



Soutenance 1

PAC-MANIA

par A2CR STUDIO

Composé par :

Alexandre Bourcier

Clément Bruley

Clément Iliou

Rémi Monteil

Table des matières

1	Introduction	3
2	Présentation du projet	4
2.1	Présentation de l'équipe	4
2.2	Nature du projet	5
2.2.1	Description du projet	5
2.3	Objet d'étude	5
3	Découpage et avancement du projet	6
3.1	Découpage du projet	6
3.2	Avancement du projet	6
4	Création des graphismes	7
4.1	GTK et Glade	7
4.2	L'affichage de la carte et des Sprites	7
5	Le gameplay du joueur	9
5.1	Structure	9
5.2	Score, mort et niveaux	9
5.3	Déplacements	10
6	IA des fantômes	11
6.1	Le pathfinding	11
6.2	Le mode chasseur	11
6.3	Le mode patrouille	12
6.4	Le mode fuite	12
7	Les tâches annexes	13
7.1	Le site web	13
7.2	Gestion du dépôt git	14
7.3	L ^A T _E X	14
8	Plan de soutenance (théorique)	15
8.1	Tableau d'avancement	15
8.2	Détail de l'avancement et des objectifs	15
9	Conclusion	16

1 Introduction

Ce rapport de soutenance va présenter toutes les avancé de notre projet du S4. Ce projet a pour but de créer une intelligence artificielle et de l'a montré visuellement à l'utilisateur. Le but ici n'est évidemment pas de recoder le célèbre jeu Pac-Man, mais bien de se concentrer sur comment développer une IA qui fera un gros score..

Ce rapport va donc décrire tout d'abord l'équipe et l'origine du projet avec une présentation des différents membres. Il y aura ensuite une description du projet ainsi que les différents enjeux du projet. Après, nous parlerons de ce que nous avons réaliser sur cette première partie de projet. Nous décrirons avec précision ce qui a été réalisé et qui l'as fait. Enfin, nous ferons une conclusion.

2 Présentation du projet

2.1 Présentation de l'équipe

Alexandre Bourcier : Entré à l'EPITA en 2019, je me suis toujours efforcé de faire beaucoup de programmation pour avoir des acquis et une expérience solide dans cette discipline. Ainsi, avec le projet de S2, j'avais pris en charge une partie très importante de la programmation d'un jeu vidéo (multijoueur et gameplay). Lors du projet de S2, j'ai pris en charge l'interface graphique, le pré-traitement et la segmentation de l'OCR. Ainsi, toutes ces expériences m'ont permis d'avoir des acquis assez solides et d'aborder ce projet de S4 sereinement. L'idée de refaire une intelligence artificielle me plaît beaucoup, car lors de notre projet de S3, le réseau de neurones était loin d'être impeccable et je souhaite pouvoir prendre ma revanche sur ce projet.

Pendant les projets, j'apprécie toujours commencer tout de suite les projets pour pouvoir vraiment étaler tout le travail sur le temps imparti et ne pas courir à la fin et rentrer quelque chose de moyennement fonctionnel. Un point très important aussi est de discuter de la conception du projet, notamment le prototype des fonction et comment les faire interagir. Tout cela nous permettra, j'espère de faire un beau projet fonctionnel.

Clément Bruley : La plupart des élèves d'EPITA ont commencé à toucher à l'informatique depuis de nombreuses années. Pour ma part, j'ai commencé réellement lorsque je suis entré à EPITA. Lors des deux derniers projets, nous avons approfondi nos connaissances en `c#` lors du S2 et en `C99` lors du S3. Ce projet va me permettre d'avoir de plus grandes connaissances en `C` qui est un langage très utile dans le domaine de l'informatique. Possiblement, nous allons pouvoir appliquer ce que nous avons appris lors du séminaire UNIX. J'ai toujours été intéressé par les IA. Grâce à ce projet, je vais pouvoir développer mes compétences dans ce domaine.

Avec le reste du groupe nous avons décidé de m'attribuer le rôle de chef de projet. Il me semble que c'est important d'avoir quelqu'un qui va "diriger" le projet. Néanmoins, ce rôle ne va pas être très grand, car il ne s'agit pas d'un grand projet tel que pourrait le faire des entreprises.

Clément Iliou : Je suis étudiant en deuxième année de classe préparatoire à EPITA. J'ai effectué une terminale S spécialité SVT. Avant EPITA, je n'avais pas fait de programmation dans un cadre scolaire. Cependant, je me suis toujours intéressé à l'informatique en général. J'aime beaucoup les travaux de groupe, car l'on peut ainsi profiter des différents atouts de tous. La partie graphique du jeu m'encourage à approfondir le domaine du traitement d'image que je trouve passionnant. Ce projet est un moyen pour moi de croiser les connaissances acquises en développement de jeu vidéo durant le projet de S2 et celles sur le machine learning et les réseaux de neurones vu pendant le projet de S3. J'ai hâte après EPITA de pouvoir développer ce genre de programme dans un cadre professionnel. Ce projet devrait me permettre d'approfondir mes connaissances en `C99` et en `GTK`.

Rémi Monteil : Je suis rentré à EPITA en 2019. Depuis j'ai appris plusieurs langages tels que le `Caml`, `c` et le `C`. Mais surtout j'ai appris à les exploiter au maximum avec les projets. J'avais développé le gameplay du jeu de S2 et je me suis occupé du Neural-Network de l'OCR. Je suis très attiré par ce qui touche à l'IA autant dans ces utilisations exotiques que de son apprentissage si divers. La difficulté et les erreurs que j'ai commises lors de l'OCR ne sont que de l'expérience supplémentaire pour appréhender cette nouvelle IA qui me semble plus difficile par sa méthode d'apprentissage non supervisée. J'espère partager au mieux mes connaissances des projets précédents pour accompagner mes collègues dans ce projet.

2.2 Nature du projet

2.2.1 Description du projet

Le principe du projet est assez simple. Il s'agit de reproduire le jeu Pac-Man et de faire une intelligence artificielle qui essaie de faire le score le plus élevé possible. Le but de ce projet n'est pas de coder un jeu vidéo, mais de se concentrer vraiment sur de l'intelligence artificielle avec du traitement de donnée dans un réseau de neurones. L'implémentation visuelle du jeu nous permettra également de faire un peu de traitement d'image.

2.3 Objet d'étude

Ce projet peut apporter beaucoup en terme d'expérience de programmation et de conception d'IA. Déjà dans un premier temps, il faudra programmer le jeu. Cela va demander surtout d'utiliser des connaissances en GTK+ et en traitement d'image. Il faudra programmer une mini-intelligence des fantômes, les faire bouger, rendre les pac-goms interactifs, activer les super pac-goms, augmenter le score et évidemment afficher graphiquement tout ça dans notre fenêtre avec GTK+.

Il faudra ensuite se concentrer sur l'intelligence artificielle et là, il y a une grosse partie de réflexion sur comment nous allons faire une intelligence artificielle. Nous pensons utiliser un réseau de neurones, mais nous allons probablement devoir faire de nombreux tests de modèle d'IA, changer les paramètres d'entrée,...

3 Découpage et avancement du projet

3.1 Découpage du projet

Tâches à effectuer	Alexandre Bourcier	Clément Bruley	Clément Iliou	Rémi Monteil
Création graphique	⊗		✓	
Gameplay du joueur	✓		⊗	
IA des fantômes		⊗		✓
IA de Pac-Man	⊗	⊗	⊗	✓
Site web	⊗	✓		
Git	✓	⊗		
L ^A T _E X	⊗	✓		

✓ = représentant

⊗ = *assistant*

Les tâches énumérées précédemment ne sont pas de difficulté ni de durée équivalente. En effet, la gestion du git est très rapide à faire tandis que la création de l'IA du joueur est assez difficile.

3.2 Avancement du projet

Tâches à effectuer	Prévu	Effectué
Création du visuel	99%	90%
Animation du joueur	70%	90%
Déplacement des fantômes	70%	70%
IA du joueur	0%	0%
Site web	30%	80%

4 Création des graphismes

4.1 GTK et Glade

Afin d’afficher le jeu, nous utilisons la bibliothèque GTK. Cette bibliothèque permet de créer des boutons, ainsi qu’une interface visuelle afin que le joueur puisse jouer. Pour structurer l’interface, nous utilisons Glade. Dans Glade, nous avons créé deux boutons : Start et Pause, qui permettent de démarrer ou de mettre le jeu en pause. Il y a aussi trois GtkLabel qui permettent d’afficher le score, le nombre de vies restantes au joueur et le niveau.

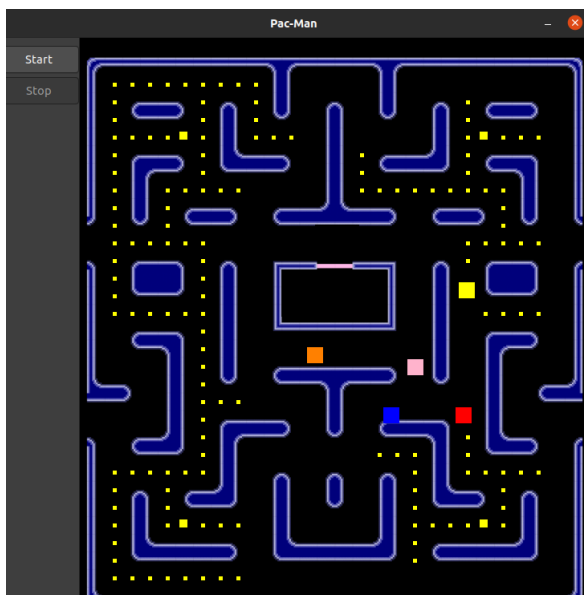


FIGURE 4.1 – Premier graphisme

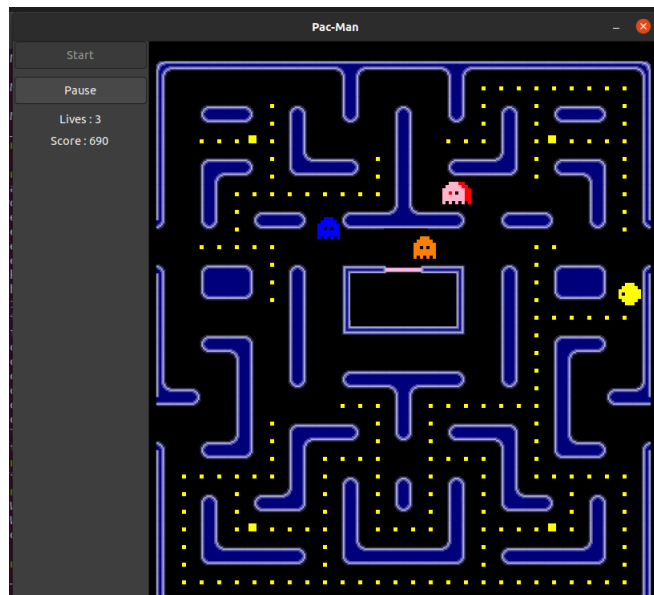


FIGURE 4.2 – Graphismes finaux

4.2 L’affichage de la carte et des Sprites

L’affichage de la carte et des Sprites sont séparés. Nous superposons grâce à un GtkOverlay une image et une Drawing Area. Nous avons choisi ce mode d’affichage, car cela nous permet d’avoir un jeu plus agréable visuellement et plus optimisé puisqu’à chaque lancement, la carte n’a pas besoin d’être reconstruite. Pour savoir si une case est un mur, un couloir vide, un couloir avec une pacgum ou une super pacgum nous avons réalisé une matrice de 28 par 31 comme dans le jeu original. Au début, de chaque partie, nous affichons toute les pacgums à partir de la matrice. Pour afficher Pac-man et les fantômes nous avons créé d’autres matrices afin de les afficher en Pixel Art. Pour Pac-man, une image sur trois celui-ci ouvre la bouche comme dans le jeu original. Pour que la direction du pixel de Pac-man change, nous changeons juste le sens du parcours de la matrice.

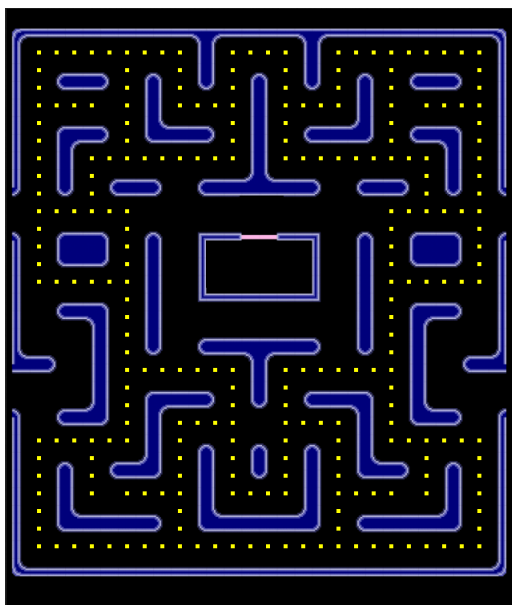


FIGURE 4.3 – Map affichée

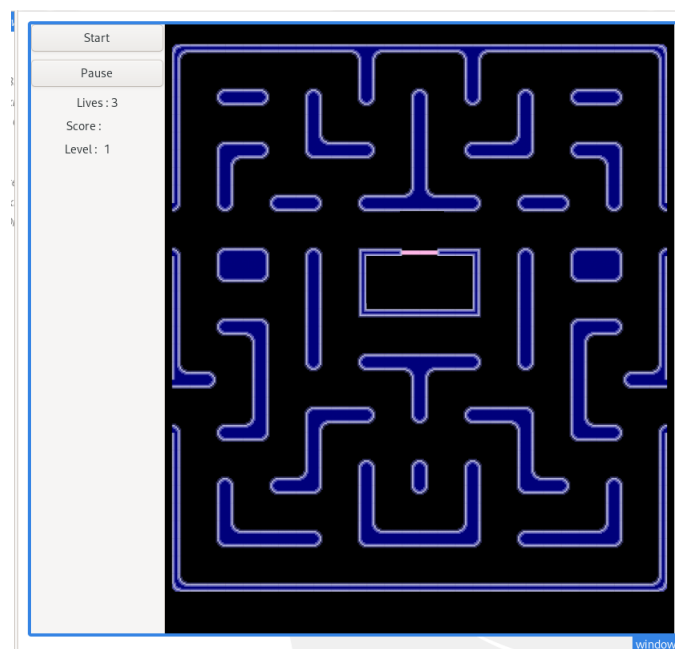


FIGURE 4.4 – Fichier glade utilisé pour faire l'interface et l'affichage de la carte

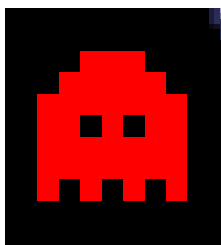


FIGURE 4.5 – Sprite de fantôme

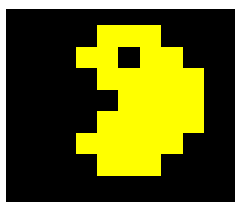


FIGURE 4.6 – Sprite de Pac-man avec la bouche ouverte

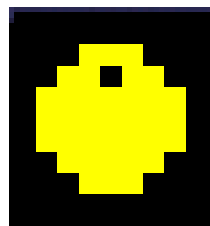


FIGURE 4.7 – Sprite de Pac-man avec la bouche fermée

5 Le gameplay du joueur

5.1 Structure

La structure du jeu comporte de nombreux paramètres. Cette structure comporte d'abord 5 joueurs : Pac-man et les 4 fantômes. Ces joueurs possèdent 3 caractéristiques : leur coordonnées (en x et en y) et leur direction. La structure du jeu possède aussi le statut pour savoir si le jeu est en pause ou en cours, le mode chasse pour savoir si Pac-man est invincible ou pas, le score, le nombre de vies et le niveau.

```
//-----INITIALISATION-----
int map[31][28] ={
// 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}, //0
{0,2,2,2,2,2,2,2,2,0,0,2,2,2,2,0,0,2,2,2,2,2,2,2,2,0}, //1
{0,2,0,0,0,0,2,0,0,2,0,0,2,0,0,2,0,0,2,0,0,2,0,0,0,0,2,0}, //2
{0,2,0,0,0,0,2,0,0,2,0,0,2,0,0,2,0,0,2,0,0,2,0,0,0,0,2,0}, //3
{0,2,2,2,2,3,2,0,0,2,2,2,2,0,0,2,2,2,2,0,0,2,3,2,2,2,0}, //4
{0,2,0,0,0,0,2,0,0,0,0,0,2,0,0,2,0,0,0,0,0,2,0,0,0,0,2,0}, //5
{0,2,0,0,0,0,2,0,0,0,0,0,2,0,0,2,0,0,0,0,0,2,0,0,0,0,2,0}, //6
{0,2,0,0,2,2,2,2,2,2,2,2,2,0,0,2,2,2,2,2,2,2,2,2,0,0,2,0}, //7
{0,2,0,0,2,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,2,0,0,2,0}, //8
{0,2,0,0,2,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,2,0,0,2,0}, //9
{5,2,2,2,2,2,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,2,2,5}, //10
{0,2,0,0,0,0,2,0,0,1,0,0,0,4,4,0,0,0,1,0,0,2,0,0,0,0,2,0}, //11
{0,2,0,0,0,0,2,0,0,1,0,4,4,4,4,4,4,0,1,0,0,2,0,0,0,0,2,0}, //12
{0,2,0,0,0,0,2,0,0,1,0,4,4,4,4,4,4,0,1,0,0,2,0,0,0,0,2,0}, //13
{0,2,2,2,2,2,2,0,0,1,0,4,4,4,4,4,4,0,1,0,0,2,2,2,2,2,0}, //14
{0,1,0,0,0,0,2,0,0,1,0,0,0,0,0,0,0,0,1,0,0,2,0,0,0,0,1,0}, //15
{0,1,0,0,0,0,2,0,0,1,1,1,1,1,1,1,1,1,1,0,0,2,0,0,0,0,1,0}, //16
{0,1,1,1,0,0,2,0,0,1,0,0,0,0,0,0,0,0,1,0,0,2,0,0,1,1,1,0}, //17
{0,0,0,1,0,0,2,0,0,1,0,0,0,0,0,0,0,0,1,0,0,2,0,0,1,0,0,0}, //18
{0,0,0,1,0,0,2,2,2,2,2,2,2,0,0,2,2,2,2,2,2,2,2,0,0,1,0,0,0}, //19
{5,1,1,1,0,0,2,0,0,0,0,0,2,0,0,2,0,0,0,0,0,2,0,0,1,1,1,5}, //20
{0,1,0,0,0,0,2,0,0,0,0,0,2,0,0,2,0,0,0,0,0,2,0,0,0,0,1,0}, //21
{0,1,0,0,0,0,2,0,0,2,2,2,2,1,1,2,2,2,2,0,0,2,0,0,0,0,1,0}, //22
{0,2,2,2,2,2,2,2,0,0,2,0,0,2,0,0,2,0,0,2,0,0,2,2,2,2,2,0}, //23
{0,2,0,0,2,0,0,0,0,2,0,0,2,0,0,2,0,0,2,0,0,0,0,2,0,0,2,0}, //24
{0,2,0,0,2,0,0,0,0,2,0,0,2,0,0,2,0,0,2,0,0,0,0,2,0,0,2,0}, //25
{0,2,0,0,2,3,2,2,2,2,0,0,2,2,2,2,0,0,2,2,2,2,3,2,0,0,2,0}, //26
{0,2,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,2,0}, //27
{0,2,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,2,0}, //28
{0,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2}, //29
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}}; //30
```

FIGURE 5.1 – Matrice utilisé pour la map

5.2 Score, mort et niveaux

Le joueur possède trois vies en début de partie, il en perd une à chaque fois qu'il se fait manger pas un fantôme. Nous vérifions que Pac-man et un des fantômes ne se trouvent pas sur la même case. Lorsque Pac-man se fait manger, celui-ci ainsi que les fantômes retournent à leur emplacement de départ. Nous utilisons alors la fonction Sleep(3) pour permettre au joueur de réfléchir à une nouvelle stratégie. Il possède 2 secondes pour réfléchir.

Lorsque le joueur mange une pacgum nous modifions la valeur de la case dans la matrice. Pour vérifier si le joueur a mangé toutes les pacgums nous utilisons un compteur du nombre de pacgum mangé. Nous réinitialisons la matrice à chaque changement de niveau. Nous ne mettons pas de limite de niveau comme dans le jeu original.

Lorsque Pac-man mange une super-pacgum (il y en a 4 par niveau) celui-ci devient bleue et il peut manger les fantômes. Les mouvements des fantômes changent et ils passent en mode fuite. L'effet de la pacgum dure environ 10 secondes, à la fin Pac-man redevient jaune.

5.3 Déplacements

Comme dit plus haut, nous avons créé une matrice de 28 par 31, et nous avons donc un premier système de coordonnées, c'est-à-dire que pour chaque entité nous avons des coordonnées dans cette matrice. Ensuite, nous avons un deuxième système de coordonnées, celles qui correspondent au pixels de zone de dessin de GTK.

Ainsi, pour ne pas se perdre dans ce double système de coordonnées, nous avons décidé de séparer en 2 fichiers :

- un premier fichier GTK.c qui traite de tout ce qui est en lien avec l'interface graphique et donc le système de coordonnées relatives à la zone de dessin.
- un second fichier pac-man.c qui traite de la gestion du jeu en lui-même avec le système de coordonnées de la matrice.

Pour la direction de Pac-man, nous créons une fonction d'appel dans GTK.c, qui est déclenché lorsqu'on appuie sur une touche. Ensuite on appelle une seconde fonction dans pac-man.c qui vérifie grâce aux coordonnées de la matrice si Pac-man ne va rentrer dans un mur en changeant de direction. On change ensuite la direction de Pac-man.

Concernant les déplacements de Pac-man, on définit la vitesse de celui-ci comme étant le nombre de pixel de déplacement par frame (le jeu est en 24 frames par seconde). Ainsi, à chaque frame, on vérifie que Pac-man ne va pas rentrer dans un mur avec les coordonnées en de la matrice et la direction puis on modifie les coordonnées en pixel pour que Pac-man soit ensuite re-dessiner dans la zone de dessin GTK.

Pour le déplacement des fantômes, la procédure est similaire à celle de Pac-man sauf que la direction est redéfini à tous les frames et que celle ci tient compte des obstacles. Ainsi il ne reste plus qu'à mettre à jours le coordonnées.

6 IA des fantômes

6.1 Le pathfinding

Les fantômes utilisent un algorithme de pathfinding. Au moment de la création de la fonction, nous étions et nous sommes toujours dans l'étude des graphes en cours d'algorithmique. L'algorithme privilégié pour trouver le plus court chemin était le parcours largeur. Nous avons donc adapté ce parcours de graphe à celui de l'exploration de labyrinthe dans la matrice.

Comme vu précédemment nous utilisons une matrice qui définit chaque case du plateau de jeu. Pour faire le plus simple possible nous avons attribué des numéros à chaque "état" que peut prendre la case. Par exemple les cases "0" sont des murs et "1", "2" et "3" sont des couloirs avec ou sans pacgum. Les cases "4" correspondent à la maison des fantômes et les cases 5 sont les télé-porteurs que nous ignorons pour le moment. Pour réussir l'implémentation du parcours largeur, il a fallu avoir les queues qu'on utilise en cours d'algorithmique. Nous avons trouvé la librairie "sys/queue.h". Néanmoins, suite à des difficultés de manipulation et des doutes ressentis, nous avons donc décidé de créer notre propre struct queue qui a été optimisé et corrigé au fil du projet.

Nous avons effectué des recherches sur d'autres algorithmes de pathfinding et avons trouvé le a* qui est décrit comme le plus performant. Mais au vu de la première soutenance, nous avons conclu que ce serait une perte de temps et dangereux de le faire à 1 semaine de la soutenance d'autant plus que notre fonction de pathfinding actuel fonctionne. Mais le changement vers cet algorithme a* sera probablement envisagé et étudié pour la prochaine soutenance.

6.2 Le mode chasseur

En mode chasseur les fantômes : Blinky (le rouge), Pinky (le rose), Inky (le bleu) et Clyde (le orange) vont pourchasser Pac-man. Mais ils ont chacun leur propre technique :

- Blinky : Il va simplement poursuivre Pac-man.
- Pinky : Il est un peu plus malin. Il va tenter de piéger Pac-man. Pour cela il se dirige vers le mur où Pac-man irait s'il continuait dans sa direction actuel.
- Inky : Il est un peu plus compliqué. Il va considérer que Pac-man essayera de fuir en priorité Blinky (étant donné qu'il semble le plus agressif). Ainsi, il prend comme destination une coordonnée qui est le symétrique ou au alentours du symétrique de la position de Blinky par rapport à Pac-man. Dans certains cas, ce symétrique se trouve hors de la map, il va alors effectuer la même chose que Blinky.
- Clyde : Il est un peu plus fantaisiste. Il va toujours fuir Pac-man. Il va vers les coordonnées étant la plus proche de lui et dans un rayon de 5 cases autour de Pac-man.

Note : les noms japonais expriment très bien leur mode chasseur respectif :

- Blinky = chasseur
- Pinky = embusqué
- Inky = inconstants
- Clyde = stupide.

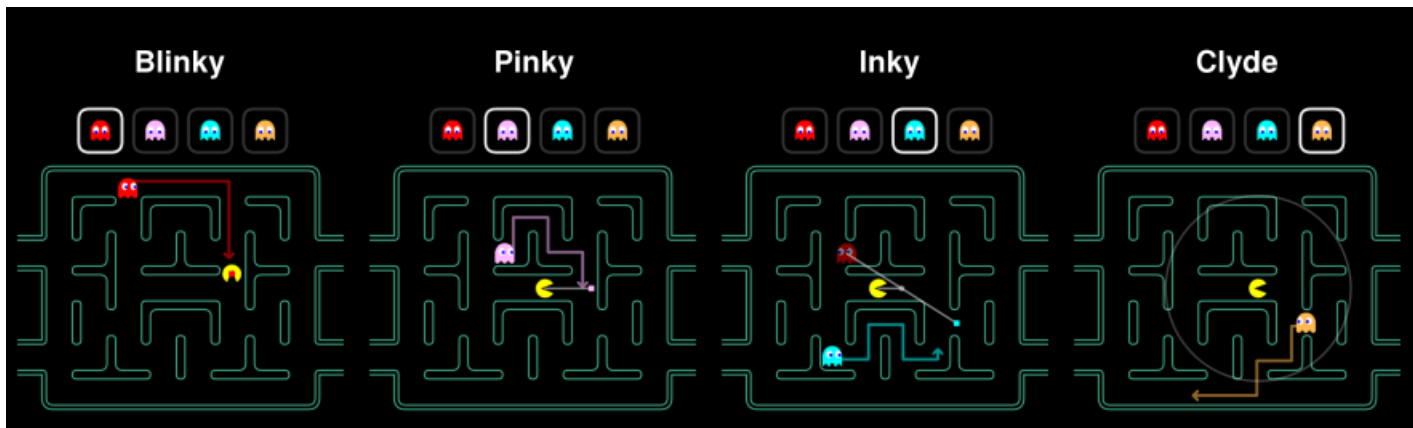


FIGURE 6.1 – Mode chasseur de chaque fantômes

6.3 Le mode patrouille

Les fantômes ne poursuivent pas constamment Pac-man. Au bout d’une certaine distance de Pac-man, il décide d’arrêter leur chasse et de passer en mode patrouille. Ainsi le fantôme en question va se promener dans le coin le plus proche ou il se trouve. Pour le moment ce mode n’est pas activé.

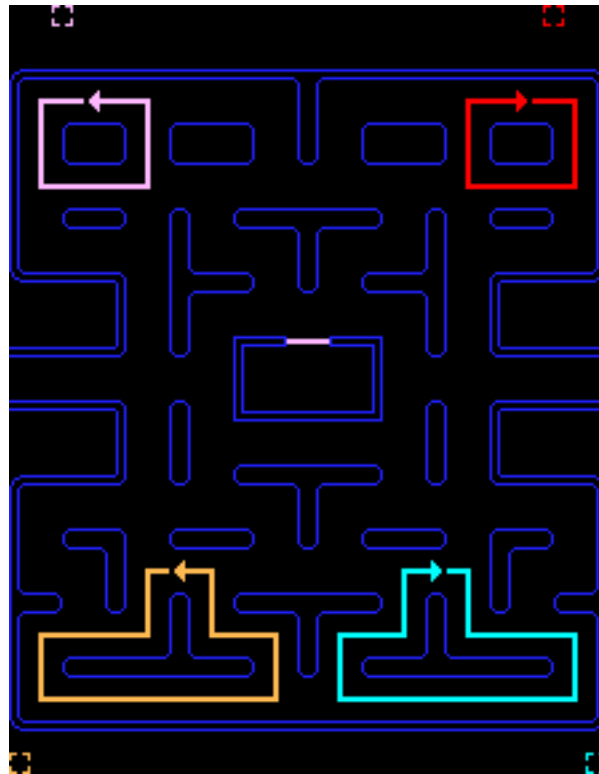


FIGURE 6.2 – Mode patrouille de chaque fantôme

6.4 Le mode fuite

Ce mode peut être déclenché uniquement par Pac-man lorsqu’il mange une super pacgum. A partir de ce moment Pac-man a 10 secondes pour manger les 4 fantômes et ainsi obtenir plus de point. A ce moment les mouvements des fantômes sont aléatoires comme dans le jeu original.

7 Les tâches annexes

7.1 Le site web

Pour le site web, nous avons décidé de faire un site sobre et épuré. Il y a donc un bandeau noir en haut de la page avec le logo du projet ainsi que les différents liens pour accéder aux pages. Sur les pages en elle-même il y aura un visuel de notre application et au-dessus du texte avec un style graphique assez simpliste.

Au départ, nous avions prévu de faire le site web plutôt vers la fin du projet mais ayant eu un peu de temps libre lors du commencement du projet Alexandre à entrepris de créer le site.



FIGURE 7.1 – Page d'accueil



FIGURE 7.2 – À propos de nous

7.2 Gestion du dépôt git

Bien que l'utilisation d'un git n'ait pas été voté à l'unanimité, nous en avons néanmoins créé un pour pouvoir faire une structure propre. Ainsi, Clément c'est lancé dans la création de branches pour que chacun puisse travailler de son côté sans détruire le travail d'un autre membre du groupe. Nous n'avons eu aucun problèmes avec l'utilisation des branches jusqu'au moment où il a fallu faire un git merge des branches. Comme tout le monde le sait il peut y avoir des conflits entre les branches lorsque nous les réunissons. C'est ce qui nous est arrivé. Il nous a fallu un peu de temps pour comprendre comment il fallait s'y prendre pour régler ces conflits et ainsi terminer la réunion des codes de chacun.

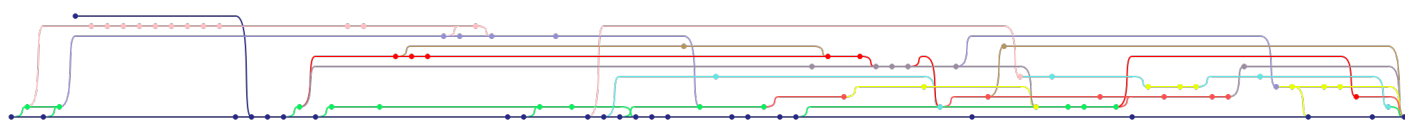


FIGURE 7.3 – Tout les commits que nous avons fait

Anecdote : Le PC de Clément étant un français, lors de l'exécution d'une commande, un message est apparu comme quoi il a essayé de "picorer un commit".

7.3 L^AT_EX

Pour pouvoir effectuer le L^AT_EX correctement nous avons pris de l'avance. En effet nous avons commencé à écrire ce rapport de soutenance plus d'une semaine avant la soutenance. Ainsi nous avons pu nous remémorer plus facilement tout ce que nous avons fait pour le projet depuis le début du mois de mars.

8 Plan de soutenance (théorique)

8.1 Tableau d'avancement

	Soutenance 2	Soutenance 3
Création du visuel	99,9%	100%
Animation du joueur	100%	100%
Déplacement des fantômes	100%	100%
IA du joueur	50%	100%
Site web	95%	100%

8.2 Détail de l'avancement et des objectifs

Soutenance 2 :

- Avoir un système de patrouille pour les fantômes
- Avoir une base de réseau de neurones fonctionnel
- système de sauvegarde des poids et biais
- réflexion autour du modèle de réseau de neurones et des données d'entrée bien avancé.
- Site web prêt avec toutes les pages, mais graphisme pas encore terminé.

Soutenance final :

- Avoir une IA fonctionnelle qui fait des gros scores.
- Site web et graphique finis.
- Interface graphique du logiciel et finitions terminés.

9 Conclusion

Pour conclure, notre projet avance bien comme nous le souhaitions. Nous n'imaginions pas vraiment que la programmation du jeu prenne autant de temps mais nous avons finalement réussi à avoir un jeu jouable. Nous avons donc une interface graphique fonctionnelle, un déplacement de Pac-man qui marche, et une intelligence artificielle des fantômes qui est presque terminée.

Lors de ce début de projet, nous avons approfondi beaucoup de sujet comme le Git, où nous avons pour la première fois appris bien utiliser le système de branches. Nous avons aussi pu approfondir la bibliothèque GTK+ avec notamment sa zone de dessin.

Concernant l'équipe autour de ce projet, nous restons tous bien soudés et l'ambiance est vraiment bonne, nous sommes vraiment confiants pour la suite et pensons pouvoir mener à bien ce projet et rendre une intelligence artificielle qui marchera parfaitement.