Projet langage C
ING3 Semestre 1

La revanche de Snoopy

Créé sur Gameboy en noir et blanc en 1990, voici l'écran originel.

Snoopy's Magic Show est un jeu vidéo créé en 1990 qui met en scène le personnage de Snoopy. C'est un jeu de réflexion de type "puzzle game" où le but est de récupérer 4 oiseaux pour passer au niveau suivant... mais le chemin le long des niveaux est semé d'embûches...

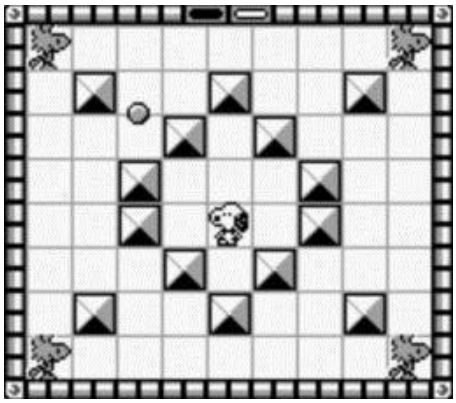
SOMMAIRE

Introduction

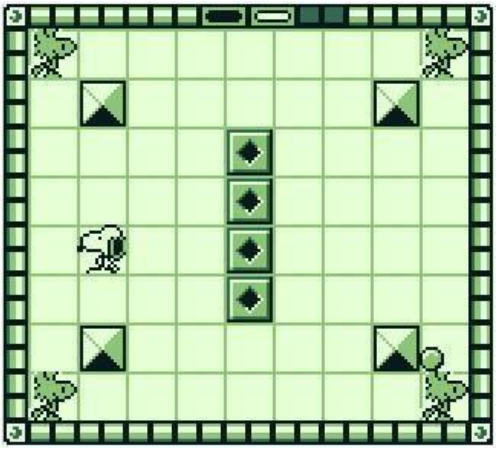
Le but de Snoopy est de récupérer 4 oiseaux aux 4 coins du niveau **en un temps imparti** (2 minutes par niveau). Le problème est que ces 4 oiseaux ne sont pas si faciles à récupérer. Une balle rebondit constamment dans le niveau afin de freiner Snoopy dans sa quête. Mais ce n'est pas tout, d'autres pièges sont présents comme des téléporteurs que la balle peut emprunter ou des cases piégées, voir même des blocs à pousser ou à casser...

Le jeu original comporte 120 niveaux. Un mot de passe est disponible pour chaque niveau.

Voici quelques captures d'écran du jeu original :



Les quatre oiseaux à sauver sont situés dans les quatre coins du niveau. La balle se trouve dans la zone supérieure gauche du niveau. Dans ce niveau, Snoopy doit pousser des blocs...



Dans ce niveau, Snoopy peut casser les blocs du milieu pour avoir des objets (bonus ou malus)

Lorsqu'ils sont poussés, les blocs "flèches" se déplacent automatiquement dans le sens de la flèche jusqu'à se stabiliser sur un obstacle quelconque.

Le temps imparti est symbolisé par des rectangles qui entourent le niveau.

Illustration de la gestion du temps Chaque unité de temps représente deux secondes

Parmi les objets disponibles dans le jeu original, on trouve :

- Une horloge : objet pour figer le temps (et donc aussi la balle mais pas Snoopy)
- Des objets symbolisant l'invincibilité
- Des blocs qu'on peut pousser d'une case à chaque fois
- Des blocs qu'on peut casser
- Des blocs qui disparaissent et réapparaissent à intervalle de temps régulier
- Des flèches "bloc mobile" qu'on pousse une fois et qui se déplacent dans le sens de la flèche jusqu'à un obstacle quelconque

Les ennemis du jeu original sont les suivants :

- Les fameuses balles qui rebondissent dans le niveau (il peut y en avoir 2 maximum)
- un méchant Snoppy qui a la caractéristique de suivre le vrai Snoopy :)

Présentation du projet : "La revanche de Snoopy"

Le jeu proposé cette année s'inspire fortement de son ancêtre présenté dans le paragraphe précédent mais il devra être **en mode console sans graphique** ! Pour simplifier, le jeu comportera **au MINIMUM 1 niveau par membre de l'équipe de codeurs (donc 4 niveaux pour une équipe de 4), les niveaux devront être de difficulté croissante en faisant intervenir un bloc d'un nouveau genre à chaque nouveau niveau.** Une gestion des scores et des mots de passe par niveau et la possibilité de charger une partie sauvegardée devront être prévues.

Une fois lancé, le jeu proposera un menu classique permettant de réaliser les actions suivantes :

1. Règles du jeu (affichées à l'écran)
2. Lancer un nouveau Jeu à partir du niveau 1
3. Charger une partie
4. Lancer directement un niveau via son Mot de passe
5. Scores
6. Quitter

Lire attentivement et respecter scrupuleusement toutes les règles du jeu qui suivent.

Initialement, le joueur possède 3 vies.

Chaque niveau devra être résolu en moins de 120 secondes. Si le temps est écoulé, le joueur perd une vie et recommence le niveau. Le but est de récupérer les 4 oiseaux du niveau sans se faire toucher par la balle et/ou les ennemis (si présents).

Représentation d'un niveau

Chaque niveau sera représenté par une matrice rectangulaire de caractères de 10 lignes par 20 colonnes (à adapter à votre écran pour éviter le scroll), contenant les objets suivants :

- Snoopy (le personnage)
- La balle sur au moins un niveau
- Les oiseaux à récupérer
- Les blocs qu'il faut pousser (case par case ou à déplacement auto (flèche))
- Les blocs qu'il faut casser
- Les blocs piégés (nouveauté !!)
- Autres blocs possibles facultatifs mais pouvant donner des points bonus : voir la section « jeu de base »

Déplacement de Snoopy

Snoopy ne peut pas se déplacer en diagonale. Il ne peut se déplacer que dans les 4 directions classiques (Haut, Bas, Gauche et Droite) et d'une seule case à la fois. Evidemment, en cas d'obstacle, Snoopy ne pourra pas effectuer son déplacement. Il ne peut pas sortir du niveau.

Mouvement de la balle

La balle se déplace exclusivement en diagonale et rebondit uniquement sur les murs (les bords de la matrice). La vitesse de la balle est fixe. La balle « traverse » tous les obstacles du terrain sans changer de direction. Elle « tue » Snoopy quand elle le touche.

Gestion des objets

Un bloc poussable ne peut être poussé **qu'une seule fois** dans une direction précise mais ne peut pas sortir du niveau ! Ce type de bloc n'est pas traversable.

Un bloc cassable ne peut pas être traversé. Pour le casser, il faut appuyer sur une touche spéciale. Il ne libère aucun item.

Un bloc piégé tue instantanément si on le touche.

Vous pourrez inventer des blocs « surprise » qui offrent un avantage quand ils sont cassés (vie supplémentaires, invincibilité, gel de la balle…)

Condition de victoire ou de défaite

Pour gagner, il faut récupérer les 4 oiseaux du niveau. Un fois un niveau terminé, on charge automatiquement le niveau suivant en donnant son code d'accès et ainsi de suite.

Quand le joueur perd toutes ses vies, on affiche un écran de GameOver et le jeu revient au menu principal.

Gestion du temps

Chaque niveau aura un timer initialisé à 120 secondes. Quand le timer atteint 0, le joueur perd une vie. Le timer pourra être représenté par des cases autour du terrain qui changent de couleur pour symboliser un décompte comme dans le jeu originel, ou par un simple affichage type compte à rebours.

Gestion des scores

La gestion des scores s'effectue de cette manière pour chaque niveau :

$$S_{niveau} = temps\ restant * 100$$

Au fur et à mesure des niveaux, les scores s'additionnent pour former le score final.

Exemple : Le niveau 1 est fini en 30 secondes, le niveau 2 en 55 secondes et le dernier niveau en 59 secondes. Le score du joueur à la fin du niveau 3 sera :

$$S_{total} = S_{niveau1} + S_{niveau2} + S_{niveau3} = 30 * 100 + 5 * 100 + 1 * 100 = 3600 \text{ points}$$

Sauvegarde et chargement d'une partie

A chaque instant, le joueur peut s'il le souhaite sauvegarder sa partie en appuyant sur la touche 's' du clavier. Dès qu'il le fait, le programme lui demande le nom du fichier de sauvegarde puis retourne sur le menu principal (la partie en cours est donc quittée).
La sauvegarde se fera au choix soit dans un fichier texte ou un fichier binaire et comprend les éléments suivants :

- La position de Snoopy
- La position de la balle (ou des balles)
- La position de tous les éléments du décor (blocs, oiseaux, ...) restant
- Le temps restant
- Le nombre de vies
- Le score courant

Pour charger une partie, il faut passer par le menu principal et choisir "Charger une partie". Le joueur est ensuite invité à entrer le nom de son fichier de sauvegarde.

Gestion des mots de passe

Chaque niveau sera accessible par un mot de passe unique. Un joueur peut donc s'il connait le mot de passe accéder au niveau de son choix à partir du menu principal. Ce mode devrait être très utile en soutenance pour montrer rapidement vos fonctionnalités...

Mode "pause"

Si le joueur appuie sur une touche de « pause », le jeu se met en pause, c'est à dire :

- la(es) balle(s) se fige(nt) (et les ennemis si présents)
- Snoppy ne peut plus se déplacer
- le timer s'arrête

Pour enlever la pause, il suffit d'appuyer à nouveau sur la touche de « pause ».

Cahier des charges

1) Jeu à développer

Votre jeu commence par le menu.

Implémenter le jeu en suivant les instructions suivantes :

- Pour chaque niveau de jeu, le plateau de jeu sera représenté par une matrice 2D en mémoire et un chronomètre défile.
- Les plateaux de jeu seront stockés dans des fichiers texte, **chaque élément du décor étant identifié par un chiffre dans le fichier et affiché à l'écran par un caractère de votre choix en console**, comme par exemple un caractère de la table ANSI de la page de code 850 (voir exemples ci-dessous) :

Identifiants (chiffres) d'éléments de décor dans le fichier	Exemples d'éléments du décor affichés en console (caractères ANSI)
0 (case vide)	
1 (bloc cassable)	♠
2 (bloc poussable à déplacement auto)	→
3 (bloc piégé)	♣
4 (bloc invincible)	☼
5 (bloc disparition/apparition)	◙
6 (bloc à pousser case par case)	▬
7 (Snoopy)	☺
8 (balle)	♂
9 (oiseau)	♫

NB : Pour la table ANSI de la page de code 850 voir le site Page de code 850 — Wikipédia (wikipedia.org).

Exemple de code qui affiche un cœur suivi d'un carré :

```
int main ()

{
   printf("%c %c", 0x03, 0xDB);  // 0x pour le codage hexadécimal
   return 0;
}
```

Exemple de texte affiché en couleur en langage C :

Si vous souhaitez de la couleur dans votre texte, téléchargez l'exemple de code donné dans le pdf « ANNEXE EXEMPLE DE TEXTE AFFICHE EN COULEUR EN LANGAGE C » dans le lien suivant **https://drive.google.com/file/d/1EoYYyhrhqU_ffBJO0INkEf4nEavlYPfu/view?usp=sharing**.

- Le déplacement du personnage se fera case par case (gérer les collisions) manuellement en utilisant 4 touches du clavier, sans avoir besoin d'appuyer sur ENTER à chaque fois (indiquez ces touches de déplacement pour guider l'utilisateur).

Exemple de manipulation du curseur à l'écran sous Windows en langage C :

Si vous souhaitez déplacer le personnage avec le curseur à l'écran, téléchargez l'exemple de code donné dans le pdf « ANNEXE EXEMPLE DE MANIPULATION DU CURSEUR SUR WINDOWS EN LANGAGE C» dans le lien suivant **https://drive.google.com/file/d/18fvpcEoQ2U13pNM99CGnzUxqKY2hwVtb/view?usp=sharing**.

- Le joueur peut mettre le jeu en « pause » ou annuler cette « pause » à tout moment.
- L'utilisateur peut revenir au menu et sauvegarder son niveau de jeu en cours : le niveau, le chronomètre et le plateau avec les identifiants associés aux éléments du décor. Il peut accéder à un niveau directement par un mot de passe.
- Quand on termine un niveau (c'est-à-dire que Snoopy a récupéré les 4 oiseaux), un score est établi en fonction du temps mis et du niveau, puis on charge le niveau suivant et on affiche son code d'accès direct. Les scores sont additionnés de niveau en niveau.

Pour vous aider, voici une liste possible de fonctionnalités (sous-programmes) pour gérer le jeu de base :

- Charger un niveau à partir d'un fichier texte
- Vérifier si le niveau est résolu
- Déplacer Snoopy, en tenant compte du type de bloc
- Afficher à l'écran le plateau de jeu

Et n'oubliez pas de tester que vos niveaux sont jouables à l'aide du clavier…

2) Un niveau par membre de l'équipe

Afin d'impliquer équitablement tous les membres de l'équipe de développeur, il vous est demandé de créer au moins un niveau par membre de l'équipe. Les niveaux devant être de difficulté croissante, réfléchissez bien aux compétences de chacun.
Lors de la soutenance, chacun devra pouvoir expliquer son niveau et répondre aux questions spécifiques sur le code posé par le jury.

Versioning de votre projet : GIT

Vous utiliserez l'outil de **versioning GIT** suivant pour créer votre compte GitHub classroom dans le lien suivant : ing3-paris-classroom-2023-2024 (github.com).

Pour plus d'informations, consultez la page BoostCamp Cours : Projet Informatique ING3 Nvx, Section : GitHub classroom : pour le versioning du code en équipe (omneseducation.com)

Lors de la soutenance, vous pourrez être amené à montrer les différentes phases de développement de votre projet (les branches des "versions" du projet). Vous devez aussi être en mesure de montrer "qui a fait quoi" dans le projet.

Grâce au versioning, vous n'aurez plus de problèmes et d'excuses du type :

- c'est mon camarade qui a tout le projet, je n'ai pas pu corriger les erreurs...
- mon disque dur est mort la veille de la soutenance...
- on m'a volé mon ordinateur le jour de la soutenance...
- j'ai renversé du liquide sur mon clavier...
- je ne comprends pas, mon code a été écrasé ? et je n'ai pas fait de backups...
- ...

Ça sent le vécu…

Planning et organisation du travail

Le planning, sujet et évaluation de trouve sur la page BoostCamp Cours : Projet Informatique ING3 Nvx, Section : Projet : Planning, sujet et évaluation (omneseducation.com).

Période de réalisation du projet :
De la semaine du 23 octobre à la semaine du 20 novembre 2023 incluse.

Équipes : 3 ou 4 dans un même groupe de TD (en raison des soutenances).

Evaluation : La soutenance sera d'environ 20 minutes par équipe, incluant une présentation PowerPoint pour la conception et une démonstration du projet, en respect du **cahier des charges demandé ci-dessus, ainsi que des contraintes et consignes ci-dessous**.

Travail demandé, contraintes & consignes (dans le fichier powerpoint)

1- Analyse et conception générale.

- A partir du cahier des charges (CDC) : extraire les données pertinentes, si besoin les regrouper (structures, tableaux), spécifier et caractériser les fonctionnalités (sous-programmes) nécessaires, en déduire les principaux algorithmes de ces fonctionnalités, …
- Interface Homme-Machine : lister les choix à offrir au démarrage, déterminer l'organisation et le contenu de l'écran de jeu, …
- Répartir les tâches au sein de l'équipe.

2- Analyse et conception détaillée

- A partir de vos algorithmes, déduits des fonctionnalités, lister les prototypes de toutes les sous-programmes requis en précisant ses paramètres d'entrée et de sortie.
- Réaliser progressivement une maquette du jeu en testant au fur et à mesure du codage et en tenant compte des différents scénarios. Ajouter des copies d'écran à votre PowerPoint.

3- Développement dans le langage C (codage, tests).

Entre autres critères de qualité, le programme final devra être entièrement commenté pour être très facilement adaptable par tout autre développeur (exemples : changement des valeurs d'initialisation, changement des caractères et couleurs d'affichage, …).

Dans le langage **C**, implémenter le jeu en respect de votre analyse des étapes précédentes : organisation modulaire multi-fichiers (avec headers(s) .h et fichier(s) source(s) .c), commenter les prototypes des sous-programmes (dans le header en précisant les paramètres IN/OUT), vos sous-programmes, votre programme principal.

4- Critères de notation.

Vous devez réaliser l'ensemble du cahier des charges du « jeu à développer ». Le langage de programmation doit être le langage C... **Votre code devra être modulaire, respecter l'interface du jeu en mode console et bien commenté !**

Un code qui ne compile pas ou qui plante au démarrage ne vaut pas plus de 10/20. Tester donc votre programme avant de le déposer sur BoostCamp...

Votre travail sera jugé sur les critères suivants :

- Le respect rigoureux du cahier de charge (CDC) du jeu énoncé précédemment
- La modularité de votre conception et donc de votre code
- La bonne répartition des tâches entre les membres de l'équipe
- L'intérêt, l'originalité, la jouabilité et toutes les caractéristiques que vous prendrez soin de mettre en avant lors de la soutenance.

5- Deadlines du livrable à déposer sur BoostCamp.

Les dates et toutes les consignes sur les livrables à déposer sont spécifiées dans la section de la page BoostCamp Cours : Projet Informatique ING3 Nvx, Section : Livrables à déposer : consignes à respecter (omneseducation.com) :

- Le PowerPoint et le code entier du jeu seront à rendre au plus tard le dimanche 26/11/2023 sur le lien Rendu du livrable de la version finale : PowerPoint et code entier du jeu deadline le dimanche 26/11/2023 de la section de la page BoostCamp ci-dessus.