

PRÁCTICA FINAL PROGRAMACIÓN

DOCUMENTACIÓN



31 de mayo de 2019

BORJA SERRANO GARCÍA

**ÍNDICE**

**Manual de Usuario 2 - 20**

**Instrucciones de Ejecución e Instalación 21**

**Estructuras de Datos 22 - 27**

**Manual de Procedimientos y Funciones 28 - 42**

**MANUAL USUARIO**

CONTROLES

Controles Menús

Para manejarse en el menú es necesario el uso del ratón:

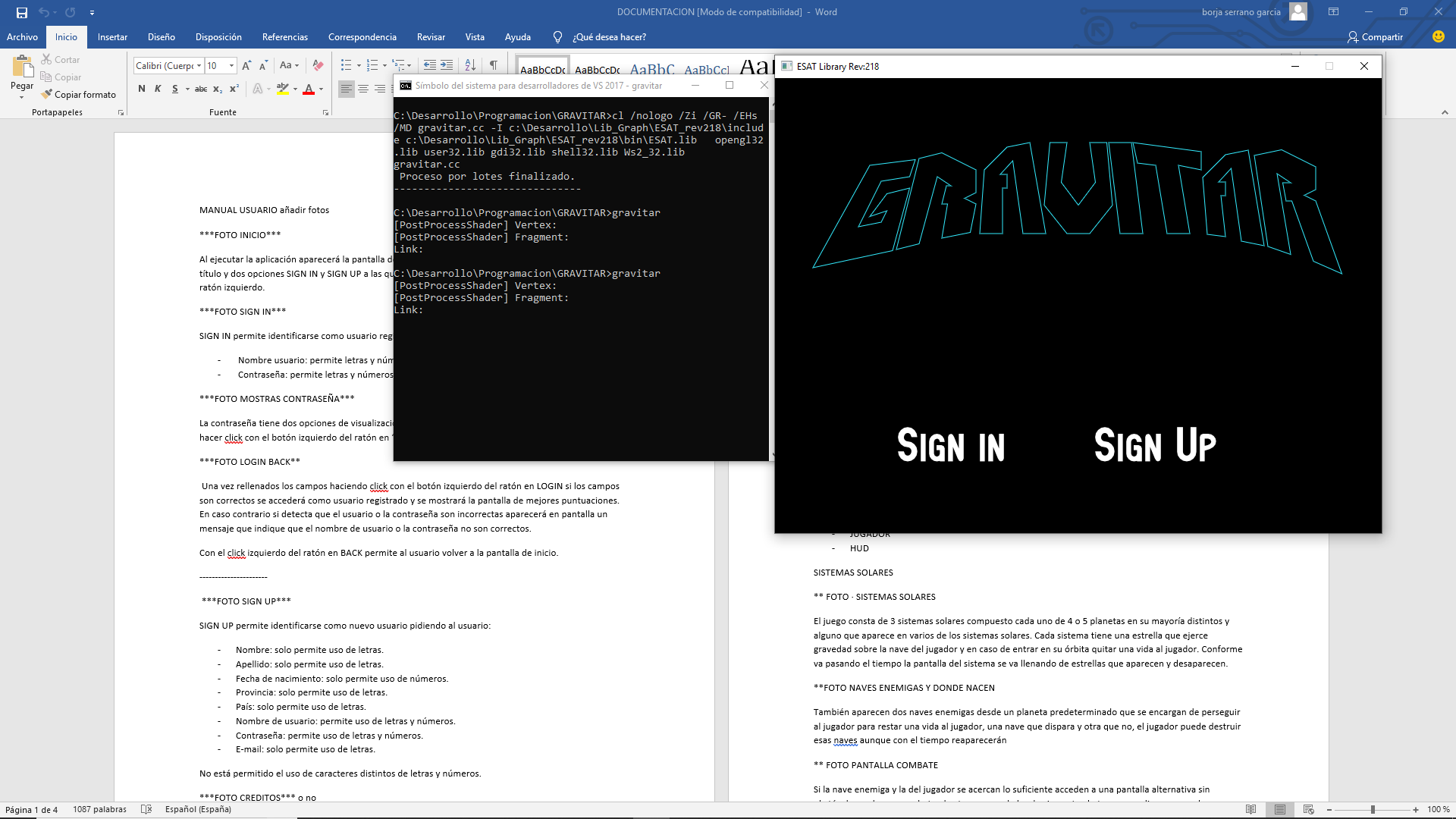
* Click Izquierdo : permite avanzar en los menús y activar los botones.

Controles Gameplay

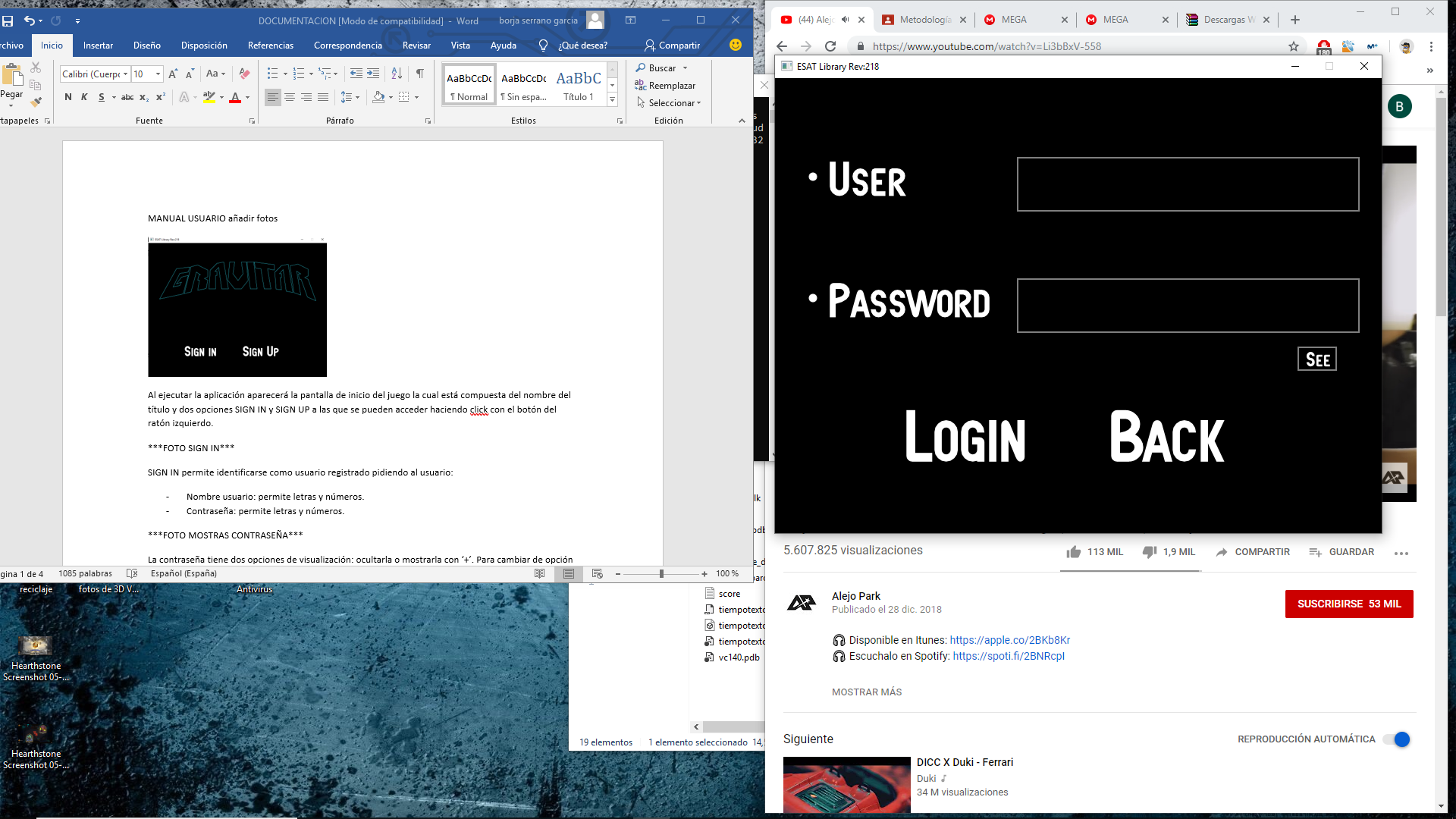
Para dirigir la nave se usan distintas teclas:

* Flecha Arriba: permite avanzar a la nave.
* Flecha Abajo: activa el modo escudo de la nave.
* Flecha Izquierda: rota la nave hacia la izquierda.
* Flecha Derecha: rota la nave hacia la derecha.
* Espacio: activa un disparo de la nave.

MENÚS

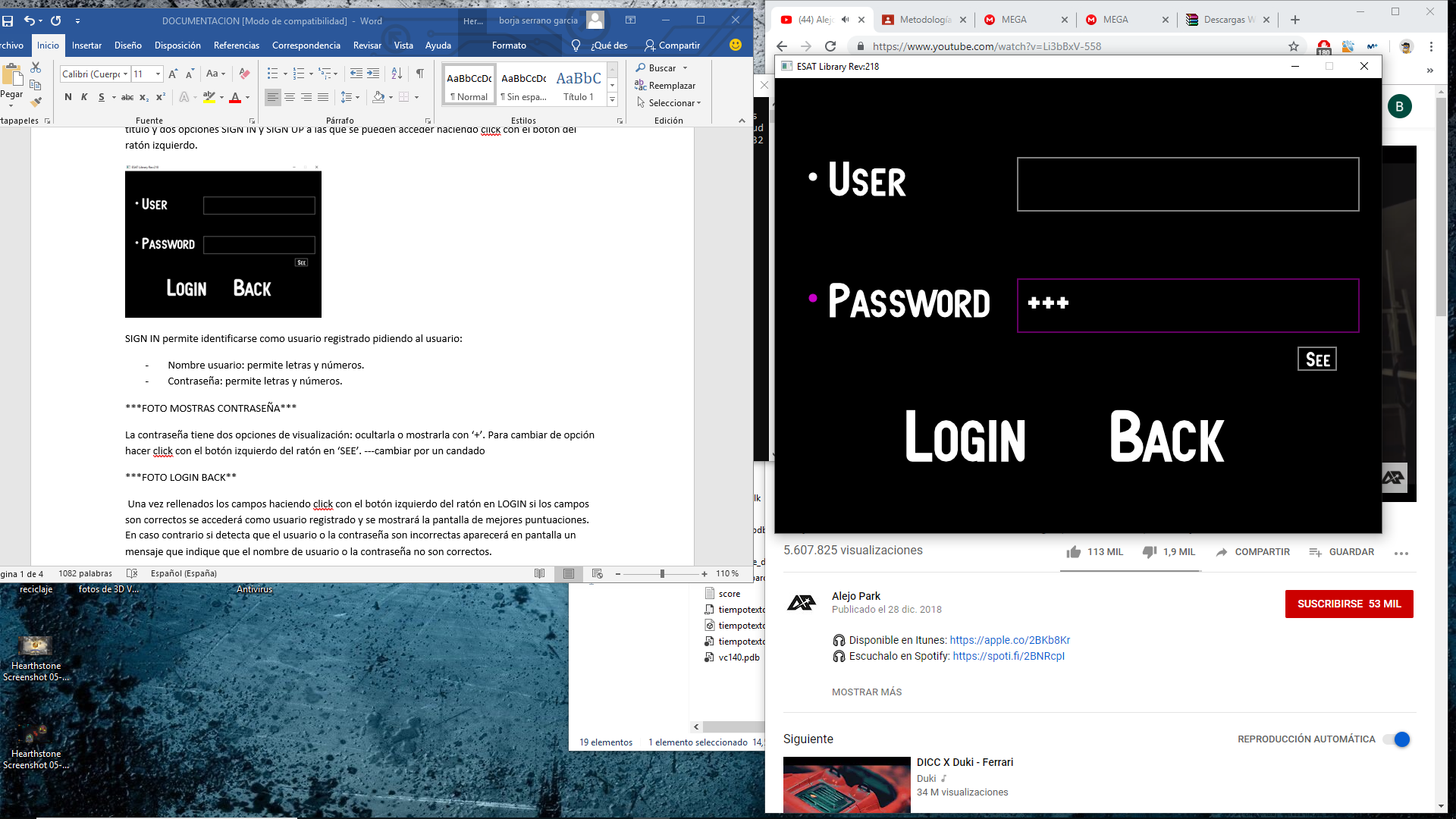


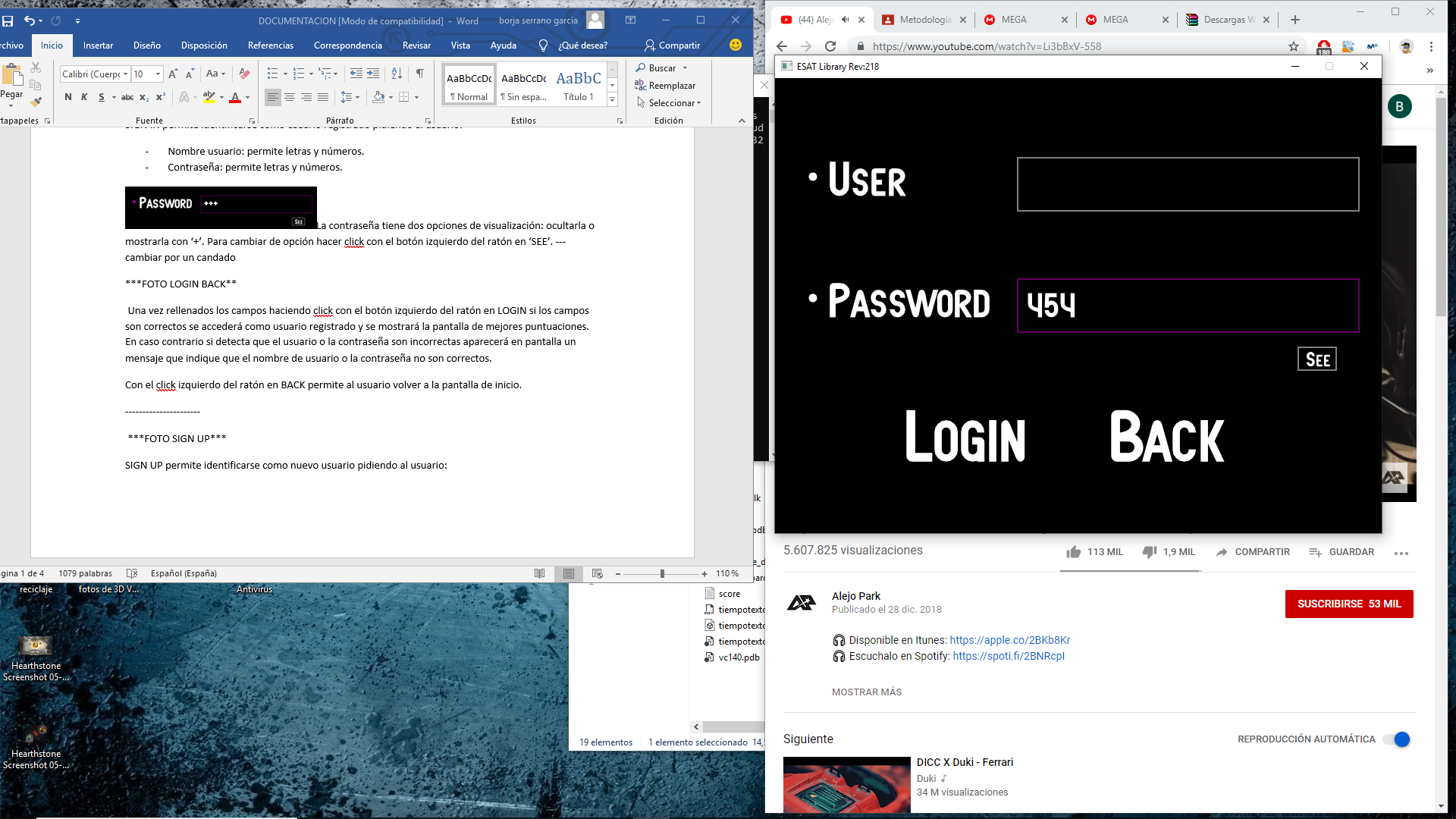
Al ejecutar la aplicación aparecerá la pantalla de inicio del juego la cual está compuesta del nombre del título y dos opciones SIGN IN y SIGN UP a las que se pueden acceder haciendo click con el botón del ratón izquierdo.



SIGN IN permite identificarse como usuario registrado pidiendo al usuario:

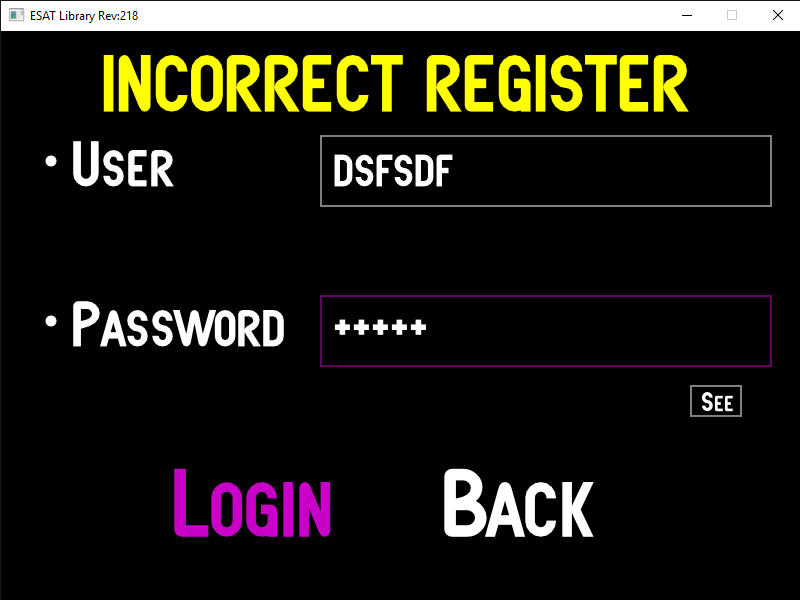
* Nombre usuario: permite letras y números.
* Contraseña: permite letras y números.





La contraseña tiene dos opciones de visualización: ocultarla o mostrarla con ‘+’. Para cambiar de opción hacer click con el botón izquierdo del ratón en ‘SEE’. ---cambiar por un candado

Una vez rellenados los campos haciendo click con el botón izquierdo del ratón en LOGIN si los campos son correctos se accederá como usuario registrado y se mostrará la pantalla de mejores puntuaciones. En caso contrario si detecta que el usuario o la contraseña son incorrectas aparecerá en pantalla un mensaje que indique que el nombre de usuario o la contraseña no son correctos.



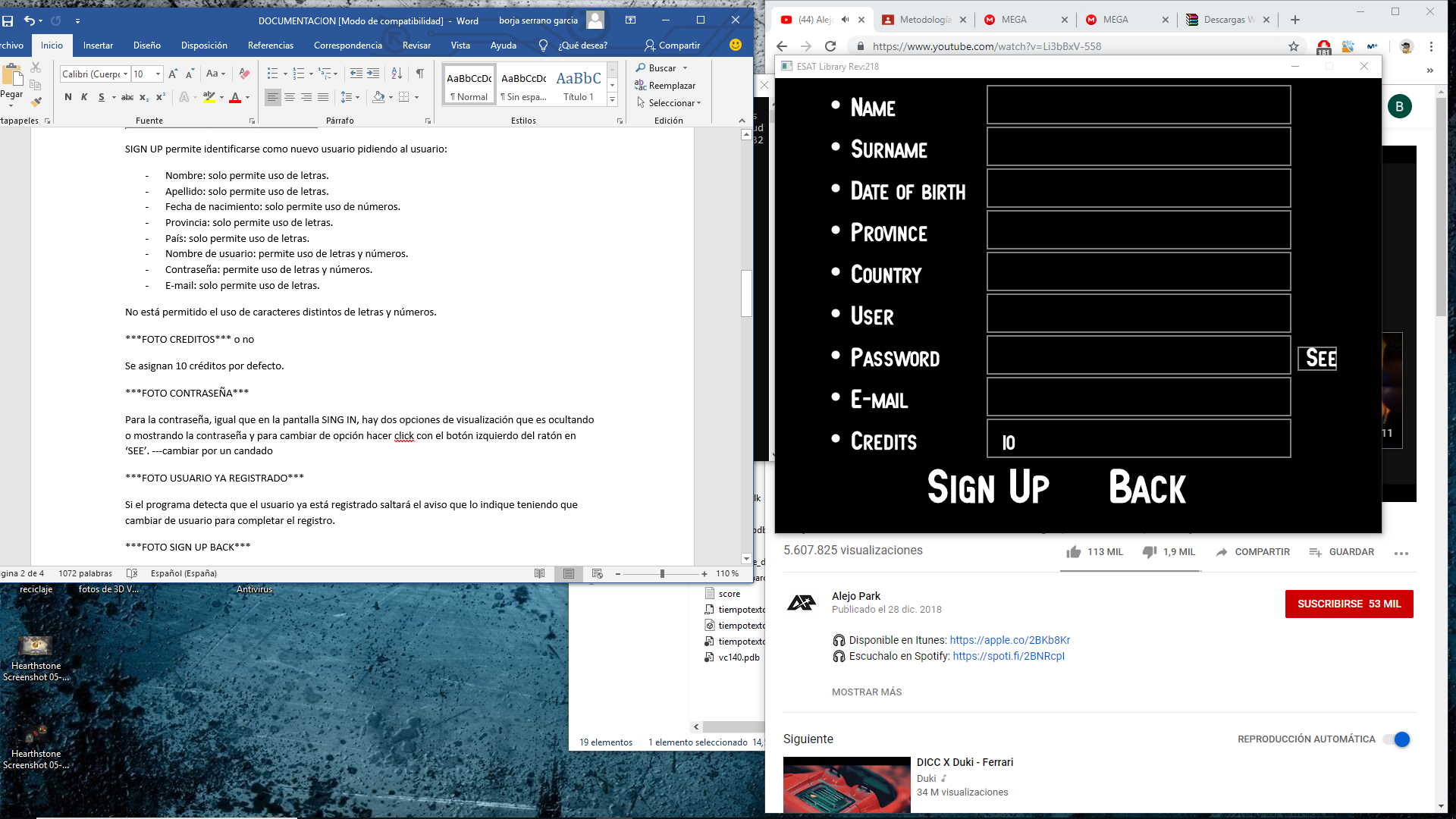
Con el click izquierdo del ratón en BACK permite al usuario volver a la pantalla de inicio.



SIGN UP permite identificarse como nuevo usuario pidiendo al usuario:

* Nombre: solo permite uso de letras.
* Apellido: solo permite uso de letras.
* Fecha de nacimiento: solo permite uso de números.
* Provincia: solo permite uso de letras.
* País: solo permite uso de letras.
* Nombre de usuario: permite uso de letras y números.
* Contraseña: permite uso de letras y números.
* E-mail: solo permite uso de letras.

No está permitido el uso de caracteres distintos de letras y números.



Se asignan 10 créditos por defecto.

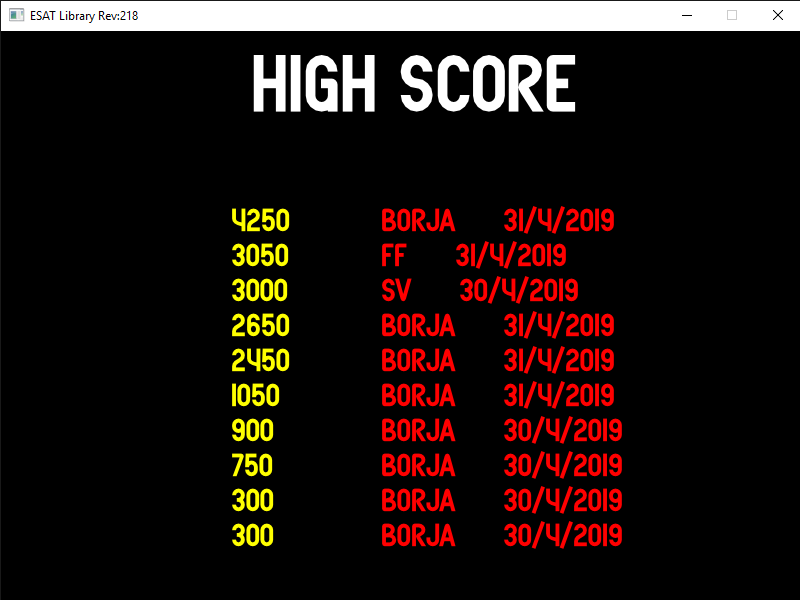
Para la contraseña, igual que en la pantalla SING IN, hay dos opciones de visualización que es ocultando o mostrando la contraseña y para cambiar de opción hacer click con el botón izquierdo del ratón en ‘SEE’. ---cambiar por un candado

Si el programa detecta que el usuario ya está registrado saltará el aviso que lo indique teniendo que cambiar de usuario para completar el registro.

Una vez rellenados los campos haciendo click con el botón izquierdo del ratón en SIGN UP si los campos son correctos se accederá como nuevo usuario registrado y se mostrará la pantalla de mejores puntuaciones. En caso contrario si detecta que el usuario o la contraseña son incorrectas aparecerá en pantalla un mensaje que indique que el nombre de usuario o ya está en uso.

Con el click izquierdo del ratón en BACK permite al usuario volver a la pantalla de inicio.

PANTALLA MEJORES PUNTUACIONES



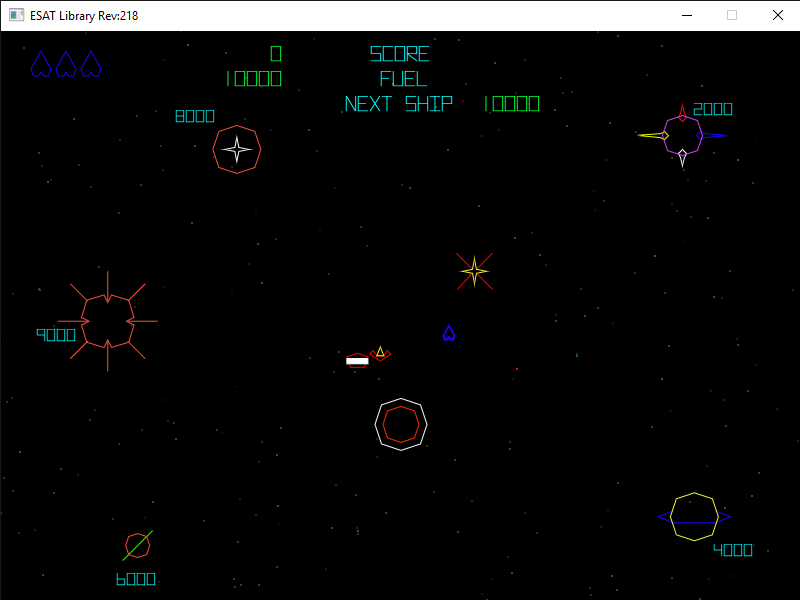
Se muestra el TOP 10 de mejores jugadores con los puntos que ha conseguido, la fecha y el nombre de usuario. Al cabo de unos segundos se accede directamente al gameplay del juego.

DENTRO DEL JUEGO

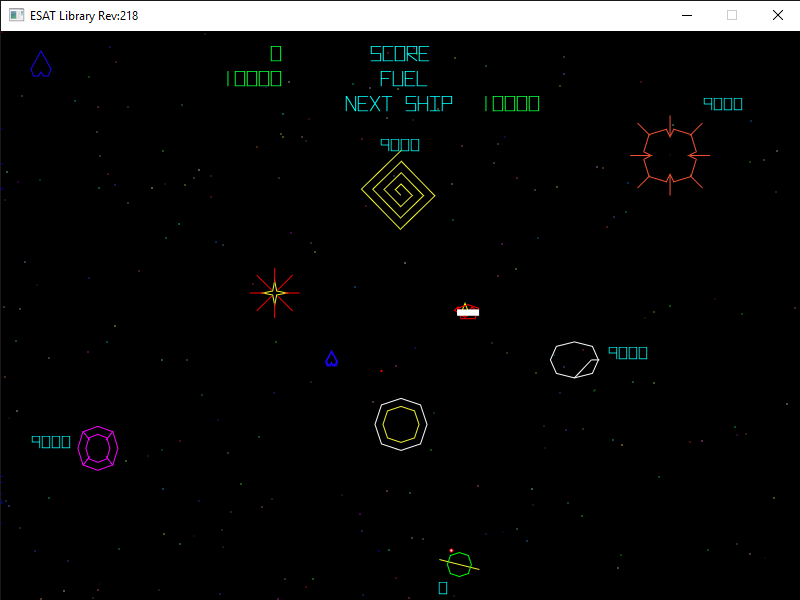
El juego consta de 3 sistemas solares compuesto cada uno de 4 o 5 planetas en su mayoría distintos y alguno que aparece en varios de los sistemas solares. Cada sistema tiene una estrella que ejerce gravedad sobre la nave del jugador y en caso de entrar en su órbita quitar una vida al jugador. Conforme va pasando el tiempo la pantalla del sistema se va llenando de estrellas que aparecen y desaparecen.

Para acceder a cada planeta diferente hay que entrar en la órbita de cada uno.

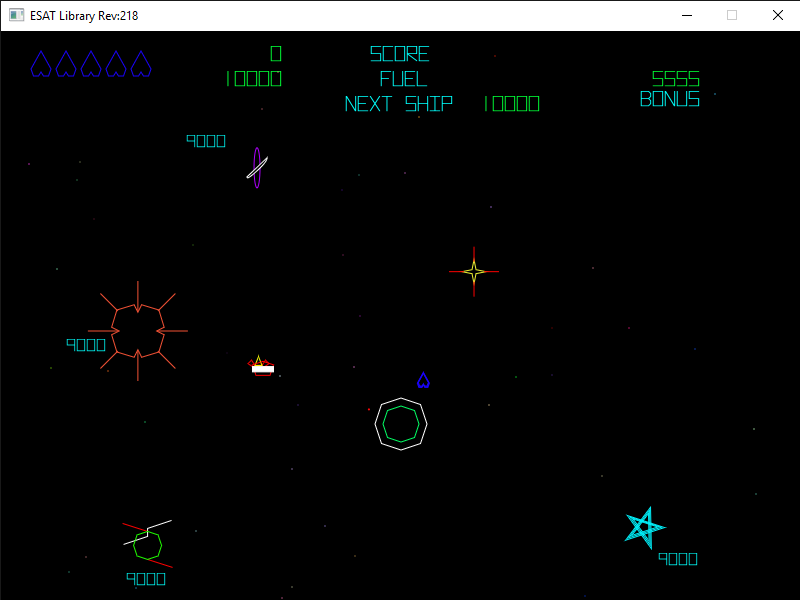
SISTEMA SOLAR 1



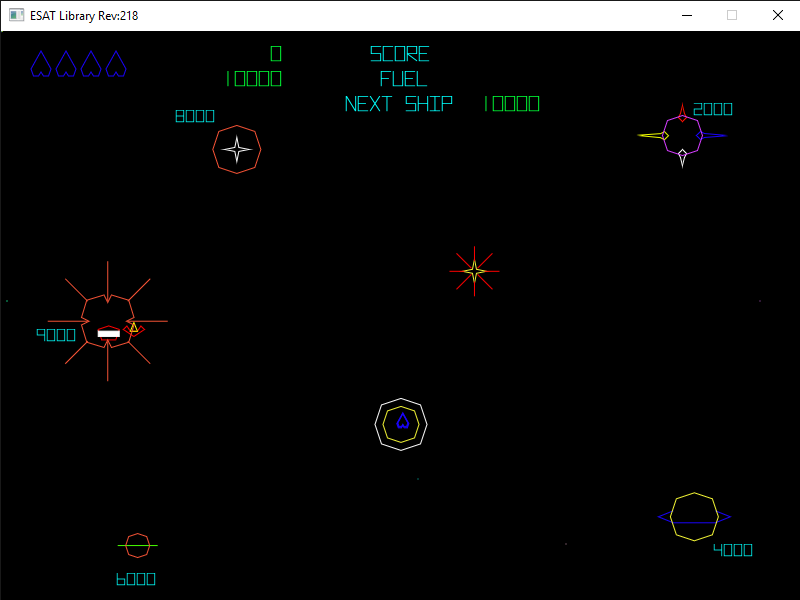
SISTEMA SOLAR 2



SISTEMA SOLAR 3

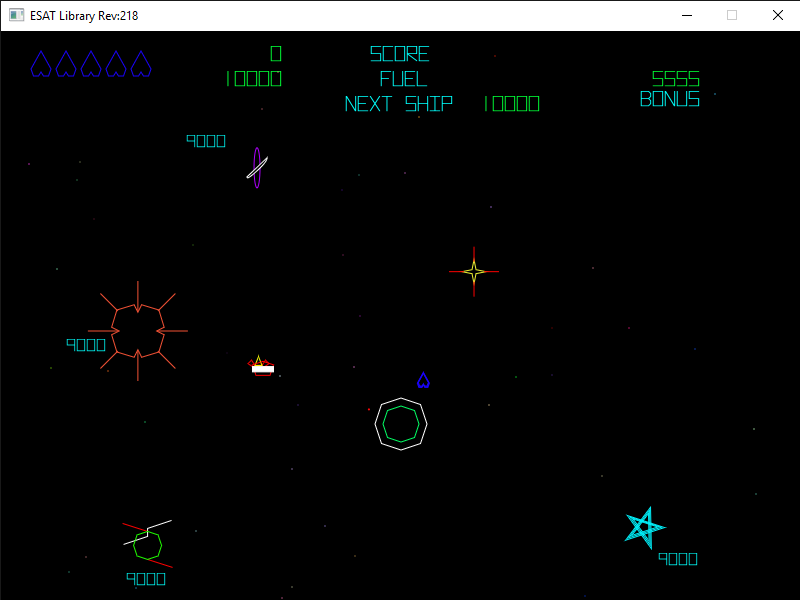


En cada sistema solar aparecen dos naves enemigas desde un planeta predeterminado que se encargan de perseguir al jugador para restar una vida al jugador, una nave que dispara y otra que no, el jugador puede destruir esas naves aunque con el tiempo reaparecerán. Destruir alguno de estos enemigos dota al jugador de 100 puntos.

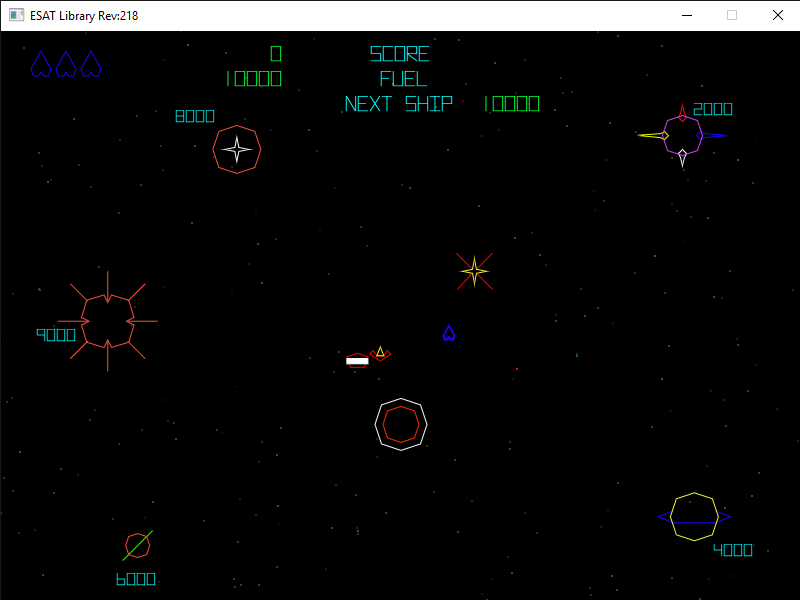




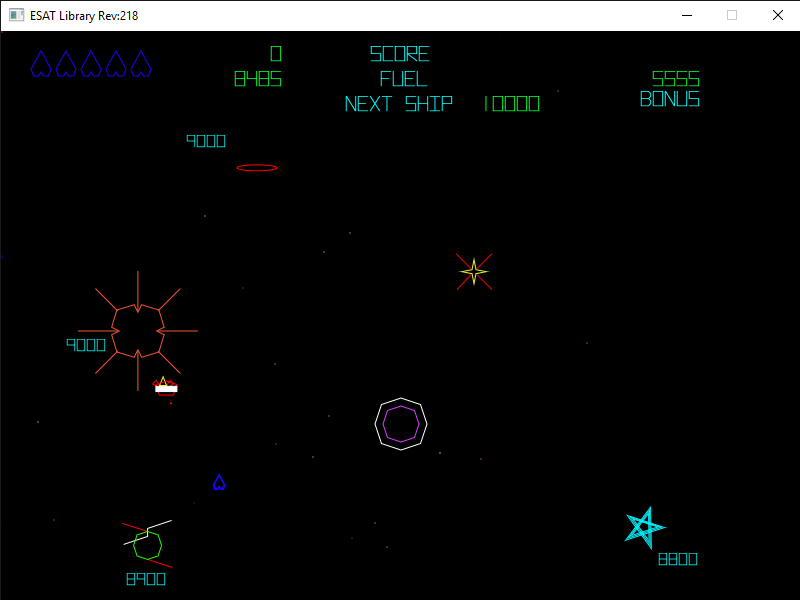
Si la nave enemiga y la del jugador se acercan lo suficiente acceden a una pantalla alternativa sin obstáculos en la que combaten hasta que una de las dos impacte al otro con un disparo o se choquen entre ellas.



Para acceder a cada planeta hay que entrar en su órbita y tras ser completado otorga la puntuación indicada justo al lado restándose cada 5 segundos que se pasa dentro del mapa 100 puntos



La nave del jugador comienza en cada sistema solar desde un hexágono situado prácticamente en el centro del sistema. La nave rebota con los bordes que limitan la pantalla.

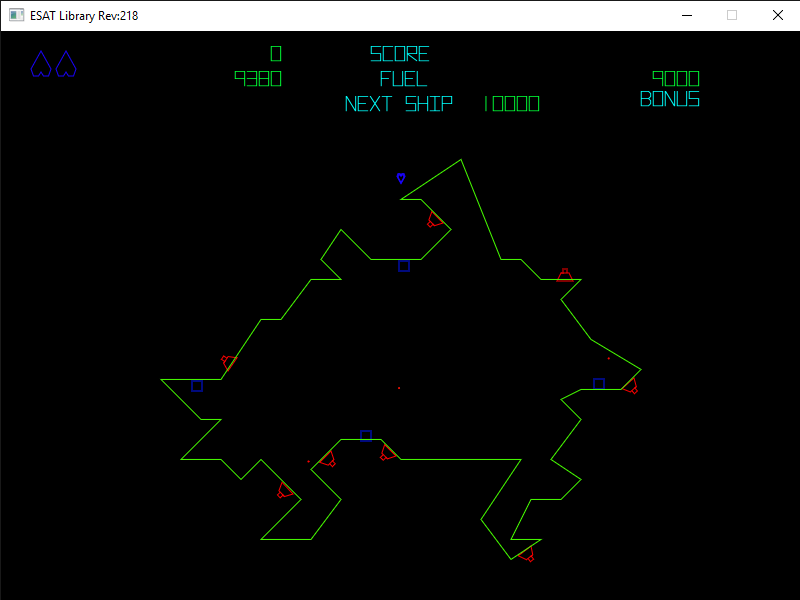


Esta es la estrella que ejrece la gravedad de cada sistema solar.

MAPAS DE PLANETAS

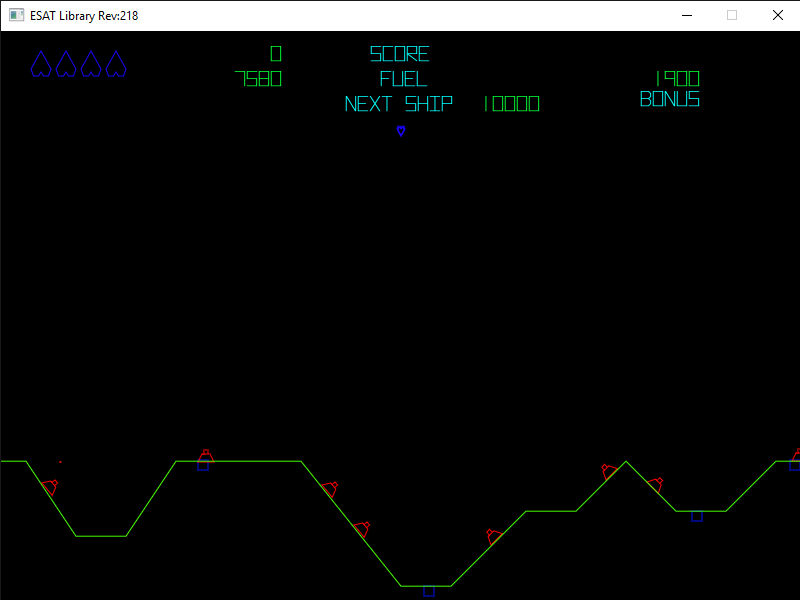
A cada planeta le corresponde un mapa con características propias y compartidas.

Mapas Sistema Solar 1:

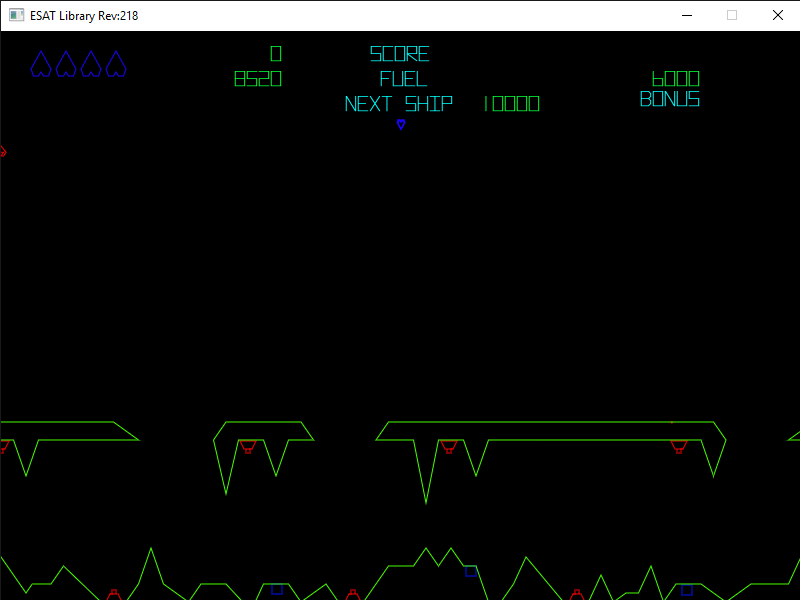


Mapas Sistema Solar 1:

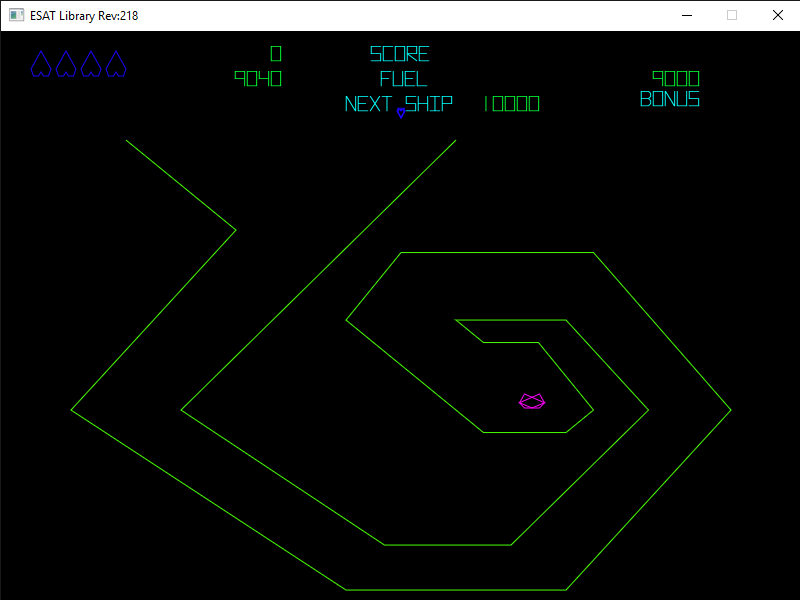
La gravedad de este mapa atrae desde el centro de la figura y para abandonar del planeta la nave tiene que alcanzar cualquier límite posible de la pantalla. No hay ningún tipo de zoom al mapa.



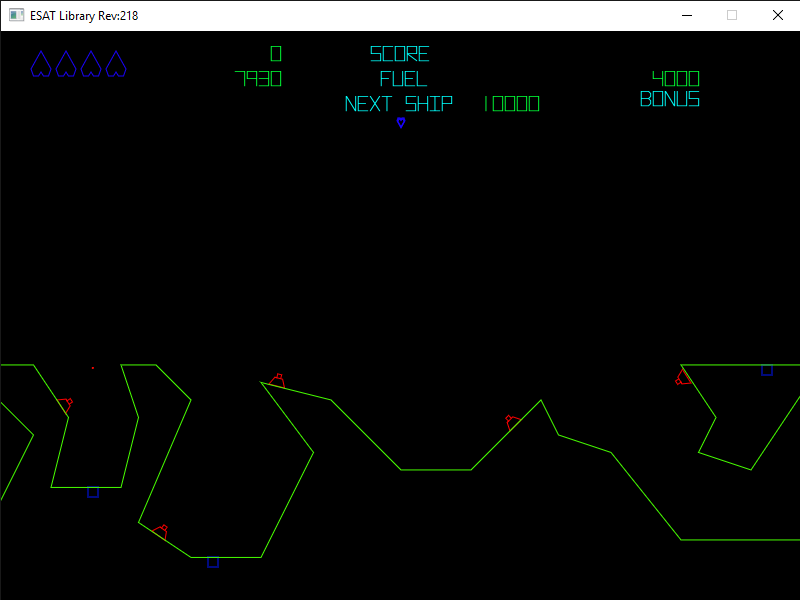
La gravedad de este mapa atrae desde el límite inferior de la pantalla y para abandonar el planeta la nave tiene que alcanzar cualquier límite posible de la pantalla. El jugador aparece cerca del límite superior en el centro del mapa y si se acerca hay zoom y al contrario si se aleja, también permite scroll lateral infinito ya que irá repitiendo el dibujado. Para abandonar el planeta la nave tiene que alcanzar el límite superior de la pantalla.



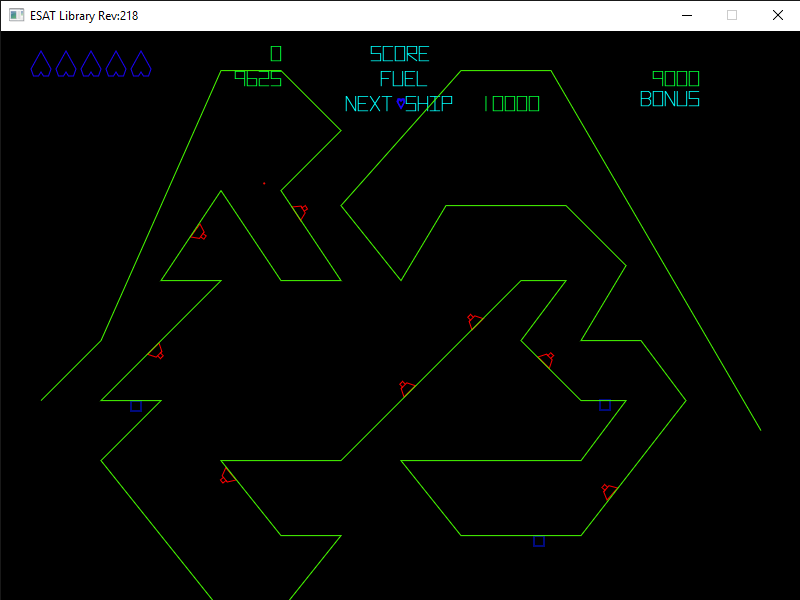
La gravedad de este mapa atrae desde el límite inferior de la pantalla y para abandonar el planeta la nave tiene que alcanzar cualquier límite posible de la pantalla. El jugador aparece cerca del límite superior en el centro del mapa y si se acerca hay zoom y al contrario si se aleja, también permite scroll lateral infinito ya que irá repitiendo el dibujado. Para abandonar el planeta la nave tiene que alcanzar el límite superior de la pantalla.



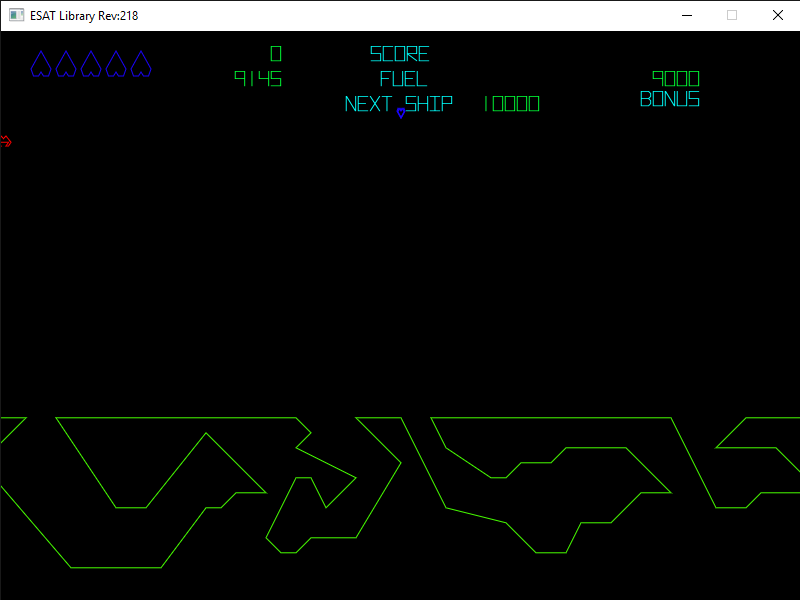
Este mapa carece de gravedad y la nave del jugador aparece dentro de un circuito cerrado que colisionando con cualquier borde conllevará a la pérdida de una vida y el objetivo es alcanzar el reactor con un disparo, momento en el que se activan las torretas enemigas pero esta vez sin ser obligación acabar con ellas, y poder escapar antes de que acabe el tiempo. Si consigues activar el reactor y escapar del mapa se accede al siguiente sistema solar.



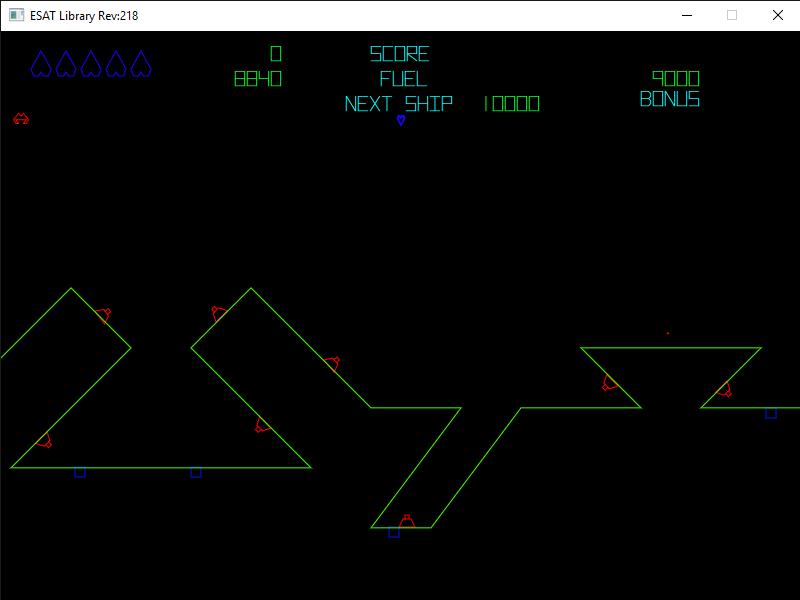
La gravedad de este mapa atrae desde el límite inferior de la pantalla y para abandonar el planeta la nave tiene que alcanzar cualquier límite posible de la pantalla. El jugador aparece cerca del límite superior en el centro del mapa y si se acerca hay zoom y al contrario si se aleja, también permite scroll lateral infinito ya que irá repitiendo el dibujado. Para abandonar el planeta la nave tiene que alcanzar el límite superior de la pantalla.



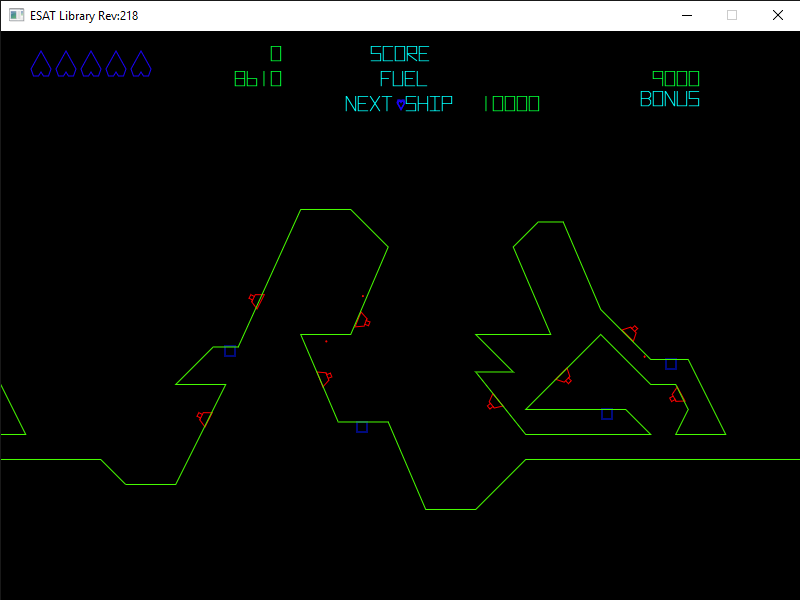
La gravedad de este mapa atrae desde el límite inferior de la pantalla y para abandonar el planeta la nave tiene que alcanzar cualquier límite posible de la pantalla. El jugador aparece cerca del límite superior en el centro del mapa no permite ningún tipo de scroll. Para abandonar el planeta la nave tiene que alcanzar cualquier límite de la pantalla.



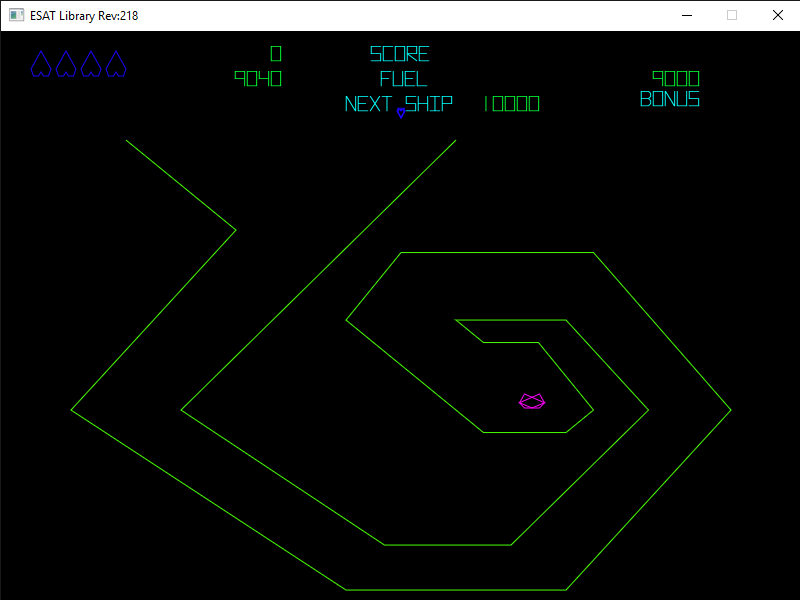
La gravedad de este mapa atrae desde el límite inferior de la pantalla y para abandonar el planeta la nave tiene que alcanzar cualquier límite posible de la pantalla. El jugador aparece cerca del límite superior en el centro del mapa y si se acerca hay zoom y al contrario si se aleja, también permite scroll lateral infinito ya que irá repitiendo el dibujado. Para abandonar el planeta la nave tiene que alcanzar el límite superior de la pantalla.



La gravedad de este mapa atrae desde el límite inferior de la pantalla y para abandonar el planeta la nave tiene que alcanzar cualquier límite posible de la pantalla. El jugador aparece cerca del límite superior en el centro del mapa y si se acerca hay zoom y al contrario si se aleja, también permite scroll lateral infinito ya que irá repitiendo el dibujado. Para abandonar el planeta la nave tiene que alcanzar el límite superior de la pantalla.

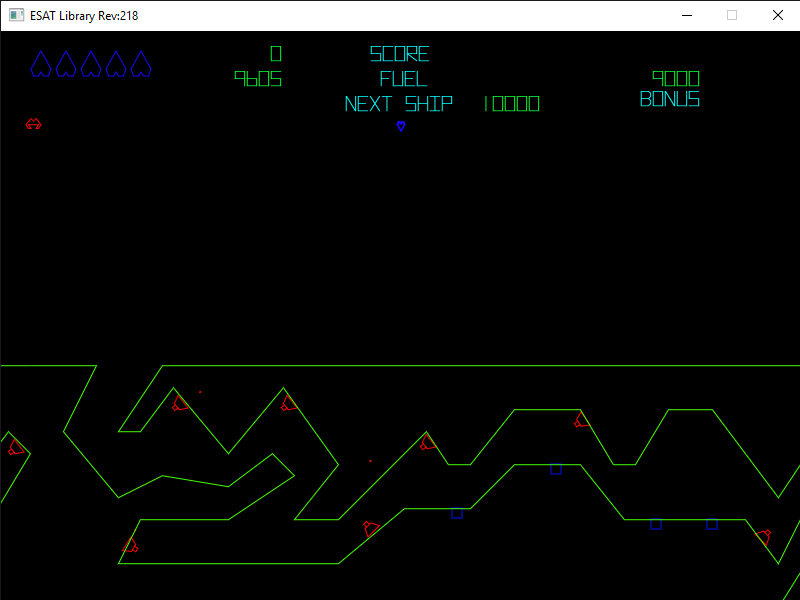


La gravedad de este mapa atrae desde el límite inferior de la pantalla y para abandonar el planeta la nave tiene que alcanzar cualquier límite posible de la pantalla. El jugador aparece cerca del límite superior en el centro del mapa y si se acerca hay zoom y al contrario si se aleja, también permite scroll lateral infinito ya que irá repitiendo el dibujado. Para abandonar el planeta la nave tiene que alcanzar el límite superior de la pantalla.

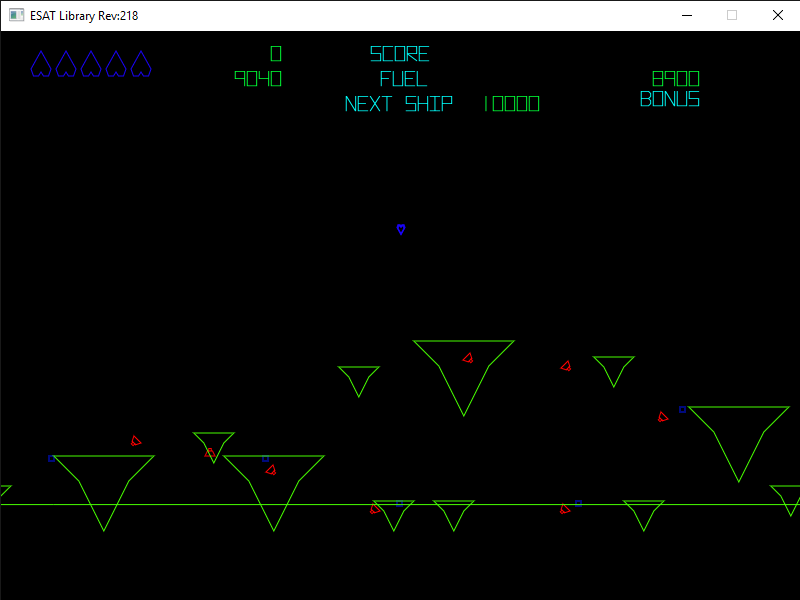


Este mapa carece de gravedad y la nave del jugador aparece dentro de un circuito cerrado que colisionando con cualquier borde conllevará a la pérdida de una vida y el objetivo es alcanzar el reactor con un disparo, momento en el que se activan las torretas enemigas pero esta vez sin ser obligación acabar con ellas, y poder escapar antes de que acabe el tiempo. Si consigues activar el reactor y escapar del mapa se accede al siguiente sistema solar.

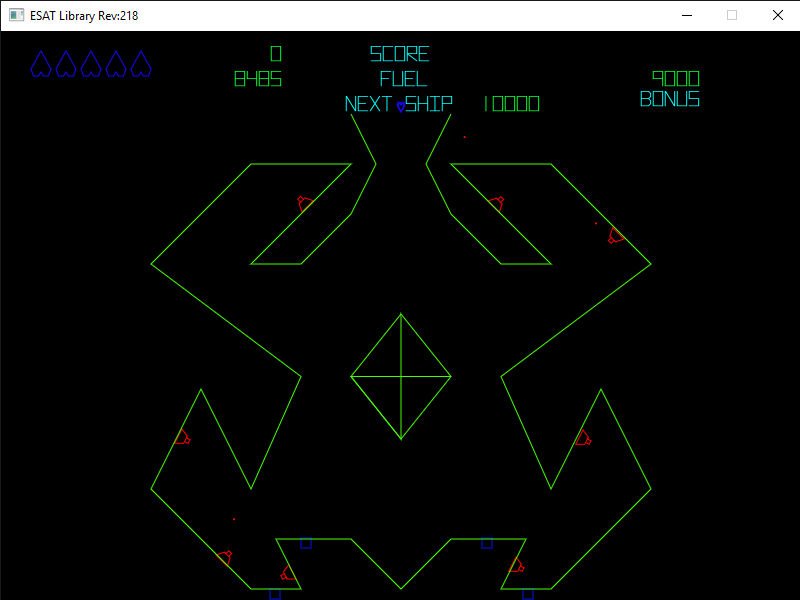
MAPAS SISTEMA SOLAR 3



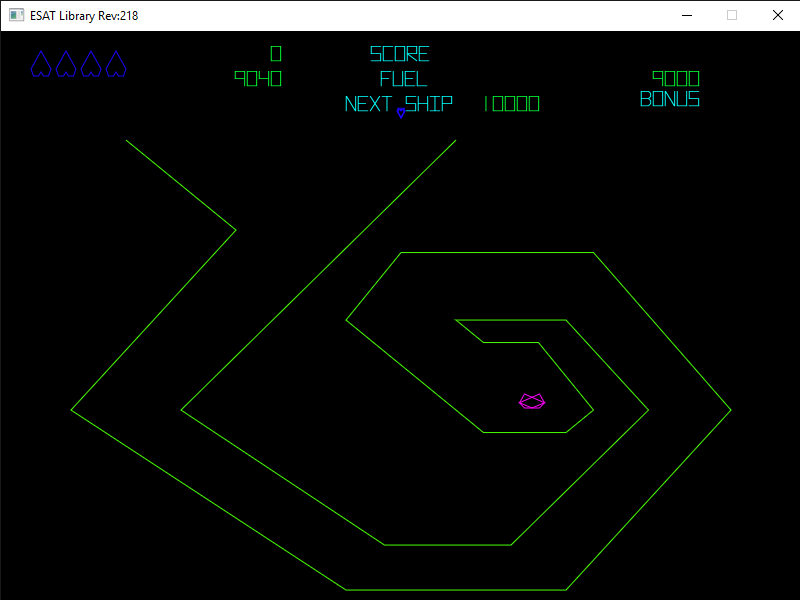
La gravedad de este mapa atrae desde el límite inferior de la pantalla y para abandonar el planeta la nave tiene que alcanzar cualquier límite posible de la pantalla. El jugador aparece cerca del límite superior en el centro del mapa y si se acerca hay zoom y al contrario si se aleja, también permite scroll lateral infinito ya que irá repitiendo el dibujado. Para abandonar el planeta la nave tiene que alcanzar el límite superior de la pantalla.



La gravedad de este mapa atrae desde el límite inferior de la pantalla y para abandonar el planeta la nave tiene que alcanzar cualquier límite posible de la pantalla. El jugador aparece cerca del límite superior en el centro del mapa y si se acerca hay zoom y al contrario si se aleja, también permite scroll lateral infinito ya que irá repitiendo el dibujado. Para abandonar el planeta la nave tiene que alcanzar el límite superior de la pantalla.



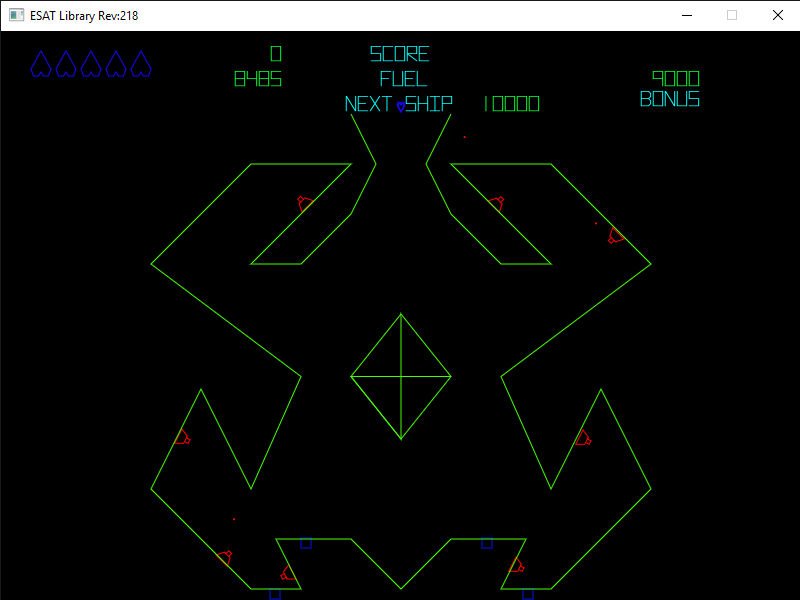
La gravedad de este mapa atrae desde el límite inferior de la pantalla y para abandonar el planeta la nave tiene que alcanzar cualquier límite posible de la pantalla. El jugador aparece cerca del límite superior en el centro del mapa no permite ningún tipo de scroll. Para abandonar el planeta la nave tiene que alcanzar cualquier límite de la pantalla.



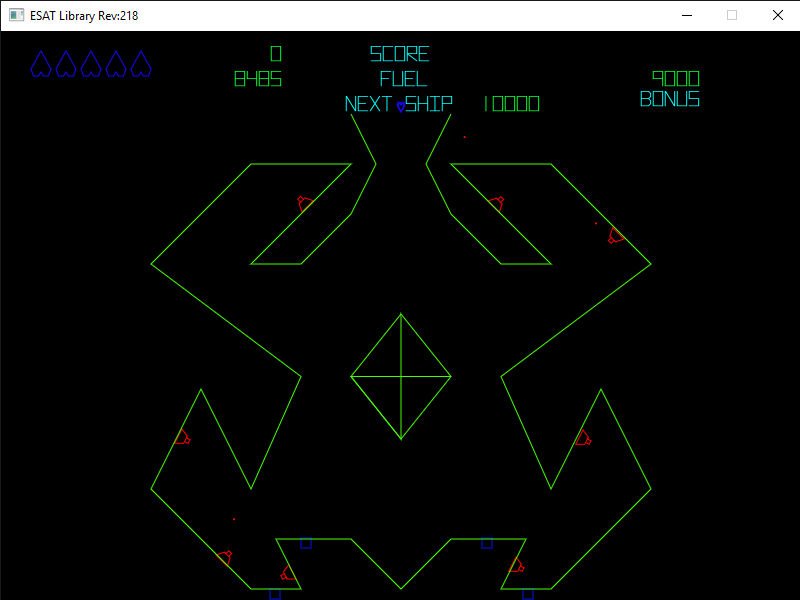
Este mapa carece de gravedad y la nave del jugador aparece dentro de un circuito cerrado que colisionando con cualquier borde conllevará a la pérdida de una vida y el objetivo es alcanzar el reactor con un disparo, momento en el que se activan las torretas enemigas pero esta vez sin ser obligación acabar con ellas, y poder escapar antes de que acabe el tiempo. Si consigues activar el reactor y escapar del mapa se accede al siguiente sistema solar.

OBJETIVO DENTRO DE CADA PLANETA

Cada mapa tiene 8 torretas enemigas que disparan al jugador y 4 cajas azules repartidas por el mapa para recargar el fuel, también hay naves rastreadoras enemigas que se mueven de lado a lado para acabar con el jugador. Para completar el mapa hay que destruir las 8 torretas correspondientes, si se abandona el mapa sin completar y posteriormente se accede la puntuación corresponderá al momento en el que se abandonó la vez anterior.

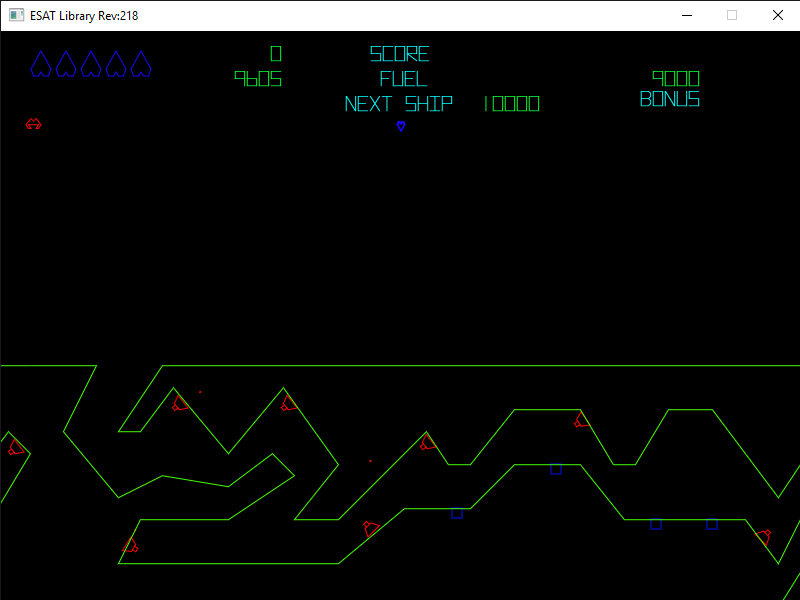


Cada torreta otorga 250 puntos al jugador. Las torretas disparan al jugador.



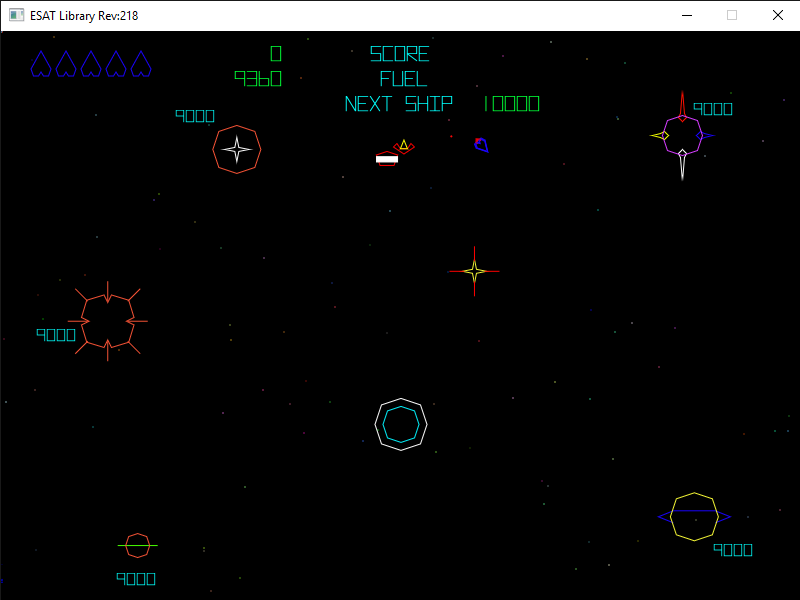
En cada mapa también hay distintos cuadrados que otorgan fuel al jugador, para recogerla hay que activar el escudo del que dispone al jugador. Otorga 1100 puntos de fuel.

En los mapas que existe el scroll o el movimiento lateral existen unos enemigos llamados rastreadores que van de un lado al otro. No disparan.

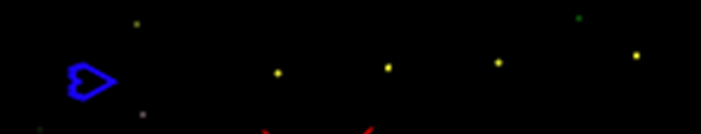


JUGADOR

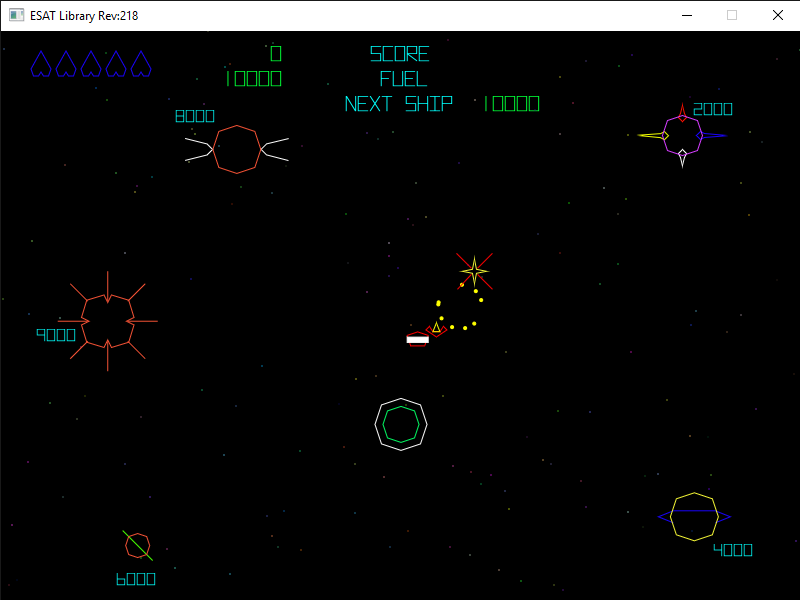
La nave del jugador aparece en su lugar de nacimiento ya indicado. Para desplazarse la nave usa fuel que se va consumiendo y se activa el motor de la nave.



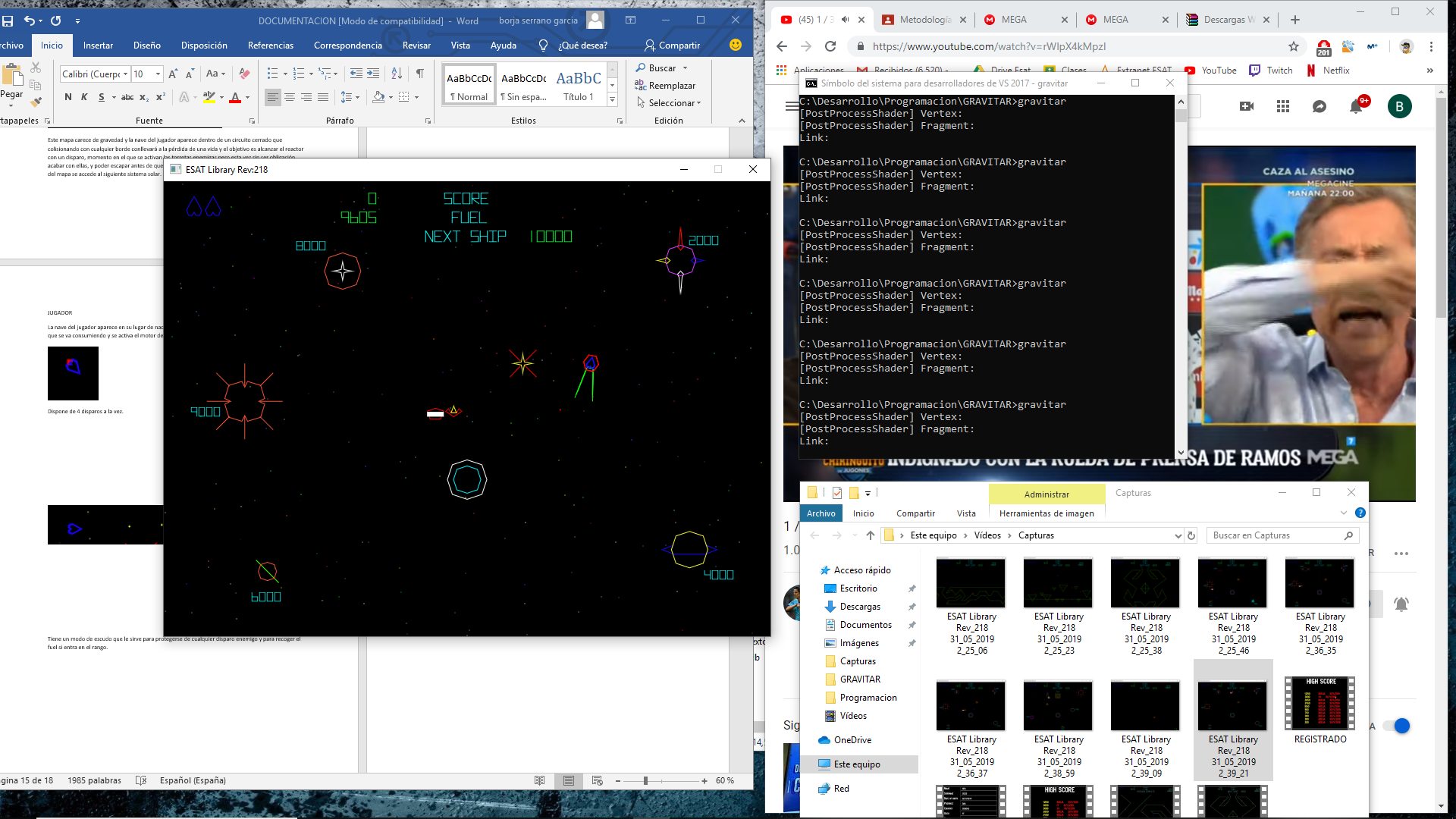
Dispone de 4 disparos a la vez.



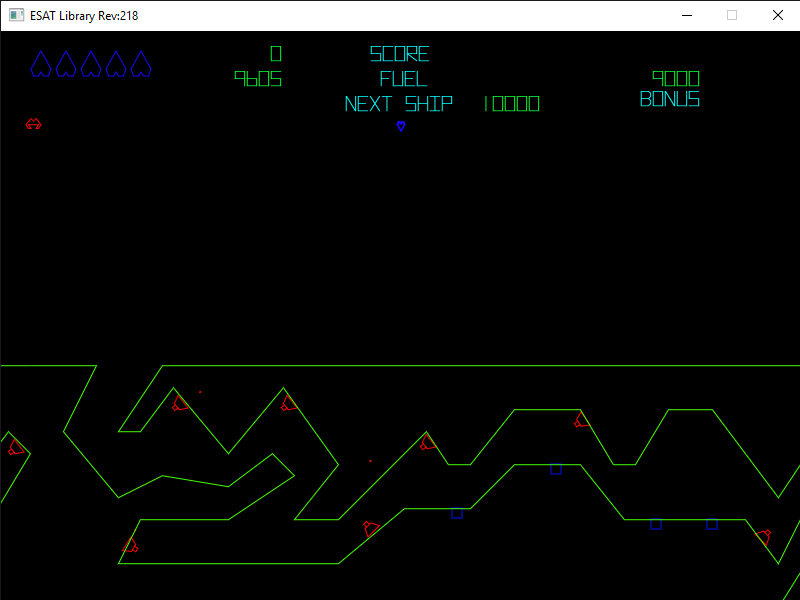
Cuando el jugador colisiona con algo explota generando pequeñas bolas con el número total y la dirección aleatoria.



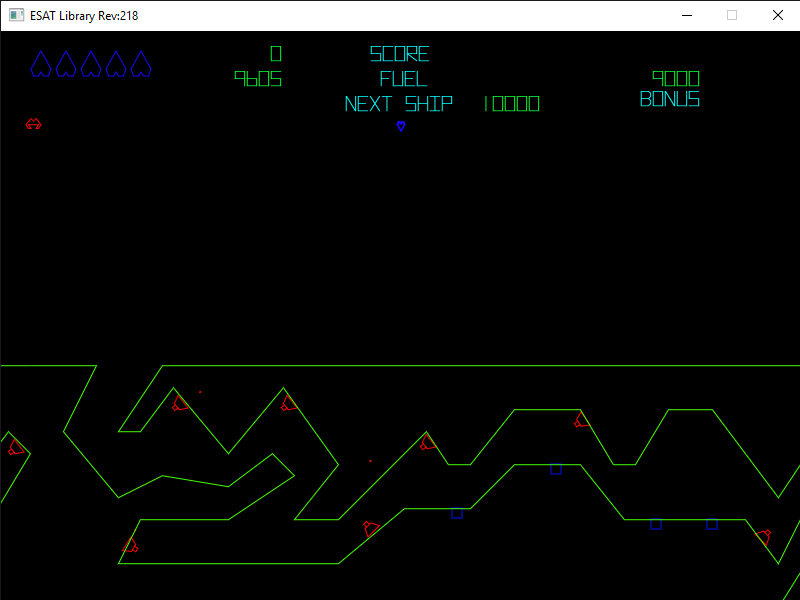
Tiene un modo de escudo que le sirve para protegerse de cualquier disparo enemigo y para recoger el fuel si entra en el rango.



HUD

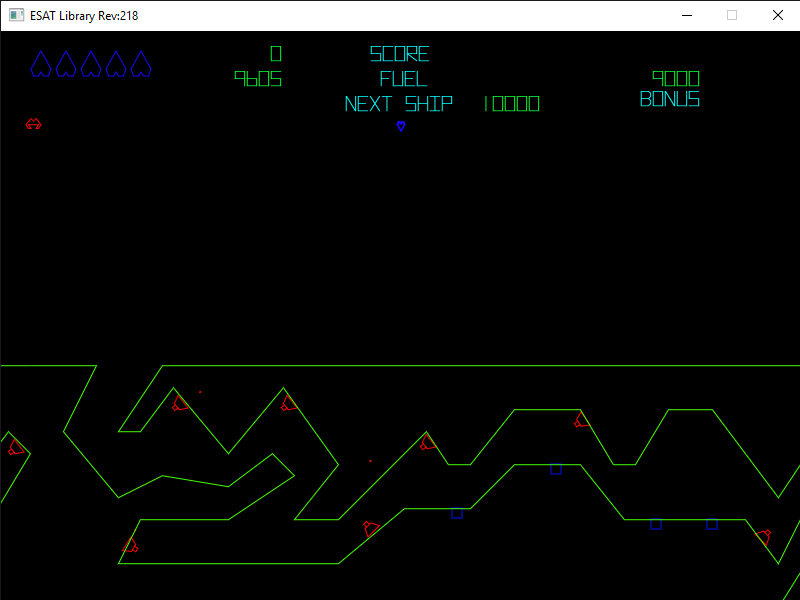


La pantalla de gameplay costa de distintos elementos que indican el estado de juego.



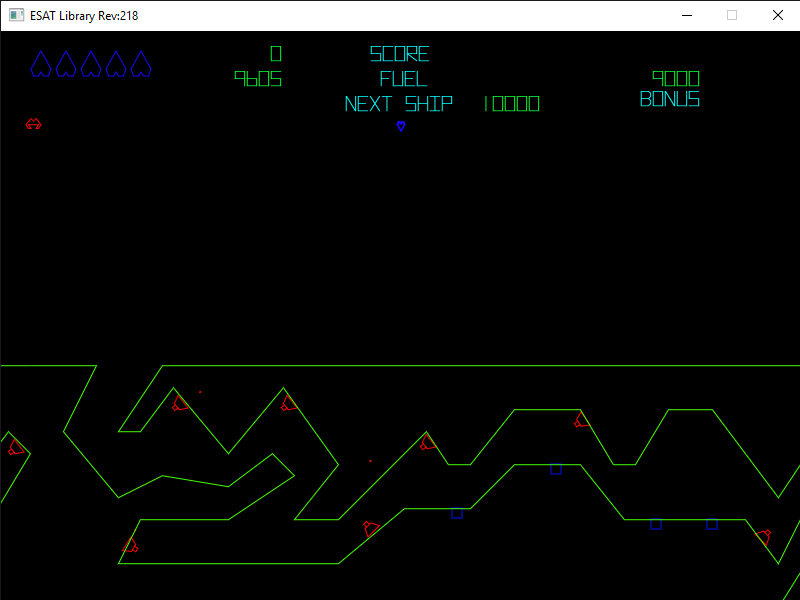
Las vidas restantes del jugador aparecen en la esquina superior izquierda representadas con naves del jugador a mayor escala. Si el jugador pierde todas las vidas se acaba la partida actual.

Se pierden vidas con cada colisión de la nave del jugador con enemigos o con el propio mapa o incluso entrando en la órbita de la estrella pero también se pueden ganar vidas cuando alcanzas ciertos objetivos en la puntuación. Esos objetivos aparecen a la derecha de NEXT SHIP y una vez alcanzando el siguiente objetivo está a 20000 puntos más.

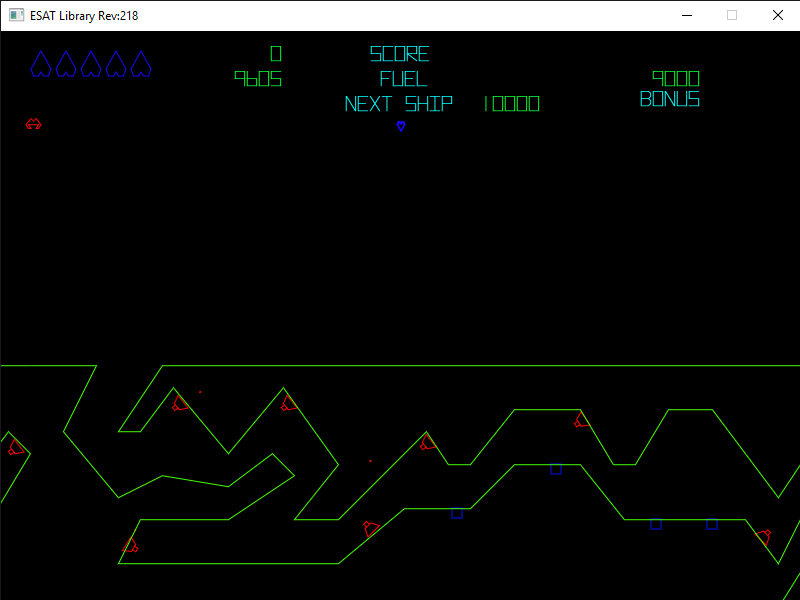


La puntuación actual de la partida aparece desde el medio de la pantalla.

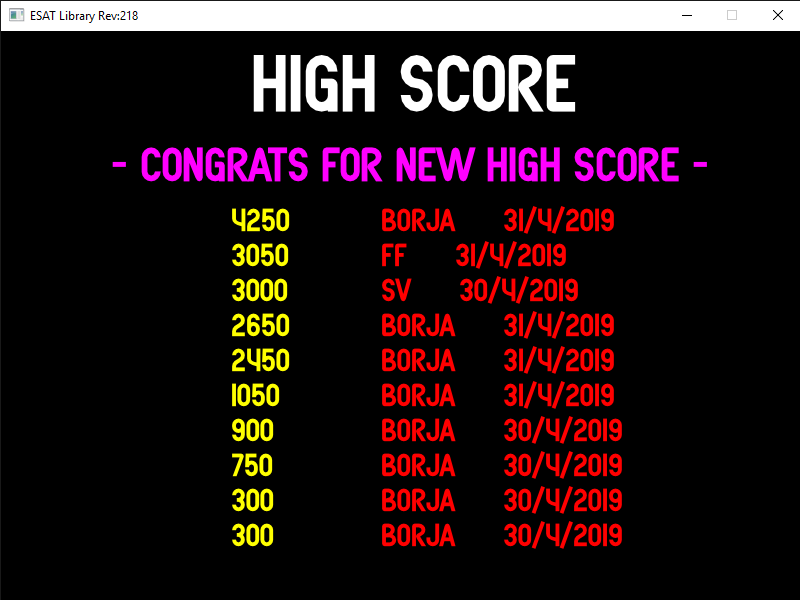
Lo mismo el indicador de fuel



Cada vez que entras en un planeta tiene una puntuación determinada y por cada 5 segundos que pases dentro va disminuyendo, si completas el mapa acabando con todas las torretas al salir consigues sumar esa puntuación, en caso contrario si vuelves a entrar seguirá con la puntuación con el que lo dejaste.



El juego continuará mientras el jugador no pierda las vidas pero una vez se pierdan se mostrará la pantalla de mejores puntuaciones y si se ha entrado en ese TOP 10 aparecerá un mensaje de felicitación. Al cabo de los segundos se vuelve a la pantalla de gameplay.



**INSTRUCCIONES DE EJECUCIONES E INSTALACIÓN**

Para ejecutar el juego solo hace falta abrir el archivo llamado “gravitar.exe” que corresponde al ejecutable.

**ESTRUCTURAS DE DATOS**

El código consta de 3 archivos .cc independientes que en el ejecutable se unen. Por tanto se va a dividir la sección en 3 partes, una para cada archivo .cc. Los archivos son “gravitar.cc”, “lista\_disparos.cc” y “base\_datos.cc”.

LISTA\_DISPAROS

**ESTRUCTURAS DE DATOS**

|  |  |  |  |
| --- | --- | --- | --- |
| NOMBRE ESTRUCTURA | VARIABLES ESTRUCTURA | USO | DECLARACIÓN |
| TCoord | Float x,y | Determinar la posición | Global |
| TNodo | TCoord position | Determinar la posición | Global |
|  | esat::Vec2 direction | Determinar dirección del nodo | Global |
|  | int count | Contador | Global |
|  | bool alive | Determinar si vive o no | Global |
|  | TNodo \*prox |  | Global |

**VARIABLES**

|  |  |  |  |
| --- | --- | --- | --- |
| NOMBRE VARIABLE O VARIABLES | TIPO VARIABLE | USO | DECLARACIÓN |
| \*primero  \*ultimo | TNodo | Guardar posición del nodo | Global |
| \*ultimo | TNodo | Guardar posición del nodo | Global |
| d  change\_direction last\_direction | esat::Vec2 | Determinar dirección del jugador | Global |
|  | int count | Contador | Global |
|  | bool alive | Determinar si vive o no | Global |

BASE\_DATOS

**ESTRUCTURAS DE DATOS**

|  |  |  |  |
| --- | --- | --- | --- |
| NOMBRE ESTRUCTURA | VARIABLES ESTRUCTURA | USO | DECLARACIÓN |
| TRegister | char \*name  \*surname  \*birthday  \*province  \*country  \*username  \*password \*secret\_password \*mail  \*credits | Guardar las cadenas para el registro o para el acceso a la aplicación | Global |

GRAVITAR

**ESTRUCTURAS DE DATOS**

|  |  |  |  |
| --- | --- | --- | --- |
| NOMBRE ESTRUCTURA | VARIABLES ESTRUCTURA | USO | DECLARACIÓN |
| TShoot | TCoord position | Determinar la posición | Global |
|  | int count\_alive | Contador | Global |
| TExplosion | TCoord position | Determinar posición | Global |
|  | int count\_explosion | Contador | Global |
|  | bool alive | Determinar si vive o no | Global |
|  | esat::Vec2 direction | Dirección del disparo | Global |
| TPlayer | TCoord position, last\_position, initial\_position | Determinar las distintas posiciones del jugador | Global |
|  | TShoot shoot | Manejar disparos | Global |
|  | TExplosion \*explosion | Activar explosiones | Global |
|  | float gravity  speed  acceleration  rotation  scale | Gravedad, velocidad, aceleración, rotación y escala del jugador | Global |
|  | int lifes  score  fuel  bonus  total\_shots  last\_screen  n\_explosion  exploding\_count  next\_ship | Contar vidas  Puntuación  Gasolina  Puntos de bonus a sumar  Total disparos  Pantalla anterior a la actual para regresar al acabar el mapa  Número de explosiones  Contador de las explosiones  Puntuación hasta la siguiente nave extra | Global |
|  | bool turbo  exploding  alive  shield  complete\_map  reactorrecord\_score | Comprobar turbo activado, explotando, si está vivo, escudo activado, mapa completado  Reactor activado y récord en la puntuación | Global |
| TStars | TCoord position | Posición de las estrellas | Global |
|  | int aparition, rgb[3] | Aparición y colores | Global |
|  | bool alive | Si la estrella está viva | Global |
| TPlanet | TCoord position | Posición del planeta | Global |
| TEnemy | TCoord position, last\_position | Posición del enemigo | Global |
|  | TShoot shoot | Disaparo del enemigo | Global |
|  | float speed | Velocidad del enemigo | Global |
|  | bool alive, shooting | Activar si vive y si dispara | Global |
|  | esat::Vec2 shoot\_direction | Dirección del disparo | Global |
| TScore | int number | Guardar el número para pantalla puntuaciones | Global |
|  | int position | Guardar posición para pantalla puntuaciones | Global |
|  | char user[60] | Guardar nombre y fecha usuario para pantalla puntuaciones | Global |

**VARIABLES**

|  |  |  |  |
| --- | --- | --- | --- |
| NOMBRE VARIABLE O VARIABLES | TIPO VARIABLE | USO | DECLARACIÓN |
| \*score | TScore | Guardar registros de puntuación | Global |
| player | TPlayer | Llevar todo lo correspondiente al jugador | Global |
| \*shot | TNodo | Llevar todo lo correspondiente al disparo | Global |
| \*stars | TStars | Llevar todo lo correspondiente a las estrellas | Global |
| \*planet | TPlanet | Llevar todo lo correspondiente a los planetas | Global |
| \*enemy | TEnemy | Llevar todo lo correspondiente a los enemigos | Global |
| \*planet\_alive | int | Si planeta está vivo o no | Global |
| \*planet\_bonus | int | Puntuación de cada planeta | Global |
| count\_bonus | int | Contador para restar el bonus de cada planeta | Global |
| \*count\_figure | float | Contador para todas las animaciones del juego | Global |
| scale\_map | float | Llevar la escala del juego completo | Global |
| planet\_speed | float | Velocidad de movimiento de los planetas | Global |
| counttt | float | Contador para los distintos zoom | Global |
| ppp | float | Posición en X de una de las 2 copias del mapa de cada planeta | Global |
| pppp | float | Posición en X de una de las 2 copias del mapa de cada planeta | Global |
| zoomin | bool | Activar zoom para dentro | Global |
| zoomout | bool | Activar zoom para fuera | Global |
| zoomleft | bool | Activar scroll a la izquierda | Global |
| zoomright | bool | Activar scroll para la derecha | Global |
| start | bool | Para iniciar el juego cuando se pierden todas las vidas | Global |
| change\_ss | bool | Para iniciar el juego cuando se cambia de sistema solar | Global |
| new\_ss | bool | Para iniciar el juego cuando se cambia de sistema solar | Global |
|  |  |  |  |
| p\_direction | esat::Vec2 | usar vector direccion para saber el movimeinto de cada planeta | void PlanetZoom()  Local |
| Dots  ……  Muchos nombres parecidos | esat::Vec2 | Vectores para guardar el dibujado de los mapas y distintos objetos | Todas las funciones que tengan dibujado  Local |
| gravity\_direction | esat::Vec2 | Dirección de la gravedad | void Gravity()  Local |
| total\_size  original\_size | float | Tamaño del dibujado del mapa original y según la escala para poder dibujar los elementos del mapa cuando haya zoom o no | Todas las funciones que se llamen DrawMap..()  Local |
| S  T  R  M | esat::Mat3  Identity | La escala  El tamaño  La rotación  Matriz identidad que va a llevar todo el dibujado del juego | void ScreensR() |

**MANUAL DE PROCEDIMIENTOS Y FUNCIONES**

LISTA\_DISPAROS

TNodo \*InsertarNodo(float xx, float yy) -> recibe un dato tipo nodo con una x y una y y crea el nodo.

Líneas 16 – 41

void VaciarLista(TNodo \*lista) -> recibe un tipo nodo y lo vacía de la lista

Líneas 43 – 72

BASE\_DATOS

void IniTRegister() -> inicia variables de registro.

Líneas 51 – 63

void Quad (int x, int y, int width, int height) -> dibuja el quad recibiendo una x, y, ancho y alto.

Líneas 66– 75

void Circle(float g\_circle[], float g\_center\_x, float g\_center\_y, float g\_radius\_x, float g\_radius\_y) -> dibuja un círculo. Recibe el centro de x y de y y el radio de x y de y.

Líneas 78 – 84

bool DetectMouseQuad(int x, int y, int width, int height) -> función para dar un rango de detección de click del ratón, devuelve cierto o falso.

Líneas 108 – 114

void WriteWord(char \*&word) -> recibe una cadena para escribir la palabra letra a letra y poder borrar letras.

Líneas 118 – 167

bool CompleteSignUp() -> revisa si se ha completado el registro y devuelve verdadero o falso.

Líneas 170 – 185

bool CompleteSignIn() -> revisa si se puede acceder con el usuario y la contraseña y devuelve verdadero o falso.

Líneas 188 – 231

void SecretPassword() -> mostrar la contraseña o cambiarla por signos.

Líneas 234 – 241

void ScreenR1() -> pantalla de registrarse o registrado.

Líneas 244 – 272

void ScreenR2() -> pantalla de registrarse.

Líneas 275 – 371

void ScreenR3() -> pantalla de registrado.

Líneas 374 – 447

GRAVITAR

void IniBonus() -> inicializar bonus.

Líneas 107 – 129

void IniPlayer () -> inicializar jugador.

Líneas 131 – 177

void IniFigures () -> inicializar contador animaciones.

Líneas 179 – 196

void IniStars () -> inicializar estrellas.

Líneas 198 – 203

void IniPlanet() -> inicializar planetas.

Líneas 205 – 219

void IniEnemy() -> inicializar enemigos.

Líneas 221 – 270

void IniExplosion () -> inicializar explosiones.

Líneas 272 – 274

void PlanetAlive () -> inicializar planetas vivos.

Líneas 276 – 278

void AllIni () -> unir todas las inicializaciones.

Líneas 280 – 290

void NormalizeVector(float &i, float &j) -> normaliza vector que recibe.

Líneas 293 – 298

void ScreenR5() -> pantalla de puntuaciones

Líneas 302 – 383

void SaveScore() -> guarda la puntuación final

Líneas 385 – 408

bool CheckRectangleColision(short x, short y, short width, short height, short x2, short y2, short width2, short height2) -> detecta si hay collision entre cuadrados recibiendo su x, y , ancho y alto

Líneas 411 – 416

bool CheckCircleColision(float c1\_x, float c1\_y, float c1\_radio,float c2\_x, float c2\_y, float c2\_radio) -> detecta si hay collision entre cuadrados recibiendo su x, y y su radio

Líneas 418 – 425

bool CheckLineColision(esat::Vec2 d[100], int total, float scaleX, float scaleY, float x\_center, float y\_center) -> detecta si hay collision del jugador con una figura dibujada de punto a punto

Líneas 427 – 451

void DrawFigure(esat::Mat3 figure, esat::Vec2 dots[], esat::Vec2 trans\_vec[], int n, bool solidpath) -> recibe la matriz, los puntos, cuantos puntos son y si tiene relleno o no la figura

Líneas 467 – 480

void OctagonRGB() -> contador de figura y uso.

Líneas 485 – 493

void LineU() -> contador de figura y uso.

Líneas 495 – 498

void SemiArrow()-> contador de figura y uso.

Líneas 500 – 503

void LineSize()-> contador de figura y uso.

Líneas 505 – 508

void LineLarge() -> contador de figura y uso.

Líneas 510 – 514

void RingPosition()-> contador de figura y uso.

Líneas 515 – 518

void StarLines() -> contador de figura y uso.

Líneas 520 – 523

void MazeRotation() -> contador de figura y uso.

Líneas 525 – 528

void CircleDots() -> contador de figura y uso.

Líneas 530 – 533

void EyeAnimation() -> contador de figura y uso.

Líneas 535 – 538

void BlueStar() -> contador de figura y uso.

Líneas 540 – 544

void GreenOctagonS3() -> contador de figura y uso.

Líneas 545 – 549

void CircleX3() -> contador de figura y uso.

Líneas 550 – 554

void Bonus(int &bon) -> resta la puntuación del planeta y lo almacena en la variable que recibe.

Líneas 555 – 564

void RestartPlanet() -> reinicia posición de planetas.

Líneas 565 – 570

void PlanetZoom(float &actualX, float &actualY,float xx, float yy) -> hace desplazamiento de los planetas recibiendo la X y la Y actual y hacia donde se desplaza, lo guarda.

Líneas 572 – 585

void PlanetScale(float &scale) -> cambia la escala del mapa.

Líneas 586 – 591

void StarsBackGround()-> funcionamiento del fondo de las estrellas.

Líneas 593 – 613

void MapComplete()-> controla si el mapa ha sido completado destruyendo todas las torretas.

Líneas 615 – 643

void ResetM(esat::Mat3 &m, esat::Mat3 &s, esat::Mat3 &r, esat::Mat3 &t) -> reinicia la matriz recibiendo y guardando la matriz, escala, rotación y posición.

Líneas 645 – 651

void DotsPlayer(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r, esat::Mat3 t) -> puntos de dibujado.

Líneas 654 – 668

void DotsTurbo(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r, esat::Mat3 t) -> puntos de dibujado.

Líneas 669 – 677

void DotsFuel(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r, esat::Mat3 t) -> puntos de dibujado.

Líneas 678 – 703

void DotsOctagon(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r, esat::Mat3 t) -> puntos de dibujado.

Líneas 704 – 718

void DotsTurret(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r, esat::Mat3 t) -> puntos de dibujado.

Líneas 719 – 734

void DotsOctagon2(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r, float x, float y, float size\_line) -> puntos de dibujado.

Líneas 735 – 769

void DotsStar(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r, float x, float y, bool lines) -> puntos de dibujado.

Líneas 770 – 803

void DotsOctagonStart(esat::Mat3 m, float x, float y) -> puntos de dibujado.

Líneas 804 – 818

void DotsEnemy(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r, esat::Mat3 t) -> puntos de dibujado.

Líneas 819 – 886

void DotsEnemyRastreator(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r, esat::Mat3 t) -> puntos de dibujado.

Líneas 887 – 914

void Title(esat::Mat3 m) -> dibuja el título y recibe la matriz.

Líneas 915 – 1058

void Hud(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> contola el HUD y recibe la matriz, escala, posición y rotación

Líneas 1061 – 1146

void CountSpeed()-> controla la velocidad del jugador

Líneas 1149 – 1154

void PlayerShots()-> controla los disparos del jugador

Líneas 1155 – 1181

void Gravity(float xx, float yy) -> recibe el punto en X,Y de la gravedad y la aplica

Líneas 1155 – 1181

void LastPosition(float displacementX, float displacementY) -> guarda la posición y la screen antes de cambiar dando un desplazamiento en X e Y.

Líneas 1188 – 1201

void NewPosition(int new\_screen, float xx, float yy, float new\_scale) -> cambia la screen recibiendo la nueva, la posición en X e Y y la nueva escala.

Líneas 1202 – 1214

void LimitsRebound()-> rebota con los límites de la pantalla.

Líneas 1215 – 1229

void LimitsMaps()-> límites de cada mapa para salir.

Líneas 1230 – 1246

void ZoomIn()-> maneja el zoom haci dentro.

Líneas 1247 – 1252

void ZoomOut()-> maneja el zoom hacia fuera.

Líneas 1253 – 1258

void ZoomLeft()-> maneja scroll lateral.

Líneas 1259 – 1267

void ZoomRight()->maneja scroll lateral.

Líneas 1268 – 1276

void LimitsMaps2(float size, float original) -> límites de los mapas que no son sistemas solares que recibe el tamaño del mapa original y escalado.

Líneas 1277 – 1328

void ExplosionPlayer()-> maneja la explosión del jugador

Líneas 1329 – 1432

void Movement()-> maneja el movimiento del jugador.

Líneas 1433 – 1473

void DrawPlayer(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> se encarga de dibujar al jugador y recibe la matriz, escala , rotación y posición.

Líneas 1474 – 1522

void OvnisMovement(float &xx, float &yy, float ss) -> maneja el movimiento de los ovnis recibiendo y guardando la X y la Y y la velocidad.

Líneas 1525 – 1532

void OvnisShoots()-> disparos de los ovnis.

Líneas 1533 – 1557

void Ovnis(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> maneja todo lo relacionado con los ovnis.

Líneas 1558 – 1647

void RastreatorMovement(esat::Mat3 m, esat::Mat3 s, esat::Mat3 r, esat::Mat3 t, float maxY) - -> movimiento del enemigo rastreados y recibe la matriz con su escala, rotación y posición.

Líneas 1648 – 1713

void TurretsShooting(int i, float xx, float yy, float total\_size) -> se encarga de disparos de las torretas y de las colisiones.

Líneas 1714 – 1788

void ColisionFuel(int i, float xx, float yy, float total\_size) -> maneja la colisión con el fuel.

Líneas 1789 – 1807

void DrawMap1S1(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> dibuja el mapa y recibe la matriz con su escala, rotación y posición.

Líneas 1810 – 2085

void DrawMap2S1(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> dibuja el mapa y recibe la matriz con su escala, rotación y posición.

Líneas 2086 – 2241

void DrawMap3S1(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> dibuja el mapa y recibe la matriz con su escala, rotación y posición.

Líneas 2242 – 2577

void DrawMap4S1(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> dibuja el mapa y recibe la matriz con su escala, rotación y posición.

Líneas 2578 – 2798

void DrawMap5S1(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> dibuja el mapa y recibe la matriz con su escala, rotación y posición.

Líneas 2799 – 2949

void DrawMap6S1(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> dibuja el mapa y recibe la matriz con su escala, rotación y posición.

Líneas 2950 – 3150

void DrawMap1S2(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r, esat::Mat3 t) -> dibuja el mapa y recibe la matriz con su escala, rotación y posición.

Líneas 3151 – 3329

void DrawMap2S2(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> dibuja el mapa y recibe la matriz con su escala, rotación y posición.

Líneas 3330 – 3482

void DrawMap3S2(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> dibuja el mapa y recibe la matriz con su escala, rotación y posición.

Líneas 3483 – 3688

void DrawMap4S2(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> dibuja el mapa y recibe la matriz con su escala, rotación y posición.

Líneas 3689 – 3927

void DrawMap5S2(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> dibuja el mapa y recibe la matriz con su escala, rotación y posición.

Líneas 3928 – 4009

void DrawMap1S3(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r, esat::Mat3 t) -> dibuja el mapa y recibe la matriz con su escala, rotación y posición.

Líneas 4011 – 4211

void DrawMap6S3(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> dibuja el mapa y recibe la matriz con su escala, rotación y posición.

Líneas 4212 – 4453

void DrawMap7S3(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r, esat::Mat3 t) -> dibuja el mapa y recibe la matriz con su escala, rotación y posición.

Líneas 4454 – 4605

void DrawMap8S3(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r, esat::Mat3 t) -> dibuja el mapa y recibe la matriz con su escala, rotación y posición.

Líneas 4606 – 4830

void ScreenG4(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> controla las funciones usadas en la screen y recibe la matriz con su escala, rotación y posición.

Líneas 4834 – 4854

void ScreenG5(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> controla las funciones usadas en la screen y recibe la matriz con su escala, rotación y posición.

Líneas 4855 – 4870

void ScreenG6(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> controla las funciones usadas en la screen y recibe la matriz con su escala, rotación y posición.

Líneas 4871 – 4885

void ScreenG7(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> controla las funciones usadas en la screen y recibe la matriz con su escala, rotación y posición.

Líneas 4886 – 4899

void ScreenG8(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> controla las funciones usadas en la screen y recibe la matriz con su escala, rotación y posición.

Líneas 4900 – 4921

void ScreenG15(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> controla las funciones usadas en la screen y recibe la matriz con su escala, rotación y posición.

Líneas 4922 – 4935

void ScreenG12(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> controla las funciones usadas en la screen y recibe la matriz con su escala, rotación y posición.

Líneas 4937 – 4958

void ScreenG16(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> controla las funciones usadas en la screen y recibe la matriz con su escala, rotación y posición.

Líneas 4959 – 4970

void ScreenG17(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> controla las funciones usadas en la screen y recibe la matriz con su escala, rotación y posición.

Líneas 4971 – 4983

void ScreenG18(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> controla las funciones usadas en la screen y recibe la matriz con su escala, rotación y posición.

Líneas 4984 – 4996

void ScreenG19(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> controla las funciones usadas en la screen y recibe la matriz con su escala, rotación y posición.

Líneas 4997 – 5009

void ScreenG13(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> controla las funciones usadas en la screen y recibe la matriz con su escala, rotación y posición.

Líneas 5011 – 5032

void ScreenG11(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> controla las funciones usadas en la screen y recibe la matriz con su escala, rotación y posición.

Líneas 5033 – 5044

void ScreenG10(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> controla las funciones usadas en la screen y recibe la matriz con su escala, rotación y posición.

Líneas 5045 – 5056

void ScreenG9(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> controla las funciones usadas en la screen y recibe la matriz con su escala, rotación y posición.

Líneas 5057 – 5069

void ScreenG14(esat::Mat3 m, esat::Mat3 s,esat::Mat3 r,esat::Mat3 t) -> controla las funciones usadas en la screen y recibe la matriz con su escala, rotación y posición.

Líneas 5070 – 5077

void ScreensR() -> controla el manejo de screens y, por tanto, todo el juego.

Líneas 5079 - 5106