
PROJET C – LES PINGOUINS

Réalisé par : Delpech Marc, Levy Yoni, Rousselle Matthieu, Tissier Romain

TELECOM Nancy 1^{ère} année

Date : 25 Mai 2015



Sommaire

RESUME	3
CAPTURES D'ECRAN	3
DESCRIPTIF DES DIFFERENTES ETAPES.....	4
AVANT DE COMMENCER A CODER.....	4
APRES AVOIR COMMENCER A CODER	5
SOURCES D'AIDE ET D'INSPIRATION	6
INSPIRATIONS / JOUABILITE	6
GESTION DU PROJET	6
BIBLIOTHEQUES C.....	6
GRAPHISMES.....	6
AUTRES REFERENCES	6

Résumé

Le but de notre projet de C est de ré-implémenter le jeu des Lemmings. Pour cela, nous avons choisi d'utiliser des images et des sprites qui sont tirés du jeu Pingus, une version libre de droits des Lemmings multiplateforme et notamment sous Linux. Le principe du jeu est le suivant : plusieurs pingouins arrivent à intervalle régulier sur la carte depuis l'entrée du niveau et ils doivent atteindre la sortie située ailleurs sur l'écran avec l'aide de l'utilisateur. En effet, ils peuvent effectuer différentes actions influant sur l'état du pingouin, comme planer, et sur le décor, comme creuser un trou ou créer un pont.

Notre projet étant un projet étudiant, il nous paraît plus approprié que notre projet soit libre de droits afin que d'autres personnes puissent le réutiliser par la suite. C'est pourquoi il nous semble plus adapté d'utiliser la Licence Publique Générale GNU. De plus, nous utilisons seulement des images et sprites ayant une licence Creative Commons et par conséquent notre projet ne peut pas être commercialisé.

Captures d'écran

Voici quelques captures d'écran de notre projet :



Fenêtre de sélection des niveaux



Fenêtre de jeu du niveau 2

Descriptif des différentes étapes

Avant de commencer à coder

Avant de commencer à coder le projet, nous avons envisagé notre travail en 5 grandes étapes.

- Créer et afficher un pingouin, puis le faire avancer jusqu'à un obstacle pour enfin lui faire réaliser un demi-tour ; cette étape avait pour principal objectif de mettre en place la gestion de la collision verticale.
- Gérer la collision horizontale, utilisée lors de la chute libre du pingouin, et de faire apparaître à l'écran un nombre donné de pingouins.
- Mise en place de niveaux, gestion des éléments graphiques constituant les différents niveaux et mise en place d'une entrée et d'une sortie.

- Améliorations potentielles du jeu comme un plateau « scrollable », des raccourcis claviers, la possibilité de changer la vitesse globale du jeu, le mettre en pause, ajouter des sons, changer la vitesse d'apparition des pingouins selon le niveau, ajouter un menu pour le jeu ainsi qu'un éditeur de niveaux, ou encore de permettre la sauvegarde de l'avancement du joueur.

Après avoir commencer à coder

Les deux premières étapes ont été réalisées simultanément puisqu'elles sont assez proches.

La génération des niveaux se fait grâce à des fichiers textes qui contiennent le nombre de pingouins, les différents éléments du décor, leurs positions et leurs tailles.

Nous avons réussi à implémenté différentes actions pour les pingouins : creuser horizontalement ou verticalement dans le décor, planer lors de la chute libre pour ne pas tuer le pingouins lors du contact avec le sol, faire stopper un pingouin afin que les autres ne continuent pas leur chemin en arrivant à son niveau, faire exploser un pingouin pour détruire le décor ou encore construire un escalier.

Cependant, certaines actions ont du être modifiées. Par exemple, l'action de construire un escalier est devenue l'action de créer un pont. Nous n'avons pas implémenté la possibilité de creuser en diagonale ni celle de grimper sur les parois verticales.

Un menu général, un menu de sélection des niveaux, un menu de victoire et de défaite ont été créés. Il y a aussi la possibilité de mettre en pause le jeu. La vitesse d'apparition des pingouins a été standardisée pour tous les niveaux du jeu.

Les autres idées citées précédemment n'ont pas pu être implémentées.

Nous avons décidé de rajouter une dernière étape qui est celle de la correction des fuites mémoires car lorsque nous avons commencé à coder, le lancement du projet retournait de nombreuses erreurs de segmentations. Nous avons donc au fur et à mesure de l'avancement du projet utilisé le Framework Valgrind qui a permis de trouver de nombreuses fuites mémoire que l'on a corrigé par la suite. La version finale du projet, lorsqu'elle est lancée avec Valgrind, comporte cinq erreurs qui proviennent de la librairie graphique SDL qui comporte des conflits avec le driver NVIDIA.

Sources d'aide et d'inspiration

Inspirations / Jouabilité

- Martin Quinson / Vincent Thomas
- Jeu original des Lemmings de DMA Design
- Pingus : <http://pingus.seul.org/>
- Reproduction du jeu en DHTML (JavaScript) : elizium.nu/scripts/lemmings
- Précisions sur le jeux : wikipedia.org/wiki/Lemmings
- Lemmings sur PS VITA : jeuxvideoz.net/les-lemmings-s%E2%80%99offrent-une-sortie-sur-vita
- Jeu original Lemmings Revolution : <http://www.old-games.com/download/6569/lemmings-revolution>

Gestion du projet

- Tutoriel rédaction cahier des charges : openclassroom.fr
- Note sur le cahier des charges : Mémotech - Sciences de l'ingénieur

Bibliothèques C

- Utilisation de la SDL 2.0 (site officiel) : libsdl.org
- Tutorat SDL 2.0 :
 - willusher.io/pages/sdl2
 - sdltutorials.com
- Tutoriel installation SDL Fedora : fedora-fr.org
- Tutoriel SDL en français : <http://alexandre-laurent.developpez.com/tutoriels/sdl-2/creer-premieres-fenetres/>

Graphismes

- Lemmings (la licence étant ambiguë, nous avons choisi de ne pas utiliser les images et sprites de ce site) : users.tpg.com.au/wagnerbe/lemdoom/lemsprite.html
- Liste éléments graphiques / sonore libres (finalement pas utilisée) : <http://jeux.developpez.com/medias/>
- Sprite de Pingus : <http://pingus.seul.org/>

Autres références

- Tutoriel utilisation Git : doc.ubuntu-fr.org/git
- Conseil divers : stackoverflow.com