



PACMAN

Rapport de projet Info 3-1 P.C.S

I. Explication du programme:

Après avoir téléchargé le dossier **TAYLOR_SOCHAJ.zip** il faut l'extraire avant de le lancer. Vous devriez avoir **8 fichiers.pde** et **1 dossier data** contenant toutes les images, font, sons, level1.txt...

Une fois le programme lancé vous arriverez sur l'avant menu du jeu. Appuyez sur **ENTER** pour continuer. Pour naviguer il suffit de cliquer sur l'option voulue (dans le rectangle qui l'entoure) : **//New Game / Load Game / Save Game / Highscores / Exit**

- **New Game**
Permet de réinitialiser le plateau de jeu (correspondant au level1.txt) et le score du joueur. La partie (re)commence.
- **Load Game**
Permet de charger une partie sauvegardé auparavant en entrant le numéro de votre sauvegarde.
- **Save Game**
Permet de sauvegarder l'état actuel du plateau dans un fichier.txt qu'on pourra charger avec le bouton Load Game par la suite.
- **Highscores**
Cette page vous affiche les 5 meilleurs scores du pacman avec le nom correspondant (pas fini)
- **Exit**
En cliquant sur Exit vous arriverez à une page qui vous demande de confirmer votre choix. Si oui, le programme se ferme. Si non, vous revenez au menu principal.

Le **MENU** fonctionne grâce à des "stages" chaque moment du jeu correspond à un stage différent (ex: stage 5=jeu). Pour changer de stage il existe des fonctions **menuMove()** pour vérifier les positions mouseX et mouseY, si on click sur ce bouton alors on va à ce stage. Toutes les étapes de menu ont aussi une animation pour les rendre plus vivant.

En lançant une nouvelle partie vous commencez directement sur le plateau de jeu, utilisez les touches **WASD** pour vous déplacer (le code a principalement été écrit sur un clavier QWERTY d'où ce choix) vous pouvez aussi utiliser les **flèches directionnelles**.

En bas de la fenêtre d'affichage il y a certaines informations comme le nombre de **vies restant** (affichés avec des PacMan), votre **score** (+10 pour DOT, +50 pour SUPERDOT), et le **temps écoulé** en (minutes:secondes)

A tout moment vous pouvez mettre la partie en **Pause** avec la touche **ESC** (échap) de votre clavier. Vous serez redirigé vers un écran de pause qui vous permettra de reprendre la partie (**Resume**), de revenir au menu principal (**Main Menu**) ou de sauvegarder votre progrès en cours (**Save Game**).

En sauvegardant votre partie votre numéro de fichier sera écrit dans la console de Processing, retenez le! Vous pouvez maintenant cliquez sur **Load Game** un numéro de sauvegarde vous sera demandé, entrez le! (le temps, le score ne sera pas sauvegardé). La position du PacMan sera sauvegardé seulement si vous ne fermez pas le programme entre temps.

Si vous **mangez** un **SUPERDOT** les fantômes deviennent vulnérables pendant 10 secondes, vous pouvez donc les manger le score augmente de plus en plus si vous mangez des fantômes pendant ce laps de temps.

Si vous mangez suffisamment de **DOTS** un **BONUS** apparaît au spawn initial du PacMan si vous le récupérez vous remplissez le plateau de jeu avec des DOTS, vous pouvez donc battre votre score!

Si vous avez moins de **3 vies** et vous obtenez plus de **10000 POINTS** alors vous gagnerez **1 vie** supplémentaire pour vous aider.

Les **FANTÔMES** marchent un peu comme le héro. C'est à dire qu'ils ont quasiment les mêmes fonctions mais on assigne à la fonction **launchmove()** un vecteur quasiment aléatoire. En d'autres termes on va lui assigner à chaque fois que le fantôme **croisent un mur ou au bout d'un certain temps** va **changer de direction de manière aléatoire** (bien sûr il va faire faire un aléatoire des directions possibles). Pour éviter de recopier le code pour chaque fantôme nous avons utilisé **l'héritage**.

Nous vous expliquerons plus en détail notre programme à l'oral.

II. Problemes rencontrés !

- **Déplacement**

Notre problème principal du projet était le déplacement du héro. Pour le déplacer case par case dans le plateau de jeu on avait pas de problème particulier rendre ce **déplacement fluide** était le challenge. De Plus il y avait un second problème, le hero devait rester au centre de la case pour éviter un décalage qui rendrait le jeu pas très agréable visuellement.

- **Mémoire**

Un autre problème rencontré était l'utilisation **mémoire** de Processing qui montait sans cesse jusqu'à atteindre la limite. Cette utilisation importante était dû aux images qui n'étaient pas initialisés dans le void setup() et donc étaient chargés toutes les 60 tièmes de secondes (60FPS).

- **Son**

Nous avons voulu mettre du son pour rendre le jeu plus agréable hors nous avons eu un probleme "unable to decode sound file" nous avons fait beaucoup de recherche sur ce sujet mais nous n'avons rien trouvé pour résoudre ce problème, nous avons donc pas réussi à mettre cette fonctionnalité.

III. Solutions mises en place

- Pour avoir un déplacement fluide et sans décalage nous ajoutons au fil du temps dans le boucle void draw() le déplacement * la direction + une vitesse, et pour éviter le décalage on a décidé de faire une mini "téléportation" du pacman au milieu de la case quand il va changer de direction. Tout ceci permet d'avoir un déplacement fluide et sans décalage.
- Pour résoudre le problème de la mémoire nous avons juste changer l'endroit ou l'on initialise les variables image, car elles étaient dans une fonction qui tournait en boucle une fois corrigé la mémoire RAM utilisée est moindre.