**DOKUMENTASI TUGAS PYGAME**

**PEMROGRAMAN BERORIENTASI OBJECT**



Oleh :

1. Dhea Oky Safitri (170411100006)
2. Fitri Anisa (170411100014)

**Teknik Informatika**

**Fakultas Teknik**

**Universitas Trunojoyo Madura**

1. Penjelasan Pygame

pygame (perpustakaan) yaitu perpustakaan bahasa pemrograman phyton Bebas dan Open Source untuk membuat aplikasi multimedia seperti permainan yang dibangun di atas perpustakaan SDL yang baik. Seperti SDL, pygame sangat portabel dan berjalan di hampir semua platform dan sistem operasi.

1. Cara instalasi pygame :
2. mengecek versi python yang dimiliki.
3. Mendownload atau mengunduh file instalasi pygame, dengan cara masuk ke laman link pygame.org/download.shtml . lalu pilih file library pygame sesuai versi python yang sudah di cek tadi dan download file yang sesuai versi python yang dimiliki.
4. extract file hasil download, buka folder “pygame-1.9.2.data” hasil dari ‘extract’ tadi, Lalu buka forder “Headers”, Setelah itu select semuanya, setelah itu klik kanan pada mouse lalu pilih “copy”.
5. Paste file yang sudah di copy, setelah menyalin file – file nya, selanjutnya silahkan kalian pergi ke drive “C”. lalu pilih folder python sesuai versi nya. Contoh “pythonXX” huruf XX ini menandakan versi python nya. Contoh “python34”. Silahkan masuk ke folder “pythonxx” nya, lalu masuk ke folder “include”, Nah di folder ini silahkan buat folder baru dengan nama “pygame”, Lalu paste file yang sudah di copy tadi kedalam folder “pygame” ini.
6. copy paste folder pygame, silahkan copy folder “pygame” dan “pygame-1.9.2.data” hasil extract tadi, Lalu silahkan masuk ke dalam drive “C” lalu masuk ke folder “pythonxx”. Setelah itu masuk ke folder “Lib”. Lalu masuk ke folder “site-packages”. Nah silahkan paste di dalam folder “site-packages”.
7. Finsih dan pengecekan dengan cara buka IDLE dan coba ketikkan “import pygame” jika tidak error berarti tandanya sudah berhasil terinstal.
8. Penjelasan OOP

Pygame.sprite.Sprite

|  |
| --- |
| Player |
| * bullet\_img * bullet\_rect * bullet\_speed * bullets * score |
| * shoot () * move\_bullets() |

|  |
| --- |
| Sprites |
| * image * img\_rect * s\_Pos * img\_rect.topleft * speed |
| * set\_Speed() * update() * move() |

|  |
| --- |
| Display |
|  |
| * createWindow() * setTittle() * createText() |

Pygame.sprite.Sprite

|  |
| --- |
| Transform |
|  |
| * scaleimage() * rotateimage() * load\_sound() * load\_image() * terminate() * wait\_forkey() |

|  |
| --- |
| Enemy |
| * img * b\_img * position * b\_speed |
| * move() * move\_bullets() |

Method and Class from pygame :

+ init()

+ sprite.Sprite

+ surface.Surface

+ sprite.Group

+ surface.Surface.fill

+ surface.Surface.rect

+ time.Clock

+ display.set\_mode

+ display.set\_caption

+ constants.\*

+ event.get

+ event.type

+ sprite.Group.draw

+ display.flip

+ time.Clock.tick

+ quit()

|  |
| --- |
| Sprites |
| * image * img\_rect * s\_Pos * img\_rect.topleft * speed |
| * set\_Speed() * update() * move() |

1. Konsep Class dan Method for Shooter Game

Class Sprites (Contoh OOP jenis Class dan Object)

class Sprite():

def \_\_init\_\_(self, img, position): method constructor

self.img = img

self.img\_rect = self.img.get\_rect()

self.s\_pos = position

self.img\_rect.topleft = self.s\_pos

self.speed =0

Method :

* set\_Speed untuk mengatur kecepatan atau pergerakan dari player
* update untuk perubahan atau pergerakan dari player
* move untuk mencontrol pergerakan player dengan menggunakan keyboard

|  |
| --- |
| Player |
| * bullet\_img * bullet\_rect * bullet\_speed * bullets * score |
| * shoot () * move\_bullets() |

Class Player

class Player(Sprites.Sprite):

def \_\_init\_\_(self, img, b\_img, position, b\_speed):

Sprites.Sprite.\_\_init\_\_(self,img, position)

self.bullet\_img = b\_img

self.bullet\_rect = self.bullet\_img.get\_rect()

self.bullet\_speed = b\_speed

self.bullets = []

self.score = 0

Method :

* shoot untuk mengcreate atau membuat peluru
* move\_bullets mendefinisikan metode yang menggerakkan peluru yang ditargetkan pada musuh

|  |
| --- |
| Display |
|  |
| * createWindow() * setTittle() * createText() |

Class Display:

class Display():

def \_\_init\_\_(self):

pass

def createWindow(self, width, height):

return pygame.display.set\_mode((width, height))

def setTitle(self, title):

pygame.display.set\_caption(title)

def createText(self, text, fontSize, fontColor,x, y,BgColor = None):

font = pygame.font.Font('freesansbold.ttf', fontSize)

fontObj = font.render(text, True, fontColor, BgColor)

fontObjRect = fontObj.get\_rect()

fontObjRect.topleft = x, y

surface = pygame.display.get\_surface()

surface.blit(fontObj, fontObjRect)

Method :

* createWindow untuk membuat screen atau tampilan window
* setTittle untuk mengatur judul atau caption window
* createText untuk mengatur text

|  |
| --- |
| Transform |
|  |
| * scaleimage() * rotateimage() * load\_sound() * load\_image() * terminate() * wait\_forkey() |

Class Transform:

class Transfrom(object):

def \_\_init\_\_(self):

pass

@staticmethod

def scaleImage(self, img, size):

return pygame.transform.scale(img,(size))

@staticmethod

def rotateImage(self, img, angle):

return pygame.transform.rotate(img, angle)

def load\_sound(path):

path = os.path.join(main\_dir, "data", path)

if not pygame.mixer:

return None

try:

data = pygame.mixer.Sound(path)

except pygame.error:

print("Sorry, couldn't load image " +(path) + " " + pygame.get\_error())

Memiliki Method :

* scaleimage yaitu untuk mengatur skala gambar
* rotateimage yaitu untuk mengatur rotasi gambar
* load\_sound yaitu untuk memasukkan suara
* load\_image yaitu untuk memasukkan gambar atau image
* terminate yaitu untuk keluar dari aplikasi

|  |
| --- |
| Enemy |
| * img * b\_img * position * b\_speed |
| * move() * move\_bullets() |

* wait\_forkey yaitu untuk memulai game baru atau keluar dari aplikasi

Class Enemy:

class Enemy(Player):

def \_\_init\_\_(self, img, b\_img, position,b\_speed):

Player.\_\_init\_\_(self, img,b\_img, position, b\_speed)

def move(self):

x = 0

y = self.speed

self.img\_rect.move\_ip(x,y)

def move\_bullets(self):

for bullet in self.bullets:

s = pygame.display.get\_surface()

s.blit(self.bullet\_img, bullet)

bullet.move\_ip(0, self.bullet\_speed)

for bullet in self.bullets:

if bullet.y > 640:

self.bullets.remove(bullet)

Memiliki Method :

* move yaitu untuk mengatur posisi pergerakan musuh
* move\_bullets untuk mengatur pergerakan peluru musuh yang ditargetkan kepada player

1. Contoh konsep Super,Override atau Multiple Inheritance

Dalam program ini tidak memiliki konsep tersebut.

1. Contoh Konsep Polymorphism

Dalam program ini tidak memiliki konsep tersebut.

1. Contoh Konsep Overloading

Dalam program ini tidak memiliki konsep tersebut.

1. Contoh Konsep Enkapsulasi

Dalam program ini tidak memiliki konsep tersebut.

1. Source Code main program game shooter

|  |
| --- |
| from Player import \*  from Enemy import \*  from Modules import \*  import pygame  pygame.mixer.init()  def createText(text, fontSize, fontColor,x, y,BgColor = None):  font = pygame.font.Font('freesansbold.ttf', fontSize)  fontObj = font.render(text, True, fontColor, BgColor)  fontObjRect = fontObj.get\_rect()  fontObjRect.topleft = x, y  surface = pygame.display.get\_surface()  surface.blit(fontObj, fontObjRect)  def main():  pygame.init()  Text\_color = (0,0,200)  score = 0  sound = pygame.mixer.music.load("data/main.ogg")  frame = pygame.time.Clock()  screen = Display().createWindow(640, 640)  player\_shoot = True  enemy\_shoot = True  display\_enemy = True  screen.fill((0,0,234))  size = pygame.display.get\_surface().get\_rect().size  P\_pos = [size[0]/2, size[1]-100]  bullet\_speed = 6  P\_image = load\_image("player.png", True)  E\_image = load\_image("enemy.png", False)  E\_shot = load\_image("Enemyshot.png", False)  P\_shot = load\_image("Playershot.png", False)  back = load\_image("background.png", True)  back = pygame.transform.scale(back,(650, 650))  back\_rect = back.get\_rect()  back\_rect.topleft = 0, -10  player = Player(P\_image, P\_shot, P\_pos, bullet\_speed)  player.set\_speed(50)  EnemyList = []  displayPlayer = True  EnemyAddCounter = 120  EnemyOneCounter = 60  pygame.mouse.set\_visible(True)  bullets = []  shot\_speed = 8  Display().setTitle("Space Shooter")  life = 5  file = open("data/score.pck", "r")  f = file.read()  p\_score = int(f)  file.close()  if(p\_score == 0 or p\_score < 20):  file = open("data/score.pck", "w")  file.write(str(20))  p\_score = 20  file.close()  pygame.mixer.music.play(-1, 0.0)  while True:  screen.fill((0,0,234))  EnemyAddCounter -= 1  EnemyOneCounter -= 1  if(EnemyAddCounter == 0):  EnemyAddCounter = 120  enemy = Enemy(E\_image, E\_shot, (random.randint(0, 315), -10), shot\_speed)  enemy.set\_speed(5)  EnemyList.append(enemy)    if(EnemyOneCounter ==0):  EnemyOneCounter = 60  enemy = Enemy(E\_image, E\_shot, (random.randint(320, 575), -10),shot\_speed)  enemy.set\_speed(5)  EnemyList.append(enemy)    screen.blit(back, back\_rect)  createText("Score: " +str(score), 12, (255, 255, 255),10, 10,None )  createText("Top Score: "+str(p\_score), 12, (255,255,255) ,10, 30, None)  createText("Life: "+str(life), 12, Text\_color ,10, 50, None)    for event in pygame.event.get():  if event.type == QUIT:  pygame.quit()  sys.exit()  if event.type == KEYDOWN:  if event.key == K\_ESCAPE:  pygame.quit()  sys.exit()  if event.key == K\_LEFT and (player.img\_rect.left - 50) >= 0:  player.move("left")  if event.key == K\_RIGHT and (player.img\_rect.right + 50) <= 670:  player.move("right")  if event.key == K\_UP and (player.img\_rect.top - 30) >=0:  player.move("up")  if event.key == K\_DOWN and (player.img\_rect.bottom + 50) <= 640:  player.move("down")  if event.key == K\_SPACE and player\_shoot:  player.shoot()  for enemy in EnemyList:  if(random.randint(0, 50) == 2) and enemy\_shoot == True:  enemy.shoot()  if display\_enemy:  enemy.move\_bullets()  enemy.move()  enemy.update()  for enemy in EnemyList:  if enemy.img\_rect.y > 640:  EnemyList.remove(enemy)  else:  for bull in player.bullets:  if bull.colliderect(enemy.img\_rect):  bullets = enemy.bullets  EnemyList.remove(enemy)  score+= 1  player.bullets.remove(bull)  for bull in enemy.bullets:  if(bull.colliderect(player.img\_rect)):  enemy.bullets.remove(bull)  life -= 1    for rect in bullets:  if(rect.colliderect(player.img\_rect)):  bullets.remove(rect)  life -= 1  if(rect.y > 640):  bullets.remove(rect)  else:  rect.move\_ip(0, 8)  screen.blit(E\_shot, rect)    if(life < 1):  for bullet in player.bullets:  player.bullets.remove(bullet)  for bullet in bullets:  bullets.remove(bullet)  for bullet in enemy.bullets:  enemy.bullets.remove(bullet)  EnemyList = []  displayPlayer = False  player\_shoot = False  enemy\_shoot = False  display\_enemy = False  screen.blit(back, back\_rect)  if(p\_score < score):  file = open("data/score.pck", "w")  file.write(str(score))  file.close()  createText("CONGRATULATION, YOU MADE A NEW HIGH SCORE OF ",20, Text\_color,50, 310, None)  p\_score = score  else:  createText("GAME OVER, YOU FAILED TO MAKE A NEW HIGH SCORE ", 20, Text\_color,50, 310, None)  createText("PRESS Q TO PLAY AGAIN!!!" , 20, Text\_color ,50, 330, None)  pygame.display.update()  wait\_forkey()  score = 0  life = 5  displayPlayer=True  display\_enemy = True  enemy\_shoot = True  player\_shoot = True        player.move\_bullets()  if(displayPlayer == True):  player.update()  pygame.display.update()  frame.tick(60)  if \_\_name\_\_== "\_\_main\_\_":  main() |

1. Source Code Sprites

|  |
| --- |
| import pygame  class Sprite():  LEFT = "left"  RIGHT = "right"  UP = "up"  DOWN = "down"  def \_\_init\_\_(self, img, position):  self.img = img  self.img\_rect = self.img.get\_rect()  self.s\_pos = position  self.img\_rect.topleft = self.s\_pos  self.speed =0    def set\_speed(self, s):  if(s > 0 and s <= 50):  self.speed = s  else:  raise ValueError("the speed must be between one and 50")  def update(self):  surface = pygame.display.get\_surface()  surface.blit(self.img,self.img\_rect)  def move(self, const):  size = pygame.display.get\_surface().get\_rect().size  if(const == "left"):  self.img\_rect.move\_ip(-1 \*self.speed, 0)  if(const=="right"):  self.img\_rect.move\_ip(self.speed, 0)  if(const == "up"):  self.img\_rect.move\_ip(0, -1\*self.speed)  if(const == "down"):  self.img\_rect.move\_ip(0, self.speed) |

1. Source Code Player

|  |
| --- |
| import pygame  import Sprites  class Player(Sprites.Sprite):  def \_\_init\_\_(self, img, b\_img, position, b\_speed):  Sprites.Sprite.\_\_init\_\_(self,img, position)  self.bullet\_img = b\_img  self.bullet\_rect = self.bullet\_img.get\_rect()  self.bullet\_speed = b\_speed  self.bullets = []  self.score = 0  def shoot(self):  bullet = self.bullet\_img.get\_rect()  bullet.topleft = self.img\_rect.center  self.bullets.append(bullet)  def move\_bullets(self):  for bullet in self.bullets:  s = pygame.display.get\_surface()  s.blit(self.bullet\_img, bullet)  bullet.move\_ip(0, -self.bullet\_speed)  for bullet in self.bullets:  if bullet.y < 0:  self.bullets.remove(bullet) |

1. Source Code Enemy

|  |
| --- |
| from Player import \*  import random  class Enemy(Player):  def \_\_init\_\_(self, img, b\_img, position,b\_speed):  Player.\_\_init\_\_(self, img,b\_img, position, b\_speed)  def move(self):  x = 0  y = self.speed  self.img\_rect.move\_ip(x,y)  def move\_bullets(self):  for bullet in self.bullets:  s = pygame.display.get\_surface()  s.blit(self.bullet\_img, bullet)  bullet.move\_ip(0, self.bullet\_speed)  for bullet in self.bullets:  if bullet.y > 640:  self.bullets.remove(bullet) |

1. Source Code Modules

|  |
| --- |
| import pygame, sys, os  from pygame.locals import \*  main\_dir = os.path.split(os.path.abspath(\_\_file\_\_))[0]  class Display():  def \_\_init\_\_(self):  pass  def createWindow(self, width, height):  return pygame.display.set\_mode((width, height))  def setTitle(self, title):  pygame.display.set\_caption(title)  def createText(self, text, fontSize, fontColor,x, y,BgColor = None):  font = pygame.font.Font('freesansbold.ttf', fontSize)  fontObj = font.render(text, True, fontColor, BgColor)  fontObjRect = fontObj.get\_rect()  fontObjRect.topleft = x, y  surface = pygame.display.get\_surface()  surface.blit(fontObj, fontObjRect)    class Transfrom(object):  def \_\_init\_\_(self):  pass  @staticmethod  def scaleImage(self, img, size):  return pygame.transform.scale(img,(size))  @staticmethod  def rotateImage(self, img, angle):  return pygame.transform.rotate(img, angle)    # sound  def load\_sound(path):  path = os.path.join(main\_dir, "data", path)  if not pygame.mixer:  return None  try:  data = pygame.mixer.Sound(path)  except pygame.error:  print("Sorry, couldn't load image " +(path) + " " + pygame.get\_error())  return data  def load\_image(path, transparent):  path = os.path.join(main\_dir, "data", path)  if not pygame.image:  return None  try:  data = pygame.image.load(path)  except pygame.error:  print("Couldn't load image file " + path+ " " + pygame.get\_error())  if transparent:  corner = data.get\_at((0,0))  data.set\_colorkey(corner, RLEACCEL)  return data.convert()  def terminate():  pygame.quit()  sys.exit()  def wait\_forkey():  while True:  for event in pygame.event.get():  if event.type == QUIT:  terminate()  if event.type == KEYDOWN:  if event.key == K\_ESCAPE:  terminate()  if event.key == ord("q"):  return |

1. Screen Shoot game Shooter



