BINUS UNIVERSITY
BINUS INTERNATIONAL

Assignment Cover Letter

(Individual Work)

Student Information:		Surname	Given Names	Student ID Number
	1.	Andi Yusuf	Muhammad	2101718476

Course Code	**: COMP6502**	**Course Name**	**: Introduction to Programming**
Class	**: L1AC**	**Name of Lecturer(s)**	: 1. Bagus Kerthyayana 2. Tri Asih Budiono
Major	**: CS**		
Title of Assignment (if any)	: Car Simulation		
Type of Assignment	**: Final Project**		
Submission Pattern			
Due Date	**: 8-11-2017**	**Submission Date**	**: 8-11-2017**

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer's instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

Plagiarism/Cheating

BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

Declaration of Originality

By signing this assignment, I understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student:

Muhammad Andi Yusuf

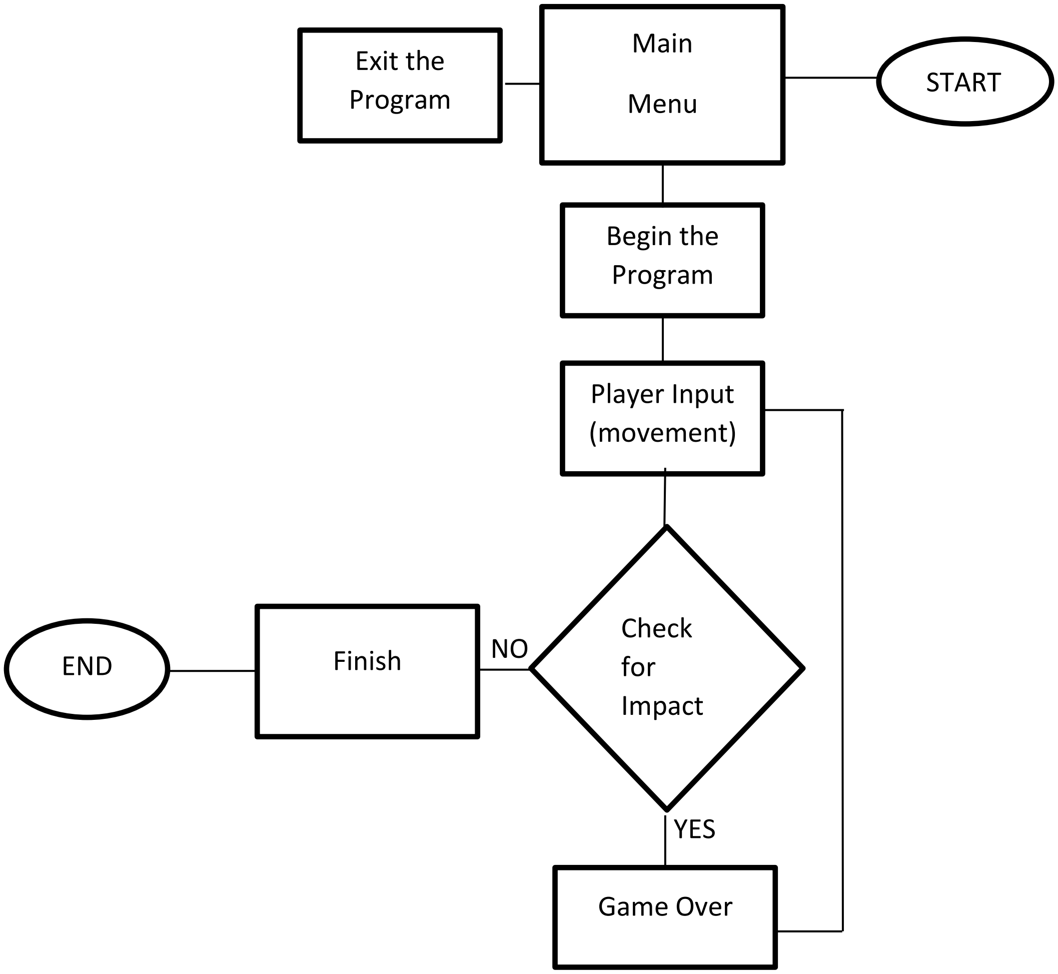
“Car Simulation”
Name = Muhammad Andi Yusuf
ID = 2101718476

I. Description

Function this Program :

This program acts to entertain people who play this game. However, this game includes a scary sound if the car hit the wall. On other word, this car simulation is almost the same as scary maze, but without ghost screen. This game is for the people whom enjoys playing scary games.

II. Flow Chart

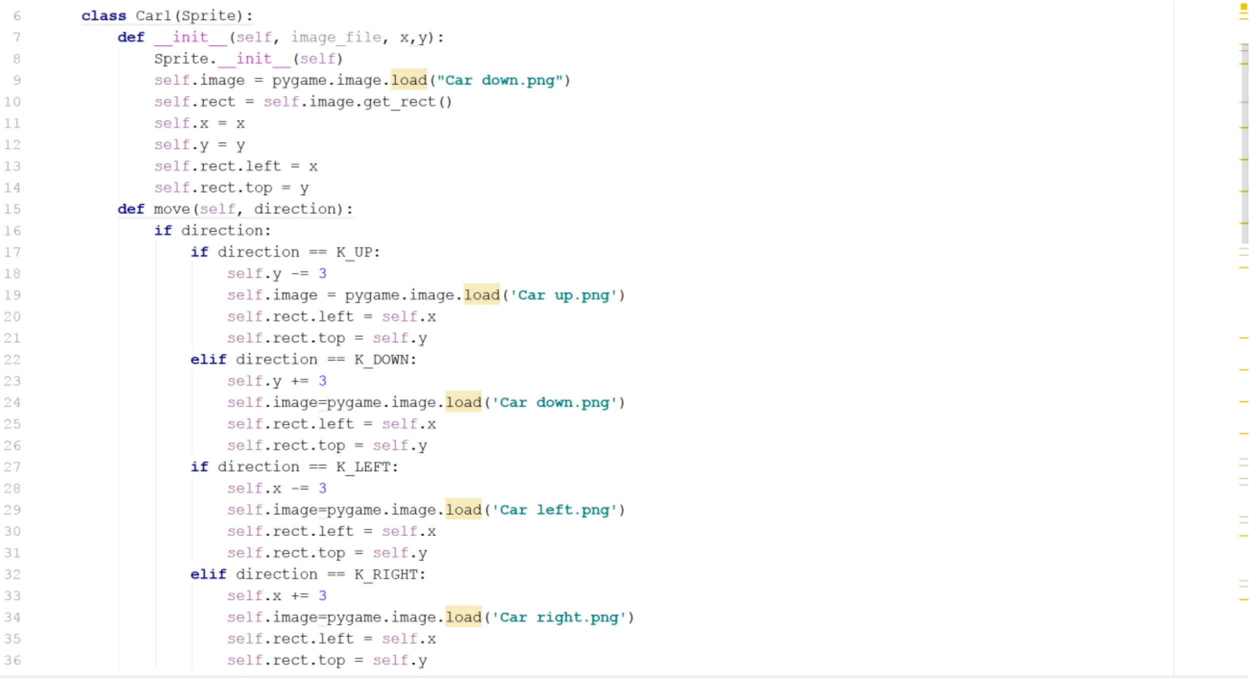


III. Each Function

Class (Class.py):

Car1() and Car2():

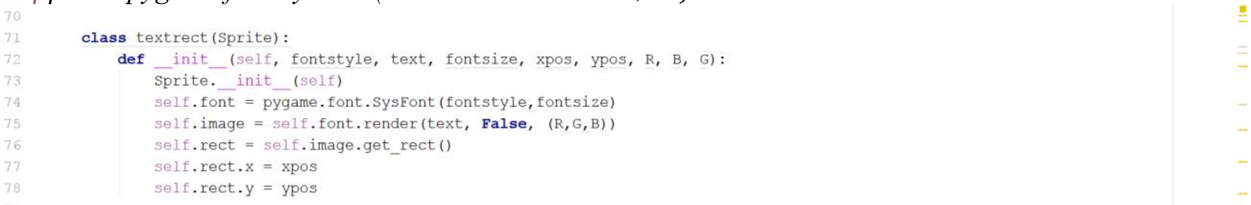
- Car 1 and Car2 Class are the blue print of the car that player will play with during the game
- Def move function is the function that will move the care position so that the car looked like as if it is moving
- For *x -= 3, x+=3, y-=3, y+=3* about car move speed in the game and directions.
- The different between Car1 and Car2 is the stage. Its mean, Car1 for the stage one and Car2 for the stage 2.
-



```python
class Car1(Sprite):
    def __init__(self, image_file, x,y):
        Sprite.__init__(self)
        self.image = pygame.image.load("Car down.png")
        self.rect = self.image.get_rect()
        self.x = x
        self.y = y
        self.rect.left = x
        self.rect.top = y
    def move(self, direction):
        if direction:
            if direction == K_UP:
                self.y -= 3
                self.image = pygame.image.load('Car up.png')
                self.rect.left = self.x
                self.rect.top = self.y
            elif direction == K_DOWN:
                self.y += 3
                self.image=pygame.image.load('Car down.png')
                self.rect.left = self.x
                self.rect.top = self.y
            if direction == K_LEFT:
                self.x -= 3
                self.image=pygame.image.load('Car left.png')
                self.rect.left = self.x
                self.rect.top = self.y
            elif direction == K_RIGHT:
                self.x += 3
                self.image=pygame.image.load('Car right.png')
                self.rect.left = self.x
                self.rect.top = self.y
```

Textrect ():

- Text class are the class that will be used as the blue print to create the object on the screen
- In *self.image = self.font.render(text,* ***False****, (R,G,B))* ➔R,G,B are the color's code for the text. R is red, G is green, B is Blue.
- *self.rect.x = xpos* and *self.rect.y = ypos* are the position text in the screen game.
- *self.font = pygame.font.SysFont(fontstyle,fontsize)* for fontstyle and fontsize. The example is *self.font = pygame.font.SysFont("Times New Roman", 20)*

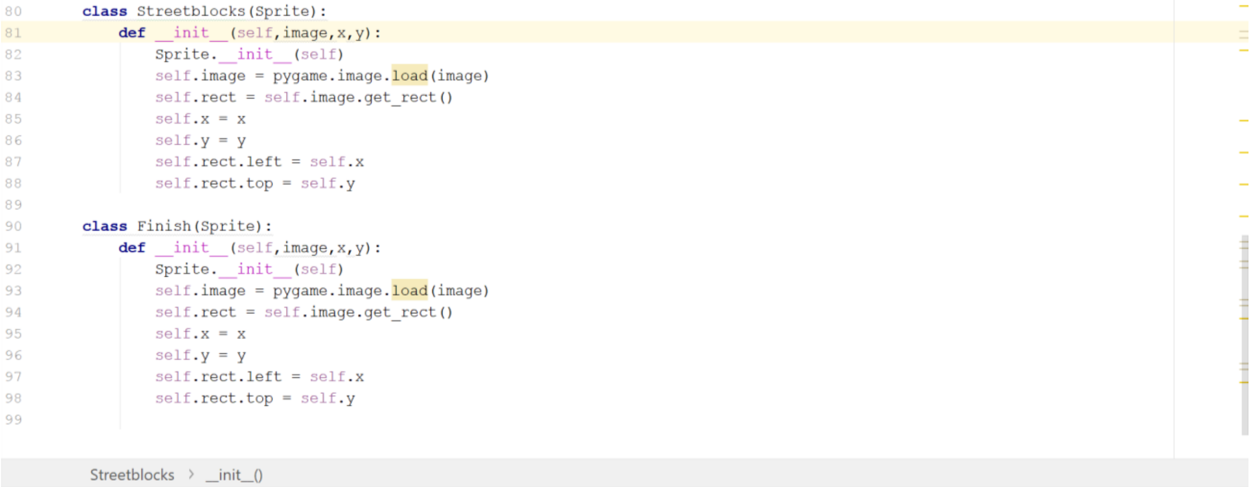


```python

class textrect(Sprite):
    def __init__(self, fontstyle, text, fontsize, xpos, ypos, R, B, G):
        Sprite.__init__(self)
        self.font = pygame.font.SysFont(fontstyle,fontsize)
        self.image = self.font.render(text, False, (R,G,B))
        self.rect = self.image.get_rect()
        self.rect.x = xpos
        self.rect.y = ypos
```

StreetBlocks() and Finish():

- These class function its same, but for different usage
- StreetBlocks() function usage for block the wall, if the car crash the wall it will be game over and you will start in the beginning.
- Finish() function the usage of finish function is to move the game whenever the game is completed



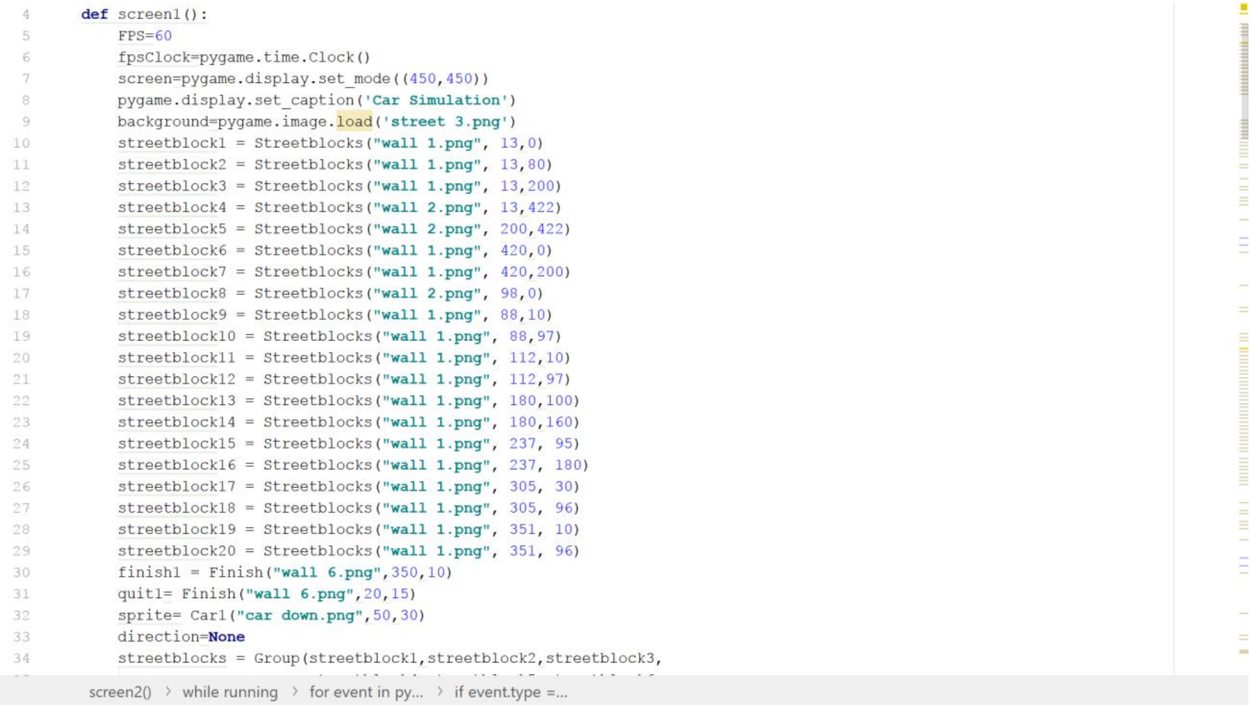
```
class Streetblocks(Sprite):
    def __init__(self,image,x,y):
        Sprite.__init__(self)
        self.image = pygame.image.load(image)
        self.rect = self.image.get_rect()
        self.x = x
        self.y = y
        self.rect.left = self.x
        self.rect.top = self.y

class Finish(Sprite):
    def __init__(self,image,x,y):
        Sprite.__init__(self)
        self.image = pygame.image.load(image)
        self.rect = self.image.get_rect()
        self.x = x
        self.y = y
        self.rect.left = self.x
        self.rect.top = self.y
```

Screen(Screen.py)

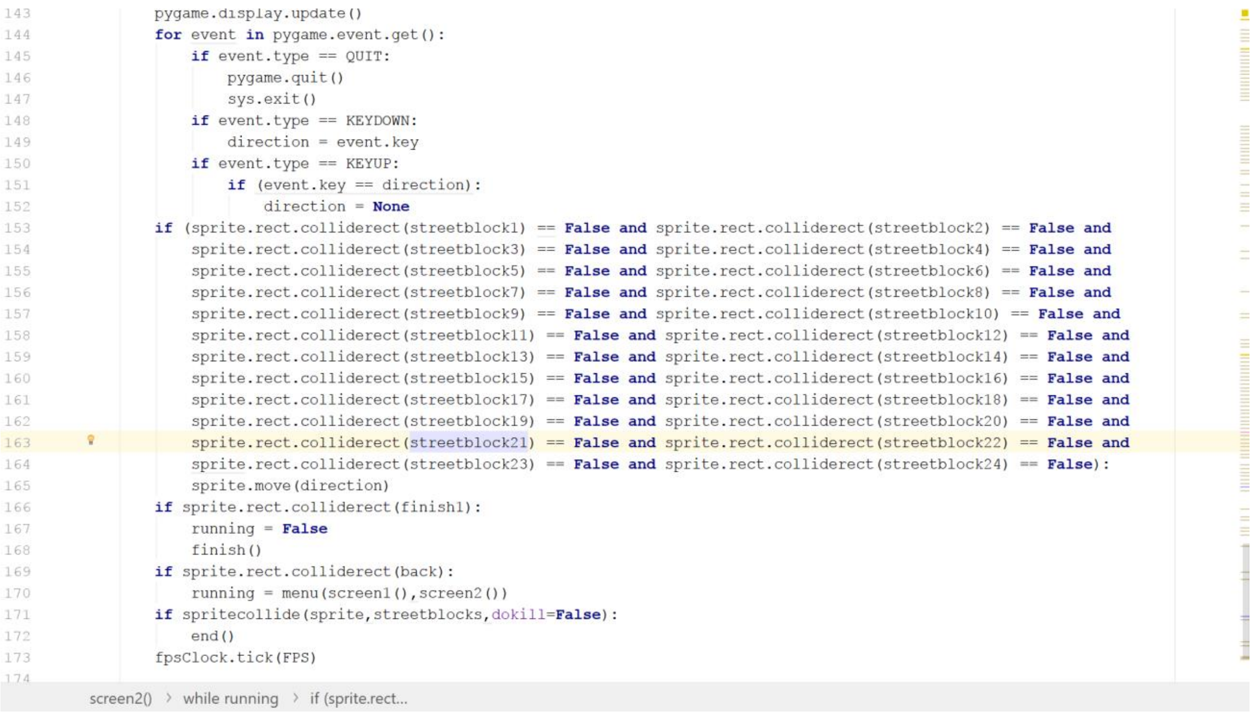
Screen1() and Screen2() :

- The different between Screen1() and Screen2() are the street, position, and streetblocks.
- *fpsClock=pygame.time.Clock()* for the FPS work. If you put FPS = 60 the car will work smoothly and if you put FPS = 30 the car will not smoothly.
- Streetblock for blocked the wall in the street image. Every line wall, I used streetblock and set the position for parallel line with wall street.
- *screen=pygame.display.set_mode((450,450))* for game resolution.
- *seconds=(pygame.time.get_ticks() - start_tick)/1000* for timer in the game. I used "*/1000*" because the time limit use millisecond.
- ***for** event **in** pygame.event.get():* for condition in the game.



```
def screen1():
    FPS=60
    fpsClock=pygame.time.Clock()
    screen=pygame.display.set_mode((450,450))
    pygame.display.set_caption('Car Simulation')
    background=pygame.image.load('street 3.png')
    streetblock1 = Streetblocks("wall 1.png", 13,0)
    streetblock2 = Streetblocks("wall 1.png", 13,80)
    streetblock3 = Streetblocks("wall 1.png", 13,200)
    streetblock4 = Streetblocks("wall 2.png", 13,422)
    streetblock5 = Streetblocks("wall 2.png", 200,422)
    streetblock6 = Streetblocks("wall 1.png", 420,0)
    streetblock7 = Streetblocks("wall 1.png", 420,200)
    streetblock8 = Streetblocks("wall 2.png", 98,0)
    streetblock9 = Streetblocks("wall 1.png", 88,10)
    streetblock10 = Streetblocks("wall 1.png", 88,97)
    streetblock11 = Streetblocks("wall 1.png", 112,10)
    streetblock12 = Streetblocks("wall 1.png", 112,97)
    streetblock13 = Streetblocks("wall 1.png", 180,100)
    streetblock14 = Streetblocks("wall 1.png", 180,160)
    streetblock15 = Streetblocks("wall 1.png", 237, 95)
    streetblock16 = Streetblocks("wall 1.png", 237, 180)
    streetblock17 = Streetblocks("wall 1.png", 305, 30)
    streetblock18 = Streetblocks("wall 1.png", 305, 96)
    streetblock19 = Streetblocks("wall 1.png", 351, 10)
    streetblock20 = Streetblocks("wall 1.png", 351, 96)
    finish1 = Finish("wall 6.png",350,10)
    quit1= Finish("wall 6.png",20,15)
    sprite= Car1("car down.png",50,30)
    direction=None
    streetblocks = Group(streetblock1,streetblock2,streetblock3,
```

screen2() › while running › for event in py... › if event.type =...

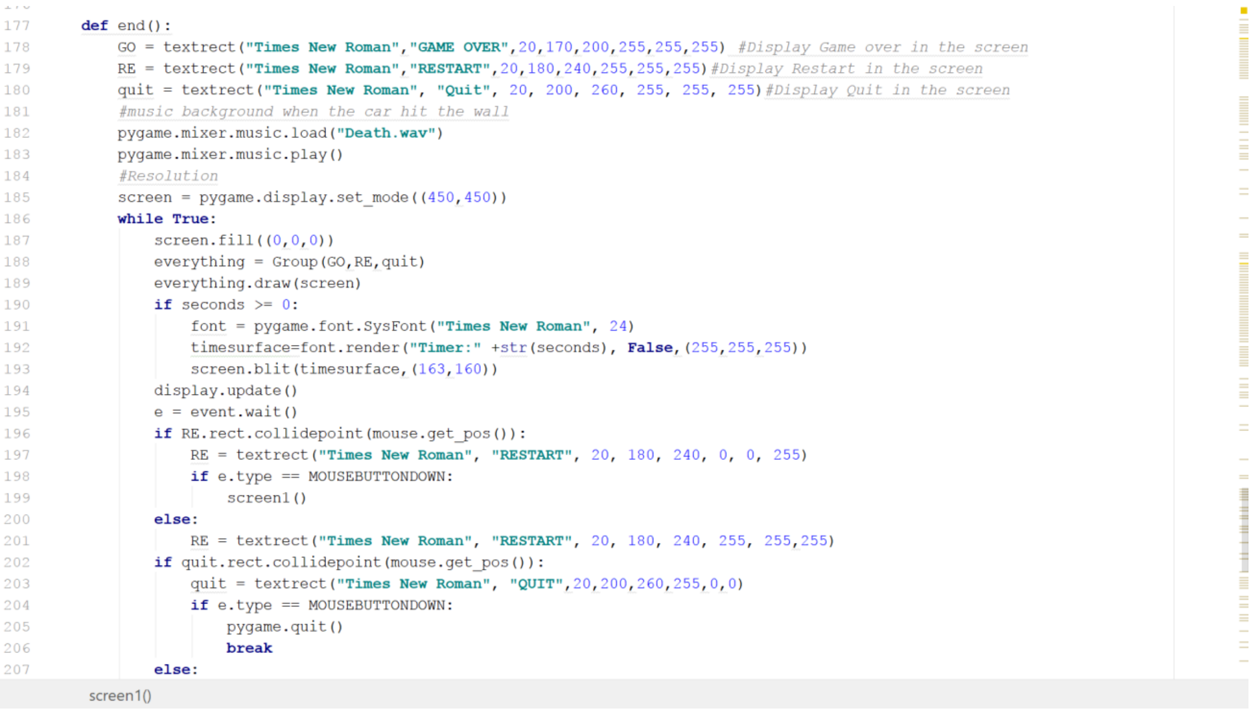


```
pygame.display.update()
for event in pygame.event.get():
    if event.type == QUIT:
        pygame.quit()
        sys.exit()
    if event.type == KEYDOWN:
        direction = event.key
    if event.type == KEYUP:
        if (event.key == direction):
            direction = None
if (sprite.rect.colliderect(streetblock1) == False and sprite.rect.colliderect(streetblock2) == False and
    sprite.rect.colliderect(streetblock3) == False and sprite.rect.colliderect(streetblock4) == False and
    sprite.rect.colliderect(streetblock5) == False and sprite.rect.colliderect(streetblock6) == False and
    sprite.rect.colliderect(streetblock7) == False and sprite.rect.colliderect(streetblock8) == False and
    sprite.rect.colliderect(streetblock9) == False and sprite.rect.colliderect(streetblock10) == False and
    sprite.rect.colliderect(streetblock11) == False and sprite.rect.colliderect(streetblock12) == False and
    sprite.rect.colliderect(streetblock13) == False and sprite.rect.colliderect(streetblock14) == False and
    sprite.rect.colliderect(streetblock15) == False and sprite.rect.colliderect(streetblock16) == False and
    sprite.rect.colliderect(streetblock17) == False and sprite.rect.colliderect(streetblock18) == False and
    sprite.rect.colliderect(streetblock19) == False and sprite.rect.colliderect(streetblock20) == False and
    sprite.rect.colliderect(streetblock21) == False and sprite.rect.colliderect(streetblock22) == False and
    sprite.rect.colliderect(streetblock23) == False and sprite.rect.colliderect(streetblock24) == False):
    sprite.move(direction)
if sprite.rect.colliderect(finish1):
    running = False
    finish()
if sprite.rect.colliderect(back):
    running = menu(screen1(),screen2())
if spritecollide(sprite,streetblocks,dokill=False):
    end()
fpsClock.tick(FPS)
```

Main(Main.py):

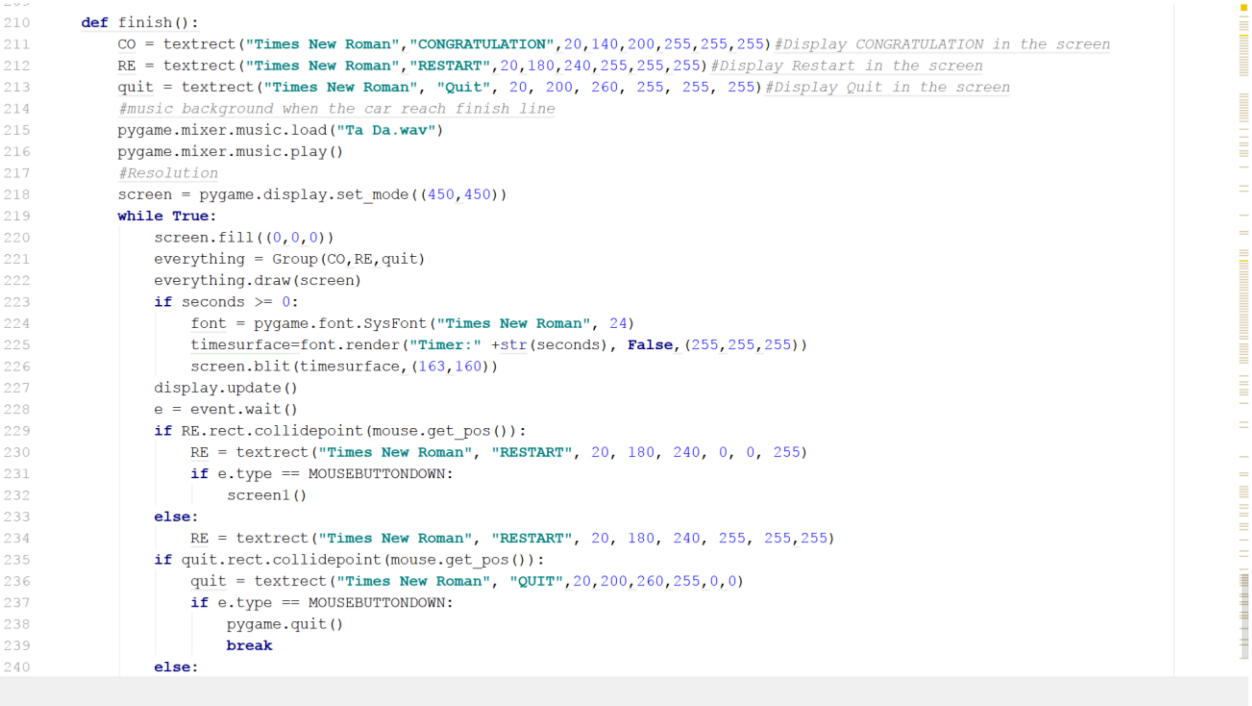
end(), finish(), main():

- end() function is use to check streetblock. If the car collide with the wall, the game will be over and you need to restart the game again.



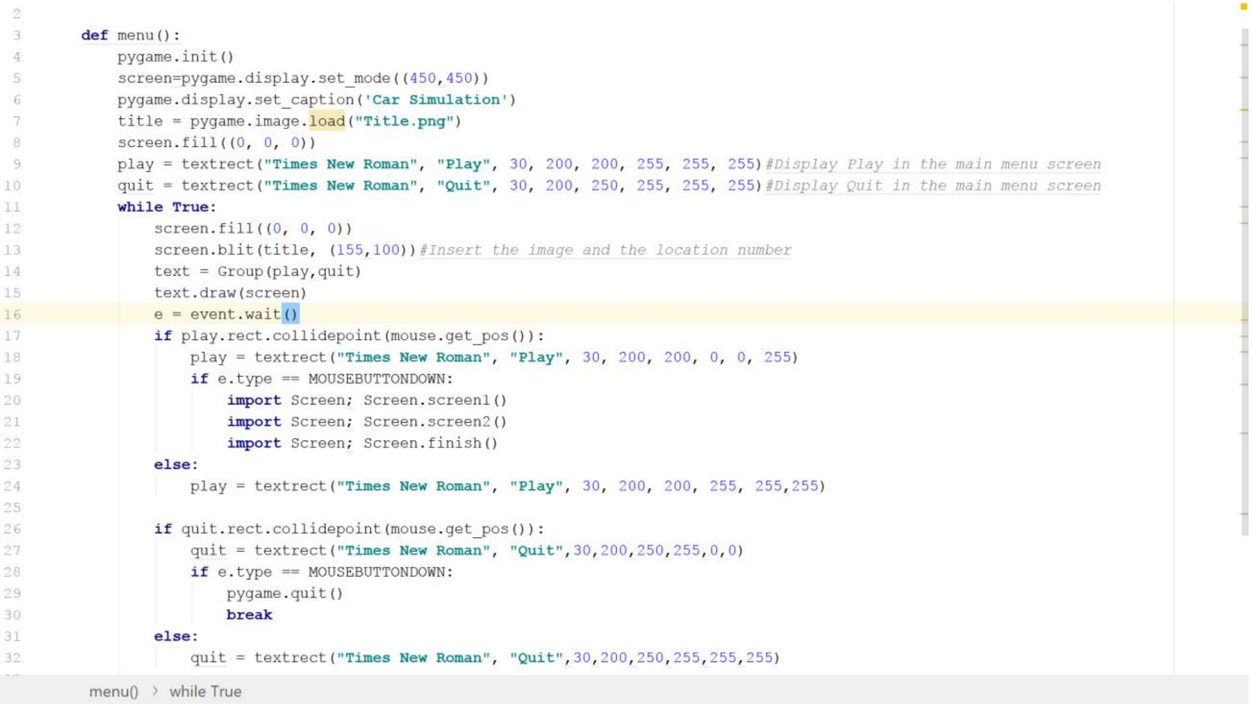
```
def end():
    GO = textrect("Times New Roman","GAME OVER",20,170,200,255,255,255)  #Display Game over in the screen
    RE = textrect("Times New Roman","RESTART",20,180,240,255,255,255)#Display Restart in the screen
    quit = textrect("Times New Roman", "Quit", 20, 200, 260, 255, 255, 255)#Display Quit in the screen
    #music background when the car hit the wall
    pygame.mixer.music.load("Death.wav")
    pygame.mixer.music.play()
    #Resolution
    screen = pygame.display.set_mode((450,450))
    while True:
        screen.fill((0,0,0))
        everything = Group(GO,RE,quit)
        everything.draw(screen)
        if seconds >= 0:
            font = pygame.font.SysFont("Times New Roman", 24)
            timesurface=font.render("Timer:" +str(seconds), False,(255,255,255))
            screen.blit(timesurface,(163,160))
        display.update()
        e = event.wait()
        if RE.rect.collidepoint(mouse.get_pos()):
            RE = textrect("Times New Roman", "RESTART", 20, 180, 240, 0, 0, 255)
            if e.type == MOUSEBUTTONDOWN:
                screen1()
        else:
            RE = textrect("Times New Roman", "RESTART", 20, 180, 240, 255, 255,255)
        if quit.rect.collidepoint(mouse.get_pos()):
            quit = textrect("Times New Roman", "QUIT",20,200,260,255,0,0)
            if e.type == MOUSEBUTTONDOWN:
                pygame.quit()
                break
        else:
```

- finish() function is use the end every stage. If the car collide the finish line, it will next another stage or the end of the game.



```
def finish():
    CO = textrect("Times New Roman","CONGRATULATION",20,140,200,255,255,255)#Display CONGRATULATION in the screen
    RE = textrect("Times New Roman","RESTART",20,180,240,255,255,255)#Display Restart in the screen
    quit = textrect("Times New Roman", "Quit", 20, 200, 260, 255, 255, 255)#Display Quit in the screen
    #music background when the car reach finish line
    pygame.mixer.music.load("Ta Da.wav")
    pygame.mixer.music.play()
    #Resolution
    screen = pygame.display.set_mode((450,450))
    while True:
        screen.fill((0,0,0))
        everything = Group(CO,RE,quit)
        everything.draw(screen)
        if seconds >= 0:
            font = pygame.font.SysFont("Times New Roman", 24)
            timesurface=font.render("Timer:" +str(seconds), False,(255,255,255))
            screen.blit(timesurface,(163,160))
        display.update()
        e = event.wait()
        if RE.rect.collidepoint(mouse.get_pos()):
            RE = textrect("Times New Roman", "RESTART", 20, 180, 240, 0, 0, 255)
            if e.type == MOUSEBUTTONDOWN:
                screen1()
        else:
            RE = textrect("Times New Roman", "RESTART", 20, 180, 240, 255, 255,255)
        if quit.rect.collidepoint(mouse.get_pos()):
            quit = textrect("Times New Roman", "QUIT",20,200,260,255,0,0)
            if e.type == MOUSEBUTTONDOWN:
                pygame.quit()
                break
        else:
```

- main() function is first screen before game start. If Play button is press, it will start the game and if Quit button is press, it will quit the game.



```

def menu():
    pygame.init()
    screen=pygame.display.set_mode((450,450))
    pygame.display.set_caption('Car Simulation')
    title = pygame.image.load("Title.png")
    screen.fill((0, 0, 0))
    play = textrect("Times New Roman", "Play", 30, 200, 200, 255, 255, 255)#Display Play in the main menu screen
    quit = textrect("Times New Roman", "Quit", 30, 200, 250, 255, 255, 255)#Display Quit in the main menu screen
    while True:
        screen.fill((0, 0, 0))
        screen.blit(title,  (155,100))#Insert the image and the location number
        text = Group(play,quit)
        text.draw(screen)
        e = event.wait()
        if play.rect.collidepoint(mouse.get_pos()):
            play = textrect("Times New Roman", "Play", 30, 200, 200, 0, 0, 255)
            if e.type == MOUSEBUTTONDOWN:
                import Screen; Screen.screen1()
                import Screen; Screen.screen2()
                import Screen; Screen.finish()
        else:
            play = textrect("Times New Roman", "Play", 30, 200, 200, 255, 255,255)

        if quit.rect.collidepoint(mouse.get_pos()):
            quit = textrect("Times New Roman", "Quit",30,200,250,255,0,0)
            if e.type == MOUSEBUTTONDOWN:
                pygame.quit()
                break
        else:
            quit = textrect("Times New Roman", "Quit",30,200,250,255,255,255)
```

IV. Source Code

V.

```
import pygame, sys, time
from pygame.locals import *
from pygame.sprite import *
from pygame import *
pygame.init()
class Car1(Sprite):
    def __init__(self, image_file, x,y):
        Sprite.__init__(self)
        self.image = pygame.image.load("Car down.png")
        self.rect = self.image.get_rect()
        self.x = x
        self.y = y
        self.rect.left = x
        self.rect.top = y
    def move(self, direction):
        if direction:
            if direction == K_UP:
                self.y -= 3
                self.image = pygame.image.load('Car up.png')
                self.rect.left = self.x
                self.rect.top = self.y
            elif direction == K_DOWN:
                self.y += 3
                self.image=pygame.image.load('Car down.png')
                self.rect.left = self.x
                self.rect.top = self.y
            if direction == K_LEFT:
                self.x -= 3
                self.image=pygame.image.load('Car left.png')
                self.rect.left = self.x
                self.rect.top = self.y
            elif direction == K_RIGHT:
                self.x += 3
                self.image=pygame.image.load('Car right.png')
                self.rect.left = self.x
                self.rect.top = self.y

class Car2(Sprite):
    def __init__(self, image_file, x,y):
        Sprite.__init__(self)
        self.image = pygame.image.load("Car up.png")
        self.rect = self.image.get_rect()
        self.x = x
        self.y = y
        self.rect.left = x
        self.rect.top = y

    def move(self, direction):
        if direction:
            if direction == K_UP:
                self.y -= 3
                self.image = pygame.image.load('Car up.png')
                self.rect.left = self.x
                self.rect.top = self.y
            elif direction == K_DOWN:
                self.y += 3
                self.image=pygame.image.load('Car down.png')
                self.rect.left = self.x
                self.rect.top = self.y
            if direction == K_LEFT:
                self.x -= 3
                self.image=pygame.image.load('Car left.png')
                self.rect.left = self.x
                self.rect.top = self.y
            elif direction == K_RIGHT:
                self.x += 3
                self.image=pygame.image.load('Car right.png')
                self.rect.left = self.x
                self.rect.top = self.y

class textrect(Sprite):
    def __init__(self, fontstyle, text, fontsize, xpos, ypos, R, B, G):
        Sprite.__init__(self)
        self.font = pygame.font.SysFont(fontstyle,fontsize)
        self.image = self.font.render(text, False, (R,G,B))
        self.rect = self.image.get_rect()
        self.rect.x = xpos
        self.rect.y = ypos

class Streetblocks(Sprite):
    def __init__(self,image,x,y):
        Sprite.__init__(self)
        self.image = pygame.image.load(image)
        self.rect = self.image.get_rect()
        self.x = x
        self.y = y
        self.rect.left = self.x
        self.rect.top = self.y

class Finish(Sprite):
    def __init__(self,image,x,y):
        Sprite.__init__(self)
        self.image = pygame.image.load(image)
        self.rect = self.image.get_rect()
        self.x = x
        self.y = y
        self.rect.left = self.x
        self.rect.top = self.y

def screen1():
    FPS=60
    fpsClock=pygame.time.Clock()
    screen=pygame.display.set_mode((450,450))
    pygame.display.set_caption('Car Simulation')
    background=pygame.image.load('street 3.png')
    streetblock1 = Streetblocks("wall 1.png", 13,0)
    streetblock2 = Streetblocks("wall 1.png", 13,80)
    streetblock3 = Streetblocks("wall 1.png", 13,200)
    streetblock4 = Streetblocks("wall 2.png", 13,422)
    streetblock5 = Streetblocks("wall 2.png", 200,422)
    streetblock6 = Streetblocks("wall 1.png", 420,0)
    streetblock7 = Streetblocks("wall 1.png", 420,200)
    streetblock8 = Streetblocks("wall 2.png", 98,0)
    streetblock9 = Streetblocks("wall 1.png", 88,10)
    streetblock10 = Streetblocks("wall 1.png", 88,97)
    streetblock11 = Streetblocks("wall 1.png", 112,10)
    streetblock12 = Streetblocks("wall 1.png", 112,97)
    streetblock13 = Streetblocks("wall 1.png", 180,100)
    streetblock14 = Streetblocks("wall 1.png", 180,160)
    streetblock15 = Streetblocks("wall 1.png", 237, 95)
    streetblock16 = Streetblocks("wall 1.png", 237, 180)
    streetblock17 = Streetblocks("wall 1.png", 305, 30)
    streetblock18 = Streetblocks("wall 1.png", 305, 96)
    streetblock19 = Streetblocks("wall 1.png", 351, 10)
    streetblock20 = Streetblocks("wall 1.png", 351, 96)
    finish1 = Finish("wall 6.png",350,10)
    quit1= Finish("wall 6.png",20,15)
    sprite= Car1("car down.png",50,30)
    direction=None
    streetblocks = Group(streetblock1,streetblock2,streetblock3,
                         streetblock4,streetblock5,streetblock6,
                         streetblock7,streetblock8,streetblock9,
                         streetblock10,streetblock11,streetblock12,
                         streetblock13,streetblock14,streetblock15,
                         streetblock16,streetblock17,streetblock18,
                         streetblock19,streetblock20)
    everything = Group(sprite)
    global start_tick, seconds
    start_tick=pygame.time.get_ticks()
    running = True
    while running:
        seconds=(pygame.time.get_ticks() - start_tick)/1000
        streetblocks.draw(screen)
        screen.fill((0,0,0))
        screen.blit(background,(0,0))
        if seconds >= 0:
            font = pygame.font.SysFont("Times New Roman", 24)
            timesurface=font.render("Timer:" +str(seconds), False,(0,255,0))
            screen.blit(timesurface,(170,0))
        everything.draw(screen)
        pygame.display.update()

        for event in pygame.event.get():
            if event.type == QUIT:
                pygame.quit()
                sys.exit()
            if event.type == KEYDOWN:
                direction = event.key
            if event.type == KEYUP:
                if (event.key == direction):
                    direction = None
        if (sprite.rect.colliderect(streetblock1) == False and
sprite.rect.colliderect(streetblock2) == False
            and sprite.rect.colliderect(streetblock3) == False and
sprite.rect.colliderect(streetblock4) == False
            and sprite.rect.colliderect(streetblock5) == False and
sprite.rect.colliderect(streetblock6) == False
            and sprite.rect.colliderect(streetblock7) == False and
sprite.rect.colliderect(streetblock8) == False
            and sprite.rect.colliderect(streetblock9) == False and
sprite.rect.colliderect(streetblock10) == False
            and sprite.rect.colliderect(streetblock11) == False and
sprite.rect.colliderect(streetblock12) == False
            and sprite.rect.colliderect(streetblock13) == False and
sprite.rect.colliderect(streetblock14) == False
            and sprite.rect.colliderect(streetblock15) == False and
sprite.rect.colliderect(streetblock16) == False
            and sprite.rect.colliderect(streetblock17) == False and
sprite.rect.colliderect(streetblock18) == False
            and sprite.rect.colliderect(streetblock19) == False and
sprite.rect.colliderect(streetblock20) == False):
            sprite.move(direction)
        if sprite.rect.colliderect(finish1):
            running = False
            screen2()
        if sprite.rect.colliderect(quit1):
            running = menu()
        if spritecollide(sprite,streetblocks,dokill=False):
            end()
        fpsClock.tick(FPS)

def screen2():
    FPS=60
    fpsClock=pygame.time.Clock()
    screen=pygame.display.set_mode((450,450))
    pygame.display.set_caption('Car Simulation')
    background=pygame.image.load('another street.png')
    streetblock1 = Streetblocks("wall 1.png", 0,28)
    streetblock2 = Streetblocks("wall 1.png", 0,200)
    streetblock3 = Streetblocks("wall 1.png", 420,180)
    streetblock4 = Streetblocks("wall 1.png", 420,20)
    streetblock5 = Streetblocks("wall 7.png", 85,357)
    streetblock6 = Streetblocks("wall 7.png", 165,357)
    streetblock7 = Streetblocks("wall 2.png", 15,435)
    streetblock8 = Streetblocks("wall 2.png", 90,435)
    streetblock9 = Streetblocks("wall 2.png", 252,182)
    streetblock10 = Streetblocks("wall 1.png", 252,113)
    streetblock11 = Streetblocks("wall 2.png", -75,239)
    streetblock12 = Streetblocks("wall 2.png", -75,215)
    streetblock13 = Streetblocks("wall 7.png", 70,130)
    streetblock14 = Streetblocks("wall 7.png", 150,130)
    streetblock15 = Streetblocks("wall 7.png", 150,120)
    streetblock16 = Streetblocks("wall 7.png", 70,120)
    streetblock17 = Streetblocks("wall 7.png", 85,330)
    streetblock18 = Streetblocks("wall 7.png", 165,330)
    streetblock19 = Streetblocks("wall 1.png", 350,275)
    streetblock20 = Streetblocks("wall 1.png", 325,275)
    streetblock21 = Streetblocks("wall 2.png", 0,25)
    streetblock22 = Streetblocks("wall 2.png", 80,25)
    streetblock23 = Streetblocks("wall 1.png", 350,-142)
    streetblock24 = Streetblocks("wall 1.png", 320,-142)
    finish1 = Finish("wall 6.png",360,6)
    back = Finish("wall 6.png",380,420)
    sprite= Car2("car up.png",390,380)
    direction=None
    streetblocks = Group(streetblock1,streetblock2,streetblock3,
                         streetblock4,streetblock5,streetblock6,
                         streetblock7,streetblock8,streetblock9,
                         streetblock10,streetblock11,streetblock12,
                         streetblock13,streetblock14,streetblock15,
                         streetblock16,streetblock17,streetblock18,
                         streetblock19,streetblock20,streetblock21,
                         streetblock22,streetblock23,streetblock24)
    everything = Group(sprite)
    running = True
    global seconds
    while running:
        seconds=(pygame.time.get_ticks() - start_tick)/1000
        streetblocks.draw(screen)
        screen.fill((0,0,0))
        screen.blit(background,(0,0))
        if seconds >= 0:
            font = pygame.font.SysFont("Times New Roman", 24)
            timesurface=font.render("Timer:" +str(seconds), False,(0,255,0))
            screen.blit(timesurface,(170,0))
        everything.draw(screen)
        pygame.display.update()
        for event in pygame.event.get():
            if event.type == QUIT:
                pygame.quit()
                sys.exit()
            if event.type == KEYDOWN:
                direction = event.key
            if event.type == KEYUP:
                if (event.key == direction):
                    direction = None
        if (sprite.rect.colliderect(streetblock1) == False and
sprite.rect.colliderect(streetblock2) == False and
            sprite.rect.colliderect(streetblock3) == False and
sprite.rect.colliderect(streetblock4) == False and
            sprite.rect.colliderect(streetblock5) == False and
sprite.rect.colliderect(streetblock6) == False and
            sprite.rect.colliderect(streetblock7) == False and
sprite.rect.colliderect(streetblock8) == False and
            sprite.rect.colliderect(streetblock9) == False and
sprite.rect.colliderect(streetblock10) == False and
            sprite.rect.colliderect(streetblock11) == False and
sprite.rect.colliderect(streetblock12) == False and
            sprite.rect.colliderect(streetblock13) == False and
sprite.rect.colliderect(streetblock14) == False and
            sprite.rect.colliderect(streetblock15) == False and
sprite.rect.colliderect(streetblock16) == False and
            sprite.rect.colliderect(streetblock17) == False and
sprite.rect.colliderect(streetblock18) == False and
            sprite.rect.colliderect(streetblock19) == False and
sprite.rect.colliderect(streetblock20) == False and
            sprite.rect.colliderect(streetblock21) == False and
sprite.rect.colliderect(streetblock22) == False and
            sprite.rect.colliderect(streetblock23) == False and
sprite.rect.colliderect(streetblock24) == False):
            sprite.move(direction)
        if sprite.rect.colliderect(finish1):
            running = False
            finish()
        if sprite.rect.colliderect(back):
            running = menu()
        if spritecollide(sprite,streetblocks,dokill=False):
            end()
        fpsClock.tick(FPS)

def end():
    GO = textrect("Times New Roman","GAME OVER",20,170,200,255,255,255)
    RE = textrect("Times New Roman","RESTART",20,180,240,255,255,255)
    quit = textrect("Times New Roman", "Quit", 20, 200, 260, 255, 255, 255)
    pygame.mixer.music.load("Death.wav")
    pygame.mixer.music.play()
    screen = pygame.display.set_mode((450,450))
    while True:
        screen.fill((0,0,0))
        everything = Group(GO,RE,quit)
        everything.draw(screen)
        if seconds >= 0:
            font = pygame.font.SysFont("Times New Roman", 24)
            timesurface=font.render("Timer:" +str(seconds),
False,(255,255,255))
            screen.blit(timesurface,(163,160))
        display.update()
        e = event.wait()
        if RE.rect.collidepoint(mouse.get_pos()):
            RE = textrect("Times New Roman", "RESTART", 20, 180, 240, 0, 0,
255)
            if e.type == MOUSEBUTTONDOWN:
                screen1()
        else:
            RE = textrect("Times New Roman", "RESTART", 20, 180, 240, 255,
255,255)
        if quit.rect.collidepoint(mouse.get_pos()):
            quit = textrect("Times New Roman", "QUIT",20,200,260,255,0,0)
            if e.type == MOUSEBUTTONDOWN:
                pygame.quit()
                break
        else:
            quit = textrect("Times New Roman",
"QUIT",20,200,260,255,255,255)

def finish():
    CO = textrect("Times New Roman","CONGRATULATION",20,140,200,255,255,255)
    RE = textrect("Times New Roman","RESTART",20,180,240,255,255,255)
    quit = textrect("Times New Roman", "Quit", 20, 200, 260, 255, 255, 255)
    pygame.mixer.music.load("Ta Da.wav")
    pygame.mixer.music.play()
    screen = pygame.display.set_mode((450,450))
    while True:
        screen.fill((0,0,0))
        everything = Group(CO,RE,quit)
        everything.draw(screen)
        if seconds >= 0:
            font = pygame.font.SysFont("Times New Roman", 24)
            timesurface=font.render("Timer:" +str(seconds),
False,(255,255,255))
            screen.blit(timesurface,(163,160))
        display.update()
        e = event.wait()
        if RE.rect.collidepoint(mouse.get_pos()):
            RE = textrect("Times New Roman", "RESTART", 20, 180, 240, 0, 0,
255)
            if e.type == MOUSEBUTTONDOWN:
                screen1()
        else:
            RE = textrect("Times New Roman", "RESTART", 20, 180, 240, 255,
255,255)
        if quit.rect.collidepoint(mouse.get_pos()):
            quit = textrect("Times New Roman", "QUIT",20,200,260,255,0,0)
            if e.type == MOUSEBUTTONDOWN:
                pygame.quit()
                break
        else:
            quit = textrect("Times New Roman",
"QUIT",20,200,260,255,255,255)

def menu():
    pygame.init()
    screen=pygame.display.set_mode((450,450))
    pygame.display.set_caption('Car Simulation')
    title = pygame.image.load("Title.png")
    screen.fill((0, 0, 0))
    play = textrect("Times New Roman", "Play", 30, 200, 200, 255, 255, 255)
    quit = textrect("Times New Roman", "Quit", 30, 200, 250, 255, 255, 255)
    while True:
        screen.fill((0, 0, 0))
        screen.blit(title, (155,100))
        text = Group(play,quit)
        text.draw(screen)
        e = event.wait()
        if play.rect.collidepoint(mouse.get_pos()):
            play = textrect("Times New Roman", "Play", 30, 200, 200, 0, 0,
255)
            if e.type == MOUSEBUTTONDOWN:
                screen1()
                screen2()
                finish()
        else:
            play = textrect("Times New Roman", "Play", 30, 200, 200, 255,
255,255)

        if quit.rect.collidepoint(mouse.get_pos()):
            quit = textrect("Times New Roman", "Quit",30,200,250,255,0,0)
            if e.type == MOUSEBUTTONDOWN:
                pygame.quit()
                break
        else:
            quit = textrect("Times New Roman",
"Quit",30,200,250,255,255,255)

        pygame.display.update()
menu()
```