

Tutorial Préchargement de Texture, Proto C3D_ResourceLoader (Brian Hay)
Optimisé pour BS Contact et Internet Explorer.



Petit Rappel :

Lorsqu'un fichier vrmf est ouvert en ligne, bien souvent vous apercevez la géométrie sans texture, et au fur et à mesure du téléchargement les textures apparaissent les unes après les autres. La rapidité de cette action dépend de votre débit de connexion (exprimé en Mbit/s) et de la taille des fichiers images (Ko, Mo).

Un exemple : Vous disposez d'un débit Internet de 2 Mbit/s vous chargez le monde zinzin en vrmf, la géométrie pèse 200Ko et les textures 2Mo en tout.

En sachant que 1octet = 8bits et que 1000Ko = 1Mo

$(\text{Poids du fichier} \times 8) / \text{Vitesse de connexion} = \text{temps de chargement idéal.}$

Pour charger la géométrie il faut $(200 \times 8) / 2000 = 0,8$ seconde

Pour charger les textures $(2000 \times 8) / 2000 = 8$ secondes

Durant 8 secondes vos géométries seront composées d'une mosaïque d'éléments semi texturés, et le chargement provoquera des saccades.

A noter : les textures sont ensuite placées dans le cache de votre plug'in Vrmf ainsi que dans les fichiers temporaires de votre Browser Internet (IE, FireFox...). Lors d'un nouveau chargement du même espace vrmf, les textures sont récupérées de votre disque dur, le chargement est ainsi quasi immédiat.

Voici un exemple avec 8 cubes texturés, chaque image jpg pèse environ 1 Mo afin de réellement tester l'efficacité du pré chargement.

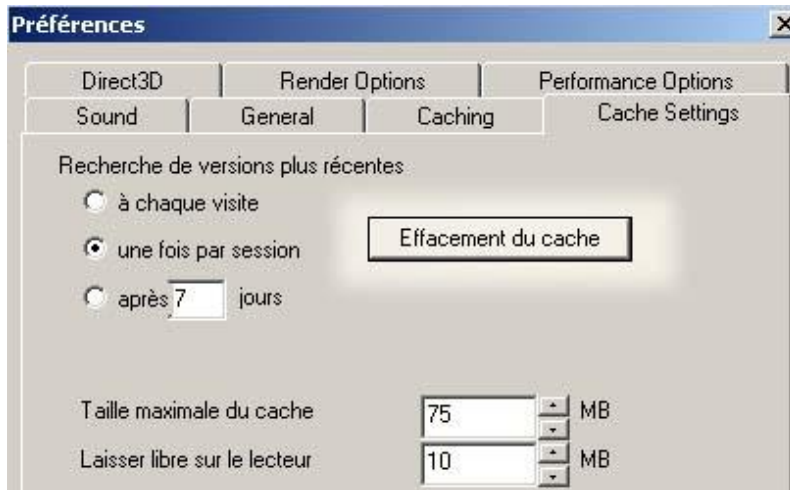
Pour comparer l'effet avec ou sans preload, tentez d'abord de visualiser cette même scène sans préchargement.

<http://christler.cedric.free.fr/test/test0.wrl>

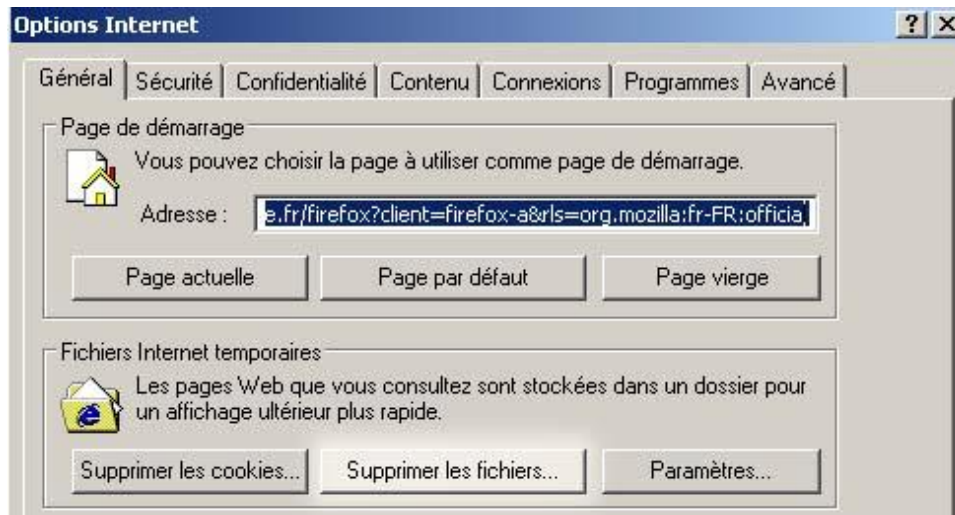


CQFD Les textures apparaissent les unes après les autres.

A présent videz votre cache (haut les mains ! :D) laissez votre fenêtre avec votre scène vrmf sous Bs contact ouverte bouton de droite>Configuration>Préférence>Cache Settings>Effacement du Cache



Ensuite avec Internet Explorer dans Outils>Options Internet>Général>Supprimer les fichiers.



Voilà qui est fait !

Vous pouvez tester à présent la version avec préchargement :

<http://christler.cedric.free.fr/test/test.wrl>



haaa voilà qui est mieux.

Comment mettre en place ce préchargement ?

Il faut tout d'abord récupérer ces 3 protos ici :

http://christler.cedric.free.fr/test/proto_C3D/

Comment cela fonctionne en gros :

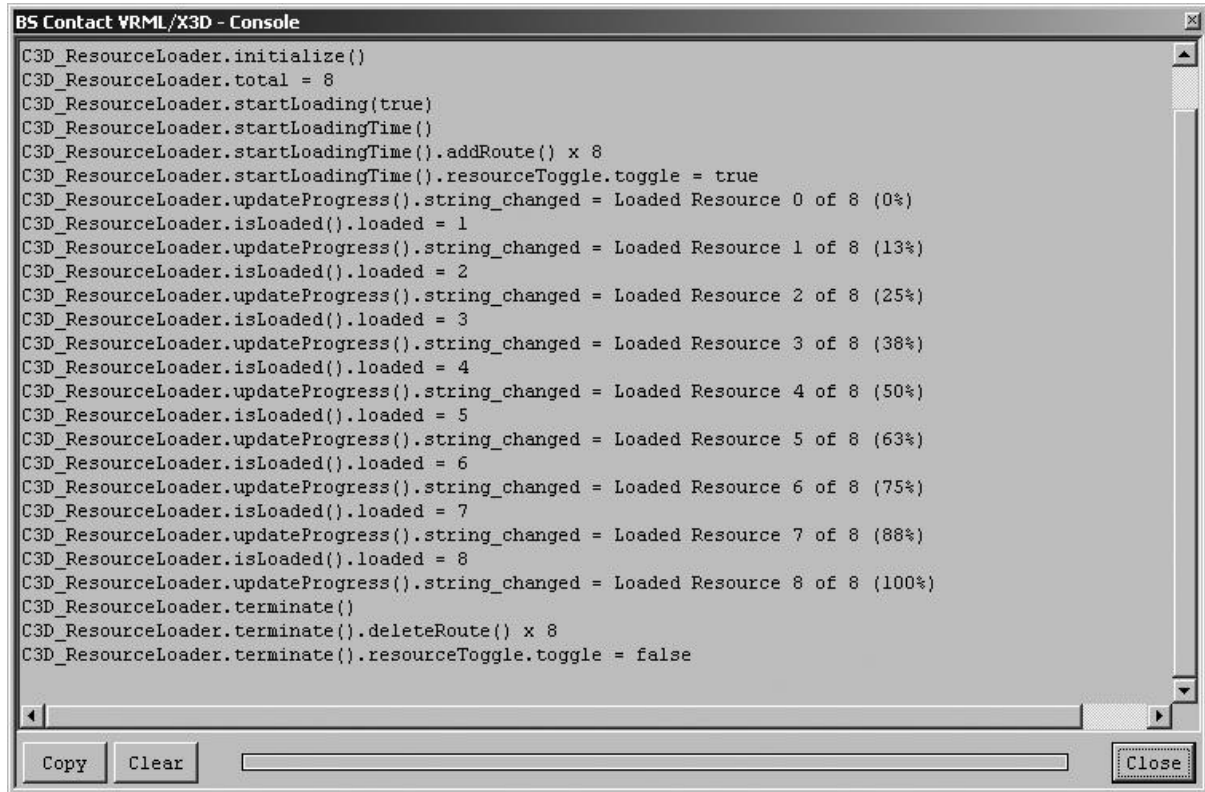
Un nœud children est créé dans lequel on place vos images une à une.

Le C3D_ResourceLoader utilise alors la fonction de Bs contact Browser.prefetch qui pré charge réellement le fichier.

Cette opération se renouvelle autant de fois qu'il y a d'images, et incrémente la valeur du %.

Selon le % la barre de progression progresse :D

A 100% le programme se termine et supprime le nœud children. Il n'y a plus qu'à laisser apparaître votre création vrml.



```
BS Contact VRML/X3D - Console
C3D_ResourceLoader.initialize()
C3D_ResourceLoader.total = 8
C3D_ResourceLoader.startLoading(true)
C3D_ResourceLoader.startLoadingTime()
C3D_ResourceLoader.startLoadingTime().addRoute() x 8
C3D_ResourceLoader.startLoadingTime().resourceToggle.toggle = true
C3D_ResourceLoader.updateProgress().string_changed = Loaded Resource 0 of 8 (0%)
C3D_ResourceLoader.isLoaded().loaded = 1
C3D_ResourceLoader.updateProgress().string_changed = Loaded Resource 1 of 8 (13%)
C3D_ResourceLoader.isLoaded().loaded = 2
C3D_ResourceLoader.updateProgress().string_changed = Loaded Resource 2 of 8 (25%)
C3D_ResourceLoader.isLoaded().loaded = 3
C3D_ResourceLoader.updateProgress().string_changed = Loaded Resource 3 of 8 (38%)
C3D_ResourceLoader.isLoaded().loaded = 4
C3D_ResourceLoader.updateProgress().string_changed = Loaded Resource 4 of 8 (50%)
C3D_ResourceLoader.isLoaded().loaded = 5
C3D_ResourceLoader.updateProgress().string_changed = Loaded Resource 5 of 8 (63%)
C3D_ResourceLoader.isLoaded().loaded = 6
C3D_ResourceLoader.updateProgress().string_changed = Loaded Resource 6 of 8 (75%)
C3D_ResourceLoader.isLoaded().loaded = 7
C3D_ResourceLoader.updateProgress().string_changed = Loaded Resource 7 of 8 (88%)
C3D_ResourceLoader.isLoaded().loaded = 8
C3D_ResourceLoader.updateProgress().string_changed = Loaded Resource 8 of 8 (100%)
C3D_ResourceLoader.terminate()
C3D_ResourceLoader.terminate().deleteRoute() x 8
C3D_ResourceLoader.terminate().resourceToggle.toggle = false
```

Vous devez donc répertorier toutes vos images et les placer dans le nœud children du C3D_ResourceLoader.

Vous pouvez vous-même créer un proto de barre de chargement en dirigeant le SFFloat loader.fraction_changed vers ce que vous voulez (rotation[3] , scale.x , translation[0] ...)

Bon C3D_ResourceLoader à tous ;)

Encore un grand merci à Brian Hay (www.construct3d.com) pour son proto.

Merci à Sylvie pour sa relecture.

```
#VRML V2.0 utf8
EXTERNPROTO Layer3D[
exposedField SFVec3f bboxSize
exposedField SFVec3f bboxCenter
eventIn MFNode addChildrenLayer
eventIn MFNode removeChildrenLayer
exposedField MFNode childrenLayer
exposedField SFVec2f translation
exposedField SFInt32 depth
exposedField SFVec2f size
exposedField SFNode background
```

```

exposedField SFNode fog
exposedField SFNode navigationInfo
exposedField SFNode viewpoint
exposedField MFNode children
eventIn MFNode addChildren
eventIn MFNode removeChildren
]
[
"urn:inet:blaxxun.com:node:Layer3D"
"http://www.blaxxun.com/vrml/protos/nodes.wrl#Layer3D"
]
#Le Layer3d sert à créer cet effet de fondu pour la transition du preload à la scène vrml

```

```

EXTERNPROTO C3D_ResourceLoader [
    eventIn SFBool startLoading
    eventIn SFTime startLoadingTime
    eventOut SFFloat fraction_changed
    eventOut SFBool isFinished
    eventOut SFTime finishedTime
    eventOut MFString string_changed
    field SFBool debug # affiche dans la console de Bs Contact les infos de chargement
    field MFNode children # Noeud dans lequel vous placez vos ImagesTextures
    field SFTime timeoutInterval # terminate loader after this time, regardless of how many
assets still to load

```

```

]
"C3D_ResourceLoader_1.07.wrl#C3D_ResourceLoader"

```

#Placez ici toutes vos textures utilisées dans votre monde vrml

```

DEF loader C3D_ResourceLoader {
    debug FALSE
    children [
        ImageTexture {
            url ["http://cedric.christler.free.fr/sccg/stpierrein1.jpg"]
        }
        ImageTexture {
            url ["http://cedric.christler.free.fr/sccg/stpierrein2.jpg"]
        }
        ImageTexture {
            url ["http://cedric.christler.free.fr/sccg/stpierrein3.jpg"]
        }
        ImageTexture {
            url ["http://cedric.christler.free.fr/sccg/stpierreout1.jpg"]
        }
        ImageTexture {
            url ["http://cedric.christler.free.fr/sccg/stpierreout2.jpg"]
        }
        ImageTexture {
            url ["http://cedric.christler.free.fr/sccg/stpierreout3.jpg"]
        }
        ImageTexture {
            url ["http://cedric.christler.free.fr/sccg/stpierreout4.jpg"]
        }
        ImageTexture {

```

```

    url ["http://cedric.christler.free.fr/sccg/stpierreout5.jpg"]
  }
]
  timeoutInterval 10
}

```

#PROTO C3D_BarreChargement par Cédric CHRISTLER, une barre se remplit progressivement à partir de son milieu, le fond blanc devient transparent quand le chargement est à 100%

```

PROTO C3D_BarreChargement [
  eventIn SFFloat fraction_changed

  ]
  {
    DEF preloadok Switch {
      whichChoice 0
      choice [
        Layer3D {
          size 2 2
          children [
            Transform {
              translation 0 0 0
              children [
                Shape {
                  appearance Appearance {
                    material DEF fondublan debut Material {
                      ambientIntensity 1
                      diffuseColor 1 1 1
                      emissiveColor 1 1 1
                      transparency 0
                    }
                  }
                  geometry Box {
                    size 10 10 .1 }
                }
              ]
            }
          ]
        }
      ]
    }
  }
}

```

```

  Transform { translation 0 0 1.2
    scale .1 .1 .1
    children [
      DEF barre2 Transform { translation 0 0 0
        translation 0 0 0
        scale 3.55 .1 .1
        children [Shape {
          appearance Appearance {
            material Material {
              diffuseColor 1 0.878 0.098
              emissiveColor 1 0.878 0.098
            }
            texture ImageTexture {

```

```

        url ["http://christler.cedric.free.fr/test/loading.gif",]
    }
    textureTransform TextureTransform {
        scale 8 1
    }
}
geometry Box {
    size 10 10 .1
    }

    geometry Box {

    }

}}
}
DEF barre1 Transform { translation 0 0 0
translation 0 0 0
scale 0 .09 .11
children [Shape {
    appearance Appearance {
        material Material {
            transparency .2
            diffuseColor 0 0.26 0.564
            emissiveColor 0 0.26 0.564
        }
    }
    geometry Box {

    }

}}
} ]
}
DEF timok TimeSensor {
loop TRUE
} ]
}

DEF scripto Script {
field SFNode bar1 USE barre1
field SFNode bar2 USE barre2
field SFNode fondblanc USE fondublanccdebut
field SFNode timeok USE timok
field SFNode preloadhide USE preloadok
eventIn SFFloatfraction_changed IS fraction_changed
eventIn SFFloattimeinput
url ["vrmlscript:
function fraction_changed(b){
bar1.scale.x=b*3.5;
if(b==1){bar1.scale.x=0;
bar2.scale.x=0;
showstart=0;
}}

```

```

function timeinput(t){

    bar2.children[0].appearance.textureTransform.translation.x=bar2.children[0].appearance.textureTransform.translation.x-.001;
    if ((bar2.scale.x==0) &&
(fondblanc.transparency<1.1)){fondblanc.transparency=fondblanc.transparency+.01;
    }

    if (fondblanc.transparency>0.01 );
    if (fondblanc.transparency>1) {timeok.loop=FALSE;
    preloadhide.whichChoice= -1;
    }
}

    "]
}
ROUTE timok.fraction_changed TO    scripto.timeinput
}

DEF barre C3D_BarreChargement    {}
ROUTE loader.fraction_changed TO barre.fraction_changed

```

#Il n'y a plus qu'à placer ici votre monde vrml

```

DEF  cube1 Transform    { translation 15 0 0
    children [Shape {
        geometry Box {

        }
        appearance Appearance {
            texture ImageTexture {
                url    ["http://cedric.christler.free.fr/sccg/stpierrein1.jpg"]
            }
        }
    }]
}

DEF  cube2 Transform    { translation 12 0 0
    children [Shape {
        geometry Box {

        }
        appearance Appearance {
            texture ImageTexture {
                url    ["http://cedric.christler.free.fr/sccg/stpierrein2.jpg"]
            }
        }
    }]
}

DEF  cube3 Transform    { translation 9 0 0
    children [Shape {
        geometry Box {

        }
        appearance Appearance {
            texture ImageTexture {

```

```

        url    ["http://cedric.christler.free.fr/sccg/stpierrein3.jpg"]
    }
}
}}
}
DEF  cube4 Transform    { translation 6 0 0
children [Shape {
    geometry Box {

    }
    appearance Appearance {
        texture ImageTexture {
            url    ["http://cedric.christler.free.fr/sccg/stpierreout1.jpg"]
        }
    }
}
}}
}
DEF  cube5 Transform    { translation 3 0 0
children [Shape {
    geometry Box {

    }
    appearance Appearance {
        texture ImageTexture {
            url    ["http://cedric.christler.free.fr/sccg/stpierreout2.jpg"]
        }
    }
}
}}
}
DEF  cube6 Transform    { translation 0 0 0
children [Shape {
    geometry Box {

    }
    appearance Appearance {
        texture ImageTexture {
            url    ["http://cedric.christler.free.fr/sccg/stpierreout3.jpg"]
        }
    }
}
}}
}
DEF  cube7 Transform    { translation -3 0 0
children [Shape {
    geometry Box {

    }
    appearance Appearance {
        texture ImageTexture {
            url    ["http://cedric.christler.free.fr/sccg/stpierreout4.jpg"]
        }
    }
}
}}
}
DEF  cube8 Transform    { translation -6 0 0
children [Shape {

```



```
    geometry Box {  
    }  
    appearance Appearance {  
        texture ImageTexture {  
            url ["http://cedric.christler.free.fr/sccg/stpierreout5.jpg"]  
        }  
    }  
}]]  
}
```