Guia: Juego de Plataforma Basico con Pygame

Requisitos previos:
- Tener Python instalado (3.7+).
- Instalar pygame: pip install pygame

Estructura del proyecto sugerida:
game/
  main.py
  assets/
    player.png
    tiles.png

Paso 1: Ventana y loop principal
 - Inicializar pygame, crear pantalla y un reloj para gestionar FPS.

Paso 2: Crear el jugador
 - Definir una clase Player con atributos x, y, vel_x, vel_y, on_ground.
 - Dibujar el jugador como rect o imagen.

Paso 3: Movimiento y gravedad
 - Usar teclas izquierda/derecha para mover vel_x.
 - Aplicar gravedad a vel_y cada frame.
 - Permitir salto si on_ground es True.

Paso 4: Plataformas y colisiones
 - Definir plataformas como rects.
 - Detectar colisiones verticales y horizontales para detener el movimiento y colocar al jugador sobre la plataforma.

Paso 5: Nivel simple y objetivo
 - Crear un conjunto de plataformas que formen el nivel.
 - Definir un punto objetivo; si el jugador lo alcanza, mostrar 'Victory'.

Paso 6: Mejoras opcionales
 - Añadir enemigos que patrullen.
 - Animaciones del jugador con sprites.
 - Cargar niveles desde archivos de texto.

Ejemplo minimo (main loop con salto y gravedad):

```
import pygame
```

```
pygame.init()
WIDTH, HEIGHT = 800, 600
screen = pygame.display.set_mode((WIDTH, HEIGHT))
clock = pygame.time.Clock()

x, y = 50, 500
vel_y = 0
gravity = 0.5
jump = -10
on_ground = True
running = True

while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    keys = pygame.key.get_pressed()
    if keys[pygame.K_SPACE] and on_ground:
        vel_y = jump
        on_ground = False
    if keys[pygame.K_LEFT]:
        x -= 5
    if keys[pygame.K_RIGHT]:
        x += 5

    vel_y += gravity
    y += vel_y
    if y >= 500:
        y = 500
        vel_y = 0
        on_ground = True

    screen.fill((135, 206, 235))
    pygame.draw.rect(screen, (255, 0, 0), (x, y, 50, 50))
    pygame.display.flip()
    clock.tick(60)

pygame.quit()
```