

Accueil (.././fr/) > Forum (../) > Programmation (<https://openclassrooms.com/forum/categorie/programmation>) > Langage C (<https://openclassrooms.com/forum/categorie/langage-c>) > [EXERCICES]Langage C, en SDL

Liste des forums
(/forum/)

[EXERCICES]Langage C, en SDL (/forum/sujet/exercices-langage-c-en-sdl-91300)

Des consignes précises

Partage



Sujet résolu

Xavierh
(/fr/memb
ers/kx9sbf
t3g945).



(/fr/members/kx9sbft3g945).



11 août 2008 à 10:10:41

J'ai récemment ouvert un topic sur des exercices en C. Mais les TP proposés étaient presque tous des projets. C'est pourquoi j'ouvre ce topic avec des règles précises.

Ne postez pas d'exercices sans avoir lu les règles qui suivent!

Sur quoi portent les exercices

Sur ce topic, les exercices sont uniquement des exercices sur le langage C utilisant la SDL. Pas de C++ ici !

Ces exercices ne seront pas "gros", c'est à dire que ce ne seront pas des projets mais des simples TP (du même genre que les TP fournis par M@theo21 dans son tutoriel sur le langage C.

Quelles sont les règles

Tout d'abord, ce topic n'est pas un fourre tout!

Si vous voulez poster un TP, il faut que celui ci soit parfaitement expliqué et qu'il suive une présentation précise (ci dessous) :

```
1 <titre1>Titre du TP</titre1>
2
3 <titre2><souligne>Quand</souligne>: </titre2> Lieu où ci
4 <titre2><souligne>Sujet principal</souligne></titre2> :
5 <titre2><souligne>Consignes</souligne></titre2>: Consig
6 <titre2><souligne>Notions travaillées</souligne></titre2>:
7 <titre2>Réponse</titre2> :
8 <secret>Les réponses sont à placées dans une balise "se
9 <titre2><souligne>Améliorations possibles</souligne></t:
```

Sur la ligne Quand, vous devez mettre ce TP ce situe à quel chapitre du tutoriel sur le langage C.

Les repères:

1. Avant l'apprentissage de SDL_image
2. Avant la gestion des événements
3. Avant le Mario Sokoban (avec la gestion des événements)
4. Après le Mario Sokoban (maîtrise des collisions etc...)
5. Après la gestion du temps
6. Après la gestion du temps et de SDL_ttf
7. Tous chapitres

Si votre TP possède des images hors du tutoriel de M@teo, il est préférable d'envoyer un zip du dossier. S'il requiert des images fournies par m@teo, il serait sympas de mettre un lien vers la page du tutoriel qui les met à téléchargement :)

Les réponses

Si vous postez un TP, vous êtes obligés de donner la solution au problème. Celle ci doit obligatoirement se situer dans une balises **secret**.

Le code doit avoir été testé.

N'utilisez aucune bibliothèque non travaillée avec M@teo21 lors de son tutoriel.

Si votre TP suit bien ces règles, alors il sera mentionné dans le tableau ci dessous.

Exercices

Exercices/TP SDL

Titre du TP	Notions abordées	Envoyé par
<u>SDL Button</u> (http://www.siteduzero.com/forum-83-308326-2842727-exercices-langage-c-en-sdl.html#r2842727).	maîtrise du C (niveau correct)/SDL/Pointeurs de fonction et méthodologie	<u>gogohan</u> (http://www.siteduzero.com/forum-83-308326-2842727-exercices-langage-c-en-sdl.html#r2842727).
<u>Je dessine</u> (http://www.siteduzero.com/forum-83-308326-2842802-exercices-langage-c-en-sdl.html#r2842802).	événements, conditions, surfaces et pointeurs	<u>xavie</u> (http://www.siteduzero.com/forum-83-308326-2842802-exercices-langage-c-en-sdl.html#r2842802).
<u>Coordonnées de la souris</u> (http://www.siteduzero.com/forum-83-308326-2849988-exercices-langage-c-en-sdl.html#r2849988).	Utilisation des événements en SDL et de SDL_ttf	<u>askaf</u> (http://www.siteduzero.com/forum-83-308326-2849988-exercices-langage-c-en-sdl.html#r2849988).
<u>Bal littéraire</u> (http://www.siteduzero.com/forum-83-308326-2878577-exercices-langage-c-en-sdl.html#r2878577).	Timers, tableaux de char, affichage de texte	<u>dann</u> (http://www.siteduzero.com/forum-83-308326-2878577-exercices-langage-c-en-sdl.html#r2878577).

Si vous avez besoin d'aide à propos des exercices ou que vous voulez signaler une erreur ou même demander de l'aide à propos des améliorations, n'hésitez pas à poster votre message. Les TP sont faits pour vous entraîner!

gogohan
(/fr/members/zwdkjr5j8tc3)



([/fr/members/zwdkjr5j8tc3](http://www.siteduzero.com/forum-83-308326-2878577-exercices-langage-c-en-sdl.html#r2878577)).



SDL button

Quand:

Partie 3 du tuto de C -> maîtrise de la SDL requise

Sujet principal

: Création de boutons personnalisés

Consignes

: Réalisez une structure nommée `SDL_Button`, ainsi que des fonctions afin de s'occuper des boutons. Je n'impose rien, à vous de chercher.

Notions travaillées

: maîtrise du C (niveau correct)/SDL/Pointeurs de fonction et méthodologie 🤔

Aide

:

Pensez à mettre un pointeur de fonction dans la structure `SDL_Button`.

exemple : (structure incomplète 🤔, à vous de chercher quoi rajouter)

```
1 | typedef struct _SDL_Button{
2 |
3 |     SDL_Rect pos;
4 |     SDL_Surface *surface_normale; /* On n'a pas le curseur :
5 |
6 |     void *(*fonction)(void); /* le pointeur de fonction, rei
7 |
8 | }SDL_Button;
```

Réponse

:

Bon, j'ai retrouvé ma version de la lib.

A ne lire qu'une fois votre lib entamée, ou une fois finie (ne lisez pas mes sources si vous n'avez aucun problème 🤔).

Télécharger les sources (<https://www.mediafire.com/?sharekey=8eddc5ccd5dcaed2d2db6fb9a8902bda>).

Améliorations possibles

: Personnalisez le bouton en y mettant du texte (SDL_ttf 🤖), ou encore en chargeant une image 🤖

Bon courage à tous 🤖

Edit: Je cherche ma correction 🤖 J'éditerai quand j'aurai retrouvé mon code 🤖

Edit 2: Fait 🤖

Note: Pour mon code, j'ai créé une fonction pour écrire du texte plus rapidement sur une surface, mais je n'ai pas optimisé le code donc il est assez lourd si vous devez créer 30-40 boutons (ouverture et fermeture 30 fois des polices 🤖). Aussi, je ne stocke pas la position du bouton dans ma structure car on n'a pas accès aux membres de la structure 🤖

Xavierh
(/fr/memb
ers/kx9sbf
t3g945).



[\(/fr/members/kx9sbf3g945\).](#)



[11 août 2008 à 11:27:06](#)

Je dessine

Quand:

Avant Mario Sokoban (maîtrise de la gestion des événements requise).

Sujet principal

: Dans ce TP, vous allez faire un petit jeu où l'utilisateur pourra dessiner.

Consignes

:

Ce TP vous fera réaliser un programme dans lequel l'utilisateur pourra dessiner sur l'écran. Il dessinera des petits carrés de 3 pixel sur 3 pixel mais attention : il faut que l'on puisse modifier la couleur du

carré.

Voici les couleurs associées à la touche:

- Touche 1: rgb = 0, 0, 0
- 2 : rgb = 255, 0, 0
- 3 : rgb = 0, 255, 0
- 4 : rgb = 0, 0, 255
- 5 : rgb = 255, 255, 0
- 6 : rgb = 255, 134, 0
- 7 : rgb = 100, 50, 0
- 8 : rgb = 255, 91, 173
- 9 : rgb = 53, 154, 255
- 0 : rgb = 255, 255, 255

En plus il faut que lorsque l'on fait flèche du haut, la taille du crayon (le carré) grossisse et qu'elle diminue lorsque l'on fait flèche du bas. Allez, Au boulot!

Notions travaillées

: évènements, conditions, surfaces et pointeurs

Réponse

:

Secret ([cliquez pour afficher](#)).

Solution:

Je récapitulerais le code à la fin...

Tout d'abord on inclus SDL.h

```
1 | #include <SDL/SDL.h>
```

Ensuite:

```
1 | SDL_Surface *ecran = NULL, *SurfaceColler, *pCouleur;
2 |     SDL_Event event;
3 |     int continuer = 1;
4 |     long t = 3;
5 |     long r;
6 |     long g;
7 |     long b;
8 |     SDL_Rect position;
9 |
10 |     position.x = 0;
11 |     position.y = 0;
```

Explication :

*ecran designe l'écran ^^.

SurfaceColler est un pointeur qui designe quelle surface va être collée. Dans ce corrigé, il est inutile. Pourquoi l'avoir mis? Je vous expliquerai plus tard, lors des améliorations possibles.

pCouleur c'est la surface que l'utilisateur collera pour dessiner.

continuer est la variable qui, tant qu'elle est vraie, fera une boucle infinie.

La suite, plus tard.

Ensuite, on initialise la SDL et on ouvre une fenêtre avec un fond blanc.

```
1 | SDL_Init(SDL_INIT_VIDEO);
2 |
3 |     ecran = SDL_SetVideoMode(640, 480, 32, SDL_HWSURFACE);
4 |     SDL_FillRect(ecran, NULL, SDL_MapRGB(ecran->format,
5 |     SDL_WM_SetCaption("Je dessine", NULL);
```

Puis on crée pCouleur qui n'est que un petit carré. Mais attention, pas n'importe comment. De sorte à ce que l'on puissent plus tard modifier sa taille et sa couleur. On va donc mettre la variable "t" pour sa taille, et "r", "g", "b" pour sa couleur.

```
1 | r = 0; g = 0; b = 0;
2 |     pCouleur = SDL_CreateRGBSurface(SDL_HWSURFACE, t, t,
3 |     SDL_FillRect(pCouleur, NULL, SDL_MapRGB(ecran->format,
```

Comme je vous l'ai dit plus tôt, la surface que l'utilisateur dessinera est SurfaceColler qui va pointer sur pCouleur.

```
1 | SurfaceColler = pCouleur;
```

Ensuite on met l'écran à jour et on active la répétition de touches.

```
1 | SDL_Flip(ecran);
2 |
3 |     SDL_EnableKeyRepeat(10, 10);
```

Maintenant vient la boucle des événements. Il faut ici dire que lorsque l'on tappe sur 0, 1, 2, 3, 4, ..., 9, alors pCouleur change de couleur et que lorsque l'on fait flèche du haut, pCouleur grossi (la valeur de "t" augmente) et que lorsque l'on fait flèche du bas, pCouleur diminue.

```
1  while (continuer)
2      {
3          SDL_WaitEvent(&event);
4          switch(event.type)
5          {
6              case SDL_QUIT://Arrêt du programme
7                  continuer = 0;
8                  break;
9
10             case SDL_KEYDOWN:
11
12                 switch(event.key.keysym.sym)
13                 {
14                     case SDLK_ESCAPE://Effacement
15                         position.x = 0;
16                         position.y = 0;
17                         SDL_FillRect(ecran, NULL,
18                         break;
19
20                     case SDLK_1://couleur rgb de |
21                         r = 0;
22                         g = 0;
23                         b = 0;
24                         SDL_FillRect(pCouleur, NU
25                         SurfaceColler = pCouleur;
26                         break;
27
28                     case SDLK_2://couleur rgb de |
29                         r = 255;
30                         g = 0;
31                         b = 0;
32                         SDL_FillRect(pCouleur, NU
33                         SurfaceColler = pCouleur;
34                         break;
35
36                     case SDLK_3://0.255.0
37                         r = 0;
38                         g = 255;
39                         b = 0;
40                         SDL_FillRect(pCouleur, NU
41                         SurfaceColler = pCouleur;
42                         break;
43
44                     case SDLK_4://0.0.255
45                         r = 0;
46                         g = 0;
47                         b = 255;
48                         SDL_FillRect(pCouleur, NU
49                         SurfaceColler = pCouleur;
50                         break;
51
52                     case SDLK_5://255.255.0
53                         r = 255;
54                         g = 255;
55                         b = 0;
56                         SDL_FillRect(pCouleur, NU
57                         SurfaceColler = pCouleur;
58                         break;
59
60                     case SDLK_6://255.134.0
```



```

        r = 255;
        g = 134;
        b = 0;
        SDL_FillRect(pCouleur, NU
        SurfaceColler = pCouleur;
break;

case SDLK_7://100.50.0
    r = 100;
    g = 50;
    b = 0;
    SDL_FillRect(pCouleur, NU
    SurfaceColler = pCouleur;
break;

case SDLK_8://255.91.173
    r = 255;
    g = 91;
    b = 173;
    SDL_FillRect(pCouleur, NU
    SurfaceColler = pCouleur;
break;

case SDLK_9://53.154.255
    r = 53;
    g = 154;
    b = 255;
    SDL_FillRect(pCouleur, NU
    SurfaceColler = pCouleur;
break;

case SDLK_0://255.255.255
    r = 255;
    g = 255;
    b = 255;
    SDL_FillRect(pCouleur, NU
    SurfaceColler = pCouleur;
break;

case SDLK_UP://taille de pCou
    t++;
    switch(t)
    {
        case 479:
            t = 478;
            break;
    }
    pCouleur = SDL_CreateRGBSi
    SDL_FillRect(pCouleur, NU
    SurfaceColler = pCouleur;
break;

case SDLK_DOWN: //moins grossi
    t--;
    switch(t)
    {
        case 0:
            t = 1;
            break;
    }
    pCouleur = SDL_CreateRGBSi
    SDL_FillRect(pCouleur, NU
    SurfaceColler = pCouleur;

break;

```

```

    }

```

```

129
130         break;
131
132     case SDL_MOUSEMOTION:
133
134         switch(event.button.button)
135         {
136             case SDL_BUTTON_LEFT:
137
138                 position.x = event.motion
139                 position.y = event.motion
140                 SDL_BlitSurface(SurfaceCo.
141                 break;
142         }
143     break;
144
145     break;
146
147     case SDL_MOUSEBUTTONDOWN:
148         switch(event.button.button)
149         {
150             case SDL_BUTTON_LEFT:
151                 position.x = event.motion
152                 position.y = event.motion
153                 SDL_BlitSurface(SurfaceCo.
154                 break;
155         }
156     break;
157 }
158
159 SDL_Flip(ecran);
160 }

```

Vous avez remarqué que les valeurs de r g b change ainsi que t.
D'ailleurs, j'ai mis que lorsque t atteignait 0, celui-ci était = 1, comme
ça, pas de t en dessous de 0 (si "t" à une valeur en dessous de 0, le
programme plante, essayez!). Et de même lorsqu'il arrive à 479.

Ensuite on quitte :

```

1 | SDL_FreeSurface(pCouleur);
2 | SDL_Quit();

```

Je récapitul le tout :

```

1 | #include <SDL/SDL.h>
2 |
3 | int main(int argc, char *argv[])
4 | {
5 |     SDL_Surface *ecran = NULL, *SurfaceColler;
6 |     SDL_Surface *pCouleur;
7 |     SDL_Event event;
8 |     int continuer = 1;
9 |     long t = 3;
10 |    long r;

```

```

11 long g;
12 long b;
13 SDL_Rect position;
14
15 position.x = 0;
16 position.y = 0;
17
18 SDL_Init(SDL_INIT_VIDEO);
19
20 ecran = SDL_SetVideoMode(640, 480, 32, SDL_HWSURF,
21 SDL_FillRect(ecran, NULL, SDL_MapRGB(ecran->format, 0, 0, 0));
22 SDL_WM_SetCaption("Je dessine", NULL);
23
24 r = 0; g = 0; b = 0;
25 pCouleur = SDL_CreateRGBSurface(SDL_HWSURFACE, t,
26 SDL_FillRect(pCouleur, NULL, SDL_MapRGB(ecran->format, 0, 0, 0));
27
28 SurfaceColler = pCouleur;
29
30 SDL_Flip(ecran);
31
32 SDL_EnableKeyRepeat(10, 10);
33
34 while (continuer)
35 {
36     SDL_WaitEvent(&event);
37     switch(event.type)
38     {
39         case SDL_QUIT://Arrêt du programme
40             continuer = 0;
41             break;
42
43         case SDL_KEYDOWN:
44
45             switch(event.key.keysym.sym)
46             {
47                 case SDLK_ESCAPE://Effacement
48                     position.x = 0;
49                     position.y = 0;
50                     SDL_FillRect(ecran, NULL,
51 SDL_MapRGB(ecran->format, 0, 0, 0));
52                     break;
53
54                 case SDLK_1://couleur rgb de |
55                     r = 0;
56                     g = 0;
57                     b = 0;
58                     SDL_FillRect(pCouleur, NU
59 SDL_MapRGB(pCouleur->format, 0, 0, 0));
60                     SurfaceColler = pCouleur;
61                     break;
62
63                 case SDLK_2://couleur rgb de |
64                     r = 255;
65                     g = 0;
66                     b = 0;
67                     SDL_FillRect(pCouleur, NU
68 SDL_MapRGB(pCouleur->format, 255, 0, 0));
69                     SurfaceColler = pCouleur;
70                     break;
71
72                 case SDLK_3://0.255.0
73                     r = 0;
74                     g = 255;
75                     b = 0;
76                     SDL_FillRect(pCouleur, NU
77 SDL_MapRGB(pCouleur->format, 0, 255, 0));
78                     SurfaceColler = pCouleur;
79                     break;
80
81                 case SDLK_4://0.0.255
82                     r = 0;
83                     g = 0;
84                     b = 255;
85                     SDL_FillRect(pCouleur, NU
86 SDL_MapRGB(pCouleur->format, 0, 0, 255));
87                     SurfaceColler = pCouleur;
88                     break;
89
90             }
91     }
92 }

```

```

79         g = 0;
80         b = 255;
81         SDL_FillRect(pCouleur, NU
82         SurfaceColler = pCouleur;
83     break;
84
85     case SDLK_5://255.255.0
86         r = 255;
87         g = 255;
88         b = 0;
89         SDL_FillRect(pCouleur, NU
90         SurfaceColler = pCouleur;
91     break;
92
93     case SDLK_6://255.134.0
94         r = 255;
95         g = 134;
96         b = 0;
97         SDL_FillRect(pCouleur, NU
98         SurfaceColler = pCouleur;
99     break;
100
101     case SDLK_7://100.50.0
102         r = 100;
103         g = 50;
104         b = 0;
105         SDL_FillRect(pCouleur, NU
106         SurfaceColler = pCouleur;
107     break;
108
109     case SDLK_8://255.91.173
110         r = 255;
111         g = 91;
112         b = 173;
113         SDL_FillRect(pCouleur, NU
114         SurfaceColler = pCouleur;
115     break;
116
117     case SDLK_9://53.154.255
118         r = 53;
119         g = 154;
120         b = 255;
121         SDL_FillRect(pCouleur, NU
122         SurfaceColler = pCouleur;
123     break;
124
125     case SDLK_0://255.255.255
126         r = 255;
127         g = 255;
128         b = 255;
129         SDL_FillRect(pCouleur, NU
130         SurfaceColler = pCouleur;
131     break;
132
133     case SDLK_UP://taille de pCou
134         t++;
135         switch(t)
136         {
137             case 479:
138                 t = 478;
139                 break;
140         }
141         pCouleur = SDL_CreateRGBSi
142         SDL_FillRect(pCouleur, NU
143         SurfaceColler = pCouleur;
144     break;
145
146     case SDLK_DOWN: //moins grossi

```

```

147         t--;
148         switch(t)
149         {
150             case 0:
151                 t = 1;
152                 break;
153         }
154         pCouleur = SDL_CreateRGBSurface(
155             SDL_FillRect(pCouleur, NULL,
156             SurfaceColler = pCouleur;
157
158         break;
159
160
161     }
162
163     break;
164
165     case SDL_MOUSEMOTION:
166
167         switch(event.button.button)
168         {
169             case SDL_BUTTON_LEFT:
170
171                 position.x = event.motion.x;
172                 position.y = event.motion.y;
173                 SDL_BlitSurface(SurfaceColler, NULL,
174                 break;
175         }
176     break;
177
178     break;
179
180     case SDL_MOUSEBUTTONDOWN:
181         switch(event.button.button)
182         {
183             case SDL_BUTTON_LEFT:
184                 position.x = event.motion.x;
185                 position.y = event.motion.y;
186                 SDL_BlitSurface(SurfaceColler, NULL,
187                 break;
188         }
189     break;
190 }
191
192     SDL_Flip(ecran);
193 }
194
195 SDL_FreeSurface(pCouleur);
196 SDL_Quit();
197
198 return EXIT_SUCCESS;
199 }

```

Améliorations possibles

: 1. Vous pouvez rajouter des dessins à ce programme : c'est là qu'intervient l'utilité de SurfaceColler ! Imaginons que vous voulez un soleil, vous chargez le soleil et vous écrivez :

1 | SurfaceColler = soleil

2. Pourquoi ne pas pouvoir changer la couleur de fond 🤔.
3. Faire un menu (avec des boutons) qui permet de sélectionner la couleur du crayon.

Allez, Au boulot!

Xavierh
(/fr/memb
ers/kx9sbf
t3g945).



[\(/fr/members/kx9sbft3g945\).](#)



[12 août 2008 à 21:33:51](#)

S'il vous plaît, aidez nous et remplissez ce post d'exercices clairs et précis (je n'en demande pas trop quand même 🤔).

Un petit **UP** ne fait pas de mal...

Merci d'avance pour vos futures aides...

Xavierh, débutant qui veut aidez des autres débutants...

Askannz
(/fr/memb
ers/gn72v6
c38645).



[\(/fr/members/gn72v6c38645\).](#)

[13 août 2008 à 7:39:26](#)

Coordonnées de la souris

Quand:

Après l'apprentissage de SDL_ttf

Sujet principal:

afficher les coordonnées de la souris à l'écran

Consignes

: Créez une fonction qui affiche les coordonnées actuelles de la souris dans un coin de l'écran. Aide : voici le prototype de la fonction :

```
1 | void afficherPositionSouris(int x, int y, SDL_Rect posi
```

Notions travaillées

: Utilisation des évènements en SDL et de SDL_ttf.

Réponse

:

```

1  #include <SDL/SDL.h>
2  #include <SDL/SDL_ttf.h>
3  #include <SDL/SDL_Image.h>
4
5  void afficherPositionSouris(int x, int y, SDL_Rect pos:
6
7  int main(int argc, char *argv[])
8  {
9      SDL_Surface *ecran = {NULL};
10     SDL_Event event = {0};
11     SDL_Rect position = {0};
12     TTF_Font *police = NULL;
13     SDL_Color couleur = {0, 255, 0};
14
15     int continuer = 1;
16
17     SDL_Init(SDL_INIT_VIDEO);
18     TTF_Init();
19
20
21     écran = SDL_SetVideoMode(500, 500, 32, SDL_HWSURFAI
22     police = TTF_OpenFont("arial.ttf", 20);
23
24
25     while(continuer)
26     {
27         SDL_WaitEvent(&event);
28
29         SDL_FillRect(ecran, NULL, SDL_MapRGB(ecran->foi
30
31         switch(event.type)
32         {
33             case SDL_QUIT :
34                 continuer = 0;
35                 break;
36
37             case SDL_MOUSEMOTION :
38                 afficherPositionSouris(event.motion.x, evei
39                 break;
40
41         }
42         SDL_Flip(ecran);
43
44     }
45
46     TTF_CloseFont(police);
47     TTF_Quit();
48     SDL_Quit();
49
50     return EXIT_SUCCESS;
51 }
52
53 void afficherPositionSouris(int x, int y, SDL_Rect pos:
54 {
55     char chaine[100] = {0};
56     SDL_Surface *texte = NULL;
57
58     sprintf(chaine, "x : %d -- y : %d", x, y);
59
60     texte = TTF_RenderText_Solid(police, chaine, coulei
61     SDL_BlitSurface(texte, NULL, écran, &position);
62
63     SDL_FreeSurface(texte);
64 }

```


Améliorations possibles

:

- Vous remarquerez que la fonction prend un grand nombre de paramètres. Pourquoi ne pas les regrouper dans une structure ?
- Essayez d'afficher également les touches enfoncées. Vous devrez peut-être modifier le prototype afin qu'il ne soit pas trop long.
- Faites en sorte que l'affichage des coordonnées s'active ou se désactive lorsque l'utilisateur appuie sur une touche.

Xavierh
(/fr/memb
ers/kx9sbf
t3g945).



[\(/fr/members/kx9sbft3g945\).](#)



[13 août 2008 à 17:08:15](#)

Merci pour ces exercices!

Xavierh
(/fr/memb
ers/kx9sbf
t3g945).



[\(/fr/members/kx9sbft3g945\).](#)



[15 août 2008 à 12:05:54](#)

UP

danman
(/fr/memb
ers/drmg9
pj2lrq5).



(/fr/members/drmg9pj2lrq5).

20 août 2008 à 19:59:49

Bal littéraire

Quand:

Tous en général

Sujet principal

:

Créer une fonction qui permet d'afficher du texte a coté de la position du curseur avec des effets.

Consignes

:

Si le curseur bouge, ce n'est pas tout le texte d'un coup mais des lettres avec un décalage (la 1 bouge, puis au bout de 100 ms , la deux bouge ,etc;

argument : le texte a afficher, le temps de décalage (en ms)[,la couleur,le type d'affichage, la taille ,la police]. (les arguments entre [] ne sont pas obligatoire.

Notions travaillées

: Timers, tableaux de char, affichage de texte

Réponse

:

<secret>Aucun code actuellement fait ... désolé 😞 <secret>

Améliorations possibles

:

Gestion des collisions entre les lettres et l'écran;

Ajout d'autre effets comme la rotation, le changement de taille, le fondu , etc

ers/kx9sbf
t3g945)



[\(/fr/members/kx9sbft3g945\).](#)



[23 août 2008 à 14:19:16](#)

Aidez nous SVP!

Xavierh
(/fr/memb
ers/kx9sbf
t3g945).



[\(/fr/members/kx9sbft3g945\).](#)



[25 août 2008 à 9:30:50](#)

exercice?

Xavierh
(/fr/memb
ers/kx9sbf
t3g945).



[\(/fr/members/kx9sbft3g945\).](#)



[28 août 2008 à 19:47:51](#)

up

Xavierh
(/fr/memb

ers/kx9sbf
t3g945).



[\(/fr/members/kx9sbft3g945\).]((/fr/members/kx9sbft3g945).)



[31 août 2008 à 13:35:13](#)

Dernier up

Xavierh
(/fr/memb
ers/kx9sbf
t3g945).



[\(/fr/members/kx9sbft3g945\).]((/fr/members/kx9sbft3g945).)



[2 septembre 2008 à 7:18:11](#)

Je ferme.

Anonyme



[5 novembre 2011 à 17:37:28](#)

Titre du TP

Quand:

Avant TP Marion Sokoban

Sujet principal

: Ce sera un paint pour petit (exercice de m@teo21).

Consignes

: Ce logiciel devra coller des dessins sur une surface blanche quand on clique. L'utilisateur choisit l'image à coller avec les touches de 1 à 9. Il y aura donc 9 dessins. L'utilisateur pourra aussi effacer l'écran en appuyant sur `suppr`.

Notions travaillées

: Gestion des événements.

Réponse

:

```
1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <SDL.h>
4  #include <SDL_image.h>
5  #define ZOZOR 0
6  #define SAPIN 1
7  #define SOLEIL 2
8  #define NUAGE 3
9  #define HERBE 4
10 #define FLEUR 5
11 #define COEUR 6
12 #define ETOILE 7
13 #define MAISON 8
14
15
16 int main(int argc , char *argv[])
17 {
18     SDL_Surface *screen = NULL ;
19     SDL_Surface *banderolle = NULL;
20     SDL_Surface *image[9]; // on crée deux surface ou
21     int continuer = 1;
22     SDL_Event event;
23     int clic = 0;
24     SDL_Rect posSouris;
25     SDL_Rect posBanderolle;
26     int suppr = 0;
27     int imageActuelle = ZOZOR;//par default c'est zozi
28
29     if(SDL_Init(SDL_INIT_VIDEO) == -1)
30     {
31         fprintf(stderr , "%s\n" , SDL_GetError());
32         exit(-1);
33     }
34
35     SDL_WM_SetIcon(IMG_Load("image/icone.png"), NULL)
36
37     screen = SDL_SetVideoMode(640 , 480 , 32 , SDL_HW
38     if(screen == NULL)
39     {
40         fprintf(stderr , "%s\n" , SDL_GetError());
41         exit(-1);
42     }
43
44     image[ZOZOR] = SDL_LoadBMP("image/zozor.bmp");
45     if(image[ZOZOR] == NULL)
46     {
47         fprintf(stderr , "%s\n" , SDL_GetError());
48         exit(-1);
49     }
50
```

```

51 image[SAPIN] = IMG_Load("image/sapin.png");
52 if(image[SAPIN] == NULL)
53 {
54     fprintf(stderr, "%s\n", SDL_GetError());
55     exit(-1);
56 }
57
58 image[SOLEIL] = IMG_Load("image/soleil.png");
59 if(image[SOLEIL] == NULL)
60 {
61     fprintf(stderr, "%s\n", SDL_GetError());
62     exit(-1);
63 }
64
65 image[NUAGE] = IMG_Load("image/nuage.png");
66 if(image[NUAGE] == NULL)
67 {
68     fprintf(stderr, "%s\n", SDL_GetError());
69     exit(-1);
70 }
71
72 image[HERBE] = IMG_Load("image/herbe.png");
73 if(image[HERBE] == NULL)
74 {
75     fprintf(stderr, "%s\n", SDL_GetError());
76     exit(-1);
77 }
78
79 image[FLEUR] = IMG_Load("image/fleur.png");
80 if(image[FLEUR] == NULL)
81 {
82     fprintf(stderr, "%s\n", SDL_GetError());
83     exit(-1);
84 }
85
86 image[COEUR] = IMG_Load("image/coeur.png");
87 if(image[COEUR] == NULL)
88 {
89     fprintf(stderr, "%s\n", SDL_GetError());
90     exit(-1);
91 }
92
93 image[ETOILE] = IMG_Load("image/etoile.png");
94 if(image[ETOILE] == NULL)
95 {
96     fprintf(stderr, "%s\n", SDL_GetError());
97     exit(-1);
98 }
99
100 banderolle = IMG_Load("image/banderolle.png");
101 if(banderolle == NULL)
102 {
103     fprintf(stderr, "%s\n", SDL_GetError());
104     exit(-1);
105 }
106
107 image[MAISON] = IMG_Load("image/maison.png");
108 if(image[MAISON] == NULL)
109 {
110     fprintf(stderr, "%s\n", SDL_GetError());
111     exit(-1);
112 }
113
114 SDL_SetColorKey(image[ZOZOR], (SDL_SRCCOLORKEY |
115
116 SDL_FillRect(screen, NULL, SDL_MapRGB(screen->f
117
118 posBanderolle.x = 0;

```

```

119 posBanderolle.y = 0;
120
121 while(continuer)
122 {
123     while(SDL_PollEvent(&event))
124     {
125         switch(event.type)
126         {
127             case SDL_QUIT :
128                 continuer = 0;
129                 break;
130
131             case SDL_MOUSEBUTTONDOWN :
132                 if(event.button.button == SDL_BUTTON_
133                 {
134                     posSouris.x = event.button.x;
135                     posSouris.y = event.button.y;
136                     clic = 1;
137                 }
138
139                 else if(event.button.button == SDL_BU
140                 {
141                     suppr = 1;
142                 }
143                 break;
144             case SDL_KEYUP:
145                 switch(event.key.keysym.sym)
146                 {
147                     case SDLK_DELETE:
148                         suppr = 1;
149                         break;
150
151                     case SDLK_ESCAPE:
152                         continuer = 0;
153                         break;
154
155                     case SDLK_1:
156                         imageActuelle = ZOZOR;
157                         break;
158
159                     case SDLK_2:
160                         imageActuelle = SAPIN;
161                         break;
162
163                     case SDLK_3:
164                         imageActuelle = SOLEIL;
165                         break;
166
167                     case SDLK_4:
168                         imageActuelle = NUAGE;
169                         break;
170
171                     case SDLK_5:
172                         imageActuelle = HERBE;
173                         break;
174
175                     case SDLK_6:
176                         imageActuelle = FLEUR;
177                         break;
178
179                     case SDLK_7:
180                         imageActuelle = COEUR;
181                         break;
182
183                     case SDLK_8:
184                         imageActuelle = ETOILE;
185                         break;
186

```

```

187         case SDLK_9:
188             imageActuelle = MAISON;
189             break;
190
191         default :
192             ;
193     }
194     break;
195
196     default :
197         ;
198     }
199 }
200 SDL_BlitSurface(banderolle, NULL, screen, &po:
201
202 if(clic == 1)
203 {
204     SDL_BlitSurface(image[imageActuelle] , NU
205
206     clic = 0;
207 }
208
209 else if(suppr == 1)
210 {
211     SDL_FillRect(screen , NULL , SDL_MapRGB(s
212     suppr = 0;
213 }
214
215 SDL_Flip(screen);
216 }
217
218 SDL_FreeSurface(image[ZOZOR]);
219 SDL_FreeSurface(image[SAPIN]);
220 SDL_FreeSurface(image[SOLEIL]);
221 SDL_FreeSurface(image[NUAGE]);
222 SDL_FreeSurface(image[HERBE]);
223 SDL_FreeSurface(image[FLEUR]);
224 SDL_FreeSurface(image[COEUR]);
225 SDL_FreeSurface(image[ETOILE]);
226 SDL_FreeSurface(image[MAISON]);
227 SDL_Quit();
228
229 return EXIT_SUCCESS;
230 }

```

Améliorations possibles

: (rien pour le momoent)

Anonyme



5 novembre 2011 à 18:04:47

Pas la peine de déterrer pour poster l'exercice s'il est déjà dans le tuto.

[EXERCICES]Langage C, en SDL

Après avoir cliqué sur "Répondre" vous serez invité à vous connecter pour que votre message soit publié. ✕

Attention, ce sujet est très ancien. Le déterrer n'est pas forcément approprié. Nous te conseillons de créer un nouveau sujet pour poser ta question. ✕

Editeur · Markdown

Répondre

OPENCCLASSROOMS ▼

ENTREPRISES ▼

CONTACT ▼

EN PLUS ▼

 Français ▼



<https://itunes.apple.com/fr/app/openclassrooms-cours-en-ligne/id1164140533?pt=118427510&ct=FooterWeb&mt=8>