

Materia	Programación III	Carrera	ING. EN SISTEMAS COMPUTACIONALES
Profesor	Eduardo Serna-Pérez	Semestre	4
PROYECTO FINAL			
Nombre del Proyecto:	El alumno selecciona alguno	Fecha de Entrega:	
Observaciones:	La entrega del proyecto será a partir de la hora especificada, el profesor asignará sesiones de 20 minutos para revisar cada proyecto de manera que en el cubículo aparecerá un registro de llegada para cada equipo.		
Alumnos por equipo	2 personas, salvo excepciones		

Instrucciones Generales:

Deberá entregar el código fuente en la plataforma de Aula Virtual. Además de un manual de usuario que muestre su funcionamiento y todos aquellos documentos que crea necesarios como investigación adicional.

Entregar PDF en el aula virtual de la documentación con lo siguiente:

- Portada
- Resumen descriptivo
- Temas investigados para la realización del proyecto.
- Bitácora de trabajo.
- Conclusiones
- Referencias consultadas

Proyecto copiado de Internet o de alguna otra fuente causará anulación de la calificación final del curso, así como proyectos con errores de compilación o que no se realice lo que se pide.

TEMAS A EVALUAR: Se evaluará que el proyecto no tenga errores de compilación y/o sintaxis, errores de ejecución. Además, debe cumplir con todos los temas que marca el programa de las materias de Estructuras de Datos y Programación II.

DOCUMENTACION

El documento incluye:

1. Portada
 - Imagen con título de la UAA
 - Título del proyecto
 - Nombre de la materia
 - Nombres de los integrantes
 - Carrera, semestre y grupo
 - Nombre del profesor
 - Fecha de entrega
2. Resumen descriptivo
 - Fortalezas
 - Debilidades

Estos puntos los deben redactar con sus propias palabras, indiquen que fortalezas tiene su proyecto y cuales debilidades, justifiquen sus comentarios.
3. Temas Investigados para la realización del proyecto

4. Bitácora de trabajo
Hacer una tabla donde incluyan el nombre del integrante del equipo, actividades realizadas y tiempo (días) que se llevó en realizar dicha actividad (sean honestos y escriban lo que hizo cada uno)
5. Conclusiones
Con sus palabras expliquen de forma breve que fue lo que aprendieron con el desarrollo del proyecto.
6. Bibliografía o Referencias consultadas
Escribir la bibliografía que fue consultada para realizar el proyecto (libros, páginas de Internet, etc.) Recuerden que un formato adecuado para citar fuentes de libros, revistas, páginas Web, periódicos, etc. es el formato APA (American Psychological Association)

CODIGO

- Todo programa debe ser implementado en programación orientada a objetos.
- Debe emplear métodos y hacer uso de parámetros.
- No debe utilizar variables globales a menos que se justifique su uso (solo puede usar de forma global las estructuras y archivos).
- Debe aplicar todos los temas vistos en el curso de ser necesario.
- Utilizar los menús necesarios para acceder a cada parte del programa.
- Validar las capturas (que no existan errores de ejecución, es decir, que no “true” el programa).
- Cada sección de código del programa debe estar perfectamente bien comentada, es decir, cada método debe tener comentarios acerca de lo que se realiza.
- Los nombres de los métodos y atributos deben ser de acuerdo al dato que almacenan o la acción que realizan

Proyectos: Cada equipo tiene que seleccionar un problema en particular de los siguientes

Problema 1.



El nombre del juego proviene de la onomatopeya japonesa paku (パク?), sonido que se produce al abrir y cerrar la boca. El nombre se romanizó como Puck-Man en Japón (en inglés, puck se pronuncia pac en español). Sin embargo, Midway lo modificó a Pac-Man para el mercado estadounidense (y posteriormente otros mercados occidentales) debido a que la gente podría cambiar la «P» de Puck-Man, por una «F», formando una palabra soez en inglés.

El protagonista del videojuego Pac-Man es un círculo amarillo al que le falta un sector, por lo que parece tener boca. Aparece en laberintos donde debe comer puntos pequeños «Pac-dots», puntos mayores y otros premios con forma de frutas y otros objetos. El objetivo del personaje es comer todos los puntos de la pantalla, momento en el que se pasa al siguiente nivel o pantalla. Sin embargo, cuatro fantasmas, Blinky, Pinky, Inky y Clyde, recorren el laberinto para intentar capturar a Pac-Man. Estos fantasmas son, respectivamente, de colores rojo, rosa, azul y naranja. Los fantasmas no son iguales, así mientras Blinky es muy rápido, y tiene la habilidad de encontrar a Pac-Man en el escenario, Inky es muy lento y muchas veces evitará el encuentro con Pac-Man.

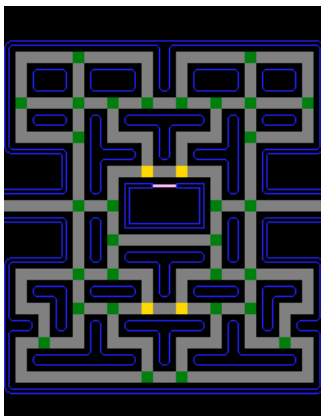
Instrucciones Generales:

1. Modo de juego Clásico:

Hay un "pasillo" a los costados del laberinto que permiten a Pac-Man o sus enemigos transportarse al costado opuesto. Cuatro puntos más grandes de lo normal situados cerca de las esquinas del laberinto nombrados en inglés «Power Pellets», proporcionan a Pac-Man, durante un tiempo limitado, la habilidad de comerse a los monstruos (todos ellos se vuelven azules mientras Pac-Man tiene esa habilidad), tras lo cual todo vuelve a ser como al principio.



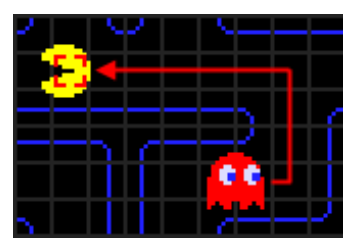
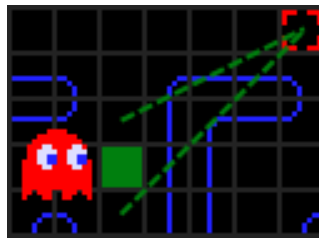
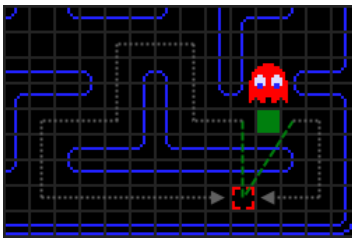
Después de haber sido comidos por Pac-Man, los fantasmas se regeneran en «casa» (una caja situada en el centro del laberinto). El tiempo en que los monstruos permanecen vulnerables varía según la pantalla, pero tiende a decrecer a medida que progresa el juego, y al cabo de muchas pantallas los puntos especiales no tienen ningún efecto sobre los fantasmas.



Los Fantasmas están limitados por el laberinto de la misma manera que Pac-Man, pero por lo general se mueven ligeramente más rápido que el jugador. Pac-Man se ralentiza ligeramente mientras come puntos, permitiendo potencialmente a un fantasma perseguidor que le capture durante esos momentos.

Los fantasmas cuentan con puntos de control donde pueden hacer cambios a su comportamiento, como decidir ubicar a Pac-Man en una nueva posición o decidir recorrer el mapa en otra dirección

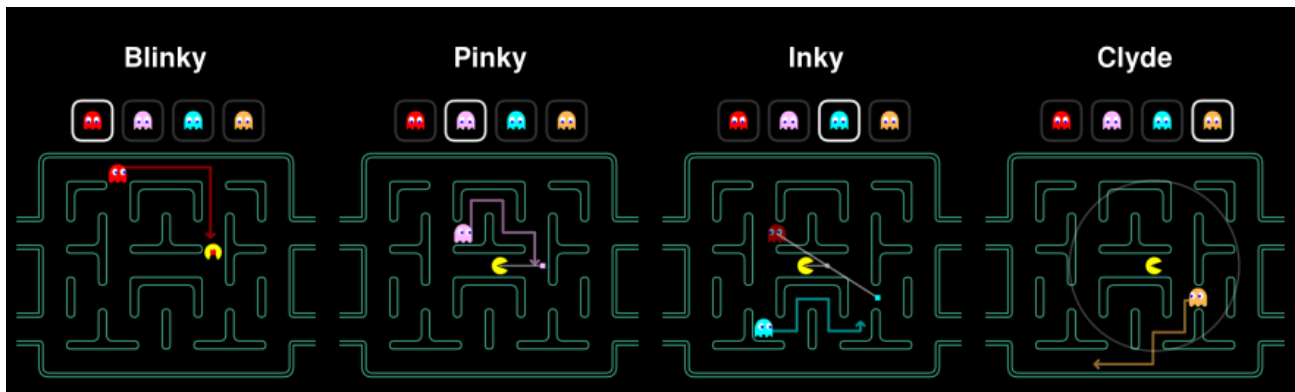
Blinky: Después de que Pac-Man coma cierta cantidad de puntos, su velocidad incrementa considerablemente, además es capaz de determinar la ubicación de Pac-Man y perseguirlo. Cuando es necesaria una decisión sobre qué dirección girar, la elección se realiza en función de qué mosaico contiguo a la intersección colocará al fantasma más cercano a su mosaico objetivo, medido en línea recta. Se mide la distancia desde cada posibilidad al mosaico objetivo, y se seleccionará el mosaico más cercano al objetivo.



Pinky: Rodea los obstáculos al contrario de las manecillas del reloj. Esta suele colocarse en frente de Pac-Man a 4 casillas y cortar el paso u orbita alrededor de Pac-Man para confundir y distraer al jugador para que uno de sus compañeros lo atrape, pero su esquema de objetivos intenta moverlo al lugar donde va Pac-Man, en lugar de donde está actualmente. El mosaico objetivo de Pinky se determina mirando la posición y orientación actuales de Pac-Man, y funciona cuando Pac-Man está mirando hacia la izquierda, hacia abajo o hacia la derecha.

Inky: No es tan rápido como Blinky pero actúa de manera errática así que es difícil predecir cómo va a reaccionar.

Clyde: Él no persigue a Pac-Man, sino que deambula por el laberinto sin una ruta específica. Se podría decir que incluso no lo quiere evitar.



El modo "Asustado" ocurre cuando Pac-Man come un energizador dentro del laberinto. Hay cuatro energizadores ubicados en el laberinto y los cuatro fantasmas cambian de modo. Los fantasmas se vuelven azul oscuro y deambulan por el laberinto siendo vulnerables. Parpadearán momentos antes de volver al modo normal.

2. Consideraciones técnicas:

- Implementar los conceptos de POO aprendidos en el curso de programación (abstracción, encapsulación, herencia y polimorfismo)
- Implementar alguna(s) estructura de datos en la solución del proyecto.
- Manejo de archivos de texto o binarios para registrar la puntuación obtenida (**jugador (alias), puntuación, fecha**). Para el mismo jugador se debe permitir acumular puntos y el juego debe tener alguna opción para mostrar el reporte de jugadores ordenado de mayor a menor puntuación.
- Manejo de 3 niveles, verificar en el juego original las diferencias entre los niveles (considera que hay varias versiones de este juego revísalas).

*Es necesario redactar por equipo un documento que indique el modo de operación que programaran en cada nivel **es decir el alcance (reglas del juego)**, este documento debe ser entregado al profesor el día 1 de diciembre y será considerado el día de la revisión.*

- Manejo de sonidos (opcional música)
- El IDE para programar el juego se deja a consideración del equipo, siempre y cuando sea un IDE para programar en lenguaje Java. En el entendido de que, elegir alguno diferente al utilizado en clase requiere tiempo de investigación por parte del equipo. Y de continuar con el IDE utilizado en clase también requiere investigación y pruebas para incorporar el manejo de gráficos.*

- Sobre la calificación:** El proyecto representa el 20% de la calificación final y es requisito entregarlo para tener calificación de la materia.

Puntos por evaluar durante la revisión:

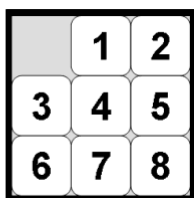
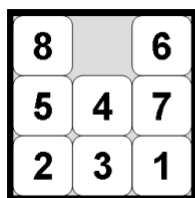
- Gráficos del juego, sonidos (si fue integrado).
- Funcionalidad del juego. Manejo de 3 niveles. Cubrir requisitos de cada nivel en base a documento entregado al profesor.

3. Uso de estructuras de datos (pilas, colas, listas, arboles, grafos, arreglos, etc.)
4. Uso de Programación Orientada a Objetos.
5. Uso de archivos de texto o binarios para guardar información sobre puntuaciones obtenidas.
6. Reporte que incluya portada y una explicación de la forma en que se organizó y realizó la programación, clases utilizadas, donde se utilizó herencia y/o polimorfismo, estructuras de datos utilizadas y en donde, una conclusión por equipo sobre la experiencia de hacer este proyecto, así como un informe donde se indique con detalle la participación que tuvo cada integrante durante el desarrollo del sistema.
7. Entregar en aula virtual el código fuente y un archivo que indique instrucciones de como instalar el juego y/o como ejecutarlo (requerimientos del sistema), así como los integrantes del equipo.

Notas importantes:

- Es requisito la presencia de todos los integrantes del equipo durante la revisión, entendiendo que el integrante que no se presente, no tiene calificación por concepto de proyecto final.
- Todos los documentos y programas fuente deben estar presentes en aula virtual antes de su presentación física el día de la revisión.
- La revisión del proyecto se realizará el 16 de diciembre de 2019.

Problema 2.



El juego 8-puzzle es una versión reducida del 15-puzzle, un juego conocido en muchas partes del mundo, aunque no siempre se conoce por ese nombre. El juego está formado por una caja cuadrada con 8 piezas móviles, también cuadradas, numeradas entre 1 y 8. El objetivo del juego es ordenar las piezas de 1 a 8 realizando el desplazamiento de una pieza a la vez, utilizando el único espacio libre disponible. Las piezas no pueden sacarse de la caja así que no es posible ordenarlas de cualquier forma.

Instrucciones Generales:

1. Generalidades:

Se requiere construir un programa que permita al usuario jugar en dos modos: manual donde el jugador desplaza las piezas, e inteligente, donde la cuadrícula comienza de un punto aleatorio y el jugador determina cual será la solución deseada (meta).

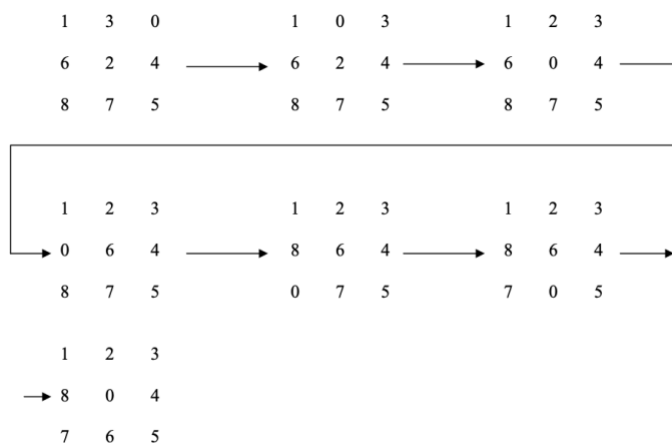
En el modo manual el usuario debe poder:

1. Mover las piezas del puzzle-8 según las reglas del juego.
2. Crear un nuevo juego de manera aleatoria
3. Una opción donde el programa pueda sugerir un movimiento.

En modo inteligente el usuario debe poder:

1. Determinar el punto de inicio
2. Determinar cuál será la meta por solucionar.
3. Visualizar la ejecución paso a paso de la solución del juego, en caso de que exista el árbol de soluciones.

Para realizar un movimiento, el usuario debe hacer clic sobre una de las piezas y, si esta se encuentra al lado del espacio disponible, se mueve a dicha posición. Para sugerir una jugada se utilizará un algoritmo muy sencillo: a partir del estado actual del juego se genera un árbol con los estados alcanzables.



La complejidad del juego hace que el número de nodos en el árbol sea exponencial y cada nivel adicional en el árbol multiplica el número total de nodos por casi 2 (en un árbol de altura 20 puede haber cerca de 900,000 nodos). Es por esto que la altura del árbol que se quiere generar debe ser controlado por el usuario. Luego de tener construido el árbol se busca el nodo alcanzable que está más cerca de la solución y la jugada sugerida es la raíz del árbol que lleva ese nodo. Todo esto se basa en un valor que se le da a cada estado y que indica qué tan cerca está ese estado al final del juego.

Técnica de Ramificación y Poda

En el algoritmo de **Branch and bound** se suele interpretar en un árbol de soluciones, donde cada rama nos lleva a una posible solución posterior a la actual. La característica de esta técnica con respecto a otras anteriores (y a la que debe su nombre) es que el algoritmo se encarga de detectar en qué ramificación las soluciones dadas ya no están siendo óptimas, para «podar» esa rama del árbol y no continuar malgastando recursos y procesos en casos que se alejan de la solución óptima (costo).

2. Consideraciones técnicas:

- Implementar los conceptos de POO aprendidos en el curso de programación (abstracción, encapsulación, herencia y polimorfismo)
- Implementar alguna(s) estructura de datos en la solución del proyecto.
- Manejo de archivos de texto o binarios para registrar la puntuación obtenida (**jugador (alias), puntuación, fecha**). Para el mismo jugador se debe permitir acumular puntos y el juego debe tener alguna opción para mostrar el reporte de jugadores ordenado de mayor a menor puntuación.
- Manejo de 3 niveles, verificar en el juego original las diferencias entre los niveles (considera que hay varias versiones de este juego revísalas).

*Es necesario redactar por equipo un documento que indique el modo de operación que programaran en cada nivel **es decir el alcance (reglas del juego)**, este documento debe ser entregado al profesor el día 1 de diciembre y será considerado el día de la revisión.*

- Manejo de sonidos (opcional música)
- El IDE para programar el juego se deja a consideración del equipo, siempre y cuando sea un IDE para programar en lenguaje Java. En el entendido de que, elegir alguno diferente al utilizado en clase requiere tiempo de investigación por parte del equipo. Y de continuar con el IDE utilizado en clase también requiere investigación y pruebas para incorporar el manejo de gráficos.*

- Sobre la calificación:** El proyecto representa el 20% de la calificación final y es requisito entregarlo para tener calificación de la materia.

Puntos por evaluar durante la revisión:

1. Gráficos del juego, sonidos (si fue integrado).
2. Funcionalidad del juego. Manejo de 2 Modos. Cubrir requisitos de cada modo en base a documento autorizado por el profesor.
3. Uso de estructuras de datos (pilas, colas, listas, arboles, grafos, arreglos, etc.)
4. Uso de Programación Orientada a Objetos.
5. Uso de archivos de texto o binarios para guardar información sobre puntuaciones obtenidas.
6. Reporte que incluya portada y una explicación de la forma en que se organizó y realizó la programación, clases utilizadas, donde se utilizó herencia y polimorfismo, estructuras de datos utilizadas y en donde, una conclusión por equipo sobre la experiencia de hacer este proyecto, así como un informe donde se indique con detalle la participación que tuvo cada integrante durante el desarrollo del sistema.
7. Entregar en aula virtual el código fuente y un archivo que indique instrucciones de como instalar el juego y/o como ejecutarlo (requerimientos del sistema), así como los integrantes del equipo.

Notas importantes:

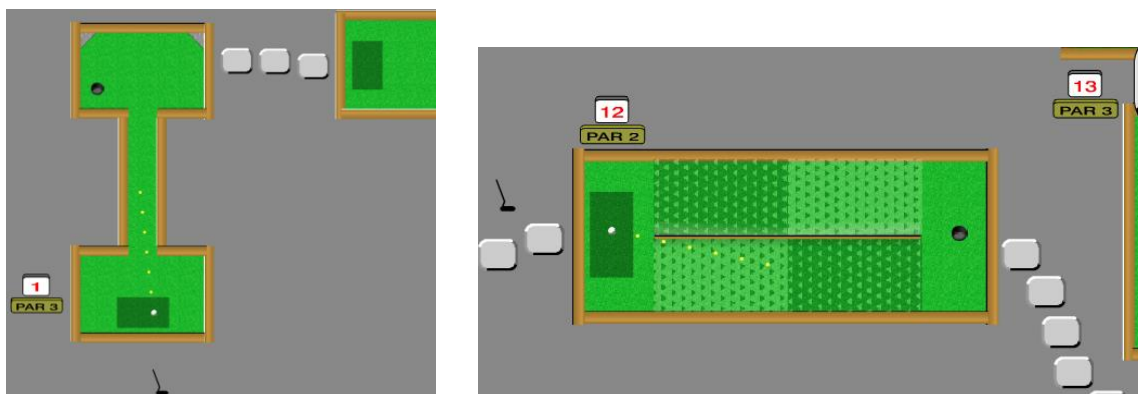
- Es requisito la presencia de todos los integrantes del equipo durante la revisión, entendiendo que el integrante que no se presente, no tiene calificación por concepto de proyecto final.
- Todos los documentos y programas fuente deben estar presentes en aula virtual antes de su presentación física el día de la revisión.
- La revisión del proyecto se realizará el 16 de diciembre de 2019.

Problema 3.

Un Mini-Golf es un juego entretenido que existe en muchos centros de diversión, la idea central es utilizar un palo de golf simple, una pelota y 18 diferentes áreas conocidas como hoyos, con una diferente cantidad de obstáculos distribuidos dentro de cada área.

Aspectos del juego

Se cuenta con 18 áreas diferentes, dentro de cada área existe un espacio específico donde comienza el juego y un pequeño punto del otro extremo, conocido como hoyo, dentro de este espacio la pelota puede rebotar y perder velocidad conforme avanza, además de existir una gran variedad de obstáculos y diversas superficies que provocaran que la pelota se comporte de diferente manera debido a la fricción.



Niveles

Solo se tienen un único nivel con 18 hoyos, y es posible jugarlo por más de una persona, de manera que para estos casos se deberá de intercalar las jugadas de cada participante cuando ha tocado la pelota sin lograr atinar al hoyo. Sucesivamente se puede tener una opción para cronometrar el juego y hacerlo más interesante.

Personajes

Solo existe un personaje al tiempo, aunque pueden existir obstáculos fijos y móviles.

Puntos importantes

La pelota tendrá una velocidad que será adquirida por la fuerza que se aplique al golpe del palo, de manera que debe de considerar este aspecto importante, además deberá de re calcular los cambios de ángulos. Es posible que el juego se pause y continúe, así como ver la puntuación hasta el momento del juego.

Algunas ligas que podrían resultar muy útiles

<http://www.mini-putt.org>

<http://www.jbox2d.org/>

<https://code.google.com/p/jbox2d/wiki/QuickStart>

<https://code.google.com/p/jbox2d/wiki/Testbed>

<http://mvnrepository.com/artifact/org.jbox2d/jbox2d-library/2.2.1.1>

<http://thisiswhatiknowabout.blogspot.mx/2011/12/jbox2d-tutorial.html>

Problema 4.

Asteroides es un popular juego de 1979, el objetivo del juego es desde una nave se tendrá que disparar y destruir asteroides de diverso tamaño evitando chocar con los fragmentos de éstos.

El objetivo principal del juego de asteroides es destruir asteroides y platillos. El jugador está al mando de una nave espacial triangular que puede girar hacia la izquierda y hacia la derecha, disparar hacia adelante y empujar hacia adelante. Una vez que la nave comienza a viajar en una dirección, continuará en esa ruta indefinidamente a menos que el jugador agregue empuje en una dirección diferente.

Cuando no está empujando, la nave finalmente se detiene. Es destruir asteroides y platillos. El jugador está al mando de una nave espacial triangular que puede girar hacia la izquierda y hacia la derecha, disparar hacia adelante y empujar hacia adelante. Una vez que la nave comienza a viajar en una dirección, continuará en esa ruta indefinidamente a menos que el jugador agregue empuje en una dirección diferente. Cuando no está empujando, la nave finalmente se detiene.



El control de la nave consiste en aceleración y botón de fuego, y giros en el sentido del reloj y, al contrario, para avanzar hay que ir en cualquier dirección y aceleras constantemente cuando llegas al borde de la pantalla apareces en el lado contrario, para desacelerar tienes que ir al lado contrario, hay también un botón de hiperspacio en te hace aparece al azar en la pantalla, con el riesgo de explotar (si te materializas dentro de un asteroide).

El jugador comienza en el medio de la pantalla, con cuatro grandes asteroides viajando a diferentes velocidades. Cada asteroide grande (20 puntos) disparado se divide en 2 partes de tamaño medio (50 puntos), y cuando les disparas se dividen en otros 2 más pequeños (100 puntos). Los asteroides medio y pequeños pueden viajar a diferentes velocidades.

Periódicamente aparece un ovni que dispara en direcciones al azar (100 puntos). Tiende aparecer cuando quedan pocos asteroides en la pantalla y/o el jugador no ha disparado recientemente a un asteroide.

Cada vez que pases un nivel se limpia la pantalla y aparecen más asteroides (de 2 en 2) hasta llegar un máximo de 12. El puntaje máximo posible es de 99.990 puntos, después de la cual da vuelta a cero. Para hacer un poco más interesante el juego, agregaremos una opción de campo de fuerza que pueda proteger a la nave por un lapso de 5 segundos.

[https://en-m-wikipedia-org.translate.goog/wiki/Asteroids_\(video_game\)?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc](https://en-m-wikipedia-org.translate.goog/wiki/Asteroids_(video_game)?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)