```
import pygame
import random

pygame.init()

clock = pygame.time.Clock()

sw = 800
sh = 600

screen = pygame.display.set_mode((sw, sh))
pygame.display.set_caption("Hui-Hui")
font_32 = pygame.font.Font('fonts/SF-Pro-Text-Bold.otf', 32)
font_64 = pygame.font.Font('fonts/SF-Pro-Text-Bold.otf', 64)

score = 0


# Time-Out Mode
def time_out_mode(st):
    start_time = st

    game_font = pygame.font.Font('fonts/SF-Pro-Text-Regular.otf', 32)
    xy = [100, 100]

    def print_score(scr):
        screen.blit(game_font.render("Score: " + str(scr), True, (255, 255, 255)), (10, 10))

    def generate_box(x, y):
        return (pygame.Rect(x, y, 100, 100))

    def isClicked(xy, mx, my):
        global score
        if xy[0] < mx < xy[0] + 100 and xy[1] < my < xy[1] + 100:
            score += 1
            print(score)
            return True
        return False

    clicked = False
    start = pygame.time.get_ticks()
    TimeOutRun = True
    while TimeOutRun:
        screen.fill((0, 0, 0))
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                TimeOutRun = False
            if event.type == pygame.MOUSEBUTTONDOWN:
                if event.button == 1:
                    clicked = True
            if event.type == pygame.MOUSEBUTTONUP:
                if event.button == 1:
                    clicked = False

        box = generate_box(xy[0], xy[1])
        pygame.draw.rect(screen, (0, 255, 0), box)
        mx, my = pygame.mouse.get_pos()

        current_time = pygame.time.get_ticks()
        if current_time - start > 1000:
            start = pygame.time.get_ticks()
            xy = [random.randint(0, 750), random.randint(0, 500)]

        if clicked:
            if (current_time - start < 1000) and isClicked(xy, mx, my):
                xy = [random.randint(0, 700), random.randint(0, 500)]
                pygame.draw.rect(screen, (0, 255, 0), box)
                start = pygame.time.get_ticks()

        game_time = pygame.time.get_ticks()

        if game_time - start_time >= 10000:
            screen.fill((200, 0, 0))
            msg = pygame.font.Font('fonts/SF-Pro-Text-Bold.otf', 64)
            screen.blit(msg.render("GAME OVER!!!", True, (255, 255, 255)), (170, 200))

            fsc = pygame.font.Font('fonts/SF-Pro-Text-Bold.otf', 32)
            screen.blit(fsc.render("FINAL SCORE: " + str(score), True, (255, 255, 255)), (265, 275))

        print_score(score)
        clock.tick(60)
        pygame.display.update()

# Arcade Mode
def arcade_mode():
    life = 10
    clicked = False
    game_font = pygame.font.Font('fonts/SF-Pro-Text-Regular.otf', 32)
    xy = [100, 100]


    def print_score(scr):
        screen.blit(game_font.render("Score: " + str(scr), True, (255, 255, 255)), (10, 10))
```

```
def generate_box(x, y):
    return (pygame.Rect(x, y, 100, 100))


def isClicked(xy, mx, my):
    global score
    if xy[0] < mx < xy[0] + 100 and xy[1] < my < xy[1] + 100:
        score += 1
        print(score)
        return True
    return False

def draw_lives(l):
    lives = l
    for i  in range (lives):
        pygame.draw.circle(screen, (255, 0, 0), (760 - 30*i, 20), 15)

start = pygame.time.get_ticks()
ArcadeRun = True
while ArcadeRun:
    screen.fill((0, 0, 0))
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            ArcadeRun = False
        if event.type == pygame.MOUSEBUTTONDOWN:
            if event.button == 1:
                clicked = True
        if event.type == pygame.MOUSEBUTTONUP:
            if event.button == 1:
                clicked = False

    box = generate_box(xy[0], xy[1])
    pygame.draw.rect(screen, (0, 255, 0), box)
    mx, my = pygame.mouse.get_pos()

    current_time = pygame.time.get_ticks()
    if current_time - start > 1000 and not (clicked):
        life -= 1
        print(life)
        start = pygame.time.get_ticks()
        xy = [random.randint(0, 750), random.randint(0, 500)]

    if clicked:
        if (current_time - start < 1000) and isClicked(xy, mx, my):
            xy = [random.randint(0, 700), random.randint(0, 500)]
            pygame.draw.rect(screen, (0, 255, 0), box)
```

```
                start = pygame.time.get_ticks()

            elif (current_time - start < 1000) and not (isClicked(xy, mx, my)):
                clicked = False
                life -= 1
                pygame.draw.rect(screen, (0, 0, 255), box)
                xy = [random.randint(0, 700), random.randint(0, 500)]
                start = pygame.time.get_ticks()
                print(life)


        game_time = pygame.time.get_ticks()

        draw_lives(life)
        if life <= 0:
            screen.fill((200, 0, 0))
            msg = pygame.font.Font('fonts/SF-Pro-Text-Bold.otf', 64)
            screen.blit(msg.render("GAME OVER!!!", True, (255, 255, 255)), (170, 200))

            fsc = pygame.font.Font('fonts/SF-Pro-Text-Bold.otf', 32)
            screen.blit(fsc.render("FINAL SCORE: " + str(score), True, (255, 255, 255)), (265, 275))

        print_score(score)
        clock.tick(60)
        pygame.display.update()


# Main Menu
clicked = False
MainRun = True
mode = 0  # Time-Out or Arcade
state = 0  # Game started or not
while MainRun:
    screen.fill((0, 0, 0))
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            MainRun = False
        if event.type == pygame.MOUSEBUTTONDOWN:
            if event.button == 1:
                clicked = True
        if event.type == pygame.MOUSEBUTTONUP:
            if event.button == 1:
                clicked = False

    Welcome_Message = font_64.render("Hui-Hui", True, (255, 255, 255))
    screen.blit(Welcome_Message, (sw // 2 - 130, 20))
```

```
Select_Mode = font_32.render("Select Mode:", True, (255, 255, 255))
screen.blit(Select_Mode, (10, sh - 400))

time_out = font_32.render("Time-Out", True, (255, 255, 255))
screen.blit(time_out, (290, sh - 360))

arcade = font_32.render("Arcade", True, (255, 255, 255))
screen.blit(arcade, (290, sh - 320))

Start = font_64.render("Start", True, (255, 255, 255))
screen.blit(Start, (320, sh - 100))

mx, my = pygame.mouse.get_pos()

if clicked and state == 0:
    if 285 < mx < 420 and sh - 360 < my < sh - 323:
        mode = 1  # Time-Out
    elif 285 < mx < 420 and sh - 320 < my < sh - 283:
        mode = 2  # Arcade
    elif sw // 2 - 95 < mx < sw // 2 + 95 and sh - 105 < my < sh - 30:
        state = 1

if mode == 1:
    pygame.draw.rect(screen, (255, 0, 0), pygame.Rect(275, sh - 360, 190, 37), 2)
elif mode == 2:
    pygame.draw.rect(screen, (255, 0, 0), pygame.Rect(275, sh - 320, 190, 37), 2)

if state == 1:
    pygame.draw.rect(screen, (0, 255, 0), pygame.Rect(sw // 2 - 90, sh - 97, 180, 70), 2)
    if mode == 1:  # Time-Out
        MainRun = False
        time_out_mode(pygame.time.get_ticks())

    elif mode == 2:  # Arcade
        MainRun = False
        arcade_mode()

pygame.display.update()
```