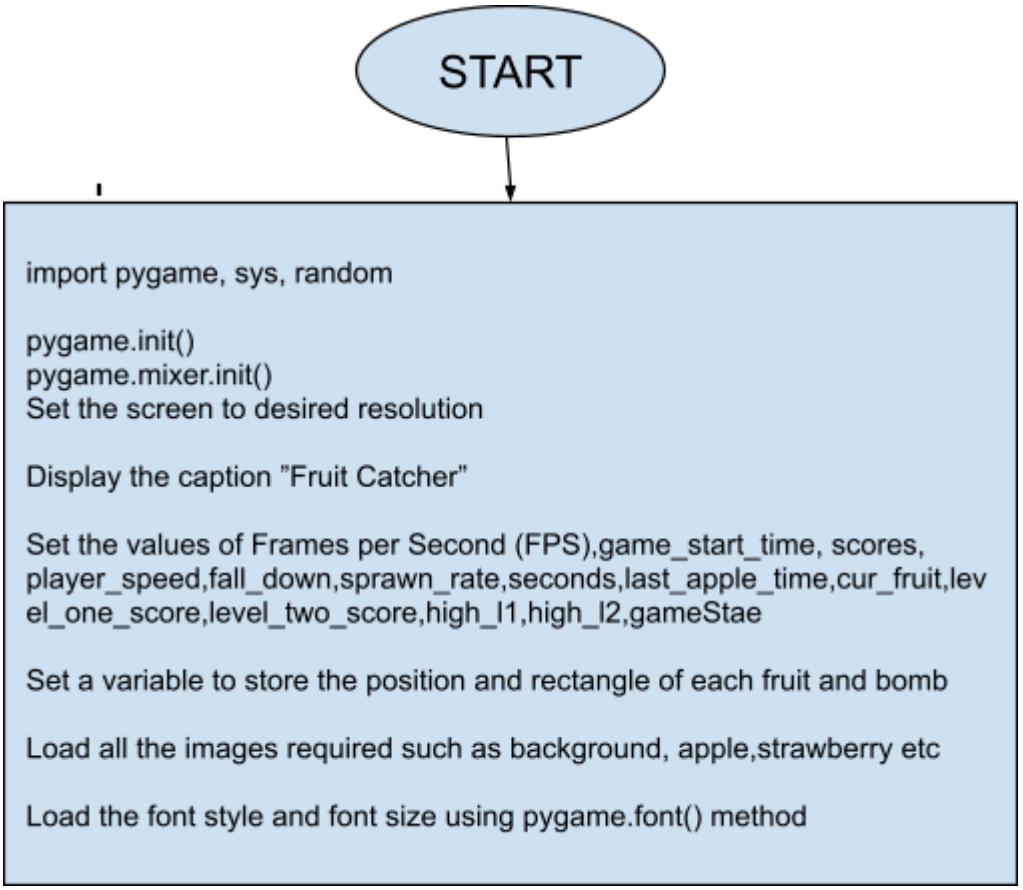
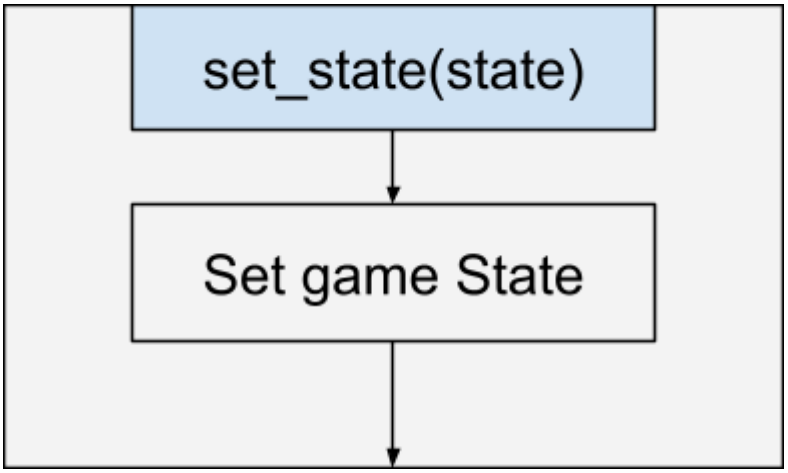
Project – Fruit Catcher Game

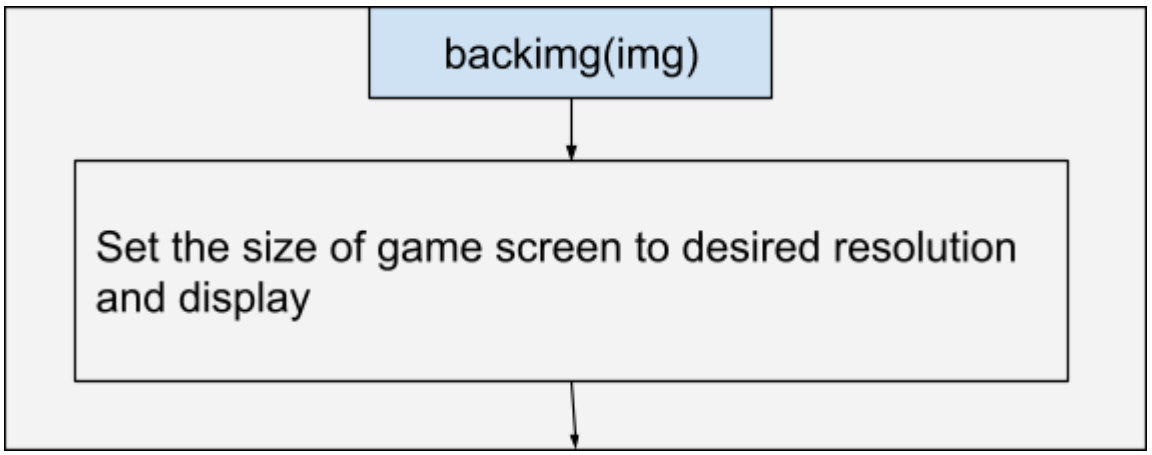
Problem Statement

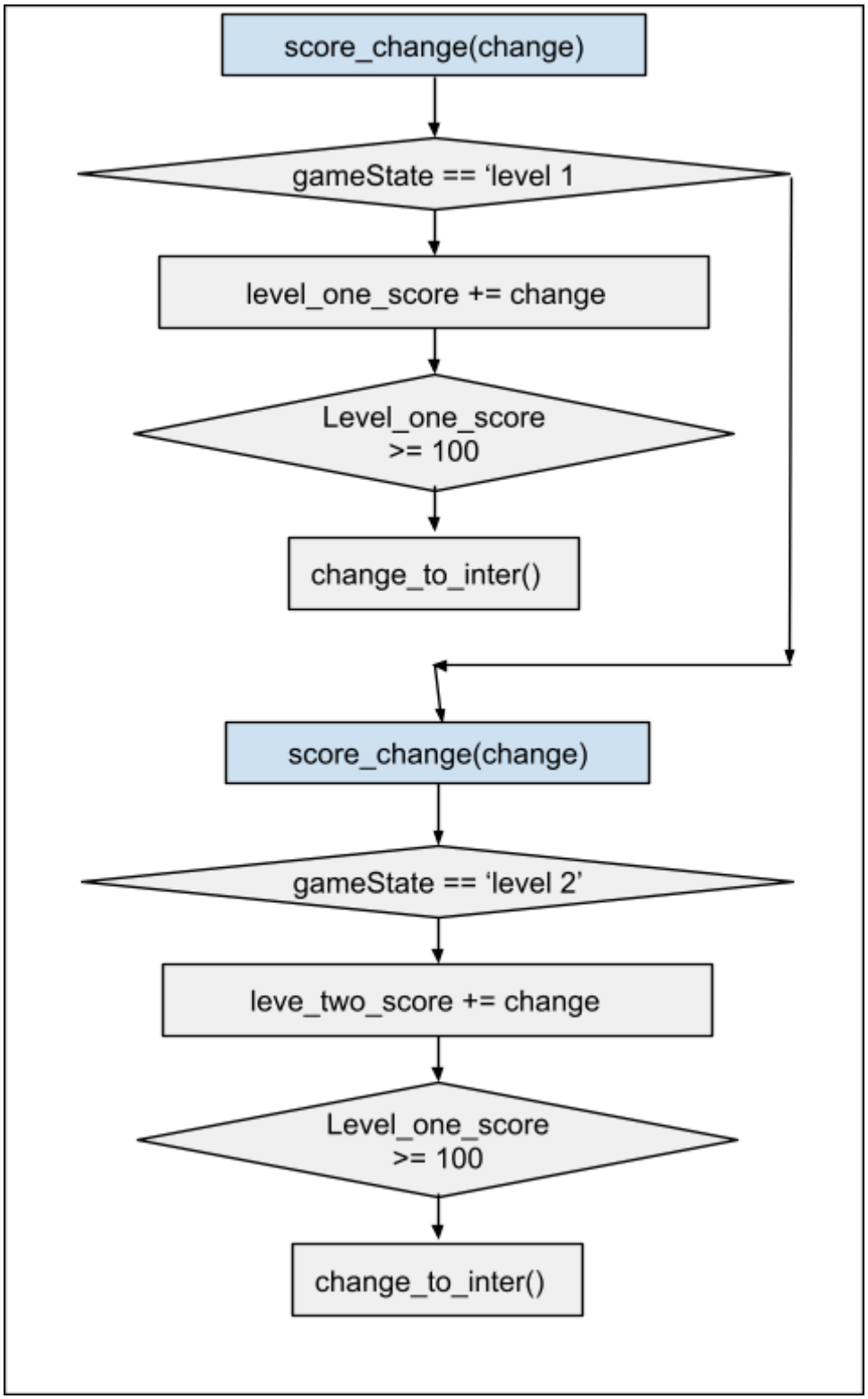
Catch falling fruits while avoiding bombs using Pygame(). Score points by catching fruits, but lose points for bombs. The game has multiple levels with increasing difficulty. Follow the specifications to design and develop the game.

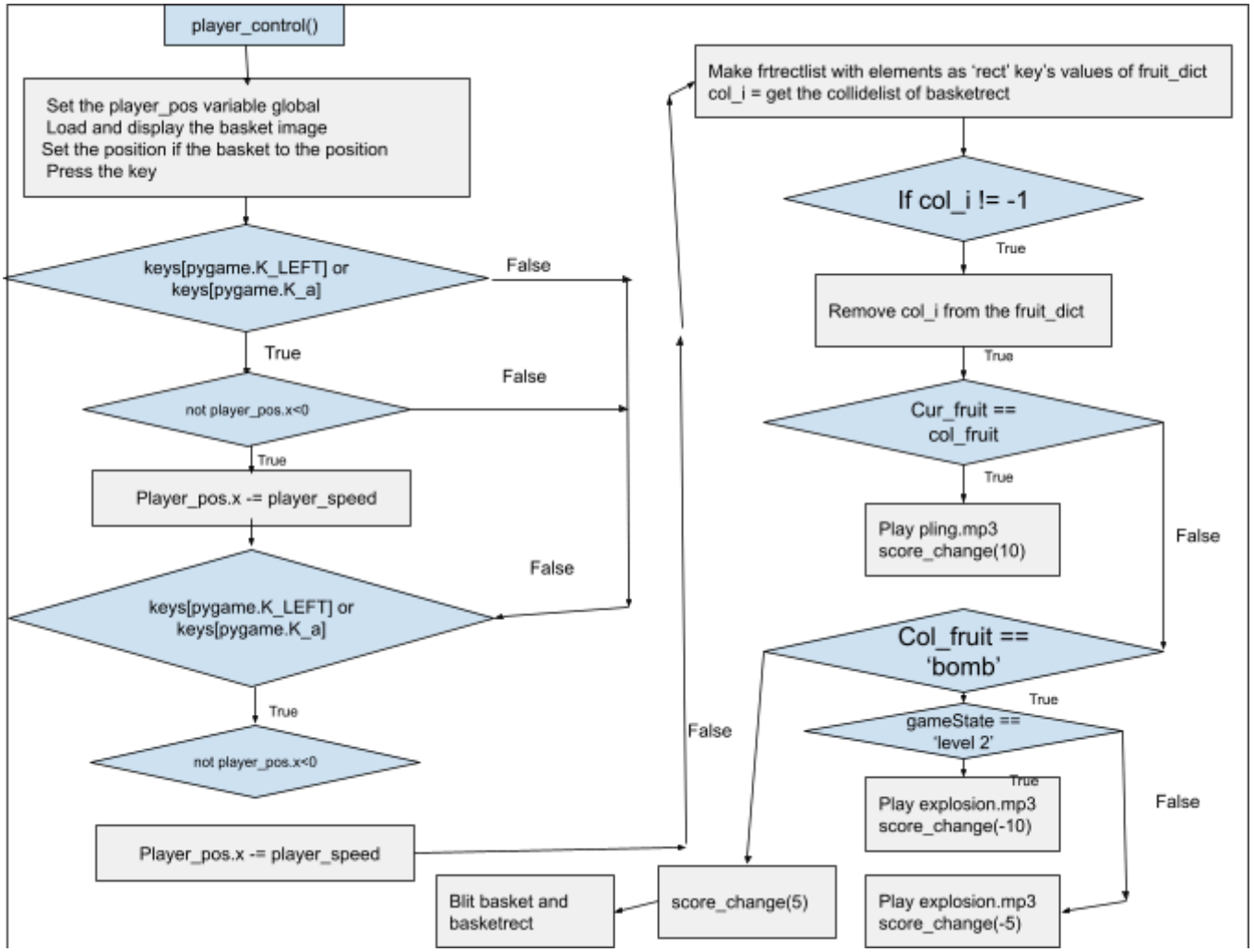
Flowchart

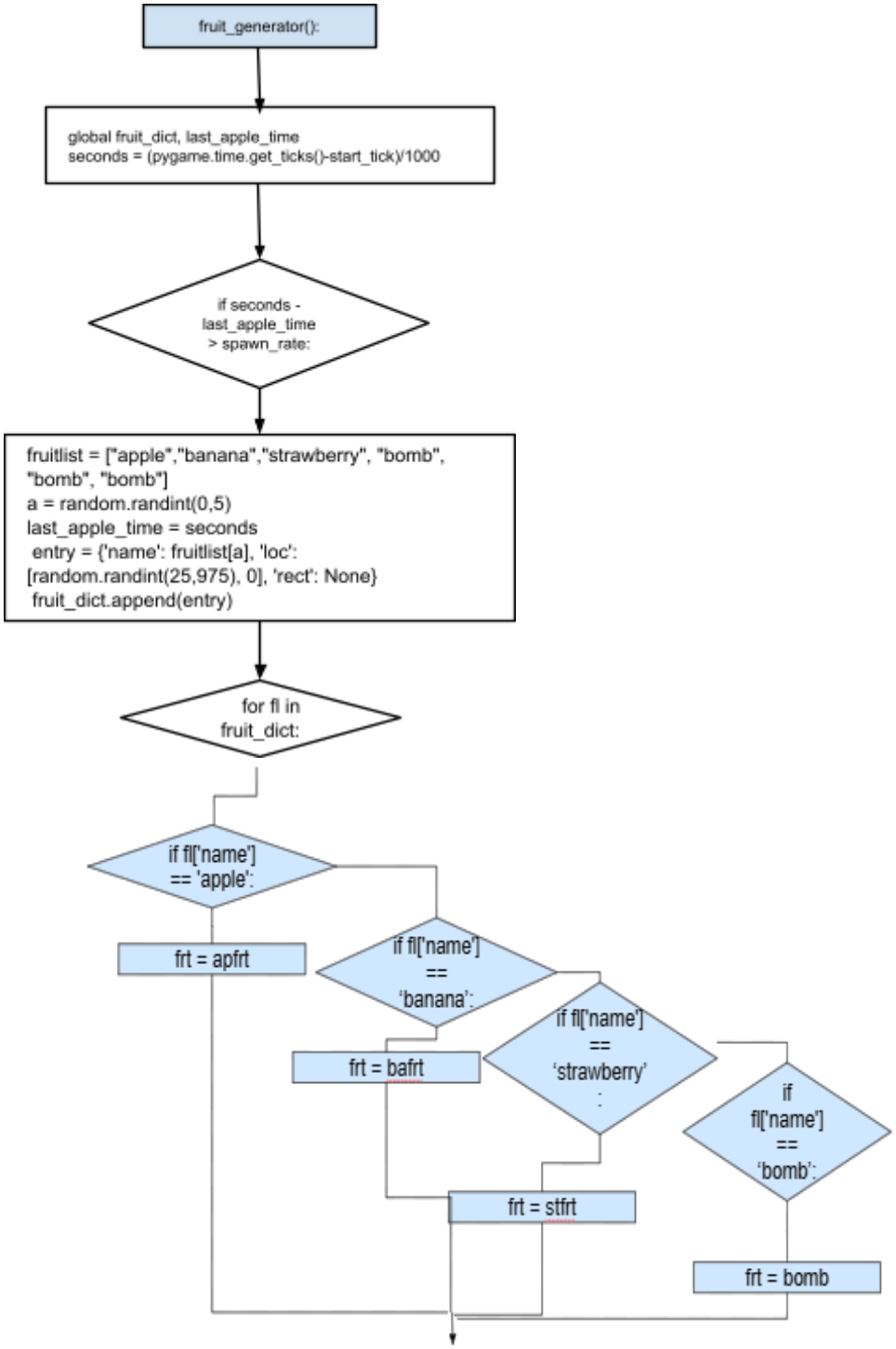


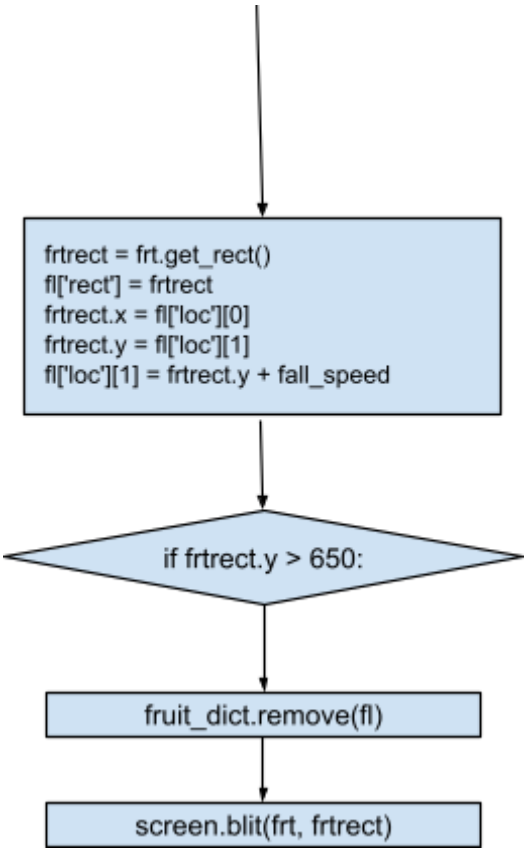

set_state(state)
Set game State

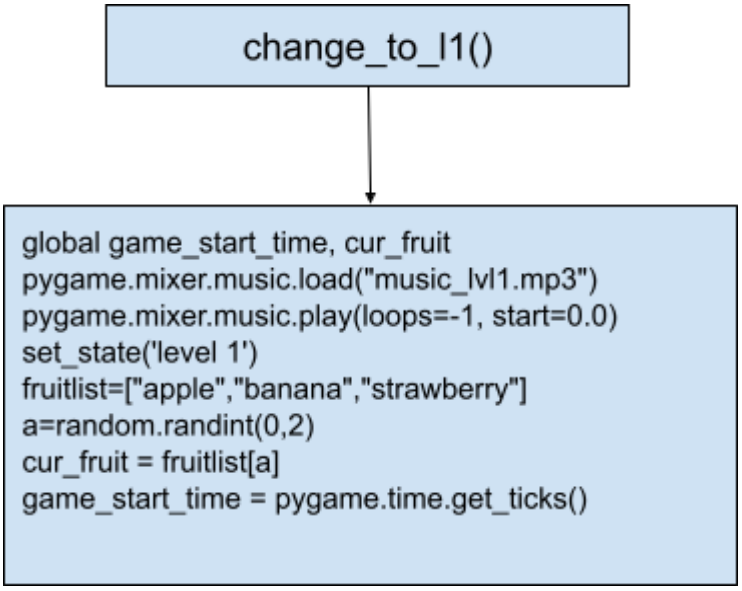

backimg(img)
Set the size of game screen to desired resolution and display

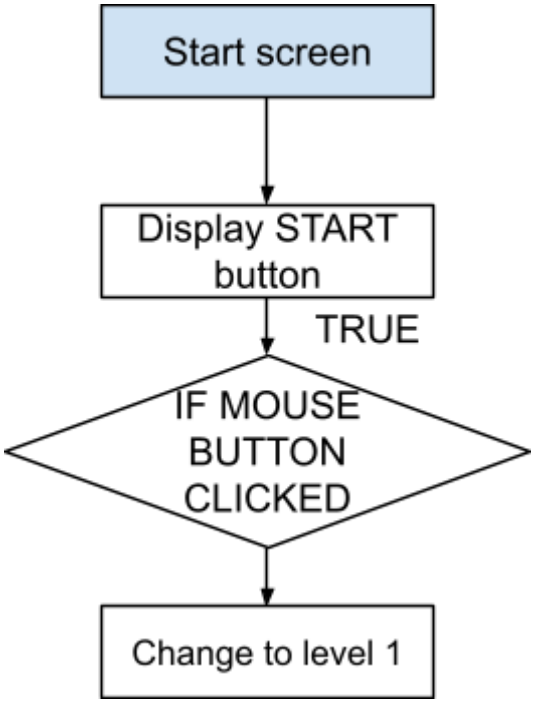

score_change(change)
gameState == 'level 1
level_one_score += change
Level_one_score
>= 100
change_to_inter()
score_change(change)
gameState == 'level 2'
leve_two_score += change
Level_one_score
>= 100
change_to_inter()

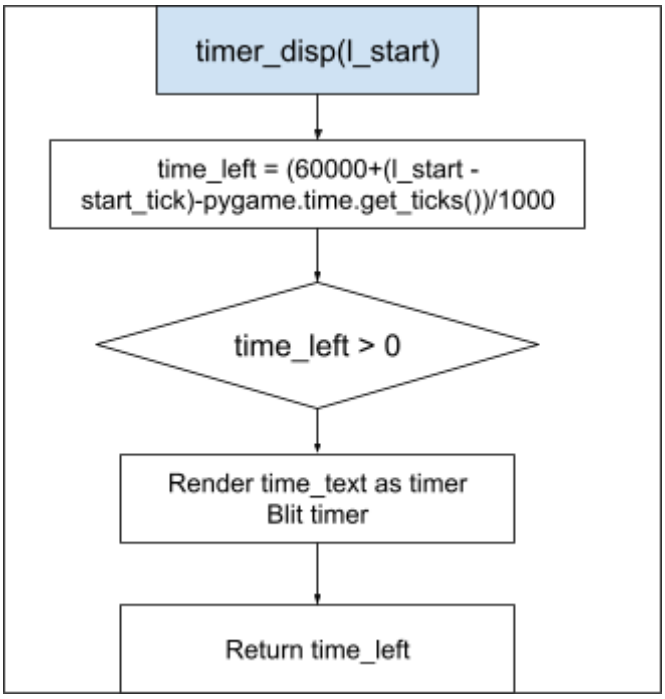

player_control()
Set the player_pos variable global
Load and display the basket image
Set the position if the basket to the position
Press the key
keys[pygame.K_LEFT] or keys[pygame.K_a]
False
True
not player_pos.x<0
False
True
Player_pos.x -= player_speed
False
keys[pygame.K_LEFT] or keys[pygame.K_a]
True
not player_pos.x<0
Player_pos.x -= player_speed
Blit basket and basketrect
score_change(5)
Make frtrectlist with elements as 'rect' key's values of fruit_dict
col_i = get the collidelist of basketrect
If col_i != -1
True
Remove col_i from the fruit_dict
True
Cur_fruit == col_fruit
True
Play pling.mp3
score_change(10)
False
Col_fruit == 'bomb'
True
gameState == 'level 2'
True
Play explosion.mp3
score_change(-10)
False
Play explosion.mp3
score_change(-5)
False

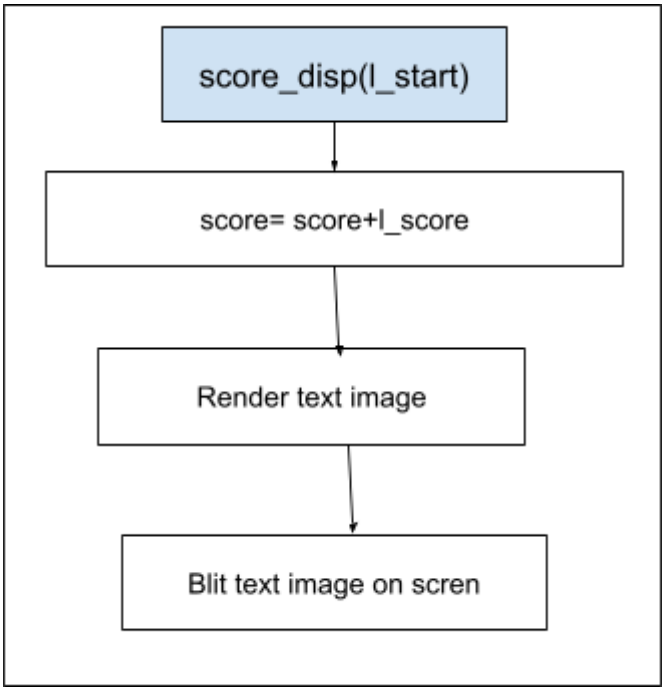

fruit_generator():
global fruit_dict, last_apple_time
seconds = (pygame.time.get_ticks()-start_tick)/1000
if seconds - last_apple_time > spawn_rate:
fruitlist = ["apple","banana","strawberry", "bomb", "bomb", "bomb"]
a = random.randint(0,5)
last_apple_time = seconds
entry = {'name': fruitlist[a], 'loc': [random.randint(25,975), 0], 'rect': None}
fruit_dict.append(entry)
for fl in fruit_dict:
if fl['name'] == 'apple':
frt = apfrt
if fl['name'] == 'banana':
frt = bafrt
if fl['name'] == 'strawberry' :
frt = stfrt
if fl['name'] == 'bomb':
frt = bomb

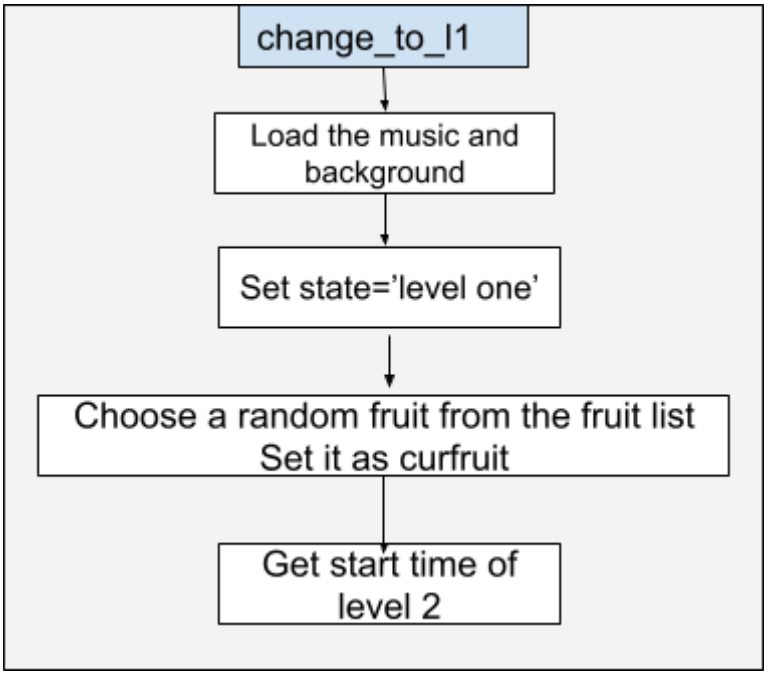

frtrect = frt.get_rect()
fl['rect'] = frtrect
frtrect.x = fl['loc'][0]
frtrect.y = fl['loc'][1]
fl['loc'][1] = frtrect.y + fall_speed
if frtrect.y > 650:
fruit_dict.remove(fl)
screen.blit(frt, frtrect)

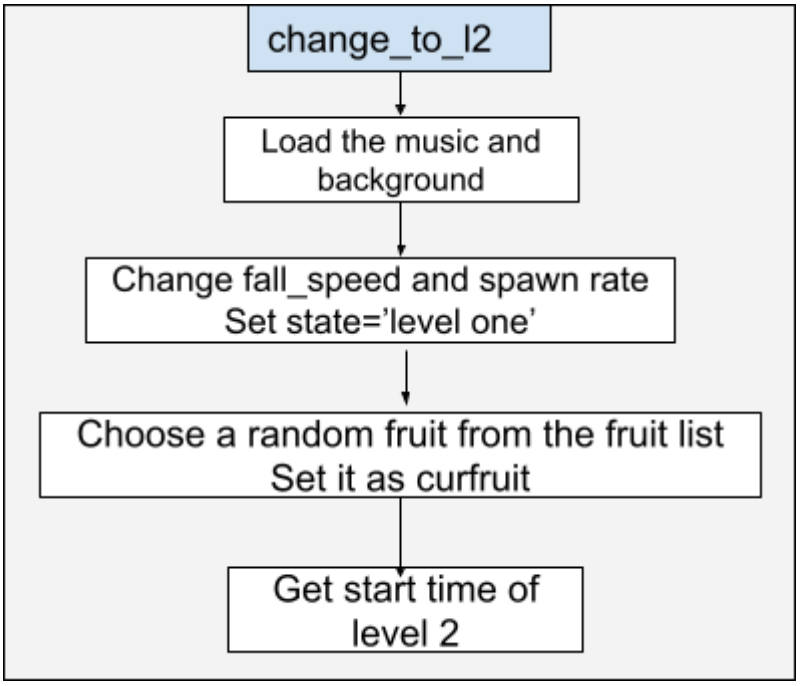

change_to_l1()
global game_start_time, cur_fruit
pygame.mixer.music.load("music_lvl1.mp3")
pygame.mixer.music.play(loops=-1, start=0.0)
set_state('level 1')
fruitlist=["apple","banana","strawberry"]
a=random.randint(0,2)
cur_fruit = fruitlist[a]
game_start_time = pygame.time.get_ticks()

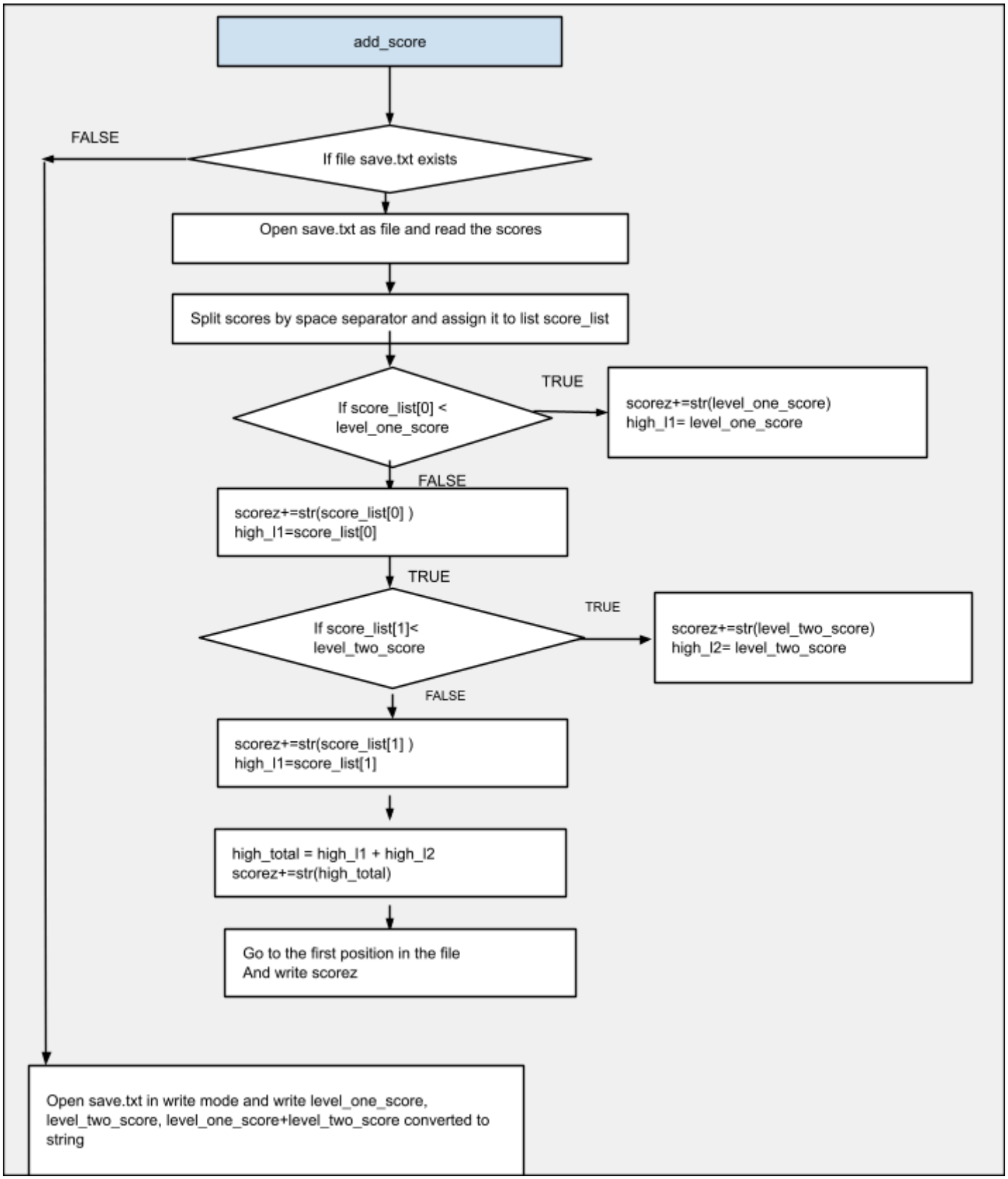

Start screen
Display START button
TRUE
IF MOUSE BUTTON CLICKED
Change to level 1

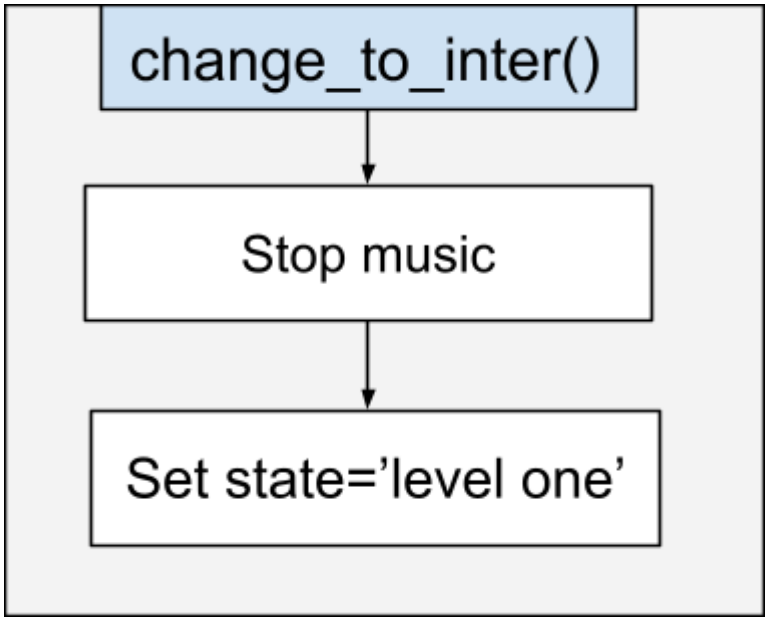

timer_disp(l_start)
time_left = (60000+(l_start - start_tick)-pygame.time.get_ticks())/1000
time_left > 0
Render time_text as timer
Blit timer
Return time_left

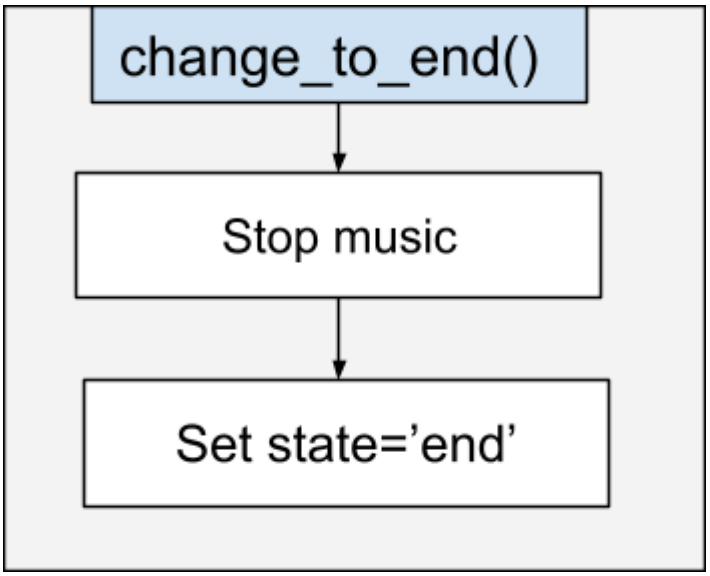

score_disp(l_start)
score= score+l_score
Render text image
Blit text image on scren

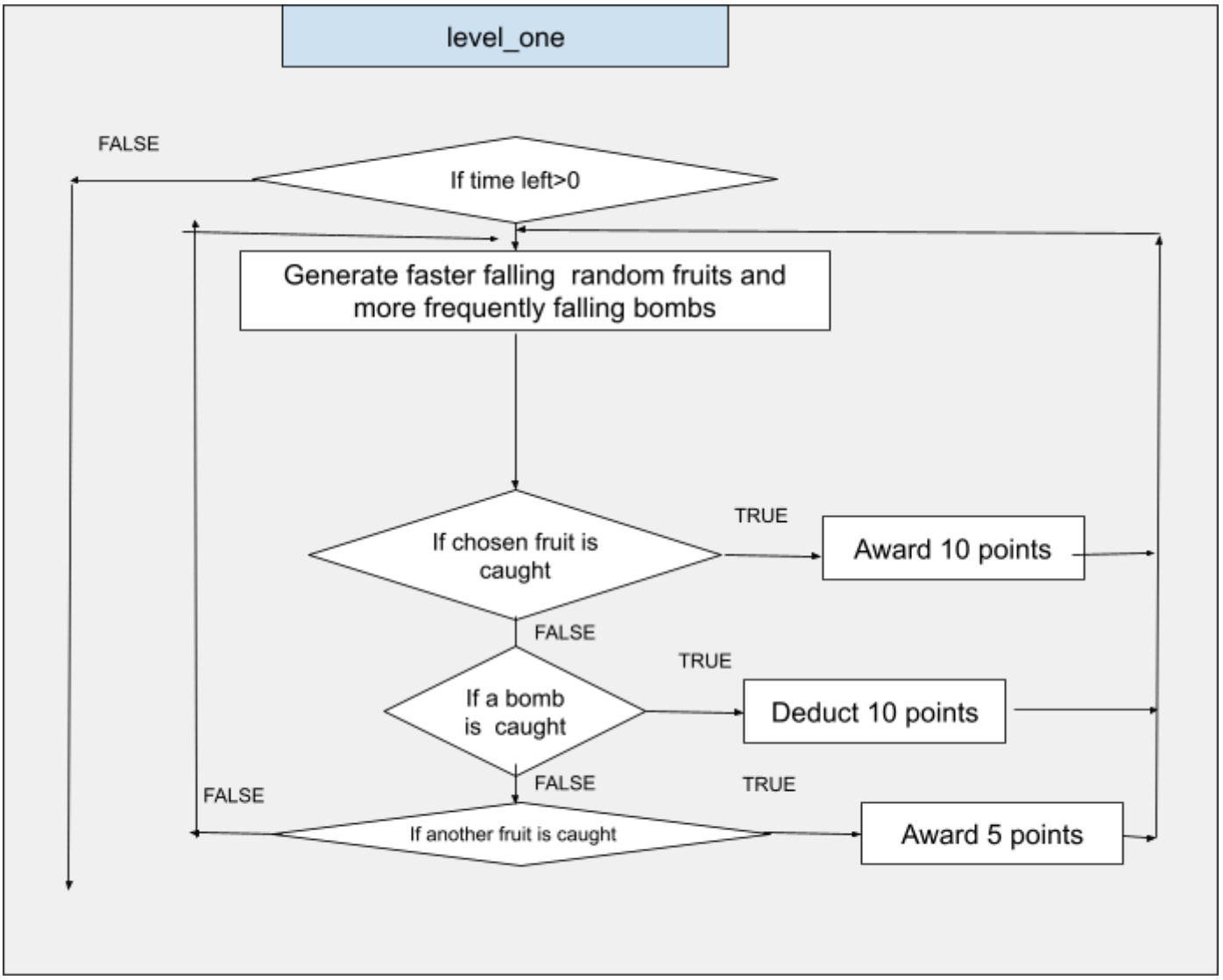

change_to_l1
Load the music and background
Set state='level one'
Choose a random fruit from the fruit list
Set it as curfruit
Get start time of level 2

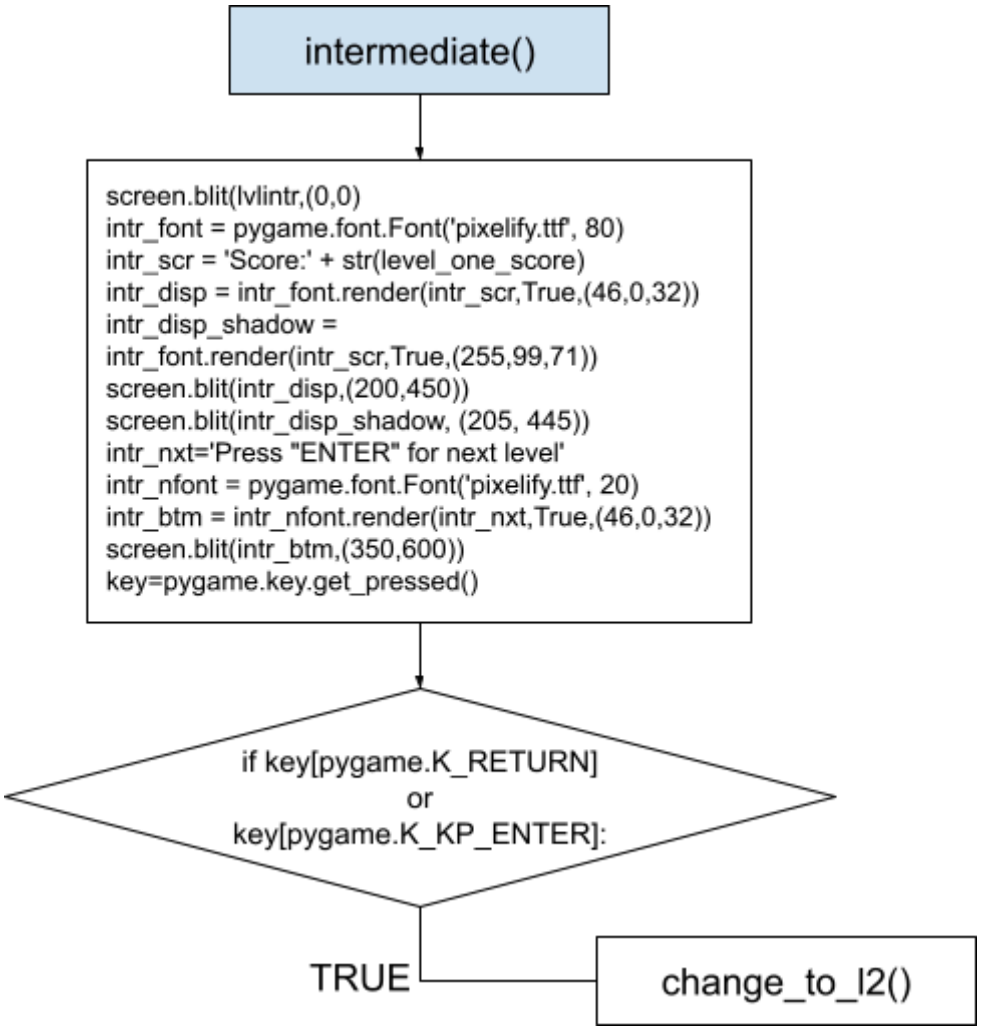

change_to_l2
Load the music and background
Change fall_speed and spawn rate
Set state='level one'
Choose a random fruit from the fruit list
Set it as curfruit
Get start time of level 2

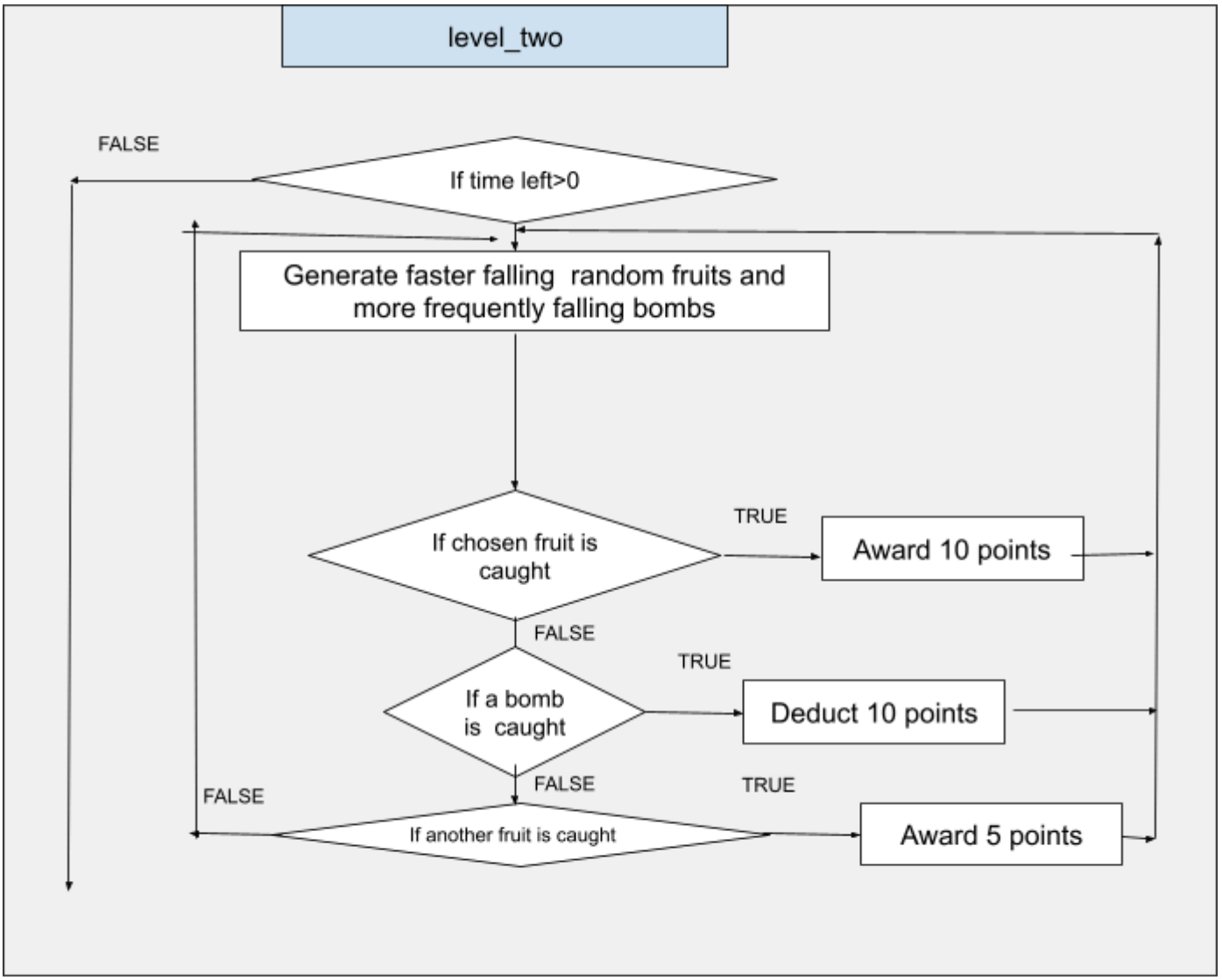

add_score
If file save.txt exists
FALSE
Open save.txt as file and read the scores
Split scores by space separator and assign it to list score_list
If score_list[0] < level_one_score
TRUE
scorez+=str(level_one_score)
high_l1= level_one_score
FALSE
scorez+=str(score_list[0])
high_l1=score_list[0]
TRUE
If score_list[1]< level_two_score
TRUE
scorez+=str(level_two_score)
high_l2= level_two_score
FALSE
scorez+=str(score_list[1])
high_l1=score_list[1]
high_total = high_l1 + high_l2
scorez+=str(high_total)
Go to the first position in the file
And write scorez
Open save.txt in write mode and write level_one_score, level_two_score, level_one_score+level_two_score converted to string

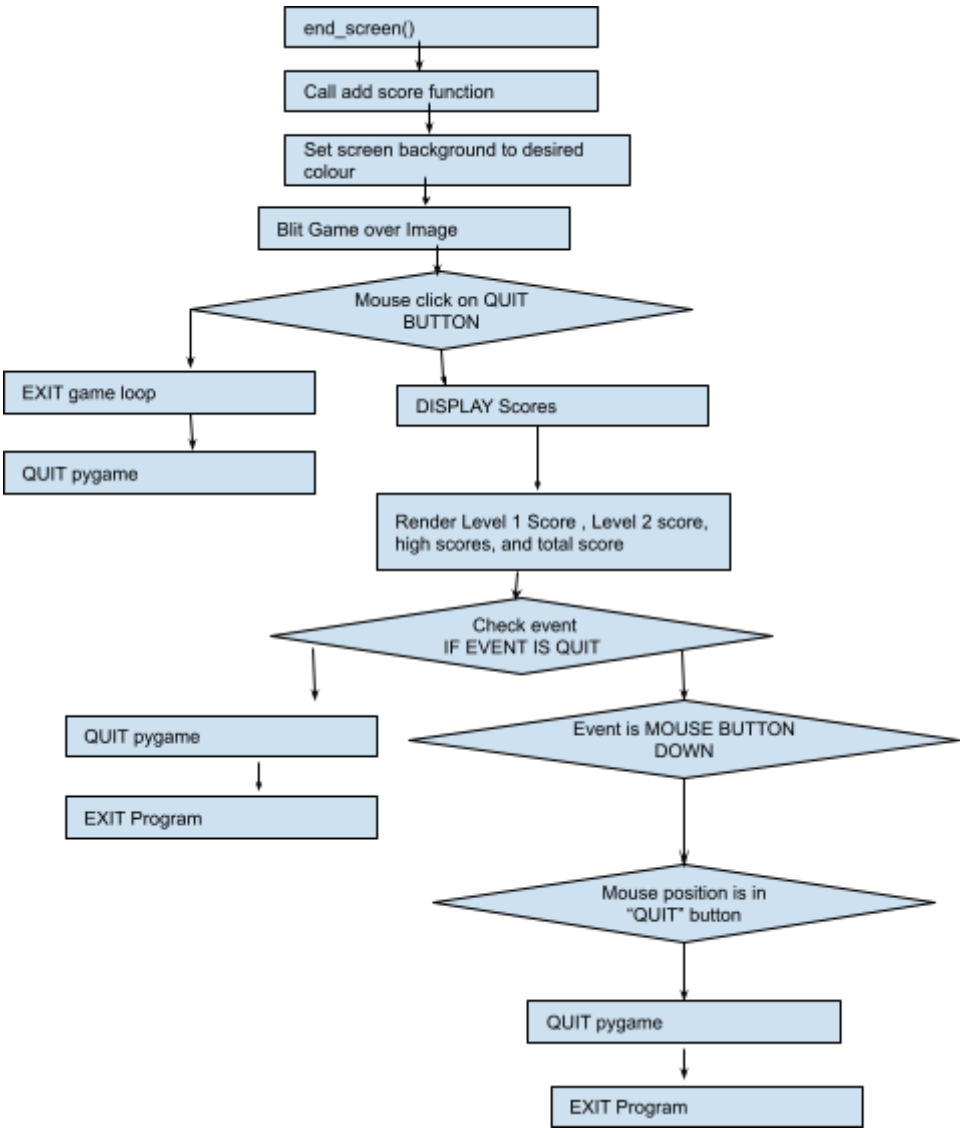

change_to_inter()
Stop music
Set state='level one'

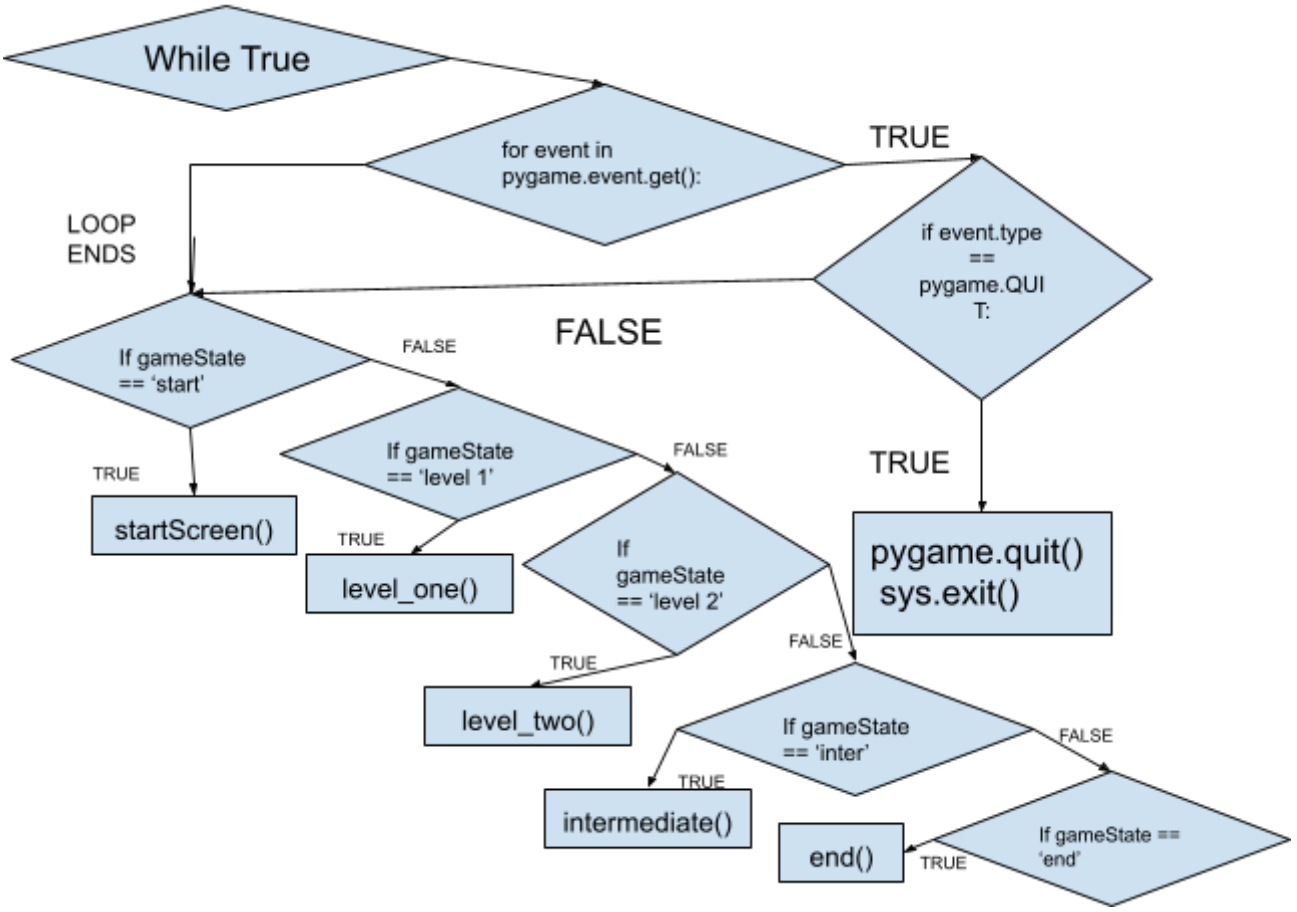

change_to_end()
Stop music
Set state='end'


level_one
FALSE
If time left>0
Generate faster falling random fruits and more frequently falling bombs
If chosen fruit is caught
TRUE
Award 10 points
FALSE
If a bomb is caught
TRUE
Deduct 10 points
FALSE
FALSE
TRUE
If another fruit is caught
Award 5 points


intermediate()
screen.blit(lvlintr,(0,0)
intr_font = pygame.font.Font('pixelify.ttf', 80)
intr_scr = 'Score:' + str(level_one_score)
intr_disp = intr_font.render(intr_scr,True,(46,0,32))
intr_disp_shadow =
intr_font.render(intr_scr,True,(255,99,71))
screen.blit(intr_disp,(200,450))
screen.blit(intr_disp_shadow, (205, 445))
intr_nxt='Press "ENTER" for next level'
intr_nfont = pygame.font.Font('pixelify.ttf', 20)
intr_btm = intr_nfont.render(intr_nxt,True,(46,0,32))
screen.blit(intr_btm,(350,600))
key=pygame.key.get_pressed()
if key[pygame.K_RETURN]
or
key[pygame.K_KP_ENTER]:
TRUE
change_to_l2()


level_two
FALSE
If time left>0
Generate faster falling random fruits and more frequently falling bombs
TRUE
If chosen fruit is caught
Award 10 points
FALSE
TRUE
If a bomb is caught
Deduct 10 points
FALSE
TRUE
FALSE
If another fruit is caught
Award 5 points


end_screen()
Call add score function
Set screen background to desired colour
Blit Game over Image
Mouse click on QUIT BUTTON
EXIT game loop
QUIT pygame
DISPLAY Scores
Render Level 1 Score , Level 2 score, high scores, and total score
Check event IF EVENT IS QUIT
QUIT pygame
EXIT Program
Event is MOUSE BUTTON DOWN
Mouse position is in "QUIT" button
QUIT pygame
EXIT Program


While True
for event in pygame.event.get():
TRUE
LOOP ENDS
if event.type == pygame.QUIT:
FALSE
If gameState == 'start'
FALSE
TRUE
startScreen()
If gameState == 'level 1'
FALSE
TRUE
level_one()
If gameState == 'level 2'
TRUE
level_two()
TRUE
pygame.quit() sys.exit()
FALSE
If gameState == 'inter'
FALSE
TRUE
intermediate()
If gameState == 'end'
TRUE
end()

Algorithm

```
1. import modules pygame,sys,random
2. initialize global variables
     a. FPS=60
     b. game_start_time=0
     c. lvl2_start_time=0
     d. scorez=''
     e. player_speed = 10
     f. fall_speed = 2
     g. spawn_rate = 2
     h. seconds = 0
     i. last_apple_time = 0
     j. bckgrnd = pygame.image.load('start-title.png')
     k. cur_fruit = ""
     l. l1b = pygame.image.load('l1_back.png')
     m. l2b = pygame.image.load('l2_back.png')
     n. lvlintr = pygame.image.load('lvl1.png')
     o. level_one_score = 0
     p. level_two_score = 0
     q. high_l1 = 0
     r. high_l2 = 0
     s. apfrt = pygame.image.load("apples.png")
     t. bafrt = pygame.image.load("banana.png")
     u. stfrt = pygame.image.load("strawberry.png")
     v. bomb=pygame.image.load("bomb.png")
     w. hud_font = pygame.font.Font('pixelify.ttf', 40)
3. set the standard template
     a. pygame.init()
     b. pygame.mixer.init()
     c. SCREEN_WIDTH=1000
     d. SCREEN_HEIGHT=650
     e. screen = pygame.display.set_mode((1000, 650))
     f. pygame.display.set_caption("Fruit Catcher!")
     g. clock = pygame.time.Clock()
     h. start_tick = pygame.time.get_ticks()
4. set default state by defining gameState as start
5. set the basket’s starting position
6. establish locations of the fruit
7. create main game loop
8. make a way to quit the game without finishing it using events
9. check gameState next
10.   if gameState is ‘start’
     a. call startscreen()
```

b. startscreen sets the background of start screen
c. creates a start button in a way where if mouse is hovered over it the colour changes
d. it makes use of button1() and mouse1()
e. if we click on start change_to_l1() is called
   i. change_to_l1 starts the timer, starts the music and creates the fruit list
   ii. it then sets gameState as level 1

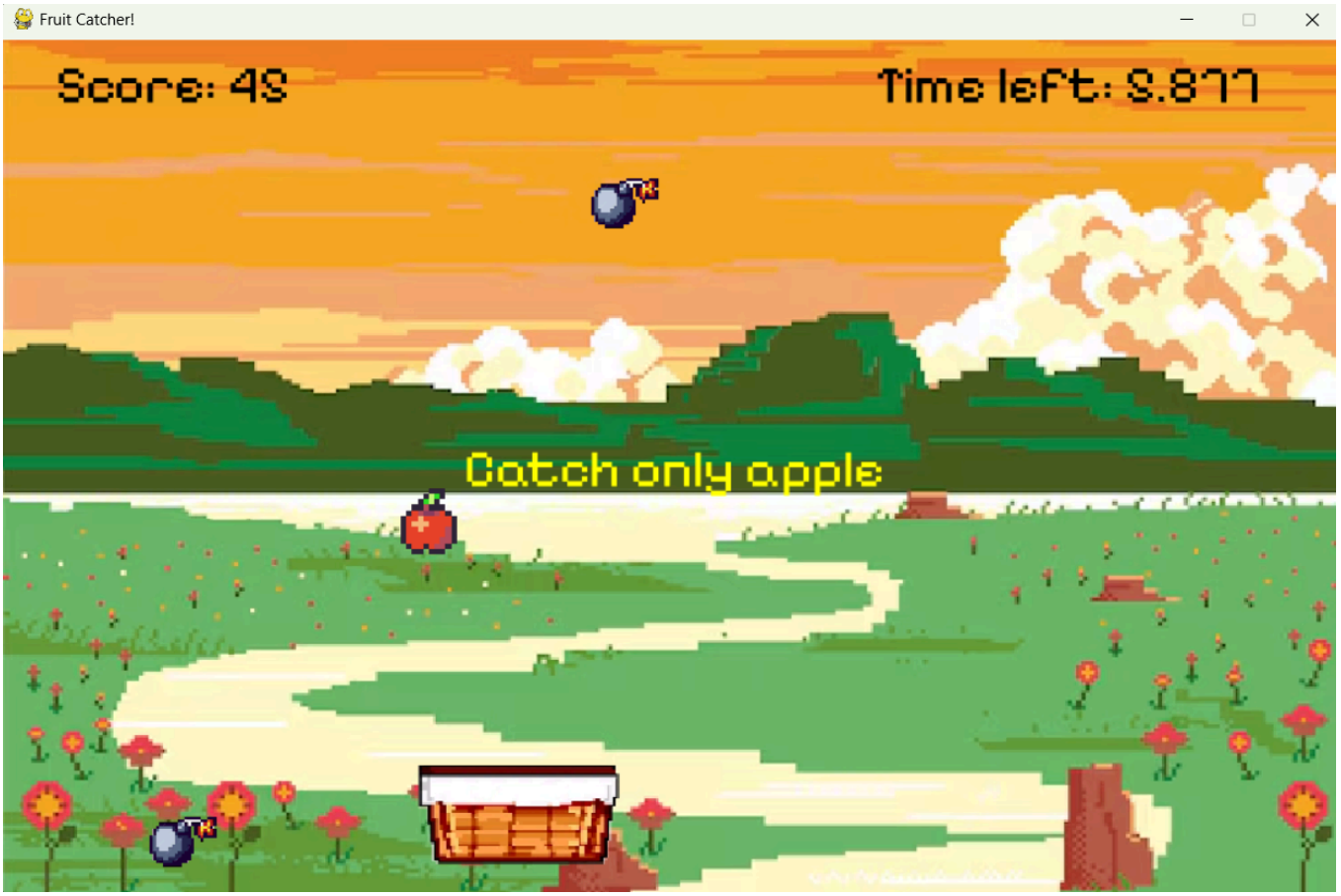
11. as gameState is level 1 call level_one()
    a. set the background
    b. display timer and score
    c. display ‘catch only’ text
    d. if timer runs out, call change_to_inter, music plays
    e. calls player_control()
       i. loads basket image
       ii. makes the image move by 10 pixels on the direction the user wants
       iii. doesn’t let basket leave the boundaries
       iv. also establishes collide system that increases or decreases points according to the rules
    f. calls fruit_generator()
       i. starts a timer of its own
       ii. spawns in images of fruits at every spawn time
       iii. makes the image drop
       iv. removes it if it exceeds the height
12. change_to_inter()
    a. music stops
    b. sets state as inter
13. gameState is inter, intermediate() is called
    a. display intermediate background
    b. displays score of level 1
    c. ‘Press ENTER to for next level’ is printed at the bottom
    d. if enter or return is pressed call change_to_l2()
14. change_to_l2()
    a. starts timer
    b. increases spawn rate
    c. increases fall speed
    d. starts level 2 music
    e. starts timer
    f. sets state as level 2
15. as gameState is level 2, level_two() is called
    a. level 2 background is set
    b. catch fruit is specified
    c. if timer runs out change_to_end() is called
16. change_to_end()
    a. music stops
    b. state is set to end

17. as state is set as end, endscreen() is called
    a. displays end screen background
    b. creates a quit button that exits the game if pressed
    c. calls addscore()
        i. creates a file if no data has been saved prior and saves your current score
        ii. if the file has data it compares each of the scores(level one, level two, total) with the saved values
        iii. it updates to the higher score
    d. finally your current scores is displayed along with the highest scores attained in each level and totally

Test Cases

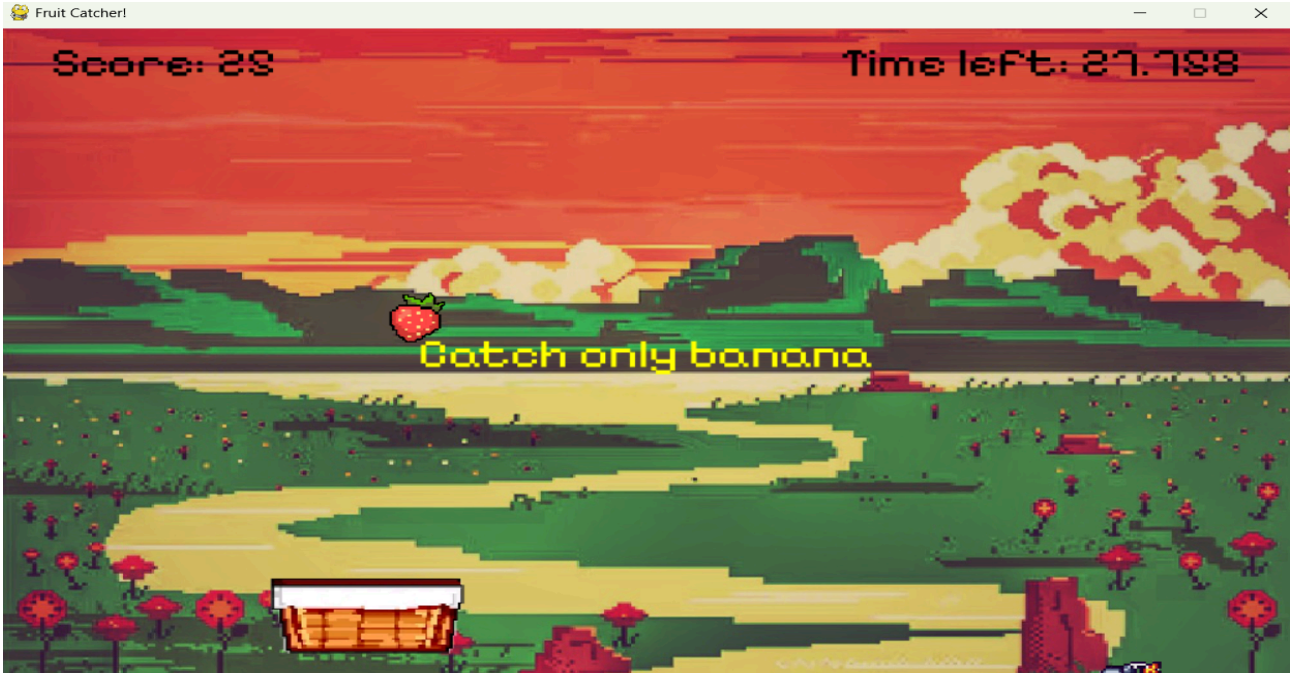
Start screen

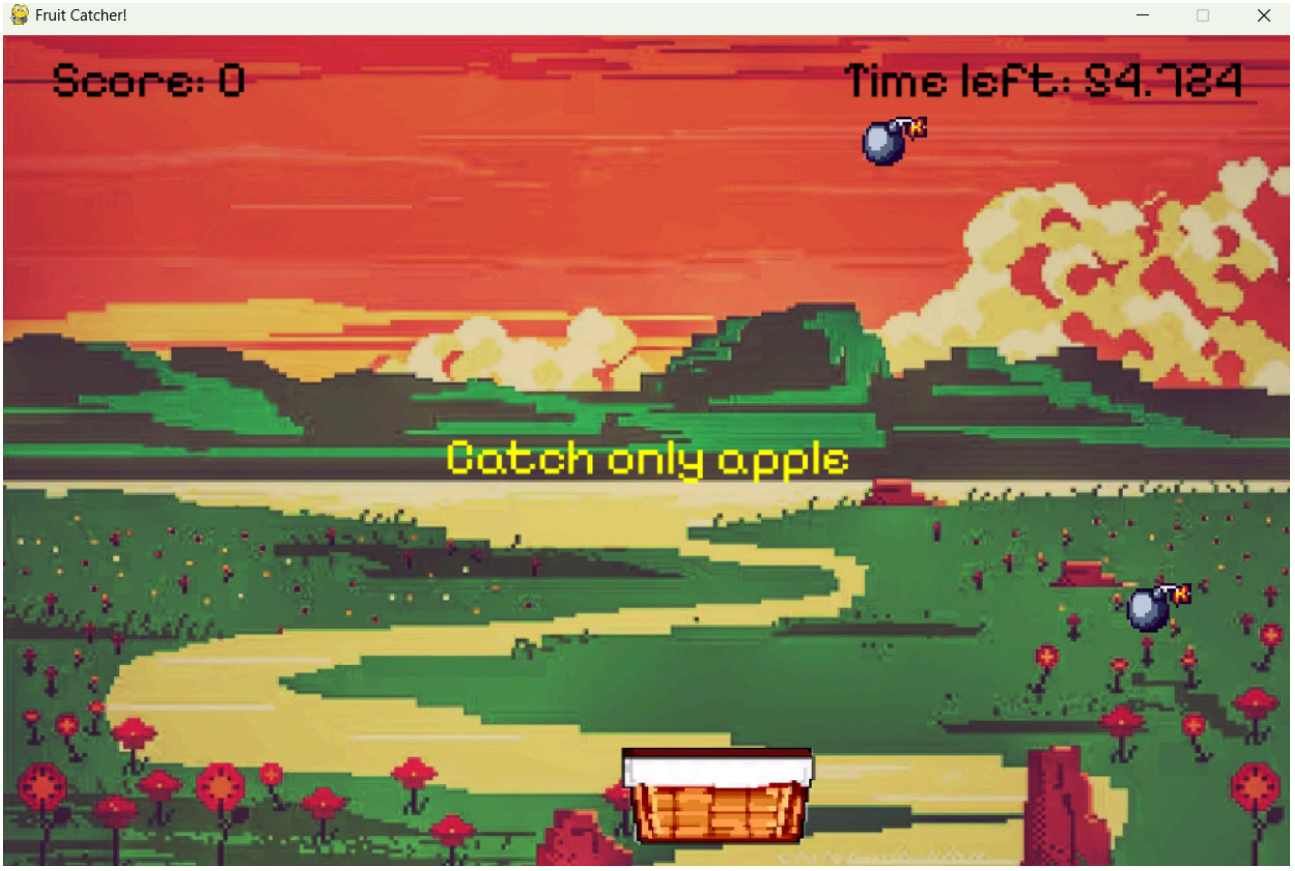
Level one screen



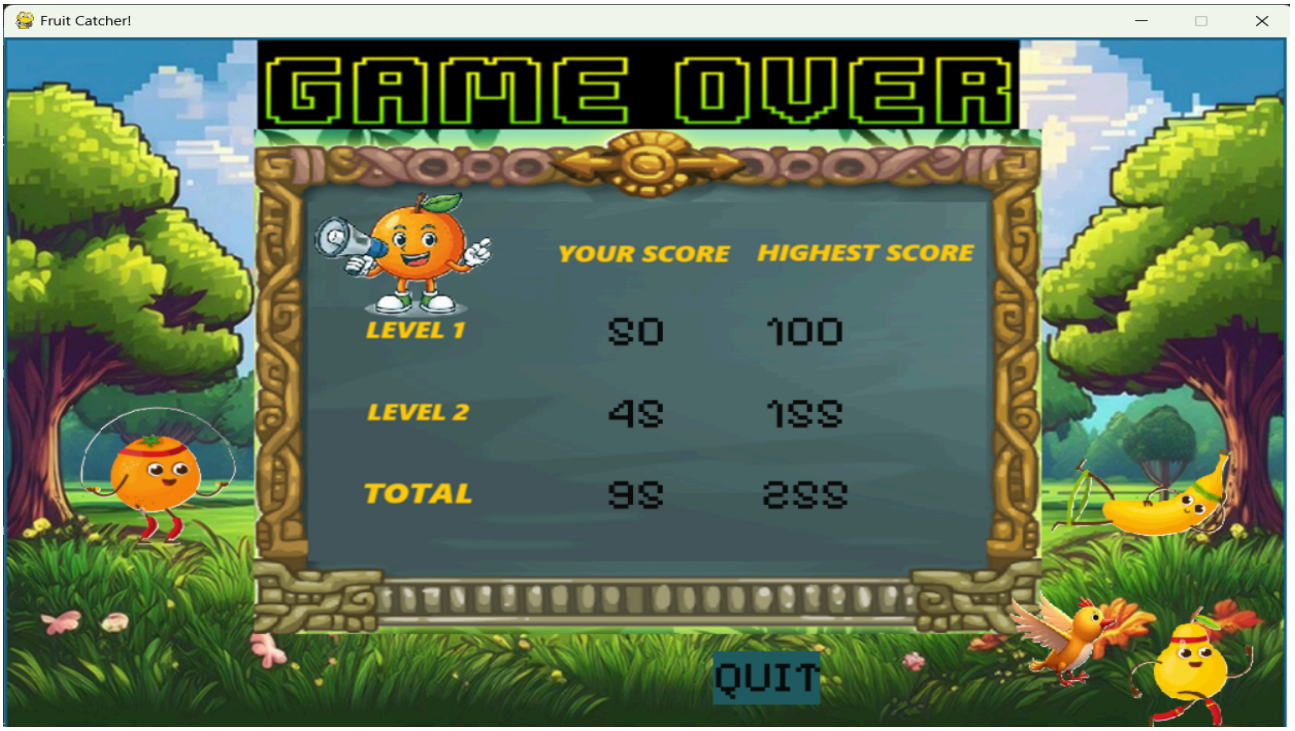
Level change Screen

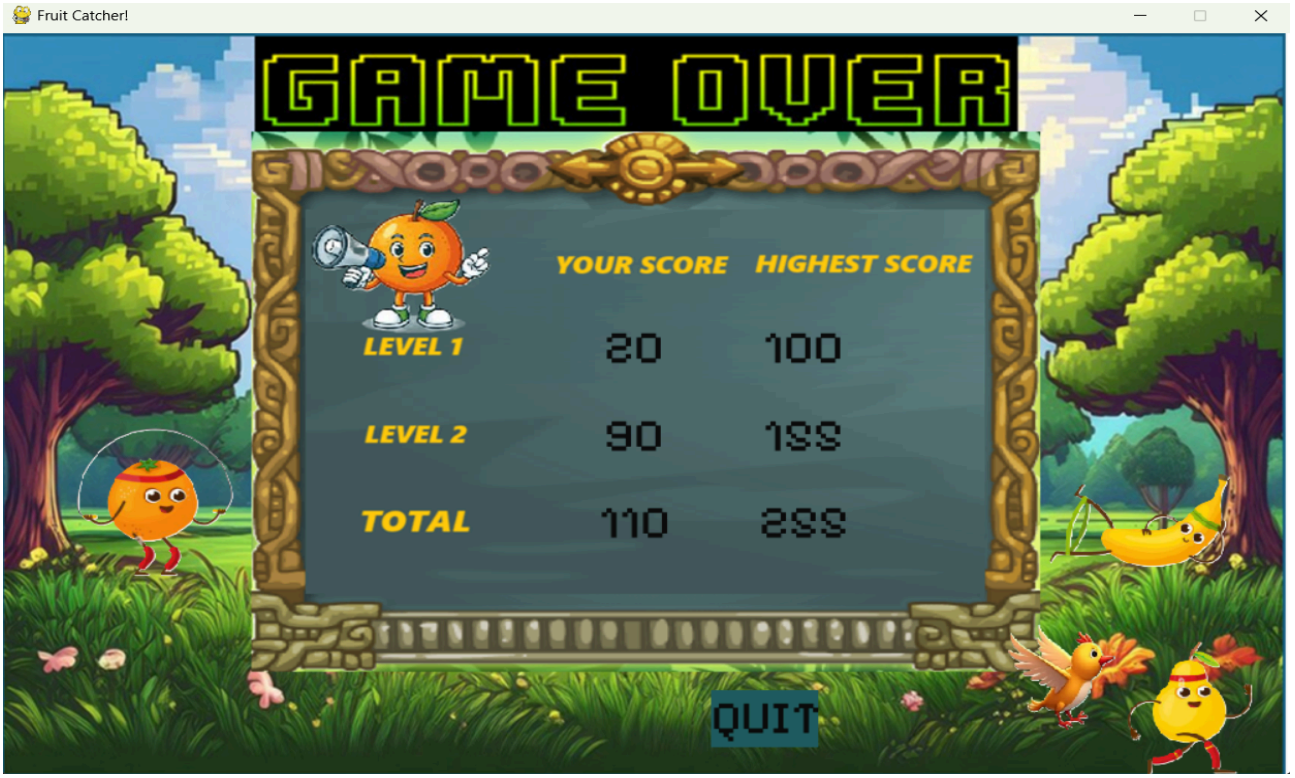
level 2 screen


Fruit Catcher!
Score: 28
Time left: 27.788
Catch only banana


Fruit Catcher!
Score: 0
Time left: 34.724
Catch only apple

End screen


Fruit Catcher!
GAME OVER
YOUR SCORE
HIGHEST SCORE
LEVEL 1
50
100
LEVEL 2
45
155
TOTAL
95
255
QUIT


Fruit Catcher!
GAME OVER
YOUR SCORE
HIGHEST SCORE
LEVEL 1
20
100
LEVEL 2
90
155
TOTAL
110
255
QUIT

Source Code

```
import pygame, sys, random

### Constants ###
FPS = 60
game_start_time=0
lvl2_start_time=0
SCREEN_WIDTH=1000
SCREEN_HEIGHT=650

pygame.init()
pygame.mixer.init()
screen = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))
pygame.display.set_caption("Fruit Catcher!")
clock = pygame.time.Clock()
start_tick = pygame.time.get_ticks()

### GLOBAL VARIABLES ###
scorez=''
player_speed = 10
fall_speed = 2
spawn_rate = 2

seconds = 0
last_apple_time = 0
cur_fruit = ""
level_one_score = 0
level_two_score = 0
high_l1 = 0
high_l2 = 0

gameState = 'start'
player_pos = pygame.Vector2((screen.get_width() / 2)-100,
(screen.get_height() - 100))
fruit_dict = [{'name': "apple", 'loc': [500,0], 'rect': None}]

bckgrnd = pygame.image.load('start-title.png')
l1b = pygame.image.load('l1_back.png')
l2b = pygame.image.load('l2_back.png')
```

```
lvlintr = pygame.image.load('lvl1.png')
apfrt = pygame.image.load("apples.png")
bafrt = pygame.image.load("banana.png")
stfrt = pygame.image.load("strawberry.png")
bomb = pygame.image.load("bomb.png")
hud_font = pygame.font.Font('pixelify.ttf', 40)
sfont = pygame.font.Font('pixelify.ttf',50)

def set_state(state):
    global gameState
    gameState = state

def backimg(img):
    size = pygame.transform.scale(img,(1000,650))
    screen.blit(size,(0,0))

### PLAYER SPECIFIC FUNCTIONS ###

def score_change(change):
    global level_one_score, level_two_score
    if gameState == 'level 1':
        level_one_score += change
        if level_one_score >= 100:
            change_to_inter()
    elif gameState == 'level 2':
        level_two_score += change
        if level_two_score >= 150:
            change_to_end()

def player_control():

    global player_pos

    basket = pygame.image.load("new_basket.png")
    basketrect = basket.get_rect()
    basketrect.x = player_pos.x
    basketrect.y = player_pos.y

    keys = pygame.key.get_pressed()
```

```
    if keys[pygame.K_LEFT] or keys[pygame.K_a]:
        if not player_pos.x < 0:
            player_pos.x -= player_speed
    if keys[ pygame.K_RIGHT] or keys[pygame.K_d]:
        if not player_pos.x > 850:
            player_pos.x += player_speed

    frtrectlist = []
    for frt in fruit_dict:
        frtrectlist.append(frt['rect'])
    col_i = basketrect.collidelist(frtrectlist)
    if col_i != -1:
        col_fruit = fruit_dict[col_i]['name']
        fruit_dict.pop(col_i)

        if cur_fruit == col_fruit:
            sound1=pygame.mixer.Sound("pling.mp3")
            sound1.play(loops=0, maxtime=0, fade_ms=0)
            score_change(10)
        elif col_fruit == 'bomb':
            sound2=pygame.mixer.Sound("small-explosion-129477.mp3")
            sound2.play(loops=0, maxtime=0, fade_ms=0)
            if gameState == 'level 2':
                score_change(-10)
            else:
                score_change(-5)
        else:
            score_change(5)

    screen.blit(basket, basketrect)

### FRUIT SPECIFIC FUNCTIONS ###

def fruit_generator():
    global fruit_dict, last_apple_time

    seconds = (pygame.time.get_ticks()-start_tick)/1000

    if seconds - last_apple_time > spawn_rate:
        fruitlist = ["apple","banana","strawberry", "bomb", "bomb",
```

```
"bomb"]
        a = random.randint(0,5)
        last_apple_time = seconds
        entry = {'name': fruitlist[a], 'loc': [random.randint(25,975),
0], 'rect': None}
        fruit_dict.append(entry)

    for fl in fruit_dict:
        if fl['name'] == 'apple':
            frt = apfrt
        elif fl['name'] == 'banana':
            frt = bafrt
        elif fl['name'] == 'strawberry':
            frt = stfrt
        elif fl['name']=='bomb':
            frt=bomb

        frtrect = frt.get_rect()
        fl['rect'] = frtrect
        frtrect.x = fl['loc'][0]
        frtrect.y = fl['loc'][1]

        ## Update the falling for next round
        fl['loc'][1] = frtrect.y + fall_speed
        if frtrect.y > 650:
            fruit_dict.remove(fl)

        screen.blit(frt, frtrect)

### GAME STATES ###

def mouse1():
    m = pygame.mouse.get_pos()
    if 410<m[0]<410+140 and 525<m[1]<525+80:
        pygame.draw.rect(screen, (135, 232, 97), (410, 525, 140, 80))
        button1()

def button1():
    b=sfont.render('START',True,(255,255,255))
    screen.blit(b,(410,535,100,50))
```

```
def change_to_l1():
    global game_start_time, cur_fruit
    pygame.mixer.music.load("music_lvl1.mp3")
    pygame.mixer.music.play(loops=-1, start=0.0)
    set_state('level 1')
    fruitlist=["apple","banana","strawberry"]
    a=random.randint(0,2)
    cur_fruit = fruitlist[a]
    game_start_time = pygame.time.get_ticks()

def startScreen():
    global game_start_time, cur_fruit

    screen.fill('blue')
    backimg(bckgrnd)
    pygame.draw.rect(screen,(247, 159, 27),(410,525,140,80))
    button1()
    mouse1()

    moupos=pygame.mouse.get_pos()
    moupres=pygame.mouse.get_pressed()[0]
    is_hovered=(410<=moupos[0]<=410+140 and 525<=moupos[1]<=525+80)
    for eve in pygame.event.get():
        if eve.type == pygame.MOUSEBUTTONDOWN and is_hovered:
            change_to_l1()

def timer_disp(l_start):
    global hud_font

    time_left = (60000+(l_start -
start_tick)-pygame.time.get_ticks())/1000
    time_text='Time left: '+str(time_left)
    if time_left > 0:
        timer=hud_font.render(time_text, True, (0,0,0))
        screen.blit(timer,(650,10))
    return time_left

def score_disp(l_score):
    score='Score: ' + str(l_score)
```

```
    text_image = hud_font.render(score, True, (0,0,0))
    screen.blit(text_image, (40,10))

def add_score():
    global high_l1, high_l2
    try:
        file = open('save.txt','r+')
        scores = file.readline()
        score_list = scores.split()

        scorez = ""
        if int(score_list[0]) < level_one_score:
            scorez += str(level_one_score)+' '
            high_l1 = level_one_score
        else:
            scorez += str(score_list[0])+' '
            high_l1 = int(score_list[0])

        if int(score_list[1]) < level_two_score:
            scorez += str(level_two_score)+' '
            high_l2 = level_two_score
        else:
            scorez += str(score_list[1])+' '
            high_l2 = int(score_list[1])

        high_total = high_l1 + high_l2
        scorez += str(high_total)
        file.seek(0)
        file.write(scorez)
    except FileNotFoundError:
        file = open('save.txt', 'w+')
        file.write(str(level_one_score)+' '+str(level_two_score)+'
'+str(level_one_score+level_two_score))

# Scene switchers

def change_to_inter():
    pygame.mixer.music.stop()
    set_state('inter')
```

```
# Level One Logic
def level_one():

    backimg(l1b)
    time_left = timer_disp(game_start_time)
    score_disp(level_one_score)

    catch = "Catch only " + cur_fruit

    catch_text = hud_font.render(catch, True, (255,255,0))
    catch_rect= catch_text.get_rect()
    catch_rect.center = (1000// 2, 650// 2)
    screen.blit(catch_text,catch_rect)
    keys = pygame.key.get_pressed()

    if time_left<=0:
        change_to_inter()

    if keys[pygame.K_SPACE]:
        pygame.mixer.music.stop()
        set_state('inter')

    fruit_generator()
    player_control()

def change_to_l2():
    global lvl2_start_time, cur_fruit
    global fall_speed, spawn_rate

    pygame.mixer.music.load("music_lvl2.mp3")
    pygame.mixer.music.play(loops=-1, start=0.0)

    fall_speed = 6
    spawn_rate = 1

    set_state('level 2')
    fruitlist=["apple","banana","strawberry"]
    a = random.randint(0,2)
    cur_fruit = fruitlist[a]
    lvl2_start_time = pygame.time.get_ticks()
```

```
def intermediate():

    screen.blit(lvlintr,(0,0))
    intr_font = pygame.font.Font('pixelify.ttf', 80)
    intr_scr = 'Score:' + str(level_one_score)
    intr_disp = intr_font.render(intr_scr,True,(46,0,32))
    intr_disp_shadow = intr_font.render(intr_scr,True,(255,99,71))
    screen.blit(intr_disp,(200,450))
    screen.blit(intr_disp_shadow, (205, 445))

    intr_nxt='Press "ENTER" for next level'
    intr_nfont = pygame.font.Font('pixelify.ttf', 20)
    intr_btm = intr_nfont.render(intr_nxt,True,(46,0,32))
    screen.blit(intr_btm,(350,600))
    key=pygame.key.get_pressed()

    if key[pygame.K_RETURN] or key[pygame.K_KP_ENTER]:
        change_to_l2()

def change_to_end():
    pygame.mixer.music.stop()
    set_state('end')

def level_two():

    backimg(l2b)
    tl = timer_disp(lvl2_start_time)
    score_disp(level_two_score)

    catch2 = "Catch only " + cur_fruit
    catch_text2 = hud_font.render(catch2,True,(255,255,0))
    catch_rect2= catch_text2.get_rect()
    catch_rect2.center = (1000// 2, 650// 2)
    screen.blit(catch_text2,catch_rect2)
    if tl<=0:
        change_to_end()
    player_control()
    fruit_generator()
```

```
def end_screen():

    add_score()

    mpos=pygame.mouse.get_pos()
    mpres=pygame.mouse.get_pressed()[0]
    gamovr=pygame.image.load('final_background image.png')
    gameover= gamovr.get_rect()
    screen.fill('yellow')
    screen.blit(gamovr,gameover)

    qtext = hud_font.render('QUIT', True, (10,10,10))
    pygame.draw.rect(screen,(33,94,97),(550,575,83,50))
    screen.blit(qtext, (550,575))
    quitrect = qtext.get_rect()
    quitrect.topleft = (550, 575)

    #displaying score
    l1 = level_one_score
    l2 = level_two_score
    highl1 = high_l1
    highl2 = high_l2
    sumyour=l1+l2
    sumhigh=highl1+highl2

    l1dis=hud_font.render(str(l1), True, (10,10,10))
    l2dis=hud_font.render(str(l2), True, (10,10,10))
    highl1dis=hud_font.render(str(highl1), True, (10,10,10))
    highl2dis=hud_font.render(str(highl2), True, (10,10,10))
    sumyourdis=hud_font.render(str(sumyour), True, (10,10,10))
    sumhighdis=hud_font.render(str(sumhigh), True, (10,10,10))

    l1dis_rect= l1dis.get_rect()
    l1dis_rect.center = (490,275)
    screen.blit(l1dis,l1dis_rect)

    l2dis_rect= l2dis.get_rect()
    l2dis_rect.center = (490,352)
    screen.blit(l2dis,l2dis_rect)
```

```
    highl1dis_rect= highl1dis.get_rect()
    highl1dis_rect.center = (621,275)
    screen.blit(highl1dis,highl1dis_rect)

    highl2dis_rect=highl2dis.get_rect()
    highl2dis_rect.center = (621,352)
    screen.blit(highl2dis,highl2dis_rect)

    sumyourdisdis_rect= sumyourdis.get_rect()
    sumyourdisdis_rect.center = (490,428)
    screen.blit(sumyourdis,sumyourdisdis_rect)

    sumhighdis_rect=sumhighdis.get_rect()
    sumhighdis_rect.center = (621,428)
    screen.blit(sumhighdis,sumhighdis_rect)

    ### END GAME LOOP ###
    running = True
    while running:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            elif event.type == pygame.MOUSEBUTTONDOWN:
                mouse_pos = pygame.mouse.get_pos()
                if quitrect.collidepoint(mouse_pos):
                    pygame.quit()
                    sys.exit()

        pygame.display.flip()


### MAIN GAME LOOP ###
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

    if gameState == 'start':
```

```
        startScreen()
    elif gameState == 'level 1':
        level_one()
    elif gameState == 'inter':
        intermediate()
    elif gameState == 'level 2':
        level_two()
    elif gameState == 'end':
        end_screen()

    pygame.display.flip()
    clock.tick(FPS)
```