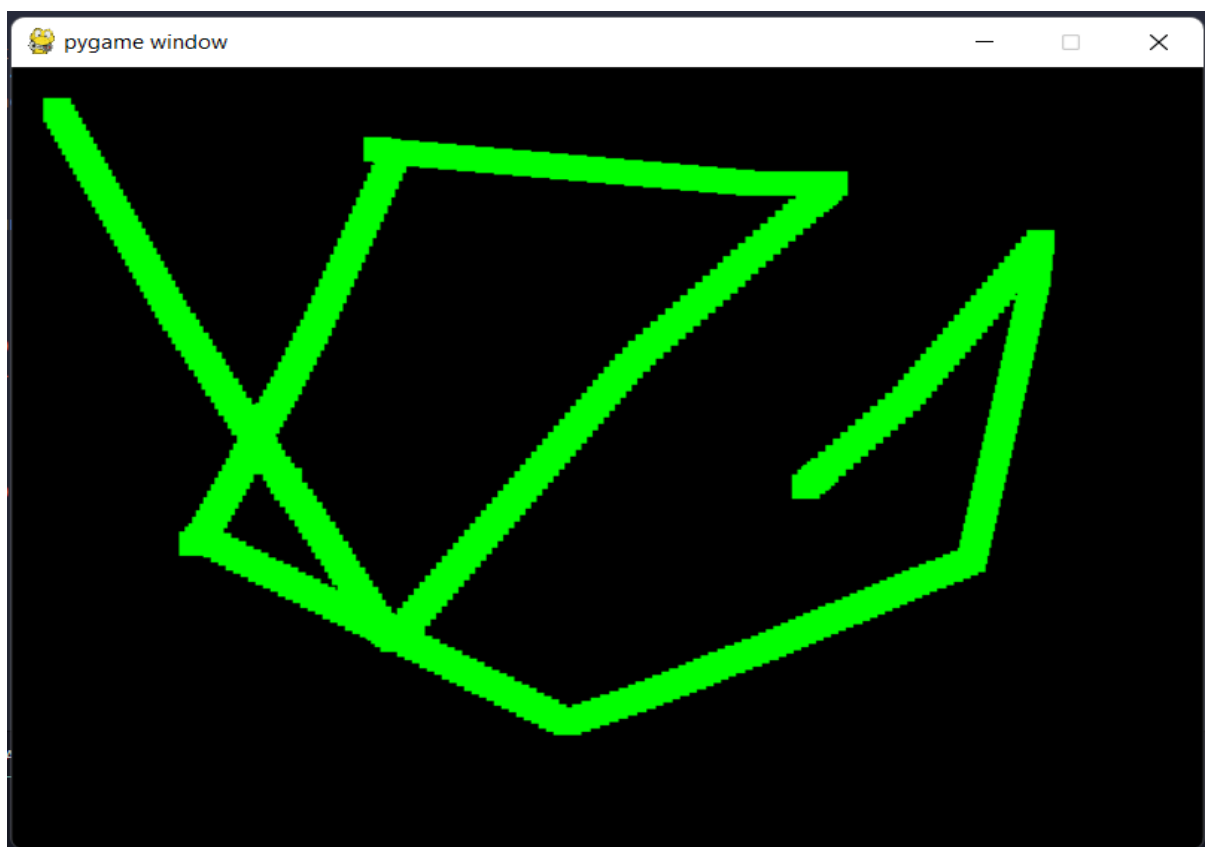


NAMA : KRESHNA PUTRA ADI WICAKSANA  
NIM / KELAS : V3920032 / TI-D  
MATAKULIAH : PRAKTIK GAME DEVELOPMENT

```
1  import pygame
2  import sys
3  import math
4  import random
5  from pygame.locals import *
6
7  pygame.init()
8  screen = pygame.display.set_mode((640, 480))
9  clock = pygame.time.Clock()
10 player = pygame.Rect(15, 15, 15, 15)
11 vec = None
12 direction = None
13 target = 141, 250
14 while True:
15     for event in pygame.event.get():
16         if event.type == pygame.QUIT:
17             pygame.quit()
18             sys.exit()
19
20         if event.type == pygame.MOUSEBUTTONDOWN:
21             target = pygame.mouse.get_pos()
22
23     currentx = player.x
24     currenty = player.y
25
26     vec = (target[0] - currentx, target[1] - currenty)
27     distance = math.sqrt(vec[0]**2 + vec[1]**2)
28     print("distance : ", distance)
29
30     if(distance == 0):
31         pass
32     else:
33         normalize = (vec[0]/round(distance), vec[1]/round(distance))
34         direction = normalize
35         print(normalize)
36         print(direction)
37     print("vector : ", vec)
38     speed_x = round(direction[0] * 5)
39     speed_y = round(direction[1] * 5)
40
41     player.x += speed_x
42     player.y += speed_y
43
44     pygame.draw.rect(screen, (0, 255, 0), player)
45
46     clock.tick(60)
47     pygame.display.update()
```

Pada line 1 – 4 berfungsi untuk mengambil sebuah library pada `pygame`, `sys`, `math`, dan `random` yang kemudian pada line 5 kita ambil dari locals. Pada line 7 kita `pygame.init()` yang dimana pada bagian ini berfungsi untuk menginisialisasi program, kemudian kita membuat deklarasi `screen = pygame.display.set_mode((640, 480))` yang berfungsi untuk menampilkan sebuah display dengan ukuran 640 x 480. Kemudian kita deklarasi clock untuk membuat waktu dengan `pygame.time.Clock()` yang selanjutnya kita membuat player dengan menuliskan `player = pygame.Rect(15, 15, 15, 15)`. Selanjutnya kita buat `vec = None`, dan `direction = None`. Kemudian kita membuat target dengan 141 dan 250. Kemudian pada line 14 berisi perulangan True menggunakan while yang dimana `for event in pygame.event.get()` berfungsi untuk mendapatkan pesan, kemudian jika `event.type == pygame.QUIT` yang artinya jika event.type sama dengan pygame.QUIT maka pygame dan sys keluar. Kemudian jika `event.type == pygame.MOUSEBUTTONDOWN` yang dimana jika mouse bergerak kebawah maka `target = pygame.mouse.get_pos()`.

Selanjutnya kita deklarasi `currentx = player.x` dan `currenty = player.y`. lalu kita deklarasi vec dengan ditulis `vec = (target[0] - currentx, target[1] - currenty)` yang dimana ini untuk membuat sebuah targetnya yang kemudian kita membuat deklarasi kembali untuk membuat jarak dengan `distance = math.sqrt(vec[0]**2 + vec[1]**2)` yang dimana `math.sqrt` digunakan untuk mencari hasil dari akar kuadrat pada sebuah nilai, fungsi tersebut perlu 1 argumen yaitu sebuah nilai yang ingin dihitung. Dan kemudian kita print distancenya. Kemudian pada line 30 jika `distance == 0` maka `pass` sedangkan jika else maka kita deklarasi `normalize = (vec[0]/round(distance), vec[1]/round(distance))` yang kemudian `direction = normalize`. Selanjutnya kita cetak normalize dan direction tadi. Dan pada line 37 kita print vec nya. Kemudian kita deklarasi `speed_x` dan `speed_y`, pada `speed_x` ditulis seperti `speed_x = round(direction[0] * 5)` sedangkan `speed_y` ditulis seperti `speed_y = round(direction[1] * 5)`. Selanjutnya kita menuliskan `player.x += speed_x` dan juga `player.y += speed_y`. kemudian kita membuat gambar dengan `pygame.draw.rect`. selanjutnya `clock.tick(60)` yang berfungsi untuk menentukan timer pada sistem real time. Dan yang terakhir kita menuliskan `pygame.display.update()` untuk mengupdate display / tampilannya.





```
1  import pygame
2  import random
3
4
5  class Ball(pygame.sprite.Sprite):
6      def __init__(self, startpos, velocity, startdir):
7          super().__init__()
8          self.pos = pygame.math.Vector2(startpos)
9          self.velocity = velocity
10         self.dir = pygame.math.Vector2(startdir).normalize()
11         self.image = pygame.image.load("bola.jpg").convert_alpha()
12         self.image = pygame.transform.scale(self.image, (100, 60))
13         self.rect = self.image.get_rect(
14             center=(round(self.pos.x), round(self.pos.y)))
15
16         def reflect(self, NV):
17             self.dir = self.dir.reflect(pygame.math.Vector2(NV))
18
19         def update(self):
20             self.pos += self.dir * self.velocity
21             self.rect.center = round(self.pos.x), round(self.pos.y)
22
23
24  pygame.init()
25  window = pygame.display.set_mode((640, 480))
26  clock = pygame.time.Clock()
27
28  all_groups = pygame.sprite.Group()
29  start, velocity, direction = (250, 250), 5, (random.random(), random.random())
30  ball = Ball(start, velocity, direction)
31  all_groups.add(ball)
32
33  GREY = (128, 128, 128)
34
35  run = True
36  while run:
37      clock.tick(60)
38      for event in pygame.event.get():
39          if event.type == pygame.QUIT:
40              run = False
41
42      all_groups.update()
43
44      if ball.rect.left ≤ 100:
45          ball.reflect((1, 0))
46      if ball.rect.left ≥ 300:
47          ball.reflect((-1, 0))
48      if ball.rect.top ≤ 100:
49          ball.reflect((0, 1))
50      if ball.rect.top ≥ 300:
51          ball.reflect((0, -1))
52
53      window.fill(GREY)
54      pygame.draw.rect(window, (255, 0, 0), (100, 100, 300, 300), 1)
55      all_groups.draw(window)
56      pygame.display.flip()
57
58  pygame.quit()
```

Yang pertama kita lakukan adalah memanggil sebuah modul *pygame* dan *random*. Selanjutnya kita membuat class *Ball* yang terdiri dari 3 variabel, kemudian *def \_\_init\_\_(self, startpos, velocity, startdir):* digunakan untuk membuat sebuah fungsi konstruktor. *super().\_\_init\_\_()* berfungsi untuk membangun atau menginisialisasi. Kemudian *self.pos = pygame.math.Vector2(startpos)* digunakan untuk membuat post secara otomatis, *self.velocity = velocity* berfungsi untuk mendeklarasikan sebuah persamaan, dan kemudian *self.dir = pygame.math.Vector2(startdir).normalize()* berfungsi untuk membuat sebuah *self* directory yang berisi vector dan cara terdekat. *self.image = pygame.image.load("bola.jpg").convert\_alpha()* digunakan untuk memanggil sebuah file yang berbentuk gambar. Kemudian dibawahnya pada line 12 ini digunakan untuk mengatur ukuran dari gambar yang sebelumnya sudah dimasukkan. Kemudian kita mendeklarasikan *self.rect = self.image.get\_rect(center=(round(self.pos.x), round(self.pos.y)))* yang berfungsi untuk membuat sebuah fungsi reflect atau pantulan. Kemudian *self.dir = self.dir.reflect(pygame.math.Vector2(NV))* *def update(self):* digunakan untuk mengupdate data secara otomatis atau individu. *self.pos += self.dir \* self.velocity* pada line 20 ini berfungsi untuk mendeklarasikan sebuah post yang kurang lebih sama dengan directory dan kecepatan dari pergerakannya. *self.rect.center = round(self.pos.x), round(self.pos.y)* pada line 21 ini digunakan saat object berada pada posisi center pada bangun yang akan meround x dan y. selanjutnya *pygame.init()* berfungsi untuk menginisialisasi. Selanjutnya kita mendeklarasikan window pada line 25 dan clock pada 26. Line 25 ini berfungsi untuk menampilkan display atau window, kemudian pada line 26 ini berfungsi untuk mendeklarasikan waktu pada pygame. Selanjutnya *all\_groups = pygame.sprite.Group()* digunakan untuk membuat gabungan dari beberapa bagian. Kemudian pada line 29 ini digunakan untuk mendeklarasikan nilai koordinat nilai kecepatan dan nilai jarak yang ditulis dengan *start, velocity, direction = (250, 250), 5, (random.random(), random.random())*. Selanjutnya kita mendeklarasikan *ball = Ball(start, velocity, direction)* yang digunakan untuk menjalankan code yang sudah dideklarasikan sebelumnya, dan pada line 31 *all\_groups.add(ball)* digunakan untuk menambahkan variabel bola / ball pada grup yang sudah dibuat.

*GREY = (128,128,128)* digunakan untuk mendeklarasikan sebuah warna yang kemudian pada line 35 kita mendeklarasikan *run = True* yang dimana jika nilai run adalah benar. Selanjutnya while run ini digunakan untuk menjakanan sebuah perulangan. Kemudian *clock.tick(60)* digunakan untuk waktu secara realtime. Kemudian *for event in pygame.event.get():* untuk memasukkan perintah yang dilakukan. Kemudian pada line 39 jika *event.type == pygame.QUIT* maka nilai run yang tadi true menjadi false. Lalu ada *all\_groups.update()* yang digunakan untuk mengupdate data yang ada didalam grup, kemudian ada *if ball.rect.left <= 100:* percabangan jika nilai bangun dari bola kurang dari sama dengan koordinat 100 maka akan menjalankan program didalamnya, selanjutnya *ball.reflect((1,0))* digunakan untuk membuat / menambahkan koordinat pada sumbu x sehingga akan memantul. Line 46 – 51 hampir sama dengan line 44 dan 45. Kemudian pada line 53 yaitu *window.fill(GREY)* adalah digunakan untuk memberikan warna pada dasar windownya, kemudian *pygame.draw.rect(window, (255, 0, 0), (100, 100, 300, 300), 1)* digunakan untuk membuat sebuah batasan untuk bola memantul, kemudian *all\_groups.draw(window)* digunakan untuk menggambar seluruh data dari grup ke window, dan lalu *pygame.display.flip()* digunakan untuk membalikkan displaynya, dan yang terakhir *pygame.quit()* digunakan untuk memberhentikan programnya.

