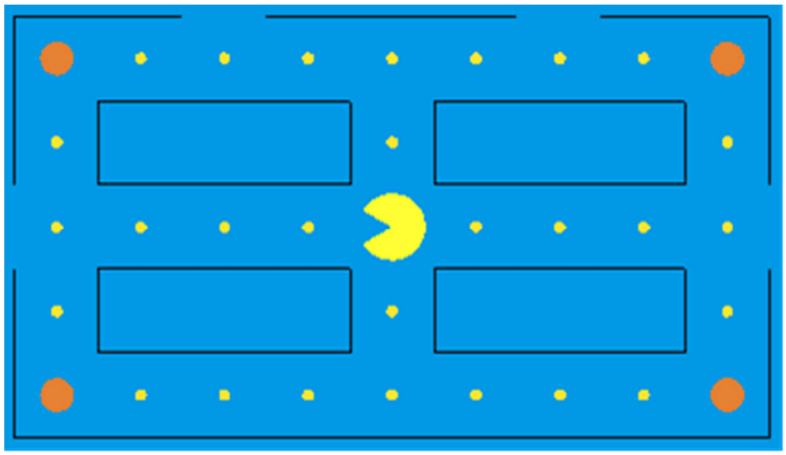
Projet C++ - PPGR : Pacman – Partie 1

Remarques préliminaires :

- Ce projet doit être effectué par groupe de 2 étudiants, sauf autorisation expresse des professeurs. Le ou les fichiers contenant le code source doivent être postés sur l'extranet, par l'un des 2 étudiants de chaque groupe, au plus tard le dimanche 16 octobre 2011 à 23h59.
- En plus des aspects programmation graphique et esthétique (le jeu doit rester agréable à regarder et à manipuler), les critères suivants seront pris en compte dans l'évaluation : le code doit être commenté de telle manière qu'une personne étrangère au projet puisse s'y retrouver aisément ; il doit être modulaire (chaque fonctionnalité doit être programmée le plus indépendamment possible des autres) ; il doit être le plus efficace possible du point de vue algorithmique (pas d'alternatives inutiles, pas de redondances, etc) et enfin il doit être le plus général possible (paramétrer tout ce qui peut l'être, soit à l'aide de constantes globales, soit via des paramètres d'entrées de fonctions, choisir des structures de données qui soient adaptatives, etc).
- Chaque groupe remet un travail original : si vous collaborez entre groupes, vous veillerez cependant à garder un caractère original/spécifique à votre programme.
- **Vous devez programmer les volets 1 et 2 dans leur entièreté**. Vous irez ensuite le plus loin possible dans le volet 3.

Pacman :



Le Pacman se déplace dans un labyrinthe jusqu'à ce qu'il ait mangé toutes les pastilles qui s'y trouvent, ce qui lui permet de passer au niveau supérieur. Un ou plusieurs fantômes peuvent aussi parcourir le labyrinthe : le Pacman doit alors les éviter en mode normal et les avaler en mode "super". Ce dernier mode s'active pour une durée limitée lorsque le Pacman avale une des pastilles spéciales. Chaque objet avalé rapporte un nombre de points spécifique. Ci-dessus, un exemple très élémentaire …

Objectif de l'application :

- Dessiner le plateau de jeu dans sa position initiale, similaire à la position proposée dans la figure ci-dessous, dans une fenêtre d'écran ;
- Animer la bouche du pacman et gérer ses déplacements tout en actualisant l'affichage pour tenir compte des endroits déjà parcourus ;
- Introduire un fantôme se déplaçant de manière aléatoire et gérer la logique du jeu (vies, score, fin du niveau).

Au terme de cette première partie, l'application prévoira :

- Une sous-fenêtre contenant le labyrinthe, une autre des boutons graphiques de gestion du jeu, et une troisième permettant d'afficher les informations de l'état du jeu et des messages à l'utilisateur sous forme de texte.
- La gestion des mouvements du pacman et de la logique du jeu avec prise en compte des actions de l'utilisateur et mise à jour des informations sur l'état du jeu. Les actions de l'utilisateur devront être possibles d'une part via le clavier, et d'autre part via la souris en cliquant sur les boutons graphiques.

Votre soumission devra aussi **impérativement** inclure un manuel de l'utilisateur expliquant clairement les fonctionnalités des touches du clavier et des boutons graphiques disponibles.

Du point de vue graphique

Volet 1.

1) Choisissez des structures de données pour stocker l'information nécessaire au dessin (positions du Pacman, des parois et des pastilles). Définissez ensuite des formes (élémentaires) pour représenter les différents éléments du jeu (Pacman, pastille, morceau de paroi). **Attention** : il ne faut **pas** réfléchir en pixels ! Choisissez des unités pratiques pour travailler dans votre univers de dessin.
2) Décrivez le jeu dans la situation initiale à partir de la fonction "display" en parcourant les structures de données choisies et en veillant à utiliser des instructions de translation et rotation pour positionner les différentes formes nécessaires.
3) En parallèle avec les points 1) et 2), choisissez des dimensions en pixels pour la fenêtre d'écran de telle sorte que son rapport largeur sur hauteur soit identique à celui du labyrinthe. Dans la fonction "viewport", définissez à l'aide d'une instruction de projection la zone de dessin qui doit être projetée dans la fenêtre d'écran.
4) Programmez le déplacement du pacman : choisissez 4 touches du clavier permettant à l'utilisateur de faire faire au Pacman un saut d'une "case" dans les 4 directions, lorsque les parois le permettent. Il est interdit de sortir du « labyrinthe », même par les entrées prévues pour les fantômes. N'oubliez pas de faire disparaître les pastilles ingurgitées. Faites changez le pacman d'aspect lorsqu'il ingurgite l'une des pastilles spéciales.
5) Ajoutez une viewport permettant d'afficher des messages à l'utilisateur, tels que : le niveau, le score, le nombre de vies restant, la vulnérabilité aux fantômes, la fin du jeu.
6) Programmez le mouvement de va et vient de la bouche du pacman.

Volet 2.

7) Eliminez l'effet de distorsion lors d'un redimensionnement de la fenêtre d'écran par l'utilisateur. Dans un premier temps, arrangez-vous pour éliminer la distorsion de la viewport contenant le labyrinthe. Traitez ensuite aussi la seconde viewport.
8) Ajoutez une 3ième viewport qui contiendra 4 boutons graphiques. Ces boutons permettront à l'utilisateur de gérer les déplacements à l'aide de clics de la souris. Gérez la distorsion aussi pour cette 3ième viewport.
9) Prévoyez la possibilité pour l'utilisateur de recommencer le niveau en cours et de quitter l'application. Ces deux actions doivent pouvoir se faire via le clavier d'une part, et via deux boutons graphiques supplémentaires d'autre part.

Volet 3.

10) Améliorez le graphisme des différents éléments du jeu.

11) Améliorez le déplacement du pacman : il doit donner l'impression d'avoir lieu en continu et non plus par sauts et doit continuer tout droit soit jusqu'à rencontrer une paroi, soit jusqu'à ce que l'utilisateur change de direction.

12) Imaginez un dessin simple de fantôme. Faites apparaître ce fantôme par l'une des entrées du « labyrinthe » et faites-lui faire un trajet soit fixé en boucle, soit aléatoire, à votre choix. Changez l'aspect du fantôme lorsque le pacman peut l'ingurgiter.

Du point de vue C++

La partie C++ du projet sera évaluée en tenant compte des critères suivants:

- Respecter la programmation modulaire (séparation .h .cpp)
- Gérer la mémoire sans fuite (allocation/libération des ressources)
- Respecter l'encapsulation (usage public/private/protected)
- Respecter le modèle MVC
- Démontrer la maîtrise des différents types d'accès mémoire (pointeurs, références et valeurs) en les utilisant judicieusement dans votre programme notamment lors des passages de paramètres.
- Montrer la maîtrise des aspects OO de C++ (ne pas écrire de code « à la C »).
- Produire un code maintenable par une personne tierce : architecture claire, code auto-commenté, spécification claire du contrat de chaque classe, commentaires synthétiques pour les parties plus compliquées du code.

Volet 1 : MVC

1) Votre application sera architecturée selon le modèle MVC vu aux cours de GUI et de pattern. Déclarez les différentes classes et méthodes correspondant ce modèle.
2) Le modèle contiendra les structures de données permettant de connaître la configuration et l'état du jeu (position des pastilles, parois, pac-man, fantômes, mode (normal ou « super »), score, etc.
   Cette structure de données suivra un modèle « OO » : nous voulons voir apparaître une classe (ou plusieurs classes) supplémentaires représentant le jeu. Cette classe ne sera pas un singleton et pourrait donc être instanciée en plusieurs exemplaires.
3) La mémoire occupée par les objets représentants le jeu devra être gérée correctement : elle devra être allouée dans le constructeur et libérée dans le destructeur.
4) Les structures de données ne seront pas exposées au monde extérieur. Il faudra donc créer les getters appropriés ainsi que les méthodes modifiant l'état du jeu.
5) La vue utilisera les getters pour gérer l'affichage.
6) Le modèle doit pouvoir être contrôlé par le contrôleur en lui offrant les services qui lui sont nécessaires. Il doit en outre détecter si une action demandée est impossible et le signaler au contrôleur.
7) La gestion des ordres clavier et souris et de la relance de la partie se fera au niveau du contrôleur.

Volet 2: Tableau de jeu « custom »

8) .Il doit être possible de charger, à partir d'un fichier, différentes configurations pour le tableau de jeu. On devra pouvoir configurer dans ce fichier la taille du tableau, l'emplacement des parois et des pastilles. **Le chargement doit impérativement se faire à l'aide de l'opérateur << surchargé !**