



Simulateur de carrefour à sens giratoire :
spécifications

Nom du projet	Pac-Man
Livrables	

Cahier des charges

	Nom et prénom	Affiliation	Contact
Auteur principal	Vincent P.	Etudiant	Pierre.vincent.1@etu.univ-amu
Chef de Projet	Vincent P.	Etudiant	Pierre.vincent.1@etu.univ-amu
Approbateur	Deutsch R.	Etudiant	Remi.deutsch@etu.univ-amu
Approbateur	Loignon L.	Etudiant	Lucas.loignon@etu.univ-amu
Approbateur	Fauconnier A.	Etudiant	Axel.fauconnier@etu.univ-amu
Approbateur	Cauet C.	Etudiant	Christopher.cauet@etu.univ-amu

Livré le : 08 /12/2017	Approuvé le : 05 /12/2017	Validé le : 05/12/2017
---------------------------	------------------------------	---------------------------

Entité	Nom et prénom	Mode de distribution
à	Amine HAMRI	Depôt GitHub
Copie à		

Nom du fichier : PacMan.zip	Etat :	Nombre de page :
--------------------------------	--------	------------------

Evolution	Date	Version
Fin du sprint 3	05/12/2017	1.4

Table des matières

I.	Documentation.....	4
1.	Le menu principal	4
2.	Le menu de chargement des profils	5
3.	Règles et principes du jeu.....	6
a)	Différences avec la version originale.....	6
b)	Règles du jeu	6
4.	L'interface de jeu.....	7
a)	ATH	7
b)	Zone de jeu et éléments graphiques	7
c)	Touches de contrôles	8
II.	Détails techniques	8
1.	Implémentation des IAs des fantômes.....	8
1.	Blinky (le fantôme rouge)	8
2.	Inky (le fantôme bleu)	9
3.	Pinky et Clyde (respectivement les fantômes rose et orange)	9
2.	Cas d'utilisations.....	9
3.	Scénarios	10
4.	Diagramme de classes	11
1.	Architecture générale.....	12
2.	Architecture des entités	12
3.	Architecture des objets collectables	12
5.	Générateur de niveau.....	13
6.	Compatibilité	13

Le logiciel livré est un jeu vidéo pouvant être utilisé sous les plateformes Windows et Linux, ce jeu vidéo est un remake du célèbre jeu vidéo des années 80 : Pac-Man. Il permet à l'utilisateur de contrôler un petit disque jaune dans un labyrinthe qui doit, en tant qu'invité et en évitant les ennemis, manger toutes les petites gomme présentes dans le labyrinthe.

Dans la suite de ces spécifications nous expliquerons le fonctionnement et les diverses fonctionnalités offertes par notre produit, puis nous détaillerons le schéma de production avec des scénarios de cas d'utilisation, ainsi que le patron de conception UML et un diagramme de gant.

I. Documentation

1. Le menu principal

Le menu principal est la première fenêtre sur laquelle l'utilisateur va tomber en lançant le programme, elle se présente comme ceci :

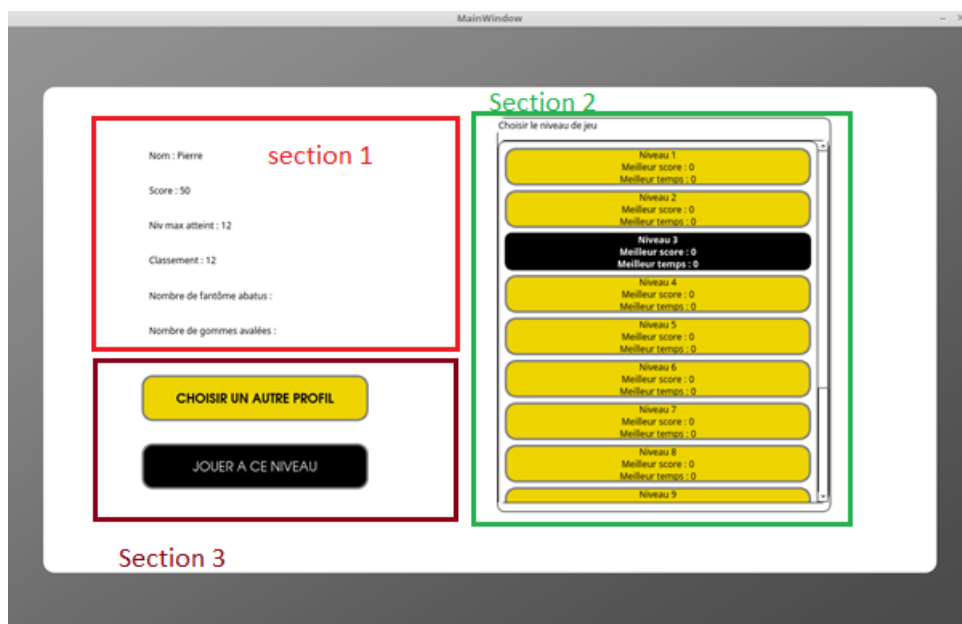


Comme l'on peut le voir sur l'image deux options s'offrent au joueur. La première option proposée est « jouer en tant qu'invité », cette option permet de faire une partie rapide sans sauvegarder son score. Lorsque le joueur choisit cette option le jeu se lance automatiquement au niveau 1 et la progression n'est pas sauvegardée.

La deuxième option disponible est « Charger un profil », cette option permet de choisir dans la liste des profils sauvegardés un profil de joueur puis de jouer avec ce profil. Avec cette option lorsqu'un joueur termine un niveau son profil est actualisé en conséquence et il pourra revenir ultérieurement sur l'application en reprenant là où il s'était arrêté.

2. Le menu de chargement des profils

Le menu de chargement des profils, accessible lorsque le joueur clique sur le bouton « charger un profil » du menu principal, se présente comme cela :



Ce menu comporte trois grandes sections :

- La première section nous informe sur les statistiques du profil chargé. Ces statistiques comprennent : le nom du profil, le score, le niveau maximum atteint, le classement, le nombre de fantômes abattus et le nombre de gommes avalées.
- La deuxième section présente une liste de boutons représentant des niveaux. Une fois le profil chargé, le joueur a le choix entre recommencer un niveau déjà effectué dans le but d'améliorer son score ou de continuer sa progression dans les niveaux du jeu.
- La troisième section présente deux boutons, le premier bouton « choisir un autre profil » permet comme son nom l'indique de choisir un autre profil parmi les profils déjà sauvegardé. Le second bouton « jouer à ce niveau » permet de lancer le niveau sélectionné dans la section 2.

3. Règles et principes du jeu

Dans cette partie nous allons détailler les grands principes ainsi que les règles de notre version du Pac-Man.

a) Différences avec la version originale

Notre version du jeu de Pac-Man apporte quelques modifications au jeu original de Tōru Iwatani, nous avons ajouté une gestion des profils qui permet à un joueur de pouvoir reprendre sa partie même en l'ayant interrompu. Nous proposons également une gestion des statistiques un peu plus poussée que dans le jeu original.

Dans notre version, les sprites utilisés ont été redessinés pour l'occasion (tous les sprites utilisés ont donc une licence qui nous appartient) et ces sprites sont différents des originaux. Cette différence est principalement notable sur les sprites du labyrinthe, en effet nous avons adopté une approche « flat design » qui rend plus simple le dessin des niveaux.

Dans notre version du jeu nous avons ajouté un élément de jeu différent de l'original, le principe de la boîte mystère. La boîte mystère est un élément collectable par le Pac-Man qui va apparaître à une position aléatoire dans le labyrinthe. Lorsque le joueur va collecter cette boîte, d'une part il va augmenter son score (comme avec n'importe quel autre item collectable du jeu) mais il va aussi obtenir un objet aléatoire contenu dans cette boîte. Il va pouvoir garder cet objet jusqu'à ce que le joueur décide de l'utiliser. Lorsque le joueur obtient un tel objet il est stocké dans son « Mystery-slot ».

b) Règles du jeu

Les règles du jeu sont très proches des règles originales des années 80, le joueur incarne le Pac-Man un petit disque couleur orange à l'appétit dévorant et doit manger toutes les gommes du labyrinthe. Le Pac-Man doit absolument éviter les fantômes qui rôdent et le poursuivent dans le labyrinthe, si le joueur touche un fantôme alors Pac-Man perd une vie.

Lorsque Pac-Man n'a plus de vie c'est un Game Over et le joueur doit recommencer le niveau depuis le début.

Lorsque le Pac-Man mange une super gomme il devient insensible aux pouvoirs des fantômes et il peut les manger pour gagner des points.

Un timer se déclenche en début de niveau, si le timer arrive à zéro avant que le joueur ne termine le niveau alors c'est un Game Over.

4. L'interface de jeu

L'interface de jeu est la section la plus importante du logiciel, en effet 90% de l'utilisation du logiciel se fera dans cette fenêtre elle doit donc être à la fois fonctionnelle et ergonomique pour l'utilisateur.

Cette interface se présente comme ceci :

/ insert screen game */*

a) ATH

L'ATH ou Affichage Tête Haute est un grand classique du jeu vidéo, c'est une petite zone placée en périphérie de la zone de jeu qui informe le joueur de l'état de la partie.

Notre ATH va donner au joueur des informations essentielles sur le déroulement de la partie, il va donner ces informations :


- Le nombre de vie restantes de Pac-Man.
- Le score effectué jusqu'à présent.
- Le temps restant avant la fin du timer de partie.
- L'item présent dans le Mystery-slot (ce concept est détaillé à la section « règles et principes du jeu »).

Cet ATH se veut simple et ergonomique, se fondant totalement dans le fenêtre de jeu.

b) Zone de jeu et éléments graphiques





La zone de jeu est délimitée par les bords du labyrinthe, le joueur ne pourra jamais contrôler le Pac-Man en dehors de ces limites.


Plusieurs éléments graphiques sont présents dans cette zone et font partie à proprement parler du jeu en lui-même.


Les zones bleues comme celles-ci :  représente des murs du labyrinthe, le Pac-Man ainsi que les fantômes ne peuvent pas les traverser.


Les zone sombres comme celle-ci :  représente les chemins empruntables par le Pac-Man.

Les points jaunes comme ceci  ou  représentent respectivement les gomme et les supers gomme le Pac-Man doit les récolter.

Les fantômes :     représentent les quatre ennemis mortels du Pac-Man si celui-ci entre en collision avec l'un d'eux le Pac-Man perd une vie.

Les fantômes effrayés :  représente des ennemis affaiblis, si le Pac-Man entre en collision avec l'un d'eux le fantôme se transforme en fantôme mort.

Les fantômes morts :  représente des ennemis vaincus par le Pac-Man leurs collisions avec le Pac-Man ne produit aucun événement.

Les caisses mystère :  représente des objets collectables par le Pac-Man, une fois cet objet collecté le Pac-Man reçoit un item mystère (fonctionnement détaillé à la section « règles et principes du jeu »).

Les fantômes transparents :  représente les points d'apparitions des divers fantômes.

c) Touches de contrôles

Les contrôles du jeu de Pac-Man sont très simples :

- Les flèches directionnelles servent à changer de direction dans le labyrinthe, notez que les changements de direction sont mis en tampons jusqu'à ce que le Pac-Man puissent effectuer le changement de direction.
Exemple : Le Pac-Man se dirige vers la droite et nous effectuons un appui sur la touche « flèche du haut » le Pac-Man ne va pas changer de direction tout de suite mais il va tourner sur la première possibilité à sa gauche.
De cette manière le Pac-Man ne peut pas se coincer dans un mur sur ordre du joueur.⁷
- La touche espace permet d'utiliser l'objet présent dans le Mystery-slot, si le Mystery-slot est vide l'appui sur cette touche ne provoque rien (se référer à la section « règles et principes de jeu » pour la définition du Mystery-slot).

II. Détails techniques

1. Implémentation des IAs des fantômes

Voici le détail sur l'implémentation des IAs des fantômes. Pour chaque niveau du jeu, un graphe appelé « graphe de contrôle » va être généré. Ce graphe représente tous les chemins possibles au sein du labyrinthe et permet l'établissement d'une table de routage avec un algorithme de recherche de plus court chemin. L'algorithme de recherche de plus court chemin utilisé ici est l'algorithme de Dijkstra. Avec cette table de routage et ce graphe il est donc possible d'implémenter les différentes IAs des fantômes décrite ci-dessous.

1. Blinky (le fantôme rouge)

Blinky va toujours chercher à poursuivre le Pac-Man peu importe où celui-ci se trouve dans le labyrinthe. Chaque mouvement de Blinky le rapproche du Pac-Man dans le sens où la trajectoire de Blinky dans le labyrinthe est le plus court chemin le séparant du Pac-Man.

Blinky à la même vitesse que Pac-Man.

2. Inky (le fantôme bleu)

Inky est un peu plus malin que Blinky, en effet il va chercher à se placer en embuscade devant Pac-Man pour l'empêcher de poursuivre son chemin.

La trajectoire de Inky est le plus court chemin le séparant de la prochaine intersection que va devoir franchir le Pac-Man. De cette manière Inky va parfois poursuivre le Pac-Man comme Blinky, mais parfois celui-ci va essayer de faire un détour afin de surprendre le Pac-Man à la prochaine intersection.

Inky à la même vitesse que Pac-Man.

3. Pinky et Clyde (respectivement les fantômes rose et orange)

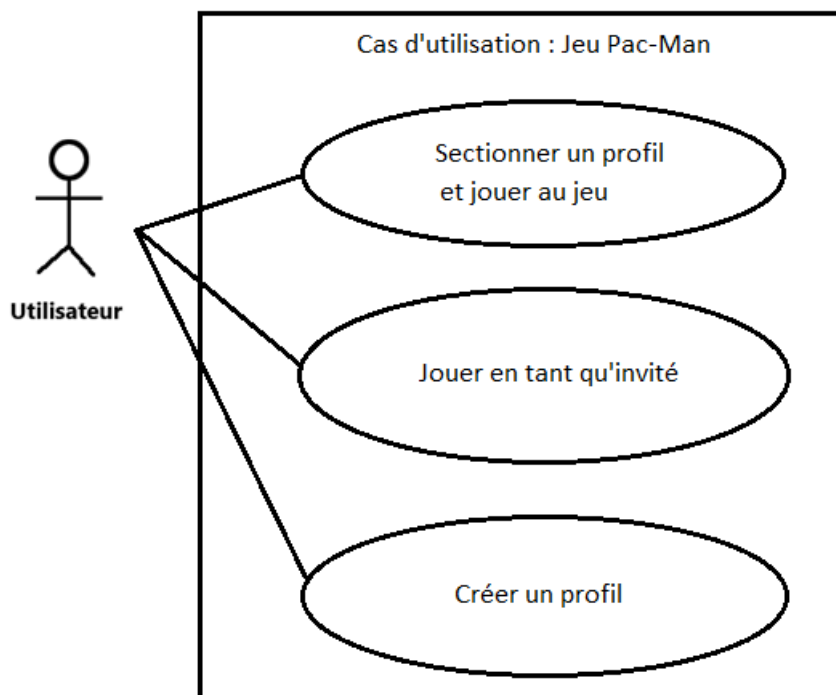
L'IA de Pinky et Clyde est relativement simple, en effet leurs mouvements sont totalement aléatoires. La seule contrainte qu'il connaisse c'est d'éviter de repartir en arrière sauf si cela est nécessaire (cul de sac).

Pinky et Clyde sont plus rapides que le Pac-Man ce qui les rends tout de même dangereux malgré leurs mouvements imprévisibles

2. Cas d'utilisations

Maintenant nous allons aborder les aspects techniques du produit. Dans un premier temps nous présenterons un diagramme de cas d'utilisation avec quelques scénarios de mise en situations, puis nous montrerons la solution adoptée pour la conception du produit.

Voici le diagramme de cas d'utilisation :



Les fonctionnalités prévues pour le produit sont donc :

- Permettre à l'utilisateur la création de son profil de joueur.
- Permettre à l'utilisateur de jouer avec un profil de joueur qu'il choisit.
- Permettre au joueur de faire une partie simple sans enregistrer son score.

3. Scénarios

Titre : Utilisation du Jeu Pac-Man par un joueur.

Acteur : Utilisateur.

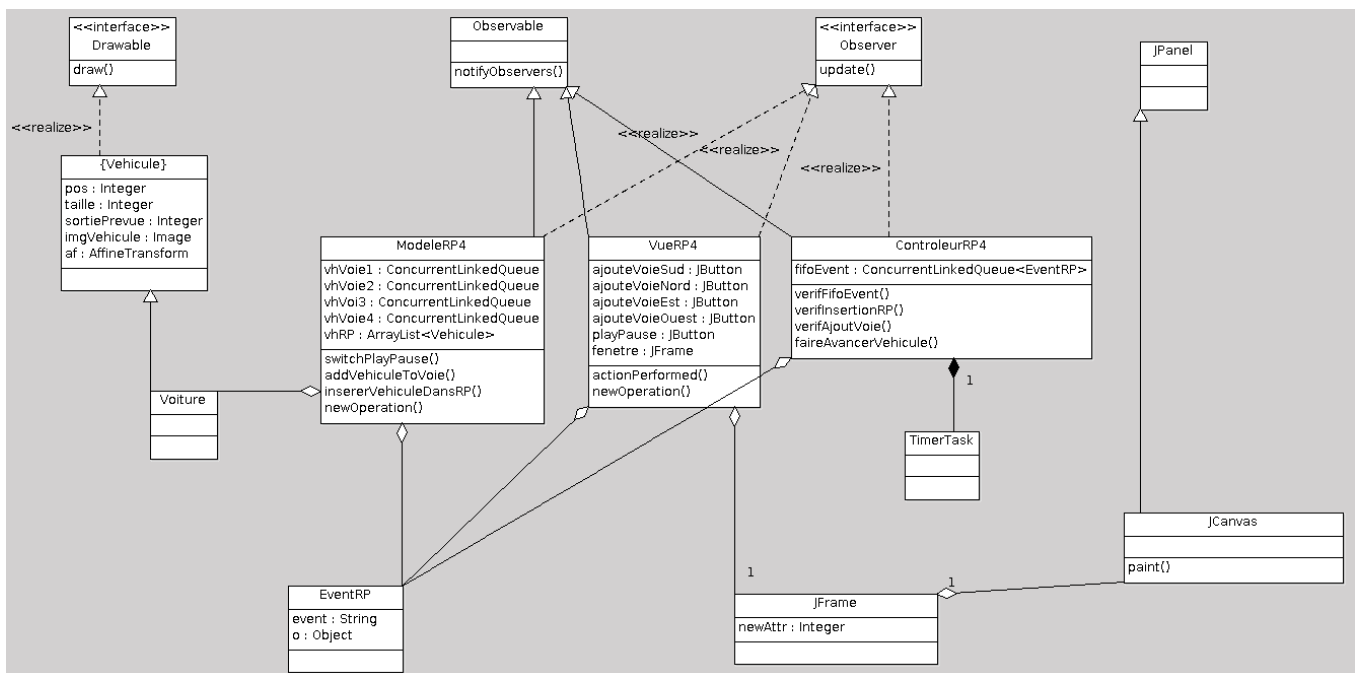
Condition(s) : Lancer le programme.

- **Scénario 0** : L'utilisateur veut effectuer une partie rapide sans enregistrer son score :
 - 1) L'utilisateur lance le programme ;
 - 2) Une fenêtre s'ouvre, il tombe sur le menu général du jeu ;
 - 3) L'utilisateur choisit « jouer en tant qu'invité » ;
 - 4) Le jeu se lance directement sans demander plus d'informations au joueur.
- **Scénario 1** : L'utilisateur créer un profil :
 - 1) L'utilisateur lance le programme ;
 - 2) L'utilisateur tombe sur le menu principal du jeu ;
 - 3) L'utilisateur clique sur le bouton « créer un profil » ;
 - 4) L'utilisateur tombe sur le menu de création de profil ;
 - 5) L'utilisateur remplit les champs puis clique sur le bouton « créer mon profil »
 - 6) Le programme enregistre un nouveau profil dans le dossier correspondant, attention si l'utilisateur rentre un nom de profil déjà utilisé le programme va écraser ce profil pour enregistrer le nouveau profil.
- **Scénario 2** : L'utilisateur veut jouer sur son profil :
 - 1) L'utilisateur lance le programme ;
 - 2) L'utilisateur tombe sur le menu principal du jeu ;
 - 3) L'utilisateur clique sur « charger un profil »
 - 4) L'utilisateur clique sur « choisir un autre profil » et il choisit un profil dans la boîte de dialogue qui s'ouvre ;
 - 5) L'utilisateur clique sur le niveau souhaité puis sur « jouer à ce niveau »
 - 6) Le jeu se lance.
- **Scénario 3** : L'utilisateur veut consulter son profil :
 - 1) L'utilisateur lance le programme ;
 - 2) L'utilisateur tombe sur le menu principal du jeu ;
 - 3) L'utilisateur clique sur « charger un profil »
 - 4) L'utilisateur clique sur « choisir un autre profil » et il choisit un profil dans la boîte de dialogue qui s'ouvre ;
 - 5) L'utilisateur consulte son profil ;

- **Scénario 4** : L'utilisateur termine un niveau :
Condition : l'utilisateur a lancé le jeu.
 - 1) L'utilisateur termine le niveau ;
 - 2) Deux boutons apparaissent à l'écran ;
 - 3) L'utilisateur clique sur « continuer » pour lancer le prochain niveau ou il clique sur « retour » pour revenir au menu de sélection des profils.
- **Scénario 5** : L'utilisateur perd :
Condition : l'utilisateur a lancé le jeu.
 - 1) L'utilisateur perd un niveau ;
 - 2) Deux boutons apparaissent à l'écran ;
 - 3) L'utilisateur clique sur « recommencer » pour recommencer le niveau actuel ou alors il clique sur « retour » pour revenir à l'écran de sélection des profils.

4. Diagramme de classes

Le produit est fourni avec le diagramme de classes complet suivant (une version PNG sera disponible attaché à ce rapport) :



1. Architecture générale

Afin de réaliser cette application le mieux possible nous avons utilisé les fonctionnalités graphiques de Qt qui présente une architecture de type Modèle/Vue (le contrôleur est en effet inclus dans les widgets de Qt).

Plus particulièrement pour l'implémentation de la partie jeu, nous avons utilisé une architecture propre à QT de type QGraphicsView/QGraphicsScene afin de séparer la scène de la vue (ce qui était en effet approprié dans notre cas).

Nous avons quatre vues importantes dans ce logiciel :

- Le menu principal.
- Le menu de sélection des profils.
- Le menu de création des profils.
- Le jeu.

Ces quatre vues sont regroupées au sein d'un widget de Qt : QStackedWidget qui permet de stocker plusieurs widgets mais d'en afficher qu'un seul à l'écran.

2. Architecture des entités

Les entités sont des éléments centraux pour la plupart des jeux vidéo, nous avons décidé d'implémenter ces entités grâce à un héritage de la classe QGraphicsPixmapItem qui permet une gestion propre et efficace des collisions.

La classe Entity dérive donc de la classe QGraphicsPixmapItem, et définit le comportement général d'une entité comme par exemple le fait que cette entité ait une vitesse propre et une manière de se déplacer.

La classe Ghost dérive de la classe Entity et définit le comportement global d'un fantôme comme par exemple l'implémentation de son IA. Cette manière de faire a l'avantage de faciliter l'ajout de nouveau fantôme si besoin.

Exemple : l'ajout d'un nouveau fantôme allié au Pac-Man par exemple.

3. Architecture des objets collectables

Les objets collectables sont présents dans le jeu sous la forme de gommes ou de super gommes mais nous pouvons imaginer d'autres objets collectables qui pourrions enrichir l'expérience de jeu de l'utilisateur. Par conséquent il est nécessaire d'adopter une bonne architecture objet afin que l'ajout de nouveaux objets collectables soit le plus simple possible.

Dans ce souci de simplicité nous avons adopté la solution suivante : une classe Collectables héritant de QGraphicsPixmapItem définit le comportement général des objets collectables comme par exemple les points qu'ils apportent au joueur. La collision est gérée par Qt grâce aux fonctionnalités de la classe QGraphicsPixmapItem, de cette manière il est facile de détecter la collision entre un objet collectable et une entité (tel que le Pac-Man par exemple).

5. Générateur de niveau

Les niveaux sont représentés par des fichiers xml, le nombre d'information contenu dans ces fichiers est assez conséquent. En effet ces fichiers contiennent en fait la « Tilemap » de chaque niveau ainsi que d'autres informations importantes.

De ce fait, écrire ces fichiers de niveau « à la main » est une tâche plus que désagréable sans compter le fait que nous pouvons introduire dans ces fichiers des erreurs qui mettraient le bon fonctionnement de notre application en péril.

C'est pour cette raison que, en parallèle du développement du jeu, nous avons développé un éditeur de niveau qui présente une interface graphique sommaire. Cet éditeur permet de créer un fichier xml de niveau qui sera ensuite utilisé par l'application de jeu.

Ce générateur de niveau sera fourni avec l'application lors du rendu, il a été développé en java dans un souci de rapidité de développement ainsi qu'un souci de compatibilité entre systèmes.

Il sera fourni sous la forme d'un exécutable jar (tapez `java -jar Editeur_de_niveau.jar`) pour le lancer. Une grille vierge de labyrinthe apparaît alors, les commandes d'utilisation de ce générateur sont les suivantes :

- L'appui sur la touche « n » permet de passer en mode normal tous les blocs bleus sont des murs lorsque l'on clique sur un de ces blocs il devient sombre (il représente alors un chemin) il est alors possible de « sculpter » un labyrinthe.
- L'appui sur la touche « p » permet de positionner le point d'apparition du fantôme Pinky.
- L'appui sur la touche « i » permet de positionner le point d'apparition du fantôme Inky.
- L'appui sur la touche « b » permet de positionner le point d'apparition du fantôme Blinky.
- L'appui sur la touche « c » permet de positionner le point d'apparition du fantôme Clyde.
- L'appui sur la touche « d » permet de positionner une super gomme.
- L'appui sur la touche « j » permet de positionner le point d'apparition du joueur.
- Le bouton en bas de la fenêtre sert à écrire le niveau actuel dans un fichier « `xml_level.xml` » qui devra être renommé en « `xml_levelX.xml` » avec X étant le numéro du niveau que l'on souhaite créer et placé dans le répertoire des niveaux.

6. Compatibilité

L'application a été entièrement réalisée avec le Framework Qt5, nous fournirons une version exécutable sous Windows et sous Linux.