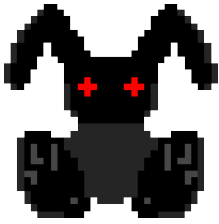




TekGUI



LAPINS NOIRS



TekGUI

Bibliothèque d'interface homme-machine

- Le Laboratoire aux Lapins Noirs -
lapinsnoirs@epitech.eu

Ce projet consiste à construire une bibliothèque d'interface homme-machine : un ensemble de composants graphiques interactifs permettant à un utilisateur novice d'utiliser son ordinateur de manière intuitive, le plus souvent sans recours à un manuel.

Nom du dépôt de rendu : tekgui



- 1 – Détails administratifs
- 2 – Fonctions autorisées
- 3 – Sujet
- 4 – Format
- 5 – Documentation
- 6 – Bonus
- 7 – Interface de correction automatique



1 – Détails administratifs

Votre dépôt de rendu doit s'appeler tekgui.

Votre programme devra être compilé avec un Makefile.

Votre bibliothèque devra être nommée « libtekgui.a ».

Vous devez disposer d'un ou plusieurs binaires de démonstration qui seront utilisés en pitch pour faire l'article de votre projet. Seront évalués seulement les aspects démontrés par ces démonstrations.

Vous devez respecter la norme.

Votre rendu ne devra pas comporter votre programme compilé, un fichier .o ou un fichier tampon type "#*" ou "*~" sous peine d'avoir 1,5.

Votre programme doit compiler avec -W -Wall -Werror.

Les lignes de liaison avec les bibliothèques doit être les suivantes :

```
-I/home/${USER}/.froot/include  
-L/home/${USER}/.froot/lib  
-llapin -lsfml-audio -lsfml-graphics -lsfml-window -lsfml-system -lstdc++ -ldl -lm
```

Votre programme **devra** s'exécuter en 1024*768 ou au-delà.

La première ligne de code votre main, directement après les déclarations de variables, **doit** être un appel à la fonction bunny_set_maximum_ram. La quantité **maximale** de RAM autorisé pour ce projet est de **15Mo**.

La quantité de RAM est susceptible d'être modifiée durant l'évaluation.

Vous **devez** appeler la fonction bunny_set_memory_check en lui passant **true**, cela avant de quitter votre programme ou encore plus tôt.

La taille **totale** de votre dépôt ne **doit pas** excéder **10Mo**.

Prenez garde à régler les droits d'accès de votre dépôt et de vos dossiers et fichiers.



2 – Fonctions autorisées

Ci-dessous, la liste des fonctions systèmes autorisées pour réaliser ce projet :

open, read, write, close
alloca, setjmp, longjmp, assert
rand, srand, time
opendir, readdir, closedir
stat, clock_gettime

Les libs math et dlfcn

Ainsi que les fonctions de la Liblapin présente dans lapin/basic.h

Toute fonction permettant d'allouer de la mémoire en dehors de `bunny_*alloc`, `alloca` ainsi que tous moyens de précipiter la fin du programme (`exit...`) sont **interdits**.

Les fonctions présentes dans lapin/advanced.h sont soumises à condition :

Toutes les fonctions **associées à un pointeur sur fonction** portant un nom type `"gl_bunny_my*"` doivent avoir été reprogrammées parfaitement. Une fonction disponible dans lapin/self_test.h vous aidera à vérifier leur fonctionnement. Les autres sont autorisées.

Si la fonction de test vous indique que votre fonction n'est pas une copie conforme, vous ne pouvez **pas** utiliser la fonction.



3 – Sujet

Le projet TekGUI consiste à programmer une bibliothèque d'interface homme-machine. Une interface homme-machine est un ensemble d'éléments graphiques pouvant réagir aux manipulations humaines : fenêtre, boutons, texte...

Votre travail va consister à créer une bibliothèque statique disposant de la fonction suivante :

```
t_tekgui *tekgui_load(const char *file) ;
```

Cette fonction chargera la description d'une interface graphique et ses réactions depuis le fichier passé en paramètre. Ce fichier doit être un fichier INI chargeable avec la LibLapin. Cette interface est représentée par la structure `t_tekgui` que vous **devez** déclarer dans un fichier en-tête appelé `tekgui.h`.

`tekgui_load` doit retourner `NULL` en cas d'erreur.

Vos fichiers en-tête (.h) doivent être situés à la racine de votre dépôt, dans le dossier `./inc/` ou dans le dossier `./include/` eux-mêmes situés à la racine de votre dépôt.

Cette bibliothèque devra également fournir cette fonction :

```
void tekgui_display(t_bunny_pixelarray *pix,  
                   t_tekgui *gui) ;
```

Cette fonction permet de dessiner dans `pix` l'interface graphique contenue par `gui`.

Ces fonctions sont les seules imposées. Elles seront utilisées pour la correction de votre programme. Cela ne signifie nullement que votre bibliothèque ne contiendra que ces fonctions ! Celle-ci doit contenir tout ce qu'il faut pour que votre interface fonctionne.

Vous devez mettre à disposition dans un dossier `./demo` au moins un fichier INI permettant de lancer avec succès la fonction `tekgui_load`. Ce dossier ne doit contenir que des fichiers INI permettant **même seul** de lancer avec succès cette interface.



4 – Format

La création du format à employer pour le fichier INI est à votre charge. Voici néanmoins des indications quand aux fonctionnalités obligatoires :

- Votre fichier doit permettre de définir les objets graphiques suivants :
 - ° Des boutons
 - ° Des boîtes de texte
 - ° Des boîtes d'image
 - ° Des cases à cocher
- Votre fichier doit permettre de définir aux objets graphiques une apparence et une organisation :
 - ° La position des objets
 - ° La couleur des objets
 - ° Le contenu des objets (Texte, label, image, coché ou pas, etc.)
- Votre fichier doit définir quelle fonction va réagir lorsqu'un événement se produit sur un objet :
 - ° Un événement de type click
 - ° Un événement de type text entered

Comment appeler une fonction à partir de son nom dans une chaîne de caractère ?
man dlopen
google « -rdynamic »

Les fonctions appelées doivent pouvoir modifier
les propriétés des éléments graphiques.



5 – Documentation

Étant donné la richesse du format que vous aurez à générer, vous devez écrire de la documentation. Le rôle de la documentation que vous devez écrire est de permettre à un utilisateur novice d'utiliser votre bibliothèque, il ne s'agit pas d'une documentation à usage interne.

Cette documentation se découpera en deux grandes parties, une première partie tutoriel, mettant en place des exemples primitifs d'utilisation de votre bibliothèque, et une seconde partie manuel détaillant l'intégralité des aspects de vos objets.

La documentation représente un tiers des points de ce projet.

Ci-dessous, une liste non exhaustive de bonus :

Gestion des caractères unicode

Possibilité d'établir des réactions aux événements dans des bibliothèques partagées

Mini-langage de script interne au fichier INI établissant une réaction aux événements.

Système de compilation mêlant INI et C visant à établir une système « IDE Like »

Sous-fenêtre

Boutons radios

Boite de dialogue « Attention ! » avec un bouton OK.

Boite de dialogue « Êtes-vous sur ? » avec les boutons Oui et Non.

Boite de dialogue « Flux » avec les boutons Annuler, Continuer et Ré-essayer.

Boite de dialogue « Parcourir »

Scrollbar

Menu déroulant en haut de fenêtres

Menu déroulant « clic droit »

Gestion du focus sur objet (Avec gestion de Tab, Shift+Tab, Return et Escape)

Listbox



7 – Interface de correction automatique

Afin de permettre une correction automatique de votre travail, nous vous demandons d'implémenter les fonctions suivantes :

```
void          tekpixel(t_bunny_pixelarray    *pix,  
                      t_bunny_position      *pos,  
                      t_color               *color) ;  
  
void          *tekfunction(const char       *funcname) ;  
  
void          tektext(t_bunny_pixelarray    *out,  
                      const char           *str,  
                      t_bunny_pixelarray    *font_png) ;
```

Ces fonctions peuvent tout à fait ne pas être utilisé dans votre projet si vous ne le souhaitez pas.

L'échec à la correction automatique vous fermera l'accès à l'évaluation.
Faites donc très attention en travaillant ces fonctions.

Ces fonctions doivent se trouver dans des fichiers .c situés dans le dossier ./tcore/, lui même situé à la racine de votre dépôt. L'intégralité du contenu de ce dossier et exclusivement de ce dossier sera compilé avec la moulinette.

Faites attention à ce qu'aucun main ne s'y trouve !

Si vos fichiers incluent des fichier en-têtes (.h), veuillez à les placer soit à la racine de votre dépôt, soit dans un dossier ./include/, soit dans un dossier ./inc/ eux même à la racine.

tekpixel dépose dans pix un pixel de couleur color à la position indiqué dans pos.

tekfunction prend en paramètre le nom d'une fonction situé dans le programme et en retourne le pointeur sur fonction. (man dlopen)

tektext prend en paramètre une image out sur laquelle devra être écrite la chaîne str, à l'aide de l'image font_png (Fichier font.png, fourni avec la Liblapin, chargé en mémoire). Le texte doit être écrit en haut à gauche de l'image out. L'image font_png sera utilisée sans transformations.

Les caractères doivent être séparés d'un espace large d'un pixel situé sur la droite.
Tabulations horizontales, verticales et saut de ligne ne seront pas testé.

Le non-fonctionnement de l'une des trois fonctions est éliminatoire.