Projet C++

Nibbler

Giron David thor@epitech.net
Baudry Dan baudry_d@gmail.com

Résumé: Le but de ce projet est de creer votre version du jeu Snake capable d'utiliser au moins 3 GUI differentes. Ces GUI prendront la forme de bibliotheques partagees.

Table des matières

I Snake

`Snake` est un jeu video classique cree dans les années 1970. Sa simplicite et son addictivite l'ont rendu disponible sur l'ensemble des plate-formes de jeu existantes sous des noms divers et varies. Pour les vieux jeunes, `Snake` est synonyme de nombreuses heures de glandouille au lycee dans sa version sur le telephone portable `Nokia 3310` et assimiles. Pour les plus jeunes, `Snake` est un projet de tech2 faisant reference a un obscur jeu prehistorique qui n'interesse plus que les fossiles qui se croient encore dans le coup.

Il n'est jamais trop tard pour decouvrir un classique : http://jeux-flash.jeu-gratuit.net/jeux_classiques/snake_250.html

Comme vous avez pu le (re)decouvir, `Snake` consiste a deplacer dans un plan un serpent represente par des sections devant manger de la nourriture, ce qui le fait grandir d'une section a chaque fois. La partie s'arrete quand la tete du serpent touche un des bords du plan ou une de ses sections. Le but du jeu est d'obtenir le serpent le plus long possible.

De tres nombreuses variantes de `Snake` ont vu le jour et certaines proposent des obstacles pour augmenter la difficulte, un systeme de score, des bonus, etc.

II Le projet

`Nibbler` est un jeu video ou le joueur controle un serpent se deplacant dans un plan dont le bord de la fenetre/ecran constitue les limites. Si le serpent percute un de ces bords ou une des sections qui composent son corps, la partie se termine.

II.1 Generalites

Pour votre culture, il est important d'avoir ete initie a plusieurs bibliotheques graphiques. C'est pourquoi votre rendu devra obligatoirement en utiliser 3 differentes. L'interet de ce projet etant de vous faire manipuler des bibliotheques dynamiques a l'execution, le rendu graphique et les inputs devront se trouver dans une bibliotheque dynamique utilisee a l'execution. Le corps de votre programme devra donc interagir de maniere uniforme avec l'une ou l'autre de vos bibliotheques.

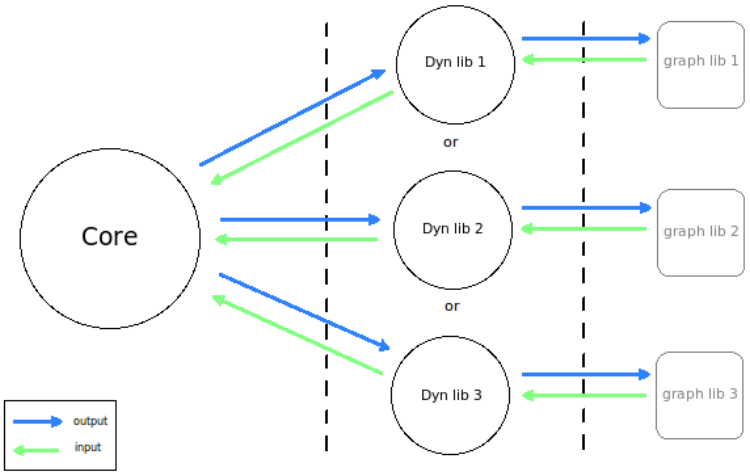

FIGURE 1 – `Architecture`

Chaque GUI possible pour le programme devra prendre la forme d'une bibliotheque partagee qui sera chargee et utilisee dynamiquement par le programme principal. Il est rigoureusement **INTERDIT** de faire une reference quelconque a l'une ou l'autre des bibliotheques graphiques que vous souhaitez utiliser dans votre programme principal. Seules vos bibliotheques dynamiques peuvent le faire.

`Je repete : Il est rigoureusement INTERDIT de faire une reference quelconque a l'une ou l'autre des bibliotheques graphiques que vous souhaitez utiliser dans votre programme principal.`

II.2 Les bibliotheques dynamiques

Vos bibliotheques dynamiques doivent etre utilisees a l'execution de votre programme. Cela signifie que vous DEVEZ utiliser les fonctions `dlopen`, `dlclose`, `dlsym` et `dlerror` pour manipuler vos bibliotheques dynamiques, comme vu dans le cours. En passant votre programme en parametre a la commande `ldd`, les dependances vers vos bibliotheques ne doivent donc pas apparaitre.

Ces bibliotheques peuvent etre vues comme des plugins pour votre programme principal mettant a sa disposition differentes interfaces graphiques. Elle ne doivent en AUCUN CAS traiter la logique du jeu. Elle ne servent qu'a afficher l'etat du jeu au joueur et a recuperer les inputs de ce dernier pour les transmettre au programme principal.

Vous ne DEVEZ PAS faire de difference entre l'une ou l'autre de vos bibliotheques dans votre programme principal. Toutes vos bibliotheques DOIVENT etre manipulees de maniere generique et uniforme. C'est cette genericite qui interessera le plus vos correcteurs lors de la soutenance.

Indices

Le cours, les sources a votre disposition sur l'e-learning et le TP sont des sources d'information precieuses pour realiser cette partie du projet de maniere optimale. Toutefois, rien ne remplacera la reflexion.

II.3 Les bibliotheques graphiques

Vous pouvez choisir les 3 bibliotheques graphiques que vous souhaitez utiliser dans vos bibliotheques dynamiques parmis les suivantes :

- NCurses
- NDK++
- MinilibX
- Xlib
- GTK++
- OpenGL
- SFML ou SDL (Pas les deux)
- Qt

Si vous souhaitez utiliser une bibliotheque graphique qui n'apparait pas dans cette liste, contactez moi pour obtenir l'autorisation. Je vous recommande de me contacter via le forum C++ de l'intranet afin que votre proposition profite a tout le monde.

`Certaines de ces bibliotheques ne sont peut-etre pas installees sur les dumps standards de l'ecole. Il sera donc peut-etre necessaire que vous les installiez manuellement sans droits privilegies. Cela ne vous donnera que plus de culture.`

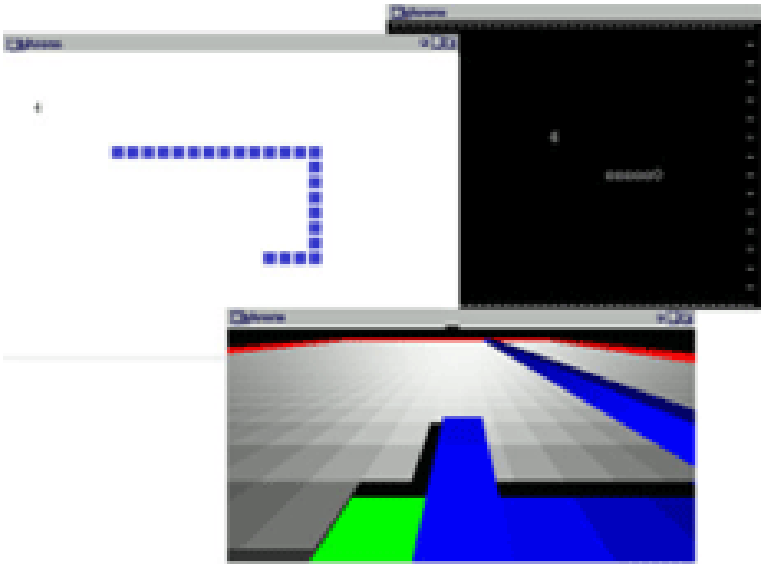

FIGURE 2 – `Exemples de rendus`

II.4 Le jeu

Les regles du jeu sont particulierement simples et DOIVENT etre respectees. Voici les bases :

- L'unite de mesure est la "case". La taille d'une case est a votre discretion mais DOIT etre raisonnable et PEUT dependre de la bibliotheque graphique en cours d'utilisation. Une case d'1px est trop petite et une case de 1000px est trop grande par exemple.
- L'aire de jeu est un plan fini de cases dont les bords ne sont pas traversables.
- Le serpent commence avec une taille de 4 cases au milieu de l'aire de jeu.
- Le serpent avance droit devant lui automatiquement a une vitesse fixe. Chaque section de la queue suit le chemin exact de la tete.
- Le serpent peut tourner vers la droite ou vers la gauche de 90° lorsque la touche correspondante du clavier est pressee.
- Le but du jeu etant de faire manger de la nourriture a votre serpent pour le faire grandir, votre aire de jeu ne DOIT JAMAIS contenir strictement moins d'un element de nourriture.
- La nourriture ne peut jamais occuper plus d'une case de surface.
- Lorsque la tete du serpent se trouve sur une case contenant de la nourriture, celle-ci disparait et une section d'une case est ajoutee a la queue du serpent. Si la case suivant la derniere case de la queue du serpent est un mur ou une autre case de la queue du serpent, la section supplementaire est ajoute dans la premiere case libre autour de la derniere case de la queue du serpent. Si aucune case n'est libre, la partie s'acheve. Si la nouvelle section du serpent a pu etre ajoutee, une nouvelle nourriture apparait.

Une fois votre projet complet et fonctionnel avec ses 3 bibliotheques dynamiques, vous pouvez envisager d'etendre les regles du jeu en tant que bonus. Vous pouvez vous inspirer des idees suivantes :

- Un systeme de score
- De la nourriture bonus apparaissant un court laps de temps
- La section de la tete a une apparence differente des sections du corps du serpent
- La vitesse de deplacement augmente au cours de la partie
- L'aire de jeu comporte des obstacles
- Le serpent grandit d'un nombre aleatoire de sections a chaque fois qu'il mange
- Un boost de vitesse limite en appuyant sur la barre espace
- ...

II.5 Utilisation

Le resultat de la compilation de votre rendu DEVRA etre un programme et 3 bibliotheques dynamiques. Le programme DEVRA etre nomme "nibbler" et les bibliotheques dynamiques DEVRONT etre nommees d'apres le nom de la bibliotheque graphique qu'elles utilisent. Seul le nom de l'executable est impose, toutefois vos bibliotheques DEVRAIENT avoir un nom similaire a "lib_nibbler_XXX.so", ou "XXX" est le nom de la bibliotheque graphique utilisee.

Votre executable "nibbler" devra prendre 3 parametres :

- La largeur de l'aire de jeu
- La hauteur de l'aire de jeu
- Le nom de la bibliotheque dynamique a utiliser

Exemple :

```
1    >./nibbler 30 20 lib_nibbler_opengl.so
```

Vous DEVEZ traiter les cas suivants :

- Si on passe un nombre d'arguments strictement different de 3 a votre programme, celui-ci DOIT afficher un usage et quitter proprement.
- Si la taille de l'aire de jeu est impossible (valeurs negatives, non numeriques, trop petites, trop grandes, ...), votre programme DOIT afficher un message d'erreur pertinent et quitter proprement.
- Si la bibliotheque dynamique n'existe pas ou n'est pas compatible, votre programme DOIT afficher un message d'erreur pertinent et quitter proprement.

Au cours de l'execution de votre programme, les 3 touches suivantes du clavier DOIVENT avoir le comportement suivant :

- Fleche gauche : Fait tourner le serpent de 90° vers la gauche.
- Fleche droite : Fait tourner le serpent de 90° vers la droite.
- Echap : Termine la partie, ferme la fenetre et termine le programme proprement.

III Consignes

Vous etes globalement libres de faire l'implementation que vous voulez. Cependant, il y a quelques restrictions :

- Les seules et uniques fonctions de la `libc` autorisees sont celles qui encapsulent les appels systemes et qui n'ont pas d'equivalent en C++.
- Toute valeur passee par copie plutot que par reference ou par pointeur doit etre justifiee, sinon vous perdrez des points.
- Toute valeur non `const` passee en parametre doit etre justifiee, sinon vous perdrez des points.
- Toute fonction membre ou methode ne modifiant pas l'instance courante n'etant pas `const` doit etre justifiee, sinon vous perdrez des points.
- Il n'existe pas de norme en `C++`. Cependant, tout code que nous jugerons illisible ou trop sale pourra etre sanctionne arbitrairement. Soyez serieux !
- Gardez un oeil sur ce sujet regulierement car il est susceptible d'etre modifie.
- J'attache une grande importance a la qualite de mes sujets, donc si vous trouvez des fautes de frappe, d'orthographe ou des incoherences, merci de me contacter a l'adresse thor@epitech.net pour que je puisse les corriger.

IV Consignes de rendu

Vous devez rendre votre projet sur le depot `SVN` mis a votre disposition par le `Koalab`. Vos depots seront ouverts au maximum 48h apres la date de fin d'inscription au projet, intranet `Epitech` faisant foi. Il ne sera plus possible de s'inscrire au projet et de passer en soutenance une fois cette date depassee.

Vos depots seront fermes en ecriture a l'heure exacte de la fin du projet, intranet `Epitech` faisant foi.

Seul le code present sur votre depot sera evalue lors de la soutenance. La documentation relative aux depots fournis par le `Koalab` est fournie avec ce sujet.

Bon courage !