

**Escuela
Politécnica
Nacional**



INFORME

Diseño de Software

Proyecto Pac-Man



Grupo D

Averos D, Quishpe G, Roman P, Salazar J, Salazar S,
Yáñez D

Profesor

Ing. Richard Valarezo PhD.

INDICE

| | |
|------------------------------------|----|
| I. Tema | 3 |
| II. Objetivos..... | 3 |
| III. Introducción | 3 |
| IV. Desarrollo | 4 |
| Requisitos funcionales | 4 |
| Requisitos no funcionales | 5 |
| Tarjetas CRC | 6 |
| Diagrama de casos de uso | 9 |
| Diagramas de Clases | 11 |
| Diagramas de Secuencias..... | 12 |
| Diagrama de estados..... | 13 |
| Diagramas de Actividades | 14 |
| Patrón de diseño (Singleton) | 14 |
| V. Conclusiones | 15 |
| VI. Bibliografía | 16 |

I. Tema

Diseño de los casos de uso para el juego Pac Man bajo el apoyo de los requisitos funcionales y no funcionales.

II. Objetivos

1. Identificar requisitos funcionales y no funcionales del juego a partir de la inspección y la interacción con el jugador.
2. Elaborar el diagrama de casos de uso a partir de las funcionalidades correspondientes a cada actor.
3. Crear tarjetas CRC partiendo del establecimiento las clases identificadas en el juego, las tareas correspondientes y las clases colaboradoras.

III. Introducción

Pac-Man, lanzado por Namco en 1980, es un juego históricamente influyente y emblemático en el mundo de los videojuegos. Este clásico de los salones recreativos ha dejado una profunda marca en la cultura pop y ha tenido un impacto duradero en la industria del entretenimiento digital. Su mezcla única de jugabilidad fácil de entender, pero desafiante, personajes carismáticos y niveles ingeniosamente diseñados, cautivaron a un público global, allanando el camino para otros juegos exitosos.

La historia de Pac-Man se inicia con el diseñador de videojuegos Toru Iwatani, quien buscaba crear un juego que atrajera a un público diverso y apelara tanto a hombres como mujeres. Inspirado por la forma de una pizza a medio comer, Iwatani imaginó un personaje principal simpático y hambriento que debía recoger puntos en un laberinto mientras evitaba a los fantasmas que lo perseguían. La simplicidad, pero efectividad del diseño de Pac-Man lo hizo accesible para jugadores de todas las edades y géneros. A diferencia de muchos juegos de la época que se centraban en la violencia y la competencia directa, Pac-Man se enfocaba en estrategia y evasión. Los jugadores debían esquivar a los coloridos fantasmas Blinky, Pinky, Inky y Clyde mientras intentaban comer todos los puntos para avanzar al siguiente nivel.

La popularidad de Pac-Man fue instantánea, llenando las salas recreativas de jugadores ansiosos por probar esta novedosa experiencia de juego. La sencillez y la adictividad de Pac-Man lo convirtieron en un fenómeno cultural, generando una amplia gama de productos relacionados, desde juguetes y ropa hasta una exitosa serie de televisión y canciones populares. Además, el éxito de Pac-Man impulsó la popularidad de los juegos de arcade y abrió el camino para futuros éxitos en la industria.

En términos de su diseño, Pac-Man presentaba gráficos 2D simples pero efectivos que permitían a los jugadores identificar fácilmente a los personajes y el entorno del juego. El laberinto tenía una vista cenital que proporcionaba una visión clara de los movimientos y acciones de Pac-Man y los fantasmas. Los controles eran simples, utilizando un joystick para mover al personaje principal a través del laberinto.

En el marco de este proyecto, se introduce el icónico videojuego PACMAN, que ha sido abordado desde una perspectiva de diseño y arquitectura. En este contexto, se ha adoptado un enfoque fundamentado en los principios de la programación orientada a objetos, la creación de tarjetas CRC y la elaboración de diagramas conforme al lenguaje de modelado UML. Esta metodología proporciona una estructura organizada para la

implementación del software, permitiendo un desarrollo más eficiente y comprensible. Para llevar a cabo estas tareas, se ha elegido el lenguaje de programación Python, conocido por su robustez en lo que respecta a la programación orientada a objetos, pero sin las restricciones sintácticas más rígidas que se encuentran en lenguajes como Java o C++. Este enfoque no solo facilita la creación del juego, sino que también promueve la mantenibilidad y la escalabilidad del código, brindando así una base sólida para la materialización de un PACMAN interactivo y de alta calidad.

IV. Desarrollo

Requisitos funcionales

1. El juego dispondrá de un menú principal el cual consta de las siguientes opciones: Jugar, Marcador y Salir.
2. El juego dispondrá de un solo laberinto.
3. El juego permitirá al jugador mover a Pac-Man a lo largo del laberinto en 2D, según la tecla de dirección que el usuario digite, teniendo 4 posibilidades: arriba, abajo, izquierda, derecha.
4. El juego incrementará la dificultad y velocidad de los fantasmas cada vez que se complete el laberinto.
5. El juego colocará dos tipos de píldoras (normales y especiales) a lo largo del laberinto que puede comer Pac-Man.
6. El juego incluirá frutas en ubicaciones específicas del laberinto para ofrecer puntos adicionales (100pts) al ser consumidas.
7. El juego incrementará 10 o 50 puntos cuando Pac-Man coma una píldora normal o especial respectivamente.
8. El juego incrementará la puntuación cuando Pac-Man se coma a los fantasmas. Los puntos ganados deben seguir la siguiente lógica:
 - Si Pac-Man come al primer fantasma, se deben sumar 200 puntos a su puntuación actual.
 - Si Pac-Man come al segundo fantasma, se deben sumar 400 puntos adicionales.
 - Si Pac-Man come al tercer fantasma, se deben sumar 800 puntos adicionales.
 - Si Pac-Man come al cuarto fantasma, se deben sumar 1600 puntos adicionales.
9. El juego esconderá la fruta de cada mapa después de que pase 60 segundos sin ser consumida.
10. El juego hará que los fantasmas sean vulnerables y comestibles durante un tiempo limitado (10 segundos), si Pac-Man come una píldora especial.
11. El juego dispondrá con un contador de vidas del jugador (Max. 3).
12. El juego decrementará una vida al jugador cada vez que un fantasma atrape a Pac-Man.
13. El juego incrementará una vida al jugador cada vez que complete el laberinto exitosamente.
14. El juego terminará la partida en el caso de que el jugador tenga cero vidas.
15. El juego debe implementar el comportamiento del fantasma rojo (Blinky) de manera que persiga directamente a Pac-Man aumentando su velocidad a medida que Pac-Man consigue puntos.

16. El juego debe implementar el comportamiento del fantasma azul (Inky) de manera que calcule la distancia entre Blinky y Pac-Man, y luego imite la dirección y la distancia invertida de Blinky a 180°. Esto hará que Inky sea peligroso y a menudo atrape a Pac-Man entre él y otro fantasma.
17. El juego debe implementar el comportamiento del fantasma rosa (Pinky) de manera que siempre trate de atrapar a Pac-Man por enfrente. Pinky debe moverse en dirección al punto cuatro espacios por delante de Pac-Man en lugar de seguir directamente a su posición actual.
18. El juego debe implementar el comportamiento del fantasma naranja (Clyde) de manera que persiga a Pac-Man de manera similar a Blinky, pero cuando esté dentro de un radio de ocho espacios de Pac-Man, comenzará a retirarse a la esquina inferior izquierda del laberinto.
19. El juego desplegará la tabla de marcadores cuando el jugador elija la opción "Marcadores" en el menú principal.
20. El juego permitirá al jugador pausar la partida en cualquier momento mediante una tecla o un botón específico, lo que congelará la acción en pantalla y proporcionará opciones para reanudar o salir del juego.
21. El juego incluirá efectos de sonido distintivos para acciones como mover a Pac-Man, comer píldoras, comer fantasmas, perder una vida y completar el laberinto. También se puede incluir música de fondo opcional para mejorar la atmósfera del juego.
22. El juego le permitirá al jugador terminar el programa en cualquier momento al presionar la tecla ESC.

Requisitos no funcionales

1. Rendimiento:

El juego debe funcionar de manera fluida y sin demoras sustanciales en una variedad extensa de dispositivos y configuraciones, en consonancia con el Requerimiento Funcional 4. Dado que el incremento de la velocidad de los fantasmas puede tener un impacto en el rendimiento del juego, es crucial mantener la armonía entre estos elementos.

2. Usabilidad:

La interfaz del juego ha de ser diseñada de manera intuitiva y presentar una usabilidad accesible, permitiendo a los jugadores aprender y participar en el juego sin dificultades o barreras significativas.

3. Estabilidad

El juego debe mantenerse libre de bloqueos, congelamientos y errores graves a lo largo de la experiencia de juego, garantizando una ejecución fluida y sin interrupciones.

4. Estética:

Los elementos visuales, auditivos y efectos gráficos del juego deben ser estéticamente agradables y mantener coherencia con el estilo característico de Pac-Man. De esta manera, se busca mantener la identidad visual y sonora reconocible del juego, brindando una experiencia visualmente atractiva y coherente con la temática original de Pac-Man.

5. Tiempo de carga:

El juego debe contar con tiempos de carga reducidos, de manera que los jugadores puedan iniciar su partida de manera rápida y sin demoras significativas. Este requisito busca asegurar una experiencia fluida y sin interrupciones desde el momento en que el jugador inicia el juego, permitiéndole sumergirse rápidamente en la acción sin tener que esperar largos periodos de carga.

6. Respuesta rápida:

El juego debe ofrecer una respuesta inmediata a las acciones del jugador, evitando retrasos significativos que puedan afectar la jugabilidad. Este requisito busca garantizar una experiencia de juego fluida y en tiempo real, donde las acciones del jugador sean procesadas y reflejadas de manera instantánea en el juego, sin demoras perceptibles que puedan afectar la interacción y la inmersión del jugador.

7. Cohesión temática:

Todos los elementos del juego, desde los niveles y personajes hasta la música, deben mantener una coherencia estrecha con la temática original de Pac-Man. Este requisito busca asegurar que todos los aspectos del juego estén alineados con la esencia y el estilo distintivo de Pac-Man, preservando así la atmósfera y la identidad del juego original. Al mantener esta sintonía temática, se busca brindar a los jugadores una experiencia auténtica y fiel a la esencia del clásico Pac-Man, permitiéndoles sumergirse en un mundo cohesivo y reconocible dentro del juego.

8. Compatibilidad:

El juego deberá ser compatible con todas las plataformas y sistemas operativos que admitan Python y Pygame sin necesidad de modificaciones significativas.

Tarjetas CRC

| Main | |
|--|---|
| Responsabilidades | Colaboradores |
| <ul style="list-style-type: none">• Inicialización del videojuego.• Bucle del videojuego.• Control de flujo. | <ul style="list-style-type: none">• Level• Game• Pacman• Ghost• Path• MainMenu• Sound |

| Main Menu | |
|--|--|
| Responsabilidades | Colaboradores |
| <ul style="list-style-type: none">• Inicialización del videojuego. | <ul style="list-style-type: none">• Main |

| | |
|--|--|
| <ul style="list-style-type: none"> • Bucle del videojuego. • Control de flujo. | |
|--|--|

| Game | |
|---|--|
| Responsabilidades | Colaboradores |
| <ul style="list-style-type: none"> • Iniciar una nueva partida. • Dibujar el mapa. • Dibujar las vidas. • Pausar videojuego. • Terminar videojuego. • Completar nivel. • Revisar inputs de teclado. • Dibujar puntaje. • Dibujar pastillas y frutas. • Salir del juego. | <ul style="list-style-type: none"> • Ghost • Level • Pacman • Path • Utils • Sound |

| Level | |
|--|---|
| Responsabilidades | Colaboradores |
| <ul style="list-style-type: none"> • Cargar nivel. • Verificar si es pared. • Reiniciar nivel al momento que Pac-Man es comido. | <ul style="list-style-type: none"> • Game • Pacman • Ghost • Path |

| Pacman | |
|---|--|
| Responsabilidades | Colaboradores |
| <ul style="list-style-type: none"> • Dibujar Pac-man. • Movimiento de Pac-man. • Incrementar puntaje de Pac-man. • Verificar si choca con paredes, fantasmas, pastillas y frutas. | <ul style="list-style-type: none"> • Level • Game • Ghost • Path • Utils • Sound |

| Ghost | |
|--|---|
| Responsabilidades | Colaboradores |
| <ul style="list-style-type: none"> • Reiniciar propiedades de fantasma al reiniciar juego. • Dibujar fantasma. | <ul style="list-style-type: none"> • Level • Game • Pacman |

| | |
|--|---|
| <ul style="list-style-type: none"> • Movimiento de fantasma. • Buscar siguiente ruta para fantasma. • Incrementar la velocidad de fantasma al completar el laberinto. | <ul style="list-style-type: none"> • Path • Utils |
|--|---|

| Path Finder | |
|---|--|
| Responsabilidades | Colaboradores |
| <ul style="list-style-type: none"> • Buscar una ruta a seguir por los fantasmas. | <ul style="list-style-type: none"> • Node |

| Node | |
|--|---|
| Responsabilidades | Colaboradores |
| <ul style="list-style-type: none"> • Representar nodos en un grafo, en este caso el mapa. | <ul style="list-style-type: none"> • Path Finder |

| Utils | |
|---|---|
| Responsabilidades | Colaboradores |
| <ul style="list-style-type: none"> • Obtener y retornar imágenes del source. | <ul style="list-style-type: none"> • Ghost • Pacman • Game |

| Sound | |
|---|--|
| Responsabilidades | Colaboradores |
| <ul style="list-style-type: none"> • Obtener y reproducir sonidos dependiendo del número que se mande. | <ul style="list-style-type: none"> • Pacman • Main • Game |

Diagrama de casos de uso

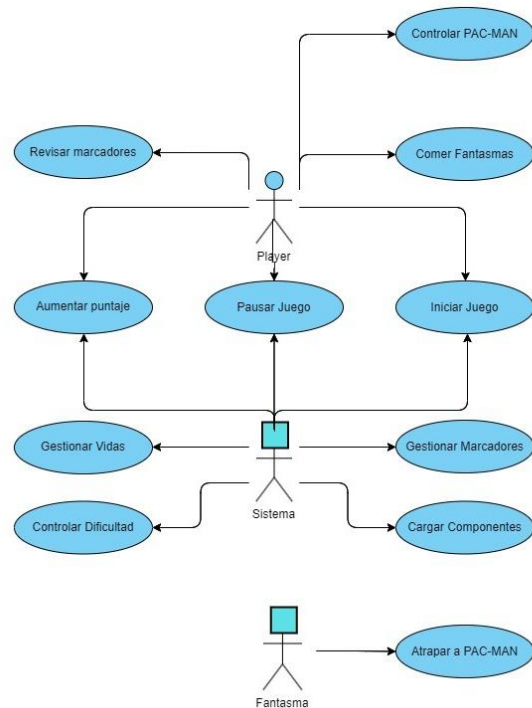


Figura 1. Diagrama de casos de uso para el juego de Pac-Man.

- Controlar PAC-MAN
 - El jugador puede mover a Pac-Man hacia arriba, abajo, izquierda o derecha utilizando las teclas de dirección.
 - Pac-Man se desplaza a lo largo del laberinto según las teclas de dirección ingresadas por el jugador.
 - Pac-Man puede comer las píldoras normales y especiales mientras se desplaza por el laberinto.
- Comer Fantasmas
 - Pac-Man obtiene una puntuación adicional por comer fantasmas, que varía dependiendo del orden en que los come.
- Revisar marcadores
 - El jugador puede seleccionar la opción "marcadores" para ver la tabla de clasificación.
- Aumentar puntaje
 - Si Pac-Man come una píldora especial, los fantasmas se vuelven vulnerables y comestibles durante 10 segundos.
- Pausar juego
 - El jugador puede pausar la partida en cualquier momento mediante una tecla o un botón específico.
 - La acción en pantalla se congela y se proporcionan opciones para reanudar o salir del juego.
- Iniciar juego
 - El jugador puede seleccionar la opción "jugar" para comenzar una partida.

- Gestionar vida
 - El jugador pierde una vida cada vez que un fantasma atrapa a Pac-Man.
 - El jugador gana una vida cada vez que completa exitosamente el laberinto.
 - El juego termina si el jugador tiene cero vidas.
- Gestionar marcadores
 - Cuando Pac-Man consume una fruta, se agregan 100 puntos adicionales a la puntuación del jugador.
 - Pac-Man obtiene 10 puntos adicionales por comer una píldora normal.
 - Pac-Man obtiene 50 puntos adicionales por comer una píldora especial.
 - Por comer al primer fantasma, Pac-Man obtiene 200 puntos adicionales.
 - Por comer al segundo fantasma, Pac-Man obtiene 400 puntos adicionales.
 - Por comer al tercer fantasma, Pac-Man obtiene 800 puntos adicionales.
 - Por comer al cuarto fantasma, Pac-Man obtiene 1600 puntos adicionales.
- Controlar dificultad
 - Después de completar cada laberinto, el juego incrementa la dificultad y velocidad de los fantasmas.
- Cargar componentes
 - El juego presenta el laberinto donde se desarrolla la partida.
 - El juego coloca píldoras normales y especiales en el laberinto.
 - El juego incluye frutas en ubicaciones específicas del laberinto.
 - El juego muestra un contador de vidas del jugador con un máximo de 3 vidas.
 - El juego incluye efectos de sonido distintivos para acciones como mover a Pac-Man, comer píldoras, comer fantasmas, perder una vida y completar el laberinto.
- Atrapar a PAC-MAN
 - Comportamiento del fantasma rojo (Blinky)
 - El fantasma rojo persigue directamente a Pac-Man y aumenta su velocidad a medida que Pac-Man obtiene puntos.
 - Comportamiento del fantasma azul (Inky)
 - El fantasma azul calcula la distancia entre Blinky y Pac-Man.
 - Inky imita la dirección y la distancia invertida de Blinky a 180°, lo que a menudo atrapa a Pac-Man entre él y otro fantasma.
 - Comportamiento del fantasma rosa (Pinky)
 - El fantasma rosa siempre trata de atrapar a Pac-Man por delante.
 - Pinky se mueve hacia el punto cuatro espacios por delante de Pac-Man en lugar de seguir directamente su posición actual.
 - Comportamiento del fantasma naranja (Clyde)
 - El fantasma naranja persigue a Pac-Man de manera similar a Blinky.
 - Cuando está dentro de un radio de ocho espacios de Pac-Man, Clyde comienza a retirarse a la esquina inferior izquierda del laberinto.

Diagramas de Clases

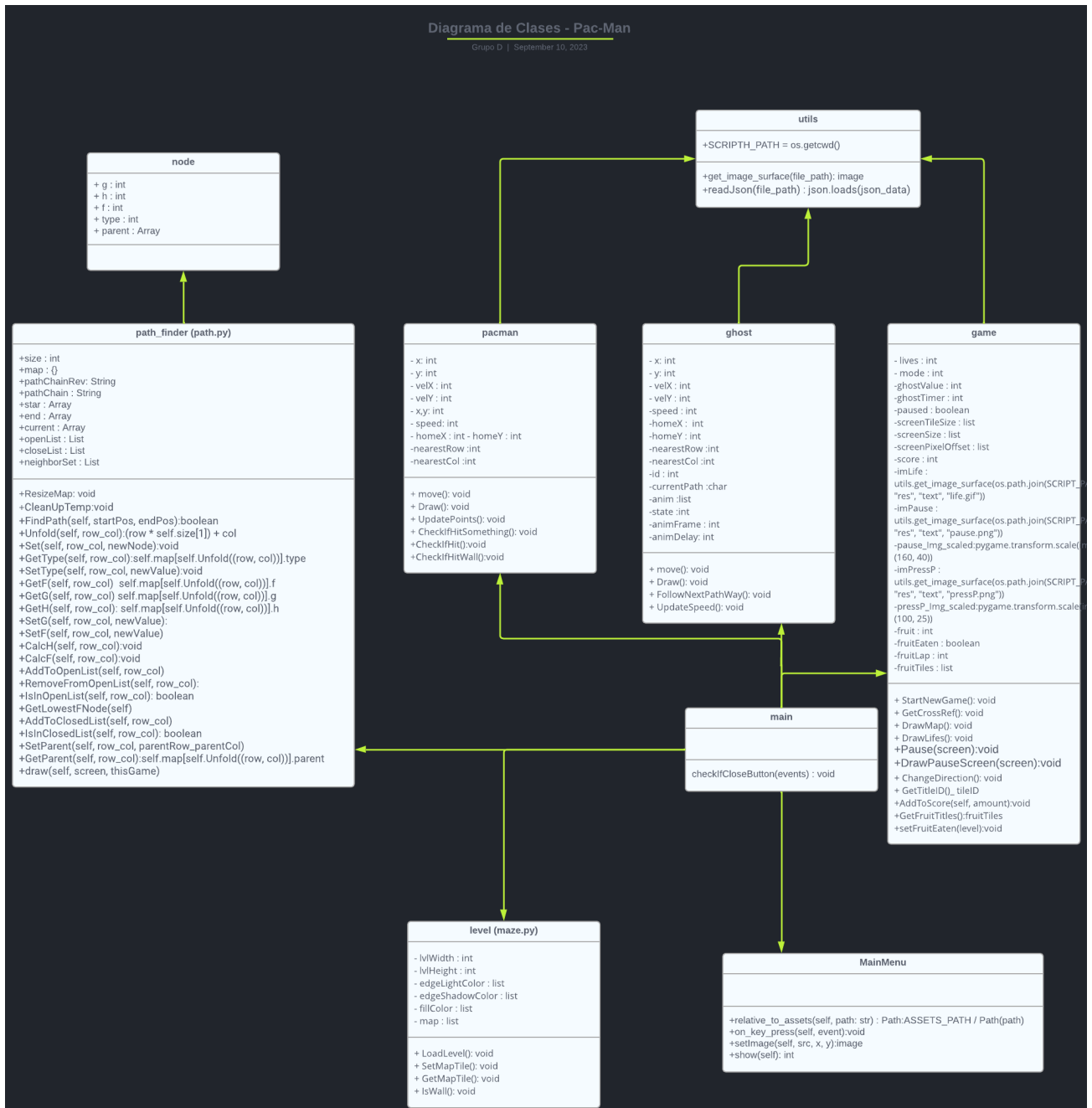


Figura 2. Diagrama de clases del videojuego Pacman

Diagramas de Secuencias

○ Comer pastillas

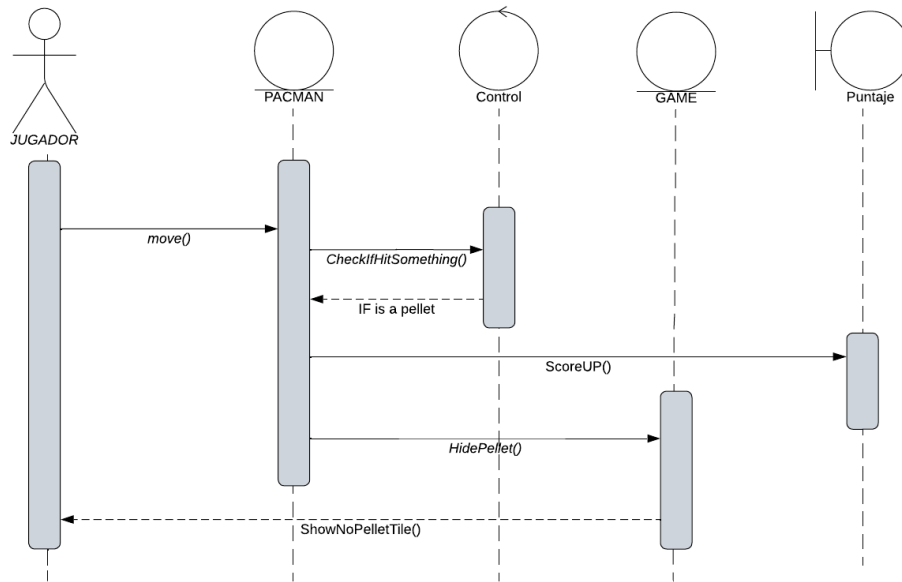


Figura 3. Diagrama de secuencias de "comer pastillas"

○ Comer Power Up

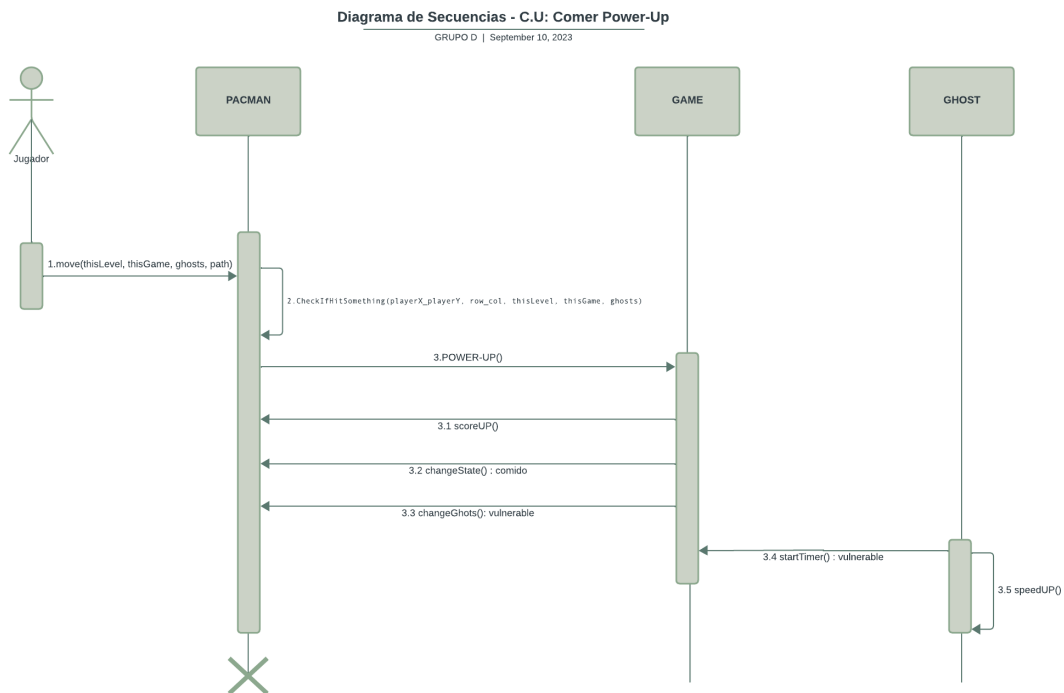


Figura 4. Diagrama de secuencias de "comer power up"

○ Diagrama Cargar score

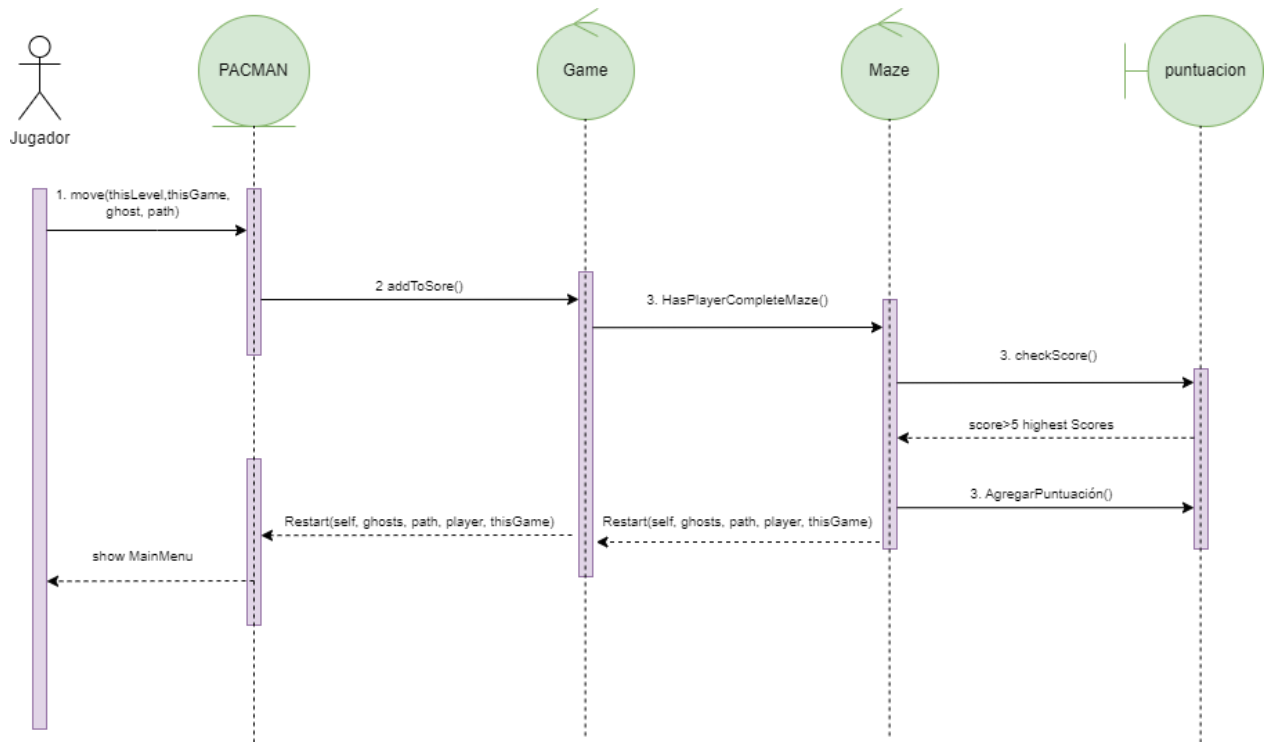


Diagrama de estados

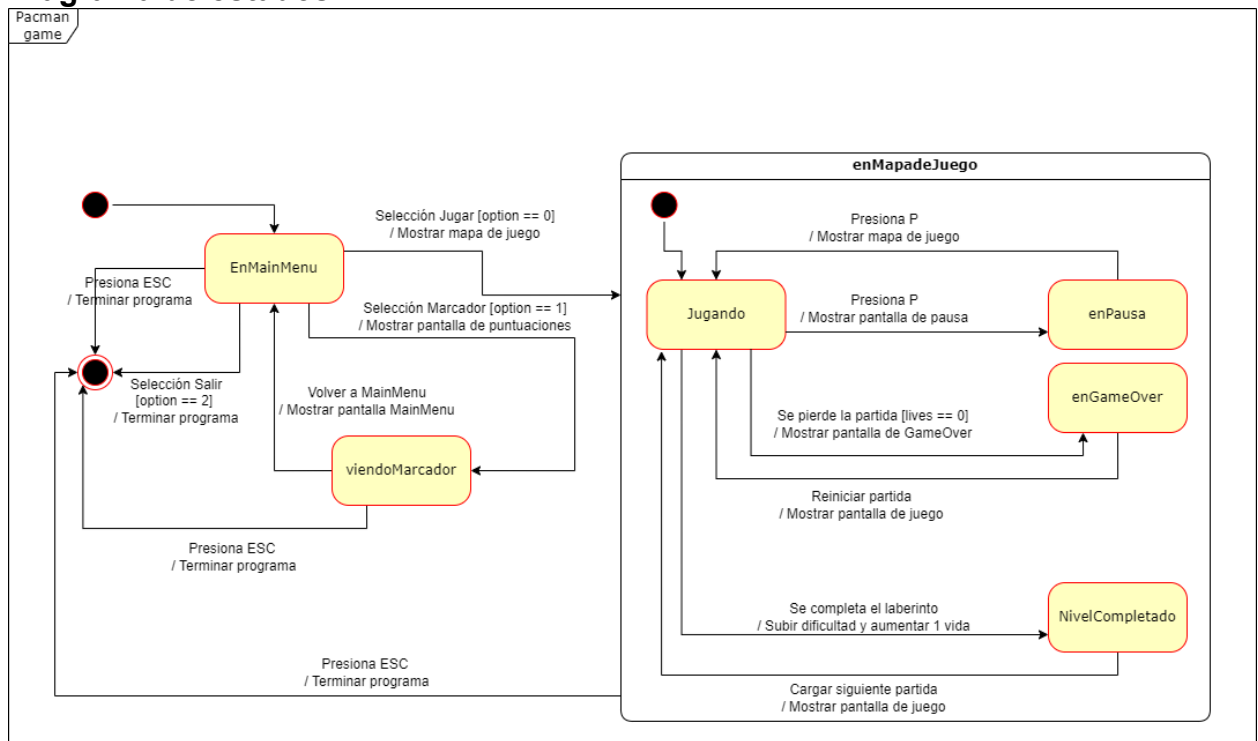


Figura 5. Diagrama de estados del videojuego Pacman

Diagramas de Actividades

Movimiento de Pac-Man

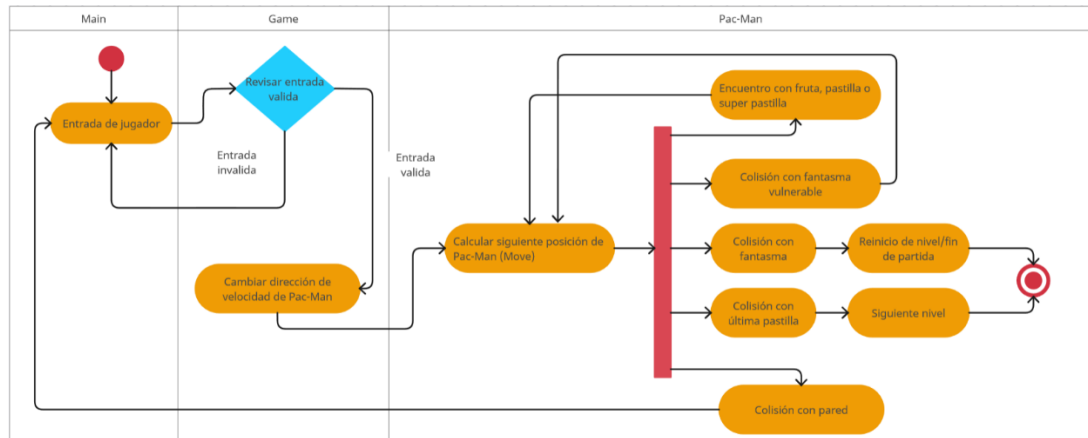


Figura 6. Diagrama de actividades del movimiento de Pacman

Patrón de diseño (Singleton)

La implementación del patrón Singleton se mostró como una elección estratégica fundamental para garantizar un control coherente y eficiente de los recursos compartidos en el juego. El patrón Singleton se aplicó en la gestión de la instancia de la clase Sonido. Al hacerlo, se aseguró que solo existiera una única instancia de este elemento en toda la aplicación, lo que facilitó la sincronización y la reproducción de los sonidos dentro del juego. Esta implementación se justificó por su capacidad para prevenir problemas de concurrencia al evitar la reproducción de sonidos en bucle. Se optó por este patrón con el objetivo de no crear múltiples instancias, optimizando así la administración de recursos y la coherencia de la experiencia del jugador.

```
class sound:
    _instance = None # Almacena la única instancia de la clase

    def __new__(cls):
        if cls._instance is None:
            cls._instance = super(sound, cls).__new__(cls)
        return cls._instance
```

Figura 7. Implementación del patrón Singleton en la clase Sound

Ejemplo de uso

En las clases **Game** y **Pacman** se creó un atributo llamado "soundInstance" en sus respectivos métodos constructores, como se puede ver en la Figura 8.

```
class game:
    def __init__(self):
        self.soundInstance = sound.sound()
```

Figura 8. Instanciación de la clase Sound

Finalmente, a partir de esta instancia se llaman los distintos métodos para reproducir los sonidos según las acciones correspondientes ocurridas en el juego, como se indica en la Figura 9 con un método de la clase Game.

```
def StartNewGame(self, thisPath, thisLevel, player, ghosts):
    self.score = 0
    self.lives = 3
    self.pause = False
    #Music
    self.soundInstance.SetMode(1)
    self.soundInstance.snd_levelintro.play()
    self.mode = 1
    thisLevel.LoadLevel(thisPath)
    thisLevel.Restart(ghosts, thisPath, player, self)
```

Figura 9. Uso de la instancia de Sound

V. Conclusiones

A partir del planteamiento de los requisitos funcionales se pudo establecer una base sólida para el desarrollo del juego de Pac-Man. Los requisitos funcionales identificados nos proporcionaron una comprensión clara de las funcionalidades y comportamientos esperados del juego, permitiéndonos definir de manera precisa las interacciones entre los actores y el sistema.

Por otro lado, con la definición de los requisitos no funcionales pudimos establecer las restricciones y características técnicas necesarias para el desarrollo del juego de Pac-Man. Estos requisitos no funcionales nos permitieron considerar aspectos como el rendimiento, la seguridad, la usabilidad y la escalabilidad, asegurando que el juego se cumpla con los estándares y requisitos de calidad esperados.

Mediante el diagrama de casos de uso, pudimos visualizar de forma clara y concisa cómo los distintos elementos del juego interactúan entre sí, lo que facilitó la comprensión general del flujo de juego.

Además, las tarjetas CRC nos permitieron asignar las responsabilidades a cada clase, asegurando una distribución adecuada de tareas y una identificación clara de las clases colaboradoras involucradas en el desarrollo del juego., logrando un diseño más claro y coherente del sistema del juego de Pac-Man.

El juego contiene todas las características necesarias para que Pac-Man se mueva adecuadamente en el juego y choque con los elementos del entorno de acuerdo con las reglas del juego.

A partir del diagrama de estados se obtiene una representación visual de todas las posibles transiciones entre estados que el juego puede experimentar durante su ejecución. Esto permite una comprensión más profunda de cómo funcionan los componentes del juego y cómo interactúan entre sí.

El diseño de un videojuego es un arte que combina creatividad y lógica. Este proyecto ilustra cómo la planificación cuidadosa y la implementación de patrones de desarrollo de software pueden dar vida a experiencias interactivas memorables. Al fusionar la imaginación con la

estructura, se demuestra que los videojuegos son un equilibrio entre la visión artística y la ingeniería de software.

La implementación del patrón de diseño Singleton resultó fundamental para abordar la necesidad de instancias de la clase "sonido" en diversas clases, lo que permitió un ahorro significativo de memoria al mantener una única instancia.

VI. Bibliografía

Barbero, C., & María, E. (s/f). *Desarrollo del pensamiento computacional a través de la programación y la robótica*. Educalab.es. Recuperado el 3 de julio de 2023, de <http://educalab.es/documents/10180/640047/TallerProgramacionvideojuegosI-pacman.pdf/de15bd77-f0dc-410b-88f1-d6616f0dd6bf>

de Grado para acceder al, T. de F. (s/f). *FANTASMAS INTELIGENTES PARA EL PAC-MAN BASADOS EN SISTEMAS MULTIAGENTE*. Unican.es. Recuperado el 3 de julio de 2023, de <https://repositorio.unican.es/xmlui/bitstream/handle/10902/7685/Sergio%20Salomon%20Garcia.pdf?sequence=1>

(S/f). Ucm.es. Recuperado el 3 de julio de 2023, de <https://gaia.fdi.ucm.es/research/mspacman/files/tutorial.pdf>