Pythonprogramming.net

Structure of Game

Import pygame
Initialize pygame
Setup a window
Set a name
Define a Clock
Setup a game loop
((LOGIC))
((Pygame grab event))

// Come out of loop
// after for loop
update everything
Enter your game's Speed (FPS)
To QUIT the GAME

Displaying Images

Loading Images (as characters of game)
Displaying Images
Positioning Images
.
To show our Car
Background Color

=> flipbooks.

Game Dev with Python

```
import pygame
pygame.init()
gameDisplay = pygame.display.set_mode((800,600))   # Tupple
pygame.display.set_caption('Racing')
clock = pygame.time.Clock()
crashed = False
while not crashed:                                  # Tip
#Color Definations
    using pygame.event.get()
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            crashed = True
    pygame.display.update() or flip()
    clock.tick(60)
Python pygame.quit()
quit()
```

Put all in gameLoop

```
def gameLoop():
```

```
carImg = pygame.image.load('car.png')
def car(x,y):
    gameDisplay.blit(carImg,(x,y))
x = 800 * 0.45
y = 600 * 0.8  -> a middle.
car(x,y) (in loopgame)
gameDisplay.fill(white)
```

Use it in start of loop.

Sundaram

x-change = 0 ↗ declare it in init().

Events

If any key pressed down	if event.type == pygame.KEYDOWN:
If left key pressed Move Car Left	if event.key == pygame.K_LEFT:
((new location))	x_change = -5
	elif event.key == pygame.K_RIGHT:
	x_change = 5
If key lifted up ↳ keep car there	if event.type == pygame.KEYUP:
	if event.key == LEFT or event == RIGHT
	x-change = 0
	x = x + x_change

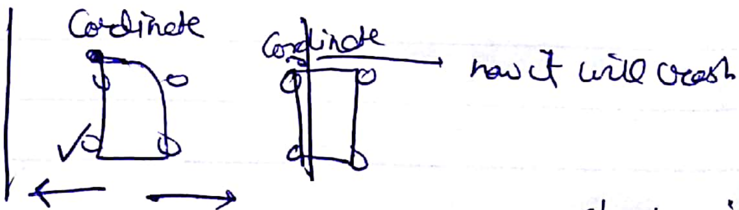
Setting Boundaries:

((Note: If clock ↑ → car moves slower))

If x > 800 or X < 0 :

↓ crashed = True

some of your car will pass the screen



check in PS

∴ DO

Boundaries ← If X + Car width > 800 or x < 0 x - carwidth

crash ()

If crash:

message display ('crashed')

def message_display(text):
Large Text = pygame.font.Font('Arial', 200) → size → select the font
TextSurface, TextRect = text_objects(text, largeText) → write it out on rectangle (MPS)

Display Message
Put it in Crash()

def crash()
~~gameDisp~~:
message_display('You crashed').

Drawing things
+
Blocks falling down

in pygame.init()
in start:
obvious

def things(thingx, thingy, thingw, thingh, color)
pygame.draw.rect (gameDisplay, color,
[thingx, thingy,
thingwidth, thing height])
↓
documentation

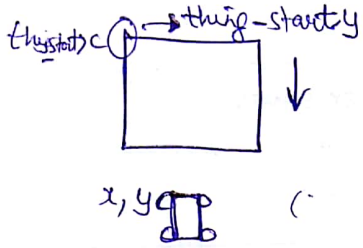
In Game loop:
thing_startx = random.randrange(0, 800) Between
thing_starty = -600
thing_speed = 7
thing_width = 100
thing_height = 100

Call the things we need it in game
We want falling Block
↓
((It will fall but never come back))

in Game's while loop:
things(thing_startx, thing_starty, thing_width, thing_height)
thing_starty += thing_speed
((each time in loop redraw with 7 over it))

Block goes beyond window
↳ make new x, y.

if thing_starty > display_height
thing_starty = 0 - thing_height
thing_startx = random.randrange(0, 800)



Crash

(When Bikes touch)

Car

y crossover

x crossover

Logic:

if y < thing_start y + thing_height;
+ (One of the sides should also touch)

if x > thing_start x and x < thing_start x + thing_width
or
x + car_width > thing_start x
and
x + car_width < thing_start x + thing_width

Crash ( )