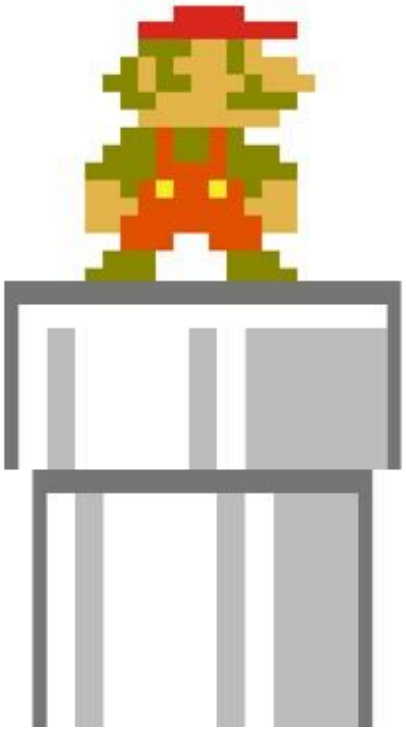


PROYECTO:

¡Run Racket Run! Animaciones y Juegos en Racket

Programación I - **2016**

Ciencias de la Computación

Universidad Nacional de Rosario – Facultad de ciencias exactas, ingeniería y agrimensura.

Simón Martín Acosta

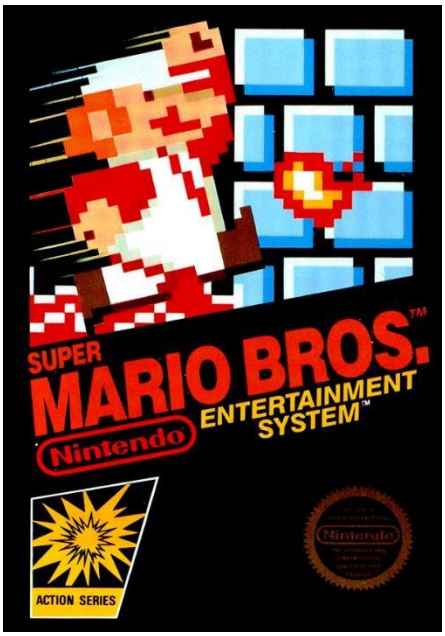
D.N.I.: 38.815.437

Mail: sma2347@gmail.com

El **"Proyecto: Run Racket Run! Animaciones y Juegos en Racket"**, de la catedra Programación I de la carrera Ciencias de la Computación de la *U.N.R. – F.C.E.I.A.*, tiene la finalidad de realizar un programa interactivo utilizando todos los conocimientos aprendidos sobre DrRacket.

Se optó por realizar una pequeña recreación similar del conocido video-juego de plataformas, *Super Mario Bros* de *Nintendo* diseñado por **Shigeru Miyamoto** y lanzado el *13 de septiembre de 1985*. La elección de dicho tema, surgió de la presentación del proyecto anterior, en donde se hizo una imagen del mismo juego. El simple objetivo del juego será llegar hasta la princesa antes de que el tiempo se agote, para lograrlo debemos avanzar por los diferentes niveles y utilizar los distintos tubos, los mismos podrían ahorrarnos el tiempo de llegada o retrasarnos.

Información mencionada en el Proyecto 01: Racketeando! Dibujos y Arte en Racket:



Shigeru Miyamoto:

Es un diseñador y productor de video-juegos japonés que trabaja para Nintendo desde el año 1977.

--->

Portada del video-juego *Super Mario Bros,* 1985.

Programando en DrRacket:

El desarrollo del programa comenzó con la creación de los escenarios, los mismos se construyeron con una temática similar a los vistos en el juego original, tomando como referencias los niveles 1-4 y 8-4 del *Super Mario Bros*. A medida que se fue avanzando con el proyecto se fueron creando las diferentes constantes que se consideraron necesarias, entre ellas colores, imágenes, números.

Construcción de los escenarios:

Para construir los escenarios se creó y utilizo una variable llamada ***repetir/img*** cuya utilidad es repetir N veces una misma imagen. Para su funcionamiento en su entrada necesita un operador, la imagen que queremos repetir y un número que establece la cantidad de veces que la imagen se repetirá. Los operadores utilizados en el programa fueron *beside* y *above*, los mismos se utilizaron para repetir imágenes de forma horizontal (usando *beside*) y para repetir imágenes de forma vertical (usando *above*). Además de utilizar ***repetir/img*** para construir los escenarios se reutilizo código del proyecto pasado, el característico **tubo** de *Super Mario Bros*, y utilizando la función *place-image* se logró posicionar cada construcción en su respectivo lugar.

Para finalizar se crearon los carteles de fin de juego ganaste (**Ganaste**) y fin de juego perdiste. (**Fin-P**)

Construcción del programa interactivo:

El programa en su estado contiene una estructura, ***ST***, que contiene seis campos. A través de ella se logran obtener los datos necesarios para el correcto funcionamiento del programa. Cinco de los seis campos de la estructura están constituidos por números, estos campos se utilizan para establecer condiciones de animación de movimiento, determinar la ubicación/posición del ancho y alto del personaje, el nivel en donde se encuentra y el reloj utilizado por el cronometro, el campo faltante por mencionar es una imagen del personaje.

Diseño de estructura:

- ***NMB:*** Unidad numérica, usado para activar funciones relacionadas a animaciones.
- ***NMB/ANCHO:*** Posición en eje X del personaje.
- ***NMB/ALTO:*** Posición en eje Y del personaje.
- ***IZ/DE:*** Imagen del personaje durante el movimiento o acción.
- ***Level:*** Indica el número de nivel en funcionamiento.
- ***Tick:*** Contador, usado en el cronometro.

Manejadores de eventos:

- ***Manejadores de eventos de pantalla:***

El manejador de eventos de pantalla, se encarga de que nosotros podamos visualizar, en el código llamado ***pantalla***. Este manejador utiliza una estructura **ST** y de ella extrae los datos de posicionamiento del personaje, la imagen del personaje, el dato de contador para mostrar en el centro superior de la imagen y el nivel, este último es el que determina el escenario a mostrar por el manejador.

- ***Manejadores de eventos de teclado:***

Encargado de los eventos producidos al presionar una tecla del teclado, en el código llamado ***key?***. Este manejador se encarga de las funciones de movimiento del personaje y de reinicio del juego. A su entrada ingresa una estructura **ST**, donde luego de presionar alguna de las teclas configuradas se encargará de activar las funciones respectiva de la tecla presionada. **-** ***Manejadores de eventos de reloj:***

Establece el tiempo restante de juego, en caso de llegar a cero... Perdiste. También se encarga de activar el cambio de nivel cuando se active la condición debida.

- ***Manejadores de eventos de fin del programa.***

En caso de que una de las condiciones establecidas en este manejador se cumpla y como resultado entregue un, #t, un valor verdadero, el programa finalizara automáticamente.

Funciones de posicionamiento:

Las funciones de posicionamiento se encargan de la respectiva ubicación del personaje en determinados sector o puntos del plano. En el código se encuentran tres funciones encargadas de esto:

- ***cond/ancho+***, encargada de las condiciones del ancho cuando el personaje avanza hacia el lado derecho. Esta función es la encargada de identificar si en frente del personaje existe un elemento que nos bloquee el paso o no. También es la encargada del movimiento hacia la derecha.

- ***cond/ancho-***, encargada de las condiciones del ancho cuando el personaje avanza hacia el lado izquierdo. Al igual que la función mencionada recientemente, esta función se encarga de identificar si en frente existe un elemento que nos bloquee el paso o no. También es la encargada del movimiento hacia la izquierda

- ***Cond/alto***, su función es la de determinar la altura en donde debe estar posicionado el personaje. *Ejemplo: al bajar de uno de los tubos o escaleras.*

Las funciones mostradas anteriormente constituyen la rama principal del posicionamiento dentro del escenario, pero además de las mencionadas existen dos funciones exclusivamente encargada para la acción de saltar. Las mismas a su entrada reciben una estructura **ST**, la cual se modifica a su salida agregando o quitando delta unidades a la posición X e Y para ir hacia la derecha o izquierda durante un lapso de tiempo. Al terminar ese lapso se activa el condicional de altura para determinar la posición del personaje, y el proceso se vuelve a repetir de ser requerido.

- ***Salto/d,*** se encarga del salto hacia la derecha.
- ***Salto/i,*** se encarga del salto hacia la izquierda.

Animaciones de movimiento:

Las animaciones mientras el personaje avanza son determinadas por el campo **NMB** de la estructura **ST**, en donde al número del campo se le agrega 1 y usando un condicional se determina la imagen asociada al valor obtenido, en caso de que el número no coincida con las condiciones se activa una recursión pero con el valor de **NMB** reiniciado, y dejándolo en condiciones nuevamente listo para repetir el proceso.

Recursos:

Para este proyecto de recreación de Super Mario Bros de Nintendo, fueron requeridos datos externos. Dichos datos son exclusivamente imágenes importadas al código de DrRacket, las mismas se descargaron de la página *"The Spriters Resource"* (www.spriters-resource.com)