

Projet LINGI1131 : PacmOz

Florence Blondiaux 06521500, Noémie Verstraete 25021500

Avril 2018

1 Introduction

Si on ne se foire pas la suite du rapport arrive

2 Modification du GUI.oz

3 Comportement du Main.oz

Notre *Main* se divise en deux parties. Dans la première, nous initialisons la *window* et les ports. Ensuite nous lisons entièrement la *map* et initialisons toutes les variables utiles pour lancer le jeu soit en mode *turn-by-turn*, soit en mode *simultané*. Plus particulièrement, nous stockons :

- Une liste contenant tous les ports des *pacmans*. Chaque élément de la liste est un port et les ports sont triés dans la liste par ordre chronologique des ID des *pacmans*.
- Une liste contenant tous les ports des *ghosts*. Chaque élément de la liste est un port et les ports sont triés dans la liste par ordre chronologique des ID des *ghosts*.
- Une liste contenant tous les IDs des pacmans. Chaque élément de la liste est un ID de type `ipacmani`.
- Une liste contenant tous les IDs des ghosts. Chaque élément de la liste est un ID de type `ighosti`.
- Une liste contenant tous les IDs des pacmans et des ghosts. Chaque élément est un ID de type `ipacmani` ou de type `ighosti`. L'ordre des IDs dans la liste a été choisi aléatoirement (grâce à *OS.rand*) et c'est cet ordre qui sera utilisé pour déterminer l'ordre des joueurs pour le mode *turn-by-turn*. Cette liste ne sera pas utilisée pour le mode *simultané*.
- Une liste contenant tous les emplacements des points de la *map*. Chaque élément est de type `ipositioni`.

- Une liste contenant tous les emplacements de bonus de la *map*. Chaque élément est de type `jpositioni`.
- Une liste contenant tous les emplacements "murs" de la *map*. Chaque élément est de type `jpositioni`.
- Une liste de records pour stocker les emplacements "spawn" assignés aléatoirement à chaque pacman. Chaque élément de la liste est de type `jpacmanijpositioni`.
- Une liste de records pour stocker les emplacements "spawn" assignés aléatoirement à chaque ghost. Chaque élément de la liste est de type `jghostijpositioni`.

C'est également dans la première partie de la *Main* que nous envoyons tous les messages nécessaires aux pacmans et aux ghosts. Par exemple, nous informons les pacmans de leurs emplacements "spawns" respectifs, de l'emplacement des points, bonus, ghosts, etc.

Dans la deuxième partie de la main, nous lançons le jeu soit en mode *turn-by-turn*, soit en mode *simultané* en fonction de ce qui est demandé dans le fichier

Dans la main, nous commençons par créer un port pour la window et ouvrir celle-ci. Nous créons ensuite des ports pour les pacmans et lisons entièrement la map pour stocker dans des listes les emplacements de tous les points, de tous les murs et de tous les bonus.

3.1 Mode Turn-by-Turn

3.2 Mode simultané

4 Comportement du pacman

5 Comportement du ghost

6 Interopérabilité

Nous avons échangés nos pacmans et ghost avec 3 autres groupes, les groupes 18, 47 et 60. Certaines petites erreurs ont pu être découvertes grâce à ça. Nous avons pu découvrir que le groupe 47 envoyait un message ne respectant pas la syntaxe. Le groupe 18 a pu se rendre compte qu'ils envoyait une valeur qui était déjà liée alors qu'elle devait être liée suite à l'action du message, mais également que lorsque les pacmans apparaissaient pour la première fois sur la map, les ghosts n'étaient pas prévenus. Ce qui posait problème lorsque la

6.1 Groupe 47

Tout fonctionnait pour leurs pacmans et ghost. Un des messages envoyé par l'autre groupe au pacman ne respectait pas la spécification, ce problème fut résolu rapidement.

6.2 Groupe 18

Lorsque le pacman spawn pour la première fois, le groupe 18 ne prévenait pas les ghosts de sa position. Cela posait donc un problème si le pacman mourrait avant de jouer pour la première fois. Le même problème fut rencontré avec l'autre groupe.

6.3 Groupe 60

-! Rien d'intelligent avec ma maps