

Tecnológico de Costa Rica
Área Académica de Ingeniería en Computadores
Taller de Programación

Grupo 01

Documentación externa del Proyecto: Monkey

Profesor:
Jeff Schmidt Peralta

Estudiante:
Brayan Alfaro Gonzalez 2019380074

I semestre
05 de abril, 2019

Tabla de contenidos

Introducción	2
Descripción del Problema.....	3
Análisis de resultados.....	4
Dificultades encontradas.....	9
Bitácora.....	10
Estadística de tiempos.....	12
Conclusion.....	13

Introducción

Se desarrolló el proyecto Monkey, el cual está basado en el juego de plataformas Donkey Kong, de los años 80. El juego elaborado usa las mismas imágenes del juego original y busca parecerse a este en lo más posible, sin embargo se llevaron a cabo cambios apropiados debido al límite de tiempo disponible para llevar a cabo el proyecto.

El juego consiste en el personaje, Mario, que trata de llegar hasta la princesa mientras el mono le lanza barriles para detenerlo. En esta implementación, cuando Mario llega al final, este reaparece abajo para empezar el próximo nivel. Esto sucede así, hasta que Mario logra ganar. En esta implementación se utilizaron varias imágenes tomadas de internet para llevar a cabo las animaciones, todos los personajes tienen algún tipo de animación y se buscó que se viera natural y lo más realista posible. Además, se agregaron audios apropiadamente.

El juego tiene un menú en el que se pueden ver los puntajes más altos alcanzados por otros jugadores, y también se puede configurar la dificultad del juego, entre otras cosas. De esta manera se busca que el juego sea flexible y que se adapte a las habilidades y deseos del jugador. Además, el salón de la fama le agrega un espíritu de competitividad al juego, el cual es valioso para que los jugadores se apasionan más por jugar y lograr superarse.

La implementación gráfica se desarrolló con la librería Pygame, aunque también se usó Tkinter para que el usuario introduzca su nombre, ya que Pygame no tiene ninguna funcional similar, e implementarlo con Pygame implicaría una cantidad de trabajo extra innecesario.

La cantidad de líneas de código que se utilizaron en el proyecto ronda las 1000, y el proyecto tomó alrededor de dos semanas en completarse, a partir del momento en que se inició la programación.

Descripción del Problema

El problema consistió en lograr la implementación de un juego similar a Donkey Kong. Lo único requerido era que este juego fuera de plataformas, y que tuviera tres niveles, además de que era necesario contar los puntos que ha logrado el jugador. Por lo cual, las imágenes y animaciones fueron escogidas por el programador. En pantalla, en medio del juego, se debían de mostrar los puntos, las vidas restantes y el nivel actual.

En general, se requería que Mario subiera por algún tipo de escalera u otro similar y que se desplace a través de las plataformas, además se requería que hubieran obstáculos que Mario debiera de brincar.

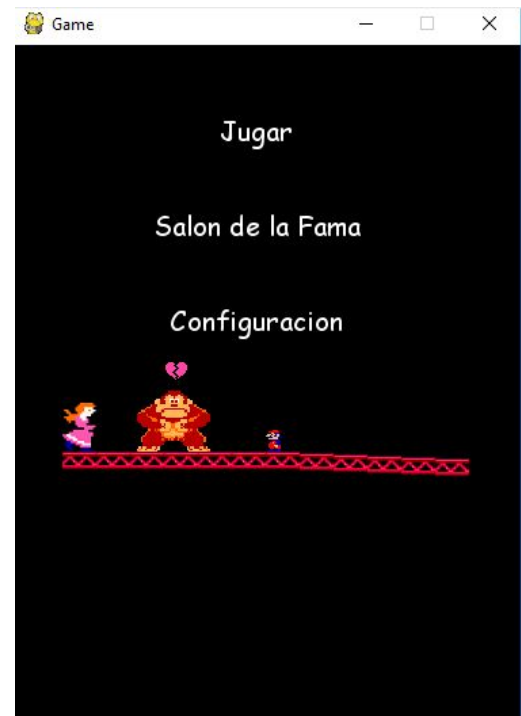
En cuanto al menú, este debía de tener acceso a un salón de la fama, en el cual se mostrara la información sobre los mayores puntajes obtenidos, y que se hiciera por medio de archivos de texto, esto para almacenar la información para futuros juegos. De nuevo, la forma gráfica del menú fue elegida por el programador.

Además de lo anterior, se debía de crear un menú para las configuraciones del juego, dificultad, manera de obtener los puntos y vidas del jugador.

Finalmente, era necesaria también la implementación de una animación al empezar el juego.

Análisis de Resultados

-Se creó el menú inicial de manera exitosa, en donde el usuario puede acceder a los otros menús del juego, o bien, iniciar el juego en sí. Este menú cuenta con una imagen representativa del juego.



- Se creó un menú de configuración, en el cual el usuario puede escoger la cantidad de vidas, la dificultad y la manera de obtener puntos.

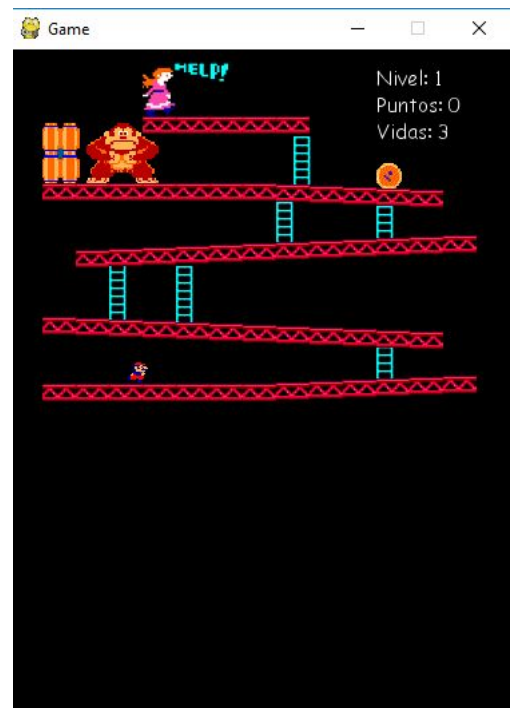
Los elementos en color rojo son los que están seleccionados. Cuando se cambian la cantidad de vidas o la dificultad, los colores cambian de acorde.

Similarmente, al hacer clic en las opciones de brincar, subir plataformas o pasar nivel estas cambian de color, a menos de que la opción seleccionada sea la única en rojo.

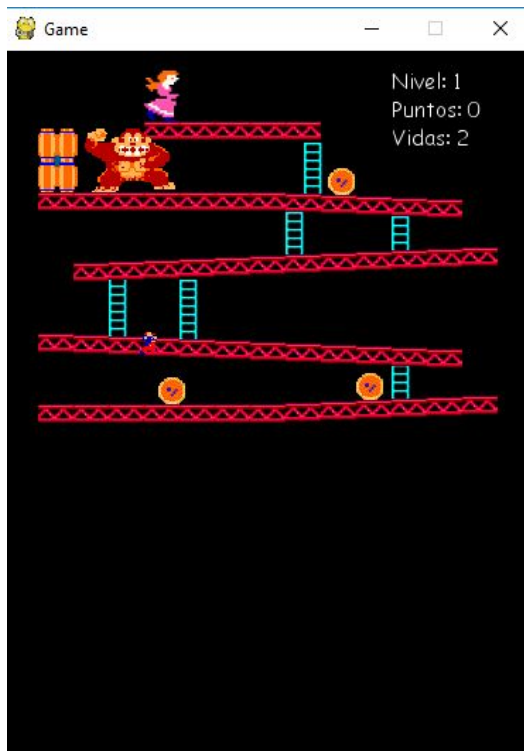
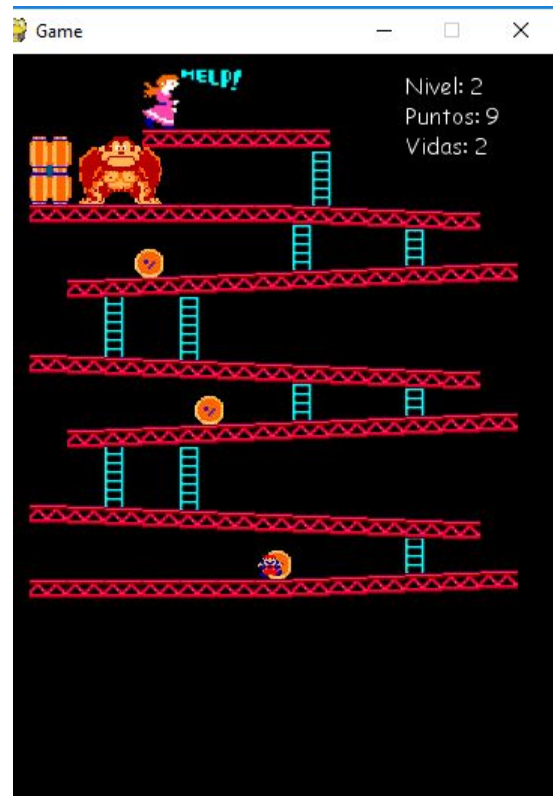


- Se creó exitosamente el menú del salón de la fama, en el cual se observan las mejores puntuaciones logradas en el juego. Esto se logra mediante un archivo de texto que guarda la información, este se edita al final de cada partida para agregar puntajes nuevos.

-Se creó el juego, en el cual mario se mueve según la dirección que el usuario presione en las teclas. Además, los barriles caen y se desplazan mientras son animados. En esta foto se observa que se está en el nivel 1 y no se tienen puntos. Además se tienen 3 vidas.



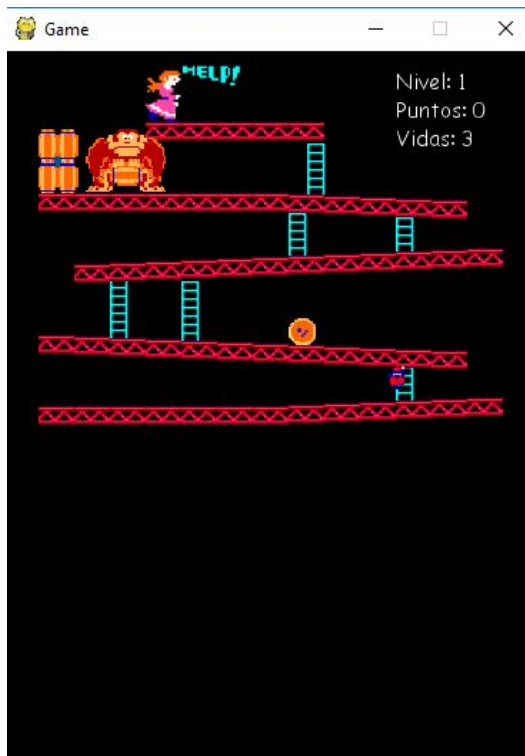
- En la imagen de la derecha se observa como Mario choca con un barril y cambia su imagen. Esta derrota causa una pérdida de vidas. Además, aquí se observa que se está jugando en el nivel 2, osea que anteriormente se ganó un nivel.



- Aquí se observa a Mario brincando sobre un barril, luego del brinco los puntos van a haber aumentado. Para brincar, el usuario debe presionar la tecla de espacio.

En esta imagen se ve que la vida disminuyó debido a un choque.

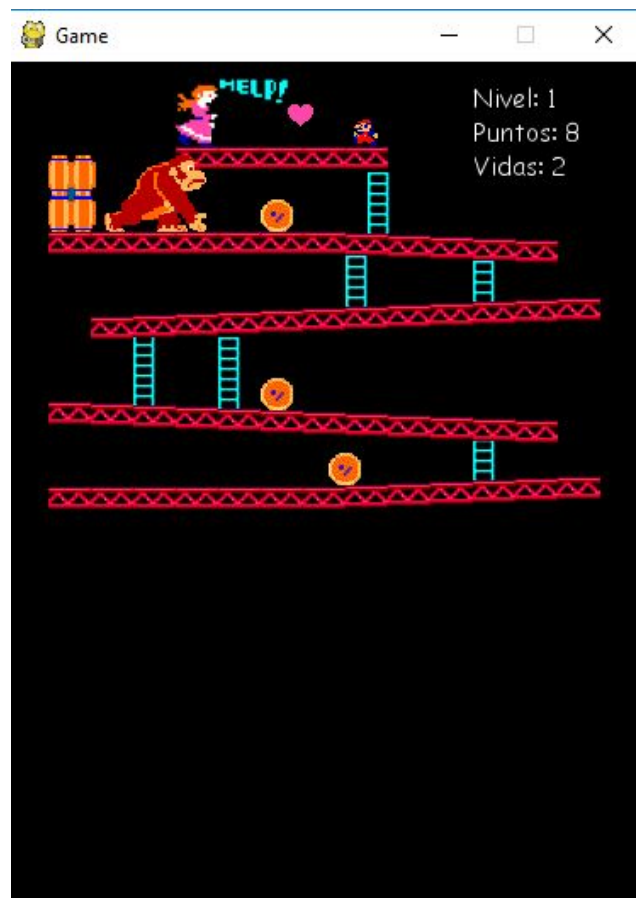
Además, se nota que la princesa y el mono tienen imágenes diferentes a las vistas anteriormente.



-Aquí se observa como Mario sube una escalera, Mario solo puede cambiar de plataforma cuando está en una escalera, y la imagen de Mario cambia a como se ve en la imagen.

Hay restricciones al movimiento en una escalera, solo se puede ir hacia arriba y abajo, y cuando se llega al final de la escalera se puede mover hacia la izquierda y la derecha.

- Aquí se observa que cuando Mario llega a la última plataforma y ve hacia la princesa, aparece un corazón y se cambia de nivel. Justo después de aparecer el corazón el juego se pausa por un corto instante.





-Cuando el usuario pierde todas sus vidas o gana el último nivel, se acaba el juego, y se le pide al usuario que escriba su nombre, esto para guardarlo en el salón de la fama, si logra el puntaje suficiente. Si el usuario presiona la tecla cancel, no se hacen cambios al salón de la fama y se termina el juego.

Adicionalmente, se incluyeron efectos de audio en el juego. Existe un efecto de audio cuando se pierde, cuando se gana y al final del juego. Además, se agregó una pista de audio de fondo para cuando se está jugando. Y finalmente, se agregó un audio a la animación inicial.

En cuanto a las funcionalidades no logradas, se puede mencionar que el juego no tiene ninguna opción de pausa, ni ningún menú cuando se está jugando, lo que constituye una pérdida de jugabilidad para el usuario, por ejemplo, al no poder pausar y cambiar la dificultad. Si a un jugador le parece demasiado complicado, deberá esperar a perder para cambiar esta dificultad.

También, se puede mencionar que la animación de los barriles al caer es diferente a la del juego de verdad, esto debido a que no se agregaron imágenes especiales para estas situaciones, ya que se debían de hacer esfuerzos adicionales para que se detectaran los choque de Mario con los barriles cayendo.

Dificultades Encontradas

- El manejo de las animaciones fue algo que se mejoró poco a poco, ya que inicialmente esto se manejaba con muchos hilos; había un hilo por cada barril, por ejemplo. Esto causaba grandes complicaciones a la hora de calcular los choques y al actualizar la pantalla, ya que los elementos del juego se actualizaban a destiempo. Finalmente, esto se solucionó al crear la clase Animación, la cual maneja todos los cambios en pantalla de todos los elementos, y hace que estos cambien de imagen o posición.
- Hubo problemas al calcular los choques, ya que al usar la clase Rect de Pygame, el jugador perdía aunque realmente no estuviera en contacto con barril. Esto porque los barriles son circulares y los Rect, son rectangulares. Para solucionar este problema se implementó un análisis de pixel por pixel una vez que hubieran colisiones en los Rect.
- Hubieron también complicaciones a la hora de actualizar la pantalla, ya que algunos elementos quedaban dibujados varias veces, lo cual es algo completamente indeseable en el juego. La solución consistía en hacer una copia del frame, cuando en este solo estaban los elementos del escenario, y cuando se iba a actualizar la pantalla primero se pone el fondo en pantalla y luego se dibujan los diferentes personajes y obstáculos.
- También, en el momento de agregar el puntaje al salón de la fama, se necesitaba un nombre, pero en Pygame no existe ninguna manera de ingresar texto desde la interfaz, por lo cual se decidió acudir a la librería Tkinter, y se utilizó la función `simpdialog.askstring(args)`, para pedirle al usuario que introduzca su nombre.

Bitácora

Tipo de Actividad	Duración	Descripción
Programación de las plataformas	3 horas	Se creó la clase Plataforma y la clase Peldano y se asignaron sus imágenes y funciones
Programación de las escaleras	1 hora	Se creó la clase Escalera la clase Escalón y se asignaron sus imágenes y funciones
Programación de Mario	6 horas	Se creó la clase Mario y se añadieron las funcionalidades de movimiento básicas en las escaleras y plataformas
Programación de los obstáculos	3 horas	Se crearon los obstáculos y se agregaron las funciones de movimiento
Agregar Brincos	6 horas	Se agrego la posibilidad de brincar, de manera que solo se pueda brincar una vez y que la dirección en un brinco no cambie
Calcular choque	6 horas	Se agregó una función que calcula si hubo choque entre dos elementos
Manejo General de las animaciones	4 horas	Se integraron animaciones simples en la clase Animación
Añadir otros personajes	1 horas	Se añadieron los personajes de Mono y Princesa
Manejo General de las animaciones	8 horas	Se añadieron todas las imágenes necesarias y se programaron las animaciones en todos los personajes

Manejo de un Juego	4 horas	Se creó la clase Partida para manejar una partida. Esta maneja los puntos y las vidas
Manejo de eventos	4 horas	Se creó la clase Game_handler que maneja los eventos que hace el usuario en medio del juego
Creacion de los menu	5 horas	Se crearon e implementaron los menús y el manejo de archivos de texto para guardar información
Animación inicial	1.5 horas	Se creó la animación inicial
Sonidos	4 horas	Se agregaron todos los sonidos al juego

Estadística de Tiempos

FUNCION	TOTAL HORAS
Requerimientos/Diseño	5 horas
Investigación de funciones	5 horas
Programación	57 horas
Documentación interna	5 horas
Pruebas	10 horas
Elaboración del Documento	6 horas
TOTAL HORAS	88 horas

Conclusión Personal

Como conclusión personal, considero que este fue un proyecto muy retador que requirió de muchas horas de programación e investigación. De este proyecto hubo mucho aprendizaje, y se aplicaron muchas de las ideas que he aprendido en el curso de Introducción a la Programación.

El juego que se logró programar es muy similar al original, y el resultado general del proyecto fue muy satisfactorio desde el punto de vista personal, ya que a pesar de que no se tenía mucho conocimiento de juegos, gracias a la investigación y a los talleres que dieron estudiantes de la carrera, se facilitó un poco la tarea, ya que con este conocimiento adquirido fue posible empezar el proyecto.

Por otro lado, a pesar de que el producto final lo veo con buenos ojos, no es de mi mayor agrado la programación de interfaces de usuario y de juegos, sino que son de mayor interés para mí otros temas de la computación y programación.