

Universidad Mariano Gálvez de Guatemala

Facultad de Ingeniería en Sistemas de Información y Ciencias de la Computación

Ingeniera: Belén de la Cruz



Yoshua Kevin Yafeth de León Ovalle - 2290-21-19116

Jhony Abel Ajcalón Samines - 2290-21-2725

Daniel Orlando Muj Vásquez - 2290-21-7554

E'kefa Jorge Lisandro Iboy Saloj - 2290-21-17905

Grupo "3"

Sololá, Sololá 21 de octubre 202

Contenido

INTRODUCCIÓN.....	1
1. Historia PAC-MAN	2
Razones por la que fue un éxito.....	3
2. Diagramas UML	4
¿Para qué usarlos?.....	4
Diagramas UML PAC-MAN.....	5
1. Diagrama de Clases	5
2. Diagrama de Objetos	6
3. Diagrama de Secuencia.....	6
4. Diagrama de Colaboración	7
5. Diagrama de Estado.....	7
6. Diagrama de Polimorfismo.....	8
3. Fases de entrega de proyecto.....	9
1. Fase 1:	9
2. Fase 2:	9
3. Fase 3:	9
4. Fase 4	9
4. Explicación del código del videojuego	10
5. Manual de Usuario.....	11
6. Capturas del código.....	12

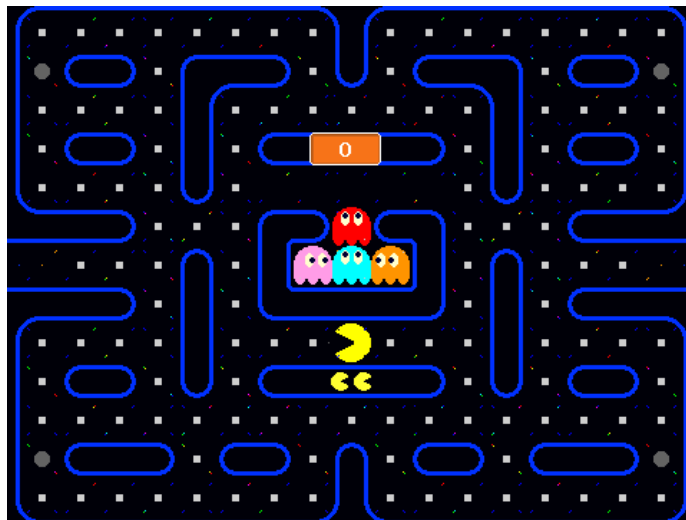
INTRODUCCIÓN

Como estudiantes de Ingeniería y trabajando en este caso el curso de Programación nos correspondió trabajar un proyecto haciendo uso del conocimiento impartido durante los meses de clases que se recibieron en base al lenguaje de programación Java NetBeans trabajando como proyecto final la creación de un juego que fue proporcionado durante clases, dicho juego es muy antiguo como famoso y el cual es Pac-Man, el cual fue un proyecto complicado de realizar teniendo que descartar algunas ideas que teníamos al principio además de estar trabajado en el transcurso de las clases para poder ser presentado en el presente trabajo.

1. Historia PAC-MAN

La leyenda cuenta que a fines de los años 70', Tohru Iwatani, diseñador de Namco, empresa japonesa de software en el campo de los videojuegos, salió una noche a comer pizza en compañía de algunos amigos. Luego de coger el primer trozo y mirar la forma que quedaba (un círculo al que le faltaba una parte, por lo que parecía tener una boca) tuvo un súbito momento de inspiración para crear un nuevo juego. Había nacido el famoso Pac man, el videojuego más famoso de todos los tiempos.

Luego de un año y medio de trabajo, Iwatani, junto a su grupo de colaboradores, terminaron de desarrollar el juego. El protagonista, Pac-Man, un círculo amarillo al que le faltaba un sector, por lo que parecía tener una boca, aparecía en un laberinto donde debía comer puntos pequeños, mayores y otros premios en forma de frutas y otros objetos. El propósito del Pac-Man era comerse todos los puntos de la pantalla para pasar al siguiente nivel o pantalla (el juego original tiene 255 niveles). Sin embargo, el Pac-Man debía sortear la presencia de cuatro peligrosos fantasmas de colores que no sólo se movían más rápido que el jugador sino que también intentaban atraparlo. Cuatro puntos más grandes de lo normal, ubicados en las esquinas del laberinto, le proporcionaban en todo caso a Pac Man la posibilidad de comerse a los fantasmas, que se tornaban azules mientras Pac-Man disfrutaba de esa habilidad. No obstante, los fantasmas, una vez devorados, podían regenerarse en una especie de corral ubicado al medio del laberinto, lo que volvía cada vez más frenético el juego.



Con respecto al peculiar nombre del juego, proviene de la onomatopeya japonesa “paku”, que es el sonido que se produce al abrir y cerrar la boca al comer. El nombre se modificó así a Puck-Man, pero fue modificado a Pac-Man para el mercado

norteamericano y occidental, debido a que la gente podía cambiar la palabra “puck” por “fuck”, procaz término del idioma inglés.

El juego Pac-Man fue lanzado oficialmente al mercado el 21 de mayo de 1980 y fue distribuido por la empresa Midway Games al mercado estadounidense en ese mismo año. Fue, por supuesto, un éxito absoluto, convirtiéndose en un verdadero fenómeno mundial en la industria de los videojuegos, acabando de paso con la supremacía del famoso Space Invaders, donde la acción predominante era disparar frenéticamente a varias columnas de marcianos que se acercaban cada vez más al jugador.

Razones por la que fue un éxito

Las razones del éxito fulminante del Pac-Man fueron varias. En primer lugar, podía ser jugado tanto por hombres como mujeres, y su formato era más humorístico y menos violento, lo que terminó de conquistar a todos los usuarios, algunos de los cuales descubrirían por medio de la reiterada práctica varios secretos para avanzar cada vez más en el juego (Para engañar mejor a los fantasmas había que girar a un lado y luego al contrario rápidamente; los ojos de los fantasmas indicaban la dirección que tomaban en su próximo giro; la parte más peligrosa de la pantalla era la línea inferior, ya que los fantasmas intentaban arrinconar al jugador, mientras que la parte más segura era cerca de la casa de los fantasmas, que ofrecía múltiples caminos de escape; cada fantasma, por lo demás, tenía su personalidad y una esquina en la que era más efectivo; y, finalmente, el Pac-man era más rápido en un camino despejado de puntos, algo bastante útil cuando el jugador intentaba despistar a los implacables fantasmas).

El juego Pac-Man, que inspiraría canciones, juegos y hasta una película, llegó a tener el récord Guinness del videojuego arcade más exitoso de todos los tiempos con un total de 293.822 máquinas vendidas de 1981 a 1987. El mejor jugador del mundo, por lo demás, fue Billy Mitchell, quien hace más de dos décadas consiguió una puntuación de 3.333.360 puntos llegando al nivel 255 con la primera vida. El año 2009 incluso se realizó un campeonato mundial patrocinado por Namco.

Hoy, a 35 años de su nacimiento, Pac-Man ya es parte de la cultura occidental y parte del inconsciente colectivo de millones de personas. La prestigiosa revista norteamericana “Time”, por lo pronto, hizo hace un tiempo un ranking (donde los usuarios votaban por el videojuego que más lo había marcado en su infancia) con los mejores videojuegos de todos los tiempos, con títulos que abarcaban juegos de PC, consola y teléfonos móviles, según su importancia e influencia en cada década ¿Adivine quién ganó? Exacto. Pac-man se impuso con comodidad en el primer lugar, seguido consecutivamente por “Super Mario Bros” (1985), “Pong” (1972) y “Space Invaders” (1978).

2. Diagramas UML

UML son las siglas de “Unified Modeling Language” o “Lenguaje Unificado de Modelado”. Se trata de un estándar que se ha adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativa a los desarrollos de software (programas informáticos).

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos.

La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Recordemos que un modelo es una representación simplificada de la realidad; el modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.



¿Para qué usarlos?

Los modelos o diagramas de UML nos ayudan a trabajar a un mayor nivel de abstracción. Permite modelar cualquier tipo de aplicación corriendo en cualquier combinación de hardware y software, sistema operativo, lenguaje de programación y red, es decir, UML es independiente de la plataforma hardware sobre la que actúa el software. Su flexibilidad permite modelar cualquier tipo de aplicación e, incluso, otros tipos de proyecto que no son puramente software.

UML ofrece ese modelado utilizando diagramas y se denomina lenguaje por ser una forma común de expresarse por todos los analistas, desarrolladores y usuarios. Está desarrollado para ayudar a todos estos (y más) perfiles a especificar, visualizar, construir y documentar todos los componentes de un proyecto. A pesar de que cada diagrama UML en particular aporta su visión particular al modelado, el lenguaje en su conjunto tiene algunas características que interesa resaltar:

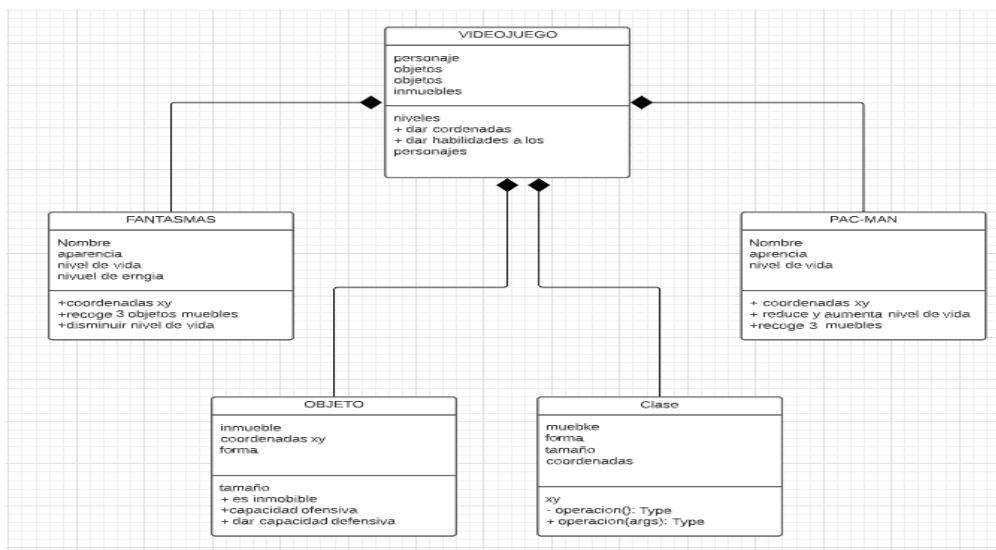
- Es muy sencillo. Pese a que si es usado de forma completa puede llegar a complicarse, lo normal es que se simplifique.
- Es capaz de modelar todo tipo de sistemas.
- Es un lenguaje universal, haciendo que todos los miembros del equipo se relacionen a través de sus diagramas sean del ámbito que sean.
- Es fácilmente extensible. Tiene mecanismos sencillos para especializar los conceptos fundamentales.
- Es visual y, por lo tanto, intuitivo.
- Es independiente del desarrollo, del lenguaje y de la plataforma.
- Bien ejecutado aporta un conjunto considerable de buenas prácticas.
- No está completo. Utilizando los distintos diagramas no podemos estar seguros de comprender con totalidad el sistema que va a desarrollarse. Los diagramas, para facilitar su comprensión pueden (y suelen) omitir información, pueden tener partes que se entienden de distintas maneras o, incluso, pueden tener conceptos que no pueden ser representados por ningún diagrama.

Diagramas UML PAC-MAN

1. Diagrama de Clases

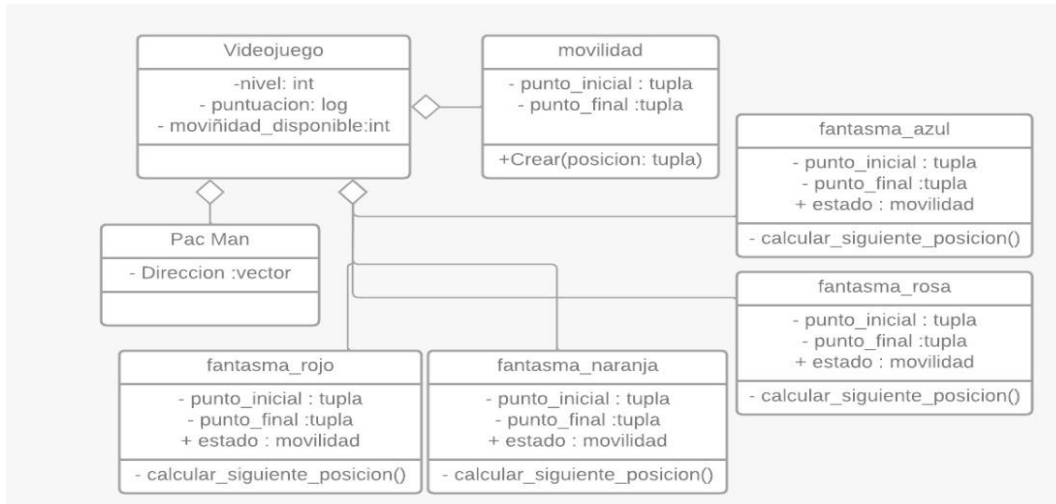
El propósito de un diagrama de clase es describir las clases que conforman el modelo de un determinado sistema. Dado el carácter de refinamiento iterativo que caracteriza un desarrollo orientado a objetos, el diagrama de clase va a ser creado y refinado durante las fases de análisis y diseño, estando presente como guía en la implementación del sistema

Los diagramas de clases describen la estructura estática de un sistema.



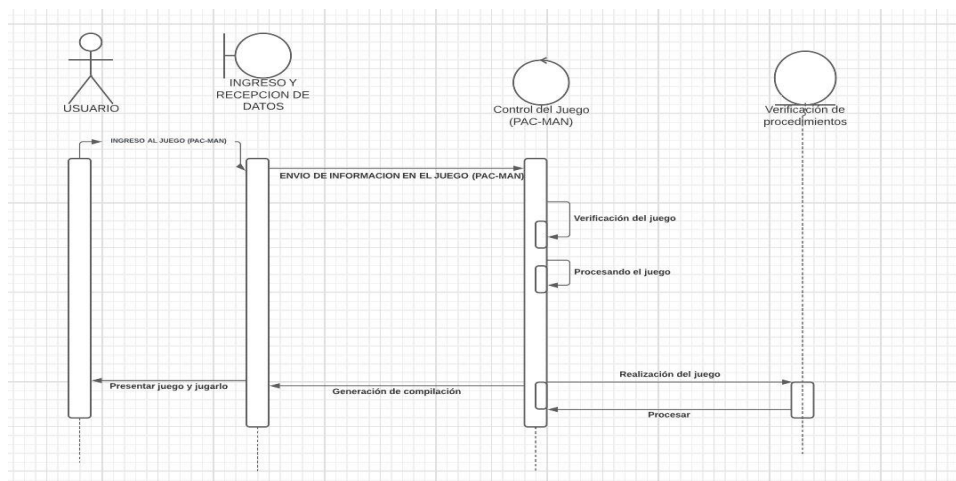
2. Diagrama de Objetos

Un diagrama de objetos UML representa una instancia específica de un diagrama de clases en un momento determinado en el tiempo. Cuando se lo representa visualmente, verás muchas similitudes con el diagrama de clases



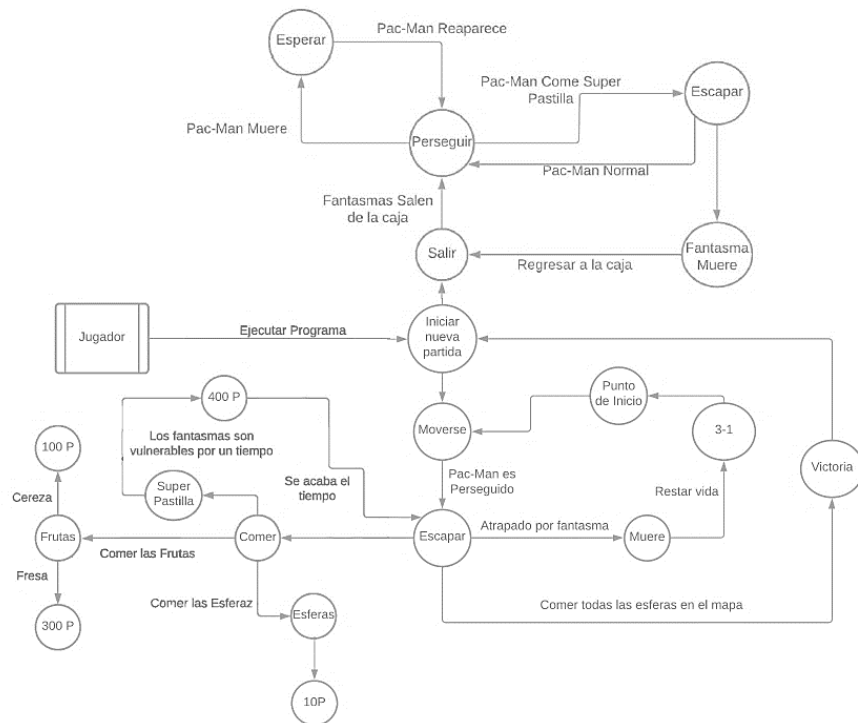
3. Diagrama de Secuencia

Para comprender lo que es un diagrama de secuencia, es importante conocer la función del Lenguaje Unificado de Modelado, mejor conocido como UML. El UML es un conjunto de herramientas de modelado que orienta la creación y notación de muchos tipos de diagramas, incluidos los diagramas de comportamiento, los diagramas de interacción y los diagramas de estructuras.



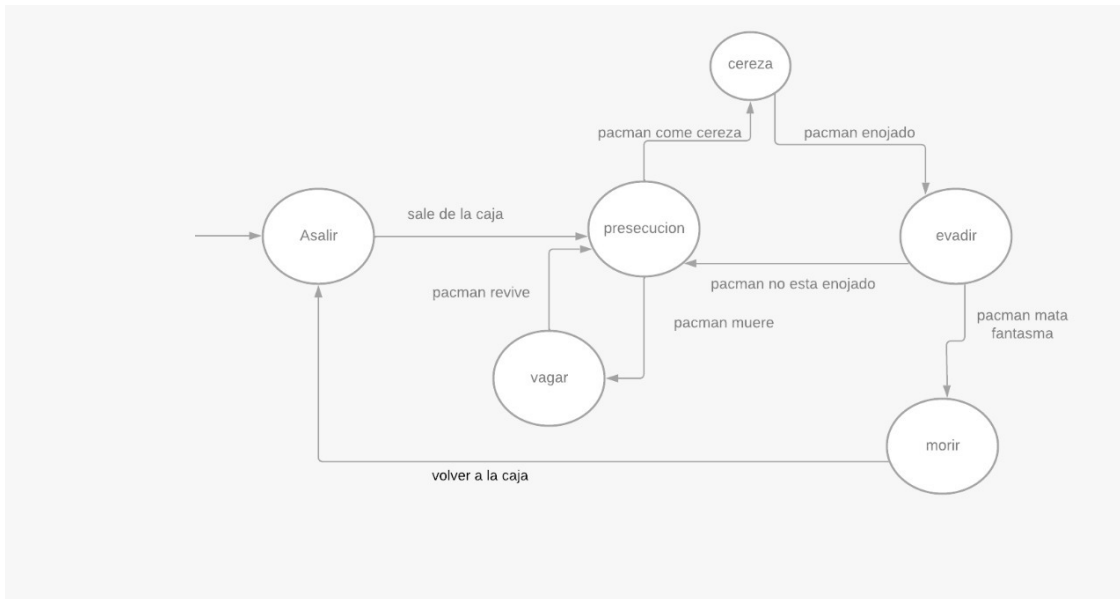
4. Diagrama de Colaboración

EL Diagrama de Colaboración presenta una alternativa al diagrama de secuencia para modelar interacciones entre objetos en el sistema. Un uso de un diagrama de colaboración es mostrar la implementación de una operación. La comunicación muestra los parámetros y las variables locales de la operación, así como asociaciones más permanentes. Cuando se implementa el comportamiento, la secuencia de los mensajes corresponde a la estructura de llamadas anidadas y el paso de señales del programa



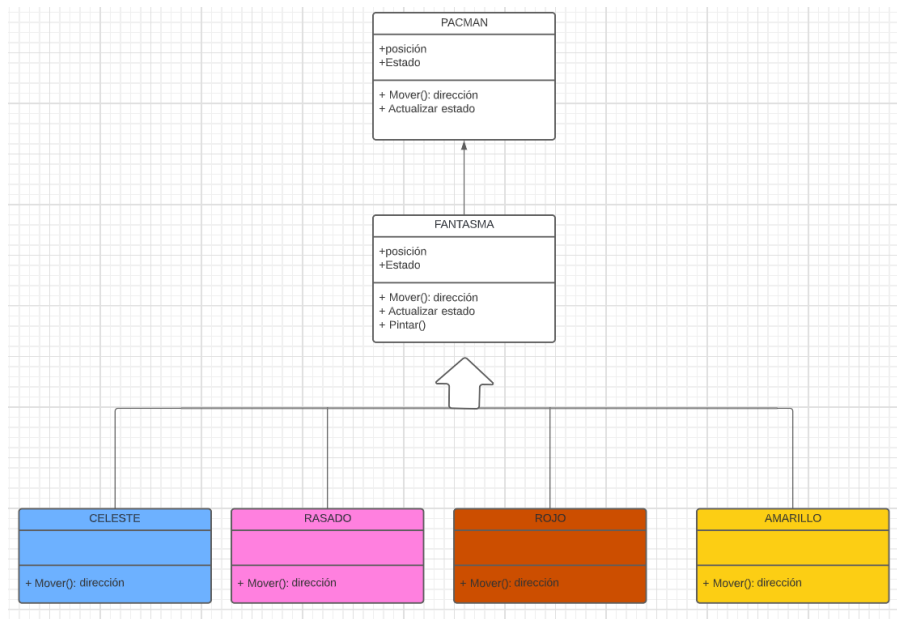
5. Diagrama de Estado

Un diagrama de estado UML (también llamado diagrama de estado, diagrama de transición de estados o diagrama de máquina de estados) muestra los estados por los que pasa una máquina de estados finitos, es decir, un modelo de comportamiento que consiste en acciones y estados o transiciones a otros estados. El diagrama proporciona un estado inicial y uno final, así como al menos un estado intermedio para cada objeto. El diagrama de estado permite, de este modo, representar el ciclo de vida completo de cualquier sistema, subsistema o componentes o clases de este, como podrían ser una máquina de café, un lector de libros electrónicos o un componente tecnológico de un vehículo



6. Diagrama de Polimorfismo

El Polimorfismo es uno de los 4 pilares de la programación orientada a objetos (POO) junto la Abstracción, Encapsulación y Herencia. Para entender con que es el polimorfismo es muy importante que tengáis bastante claro el concepto de la Herencia, por tanto, recomendamos que veáis la entrada en la que hablamos de la Herencia



3. Fases de entrega de proyecto

1. Fase 1:

En esta fase creamos los diagramas UML del tema del videojuego de PAC-MAN, los diagramas que hicimos fue los siguientes: Clases, Objetos, secuencia, colaboración, estado y polimorfismo.

2. Fase 2:

En esta segunda fase nos enfocamos en la forma visual del videojuego como lo queríamos que al final del videojuego quedara, por ejemplo, diseñamos y buscamos imágenes de internet sobre el videojuego PAC-MAN, para crear menús, botones y jugabilidad.

3. Fase 3:

En esta tercera fase empezamos a programar, diseñando el videojuego, para tenerlo terminado un 80% antes de la entrega del videojuego.

4. Fase 4


En esta fase cuatro es la entrega del proyecto, donde nos pusimos darles los últimos retoques al videojuego y realizar la documentación debida.

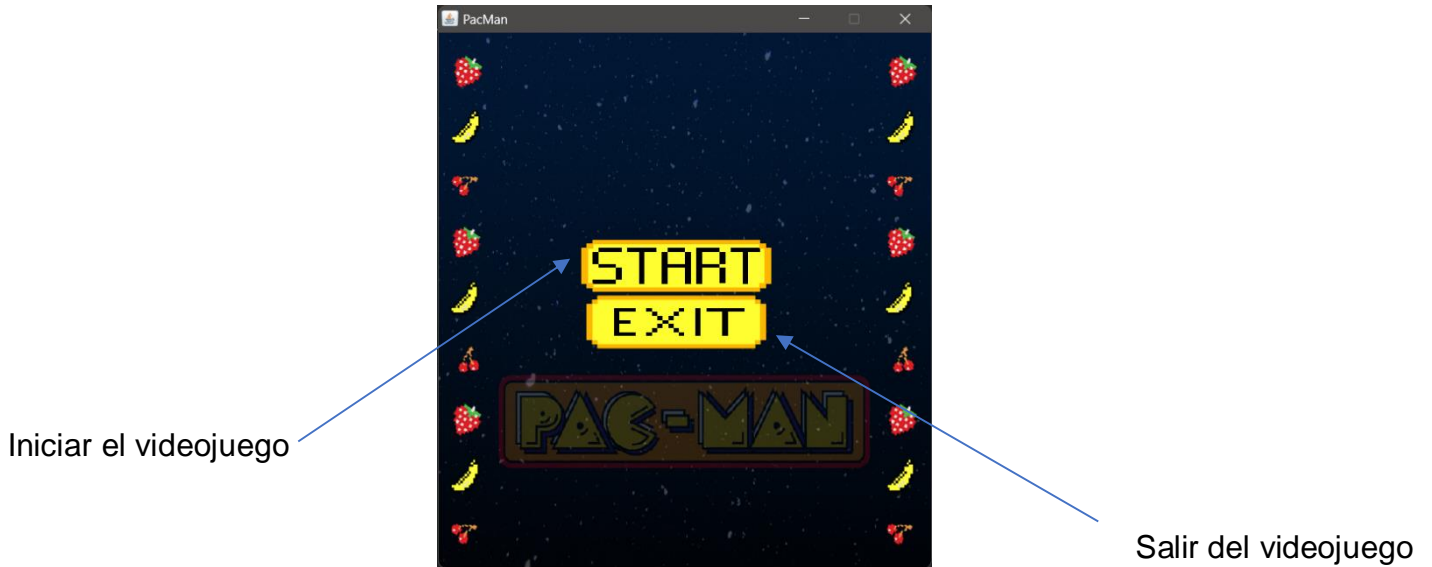
TO-DO LIST							
Task	Priority	Status	Start Date	Due Date	% Complete	Done/Overdue?	Notes
Crear diagramas UML	Normal	Complete	26/09/2022	1/10/2022	100%	✓ Done	DIAGRAMAS UML PACMAN
Desarrollo visual	High	Not Started	3/10/2022	8/10/2022	50%	🚩 Overdue	FORMA VISUAL
Desarrollo visual y logico	High	Not Started	10/10/2022	15/10/2022	75%	🚩 Overdue	EJECUCIÓN DEL PROYECTO
Entrega Final	High	Not Started	17/10/2022	22/10/2022	100%		TERMINANDO (ERRORES, OBSERVACIONES,)PROYECTO
Entrega de documentación	High	Not Started	17/10/2022	22/10/2022	100%	#¡VALORI	TERMINANDO (ERRORES, OBSERVACIONES,)PROYECTO

4. Explicación del código del videojuego

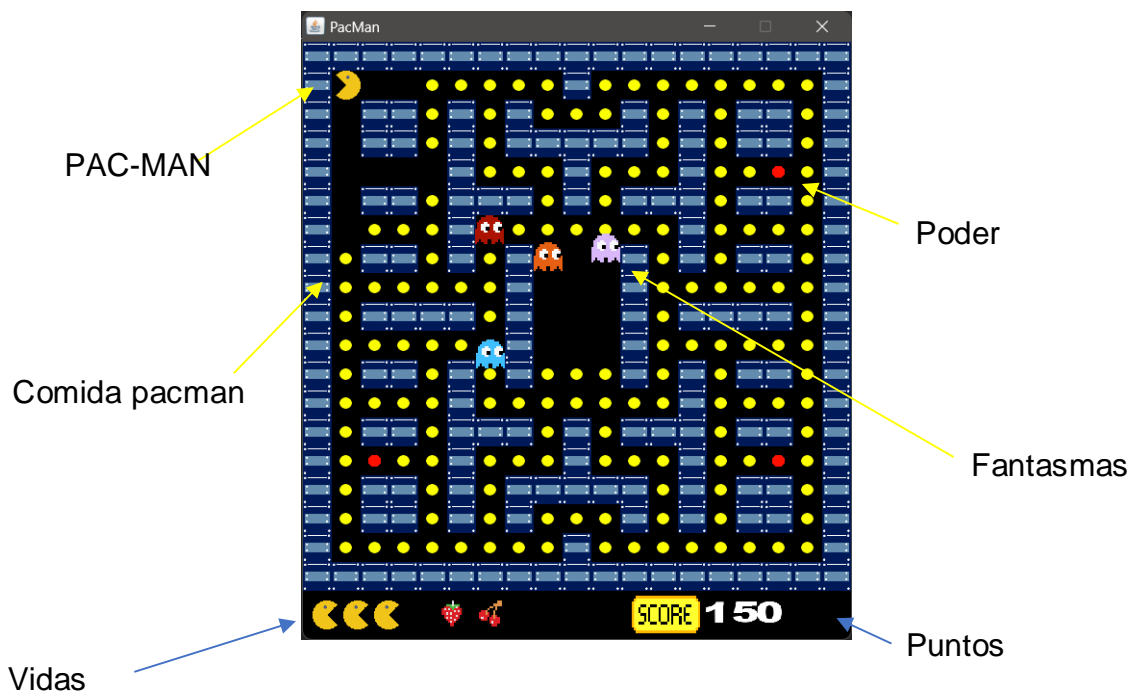
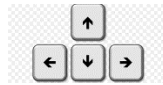
El videojuego de PAC-MAN se basó en la programación orientada objetos, en el lenguaje de java, se uso todo lo aprendido durante el curso de Programación II, se hizo un menú donde lleva dos botones lo cuales son: iniciar, salir, cada botón se programo para que obedezca su función, también se pusieron imágenes donde en cada clase, objetos o métodos se llamaban para que el videojuego se mirara animado, también se llamaba funciones de otras clases para que el código sea lo mas reducido que se pueda y funcional, se usaron varios paquetes, varias clases donde en cada una tiene una función,

5. Manual de Usuario

- Paso 1: Abrir el proyecto en NetBeans
- Paso 2: ejecutar el proyecto en el símbolo  o f6
- Paso 3: se abrirá una ventana donde tiene dos botones:

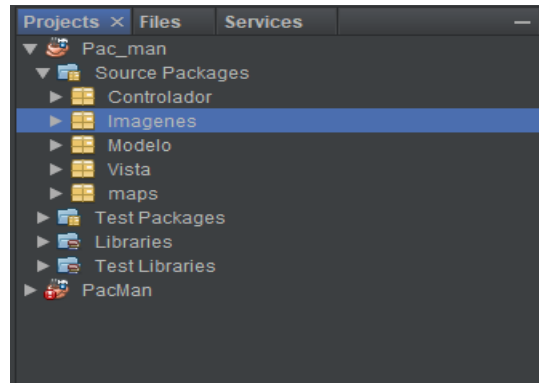


- Paso 4: al hacer clic en el botón START se abrirá el videojuego donde para controlar al pacman debemos de presionar las flechas de la computadora

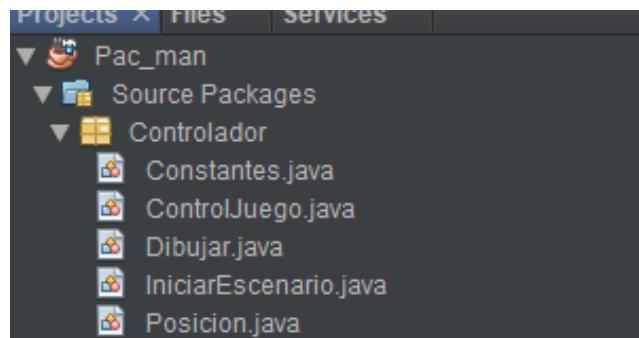


6. Capturas del código

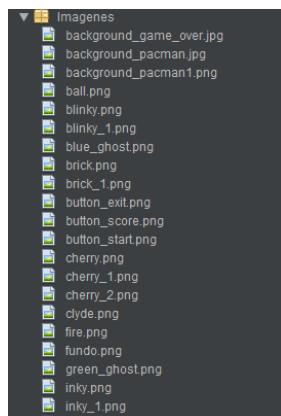
Creamos varios paquetes como se observa en la imagen que cada uno de esos paquetes tienen sus funciones que nos ayudan a que el juego funcione correctamente



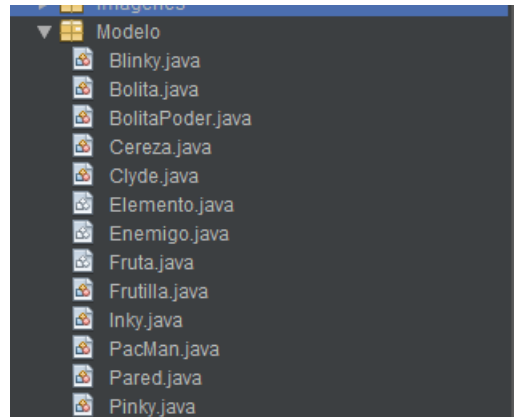
En el paquete de controlador encontramos constantes que nos ayudan en general con el videojuego, control del juego como bien el nombre lo dice es el que controla el juego y pone sus reglas, en dibujar es el que genera elementos dentro del laberinto, iniciarEcenario nos ayuda a que el juego ejecute correctamente y en posición ayuda a que al arrancar el juego pueda poner todo en orden.



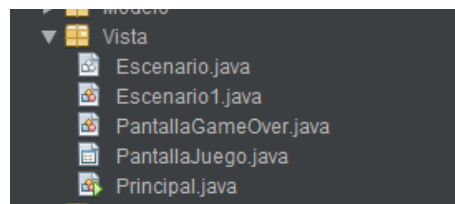
En el paquete de imágenes colocamos todas las imágenes que nos ayudan a darle vista y nos ayuda a la funcionalidad del videojuego



En el paquete modelo programamos los enemigos del pacman, programamos las bolitas de poder y las bolitas normales la que come el pacman, también programamos las paredes, la fruta y las cerezas.



En el paquete de vista encontramos clases que nos ayudan a lo visual del videojuego, como el escenario, pantalla principal del videojuego, también la pantalla al perder, y la clase principal



En el paquete maps creamos una clase map que contiene el mapa del videojuego donde el pacman juega que es el laberinto

