



Programmation graphique



Capes et epees 1

- Le Laboratoire aux Lapins Noirs -
pedagogie@ecole-89.com

Ce document est strictement personnel et ne doit en aucun cas être diffusé.

INDEX



Avant-propos :

- 01 – Détails administratifs
- 02 – Propreté de votre rendu
- 03 – Règlement quant à la rédaction du code C
- 04 – Construction de votre rendu
- 05 – Fonctions interdites

Projet :

- 06 – L'invasion commence
- 07 – Le cor de Helm résonne
- 08 – La pluie tombe
- 09 – Il ne faut jamais boire le sang d'un mort



01 – Détails administratifs

Votre travail doit être envoyé via l'interface de ramassage de l'**Infosphère** :

Pour cette activité, vous rendrez votre travail sous la forme d'une archive au format tar.gz. Cette archive devra contenir l'ensemble de votre travail tel que demandé dans la section 4.

Pour créer cette archive .tar.gz, il vous suffit d'utiliser la commande suivante :

```
$> tar cvfz mon_fichier.tar.gz fichier1 fichier2 fichier3
```

Le nom « mon_fichier.tar.gz » étant à remplacer par le nom que vous souhaitez donner votre fichier archive, et « fichier1 », « fichier2 », « fichier3 » par les fichiers ou dossiers que vous souhaitez mettre dans cette archive. Vous pouvez vérifier le contenu de votre archive à l'aide de la commande « **tar -t mon_archive.tar.gz** ».

Ce travail est à effectuer seul. Vous pouvez bien sûr échanger avec vos camarades, néanmoins vous devez être l'auteur de votre travail. Utiliser le code d'un autre, c'est **tricher**. Et tricher annule **toutes** les médailles que vous avez reçu sur l'activité. La vérification de la triche est réalisée de la même manière que la correction : de manière **automatique**. Prenez garde si vous pensez pouvoir passer au travers.

Médailles accessibles :



Réussir à rendre :

Vous avez réussi à envoyer votre travail au système de correction.



02 – Propreté de votre rendu

Votre rendu, c'est à dire le contenu de l'archive ou du dépôt que vous entrez sur l'interface du TechnoCentre, doit respecter **strictement** l'ensemble des règles suivantes :

- Il ne doit contenir **aucun** fichier objet. (*.o)
- Il ne doit contenir **aucun** fichier tampon. (*.~, #*#)
- Il ne doit pas contenir votre production finale (programme ou bibliothèque)

La présence d'un fichier interdit mettra immédiatement fin à votre évaluation.

Médailles accessibles :



Rendu propre

Votre rendu respecte les règles de propretés imposées.



03 – Règlement quant à la rédaction du code C

Vous devez respecter l'intégralité des tables de norme pour ce projet.

04 – Construction de votre rendu



Le programme de correction va construire une sous-partie déterminée de votre rendu afin d'effectuer des tests dessus. En voici les paramètres :

- Les fichiers qui seront compilés sont ceux qui auront l'extension *.c.
- Seuls les fichiers dans le(s) dossier(s) ./ seront compilés.
- Les fichiers seront compilés avec -W -Wall -Werror.

- Tous les fichiers seront compilés **ensemble**, cela signifie donc que chaque fonction doit être unique, et que vous pouvez utiliser les fonctions des autres exercices dans chaque exercice.

**L'ensemble du code que vous rendez doit pouvoir être compilé.
En cas d'échec de la compilation, vous ne serez pas évalué.**

Médailles accessibles :



Construction partielle

Les éléments requis de votre projet se construisent séparément.



05 – Fonctions interdites

Vous n'avez le droit à aucune autre fonction que celle précisée dans la liste ci-dessous :

Fonctions systèmes

- open
- close
- write
- read
- ioctl
- sin
- cos
- rand
- srand

Fonctions de la LibLapin

- bunny_start
- bunny_stop
- bunny_new_pixelarray
- bunny_delete_clipable
- bunny_save_picture
- bunny_blit
- bunny_set_*_function
- bunny_set_*_response
- bunny_loop
- bunny_malloc
- bunny_free
- bunny_open_configuration
- bunny_configuration_getf
- bunny_delete_configuration

**L'utilisation d'une fonction interdite est assimilée à de la triche.
La triche provoque l'arrêt de l'évaluation et la perte des médailles.**



06 – L'invasion commence

Réalisez un programme qui ouvre une fenêtre, la remplit de noir puis dessine 10 points blancs à des positions aléatoires.

Ensuite, à chaque tour de boucle, vous dessinerez un point supplémentaire **si et seulement si** il y a déjà un point blanc à gauche, à droite, en haut ou en bas de ce point, dans la case d'à côté.

Votre programme quittera sur une pression de la touche echap.



07 – Le cor de Helm résonne

Écrivez un programme qui affiche une onde sinusoïdale à l'écran. Vous n'aurez besoin que des fonctions de la LibLapin et de la fonction **cos**. Vous n'afficherez qu'une unique période (un aller retour). Le point de départ de l'onde sera à mi hauteur sur la partie gauche de la fenêtre et l'onde s'achèvera à mi hauteur sur la partie droite de la fenêtre, toujours sur un bord.

Votre onde aura comme amplitude la hauteur de la fenêtre.

Votre programme quittera sur une pression de la touche echap.



08 – La pluie tombe

Écrivez un programme commençant par noircir toute la fenêtre puis affichant des pixels aléatoires blanc en haut de la fenêtre. A chaque tour de boucle, les pixels blancs chuteront de hauteur, de sorte à imiter la pluie.

Votre programme quittera sur une pression de la touche echap.



09 – Il ne faut jamais boire le sang d'un mort

Le dernier programme que vous allez réaliser commence également par remplir de noir la fenêtre. Cette fois rien ne se passera sans intervention de l'utilisateur.

A chaque clic de souris, vous placerez un point rouge à la position du clic. Ce point rouge devra tomber de la même manière que la pluie dans l'exercice précédent.

Votre programme quittera sur une pression de la touche echap.