

```

1 #include <stdlib.h>
2 #include <SDL.h>
3 int main ( int argc, char** argv )
4 {
5     SDL_Surface* fen, *imal;
6     fen = SDL_SetVideoMode(640, 480, 16, SDL_HWSURFACE|SDL_DOUBLEBUF);
7     imal = SDL_LoadBMP("cb.bmp");
8
9     SDL_Rect pos1;
10    pos1.x = (fen->w - imal->w)/2;
11    pos1.y = (fen->h - imal->h)/2;
12
13    bool done = false;
14    while (!done)
15    {
16        SDL_Event event;
17        while (SDL_PollEvent(&event))
18        {
19            switch (event.type)
20            {
21                case SDL_QUIT:
22                    done = true;
23                    break;
24                case SDL_KEYDOWN:
25                    {
26                        if (event.key.keysym.sym == SDLK_ESCAPE)
27                            done = true;
28                        break;
29                    }
30            }
31        }
32        SDL_FillRect(fen, 0, SDL_MapRGB(fen->format, 0, 0,0));
33        SDL_BlitSurface(imal, 0, fen, &pos1);
34        SDL_Flip(fen);
35    }
36    SDL_FreeSurface(imal);
37    return 0;
38 }

```

Pour gérer le clavier voici quelques définitions **SDL Keysym**

(toutes les définitions ici : <https://www.libsdl.org/release/SDL-1.2.15/docs/html/sdlkey.html>)

SDLKey	ASCII value	Common name
SDLK_BACKSPACE	'\b'	backspace
SDLK_TAB	'\t'	tab
SDLK_RETURN	'\r'	return
SDLK_ESCAPE	'^['	escape
SDLK_SPACE	' '	space
SDLK_0	'0'	0
SDLK_1	'1'	1
SDLK_2	'2'	2
...		
SDLK_9	'9'	9
SDLK_a	'a'	a
SDLK_b	'b'	b
SDLK_c	'c'	c
...		
SDLK_z	'z'	z

Pour chaque question indiquer les modifications à apporter au code source fourni pour obtenir les résultats demandés. Les lignes du code source sont numérotées pour faciliter les réponses.

0/ Modifier la taille de la fenêtre de l'application : taille = .900x600

Réponse : Remplacer la ligne n° 6 par
`fen = SDL_SetVideoMode(900, 600, 16,SDL_HWSURFACE|SDL_DOUBLEBUF);`

1/ Afficher l'image cb.bmp sur fond blanc.

Réponse : Remplacer la ligne n° 32 par
`SDL_FillRect(fen, 0, SDL_MapRGB(fen->format, 255, 255, 255));`

2/ Quitter le programme avec la touche S du clavier au lieu de Escape.

Réponse : Remplacer la ligne n° 26 par
`if (event.key.keysym.sym == SDLK_s)`

3/ Afficher l'image cb.bmp comme indiqué

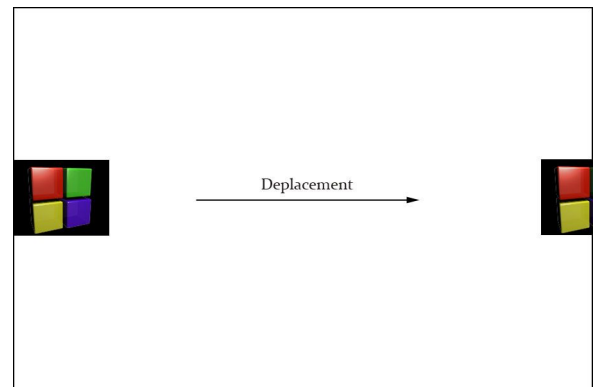
Réponse: Remplacer la ligne n° 10 par: `pos1.x = (fen->w - im1->w);`



4/ Afficher l'image cb.bmp, effectuer le déplacement et l'image s'arrête comme indiqué

Réponse: Remplacer la ligne n° 10 par: `pos1.x = 0;`

Insérer après L n°34 : `if(pos1.x < fen->w -(im1->w)/2)`
`pos1.x++;`



5/ Afficher plusieurs fois l'image cb.bmp comme indiqué .Le fond noir n'est pas affiché.

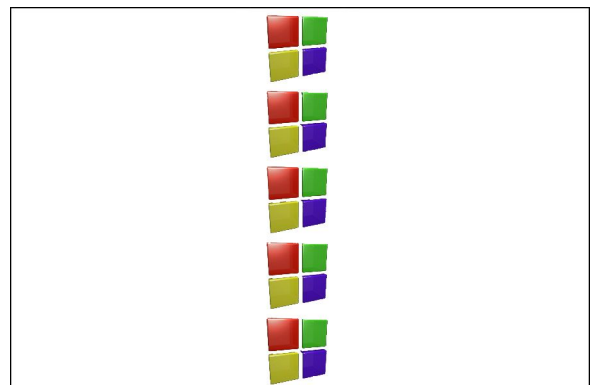
Réponse:

Insérer après L n°12 : `SDL_SetColorKey(im1, SDL_SRCCOLORKEY,SDL_MapRGB(im1->format,0,0,0));`

Remplacer L n°33 par:

```
pos1.y = 0;
SDL_BlitSurface(im1, 0, fen, &pos1);
pos1.y = im1->h;
SDL_BlitSurface(im1, 0, fen, &pos1);
pos1.y = 2*im1->h;
SDL_BlitSurface(im1, 0, fen, &pos1);
pos1.y = 3*im1->h;
SDL_BlitSurface(im1, 0, fen, &pos1);
pos1.y = 4*im1->h;
SDL_BlitSurface(im1, 0, fen, &pos1);
```

6/Utiliser une boucle for pour la question 5



7/ Remplacer l'image cb.bmp par horse_1.bmp (jointe au mail).

8/ Supprimer la couleur de fond de l'image Horse_1.bmp (la couleur de fond est R: 7, V: 159, B: 143).

9/Utiliser l'image Horse_D.bmp pour simuler le déplacement du cheval en utilisant la séquence inscrite dans l'image (sprite).