

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

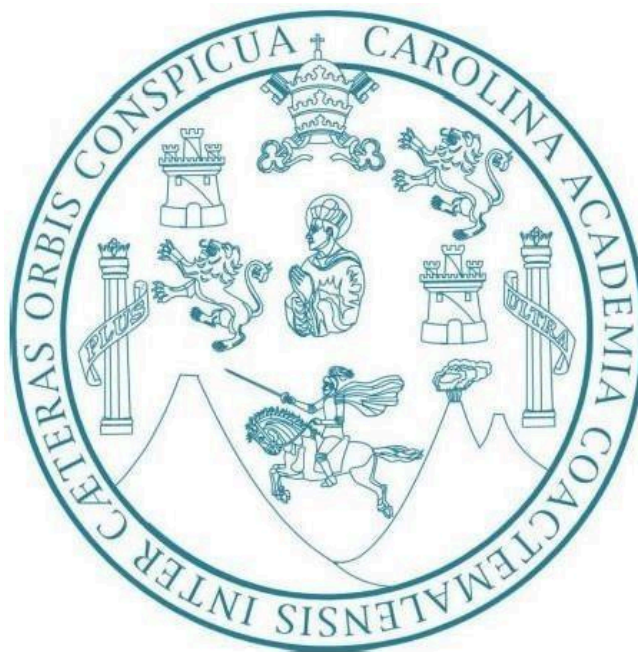
Escuela de Ciencias y Sistemas

Introducción a la Programación y Computación 1

Sección: A

Cat. Ing. Nefthali de Jesus Calderon Mendez

Tutor académico: Josué Rodolfo Morales Castillo



PRÁCTICA 2

Space Invaders

ÍNDICE

ÍNDICE.....	1
OBJETIVOS.....	2
❖ GENERALES.....	2
❖ ESPECÍFICOS.....	2
DESCRIPCIÓN GENERAL.....	3
DEFINICIÓN:.....	3
APLICACIÓN.....	4
PANTALLA PRINCIPAL.....	4
Nuevo juego:.....	4
Cargar juego:.....	4
Puntuación máxima:.....	4
Salir:.....	4
PANTALLA DE JUEGO.....	5
Indicadores:.....	5
Nave:.....	6
Enemigos:.....	6
Tipos de enemigos:.....	6
Ítems:.....	7
Tipos de ítems:.....	7
Fin del juego:.....	8
PANTALLA CARGAR JUEGO.....	8
Carga de un juego serializado:.....	8
PANTALLA PUNTUACIÓN MÁXIMA.....	9
Top 5 mejores scores:.....	9
REQUERIMIENTOS PARA EL DESARROLLO DEL PROYECTO.....	10
❖ LIBRERÍAS PERMITIDAS:.....	10
❖ DOCUMENTACIÓN:.....	10
❖ RESTRICCIONES:.....	11
❖ HABILIDADES POR EVALUAR:.....	11
❖ ENTREGA:.....	12

OBJETIVOS

❖ GENERALES

- Aplicar conocimientos de programación orientada a objetos en el lenguaje de programación Java.
- Comprender el concepto de hilo o subproceso aplicándolo en distintas funcionalidades del juego.
- Que el estudiante aplique los conocimientos adquiridos en el curso de Introducción a la Programación y computación 1.
- Elaborar la lógica para presentar una solución a la propuesta planteada.

❖ ESPECÍFICOS

- Utilizar el lenguaje de programación Java como herramienta de desarrollo de software.
- Construcción de aplicaciones usando interfaces gráficas en Java.
- Crear animaciones por medio de la implementación y manipulación de subprocesos

DESCRIPCIÓN GENERAL

DEFINICIÓN:

Space Invaders es un juego en dos dimensiones controlado por tiempo en el cual el jugador controla una nave cuyo objetivo es eliminar hordas de invasores espaciales que aparecen en la pantalla. La nave puede moverse únicamente en dos direcciones: hacia arriba o abajo. El jugador debe destruir a los extraterrestres invasores que se acercan a la nave a medida que pasa el tiempo. El juego cuenta con una serie de ítems que son catalogados como potenciadores y debilitadores. Si uno de estos ítems toca la nave del jugador, ocasionará un efecto positivo o negativo en la nave. Si los invasores llegan a tocar la nave controlada por el jugador o si el contador de tiempo llega a 0 el juego termina.

APLICACIÓN

La aplicación tendrá las siguientes pantallas:

PANTALLA PRINCIPAL

La pantalla principal debe contar con las siguientes opciones:



Nuevo juego:

Inicia un juego nuevo con las configuraciones por defecto.

Cargar juego:

Da la opción de abrir un juego guardado previamente.

Puntuación máxima:

Abre una lista con los nombres y puntuaciones de los 5 jugadores con las mejores puntuaciones.

Salir:

Cierra la aplicación.

PANTALLA DE JUEGO



Indicadores:

- **Contador de puntos:** Registra el total de puntos del jugador. Los puntos del jugador pueden verse afectados por ítems o por destruir enemigos, esto se explica más a detalle en la sección “ENEMIGOS”. El valor por defecto del contador de puntos siempre que se cree un nuevo juego será **0**.
- **Temporizador:** Es una cuenta regresiva que indica qué tanto se extiende la partida. El temporizador puede ser afectado por ítems, esto se explica más a detalle en la sección “ITEMS”. El valor por defecto del temporizador siempre que se cree un nuevo juego será de **90 segundos**.

Si el jugador desea salir del juego, puede hacerlo presionando la tecla ESC, no se guardará nada al presionar esta tecla, simplemente cerrará el juego actual y mostrará la “PANTALLA PRINCIPAL”.

Si el jugador desea guardar el estado actual de su juego entonces deberá presionar la tecla “S”, esta automáticamente serializará todo lo que el estudiante crea necesario para guardar el estado actual del juego y los archivos que se vayan serializando deberán ser guardados en una carpeta dentro de su proyecto que se llame “Juegos”, en esta carpeta se deben ir guardando los archivos con el siguiente formato: hora_min_dia_mes_año.bin, Ej: 13_05_13_06_2024.bin.

Nave:

La nave del jugador debe encontrarse posicionada en la parte izquierda de la pantalla. La nave únicamente puede moverse en el eje vertical, respetando los límites del área de juego. Para mover la nave el jugador debe usar las flechas arriba y abajo del teclado (↑ ↓). La nave puede realizar disparos con su cañón. Para poder disparar el cañón, el jugador debe presionar la barra espaciadora del teclado. El hitbox de la nave debe ser del mismo tamaño que el de los enemigos. La velocidad con la que se mueve la nave y sus disparos queda a discreción del estudiante, pero la velocidad debe ser la misma para ambos.

Enemigos:

Existen tres tipos de enemigos que inician en una formación de 8 filas por 5 columnas, separadas por tipo de enemigo, en el siguiente orden: tipo 1, tipo 2 y tipo 3. Deben moverse desde la parte derecha de la pantalla hacia la nave del jugador que se encuentra en la parte izquierda de la pantalla, mientras se van moviendo de arriba hacia abajo respetando los límites del área de juego. Cada tipo de enemigo cuenta con atributos propios como la salud (cantidad de disparos necesarios para eliminarlo) y puntos (cantidad de puntos que el jugador gana al eliminarlo). Cuando un enemigo es eliminado, su hitbox desaparece y su ícono debe cambiar por el ícono de una explosión por 0,5 segundos y luego desaparecer.

Tipos de enemigos:

- ❖ Tipo 1:
 - Salud: 2 disparos
 - Puntos: 10 puntos
 - Cantidad: 1 columna
- ❖ Tipo 2:
 - Salud: 3 disparos
 - Puntos: 20 puntos
 - Cantidad: 2 columnas
- ❖ Tipo 3:
 - Salud: 4 disparos
 - Puntos: 30 puntos
 - Cantidad: 2 columnas

PRÁCTICA 2 - IPC1

Cabe mencionar que no se pueden destruir a todos los enemigos de tipo 2 sin haber destruido antes a todos los enemigos de tipo 1, porque todos los enemigos comparten el mismo tamaño de hitbox, lo mismo pasa con los enemigos de tipo 3, no pueden ser destruidos todos si no se han destruido anteriormente los del tipo 1 y 2.

Ítems:

Los ítems afectan la jugabilidad, son activados cuando chocan contra la nave del jugador. Estos deben aparecer en posiciones aleatorias del eje vertical y moverse sobre el eje horizontal hacia donde se encuentra la nave del jugador. El tipo de ítem que aparece debe ser aleatorio. El hitbox de todos los ítems debe ser del mismo tamaño. La velocidad con la que se mueven los ítems y frecuencia de aparición quedan a criterio del estudiante.

Tipos de ítems:

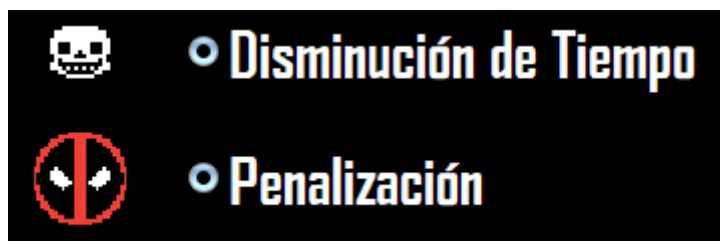
❖ Potenciadores:

- Aumento de tiempo: Agrega 10 segundos al temporizador.
- Puntos extra: Agrega 10 puntos al contador de puntos.



❖ Debilitadores:

- Disminución de tiempo: Quita 10 segundos al temporizador.
- Penalización: Resta 10 puntos al contador de puntos



Fin del juego:

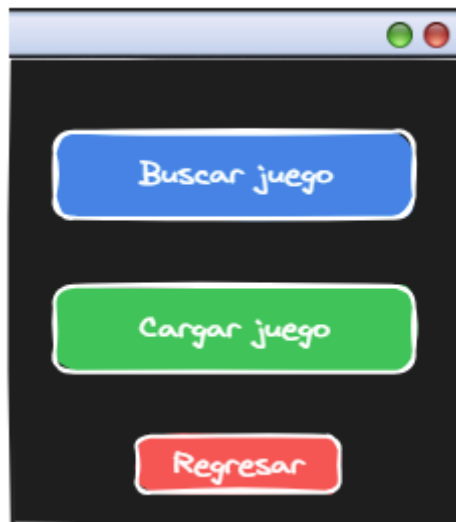
Al terminar el juego, se debe solicitar el nombre del jugador y guardarlo junto con su puntuación. El juego termina cuando:

1. El jugador destruye a todos los enemigos.
2. La nave del jugador es impactada por un enemigo.
3. Si el tiempo termina.

PANTALLA CARGAR JUEGO

Carga de un juego serializado:

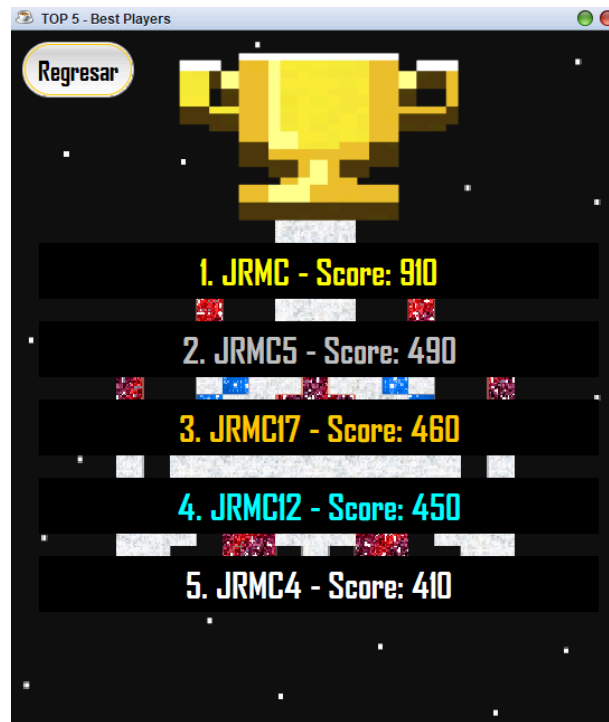
En esta pantalla se deben implementar dos botones, uno el cual abra un JFileChooser el cual solamente permita seleccionar archivos .bin, por medio de este botón se seleccionará el archivo serializado de un juego guardado anteriormente, luego de haber seleccionado el archivo se tendrá el otro botón el cual cargará la partida, cerrando la ventana actual y abriendo la “PANTALLA DE JUEGO” la cual mostrará el estado de los objetos guardados de una partida.



PANTALLA PUNTUACIÓN MÁXIMA

Top 5 mejores scores:

En esta ventana se debe observar los 5 mejores scores junto con el nombre del usuario que lo obtuvo, se deben mostrar de manera descendente.



REQUERIMIENTOS PARA EL DESARROLLO DEL PROYECTO

❖ LIBRERÍAS PERMITIDAS:

➤ Para el desarrollo de este proyecto, tiene permitido el uso de las siguientes librerías:

- AWT
- SWING
- Librerías IO para la lectura y escritura de archivos
- JFileChooser
- ArrayList, LinkedList
- Si tiene duda del uso de alguna otra librería, consultar primero al auxiliar por medio de los medios oficiales de comunicación.

❖ DOCUMENTACIÓN:

- Diagrama de clases
- Manual técnico
- Manual de usuario
- Para los manuales (técnico y de usuario) seguir la plantilla la cual está en Uedi en material de apoyo. Para el manual técnico deberá insertar imágenes como mínimo del número de línea y el nombre de su método, procedimiento o función, variables globales, librerías utilizadas y su respectiva descripción. Para el manual de usuario deberá obligatoriamente adjuntar imágenes y su respectiva descripción de cómo funciona su programa, qué ventanas muestra, resultados, etc...

❖ RESTRICCIONES:

- La aplicación debe ser desarrollada en el lenguaje de programación JAVA
- No se permite utilizar código copiado o bajado de Internet, ni ayuda entre compañeros, el trabajo es **individual**.
- El IDE por utilizar queda a discreción del estudiante (se recomienda el uso de NetBeans)
- Copias obtendrán una nota de 0 y reporte a la Escuela de Ciencias y Sistemas.
- Para realizar la interfaz gráfica puede hacer uso únicamente de las librerías AWT, Swing o Drag and drop.
- Durante la calificación se le solicitará al estudiante modificar el código del proyecto con el objetivo de validar la creación de este.
- Su repositorio debe de estar **privado**.
- Cualquier librería que quiera implementar debe de consultarlo primero con el auxiliar.
- El estudiante no tendrá derecho a calificación si no presenta interfaz gráfica, no se calificará ninguna funcionalidad en consola.
- El estudiante no tendrá derecho a calificación si no maneja su código utilizando un repositorio de github.
- Se calificará únicamente desde el .jar que envíen.
- Debe adjuntar su código fuente en el repositorio, junto con el jar y la documentación.

❖ HABILIDADES POR EVALUAR:

- Uso de variables globales y locales.
- Uso de memoria estática y dinámica.
- Uso de estructuras de control y de selección.
- Uso correcto de los arreglos.
- Conocimientos sobre sistemas computacionales.
- Habilidad para analizar y sintetizar información.
- La habilidad de comprender y realizar diagramas.
- Habilidad para resolver problemas.

- Capacidad de crear interfaces gráficas de usuario.

❖ ENTREGA:

- **FECHA DE ENTREGA:** 20/06/2024 antes de las **13:00** horas (No se aceptarán entregas, ni commits a partir de esa fecha y hora)
- Adjuntar el código fuente, el ejecutable de su aplicación (.jar) y la documentación en el repositorio privado de GitHub con el siguiente formato: [IPC1-A]Practica2_carnet, por ejemplo: [IPC1-A]Practica2_202010033.
- Subir el enlace del repositorio de GitHub en la tarea asignada en UEDI.
- Agregar de colaborador al repositorio al auxiliar: RMorales202010033