

Programmation C

PROJET C

Snoopy

Réalisé par :

KARROUM Houssam (BDCC1)



Programmation C

PROJET C

Snoopy

Réalisé par :

KARROUM Houssam (BDCC1)

[Introduction générale 3](#_Toc25571)

[Chapitre I :  Conception 6](#_Toc23542)

[1. Fonctionnalités du Jeu 6](#_Toc27640)

[2. Représentation d'un Niveau 6](#_Toc7853)

[3. Déplacements et Actions 7](#_Toc15372)

[4. Conditions de Victoire/Défaite 7](#_Toc27242)

[5. Gestion du Temps et des Scores 7](#_Toc5908)

[6. Sauvegarde et Chargement 7](#_Toc18486)

[7. Mots de Passe et Mode "Pause" 7](#_Toc13341)

[8. Diagrammes 8](#_Toc17524)

[8.1. Diagramme de Cas d'Utilisation Global 8](#_Toc24952)

[8.2. Diagramme de Séquence Global 9](#_Toc449)

[9. Les classes 10](#_Toc22173)

[10. Graphe d’appel 11](#_Toc9172)

[11. Analyse Chronologique et Descendante 1](#_Toc20986)

[11.3. ACD du menu 1](#_Toc23193)

[11.4. ACD de la gestion de transition des niveaux 1](#_Toc19000)

[11.5. ACD de niveau 2](#_Toc22227)

[Chapitre II : Réalisation du mini projet 3](#_Toc30409)

[1. Implimentation 3](#_Toc32003)

[Bibliographies 7](#_Toc20917)

Figure 1 : Diagramme de Cas d'Utilisation Global 8

Figure 2 : Diagramme de Séquence Global 9

Figure 3 : Diagramme de classe 10

Figure 4 : Graphe d'appel 12

Figure 5 : partie telechargement de jeu 3

Figure 6 : partie de controlls 3

Figure 7 : main page de jeu 4

Figure 8 : main menu de jeu 4

Figure 9 : le jeu 5

Figure 10 : Game over screen 6

Figure 11 : You win screen 6

# Introduction générale

Le projet a récemment évolué avec le développement d'une version graphique, une étape que j'ai entreprise après une évaluation approfondie. Initialement axé sur le langage C et conçu pour être joué en mode console, le jeu a gagné en complexité et en esthétique grâce à l'ajout d'éléments graphiques. Cette transition vers une interface visuelle s'est effectuée de manière délibérée, intervenant après une phase d'examen critique de la version existante. Cela témoigne de l'engagement continu à améliorer et à affiner le projet, tout en maintenant une base solide dans le langage C.

# Conception

Le projet consiste en la conception et la réalisation d'un jeu en langage C, inspiré d'un jeu classique.

Le jeu sera basé sur un mode console, sans graphique, et comportera plusieurs niveaux de difficulté croissante.

## Fonctionnalités du Jeu

Le jeu comprendra les fonctionnalités suivantes :

Niveaux : Chaque membre de l'équipe de codeurs développera un niveau, totalisant 4 niveaux pour une équipe de 4 membres.

Difficulté croissante : Chaque nouveau niveau introduira un bloc d'un nouveau genre.

Gestion des scores : Les scores seront calculés en fonction du temps restant pour chaque niveau.

Gestion des vies : Le joueur commencera avec 3 vies et perdra une vie en cas de dépassement du temps imparti pour un niveau.

Menu principal : Les options suivantes seront disponibles dans le menu principal :

* Règles du jeu
* Lancer un nouveau jeu à partir du niveau 1
* Charger une partie
* Lancer directement un niveau via son mot de passe
* Scores
* Quitter

## Représentation d'un Niveau

Chaque niveau sera représenté par une matrice rectangulaire de 10 lignes par 20 colonnes. Les éléments de la matrice incluront :

* Snoopy (le personnage)
* La balle
* Les oiseaux à récupérer
* Les blocs poussables
* Les blocs cassables
* Les blocs piégés
* Autres blocs facultatifs donnant des points bonus

## Déplacements et Actions

* **Déplacement de Snoopy :** Il se fera dans les 4 directions classiques et d'une seule case à la fois.
* **Mouvement de la balle :** La balle se déplacera en diagonale, rebondissant sur les murs. Elle tue Snoopy au contact.
* **Gestion des objets :** Différents types de blocs auront des propriétés spécifiques (poussable, cassable, piégé).

## Conditions de Victoire/Défaite

* **Victoire :** Récupérer les 4 oiseaux pour passer au niveau suivant.
* **Défaite :** Perdre toutes les vies entraîne un écran de GameOver et retour au menu principal.

## Gestion du Temps et des Scores

* Chaque niveau a un timer de 120 secondes.
* La gestion des scores se fait en fonction du temps restant.

## Sauvegarde et Chargement

* Le joueur peut sauvegarder à tout moment, stockant la position de Snoopy, la balle, les éléments du décor, le temps restant, le nombre de vies et le score courant.
* Chargement via le menu principal en spécifiant le fichier de sauvegarde.

## Mots de Passe et Mode "Pause"

* Chaque niveau est accessible par un mot de passe unique.
* Le jeu peut être mis en pause, figeant la balle, Snoopy et le timer.

## Diagrammes

### Diagramme de Cas d'Utilisation Global

Figure 1 : Diagramme de Cas d'Utilisation Global

Le diagramme de cas d'utilisation global représente les différentes actions possibles dans lesystème.

**Acteurs :** Joueur

**Cas d'Utilisation :**

* Afficher les règles du jeu
* Lancer un nouveau jeu
* Charger une partie
* Lancer un niveau via un mot de passe
* Afficher les scores
* Quitter le jeu
* Déplacer Snoopy
* Casser un bloc
* Pousser un bloc
* Mettre en pause le jeu

### Diagramme de Séquence Global

Figure 2 : Diagramme de Séquence Global

Le diagramme de séquence global illustrera les interactions entre les différents acteurs et les systèmes du jeu. Voici une description textuelle :

**Acteurs :** Joueur **,** Système de Jeu

**Séquence :**

* Le joueur lance le jeu.
* Le système affiche le menu principal.
* Le joueur choisit une option du menu.
* Le système exécute l'action correspondante (nouveau jeu, charger partie, etc.).
* Le système met à jour l'état du jeu.
* Le joueur interagit avec le jeu (déplacement, actions).
* Le système met à jour en temps réel le déroulement du jeu.
* Répéter les étapes 6 et 7 jusqu'à la victoire ou la défaite.
* Le système affiche l'écran de fin de partie.

## Les classes

Figure 3 : Diagramme de classe

**Vecteur2D** contient des coordonnées cartésiennes entières, offrant la possibilité d'effectuer divers calculs et affichages grâce à la surcharge des opérateurs arithmétiques et externes.

**Dessinable** est une classe abstraite comprenant uniquement la méthode virtuelle pure "dessine()". À l'origine, nous avions envisagé son utilisation pour regrouper les bibliothèques d'images et de sons pour Allegro en utilisant un pointeur sur une structure. Cependant, nous avons réalisé que cette structure serait constamment dupliquée pour éviter d'être supprimée, entraînant une consommation excessive de mémoire. Nous avons ensuite exploré l'idée d'utiliser des pointeurs intelligents boost::shared\_ptr, mais après de nombreuses recherches et tentatives, leur implémentation s'est révélée trop complexe.

**Snoopy** est le personnage principal du jeu, doté de méthodes spécifiques à ses actions.

**Case** représente une case du plateau de jeu avec des coordonnées, un type (défini par des macros dans config.h) et des méthodes permettant de la déplacer, de la modifier et de l'afficher.

**Balle** possède une position et une vitesse en tant que réels, permettant un déplacement lent en fonction des timers.

**Plateau** encapsule les éléments du jeu tels que Snoopy, un tableau de cases et un tableau de balles. Cette classe coordonne leurs interactions. Sa fonction "dessine" appelle les fonctions "dessine" des éléments qui la composent, sans avoir besoin de savoir si le jeu est en mode graphique ou console.

**Jeu** gère la boucle de jeu, encapsulant ainsi tous les autres éléments mentionnés ci-dessus, et répond aux commandes de l'utilisateur.

**Partie** coordonne la transition entre les niveaux, le chargement et la sauvegarde des parties.

**Menu** morcelle le fonctionnement d'un .

**Timer** est un élément essentiel pour le moteur de jeu, permettant de mesurer une durée de manière unique et stable. Les timers sont utilisés dans les boucles de jeu pour mesurer le temps, garantissant ainsi que le jeu se déroule toujours à la même vitesse

## Graphe d’appel

Voici le graphe d’appel du constructeur de la classe menu qui est appelé dans le main. C’est ce constructeur qui gère ensuite l’appel des autres fonctions.

Seules les fonctions globales et méthodes publiques des classes sont représen-

tées.

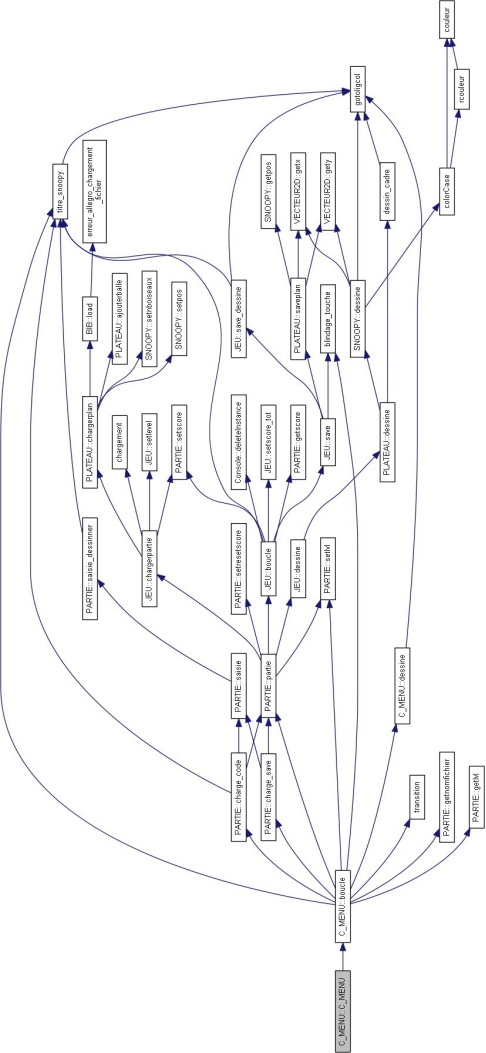


Figure 4 : Graphe d'appel

## Analyse Chronologique et Descendante

### ACD du menu

1. Initialisation des choix possibles
2. Tant que quit!=1
3. Affiche les choix
4. Affiche le curseur du choix actuellement sélectionné
5. Si appuie de la touche enter
   1. Regarde choix sélectionné
      1. Si nouvelle partie lancer une partie
      2. Si charger partie
         1. Demande du pseudo
         2. Lancement de la partie avec l’adresse du fichier déduite du pseudo
      3. Si mot de passe
         1. Demande du mot de passe
         2. Comparaison avec les codes du jeux
         3. Si concordance lance le niveau choisit
         4. Sinon revient au menu
      4. Si quittez quit=1
6. Quitte niveau

### ACD de la gestion de transition des niveaux

1. Chargement du niveaux avec l’adresse du fichier passer en paramètre
   1. Vérification que le fichier existe sinon renvoi message d’erreur
   2. Si le fichier existe lance lance la boucle de jeu
      1. la boucle se déroule
      2. Renvoi un char selon la façon du joueur pour quittez le jeux (escape,
2. Analyse le char
   1. Si perdu propose de sauvegarder le score
      1. Si jouer veut sauvegarder demande de saisie du pseudo
   2. Si win et si il y a un niveau après lance le niveau suivant sinon propose d
3. Revient au menu

### ACD de niveau

1. Si allegro on lance la musique
2. Initialisation des variables
3. Tant que quit!=’W’ ou ’E’ ou ’X’
   1. Tant que game\_time>0
      1. Recuperation des touches et action selon la touche
      2. Blindage position Snoopy
      3. Déplacement balles
      4. Si la vie du joueur est en dessous de 0
         1. quit=’X’
         2. On affiche l’écran de game over

viii. on décrémente game\_time

* 1. Si on change de secondes, on met à jour les compteurs et les FPS
  2. On gère les collisions
  3. On affiche tout

1. Si le temps est écoulé
   1. On enleve une vie au joueur
2. Si tout les oiseaux sont récupérés
   1. Si on est au dernier niveau
      1. On affiche l’écran de victoire
   2. Sinon on affiche l’écran de victoire et on passe au niveau suivant
3. Si le nombre de vie du joueur est à 0 on le remet au niveau 1
4. On s’occupe des scores
5. On arrete la musique
6. On free les variables
7. On quitte

# Réalisation du mini projet

## Implimentation

Figure 5 : partie telechargement de jeu

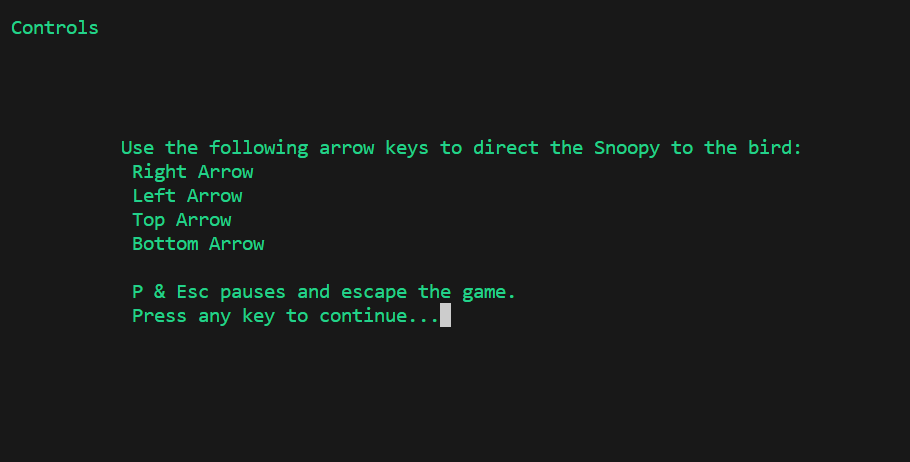


Figure 6 : partie de controlls



Figure 7 : main page de jeu

Figure 8 : main menu de jeu



Figure 9 : le jeu

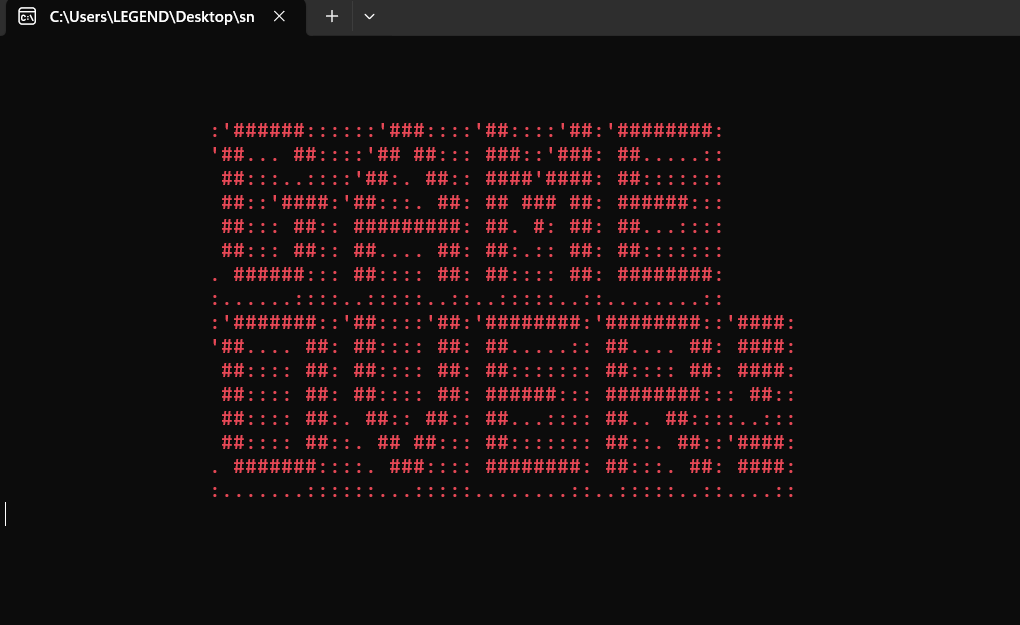


Figure 10 : Game over screen

Figure 11 : You win screen

# Bibliographies

<https://visualgo.net/en/sorting>

<https://en.wikipedia.org/wiki/Interpolation>

<https://en.wikipedia.org/wiki/Approximation>