

Mid-Term Test Practical Portion

EEE243 Applied Computer Programming
27 October 2016, 0800–1050 hrs

Examiner: S. Givigi, PhD

Instructions

- You may begin the practical portion of the exam immediately.
- The practical portion is out of 30 marks.
- The practical portion of this test is open book. You may refer to course notes and you may access the Internet.
- Communication with any person other than the invigilators during the practical portion of the test **is forbidden**.
- You will not receive an exam booklet for the practical portion of the test; you have received three pages of notepaper instead.
- Write your name and student number on the notepaper, and turn it in at the end of the exam if you wish it to be marked by the examiner.
- Ask the invigilator for more notepaper if you require it.

Hint: Compile and execute your program **often**. You will lose marks if your program does not compile and execute as submitted, or if it generates compiler warnings. Start small and build from there.

Test de mi-session Partie Pratique

GEF243 Programmation informatique appliquée
27 octobre 2016, 8h00–10h50

Examineur: Capt Adrien Lapointe, CD, MSc

Instructions

- Vous pouvez commencer la partie pratique tout de suite.
- La valeur de la partie pratique est de 30 points.
- La partie pratique de l'examen est à livre ouvert. Vous pouvez faire référence aux notes du cours et accéder à l'Internet.
- La communication avec toutes personnes autre que les surveillants pendant la partie pratique de l'examen **est interdite**.
- Vous ne recevrez pas de livret d'examen pour la partie pratique du test; vous avez plutôt reçu trois feuilles de notes.
- Écrivez votre nom et numéro de collège sur le papier de notes et remettez-le à la fin de l'examen si vous voulez qu'elles soient corrigées par l'examineur.
- Demandez plus de feuilles de notes si vous en avez besoins.

Indice : Compilez et exécutez votre code **souvent**. Vous serez pénalisé si votre programme ne compile pas, n'exécute pas ou génère des avertissements de compilateur. Commencez petit et ajoutez des fonctionnalités au fur et à mesure.

Introduction

Open the Pacman project that you created at the beginning of the examination.

Add your name and college number to the comment blocks at the beginnings of the `pacman_test.c` and `pacman.c` files.

The code provided is the skeleton for a basic implementation of the classic game Pacman. This initial implementation is not interactive and does not have a fancy display. ASCII characters instead represent the game board and each game turn is displayed after another.

Your job is to write the functions that run the game and the ones that display it.

You must provide evidence of your design for each of the tasks. Clear, well-structured code that works and includes appropriate comments is sufficient. However, if your code does not work or is not clear or well structured, you may receive part marks for design documentation in the form of flowcharts, pseudo-code, memory diagrams, or textual descriptions of your design. Write these on the provided notepaper.

Requirements

You have been provided with three files:
`pacman_test.c`, `pacman.c` and `pacman.h`.

In `pacman.c` you must implement the three functions `display_gameplay`, `play`, and `game_is_over`.

In `pacman_test.c` you must write code that simulates a series of moves by Pacman ('G' character), displaying the state of the game after each move. You may also implement any other functions that you judge necessary.

You **must not modify** `pacman.h`.

Game representation

The game board is represented by a two-dimensional array of characters, each of which represents a game element.

Walls are represented by the '#' character and cannot be crossed. The character ' ' represents empty spaces. Dots ('.' character) are placed on the game board and must be collected by Pacman to win the game. The '@' character

Introduction

Ouvrez le projet Pacman que vous avez créé au début de l'examen.

Ajoutez votre nom et numéro de collège au bloc commentaire au début des fichiers `pacman_test.c` et `pacman.c`.

Le code fourni est le squelette d'une version élémentaire du jeu classique Pacman. Cette implémentation initiale n'est pas interactive et n'a pas un affichage très élaboré. La partie est plutôt représentée par des caractères ASCII et les tours du jeu sont montrés l'un à la suite de l'autre.

Votre tâche est d'écrire les fonctions pour jouer et afficher la partie.

Vous devez fournir des traces de votre conception pour chaque tâche. Du code clair, bien structuré, qui fonctionne et qui inclut des commentaires appropriés est suffisant. Toutefois, si votre code ne fonctionne pas ou s'il n'est pas clair ou bien structuré, vous pouvez recevoir des points partiels pour la documentation de conception sous la forme d'organigrammes, pseudo-code, schémas de mémoire ou descriptions textuelles de votre conception. Écrivez-les sur les feuilles de notes fournies.

Exigences

Trois fichiers vous sont fournis : `pacman_test.c`, `pacman.c` et `pacman.h`.

Dans `pacman.c`, vous devez implémenter les trois fonctions `display_gameplay`, `play`, et `game_is_over`.

Dans `pacman_test.c`, vous devez écrire du code qui simule une série de déplacements de Pacman (la lettre 'G') et montrant l'état du jeu après chacun de ceux-ci. Vous pouvez aussi implémenter toute autre fonction que vous jugez nécessaire.

Vous **ne devez pas modifier** `pacman.h`.

Représentation de la partie

La planche de jeu est représentée par un tableau de caractères à deux dimensions où chacun représente un élément du jeu.

Les murs sont représentés par le caractère '#' et ne peuvent être traversés. Les espaces vides sont représentés par le caractère ' '. Des points (caractère '.') sont placés sur la planche de jeu et doivent être ramassés par

represents a ghost, which in the final version of the game would move, but will be static in your implementation.

main

While working through this exam, you should use the `main` function in `pacman_test.c` to test the functionality of the `display_gameplay`, `play`, and `game_is_over` functions described below.

When you are done, the `main` function must:

- Initialize the game board using the `initialize_game` function.
- Do the successive game turns that are represented by the array `moves`.
- Check whether the game is over either because Pacman collected all the dots or because it got killed by a ghost

play

The `play` function takes four parameters: the `game_board` array, Pacman's next move, its current location and the current turn number.

Pacman can move in four different directions: `NORTH`, `EAST`, `SOUTH` and `WEST`. Diagonal movements are not allowed. When Pacman gets to any game board edge, it has to wrap around to the opposite side. For instance, if it moves North and gets to the upper edge, it will re-emerge at the bottom. Pacman's world is donut shaped!

When Pacman tries to move on a wall ('#'), a message saying the move is not possible must be displayed and Pacman does not move.

When Pacman moves on a ghost, Pacman disappears and is killed. The ghost stays put.

When a dot is collected, it must disappear from the game board. When all the dots are collected, the game is over.

The game board must be displayed at the end of each turn.

display_gameplay

The `display_gameplay` function displays the current state of the game board as represented by the array of characters `game_board` that is given as a parameter. It also displays the string `label` given as a parameter.

Pacman afin de gagner la partie. Le caractère '@' représente un fantôme, qui bougerait dans la version finale, mais qui sera statique dans votre implémentation.

main

Pendant que vous travaillez sur le code, vous devriez utiliser la fonction `main` de `pacman_test.c` afin de tester les fonctionnalités des fonctions `display_gameplay`, `play` et `game_is_over` décrites ci-dessous.

Lorsque vous aurez terminé, la fonction `main` doit :

- Initialiser la planche de jeu en utilisant la fonction `initialize_game`.
- Afficher les tours successifs de la partie qui sont représentés par le tableau `moves`.
- Vérifier que la partie est terminée soit parce que Pacman a ramassé tous les points ou parce qu'il a été tué par un fantôme.

play

La fonction `play` prend quatre paramètres: le tableau `game_board`, le prochain déplacement de Pacman, sa position actuelle et le numéro du tour courant.

Pacman peut se déplacer dans quatre directions : `NORTH`, `EAST`, `SOUTH` et `WEST`. Les déplacements en diagonale ne sont pas permis. Lorsque Pacman arrive au bord de la planche de jeu, il doit réapparaître du côté opposé. Ainsi, s'il se déplace vers le nord et qu'il arrive à la limite supérieure de la planche, il doit réapparaître au bas. Le monde de Pacman a la forme d'un beigne!

Lorsque Pacman essaie de se déplacer sur un mur ('#'), un message indiquant que le déplacement est impossible est affiché et Pacman reste à la même position.

Lorsque Pacman se déplace sur un fantôme, Pacman disparaît et est tué. Le fantôme reste en place.

Lorsqu'un point est ramassé, il doit disparaître de la planche de jeu. Quand tous les points sont ramassés, la partie est terminée.

La planche de jeu doit être affichée à la fin de chaque tour.

display_gameplay

La fonction `display_gameplay` affiche l'état courant de la planche de jeu tel que représenté par le tableau de caractères `game_board` qui est passé en paramètre. Elle affiche aussi la chaîne de caractères `label` aussi passée en paramètre.

In order to see the edge of the game board, this function will add delimiter characters that are not part of the `game_board` array.

You will find below an example of the initial display and a first turn where Pacman went North. Your display should look like this.

```
Ready to play:
+-----+
|  @   |
|  . . . |
|  #   |
|  #   |
|  G #  |
+-----+

Turn 0:
+-----+
|  @   |
|  . . . |
|  #   |
|  G #  |
|  #   |
+-----+
```

game_is_over

The `game_is_over` function takes the `game_board` as a parameter and returns true if Pacman collected all the dots or if a ghost killed it. If none of those conditions are met, it returns false.

The function is also responsible to print if the game is over and the reason of its termination.

When you are finished

Ensure your name is in the header comment of the `pacman_test.c` and `pacman.c` files and on any notes pages. Leave the computer logged in and your project open in Eclipse, and leave your notes pages on the keyboard. Leave quietly by the closest door to minimize disturbance to others.

End of the examination

Afin de bien voir les limites de la planche, la fonction ajoute des caractères délimiteurs qui ne font pas partis du tableau `game_board`.

Vous trouverez ci-dessous un exemple de l’affichage initial et du premier tour après un déplacement de Pacman vers le nord. Votre affichage devrait être semblable.

game_is_over

La fonction `game_is_over` prend le `game_board` en paramètre et retourne vrai si Pacman a ramassé tous les points ou s’il a été tué par un fantôme. Si aucune de ces conditions n’a été rencontrée, elle retourne faux.

La fonction est aussi responsable d’afficher si la partie est terminée et la raison de la fin.

Lorsque vous avez terminé

Assurez-vous que votre nom est dans le bloc commentaire des fichiers `pacman_test.c` et `pacman.c`, ainsi que sur vos pages de notes. Laissez l’ordinateur connecté, votre projet ouvert dans Eclipse et laissez vos feuilles de notes sur le clavier. Sortez discrètement par la porte la plus proche pour ne pas déranger les autres.

Fin de l’examen