

Apr, 2021

T.A.: José Fernando Morales

COMPUTER ENGINEERING  
(CEAB SUBSTANTIALLY EQUIVALENT PROGRAM AND CFIA  
ACCREDITED SEALS)

Project I:

Moon to Mars

Professor:

Milton Villegas Lemus

*The Earth is the cradle of Humanity.  
But one doesn't always live in the cradle.*

Konstantin Tsiolkovsky

*The best computer is a man,  
and it's the only one that can be mass-produced by unskilled labor.*

Wernher von Braun

*The future of humanity is going to bifurcate in two directions:  
Either it's going to become multiplanetary,  
or it's going to remain confined to one planet and  
eventually there's going to be an extinction event.*

Elon Musk

## 1. Objetivo General

---

- Diseñar una solución computacional relacionada con conceptos de recursividad, almacenamiento en disco, con el fin de aplicar la teoría en problemas específicos, considerando las buenas prácticas de programación.

## Objetivos Específicos

---

- Desarrollar un proyecto software de pequeña escala, aplicando recursividad para repetición en Python.
- Aplicar las habilidades adquiridas en los talleres y experimentadas de forma individual.
- Realizar la documentación requerida para el proyecto.

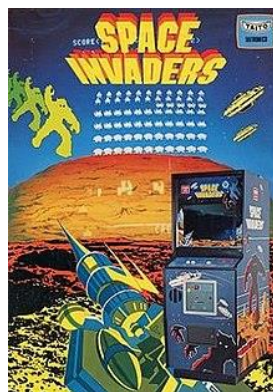
## 2. Antecedentes

---

En la historia de los videojuegos un clásico es el juego Space Invaders.

En 1978, en Japón, la empresa Taito lanzó un Arcade (máquina de video juego) con un juego del tipo de Shoot 'em up. El juego llamado Space Invaders. se vino consolidar lo que se ha clasificado como genero Shoot 'em up.

El juego creado por el ingeniero japonés Tomohiro Nishikado quien diseñó y desarrolló el hardware necesario para producir Space Invaders. Como todo en la industria del mundo computacional, el concepto del juego se inspiró en juegos previos tales como Space Monsters lanzado por Taito Corporation en 1972, el juego tipo arcade Breakout, desarrollado por Atari, la influencia es palpable en el diseño de Space Invaders, aunque la mecánica de juego. Se identifica en Invaders la intención de Nishikado de incluir en el videojuego de disparos un tipo clase de avance o progreso en el tiempo de juego mediante la conclusión de niveles, proporcionando en sí el efecto adictivo.



Flyer promocional del Juego lanzado por Taito

## Descripción del Proyecto

---

La organización para la que usted trabaja está diseñando una SBC (computadora de una sola placa) multiuso basada alrededor de la arquitectura RISC-V. Como parte del plan de pruebas del dispositivo la organización necesita comprobar que la computadora es capaz de ejecutar un programa de interfaz gráfica escrito en python. Se observa que para asegurar una buena funcionalidad de otros programas se debe comprobar al menos que el computador sea capaz de procesar interacciones con la interfaz, animaciones y audio.

La organización decide entonces que la prueba ideal sería un juego simple, ya que además este les permitiría probar si la implementación de la arquitectura procesa bien las operaciones recursivas.

Su jefe decide asignarle la tarea de programar dicho juego. Luego de una reunión logran formalizar y resumir los requisitos del juego de la siguiente manera:

### Especificaciones técnicas:

- Resolución de relación de aspecto 3:4. Escoja a conveniencia (ejem. 600x800 px)
- Programa desarrollado en python 3
- Uso únicamente de tkinter como biblioteca de interfaz gráfica
- Controles con las cuatro flechas del teclado para movimiento, tecla espacio para disparo
- Biblioteca de sonido preferiblemente de VideoLAN (python-vlc)

### Mecánicas de juego:

El juego será de tipo "Shoot 'em up", la nave del jugador se puede mover tanto en el eje Y como en el eje X, no existirá navegación sobre el universo, sino que cada nivel tendrá lugar en una pantalla estática.

La nave del jugador tiene una vida máxima de 50 puntos. Cada ataque recibido o contacto con enemigo disminuirá la vida del jugador en una cantidad determinada de puntos (ver sección Descripción de niveles). Al llegar a los 0 puntos de vida, el jugador pierde el juego.

La nave del jugador dispone de un cañón, el cual dispara proyectiles que hacen 1 punto de daño. Sería conveniente que el código se pueda adaptar para diferentes patrones de disparo, aunque no es necesario implementarlos en este Proyecto. Cada presión de tecla de disparo resulta en un solo disparo, es decir, el jugador tiene que dejar de presionar la tecla para lanzar un nuevo proyectil. Esto para evitar que el comportamiento del cañón del jugador sea como un rayo láser.

## Descripción de Niveles (mapas):

El objetivo del jugador es sobrepasar tres niveles distintos en el menor tiempo posible y perdiendo la menor cantidad de vida por cada nivel. Se debe tomar en cuenta que, al iniciar un nuevo nivel, la vida se restaura por completo.

Para probar el concepto se decidió programar primero los jefes finales de nivel antes que enemigos subordinados, cada nivel únicamente tendrá un enemigo: el jefe. Cada jefe tiene un patrón de disparo característico, y una cantidad de vida determinada. Todos los jefes aparecen en el lado contrario al jugador (posición vertical superior de la pantalla).

### Primer Nivel:

Este enemigo se mueve constantemente de izquierda a derecha. Cada 2 segundos recorridos el enemigo decide si va a embestir al jugador. Para esto, el enemigo genera un número al azar entre 1 y 10, si el mismo es múltiplo de 3, embiste, si no, sigue su camino.

La embestida consiste en que el enemigo pausa su movimiento de derecha izquierda, espera medio segundo, y luego baja rápidamente hacia la parte inferior de la pantalla. Una vez llega al límite inferior, la nave se devuelve a su posición anterior subiendo rápidamente. El enemigo causa 10 puntos de daño al hacer contacto con el jugador, se encuentre en modo embestida o no. Durante la embestida, el generador de números al azar debe desactivarse, de manera que no se pueda activar una segunda embestida mientras que una ya está en progreso.

Este jefe dispone de 30 puntos de vida. Cada vez que hace una embestida, se quita a sí mismo 1 punto de vida.

### Segundo Nivel:

En este nivel el jefe no se desplaza normalmente por la pantalla, sino que cada 2 segundos escoge una posición horizontal al azar y se teletransporta a ella. Siempre se mantiene en la misma posición vertical. Cada segundo el enemigo dispara 3 proyectiles distintos. Un impacto de uno de estos proyectiles causa 3 puntos de daño. La separación entre estos proyectiles debe ser tal que sea posible para el jugador recibir el impacto de los 2 al mismo tiempo. El jefe en este nivel dispone de 40 puntos de vida.

### Tercer Nivel:

Este enemigo es la combinación entre el primer jefe y el segundo. El enemigo se mueve horizontalmente de izquierda a derecha igual que el primer jefe. Cada segundo, dispara 3 proyectiles igual que el segundo jefe. Cada 6 segundos realiza una embestida (durante la misma, deja de disparar).

Al terminar una embestida, el enemigo tiene 50% de probabilidad de hacer una de las siguientes dos cosas:

1. Teletransportarse a su posición inicial
2. Teletransportarse a una posición horizontal al azar que se encuentre en la misma coordenada vertical que la posición inicial. Una vez llegada a esta posición, inicia una segunda embestida.

El jefe de este nivel dispone de 50 puntos de vida, pero no sufre daños al embestir.

### Acerca de los Puntajes:

El jugador irá acumulando un puntaje según avance en el juego. Cada disparo acertado dará un punto. Si durante un nivel el jugador no recibe daño, obtiene un bonus de 10 puntos.

Debe haber un bono por completar el nivel rápidamente. Se propone que, si el jugador completa el nivel en menos de 30 segundos, recibirá un bono de 20 puntos. El valor de 30 segundos es algo arbitrario, por lo que se le recomienda ajustar dicho bonus al tiempo que usted considere más apropiado.

Si al finalizar el juego (complete o no el nivel) el puntaje de jugador supera alguno de los mejores 5 anteriores, se le comunica por medio de un mensaje cuál posición obtuvo y con cuántos puntos llegó a dicha posición.

### Descripción de la Pantallas de Juego:

Toda pantalla que no sea la de inicio debe proveer alguna manera de volver a la pantalla de inicio

#### Pantalla de inicio:

En esta pantalla el jugador puede acceder a cualquier otra pantalla del juego por medio de botones. Además de los accesos a otras pantallas, se muestra una caja de texto y una selección de nivel. La caja de texto (Entry) se usará para permitirle al jugador ingresar el nombre con el que se registrará su puntaje. Los selectores de nivel permitirán empezar por el nivel que el jugador escoja, aunque el valor por defecto será el nivel 1.

#### Pantalla Complementaria o About:

En esta ventana se mostrará la información complementaria. Se deben incluir los siguientes datos:

- País de producción
- Universidad y Carrera
- Asignatura, año que cursa y grupo
- Nombre del profesor
- Versión del programa
- Nombre del Autor
- Autores de módulos modificados/utilizados por usted
- Instrucciones o datos que considere importantes para el uso del programa

## Pantalla Mejores puntajes

Esta pantalla contiene una lista de los 5 mejores puntajes obtenidos. Estos puntajes se leen desde un archivo .txt, el cual es procesado como un archivo secuencial. Cuando un jugador termina una partida, debe cargar el .txt, comparar los resultados guardados y actualizar el archivo en caso de ser necesario. Si luego de terminar una partida se obtiene uno de los 5 mejores puntajes, esto tiene que mostrarse al volver a la pantalla principal y abrir la ventana de mejores puntajes.

No se permite el uso de bibliotecas de manejo de bases de datos o automatización de lectura de archivos secuenciales. Los datos de puntajes deberán ser manejados únicamente utilizando las funciones de acceso a archivos vistas en clase.

## Pantalla de juego

La pantalla de juego tendrá un fondo inmóvil (aunque usted puede animarlo si gusta), agradable a la vista y con temática apropiada según el diseño de su juego. Además de los elementos de juego ya abarcados como enemigos y jugador, deberán mostrarse los siguientes elementos de interfaz gráfica:

- Puntaje
- Tiempo transcurrido
- Nombre del jugador
- Puntos de vida de jugador
- Puntos de vida del enemigo actual
- Botón de regreso a pantalla principal (automáticamente aborta el juego, procesar el puntaje es opcional en este caso)

## 4. Aspectos Generales

---

- Se requiere el uso de Tkinter para desarrollar la interfaz gráfica.
- El Proyecto es individual. Se debe enviar el software el día Viernes 21 de Mayo de 2021, por aparte en una asignación específica enviar la documentación (incluir bitácora) vía el Classroom, a más tardar a las 23:30. Debe entregar todos los componentes software necesarios para probar todos los comandos de forma que demuestre pleno funcionamiento.
- Debe indicarse un archivo readme. txt con la versión de Python ( $\geq 3.3$ ) a utilizar para la revisión, alguna otra indicación que se considere importante. Este archivo contiene el nombre y la descripción de una función o procedimiento especial designado `mi_auto_doc()`, que genera de forma

automática (con print de Python), todas las autodocumentaciones de los principales módulos (mínimo 3).

- Cada archivo base debe enviarse el nombre construido de la siguiente forma: Iniciales del nombre (del autor) en mayúsculas, guion bajo primer apellido, guion bajo segundo apellido y extensiones txt y doc.
- Cualquier duda, omisión o contradicción en la especificación, se debe aclarar con el profesor y esta se difundirá, a través del Classroom de Google.
- La documentación (archivo .doc) debe contar con lo siguiente: Introducción; Conclusiones; Recomendaciones; Análisis de Resultados, dentro de este punto incluir: Diagramas de Módulos, Plan de Pruebas de los módulos principales; y Literatura o Fuentes Consultadas.
- Toda la documentación formará parte de la defensa.
- El código debe estar suficientemente documentado de tal forma que el autor(a), se pueda orientar fácilmente durante la defensa.
- El proyecto se debe defender previa cita con el profesor y el asistente. El profesor o asistente publicará unas fechas de defensa. Ud debe inscribirse para defender el proyecto, de no asistir a la defensa, únicamente obtendrá la nota correspondiente a la documentación del proyecto.
- Las funciones de repetición dentro de la lógica del programa deben ser desarrolladas recursivamente, no se aceptará con iteración. Desviarse de ésta directriz implicará la anulación de esa parte del código mediante comentarios en él y su correspondiente pérdida de puntos.
- Cualquier clase de copia de código será sancionada de acuerdo con el reglamento vigente y se llevará hasta la consecuencia de amonestación con carta al expediente. Código adoptado para el manejo de interfaz debe especificarse claramente la fuente y reconocer los créditos. Está prohibida la copia de código que involucre la solución lógica general del algoritmo.
- Si en la defensa no demuestra su autoría con el dominio propio de esa calidad solo se le otorgarán los puntos correspondientes a los obtenidos en la documentación.

## 4.1. Evaluación

---

Nombre \_\_\_\_\_

Carnet \_\_\_\_\_

### Project I: Moon to Mars

#### Tabla de evaluación

Rubro	Valor	Obtenido
<b>Aspectos Generales (23 pts)</b>		
<b>Pantalla Inicial</b> <ul style="list-style-type: none"><li>• Captura del nombre</li><li>• Acceso a otras pantallas</li><li>• Botón/acceso a inicio de juego</li><li>• Selección de dificultad inicial</li></ul>	4	
<b>Pantalla Información complementaria</b> <ul style="list-style-type: none"><li>• Información completa</li><li>• Retorno a pantalla de inicio</li></ul>	4	
<b>Pantalla de mejores puntajes</b> <ul style="list-style-type: none"><li>• Muestra mejores puntajes</li><li>• Se actualiza correctamente</li><li>• Mejores puntajes se mantienen al cerrar y abrir el juego</li></ul>	12	
<b>Sonidos</b> <ul style="list-style-type: none"><li>• Sonidos asociados a ataques</li><li>• Música de fondo en pantalla de juego y pantalla principal</li><li>• Manejo de cambios de música de fondo al cambiar de pantalla</li></ul>	3	
<b>Aspectos de juego (45 pts)</b>		
<b>Pantalla de juego</b> <ul style="list-style-type: none"><li>• Muestra indicador de puntaje</li><li>• Muestra indicador de tiempo</li><li>• Muestra vida del jugador</li><li>• Muestra vida del jefe de nivel actual</li><li>• Muestra nombre del jugador</li><li>• Se le comunica al jugador cuando pierde/gana</li><li>• Si el jugador obtiene uno de los 5 mejores puntajes, se le comunica el puntaje y su posición</li><li>• Botón de regreso a pantalla principal</li><li>• Muestra fondo agradable y aspecto de imagen solicitado</li></ul>	9	
<b>Jugador</b> <ul style="list-style-type: none"><li>• Jugador puede moverse en las 4 direcciones sin salirse de la pantalla</li><li>• Jugador puede lanzar proyectiles</li></ul>	6	



<ul style="list-style-type: none"> <li>• Implementa restricción sobre disparo de proyectiles</li> <li>• Restauración de vida al cambiar de nivel</li> <li>• Animación de sprite del jugador</li> <li>• Bonus de puntaje al completar nivel rápidamente</li> </ul>		
<b>Jefe Nivel 1</b> <ul style="list-style-type: none"> <li>• Implementa patrón de movimiento del enemigo por completo</li> <li>• Implementa aspectos relacionados a daño de ataques, vida y comportamiento de enemigo según especificado</li> </ul>	6	
<b>Jefe Nivel 2</b> <ul style="list-style-type: none"> <li>• Implementa patrón de movimiento del enemigo por completo</li> <li>• Implementa aspectos relacionados a daño de ataques, vida y comportamiento de enemigo según especificado</li> </ul>	6	
<b>Jefe Nivel 3</b> <ul style="list-style-type: none"> <li>• Implementa patrón de movimiento del enemigo por completo</li> <li>• Implementa aspectos relacionados a daño de ataques, vida y comportamiento de enemigo según especificado</li> </ul>	6	
<b>Colisiones</b> <ul style="list-style-type: none"> <li>• Detecta colisiones jugador – proyectil</li> <li>• Detecta colisiones enemigo – proyectil</li> <li>• Detecta colisiones jugador – enemigo</li> </ul>	12	
<b>Documentación 32 pts</b>		
<b>Introducción</b>	1	
<b>Conclusiones</b>	6	
<b>Recomendaciones</b>	5	
<b>Análisis de resultados</b> <ul style="list-style-type: none"> <li>• Diagrama de módulos</li> <li>• Plan de pruebas</li> </ul>	4	
<b>Bitácora</b>	4	
<b>Literatura o fuentes bibliográficas</b>	1	
<b>Módulo de Generación de Autodocumentación de los Módulos Principales</b>	2	
<b>Auto documentación sigue formato</b>	6	
<b>Documentación Interna o comentarios en partes de los módulos más importantes</b>	3	