



## Introducción a la Ingeniería de Software. Prácticas curso 2014-2015.

### Introducción

Se presenta el enunciado de las prácticas de la asignatura Introducción a la Ingeniería del Software del Grado de Ingeniería Informática y Grado de Tecnologías de la Información. También se presentará las condiciones de entrega de los trabajos enunciados a los tutores, las condiciones de corrección y posterior entrega de calificaciones.

Como se podrá comprobar por el enunciado, el trabajo propuesto es una “revisita” sobre un trabajo previamente realizado por los alumnos que hayan cursado la asignatura de Programación Orientada a Objetos. Esta asignatura es una asignatura obligatoria del primer curso de los dos grados.

En el supuesto de que algún alumno no haya cursado esta asignatura, se encontrará con una propuesta nueva que podrá trabajar por completo desde el principio. El equipo docente entiende que esto no supone ninguna limitación, simplemente varía el enfoque de la elaboración de la práctica.

El contenido de estas prácticas integra cuatro partes progresivas que el alumno irá realizando y entregado a su tutor para su corrección.

### Objetivos

El objetivo de esta práctica es la consolidación de los conocimientos adquiridos con el estudio de la asignatura mediante un ejercicio que abarca los contenidos del curso. Es importante haber estudiado con anterioridad los diferentes temas que integran el curso y haber realizado alguno de los ejercicios propuestos. La práctica está estructurada de tal forma que se puede ir realizando a medida que se van estudiando los temas y permite al alumno comprobar hasta qué punto ha asimilado los conocimientos.

Un objetivo secundario de esta práctica es comparar como se aborda la realización de un

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

---

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70**

Cartagena99



## Enunciado

Se desea realizar la formalización del proceso de desarrollo de una aplicación que permita jugar al legendario juego tipo arcada Pacman en un computador.

La descripción pormenorizada del juego y de la aplicación aparece detallada, según el enunciado propuesto en la práctica de la asignatura Programación Orientada a Objetos, en el apéndice A.

## Particularmente se pide

Práctica 1: Elaborar una propuesta justificada de ciclo de vida que se va a elegir para el desarrollo de la aplicación. Contemplar el hecho de que se está elaborando un proyecto formal sobre una software que previamente estaba hecho y funcionando y donde el potencial cliente nos pedirá modificaciones. En el caso de no haber realizado la práctica de POO, considere el proyecto comenzando de cero.

Práctica 2: Elaborar los documentos de requisitos y de diseño del proyecto. Ambos documentos deben seguir el formato establecido en el libro de texto recomendado para la asignatura, si bien debe utilizar los diagramas de UML: clases de uso, clases, secuencia, colaboración, estado y actividad. Valorar que diagrama debe figurar en cada documento o, si lo considera oportuno, en los dos.

Practica 3: Desarrollar una serie de pruebas “de caja negra” para comprobar el funcionamiento de la aplicación.

Opcional: Si bien no es obligatorio, se recomienda implementar en Java la aplicación, de acuerdo con las pautas establecidas en los documentos entregados.

## Condiciones de presentación

1°. Las soluciones se entregarán al tutor en los plazos y forma que éste indique.

2°. Para la representación de los diagramas UML que se piden se deberá utilizar alguna herramienta software. Se recomienda utilizar Argo, que es de libre distribución, y puede descargarse en:

<http://argouml.tigris.org/>

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

---

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70**

Cartagena99



## APENDICE A

La práctica del presente curso va a estar basada en el legendario arcade "Pac-Man". Esto nos servirá para estudiar y practicar los mecanismos de la Programación Orientada a Objetos y hacer uso de sus características gráficas.

### Historia del Juego

Según la descripción en Wikipedia, Pac-Man es un videojuego arcade creado por el diseñador de videojuegos Toru Iwatani de la empresa Namco (basado supuestamente en la forma de una pizza con un trozo faltante) a principios de los años 1980.

### Modo de Juego

El protagonista del videojuego Pac-Man es un círculo amarillo al que le falta un sector por lo que parece tener boca. Aparece en laberintos donde debe comer puntos pequeños (llamados Pac-dots en inglés), puntos mayores y otros premios con forma de frutas y otros objetos. El objetivo del personaje es comer todos los puntos de la pantalla, momento en el que se pasa al siguiente nivel o pantalla. Sin embargo, cuatro fantasmas o monstruos, Shadow (Blinky), Speedy (Pinky), Bashful (Inky) y Pokey (Clyde), recorren el laberinto para intentar comerse a Pac-Man. Estos fantasmas son, respectivamente, de colores rojo, rosa, cian y naranja. Los fantasmas no son iguales, así mientras Blinky es muy rápido, y tiene la habilidad de encontrar a Pac-Man en el escenario, Inky es muy lento y muchas veces evitará el encuentro con Pac-Man.

Hay cuatro puntos más grandes de lo normal situados cerca de las esquinas del laberinto nombrados en inglés Power Pellets, y que proporcionan a Pac-Man la habilidad temporal de comerse a los monstruos (todos ellos se vuelven azules mientras Pac-Man tiene esa habilidad). Después de haber sido tragados, los fantasmas se regeneran en "casa" (una caja situada en el centro del laberinto). El tiempo en que los monstruos permanecen vulnerables varía según la pantalla, pero tiende a decrecer a medida que progresa el juego, y al cabo de muchas pantallas los puntos especiales no tienen ningún efecto sobre los fantasmas.

Además de comer los puntos, Pac-Man puede obtener puntuación adicional si se come alguno de los objetos que aparecen dos veces por pantalla justo debajo de la caja en el centro del laberinto de donde salen los monstruos. El objeto cambia cada pantalla o dos, y su valor en puntos aumenta, de forma que dos cerezas (el premio de la primera pantalla) valen 100 puntos, mientras que el último objeto, la llave, vale 5.000.

### Fantasmas

Los fantasmas están limitados por el laberinto de la misma manera que Pac-Man, pero por lo general se mueven ligeramente más rápido que el jugador, aunque se vuelven más lentos cuando se activan los puntos especiales de las esquinas.

Blinky, el fantasma rojo, aumenta su velocidad después de que un cierto número de puntos

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

---

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70**

Nivel 2: Fresa 500 puntos.

Niveles 3 y 4: Naranja 500 puntos.



Niveles 5 y 6: Manzana 700 puntos.

Niveles 7 y 8: Uvas 1000 puntos.

Niveles 9 y 10: Galaxian 2000 puntos.

Niveles 11 y 12: Campana 3000 puntos.

Niveles 13 al 255: Llave 5000 puntos.

En cada nivel aparecen dos veces los premios.

Si Pac-Man pierde una vida cuando aparece un premio, este desaparece a la vida siguiente.

Puedes ver el juego en acción en este vídeo en You Tube:

<http://www.youtube.com/watch?v=XlPq7dgQrQ0>

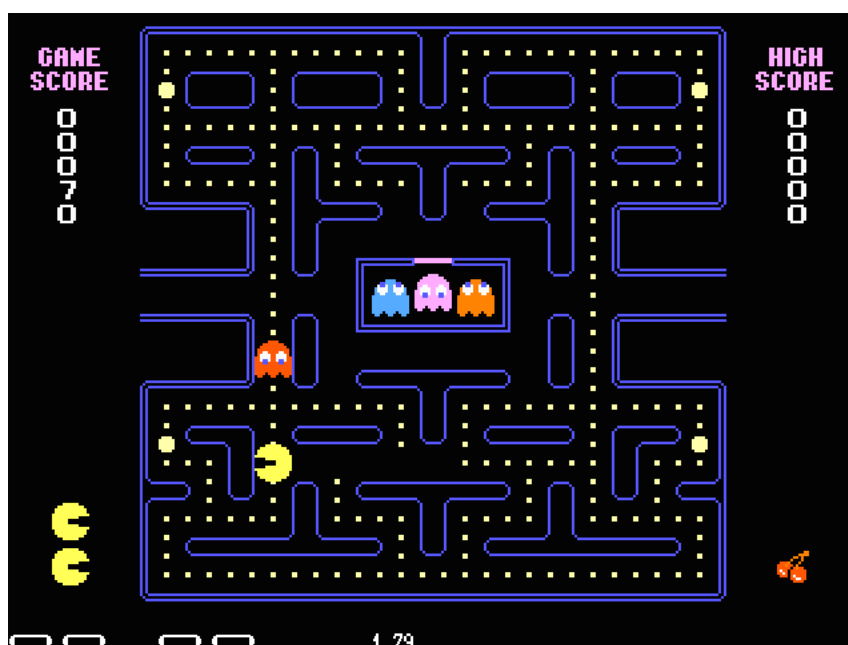


Figura 1. Imagen del juego original

### Implementación Obligatoria

El alumno deberá implementar un juego del estilo Pac-Man cumpliendo los siguientes requisitos:

- 1- El juego constará de un solo nivel donde el jugador deberá comer todos los puntos de la pantalla.
- 2- El jugador controlará a Pac-Man y dispondrá de 1 vida.
- 3- Los fantasmas serán controlados por el ordenador teniendo en cuenta el comportamiento

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

---

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70**

habrá que tener en cuenta las colisiones.



8- Los fantasmas deben implementar comportamientos diferentes:

- a. Blinky, el fantasma rojo, buscará colisionar con Pac-Man. Para acercarse a Pac-Man calculará la distancia (por ejemplo medido en filas y columnas) e intentará primero acercarse verticalmente y luego horizontalmente.
- b. Pinky. Buscará colisionar con Pac-Man. Para acercarse a Pac-Man calculará la distancia (por ejemplo medido en filas y columnas) e intentará primero acercarse horizontalmente y luego verticalmente.
- c. Clyde. Él no persigue a Pac-Man, si no que deambula sin una ruta específica.

9- Se deberán de detectar dos tipos de colisiones.

- a. Las colisiones entre Pac-Man y los fantasmas, lo que supondrá la pérdida de una vida o el final del juego en caso de no disponer de más vidas.
- b. Las colisiones entre los fantasmas, que supondrá un cambio de dirección en los fantasmas involucrados.

10- Habrá cuatro puntos más grandes de lo normal situados cerca de las esquinas del laberinto y proporcionarán a Pac-Man la habilidad temporal (5 segundos) de comerse a los fantasmas (todos ellos se vuelven azules mientras Pac-Man tiene esa habilidad). Después de haber sido tragados, los fantasmas se regeneran en "casa de fantasmas".

11- Será necesario implementar un contador con los puntos obtenidos en cada momento, teniendo en cuenta los objetos comidos. Un punto pequeño supone 10 puntos. Comer un fantasma 100 puntos.

12- Si el jugador finaliza el nivel del juego deberá aparecer un mensaje de felicitación y se volvería a mostrar la página inicial.

### Recomendaciones

Dado que el juego a desarrollar dispone de un mapa por el cual deben circular los elementos móviles del juego (Pac-Man y fantasmas) es necesario planificar su implementación para evitar problemas. A pesar de que las decisiones de implementación son libres para cada estudiante, desde el equipo docente recomendamos una alternativa que podría facilitar su desarrollo. El mapa podría ser considerado como una matriz  $A \times B$ , en la que cada celda sería controlada por un código (por ejemplo):

"-1" Para la pared.

"0" Para una celda vacía.

"1" Para un punto pequeño.

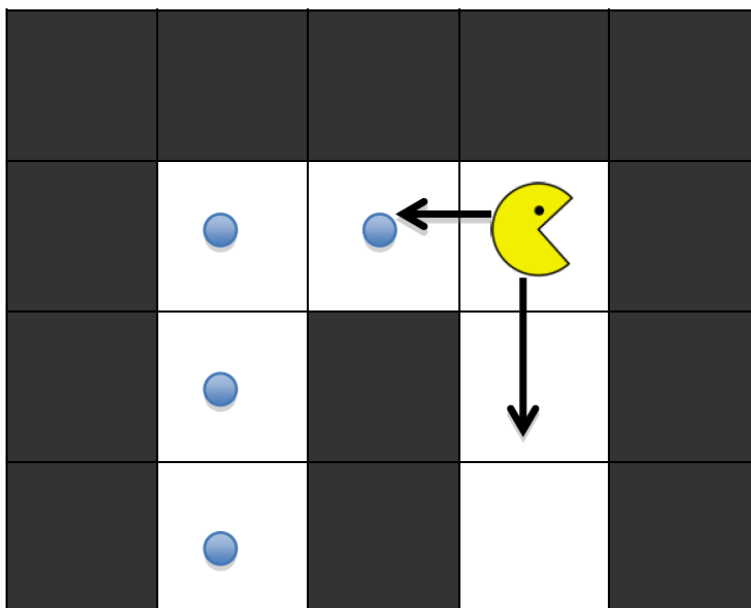
"2" Para un punto grande.

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

---

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70**

Cartagena99



Esta alternativa hará que los gráficos se muevan por la pantalla de manera más brusca. Otra aproximación para la resolución de la práctica orientada a que los gráficos se muevan más suavemente se puede llevar a cabo utilizando la detección gráfica de colisiones y el movimiento pixel a pixel explicada en los videotutoriales del curso.

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

---

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70**