

## Créer un afficheur à leds 7 segments avec RWIE

Sonelec-Musique > Logiciels > Librairies pour Proteus / RWIE

Dernière mise à jour : 23/03/2007

### Présentation

Dans ce tutoriel, nous allons apprendre à créer un modèle 3D d'un afficheur à leds sept segments, avec le logiciel RealWorld Icon Editor (RWIE, disponible sur le site <http://www.rw-designer.com>). Une fois l'objet créé avec RWIE, nous verrons comment l'exporter au format 3DS pour pouvoir l'utiliser dans le visualiseur 3D de Ares, le logiciel de création de circuit imprimé de la suite Proteus.

A partir du modèle unique créé avec RWIE, nous verrons qu'il est possible de définir le chiffre à afficher et la couleur des segments, et produire autant de fichiers 3DS que désiré : un modèle 3D avec affichage du chiffre 7 en vert, un autre modèle 3D avec affichage du chiffre 2 en rouge, etc.

**Remarque :** il est possible de créer un objet plus simple que celui que nous construirons ici, qui présenterait un chiffre fixe (affichant toujours le chiffre 8). Mais tant qu'à faire, autant mettre en oeuvre quelques mécanismes internes au logiciel RWIE, pour montrer ce dont il est capable.

### Définition de la structure du modèle 3D

Nous voulons créer un objet 3D qui ressemble à un afficheur 7 segments à leds de type HP-5082-76xx, et qui pourra être utilisé dans le visualiseur 3D de Ares (Proteus). Pour cela, nous allons définir trois groupes d'objets (containers) qui une fois assemblés, constitueront l'afficheur :

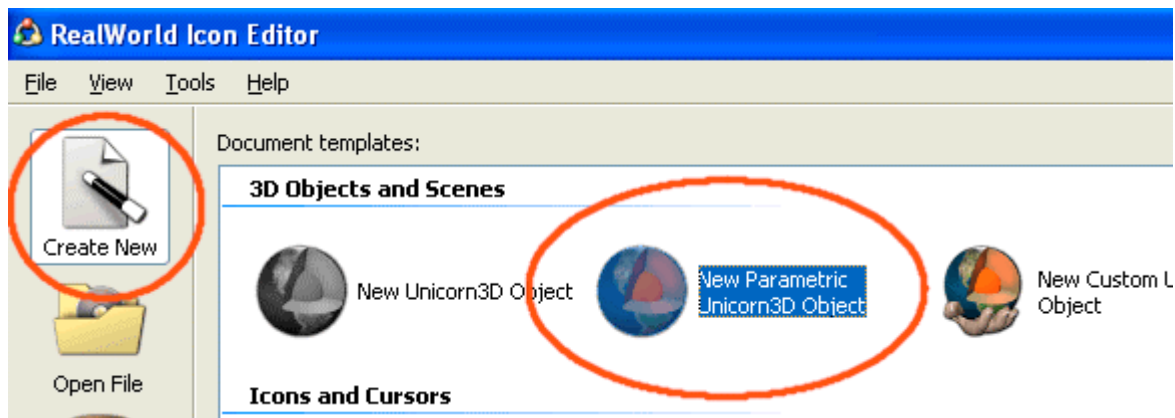
- un groupe d'objets représentant le corps principal du composant,
- un groupe d'objets représentant les segments lumineux,
- un groupe d'objets représentant les connexions électriques.

### Création de la structure générale du modèle

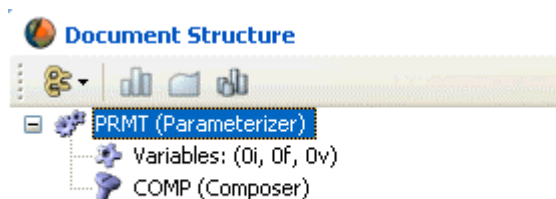
RWIE permet de travailler avec des primitives placées à un même niveau hiérarchique, ou avec des primitives placées dans différents niveaux de hiérarchie. Bien qu'il soit tout à fait possible de créer toutes les primitives au même niveau de hiérarchie, il est conseillé de structurer les éléments par groupe, comme vous devriez le faire avec vos documents informatiques situés sur votre disque dur (pour ne pas mélanger vos documents Word avec les images de votre appareil photo numérique, par exemple). Nous allons donc décider d'une structure de base possédant trois groupes (trois containers), qui est la suivante :

```
| - HP-5087-76xx (Level 0)
  | - Body (Level 1)
    | - Part_01 (Level 2)
  | - Segments (Level 1)
    | - Seg_01 (Level 2)
    | - Seg_02 (Level 2)
    | - ...
    | - Seg_07 (Level 2)
  | - Wires (Level 1)
    | - Wire_01 (Level 2)
    | - Wire_02 (Level 2)
    | - ...
    | - Wire_14 (Level 2)
```

Le choix du nom donné à chaque élément est arbitraire, j'ai choisi de donner des noms anglais pour que le tutoriel, avec ses copies d'écran, puisse aussi être écrit en anglais. Créons donc cette structure de base, en commençant par définir un nouveau projet de type Parametric Unicorn3D object, ce qui se fait en sélectionnant Create New, puis en double-cliquant sur l'icône New Parametric Unicorn3D object.

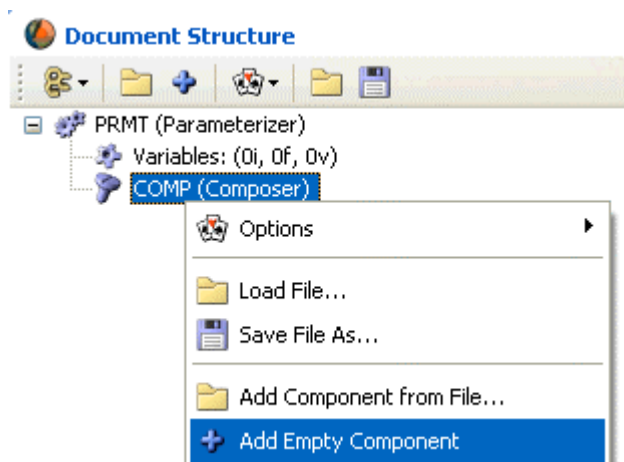


Un nouveau squelette de structure, appelé PRMT (Parameterizer) est créé. Le nouveau squelette contient un container de variables appelé Variables et un container de composants appelé COMP (Composer).

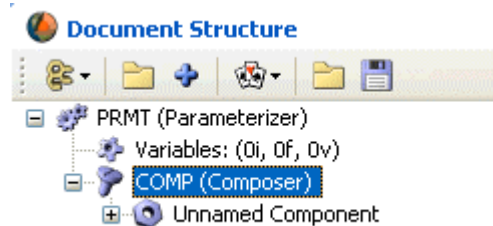


Laissons de côté pour le moment, le container Variables, et intéressons-nous au container COMP. Dans le container COMP, nous allons créer un nouveau container qui contiendra par la suite les trois groupes (containers) d'objets. Nous ne créons pas les groupes d'objets directement à ce premier niveau, ce qui nous permettra par la suite plusieurs types de manipulations sur l'objet final, dans sa totalité.

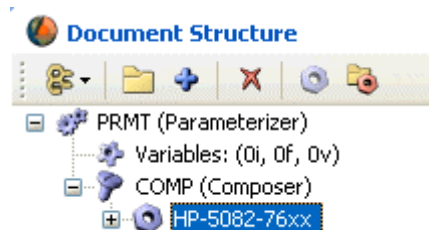
Cliquer droit sur le container COMP de façon à afficher le menu contextuel, puis activer la commande Add Empty component.



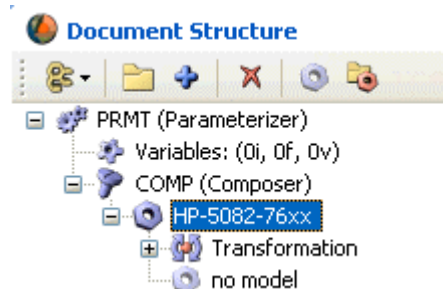
Un nouveau composant appelé Unnamed Component est créé. Pour voir le nouveau composant créé, cliquer sur le signe + situé à gauche de COMP.



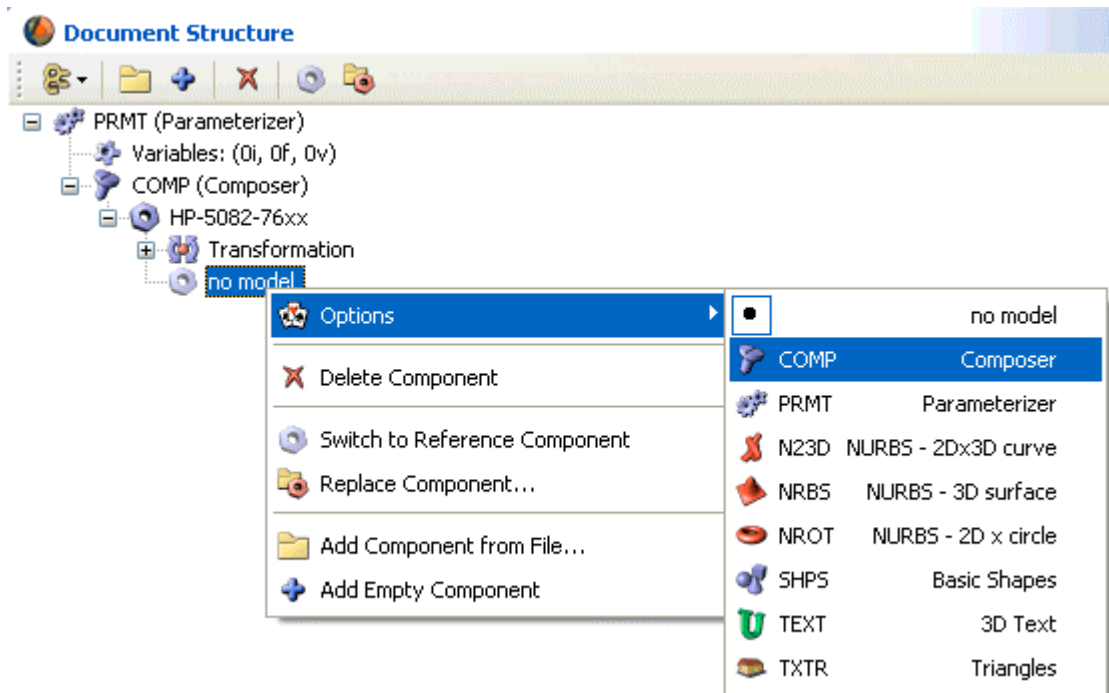
Nous allons renommer ce nouveau composant en HP-5082-76xx. Cliquer une fois sur le nom Unnamed Component, attendre une seconde et cliquer une seconde fois dessus (ou cliquer une fois et appuyer sur la touche de fonction F2), changer le nom actuel en HP-5082-76xx, puis appuyer sur la touche Entrée du clavier pour valider le nouveau nom.



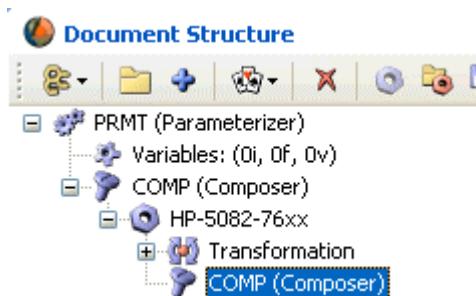
Cliquer sur le signe + situé à gauche du composant, et vous verrez apparaître deux éléments : un qui est appelé Transformation, dont nous verrons l'utilité plus tard, et un autre qui est appelé No model.



Cliquer avec le bouton droit sur No model afin d'afficher le menu contextuel, puis choisissez la commande Option, qui a pour effet de développer un nouveau sous-menu contextuel listant les composants de base que l'on peut utiliser à cet endroit. Dans cette dernière liste, sélectionner COMP.

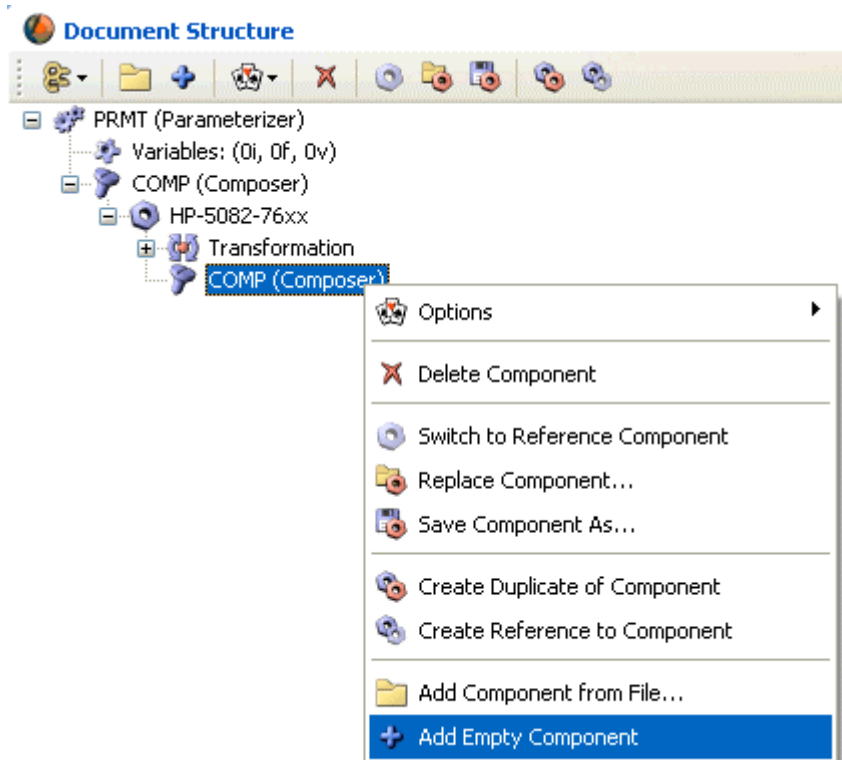


Ce qui nous donne ceci :

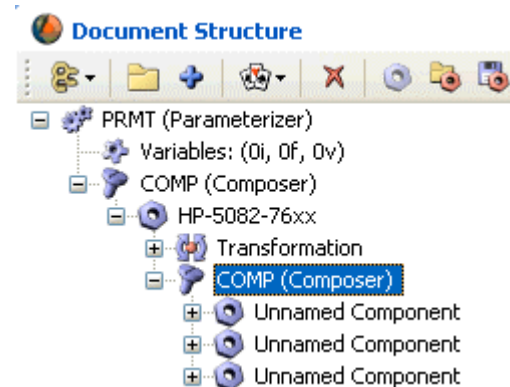
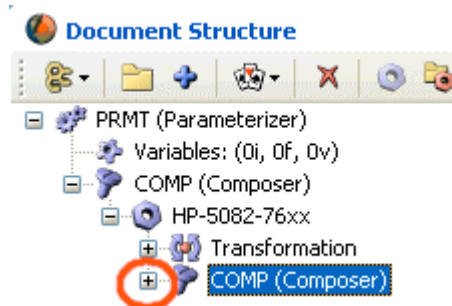


Nous avons donc maintenant un second container COMP inclus dans un premier container COMP. C'est dans ce deuxième container COMP que nous allons créer les trois containers (groupes) Body, Segments et Wires.

Ajoutons donc maintenant nos trois nouveaux containers. Pour cela, cliquer droit sur COMP, puis sélectionner la commande Add Empty Component. Répéter deux fois cette opération (clic droit sur COMP et Add Empty Component).

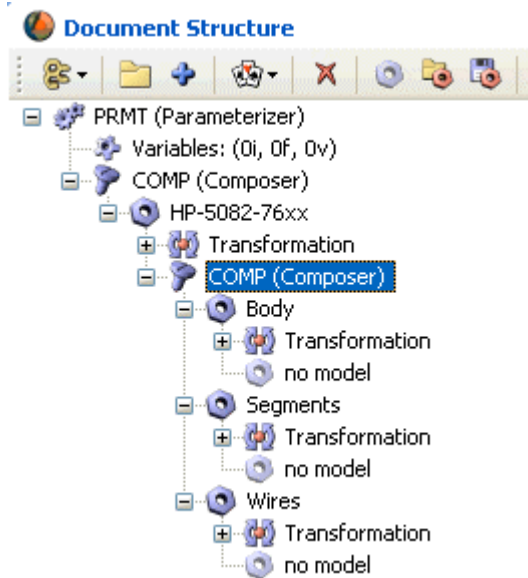


Un signe + est apparu à gauche de COMP, indiquant que le container n'est plus vide. Cliquez sur le signe + pour développer le contenu du container.



Par la même occasion, cliquer sur le signe + situé à gauche des trois nouveaux éléments appelés Unnamed Component, afin de développer leur contenu respectif. Vous verrez que là encore, nous avons pour chaque élément, deux entités appelées Transformation et No model.

Puis renommons maintenant chaque élément nouvellement créé, avec les noms précédemment définis, à savoir Body, Segments et Wires.



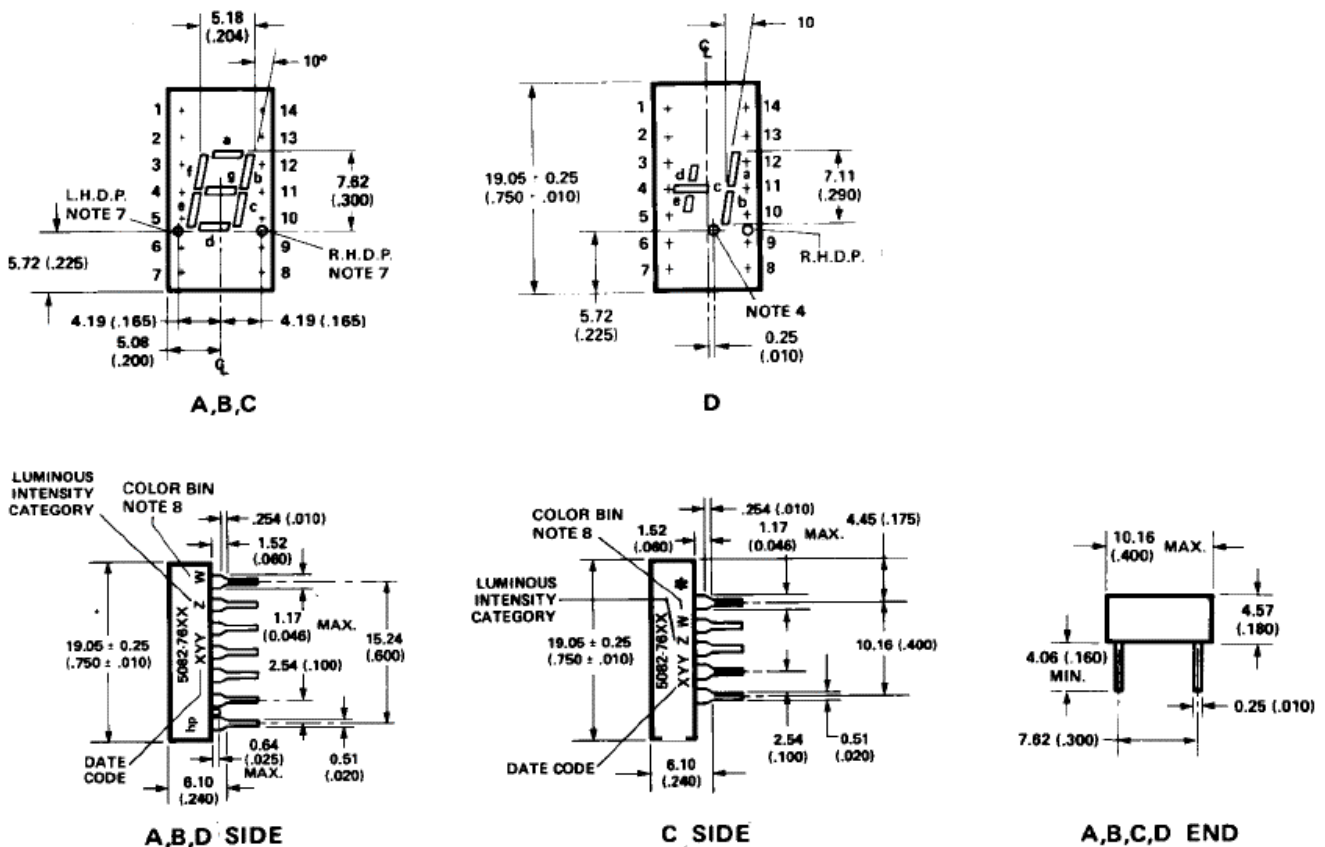
Voilà, la structure de base est créée, et nous allons pouvoir créer les sous-parties des trois groupes d'objets (containers) Body, Segments et Wires. Mais avant d'ajouter ces sous-parties, parlons un peu des dimensions du modèle à créer.

**Rappel** : pensez à sauver votre travail régulièrement ! Faites le donc maintenant.

## Dimensions de l'afficheur sept segments

Les dimensions de l'afficheur et de ses diverses parties, sont celles spécifiées dans le datasheet constructeur de l'afficheur HP-5082-7610, dont un extrait est affiché ci-après. Les dimensions sont données en inches (pouces) et en cm.

### Package Dimensions



\*The Side View of package indicates Country of Origin.

Pour le tutoriel, j'ai choisi d'adopter l'unité anglaise (inch), mais il aurait été aussi possible d'adopter l'unité métrique, puisque Ares permet de spécifier l'échelle et l'unité d'affichage. J'utiliserai donc l'unité "in" dans les lignes qui suivent.

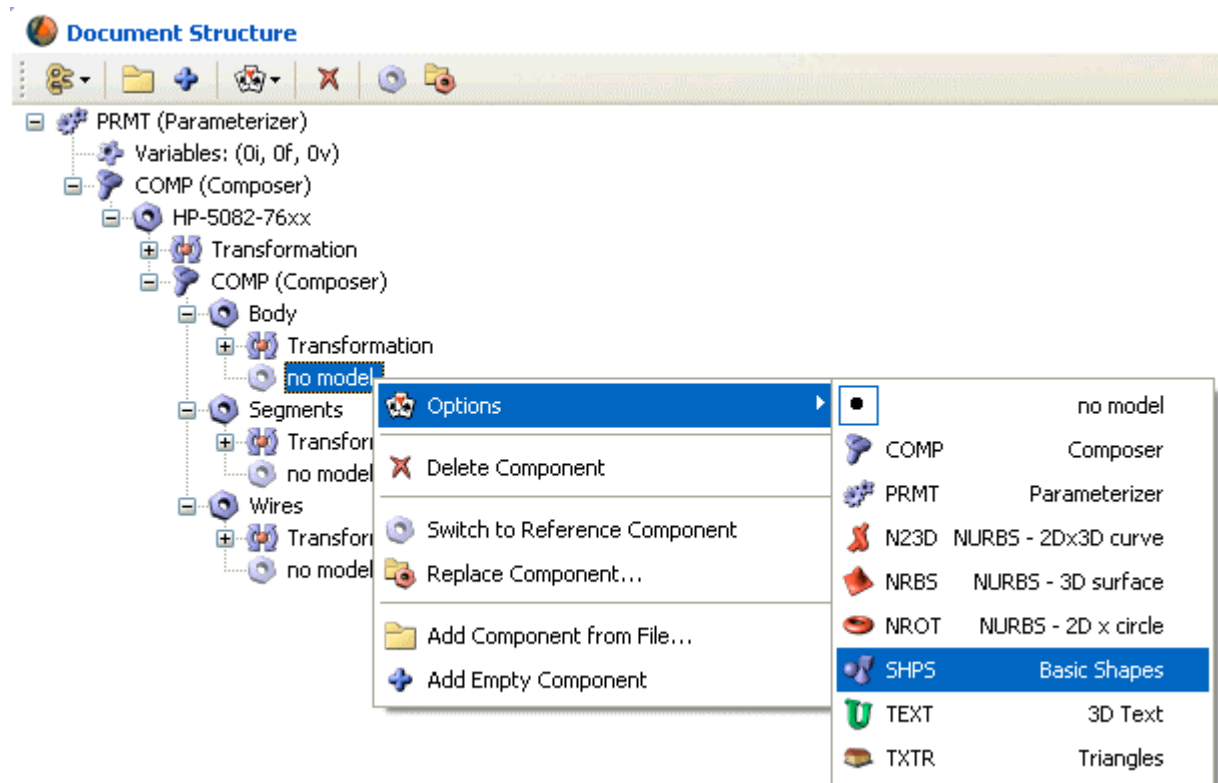
Le datasheet du 5082-7610 indique que la largeur du boîtier principal est de 0.400 in, que sa hauteur est de 0.750 in, et que son épaisseur est de 0.240 in. Nous allons donc commencer par ajouter un composant simple de type BOX qui aura ces dimensions, et qui constituera le corps de l'afficheur.

## Création du corps (container Body)

Le corps est constitué ici d'un seul élément. Vous pourrez par la suite ajouter d'autres parties pour rendre le modèle plus réaliste, si vous le souhaitez. Mais il ne faut pas oublier que plus il y a d'éléments, et plus l'affichage du composant prend du temps.

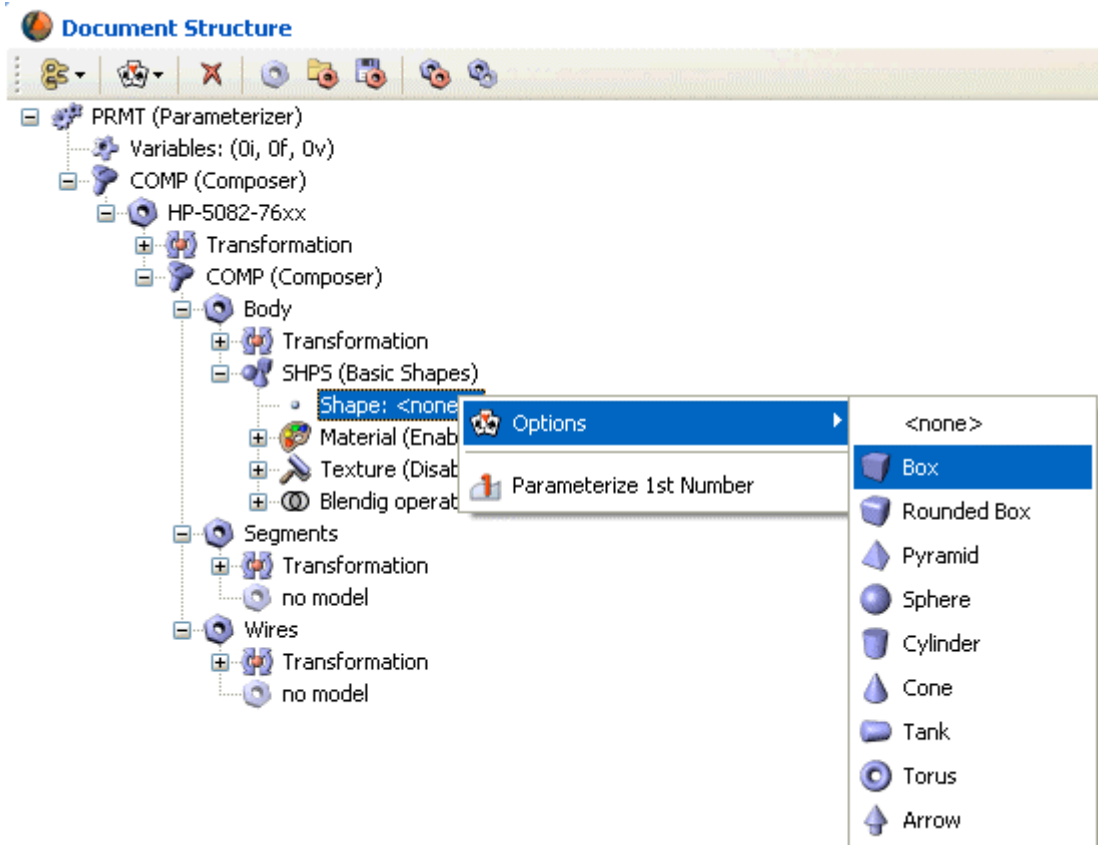
## Création du support principal

Le corps de l'afficheur peut être représenté par un simple rectangle et l'emploi d'un objet 3D simple de type "boite" convient parfaitement. Ajoutons donc un objet BOX dans notre container Body, en cliquant droit sur l'objet No model contenu dans le groupe Body, et en sélectionnant la commande Options, puis la commande SHPS (Basic Shapes).

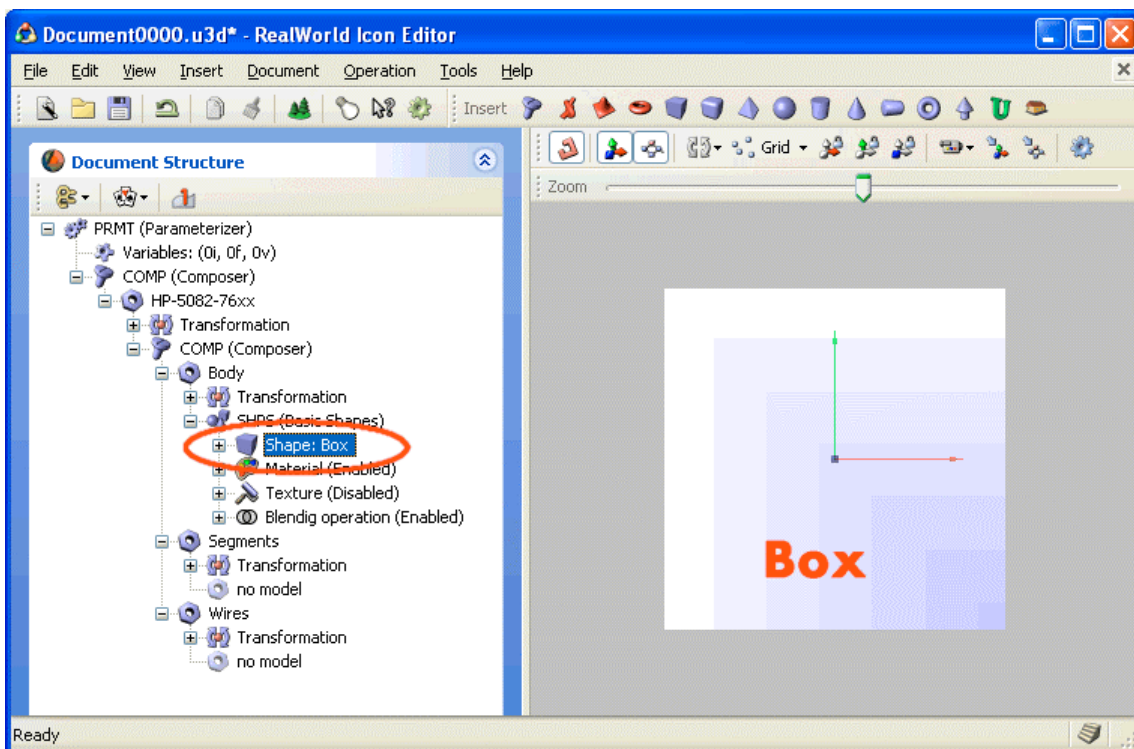


Le composant SHPS n'est pas un objet à proprement parler, mais plutôt un type d'objet. Il englobe l'ensemble des objets 3D basiques tels que cube, sphère ou cylindre. Une fois ce type d'objet sélectionné, il convient de spécifier quelle forme 3D de base on souhaite. Pour cela, développer le composant SHPS en cliquant sur le signe + situé sur sa gauche, puis cliquer droit sur Shape: <None>. Dans le menu contextuel qui apparaît, cliquer sur Options, puis sur Box.



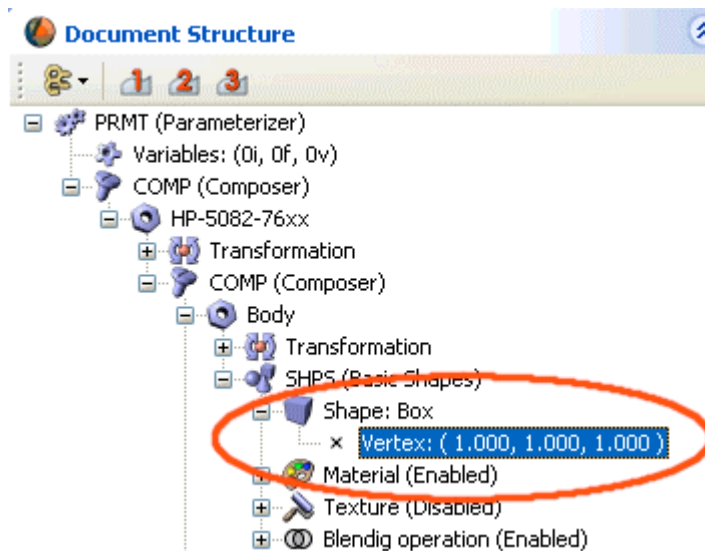


Et voilà, notre premier objet 3D apparaît dans la vue 3D, située à droite de la fenêtre de RWIE :

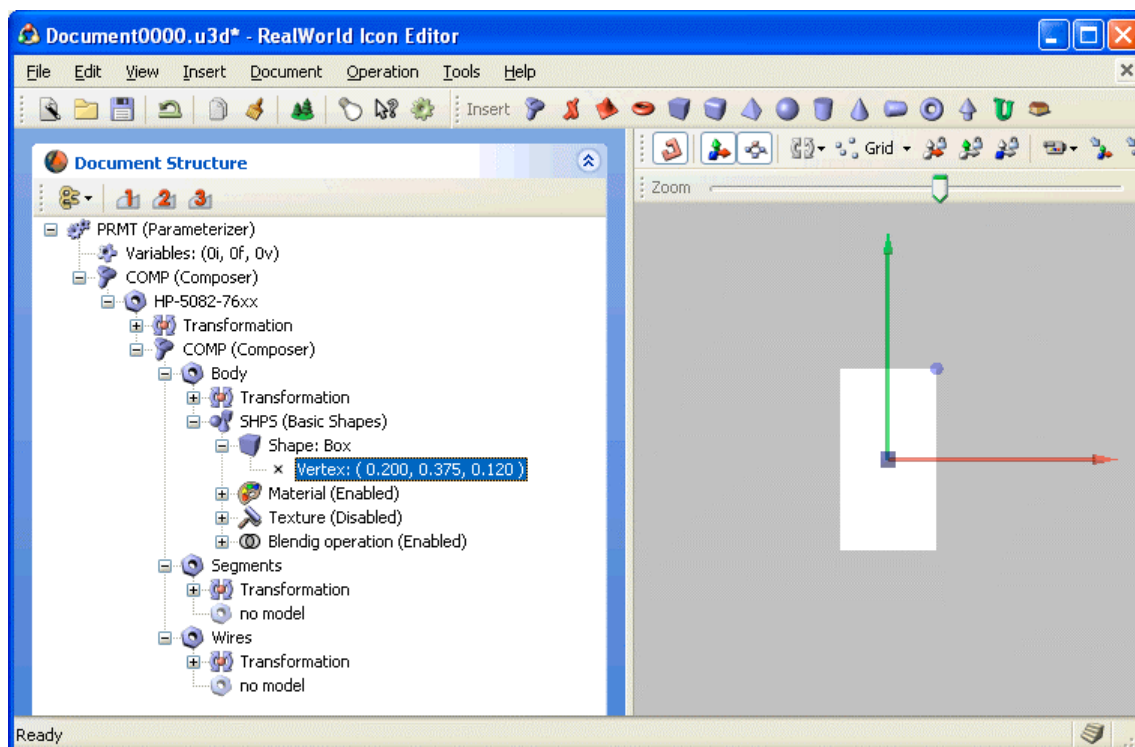


Nous allons maintenant "tailler" cet objet BOX aux dimensions requises, en tenant compte du fait que les positions des points qui permettent de dessiner le rectangle sont définies par rapport au point central de référence.

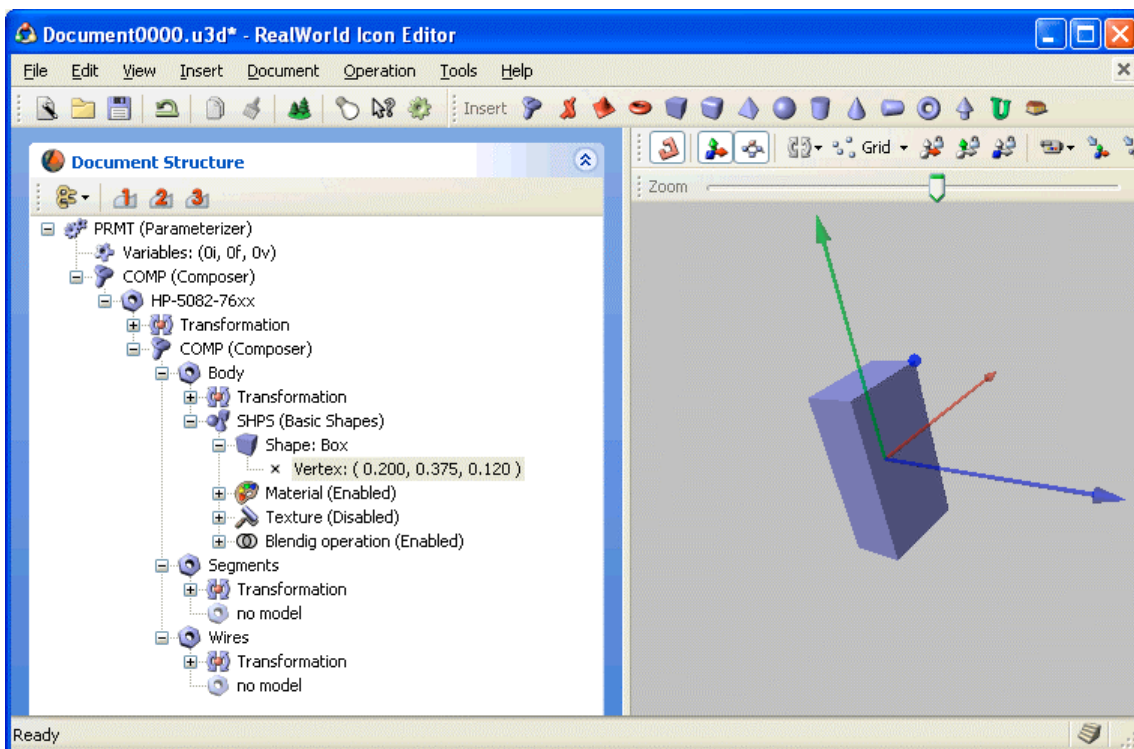
Cliquer maintenant sur le signe + qui précède Shape: Box. Vous pouvez voir que cet objet possède les coordonnées (Vertex) [1.000, 1.000, 1.000], il s'agit d'un cube dont chaque côté mesure 2.000 (+1.000 et -1.000 par rapport au centre 0.000).



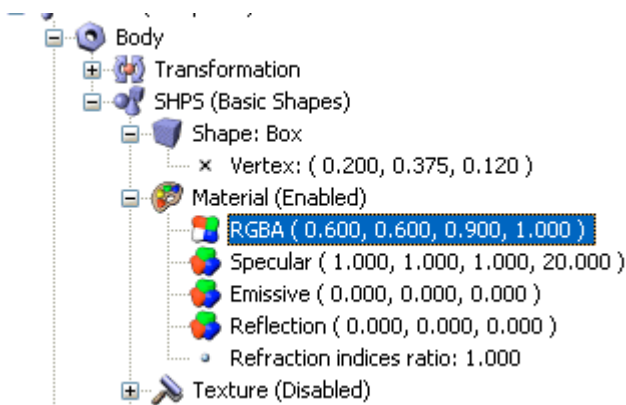
Si nous donnons directement à l'objet BOX les dimensions précitées dans le Datasheet, nous allons nous retrouver avec un composant dont les dimensions sont le double des normales. Il faut donc diviser par deux les dimensions données dans le datasheet. Nous donnons donc à l'objet BOX les valeurs de Vertex [0.200, 0.375, 0.120], et non pas les valeurs de vertex [0.400, 0.750, 0.240]. Vous devez avoir un objet ressemblant à la copie d'écran suivante.



**Remarque :** pour tourner autour de l'objet BOX, vous pouvez, dans la vue 3D, déplacer la souris tout en maintenant le bouton droit de la souris enfoncé.



La couleur n'étant pas celle désirée, nous allons la retoucher un peu. Pour cela, cliquer sur le signe + situé à gauche de Material, afin de visualiser les valeurs attribuées aux trois composantes couleur primaires Rouge, Vert et Bleu (RVB).



#### Les valeurs actuelles de RGBA sont

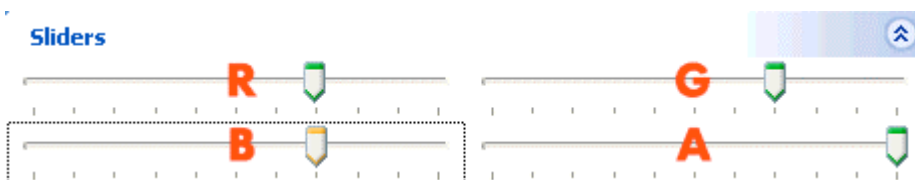
R = 0.600 (Red = rouge)  
G = 0.600 (Green = vert)  
B = 0.900 (Blue = bleu)  
A = 1.000

#### Changez-les en :

R = 0.700  
G = 0.700  
B = 0.700  
A = 1.000

**Pour cela**, cliquer une fois sur les valeurs en cours, attendez une seconde, puis cliquer à nouveau dessus (ou appuyer sur F2).

**Remarque** : les paramètres RVB peuvent être ajustés manuellement, mais il est plus facile de faire usage des sliders (potentiomètres rectilignes) pour modifier rapidement les valeurs du paramètre actuellement sélectionné.



Et voilà, le corps principal de l'afficheur est terminé !

## Création des segments (container Segments)

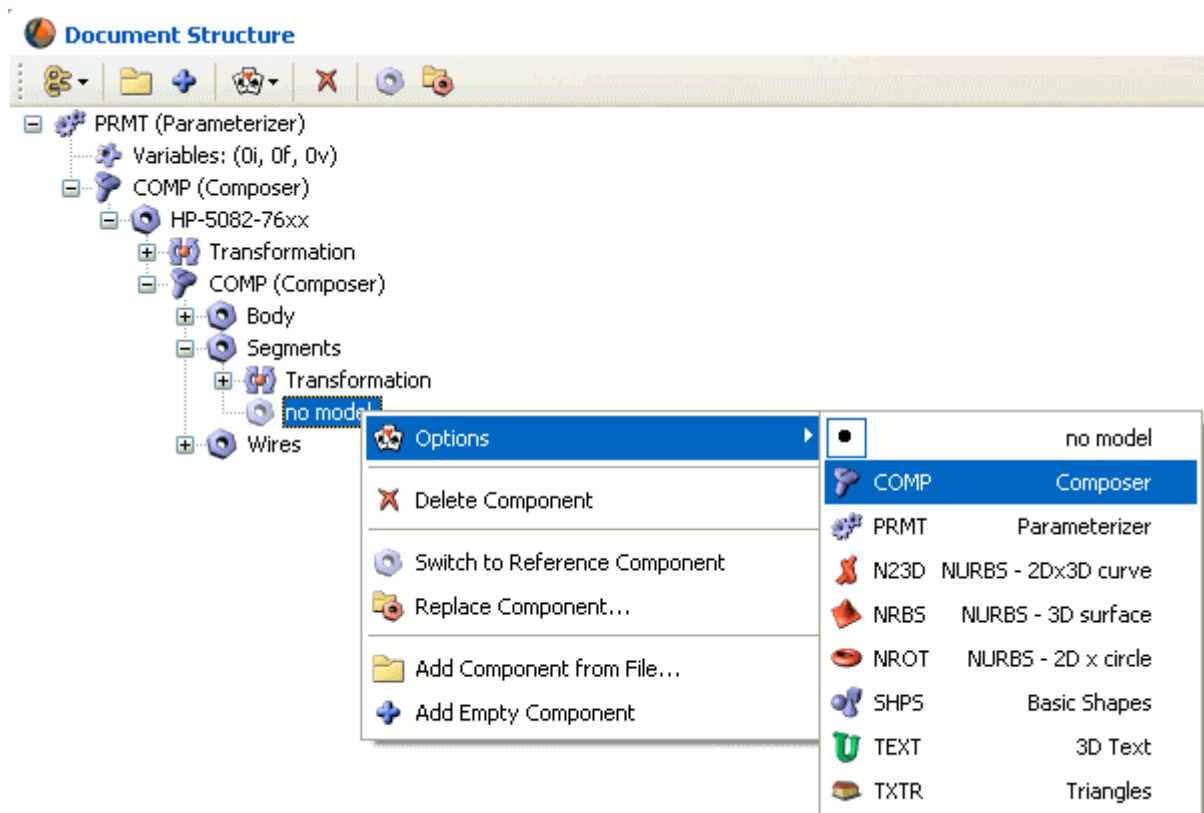
Les segments sont tous identiques, si ce n'est que trois sont horizontaux (segments A, D et G) et quatre sont verticaux (segments B, C, E et F). Pour ne pas répéter cinquante fois la même chose, nous allons créer un seul segment horizontal et un seul segment vertical, que nous dupliquerons ensuite en autant d'exemplaires que nous en avons besoin. Il ne restera ensuite qu'à positionner correctement chaque segment dans l'espace 3D.

### Création d'un segment horizontal

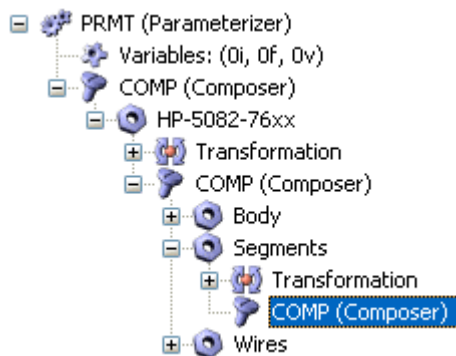
Une fois n'est pas coutume, nous allons créer le segment G en premier. Tout simplement parce qu'il s'agit du segment central, mais il n'y a aucune obligation de commencer par lui.

Si le segment avait été un rectangle simple, nous aurions pu nous contenter d'un objet 3D simple comme l'objet BOX. Mais nous avons affaire ici à une forme qui n'est pas rectangulaire, du fait que les segments sont légèrement inclinés. Nous allons donc chercher un autre composant plus adapté à la forme que nous voulons donner au segment. Dans le cas présent, l'objet 3D NRBS est celui qui semble le plus adapté, car il permet de créer des surfaces polygonales avec des angles quelconques, on n'est pas limité aux angles droits.

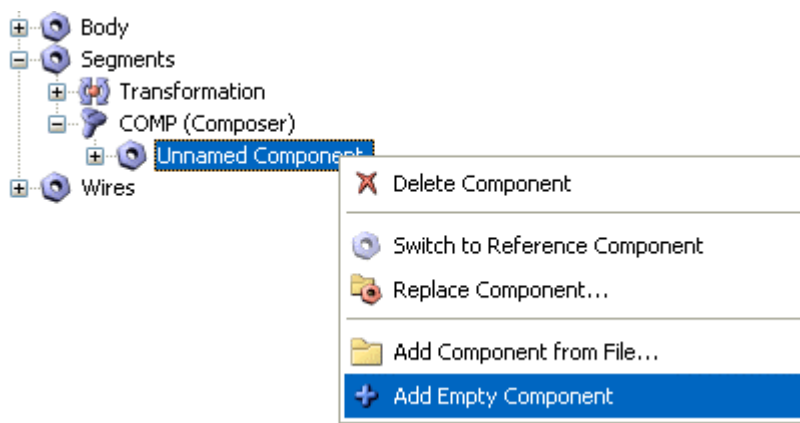
Comme cela l'a déjà été fait précédemment, nous allons créer un container pour stocker nos sept segments. Cela n'est nullement obligatoire, mais permet de les manipuler plus facilement en bloc. Dans le groupe Segments, cliquer droit sur No model, puis sélectionner successivement les commandes Options et COMP, pour créer un nouveau container.



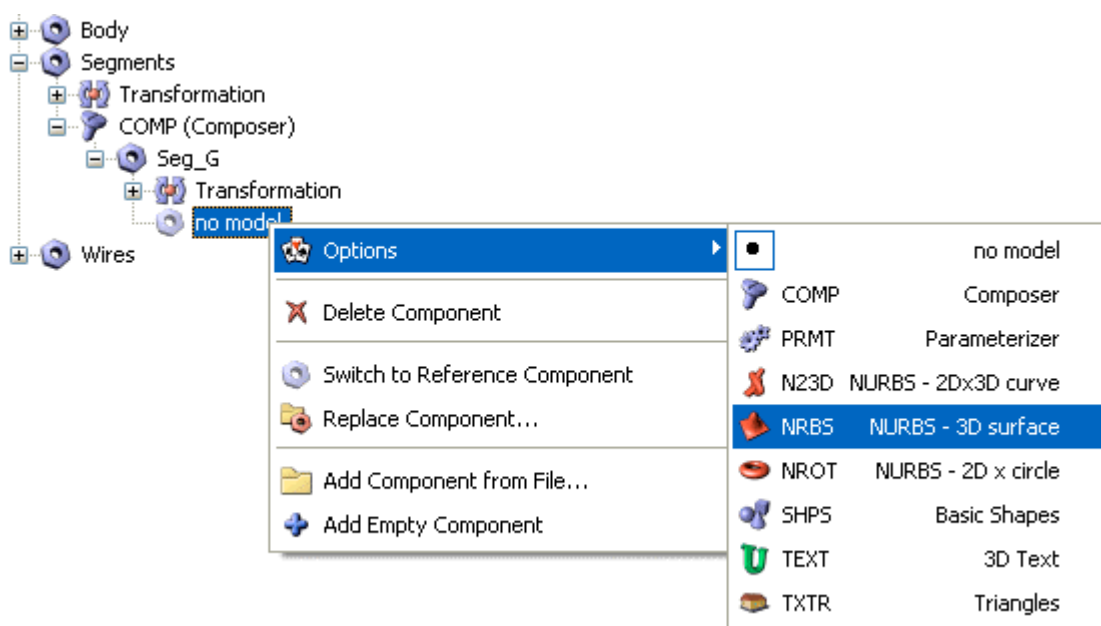
Vous devriez avoir cela :



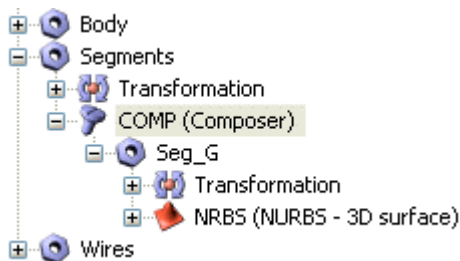
Ajoutons maintenant un composant vide dans ce nouveau container, en cliquant droit sur l'élément Unnamed Component, et en utilisant la commande Add Empty Component, qui devrait commencer à devenir familière.



Renommons l'élément appelé Unnamed Component, en Seg\_G, puis développons sa branche afin de voir ce que cet élément contient (si vous ne savez pas comment opérer ces deux actions, revenez quelques lignes en arrière). Enfin, attribuons un modèle d'objet 3D de type Surface rectangulaire (NRBS) à ce composant vide, grâce aux commandes Options et NRBS, qui sont accessibles après un clic droit sur l'élément No model.



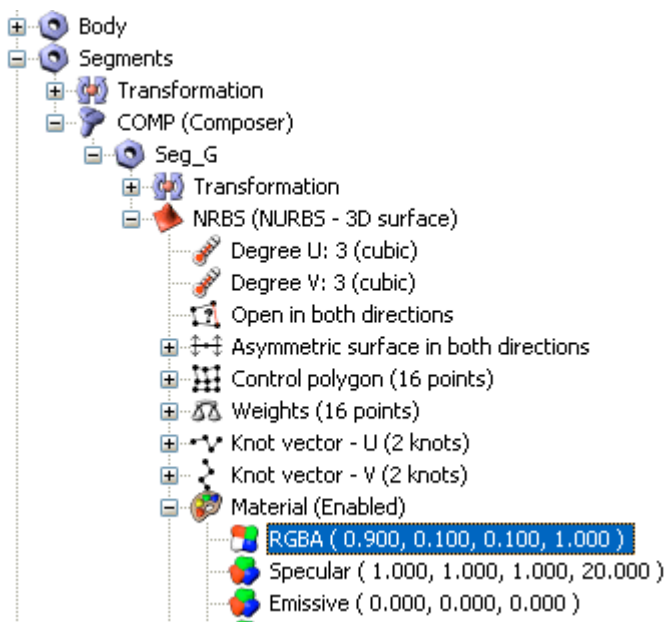
L'élément Seg\_G contient désormais un objet 3D de type NRBS.



Cet objet possède une certaine quantité de paramètres permettant de modifier son aspect visuel, il suffit de cliquer sur le signe + situé sur sa gauche pour les rendre partiellement visibles et s'en convaincre.

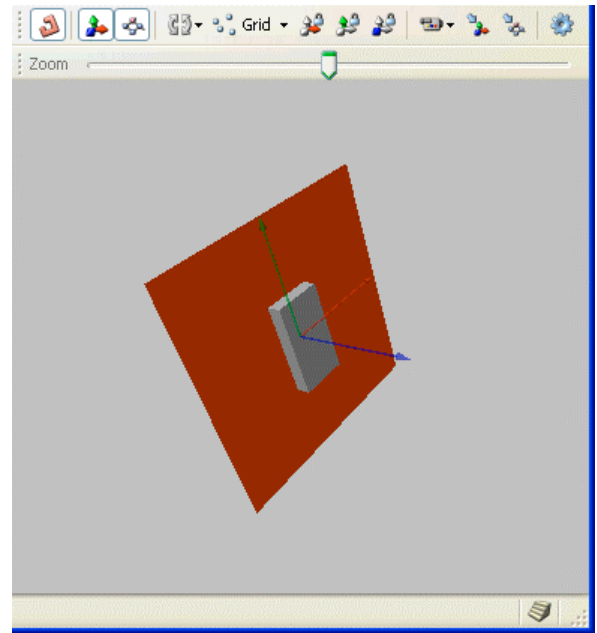
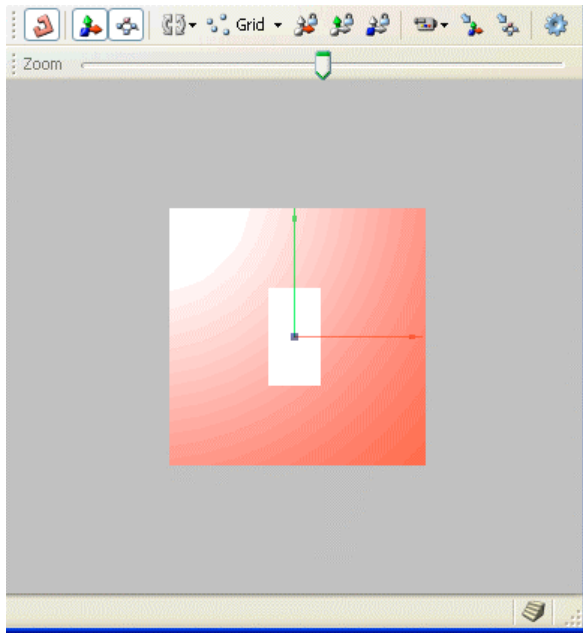
Bien, nous avons désormais deux objets visibles dans la vue 3D de RWIE. Ce que nous voyons n'est pas forcément très parlant, surtout si la vue en 3D se fait de face. Pour y voir un peu plus clair, et comme l'ordre de certaines opérations n'a pas d'importance, nous allons dès maintenant donner la couleur rouge à notre segment, avant même de lui donner la bonne dimension. Cette façon de faire permet d'avoir deux objets différemment colorés dans la vue 3D, ce qui permet de les distinguer plus facilement.

Cliquer droit sur le signe + situé à gauche du paramètre Material, de telle sorte que l'on puisse visualiser et modifier le paramètre RGBA. Comme vu auparavant, le premier paramètre est relatif à la quantité de couleur rouge, alors que les deux valeurs suivantes s'appliquent aux quantités de couleurs vert et bleu. Nous voulons un segment de couleur à dominante rouge, aussi attribuons-nous les valeurs ( 0.900, 0.100, 0.100, 1.000 ) aux paramètres RGBA (nous laisserons toujours le paramètre A à sa valeur de 1.000).



Les nouvelles valeurs affectées aux paramètres RGBA conduisent à une vue 3D qui doit ressembler aux copies d'écran suivantes. Notez au passage que notre nouvel objet Seg\_G coupe en deux l'objet Body. Cela est visible si vous faites tourner l'objet dans la vue 3D (en bougeant la souris tout en maintenant enfoncé le bouton droit de la souris).



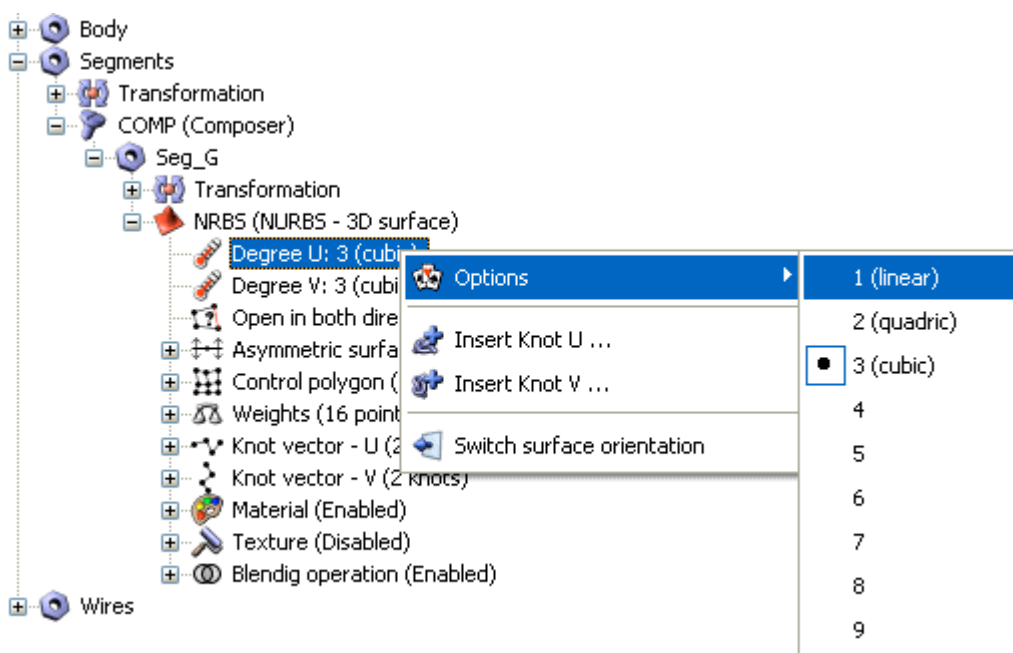


Tout cela est bien joli, mais le segment possède des dimensions qui ne sont pas encore tout à fait les bonnes, ce à quoi nous allons tout de suite remédier.

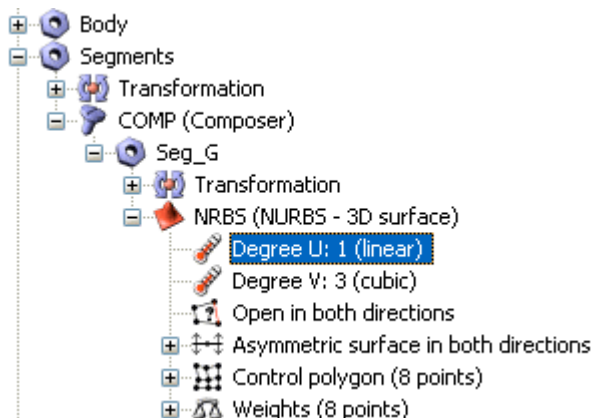
Si vous voulez prendre un café avant de continuer, vous le pouvez.

Avant de redimensionner notre objet, une petite remarque s'impose concernant l'objet NRBS. L'objet NRBS comporte un certain nombre de points de contrôle, et ce nombre de points de contrôle peut être ajusté aussi bien dans le sens vertical que dans le sens horizontal. Par défaut, cet objet comporte 16 points de contrôle, or 4 sont suffisants pour réaliser notre segment lumineux. Nous allons donc réduire ce nombre de points, d'une part pour faciliter la modification de l'objet, et d'autre part pour réduire son "poids". Pour cela, nous allons diminuer le nombre de points de contrôle dans les deux orientations "Degree U" et "Degree V".

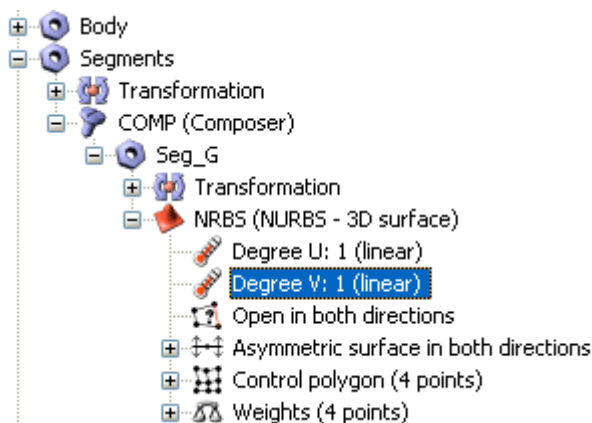
Cliquer droit sur l'élément Degree U, et dans le menu contextuel qui apparaît, sélectionner la commande Options, puis la commande 1 (linear), pour changer la valeur du paramètre sélectionné de 3 (cubic) en 1 (linear).



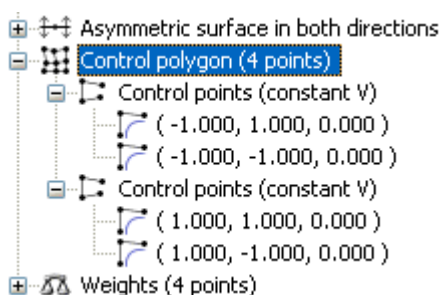
Cela occasionne les changements suivants :



Vous remarquerez que le nombre de points (paramètre Control polygon) est passé de 16 à 8. Procédons de la même façon que précédemment, mais cette fois pour le paramètre Degree V, auquel nous attribuons aussi la valeur 1 (linear) au lieu de 3 (cubic). Ce qui nous donne la vue suivante, mettant en évidence une nouvelle réduction du nombre de points de contrôle, qui passe ainsi de 8 à 4.

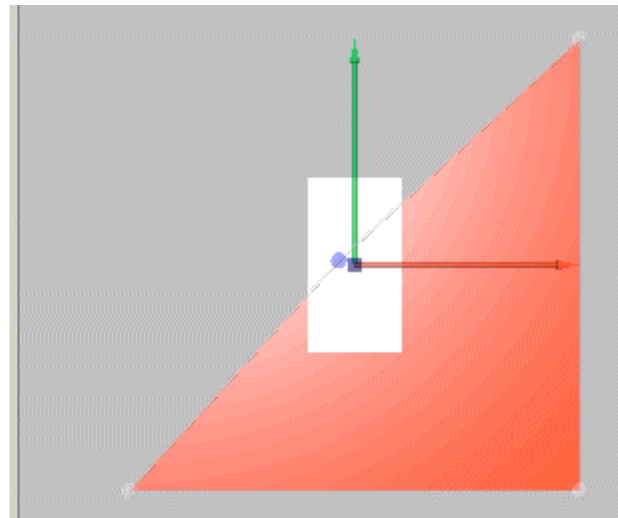
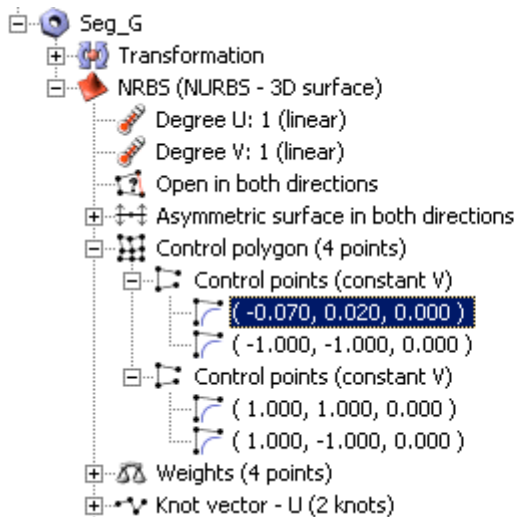


Développons entièrement la branche du paramètre Control polygon en cliquant sur les trois signes + des éléments Control polygon et Control points, afin de voir le détail des points de contrôle du polygone.

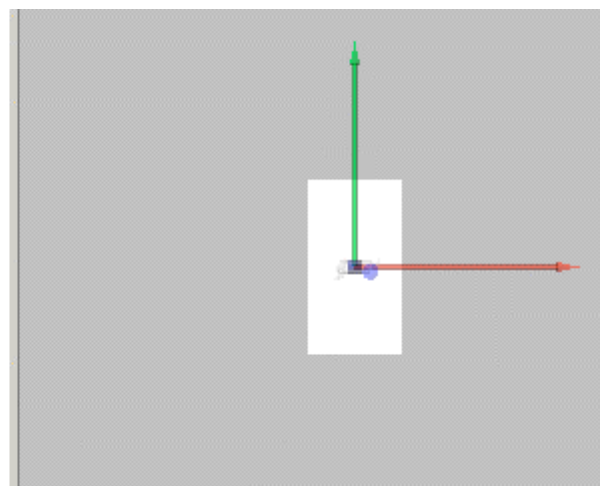
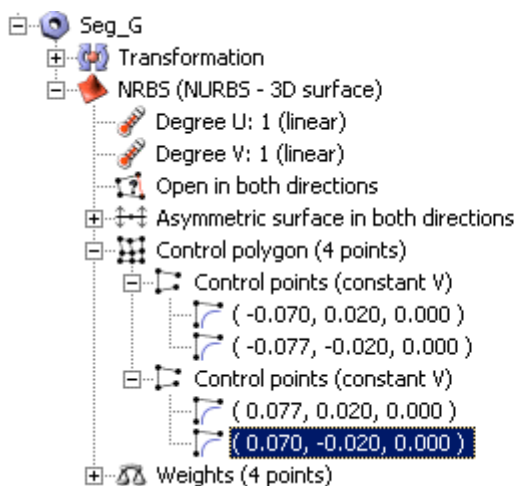


Ce que nous voyons là ressemble étrangement à ce que nous avons vu précédemment avec les paramètres Vertex ou RGBA : une suite de valeurs attribuées à plusieurs paramètres, les paramètres étant ici des points de contrôle du polygone. De façon intuitive, on pourrait presque penser que ces paramètres permettent de modifier la position des points de contrôle qui déterminent la forme de l'objet 3D. Faisons donc confiance à notre intuition, et modifions les valeurs du premier point de contrôle, en adoptant les nombres visibles sur la copie d'écran suivante, afin de voir ce qui se passe.

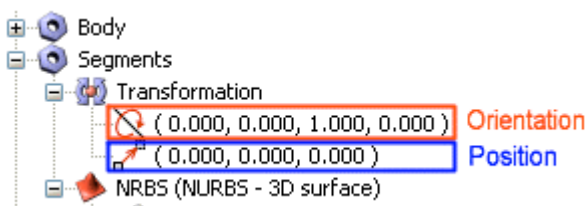




Quelque chose a effectivement changé dans la forme de l'objet 3D, qui n'est désormais plus un simple carré. Le point modifié se trouve désormais proche du centre de l'objet Body, et est masqué par ce dernier, même si l'on voit encore un petit cercle bleu pour marquer son emplacement. Imaginons un instant que cet aspect des choses ne nous gêne en rien, et continuons comme si de rien n'était, en attribuant de nouvelles valeurs de position aux trois autres points du polygone, telles qu'indiquées ci-après.

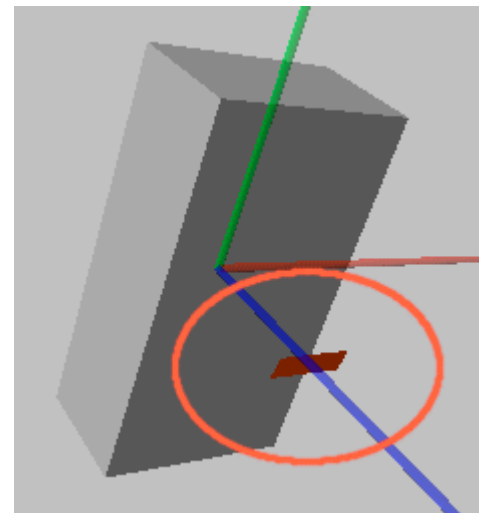
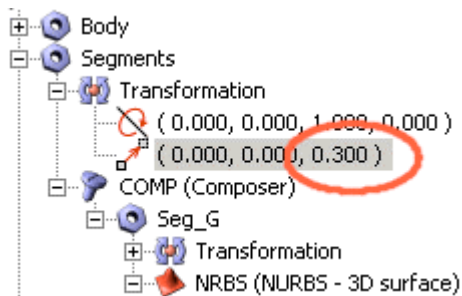


Notre objet Seg\_G a sembler-t-il totalement disparu de la vue 3D. Ce qui est normal, car les quatre points du polygone sont désormais tous dans une zone spatiale située au coeur du composant Body, qui masque presque tout. Comme vous devez vous en douter un peu, rendre à nouveau visible ce segment doit bien être possible, d'une façon ou d'une autre. Et vous avez raison, on ne va pas s'embêter à travailler pour rien. Comme le polygone est contenu dans un container appelé Segments doté de fonctions de positionnement dans l'espace 3D, nous allons en profiter. Cliquer sur le signe + situé à gauche de Transformation, directement en dessous de Segments, afin de rendre visible les paramètres d'orientation et de positionnement.

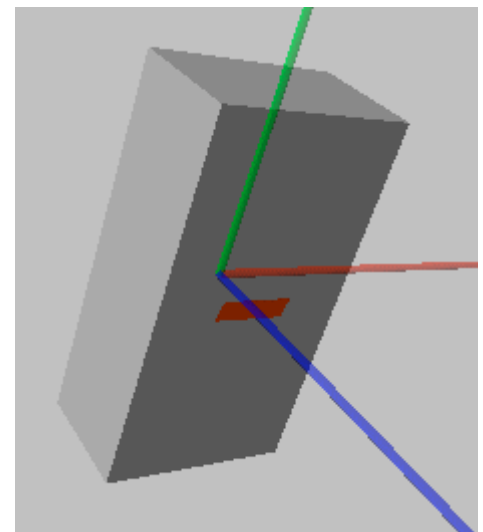
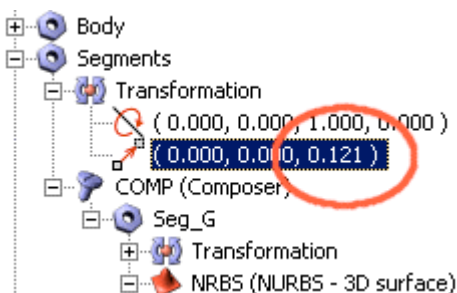


Cliquer une fois sur la ligne des paramètres de positionnement (en bleu sur la copie d'écran précédente), et modifier la position des Sliders afin de voir ce qui se passe. Vous constatez que le segment nouvellement redimensionné peut se déplacer dans l'espace 3D selon les axes X, Y et Z, comme bon vous semble.

Donner maintenant au troisième paramètre de positionnement, les valeurs 0.300 dans un premier temps, puis 0.121 dans un second temps, et observez ce qui se passe.



Avec la valeur 0.300, le segment a été ramené plus en avant (axe Z), et on le voit bien mieux. Mais il est beaucoup trop avancé et il ne touche pas le corps du boîtier (Body).



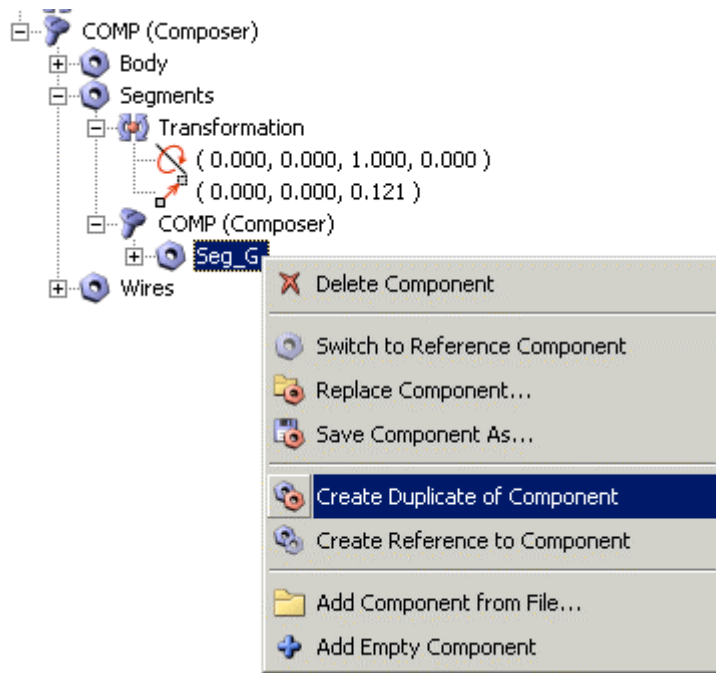
Avec la valeur 0.121 (qui correspond à l'épaisseur du boîtier, auquel a été ajouté 1 millième d'unité), le segment paraît "collé" au boîtier, ce qui procure l'effet recherché.

Nous voilà désormais avec une vue qui devrait vous rappeler quelque chose de bien précis, n'est-ce pas ? Un segment seul permettant un nombre d'affichages relativement restreint, nous allons le dupliquer pour en avoir trois identiques. Un autre segment au-dessus de l'actuel, et un autre segment en dessous.

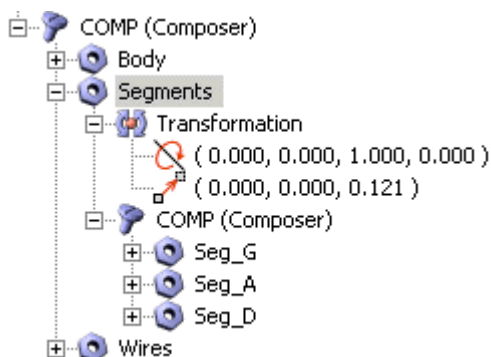
## ***Duplication du segment horizontal***

Le segment existant nous plaît, et comme nous sommes toujours un peu allergiques au travail inutile, il est normal de vouloir dupliquer ce segment plutôt que d'en recréer de nouveaux identiques. Après tout, le traitement de texte nous a un peu habitué à ce genre d'exercice avec ses fonctions de "copier / coller".

Cliquer droit sur l'élément Seg\_G, et dans le menu contextuel qui apparaît, sélectionner la commande Create Duplicate of Component.



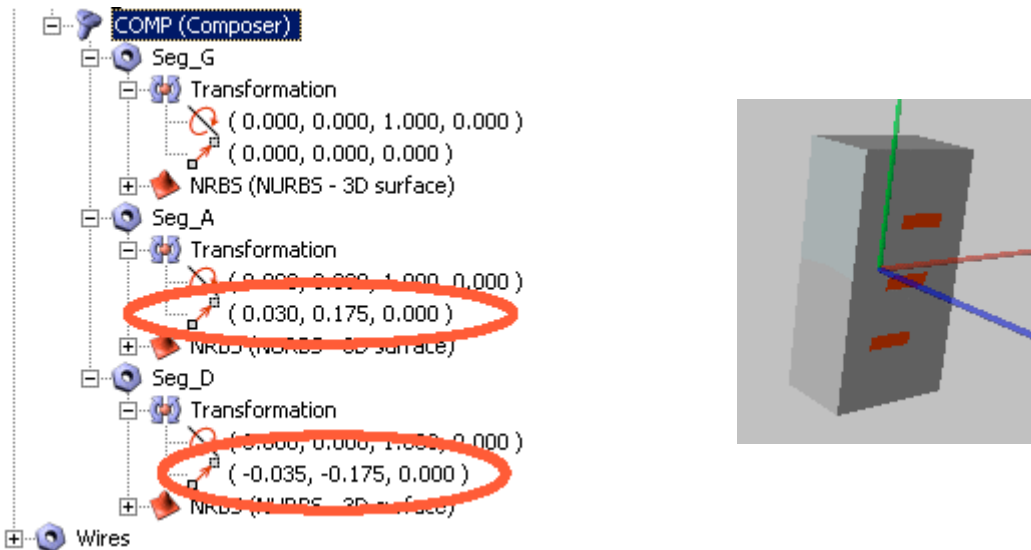
Un nouvel objet, identique au segment Seg\_G, est créé, qui porte le nom Copy of Seg\_G. Renommer le composant Copy of Seg\_G en Seg\_A. Une nouvelle fois, placer le curseur de la souris au-dessus de Seg\_G, puis cliquer droit. Dans le menu contextuel qui apparaît, sélectionner une nouvelle fois la commande Create Duplicate of Component. Une deuxième copie portant le nom Copy of Seg\_G, est créée. Cette fois, renommer ce nouveau composant Copy of Seg\_G en Seg\_D. Voici ce que vous devriez désormais avoir.



## ***Positionnement des segments horizontaux***

Positionnons maintenant les deux segments Seg\_A et Seg\_D respectivement au-dessus et en dessous du segment G. Pour respecter l'inclinaison du composant original (réel), nous allons décaler légèrement le segment Seg\_A sur la droite, et le segment Seg\_D légèrement sur la gauche, par rapport au segment Seg\_G qui lui est centré.

Les valeurs de positionnement à adopter pour ces deux nouveaux segments sont indiquées ci-après, la vue 3D en regard reflétant l'utilisation de ces valeurs.



Je ne sais pas si vous pensez comme moi, mais à ce stade, j'ai l'impression que le composant prend vraiment forme, et je n'ai pas envie de m'arrêter.

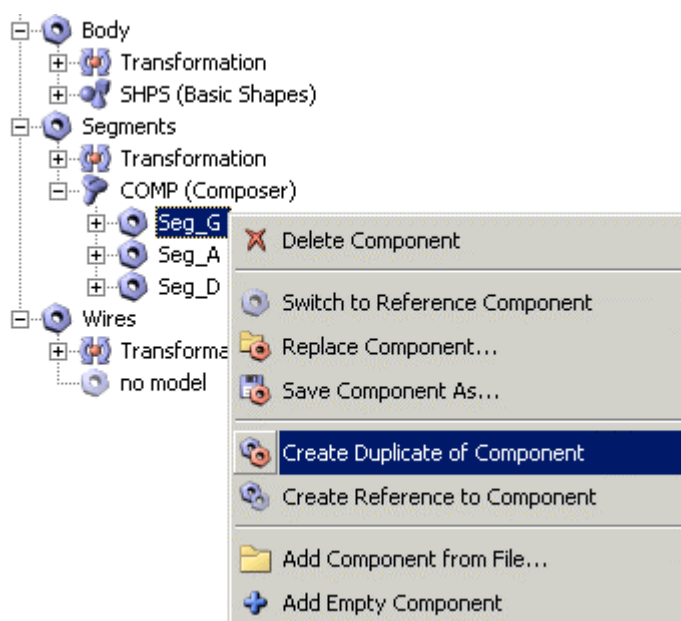
Si ce n'est pour prendre un café.

Nous allons maintenant créer un segment vertical, en procédant de la même façon que pour la création du premier segment horizontal. Puis nous dupliquerons ce segment vertical pour obtenir quatre segments de même type et même dimensions, que nous disposerons comme il le faut dans l'espace 3D.

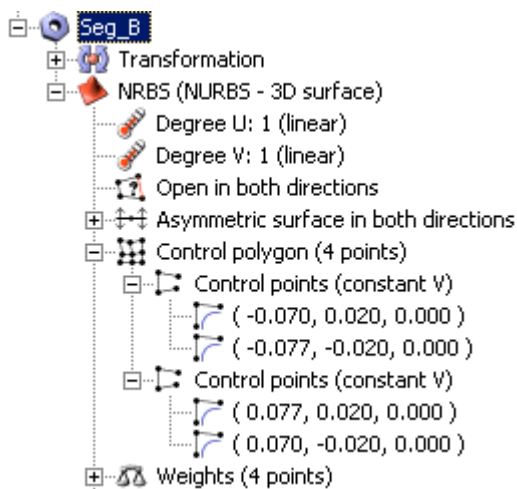
## Création d'un segment vertical

Comme les opérations à effectuer sont identiques aux précédentes, elles seront moins détaillées. Au besoin, revenir un peu en arrière si vous bloquez sur un point.

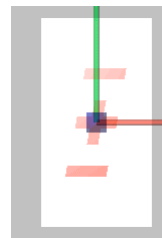
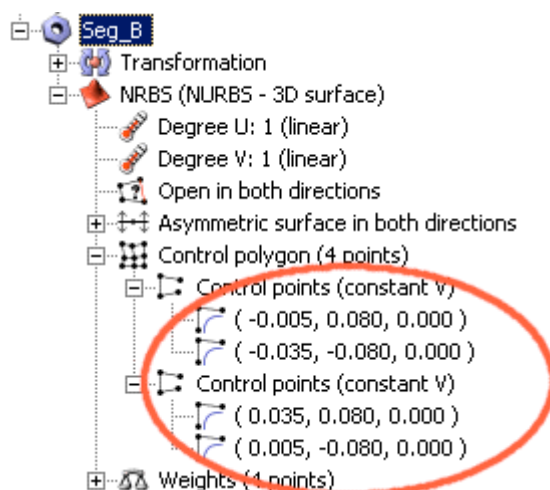
Dupliquer le composant Seg\_G en un nouvel exemplaire, en cliquant droit sur son nom et en sélectionnant la commande Create Duplicate of Component.



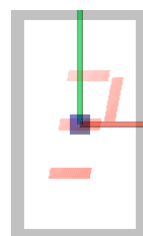
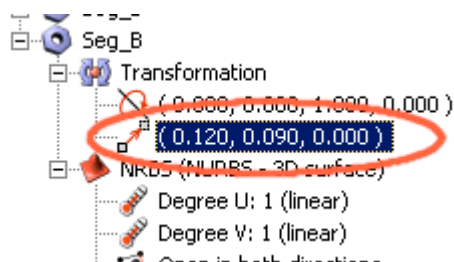
Renommer le composant Copy of Seg\_A ainsi créé en Seg\_B, puis développer la branche Seg\_B pour rendre visibles les paramètres des points de contrôle du polygone (cliquer sur les signes + qui vont bien). Vous devriez avoir la vue suivante.



La valeur des points de contrôle de ce segment Seg\_B sont identiques à celles des points de contrôle du segment Seg\_A, ce qui est normal puisque ce segment est obtenu par copie. Nous allons modifier ces valeurs pour donner au segment Seg\_B, une allure verticale. Pour cela, adopter les valeurs visibles sur la copie d'écran ci-après de gauche, la vue 3D devant vous montrer un objet ressemblant à la copie d'écran ci-après à droite.



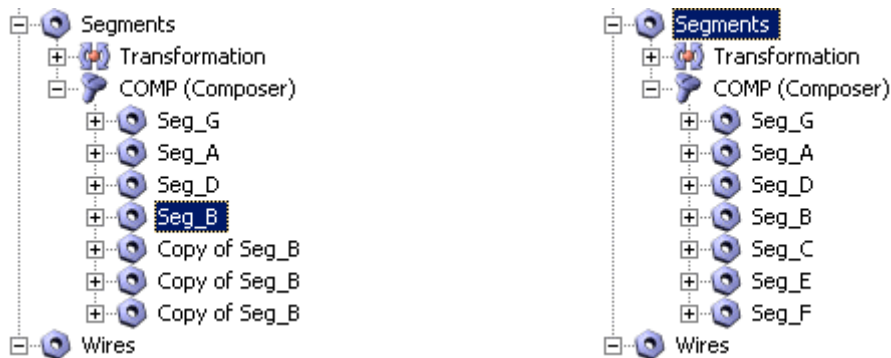
Notez qu'une fois encore, le nouveau composant est d'office positionné au centre, qui correspond aux coordonnées [0.000, 0.000, 0.000]. Déplaçons-le un peu à droite et au-dessus de sa position actuelle, en utilisant les coordonnées suivantes.



Voilà un segment vertical créé et correctement positionné. Profitons donc de sa bonne forme pour en créer trois nouveaux identiques.

## Duplication du segment vertical

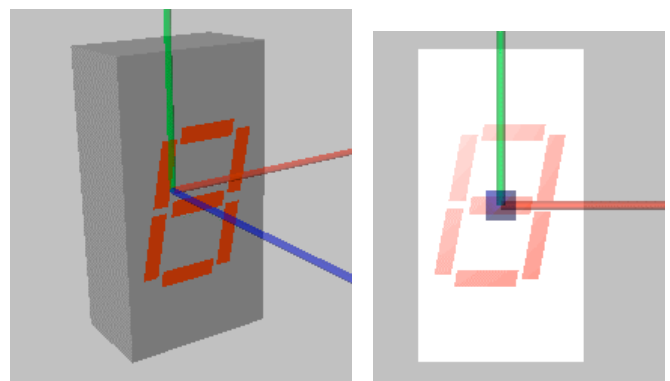
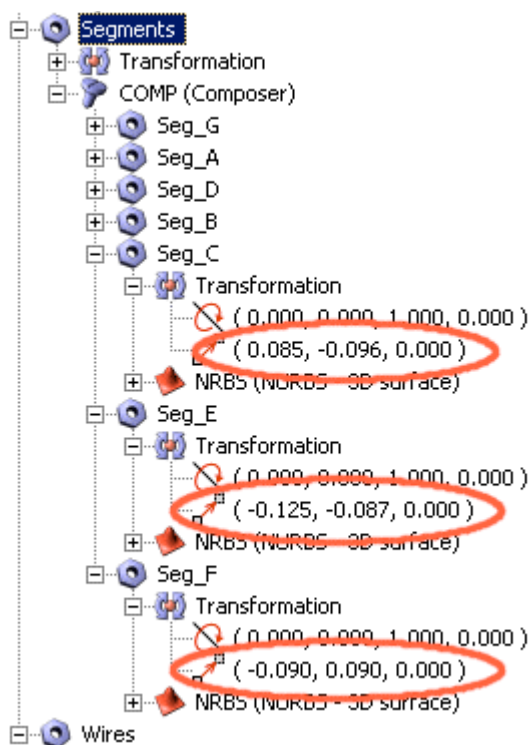
Comme vous commencez à connaître le mécanisme du logiciel RWIE, vous ne devriez pas être surpris si je vous demande de placer le curseur de la souris au-dessus de Seg\_B, et d'utiliser trois fois de suite la commande Create Duplicate of Component, et de renommer ensuite les trois nouveaux composants pour qu'ils répondent tout naturellement aux doux noms de Seg\_C, Seg\_E et Seg\_F.



A ce stade, vous devriez voir les trois nouveaux segments superposés les uns aux autres, en plein centre du corps (Body), formant un signe + avec le segment horizontal Seg\_G. A nous (et oui, encore) de faire le nécessaire pour positionner correctement ces nouveaux segments.

## Positionnement des segments verticaux

Comme il m'arrive à moi aussi de fatiguer un peu, je vous donne les valeurs de position à affecter aux trois derniers segments créés, et vous laisse faire le reste, sachant que vous devez obtenir un résultat similaire à la copie d'écran ci-après à droite.



Ce à quoi doit ressembler votre travail si tous les paramètres spécifiés jusqu'ici ont été saisis sans erreur.

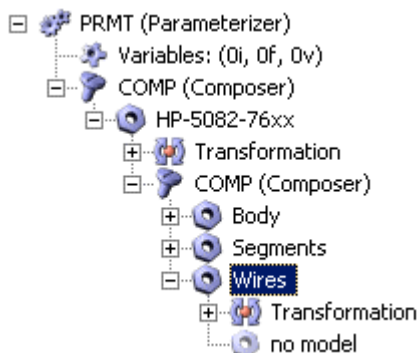
Pour rappel, le déplacement dans l'espace de la vue 3D s'effectue en bougeant la souris tout en laissant enfoncé son bouton droit.

Tout est conforme ? Et bien félicitations ! Vous avez droit à un autre café !

## Création des connexions électriques (container Wires)

Une grande partie de ce qui a été fait précédemment va nous servir pour la création des pattes de connexion, qui se trouvent être de formes et dimensions identiques : nous allons en créer une, puis la dupliquer et positionner chaque patte dupliquée, là où elle doit être. Nous utiliserons par la même occasion, des formules simples pour espacer de façon régulière (et surtout en une seule fois) la totalité des pattes.

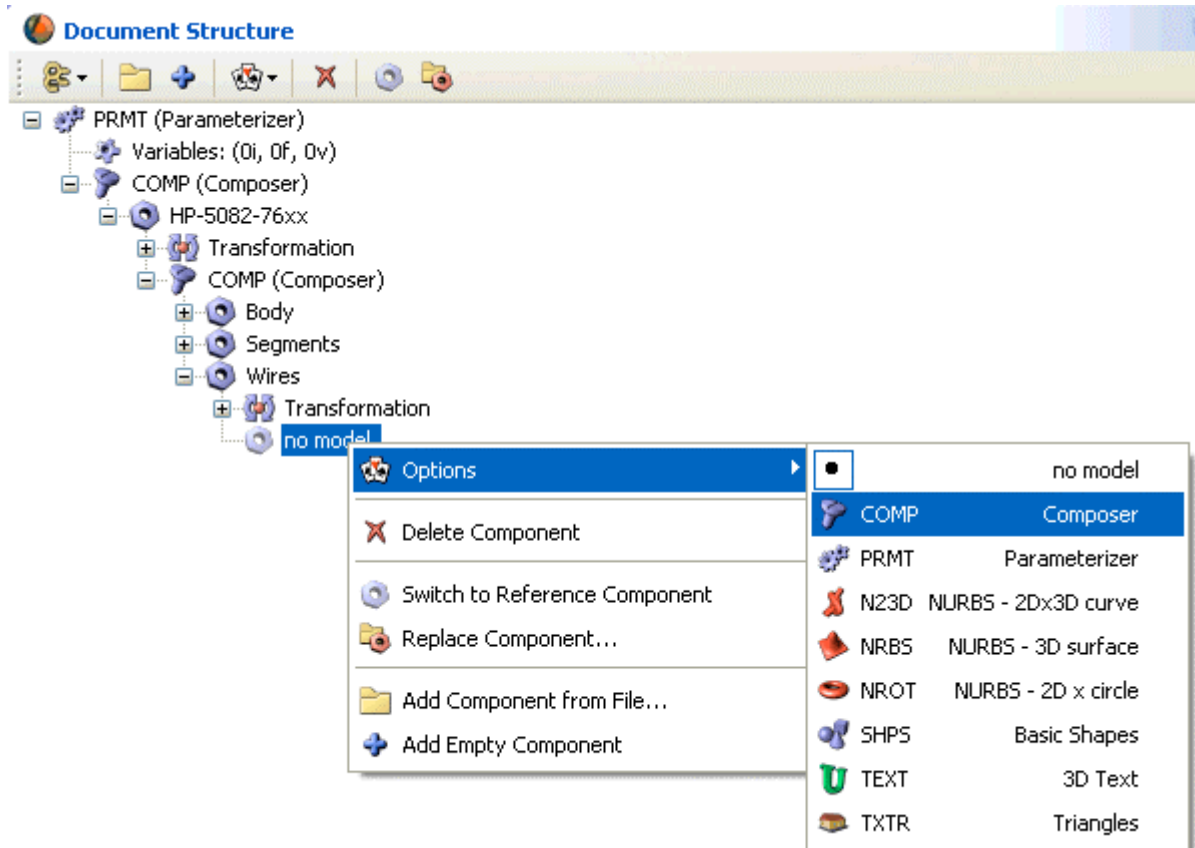
Pour y voir clair dans notre structure de composants et continuer notre travail dans de bonnes conditions, nous allons replier toutes les branches qui ne nous serviront pas, à savoir celles relatives aux containers Body et Segments, et laisser dépliée celle relative au container Wires. Pour cela, cliquer sur le signe – des branches à replier, et cliquer sur le signe + de la branche Wires à déplier.



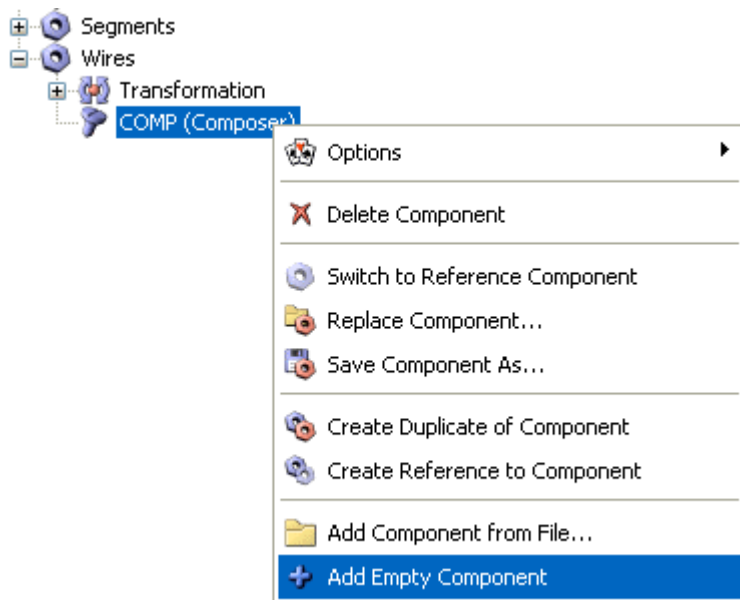
### Création d'une patte

Pour ces pattes de connexion, nous allons faire connaissance avec un autre type d'objet 3D. Il existe plusieurs façons possibles de créer des pattes de connexion, et nous pourrions les dessiner plates ou cylindriques. Mais comme nous aimons tous un peu de changement de temps en temps, nous allons utiliser des pattes rectangulaires allongées, et pour ce faire, utiliser des objets 3D simples de type TANK.

De la même façon que nous l'avons fait avec les segments, nous allons créer un container pour l'ensemble des pattes, de telle sorte qu'il soit possible de les manipuler toutes en même temps si cela s'avérait nécessaire (on ne sait jamais). Cliquons donc droit sur No model, puis sélectionnons Options, puis COMP.



Dans ce container COMP, ajoutons un objet simple de type TANK. Pour ce faire, cliquer droit sur COMP et sélectionner la commande Add Empty Component.



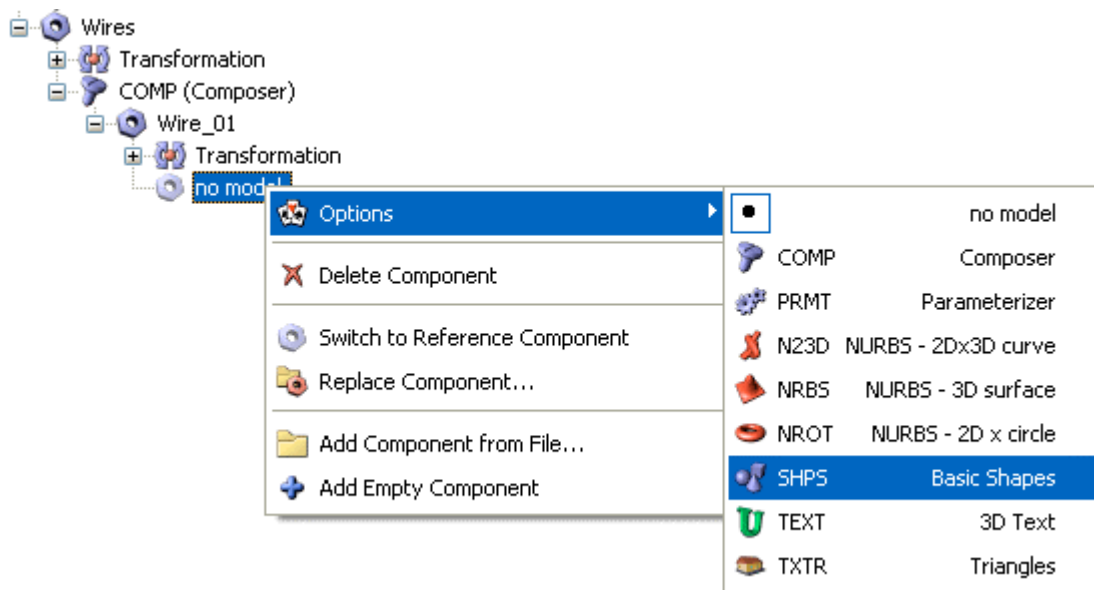
Cliquer sur le signe + situé à gauche de COMP pour voir le nouveau composant créé, qui s'appelle Unnamed Component. Renommer ce composant en Wire\_01, en cliquant une fois dessus pour le sélectionner, puis en tapant directement le nouveau nom au clavier du PC.



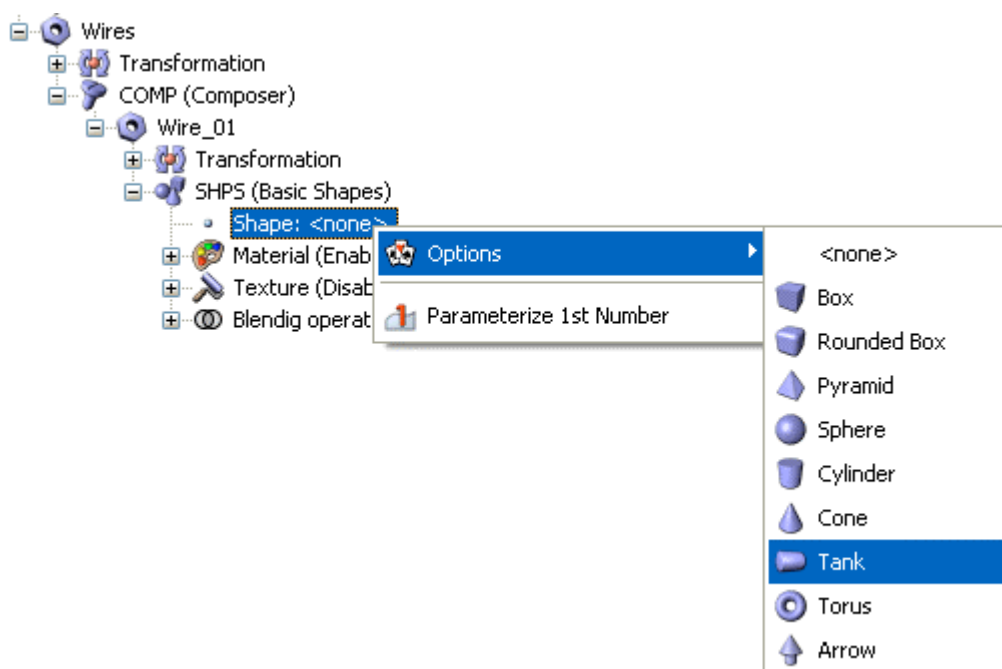


Tiens, il n'y a finalement pas eu besoin de cliquer deux fois sur le nom pour passer en mode d'édition... Je retiens cette astuce (vous pouvez faire de même).

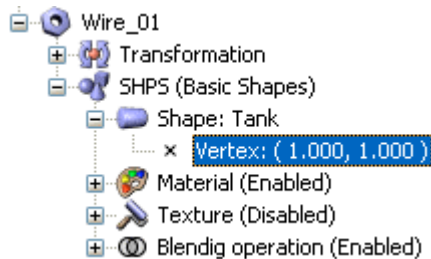
Reste à spécifier le type de composant que nous voulons utiliser pour l'élément Wire\_01. Développer la branche Wire\_01 (vous devez savoir comment on fait, désormais), puis cliquer droit sur No model, puis sélectionner successivement les commandes Options et SHPS (Basic Shapes).



Développer la branche SHPS ainsi créée, puis cliquer droit sur Shape: <none>, pour sélectionner les commandes Options puis Tank.

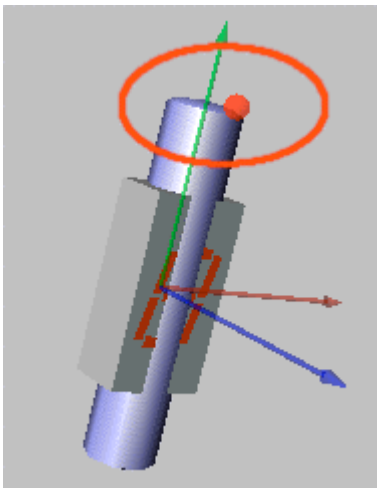


Houlà ! Un énorme bloc cylindrique recouvre entièrement tout ce que j'ai déjà fait ! Aurais-je fait une bêtise ? Je vous rassure : non, point du tout. La taille de ce nouveau composant est là aussi fixé par défaut avec les valeurs de Vertex à [1.000, 1.000], ce qui est bien plus gros que les dimensions attribuées aux parties déjà réalisées (qui sont toutes inférieures à 1.000).



Si nous voulons que les pattes de l'afficheur puissent entrer dans les trous du circuit imprimé sans trop les élargir, nous devons diminuer les dimensions de ce nouvel objet. Plutôt que de jouer avec les sliders ou entrer les valeurs manuellement, apprenons plutôt à dimensionner les composants directement dans la vue 3D.

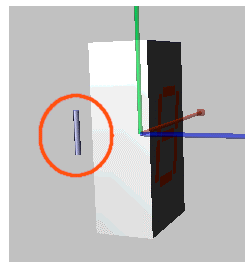
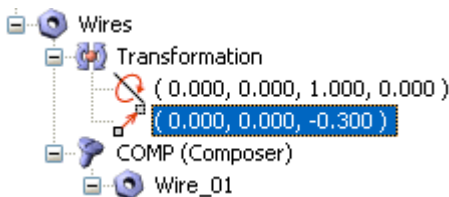
Assurez-vous que l'élément Vertex est sélectionné, cliquer dessus une fois le cas échéant. Puis dans la vue 3D, cliquer sur le rond bleu situé sur le composant actuellement sélectionné (le rond devient rouge), et laisser le bouton gauche de la souris enfoncé en même temps que vous déplacez la souris.



En cliquant sur le rond bleu situé à une extrémité du composant sélectionné (il devient rouge), il est possible de redimensionner le composant simplement en bougeant la souris. Cette méthode permet de dégrossir les valeurs de taille des éléments, de façon assez rapide et intuitive.

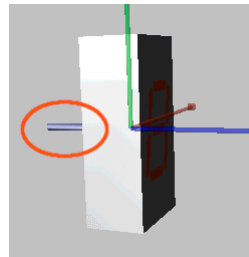
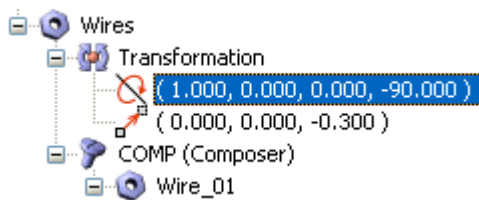
Prenez l'habitude de regarder en même temps les valeurs situées dans l'arborescence de gauche, quand vous redimensionnez ainsi un composant, cela permet de se familiariser avec le rôle de chaque paramètre.

La taille de la patte que l'on voit sur la copie d'écran ci-avant, est visiblement encore trop grosse, il faut encore réduire les valeurs de Vertex. Essayez donc avec les valeurs Vertex [0.015, 0.100]. Hum... nous voilà bien embêté, car si ces nouvelles valeurs sont correctes, nous ne voyons plus en revanche, la petite patte. Vous dites ? Au centre ? Mais vous avez tout à fait raison ! La patte est plus petite que le corps de l'afficheur (Body) et elle est masquée. Ce type de mésaventure nous est déjà arrivé, et cela fait plaisir de constater que votre mémoire fonctionne parfaitement. Sachant cela, puis-je suggérer de déplacer la patte dans l'espace 3D ? Avec les paramètres de position du bloc Wires, peut-être ? Essayons donc les valeurs suivantes et observons ce qui se passe.



Ah ! La patte réapparaît à côté du corps principal. Voilà une bonne chose.

Mais une chose me gêne tout de même depuis tout à l'heure. Si l'orientation de cette patte reste telle qu'elle est, il va être plutôt délicat de la positionner dans des trous, ne croyez-vous pas ? Il me semble qu'une réorientation s'impose. Peut-être pouvons-nous essayer de toucher un peu aux paramètres d'orientation pour voir ce que nous pouvons faire ?



Gagné du premier coup (quel hasard) ! La chose paraît finalement si simple à faire, que l'on pourrait presque se demander si elle était compliquée. Mais c'est toujours pareil : avant de savoir, on ne sait pas.

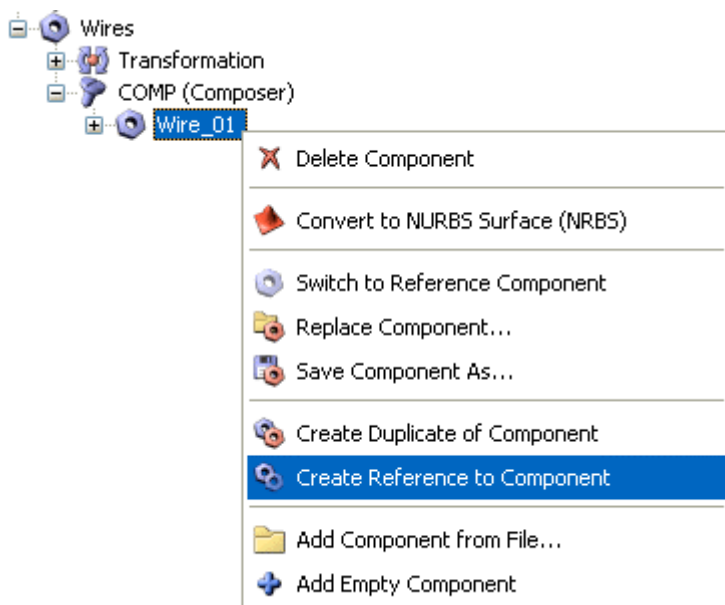
Nous avons finalement réussi sans trop de peine, à constituer une patte de connexion, qui si elle n'est pas le reflet exact de la réalité, n'en demeure pas moins suffisamment réaliste pour qu'on la laisse en l'état. Rappelez-vous que le plus important est de conserver un objet qui prend le moins de place possible en mémoire (il sera dessiné plus rapidement), et dont la forme se rapproche tout de même suffisamment de la réalité (nous ne faisons pas un concours de beauté).

Reste à créer quelques autres pattes pour donner à notre afficheur un look un peu plus sympa.

## Duplication de la patte

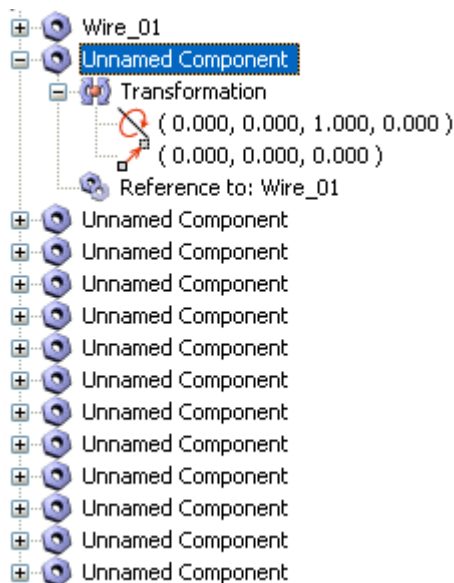
Nous allons maintenant dupliquer l'unique patte de connexion créée, afin de disposer de 14 éléments identiques. L'afficheur que nous cherchons à modéliser ne possède pas 14 pattes, mais il est plus facile de créer 14 pattes pour former un modèle d'afficheur générique, et ensuite supprimer celles qui n'existent pas pour le modèle désiré.

Cliquer droit sur l'élément Wire\_01, puis sélectionner la commande Create Reference to Component. Ne sélectionnez pas la commande Create Duplicate of Component.



En effet, nous allons cette fois créer des copies dont seules les positions changeront. C'est ce que l'on aurait pu faire pour les segments horizontaux et pour les segments verticaux, si nous avions au départ décidé de n'afficher toujours que le même chiffre.

Répéter l'opération encore douze fois, de façon à obtenir 13 composants référencés à l'objet Wire\_01, portant ainsi le nombre total de connexions à 14. Si nous développons la branche d'un des composants nouvellement créés, nous verrons qu'ils ne possèdent que peu de paramètres modifiables. En fait, seuls les paramètres d'orientation et de position sont modifiables, tous les autres paramètres, tels que dimensions ou couleur de l'objet, sont liés au composant qui sert de référence, en l'occurrence Wire\_01. Si l'on modifie les dimensions ou la couleur de l'objet Wire\_01, les treize objets liés (référéncés) à Wire\_01 seront modifiés en même temps de la même façon. Inutile de s'étendre sur l'intérêt d'une telle fonction...

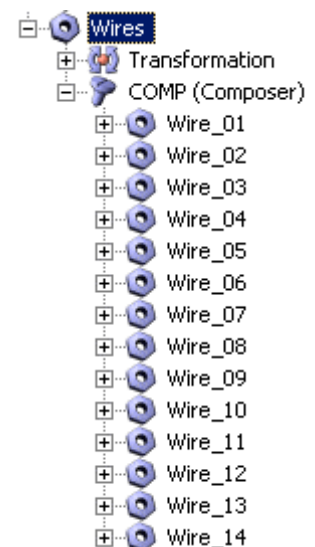


Nous avons là un objet de référence (Wire\_01) auquel sont rattachés (référéncés) 13 objets de même type. Ces 13 objets sont actuellement dénommés Unnamed Component.

<<<----

Bien entendu, vous avez anticipé la demande, et vous avez renommé ces 13 objets, de Wire\_02 à Wire\_14.

----->>>



A l'étape où nous en sommes, nous avons 14 pattes identiques, mais il semble qu'il n'y en ait qu'une seule car toutes sont actuellement superposées. Nous allons donc maintenant modifier la position de chaque patte pour les aligner en deux lignes de sept pattes chacune.

## Positionnement des pattes

Il serait tout à fait possible de spécifier manuellement la position pour chaque patte, de façon individuelle. Il nous suffirait de définir la position d'une des quatre pattes extrêmes, puis d'ajouter un certain décalage (offset), de plus en plus important, pour les pattes suivantes. C'est cette méthode, peu productive, que j'ai utilisé pour mon premier afficheur sept segments. Mais ce n'est pas celle que nous allons employer ici, car il en existe une autre bien plus pratique, qui si elle peut paraître plus longue à mettre en œuvre, peut en revanche permettre par la suite des manipulations bien plus pratiques et plus rapides.

### Utilisation de formules mathématiques

Je suis désolé autant que vous pouvez l'être, et les termes "formules mathématiques" devraient à jamais être bannis des tutoriels. Mais il faut bien admettre que certaines fois, les formules mathématiques peuvent servir à simplifier certaines tâches. Même les vôtres. Et ce qui va suivre va vous en convaincre, j'en suis sûr.

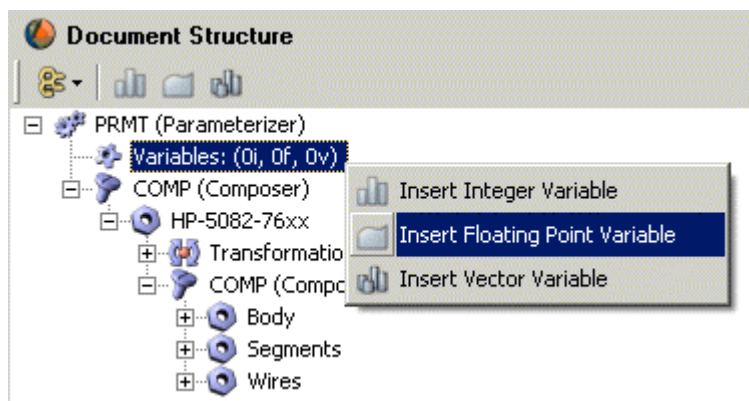
Avant de continuer, il faut savoir que les paramètres des divers éléments de RWIE, tels que position, taille, orientation ou encore couleur d'un objet, peuvent prendre une valeur que vous spécifiez manuellement, ou une valeur qui est spécifiée "ailleurs", dans une variable portant un nom bien précis que vous décidez vous-même.

Imaginons un cas pratique qui risque de se produire plus souvent que vous ne pouvez l'imaginer : vous avez un certain nombre de pattes d'une certaine longueur, que vous souhaitez modifier. S'il y a 8 pattes, vous devez effectuer l'opération 8 fois de suite, puisqu'il faut modifier la longueur de chaque patte, l'une après l'autre, pour l'amener à la valeur désirée. Imaginons maintenant que le paramètre qui spécifie la longueur des pattes, ne soit

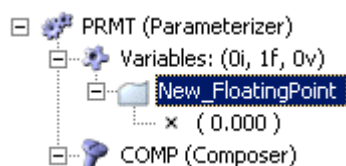
pas une valeur numérique telle que 0.100, mais un nom en toutes lettres, par exemple LongueurPattes. A un endroit bien particulier, qui pourrait bien être un container de variables, on décide d'attribuer à ce nom, la valeur 0.100. Si maintenant, au lieu de spécifier la valeur 0.100 au paramètre de taille de chaque patte, on spécifie la variable LongueurPattes, chaque composant (chaque patte) ne va pas lire la valeur 0.100, mais la valeur LongueurPattes. Bien évidemment, ce nom, en terme de valeur numérique, ne signifie rien en tant que tel. Le composant va donc chercher à savoir s'il existe un paramètre particulier qui porte ce nom (dans un container de variables par exemple), et s'il existe, il va utiliser la valeur numérique qui a été affectée à ce nom. Par la suite, si vous modifiez la valeur affectée à la variable LongueurPattes (par exemple remplacement de la valeur initiale de 0.100 par la valeur 0.150), les huit pattes vont automatiquement mettre à jour leur longueur avec cette nouvelle valeur. En résumé, on modifie la longueur de huit pattes en une seule action au lieu de huit actions. Ce principe peut aussi, bien sûr, être adopté pour la position des pattes, sur laquelle il devient possible de jouer en affectant toutes les pattes en même temps. Gain de temps de rapport 8 au moment de la modification... pour 8 pattes. Sans commentaire.

Après ce petit texte destiné à vous convaincre du bien fondé de ce choix visant à faire appel à des variables plutôt qu'à des nombres "en dur", nous pouvons passer à la pratique.

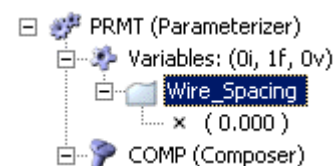
Tout en haut de la structure de nos objets 3D, cliquer droit sur Variables (directement sous PRMT), puis sélectionner la commande Insert Floating Point Variable.



Une variable appelée New\_FloatingPoint est créée, à cette variable est affectée la valeur par défaut 0.000.

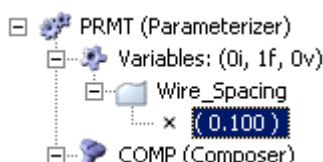


Nouvelle variable New\_FloatingPoint



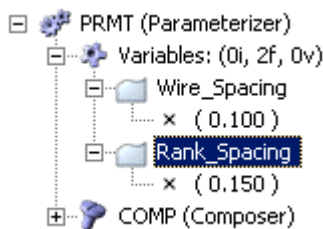
Nouvelle variable renommée en Wire\_Spacing

Comme nous utilisons l'unité de mesure anglaise (inch) et que l'espacement entre les pattes est de 0.100 in (ce qui correspond à 2.54 mm), nous attribuons la valeur 0.100 à la variable Wire\_Spacing.



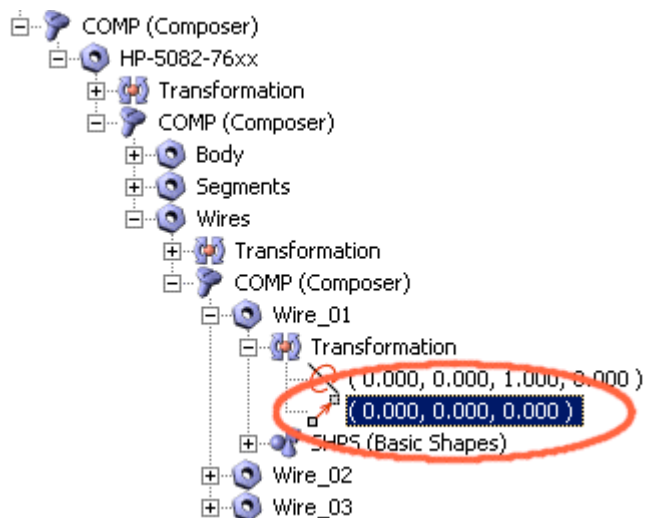
Nous désirons aligner deux rangées de 7 pattes, ces deux rangées seront éloignées autant l'une que l'autre du centre de référence qui est toujours le point [0.000, 0.000, 0.000]. Nous aurons donc une rangée dont la valeur d'éloignement sera positive, et une rangée dont la valeur d'éloignement sera négative. Le datasheet de l'afficheur nous indique que la valeur de l'espacement entre les deux rangées est de 0.300 in. Nous en déduisons qu'il nous

faut utiliser les valeurs d'espacement de -0.150 et 0.150 par rapport au point [0.000] si nous voulons centrer l'ensemble. Créons donc à cet effet une nouvelle variable dans le container des variables, qui contiendra la valeur de l'espacement des rangées par rapport au centre, et appelons cette variable Rank\_Spacing.

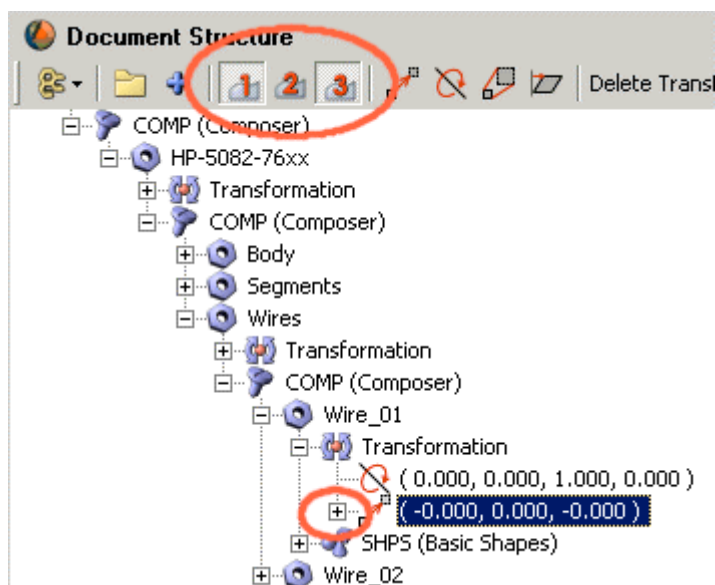


Et voilà, nous n'avons pas besoin d'aller plus loin, nos deux variables Wire\_Spacing et Rank\_Spacing vont déjà nous être très utiles, comme nous allons bientôt le voir.

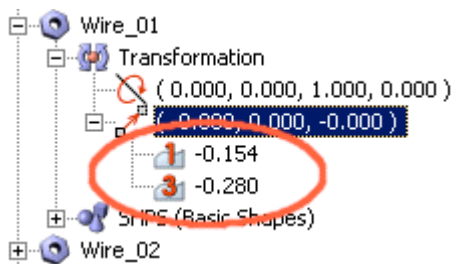
A présent, utilisons nos deux variables là où elles seront les plus utiles, à savoir dans les paramètres de positionnements de nos pattes. Cliquer sur tous les signes + qui permettent de développer la branche Wire\_01, afin d'avoir accès aux paramètres de positionnement de cet objet.



Cliquer sur les paramètres de position pour les mettre en surbrillance, puis cliquer sur les deux boutons "1" (Parameterize 1st Number) et "3" (Parameterize 3rd Number) situés dans la batte d'outils de la structure.

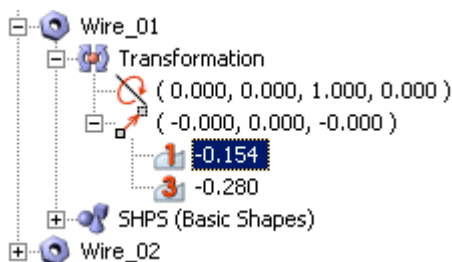


Un signe + a été ajouté à gauche des paramètres de position, indiquant que des éléments y ont été ajoutés. Allons donc voir de quoi il s'agit, en cliquant sur ce signe +.



Nous constatons que deux paramètres ont été ajoutés : un premier qui s'appelle "1" et qui correspond au premier paramètre de position, et un second qui s'appelle "3" et qui correspond au troisième paramètre de position. Les trois paramètres de position correspondent, de gauche à droite, aux axes X, Z et Y. Le paramètre "1" correspond donc à la valeur de l'axe X, et le paramètre "3" correspond à la valeur de l'axe Y. L'axe X est celui que nous utiliserons pour définir l'écartement des rangées de pattes par rapport au centre. Nous décidons donc que le paramètre "1" contiendra la valeur d'espacement des rangées de pattes, elle-même contenue dans la variable Rank\_Spacing.

Cliquer une fois sur le paramètre "1" pour le sélectionner.

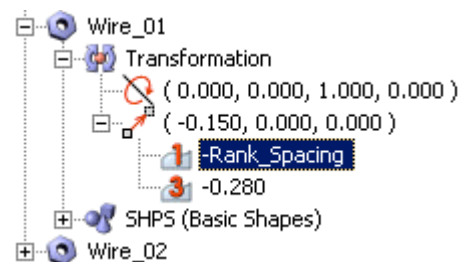


Puis en deux étapes séparées :

- Etape 1 : saisir le nom de la variable Rank\_Spacing, avec un trait de soulignement entre "Rank" et "Spacing".
- Etape 2 : ajouter un signe moins juste devant le nom Rank\_Spacing



Etape 1

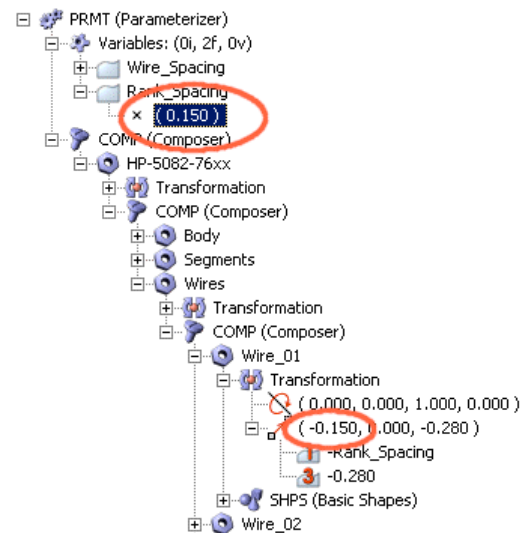
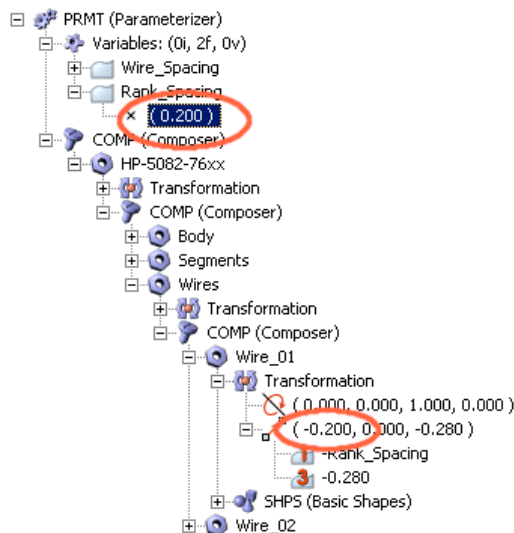


Etape 2

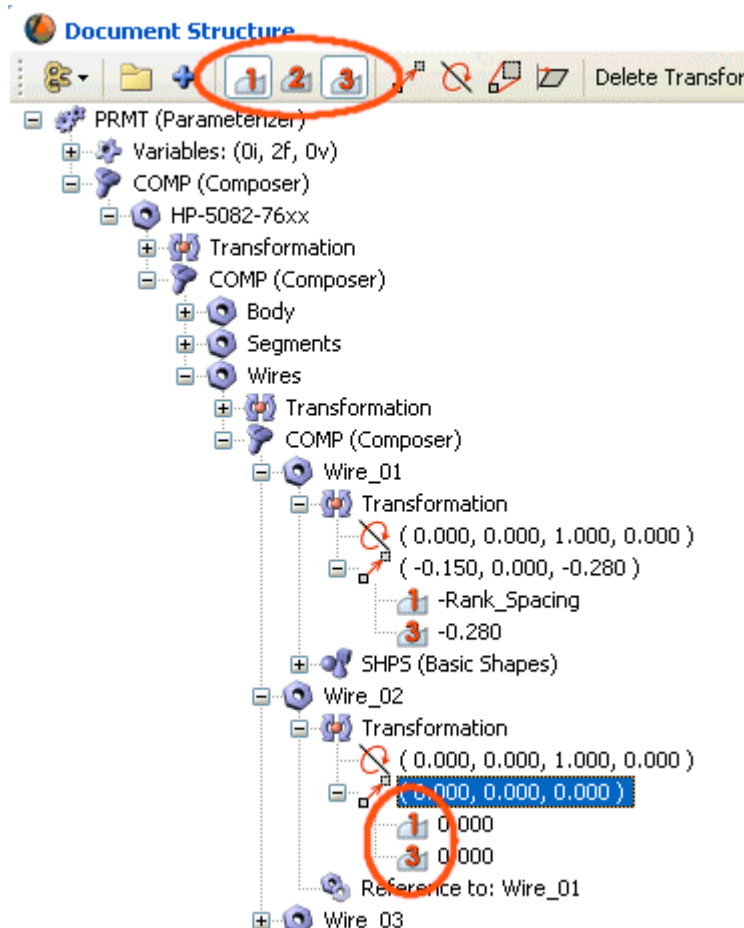
A chaque étape, notez le changement de valeur dans la ligne des paramètres de position. Vous remarquerez ainsi que la valeur de l'axe X a été modifiée, pour devenir 0.150 dans l'étape 1, et devenir -0.150 à l'étape 2. Cette valeur correspond exactement à la valeur qui a auparavant été affectée à la variable Rank\_Spacing.

Pour dissiper tout doute éventuel, essayez de modifier la valeur de la variable Rank\_Spacing, et observez ce qui se passe sur la ligne de position de la patte Wire\_01.





Maintenant que tout est clair dans tous les esprits, nous allons répéter l'opération d'affectation de la variable Rank\_Spacing au paramètre de position de l'axe X de toutes les pattes. Développer la branche de l'objet Wire\_02, cliquer sur la ligne des paramètres de position de l'objet pour la sélectionner, puis activer les deux boutons "1" et "3" de la barre d'outil.



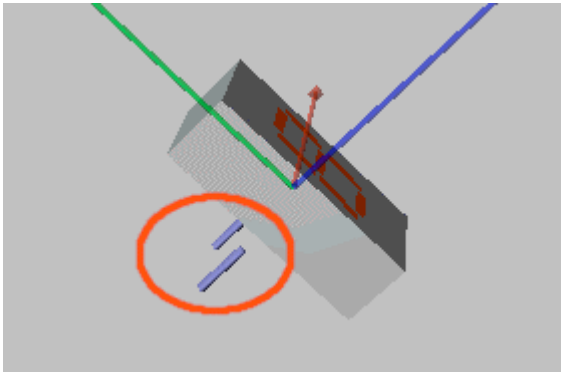
Puis faites de même pour les douze autres pattes (vous verrez, ce n'est pas si long que ça). Une fois cela réalisé, affectez le nom "-Rank\_Spacing" au paramètre 1 des sept premières pattes Wire\_01 à Wire\_07 (pour la première c'est déjà fait), et affectez le nom "Rank\_Spacing" au paramètre 1 des sept dernières pattes Wire\_08 à Wire\_14.



**Astuce pour aller plus vite :**

- cliquer sur le nom "-Rank\_Spacing" affecté au paramètre "1" de la patte Wire\_01 pour le sélectionner,
  - appuyez simultanément sur les touches CTRL+C de votre clavier (raccourci clavier de l'opération Copier),
  - cliquer sur la valeur 0.000 du paramètre "1" à modifier en "-Rank\_Spacing" pour la sélectionner,
  - appuyez simultanément sur les touches CTRL+V de votre clavier (raccourci clavier de l'opération Coller).
- Notez bien que pour les sept premières pattes, il y a le signe - devant le nom Rank\_Spacing, ce qui n'est pas le cas pour les sept autres.

Une fois ce petit travail terminé, vous devriez visualiser dans la vue 3D, deux pattes, en plus du corps et des segments, qui heureusement n'ont pas disparus entre temps (ce qui nous évitera de revenir en début du document).



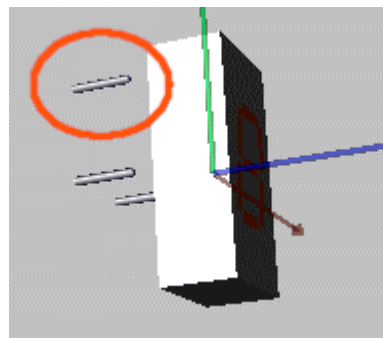
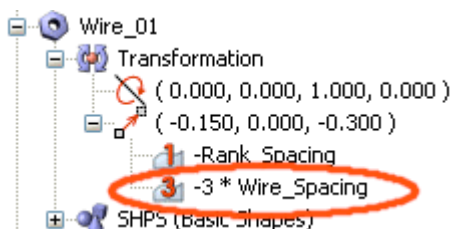
On ne voit que deux pattes, car d'un côté comme de l'autre (par rapport au centre 0.000), sept pattes identiques sont superposées. Les seuls mouvements que nous avons fait faire aux pattes ont en effet eu lieu sur l'axe X.

Afin de visualiser individuellement toutes les pattes, nous devons maintenant faire quelque chose avec l'axe Y. En l'occurrence, espacer chaque patte de chaque rangée, de sa voisine, avec un intervalle de 0.100 in. Vous vous souvenez ? C'est la valeur que nous avons donnée à la variable Wire\_Spacing. Allez, on y va !

Elevons la position de la patte Wire\_01, qui tout comme la patte Wire\_14, doit se trouver "en haut" de la rangée. Pour cela, cliquer une fois sur le paramètre "3" de la patte Wire\_01, et saisir la formule suivante :

`-3 * Wire_Spacing`

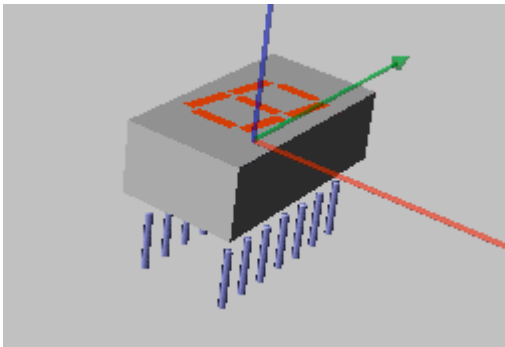
Ce qui a pour effet de déplacer la patte en question vers le haut. Nous voyons donc désormais trois pattes dans la vue 3D.



Ne nous arrêtons pas en si bon chemin, et affectons les formules (simples) suivantes à l'ensemble des pattes.

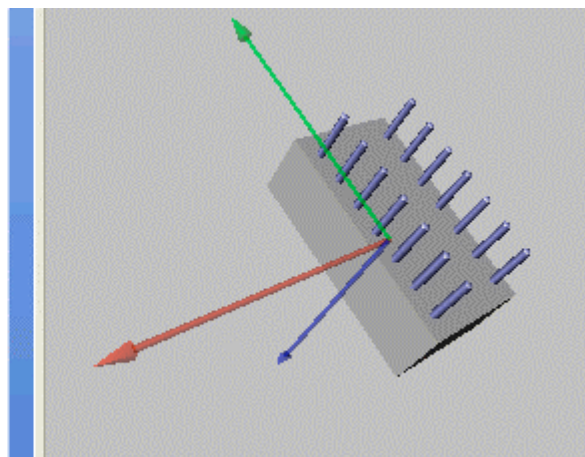
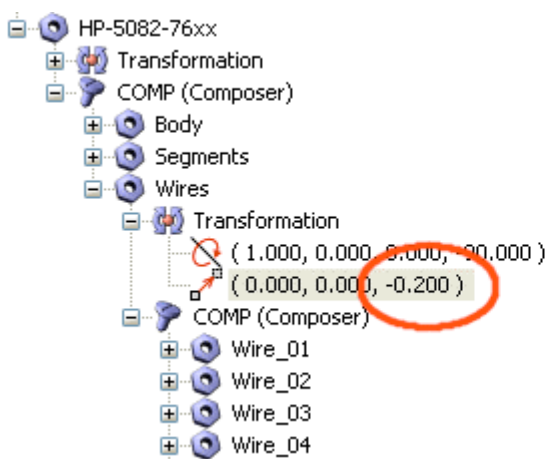
Patte	Formule pour le paramètre "1"	Patte	Formule pour le paramètre "1"
Wire_01	<code>-3 * Wire_Spacing</code>	Wire_08	<code>-3 * Wire_Spacing</code>
Wire_02	<code>-2 * Wire_Spacing</code>	Wire_09	<code>-2 * Wire_Spacing</code>
Wire_03	<code>-1 * Wire_Spacing</code>	Wire_10	<code>-1 * Wire_Spacing</code>
Wire_04	<code>0 * Wire_Spacing</code>	Wire_11	<code>0 * Wire_Spacing</code>
Wire_05	<code>1 * Wire_Spacing</code>	Wire_12	<code>1 * Wire_Spacing</code>
Wire_06	<code>2 * Wire_Spacing</code>	Wire_13	<code>2 * Wire_Spacing</code>
Wire_07	<code>3 * Wire_Spacing</code>	Wire_14	<code>3 * Wire_Spacing</code>

Et voilà ce que les formules précédentes permettent d'obtenir :



Nous avons maintenant nos 14 pattes visibles, mais elles ne sont pas encore très bien "plaquées" contre le corps de l'afficheur. Mais vous devez bien vous douter un peu que nous ne sommes plus très loin du résultat final...

Le rapprochement des 14 pattes vers le corps de l'afficheur pourrait s'opérer en travaillant individuellement la position de chaque patte l'une après l'autre. Mais ce serait faire une sacrée marche arrière par rapport à la méthodologie adoptée jusqu'ici. Procédons donc dans les règles de l'art, en modifiant la position du bloc complet des 14 pattes, grâce aux paramètres de position du container Wires.



Joli composant, vous ne trouvez pas ?

Nous pourrions fort bien nous arrêter là, puisque nous disposons dès lors, d'un objet tout à fait "fonctionnel". Mais nous avons prévu dès le départ, la possibilité d'afficher un nombre quelconque dans la couleur de notre choix. Nous allons donc dans les sections suivantes, nous attacher à rendre cela possible.

**Remarque :** les étapes suivantes ne sont nullement nécessaires pour effectuer l'export dans un fichier au format 3DS. Si vous désirez effectuer un export dès maintenant, vous pouvez tout à fait le faire sans tenir compte de ce qui suit, et aller directement au chapitre Export au format 3DS.

## Choix de la couleur des segments

Facile, allez-vous dire ! Il suffit de modifier la valeur des composantes rouge, vert et bleu de chaque segment, accessibles à la ligne Material / RGBA ! Exact, vais-je répondre. Mais auriez-vous par hasard l'intention de faire cela pour chaque segment ? Cela porte à 7, au moins, le nombre de modifications à effectuer pour un seul changement de valeur d'une composante de couleur.

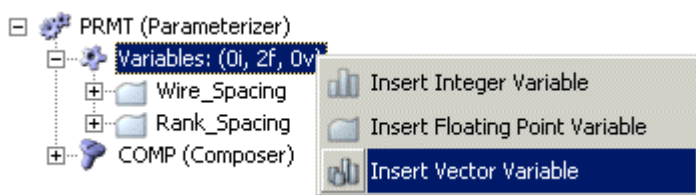
Imaginez que vous souhaitiez passer du rouge au vert : vous devez diminuer la valeur de la composante rouge, puis augmenter celle de la composante verte. Au total, 14 changements de valeur pour sept segments. Cela ne vaudrait-il pas la peine de prendre quelques minutes pour voir si plus simple n'est pas possible ?

Tenez, une suggestion. Que se passerait-il si nous affections une nouvelle variable appelée Red, à chaque paramètre "Rouge" de la ligne RGBA ? Et bien en modifiant une seule fois la valeur de la variable Red, cette valeur serait propagée à tous les segments en même temps. Merveilleux, non ? Soit, créons donc trois variables appelées Red, Green et Blue et appliquons le principe énoncé.

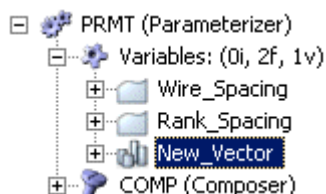
Sauf que...

Sauf qu'il existe une façon de faire encore plus élégante. Et oui, le concepteur du logiciel RWIE avait déjà pensé à ce type de besoin, qui n'est certes pas unique à notre cas d'école. Et il a déjà prévu l'usage d'une variable de type "Couleur", qui ne consomme qu'une seule ligne de variable. Mais arrêtons donc d'en parler, et passons à la pratique, car je sens que cela vous démange.

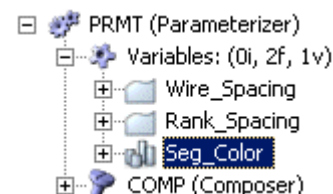
Cliquer droit sur la ligne Variables tout en haut de l'arborescence des paramètres, puis sélectionner la commande Insert Vector Variable.



Une nouvelle variable, appelée New\_Vector, est ajoutée à la suite des deux variables existantes Wire\_Spacing et Rank\_Spacing.



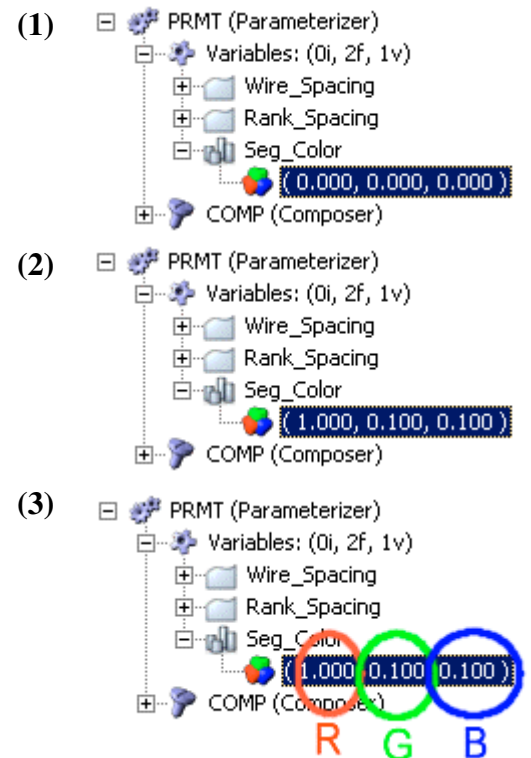
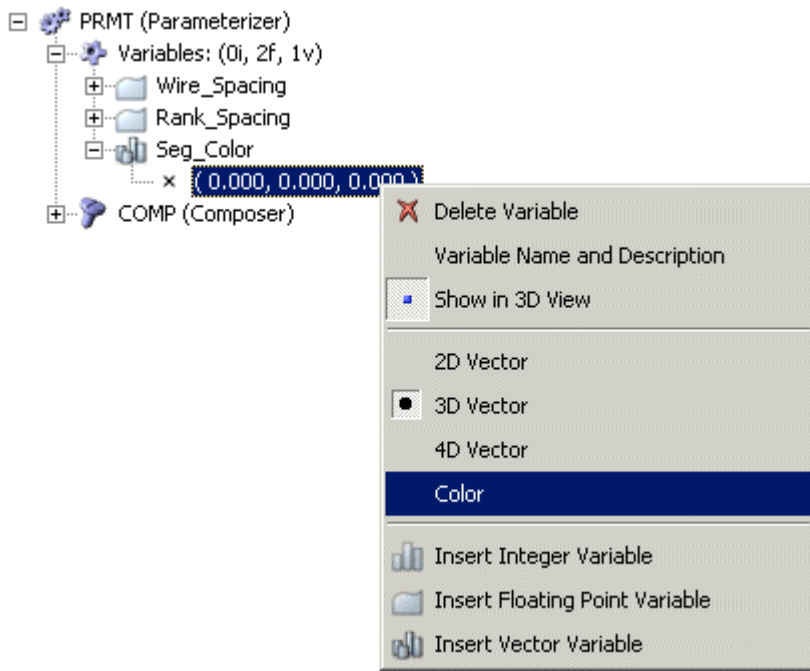
Insertion d'une nouvelle variable "New\_Vector"



Variable "New\_Vector " renommée en "Seg\_Color"

Développer la branche Seg\_Color en cliquant sur le signe + situé à sa gauche. Cliquer droit sur la ligne comportant les trois valeurs actuellement à 0.000, puis sélectionner l'option Color (actuellement fixée à 3D Vector). Cela aura pour effet de signifier au logiciel que les valeurs attribuées à la variable Seg\_Color, correspondront aux valeurs des composantes Rouge, Vert et Bleu.

Profitez-en pour affecter dès maintenant les valeurs [1.000, 0.100, 0.100] à la variable Seg\_Color.

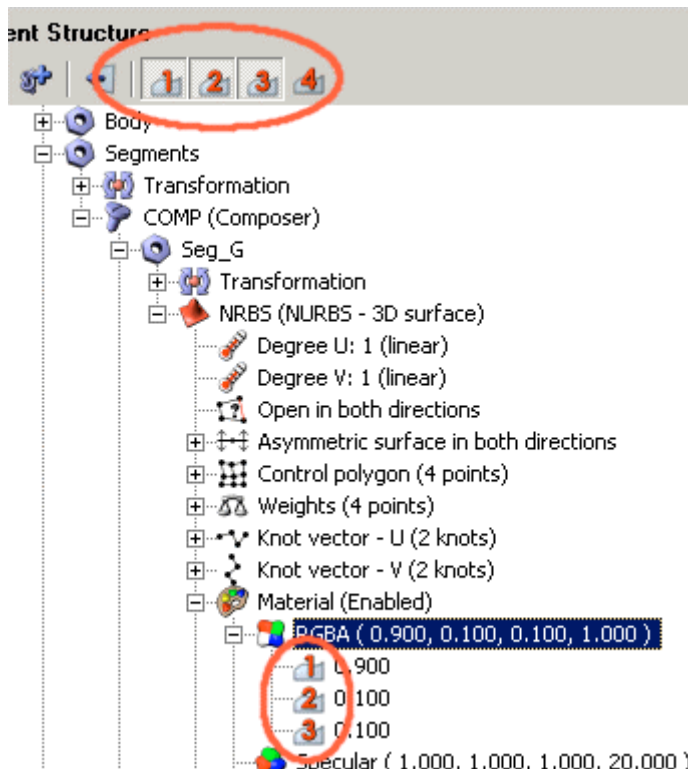


(1) : Seg\_Color a été spécifiée comme étant une variable de type Couleur.

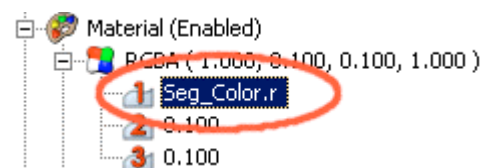
(2) : Seg\_Color ayant une couleur avec dominante de Rouge.

(3) : Valeur Rouge stockée dans le premier paramètre, Vert dans le second, Bleu dans le troisième.

Développer la branche Seg\_G de telle sorte que soit visible la ligne des paramètres de couleur RGBA. Cliquer sur la ligne des paramètres de couleur RGBA, puis activer les trois boutons "1", "2" et "3" de la barre d'outils située directement au-dessus de la structure des paramètres. Une fois cela fait, cliquer sur la ligne "1" pour la sélectionner, puis taper la ligne suivante : Seg\_Color.r

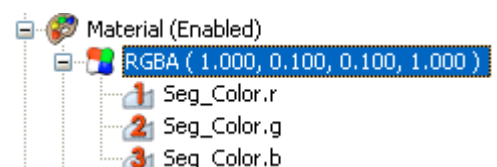


1 - Cliquer sur la ligne "1" pour la sélectionner, puis taper le texte Seg\_Color.r



2 - Cliquer sur la ligne "2" pour la sélectionner, puis taper le texte Seg\_Color.g

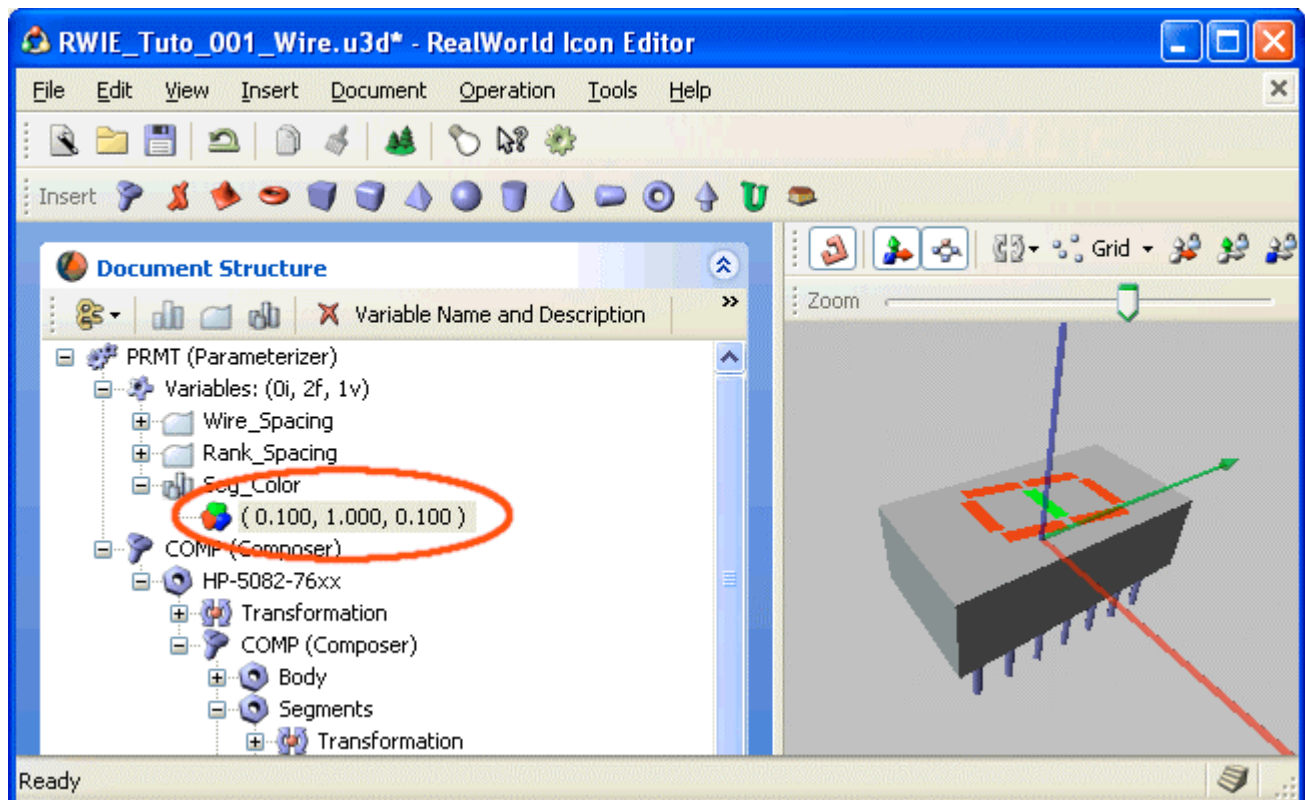
3 - Cliquer sur la ligne "3" pour la sélectionner, puis taper le texte Seg\_Color.b



### Explication :

La variable Seg\_Color, de type Couleur, possède trois paramètres, qui sont les paramètres Rouge, Vert et Bleu. Si dans une formule nous utilisons la valeur `Seg_Color.r` (Seg\_Color et lettre "R" séparés par un point), cela signifie que l'on se réfère au paramètre Rouge de la variable Seg\_Color (R pour Red, rouge). Si nous voulons utiliser la valeur du paramètre Bleu de la variable Seg\_Color, nous devons faire usage de la valeur `Seg_Color.b` (b pour Blue, bleu). De même pour le paramètre Vert, accessible au travers de `Seg_Color.g` (g pour Green, vert).

**Test :** Modification de la variable Seg\_Color pour qu'elle représente une couleur avec dominante verte. Modifier la valeur de quantité de rouge pour qu'elle égale 0.100, et modifier la valeur de quantité de vert pour qu'elle égale 1.000. Ne pas toucher à la quantité de bleu. Voici ce que vous devriez observer.



Le segment central Seg\_G s'est coloré en vert, alors que tous les autres segments restent colorés en rouge. Et cela sans modifier directement les paramètres de couleur de la ligne RGBA du segment Seg\_G. Il est aisé de comprendre que si nous avons modifié de la même façon tous les paramètres "1", "2" et "3" de chaque ligne RGBA des autres segments, tous se seraient allumés en vert à la suite de cette opération de test.

Vous avez envie de modifier les paramètres des autres segments pour que tous se comportent de la même manière, et que le changement de couleur apporté à la variable Seg\_Color se propage à tous les segments en même temps ? Vous pouvez le faire.

Mais si je peux me permettre un conseil, attendez un peu, car dans le chapitre suivant "Choix du chiffre à afficher", nous allons aller plus loin et ne plus nous contenter des simples lignes de texte de type Seg\_Color.r.



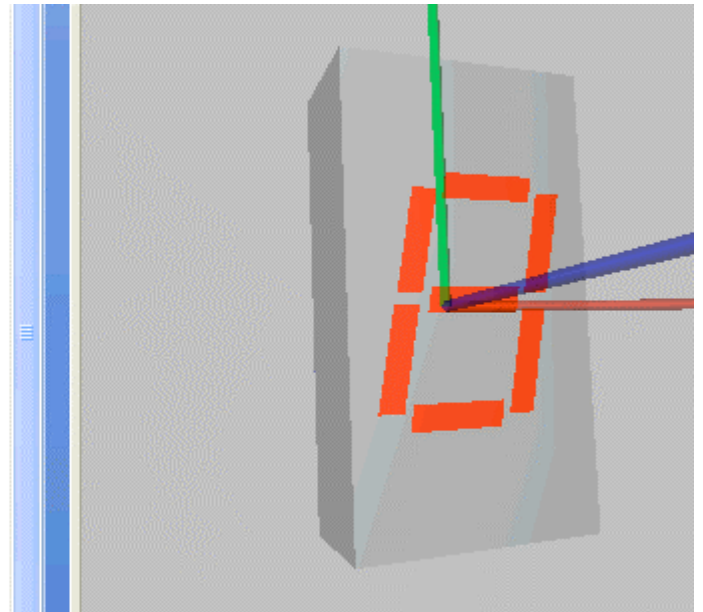
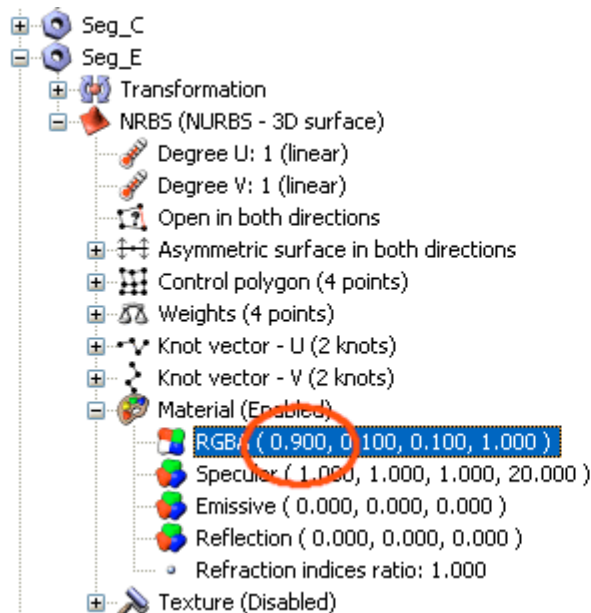
## Choix du chiffre à afficher

Question : comment faire pour "allumer" ou "éteindre" les segments ?

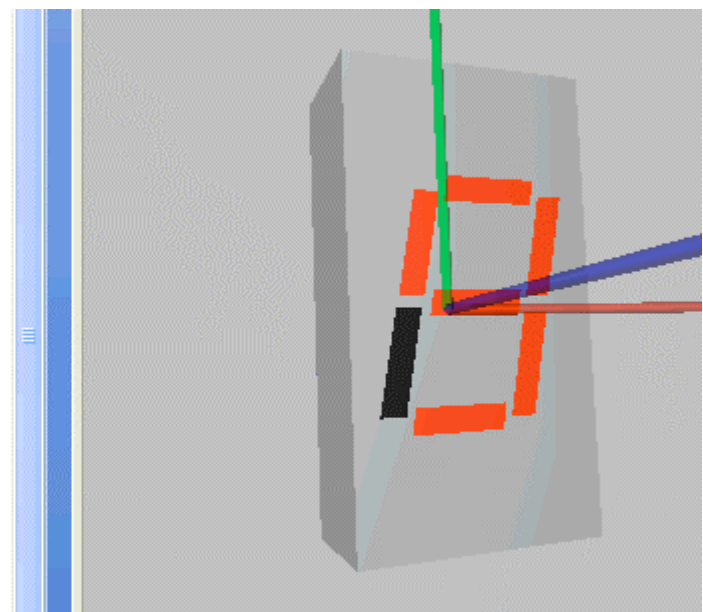
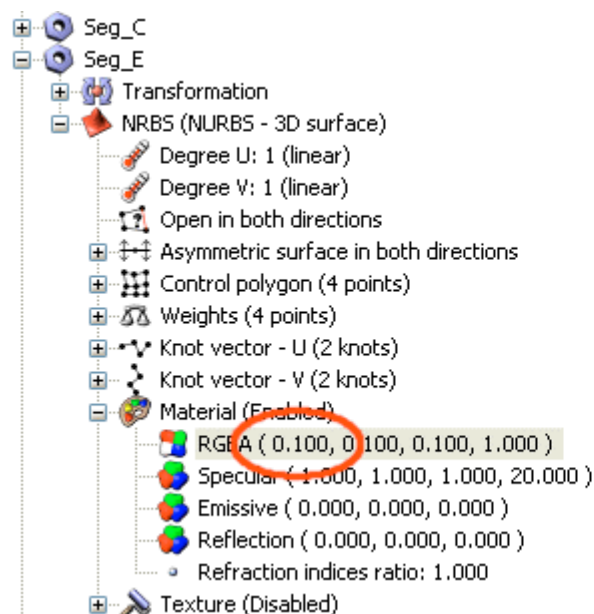
Réponse : en assombrissant les segments éteints.

Un exemple pratique va mettre cela rapidement en évidence. Eteignons par exemple le segment Seg\_E, en abaissant suffisamment la quantité de couleur rouge, et observons l'effet visuel obtenu.

Ici, tous les segments sont "allumés", la valeur de la composante de couleur rouge de chacun est élevée :



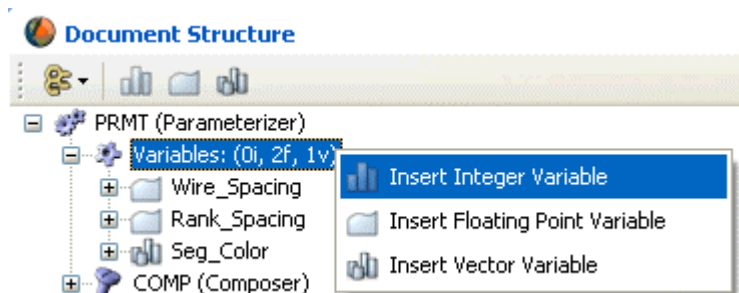
Et là, tous les segments sont "allumés" sauf le segment Seg\_E, dont la valeur de la composante de couleur rouge est faible :



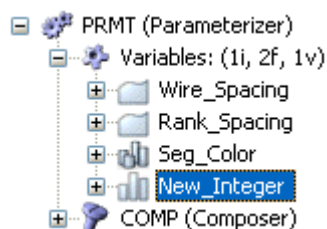
Quoi, c'est tout ? Il suffit de changer la valeur d'un paramètre, et le chiffre affiché est différent ? Et bien dans l'exemple précédent, oui. Mais faites maintenant les opérations qui s'imposent pour afficher le chiffre 1. Vous devez modifier non pas 1 paramètre, mais quatre, puisqu'il faut aussi "éteindre" les segments A, D, F et G en plus du segment E. Vous imaginez aisément que cette façon de faire risque d'être un peu lourde, même sans avoir l'intention de modifier l'affichage à tout bout de champ.

Comme nous l'avons déjà fait pour les pattes de connexion, et comme nous l'avons aussi fait pour la couleur du segment Seg\_G, nous allons définir une variable numérique que nous utiliserons dans des formules mathématiques. Ce sont ces formules mathématiques qui permettront de changer la couleur des segments désirés.

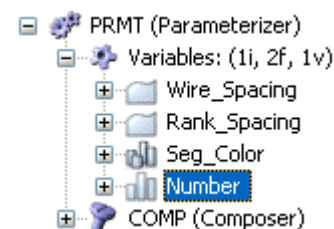
Cliquer droit sur la ligne Variables tout en haut de l'arborescence des paramètres, puis sélectionner la commande Insert Integer Variable.



Une nouvelle variable, appelée New\_Integer, est ajoutée à la suite des trois variables existantes Wire\_Spacing, Rank\_Spacing et Seg\_Color.

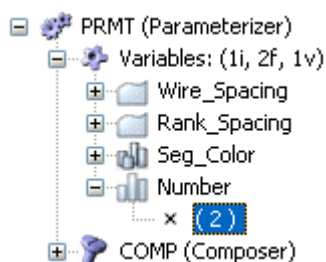


Insertion d'une nouvelle variable "New\_Integer"



Variable "New\_Integer" renommée en "Number"

Développer la branche Number en cliquant sur le signe + situé à sa gauche. Attribuer la valeur 2 à cette variable Number.



Maintenant que notre variable Number est créée, utilisons la dans les formules suivantes :

Pour le paramètre "1" de RGBA du segment Seg\_G :

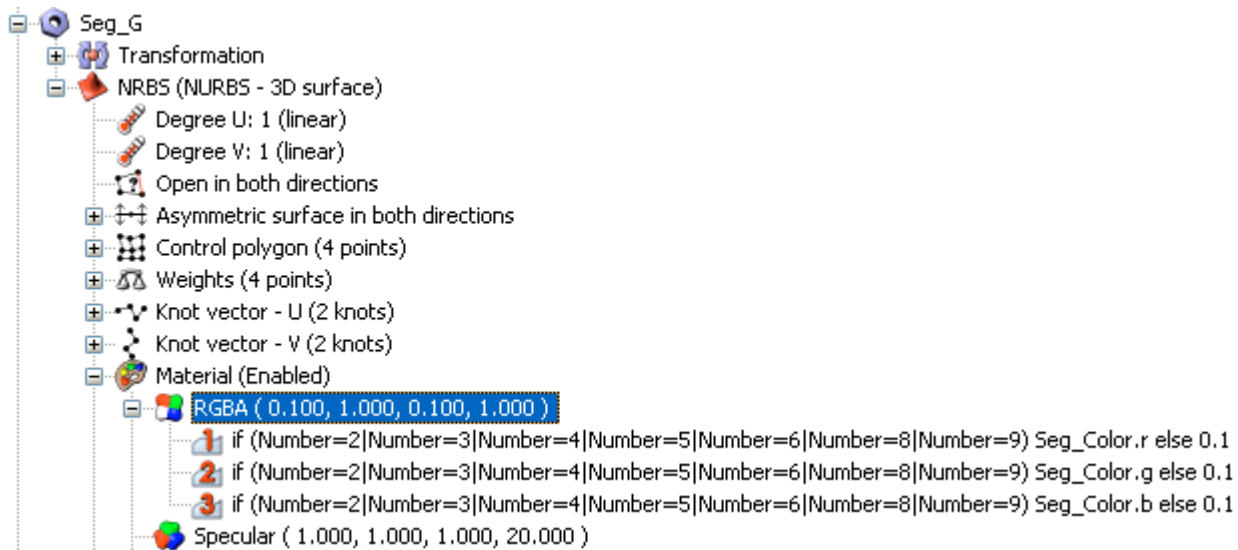
```
if (Number=2|Number=3|Number=4|Number=5|Number=6|Number=8|Number=9) Seg_Color.r else 0.1
```

Pour le paramètre "2" de RGBA du segment Seg\_G :

```
if (Number=2|Number=3|Number=4|Number=5|Number=6|Number=8|Number=9) Seg_Color.g else 0.1
```

Pour le paramètre "3" de RGBA du segment Seg\_G :

```
if (Number=2|Number=3|Number=4|Number=5|Number=6|Number=8|Number=9) Seg_Color.b else 0.1
```



Ces formules seront lues et exécutées (interprétées) au moment où le composant ira lire la valeur de ses composantes de couleur rouge, vert et bleu.

### Explications

Les formules peuvent contenir des mots clés, comme if..(then)..else, ce qui permet d'utiliser une valeur qui dépend d'une condition externe, basée ou non sur l'utilisation d'une ou de plusieurs variables. Ici, on regarde si ce qui est contenu entre les parenthèses, est vrai ou faux. Si c'est vrai (la variable Number vaut 4, par exemple), alors le paramètre "1" prendra la valeur de Seg\_Color.r. Si le contenu entre parenthèse est faux (la variable Number vaut 7, par exemple), alors le paramètre "1" prendra la valeur 0.1 (ou 0.100, c'est la même chose). En effet, les barres verticales, situées entre les égalités (par exemple entre Number=4 et Number=5) correspondent à des "ou". En clair, le contenu entre parenthèses est vrai si la variable Number vaut 2, **ou** si elle vaut 3, **ou** si elle vaut 4, etc. Et ce contenu est évalué comme faux si la valeur de la variable Number ne fait pas partie des nombres indiqués.

Notons au passage l'emploi possible de signes "supérieur à" ou "inférieur à", qui peuvent permettre dans certains cas de simplifier les formules. Dans le cas présent, les deux formules Formule version 1 et Formule version 2 qui suivent sont équivalentes et donnent exactement les mêmes résultats si la valeur attribuée à la variable Number reste comprise entre 0 et 9 (toute autre valeur n'aurait pas de sens ici).

#### Formule version 1 :

```
if (Number=2|Number=3|Number=4|Number=5|Number=6|Number=8|Number=9) Seg_Color.r else 0.1
```

#### Formule version 2 :

```
if (Number=2|Number=3|Number=4|Number=5|Number=6|Number>7) Seg_Color.r else 0.1
```

Appliquons maintenant ce type de formule à chaque paramètre de couleur RGBA de chaque segment, en tenant évidemment compte des cas où le segment doit être allumé. Par exemple, le segment A n'est allumé que lorsqu'on désire afficher les chiffres 0, 2, 3, 5, 6, 7, 8 et 9. Alors que le segment E n'est allumé que lorsqu'on désire afficher les chiffres 0, 2, 6 et 8. Afin de soulager un peu votre cerveau, qui doit pour certains d'entre vous être en pleine ébullition, voici ci-après les formules requises pour chaque segment.

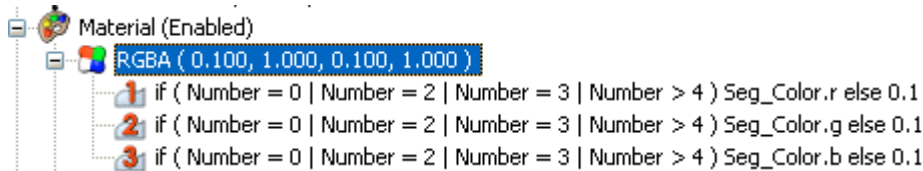
### Segment Formule requise

Seg_A	if ( Number = 0   Number = 2   Number = 3   Number > 4 ) Seg_Color.r else 0.1
Seg_B	if ( Number < 5   Number > 6 ) Seg_Color.r else 0.1
Seg_C	if ( Number < 2   Number > 2 ) Seg_Color.r else 0.1
Seg_D	if ( Number = 0   Number = 2   Number = 3   Number = 5   Number = 6   Number = 8   Number = 9 ) Seg_Color.r else 0.1
Seg_E	if ( Number = 0   Number = 2   Number = 6   Number = 8 ) Seg_Color.r else 0.1
Seg_F	if ( Number = 0   Number = 4   Number = 5   Number = 6   Number = 8   Number = 9 ) Seg_Color.r else 0.1
Seg_G	if ( Number=2   Number=3   Number=4   Number=5   Number=6   Number=8   Number=9 ) Seg_Color.r else 0.1

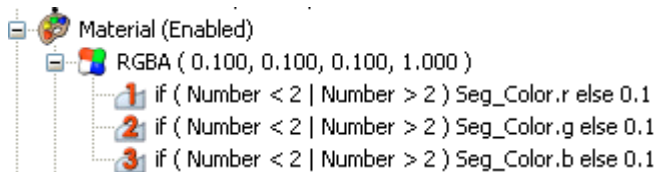


Les formules indiquées ci-avant ne concernent que les paramètres "1" relatifs à la couleur rouge. Elles sont identiques pour les trois paramètres "1", "2" et "3", sauf que l'on utilise `Seg_Color.g` pour le paramètre "2" et `Seg_Color.b` pour le paramètre "3". Voir copie d'écran précédente, ou les deux exemples suivants.

Pour le segment `Seg_A` :

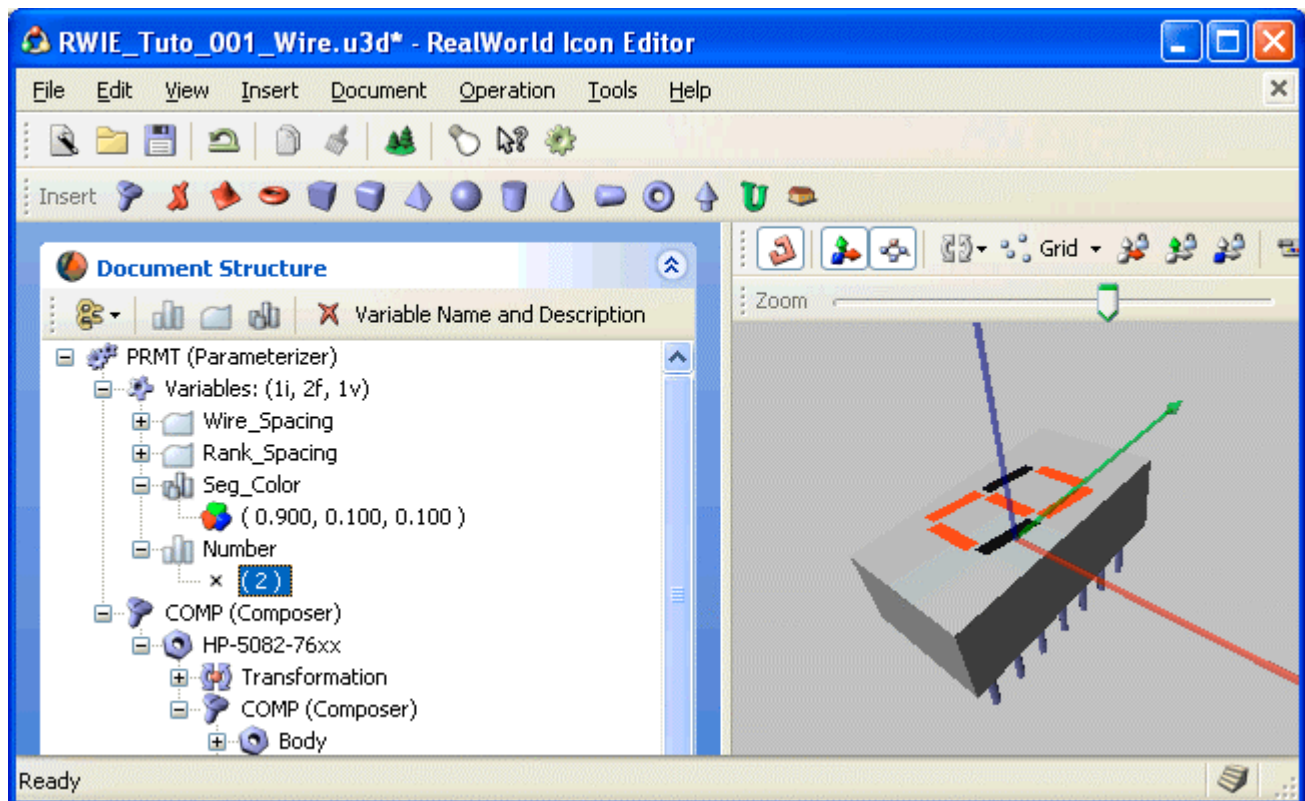


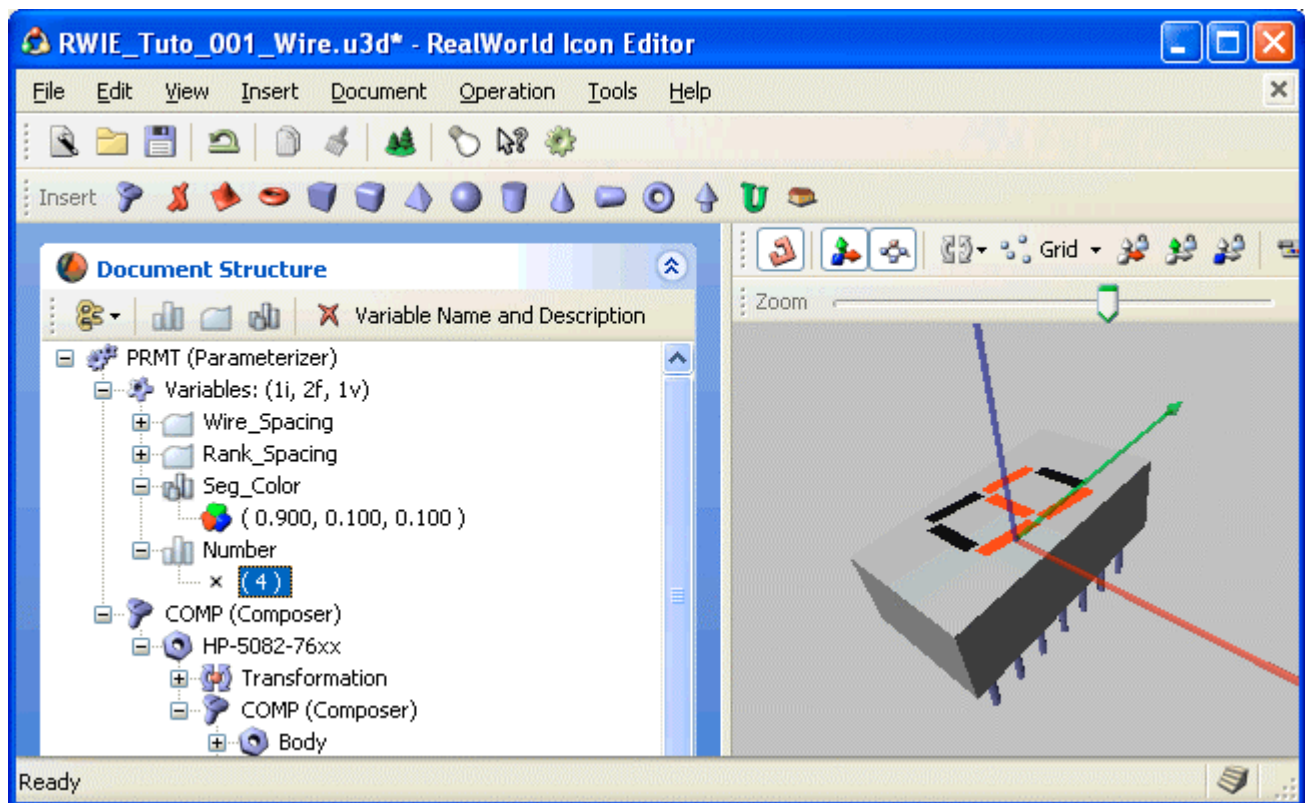
Pour le segment `Seg_C` :



**Important !** Les formules sont sensibles à la casse, les minuscules et majuscules sont importantes et doivent être respectées à tout endroit. Si vous nommez une variable "Number" (avec un N majuscule) dans le container des variables, et qu'ensuite vous y faite référence en la notant "number" (avec un n minuscule) dans une formule, la formule ne sera pas exécutée par qu'elle contiendra un élément considéré comme inexistant.

Et voilà ! Maintenant, la valeur de la variable `Number` se reflète directement sur l'afficheur lui-même.

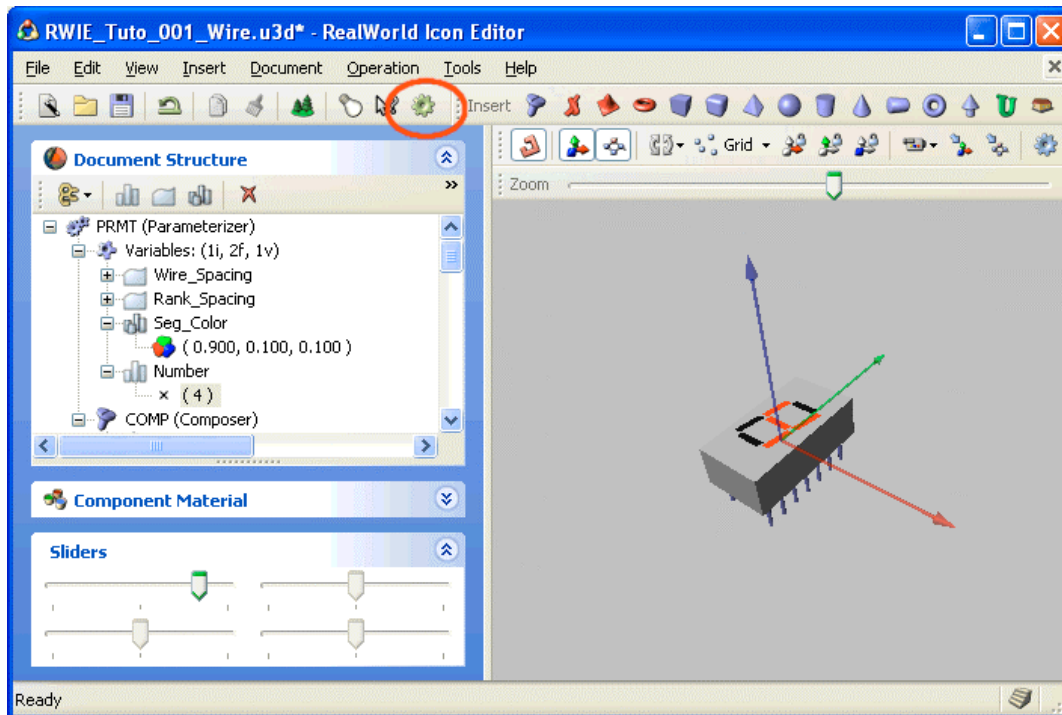




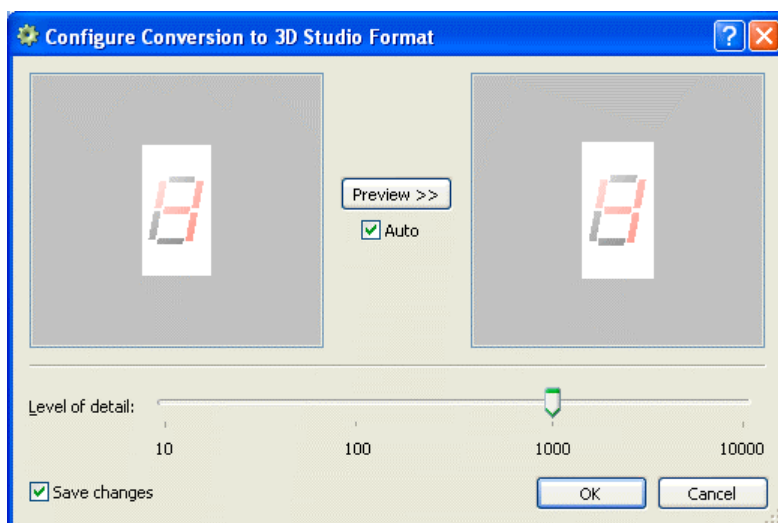
## Export au format 3DS

L'export au format 3DS requiert une version modifiée du logiciel RealWorld Icon Editor. La version actuellement sur le site [rw-designer](http://www.sonelec-musique.com), qui est la version V2006.2, ne permet pas l'export au format 3DS. Avant de pouvoir exporter votre objet au format 3DS, vous devez au préalable suivre la procédure décrite à la page RWIE - Ajout fonction export 3DS, à l'adresse suivante : [http://www.sonelec-musique.com/logiciels/rwie/rwie\\_3ds\\_export.htm](http://www.sonelec-musique.com/logiciels/rwie/rwie_3ds_export.htm)

L'objet vous plaît tel qu'il est et vous souhaitez dès maintenant l'utiliser dans le visualiseur 3D de Proteus / Ares. Il vous suffit de cliquer sur le bouton "Save as 3D Studio file" situé dans la barre d'outil des fichiers.



Une fenêtre apparaît, qui vous permet de spécifier le degré de détail du composant qui sera exporté.



La plupart du temps, une valeur comprise entre 100 et 1000 pour le degré de détail (Level of detail) est suffisant. Pour des objets possédant des arrondis, il vous faudra faire quelques tests pour voir ce qui vous satisfait le mieux.