INSTITUTO TECNOLÓGICO DE COSTA RICA

INGENIERÍA EN COMPUTADORES

TALLER DE PROGRAMACIÓN CE-1102

**Proyecto #2: Crazy Donkey Kong**

Profesor:

Milton Villegas Lemus

Alumnos:

Gabriel Abarca Aguilar. 2017110442

I Semestre, 2017

Fecha de entrega:

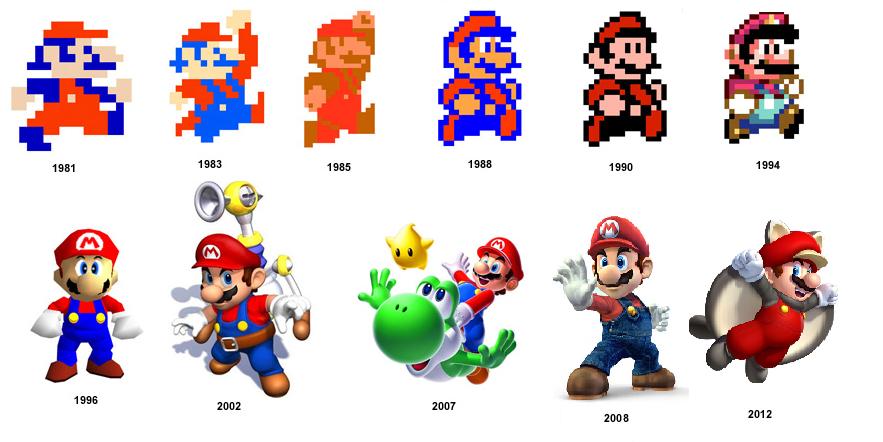
5 de mayo, 2017

# Introducción

En el siguiente proyecto se puede apreciar una versión sin fines lucrativos de Donkey Kong en lenguaje Python, el juego trata sobre un personaje, Mario Bros. El cual tiene que rescatar a Pauline de Donkey Kong, para esto tiene que pasar un mapa de obstáculos con barriles, los cuales tiene que saltar para cumplir su objetivo. Este juego fue sacado por primera vez en Arcade en Japón por Nintendo en 1981. Este juego pertenece al género de plataformas, y fue un éxito en Japón y en Estados Unidos. Durante la producción de este juego se aprecia el desarrollo de la capacidad de un programador al ejercer creatividad sobre el juego. Hoy en día se poseen herramientas que en esa época no se usaban, y esto beneficia al programador moderno a crear una réplica de este entretenido juego, el cual se mantiene como uno de los favoritos en la historia de los videojuegos Retro.

Al hablar sobre un videojuego retro, se habla de una época vieja. Los inicios de estos productos que empezaron a vender por montones, se puede exponer sobre como las versiones de Mario han madurado y han crecido junto a sus fanáticos y jugadores, no solo con Mario, si no con videojuegos que se juegan hoy en día como Fifa, PES, Zelda, etc. Como se aprecia en la figura uno, estos fueron evolucionando conforme la tecnología y sus capacidades fueron creciendo.

En el juego principal se visualizan 4 escenarios que avanzan conforme la historia del juego, donde Mario tiene que subir de 25m a 100m, la cima de la torre, para liberar a Pauline de Donkey Kong.

****

**Figura 1.**

# Conclusiones

Mario es un personaje icónico en el mundo de los videojuegos, y recrear una de sus versiones se aprecia el entretenimiento que puede ejercer la creatividad de alguien al programar un videojuego como este.

La inteligencia artificial puede iniciar desde la simpleza de una orden, por medio de la computadora, esto se experimentó en la creación de barriles, donde se le ponían ciertos ejes para que pararan los mismos.

Se puede visualizar como los medios modernos facilitan al ser humano durante la programación, al comparar códigos de otros lenguajes diferentes. Los cuales sirvieron de ayuda para visualizar diferentes ideas y transmitir su lógica al juego.

# Recomendaciones

* Averiguar sobre la interfaz usada al programar antes de iniciar un proyecto, esto debido a que el gasto de tiempo para aprender a usar una interfaz es bastante.
* Aprender sobre coordenadas cartesianas para tener una visualización sobre 2 dimensiones a la hora de programar un juego de plataformas como este.
* Aprender sobre la historia del juego, para plasmar la idea de este al recrear una versión sobre este.
* A la hora del uso van a ver unos cuantos errores, por eso se recomienda pedir ayuda a algún usuario que no tenga idea de cómo funciona el código para que pruebe el programa y pueda sacar los errores de este.
* Buscar sobre videotutoriales o lecturas referentes a algún juego, debido a que las ideas de estos pueden ayudarnos a visualizar ideas para usar alguna función del juego o incluso crear nuestra lógica en base a esta.

# Análisis de Resultados

## Errores

* Se visualizan errores en el juego como al borrar el nombre, se acumulan las letras. Esto se da debido a que no se borra el “string” puesto en la imagen, pero si se borra la variable y se vuelve a escribir sobre ella en la imagen.
* Al saltar, los booleanos están definidos de una manera, sin embargo, al llegar a ciertos ejes Mario con la “w” salta y sube escaleras, pero al mantener estripado la tecla, sube las escaleras, pero a llegar al punto máximo de este, pega un brinco.
* Se visualizaba un bug que Mario saltaba hasta la visualización del “Score” y se quedaba pegado, esto se solucionó reduciendo el salto de Mario.
* Los Barriles tienen radio de rectángulo y no de círculo, esto hace problemas con las colisiones, debido a que el rango era bastante grande.

## Recursividad

Se pudo apreciar como al usar “while” y “for” en la interfaz, se puede implementar de manera más cómoda que con recursividad, sin embargo, algunas ideas como los “Highscores” surgieron por recursividad y no por estos métodos.

## Errores Corregidos

### Saltos:

Mario saltaba al inicio de una plataforma, esto se corrigió creando una variable que guarda el valor al inicio del salto y lo compara con el valor de y original, cuando llegue a 70Px, de diferencia lo cambia a bajada la cual va a terminar hasta que toque los ejes de un plataforma.

### Puntajes:

Algunas veces los puntajes no sumaban, esto se arregló poniendo un rango para que Mario pase encima de los barriles.

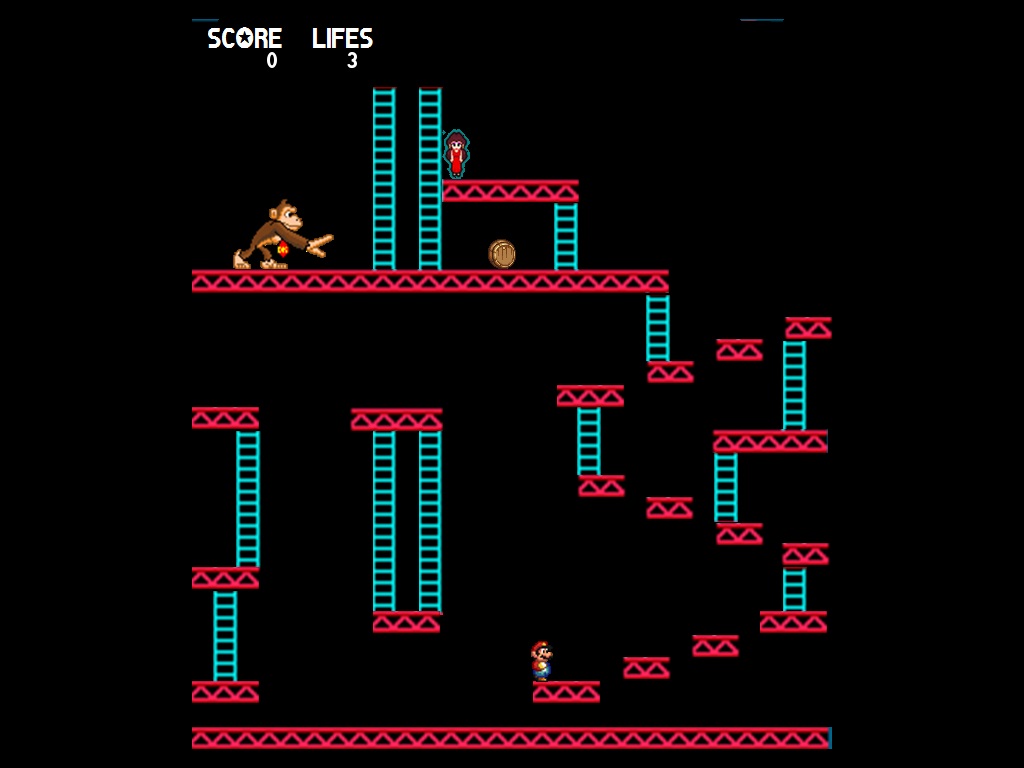
# Anexos



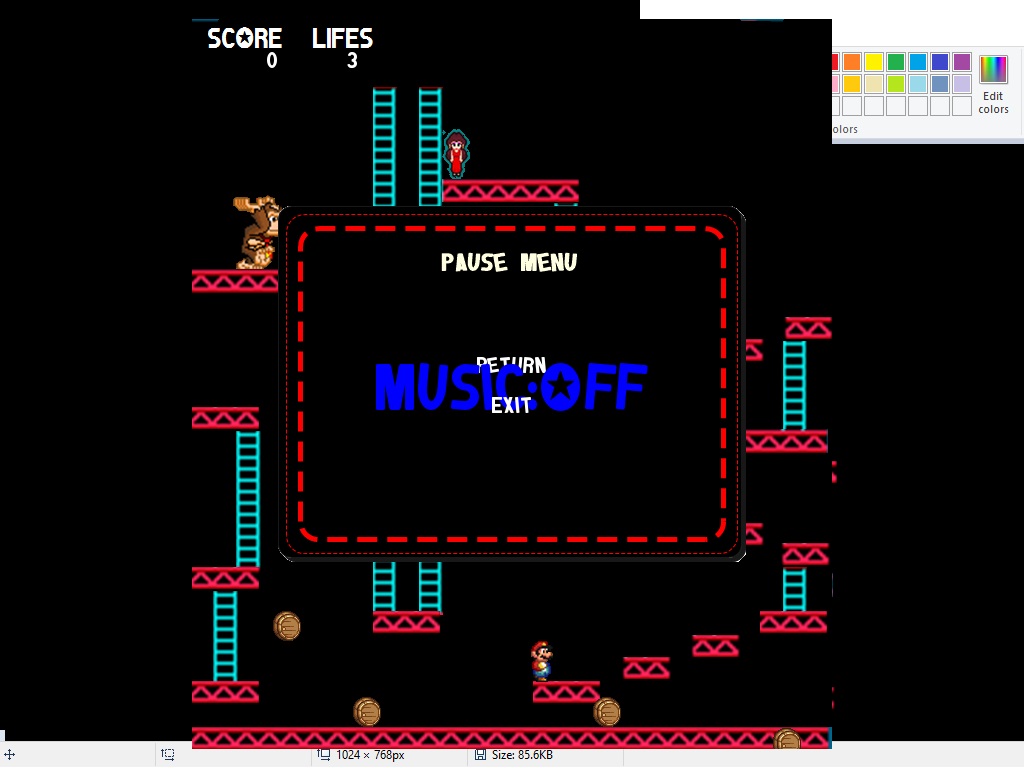
Pantalla de Inicio que lleva a otras pantallas

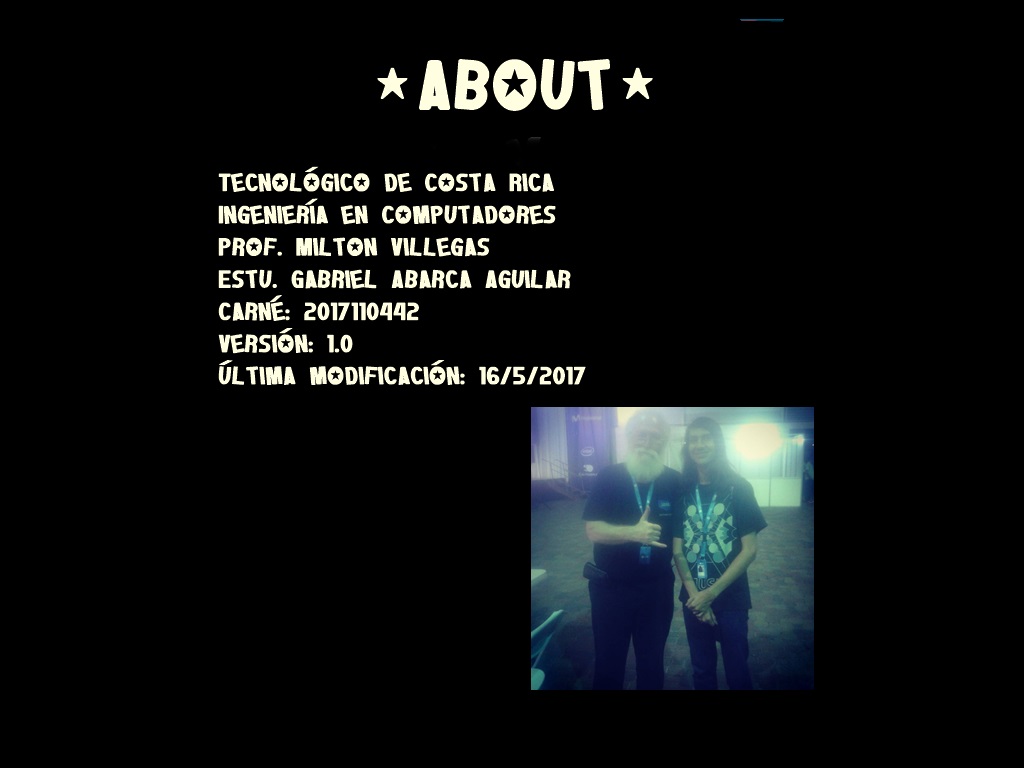
Pantalla de Puntajes más altos

Los puntajes iguales a 0 no aparecen. En caso de ser mayor va a aparecer

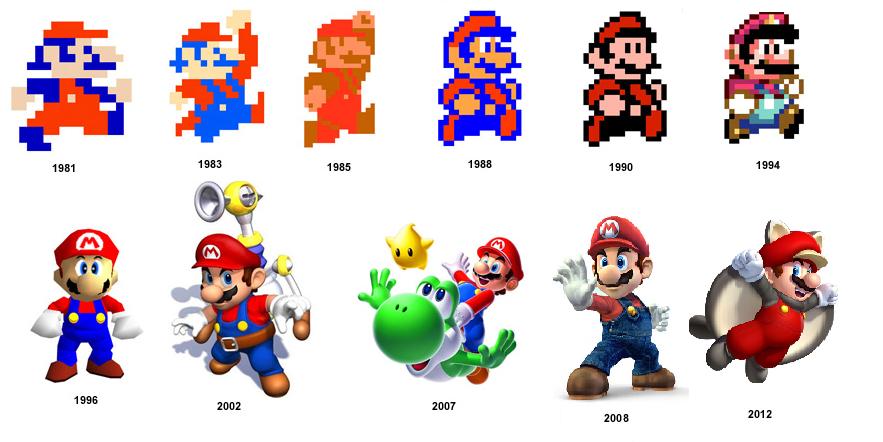


Pantalla de Mario en Juego, Donkey tira un barril

Menu de Pausa Buggeado, cuando se cambia de pantalla en el sistema operativo con Alt+Tab pasa este problema, pero no afecta al juego.

Pantalla de Información del Estudiante

Evoluciíon de Mario:



# Referencias

Gonzáles de León, F. 05 Videojuegos con Python y Pygame: Animando Sprites. Recuperado de <https://www.youtube.com/watch?v=WrCAeE-fL_k> el 30 de Abril del 2017.

Gonzáles de León, F. 06 Videojuegos con Python y Pygame: Física I El Salto. Recuperado de <https://www.youtube.com/watch?v=ddxSjCXhN9g> el 30 de Abril del 2017.

Guerrero Marrero, A. Razón Artificial, Tutoriales Pygame. Recuperado de <http://razonartificial.com/tutoriales-pygame/> el 29 de Abril del 2017.