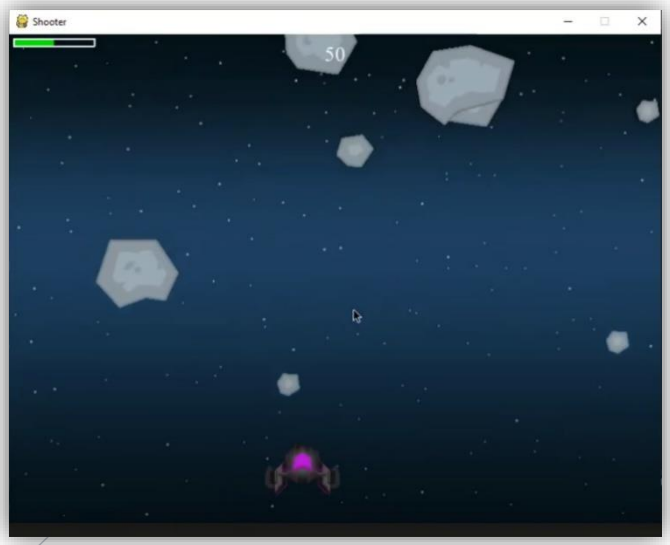
26-2-2023

Juego con Python

SHOOTER



Pere Manel Verdugo Zamora
pereverdugo@gmail.com

Este manual son los apuntes que he realizado basándome en los tutoriales que he encontrado en el YouTube en el canal Mundo Python.

Para acceder a los materiales necesarios y el código te adjunto la siguiente página web.

https://github.com/mundo-python/shooter-pygame

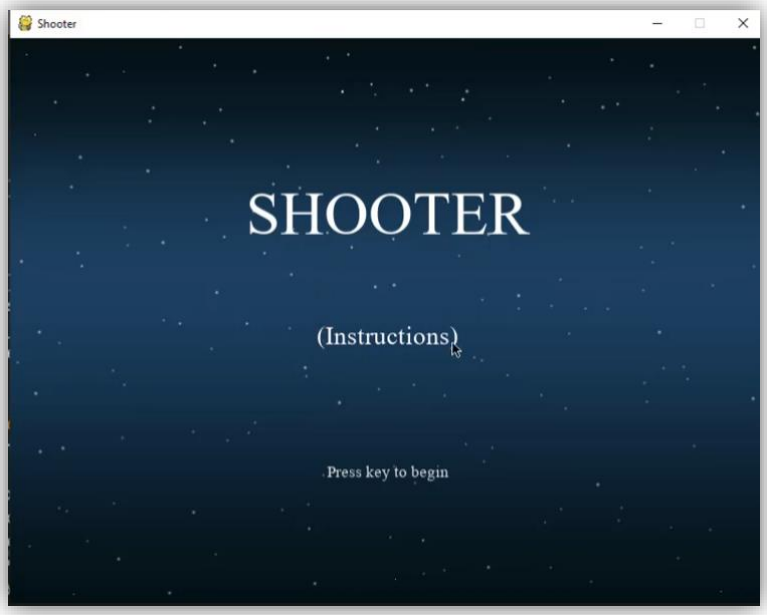
Yo solo me he descargado los materiales, el código lo dejo para cuando tengan algún problema con el seguimiento del curso y tenga que revisar el código en profundidad.

Contenido

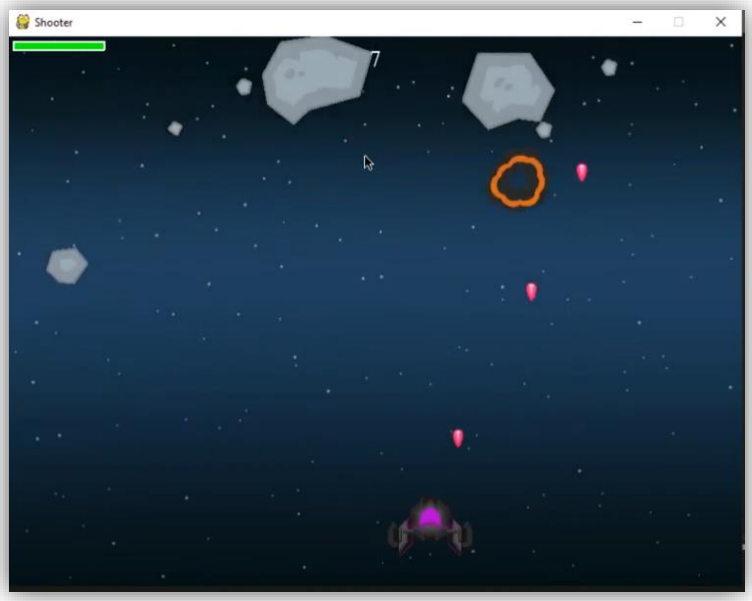
Parte 1 (Introducción)

En esta serie de tutoriales vamos a realizar un juego que tiene varios elementos:

Una pantalla de inicio

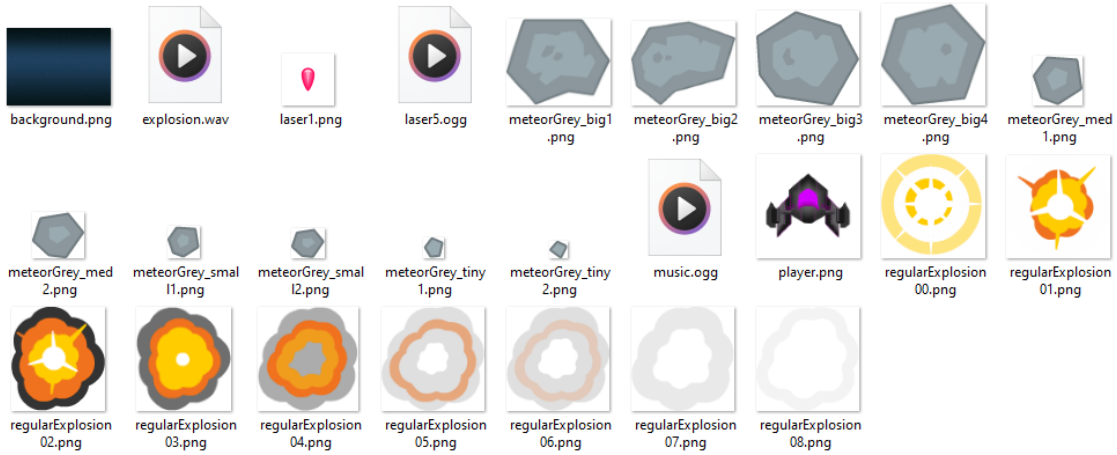


En la parte superior izquierda tendremos una barra que nos dirá como estamos de vida.



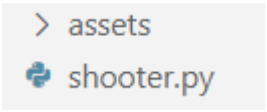
Si me tocan 4 meteoritos se finaliza el juego.

Necesitaremos todos estos elementos que al inicio del tutorial les he indicado donde lo pueden descargar.



Esta carpeta lo incluiremos en nuestro proyecto, este se tiene que llamar assets.

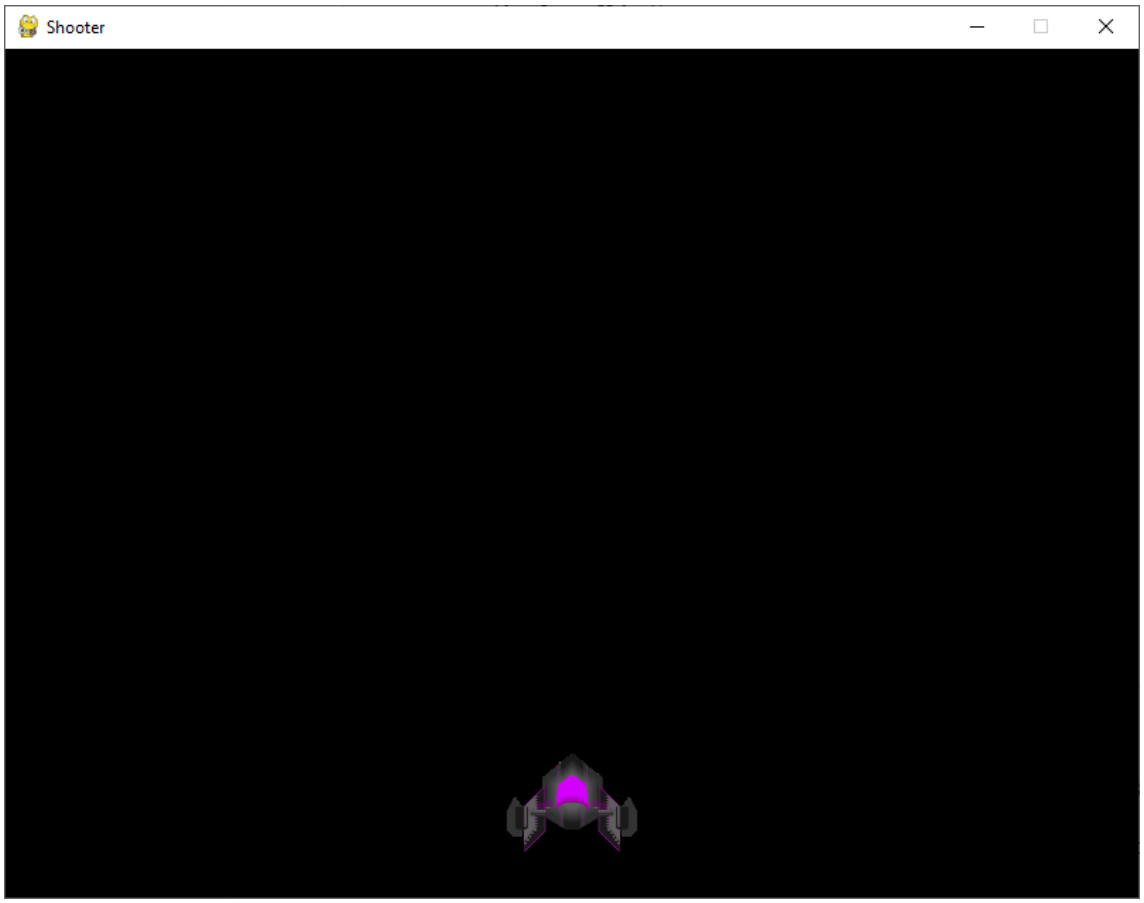
En este primer capítulo vamos a crear el archivo shooter.py.



Vamos a escribir el siguiente código:

```
import pygame, random

# Definimos las constantes para la dimensión de la ventana.
WIDTH = 800
HEIGHT = 600
# Definir color
BLACK = (0,0,0)
WHITE = (255, 255, 255)
# Inicializar pygame
pygame.init()
# Para la parte del sonido
pygame.mixer.init()
# Crear la ventana
screen = pygame.display.set_mode((WIDTH, HEIGHT))
# Título de la ventana
pygame.display.set_caption("Shooter")
# Para controlar los Frames por segundo
clock = pygame.time.Clock()

# Vamos a crear nuestra clase jugador
class Player(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        # Cargamos la imagen
        self.image = pygame.image.load("assets/player.png").convert()
        # Para borrar el borde negro
        self.image.set_colorkey(BLACK)
        # Definimos el rectángulo
```

```
        self.rect = self.image.get_rect()
        # Lo colocamos en la ventana
        self.rect.centerx = WIDTH // 2
        self.rect.bottom = HEIGHT - 10
        # Inicializamos una variable para la velocidad
        self.speed_x = 0

    def update(self):
        self.speed_x = 0
        # Para comprobar si presionamos flecha derecha e izquierda
        keystate = pygame.key.get_pressed()
        if keystate[pygame.K_LEFT]:
            self.speed_x = -5
        if keystate[pygame.K_RIGHT]:
            self.speed_x =  5
        # Para que no se salga de la derecha ni de la izquierda
        self.rect.x += self.speed_x
        if self.rect.right > WIDTH:
            self.rect.right = WIDTH
        if self.rect.left < 0:
            self.rect.left = 0

# Creamos el grupo
all_sprites = pygame.sprite.Group()

# Creamos una intancia a Player
player = Player()
# Lo agregamos a all_sprites
all_sprites.add(player)

# Bucle principal
running = True
while running:
    clock.tick(60)
    # Evento para salir de la ventana
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    all_sprites.update()
    screen.fill(BLACK)
    # Dibujar en la ventana
    all_sprites.draw(screen)
    pygame.display.flip()

pygame.quit()
```


Shooter

Parte 2 (Enemigos)

Vamos a añadir la siguiente clase:

```
class Meteor(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image =
pygame.image.load("assets/meteorGrey_med1.png").convert()
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.x = random.randrange(WIDTH - self.rect.width)
        self.rect.y = random.randrange(-100, -40)
        self.speedy = random.randrange(1, 10)
        self.speedx = random.randrange(-5, 5)

    def update(self):
        self.rect.y += self.speedy
        self.rect.x += self.speedx
```

Creamos el grupo:

```
meteor_list = pygame.sprite.Group()
```

Los añadimos a los grupos all_sprites y meteor_list un total de 8 meteoritos.

```
for i in range(8):
    meteor = Meteor()
    all_sprites.add(meteor)
    meteor_list.add(meteor)
```

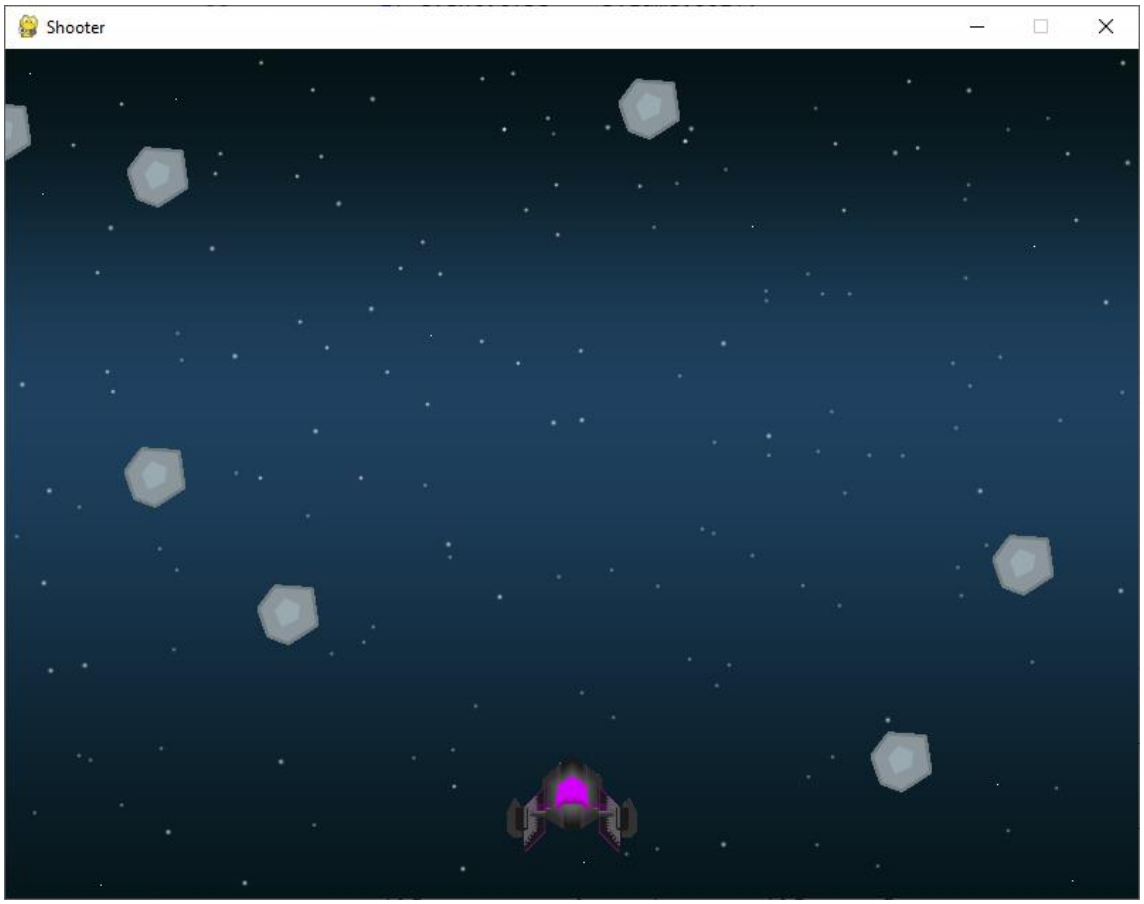
```
 def update(self):
        self.rect.y += self.speedy
        self.rect.x += self.speedx
        if self.rect.top > HEIGHT + 10 or self.rect.left < -25 or
self.rect.right > WIDTH + 25:
            self.rect.x = random.randrange(WIDTH - self.rect.width)
            self.rect.y = random.randrange(-100, -40)
            self.speedy = random.randrange(1, 10)
```

Hemos agregado estas 4 líneas para controlar si el meteorito se sale por la parte izquierda, derecha o la parte inferior de la ventana, si es así este tiene que volver a salir por la parte superior en modo aleatorio, como salen la primera vez.

Vamos a agregar un color de fondo:

```
background = pygame.image.load("assets/background.png").convert()


 # Insertar la imagen
    screen.blit(background, [0,0])
```

Cambiar el fill por blit agregando la imagen y las coordenadas.



En el siguiente capítulo vamos a añadir láser y controlar las colisiones.

El código completo hasta ahora:

```
import pygame, random

# Definimos las constantes para la dimensión de la ventana.
WIDTH = 800
HEIGHT = 600
# Definir color
BLACK = (0,0,0)
WHITE = (255, 255, 255)
# Inicializar pygame
pygame.init()
# Para la parte del sonido
pygame.mixer.init()
# Crear la ventana
screen = pygame.display.set_mode((WIDTH, HEIGHT))
# Título de la ventana
pygame.display.set_caption("Shooter")
# Para controlar los Frames por segundo
clock = pygame.time.Clock()

# Vamos a crear nuestra clase jugador
class Player(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        # Cargamos la imagen
        self.image = pygame.image.load("assets/player.png").convert()
        # Para borrar el borde negro
        self.image.set_colorkey(BLACK)
        # Definimos el rectángulo
        self.rect = self.image.get_rect()
        # Lo colocamos en la ventana
        self.rect.centerx = WIDTH // 2
        self.rect.bottom = HEIGHT - 10
        # Inicializamos una variable para la velocidad
        self.speed_x = 0

    def update(self):
        self.speed_x = 0
        # Para comprobar si presionamos flecha derecha e izquierda
        keystate = pygame.key.get_pressed()
        if keystate[pygame.K_LEFT]:
            self.speed_x = -5
        if keystate[pygame.K_RIGHT]:
            self.speed_x =  5
        # Para que no se salga de la derecha ni de la izquierda
        self.rect.x += self.speed_x
        if self.rect.right > WIDTH:
            self.rect.right = WIDTH
```

```
        if self.rect.left < 0:
            self.rect.left = 0

class Meteor(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image =
pygame.image.load("assets/meteorGrey_med1.png").convert()
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.x = random.randrange(WIDTH - self.rect.width)
        self.rect.y = random.randrange(-100, -40)
        self.speedy = random.randrange(1, 10)
        self.speedx = random.randrange(-5, 5)

    def update(self):
        self.rect.y += self.speedy
        self.rect.x += self.speedx
        if self.rect.top > HEIGHT + 10 or self.rect.left < -25 or
self.rect.right > WIDTH + 25:
            self.rect.x = random.randrange(WIDTH - self.rect.width)
            self.rect.y = random.randrange(-100, -40)
            self.speedy = random.randrange(1, 10)

# Cargar imagen de fondo
background = pygame.image.load("assets/background.png").convert()

# Creamos el grupo
all_sprites = pygame.sprite.Group()
meteor_list = pygame.sprite.Group()

# Creamos una intancia a Player
player = Player()
# Lo agregamos a all_sprites
all_sprites.add(player)

for i in range(8):
    meteor = Meteor()
    all_sprites.add(meteor)
    meteor_list.add(meteor)



# Bucle principal
running = True
while running:
    clock.tick(60)
    # Evento para salir de la ventana
    for event in pygame.event.get():
```

```
            if event.type == pygame.QUIT:
                running = False

        all_sprites.update()
        # Insertar la imagen
        screen.blit(background, [0,0])
        # Dibujar en la ventana
        all_sprites.draw(screen)
        pygame.display.flip()

    pygame.quit()
```

Parte 3 (Colisiones)

En este capítulo vamos a implementar los láseres, cada vez que presione la barra espaciadora que suene un laser y demás detectar las colisiones.

```
# Detectar las colisiones jugador - meteoro
    # el parámetro true los objeos que colisiones desaparecen
    hits = pygame.sprite.spritecollide(player, meteor_list, True)
    if hits:
        running = False
```

hits asume el valor True si el jugador colisiona con uno de los meteoritos, por ese motivo ponemos el parámetro metoer_list que es el conjunto de los 8 meteoros.

Cuando colisiona hists tiene el valor True con lo que se va a cumplir la condición running igual a False, esto hace que salgamos del bucle while running, al salir del bucle el programa finaliza.

Es para comprobar que funciona.

```
# Bucle principal
running = True
while running:
    clock.tick(60)
    # Evento para salir de la ventana
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_SPACE:
                player.shoot()
```

En el bucle while running: agregamos las tres últimas líneas, vamos a controlar que presionamos la barra espaciadora, si es así vamos a llamar al método shoot() que vamos a hacer a continuación.

Vamos a implementar otra clase para las balas.

```
class Bullet(pygame.sprite.Sprite):
    def __init__(self, x, y): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image = pygame.image.load("assets/laser1.png").convert()
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.y = y
        self.speedy = random.randrange(1, 10)
        self.rect.centerx = x
        self.speedy = -10

    def update(self):
        self.rect.y += self.speedy
        if self.rect.bottom < 0:
            self.kill()
```

A continuación creamos el siguiente método en la clase Bullet

```
def shoot(self):
        bullet = Bullet()
        all_sprites.add(bullet)
        bullets.add(bullet)
```

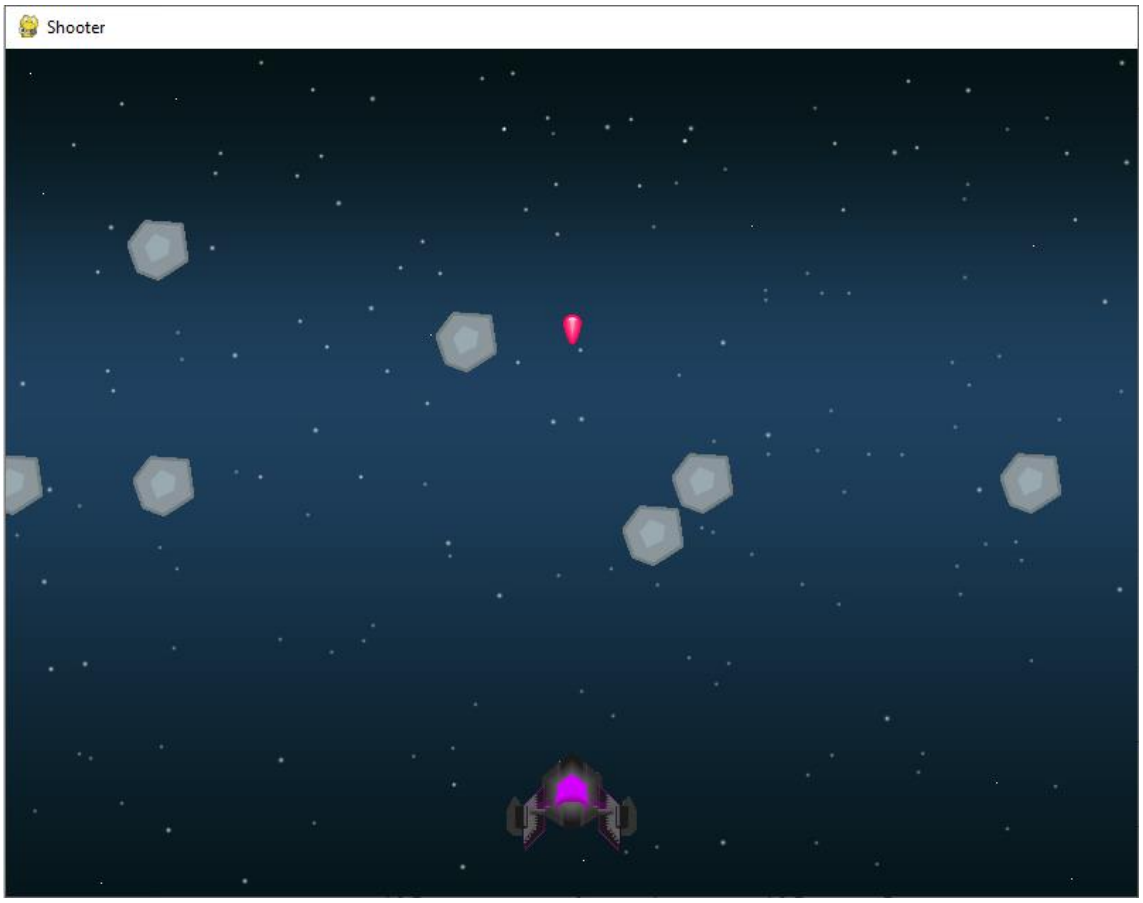
Creamos el siguiente método en la clase Player

```
def shoot(self):
        bullet = Bullet(self.rect.centerx, self.rect.top)
        all_sprites.add(bullet)
        bullets.add(bullet)
```

Definimos el grupo para los bullet.

```
# Creamos el grupo
all_sprites = pygame.sprite.Group()
meteor_list = pygame.sprite.Group()
bullets = pygame.sprite.Group()
```

Este será el resultado:



Adjunto el código completo hasta este capítulo:

```
import pygame, random

# Definimos las constantes para la dimensión de la ventana.
WIDTH = 800
HEIGHT = 600
# Definir color
BLACK = (0,0,0)
WHITE = (255, 255, 255)
# Inicializar pygame
pygame.init()
# Para la parte del sonido
pygame.mixer.init()
# Crear la ventana
screen = pygame.display.set_mode((WIDTH, HEIGHT))
# Título de la ventana
pygame.display.set_caption("Shooter")
# Para controlar los Frames por segundo
clock = pygame.time.Clock()

# Vamos a crear nuestra clase jugador
class Player(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        # Cargamos la imagen
        self.image = pygame.image.load("assets/player.png").convert()
        # Para borrar el borde negro
        self.image.set_colorkey(BLACK)
        # Definimos el rectángulo
        self.rect = self.image.get_rect()
        # Lo colocamos en la ventana
        self.rect.centerx = WIDTH // 2
        self.rect.bottom = HEIGHT - 10
        # Inicializamos una variable para la velocidad
        self.speed_x = 0

    def update(self):
        self.speed_x = 0
        # Para comprobar si presionamos flecha derecha e izquierda
        keystate = pygame.key.get_pressed()
        if keystate[pygame.K_LEFT]:
            self.speed_x = -5
        if keystate[pygame.K_RIGHT]:
            self.speed_x =  5
        # Para que no se salga de la derecha ni de la izquierda
        self.rect.x += self.speed_x
        if self.rect.right > WIDTH:
            self.rect.right = WIDTH
```

```
        if self.rect.left < 0:
            self.rect.left = 0

    def shoot(self):
        bullet = Bullet(self.rect.centerx, self.rect.top)
        all_sprites.add(bullet)
        bullets.add(bullet)


class Meteor(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image =
pygame.image.load("assets/meteorGrey_med1.png").convert()
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.x = random.randrange(WIDTH - self.rect.width)
        self.rect.y = random.randrange(-100, -40)
        self.speedy = random.randrange(1, 10)
        self.speedx = random.randrange(-5, 5)

    def update(self):
        self.rect.y += self.speedy
        self.rect.x += self.speedx
        if self.rect.top > HEIGHT + 10 or self.rect.left < -25 or
self.rect.right > WIDTH + 25:
            self.rect.x = random.randrange(WIDTH - self.rect.width)
            self.rect.y = random.randrange(-100, -40)
            self.speedy = random.randrange(1, 10)

class Bullet(pygame.sprite.Sprite):
    def __init__(self, x, y): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image = pygame.image.load("assets/laser1.png").convert()
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.y = y
        self.speedy = random.randrange(1, 10)
        self.rect.centerx = x
        self.speedy = -10

    def update(self):
        self.rect.y += self.speedy
        if self.rect.bottom < 0:
            self.kill()


# Cargar imagen de fondo
background = pygame.image.load("assets/background.png").convert()
```

```
# Creamos el grupo
all_sprites = pygame.sprite.Group()
meteor_list = pygame.sprite.Group()
bullets = pygame.sprite.Group()

# Creamos una intancia a Player
player = Player()
# Lo agregamos a all_sprites
all_sprites.add(player)

for i in range(8):
    meteor = Meteor()
    all_sprites.add(meteor)
    meteor_list.add(meteor)



# Bucle principal
running = True
while running:
    clock.tick(60)
    # Evento para salir de la ventana
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_SPACE:
                player.shoot()


    all_sprites.update()
    # Detectar las colisiones jugador - meteoro
    # el parámetro true los objeos que colisiones desaparecen
    hits = pygame.sprite.spritecollide(player, meteor_list, True)
    if hits:
        running = False
    # Insertar la imagen
    screen.blit(background, [0,0])
    # Dibujar en la ventana
    all_sprites.draw(screen)
    pygame.display.flip()

pygame.quit()
```

Parte 4 (Marcador)

En este capítulo vamos a implementar el marcador.

Creamos una variable:

```
# Variable para llevar el marcador
score = 0
```

Creamos la siguiente función, esta no pertenece a ninguna clase:

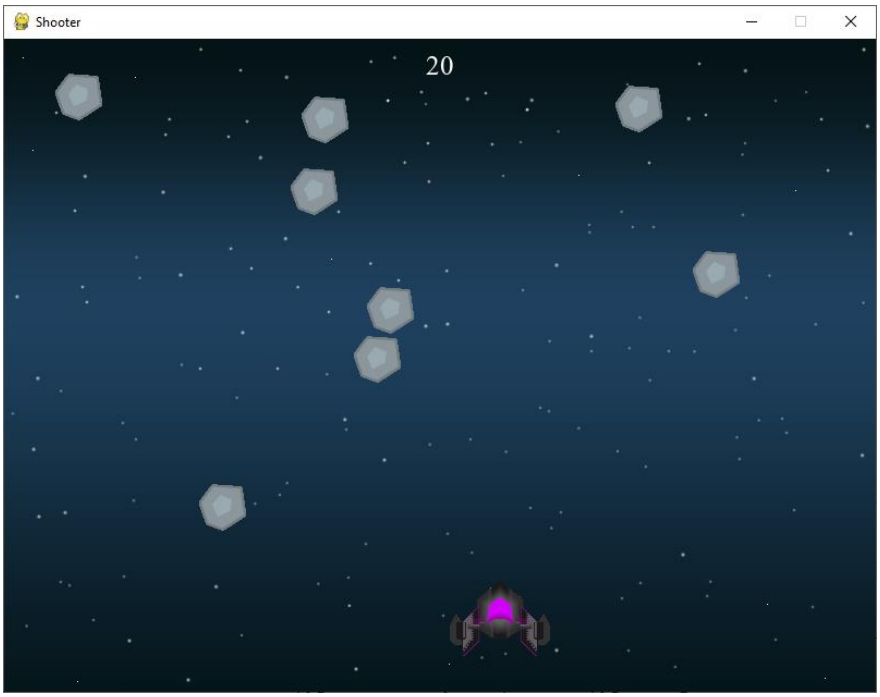
```
def draw_text(surface, text, size, x, y):
    font = pygame.font.SysFont("serif", size)
    text_surface = font.render(text,True,WHITE) # True: definición de
texto
    text_rect = text_surface.get_rect()
    text_rect.midtop = (x, y)
    surface.blit(text_surface, text_rect)
```

```
Colisión meteoro - laser
# Colision - meteo - laser
    hits = pygame.sprite.groupcollide(meteor_list, bullets, True, True)
    for hit in hits:
        score += 10
        meteor = Meteor()
        all_sprites.add(meteor)
        meteor_list.add(meteor)
```

El True, True hacen que desaparezcan la bala y el meteorito.

Agregamos este Código al final porque queremos que se sitúe por encima de los demás gráficos.

```
    # Marcador
    draw_text(screen, str(score), 25, WIDTH // 2, 10 )

    pygame.display.flip()

pygame.quit()
```



Ya tenemos un Contador que suma cuando disparamos a los meteoritos.

Aquí tienes el código completo hasta este capítulo.

```
import pygame, random

# Definimos las constantes para la dimensión de la ventana.
WIDTH = 800
HEIGHT = 600
# Definir color
BLACK = (0,0,0)
WHITE = (255, 255, 255)
# Inicializar pygame
pygame.init()
# Para la parte del sonido
pygame.mixer.init()
# Crear la ventana
screen = pygame.display.set_mode((WIDTH, HEIGHT))
# Título de la ventana
pygame.display.set_caption("Shooter")
# Para controlar los Frames por segundo
clock = pygame.time.Clock()

def draw_text(surface, text, size, x, y):
    font = pygame.font.SysFont("serif", size)
    text_surface = font.render(text,True,WHITE) # True: definición de
texto
    text_rect = text_surface.get_rect()
    text_rect.midtop = (x, y)
    surface.blit(text_surface, text_rect)
```

```
# Vamos a crear nuestra clase jugador
class Player(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        # Cargamos la imagen
        self.image = pygame.image.load("assets/player.png").convert()
        # Para borrar el borde negro
        self.image.set_colorkey(BLACK)
        # Definimos el rectángulo
        self.rect = self.image.get_rect()
        # Lo colocamos en la ventana
        self.rect.centerx = WIDTH // 2
        self.rect.bottom = HEIGHT - 10
        # Inicializamos una variable para la velocidad
        self.speed_x = 0

    def update(self):
        self.speed_x = 0
        # Para comprobar si presionamos flecha derecha e izquierda
        keystate = pygame.key.get_pressed()
        if keystate[pygame.K_LEFT]:
            self.speed_x = -5
        if keystate[pygame.K_RIGHT]:
            self.speed_x =  5
        # Para que no se salga de la derecha ni de la izquierda
        self.rect.x += self.speed_x
        if self.rect.right > WIDTH:
            self.rect.right = WIDTH
        if self.rect.left < 0:
            self.rect.left = 0

    def shoot(self):
        bullet = Bullet(self.rect.centerx, self.rect.top)
        all_sprites.add(bullet)
        bullets.add(bullet)


class Meteor(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image =
pygame.image.load("assets/meteorGrey_med1.png").convert()
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.x = random.randrange(WIDTH - self.rect.width)
        self.rect.y = random.randrange(-100, -40)
        self.speedy = random.randrange(1, 10)
        self.speedx = random.randrange(-5, 5)
```

```
    def update(self):
        self.rect.y += self.speedy
        self.rect.x += self.speedx
        if self.rect.top > HEIGHT + 10 or self.rect.left < -25 or
self.rect.right > WIDTH + 25:
            self.rect.x = random.randrange(WIDTH - self.rect.width)
            self.rect.y = random.randrange(-100, -40)
            self.speedy = random.randrange(1, 10)

class Bullet(pygame.sprite.Sprite):
    def __init__(self, x, y): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image = pygame.image.load("assets/laser1.png").convert()
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.y = y
        self.speedy = random.randrange(1, 10)
        self.rect.centerx = x
        self.speedy = -10

    def update(self):
        self.rect.y += self.speedy
        if self.rect.bottom < 0:
            self.kill()


# Cargar imagen de fondo
background = pygame.image.load("assets/background.png").convert()

# Creamos el grupo
all_sprites = pygame.sprite.Group()
meteor_list = pygame.sprite.Group()
bullets = pygame.sprite.Group()

# Creamos una intancia a Player
player = Player()
# Lo agregamos a all_sprites
all_sprites.add(player)

for i in range(8):
    meteor = Meteor()
    all_sprites.add(meteor)
    meteor_list.add(meteor)

# Variable para llevar el marcador
score = 0

# Bucle principal
running = True
```

```
while running:
    clock.tick(60)
    # Evento para salir de la ventana
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_SPACE:
                player.shoot()

    all_sprites.update()

    # Colision - meteo - laser
    hits = pygame.sprite.groupcollide(meteor_list, bullets, True, True)
    for hit in hits:
        score += 10
        meteor = Meteor()
        all_sprites.add(meteor)
        meteor_list.add(meteor)

    # Detectar las colisiones jugador - meteoro
    # el parámetro true los objeos que colisiones desaparecen
    hits = pygame.sprite.spritecollide(player, meteor_list, True)
    if hits:
        running = False
    # Insertar la imagen
    screen.blit(background, [0,0])
    # Dibujar en la ventana
    all_sprites.draw(screen)
    # Marcador
    draw_text(screen, str(score), 25, WIDTH // 2, 10 )

    pygame.display.flip()

pygame.quit()
```

Parte 5 (Enemigos)

En este capítulo queremos cargar todos los tamaños de meteros y que salgan aleatoriamente de distintos tamaños.

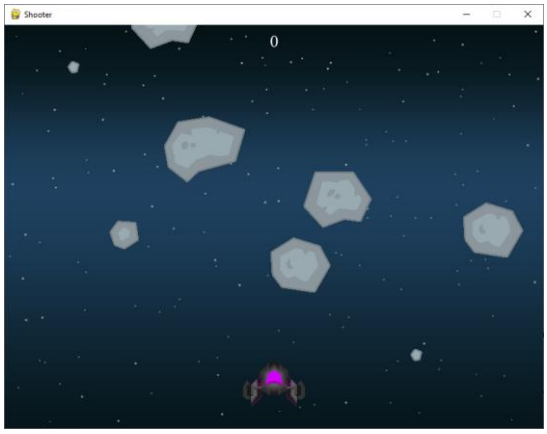
Vamos a cargar los meteoritos:

```
# Cargar todos los meteroritos
meteor_images = []
meteor_list = [ "assets/meteorGrey_big1.png",
                "assets/meteorGrey_big2.png",
                "assets/meteorGrey_big3.png",
                "assets/meteorGrey_big4.png",
                "assets/meteorGrey_med1.png",
                "assets/meteorGrey_med2.png",
                "assets/meteorGrey_small1.png",
                "assets/meteorGrey_small2.png",
                "assets/meteorGrey_tiny1.png",
                "assets/meteorGrey_tiny2.png"
                ]
for img in meteor_list:
    meteor_images.append(pygame.image.load(img).convert())
```

De la siguiente clase hemos cambiado:

```
class Meteor(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image = random.choice(meteor_images)
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.x = random.randrange(WIDTH - self.rect.width)
        self.rect.y = random.randrange(-100, -40)
        self.speedy = random.randrange(1, 10)
        self.speedx = random.randrange(-5, 5)
```

De la lista meteor_images selecciona cada ver una aleatoriamente.



```
class Meteor(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image = random.choice(meteor_images)
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.x = random.randrange(WIDTH - self.rect.width)
        self.rect.y = random.randrange(-140, -100)
        self.speedy = random.randrange(1, 10)
        self.speedx = random.randrange(-5, 5)
```

De la clase Meteor cambiamos el rango para la coordenada y de -140 y -100.

```
 def update(self):
        self.rect.y += self.speedy
        self.rect.x += self.speedx
        if self.rect.top > HEIGHT + 10 or self.rect.left < -40 or
self.rect.right > WIDTH + 40:
            self.rect.x = random.randrange(WIDTH - self.rect.width)
            self.rect.y = random.randrange(-100, -40)
            self.speedy = random.randrange(1, 10)
```

También hemos cambiado las siguientes coordenadas.

Te adjunto el código completo hasta este capítulo:

```
import pygame, random

# Definimos las constantes para la dimensión de la ventana.
WIDTH = 800
HEIGHT = 600
# Definir color
BLACK = (0,0,0)
WHITE = (255, 255, 255)
# Inicializar pygame
pygame.init()
# Para la parte del sonido
pygame.mixer.init()
# Crear la ventana
screen = pygame.display.set_mode((WIDTH, HEIGHT))
# Título de la ventana
pygame.display.set_caption("Shooter")
# Para controlar los Frames por segundo
clock = pygame.time.Clock()

def draw_text(surface, text, size, x, y):
    font = pygame.font.SysFont("serif", size)
    text_surface = font.render(text,True,WHITE) # True: definición de
texto
```

```
    text_rect = text_surface.get_rect()
    text_rect.midtop = (x, y)
    surface.blit(text_surface, text_rect)

# Vamos a crear nuestra clase jugador
class Player(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        # Cargamos la imagen
        self.image = pygame.image.load("assets/player.png").convert()
        # Para borrar el borde negro
        self.image.set_colorkey(BLACK)
        # Definimos el rectángulo
        self.rect = self.image.get_rect()
        # Lo colocamos en la ventana
        self.rect.centerx = WIDTH // 2
        self.rect.bottom = HEIGHT - 10
        # Inicializamos una variable para la velocidad
        self.speed_x = 0

    def update(self):
        self.speed_x = 0
        # Para comprobar si presionamos flecha derecha e izquierda
        keystate = pygame.key.get_pressed()
        if keystate[pygame.K_LEFT]:
            self.speed_x = -5
        if keystate[pygame.K_RIGHT]:
            self.speed_x =  5
        # Para que no se salga de la derecha ni de la izquierda
        self.rect.x += self.speed_x
        if self.rect.right > WIDTH:
            self.rect.right = WIDTH
        if self.rect.left < 0:
            self.rect.left = 0

    def shoot(self):
        bullet = Bullet(self.rect.centerx, self.rect.top)
        all_sprites.add(bullet)
        bullets.add(bullet)


class Meteor(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image = random.choice(meteor_images)
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.x = random.randrange(WIDTH - self.rect.width)
        self.rect.y = random.randrange(-140, -100)
```

```
        self.speedy = random.randrange(1, 10)
        self.speedx = random.randrange(-5, 5)

    def update(self):
        self.rect.y += self.speedy
        self.rect.x += self.speedx
        if self.rect.top > HEIGHT + 10 or self.rect.left < -40 or
self.rect.right > WIDTH + 40:
            self.rect.x = random.randrange(WIDTH - self.rect.width)
            self.rect.y = random.randrange(-100, -40)
            self.speedy = random.randrange(1, 10)

class Bullet(pygame.sprite.Sprite):
    def __init__(self, x, y): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image = pygame.image.load("assets/laser1.png").convert()
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.y = y
        self.speedy = random.randrange(1, 10)
        self.rect.centerx = x
        self.speedy = -10

    def update(self):
        self.rect.y += self.speedy
        if self.rect.bottom < 0:
            self.kill()

# Cargar todos los meteoritos
meteor_images = []
meteor_list = [ "assets/meteorGrey_big1.png",
                "assets/meteorGrey_big2.png",
                "assets/meteorGrey_big3.png",
                "assets/meteorGrey_big4.png",
                "assets/meteorGrey_med1.png",
                "assets/meteorGrey_med2.png",
                "assets/meteorGrey_small1.png",
                "assets/meteorGrey_small2.png",
                "assets/meteorGrey_tiny1.png",
                "assets/meteorGrey_tiny2.png"
                ]
for img in meteor_list:
    meteor_images.append(pygame.image.load(img).convert())



# Cargar imagen de fondo
background = pygame.image.load("assets/background.png").convert()
```

```
# Creamos el grupo
all_sprites = pygame.sprite.Group()
meteor_list = pygame.sprite.Group()
bullets = pygame.sprite.Group()

# Creamos una intancia a Player
player = Player()
# Lo agregamos a all_sprites
all_sprites.add(player)

for i in range(8):
    meteor = Meteor()
    all_sprites.add(meteor)
    meteor_list.add(meteor)

# Variable para llevar el marcador
score = 0

# Bucle principal
running = True
while running:
    clock.tick(60)
    # Evento para salir de la ventana
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_SPACE:
                player.shoot()

    all_sprites.update()

    # Colision - meteo - laser
    hits = pygame.sprite.groupcollide(meteor_list, bullets, True, True)
    for hit in hits:
        score += 10
        meteor = Meteor()
        all_sprites.add(meteor)
        meteor_list.add(meteor)

    # Detectar las colisiones jugador - meteoro
    # el parámetro true los objeos que colisiones desaparecen
    hits = pygame.sprite.spritecollide(player, meteor_list, True)
    if hits:
        running = False
    # Insertar la imagen
    screen.blit(background, [0,0])
    # Dibujar en la ventana
```

```
    all_sprites.draw(screen)
    # Marcador
    draw_text(screen, str(score), 25, WIDTH // 2, 10 )

    pygame.display.flip()

pygame.quit()
```

Parte 6 (Sonido)

En el Proyecto vamos a trabajar con 3 archivos de sonido:


```
# Cargar sonido
laser_sound = pygame.mixer.Sound("assets/laser5.ogg")
```

Cargamos el láser, lo puedes situar antes de # Creamos el grupo.

Ahora nos vamos a la clase Player

```
 def shoot(self):
        bullet = Bullet(self.rect.centerx, self.rect.top)
        all_sprites.add(bullet)
        bullets.add(bullet)
        laser_sound.play()
```

Para que se reproduzca el láser.

Ahora para las explosiones y la música de fondo.

```
# Cargar sonido
laser_sound = pygame.mixer.Sound("assets/laser5.ogg")
explosion_sound = pygame.mixer.Sound("assets/explosion.wav")
pygame.mixer.music.load("assets/music.ogg")
```

Ahora para reproducir las explosiones.

```
 # Colision - meteo - laser
    hits = pygame.sprite.groupcollide(meteor_list, bullets, True, True)
    for hit in hits:
        score += 10
        explosion_sound.play()
        meteor = Meteor()
        all_sprites.add(meteor)
        meteor_list.add(meteor)
```

Para controlar el volumen de la música de fondo.

```
# Cargar sonido
laser_sound = pygame.mixer.Sound("assets/laser5.ogg")
explosion_sound = pygame.mixer.Sound("assets/explosion.wav")
```

```
pygame.mixer.music.load("assets/music.ogg")
pygame.mixer.music.set_volume(0.2)
```

Cuanto más alto sea el valor el volumen de la música subirá, los valores están entre 0 y 1.

Para reproducir la música.

```
# Variable para llevar el marcador
score = 0
pygame.mixer.music.play(loops=-1)
```

Para que la música se reproduzca continuamente haremos un bucle de loops=-1. Si ponemos 1, 2, 3, etc. se va a repetir el número de veces que especifiquemos.

Si antes de terminar este proyecto no quieres que se reproduzca la música pásala a comentario.

```
# pygame.mixer.music.play(loops=-1)
```

Al final borraremos la almohadilla.

Te adjunto el código completo hasta este capítulo:

```
import pygame, random

# Definimos las constantes para la dimensión de la ventana.
WIDTH = 800
HEIGHT = 600
# Definir color
BLACK = (0,0,0)
WHITE = (255, 255, 255)
# Inicializar pygame
pygame.init()
# Para la parte del sonido
pygame.mixer.init()
# Crear la ventana
screen = pygame.display.set_mode((WIDTH, HEIGHT))
# Título de la ventana
pygame.display.set_caption("Shooter")
# Para controlar los Frames por segundo
clock = pygame.time.Clock()

def draw_text(surface, text, size, x, y):
    font = pygame.font.SysFont("serif", size)
    text_surface = font.render(text,True,WHITE) # True: definición de
texto
    text_rect = text_surface.get_rect()
    text_rect.midtop = (x, y)
    surface.blit(text_surface, text_rect)
```

```
# Vamos a crear nuestra clase jugador
class Player(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        # Cargamos la imagen
        self.image = pygame.image.load("assets/player.png").convert()
        # Para borrar el borde negro
        self.image.set_colorkey(BLACK)
        # Definimos el rectángulo
        self.rect = self.image.get_rect()
        # Lo colocamos en la ventana
        self.rect.centerx = WIDTH // 2
        self.rect.bottom = HEIGHT - 10
        # Inicializamos una variable para la velocidad
        self.speed_x = 0

    def update(self):
        self.speed_x = 0
        # Para comprobar si presionamos flecha derecha e izquierda
        keystate = pygame.key.get_pressed()
        if keystate[pygame.K_LEFT]:
            self.speed_x = -5
        if keystate[pygame.K_RIGHT]:
            self.speed_x =  5
        # Para que no se salga de la derecha ni de la izquierda
        self.rect.x += self.speed_x
        if self.rect.right > WIDTH:
            self.rect.right = WIDTH
        if self.rect.left < 0:
            self.rect.left = 0

    def shoot(self):
        bullet = Bullet(self.rect.centerx, self.rect.top)
        all_sprites.add(bullet)
        bullets.add(bullet)
        laser_sound.play()


class Meteor(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image = random.choice(meteor_images)
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.x = random.randrange(WIDTH - self.rect.width)
        self.rect.y = random.randrange(-140, -100)
        self.speedy = random.randrange(1, 10)
        self.speedx = random.randrange(-5, 5)
```

```
    def update(self):
        self.rect.y += self.speedy
        self.rect.x += self.speedx
        if self.rect.top > HEIGHT + 10 or self.rect.left < -40 or
self.rect.right > WIDTH + 40:
            self.rect.x = random.randrange(WIDTH - self.rect.width)
            self.rect.y = random.randrange(-100, -40)
            self.speedy = random.randrange(1, 10)

class Bullet(pygame.sprite.Sprite):
    def __init__(self, x, y): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image = pygame.image.load("assets/laser1.png").convert()
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.y = y
        self.speedy = random.randrange(1, 10)
        self.rect.centerx = x
        self.speedy = -10

    def update(self):
        self.rect.y += self.speedy
        if self.rect.bottom < 0:
            self.kill()

# Cargar todos los meteoritos
meteor_images = []
meteor_list = [ "assets/meteorGrey_big1.png",
                "assets/meteorGrey_big2.png",
                "assets/meteorGrey_big3.png",
                "assets/meteorGrey_big4.png",
                "assets/meteorGrey_med1.png",
                "assets/meteorGrey_med2.png",
                "assets/meteorGrey_small1.png",
                "assets/meteorGrey_small2.png",
                "assets/meteorGrey_tiny1.png",
                "assets/meteorGrey_tiny2.png"
                ]
for img in meteor_list:
    meteor_images.append(pygame.image.load(img).convert())

# Cargar imagen de fondo
background = pygame.image.load("assets/background.png").convert()

# Cargar sonido
laser_sound = pygame.mixer.Sound("assets/laser5.ogg")
explosion_sound = pygame.mixer.Sound("assets/explosion.wav")
pygame.mixer.music.load("assets/music.ogg")
```

```
pygame.mixer.music.set_volume(0.2)

# Creamos el grupo
all_sprites = pygame.sprite.Group()
meteor_list = pygame.sprite.Group()
bullets = pygame.sprite.Group()

# Creamos una intancia a Player
player = Player()
# Lo agregamos a all_sprites
all_sprites.add(player)

for i in range(8):
    meteor = Meteor()
    all_sprites.add(meteor)
    meteor_list.add(meteor)

# Variable para llevar el marcador
score = 0
# pygame.mixer.music.play(loops=-1)

# Bucle principal
running = True
while running:
    clock.tick(60)
    # Evento para salir de la ventana
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_SPACE:
                player.shoot()

    all_sprites.update()

    # Colision - meteo - laser
    hits = pygame.sprite.groupcollide(meteor_list, bullets, True, True)
    for hit in hits:
        score += 10
        explosion_sound.play()
        meteor = Meteor()
        all_sprites.add(meteor)
        meteor_list.add(meteor)

    # Detectar las colisiones jugador - meteoro
    # el parámetro true los objeos que colisiones desaparecen
    hits = pygame.sprite.spritecollide(player, meteor_list, True)
    if hits:
        running = False
```

```
    # Insertar la imagen
    screen.blit(background, [0,0])
    # Dibujar en la ventana
    all_sprites.draw(screen)
    # Marcador
    draw_text(screen, str(score), 25, WIDTH // 2, 10 )

    pygame.display.flip()

pygame.quit()
```

Parte 7 (Escudos)

En este vídeo implementaremos algo muy interesante que es¿ ¿Cómo poner un escudo?, es decir una barra de salud, para hacer esto nos vamos a nuestra clase de Player.

```
class Player(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        # Cargamos la imagen
        self.image = pygame.image.load("assets/player.png").convert()
        # Para borrar el borde negro
        self.image.set_colorkey(BLACK)
        # Definimos el rectángulo
        self.rect = self.image.get_rect()
        # Lo colocamos en la ventana
        self.rect.centerx = WIDTH // 2
        self.rect.bottom = HEIGHT - 10
        # Inicializamos una variable para la velocidad
        self.speed_x = 0
        # El valor del escudo
        self.shield = 100
```

Agregamos una variable para el escudo.

```
# Detectar las colisiones jugador - meteoro
    # el parámetro true los objeos que colisiones desaparecen
    hits = pygame.sprite.spritecollide(player, meteor_list, True)
    for hit in hits:
        player.shield -= 25
        if player.shield <= 0:
            running = False
```

ahora para que se termine el juego me tienen que tocar 4 meteoritos. Hay un ciclo for porque tiene que ir comparando cada uno de los meteoritos que hay en la lista meteor_list.

```
# Detectar las colisiones jugador - meteoro
    # el parámetro true los objeos que colisiones desaparecen
    hits = pygame.sprite.spritecollide(player, meteor_list, True)
    for hit in hits:
        player.shield -= 25
        meteor = Meteor()
        all_sprites.add(meteor)
        meteor_list.add(meteor)
        if player.shield <= 0:
            running = False
```

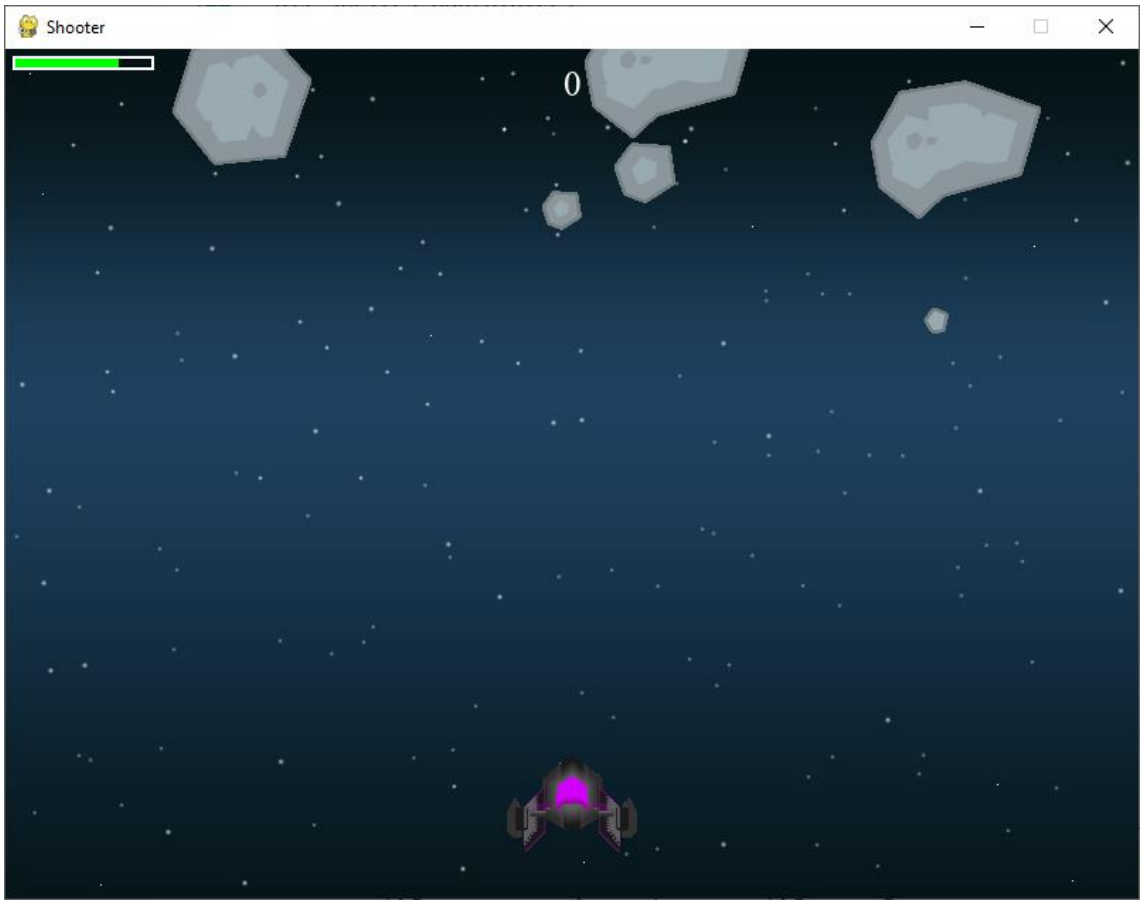
Cada vez que se colisiona con un meteoro este ya no se va a rehacer por este motivo creamos un nuevo meteoro.

Vamos a dibujar la barra de salud.

```
 # Marcador
    draw_text(screen, str(score), 25, WIDTH // 2, 10 )

    # Escudo
    draw_shield_bar(screen, 5, 5, player.shield)

    pygame.display.flip()

pygame.quit()
```

Llamaremos a esta función que ahora vamos a escribir fuera de las clases.

```
def draw_shield_bar(surface, x, y, percentage):
    BAR_LENGHT = 100
    BAR_HEIGHT = 10
    fill = (percentage / 100) * BAR_LENGHT
    border = pygame.Rect(x, y, BAR_LENGHT, BAR_HEIGHT)
    fill = pygame.Rect(x, y, fill, BAR_HEIGHT)
    pygame.draw.rect(surface, GREEN, fill)
    pygame.draw.rect(surface, WHITE, border, 2 )
```



La barra tiene para 4 vidas.

Adjunto todo el código realizado hasta este capítulo:

```
import pygame, random

# Definimos las constantes para la dimensión de la ventana.
WIDTH = 800
HEIGHT = 600
# Definir color
BLACK = (0,0,0)
WHITE = (255, 255, 255)
GREEN = (0, 255, 0)
# Inicializar pygame
pygame.init()
# Para la parte del sonido
pygame.mixer.init()
# Crear la ventana
screen = pygame.display.set_mode((WIDTH, HEIGHT))
# Título de la ventana
pygame.display.set_caption("Shooter")
# Para controlar los Frames por segundo
clock = pygame.time.Clock()

def draw_text(surface, text, size, x, y):
    font = pygame.font.SysFont("serif", size)
    text_surface = font.render(text,True,WHITE) # True: definición de
texto
    text_rect = text_surface.get_rect()
    text_rect.midtop = (x, y)
    surface.blit(text_surface, text_rect)

def draw_shield_bar(surface, x, y, percentage):
    BAR_LENGHT = 100
    BAR_HEIGHT = 10
    fill = (percentage / 100) * BAR_LENGHT
    border = pygame.Rect(x, y, BAR_LENGHT, BAR_HEIGHT)
    fill = pygame.Rect(x, y, fill, BAR_HEIGHT)
    pygame.draw.rect(surface, GREEN, fill)
    pygame.draw.rect(surface, WHITE, border, 2 )

# Vamos a crear nuestra clase jugador
class Player(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        # Cargamos la imagen
        self.image = pygame.image.load("assets/player.png").convert()
        # Para borrar el borde negro
        self.image.set_colorkey(BLACK)
        # Definimos el rectángulo
        self.rect = self.image.get_rect()
```

```
        # Lo colocamos en la ventana
        self.rect.centerx = WIDTH // 2
        self.rect.bottom = HEIGHT - 10
        # Inicializamos una variable para la velocidad
        self.speed_x = 0
        # El valor del escudo
        self.shield = 100

    def update(self):
        self.speed_x = 0
        # Para comprobar si presionamos flecha derecha e izquierda
        keystate = pygame.key.get_pressed()
        if keystate[pygame.K_LEFT]:
            self.speed_x = -5
        if keystate[pygame.K_RIGHT]:
            self.speed_x =  5
        # Para que no se salga de la derecha ni de la izquierda
        self.rect.x += self.speed_x
        if self.rect.right > WIDTH:
            self.rect.right = WIDTH
        if self.rect.left < 0:
            self.rect.left = 0

    def shoot(self):
        bullet = Bullet(self.rect.centerx, self.rect.top)
        all_sprites.add(bullet)
        bullets.add(bullet)
        laser_sound.play()

class Meteor(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image = random.choice(meteor_images)
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.x = random.randrange(WIDTH - self.rect.width)
        self.rect.y = random.randrange(-140, -100)
        self.speedy = random.randrange(1, 10)
        self.speedx = random.randrange(-5, 5)

    def update(self):
        self.rect.y += self.speedy
        self.rect.x += self.speedx
        if self.rect.top > HEIGHT + 10 or self.rect.left < -40 or
self.rect.right > WIDTH + 40:
            self.rect.x = random.randrange(WIDTH - self.rect.width)
            self.rect.y = random.randrange(-100, -40)
            self.speedy = random.randrange(1, 10)
```

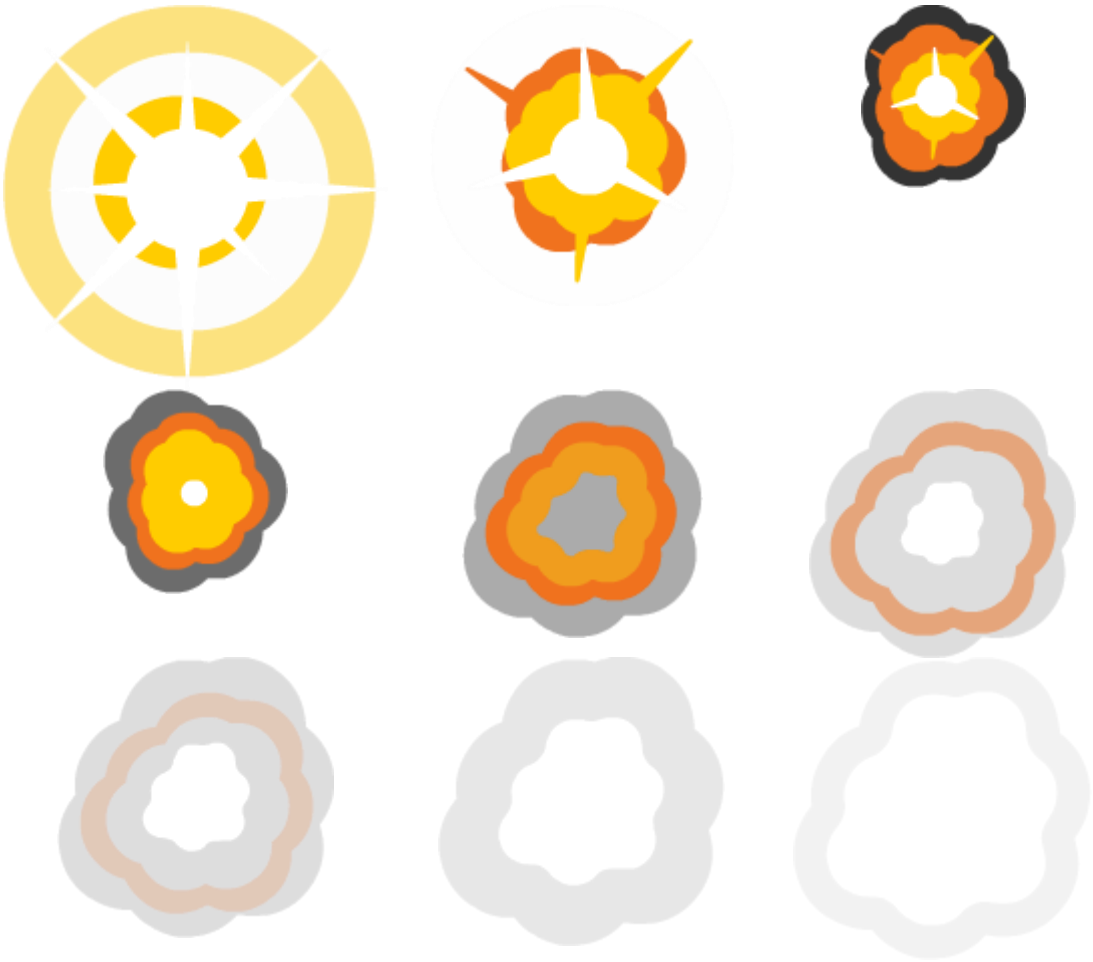
```
class Bullet(pygame.sprite.Sprite):
    def __init__(self, x, y): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image = pygame.image.load("assets/laser1.png").convert()
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.y = y
        self.speedy = random.randrange(1, 10)
        self.rect.centerx = x
        self.speedy = -10

    def update(self):
        self.rect.y += self.speedy
        if self.rect.bottom < 0:
            self.kill()

# Cargar todos los meteoritos
meteor_images = []
meteor_list = [ "assets/meteorGrey_big1.png",
                "assets/meteorGrey_big2.png",
                "assets/meteorGrey_big3.png",
                "assets/meteorGrey_big4.png",
                "assets/meteorGrey_med1.png",
                "assets/meteorGrey_med2.png",
                "assets/meteorGrey_small1.png",
                "assets/meteorGrey_small2.png",
                "assets/meteorGrey_tiny1.png",
                "assets/meteorGrey_tiny2.png"
                ]
for img in meteor_list:
    meteor_images.append(pygame.image.load(img).convert())

# Cargar imagen de fondo
background = pygame.image.load("assets/background.png").convert()

# Cargar sonido
laser_sound = pygame.mixer.Sound("assets/laser5.ogg")
explosion_sound = pygame.mixer.Sound("assets/explosion.wav")
pygame.mixer.music.load("assets/music.ogg")
pygame.mixer.music.set_volume(0.2)

# Creamos el grupo
all_sprites = pygame.sprite.Group()
meteor_list = pygame.sprite.Group()
bullets = pygame.sprite.Group()

# Creamos una intancia a Player
player = Player()
# Lo agregamos a all_sprites
```

```
all_sprites.add(player)

for i in range(8):
    meteor = Meteor()
    all_sprites.add(meteor)
    meteor_list.add(meteor)

# Variable para llevar el marcador
score = 0
pygame.mixer.music.play(loops=-1)

# Bucle principal
running = True
while running:
    clock.tick(60)
    # Evento para salir de la ventana
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_SPACE:
                player.shoot()

    all_sprites.update()

    # Colision - meteo - laser
    hits = pygame.sprite.groupcollide(meteor_list, bullets, True, True)
    for hit in hits:
        score += 10
        explosion_sound.play()
        meteor = Meteor()
        all_sprites.add(meteor)
        meteor_list.add(meteor)

    # Detectar las colisiones jugador - meteoro
    # el parámetro true los objeos que colisiones desaparecen
    hits = pygame.sprite.spritecollide(player, meteor_list, True)
    for hit in hits:
        player.shield -= 25
        meteor = Meteor()
        all_sprites.add(meteor)
        meteor_list.add(meteor)
        if player.shield <= 0:
            running = False
    # Insertar la imagen
    screen.blit(background, [0,0])
    # Dibujar en la ventana
    all_sprites.draw(screen)
    # Marcador
```

```
        draw_text(screen, str(score), 25, WIDTH // 2, 10 )

        # Escudo
        draw_shield_bar(screen, 5, 5, player.shield)

        pygame.display.flip()

    pygame.quit()
```

Parte 8 (Explosiones)

Cuando un láser toca a un meteorito se crea una animación de explosión.



```
# Explosión imagenes
explosion_anim = []
for i in range(9):
    file = "assets/regularExplosion0{}.png".format(i)
    img = pygame.image.load(file).convert()
    img.set_colorkey(BLACK)
    img_scale = pygame.transform.scale(img, (70,70))
    explosion_anim.append(img_scale)
```

Otra forma de cargar las imágenes de las explosiones.

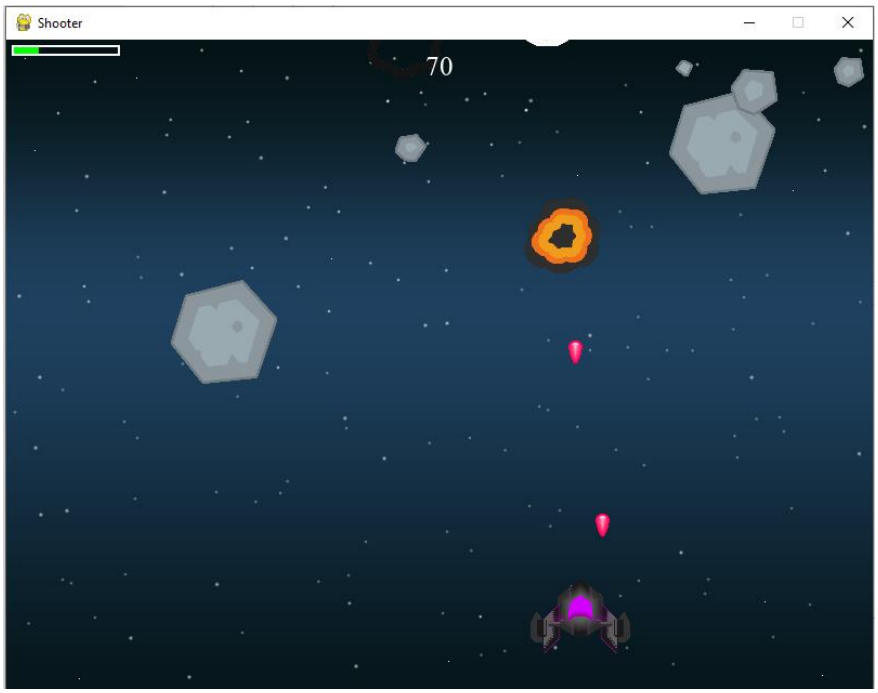
Ahora vamos a declarar una clase para las explosiones.

```
class Explosion(pygame.sprite.Sprite):
    def __init__(self, center): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image = explosion_anim[0]
        self.rect = self.image.get_rect()
        self.rect.center = center
```

```
        self.frame = 0
        self.last_update = pygame.time.get_ticks()
        self.frame_rate = 50 # Velocidad de la explosión

    def update(self):
        now = pygame.time.get_ticks() # cuanto tiempo ha transcurrido
        if now - self.last_update > self.frame_rate:
            self.last_update = now
            self.frame += 1
            if self.frame == len(explosion_anim):
                self.kill()
            else:
                center = self.rect.center
                self.image = explosion_anim[self.frame]
                self.rect = self.image.get_rect()
                self.rect.center = center
```

Ahora nos vamos a la parte inferior del código:

```
# Colision - meteo - laser
    hits = pygame.sprite.groupcollide(meteor_list, bullets, True, True)
    for hit in hits:
        score += 10
        explosion_sound.play()
        explosion = Explosion(hit.rect.center)
        all_sprites.add(explosion)
        meteor = Meteor()
        all_sprites.add(meteor)
        meteor_list.add(meteor)
```



Te adjunto todo el codigo realizado hasta este capítulo:

```
import pygame, random

# Definimos las constantes para la dimensión de la ventana.
WIDTH = 800
HEIGHT = 600
# Definir color
BLACK = (0,0,0)
WHITE = (255, 255, 255)
GREEN = (0, 255, 0)
# Inicializar pygame
pygame.init()
# Para la parte del sonido
pygame.mixer.init()
# Crear la ventana
screen = pygame.display.set_mode((WIDTH, HEIGHT))
# Título de la ventana
pygame.display.set_caption("Shooter")
# Para controlar los Frames por segundo
clock = pygame.time.Clock()

def draw_text(surface, text, size, x, y):
    font = pygame.font.SysFont("serif", size)
    text_surface = font.render(text,True,WHITE) # True: definición de
texto
    text_rect = text_surface.get_rect()
    text_rect.midtop = (x, y)
    surface.blit(text_surface, text_rect)

def draw_shield_bar(surface, x, y, percentage):
    BAR_LENGHT = 100
    BAR_HEIGHT = 10
    fill = (percentage / 100) * BAR_LENGHT
    border = pygame.Rect(x, y, BAR_LENGHT, BAR_HEIGHT)
    fill = pygame.Rect(x, y, fill, BAR_HEIGHT)
    pygame.draw.rect(surface, GREEN, fill)
    pygame.draw.rect(surface, WHITE, border, 2 )

# Vamos a crear nuestra clase jugador
class Player(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        # Cargamos la imagen
        self.image = pygame.image.load("assets/player.png").convert()
        # Para borrar el borde negro
        self.image.set_colorkey(BLACK)
        # Definimos el rectángulo
        self.rect = self.image.get_rect()
```

```
        # Lo colocamos en la ventana
        self.rect.centerx = WIDTH // 2
        self.rect.bottom = HEIGHT - 10
        # Inicializamos una variable para la velocidad
        self.speed_x = 0
        # El valor del escudo
        self.shield = 100

    def update(self):
        self.speed_x = 0
        # Para comprobar si presionamos flecha derecha e izquierda
        keystate = pygame.key.get_pressed()
        if keystate[pygame.K_LEFT]:
            self.speed_x = -5
        if keystate[pygame.K_RIGHT]:
            self.speed_x =  5
        # Para que no se salga de la derecha ni de la izquierda
        self.rect.x += self.speed_x
        if self.rect.right > WIDTH:
            self.rect.right = WIDTH
        if self.rect.left < 0:
            self.rect.left = 0

    def shoot(self):
        bullet = Bullet(self.rect.centerx, self.rect.top)
        all_sprites.add(bullet)
        bullets.add(bullet)
        laser_sound.play()

class Meteor(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image = random.choice(meteor_images)
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.x = random.randrange(WIDTH - self.rect.width)
        self.rect.y = random.randrange(-140, -100)
        self.speedy = random.randrange(1, 10)
        self.speedx = random.randrange(-5, 5)

    def update(self):
        self.rect.y += self.speedy
        self.rect.x += self.speedx
        if self.rect.top > HEIGHT + 10 or self.rect.left < -40 or
self.rect.right > WIDTH + 40:
            self.rect.x = random.randrange(WIDTH - self.rect.width)
            self.rect.y = random.randrange(-100, -40)
            self.speedy = random.randrange(1, 10)
```

```
class Explosion(pygame.sprite.Sprite):
    def __init__(self, center): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image = explosion_anim[0]
        self.rect = self.image.get_rect()
        self.rect.center = center
        self.frame = 0
        self.last_update = pygame.time.get_ticks()
        self.frame_rate = 50 # Velocidad de la explosión

    def update(self):
        now = pygame.time.get_ticks() # cuanto tiempo ha transcurrido
        if now - self.last_update > self.frame_rate:
            self.last_update = now
            self.frame += 1
            if self.frame == len(explosion_anim):
                self.kill()
            else:
                center = self.rect.center
                self.image = explosion_anim[self.frame]
                self.rect = self.image.get_rect()
                self.rect.center = center


class Bullet(pygame.sprite.Sprite):
    def __init__(self, x, y): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image = pygame.image.load("assets/laser1.png").convert()
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.y = y
        self.speedy = random.randrange(1, 10)
        self.rect.centerx = x
        self.speedy = -10

    def update(self):
        self.rect.y += self.speedy
        if self.rect.bottom < 0:
            self.kill()

# Cargar todos los meteoritos
meteor_images = []
meteor_list = [ "assets/meteorGrey_big1.png",
                "assets/meteorGrey_big2.png",
                "assets/meteorGrey_big3.png",
                "assets/meteorGrey_big4.png",
                "assets/meteorGrey_med1.png",
                "assets/meteorGrey_med2.png",
                "assets/meteorGrey_small1.png",
```

```
                "assets/meteorGrey_small2.png",
                "assets/meteorGrey_tiny1.png",
                "assets/meteorGrey_tiny2.png"
                ]
for img in meteor_list:
    meteor_images.append(pygame.image.load(img).convert())

# Explosión imagenes
explosion_anim = []
for i in range(9):
    file = "assets/regularExplosion0{}.png".format(i)
    img = pygame.image.load(file).convert()
    img.set_colorkey(BLACK)
    img_scale = pygame.transform.scale(img, (70,70))
    explosion_anim.append(img_scale)

# Cargar imagen de fondo
background = pygame.image.load("assets/background.png").convert()

# Cargar sonido
laser_sound = pygame.mixer.Sound("assets/laser5.ogg")
explosion_sound = pygame.mixer.Sound("assets/explosion.wav")
pygame.mixer.music.load("assets/music.ogg")
pygame.mixer.music.set_volume(0.2)

# Creamos el grupo
all_sprites = pygame.sprite.Group()
meteor_list = pygame.sprite.Group()
bullets = pygame.sprite.Group()

# Creamos una intancia a Player
player = Player()
# Lo agregamos a all_sprites
all_sprites.add(player)

for i in range(8):
    meteor = Meteor()
    all_sprites.add(meteor)
    meteor_list.add(meteor)

# Variable para llevar el marcador
score = 0
pygame.mixer.music.play(loops=-1)

# Bucle principal
running = True
while running:
    clock.tick(60)
    # Evento para salir de la ventana
```

```
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_SPACE:
                player.shoot()

    all_sprites.update()

    # Colision - meteo - laser
    hits = pygame.sprite.groupcollide(meteor_list, bullets, True, True)
    for hit in hits:
        score += 10
        explosion_sound.play()
        explosion = Explosion(hit.rect.center)
        all_sprites.add(explosion)
        meteor = Meteor()
        all_sprites.add(meteor)
        meteor_list.add(meteor)

    # Detectar las colisiones jugador - meteoro
    # el parámetro true los objeos que colisiones desaparecen
    hits = pygame.sprite.spritecollide(player, meteor_list, True)
    for hit in hits:
        player.shield -= 25
        meteor = Meteor()
        all_sprites.add(meteor)
        meteor_list.add(meteor)
        if player.shield <= 0:
            running = False
    # Insertar la imagen
    screen.blit(background, [0,0])
    # Dibujar en la ventana
    all_sprites.draw(screen)
    # Marcador
    draw_text(screen, str(score), 25, WIDTH // 2, 10 )

    # Escudo
    draw_shield_bar(screen, 5, 5, player.shield)

    pygame.display.flip()

pygame.quit()
```

Parte 9 Final (Game over)

Ya hemos llegado al último capítulo de este pequeño proyecto pero no deja de ser muy interesante.

Lo que queremos realizar en este capitulo es que aparezca la ventana de “GAME OVER”

```
# Variable para llevar el marcador
score = 0

pygame.mixer.music.play(loops=-1)

# Game Over
game_over = True
```

Definimos una variable de tipo bolean.

```
# Bucle principal
running = True
while running:
    if game_over:
        game_over = False
        # Creamos el grupo
        all_sprites = pygame.sprite.Group()
        meteor_list = pygame.sprite.Group()
        bullets = pygame.sprite.Group()

        # Creamos una intancia a Player
        player = Player()
        # Lo agregamos a all_sprites
        all_sprites.add(player)

        for i in range(8):
            meteor = Meteor()
            all_sprites.add(meteor)
            meteor_list.add(meteor)

        # Variable para llevar el marcador
        score = 0

    clock.tick(60)
    # Evento para salir de la ventana
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_SPACE:
                player.shoot()
```

Texto que hemos movido.

Movemos parte del código.

El texto a mover se encuentra después de las siguientes líneas.

```
# Cargar sonido
laser_sound = pygame.mixer.Sound("assets/laser5.ogg")
explosion_sound = pygame.mixer.Sound("assets/explosion.wav")
pygame.mixer.music.load("assets/music.ogg")
pygame.mixer.music.set_volume(0.2)
```

Cambiamos la siguiente línea.

```
# Detectar las colisiones jugador - meteoro
    # el parámetro true los objeos que colisiones desaparecen
    hits = pygame.sprite.spritecollide(player, meteor_list, True)
    for hit in hits:
        player.shield -= 25
        meteor = Meteor()
        all_sprites.add(meteor)
        meteor_list.add(meteor)
        if player.shield <= 0:
            game_over = True
```

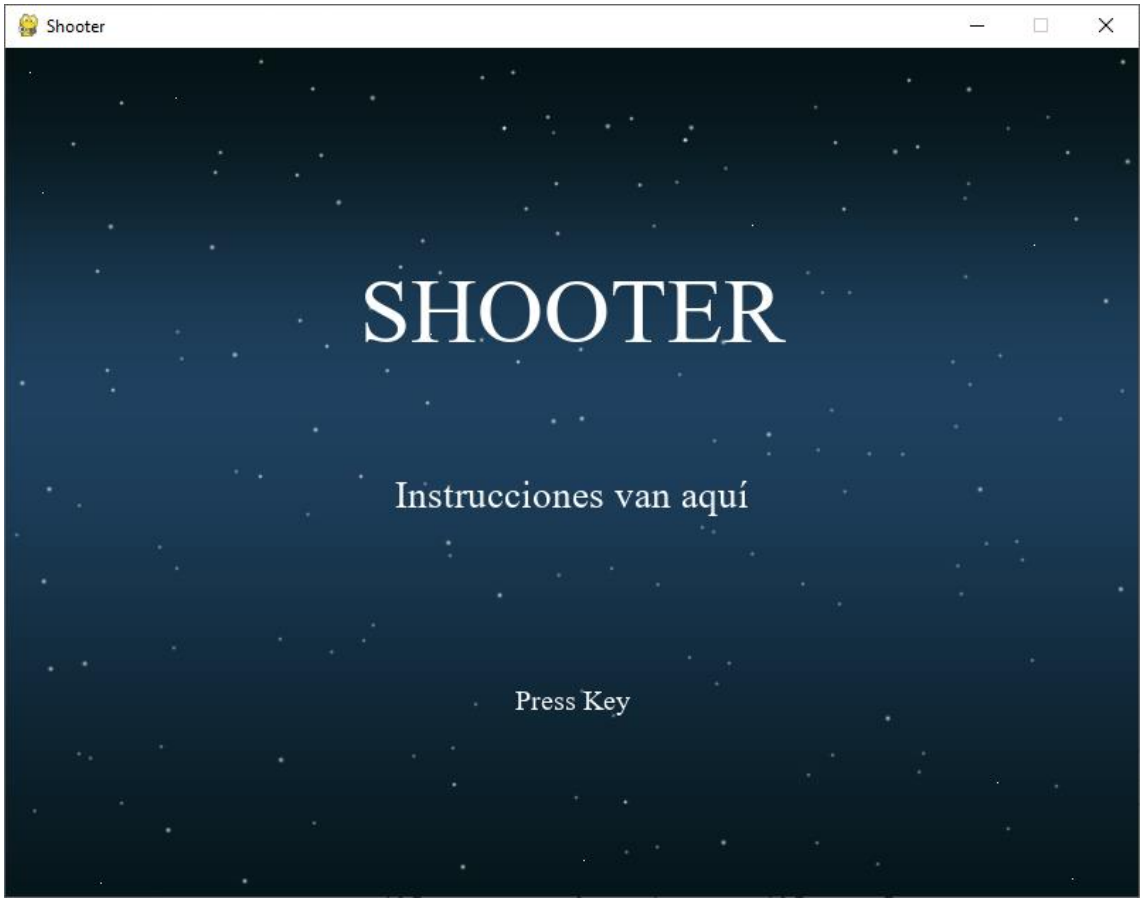
Vamos a llamar a la siguiente función que escribiremos a continuación.

```
# Bucle principal
running = True
while running:
    if game_over:

        show_go_screen()

        game_over = False
```

Vamos a crear la función show_go_screen()

```
def show_go_screen():
    screen.blit(background, [0,0])
    draw_text(screen, "SHOOTER", 65, WIDTH // 2, HEIGHT // 4)
    draw_text(screen, "Instrucciones van aquí", 27, WIDTH // 2, HEIGHT //
2)
    draw_text(screen, "Press Key", 20, WIDTH // 2, HEIGHT * 3/4)
    pygame.display.flip()
    waiting = True
    while waiting:
        clock.tick(60)
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
```

```
            if event.type == pygame.KEYUP:
                waiting = False
```



Adjunto todo el código del proyecto:

```
import pygame, random

# Definimos las constantes para la dimensión de la ventana.
WIDTH = 800
HEIGHT = 600
# Definir color
BLACK = (0,0,0)
WHITE = (255, 255, 255)
GREEN = (0, 255, 0)
# Inicializar pygame
pygame.init()
# Para la parte del sonido
pygame.mixer.init()
# Crear la ventana
screen = pygame.display.set_mode((WIDTH, HEIGHT))
# Título de la ventana
pygame.display.set_caption("Shooter")
# Para controlar los Frames por segundo
```

```
clock = pygame.time.Clock()

def draw_text(surface, text, size, x, y):
    font = pygame.font.SysFont("serif", size)
    text_surface = font.render(text,True,WHITE) # True: definición de
texto
    text_rect = text_surface.get_rect()
    text_rect.midtop = (x, y)
    surface.blit(text_surface, text_rect)

def draw_shield_bar(surface, x, y, percentage):
    BAR_LENGHT = 100
    BAR_HEIGHT = 10
    fill = (percentage / 100) * BAR_LENGHT
    border = pygame.Rect(x, y, BAR_LENGHT, BAR_HEIGHT)
    fill = pygame.Rect(x, y, fill, BAR_HEIGHT)
    pygame.draw.rect(surface, GREEN, fill)
    pygame.draw.rect(surface, WHITE, border, 2 )

# Vamos a crear nuestra clase jugador
class Player(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        # Cargamos la imagen
        self.image = pygame.image.load("assets/player.png").convert()
        # Para borrar el borde negro
        self.image.set_colorkey(BLACK)
        # Definimos el rectángulo
        self.rect = self.image.get_rect()
        # Lo colocamos en la ventana
        self.rect.centerx = WIDTH // 2
        self.rect.bottom = HEIGHT - 10
        # Inicializamos una variable para la velocidad
        self.speed_x = 0
        # El valor del escudo
        self.shield = 100

    def update(self):
        self.speed_x = 0
        # Para comprobar si presionamos flecha derecha e izquierda
        keystate = pygame.key.get_pressed()
        if keystate[pygame.K_LEFT]:
            self.speed_x = -5
        if keystate[pygame.K_RIGHT]:
            self.speed_x =  5
        # Para que no se salga de la derecha ni de la izquierda
        self.rect.x += self.speed_x
        if self.rect.right > WIDTH:
            self.rect.right = WIDTH
```

```
        if self.rect.left < 0:
            self.rect.left = 0

    def shoot(self):
        bullet = Bullet(self.rect.centerx, self.rect.top)
        all_sprites.add(bullet)
        bullets.add(bullet)
        laser_sound.play()

class Meteor(pygame.sprite.Sprite):
    def __init__(self): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image = random.choice(meteor_images)
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.x = random.randrange(WIDTH - self.rect.width)
        self.rect.y = random.randrange(-140, -100)
        self.speedy = random.randrange(1, 10)
        self.speedx = random.randrange(-5, 5)

    def update(self):
        self.rect.y += self.speedy
        self.rect.x += self.speedx
        if self.rect.top > HEIGHT + 10 or self.rect.left < -40 or
self.rect.right > WIDTH + 40:
            self.rect.x = random.randrange(WIDTH - self.rect.width)
            self.rect.y = random.randrange(-100, -40)
            self.speedy = random.randrange(1, 10)

class Explosion(pygame.sprite.Sprite):
    def __init__(self, center): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image = explosion_anim[0]
        self.rect = self.image.get_rect()
        self.rect.center = center
        self.frame = 0
        self.last_update = pygame.time.get_ticks()
        self.frame_rate = 50 # Velocidad de la explosión

    def update(self):
        now = pygame.time.get_ticks() # cuanto tiempo ha transcurrido
        if now - self.last_update > self.frame_rate:
            self.last_update = now
            self.frame += 1
            if self.frame == len(explosion_anim):
                self.kill()
            else:
                center = self.rect.center
                self.image = explosion_anim[self.frame]
```

```
                self.rect = self.image.get_rect()
                self.rect.center = center

class Bullet(pygame.sprite.Sprite):
    def __init__(self, x, y): # Inicializamos la clase
        super().__init__() # Definimos la super clase Sprite
        self.image = pygame.image.load("assets/laser1.png").convert()
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.y = y
        self.speedy = random.randrange(1, 10)
        self.rect.centerx = x
        self.speedy = -10

    def update(self):
        self.rect.y += self.speedy
        if self.rect.bottom < 0:
            self.kill()

def show_go_screen():
    screen.blit(background, [0,0])
    draw_text(screen, "SHOOTER", 65, WIDTH // 2, HEIGHT // 4)
    draw_text(screen, "Instrucciones van aquí", 27, WIDTH // 2, HEIGHT //
2)
    draw_text(screen, "Press Key", 20, WIDTH // 2, HEIGHT * 3/4)
    pygame.display.flip()
    waiting = True
    while waiting:
        clock.tick(60)
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
            if event.type == pygame.KEYUP:
                waiting = False



# Cargar todos los meteoritos
meteor_images = []
meteor_list = [ "assets/meteorGrey_big1.png",
                "assets/meteorGrey_big2.png",
                "assets/meteorGrey_big3.png",
                "assets/meteorGrey_big4.png",
                "assets/meteorGrey_med1.png",
                "assets/meteorGrey_med2.png",
                "assets/meteorGrey_small1.png",
                "assets/meteorGrey_small2.png",
                "assets/meteorGrey_tiny1.png",
                "assets/meteorGrey_tiny2.png"
```

```
                ]
for img in meteor_list:
    meteor_images.append(pygame.image.load(img).convert())

# Explosión imagenes
explosion_anim = []
for i in range(9):
    file = "assets/regularExplosion0{}.png".format(i)
    img = pygame.image.load(file).convert()
    img.set_colorkey(BLACK)
    img_scale = pygame.transform.scale(img, (70,70))
    explosion_anim.append(img_scale)

# Cargar imagen de fondo
background = pygame.image.load("assets/background.png").convert()

# Cargar sonido
laser_sound = pygame.mixer.Sound("assets/laser5.ogg")
explosion_sound = pygame.mixer.Sound("assets/explosion.wav")
pygame.mixer.music.load("assets/music.ogg")
pygame.mixer.music.set_volume(0.2)

pygame.mixer.music.play(loops=-1)

# Game Over
game_over = True

# Bucle principal
running = True
while running:
    if game_over:

        show_go_screen()

        game_over = False
        # Creamos el grupo
        all_sprites = pygame.sprite.Group()
        meteor_list = pygame.sprite.Group()
        bullets = pygame.sprite.Group()

        # Creamos una intancia a Player
        player = Player()
        # Lo agregamos a all_sprites
        all_sprites.add(player)

        for i in range(8):
            meteor = Meteor()
            all_sprites.add(meteor)
            meteor_list.add(meteor)
```

```
            # Variable para llevar el marcador
            score = 0

        clock.tick(60)
        # Evento para salir de la ventana
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                running = False
            elif event.type == pygame.KEYDOWN:
                if event.key == pygame.K_SPACE:
                    player.shoot()

        all_sprites.update()

        # Colision - meteo - laser
        hits = pygame.sprite.groupcollide(meteor_list, bullets, True, True)
        for hit in hits:
            score += 10
            explosion_sound.play()
            explosion = Explosion(hit.rect.center)
            all_sprites.add(explosion)
            meteor = Meteor()
            all_sprites.add(meteor)
            meteor_list.add(meteor)

        # Detectar las colisiones jugador - meteoro
        # el parámetro true los objeos que colisiones desaparecen
        hits = pygame.sprite.spritecollide(player, meteor_list, True)
        for hit in hits:
            player.shield -= 25
            meteor = Meteor()
            all_sprites.add(meteor)
            meteor_list.add(meteor)
            if player.shield <= 0:
                game_over = True
        # Insertar la imagen
        screen.blit(background, [0,0])
        # Dibujar en la ventana
        all_sprites.draw(screen)
        # Marcador
        draw_text(screen, str(score), 25, WIDTH // 2, 10 )

        # Escudo
        draw_shield_bar(screen, 5, 5, player.shield)

        pygame.display.flip()

pygame.quit()
```