

UNIVERSIDAD DE MARGARITA

ALMA MATER DEL CARIBE

VICERRECTORADO ACADÉMICO

DECANATO DE INGENIERIA

UNIDAD CURRICULAR: Programación II

**Informe**

**Pacman**

Profesor: Integrantes:

Cesar Requena -Manuel Casique C.I:29.864.784 -Tatiana Mero C.I: 27.261.683

El proyecto "Pac-Man" es un juego desarrollado en C# que rinde homenaje al icónico juego de arcade "Pac-Man", creado por el diseñador de videojuegos japonés Toru Iwatani y lanzado por Namco en 1980. "Pac-Man" se convirtió en uno de los videojuegos más reconocidos y queridos de todos los tiempos, estableciendo un estándar para los juegos de laberinto y marcando el inicio de una era en la industria de los videojuegos.

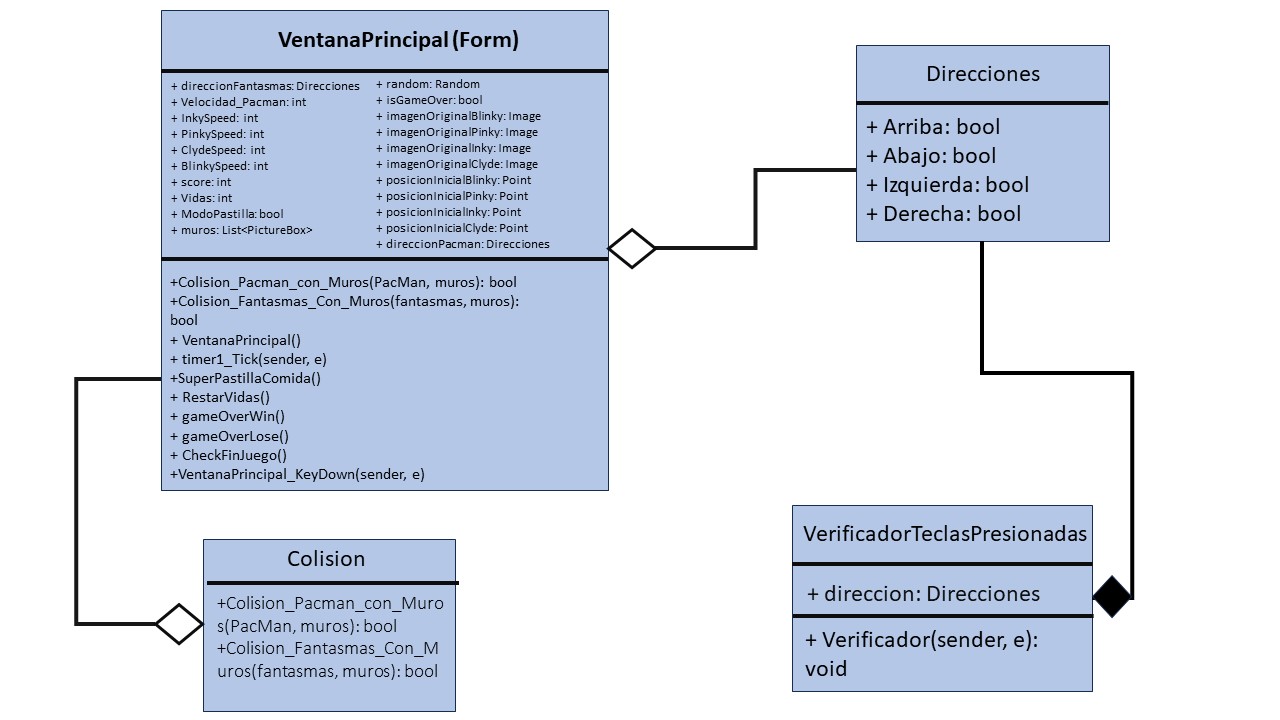
Durante el desarrollo del proyecto, existieron diversas vertientes para idear una forma de recrear el famoso juego pacman en c#, ¿Cómo crear el tablero?, ¿Cómo hacer que los fantasmas pudieran moverse solos, activar el modo pastilla y además terminar el juego? Requirió una investigación rigurosa para practicar en c# y así tener una idea mas clara de cómo realizar el pacman.

El primer paso fue crear una picture box, que representaría el personaje principal de este videojuego, “PacMan” una vez terminado eso fue cuestión de marcar los limites de la ventana, e insertar un verificador de teclas para que así la ventana reconociera las teclas AWSD y las Flechas direccionales, esto para darle instrucciones al pacman de como moverse; posterior a esto, el siguiente paso era claro, crear un sistema de colisiones del pacman para las paredes o “muros”, una vez aclarada la idea, decidí crear una lista llamada “muros” que agruparía todas las paredes o picture box que fuera creando y con las que yo quisiera que pacman chocara.

Fue bastante complicado, pero gracias a la propiedad de Intersección, se logro terminar el sistema de colisiones de forma exitosa, así como posteriormente el sistema de movimiento de pacman. Al terminar esto, solo quedaba averiguar como hacer que los Fantasmas, llamados Pinky, Blinky, Inky y Clyde respectivamente, se movieran por su cuenta; después de muchas ideas, clases y métodos fallidos, me decante por algo mas sencillo, un simple movimiento unidireccional, al que solo debía tocarse la velocidad para que así los 4 enemigos de nuestro Personaje pudieran moverse y ser un obstáculo para conseguir la Victoria.

Finalmente, se agregaron las pequeñas pastillas, que representan los puntos en el juego, y que al recolectarlos todos, estos el juego podría terminar, dándote un mensaje de felicitaciones o invitándote a volverlo a intentar si es que fallaste.

Podría decirse que el método más difícil fue el como implementar las super pastillas, pero si vemos las pastillas como una fruta o pastilla común que toman esa propiedad de forma temporal para poder ser “comido”, y que así luego fueran teletransportados a otro lugar del mapa.

Ahora a continuación el Diagrama de Clases de nuestro Proyecto:

Finalmente, el PseudoCodigo de nuestro Proyecto:

Clase VentanaPrincipal:

Variables globales:

Direcciones direccionPacman

Direcciones direccionFantasmas

Entero Velocidad\_Pacman, InkySpeed, PinkySpeed, ClydeSpeed, BlinkySpeed, score, Vidas

Booleano ModoPastilla

Lista de PictureBox muros

Objeto Random random

Booleano isGameOver

Imagen imagenOriginalBlinky, imagenOriginalPinky, imagenOriginalInky, imagenOriginalClyde

Punto posicionInicialBlinky, posicionInicialPinky, posicionInicialInky, posicionInicialClyde

Función Colision\_Pacman\_con\_Muros(PictureBox PacMan, List<PictureBox> muros):

Para cada Pared en muros:

Si PacMan.Bounds se interseca con Pared.Bounds:

Retornar Verdadero

Retornar Falso

Función Colision\_Fantasmas\_Con\_Muros(List<PictureBox> fantasmas, List<PictureBox> muros):

Para cada fantasma en fantasmas:

Para cada pared en muros:

Si el Bounds de fantasma se interseca con el Bounds de pared:

Retornar Verdadero

Retornar Falso

Constructor VentanaPrincipal():

Configurar eventos y temporizadores

Inicializar variables globales y listas de muros

Configurar las imágenes originales de los fantasmas

Agregar muros a la lista de muros

Evento PacMan\_Click(sender, e):

Manejar evento de clic en PacMan

Evento Form1\_Load(sender, e):

Manejar evento de carga del formulario

Clase Direcciones:

Propiedades Arriba, Abajo, Izquierda, Derecha

Clase VerificadorTeclasPresionadas:

Constructor VerificadorTeclasPresionadas(Direcciones D)

Función Verificador(sender, e):

Detectar teclas presionadas y actualizar direcciones

Función timer1\_Tick(sender, e):

Actualizar puntuación y vidas

Verificar si el juego ha terminado

Mover a PacMan

Controlar el teletransporte de PacMan

Manejar colisiones con monedas, frutas, superpastillas y fantasmas

Actualizar posiciones de los fantasmas

Función SuperPastillaComida():

Activar el ModoPastilla

Cambiar la imagen de los fantasmas a asustados

Configurar temporizador para revertir las imágenes de los fantasmas

Función RestarVidas():

Restar una vida

Reiniciar la posición de PacMan

Comprobar si se han agotado las vidas o si se pueden seguir jugando

Función gameOverWin():

Marcar el juego como terminado

Detener el temporizador

Mostrar mensaje de victoria

Función gameOverLose():

Marcar el juego como terminado

Detener el temporizador

Mostrar mensaje de derrota

Función CheckFinJuego():

Comprobar si se han recogido todas las monedas

Mostrar mensaje de victoria si es el caso

Evento VentanaPrincipal\_KeyDown(sender, e):

Manejar las teclas presionadas y actualizar direcciones

Otras funciones de control de eventos de PictureBox (no se muestran todas)