

RAPPORT DE DEEP

Digital Embedded Electronics Project

Livraison finale

- Nom du projet :

PAC-MAN



- Étudiants :
 - Justine LASNIER-CONFOLANT
 - Maelys LEONES

Complément obligatoire (voir l'annexe correspondante à la fin du rapport)	pts	
B- routage de PCB avec bluepill	3	<input checked="" type="checkbox"/>
B- routage de PCB avec microcontrôleur nu en CMS	4	<input type="checkbox"/>
Compléments non obligatoires (plusieurs choix possibles, voir les annexes correspondantes)		
A- utilisation d'un analyseur logique pour déchiffrer des trames	2	<input type="checkbox"/>
C- design CAO d'un boîtier	2	<input checked="" type="checkbox"/>
D- documentation doxygen du code source	1	<input checked="" type="checkbox"/>
E- mesure de conso et d'énergie selon scénarios	2	<input type="checkbox"/>
F- enregistrement de paramètres en flash	1	<input type="checkbox"/>
G- gestion de version du code source / Git ou SVN	1	<input type="checkbox"/>
T- jeu de tests pour valider une fonctionnalité software ou hardware	1	<input type="checkbox"/>
Nombre de points ciblés au maximum :		6 / 6

S O M M A I R E

1.	Compléments choisis	3
2.	Cahier des charges	3
3.	Manuel d'utilisation	5
4.	Les Ports	9
5.	Description d'un algorithme du programme	10
6.	Structure du programme	12
7.	Tests	15
8.	Cahier de suivi	17
9.	État d'avancement et analyse du projet réalisé	20
10.	Conclusion	21
11.	Bibliographie	22

1. Compléments choisis

Les compléments que nous avons choisis sont listés sur la première page de ce rapport. Les détails de ces compléments sont décrits dans les Annexes B, C et D à la fin de ce compte-rendu.

2. Cahier des charges

2.1. Contexte client

Notre client, un passionné de jeux vidéo rétro, exprime le désir de concevoir une adaptation personnalisée du célèbre jeu Pac-Man, matérialisée sous la forme d'un dispositif interactif. Cette création vise à immerger les utilisateurs dans une expérience rétro authentique en fusionnant habilement des composants tangibles tels qu'une matrice LED, des boutons pousoirs et un écran TFT. L'objectif de cette démarche est de réveiller et de retranscrire de manière fidèle l'atmosphère propre aux jeux d'arcade emblématiques de l'époque.

Le choix spécifique de la matrice LED s'explique par la volonté de saisir de manière précise l'esthétique distinctive de ce jeu rétro. Dans un monde où l'évolution du jeu vidéo s'est largement déroulée sur écrans, ce dispositif vise à rappeler l'importance des éléments tangibles et physiques qui caractérisent l'expérience de jeu autrefois. Ainsi, au-delà de la simple nostalgie, cette approche vise à révéler et à célébrer la richesse sensorielle des jeux d'arcade, rappelant à tous que l'authenticité de l'expérience va au-delà de l'évolution technologique des écrans.

2.2. Spécifications détaillées du projet

- Contrainte de délai :

Le projet est soumis à un délai strict, le client ayant expressément souhaité sa pleine opérationnalité d'ici le *17 janvier 2024*. Respecter cette échéance est impératif pour répondre aux attentes du client et assurer une mise en service dans les délais convenus.

- Contraintes techniques :

A. Boutons Pousoirs

Les boutons pousoirs doivent être minutieusement configurés pour assurer une interaction fiable et réactive avec le logiciel embarqué. La réponse aux commandes de l'utilisateur doit être cohérente et instantanée, garantissant ainsi une expérience de jeu sans délai perceptible. De plus, ils doivent être paramétrés de façon à offrir une utilisation aussi instinctive que possible pour l'utilisateur.

B. Structure Modulaire du Code Source

Le code source du logiciel embarqué doit être organisé de manière modulaire, favorisant ainsi les mises à jour ultérieures et l'intégration de fonctionnalités supplémentaires.

C. Robustesse de la Connectivité

La connectivité entre les différents composants du dispositif, y compris la communication entre le logiciel embarqué et la matrice LED, doit être mise en œuvre de manière robuste et fiable. Cette stabilité est cruciale pour garantir un fonctionnement correct et sans faille du dispositif.

D. Choix des composants

L'objectif premier est d'assurer une interface de jeu suffisamment grande pour permettre une expérience de jeu agréable. Ceci implique une sélection minutieuse des composants, en particulier de la matrice LED, pour garantir une visualisation claire et détaillée du labyrinthe, des personnages et des éléments interactifs.

- **Contraintes liées les Règles du Jeu :**

A. Changement de Niveau

Dans le jeu initial, le changement de niveau se traduit par une modification du design du labyrinthe. Toutefois, la conception de plusieurs labyrinthes individuels peut poser des défis en termes de temps et de praticité. Afin de respecter les contraintes de délai, une approche créative et efficiente doit être adoptée pour maintenir un défi croissant entre les niveaux sans nécessiter la création de multiples labyrinthes distincts.

B. Comportement des fantômes

Dans le jeu initial, les fantômes se déplacent de manière pré-déterminée, suivant des schémas de mouvement spécifiques. Ces schémas ne sont pas aléatoires, mais ils sont basés sur des algorithmes pré-établis. Cependant, pour répondre aux contraintes de délai et aux exigences des composants dans notre projet, une approche alternative devra être adoptée.

C. Point d'apparition des fantômes

Dans le jeu initial, les fantômes font leur réapparition après avoir été mangés par Pac-Man et au début de chaque niveau, émergeant d'une petite salle positionnée au centre du labyrinthe. Cependant, dans notre cas, en fonction du schéma de déplacement choisi pour les fantômes, cette salle réduite pourrait entraver leur capacité à sortir de cet espace. Afin de garantir une expérience fluide et cohérente, des ajustements pourraient être nécessaires pour adapter le point d'apparition des fantômes à la dynamique particulière de leur mode de déplacement.

Ce cahier des charges formalise la conception et les objectifs du projet DEEP (Digital Embedded Electronics Project), visant la création d'un jeu Pac-Man conforme aux attentes spécifiées par le client. En nous conformant à ces exigences, notre engagement est de produire un dispositif qui allie divertissement et fidélité au jeu classique, tout en observant rigoureusement les contraintes et limites définies.

3. Manuel d'utilisation

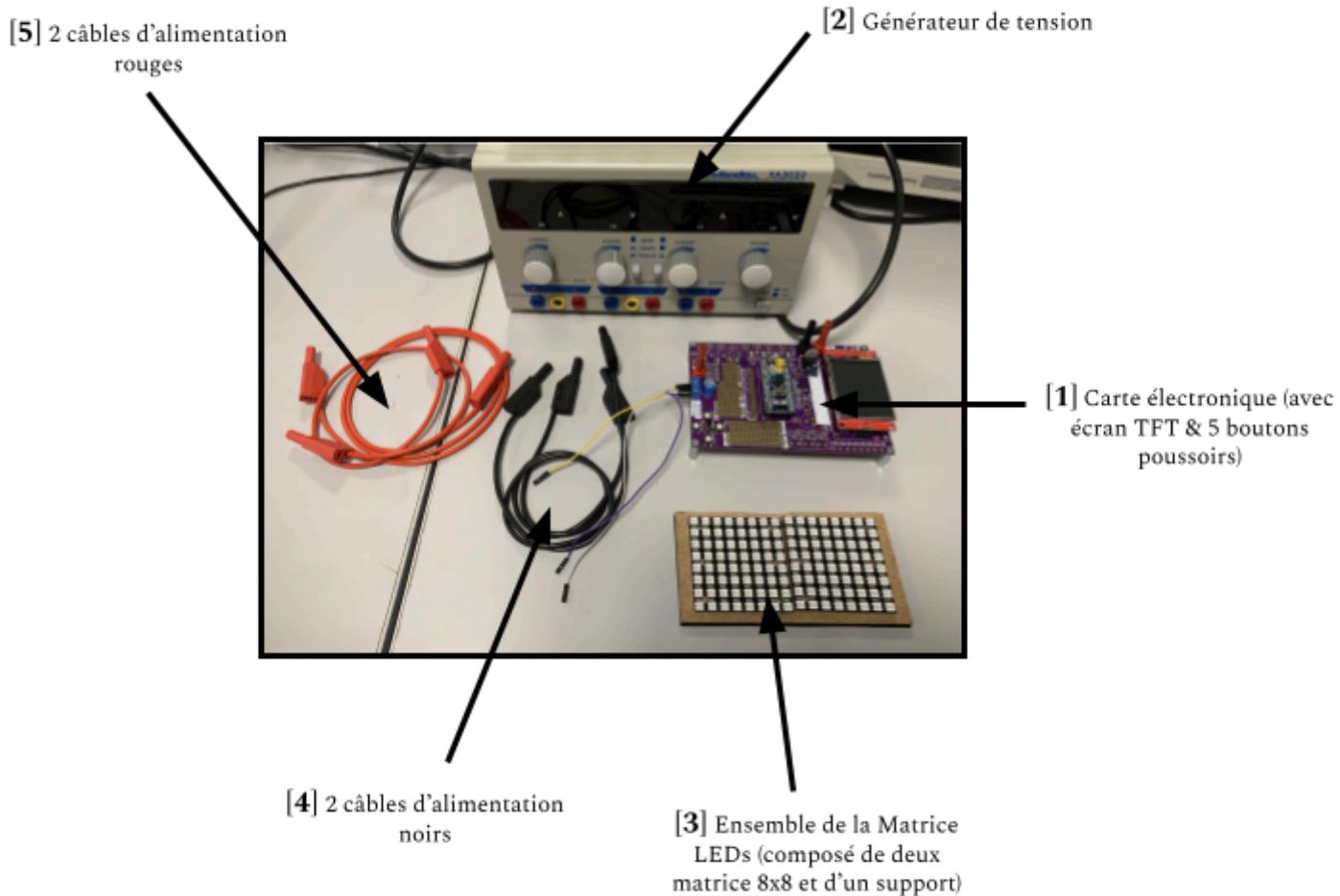
Bienvenue dans l'univers captivant du jeu Pac-Man ! Ce manuel a été élaboré pour vous guider tout au long de l'expérience de jeu et vous permettre de profiter pleinement de cette adaptation unique de ce classique intemporel.

Contenu du Package

Pour fonctionner, l'ensemble du dispositif doit comprendre les éléments suivants (**Figure n°1**):

- [1] Une carte électronique complète ;
- [2] Générateur de tension ;
- [3] Une plaque avec 128 LEDs ;
- [4] 2 câbles d'alimentation noirs ;
- [5] 2 câbles d'alimentation rouges ;

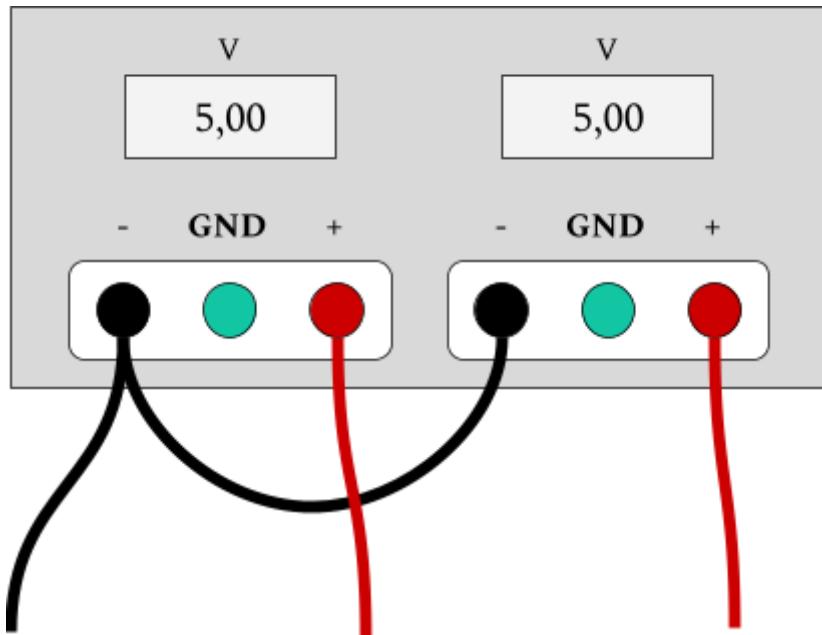
Figure n°1 : Photo des différents éléments du dispositif



Mise en place

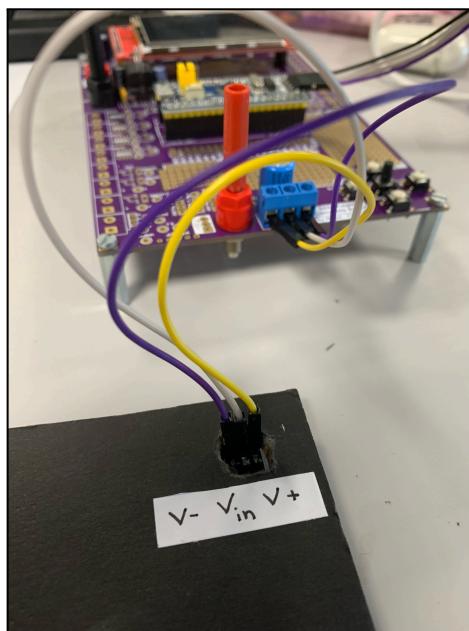
1. Connectez les quatre câbles d'alimentation fournis au générateur de tension tel que présenté dans la **Figure n°2**.

Figure n°2 : Schéma de connexion des câbles d'alimentation au générateur de tension.



2. Réglez la tension sur 5 V sur les deux cadrons du générateur de tension comme illustré sur la **Figure n°2**.
3. Suivez ensuite les étapes de branchements suivantes :

Figure n°3 : Illustration des manipulations à faire pour les branchements



4. Vérifier que l'écran et les LEDs soient allumés.

Utilité des différents composants

- L'écran permet d'abord d'afficher le menu principal, et lorsqu'une partie est lancée ce dernier affiche vos vies, votre score et votre niveau. Il permet également de vous indiquer quand vous avez perdu.
- Les cinq boutons vous permettront de sélectionner une option dans le menu principal et de déplacer Pac-Man au cours de la partie.
- Les deux matrices LEDs sont le support visuel du jeu.

Démarrage du Jeu

Dans le menu principal, vous faites face à deux options qui sont "START" et "QUIT". Vous pouvez sélectionner une des deux options en déplaçant le curseur rond en appuyant sur les boutons du haut et du bas. Vous pouvez appuyer sur le bouton central pour valider l'option que vous souhaitez.

- Sélectionner le bouton "START" affiché sur l'écran pour lancer une partie.
- Sélectionner le bouton "QUIT" affiché sur l'écran pour quitter le jeu.
-

Indications des éléments

- LED **jaune** : Pac-Man {vous}
- LED **orange** : fantôme [Clyde]
- LED **rouge** : fantôme [Blinky]
- LED **rose** : fantôme [Pinky]
- LED **bleu clair** : fantôme [Inky]
- LEDs **bleus foncés** : murs du labyrinthe.
- LEDs **noires** (éteintes) : couloirs.
- LEDs **blanches** : pac-points.

Règles du Jeu

→ Objectif : Le but principal du jeu est de manger tous les pac-points présents dans le labyrinthe sans vous faire toucher par les fantômes. L'objectif à long terme est de passer le plus de niveaux possibles et donc d'avoir le meilleur score possible.

1. Pac-Man : Vous contrôlez Pac-Man qui se déplace dans les couloirs en utilisant les boutons HAUT, BAS, GAUCHE et DROITE.
2. Le labyrinthe : Le jeu se déroule dans un labyrinthe composé de murs que Pac-Man ne peut pas traverser.
3. Les points : Les points que vous devez récupérer se nomment des "pac-points". Manger tous les pac-points permet de terminer le niveau. Au niveau suivant, de nouveaux pac-points apparaîtront.

4. Les fantômes : Le labyrinthe est hanté par quatre fantômes, chacun avec son propre nom : Pinky, Blinky, Clyde et Inky. Ils réapparaissent à leur point de réapparition lorsque le joueur les touche ou qu'ils touchent le joueur.
5. Niveau : Vous commencez avec trois vies et votre objectif est de survivre le plus longtemps possible pour ramasser un maximum de pac-points. Pour passer d'un niveau à l'autre, il faut récupérer tous les pac-points.
6. Vies : Vous commencez avec trois vies et vous perdez une vie si un fantôme vous touche.
7. Score : Votre score augmente de 10 à chaque fois que vous récupérez un pac-point.
8. Fin de partie / Game Over : Lorsque votre nombre de vie tombe à 0, un écran avec un "Game Over" affiché indique que vous avez perdu. Ce dernier affiche votre score final.

4. Les Ports

Pin	rôle
PA0	RouBR (droite)
PA1	RouBU (haut)
PA2	RouBM (milieu)
PA3	RouBD (bas)
PA4	
PA5	TFT SCK
PA6	TFT MISO
PA7	TFT MOSI
PA8	TOUCH CS
PA9	TFT DC
PA10	TFT RESET
PA11	TFT CS
PA12	
PA13	(non dispo - SWDIO)
PA14	(non dispo - SWDCLK)
PA15	non dispo sur certaines bluepill
PB0	
PB1	
PB2	(non dispo)
PB3	
PB4	
PB5	RouBL (gauche)
PB6	
PB7	
PB8	
PB9	
PB10	
PB11	
PB12	
PB13	Data envoyer dans la matrice LEDs
PB14	
PB15	
PC13	(LED bluepill)
PC14	(Quartz 32kHz bluepill)
PC15	(Quartz 32kHz bluepill)

5. Description d'un algorithme du programme

Au sein de cette section, il nous incombe de détailler au moins l'une des tâches essentielles de notre programme en employant un formalisme approprié pour représenter un algorithme. Nous avons décidé d'opter pour un pseudo-code clair et explicite afin de décrire le processus de la tâche. De manière complémentaire, nous avons élaboré le diagramme d'activité UML de la fonction choisie, à savoir la fonction `JEU_DeplacementPacMan`. Cette fonction, dédiée à la gestion des déplacements de Pac-Man, est cruciale dans le contexte du jeu. La combinaison du pseudo-code et du diagramme UML vise à offrir une compréhension du fonctionnement de cette tâche.

```
Fonction JEU_DeplacementPacMan (mouvement, grille):
    positionpacman ← TrouverPositionPacMan(qrille)
    nouvelleposition ← positionpacman + mouvement
    isValidMove ← 0

    Si grille[nouvelleposition] == PACPOINTS:
        score ← score + 10
        MENUS_ActualisationJeuEnCours(vies, score, niveau)
        indicePacPointMangé ← TrouverIndicePacPointMange(listePACPOINTS, nouvelleposition)
        SupprimerPacPointMange(listePACPOINTS, indicePacPointsMangé)

    Si grille[nouvelleposition] est un mur:
        isValidMove ← 1

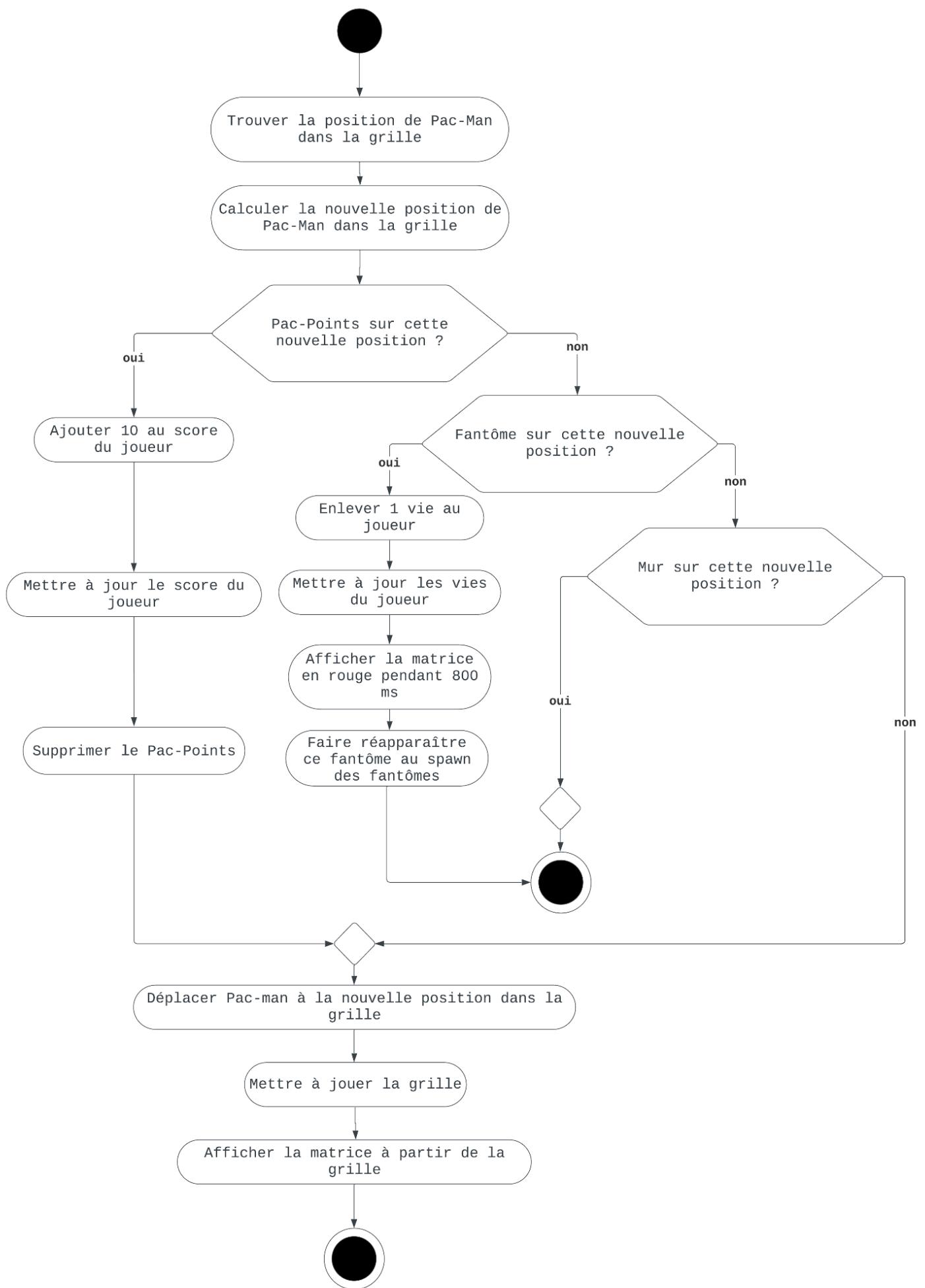
    Si grille[nouvelleposition] est un fantôme:
        isValidMove ← 1
        vies ← vies - 1
        RespawnFantome();
        grille[nouvelleposition] ← COULOIRS
        MENUS_ActualisationJeuEnCours(vies, score, niveau)
        MATRICE_PerteVie()
        MATRICE_Affichage(grille)

    Si isValidMove == 0:
        grille[positionpacman] ← COULOIRS
        grille[nouvelleposition] ← PAC-MAN
        MATRICE_Affichage(grille);
```

Veuillez noter que les fonctions suivantes sont des fonctions hypothétiques qui ne sont pas définies de cette manière dans le programme écrit en C :

- `TrouverPositionPacMan` : retrouve l'indice de pac-man dans la grille, c'est-à-dire sa position dans la matrice, là où il se situe.
- `TrouverIndicePacPointMange` : retrouve l'indice du pac-points en question dans la liste.
- `SupprimerPacPointMange` : enlève ce pac-points de la liste des pac-points à récupérer par le joueur.
- `RespawnFantome` : faire réapparaître le fantôme à un des quatre points de spawn.

Vous trouverez sur la page suivante, le diagramme UML de cette même fonction.



6. Structure du programme

Dans cette section, nous avons pour objectif de recenser exhaustivement tous les fichiers générés dans le cadre de notre projet, en fournissant une description détaillée de chacun d'entre eux ainsi que de leurs principales fonctions. Cette démarche vise à créer une documentation complète et claire de notre travail, facilitant ainsi la compréhension du code et son évolution future. En consignant cette liste détaillée, nous nous assurons de la traçabilité de notre production, ce qui est essentiel pour une gestion efficace du projet.

6.1. Fichier « Matrice.c »

Ce module permet de gérer l'affichage du jeu grâce aux LEDs.

Nom du développeur	Fonction	Description
Justine Lasnier-Confolant	MATRICE_Initialisation	Permet d'allumer les 128 LEDs en bleu pour initialiser les matrices avant que le joueur débute une partie ou quand il passe d'un niveau à l'autre.
	MATRICE_Affichage	Permet d'afficher la grille du jeu sur la matrice.
	MATRICE_PerteVie	Permet d'allumer les 128 LEDs en rouge pendant 800 ms. Cela arrive notamment quand le joueur perd une vie.

6.2. Fichier « Boutons.c »

Ce module est destiné à gérer des boutons connectés à des broches spécifiques sur le microcontrôleur.

Nom du développeur	Fonction	Description
Justine Lasnier-Confolant	BOUTONS_SiPresse	Permet de déterminer l'état de pression de chaque bouton.

6.3 Fichier « Jeu.c »

Ce module permet de gérer les aspects du gameplay de notre jeu. En outre, c'est lui qui crée et modifie la grille du jeu qui est ensuite renvoyé sur la matrice et qui gère également les 3 variables principales du jeu (vies, score et niveau).

Nom du développeur	Fonction	Description
Justine Lasnier-Confolant	JEU_DeplacementPacMan	Permet de déplacer Pac-Man (renvoie la grille après modification, après déplacement).
	JEU_DeplacementFantomes	Permet de déplacer les quatres fantômes de manière aléatoire (renvoie la grille après modification, après déplacements).
	JEU_GetNiveau	Accesseur en lecture de la variable niveau.
	JEU_GetVies	Accesseur en lecture de la variable vies.
	JEU_GetScore	Accesseur en lecture de la variable score.
	JEU_initialisationParametres	La fonction initialise les variables globales du jeu (score, vies et niveau) avec les valeurs fournies en paramètre. De plus, elle initialise la grille avec les valeurs fournies en paramètres également. En résumé, cette fonction facilite la mise en place initiale du jeu en définissant ses paramètres et sa configuration initiale.
	JEU_ChangementDeNiveau	Augmente le niveau de 1.
	JEU_SetListePACPOINTS	Permet de personnaliser les emplacements des pac-points sur la grille de jeu sans avoir à redéfinir la taille de la liste manuellement.
	JEU_CreationGrille	Cette fonction sert à configurer la disposition initiale des éléments du jeu sur la grille (spawn de Pac-Man, spawn des fantômes, pac-points, murs du labyrinthe).

6.4. Fichier « Gestion.c »

Le module encapsule la logique de jeu, la gestion des différents états de celui-ci et des interactions utilisateur pour créer une expérience de jeu.

Nom du développeur	Fonction	Description
Justine Lasnier-Confolant	GESTION_Accueil	Gère le menu principal de notre jeu Pac-Man. C'est dans cette fonction qu'on traite les actions de l'utilisateur quand il navigue entre les options du menu (START & QUIT).
	GESTION_InGame	Gère le déroulement d'une partie de Pac-Man. C'est dans cette fonction qu'on traite les actions du joueur pendant une partie et qu'on les répercute sur l'écran TFT et les matrices LEDs.
	GESTION_GameOver	Gère la fin de partie, puis le retour au menu principal.
	GESTION_Quit	Gère ce qu'il se passe lorsque le joueur sélectionne l'option "QUIT" dans le menu principal.
	GESTION_main	Fonction principale, permet le fonctionnement continu du programme en exécutant la tâche associée à l'état actuel de la machine à états, offrant ainsi une structure modulaire pour la gestion des différents aspects du jeu.

6.5. Fichier « Menus.c »

Ce module permet d'afficher les différents menus du jeu sur l'écran TFT. Les différents menus sont détaillés dans le tableau ci-dessous.

Nom du développeur	Fonction	Description
Justine Lasnier-Confolant	MENUS_Principal	Affiche sur l'écran TFT le nom du jeu, un dessin de Pac-Man, les copyrights et deux options : START & QUIT.
	MENUS_ActualisationJeuEnCours	Affiche sur l'écran TFT le score, les vies et le niveau de la partie en cours.
	MENUS_GameOver	Affiche sur l'écran TFT les mots "Game Over" et le score du joueur quand il perd.
	MENUS_Sortie	Affiche "bye".

7. Test

Dans cette section, nous décrivons les tests approfondis que nous avons réalisés pour évaluer les différentes fonctionnalités de notre jeu. Ces tests ont été conçus pour examiner la navigation dans le menu principal, la gestion des entrées utilisateur à l'aide des boutons, et la transition entre les différents états du jeu. L'objectif principal de ce rapport de test est d'obtenir une compréhension claire du stade de développement de notre réalisation. Les résultats des tests nous ont fourni des informations cruciales sur la robustesse et la fiabilité de notre logique de programme, soulignant les domaines où des améliorations pourraient être nécessaires. En résumé, ce rapport de test constitue un outil essentiel pour évaluer la maturité de notre projet, nous permettant de prendre des décisions informées pour son perfectionnement.

n°	Intitulé du test	Description de ce qu'il faut faire pour jouer le test	Observation obtenue et conclusion
1	Test du lancement du jeu après l'appuie sur START.	Lorsque le curseur est sur START, appuyer sur le bouton central.	OK L'écran TFT initialise les variables vies, niveau et score et les affiche sur l'écran TFT. La matrice affiche Pac-Man, le labyrinthe, les fantômes et les pac-points.
2	Test de sortie du jeu après l'appuie sur QUIT.	Lorsque le curseur est sur "QUIT" (le sélectionner en appuyant sur "bas"), appuyer sur le bouton central.	OK L'écran TFT affiche "bye" à l'utilisateur.

3	Test de la collision entre Pac-Man et les murs.	Lors d'une partie, faire avancer Pac-Man dans un mur à l'aide des boutons pousoirs.	OK Pac-man reste figé sur sa position initiale et ne traverse pas le mur.
4	Test de collision entre Pac-Man et les fantômes.	Lors d'une partie, foncer sur un fantôme à l'aide des boutons pousoirs.	OK Le joueur perd une vie donc la matrice s'allume en rouge pendant un court instant et le fantôme réapparaît dans l'un des quatre points de spawn.
5	Test de récupération des pac-points.	Lors d'une partie, déplacer Pac-Man sur un pac-point.	OK L'écran TFT affiche 10 de plus au score du joueur et la LED initialement allumé en blanc ne s'allume plus après le passage du joueur sur cette dernière.
6	Test de changement de niveau.	Récupérer tous les pac-points en se déplaçant sur ces derniers.	OK L'écran s'affiche entièrement en bleu pendant quelques instants puis affiche le niveau n+1 sur l'écran TFT. Fait apparaître Pac-Man et les fantômes à leur point de spawn et fait réapparaître les pac-points. La vitesse des fantômes est augmentée.
7	Test du Game Over.	Se faire toucher ou toucher 3 fois les fantômes.	OK Affiche Game Over et le score sur l'écran TFT puis au bout de quelques secondes, renvoie au menu principal et affiche la matrice en bleu.
8	Test de la non récupération des pac-points par les fantômes.	Lancer une partie.	OK le pac-point est toujours présent après le passage du fantôme.
9	Test du déplacement des fantômes.	Lancer une partie.	~OK Les 4 fantômes se déplacent aléatoirement dans tout le labyrinthe. Cependant certains restent statiques pendant quelques temps le

			temps de trouver un chemin praticable. Cela est normal étant donné le code, mais cela est à souligner pour une potentielle amélioration future.
--	--	--	---

8. Cahier de suivi

Pour chaque date, que ce soit lors de nos séances de travail, pendant ou en dehors des créneaux prévus à l'agenda, nous consignons dans ce cahier de suivi les tâches que nous avons accomplies, les personnes responsables de leur réalisation, les éventuelles difficultés rencontrées, ainsi que l'état d'avancement global du projet. Cette démarche a pour objectif de nous encourager à prendre du recul sur l'évolution de notre projet.

Date	Tâches, réalisateurs ¹	Choses à faire la prochaine fois
25/10/2023	<p>J. & M. : Choix du projet.</p> <p>J. :</p> <ul style="list-style-type: none"> Écriture des sections (livrable intermédiaire) : <ul style="list-style-type: none"> “introduction” ; “cahier des charges” ; “composants utilisés” ; <p>M. :</p> <ul style="list-style-type: none"> Soudure de la plaque 1/2. 	<input type="checkbox"/> Finir de rédiger au maximum le compte-rendu intermédiaire.
27/10/2023	<p>J. :</p> <ul style="list-style-type: none"> Soudure de la plaque 2/2. <p>M. :</p> <ul style="list-style-type: none"> Consultation attentive de documents pertinents, se familiarisant ainsi avec les spécifications techniques et les exigences matérielles du projet. 	<input type="checkbox"/> Schématiser la réalisation de la partie software, faire un plan de réalisation.
08/11/2023	<p>J. :</p> <ul style="list-style-type: none"> Installation du logiciel. Test des composants 2/3 (matrice LED & écran TFT). Développement du module logiciel Boutons.c/h. 	
15/11/2023	<p>J. :</p> <ul style="list-style-type: none"> Test des composants 3/3 (boutons poussoirs) Connecter les 4 matrices LEDs ensemble pour former une matrice 16x16 et tester l'ensemble. <p>M. :</p> <ul style="list-style-type: none"> Mise au propre & soudage du matériel, design du labyrinthe. 	<input type="checkbox"/> Faire un schéma du design du menu d'accueil du joueur.

¹ J. correspond à Justine et M. correspond à Maëlys.

17/11/2023	<p>J. :</p> <ul style="list-style-type: none"> • Développement du module logiciel Menus.c/h. <ul style="list-style-type: none"> ◦ création du Menu Principal et du Menu de Sortie (<code>MENUS_Principal</code>, <code>MENUS_Sortie</code>) ◦ création d'une fonction pour actualiser les variables affichées au joueur → <code>MENUS_ActualisationJeuEnCours</code>. • Implémentation des variables score, niveau et vies et des fonctions associés dans JEU.c.h (<code>JEU_initialisationParametres</code>, <code>JEU_ChangementDeNiveau</code>, <code>JEU_GetScore</code>, <code>JEU_GetNiveau</code>, <code>JEU_GetVies</code>). <p>M. :</p> <ul style="list-style-type: none"> • Mise au propre & soudage du matériel, design du labyrinthe. 	<input type="checkbox"/> Réaliser le design du labyrinthe sur Excel et faire une liste des cases de la matrice qui seront notées comme étant des murs dans le code.
24/11/2023	<p>J. :</p> <ul style="list-style-type: none"> • Développement du module logiciel Jeu.c/h. : <ul style="list-style-type: none"> ◦ fonction <code>JEU_DeplacementPacMan</code> • Développement du module logiciel Matrice.c/h. : <ul style="list-style-type: none"> ◦ fonction <code>MATRICE_Initialisation</code> ◦ fonction <code>MATRICE_Affichage</code> <p>M. :</p> <ul style="list-style-type: none"> • Démarrage du projet sur Altium, schématisation du code. 	<input type="checkbox"/> Schématiser l'organisation du code. <input type="checkbox"/> Faire un brouillon pour schématiser la machine à états principal du code.
29/11/2023	<p>J. :</p> <ul style="list-style-type: none"> • Problème rencontré : Pac-Man peut manger les murs → modification de la fonction <code>JEU_DeplacementPacMan</code>. • Création des pac-points : développement du module logiciel Jeu.c/h (<code>JEU_SetListePACPOINTS</code>). • Développement du module principal faisant tourner le jeu Gestion.c/h. <p>M. :</p> <ul style="list-style-type: none"> • Suite du travail sur le complément Altium, optimisation des schémas. 	<input type="checkbox"/> Lister toutes les paires de LEDs qui sont à la frontière entre deux matrices et par lesquelles Pac-Man peut passer (elles ne se suivent pas dans la grille).
01/12/2023	<p>J. :</p> <ul style="list-style-type: none"> • Modification de <code>JEU_DeplacementPacMan</code> pour que le jeu fonctionne entre les quatre matrices. • Modifications effectuées dans le module Gestion.c. • Développement de la détection de fin de partie dans le module Gestion.c/h. et création de la fonction <code>JEU_ifLevelDone</code> dans JEU.c/h. • Création du menu de fin de partie → <code>MENUS_GameOver</code>. 	<input type="checkbox"/> Se renseigner sur l'aléatoire en C.

	<p>M. :</p> <ul style="list-style-type: none"> • Revue du schéma de code, optimisation de la gestion des paires de LEDs. 	
06/12/2023	<p>J. :</p> <ul style="list-style-type: none"> • Développement du module logiciel JEU.c/h pour la création des fantômes et la mise en place de leurs déplacements (JEU_DeplacementFantomes) • Modifications effectuées dans le module Gestion.c. • Problème rencontré : Les fantômes peuvent manger les pac-points → modification de la fonction JEU_DeplacementFantomes. <p>M. :</p> <ul style="list-style-type: none"> • Finalisation du travail sur le complément Altium, vérification des connexions matérielles. 	<input type="checkbox"/> Lister l'ensemble des tâches à compléter pour finir le projet. <input type="checkbox"/> Avancer dans la rédaction du compte-rendu intermédiaire.
08/12/2023	<p>J. & M. :</p> <ul style="list-style-type: none"> • Finalisation du compte-rendu intermédiaire. <p>M. :</p> <ul style="list-style-type: none"> • Revue des spécifications matérielles, identification des composants restants. 	<input type="checkbox"/> Réorganiser le code.
13/12/2023	<p>M. :</p> <ul style="list-style-type: none"> • Tests approfondis des composants, résolution de problèmes potentiels <p>J. :</p> <ul style="list-style-type: none"> • Réécriture du code pour que les fantômes puissent réapparaître à un point de spawn spécifique pour éviter que le joueur se fasse toucher plusieurs fois d'affilée. • Détection des éventuels bugs. <ul style="list-style-type: none"> ○ Les vies du joueur passent en dessous 0 dans un cas spécifique (égal à 255). 	<input type="checkbox"/> Checker le code pour voir dans quelle section du code vient le bug énoncé.
15/12/2023	<p>M. :</p> <ul style="list-style-type: none"> • Optimisation des chemins de connexion sur le PCB. <p>J. :</p> <ul style="list-style-type: none"> • Réglage du bug des vies. • Ajout d'une fonctionnalité permettant d'afficher la matrice LED en rouge pendant un court instant lorsque le joueur perd une vie (afin que ce soit plus visible). [MATRICE_PerteVie]. 	
20/12/2023	<p>J. :</p> <ul style="list-style-type: none"> • Réorganisation du code. 	

	M. :	
10/01/2023	J. : <ul style="list-style-type: none"> Rédaction du compte-rendu (parties ‘cahier des charges’, ‘manuel d’utilisation’, ‘ports’ et ‘description d’un algorithme du programme’). M. :	
12/01/2023	M. : <ul style="list-style-type: none"> Revue de la documentation, préparation des éléments visuels pour la présentation. Création d’une petite vidéo. J. : <ul style="list-style-type: none"> Création du Powerpoint pour la présentation. Finalisation du compte-rendu. Derniers tests du jeu. 	<input type="checkbox"/> S’entraîner à l’oral pour la soutenance de 8 min.

9. État d'avancement et analyse du projet réalisé

Accomplissements et Fonctionnalités Opérationnelles

Dans cette phase avancée du projet, plusieurs aspects ont été accomplis avec succès. Le déplacement du personnage principal et des fantômes, la gestion des points, des vies et des niveaux, ainsi que l’interaction avec la matrice LED et l’écran TFT sont opérationnels.

Opportunités d'Amélioration

Cependant, une réflexion sur le projet révèle quelques opportunités d'amélioration. Si nous devions refaire le projet, nous accorderions une attention plus poussée à l'optimisation du code pour maximiser l'efficacité des algorithmes de déplacement des fantômes. Actuellement, l'algorithme choisit un déplacement aléatoire entre “haut”, “bas”, “gauche”, “droite”, ce qui peut entraîner des périodes d'immobilité lorsque le déplacement est bloqué par un obstacle.

De plus, la gestion des déplacements entre les deux matrices pourrait bénéficier d'une approche plus élaborée, car elle repose actuellement sur des structures conditionnelles manuelles. Cette méthode pourrait devenir impraticable avec un nombre plus élevé de matrices LEDs.

En ce qui concerne le gameplay lui-même, des améliorations significatives pourraient être apportées, notamment en optimisant les déplacements des fantômes pour les rendre plus intelligents. L'idée serait d'explorer des stratégies plus élaborées pour les mouvements des ennemis, offrant ainsi une expérience de jeu plus captivante. Une autre piste d'amélioration serait l'introduction de divers modes de jeu, accessibles via des options dans le menu principal. De même, l'ajout d'éléments emblématiques tels que des fruits et des super pac-gommes, comme dans le jeu original, pourrait enrichir l'expérience en introduisant de nouveaux éléments de gameplay. Les super pac-gommes, par exemple, pourraient être

intégrées pour rendre les fantômes vulnérables. Enfin, la diversification des labyrinthes pourrait également être envisagée. L'introduction de plusieurs labyrinthes au jeu pourrait offrir une variété stimulante, permettant aux joueurs de relever différents défis et de maintenir un niveau élevé d'engagement tout au long de l'expérience de jeu.

Tests et Détection des Problèmes

La mise en place d'une batterie de tests à chaque création d'une nouvelle fonction aurait permis d'identifier des points critiques plus tôt dans le processus de développement, facilitant ainsi la résolution précoce de certains problèmes. Par exemple, des problèmes tels que le déclenchement incorrect de l'écran de fin de partie lorsque Pac-Man avance sur un fantôme, ou la disparition d'un fantôme lorsqu'il se retrouve entièrement bloqué, auraient pu être détectés plus rapidement.

Limites des Composants

En ce qui concerne les limites gênantes au sujet des composants, connecter 4 matrices à la suite s'est révélé problématique. En effet, lors de la tentative de connexion de quatre matrices à la suite, une perte de tension significative a été constatée à mesure que l'on avançait de la première à la quatrième matrice. Cette diminution de tension a fini par rendre la quatrième matrice inutilisable, affectant ainsi la qualité de l'affichage global. Cette perte de tension peut être attribuée à la résistance accrue due à la longueur croissante du circuit. Par manque de temps pour trouver une solution convenable, nous avons alors décidé d'utiliser seulement 2 matrices au lieu de 4. De plus, l'écran TFT présente un léger délai d'actualisation, introduisant de la latence dans les déplacements de Pac-Man lors de la récupération d'un pac-point puisqu'il faut actualiser le score sur l'écran TFT, ce qui affecte la fluidité du jeu.

Globalement, cette phase d'analyse met en lumière des opportunités d'amélioration importantes, tout en soulignant les aspects réussis du projet. Cela offre des enseignements précieux pour des projets futurs, mettant l'accent sur l'importance de l'optimisation, du débogage précoce et d'une gestion proactive des limites.

10. Conclusion

Ce projet de développement du jeu Pac-Man sur une matrice LED a été une expérience enrichissante. La création d'un jeu classique tout en respectant des contraintes matérielles et temporelles a constitué un défi stimulant. La gestion du déplacement du personnage principal (Pac-Man) et des fantômes, la mise en place du labyrinthe, ainsi que la gestion des points, des vies et des niveaux ont été des aspects cruciaux du développement.

Le respect des contraintes de délai a exigé une planification minutieuse et une exécution efficiente du code. La modularité du code source a été une clé pour faciliter les mises à jour futures et l'ajout de fonctionnalités supplémentaires.

De plus, adapter les règles du jeu classique à une plateforme matérielle spécifique a nécessité une compréhension approfondie des mécanismes du jeu Pac-Man.

En résumé, ce projet a permis d'appliquer des concepts théoriques à une réalisation concrète, de

surmonter des défis techniques et de développer des compétences en ingénierie logicielle et matérielle. Cela témoigne du potentiel de la programmation et de l'électronique pour créer des expériences interactives, tout en soulignant l'importance de la gestion efficace des contraintes dans le processus de développement.

11. Bibliographie

[1] [Wikipédia](#), Pac-Man, 16 septembre 2023.

Annexe B – Complément « design de PCB »

Auto-validez votre design en vérifiant les points suivants :

Nous penserons à joindre à l' archive zip livrable : notre fichier SchDoc, notre fichier PcbDoc ainsi que le PDF généré par Altium contenant schéma, vue des couches de routage et vue 3D.	1
---	---

Les connecteurs sont accessibles	1
Un plan de masse est présent sur chaque couche de cuivre	1
Les GND sont correctement tous reliés par des pistes (même si le plan de masse les regroupe ensuite).	1
Le DRC (Design Rules Check) passe sans aucune erreur	1
Présence d'un connecteur UART (Rx + Tx + GND), parce que c'est toujours utile	1
Si le microcontrôleur est placé en CMS : présence d'un connecteur de programmation (SWDIO + SWDCLK + GND)	1

Validation PCB avant fabrication.pdf :

Noms des étudiants et site (Angers/Vélizy/Dijon) visibles sur le TOP.	1
Carte aux dimensions raisonnables (pas trop d'espace libre !)	1
Pistes \geq 10 mils ; et au moins 20 mils lorsque c'est possible	1
Clearance \geq 20mils lorsque c'est possible. (ponctuellement \geq 10mils)	1

Au-delà des critères indiqués ci-dessus, les critères permettant de juger votre réalisation sont les suivants.
Autoévaluez en une note de 0 à 100 votre design :

Un schéma correct	90/100
Qualité du placement (un bon placement garanti un bon routage)	80/100
Qualité du routage	70/100

Le rôle des composants et périphériques que vous avez mentionnés dans un système électronique pourrait être défini comme suit :

1. Microcontrôleur :

- Rôle :

C'est le cerveau du système, responsable de l'exécution des instructions du programme.

- Fonctions :

- Traitement des données.
- Contrôle des entrées/sorties.
- Exécution des algorithmes.
- Gestion des communications.

2. Ecran TFT ILI9341 :

- Rôle dans notre projet :

Affiche les différents menus (Menu de Démarrage, Menu de Sortie, Menu de Jeu, Menu de Game Over). Il permet d'afficher les options « START » et « QUIT » dans le menu principal, et pendant la partie il informe le joueur sur son nombre de vies, son score et le niveau en cours.

- Principe de fonctionnement :

- Contrôlé par un microcontrôleur, il reçoit des commandes pour afficher des images en convertissant les signaux électriques en variations de couleur sur l'écran, offrant ainsi une visualisation graphique.

3. Matrice LED(s) WS2812 :

- Rôle dans notre projet :

Représente l'affichage visuel du jeu. On y voit les murs, Pac Man, les fantômes, les points de score.

- Principe de fonctionnement :

- Elles sont composées de 64 LEDs disposées en une grille 8x8. La matrice est basée sur la technologie de multiplexage, où les LEDs sont activées séquentiellement pour créer l'affichage global. Ce processus est effectué très rapidement et de manière cyclique, créant l'illusion que toutes les LEDs sont allumées simultanément. En ajustant la séquence d'activation, on peut contrôler individuellement chaque LED pour afficher des motifs spécifiques

3. Boutons-poussoirs :

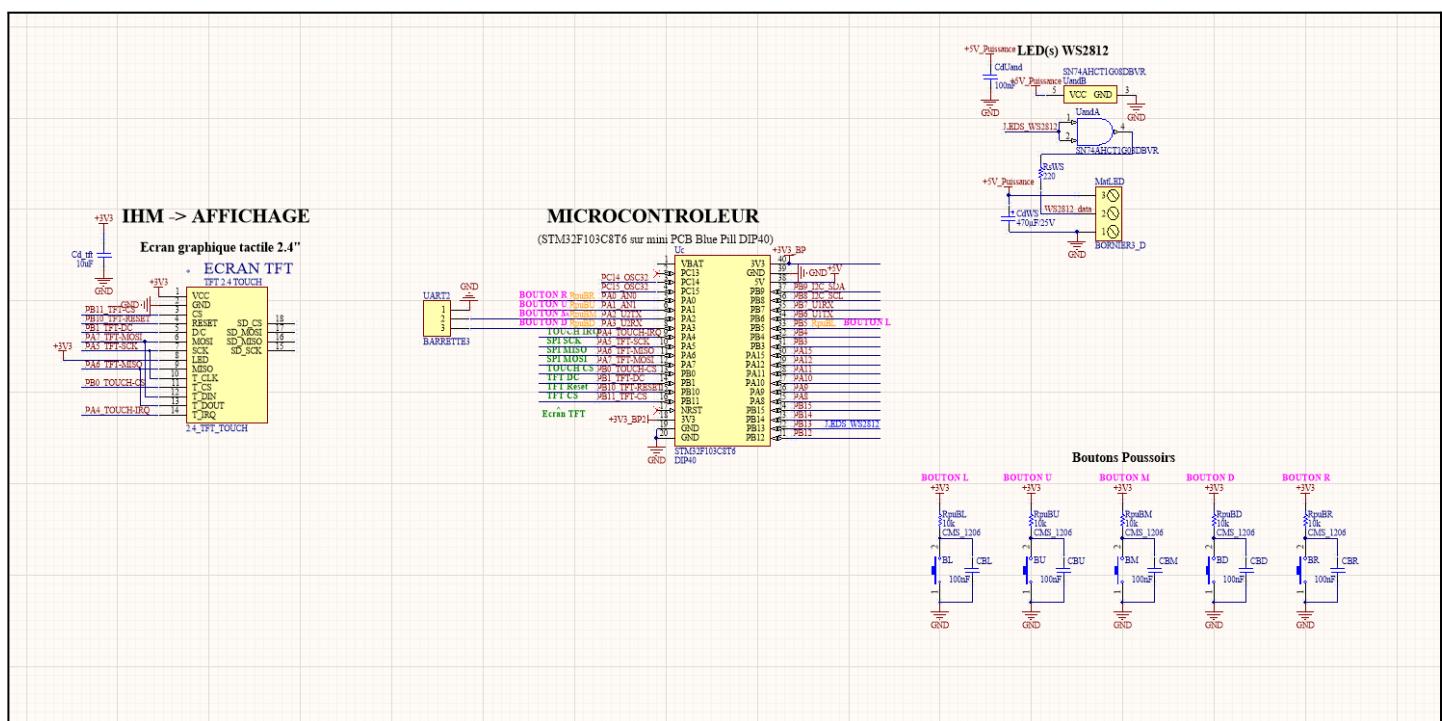
- Rôle dans notre projet :

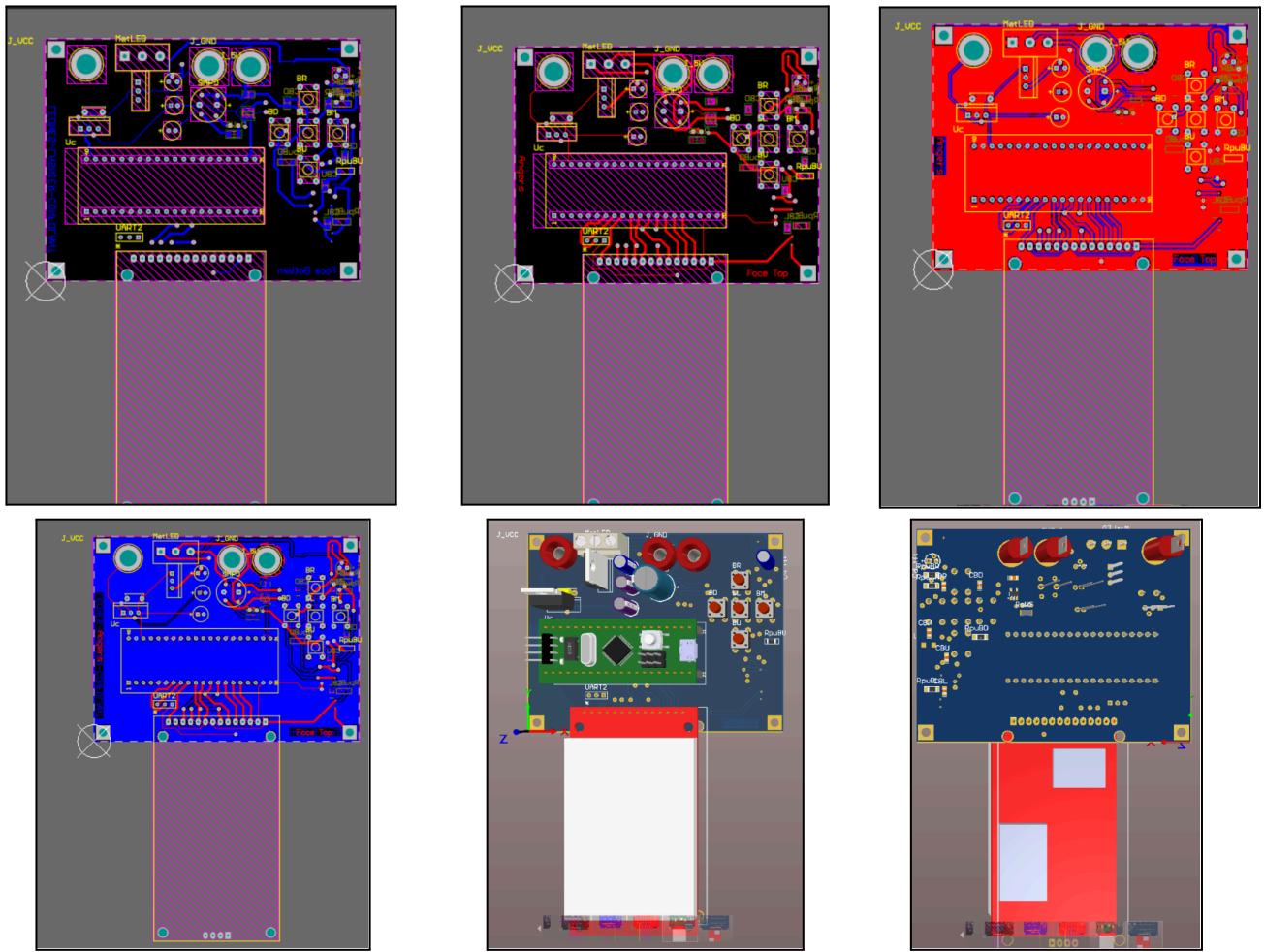
4 d'entre eux permettent à l'utilisateur de déplacer Pac Man de gauche à droite et de haut en bas. Le bouton du milieu permet de sélectionner une option dans le menu de démarrage du jeu.

- Principe de fonctionnement :

- Lorsqu'on appuie sur le bouton, deux contacts métalliques à l'intérieur du bouton se connectent, permettant au courant électrique de circuler à travers le circuit. Cette connexion momentanée est détectée par un système électronique, généralement un microcontrôleur, qui interprète ce changement comme une commande.

Chacun de ces composants joue un rôle spécifique dans le fonctionnement global du système. Le microcontrôleur agit comme le cerveau qui prend des décisions en fonction des entrées (boutons-poussoirs) et contrôle les sorties (LEDs WS2812 et écran TFT ILI9341) pour créer des fonctionnalités et des interactions spécifiques.





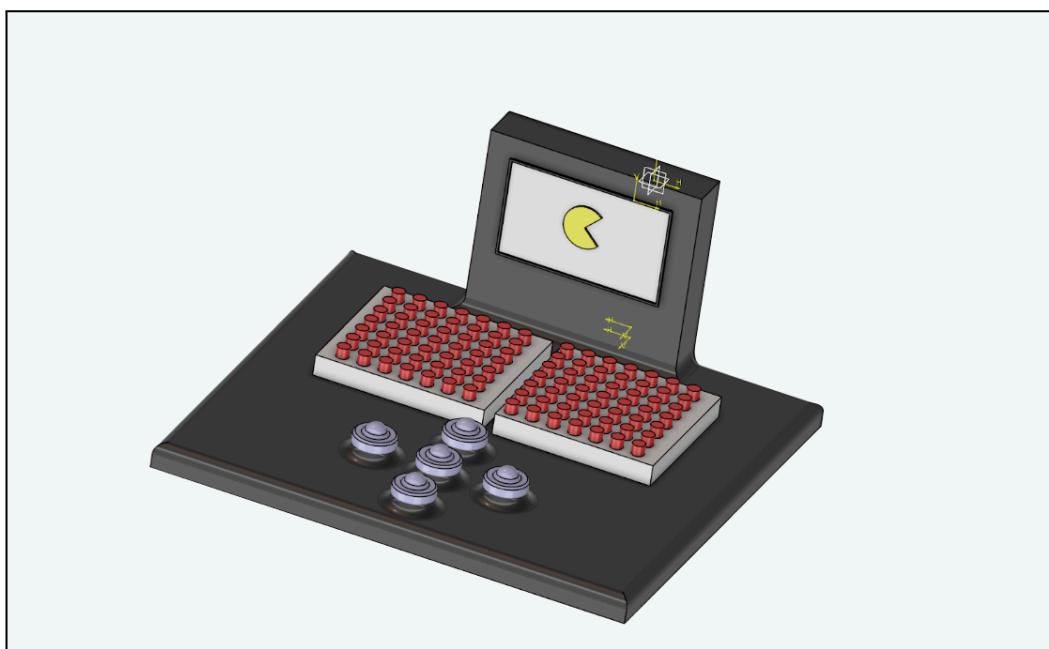
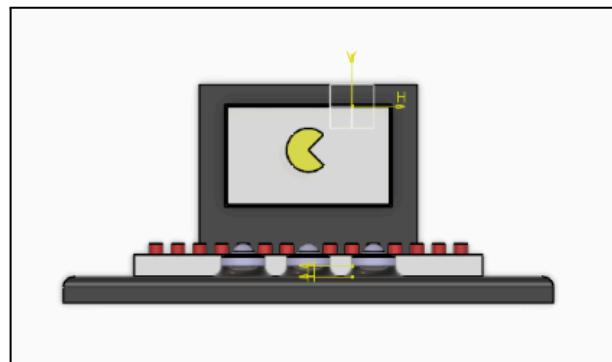
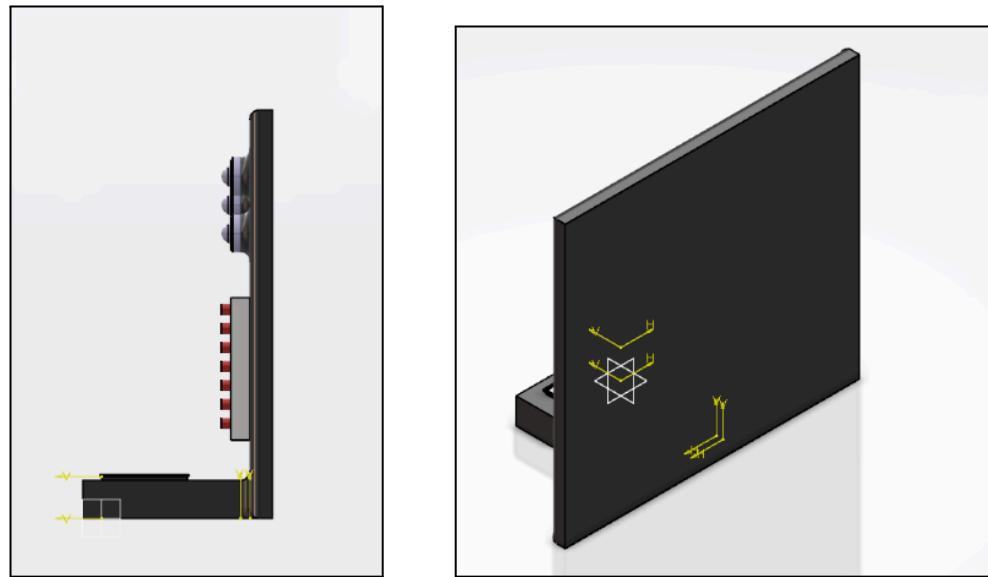
Nous avons rencontré plusieurs défis lors de la conception de la carte :

- La contrainte de taille de la carte était assez limitée, et malgré cela, nous préférions éviter de l'agrandir car la taille de la carte peut influencer les performances du logiciel.
- La gestion des croisements entre deux nets du chevelu s'est avérée complexe à résoudre, nécessitant l'utilisation de vias.

Nous avons obtenu une estimation de coût auprès d'Eurocircuit pour cette conception.

Date d'expédition prévue 17-01-2024 🗓️▼		
Prix	Prix HT	Prix TTC*
Prix unitaire	€ 52.67	€ 63.73
Prix circuits	€ 52.67	€ 63.73

Annexe C – Complément « Design CAO d'un boîtier »



- Nous avons utilisé le logiciel Catia pour réaliser notre jeu Pacman.

Matériaux Ciblés :

1. Microcontrôleur :

- Matériau : Semi-conducteur (silicium pour la puce du microcontrôleur).
- Mode de Production : Généralement fabriqué à l'aide de technologies de semi-conducteurs.

2. LED(s) WS2812 :

- Matériau : Semi-conducteur pour la puce LED, matériaux variés pour le boîtier.
- Mode de Production : Assemblage de semi-conducteurs et d'autres composants, encapsulation, éventuellement SMT (montage en surface) sur une carte de circuit imprimé.

3. Boutons-poussoirs :

- Matériau : Contacts en métal, boîtier en plastique.
- Mode de Production : Injection plastique pour le boîtier, usinage ou estampage pour les composants métalliques.

Modes de Production Ciblés :

1. Découpe Laser :

- Utilisée pour découper des matériaux fins et précis, tels que des boîtiers en acrylique.

2. Impression 3D :

- Idéale pour la production rapide de prototypes ou de pièces personnalisées avec des formes complexes.

3. Usinage :

- Approprié pour la fabrication de pièces métalliques ou plastiques avec des tolérances serrées.

4. Découpe Manuelle :

- Pour des opérations simples où une précision moindre est acceptable.

5. Injection Plastique :

- Utilisée pour la production en série de pièces en plastique avec une grande précision.

6. Emboutissage :

- Principalement pour la fabrication de pièces métalliques en série avec des formes spécifiques.

Mise en Contexte / Explication du Rôle :

1. Microcontrôleur :

- Rôle :

Cerveau du système, traitement des données, gestion des entrées/sorties.

- Particularités : Connecteurs pour les liaisons avec d'autres composants, régulateurs de tension pour l'alimentation.

2. LED(s) WS2812 :

- Rôle :

Source lumineuse programmable.

- Particularités : Connecteurs pour l'alimentation et les signaux de contrôle.

3. Boutons-poussoirs :

- Rôle :

Entrées utilisateurs.

- Particularités : Contacts spécifiques pour détecter les pressions, formes ergonomiques pour une utilisation aisée.

Annexe D – Complément « Documentation Doxygen du code source »

Nous avons élaboré la documentation au format Doxygen pour les fichiers suivants dans le cadre de ce projet :

- `Gestion.c`
- `Gestion.h`
- `Jeu.c`
- `Jeu.h`
- `main.c`
- `Matrice.c`
- `Matrice.h`
- `Menus.c`
- `Menus.h`

Cette démarche vise à améliorer la compréhension du code source en fournissant une documentation structurée et automatisée. Les commentaires inclus dans ces fichiers sont conformes aux normes Doxygen pour garantir une génération efficace de la documentation.