**Dossier ISN**

**Pac-Man remastered**



**Membres du projet :**

* SOTTAS Louis-Albert
* PIRES Alexandre

**Pourquoi avoir choisi ce projet ?**

Nous avons choisi de créer un jeu vidéo car nous sommes passionnés de jeux vidéo (sur pc principalement). Ayant l’occasion d’en créer un, nous avons sauté sur l'occasion. Au lieu de simplement jouer à un jeu, nous avons la possibilité d’en créer un de toute pièce et ainsi peut-être mieux se rapprocher de ce que pourrait être le développement des jeux en entreprise (malgré le fort clivage de nos connaissances et celui d'employer lambda).

Après mure réflexion nous avons opté pour le développement de « Pac-Man » car ce jeu d'arcade est un classique du rétro gaming. Un des pionniers dans l'industrie des jeux vidéo, autrefois présent sur les bornes d'arcade, il s'est fait démarquer par sa facilitée de prise en main et par sa forte additivité. De plus par le passé, nous avions déjà décidé de le reproduire sur Scratch en 2nd (ICN), lors d'un projet de jeu pour des maternelles. Cependant du fait du public visé le jeu se devait d'être très enfantin et donc le but n'était que de manger des balles dans un sens donné.

Cependant avant de commencer à créer les premières lignes de code, nous avions réfléchi à la difficulté et aux problèmes de développement sur lesquels nous pouvions tomber, mais petit à petit nous avons réalisé que ce jeu était un projet tout à fait réalisable malgré notre niveau en informatique.

**Qu’est-ce que « Pac-Man » ?**

Pac-Man est un jeu en 2D d'arcade, dont le but compétitif est d’obtenir le plus grand score possible. Le personnage que l'on contrôle est Pac-Man, un cercle jaune avec une bouche qui se trouve enfermé dans un labyrinthe. Son but est de manger tous les balles du labyrinthe (pac-gommes) tout en évitant de rentrer un contact avec les fantômes. Les fantômes/monstres sont au nombre de 4. Trois sont disposer dans une cage au centre de la carte et un se situe initialement en dehors. Chaque fantôme se déplace selon un algorithme plus ou moins différent :

- le fantôme rouge (Blinky), traque le pac-man ;

- le fantôme rose (Pinky) cherche toujours à se placer devant le pac-man pour lui tendre une embuscade tout comme le fantôme bleu (Inky) ;

- le fantôme orange (Clyde) se déplace aléatoirement.

C’est un jeu se jouant avec les touches directionnelles ou autrefois à l'aide d'un joystick, ce qui permet par la suite au joueur de faire avancer Pac-Man de manière automatique et linéaire dans une direction choisie (haut, bas, gauche, droite].

Le seul moyen de perdre est donc de se faire « manger » par un des fantômes, malgré nos 3 vies initiales (Il est toutefois possible d'en récupérer en mangeant un bonus). Il y a toutefois la possibilité de récolter des super balles (fruit rond) qui se comportent comme des bonus et qui permettent à Pac-Man pendant une durée limité de dévorer les fantômes. Après se faire manger, leurs yeux se redirigent vers le centre de la zone là où ils finissent par réapparaitre (cage).

**La démarche du projet :**

Pour mener à bien notre projet, nous avons utilisé processing avec Java comme langage. Nous avons tout d’abord réfléchi sur papier à l’aspect graphique mais surtout à l’aspect technique, la forme du labyrinthe, les déplacements (fantômes et Pac-man). Nous nous sommes donc répartit de manière à ce que :

* Une personne s’occupe de l’aspect technique (Louis-Albert)
* Une personne s’occupe de l’aspect graphique, sonore, et peaufinage technique (Alexandre).

Chronologie de la création de notre projet :

1. Création et importation de la carte sur processing grâce un fichier texte
2. Mise en place du Pac-man et de son déplacement (non automatique)
3. Création des balles/coins (pac-gommes)
4. Instauration des Monstres (image et placement dans le labyrinthe)
5. Élaboration de Portails/ Téléporteurs
6. Automatisation du déplacement du Pac-man
7. Initialisation des déplacements monstres
8. Apparition de la possibilité de gagner ou perdre
9. Implémentation d’un chargement et d’un menu
10. Utilisation de musiques pour le menu, lorsque l’on gagne, perd et lorsque que l’on clique sur play.

**Problèmes rencontrés :**

Nous avons tout d’abord commencé notre projet sur une machine virtuel avec le langage C mais nous avons rencontré un gros problème. Nous avons dès le début utilisé graphics.h pour créer une fenêtre graphique pour afficher ce que l’on voulait. Cependant le problème était que nous voulions afficher des images (pour l’apparence des fantômes/monstres et du pac-man et pour les balles (pac-gommes)), mais pour cela nous devions utiliser une bibliothèque externe (SDL image). Suite à l’ajout de cette bibliothèque externe, nous fallait compiler avec un Makefile en passant par le terminal au lieu de pouvoir compiler avec Geany (habituellement). Sachant que notre niveau en informatique est peu élevé comparé à ce que demande l’utilisation d’une bibliothèque externe, que ce soit le langage pour le Makefile, mais aussi dans les commandes que nous devions utiliser avec le terminal pour compiler le projet. Suite à cela nous sommes passés sur processing avec le Java comme langage, qui est un logiciel beaucoup plus simple d’utilisation pour tout ce qui est graphique avec des objets (images, formes, …). De plus lors de notre année en 2nd (en ICN), nous avions déjà appris les bases pour utiliser processing pour créer des objets graphiques.

Déplacement Joueur/SubString :

Pour déplacer le pac-man nous avons dû utiliser SubString qui permet de Garder une chaîne de caractère tout en pouvant la modifié.

.substring(A,B) = la chaîne de caractère est gardé de A à B-1, avec suppression de ce qui a avant A et possibilité d’ajout de caractère après B-1

Exemple : lines[Px]= lines[Px].substring(0,Py) + ' ' + lines[Px].substring(Py+1);

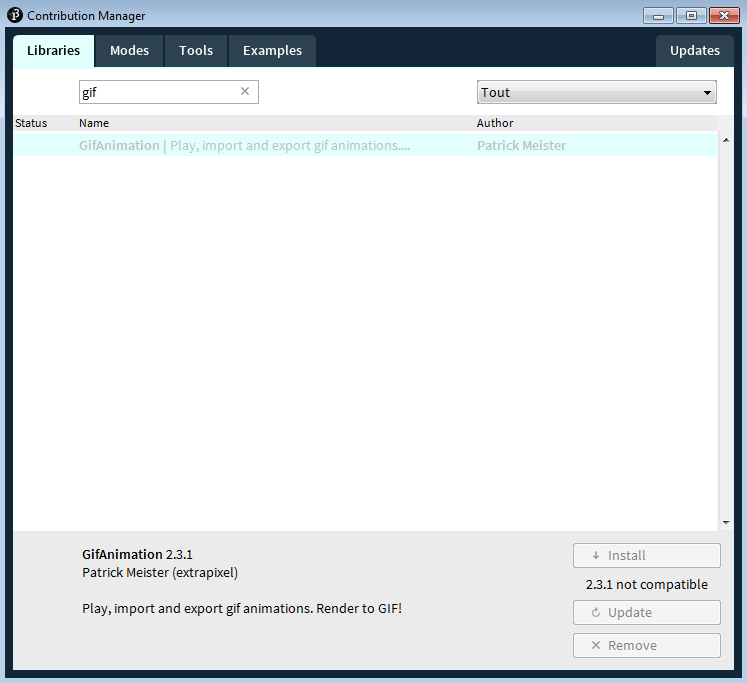
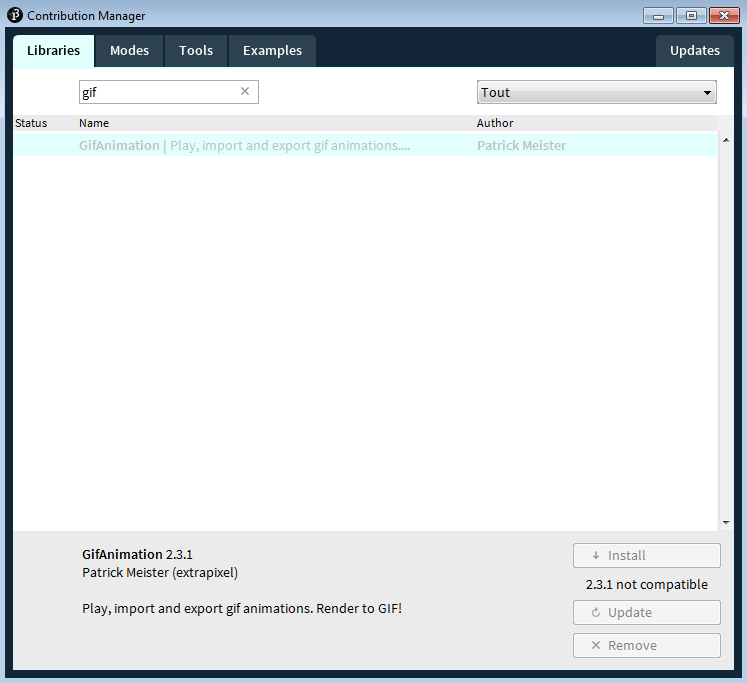
Sur ma ligne Px on modifie la chaîne de caractère en gardant de 0 à Py-1 et on rajoute à cette suite de caractère un espace noté ‘ ’. Puis on substring Py+1 garder la suite de la chaîne de caractère Px.

Gif/Sound :

Nous avion eu l’idée de rajouter dans notre projet des images, des GIF et des sons/musiques. Cependant Processing nécessite l’importation de « sketchs » qui sont des sortes de plugins permettant l’ajout de nouvelles fonctionnalités. Processing à la capacité de lire les formats d’image telle que .JPG et .GIF, mais de base il ne peut pas afficher un GIF animé. Il n’a pas non plus la capacité de proposer des fonctions pour intégrer des musiques/sons.

Pour cela il nous a fallu rajouter une librairie «gifAnimation» et une librairie «minim». La librairie minim fût facile à trouver et facile à installer, car disponible et à jour pour la version processing 3.x dans la librairie sketchs.

Néanmoins ce n’était pas aussi simple lors de la mise en place de la librairie gifAnimation car sa version proposée dans la librairie était obsolète. Il nous a fallu aller chercher sur internet une version compatible avec Processing 3.x, puis l’adapté avec notre version de Processing. Cela nous as pris plus de 3heures.



**Sitographie :**

Processing:

<https://processing.org/>

Musique et Son:

<https://www.youtube.com/watch?v=qtZ0hl-unM4> <http://www.classicgaming.cc/classics/pac-man/sounds>

Test d'un Pac-man 100% fonctionnel:

<http://gamezone.2001jeux.fr/play-00562-b-pacman.html>

Forum et aides, astuces:

[http://codelab.fr](http://codelab.fr/)

[https://forum.processing.org](https://forum.processing.org/)

<https://translate.google.fr/>

Site officiel de gifAnimation:

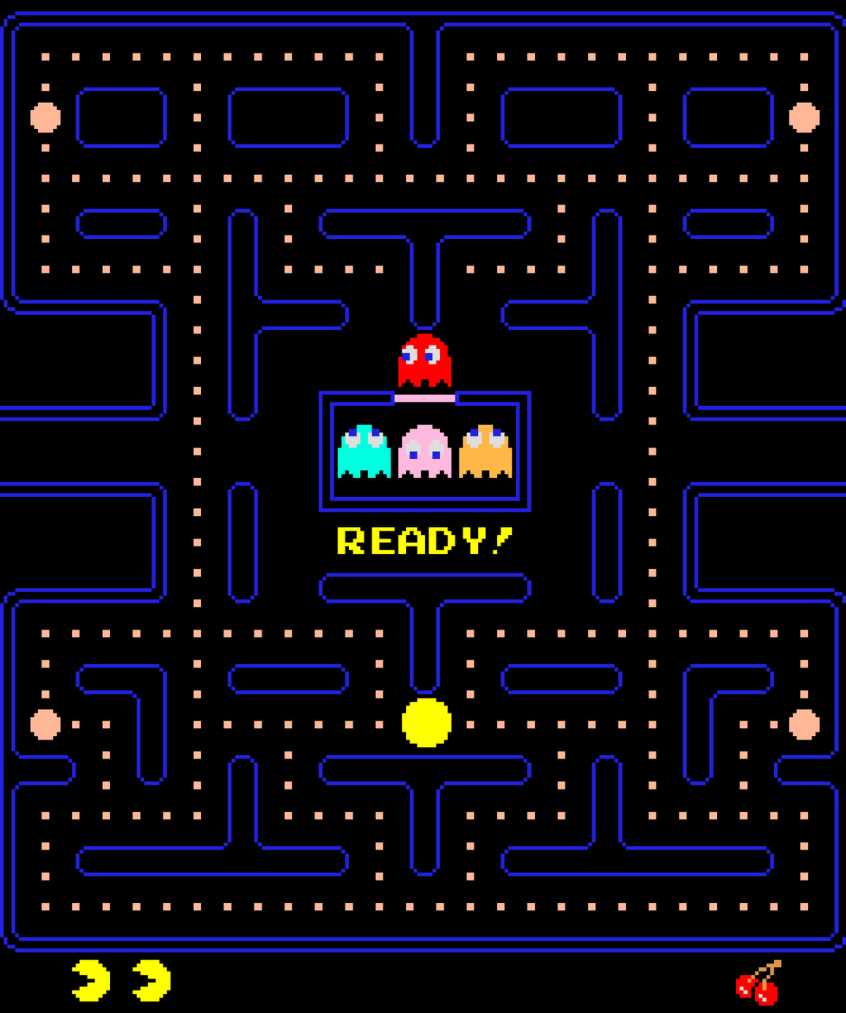
<https://github.com/01010101/GifAnimation>

**Perspectives et évolutions :**

Lors de notre sélection de projet nous étions très ambitieux, quand à notre rendu de la production finale. Nous rendons là un projet qui est à notre niveau assez complet, mais qui cependant laisse la possibilité d'amélioré un certain nombre chose tel que:

- L'aspect graphique:

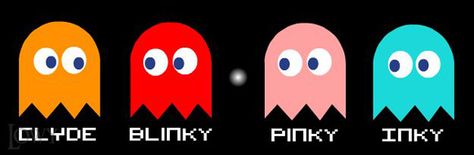
* Texture des murs.



* Amélioration esthétique du menu.

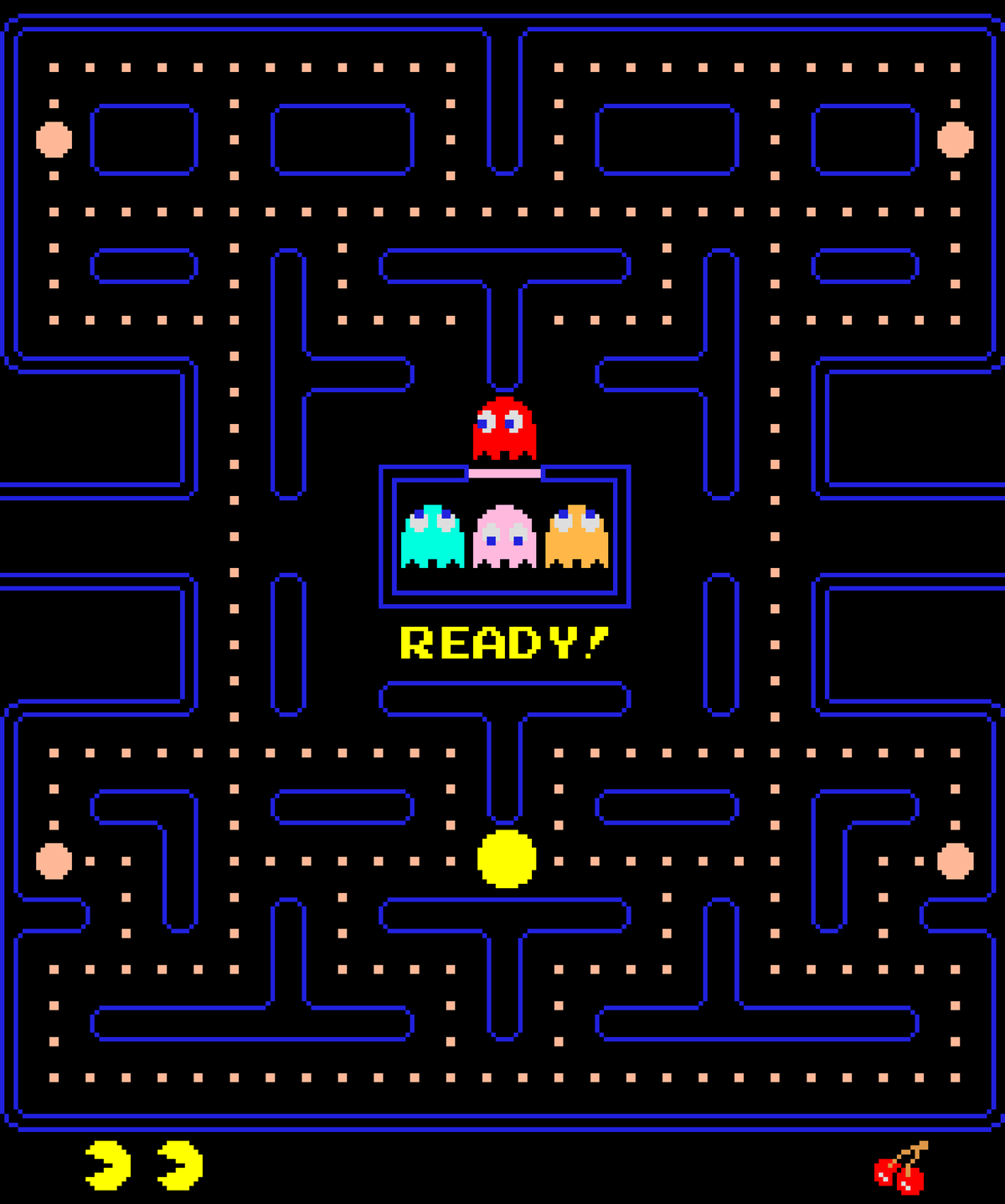
- L'algorithme des fantômes:

* Algorithme personnalisé de Blinky, Pinky, Inky, Clyde.



- La correction de certains bugs:

* Pouvoir faire apparaitre les fantômes au milieu de la carte.



* Automatisation plus performante du lancement et de l’arrêt des musiques.
* Empêcher que le Pac-Man passe à travers des fantômes sans mourir (Lorsque l’on reste appuyer sur une touche directionnelle.)

- Une interface plus ergonomique:

* Ajouter de nouveaux boutons tels que l’option et les crédits.
* Développement d’une plus grande cohérence dans le déroulement des évènements (apparition des musiques, des images, des boutons, déroulement des chargements).

- Ajout de fonctionnalités :

* Ajout du Score, du nombre de vies, affichage du niveau actuel.
* Ajouts de nouveaux bonus tel que les cerises et les super balles.
* Ajouts de nouveaux son et thèmes.
* Ajouts de la cage centrale.
* Ajouts de niveaux/nouvelles cartes.