

Luis Miguel Ossa Arias - A00369982
Santiago Trochez Velasco - A00369326

Aliens Invaders

Aliens invaders es un juego que consiste en eliminar a todos los alienígenas que provienen del espacio, donde el jugador es una nave que se encarga de disparar a dichos invasores.

A medida que los aliens son eliminados, estos van bajando con el objetivo de disparar a la nave o de llegar al punto que se encuentra protegiendo la nave para derrotar a esta, por ello, el jugador tiene que evitar ser golpeado por los proyectiles que disparan los aliens y también evitar que estos lleguen a su punto seguro.

Si el jugador es impactado por un proyectil, perderá el juego, pero si este derrota a todos los aliens, aparecerá nuevos con el mismo objetivo.

Mientras más aliens sean destruidos, más puntos obtendrá el jugador.

Cuando el juego termine, el jugador ingresará su nombre para que quede en el “hall of fame” de mejores protectores espaciales. Además, los 5 mejores jugadores se encontrarán en el podio, que por otro lado los 5 peores quedarán marcados en la sala de los fails.

Al momento en que se vayan destruyendo a más alienígenas, la dificultad del juego irá aumentando, lo que significa que los alienígenas se moverán más rápido que el nivel anterior.

Requerimientos funcionales

El juego debe estar en la capacidad de:

Req 1. Mostrar al usuario en la pantalla de inicio tres opciones.

Req 1.1. Dejar que el usuario seleccione la opción de iniciar una nueva partida.

Req 1.2. Permitir que el usuario seleccione la opción de visualizar los puntajes de usuarios anteriores que hayan jugado.

Req 1.3. Permitir que el usuario seleccione la opción de visualizar el podio de los 5 mejores jugadores.

Req 2. Permitir que el usuario se mueva de izquierda a derecha con las respectivas flechas (“←” y “→”) o cuando presione a y d.

Req 3. Dejar que el usuario dispare contra los alienígenas invasores cuando este pulse la flecha hacia arriba o la letra w.

Req 4. Eliminar a un alienígena si este es impactado por un proyectil de la nave espacial.

Req 5. Eliminar a la nave si esta es impactada por un proyectil de los alienígenas o si los alienígenas llegan al punto seguro de la nave, el cual es la posición en donde la nave se mantiene.

Req 6. Permitir al usuario ingresar su nickname al momento de terminar una partida, de esta forma el puntaje será guardado con su respectivo usuario.

Req 7. Aumentar el nivel y la velocidad del juego cuando el usuario destruye a todos los invasores de su respectivo nivel.

Req 8. Mostrar al usuario el texto de juego terminado inmediatamente la nave sea destruida.

Req 9. Mostrar los usuarios con sus respectivos niveles y puntajes cuando el usuario elija la opción de observar el salón de la fama.

Req 10. Permitir que el usuario exporte todos los jugadores del hall of fame en un archivo de texto.

Req 11. Permitir que el usuario importe a otros jugadores con puntaje y nivel alcanzados. En otras palabras, jugadores del mismo juego pero que podrían venir desde otro dispositivo.

Req 12. Dejar que el usuario ingrese su nombre antes de empezar una partida.

Req 13. Permitir que el usuario elimine a un jugador de la tabla de hall of fame proporcionando un nickname y puntaje respectivo.

Req 14. Posibilitar que el usuario busque a un jugador de la tabla de hall of fame, ya sea por su nickname o su puntaje.

Req 15. Permitir que el usuario visualice a los 5 mejores jugadores y los 5 peores jugadores.

Req 16. Dejar que el usuario organice la lista de jugadores.

Req 16.1 Permitir al usuario organizar la lista por puntaje.

Req 16.2 Permitir al usuario organizar la lista por nickname / puntaje.

Req 16.3 Permitir al usuario organizar la lista por nivel / punjate.

Req 16.4 Permitir al usuario organizar la lista por nickname / nivel.

Justificación:

El juego Aliens Invaders requiere mucho tiempo, ya que para poder desarrollar este juego se necesita una gran parte algorítmica y de diseño que tienen cierto nivel de dificultad. Dentro de la parte algorítmica encontramos temas como recursividad, concurrencia, pruebas unitarias y excepciones, las cuales requieren de un alto pensamiento lógico y analítico para poder llevarse a cabo de manera adecuada. Además de programar instrucciones lógicas de funcionamiento, se necesita que estas instrucciones se relacionen con la interfaz gráfica de usuario. Al ser Aliens Invaders un juego, se necesita que el usuario interactúe de manera fluida con controles básicos para poder tener una buena experiencia de juego. Teniendo esto en cuenta, se necesita crear una interfaz gráfica de usuario, donde esta sea intuitiva y llamativa para que el juego no se vea plano o aburrido.

Mockups

INICIO:



El botón **iniciar partida** nos dirigirá a una nueva partida.

El botón **scores** nos dirigirá al salón de la fama del juego, donde se muestran los puntajes de los anteriores juegos con el nombre del jugador, su puntuación y el nivel al que llegó.

El botón **podio** visualizar a los 5 mejores jugadores.

INGRESAR NOMBRE REAL:



Esta pantalla permite guardar el nombre real del jugador para almacenarlo en el documento, para después mostrarlo con el nickname del jugador.

El botón **continuar** permite guardar el nombre de la persona.

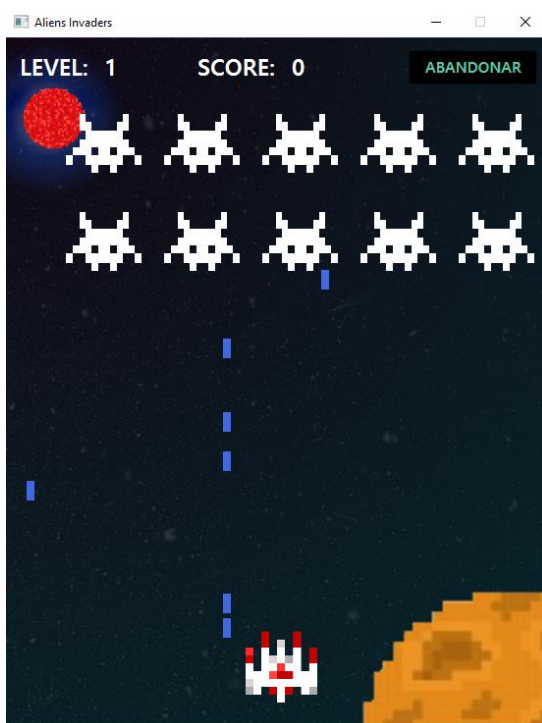
El botón **atrás** hace regresar el programa a la pantalla de inicio.

SELECCIONAR NAVE:



Permite seleccionar la dificultad del juego, además se puede presionar cualquiera de las dos naves para jugar con alguna de ellas.

PARTIDA:



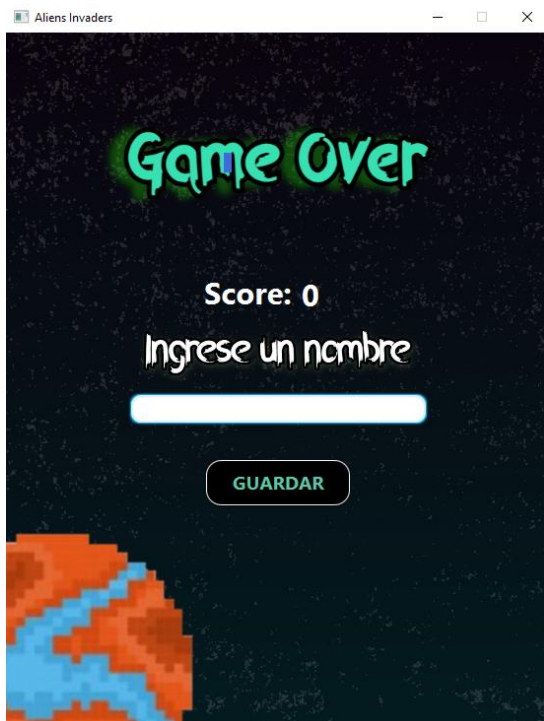
La nave podrá disparar hacia los objetivos, en este caso los aliens y también se puede desplazar hacia los lados.

Los aliens podrán disparar contra la nave, además todos los invasores irán bajando.

Habrà un label que irá sumando los aliens destruidos.

Cada alien otorga cinco punto.

GAME OVER:



Hay un label que te informa cuál fue tu puntaje del juego.

Hay un textfield que permite ingresar el nombre del usuario para así guardarlo en el salón de la fama.

Hay un botón con nombre guardar que guarda el nickname del usuario y lo llevará a el salón de la fama.

SALÓN DE LA FAMA:



Está una tabla con espacios para mostrar el nombre de cada usuario con su respectivo puntaje y nivel.

Hay un botón llamado inicio que llevará al usuario a la ventana principal inicio.

El botón importar permite agregar jugadores de un archivo.

El botón exportar permite sacar los jugadores en el sistema en un archivo de texto.

El botón buscar o eliminar permite ingresar a la pantalla buscar jugador.

BUSCAR UN JUGADOR:



Permite buscar a un jugador por su nickname o su puntaje, además también se puede eliminar a un jugador ingresando el nickname y el puntaje, y presionando el botón eliminar.

PODIO:



El podio permite visualizar a los 5 mejores jugadores, además se puede hacer clic en el título “podium” para ingresar a la pantalla de los 5 peores jugadores.

SALÓN DE LOS FAILS:



Esta pantalla permite visualizar los 5 peores jugadores, si se quiere volver a la pantalla inicio puede hacer clic en el botón **inicio**.

Requerimientos no funcionales

Req 1. Utilizar hilos para mejorar el rendimiento del programa.

Req 1.1. Utilizar un hilo para la interfaz gráfica del programa.

Req 1.2. Utilizar un hilo para dividir funciones y operaciones al momento de realizar un ordenamiento de burbuja, el cual toma bastante tiempo en ser ejecutado.

Req 1.3. Utilizar un hilo para dividir funciones y operaciones al momento de realizar un ordenamiento por selección, debido a que necesita tiempo para procesar cierta cantidad de datos.

Req 1.4. Utilizar un hilo para hacer que los alienígenas se muevan. Req 1.51 Utilizar un hilo para hacer que la nave se mueva.

Req 2. Implementar recursividad para realizar el movimiento uniforme y repetitivo de los aliens, a los cuales el jugador deberá dispararles.

Req 3. Usar las pruebas unitarias para revisar cada función del programa y evitar errores planteando situaciones que podrían generarse al momento de ejecutar la aplicación.

Req 4. Emplear la organización de datos mediante el método de burbuja.

Req 5. Emplear la organización de datos mediante el método de selección.

Req 6. Utilizar serialización de clases para guardar objetos de tipo jugador, o, mejor dicho, a un jugador con su respectivo puntaje.

Req 7. Implementar serialización de texto plano para guardar información de la aplicación, específicamente los niveles.

Req 8. Leer archivos importados o exportados.

Req 9. Escribir la información de los archivos, ya sean de texto plano de clases, para exportar o importar datos.

Req 10. Utilizar excepciones personalizadas.

Req 10.1. Utilizar excepciones para cuando se ingresen nombres con espacios al momento de registrar un jugador.

Req 10.2. Implementar excepciones para cuando no se ingrese un nombre al momento de acabar una partida.

Req 10.3. Utilizar excepciones para cuando el puntaje supera el valor de 999.

Req 11. Implementar herencia para compartir método y datos entre clases, subclases y objetos.

Req 11.1. Utilizar herencia para el usuario cuando este ingresa su nombre real y su nickName.

Req 11.2. Utilizar herencia para los tipos de niveles.

Req 11.3. Utilizar herencia para las figuras primitivas. Estas serán los proyectiles, tanto de la nave espacial como de los alienígenas.

Req 11.4. Implementar herencia para las naves que serán escogidas en el juego. Cada una tendrá un atributo diferente resaltante respecto a la otra.

Req 12. Emplear interfaces.


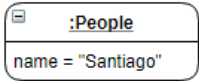
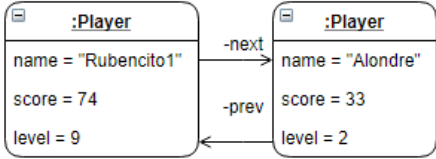
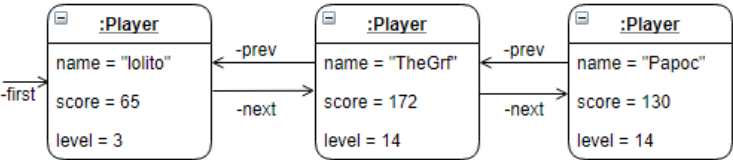
Req 12.1. Hacer un interfaz de búsqueda ya sea por nombre o puntaje.

Req 12.2. Utilizar una interfaz para calcular el puntaje de un jugador dependiendo de la acción realizada, como puede ser destruir un alien o subir de nivel.

Req 12.3. Implementar una interfaz para comparar dos objetos.

CASOS DE PRUEBA

Configuración:

Nombre	Clase	Escenario
setupScenary1	AliensInvadersTest	 <pre> graph LR A[] -- "-first" --> B[:Player] </pre>
setupScenary2	AliensInvadersTest	 <pre> classDiagram class People { name = "Santiago" } </pre>
setupScenary3	AliensInvadersTest	 <pre> classDiagram class Player1 { name = "Rubencito1" score = 74 level = 9 } class Player2 { name = "Alondre" score = 33 level = 2 } Player1 --> Player2 : -next Player2 --> Player1 : -prev </pre>
setupScenary4	AliensInvadersTest	 <pre> classDiagram class Player1 { name = "lolito" score = 65 level = 3 } class Player2 { name = "TheGrf" score = 172 level = 14 } class Player3 { name = "Papoc" score = 130 level = 14 } Player1 --> Player2 : -next Player2 --> Player1 : -prev Player2 --> Player3 : -next Player3 --> Player2 : -prev A[] --> Player1 : -first </pre>

Diseño de casos de prueba:

Objetivo de la prueba: Validar que la búsqueda de un jugador por nombre en el hall of fame retorna un objeto de tipo Player cuando este ya ha sido creado, y null cuando no se encuentra en la lista enlazada.

Clase	Método	Escenario	Valores de entrada	Resultado
-------	--------	-----------	--------------------	-----------

Objetivo de la prueba: Validar que se agrega correctamente una persona a la lista enlazada de people.

Clase	Método	Escenario	Valores de Entrada	Resultado
AliensInvaders	addPeople	setupScenary1	name = “Carlos José”	true Se agregó correctamente el nombre.
AliensInvaders	addPeople	setupScenary1	name = “Santiago”	true Se agregó correctamente el nombre.

Objetivo de la prueba: Validar que NO se pueda crear una persona si esta tiene un número en su nombre.

Clase	Método	Escenario	Valores de entrada	Resultado
AliensInvaders	addPeople	setupScenary1	name = “Jesús666”	false No se pudo agregar a la persona porque su nombre contiene números.
AliensInvaders	addPeople	setupScenary1	name= “Ana 2”	false No se pudo agregar a la persona porque su nombre contiene números

Objetivo de la prueba: Validar que se agrega correctamente un jugador a la lista enlazada de jugadores.

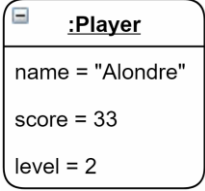
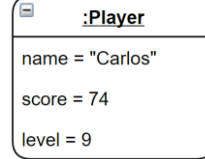
Clase	Método	Escenario	Valores de entrada	Resultado
AliensInvaders	addPlayer	setupScenary2	nick= "Jesús666" score = 62 level = 4	true El jugador se ha agregado correctamente a la lista de jugadores.
AliensInvaders	addPlayer	setupScenary2	nick= "Ana2" score = 89 level = 6	true El jugador se ha agregado correctamente a la lista de jugadores.

Objetivo de la prueba: Validar que NO se puede agregar un jugador si el nickname contiene espacios.

Clase	Método	Escenario	Valores de entrada	Resultado
AliensInvaders	addPlayer	setupScenary2	nick= "Aloha 74" score = 89 level = 6	false El jugador no se ha podido agregar a la lista de jugadores, ya que el nickname contiene espacios.
AliensInvaders	addPlayer	setupScenary2	nick= "michi xd" score = 89 level = 6	false El jugador no se ha podido agregar a la lista de jugadores, ya que el nickname contiene espacios

Objetivo de la prueba: Validar que se ha podido encontrar a un jugador gracias a su puntaje y retornando un objeto de tipo Player. Si ingresa un score no existente, se retorna null.

Clase	Método	Escenario	Valores de entrada	Resultado
-------	--------	-----------	--------------------	-----------

AliensInvaders	binarySearch	setupScenary3	score = 66	 <pre> :Player name = "Alondre" score = 33 level = 2 </pre>
AliensInvaders	binarySearch	setupScenary3	score = 66	null
AliensInvaders	binarySearch	setupScenary3	name= Carlos	 <pre> :Player name = "Carlos" score = 74 level = 9 </pre>

Objetivo de la prueba: Validar que se elimina correctamente un people de la lista enlazada.

Clase	Método	Escenario	Valores de entrada	Resultado
AliensInvaders	removePlayer	setupScenary4	nick = "lolito" score = 65 level = 3	True La persona con el nick lolito se ha eliminado de la lista enlazada. La lista enlazada ahora tiene un tamaño igual a 2.
AliensInvaders	removePlayer	setupScenary4	nick = "Papoc" score = 130 level = 14	True La persona con el nick Papocse ha eliminado de la lista enlazada. La lista enlazada ahora tiene un tamaño igual a 2.

Objetivo de la prueba: Validar que el score no sobrepase los 999 puntos.

Clase	Método	Escenario	Valores de entrada	Resultado
-------	--------	-----------	--------------------	-----------

AliensInvaders	calculate	setupScenary1	score = 920	true El score es correcto
AliensInvaders	calculate	setupScenary1	score = 80	true El score es correcto

