



UNIVERSIDAD LUTERANA SALVADOREÑA
FACULTAD DE CIENCIAS DEL HOMBRE Y LA NATURALEZA

Asignatura: Programación Orientada a Objetos
Sábado 9:40 a.m. – 12:10 p.m.

Ciclo: II-2018

Docente: Inga. Lesbia Mancía

Actividad: Proyecto Final: Videojuego “Mozo’ Space”

Apellido y Nombres Carnet

- Aguilar Cruz, Gabriela Alexandra AC01135229
- Arévalo Viera, Eduardo Alexánder AV01135084
- Flores Fuentes, Santos Félix FF01134621
- Mozo García, Josué Ezequiel MG01135024
- Ramírez Vásquez, Carlos Antonio RV01135068
- Salaverría Orellana, Gloria Elena SO01135025

Carrera: Licenciatura en Ciencias de la Computación.

Fecha de Entrega: 24/11/2018

Contenido

INTRODUCCION.....	3
OBJETIVOS.....	4
LISTADO DE REQUERIMIENTOS.....	5
CASOS DE USO.....	7
DIAGRAMA DE CLASE.....	9
MANUALES (TECNICO Y USUARIO).....	10
ANEXOS.....	20
BIBLIOGRAFIAS.....	28

INTRODUCCIÓN

Durante el desarrollo de la asignatura Programación Orientada a Objetos (POO), se ha realizado un proyecto con fines educativos para el alumno, así mismo, tiene un fin de entretenimiento, éste último determinado por el equipo de trabajo.

A continuación, se encuentra toda la información en referencia al proyecto, Nave Espacial en Python. El informe consta de los diferentes componentes que interfieren en la creación del proyecto: Diagramas UML, Listado de Requerimientos, Identificación de los Objetos que interactúan en el desarrollo de la aplicación, Manual del Desarrollador, Manual de Usuario y el código como tal.

Este proyecto consiste en la creación de un videojuego de Nave Espacial que destruye enemigos y acumula puntaje, llamado Mozo' Space; para su creación fue de mucha ayuda el manejo de Pilas Engine; ya que con dicha plataforma se logró comprender de manera más sencilla las palabras reservadas que Python utiliza a la hora de programar, debido a que está totalmente en español; además de poseer una guía interactiva.

OBJETIVOS

Objetivo general

- Mostrar el funcionamiento del videojuego, con sus distintos componentes.

Objetivo específico:

- Aplicar los conocimientos sobre UML para dibujar el Diagrama de Usos y el Diagrama de Clases, que corresponden al videojuego de éste proyecto.

LISTADO DE REQUERIMIENTOS

El videojuego permitirá el acceso de un jugador y este podrá:

- Jugar en un solo nivel.
- Esquivar enemigos.
- Destruir enemigos.
- Disparar misiles.
- Moverse sin patrón.
- Acumular puntaje, cuando:
 - 1 punto al eliminar un enemigos.
- Perder hasta 1 vida, al colisionar con enemigos.
- Perder el juego.
- Salir del juego.
- Reiniciar juego.

OBJETOS:

Objeto: Nave
Métodos: Girar, disparar, cambiar de dirección en los ejes X, Y; esquivar enemigos, explotar.
Propiedades: misil, color rojo-blanco.

Objeto: Contador
Métodos: Iniciar en cero, contar de uno en uno, pausar, contar hasta finalizar juego, reiniciar.
Propiedades: Justificado en el borde superior derecho, dimensión: 0.5x3.0cm, números color gris claro

Objeto: Vidas

Métodos: Iniciar en 1, decrementar a 0.

Propiedades: Contador: numérico.

Objeto: Enemigos

Métodos: Caer (eje Y), golpear nave, girar.

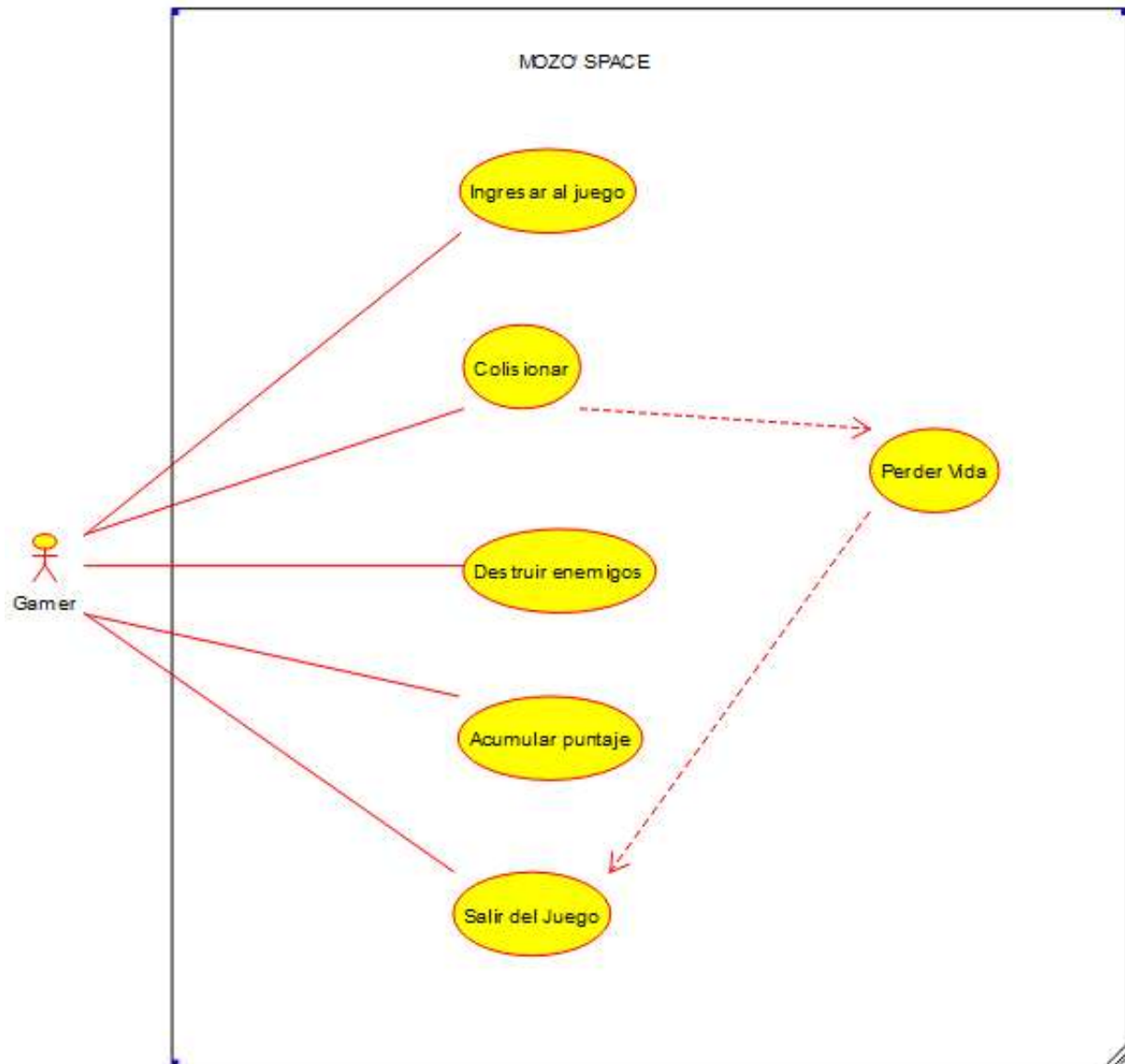
Propiedades: Tamaño: mediano, pequeño; color: café, frecuencia: 3 a 5 por 5 segundos, velocidad: 15cm x 5secs.

Objeto: Pantalla (Interfaz)

Métodos: Aparecer durante el juego al darle iniciar, cambiar al presionar botón

Propiedades: Inicial con botón de iniciar juego, segunda pantalla fondo espacial

CASOS DE USOS



Caso de uso	Ingresar al juego
objetivo	Jugar
Actor principal	Gamer
Personal involucrado	Estudiantes desarrolladores del juego
Precondición	Tener el juego descargado
Garantía de éxito	Muestra la interfaz del juego
Escenario principal	Darle play
Flujos alternativos	

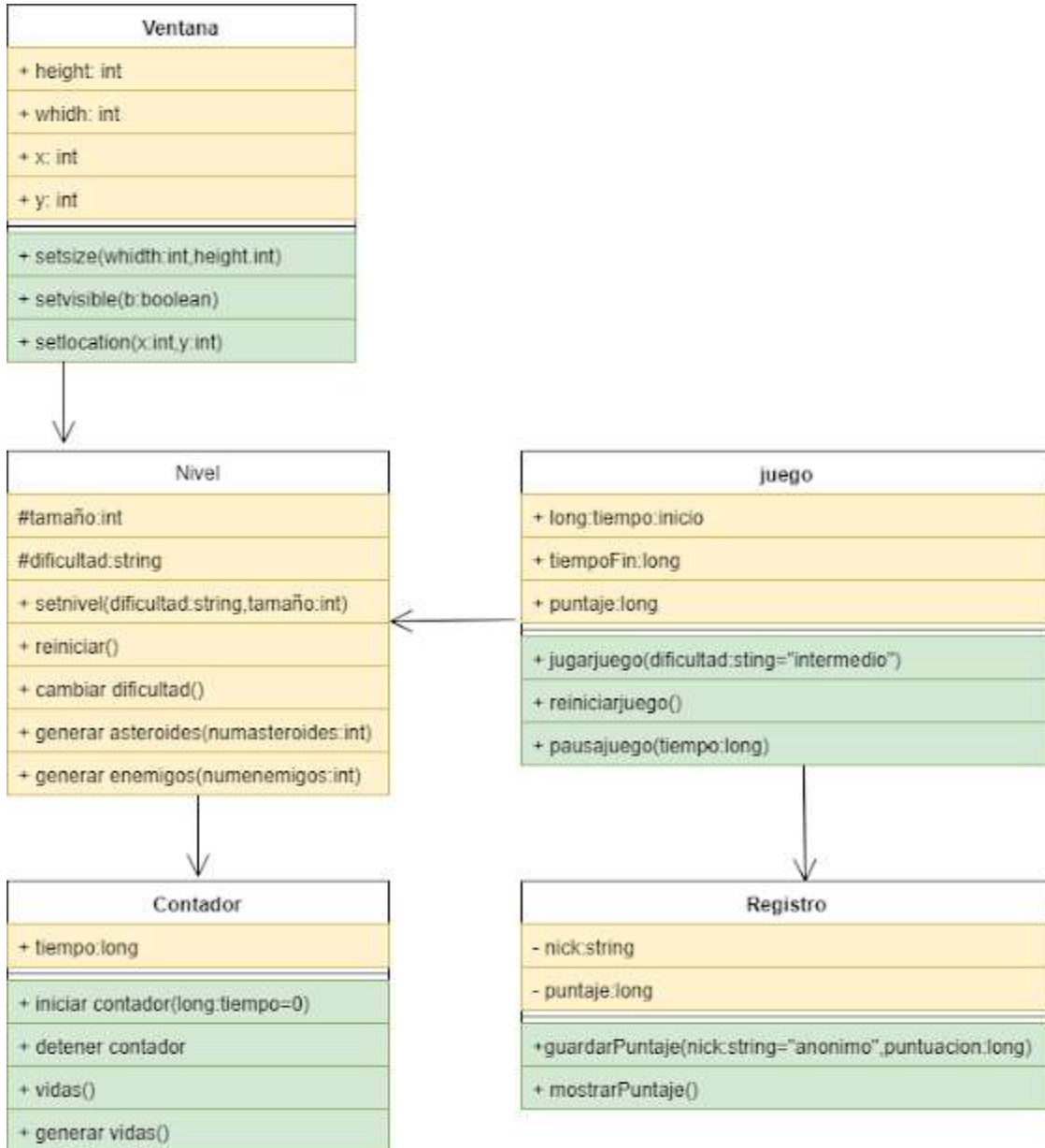
Caso de uso	Colisionar
Objetivo	Eliminar nave
Actor principal	Nave
Personal involucrado	Estudiantes desarrolladores del juego
Precondición	Tener el juego descargado
Garantía de éxito	mostrar niveles y interfaz del juego
Escenario principal	Darle play
Flujos alternativos	

Caso de uso	Acumular puntaje
objetivo	Acumular los puntos
Actor principal	Gamer
Personal involucrado	Estudiantes desarrolladores del juego
Precondición	Tener el juego descargado
Garantía de éxito	Mostrar niveles y interfaz del juego
Escenario principal	Eliminar enemigos
Flujos alternativos	Que el marcador no acumule puntos

Caso de uso	Destruir Enemigos
Objetivo	Eliminar naves enemigas
Actor principal	Gamer
Personal involucrado	Estudiantes desarrolladores del juego
Precondición	Tener el juego instalado
Garantía de éxito	Elimine a cada enemigo que le dispare
Escenario principal	Ingresar al juego
Flujos alternativos	Que no se elimine aunque se le dispare

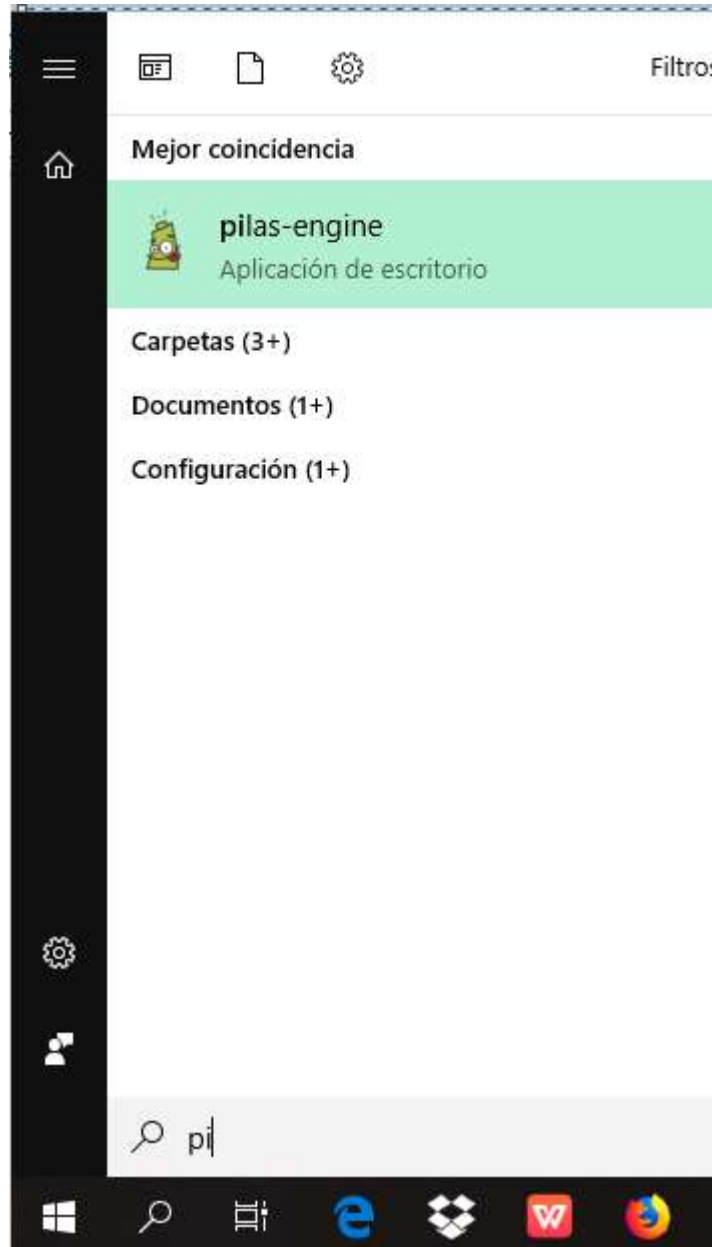
Caso de uso	Salir del juego
Objetivo	Dar por finalizado el juego
Actor principal	Gamer
Personal involucrado	Estudiantes desarrolladores del juego
Precondición	Tener el juego en ejecución
Garantía de éxito	Que valide la salida del jugador
Escenario principal	Finalizar la partida
Flujos alternativos	

DIAGRAMA DE CLASES

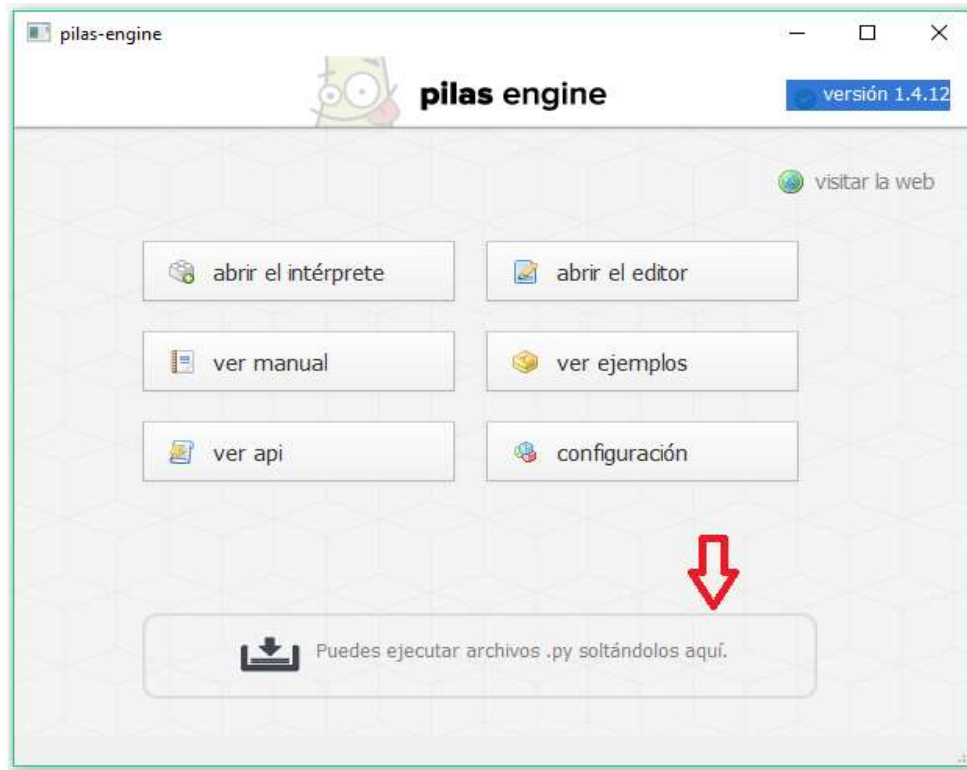


MANUAL DE USUARIO

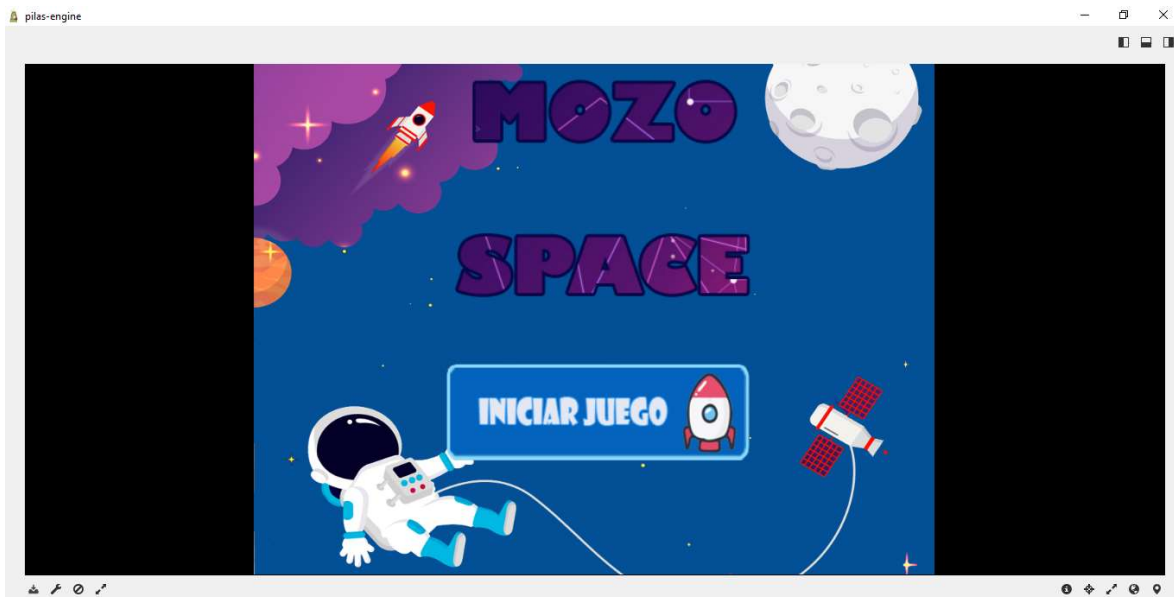
Paso 1: Abrir la Aplicación Pilas Engine, desde el menú de programas y aplicaciones



Paso 2: En la pantalla principal de Pilas Engine, arrastrar el archivo que contenga el código del videojuego: MOZOSPACE.py



Paso 3: La pantalla principal se mostrará de la siguiente manera. Para iniciar, se debe dar clic en INICIAR JUEGO



Paso 4: Para disparar misiles, el jugador debe presionar la barra espaciadora. Dispara un misil por vez que se presione.



Paso 5: Eliminar la mayoría de enemigos posibles, para acumular puntaje. Para Moverse por la pantalla, usar las cursoras del teclado.



Para salir del juego o reiniciarlo, dar clic en VOLVER A INICIO

MANUAL TÉCNICO

La elaboración de éste videojuego requiere una computadora que cumpla con los siguientes requisitos:

- Una computadora con un mínimo de 45MB de RAM para ejecutar el juego.
- Espacio en disco duro de al menos 80 MB.
- Tarjeta de video Intel HD 4400
- Teclado en buenas condiciones.

El desarrollador deberá instalar la Plataforma de Pilas-Engine desde el web oficial:

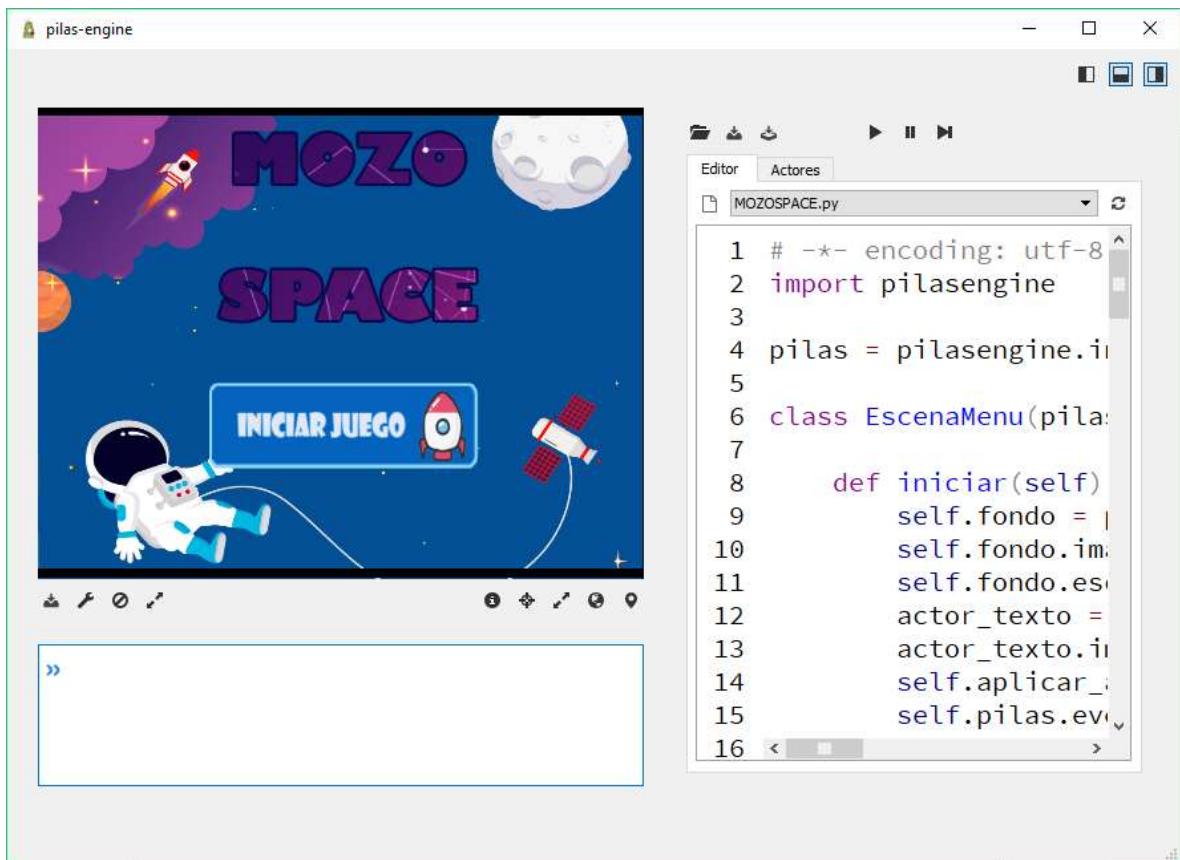
<http://pilas-engine.com.ar/>

Pasos para la instalación de Pilas-Engine:

1. Ingresar al web y seleccionar la opción DESCARGAR que se encuentra al centro de la pantalla en color verde.
2. Luego de haber descargado la aplicación ejecutable, instalar de igual forma que otros programas.
3. Abrir la aplicación Pilas-Engine y seleccionar la opción Abrir Editor:



4. Dentro del editor, realizar el código del programa.



5. Código del videojuego documentado:

```
# -*- encoding: utf-8 -*-
```

```
import pilasengine
```

```
pilas = pilasengine.iniciar()
```

```
#Primero declaramos la clase EscenaMenu para luego Vincularla con  
EscenaJuego, La definimos como escena
```

```
class EscenaMenu(pilasengine.escenas.Escena):
```

```
    def iniciar(self):
```

```
        #Con esta linea de codigo establecemos el fondo que le vamos a poner menu
```

```
        self.fondo = pilas.fondos.Fondo()
```

```
        self.fondo.imagen = pilas.imagenes.cargar('FondoMenu.png')
```

```
        self.fondo.escala = [0.64]
```

```
        #Con esta linea de codigo definimos la variable actor_texto(Este lo utilizamos  
para poder introducir el Boton Iniciar)
```

```
        actor_texto = self.pilas.actores.Actor()
```

```
        #Con esta linea de codigo le establecemos una imagen el boton iniciar
```

```
        actor_texto.imagen = "BotonIniciar.png"
```

```
        #Declaramos que le pondremos animacion al Boton Iniciar
```

```
        self.aplicar_animacion(actor_texto)
```

```
        #Esta linea de codigo para dar orden de que con un click podremos ir a  
Escena Juego(Nos podremos meter al juego)
```

```
        self.pilas.eventos.click_de_mouse.conectar(self.inicio_de_juego)
```

```
        #Declaramos Titulo
```

```
        self.Titulo()
```

```
        #Este es para el mensaje que nos saldra en el juego
```

```
        pilas.avisar(u"Pulsá espacio para disparar.")
```

```
#Declaramos la características de la animacion
```

```

def aplicar_animacion(self, texto):
    texto.y = -500
    texto.escala = 4
    texto.escala = [1], 1.5
    texto.y = [-100], 1
#Definimos el inicio de juego y lo Vinculamos con EscenaJuego
def inicio_de_juego(self, evento):
    self.pilas.escenas.EscenaJuego()
#Definimos Titulo
def Titulo(self):
    #Lo definimos como actor(Objeto)
    titulo = self.pilas.actores.Actor()
    #Le establecemos una imagen
    titulo.imagen = "MozoSpace.png"
    titulo.y = 300
    titulo.rotacion = 30
    titulo.y = [100], 1
    titulo.rotacion = [0], 1.2
#Hacemos la clase de BotonVolver El cual nos ayudara a llegar al Menu Inicio
class BotonVolver(pilasengine.actores.Actor):

    def iniciar(self):
        #Le establecemos una imagen
        self.imagen = "VolverMenu.png"
        self.cuando_hace_click = self.volver_a_inicio
        self.y = 200
        self.y = [210]
        self.x = -210
#Definimos volver_a_inicio
def volver_a_inicio(self, evento):
    self.pilas.escenas.EscenaMenu()

```


#Creamos la clase AlienDestructor el cual lo definimos como actor(Objeto)

```
class AlienDestructor(pilasengine.actores.Actor):
```

```
    def iniciar(self):
```

```
        #Le establecemos una imagen
```

```
        self.imagen = "AlienDestructor.png"
```

```
        self.imagen.escala=0.50
```

```
        #Aplicamos características
```

```
        self.aprender( pilas.habilidades.PuedeExplotarConHumo )
```

```
        self.x = pilas.azar(-200, 200)
```

```
        self.y = 290
```

```
        self.velocidad = pilas.azar(10, 40) / 10.0
```

```
    def actualizar(self):
```

```
        self.rotacion += 0
```

```
        self.y -= self.velocidad
```

```
        # Elimina al actor si sale de la pantalla
```

```
        if self.x > 400:
```

```
            self.eliminar()
```

#Declaramos la clase EscenaJuego para luego vincularla con EscenaMenu, La definimos como escena

```
class EscenaJuego(pilasengine.escenas.Escena):
```

```
    def iniciar(self):
```

```
        #Establecemos imagen de Fondo a EscenaJuego
```

```
        self.fondo = pilas.fondos.Fondo()
```

```
        self.fondo.imagen = pilas.imagenes.cargar('FondoJuego.png')
```

```
        self.fondo.escala = [0.64]
```

```
        #Definimos Boton Volver
```

```
        self.crear_boton_volver()
```

```

#Declaramos la creacion de Enemigos
self.pilas.tareas.siempre(2, self.crear_un_Alien_Destructor)
#Declaramos la variable puntaje
self.puntaje = pilas.actores.Puntaje(290, 200, color="blanco")
#Definimos y Declaramos caracteristica a la Nave(La cual heredada de la
libreria de pilas )
self.nave = pilas.actores.NaveRoja(y=-200)
self.nave.aprender(pilas.habilidades.LimitadoABordesDePantalla)
#Definimos la creacion de Enemigos
self.enemigos = pilas.actores.Grupo()
#Definimos
self.nave.definir_enemigos(self.enemigos, self.puntaje.aumentar)
#Declaramos colisiones
self.pilas.colisiones.agregar(self.nave, self.enemigos,self.nave.eliminar)
#Este es para el mensaje que nos saldra en el juego
self.pilas.avisar(u"Cuando hayas sido eliminado regresa a menu.")

def crear_boton_volver(self):
    pilas.actores.BotonVolver()

def crear_un_Alien_Destructor(self):
    actor=AlienDestructor(pilas)
    self.enemigos.agregar(actor)

## Vinculamos todas las escenas.
pilas.escenas.vincular(EscenaMenu)
pilas.escenas.vincular(EscenaJuego)

## Vinculamos los actores Personalizados
pilas.actores.vincular(BotonVolver)
pilas.actores.vincular(AlienDestructor)

```

```
# Se inicia la escena principal.  
pilas.escenas.EscenaMenu()
```

```
pilas.ejecutar()
```

CONCLUSIONES

La elaboración del videojuego, permitió que los alumnos del equipo comprendieran la aplicación de la Programación Orientada a Objetos y su diferencia respecto a la Programación Estructurada. De manera que, a pesar de tratarse de un videojuego, el proyecto cumplió su función educativa.

El uso de la plataforma Pylas Engine, ha sido una herramienta didáctica creada por el Ministerio de Educación de Argentina y ha crecido a lo largo de los 6 años desde su creación.

Una vez más se comprueba, que las herramientas creadas a través del Software Libre son igual o más efectivas que el Software Privativo, permitiendo a los estudiantes desarrollar proyectos de aprendizaje y entretenimiento, y con ello obtener resultados admirables.

Anexos.

Autoevaluación		
Nombre del Estudiante: Gloria Elena Salaverría Orellana		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Colaboré con las tareas que me fueron asignadas	9
2	Participé en forma activa en el trabajo de equipo	10
3	Mantuve comunicación con el equipo	9
4	Cumplí a tiempo con las actividades designadas	10
5	Aporté ideas de calidad	10
Total		48

Coevaluación		
Nombre del Evaluador: Gloria Elena Salaverría Orellana		
Nombre del Evaluado: Josue Ezequiel Mozo Garcia		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	10
2	Demostró respeto y tolerancia hacia los opiniones de los demás	10
3	Aportó al desarrollo del proyecto	10
4	Propicia un clima de trabajo agradable	10
5	Antes de entregar la tarea, fue revisado por el evaluado	10
Total		50

Autoevaluación		
Nombre del Estudiante: Josue Ezequiel Mozo García		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Colaboré con las tareas que me fueron asignadas	10
2	Participé en forma activa en el trabajo de equipo	10
3	Mantuve comunicación con el equipo	9
4	Cumplí a tiempo con las actividades designadas	10
5	Aporté ideas de calidad	10
Total		49

Coevaluación		
Nombre del Evaluador: Josue Ezequiel Mozo Garcia		
Nombre del Evaluado: Gloria Elena Salaverria Orellana		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	10
2	Demostró respeto y tolerancia hacia los opiniones de los demás	10
3	Aportó al desarrollo del proyecto	10
4	Propicia un clima de trabajo agradable	10
5	Antes de entregar la tarea, fue revisado por el evaluado	10
Total		50

Autoevaluación		
Nombre del Estudiante: Gabriela Alexandra Aguilar Cruz		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Colaboré con las tareas que me fueron asignadas	7
2	Participé en forma activa en el trabajo de equipo	7
3	Mantuve comunicación con el equipo	8
4	Cumplí a tiempo con las actividades designadas	7
5	Aporté ideas de calidad	8
Total		37

Coevaluación		
Nombre del Evaluador: Gabriela Alexandra Aguilar Cruz		
Nombre del Evaluado: Carlos Antonio Ramírez Vásquez		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	8
2	Demostró respeto y tolerancia hacia los opiniones de los demás	9
3	Aportó al desarrollo del proyecto	10
4	Propicia un clima de trabajo agradable	10
5	Antes de entregar la tarea, fue revisado por el evaluado	9
Total		50

Autoevaluación		
Nombre del Estudiante: Carlos Antonio Ramírez Vásquez		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Colaboré con las tareas que me fueron asignadas	9
2	Participé en forma activa en el trabajo de equipo	8
3	Mantuve comunicación con el equipo	8
4	Cumplí a tiempo con las actividades designadas	8
5	Aporté ideas de calidad	9
Total		42

Coevaluación		
Nombre del Evaluador: Carlos Antonio Ramírez Vásquez		
Nombre del Evaluado: Gabriela Alexandra Aguilar Cruz		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	7
2	Demostró respeto y tolerancia hacia los opiniones de los demás	8
3	Aportó al desarrollo del proyecto	8
4	Propicia un clima de trabajo agradable	8
5	Antes de entregar la tarea, fue revisado por el evaluado	7
Total		38

Autoevaluación		
Nombre del Estudiante: Eduardo Alexander Arévalo Viera		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Colaboré con las tareas que me fueron asignadas	8
2	Participé en forma activa en el trabajo de equipo	8
3	Mantuve comunicación con el equipo	8
4	Cumplí a tiempo con las actividades designadas	8
5	Aporté ideas de calidad	8
Total		40

Coevaluación		
Nombre del Evaluador: Eduardo Alexander Arévalo Viera		
Nombre del Evaluado: Santos Félix Flores Fuentes		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	10
2	Demostró respeto y tolerancia hacia los opiniones de los demás	10
3	Aportó al desarrollo del proyecto	9
4	Propicia un clima de trabajo agradable	10
5	Antes de entregar la tarea, fue revisado por el evaluado	9
Total		48

Autoevaluación		
Nombre del Estudiante: Santos Félix Flores Fuentes		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Colaboré con las tareas que me fueron asignadas	9
2	Participé en forma activa en el trabajo de equipo	10
3	Mantuve comunicación con el equipo	9
4	Cumplí a tiempo con las actividades designadas	10
5	Aporté ideas de calidad	9
Total		47

Coevaluación		
Nombre del Evaluador: Santos Félix Flores Fuentes		
Nombre del Evaluado: Eduardo Alexander Arévalo Viera		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	7
2	Demostró respeto y tolerancia hacia los opiniones de los demás	7
3	Aportó al desarrollo del proyecto	7
4	Propicia un clima de trabajo agradable	9
5	Antes de entregar la tarea, fue revisado por el evaluado	7
Total		37

LINK DE VIDEO

<https://www.youtube.com/watch?v=1FP5Iz6T2Hw>



MOZO SPACE



MOZO SPACE-Programacion orientada a objetos

Sin vistas



COMPARTIR



GUARDAR



BIBLIOGRAFIA

<http://pilas-engine.com.ar/>

<http://pilas-engine.com.ar/#/foro>

<http://manual.pilas-engine.com.ar/empezando/>

<https://www.youtube.com/watch?v=kzEVi0YgMgQ>