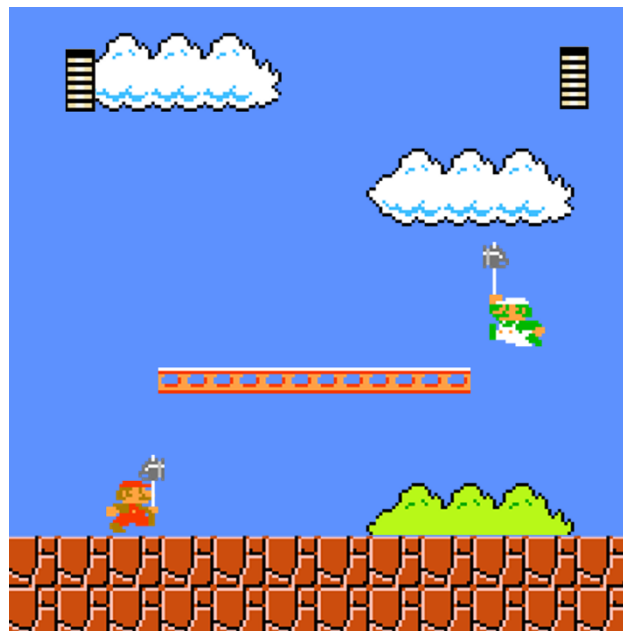


Universidad de Puerto Rico  
Recinto Universitario de Mayagüez  
CIIC 4082: Arquitectura de Computadoras II  
Profesor: Juan Felipe Patarroyo Montenegro  
Descripción Proyecto

El objetivo al final de los dos proyectos es el de tener un juego de pelea tipo “Smash” en el que dos jugadores se enfrenten en una misma pantalla. El juego deberá ser desarrollado enteramente utilizando lenguaje assembly. Ninguna solución utilizando programas de rápida edición de juegos será tomada como válida.



*Figura 1. Ejemplo del producto final*

Ambos jugadores deberán tener las siguientes características:

- Mínimo de 3 colores por personaje. El personaje 1 debe tener un color diferente al personaje 2
- Mínimo tamaño de 2x2 sprites (16x16 pixeles)
- Barra de puntos de vida
- Animación de caminar
- Animación de saltar
- Animación de golpear al frente
- Animación de cuando es golpeado
- Animación de cuando se le acaban los puntos de vida (game over)
- Detener el avance en caso de colisión con el otro jugador o cuando toca el piso
- El movimiento debe depender de la entrada del control

Además, deberán tener un fondo de pantalla que utilice al menos 4 elementos diferentes. Por ejemplo en la Figura 1, el fondo contiene el elemento del piso, el elemento de la plataforma flotante, el arbusto y las nubes. Una característica importante es que el fondo debe tener un piso y una plataforma flotante, las cuales servirán para hacer colisión y evitar que el personaje caiga. Para los colores, existe un requerimiento mínimo de 6 colores.

Todos los detalles adicionales quedan bajo su discreción de diseño. Las gráficas que usen deberán ser de su propia autoría y pueden diseñarlas tanto a mano con la tabla de excel o utilizando programas de edición (Nexxt, CHR-editor, etc.). Si deciden utilizar estos programas, queda a su responsabilidad aprender a utilizarlos y poder invocar los archivos .chr dentro de su programa.

A continuación, se detallan las instrucciones específicas para cada uno de los entregables de los dos proyectos. El proyecto 1 será la primera etapa para poder realizar el proyecto 2.

## Proyecto: Fondo y Animaciones de Movimiento

**Fecha de entrega:** Miercoles 15 de noviembre 2023

El objetivo de este proyecto es el de desarrollar un programa que muestre su personaje caminando y saltando sobre un fondo de pantalla de acuerdo a las entradas del control. A continuación se detallan cada uno de los componentes que deben entregar. Cada entregable deberá tener su propio código fuente y su archivo .nes.

### El fondo

Podrán encontrar suficiente documentación en la web y en los enlaces compartidos en tareas anteriores sobre cómo programar el fondo para el NES. Sin embargo, les recomiendo el siguiente video que hace parte de una serie de tutoriales de programación en NES:

[https://www.youtube.com/watch?v=x1OW7IS\\_RnU](https://www.youtube.com/watch?v=x1OW7IS_RnU)

Este video contiene buenas referencias y además tiene el enlace a un repositorio de GitHub que contiene códigos de ejemplo que podrán utilizar. Recuerden que en el NES todo ocurre a nivel de PPU y que se debe transferir la memoria de la CPU utilizando posiciones específicas en la memoria ROM. El proceso es similar al que hicieron programando los sprites de las letras de su nombre. Notarán que con los fondos no tienen la limitación de 8 sprites por línea horizontal.

**Entregable:** Un programa que contenga un fondo que contenga al menos 4 elementos y 6 colores diferentes. 2 de estos 4 elementos deberán ser el piso y la plataforma flotante.

### El personaje

Deberán crear su propio personaje, pueden de igual manera hacerlo a mano o pueden utilizar los programas de edición que exportan archivos .chr o el formato que sea más conveniente. Lo más importante es que el personaje debe ser de su autoría. Para crear dicho personaje deberán crear el sprite base que es el personaje quieto y luego crear los sprites correspondientes a los eventos de movimiento. En el caso de Mario (Ver Figura 2), hay diferentes tipos de sprites porque el personaje realiza labores muy complejas. En el caso de ustedes, les sugiero que diseñen tan sólo 3 sprites adicionales para caminar (recuadros azules) y uno para cada otra posición como estar de pie (violeta), saltar (verde), o game over (naranja).



*Figura 2. Sprites utilizados en el juego Super Mario*

Si el personaje está mirando hacia la dirección opuesta, pueden utilizar la palabra de configuración de sprite para invertir el sprite. Les sugiero los siguientes recursos:

<https://www.youtube.com/watch?v=oB7z-tljTVE&list=PLnjAJ75A-OrDWPCvzIB63cqsRCLHJd80E&index=11>

[https://www.youtube.com/watch?v=7Co\\_8dC2zb8](https://www.youtube.com/watch?v=7Co_8dC2zb8)

**Entregable:** Diseño de todos los sprites de los personajes visualizados en pantalla por separado. Pueden estar de manera estática. Lo importante es que se puedan ver todos los sprites dibujados en la misma pantalla sin importar su posición. De igual manera, la misma pantalla deberá contener los mismos sprites mirando hacia el lado opuesto.

### La animación

El personaje deberá tener la capacidad de moverse horizontalmente, saltar, golpear, recibir daño e indicar cuando perdió todos sus puntos de vida. De estos, el más complejo es el de moverse horizontalmente, ya que implica crear animación que hace cambio de sprites. Para el resto de acciones, basta con poner el sprite de golpear, saltar, etc.

**Entregable:** Deberán crear un programa que muestre al personaje en la misma posición (x,y) pero que muestre constantemente la animación de como si estuviera desplazándose como en el siguiente enlace:

<https://www.deviantart.com/mariofan345/art/Super-Mario-Bros-Mario-Gif-2-842284419>

### La entrada del control con animaciones

Deberán estudiar cómo mapear los botones de ambos controles del NES. Les sugiero los siguientes recursos:

<https://www.youtube.com/watch?v=tPhUjSZH6vY&list=PLnjAJ75A-OrDWPCvzIB63cqsRCLHJd80E&index=12>

<https://www.youtube.com/watch?v=nAStgQzPrAQ>

Luego de esto, deberán poder ligar los botones leídos a las animaciones del personaje. En este caso deberán utilizar los botones de izquierda, derecha para desplazamiento horizontal, el botón A para saltar y el botón B para golpear. Noten que deberán poder realizar el cambio de orientación del personaje, saltar y detectar la colisión tanto del piso como de la plataforma flotante. El personaje deberá además colisionar con los bordes de la pantalla y no ocultarse.

**Entregable:** Deberán crear un programa que permita desplazar el personaje a través del fondo de pantalla creado anteriormente con respecto a los botones presionados en el control.

### El reporte

No se utilizará la metodología de reporte escrito. En este caso se pedirá que cada grupo suba un documento PDF de una o dos páginas que contenga lo siguiente por cada uno de los elementos descritos anteriormente (fondo, personaje, animación, controles) :

- Enlace a un repositorio en GitHub con el proyecto en VS Code y el programa .nes con el programa que realiza la función de cada tarea.
- Enlace a de Youtube a un video tutorial realizado por ustedes sobre cómo se desarrolla cada actividad. El video puede ser en español o inglés. Puede ser un video por cada actividad. El video deberá tener un título y una descripción que contendrá el enlace al repositorio en GitHub. El video debe incluir lo siguiente:
  - Presentación del tema junto con el nombre de cada integrante del equipo.
  - Especificación de objetivos
  - Ejemplo del programa funcionando
  - Explicación del código y herramientas utilizadas
  - Explicación del repositorio compartido
- Capturas de pantalla del resultado de cada tarea.

Se tendrá muy en cuenta la calidad del video que compartan y la claridad con la que presenten la información. El equipo con el mejor video tendrá una bonificación adicional de 10% del puntaje total del proyecto 1 y el segundo mejor un 5%.

### Criterios de evaluación

La nota de este proyecto se dividirá de la siguiente manera:

- Fondo – 15%
- Personaje – 15%
- Animación – 15%
- Integración del control con la animación – 30%
- Reporte – 25

## Bono (15% adicional nota final): El juego de pelea

**Fecha de entrega:** Miercoles 22 de noviembre 2023

Luego de poder manipular un único personaje, tendrán las bases para poder crear su juego de pelea. Al igual que en el proyecto 1, cada entregable deberá tener su proyecto en Github que contenga el código en assembly y el programa .nes que corre el entregable.

El juego deberá contener dos personajes que puedan desplazarse por la pantalla y que puedan golpearse mutuamente. Al recibir golpe, un indicador de vida disminuirá los puntos hasta que se vuelva cero y el jugador pierda. A continuación se detallan las actividades:

### Integración del segundo jugador

Esta tarea implica realizar las mismas tareas del proyecto 1 para crear un segundo personaje que deberá tener un color diferente al primero. Para esto deberán pensar en que su código sea lo suficientemente modular para poderse ejecutar en ambos personajes sin gastar más memoria de programa.

**Entregable:** Un programa .nes con los dos personajes moviéndose en la pantalla dependiendo de las entradas de ambos controles. Noten que a este punto no deben golpearse los personajes. Solamente moverse, saltar y colisionar.

### Dinámica de la pelea:

Una vez ambos jugadores se muevan en pantalla, deberán incluir la dinámica de pelea que incluirá la animación de ejecutar golpe, recibir golpe, disminuir la barra de salud y animación de game over.

**Entregable:** Un programa .nes con el juego de pelea completado

### El reporte

Se seguirá la misma metodología que en el proyecto 1. Al final, se espera que creen una lista de reproducción en Youtube con todos los contenidos de cada una de las tareas. Noten que la bonificación adicional aplica para este proyecto también.

### Criterios de evaluación

La nota de este proyecto se dividirá de la siguiente manera:

- Integración del segundo jugador – 45%
- Dinámica de la pelea – 30%
- Reporte – 25%

## Metodología sugerida

- El trabajo se podrá hacer en parejas.
- Mi recomendación es que se reúnan primero a hacer un plan de adiestramiento en el cual definan fechas límites para que ambos miembros vean toda la documentación que puedan sobre cómo cumplir con las tareas.
- Luego realicen un plan de trabajo especificando las fechas en las que tendrán listo cada entregable. Consideren la parte del reporte que le puede quitar tiempo considerable. Luego de esto, realicen el diseño de alto nivel del proyecto.
- Definan cómo van a manejar los datos compartidos. Algunas personas utilizan servicios de nube como Google Drive o OneDrive. Yo les sugiero utilizar el mismo GitHub para tener control sobre las versiones que trabajan.
- Definan qué estructura de programa utilizarán. Les sugiero fuertemente que utilicen subrutinas y el stack para modularizar el código. De esta forma podrán separarse labores si especifican qué recibe y qué retorna cada subrutina. Por ejemplo, la rutina de dibujar el fondo y el personaje, la rutina de animar el movimiento, la rutina del salto, rutina de detección de colisión, rutina de la inversión de la dirección del personaje, rutina de golpear, etc.
- Especificar qué atributos tiene el personaje (similar a programación orientada a objetos). En este caso serían las posiciones (x,y), orientación, estado del salto (sube, baja, posición del salto, etc), nivel de salud, etc. Luego definan las posiciones en memoria RAM donde guardarán estas características. Les sugiero que averiguen el tema “zero page addressing” que hace un acceso de memoria mucho más rápido en este tipo de casos. También les sugiero que utilicen el concepto de máquinas de estado para definir los atributos del programa (<https://www.youtube.com/watch?v=8lZ53Sx5oc0> ).
- Dividan el desarrollo de cada subrutina y tarea de manera que se complementen en trabajo.
- Definan un horario en la semana en el que trabajarán juntos. Les sugiero que dediquen mínimo 2 horas de trabajo semanal juntos. Esto no quiere decir que deban trabajar en el mismo código. Cada uno puede trabajar por separado en sus cosas pero aprovecharán el espacio para compartir ideas. Este es quizás el punto más crítico que le puede garantizar el éxito en el proyecto.
- Una vez tengan las subrutinas listas, definan cómo las van a integrar.