

Dossier ISN

Dossier rédigé par :
Dubard Loïc (TS1)

en groupe avec :
Dinnat Laura (TS2)



Année scolaire 2015-2016
Lycée Ambroise Vollard

Résumé

Pac-man est un jeu vidéo sorti au Japon le 22 mai 1980. Le jeu consiste à déplacer Pac-Man, un personnage qui, vu de profil, ressemble à un diagramme circulaire à l'intérieur d'un labyrinthe, afin de lui faire manger toutes les pac-gommes qui s'y trouvent en évitant d'être touché par des fantômes.

Pac-Man est souvent crédité en tant que point de repère dans l'histoire du jeu vidéo, et est parmi les plus célèbres jeux d'arcade de tous les temps. Il est également l'un des jeux vidéo avec les plus grosses recettes de tous les temps, ayant généré plus de 2,5 milliards de dollars au cours des trimestres des années 1990.

Table des matières

1	Introduction	4
1.1	Objectif	4
1.2	Organisation	4
1.3	Problèmes rencontrés	4
2	Réalisation du projet	5
2.1	Logiciels utilisés	5
2.2	Programme commenté	6
2.2.1	Création de la carte	7
2.2.2	Gestion des scores et du résultat(game over ou gagné)	7
2.2.3	Création du personnage	7
2.2.4	Gestion des fantômes	9
2.2.5	Barre de statut	10
2.2.6	Le script du Menu	10
3	Conclusion	10
3.1	Compte rendu	10
3.2	Ouverture	10
A	Annexes	I
A.1	Programme principal	I
A.2	Script du menu	IX
A.3	Code du fichier des niveaux	XI

1 Introduction

1.1 Objectif

Lors de la réflexion pour notre projet nous avons tout de suite pensé au domaine du jeu vidéo. Après l'étude de l'histoire des jeux vidéos, "Pac-man" a retenu notre attention.

Recréer le jeu et les niveaux à notre guise, nous semblait intéressant. C'est pourquoi nous nous sommes lancés dans la conception de ce jeu des années 80. Avec l'interface graphique Tkinter de python nous avons voulu recréer ce monde et comprendre comment créer des personnages, des images et des maps pour entrer un peu plus en détail dans le domaine du jeu vidéo.

Nous ne voulions pas uniquement jouer mais aussi comprendre et savoir construire le jeu. Nous voulions en faire un, mais légèrement simplifié afin que le travail "rentre dans nos cordes" et les délais.

1.2 Organisation

Pour finir en temps et en heure nous avons adopté une certaine façon de travailler permettant la compréhension totale du programme par les 2 membres du groupe.

En cours nous nous fixons un objectif commun et chacun travaille de son côté puis nous faisons une mise en commun dès que quelque chose d'intéressant est trouvé. À la maison chacun de son côté, dans son temps libre peut faire une tâche qui manque au programme en prévenant sur le dossier Drive pour que son binôme ne la fasse pas.

La création de ce [dossier Google Drive](#) permet le transfert des documents facilement et que chacun puisse accéder au journal de bord et voir les modifications en direct. C'est donc une manière de travailler ensemble en même temps malgré l'éloignement.

1.3 Problèmes rencontrés

Lors de notre progression sur l'avancement de notre projet, nous avons dû faire face à de nombreuses contraintes.

Notre premier obstacle fut la création d'un personnage animé. Après plusieurs recherches non fructueuses sur l'intégration d'un gif animé dans le canevas de Tkinter, nous nous sommes rabattus sur les fonctions de base du canvas, comme expliqué plus bas dans la partie création du personnage.

Ensuite, de nombreux problèmes liés à la fonction collision avec les murs nous ont demandé beaucoup de temps et réflexion. En effet, la zone autour d'un mur où le pacman ne peut pas avancer a été déterminée après plusieurs heures de tests de valeurs et a demandé une séparation des tests en fonction de chaque direction. De plus, lors d'un changement de niveau, lors de la remise à zéro du canevas, tous les items sont effacés. Ce qui fait que la fonction collision se retrouvait avec des items qui n'existaient plus et conduisait à un arrêt du programme. Le problème a été résolu en ignorant la fonction collision pendant le changement de niveau.

Enfin, le dernier gros obstacle qui nous a fait face se trouvait au niveau 2, les trous créés posaient problème au déplacement des fantômes. En effet, lors de leurs passages dans les trous les fantômes se retrouvaient décalés dans leurs déplacements ou totalement hors de la map, donc hors de vue, idem pour le pac-man. Il a fallu encore tâtonner et trouver les bonnes valeurs pour la téléportation et restreindre la zone de changement de direction pour le joueur au labyrinthe pour se débarrasser de ce souci.

2 Réalisation du projet

2.1 Logiciels utilisés

Nous avons donc utilisé la version 2.7 de Python nous permettant de coder notre programme à l'aide des bibliothèques suivantes :

- Tkinter pour l'interface graphique
- os pour pouvoir exécuter le programme à partir du menu
- random pour le déplacement aléatoire des fantômes

Nous avons donc choisi d'utiliser Python pour ce projet car d'une part, c'est le langage sur lequel la plus grande partie de notre programme théorique de ISN est basé. D'autre part, ce langage nous a semblé le plus facile à utiliser, et le plus beau aussi au niveau de la forme et du rendu du code, car plus lisible par le codeur.

Pour l'interface graphique, comme la seule interface graphique étudiée en cours sur python est Tkinter ce fut donc évident que notre choix se porte sur cette interface ci. Aucun projet abouti de Pacman développé sur python avec Tkinter n'est trouvable sur Internet. Les interfaces graphiques sur lesquelles pacman a déjà été développé avec python sont pyGame, pyGTK, pyQT, pyglet et beaucoup d'autres interfaces encore. Nous ne savions donc pas si ce projet pouvait aboutir.

2.2 Programme commenté

Voici les captures d'écran du rendu graphique final :



Vous trouverez le code complet du programme en [annexe](#).

L'importation des bibliothèques aux lignes 1 à 6 : chacune permet d'ajouter une fonctionnalité au programme.

-le module random est utilisé pour donner un nombre aléatoire, utilisé pour définir aléatoirement la direction des fantômes dans la fonction `deplacementfantomes()`.

-le module `tkMessageBox` est utilisé pour afficher des boîtes de dialogue, des petites fenêtres de message, comme par exemple lorsque l'on perd ou l'on gagne.

-le module `os` permet de rajouter la possibilité d'utiliser les commandes du système, et plus précisément ici de lancer un autre script, ou programme, avec la commande `os.system("nomdufichier")`, directement à partir du script en fonctionnement.

Pour info, La ligne 12 permet d'afficher l'icône `pacman.gif` dans la barre de titre du programme et la barre d'état du système.

2.2.1 Création de la carte

Pour la création et l’affichage de la map en fonction du niveau, dans le widget canvas, nous utiliserons un système de grille modélisé par une variable tableau à deux dimensions : ici `tab_plateau[i][j]`.

Cette variable tableau contiendra des lettres codées pour chaque motif : exemple, M code pour un mur, X pour le personnage, 0 pour un bonbon, B, C, I et P pour les différents fantômes. (**voir plus bas pour le fichier `niveau.lvl`**) Ces lettres sont alors interprétées par une fonction (`interprete(ligne)`) qui traduit alors le code du niveau en éléments graphiques dans un canevas : d’abord cela crée l’item du pacman qui n’est en réalité qu’un arc de cercle rempli de jaune puis les murs qui ne sont que de simples carrés remplis de bleu et enfin les fantômes qui sont des images gifs implantés dans python.

Les id des items correspondants aux murs sont enregistrés dans une liste `listemurs[]` pour être utilisés dans la fonction `collision` (**voir plus bas**).

Cette fonction est appelée par la fonction `afficher(niveau)` qui va préparer le jeu pour un niveau : elle efface tous les items présents dans le canvas et réinitialise la grille, la variable tableau. Cette dernière va ouvrir notre fichier s’il existe et lire la ligne correspondante au niveau. La ligne lue est enregistrée dans une variable pour être interprétée plus tard. Cette fonction va aussi, si la ligne lue est vide (c’est la dernière ligne du coup) nous afficher un message pour nous informer que l’on a atteint le dernier niveau.

Elle fait donc appel à la fonction `existe(fname)` qui permet de tester la présence du fichier contenant les niveaux dans le répertoire du script. On essaie alors d’ouvrir le fichier : s’il est lu cette fonction retourne `True` sinon `False`.

2.2.2 Gestion des scores et du résultat(game over ou gagné)

La fonction `verifgagne()` teste si le pac-man a mangé tous les bonbons, c’est à dire si la liste contenant tous les id des item du canvas correspondants aux bonbons est vide. Par ce fait, on passera au niveau suivant. (cette fonction est appelée **plus bas** par `avance()`)

Sinon la fonction `gameover()` testera la collision entre le pac-man et les fantômes. Lors d’une collision, quand les coordonnées du pacman sont égales aux coordonnées du fantôme, le joueur perd.

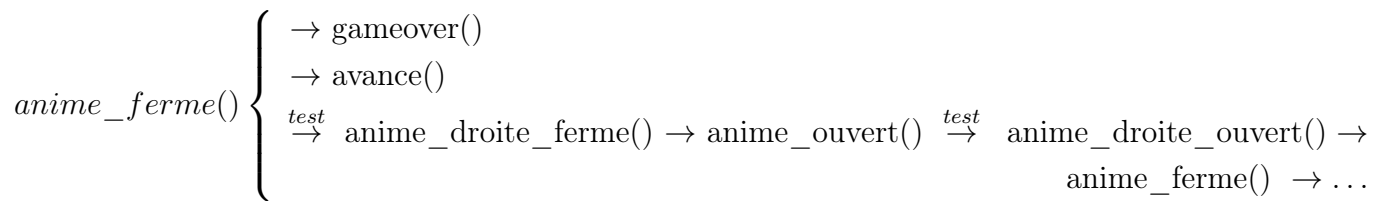
Pour compter les points et faire en sorte que les bonbons s’effacent du canvas quand Pacman passe par dessus, on utilise la fonction `mange()`. De plus celle-ci efface l’id de l’item de la liste des bonbons, pour permettre le test de la fonction `verifgagne()` et modifie le label de la barre de statut à chaque fois qu’un bonbon est mangé.

2.2.3 Création du personnage

Pour l’animation du pac-man on utilise deux principales fonctions `anime_ferme()` et `anime_ouvert()` qui gèrent des sous-fonctions chacune correspondant à une direction particulière du pacman. Ses sous-fonctions possèdent une instruction qui modifie la configuration de l’item de pac-man du canvas.

Les sous-fonctions sont toutes de la même forme (avec juste l’angle du début de l’arc de cercle qui change en fonction de la direction) : quand la bouche est ouverte l’angle coloré en jaune est de 270° alors que quand la bouche est fermée l’angle coloré est de 359° . Il y a donc 2 sous fonctions par direction, une pour l’ouverture et l’autre pour la fermeture de la bouche.

Cette partie du programme fonctionne comme un cycle, de cette manière : (exemple avec la direction droite : la flèche signifie "appelle") :



Pour permettre au joueur de contrôler la direction du pac-man, on affecte les directions aux touches grâce à la fonction `callback()`. Dans notre cas nous avons choisi les touches fléchées.

La zone où le pac-man peut changer de direction a été déterminée suite à la rencontre de plusieurs bugs. En effet, le pac-man dans le niveau 2 pouvait changer de direction avant le respawn de l'autre côté de la map en passant par les trous. Ainsi le pac-man pouvait se retrouver hors de la map. Grâce à cette délimitation nous ne rencontrons plus ce "glitch".

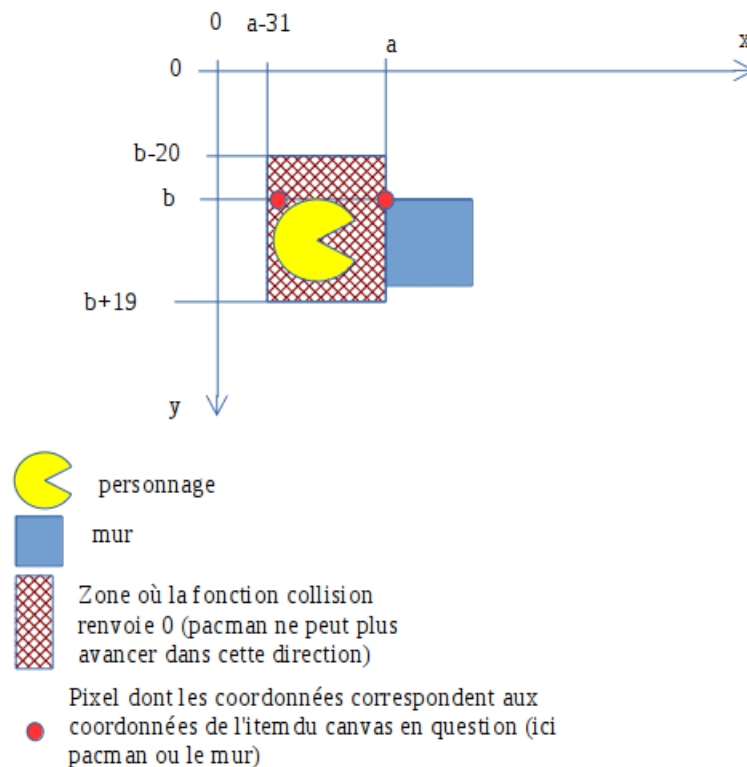
Le déplacement de pac-man est géré par la fonction `avance()`. Le pacman est donc déplacé par incréments ou décréments de 25 pixels dans ses coordonnées, uniquement si la fonction `collision` le permet (introduite plus bas).

La téléportation est possible grâce à la zone de délimitation créée dans cette fonction. C'est à dire que lorsque les coordonnées du personnage sont au delà des coordonnées du dernier mur, le personnage est téléporté de l'autre côté du labyrinthe.

On peut voir dans la fonction `avance()` que l'appel de la fonction `verifgagne()` nous permet de passer au niveau suivant et d'afficher un message de notification du succès au joueur.

La fonction `collision` est non seulement utilisée par le pac-man, mais aussi par les fantômes, que nous introduirons plus tard. Celle-ci teste pour une direction précise la proximité du pac-man avec tous les murs ; au delà d'une certaine distance (dépendante de la direction) entre le pac-man et le mur testé son mouvement est arrêté. Par exemple, pour la direction droite, la zone de test d'arrêt autour d'un mur est comme ci-dessous :

Schéma du principe de détection de collision avec un mur pour la direction droite.



Pour les direction gauche, bas et haut il suffit de déplacer la zone autour du mur en fonction de chaque direction. (les valeurs ont été trouvées en tâtonnant)

Les instructions :

```
try :
```

```
...
```

```
except :
```

```
pass
```

permettent le changement de niveau, sans le bug rencontré à cause de la fonction collision, lors d'un passage au niveau suivant.

2.2.4 Gestion des fantômes

Les fantômes sont gérés par 3 fonctions :

-une première : `deplacementfantomes()` pour donner une direction aléatoire en fonction des murs se situant autour du fantôme en paramètre de la fonction. Elle fonctionne grâce à l'instruction `random.randint(a,b)` du module `random` qui donne un nombre entier aléatoire entre `a` et `b` inclus.

-une deuxième pour faire avancer le fantôme en fonction de la direction choisie par `deplacementfantome()` et du placement du fantôme dans le labyrinthe. Le déplacement des fantomes se fait de la même manière que celui du pacman, tout les 25 pixels ;

-une dernière : `fantome()` pour appliquer les deux premières sur chaque fantôme (donc sur tous). Elle fonctionne en appliquant tour à tour `deplacementfantomes()` puis `avancefantome()` à blinky, ensuite la première fonction puis la deuxième à pinky, et ainsi de suite pour tous les fantômes.

Cette fonction est l'une des deux principales (au côté de la fonction `anime_fermer()`) qui appellent toutes les autres cycliquement.

La zone de délimitation (lignes 325 à 336 dans `avancefantome()`) sert quand le fantôme sort de la map au niveau 2 par exemple et la ligne 341 est rajoutée à cause d'un bug où le fantôme s'arrête un court moment avant de changer de direction devant un mur. Elle fonctionne de la même manière que pour la zone du pacman.

2.2.5 Barre de statut

La barre de statut est simplement composée de 2 labels (un pour l'affichage des scores et l'autre pour l'affichage du niveau) et d'un bouton de retour au menu. Ce bouton déclenche une fonction simple constituée d'une instruction pour fermer la fenêtre et d'une autre pour ouvrir le script menu (`os.system()` issue du module `os`).

Le label score est modifié lors de chaque déplacement par la fonction `mange()`, elle même appelée par la fonction `avance()` qui est appelée par la fonction principale `anime_ouvrir()`.

2.2.6 Le script du Menu

Le menu est simplement constitué de un label "rempli" avec une photo et de trois boutons :
-jeu menant vers une fonction lançant le programme principal (avec l'instruction `os.system()` provenant du module `os`) et fermant la fenêtre menu
-règles pour lancer une fonction qui affiche les principes de jeu sous la forme d'un Label qui se crée tout en bas de la fenêtre
-info qui affiche une petite fenêtre, grâce à l'instruction `show_info()` issue du module `tkMessageBox`, contenant des informations peu utiles
Vous trouverez le script complet du fichier Menu en [annexe](#).

3 Conclusion

3.1 Compte rendu

Nous avons donc réussi à terminer notre projet dans le temps imparti, de manière à ce qu'il soit présentable. Nous avons apprivoisé et maîtrisé la programmation de jeu graphique en 2D sur python avec Tkinter.

3.2 Ouverture

Si le temps nous le permettait nous aurions aimé pouvoir rajouter d'autres fonctionnalités et des niveaux supplémentaires. Les fonctionnalités à rajouter sont entre autre :

-un bonus temps (non présent dans la version originale). Ce bonus temps consiste en un genre de minuteur qui est au maximum au début de chaque niveau (quelque chose comme 150 points pour 2 minutes 30 secondes : 150s) les points bonus restants en fin de niveau sont incrémentés aux points totaux.

-des vies pour le personnage qui sont au nombre de 3 au début de partie et conservées au passage au niveau suivant. C'est à dire que si on finit notre niveau avec 2 vies au lieu de 3 on aura toujours 2 vies au niveau suivant (pas 3). De plus, le nombre de points peut, s'il dépasse une certaine valeur, rapporter une vie de plus au joueur.

-les 4 bonbons spéciaux par niveau qui permettent de changer les fantômes en fantômes peureux, auquel pacman n'est plus vulnérable, et de les manger. Ils réapparaissent, "respawn" (ou "repop" en langage de "gamer") à leur emplacement de première apparition après avoir été dévorés et reprennent leur chasse.

-Les items spéciaux qui valent plus de points que les petits bonbons blancs normaux et apparaissant chacun à partir d'un niveau. les items spéciaux sont : la cerise qui vaut 100 points, la fraise 300, l'orange 500, la pomme : 700, le melon : 1000, le galboss : 2000, cloche : 3000 et la clé : 5000. (quand le joueur dépasse les 10000 points, il obtient une vie supplémentaire)

-des IA pour les fantômes (identiques ou différents de l'original). Ceux de l'original se comportent de cette façon : blinky (le rouge, *l'Ombre*) attaque directement le pacman, il le suit comme son ombre, pinky (le rose, *la Rapide*) a tendance à se mettre en embuscade, elle vise l'emplacement où va se trouver le pacman, inky (Le bleu, *le Timide*) de temps en temps part dans la direction opposée au pacman. Clyde (le orange, *le Limité*) de temps en temps, choisit une direction au hasard qui peut être celle du pacman, il fait l'indifférent.

-rajout d'effets sonores différents de l'original et d'une musique de type électro, beaucoup plus entraînante et moins répétitive que dans l'original.

Pour les niveaux, une petite idée nous a effleuré l'esprit, celle d'ajouter des niveaux avec des labyrinthes de tailles différentes les uns des autres et beaucoup plus colorés ou même d'une taille gigantesque donnant l'impression que c'est l'écran qui se déplace sur le labyrinthe (un peu comme dans les jeux Pokémon). Le rajout de fantômes fut aussi une possibilité envisagée, de même que la possibilité de rajouter d'autres pacmans et pouvoir y jouer à plusieurs (avec une image ou "skin" personnel pour ne pas se tromper de personnage).

Enfin, la dernière idée est de proposer un éditeur de niveau pour rajouter facilement des nouveaux niveaux et modifier ceux déjà existants, à notre guise.

A Annexes

A.1 Programme principal

pacmanlabyrinthesuite.py

```
1  #!/usr/bin/python2.7
2  # -*- coding: UTF8 -*-
3  from Tkinter import *
4  import random
5  import tkMessageBox
6  import os
7
8  fenetre=Tk()
9  fenetre.title("Pacman-DubardLoïc_DinnatLaura")
10 fenetre.configure(bg="black")
11 icone=PhotoImage(file="pacman.gif")
12 fenetre.tk.call('wm', 'iconphoto', fenetre._w, icone)
13 #####variables par
14 default#####
15 NB_BLOCS=16
16 NB_LIGNES=24
17 LARGEUR_BLOC=25
18 CENTRE=LARGEUR_BLOC / 2
19 fichegrille='niveaux.lvl'
20 deplacement=["droite","gauche","haut","bas"]
21 direction="droite"
22 directioninky=deplacement[random.randint(0,len(deplacement)-1)]
23 directionpinky=deplacement[random.randint(0,len(deplacement)-1)]
24 directionclyde=deplacement[random.randint(0,len(deplacement)-1)]
25 directionblinky=deplacement[random.randint(0,len(deplacement)-1)]
26 bougepacman=1
27 accelerationx=0
28 accelerationy=0
29 NIVEAU=1
30 listemurs=[]
31 listeboules=[[[]],[[]]]
32 tab_plateau=[]
33 #chargement des images
34 blinkyimg=PhotoImage(file='blinky.GIF')
35 pinkyimg=PhotoImage(file='pinky.GIF')
36 inkyimg=PhotoImage(file='inky.GIF')
37 clydeimg=PhotoImage(file='clyde.GIF')
38 points=0
39 #####création de la
40 carte#####
41 canvas1=Canvas(fenetre,width=NB_BLOCS*LARGEUR_BLOC,height=NB_LIGNES*LARGEUR_BLOC,bg="black")
42 canvas1.pack()
43
44 for i in range(NB_BLOCS):
45     tab_plateau.append([' '] * NB_LIGNES)
46
47 def existe(fname):
48     try:
49         f = open(fname,'r') #ouvre le fichier en lecture
50         f.close() #referme le fichier
51         return True #tout est ok
52     except:
53         #code exécuté si ça n'a pas marché
54         return False #retourne : ça n'a pas marché
55
56 def afficher(NIVEAU):
57     global LARGEUR_BLOC,NB_BLOCS,NB_LIGNES
```

```
55     #interprete le code de la ligne du NIVEAU lu
56     canvas1.delete(ALL)
57     i,j=0,0
58     for i in range(NB_BLOCS):
59         for j in range(NB_LIGNES):
60             tab_plateau[i][j]=" "
61     #ouvre le fichier s'il existe
62     if existe(fichegrille):
63         fich=open(fichegrille,"r")
64         i= 1
65         while i<=NIVEAU:
66             ligne=fich.readline()
67             i=i+1
68             if ligne == "":
69                 tkMessageBox.showinfo("GAGNE","Vous avez atteint le dernier NIVEAU
70                 disponible.")
71             else :
72                 interprete(ligne)
73                 fich.close
74     else:
75         tkMessageBox.showerror("ERREUR","fichier de NIVEAU introuvable")
76 def interprete(ligne):
77     i,j=0,0
78     global
79     x,y,pacman,listemurs,blinky,clyde,inky,pinky,LARGEUR_BLOC,NB_BLOCS,NB_LIGNES
80     for lettre in ligne:
81         if lettre=="C" or lettre=="P" or lettre=="M" or lettre=="I" or lettre=="B"
82         or lettre=="0":
83             tab_plateau[i][j]=lettre
84             elif lettre=="X": #position initiale du personnage
85                 x,y=i*LARGEUR_BLOC,j*LARGEUR_BLOC
86                 pacman=canvas1.create_arc(x,y,x+25,y+25,fill="yellow",start=45,
87                 extent=270)
88                 tab_plateau[i][j]=" " #position du perso initial, inutile de la
89                 conserver, elle va changer
90             if i<NB_BLOCS-1:
91                 i=i+1
92             elif j<NB_LIGNES-1:
93                 j=j+1
94                 i=0
95             for i in range(NB_BLOCS):
96                 for j in range(NB_LIGNES):
97                     if tab_plateau[i][j]=="M" :
98                         listemurs.append(canvas1.create_rectangle(i*LARGEUR_BLOC,j*LARGEUR_B
99                         LOC,\
100                             i*LARGEUR_BLOC+20,j*LARGEUR_BLOC+20,
101                             fill="blue"))
102                     elif tab_plateau[i][j]=="0":
103                         listeboules[0].append(canvas1.create_rectangle(i*LARGEUR_BLOC+10,\
104                             j*LARGEUR_BLOC+7,i*LARGEUR_BLOC+15,j*LARGEUR_BLOC+12,
105                             fill="white"))
106             for i in range(NB_BLOCS):
107                 for j in range(NB_LIGNES):
108                     if tab_plateau[i][j]=="B":
109                         blinky=canvas1.create_image(i*LARGEUR_BLOC,j*LARGEUR_BLOC,\
```

```
103             image=blinkyimg,anchor='nw')
104     elif tab_plateau[i][j]=="P":
105         pinky=canvas1.create_image(i*LARGEUR_BLOC,j*LARGEUR_BLOC,\
106             image=pinkyimg,anchor='nw')
107     elif tab_plateau[i][j]=="I":
108         inky=canvas1.create_image(i*LARGEUR_BLOC,j*LARGEUR_BLOC,\
109             image=inkyimg,anchor='nw')
110     elif tab_plateau[i][j]=="C":
111         clyde=canvas1.create_image(i*LARGEUR_BLOC,j*LARGEUR_BLOC,\
112             image=clydeimg,anchor='nw')
113 afficher(NIVEAU)
114 #####gestion des niveaux des scores et du resultat (game over ou
gagné)#####
115 def verifgagne():
116     if listeboules==[[],[ ]]:
117         return True
118     else:
119         return False
120
121 def gameover(x,y):
122     xblinky,yblinky=canvas1.coords(blinky)
123     xinky,yinky=canvas1.coords(inky)
124     xclyde,yclyde=canvas1.coords(clyde)
125     xpinky,ypinky=canvas1.coords(pinky)
126     if (x==xblinky and y==yblinky) or (x==xpinky and y==ypinky) or (x==xinky and
y==yinky)\
127         or (x==xclyde and y==yclyde):
128         tkinter.messagebox.showinfo("Game Over","VOUS AVEZ PERDU, points :"+str(points))
129         menu()
130
131 def mange(x,y,direction): #la fonction qui permet de manger les bonbons
132     global points
133     for i in listeboules[0]:
134         a,b,c,d=canvas1.coords(i)
135         if (direction=="droite" and ((x>=a-15 and x<=a) and (y>=b-15 and y<=b))\
136             or (direction=="gauche" and ((x>=a-15 and x<=a-10) and (y>=b-15 and
y<=b-5)))\
137             or (direction=="haut" and ((x>=a-15 and x<=a-5) and (y>=b-15 and
y<=b-5)))\
138             or (direction=="bas" and ((x>=a-15 and x<=a-5) and (y>=b-15 and y<=b))):
139             canvas1.delete(i)
140             listeboules[0].remove(i)
141             points=points+1
142             score.config(text="SCORE :"+str(points))
143 #####création du personnage pacman#####
144 def anime_droite_ouvert():
145     canvas1.itemconfig(pacman,start=45,extent=270,fill="yellow")
146     fenetre.after(100,anime_fermer)
147 def anime_droite_ferme():
148     canvas1.itemconfig(pacman,start=0,extent=359,fill="yellow")
149     fenetre.after(100,anime_ouvrir)
150
151 def anime_gauche_ouvert():
152     canvas1.itemconfig(pacman,start=225,extent=270,fill="yellow")
153     fenetre.after(100,anime_fermer)
154 def anime_gauche_ferme():
```

```
155     canvas1.itemconfig(pacman,start=180,extent=359,fill="yellow")
156     fenetre.after(100,anime_ouvrir)
157
158     def anime_haut_ferme():
159         canvas1.itemconfig(pacman,start=90,extent=359,fill="yellow")
160         fenetre.after(100,anime_ouvrir)
161     def anime_haut_ouvert():
162         canvas1.itemconfig(pacman,start=135,extent=270,fill="yellow")
163         fenetre.after(100,anime_fermer)
164
165     def anime_bas_ferme():
166         canvas1.itemconfig(pacman,start=270,extent=359,fill="yellow")
167         fenetre.after(100,anime_ouvrir)
168     def anime_bas_ouvert():
169         canvas1.itemconfig(pacman,start=315,extent=270,fill="yellow")
170         fenetre.after(100,anime_fermer)
171
172     def anime_fermer():
173         global direction,accelerationx,accelerationy,x,y
174         gameover(x,y)
175         avance(accelerationx,accelerationy)           #appel de la fonction avance pour
176         pacman
177         if direction=="droite":
178             anime_droite_ferme()
179         elif direction=="gauche":
180             anime_gauche_ferme()
181         elif direction=="haut":
182             anime_haut_ferme()
183         elif direction=="bas":
184             anime_bas_ferme()
185
186     def anime_ouvrir():
187         global direction,accelerationx,accelerationy,x,y
188         gameover(x,y)
189         if direction=="droite":
190             anime_droite_ouvert()
191         elif direction=="gauche":
192             anime_gauche_ouvert()
193         elif direction=="haut":
194             anime_haut_ouvert()
195         elif direction=="bas":
196             anime_bas_ouvert()
197     #####détection des commandes du joueur#####
198     def callback(event):
199         global direction,accelerationx,accelerationy
200         if event.keysym=="Up":
201             if x>0 and x<375: #zone où le pac-man peut changer de direction
202                 direction="haut"
203                 accelerationx = 0
204                 accelerationy = -25
205
206         elif event.keysym=="Down":
207             if x>0 and x<375: #zone où le pac-man peut changer de direction
208                 direction="bas"
209                 accelerationx = 0
210                 accelerationy = 25
```



```
211     elif event.keysym=="Left":
212         if y>0 and y<575:             #zone où le pac-man peut changer de direction
213             direction="gauche"
214             accelerationx = -25
215             accelerationy = 0
216
217     elif event.keysym=="Right":
218         if y>0 and y<575:             #zone où le pac-man peut changer de direction
219             direction="droite"
220             accelerationx = 25
221             accelerationy = 0
222 fenetre.bind("<Key>",callback)
223
224 def avance(gd, hb):
225     global direction,x, y,bougepacman
226     global NIVEAU
227     if x<0:                #
228         x=400              #
229     elif x>375:           #
230         x=-25              #Zone de delimitation pour la téléportation du Pac-man
231     elif y<0:            #
232         y=600              #
233     elif y>575:         #
234         y=-25              #
235     if collision(x,y,direction)==1:
236         x, y = x +gd, y +hb
237         canvas1.coords(pacman, x, y, x+25, y+25)
238         mange(x,y,direction)
239     if verifgagne()==True:
240         tkMessageBox.showinfo("Gagné", "NIVEAU "+str(NIVEAU)+" réussi")
241         NIVEAU= int(NIVEAU)+1
242         tkMessageBox.showinfo("NIVEAU suivant :", "passage au NIVEAU:"+str(NIVEAU))
243         niv.config(text="niveau:"+str(NIVEAU))
244         afficher(NIVEAU)
245
246 #####déplacements automatiques de
pacman#####
247 def collision(x,y,direction):
248     global bougepacman,listemurs
249     if direction=="droite":
250         bougepacman=1
251         for i in listemurs:
252             try:
253                 a,b,c,d=canvas1.coords(i)
254                 if ((x>=a-31 and x<=a) and (y>=b-20 and y<=b+19)):
255                     bougepacman=0
256             except:
257                 pass
258     elif direction=="gauche":
259         bougepacman=1
260         for i in listemurs:
261             try:
262                 a,b,c,d=canvas1.coords(i)
263                 if ((x>=a and x<=a+25) and (y>=b-20 and y<=b+19)):
264                     bougepacman=0
265             except:
266                 pass
```

2

```
267     elif direction=="haut":
268         bougepacman=1
269         for i in listemurs:
270             try:
271                 a,b,c,d=canvas1.coords(i)
272                 if ((x>=a-20 and x<=a+19) and (y>=b and y<=b+25)):
273                     bougepacman=0
274             except:
275                 pass
276     elif direction=="bas":
277         bougepacman=1
278         for i in listemurs:
279             try:
280                 a,b,c,d=canvas1.coords(i)
281                 if ((x>=a-20 and x<=a+19) and (y>=b-31 and y<=b)):
282                     bougepacman=0
283             except:
284                 pass
285     else:
286         bougepacman=1
287     return bougepacman
288
289     #####création des
290     fantômes#####
291
292 def deplacementfantomes(fantome,direction):
293     x,y=canvas1.coords(fantome)
294     deplacement=["droite","gauche","haut","bas"]
295     for i in listemurs:
296         try:
297             a,b,c,d=canvas1.coords(i)
298             if ((x>=a-31 and x<=a) and (y>=b-20 and y<=b+19)):
299                 deplacement.remove("droite")
300             if ((x>=a and x<=a+25) and (y>=b-20 and y<=b+19)):
301                 deplacement.remove("gauche")
302             if ((x>=a-20 and x<=a+19) and (y>=b and y<=b+25)):
303                 deplacement.remove("haut")
304             if ((x>=a-20 and x<=a+19) and (y>=b-31 and y<=b)):
305                 deplacement.remove("bas")
306             if deplacement!=["droite","gauche","haut","bas"] or
307                 random.randint(0,1)==0:
308                 direction=deplacement[random.randint(0,Len(deplacement)-1)]
309             return direction
310         except:
311             pass
312
313 def avancefantome(fantome,direction):
314     x,y=canvas1.coords(fantome)
315     if direction=="droite":
316         accelerationx=25
317         accelerationy=0
318     elif direction=="gauche":
319         accelerationx=-25
320         accelerationy=0
321     elif direction=="haut":
322         accelerationx=0
323         accelerationy=-25
```

```
322     elif direction=="bas":
323         accelerationx=0
324         accelerationy=25
325     if x<10: #
326         x=375 #
327         canvas1.coords(fantome,x,y)#
328     elif x>375: #
329         x=0 #
330         canvas1.coords(fantome,x,y)#
331     elif y<5: #zone de délimitation pour
332         y=575 #la téléportation des fantômes
333         canvas1.coords(fantome,x,y)#
334     elif y>575: #
335         y=0 #
336         canvas1.coords(fantome,x,y)#
337     if collision(x,y,direction)==1:
338         x, y = x +accelerationx, y +accelerationy
339         canvas1.coords(fantome, x, y)
340     else:
341         avancefantome(fantome,deplacementfantomes(fantome,direction))
342 def fantomes():
343     global directionblinky,directionpinky,directioninky,directionclyde
344     directionblinky=deplacementfantomes(blinky,directionblinky)
345     avancefantome(blinky,directionblinky)
346     directionpinky=deplacementfantomes(pinky,directionpinky)
347     avancefantome(pinky,directionpinky)
348     directioninky=deplacementfantomes(inky,directioninky)
349     avancefantome(inky,directioninky)
350     directionclyde=deplacementfantomes(clyde,directionclyde)
351     avancefantome(clyde,directionclyde)
352     fenetre.after(200,fantomes)
353     #####lancement des boucles du programme#####
354     fenetre.after(1,fantomes)
355     fenetre.after(2,anime_fermer)
356     #####Barre de statut#####
357     frame1=Frame(fenetre,bg="black")
358     frame1.pack()
359     def menu(): #retour au menu
360         global fenetre
361         fenetre.destroy()
362         os.system("python Menu.py")
363         exit()
364     Button(frame1,text="Menu",command=menu,fg="yellow",bg="black").grid(row=0,column=4)
365     score=Label(frame1,text="SCORE:0",fg="yellow",bg="black") #affichage des points
366     score.grid(row=0,column=2,padx=100)
367     niv=Label(frame1,text="NIVEAU:1",fg="yellow",bg="black") #affichage du niveau
368     niv.grid(row=0,column=0)
369
370     #####
371     fenetre.mainloop()
372
```

A.2 Script du menu

Menu.py

```
1  #!/usr/bin/python2.7
2  # coding: utf-8
3
4  from Tkinter import *
5  import os
6  import tkinter
7
8  mafenetre=Tk()
9  mafenetre.resizable(width=False, height=False)
10 mafenetre.title('menu')
11 mafenetre['bg']='black'
12 icone=PhotoImage(file="pacman.gif")
13 mafenetre.tk.call('wm', 'iconphoto', mafenetre._w, icone)
14
15 photo=PhotoImage(file="pac-man-logo.gif")
16 Label(mafenetre,image=photo).grid(row=1,column=1,rowspan=10,columnspan=10)
17 def jouer():
18     global mafenetre
19     mafenetre.destroy()
20     os.system("python pacmanlabyrinthesuite.py")
21     exit()
22 def regles():
23     Frame1=Frame(mafenetre,borderwidth=2,bg='blue')
24     Frame1.grid(row=30,column=4,padx=10,pady=10,columnspan=20)
25     Label(Frame1,text="Pour jouer : utilisez les flèches pour déplacer le pacman
26     dans le labyrinthe. Le but du jeu est de manger tous les petits points blancs
27     sans se faire manger par les
28     fantômes",fg="blue",bg="white",wraplength=225).pack(fill=X)
29
30 def info():
31     tkinter.messagebox.showinfo("INFO","réalisé par Dubard Loïc et Dinnat Laura dans le
32     cadre du projet pour le bac d'ISN")
33
34 Button(mafenetre,text="JOUER",fg='blue',command=jouer).grid(row=13,column=4,padx=10,
35     pady=10)
36
37 Button(mafenetre,text="REGLES",fg='blue',command=regles).grid(row=13,column=6,padx=20,
38     pady=20)
39
40 Button(mafenetre,text="INFO",fg="blue",command=info).grid(row=17,column=5)
41
42 Label(mafenetre,bg="black").grid(row=19,column=4)
43
44 mafenetre.mainloop()
45
```

A.3 Code du fichier des niveaux

niveau.lvl

```
1  MMMMMMMMMMMMMMMMMMMX0000M00M00000MM00M00M00M00M00M00M00M0000000MM0MM0000MM00MM0000MMMMM
MM0000MMMMMMMMM000000000000000MM00000M00M00000MM000M0MBCM0M000MMM0MM0MMM0MM0MMM000M000
000M000MM000MM000MM000MM000M000000M000MM000MM00MM000MM0000000000MMM000M0MMM0M
000MM0MMM0MPIM0MMM0MM000M0M00M000MM000000000000MM00MMMMMMMMMM00MM000000000000M
M0MMM0M00M0MM0MM0000M00M0000MMMMMMMMMMMMMMMMMM

2  MMMMMMM00MMMMMMMMMX000000000000MM0MM000MM000MM0MM0M00M000M00M0MM0000M0MM0M0000MMMMM
MM0000MMMMMMMM00000000000000MM0MMM0M00M0MM0MM0M000MBCM000M0MM0M0M0MMM0M0M0MM000M000
000M000M000M0MMM0M00000000M00MM00M0000MM0000000000MM00000MM0000MM0MMM0MIPM0M
MM0MM000M0M00M0M000MMM0M000000M0MMM000M00MM00M000MM0MM00000MM0MM000MM00MMM000M
MMM0M000000M0MMM00000000000000MMMMMMMMM00MMMMMM
```