

Implementación de Pac-Man en el simulador MARIE

Isabella Tulcán, Samantha Salazar, María Eulalia Moncayo
Organización de Computadores

Abstract—Este documento presenta el desarrollo de un juego tipo Pac-Man utilizando el simulador MARIE. Se implementaron funciones básicas como el movimiento aleatorio de personajes, detección de colisiones, manejo de puntos y vidas, y condiciones de victoria y derrota, todo bajo las limitaciones de la arquitectura simplificada de MARIE.

I. INTRODUCCIÓN

MARIE (Machine Architecture that is Really Intuitive and Easy) es una arquitectura educativa simplificada usada para enseñar principios de arquitectura de computadores y programación en lenguaje ensamblador. Este trabajo presenta la implementación de un juego inspirado en Pac-Man, usando las instrucciones limitadas de MARIE.

El objetivo del proyecto fue explorar la lógica de programación de bajo nivel para implementar una versión funcional del juego, que incluya movimiento, detección de colisiones con enemigos y objetos, y la actualización de puntajes.

II. METODOLOGÍA

Para la implementación del juego se utilizó el entorno de desarrollo del simulador MARIE. Las principales etapas fueron:

- Representar el mundo como una grilla de celdas codificadas con colores (números hexadecimales).
- Crear un arreglo de 100 números aleatorios en Python para representar direcciones de movimiento.
- Implementar la lectura secuencial de movimientos desde memoria.
- Controlar el movimiento del Pac-Man usando sumas o restas de posiciones.
- Detectar colisiones con fantasmas, esteroides y monedas.
- Manejar el contador de poder activado por los esteroides.
- Controlar el *score* y condiciones de fin de juego.

A. Pseudocódigo

```
Cargar la lista de movimientos a realizar
Cargar el mapa
Calcular las posiciones iniciales de todos los personajes
Loop Movimiento Todos
    Ver si se mueve Pacman, Blinky, Pinky, Inky o Clyde
    Si se mueve pacman ir al Loop Movimiento de Pacman
    Si se mueve un fantasma ir al Loop Movimiento de Pacman

Loop Movimiento de Pacman
    Cargar el movimiento del arreglo
    Identificar el movimiento
    Realizar el movimiento
    Actualizar variable de control
    Actualizar ndice del movimiento
    Regresar al Loop Movimiento Todos

Movimiento Pacman...
    Simular movimiento
    Identificar efecto espejo
    Si efecto espejo: mover 15 celdas
    Si no: simular y verificar choque
    Si choca con pared: cancelar movimiento
    Si no choca:
        Verificar esteroide y aplicar efecto
        Verificar choque con fantasma
        Si tiene esteroide: reiniciar fantasma y +10
        Si no: reiniciar Pacman y -1 vida
        Verificar moneda: +1 score
        Actualizar variables de control

Loop Fantasmas...
    Identificar fantasma
    Guardar informac i n
    Ir a Ver Movimiento Fantasma

Ver Movimiento Fantasma
    Cargar movimiento del arreglo
    Identificar direcc i n
    Mover Fantasma

Mover Fantasma...
    Simular movimiento
    Verificar colisiones
    Regresar a Ver Movimiento Fantasma
```

III. RESULTADOS

Durante la ejecución, Pac-Man y los fantasmas se mueven de forma aleatoria siguiendo los valores cargados en memoria. Se puede observar la actualización de sus posiciones, el incremento de *score* al comer monedas o esteroides y su decremento al ser comido por fantasmas, lo que se refleja en el *output* y en el cambio de posición a la inicial.

Se detectan correctamente las colisiones de fantasmas con Pac-Man, y se reinician a su posición original si son comidos

bajo el efecto del esteroide. El juego termina al perder las tres vidas.

IV. CONCLUSIONES

Este proyecto permitió comprender a profundidad el funcionamiento de una arquitectura de computadora simple y sus limitaciones. Se exploró el uso de memoria, instrucciones de control de flujo, operaciones aritméticas y manejo de arreglos. La implementación de una lógica compleja como un juego en MARIE es desafiante pero enriquecedora.