```
import pygame
import random
import sys
import time
from pygame import mixer
pygame.init()

WHITE = 235, 255, 255
GREY = 105, 105, 105
BLACK = 0, 0, 0
RED = 255, 0, 0
SCREEN_WIDTH, SCREEN_HEIGHT = 500, 270
WIDTH, HEIGHT = 30, 80
RST_BTN_X, RST_BTN_Y = SCREEN_WIDTH / 2 - 25, SCREEN_HEIGHT / 2 + 40
RST_BTN_W, RST_BTN_H = 50, 50

win = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))
SCORE_FONT = pygame.font.SysFont("monospacebold", 30)


def jump(y, count=10):
    still_jumping = True
    direction = -1 if count < 0 else 1
    if count >= -10:
        y -= count ** 2 * 0.5 * direction
        count -= 1
    else:
        still_jumping = False
        count = 10
        y = SCREEN_HEIGHT + 1 - HEIGHT
    return y, count, still_jumping


def crouch_gen():
    cache = {'count': 12}

    def crouch(height, y):
        still_crouching = True
        direction = -1 if cache['count'] < 0 else 1
        if cache['count'] >= -12:
            height -= 6 * 0.6 * direction
            y += 6 * 0.6 * direction
```

```
            cache['count'] -= 1
        else:
            still_crouching = False
            height = HEIGHT
            y = SCREEN_HEIGHT + 1 - HEIGHT
            cache['count'] = 12
        return height, y, still_crouching
    return crouch


def obstacles(x, y, w, h, color):
    pygame.draw.rect(color, [x, y, w, h])


def message_display(text):
    font = pygame.font.Font('freesansbold.ttf', 48)
    text_surface_obj = font.render(text, True, BLACK, None)
    text_rect_obj = text_surface_obj.get_rect()
    text_rect_obj.center = SCREEN_WIDTH/2, SCREEN_HEIGHT/2 - 25
    win.blit(text_surface_obj, text_rect_obj)
    reset_button = pygame.image.load("reset.png").convert()
    reset_button = pygame.transform.scale(reset_button, (60, 60))
    win.blit(reset_button, (SCREEN_WIDTH/2 - 25, SCREEN_HEIGHT/2 +
25))
    pygame.display.update()


def game_over():
    message_display('Game Over')


def reset_button_clicked():
    mouse = pygame.mouse.get_pos()
    click = pygame.mouse.get_pressed()
    if any(click):
        print(mouse[0], mouse[1])
        print(RST_BTN_X, RST_BTN_X + RST_BTN_W)
        print(RST_BTN_Y, RST_BTN_Y + RST_BTN_H)
    if RST_BTN_X <= mouse[0] <= RST_BTN_X + RST_BTN_W and RST_BTN_Y
<= mouse[1] <= RST_BTN_Y + RST_BTN_H:
        print('clicked')
        return True
```

```
def main():

    pygame.display.set_caption("Attempt Create")
    bkgd = pygame.image.load("graystripe.jpg").convert()
    bkgd_x = 0
    height = HEIGHT
    x, y = 30, SCREEN_HEIGHT - HEIGHT

    mixer.init()
    mixer.music.load('bkgdmusic.mp3')
    mixer.music.play(-1)

    obstacle_start_x = 600
    obstacle_start_y = random.randrange(SCREEN_HEIGHT - HEIGHT + 15,
SCREEN_HEIGHT)
    obstacle_w = 15
    obstacle_h = 15
    obstacle_speed = -7

    score = 0

    jumping, jump_count = False, 10
    crouching = False
    crouch = crouch_gen()
    color = GREY
    run = True
    is_game_over, first_time = False, True
    while run:
        time.sleep(0.03)
        score += 1
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                run = False
            if is_game_over and event.type == pygame.MOUSEBUTTONDOWN:
                is_game_over = not reset_button_clicked()
                if not is_game_over:
                    color = GREY
                    first_time = True
                    obstacle_start_x = SCREEN_WIDTH + 20
                    score = 0

        if is_game_over:
```

```
            if first_time:
                game_over()
                first_time = False

        else:
            rel_bkgd_x = bkgd_x % bkgd.get_rect().width
            win.blit(bkgd, (rel_bkgd_x - bkgd.get_rect().width, 0))
            if rel_bkgd_x < SCREEN_WIDTH:
                win.blit(bkgd, (rel_bkgd_x, 0))
            bkgd_x -= 2

            keys = pygame.key.get_pressed()
            if keys[pygame.K_DOWN] and y >= SCREEN_HEIGHT - HEIGHT
and not jumping:
                crouching = True
                crouch = crouch_gen()
            if keys[pygame.K_SPACE]:
                jumping = True

            obstacles(obstacle_start_x, obstacle_start_y, obstacle_w,
obstacle_h, BLACK)
            obstacle_start_x += obstacle_speed

            if obstacle_start_x < 0:
                obstacle_start_x = SCREEN_WIDTH + 20
                obstacle_start_y = random.randrange(SCREEN_HEIGHT -
HEIGHT + 15, SCREEN_HEIGHT)

            if x <= obstacle_start_x <= x + WIDTH - 5 and y <=
obstacle_start_y <= y + HEIGHT:
                color = RED
                is_game_over = True

            if jumping:
                y, jump_count, jumping = jump(y, jump_count)
            if crouching:
                height, y, crouching = crouch(height, y)
            draw_and_update(height, WIDTH, win, x, y, color, score)

    pygame.quit()
    sys.exit()
```

```
def draw_and_update(height, width, win, x, y, color, score):
    pygame.draw.rect(win, color, (x, y, width, height))
    score_text = SCORE_FONT.render("{0}".format(score), 1, (0, 0, 0))
    score_text_rect = score_text.get_rect()
    score_text_rect.right = SCREEN_WIDTH - 10
    score_text_rect.top = 5
    win.blit(score_text, score_text_rect)
    pygame.display.update()


if __name__ == '__main__':
    main()
```