

Le paysage est créé selon les niveaux de pertinence. Les niveaux sont créés les uns après les autres.

Une fois le paysage affiché, il faut rajouter les liens qui sont des objets 3D quelconques et le texte. L’affichage du texte est traité dans la section 2.2.3 page 30.

## Le picking

Le terme français traduisant littéralement ”picking” est *cueillette*. En effet, le picking consiste à indiquer où se trouve le pointeur de la souris dans le monde en 3 dimensions.

La première méthode qui est décrite dans [WOO 2000 et coll.] consiste à trouver les coordonnées de la souris à l’écran, inverser la matrice de transformation de la perspective pour avoir les coordonnées dans le monde 3D. Ensuite il faut calculer le point d’intersection avec le plan *near* et ensuite avec le plan *far*. Ainsi, on connaît le point projeté le plus près et celui le plus loin. Il faut ensuite calculer l’intersection de cette droite avec la première face qui la coupe.

Cette méthode demande énormément de calculs (une inversion de matrice, une intersection droite/plan). De plus elle oblige à stocker toutes les faces pour savoir par laquelle le rayon est passé. Cette méthode n’est donc acceptable que pour une petite scène avec très peu d’objets.

En outre cette méthode ne permet pas de créer des animations lors ce que l’on a trouvé un objet. Les calculs se feraient sur l’objet en mouvement.

Une autre méthode consiste à dessiner dans le backbuffer\*, une image avec autant de couleurs que d’objets à cliquer. Cette méthode est initiée dans [WOO 2000 et coll.] et mise en œuvre sur le site [www.lighthouse3d.com](http://www.lighthouse3d.com).

Ici, on ne va pas projeter les coordonnées de la souris dans le le monde 3D mais on va lire les coordonnées de la souris et regarder de quelle couleur est le pixel qui se trouve derrière.



FIG. 2.12 – Exemple d’utilisation de la méthode de [www.lighthouse3d.com](http://www.lighthouse3d.com)

Cette méthode ne permet toujours pas de créer d’animation avec l’objet trouvé. En effet, il est impossible d’utiliser le backbuffer car il stocke l’image pour le picking. Il est donc impossible de créer des animations. Effectivement, pour obtenir des animations, le backbuffer sert à dessiner la scène, puis on inverse avec le frontbuffer\* qui est en fait

l'image à l'écran. Dessiner la scène en premier lieu dans le backbuffer puis l'afficher à l'écran évite les effets de clignotements.

Qt propose une méthode qui permet de capturer le contenu d'un Widget OpenGL. Ainsi au lieu d'écrire dans le backbuffer, il suffit de dessiner dans le widget normalement, puis de regarder dans l'image ainsi capturée. Cette méthode est aussi rapide que la précédente, mais permet en plus de préserver le backbuffer ce qui permet de créer des animations pour mettre en valeur l'objet pointé.

La création de l'image de picking est quasiment identique à la création de l'image normale. L'image générée est une image en niveau de gris ainsi il suffit de tester la première composante du pixel (le rouge) pour connaître l'objet pointé. Cette méthode permet d'avoir 255 objets à l'écran. Si toute la gamme des couleurs est utilisée, ce sont plus de 16 millions d'objets qui peuvent être pointés.

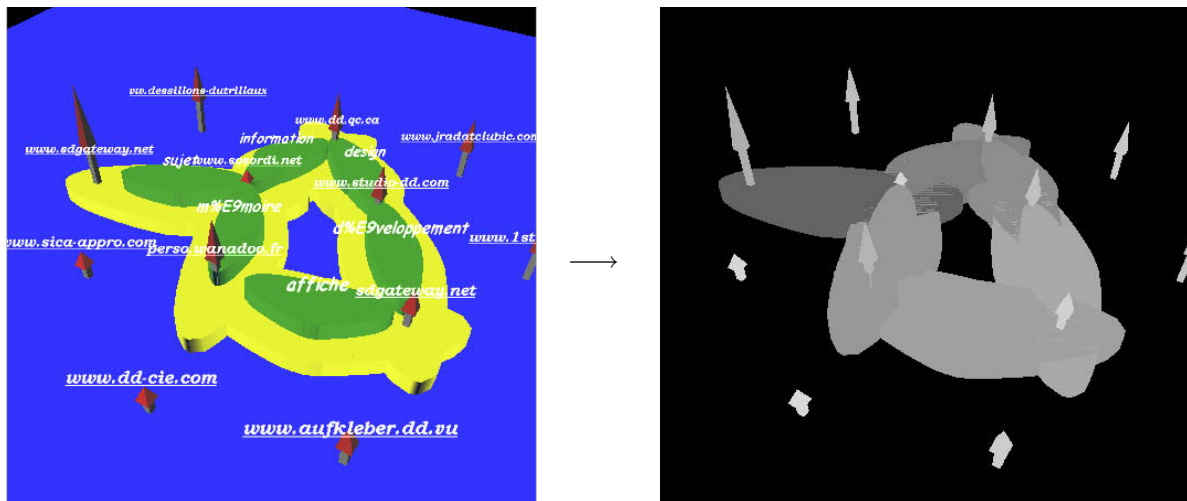


FIG. 2.13 – Image de picking générée par une recherche

En fait, ce sont d'abord les thèmes qui sont dessinés mais par ordre de pertinence décroissante. Comme cela, les thèmes ayant beaucoup de pertinence sont plus présents dans les hauts niveaux que dans les bas niveaux de la carte. Ensuite ce sont les sites qui sont dessinés.

Chaque objet (thème ou site) dessiné possède un numéro qui est en fait le niveau de gris correspondant. La couleur noire (donc le niveau de gris 0) ne correspond à aucun objet.

### Mise en évidence

Une fois qu'on a trouvé sur quel objet se trouve le pointeur de la souris, il suffit de faire la correspondance entre la couleur obtenue et les objets.

Si c'est un thème qui est trouvé alors ce site et les liens qui sont concernés sont mis en surbrillance. Sinon c'est le lien et les thèmes mis en relation qui sont mis en évidence.