



# *Space invaders*

PROGRAMACIÓN DECLARATIVA

4º Curso primer cuatrimestre

Grado en Ingeniería Informática UCO

Curso académico 2017 – 2018



Cristian Ruz Najar  
Córdoba 11/12/17

# INDICE

## 1

- Introducción
  - Comparativa visual
- ¿Que es universe.rkt teachpack?
  - Funcionamiento básico
- Desarrollo de space invader
  - Elección de estructuras de datos
  - Modelado de datos necesarios para space invaders
  - Imágenes y dibujado
  - Interacciones
  - Curiosidades

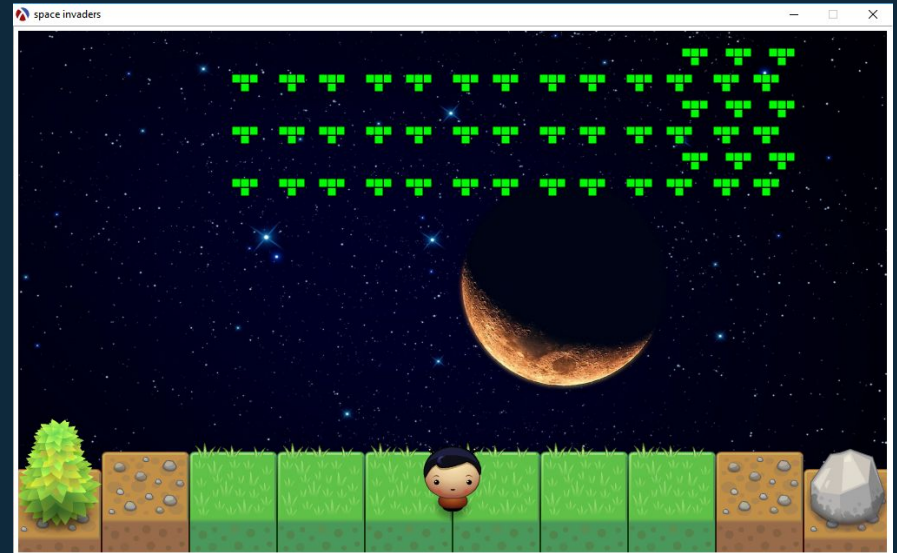
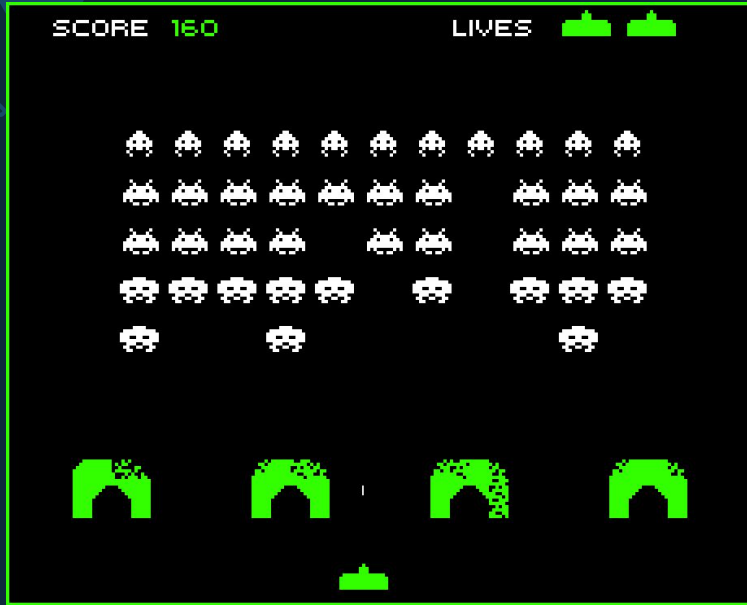
# SPACE INVADERS

## Introducción

Space Invaders es un videojuego de arcade diseñado por Toshihiro Nishikado y lanzado al mercado en 1978. En un principio fue fabricado y vendido por Taito Co. en Japón, para posteriormente ser licenciado para producción y distribución en Estados Unidos por Midway Games, división de Bally Technologies. Space Invaders es uno de los primeros juegos shoot 'em up. Es uno de los videojuegos más importantes de la historia. Su objetivo es eliminar oleadas de alienígenas con un cañón láser y obtener la mayor cantidad de puntos posible. Para el diseño del juego, Nishikado se inspiró en Breakout, La guerra de los mundos y Star Wars.

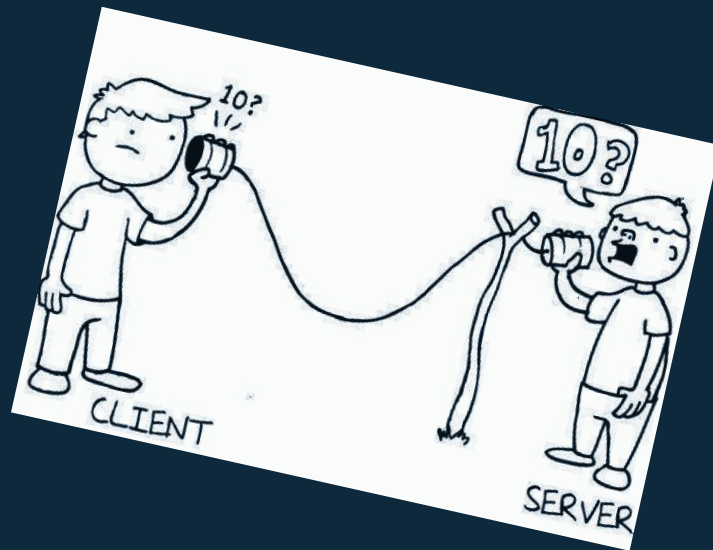
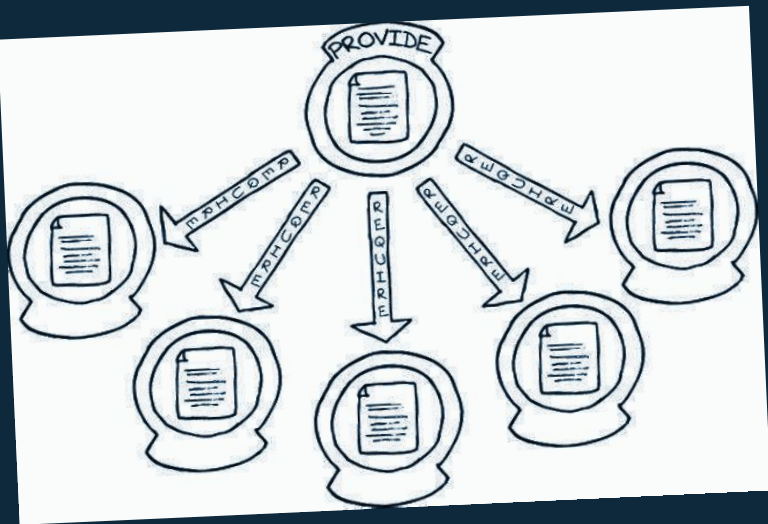


# Comparativa visual

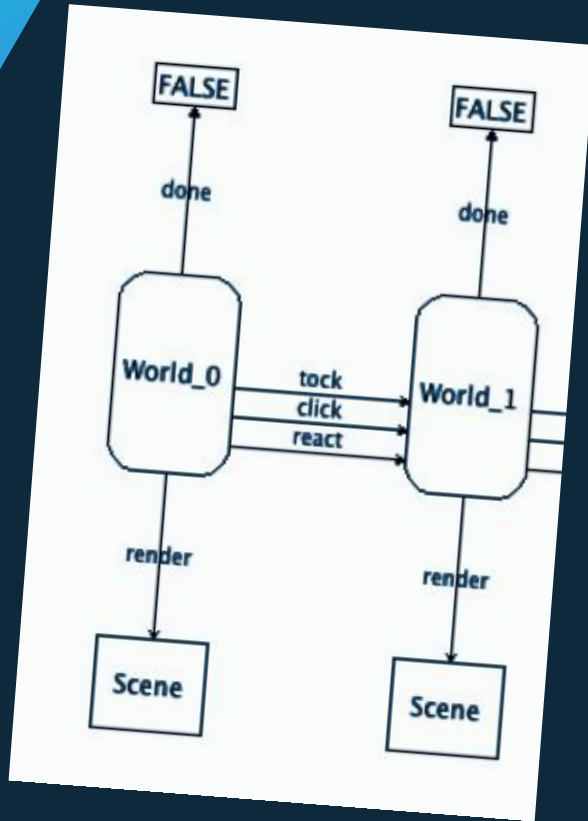


# ¿Que es universe.rkt teachpack?

Universe.rkt implementa y proporciona las funciones necesarias para crear programas gráficos interactivos , estos programas son llamados world programs y cuentan con la capacidad de comunicarse entre varios de ellos formando los denominados “universos”, que no es más que un grupo de world programs que se comunican entre ellos



# Funcionamiento básico



La forma en la que universe.rtk funciona es muy sencilla, la función bigbang recibe una serie de datos como podrían ser la posición de cada objeto en pantalla y en cada “tick” la función actualiza la información con ayuda de las funciones que tu diseñes para interactuar con el programa y redibuja toda la escena.

# Cláusulas de la función bigbang

Dentro de la función bigbang encontramos una gran variedad de cláusulas , entre las principales se encuentran:

(on-tick tick-expr): función que se ejecuta en cada “tick”

(on-key key-expr): función que se ejecuta cuando accionas una tecla

(to-draw draw-expr): función que dibuja la escena en cada “tick”

(stop-when stop-expr last-scene-expr) : función que comprueba si el programa debe acabar y dibuja la última escena(opcional)



# Control básico de teclas

Este seria un controlador estándar para los controles del programa ,  
cada tecla que definamos podrá llamar a una función .

```
(define (change w a-key)
```

```
(cond
```

```
  [(key=? a-key "left") (world-go w -DELTA)]
```

```
  [(key=? a-key "right") (world-go w +DELTA)]
```

```
  [(= (string-length a-key) 1) w] ; order-free checking
```

```
  [(key=? a-key "up") (world-go w -DELTA)]
```

```
  [(key=? a-key "down") (world-go w +DELTA)]
```

```
  [else w]))
```







# Desarrollo de space invaders

GAME  
OVER





# Elección de estructura de datos

La primera pregunta que surgió al comenzar el desarrollo era ¿como voy a almacenar la información? ¿vectores? ¿listas? , ¿como podría facilitar la representación? Y entonces apareció la respuesta, estructuras , las cuales tienen una fácil gestión y además ayudan a la fácil comprensión del programa .






# Modelado de datos necesario para space invaders

Después de elegir la manera de representar los datos , es igual de importante tener claro qué datos vas a representar .

En el caso de space invaders cuento con 4 estructuras :

- Player : representa al jugador
  - Ufo : representa una nave enemiga
  - Bullet : representa una bala disparada por el jugador
  - World : representa todos los datos del juego y está compuesta por un player , una lista de ufos y una lista de bullets
- 



# Imágenes y dibujado

Para el dibujado en pantalla he hecho uso de la librería 2htdp/planetcute y 2htdp/image .

Para el dibujado de los ufos y las bullets sigo el mismo procedimiento, una función de dibujado individual y otra que recorre la lista aplicando esa función a cada uno de los objetos .

Los ufos y bullets están dibujados a partir de “bloques” que tienen la intención de representar pixeles





# Interacciones

Para las interacciones del juego se han diseñado una amplia gama de funciones para cubrir todos los aspectos :

- Colisiones
- Movimiento de los ufos hacia el lado
- Cambio de dirección de los ufo
- Movimiento hacia abajo de los ufo
- Disparos
- Movimientos del jugador
- Ufos llegando al limite
- Todos los ufos eliminados



# Curiosidades



En el space invaders original a medida que ibas eliminando enemigos la velocidad de los que quedaban iba aumentando por lo que cada vez era más difícil dispararles, esto se producía porque la maquina arcade tenía que renderizar menos enemigos por lo que realizaba estos ciclos más rápido en ningún momento fue voluntad del desarrollador.



En esta versión en scheme curiosamente ocurre lo mismo aumentando su dificultad cuando quedan menos enemigos





# Bibliografía

Realm of Racket: Learn to Program, One Game at a Time!  
<https://docs.racket-lang.org/>





# Gracias!

## ¿Preguntas?

◇ Cristian Ruz Najar

