvXP(i,nom) TANTQUE {::Tstr_tab[i]}..X
<> {::Tstr_tab[i-1]}..X + 10 {::Tstr_tab[i]}..X = {::Tstr_tab[i]}..X-1 FIN ThreadArrête(nom)
PROCEDURE GLOBALE MouvXM(i,nom) TANTQUE {::Tstr_tab[i]}..X <> {::Tstr_tab[i-1]}..X + 10
{::Tstr_tab[i]}..X = {::Tstr_tab[i]}..X+1 FIN ThreadArrête(nom)
```