|  |
| --- |
| BHEENI AGARWAL, SAKSHAM GOEL |



|  |  |
| --- | --- |
|  | GAME OF TANKS |

**ACKNOWLEDGEMENT**

We, the students of AIS MAYUR VIHAR acknowledge the valuable contribution of **Ms. Deepshikha Sethi** in providing us the proper guidance to complete this project. The project would not have been completed without her support and kind help.

**Bheeni Agarwal**

**Saksham Goel**

**IX – A, B**

**INTRODUCTION**

The following project is a game named Tanks.

It is a simple single player game. Relying on the concepts of physics the game is about a player tank trying to destroy an enemy tank using accurate angle and power settings and position to get a perfect hit which it has to obtain from the user.

The whole game continuously relies on the user inputs for its working. It is a very interesting and fun game.

Pygame module has been largely used in the making of the game as it allows to create 2D motion by constantly changing the scene to create a motion effect. Alongside that it does uses several other modules and functions which makes it a whole lot more intriguing and attractive.

Once the player wins he can enter his name amongst the top scorers of the game ever and be remembered forever. The player can go for another round of gaming too as the game provides a feature of playing again so that the user can experience the fun again and again and again!!

**SETUP’S DESCRIPTION**

* **SPECIAL FEATURES**
* Pygame has been used to create the window and animation in the game.
* Database Management has been used to record high scores and display it when asked to.
* The program has been divided into various classes and functions for modularity and data abstraction.
* Special attention is given on the declaration of the various function so that no scope issue arises in accessing the variables.
* A sincere effort has been made to make the program run at optimum speed.
* Also program is made robust and comprehensive to help the reader and user by giving necessary instructions and writing required comments respectively.
* The setup is given a visual appeal by using colourful and interactive display.
* The game also uses mathematical calculations and coordinate system for the enemy to locate and attack the player tank giving it an AI like feeling.
* **NAMES OF FILES USED:**

# scores.db used to store the highscores.

* Door.png
* Wall.png
* Aft.png
* Bg1.png
* Bg2.png
* EXPLOSION SPRITES
  1. Exp1.png
  2. Exp2.png
  3. Exp3.png
  4. Exp4.png
  5. Exp5.png
  6. Exp6.png
  7. Exp7.png
  8. Exp8.png
  9. Exp9.png
  10. Exp10.png
  11. Exp11.png
  12. Exp12.png
  13. Exp13.png
  14. Exp14.png
  15. Exp15.png
* Firecircle1.png
* TANK SPRITES
  1. T\_body1.png
  2. T\_body2.png
  3. T\_body3.png
  4. T\_body4.png
  5. T\_body5.png
  6. T\_turret1.png
  7. T\_turret2.png
  8. T\_turret3.png
  9. T\_turret4.png
  10. T\_turret5.png
* Tank.png
* Tankicon.png

**NAME OF MODULES USED:**

* Pygame
* math
* time
* random
* sqlite3
* copy
* **CLASSES AND FUNCTIONS USED:**

**CLASSES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CLASS NAME | FUNCTION NAME | USE | PARAMETERS | USE |
| DISPLAY | msg\_2\_screen | display text on the screen. | msg | message to be displayed |
| f\_name | name of font |
| f\_color | color of the text |
| x\_displace | horizontal displacement of text from the centre |
| y\_displace | vertical displacement of text from the centre |
| size | size of the text |
| b | bold or not |
| i | italics or not |
| f\_lcn | location of font(provided if custom fonts used) |
| text\_2\_button | Displays text on the button | x | x coordinate of centre of text |
| y | y coordinate of centre of text |
| text | text to be displayed |
| f\_name | name of font |
| f\_color | color of the text |
| size | size of the text |
| b | bold or not |
| i | italics or not |
| f\_lcn | location of font(provided if custom fonts used) |
| button | Creates a button | x | x coordinate of button |
| y | y coordinate of button |
| width | width of button |
| height | height of button |
| inactive | color of button when cursor is not on it |
| active | color of button when cursor is on it |
| action | function of button |
| Button1 | Creates a button with a picture as icon | i | image number to display |
| x | x coordinate of button |
| y | y coordinate of button |
| t | check if its body or turret |
| PLAYER | \_\_init\_\_ | assigns variable to constantly changing values | self |  |
| x | x coordinate of tank |
| y | y coordinate of tank |
| angle | angle of turret of tank |
| draw | draws the player tank on the screen | win | variable for display screen |
| itur | turret number |
| ibod | body number |
| x\_barrier | x coordinate of wall |
| fire | fires and shows its animation | x\_barrier | x coordinate of wall |
| y\_barrier | y coordinate of wall |
| itur | turret number |
| ibod | body number |
| ENEMY | \_\_init\_\_ | assigns variable to constantly changing values | self |  |
| x | x coordinate of tank |
| y | y coordinate of tank |
| angle | angle of turret of tank |
| rand | provides random shift to provide some error | self.shift | random shift in position |
| self.angle\_shift | random shift in angle |
| mind | calculates the angle for perfect hit | R | distance between tanks |
| self.exp\_angle | expected angle for perfect hit |
| draw | draws the player tank on the screen | win | variable for display screen |
| x\_barrier | x coordinate of wall |
| fire | fires and shows its animation | x\_barrier | x coordinate of wall |
| y\_barrier | y coordinate of wall |

**FUNCTIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| **FUNCTION NAME** | **USE** | **PARAMETERS** | **USE** |
| colour() | returns RGB code of color asked for | colour | colour name |
| fontor() | creates text as a surface for the required font and features | f\_name | font name |
| size | size of text |
| b | bold or not |
| i | italics or not |
| f\_lcn | location of the custom font(if used) |
| pause() | display pause screen when the game is paused |  | (relies on event handling(mouse click or positioning and keystrokes)) |
| game\_intro() | displays the intro screen when the game is started |  | (relies on event handling(mouse click or positioning and keystrokes)) |
| controls() | display controls of the game |  |  |
| custom() | lets the user choose custom body and turret for their tank |  |  |
| barriers() | creates a wall of random height at a random place on the ground | x | x coordinate of the wall |
| y | y coordinate of wall |
| play() | starts the main part of the game(the game itself) | itur | turret number |
| ibod | tank body number |
| end() | animation when either player or enemy has won |  |  |
| aftergame() | displays options like entering and viewing scores,play again and quit |  |  |
| enter\_score() | displays screen where user inputs name for highscore |  |  |
| scores\_init() | accesses the scores database and retrieves the data |  |  |
| scores() | displays all the highscores on a new screen | con | sq.connect('scores.db') |
| cur | con.cursor() |

**PYTHON CODE**

import pygame

import math as m

import time

import random

import sqlite3 as sq

import copy

pygame.init()

#####################################################

#building display

d\_width=800

d\_height=600

bg=pygame.transform.scale(pygame.image.load('bg1.png'),(d\_width,d\_height))

dg=pygame.transform.scale(pygame.image.load('bg2.png'),(d\_width,d\_height\*7//60))

fps=30

gunshot=pygame.mixer.Sound('gunshot.wav')

explosion=pygame.mixer.Sound('explosion.wav')

tankmove=pygame.mixer.Sound('tankmove.wav')

clock=pygame.time.Clock()

win=pygame.display.set\_mode((d\_width,d\_height))

pygame.display.set\_caption("Tanks")

game\_icon=pygame.image.load('tankicon.png')

pygame.display.set\_icon(pygame.transform.scale(game\_icon,(32,32)))

exp=[]

door=pygame.transform.scale(pygame.image.load('door.jpg'),(d\_width,d\_height))

for i in range(1,16):

exp.append(pygame.image.load('exp%d.png'%i))

tank\_body=[pygame.image.load('t\_body1.png'),pygame.image.load('t\_body2.png'),pygame.image.load('t\_body3.png'),pygame.image.load('t\_body4.png'),pygame.image.load('t\_body5.png')]

tank\_turret=[pygame.image.load('t\_turret1.png'),pygame.image.load('t\_turret2.png'),pygame.image.load('t\_turret3.png'),pygame.image.load('t\_turret4.png'),pygame.image.load('t\_turret5.png')]

######################################################ALL CLASSES

class display(object):

def msg\_2\_screen(msg,f\_name,f\_color,x\_displace=0,y\_displace=0,size=25,b=0,i=0,f\_lcn=""):

fonte=fontor(f\_name,size,b,i,f\_lcn)

screen\_text=fonte.render(msg,True,colour(f\_color))

win.blit(screen\_text,

[(d\_width/2)-screen\_text.get\_rect().width/2+x\_displace,d\_height/2+y\_displace-screen\_text.get\_rect().height/2])

def text\_2\_button(x,y,text,f\_name,f\_color,size=25,b=0,i=0,f\_lcn=""):

fonte=fontor(f\_name,size,b,i,f\_lcn)

screen\_text=fonte.render(text,True,colour(f\_color))

win.blit(screen\_text,[x-screen\_text.get\_rect().width/2,y-screen\_text.get\_rect().height/2])

def button(x,y,width,height,inactive,active,action=None,parameter=None):

x=x-width/2

y=y-height/2

cur=pygame.mouse.get\_pos()

click=pygame.mouse.get\_pressed()

if x<cur[0]<x+width and y<cur[1]<y+height:

if parameter==None:

pygame.draw.rect(win,colour(inactive),(x+3,y+3,width,height))

pygame.draw.rect(win,colour(active),(x-3,y-3,width,height))

if click[0]==1:

if click[0]==1 and action!=None:

if action=="quit":

1/0

elif action=="controls":

return "controls"

elif action=="play":

return "play"

elif action=="home":

return "home"

elif action=="scores":

return 'scores'

elif action=='next' or 'back':

return True

else:

pass

else:

pygame.draw.rect(win,colour(active),(x,y,width,height))

return None

def button1(i,x,y,parameter=None,t=0):

l\_tank=['Prime','Demon','Dragon','Typhoon','Colossus']

l\_turret=['RZ156','VC918','CQ798','WB112','TZ553']

cur=pygame.mouse.get\_pos()

click=pygame.mouse.get\_pressed()

if t==0:

rect=tank\_turret[i].get\_rect()

x=x-rect.width/2

y=y-rect.height/2

win.blit(tank\_turret[i],(x,y))

display.msg\_2\_screen(l\_turret[i],'agencyfb',colour('black'),x-d\_width//2+tank\_turret[i].get\_rect().width\*70//200,y-d\_height//2+tank\_turret[i].get\_rect().height\*50//30,tank\_turret[i].get\_rect().height)

if x<cur[0]<x+rect.width and y<cur[1]<y+rect.height:

pygame.draw.rect(win,colour('grey'),(x-5,y-5,rect.width+10,rect.height+10),2)

if click[0]:

return parameter

elif t==1:

rect=tank\_body[i].get\_rect()

x=x-rect.width/2

y=y-rect.height/2

win.blit(tank\_body[i],(x,y))

display.msg\_2\_screen(l\_tank[i],'agencyfb',colour('black'),x-d\_width//2+tank\_body[i].get\_rect().width\*80//140,y-d\_height//2+tank\_body[i].get\_rect().height\*3//2,tank\_body[i].get\_rect().height\*3//4)

if x<cur[0]<x+rect.width and y<cur[1]<y+rect.height:

pygame.draw.rect(win,colour('grey'),(x-5,y-5,rect.width+10,rect.height+10),2)

if click[0]:

return parameter

class player(object):

def \_\_init\_\_(self,x,y,angle):

self.x = x

self.y = y

self.angle=angle

self.power=100

self.health=100

def draw(self,win,itur,ibod,x\_barrier):

self.hitbox=(self.x-2,self.y-20,4+tank\_body[4].get\_rect().width\*9//14,tank\_body[4].get\_rect().height\*3//2)

Im=pygame.transform.scale(tank\_turret[itur],(tank\_turret[itur].get\_rect().width//2,tank\_turret[itur].get\_rect().height\*4//5))

rot\_tank\_turret=pygame.transform.rotate(Im,self.angle)

turret\_rect\_center=rot\_tank\_turret.get\_rect().center

pygame.draw.circle(win,(80,80,14),(int(self.x+Im.get\_rect().center[0]),int(self.y)),Im.get\_rect().width\*18//100)

display.msg\_2\_screen("POWER:"+str(self.power)+"%",'AgencyFB',colour('black'),0,-d\_height//3,40)

display.msg\_2\_screen("ANGLE:"+str(-self.angle),'AgencyFB',colour('black'),0,-d\_height\*4//15,40)

win.blit(rot\_tank\_turret,(self.x+Im.get\_rect().center[0]-turret\_rect\_center[0],self.y-5-turret\_rect\_center[1]))

win.blit(pygame.transform.scale(tank\_body[ibod],(int(tank\_body[4].get\_rect().width\*9/14),int(tank\_body[4].get\_rect().height\*0.8))),(self.x,self.y))

pygame.draw.rect(win,colour('d\_red'),(d\_width\*5//8,d\_height//6,d\_width//4,d\_height//20))

pygame.draw.rect(win,colour('green'),(d\_width\*5//8,d\_height//6,self.health\*2\*d\_width//800,d\_height//20))

key=pygame.key.get\_pressed()

if key[pygame.K\_LEFT]==1:

self.x-=3

elif key[pygame.K\_RIGHT]==1:

self.x+=3

elif key[pygame.K\_UP]==1:

self.angle-=1

elif key[pygame.K\_DOWN]==1:

self.angle+=1

elif key[pygame.K\_a]==1:

self.power-=1

elif key[pygame.K\_d]==1:

self.power+=1

if self.x<=x\_barrier+20:

self.x+=3

elif self.x+90>d\_width:

self.x-=3

if self.angle>5:

self.angle-=1

elif self.angle<-90:

self.angle+=1

if self.power==0:

self.power+=1

if self.power==101:

self.power-=1

def fire(self,x\_barrier,y\_barrier,itur,ibod):

Im=pygame.transform.flip(pygame.transform.scale(tank\_turret[itur],(100,24)),1,0)

fire1=pygame.image.load('firecircle1.png')

g=10

v=m.sqrt(10\*d\_width)\*self.power/100

v\_x=v\*m.cos(-self.angle\*m.pi/180)

v\_y=v\*m.sin(-self.angle\*m.pi/180)

x=self.x+Im.get\_rect().center[0]-Im.get\_rect().center[0]\*m.cos(self.angle\*m.pi/180)

y=self.y+Im.get\_rect().center[0]\*m.sin(self.angle\*m.pi/180)-2

fire=True

display.msg\_2\_screen("POWER:"+str(self.power)+"%",'AgencyFB',colour('black'),0,-d\_height//3,40)

display.msg\_2\_screen("ANGLE:"+str(-self.angle),'AgencyFB',colour('black'),0,-d\_height\*4//15,40)

self.draw(win,itur,ibod,x\_barrier)

enemy\_tank.draw(win,x\_barrier)

c=0

while fire:

x=x-v\_x/10

v\_y=v\_y-g/10

y=y-v\_y/10

win.blit(pygame.transform.scale(fire1,(6,6)),(int(x)-3,int(y)-3))

clock.tick(fps\*10)

pygame.display.update()

l=enemy\_tank.hitbox

if l[0]+l[2]>x>l[0] and l[1]+l[3]>y>l[1]:

c+=1

if c==2:

win.blit(pygame.transform.scale(fire1,(60,60)),(x-30,y-30))

pygame.display.update()

fire=False

enemy\_tank.health-=5

if enemy\_tank.health==0:

for i in range(15):

time.sleep(0.03)

win.blit(bg,(0,0))

self.draw(win,itur,ibod,x\_barrier)

if i>12:

pass

else:

enemy\_tank.draw(win,x\_barrier)

pygame.draw.rect(win,colour('d\_red'),(d\_width//8,d\_height//6,d\_width//4,d\_height//20))

barriers(x\_barrier,y\_barrier)

win.blit(dg,(0,d\_height-dg.get\_rect().height))

win.blit(exp[i],(enemy\_tank.x-100,enemy\_tank.y-250))

pygame.display.update()

k=0

while k<100000:

k+=1

time.sleep(0.4)

if y>d\_height\*55//60-y\_barrier and x\_barrier<x<x\_barrier+20:

win.blit(pygame.transform.scale(fire1,(60,60)),(x-30,y-30))

pygame.display.update()

time.sleep(0.4)

fire=False

if y>d\_height\*54//60:

win.blit(pygame.transform.scale(fire1,(60,60)),(x-30,y-30))

pygame.display.update()

time.sleep(0.4)

fire=False

class enemy(object):

def \_\_init\_\_(self,x,y,angle):

self.x = 100

self.y = y

self.angle=angle

self.health=100

self.hit=0

def rand(self):

self.shift=random.randint(-d\_width//10,d\_width//10)

self.angle\_shift=random.randint(-3,3)

def mind(self):

R=tank.x+self.Im.get\_rect().center[0]-(self.x+self.Im.get\_rect().center[0]+self.Im.get\_rect().center[0]\*m.cos(self.angle\*m.pi/180))

self.exp\_angle=(m.pi/2)-0.5\*(m.asin(R\*10/(self.v)\*\*2))

self.exp\_angle=self.exp\_angle\*180/m.pi

def draw(self,win,x\_barrier):

self.hitbox=(self.x-2,self.y-20,4+tank\_body[4].get\_rect().width\*9//14,tank\_body[4].get\_rect().height\*3//2)

self.Im=pygame.transform.flip(pygame.transform.scale(tank\_turret[4],(tank\_turret[itur].get\_rect().width//2,tank\_turret[itur].get\_rect().height\*4//5)),1,0)

rot\_tank\_turret=pygame.transform.rotate(self.Im,-self.angle)

turret\_rect\_center=rot\_tank\_turret.get\_rect().center

pygame.draw.circle(win,(80,80,14),(int(self.x+self.Im.get\_rect().center[0]\*0.9),int(self.y)),18)

win.blit(rot\_tank\_turret,(self.x+self.Im.get\_rect().center[0]-turret\_rect\_center[0],self.y-5-turret\_rect\_center[1]))

win.blit(pygame.transform.flip(pygame.transform.scale(tank\_body[4],(int(tank\_body[4].get\_rect().width\*9/14),int(tank\_body[4].get\_rect().height\*0.8))),1,0),(self.x,self.y))

pygame.draw.rect(win,colour('d\_red'),(d\_width//8,d\_height//6,d\_width//4,d\_height//20))

pygame.draw.rect(win,colour('green'),(d\_width//8,d\_height//6,self.health\*2\*d\_width//800,d\_height//20))

if self.angle<-90:

self.angle+=1

elif self.angle>5:

self.angle-=1

if self.x<0:

self.x+=3

elif self.x+100>x\_barrier:

self.x-=3

def fire(self,x\_barrier,y\_barrier):

fire1=pygame.image.load('firecircle1.png')

g=10

self.v=m.sqrt(10\*d\_width)

self.rand()

if self.x<=10:

self.shift=abs(self.shift)

elif self.x>=x\_barrier-10:

self.shift=-abs(self.shift)

else:

pass

while self.shift!=0:

if self.x<0:

break

clock.tick(24)

if self.shift<0:

self.x-=3

self.shift+=1

else:

self.x+=3

self.shift-=1

win.blit(bg,(0,0))

self.draw(win,x\_barrier)

tank.draw(win,itur,ibod,x\_barrier)

barriers(x\_barrier,y\_barrier)

win.blit(dg,(0,d\_height-dg.get\_rect().height))

pygame.display.update()

self.mind()

t=int(-self.exp\_angle)+self.angle\_shift-self.angle

while t!=0:

if t<0:

self.angle-=1

t+=1

else:

self.angle+=1

t-=1

win.blit(bg,(0,0))

self.draw(win,x\_barrier)

tank.draw(win,itur,ibod,x\_barrier)

barriers(x\_barrier,y\_barrier)

win.blit(dg,(0,d\_height-dg.get\_rect().height))

pygame.display.update()

v\_x=self.v\*m.cos(-self.angle\*m.pi/180)

v\_y=self.v\*m.sin(-self.angle\*m.pi/180)

x=self.x+self.Im.get\_rect().center[0]+self.Im.get\_rect().center[0]\*m.cos(self.angle\*m.pi/180)

y=self.y+self.Im.get\_rect().center[0]\*m.sin(self.angle\*m.pi/180)-2

win.blit(bg,(0,0))

self.draw(win,x\_barrier)

tank.draw(win,itur,ibod,x\_barrier)

barriers(x\_barrier,y\_barrier)

win.blit(dg,(0,d\_height-dg.get\_rect().height))

fire=True

c=0

while fire:

win.blit(pygame.transform.scale(fire1,(6,6)),(int(x)-3,int(y)-3))

x=x+v\_x/10

v\_y=v\_y-g/10

y=y-v\_y/10

clock.tick(fps\*10)

pygame.display.update()

l=tank.hitbox

if l[0]+l[2]>x>l[0] and l[1]+l[3]>y>l[1]:

c+=1

if c==2:

fire=False

tank.health-=5

self.hit+=1

if tank.health>0:

win.blit(pygame.transform.scale(fire1,(60,60)),(x-30,y-30))

pygame.display.update()

if tank.health==0:

for i in range(15):

time.sleep(0.03)

win.blit(bg,(0,0))

self.draw(win,x\_barrier)

if i>12:

pass

else:

tank.draw(win,itur,ibod,x\_barrier)

display.msg\_2\_screen("POWER:"+str(tank.power)+"%",'AgencyFB',colour('black'),0,-200,40)

display.msg\_2\_screen("ANGLE:"+str(-tank.angle),'AgencyFB',colour('black'),0,-160,40)

pygame.draw.rect(win,colour('d\_red'),(d\_width\*5//8,d\_height//6,d\_width//4,d\_height//20))

barriers(x\_barrier,y\_barrier)

win.blit(dg,(0,d\_height-dg.get\_rect().height))

win.blit(exp[i],(tank.x-100,tank.y-250))

pygame.display.update()

k=0

while k<100000:

k+=1

time.sleep(0.4)

if y>d\_height\*55//60-y\_barrier and x\_barrier<x<x\_barrier+20:

win.blit(pygame.transform.scale(fire1,(60,60)),(x-30,y-30))

pygame.display.update()

time.sleep(0.25)

fire=False

if y>d\_height\*54//60:

win.blit(pygame.transform.scale(fire1,(60,60)),(x-30,y-30))

pygame.display.update()

time.sleep(0.25)

fire=False

######################################################ALL THE FUNCTIONS

def colour(colour):

d\_clr={"black":(0,0,0),

"white":(255,255,255),

"purple":(128,0,128),

"green":(0,255,0),

"cyan":(0,255,255),

"yellow":(255,255,0),

"magenta":(255,0,255),

"red":(235,0,0),

"b\_green":(127,255,0),

"lavender":(230,230,255),

"orange":(255,127,80),

"grey":(150,150,150),

"blue":(0,0,255),

"d\_red":(175,0,0),

"d\_green":(0,150,0),

"d\_yellow":(175,175,0),

"d\_blue":(0,0,160),

"d\_grey":(100,100,100)}

if type(colour)==tuple:

return colour

clr\_code=d\_clr[colour]

return clr\_code

def fontor(f\_name="centurygothic",size=25,b=False,i=False,f\_lcn=""):

global font

if len(f\_lcn)==0:

font=pygame.font.SysFont(str(f\_name),size,b,i)

else:

f\_lcn=f\_lcn+"\\"+f\_name

font=pygame.font.Font(f\_lcn,size,b,i)

return font

def pause():

i=1

while i>0:

display.msg\_2\_screen('PAUSED','arial',colour('black'),0,0,100,True)

pygame.display.update()

for event in pygame.event.get():

if event.type==pygame.QUIT:

pygame.display.quit()

if event.type==pygame.KEYDOWN:

i=-1

def game\_intro():

global intro,cmd

intro=True

cmd=None

while intro:

win.fill(colour("lavender"))

display.msg\_2\_screen("TANKS!","calibri","d\_green",0,-d\_height\*3//10,d\_height//5,True,True)

display.msg\_2\_screen("HS14","calibri","black",d\_width\*2//5,int(d\_height\*0.9),30,True,True)

tank=pygame.transform.scale(pygame.image.load('tank.png'),(d\_width\*3//4,d\_height//2))

win.blit(tank,((d\_width-tank.get\_rect().width)//2,(d\_height-tank.get\_rect().height)//2))

cmd=display.button(int(d\_width/2),int(0.8\*d\_height),d\_width/5,d\_height/10,"d\_red","red","controls")

if cmd==None:

cmd=display.button(int(d\_width/4),int(0.8\*d\_height),d\_width\*3/20,d\_height/10,"d\_green","green","play")

display.button(int(d\_width\*3/4),int(0.8\*d\_height),d\_width\*3/20,d\_height/10,"grey","d\_grey","quit")

display.text\_2\_button(int(d\_width/4),int(d\_height\*0.8),"PLAY","SakkalMajalla","purple",int(d\_width/20),b=True)

display.text\_2\_button(int(d\_width/2),int(0.8\*d\_height),"CONTROLS","SakkalMajalla","orange",int(d\_width/20),b=True)

display.text\_2\_button(int(d\_width\*3/4),int(0.8\*d\_height),"QUIT","SakkalMajalla","black",int(d\_width/20),b=True)

for event in pygame.event.get():

if event.type==pygame.QUIT:

1/0

if event.type==pygame.KEYDOWN:

if event.key==pygame.K\_q:

1/0

else:

pass

if cmd!=None:

intro=False

pygame.display.update()

def controls():

global g\_cont,cmd

g\_cont=True

cmd=None

while g\_cont:

win.fill(colour('lavender'))

display.msg\_2\_screen("Controls",'AgencyFB',colour('red'),0,-d\_height//5,30\*d\_height//300)

display.msg\_2\_screen("Pause:P",'calibri',colour('black'),0,-d\_height//20,30\*d\_height//600)

display.msg\_2\_screen("Move Turret:Up and Down key",'calibri',colour('black'),0,0,30\*d\_height//600)

display.msg\_2\_screen("Move Tank: Left and Right key",'calibri',colour('black'),0,d\_height//20,30\*d\_height//600)

display.msg\_2\_screen("Alter Power: A and D key",'calibri',colour('black'),0,d\_height//10,30\*d\_height//600)

display.msg\_2\_screen("Shoot:SPACE",'calibri',colour('black'),0,d\_height\*3//20,30\*d\_height//600)

cmd=display.button(int(d\_width/4),int(0.8\*d\_height),d\_width\*3/20,d\_height/10,colour('d\_green'),colour('green'),"play")

display.button(int(3\*d\_width/4),int(0.8\*d\_height),d\_width\*3/20,d\_height/10,colour('d\_red'),colour('red'),"quit")

display.text\_2\_button(int(d\_width/4),int(0.8\*d\_height),"PLAY","SakkalMajalla",colour("purple"),size=40,b=True)

display.text\_2\_button(int(3\*d\_width/4),int(0.8\*d\_height),"QUIT","SakkalMajalla",colour("black"),size=40,b=True)

pygame.display.update()

for event in pygame.event.get():

if event.type==pygame.QUIT:

pygame.display.quit()

if event.type==pygame.KEYDOWN:

if event.key==pygame.K\_q:

1/0

else:

pass

pygame.display.update()

if cmd!=None:

g\_cont=False

def custom():

tur=True

bod=True

l=[(d\_width//2,d\_height//2),(d\_width//4,d\_height//3),(d\_width\*3//4,d\_height//3),(d\_width//4,d\_height\*2//3),(d\_width\*3//4,d\_height\*2//3)]

while tur:

win.fill(colour('lavender'))

for event in pygame.event.get():

if event.type==pygame.QUIT:

pygame.display.quit()

if event.type==pygame.KEYDOWN:

if event.key==pygame.K\_q:

1/0

else:

pass

display.msg\_2\_screen('CUSTOMIZE','AgencyFB',colour('purple'),0,-d\_height\*2//5,d\_width//10,b=True)

display.msg\_2\_screen('Choose Turret','AgencyFB',colour('red'),0,-d\_height\*3//10,d\_width//20)

i0=display.button1(0,l[0][0],l[0][1],0,0)

i1=display.button1(1,l[1][0],l[1][1],1,0)

i2=display.button1(2,l[2][0],l[2][1],2,0)

i3=display.button1(3,l[3][0],l[3][1],3,0)

i4=display.button1(4,l[4][0],l[4][1],4,0)

if i0!=None or i1!=None or i2!=None or i3!=None or i4!=None:

tur=False

pygame.display.update()

l1=[i0,i1,i2,i3,i4]

time.sleep(0.15)

for me in l1:

if me!=None:

i\_turret=me

while bod:

win.fill(colour('lavender'))

for event in pygame.event.get():

if event.type==pygame.QUIT:

pygame.display.quit()

if event.type==pygame.KEYDOWN:

if event.key==pygame.K\_q:

1/0

else:

pass

display.msg\_2\_screen('CUSTOMIZE','AgencyFB',colour('purple'),0,-d\_height\*2//5,d\_width//10,b=True)

display.msg\_2\_screen('Choose Tank Body','AgencyFB',colour('red'),0,-d\_height\*3//10,d\_width//20)

i0=display.button1(0,l[0][0],l[0][1],0,1)

i1=display.button1(1,l[1][0],l[1][1],1,1)

i2=display.button1(2,l[2][0],l[2][1],2,1)

i3=display.button1(3,l[3][0],l[3][1],3,1)

i4=display.button1(4,l[4][0],l[4][1],4,1)

if i0!=None or i1!=None or i2!=None or i3!=None or i4!=None:

bod=False

pygame.display.update()

l1=[i0,i1,i2,i3,i4]

for me in l1:

if me!=None:

i\_body=me

return i\_turret,i\_body

wall=pygame.transform.scale(pygame.image.load('wall.jpg'),(d\_width//4,d\_height//2))

def barriers(x,y):

win.blit(wall,(x,(d\_height\*55//60)-y),(0,0,20,y))

def play(itur,ibod):

p\_cont=True

x\_barrier=random.randint(-d\_width//20+d\_width//2,d\_width//20+d\_width//2)

y\_barrier=random.randint(d\_height//6,d\_height//2)

while p\_cont:

win.blit(bg,(0,0))

if tank.health>0:

tank.draw(win,itur,ibod,x\_barrier)

else:

pygame.draw.rect(win,colour('d\_red'),(d\_width\*5//8,d\_height//6,d\_width//4,d\_height//20))

display.msg\_2\_screen("POWER:"+str(tank.power)+"%",'AgencyFB',colour('black'),0,-d\_height//3,40)

display.msg\_2\_screen("ANGLE:"+str(-tank.angle),'AgencyFB',colour('black'),0,-d\_height\*4//15,40)

p\_cont=False

if enemy\_tank.health>0:

enemy\_tank.draw(win,x\_barrier)

else:

pygame.draw.rect(win,colour('d\_red'),(d\_width//8,d\_height//6,d\_width//4,d\_height//20))

p\_cont=False

barriers(x\_barrier,y\_barrier)

win.blit(dg,(0,d\_height-dg.get\_rect().height))

pygame.display.update()

for event in pygame.event.get():

if event.type==pygame.QUIT:

1/0

if event.type==pygame.KEYDOWN:

if event.key==pygame.K\_q:

1/0

elif event.key==pygame.K\_SPACE:

if tank.health>0:

tank.fire(x\_barrier,y\_barrier,itur,ibod)

if enemy\_tank.health>0:

enemy\_tank.fire(x\_barrier,y\_barrier)

elif event.key==pygame.K\_p:

pause()

else:

pass

clock.tick(fps)

def end():

time.sleep(1)

for i in range(d\_width//2):

win.blit(door,(i-d\_width//2+2,0),(0,0,d\_width//2,d\_height))

win.blit(door,(d\_width-i,0),(d\_width//2,0,d\_width//2,d\_height))

pygame.display.update()

if i<d\_width\*2//3:

clock.tick(fps\*4)

else:

clock.tick(fps\*2)

if enemy\_tank.health==0:

display.msg\_2\_screen('YOU WON!','battlestar',colour('d\_red'),30,0,120)

else:

display.msg\_2\_screen('YOU LOST!','battlestar',colour('d\_red'),30,0,120)

pygame.display.update()

time.sleep(2)

win.blit(door,(0,0))

pygame.display.update()

time.sleep(0.5)

for i in range(d\_width//2,0,-1):

win.blit(door,(i-d\_width//2,0),(0,0,d\_width//2,d\_height))

win.blit(door,(d\_width-i,0),(d\_width//2,0,d\_width//2,d\_height))

pygame.display.update()

clock.tick(fps\*4)

def aftergame():

aft=True

cmd=None

while aft:

win.fill(colour('lavender'))

display.msg\_2\_screen('WELL PLAYED','agencyfb',colour('d\_grey'),0,-d\_height//3,80,True)

display.msg\_2\_screen('What would you like to do?','timesnewroman',colour('d\_red'),0,-d\_height//12,50)

cmd=display.button(int(d\_width/4),int(0.7\*d\_height),d\_width\*3/20,d\_height/10,"d\_green","green","home")

if cmd==None:

cmd=display.button(int(d\_width/2),int(0.7\*d\_height),d\_width/5,d\_height/10,"d\_red","red","scores")

display.button(int(d\_width\*3/4),int(0.7\*d\_height),d\_width\*3/20,d\_height/10,"grey","d\_grey","quit")

display.text\_2\_button(int(d\_width/4),int(d\_height\*0.7),"HOME","SakkalMajalla","purple",int(d\_width/20),b=True)

display.text\_2\_button(int(d\_width/2),int(0.7\*d\_height),"SCORES","SakkalMajalla","orange",int(d\_width/20),b=True)

display.text\_2\_button(int(d\_width\*3/4),int(0.7\*d\_height),"QUIT","SakkalMajalla","black",int(d\_width/20),b=True)

if cmd!=None:

aft=False

for event in pygame.event.get():

if event.type==pygame.QUIT:

1/0

if event.type==pygame.KEYDOWN:

if event.key==pygame.K\_q:

1/0

else:

pass

pygame.display.update()

return cmd

def enter\_score():

win.fill(colour('lavender'))

display.msg\_2\_screen('ENTER SCORE','agencyfb',colour('purple'),0,-200,90,True)

display.msg\_2\_screen('NAME','calibri',colour('d\_blue'),-180,-40,60)

display.msg\_2\_screen('HEALTH:','calibri',colour('d\_blue'),-180,30,60)

display.msg\_2\_screen('HITS:','calibri',colour('d\_blue'),-180,100,60)

display.msg\_2\_screen(str(tank.health),'calibri',colour('d\_green'),0,30,60)

display.msg\_2\_screen(str(enemy\_tank.hit),'calibri',colour('d\_green'),0,100,60)

done=False

clr=(240,240,240)

name=''

while not done:

cur=pygame.mouse.get\_pos()

click=pygame.mouse.get\_pressed()

pygame.draw.rect(win,clr,(320,235,200,40))

for event in pygame.event.get():

if event.type==pygame.QUIT:

1/0

if 320<cur[0]<520 and 235<cur[1]<275:

if click[0]==1:

clr=colour('white')

if event.type==pygame.KEYDOWN and clr==colour('white'):

if event.key==8:

name=name[:-1]

elif len(name)<10:

name+=event.unicode

else:

pass

if event.type==pygame.KEYDOWN and clr!=colour('white'):

if event.unicode=='q':

1/0

if clr!=colour('white'):

display.msg\_2\_screen('Click Here','calibri',(180,180,180),20,-45,35)

display.msg\_2\_screen(name,'calibri',colour('black'),20,-45,35)

done=display.button(int(d\_width\*3/4),int(0.8\*d\_height),d\_width\*3/20,d\_height/10,"green","d\_green","next")

display.text\_2\_button(int(d\_width\*3/4),int(0.8\*d\_height),'NEXT','agencyfb',colour('d\_red'),40)

pygame.display.update()

con,cur=scores\_init(name)

return con,cur

def scores\_init(name):

con=sq.connect('scores.db')

cur=con.cursor()

try:

cur.execute('create table score(NAME text,HITS integer,TANKHEALTH integer)')

con.commit()

except:

pass

finally:

if tank.health>0:

cur.execute('insert into score values(\'{}\',\'{}\',\'{}\')'.format(name,enemy\_tank.hit,tank.health))

print('done')

con.commit()

cur.execute('select hits from score')

r=cur.fetchall()

ar=copy.copy(r)

r.sort()

if len(r)>5:

r=r[:4]

for i in range(len(ar)):

if ar[i] in r:

pass

else:

cur.execute('delete from score where hits=%s'%ar[i])

con.commit()

con.commit()

return con,cur

def scores(con,cur):

cur.execute('select \* from score order by hits,tankhealth desc')

record=cur.fetchall()

win.fill(colour('lavender'))

display.msg\_2\_screen('HIGH SCORES','agencyfb',colour('purple'),0,-250,80)

done=False

g\_coord=[]

for j in range(100,400,50):

l=[]

for i in (200,400,500):

l.append((i,j))

g\_coord.append(l)

for j in range(100,401,50):

pygame.draw.line(win,colour('black'),(200,j),(600,j),5)

for i in (200,400,500,600):

pygame.draw.line(win,colour('black'),(i,100),(i,400),5)

pygame.display.update()

head\_in=g\_coord[0]

g\_coord=g\_coord[1:]

display.msg\_2\_screen('NAME','agencyfb',colour('black'),head\_in[0][0]-d\_width//2+100,head\_in[0][1]-d\_height//2+25,40)

display.msg\_2\_screen('HITS','agencyfb',colour('black'),head\_in[1][0]-d\_width//2+50,head\_in[1][1]-d\_height//2+25,40)

display.msg\_2\_screen('HEALTH','agencyfb',colour('black'),head\_in[2][0]-d\_width//2+50,head\_in[2][1]-d\_height//2+25,40)

while not done:

done=display.button(int(d\_width\*3/4),int(0.8\*d\_height),d\_width\*3/20,d\_height/10,"red","d\_red","back")

display.text\_2\_button(int(d\_width\*3/4),int(0.8\*d\_height),'BACK','calibri',colour('orange'),40)

try:

for i in range(5):

display.msg\_2\_screen(record[i][0],'calibri',colour('black'),g\_coord[i][0][0]-d\_width//2+100,g\_coord[i][0][1]-d\_height//2+25,30)

display.msg\_2\_screen(str(record[i][1]),'calibri',colour('black'),g\_coord[i][1][0]-d\_width//2+50,g\_coord[i][1][1]-d\_height//2+25,30)

display.msg\_2\_screen(str(record[i][2]),'calibri',colour('black'),g\_coord[i][2][0]-d\_width//2+50,g\_coord[i][2][1]-d\_height//2+25,30)

except:

pass

for event in pygame.event.get():

if event.type==pygame.QUIT:

1/0

pygame.display.update()

###############################################

#final gameloop

tank=player(5\*d\_width/8,int(5.08\*d\_height/6),0)

enemy\_tank=enemy(1\*d\_width/8,int(5.08\*d\_height/6),0)

try:

c=True

while c:

game\_intro()

if cmd=='controls':

controls()

if cmd=='play':

itur,ibod=custom()

tank.health=100

enemy\_tank.health=100

tank.angle=enemy\_tank.angle=0

tank.power=100

play(itur,ibod)

end()

if tank.health>0:

con,cur=enter\_score()

else:

con=sq.connect('scores.db')

cur=con.cursor()

while cmd!='home':

cmd=aftergame()

if cmd=='scores':

scores(con,cur)

except Exception as e:

print(e)

del font

pygame.display.quit()

pygame.quit()

print('end')

##############END#############

**OUTPUT SHOTS**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**FUTURE EXPANSION**

Our project has a lot of scope for future expansion. It can be further improvised by adding more features and animation to the game to make it more even more interesting.

It can be further improved by making the current games two player rather than single player.

Also better scoring category and highscore saving system can be created.

Therefore, our project holds a multitude of opportunities within itself to become even more useful and better in the near future.

**BIBLIOGRAPHY**

We referred to the following books:

* RIPINO Computer Science with Python for Class 12

We referred to the following sites and search engines:

* Google
* Stackoverflow.com
* Wikipedia.org
* Geeksforgeeks.com