



BREAKOUT

PROGRAMACIÓN DECLARATIVA
4º CURSO PRIMER CUATRIMESTRE
GRADO EN INGENIERÍA INFORMÁTICA UCO
CURSO ACADÉMICO 2019 – 2020

ÁNGEL HEREDIA PÉREZ

CÓRDOBA 07/12/19

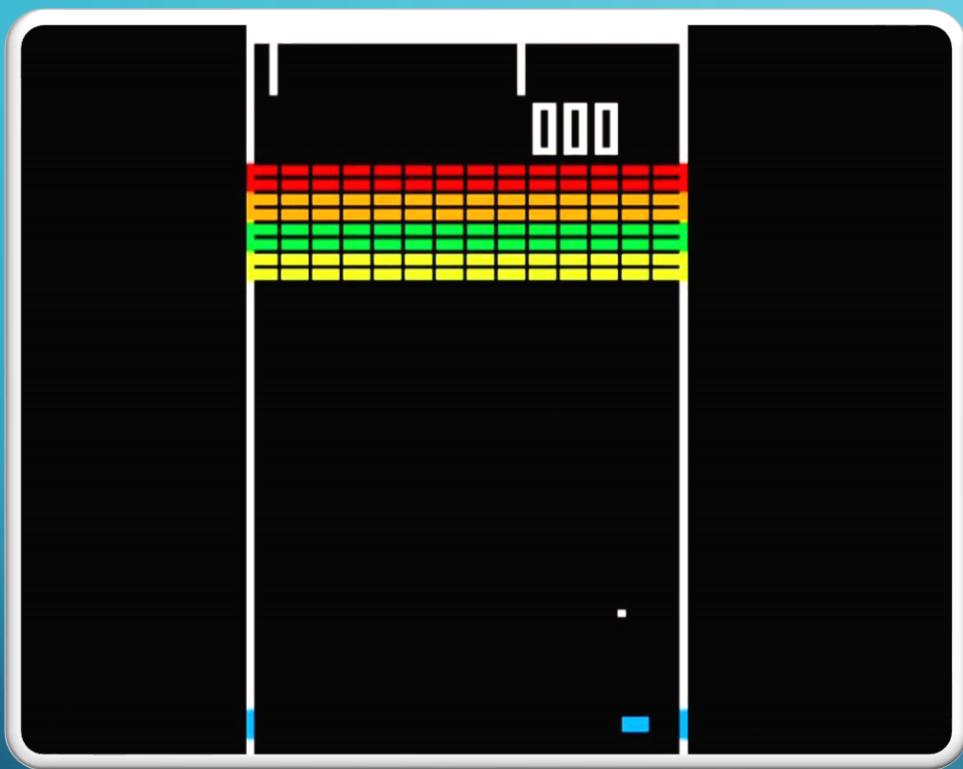
ÍNDICE

- [Introducción](#)
- [Fundamentos teóricos](#)
 - [universe.rkt teachpack](#)
 - [Función *big-bang*. Parámetros](#)
- [Desarrollo del proyecto](#)
 - [Estructuras de datos](#)
 - [Estructura world](#)
 - [Condiciones iniciales \(inicialización world\)](#)
 - [Evolución del juego](#)
 - [Renderización de escenas](#)
- [Resultados](#)
 - [Rebote y destrucción de bloque](#)
 - [Perder vida](#)
 - [Ganar y perder el juego](#)
- [Conclusiones](#)
- [Referencias](#)

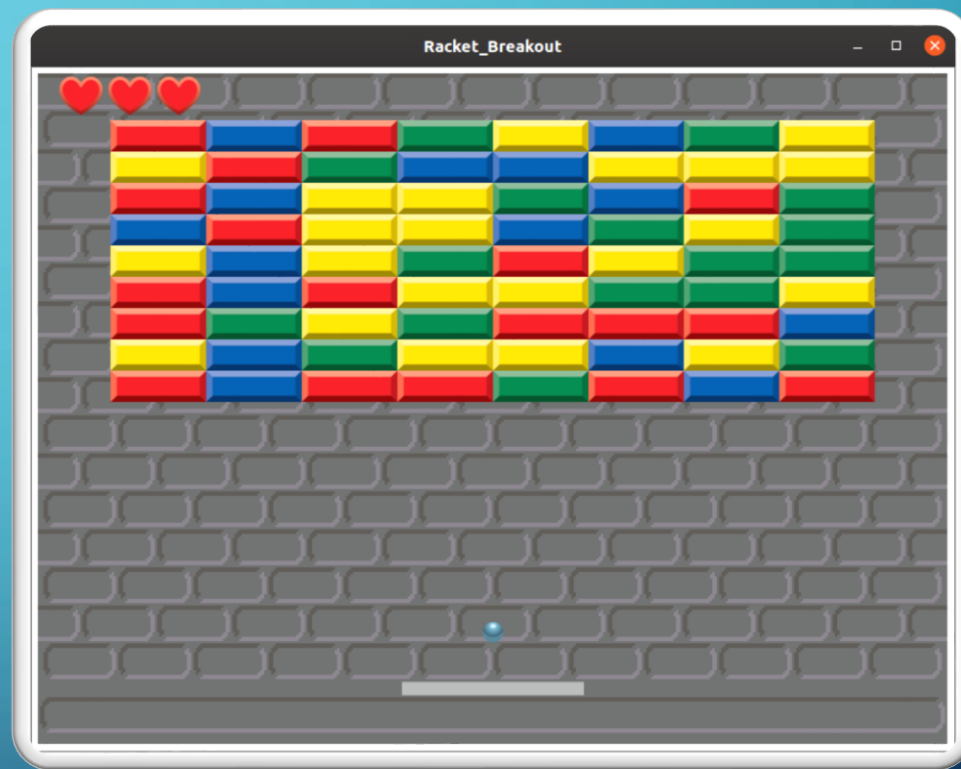
INTRODUCCIÓN



- **Breakout** es un videojuego arcade desarrollado por **Atari**.
- Fue lanzado al mercado el 13 de mayo de 1976 y fue creado por **Nolan Bushnell** y **Steve Bristow**
- Influenciado por el videojuego de 1972 **Pong**.



Breakout de Atari 2600. Año 1978



Breakout de DrRacket. Año 2019

FUNDAMENTOS TEÓRICOS



Índice

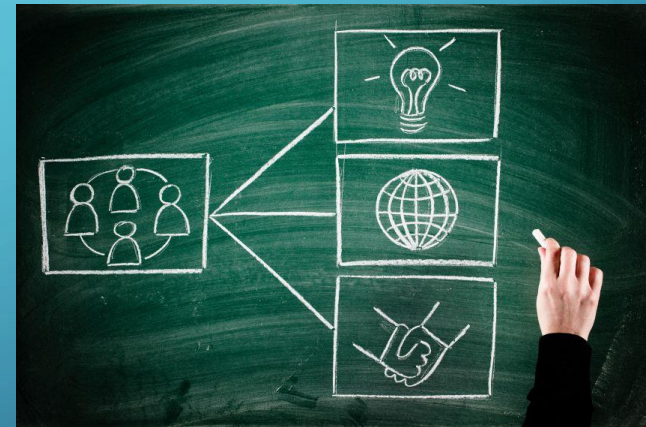
UNIVERSE.RKT TEACHPACK

- Funcionalidad necesaria para la creación de programas gráficos interactivos (*world programs*).
- *Los world programs pueden formar parte de un “Universo” e interactuar entre si.*
- Cada *world program* es gestionado por la función *big-bang*.

FUNCIÓN *BIG-BANG*. PARÁMETROS




- (on-tick tick-expr): se ejecuta 28 veces por segundo.
- (to-draw draw-expr): se ocupa de **dibujar** la escena.
- (on-key key-expr): maneja la **pulsación** de teclas.
- (on-release release-expr): maneja el evento de **soltar** una Tecla.
- (name str): **nombre** de la ventana.
- (stop-when stop-expr last-scene-exp): determina cuando el juego se debe **terminar la ejecución y dibuja la última escena**.

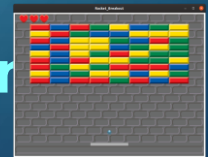
DESARROLLO DEL PROYECTO



Índice

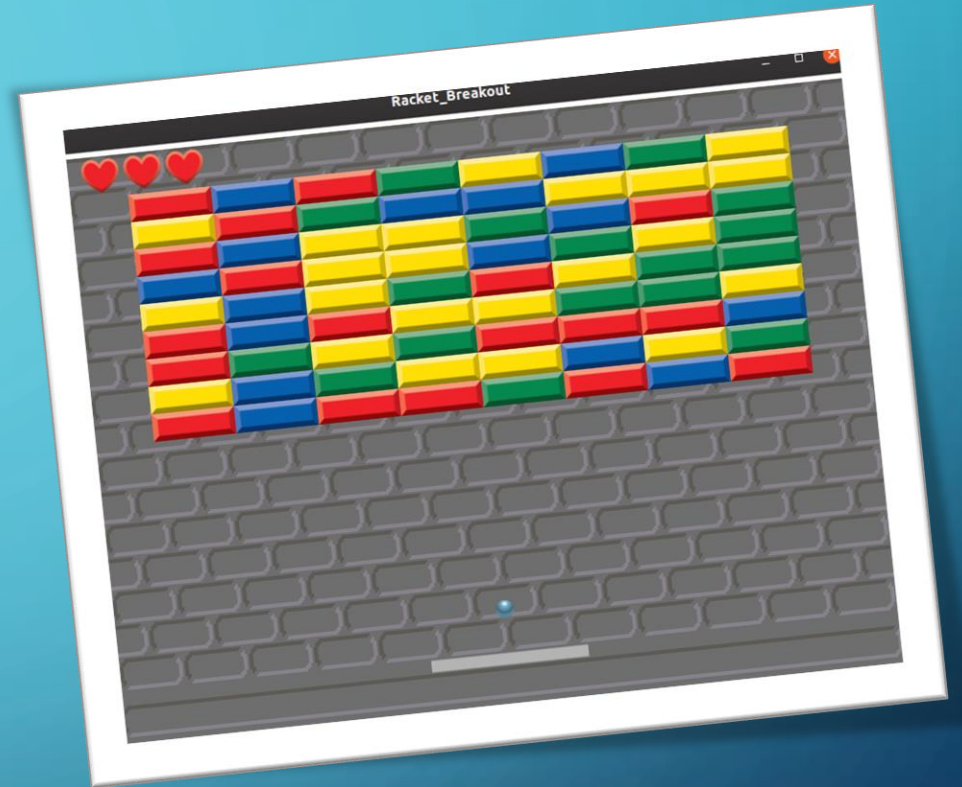
ESTRUCTURAS DE DATOS

- Ball: representa la **bola** 
- Bar: representa la **barra** que el jugador controla 
- Block: representa un **bloque** de la pantalla 
- World: representa el **juego** y **almacena toda la información**



ESTRUCTURA WORLD

- Estructura Ball
- Estructura Bar
- Lista de estructuras Block
- Número de vidas restantes del jugador
- Estado de pulsación de la flecha derecha
- Estado de pulsación de la flecha izquierda



CONDICIONES INICIALES (INICIALIZACIÓN WORLD)

- Objeto Ball en posición inicial y velocidad 0.
- Objeto Bar en posición inicial.
- Lista de objetos Block, generada por la función (make-blocks).
- Número de vidas del jugador = 3
- Estado de pulsación tecla derecha = #f
- Estado de pulsación tecla izquierda = #f

EVOLUCIÓN DEL JUEGO

- Mover barra: en función de las **teclas pulsadas** la barra se **desplaza a derecha o izquierda**.
- Actualizar vidas jugador: en caso de que la **bola caiga** se **resta 1 vida** al jugador.
- Mover bola: la bola **se desplaza con una dirección**, que **se recalcula si colisiona** con algo y **una velocidad**, que **aumenta con cada bote** hasta un máximo.
- Posición inicial: si la bola **cae**, la bola se coloca en su **posición de inicio**.
- Actualizar bloques: si un bloque recibe una **colisión** se **destruye**.
- Finalizar juego: si **no** quedan **vidas** o **bloques** el juego **termina**.

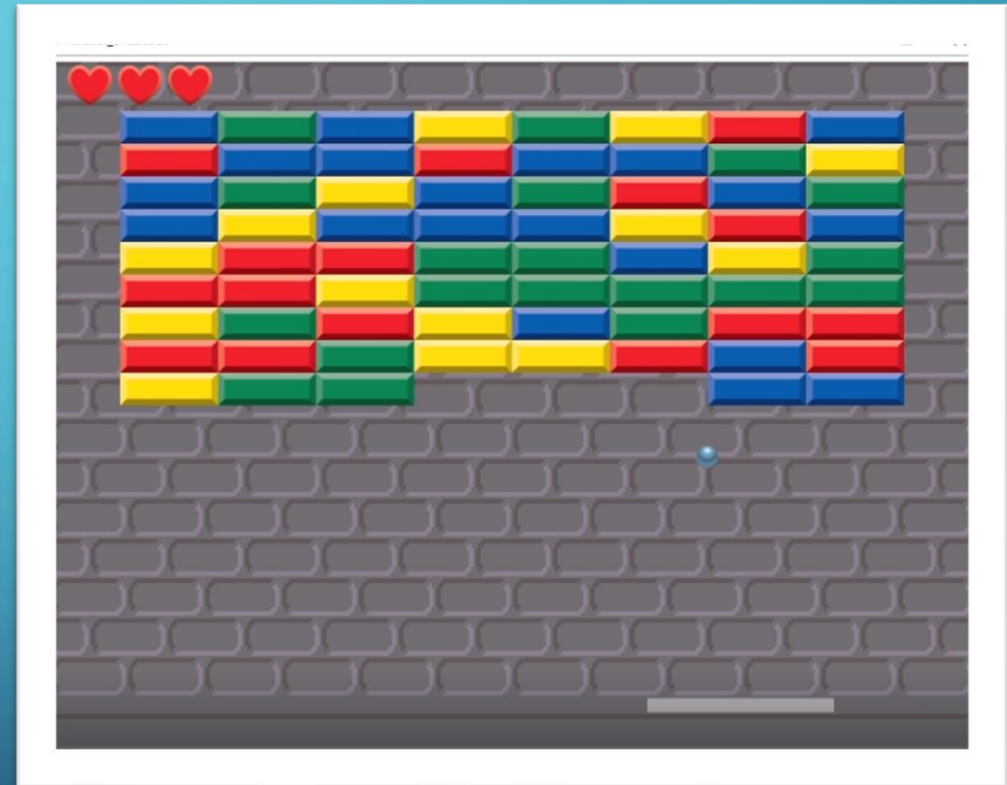
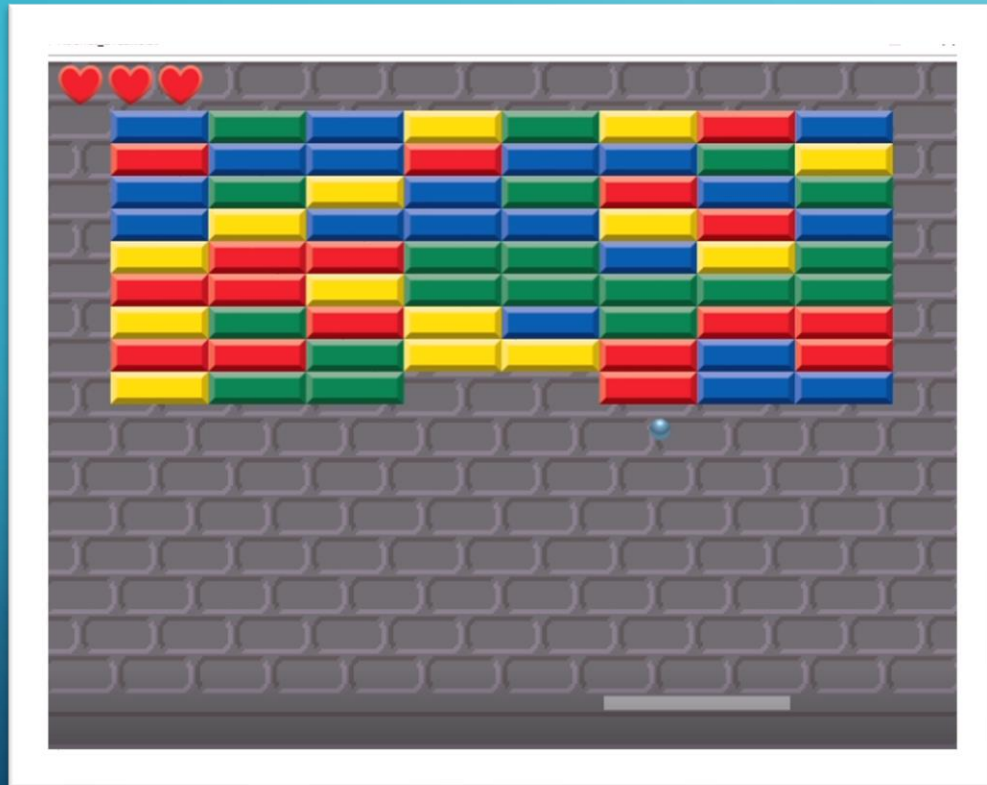
RENDERIZACIÓN DE ESCENAS

- Se usa la librería **2htdp/image**
- Se utilizó una función de dibujo **individual** para el objeto Ball y el objeto Bar.
- Se utiliza una función de dibujo **individual** para los objetos Block de forma **recursiva**, recorriendo la lista de los mismos.
- Cuando el juego termina se dibuja la pantalla de victoria o derrota en función de la **condición de finalización**.

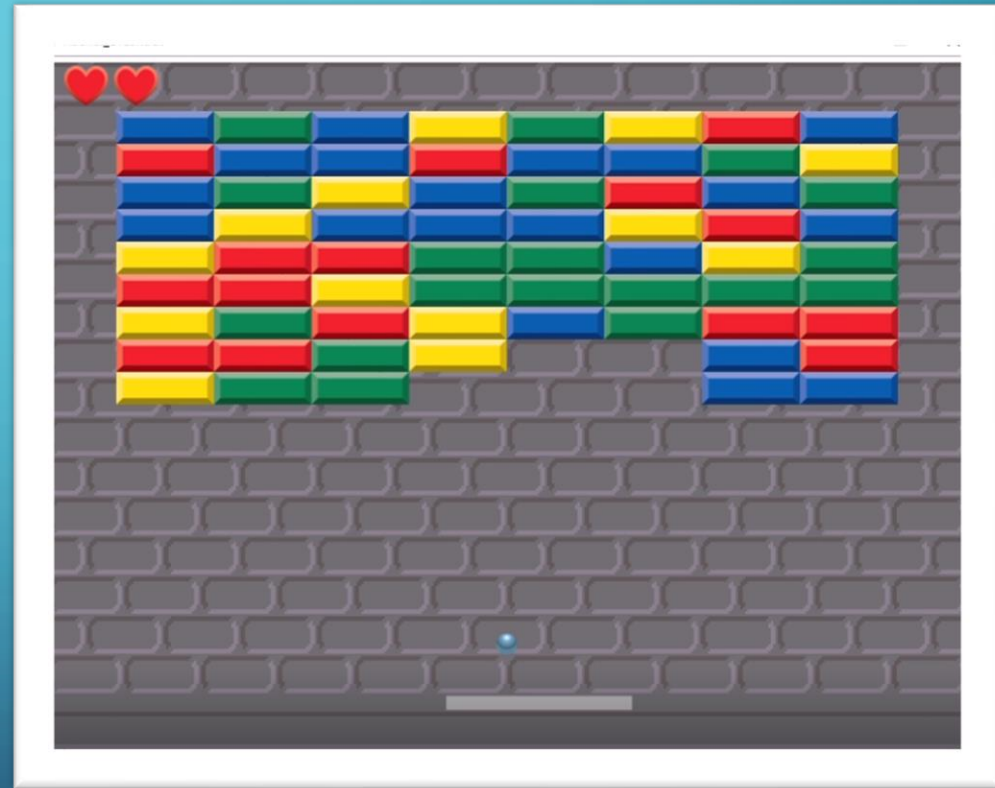
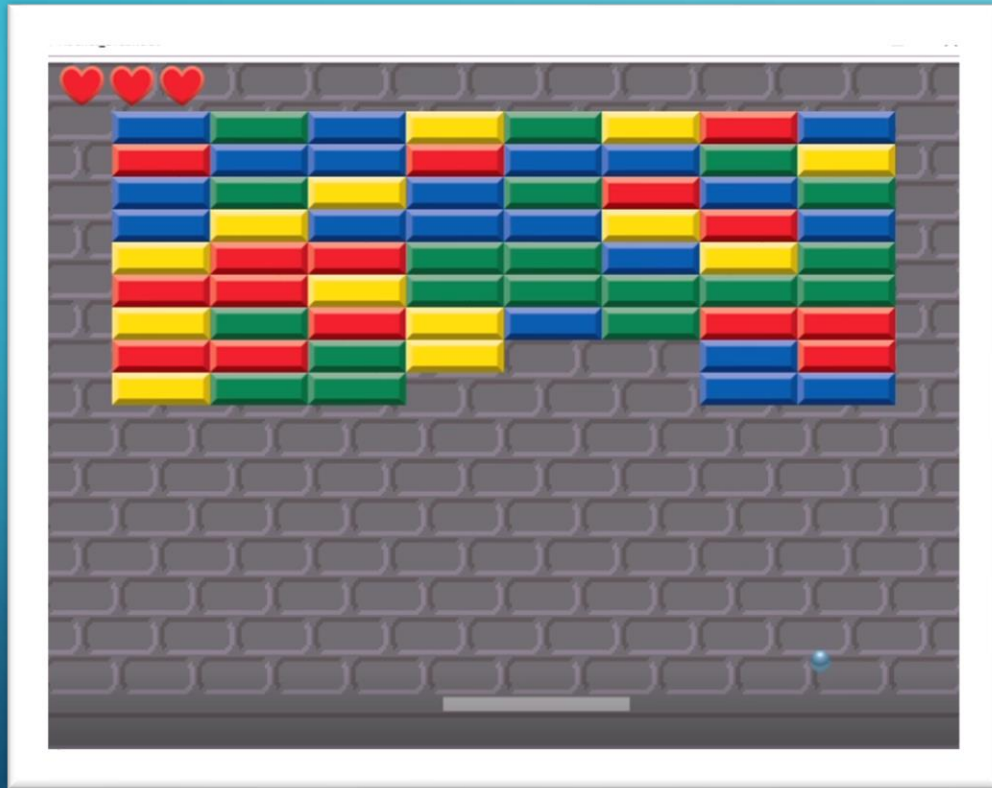
RESULTADOS

Índice

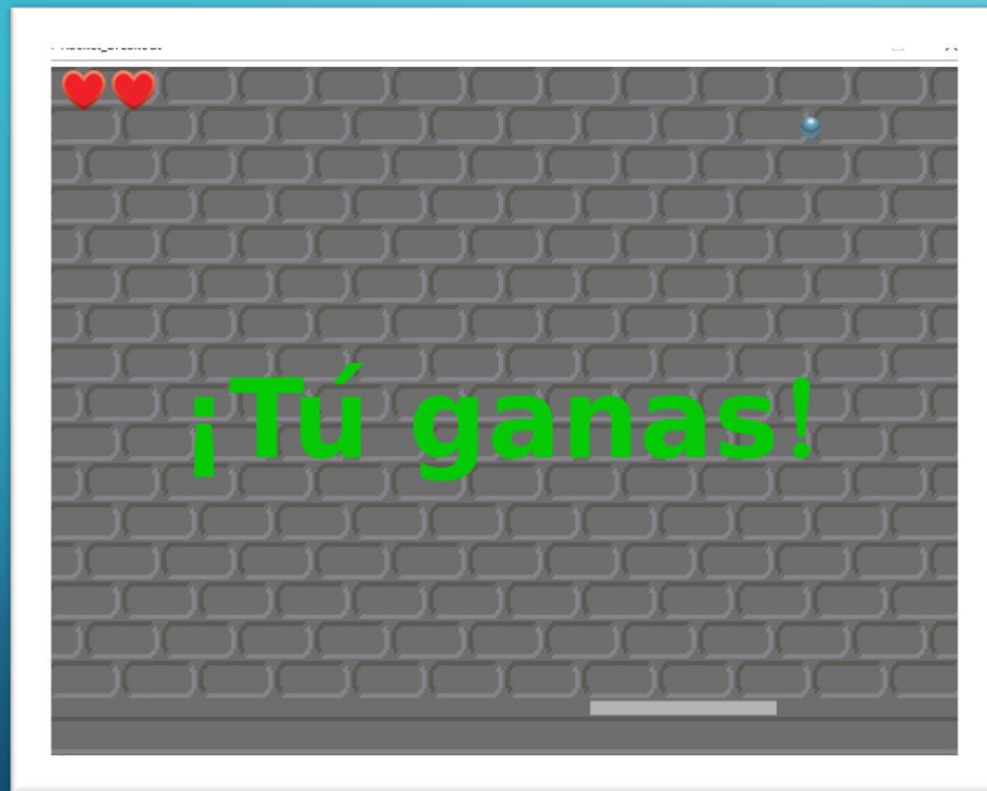
REBOTE Y DESTRUCCIÓN DE BLOQUE



PERDER VIDA



GANAR Y PERDER EL JUEGO



CONCLUSIONES

- Scheme es un lenguaje **flexible** que permite la realización de incluso videojuegos.
- La optimización no es la mejor.
- Dr Racket parece gestionar el sonido de forma distinta dependiendo del SO, provocando ralentizaciones en Windows.
- La **recursividad** es la protagonista en vez de los bucles.
- La inclusión de más niveles mejoraría la experiencia.

REFERENCIAS

- Atari Breakout. [https://es.wikipedia.org/wiki/Breakout_\(videojuego\)](https://es.wikipedia.org/wiki/Breakout_(videojuego))
- Defining Structure Types: struct. <https://docs.racket-lang.org/reference/define-struct.html>
- Worlds and the Universe: “universe.rkt”. <https://docs.racket-lang.org/teachpack/2htdpuniverse.html>
- Images: “image.rkt”. <https://docs.racket-lang.org/teachpack/2htdpimage.html>