

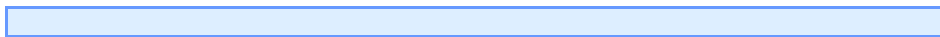
Chapitre V : Transparence



par [Loka](#)

Date de publication : 30/03/2006

Dernière mise à jour : 17/04/2006



V - Transparence

V - Transparence

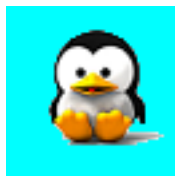
La surface SDL possède un élément nommé color key.

Le color key est la couleur que vous ne souhaitez pas afficher lorsque vous blittez une surface.

Ce tutorial a pour but de vous montrer comment s'en servir.

Nous souhaitons par exemple mettre une image d'un personnage sur un paysage.

Prenons ce mignon petit tux assis comme personnage :



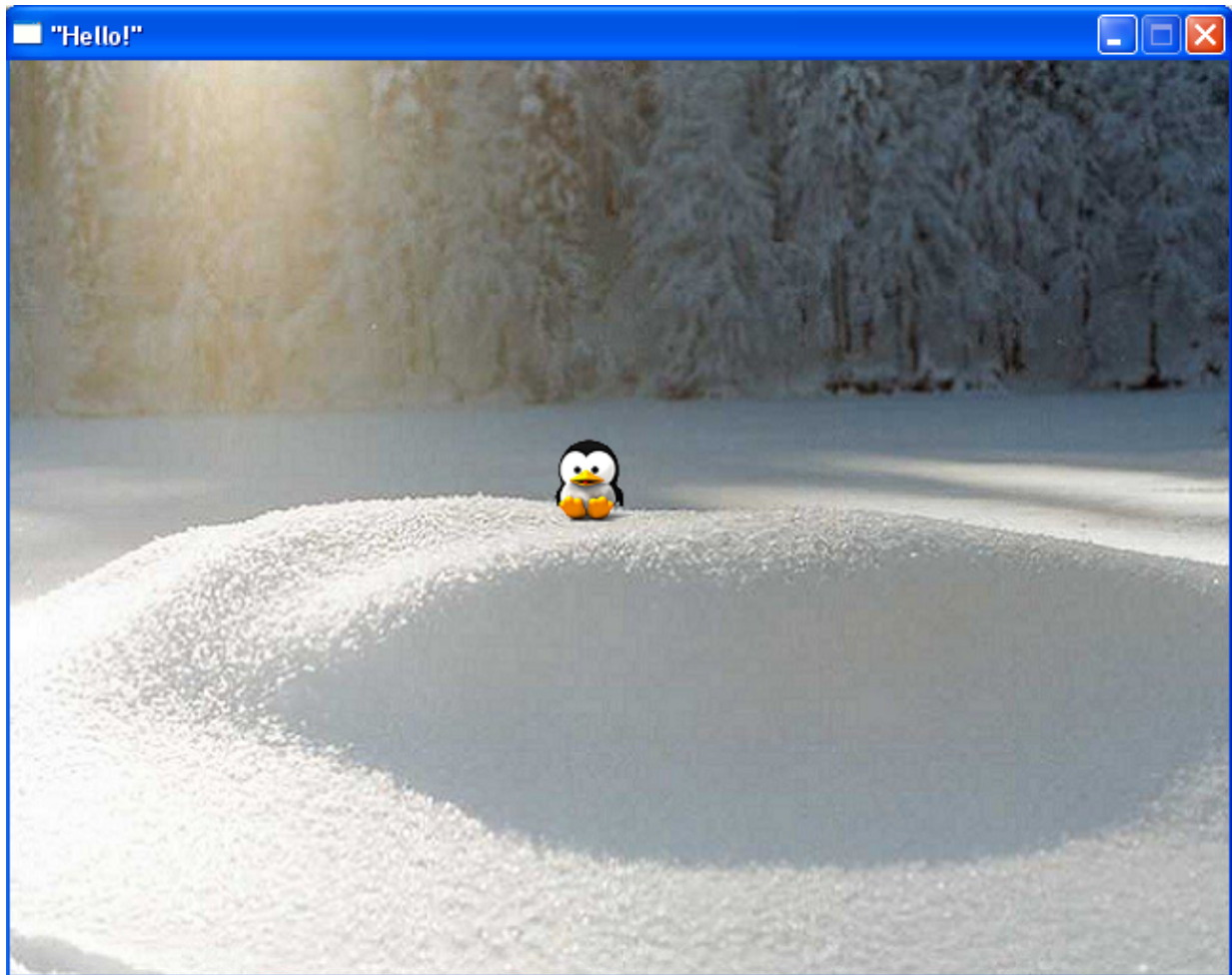
Ce tux, nous souhaitons le poser dans ce paysage :



Comme vous pouvez le voir, la couleur de fond du tux va nous gêner.

Ce que nous souhaitons obtenir n'est pas l'affichage du tux tel quel avec son fond cyan sur le paysage, mais seulement le tux sans son fond.

Et donc pour cela il nous faut rendre le fond du tux transparent afin d'obtenir comme résultat :



Il nous faut donc dire que la couleur qu'on ne souhaite pas voir pour l'image du tux est le cyan (soit Red = 0, Green = FF et Blue = FF) afin qu'il soit le color key.

Le color key est typiquement mis en place lors du chargement de l'image.

Load_image

```
SDL_Surface *load_image( std::string filename )
{
    //L'image qui est chargée
    SDL_Surface* loadedImage = NULL;

    //L'image optimisée que nous utiliserons par la suite
    SDL_Surface* optimizedImage = NULL;

    //Chargement de l'image
    loadedImage = IMG_Load( filename.c_str() );
```

Load_image

```
//Si l'image est chargée
if( loadedImage != NULL )
{
    //Création de l'image optimisée
    optimizedImage = SDL_DisplayFormat( loadedImage );

    //Libération de l'ancienne image
    SDL_FreeSurface( loadedImage );
}
```

Voici donc la fonction de chargement d'images, rien de changé par rapport à avant, ce qui est intéressant est ce qui va suivre.

SDL_MapRGB

```
//Si l'image optimisée créée est bonne
if( optimizedImage != NULL )
{
    Uint32 colorkey = SDL_MapRGB( optimizedImage->format, 0, 0xFF, 0xFF );
}
```

Après avoir vérifié si l'image a bien été optimisée, nous définissons la couleur dont on souhaite qu'elle devienne le color key (et donc la couleur transparente).

Nous appelons `SDL_MapRGB()` pour prendre les valeurs de rouge (Red), de vert (Green) et de bleu (Blue) et il nous retourne la valeur du pixel dans le même format que la surface.

Maintenant il est temps d'utiliser cette valeur obtenue afin de rendre transparente la partie qu'on souhaite rendre transparente de l'image (ici le fond).

SDL_SetColorKey

```
//On met tous les pixels de couleur R 0, G 0xFF, B 0xFF transparent
SDL_SetColorKey( optimizedImage, SDL_RLEACCEL | SDL_SRCCOLORKEY, colorkey );
```

Nous utilisons pour cela la fonction **SDL_SetColorKey()**.

Le premier argument est le pointeur vers la surface que nous allons utiliser pour le color key.

Le second argument englobe les *flags* utilisés. Ici nous choisissons `SDL_RLEACCEL` qui permet l'accélération RLE qui nous permet d'être sûr lorsque nous blitons notre surface, il n'y aura pas de ralentissements.

Le flag `SDL_SRCCOLORKEY` lui nous permet d'être sûr que nous utilisons bien le color key quand on blitte une surface sur une autre.

Le troisième et dernier argument est la couleur que nous posons comme étant le color key.

Comme vous pouvez le voir c'est la couleur mappée juste avant.

Ensuite il nous reste plus qu'à retourner l'image optimisée obtenue.

return

```
//On retourne l'image optimisée
return optimizedImage;
}
```

Maintenant que nous avons obtenu, normalement, notre surface avec le color key défini, il nous reste plus qu'à poser nos deux surfaces :

le fond en premier puis notre surface avec le color key par dessus.

apply_surface

```
//Application des surfaces sur l'écran
apply_surface( 0, 0, background, screen );
apply_surface( 270, 190, tux, screen );

//Mise à jour de l'écran
if( SDL_Flip( screen ) == -1 )
{
    return 1;
}
```

Vous devriez ainsi obtenir le résultat voulu au début, sans le fond cyan du tux pour gêner.

[Version pdf](#) 

[Télécharger les sources](#)

[<= Retour au chapitre IV : Gestion des événements](#)

[-=Index Tutos SDL=-](#)

[=> Accéder au chapitre VI : Feuille de Sprites](#)