**ACKNOWLEDGEMENT**

We would like to express our immense gratitude towards everyone who helped us to complete our project “Man vs ALIEN”

We are highly indebted to Ms. Neha Kaul Ma’am for her guidance and constant supervision as well as for providing necessary knowledge and information regarding this project.

Without her kind support and help we would never be able to complete such a project.

We would also like to thank our parents for their kind support and encouragement which helped us in completion of the project.

Lastly, we would like to extend our sincere thanks to all our friends who helped us by playing the game and finding out the errors.

**SOURCE CODE**

import pygame

pygame.init()

win = pygame.display.set\_mode((500,375))

pygame.display.set\_caption("First Game")

walkRight = [pygame.image.load('R1.png'), pygame.image.load('R2.png'), pygame.image.load('R3.png'), pygame.image.load('R4.png'), pygame.image.load('R5.png'), pygame.image.load('R6.png'), pygame.image.load('R7.png'), pygame.image.load('R8.png'), pygame.image.load('R9.png')]

walkLeft = [pygame.image.load('L1.png'), pygame.image.load('L2.png'), pygame.image.load('L3.png'), pygame.image.load('L4.png'), pygame.image.load('L5.png'), pygame.image.load('L6.png'), pygame.image.load('L7.png'), pygame.image.load('L8.png'), pygame.image.load('L9.png')]

bg = pygame.image.load('mariobgcs.png')

char = pygame.image.load('standing.png')

clock = pygame.time.Clock()

bulletSound = pygame.mixer.Sound('bullet.wav')

hitSound = pygame.mixer.Sound('hit.wav')

music = pygame.mixer.music.load('music.wav')

pygame.mixer.music.play(-1)

score = 0

class player(object):

def \_\_init\_\_(self,x,y,width,height):

self.x = x

self.y = y

self.width = width

self.height = height

self.vel = 5

self.isJump = False

self.left = False

self.right = False

self.walkCount = 0

self.jumpCount = 10

self.standing = True

self.hitbox = (self.x + 17, self.y + 11, 29, 52)

def draw(self, win):

if self.walkCount + 1 >= 27:

self.walkCount = 0

if not(self.standing):

if self.left:

win.blit(walkLeft[self.walkCount//3], (self.x,self.y))

self.walkCount += 1

elif self.right:

win.blit(walkRight[self.walkCount//3], (self.x,self.y))

self.walkCount +=1

else:

if self.right:

win.blit(walkRight[0], (self.x, self.y))

else:

win.blit(walkLeft[0], (self.x, self.y))

self.hitbox = (self.x + 17, self.y + 11, 29, 52)

#pygame.draw.rect(win, (255,0,0), self.hitbox,2)

def hit(self):

self.isJump = False

self.jumpCount = 10

self.x = 100

self.y = 265

self.walkCount = 0

font1 = pygame.font.SysFont('comicsans', 100)

text = font1.render('-5', 1, (255,0,0))

win.blit(text, (250 - (text.get\_width()/2),200))

pygame.display.update()

i = 0

while i < 200:

pygame.time.delay(10)

i += 1

for event in pygame.event.get():

if event.type == pygame.QUIT:

i = 201

pygame.quit()

class projectile(object):

def \_\_init\_\_(self,x,y,radius,color,facing):

self.x = x

self.y = y

self.radius = radius

self.color = color

self.facing = facing

self.vel = 8 \* facing

def draw(self,win):

pygame.draw.circle(win, self.color, (self.x,self.y), self.radius)

class enemy(object):

walkRight = [pygame.image.load('R1E.png'), pygame.image.load('R2E.png'), pygame.image.load('R3E.png'), pygame.image.load('R4E.png'), pygame.image.load('R5E.png'), pygame.image.load('R6E.png'), pygame.image.load('R7E.png'), pygame.image.load('R8E.png'), pygame.image.load('R9E.png'), pygame.image.load('R10E.png'), pygame.image.load('R11E.png')]

walkLeft = [pygame.image.load('L1E.png'), pygame.image.load('L2E.png'), pygame.image.load('L3E.png'), pygame.image.load('L4E.png'), pygame.image.load('L5E.png'), pygame.image.load('L6E.png'), pygame.image.load('L7E.png'), pygame.image.load('L8E.png'), pygame.image.load('L9E.png'), pygame.image.load('L10E.png'), pygame.image.load('L11E.png')]

def \_\_init\_\_(self, x, y, width, height, end):

self.x = x

self.y = y

self.width = width

self.height = height

self.end = end

self.path = [self.x, self.end]

self.walkCount = 0

self.vel = 3

self.hitbox = (self.x + 17, self.y + 2, 31, 57)

self.health = 10

self.visible = True

def draw(self,win):

self.move()

if self.visible:

if self.walkCount + 1 >= 33:

self.walkCount = 0

if self.vel > 0:

win.blit(self.walkRight[self.walkCount //3], (self.x, self.y))

self.walkCount += 1

else:

win.blit(self.walkLeft[self.walkCount //3], (self.x, self.y))

self.walkCount += 1

pygame.draw.rect(win, (255,0,0), (self.hitbox[0], self.hitbox[1] - 20, 50, 10))

pygame.draw.rect(win, (0,128,0), (self.hitbox[0], self.hitbox[1] - 20, 50 - (5 \* (10 - self.health)), 10))

self.hitbox = (self.x + 17, self.y + 2, 31, 57)

#pygame.draw.rect(win, (255,0,0), self.hitbox,2)

def move(self):

if self.vel > 0:

if self.x + self.vel < self.path[1]:

self.x += self.vel

else:

self.vel = self.vel \* -1

self.walkCount = 0

else:

if self.x - self.vel > self.path[0]:

self.x += self.vel

else:

self.vel = self.vel \* -1

self.walkCount = 0

def hit(self):

if self.health > 0:

self.health -= 1

else:

self.visible = False

print('hit')

def redrawGameWindow():

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

text = font.render('Score: ' + str(score), 1, (0,0,0))

win.blit(text, (350, 10))

man.draw(win)

goblin.draw(win)

for bullet in bullets:

bullet.draw(win)

pygame.display.update()

#mainloop

font = pygame.font.SysFont('comicsans', 30, True)

man = player(200, 265, 64,64)

goblin = enemy(100, 265, 64, 64, 450)

shootLoop = 0

bullets = []

run = True

while run:

clock.tick(27)

if goblin.visible == True:

if man.hitbox[1] < goblin.hitbox[1] + goblin.hitbox[3] and man.hitbox[1] + man.hitbox[3] > goblin.hitbox[1]:

if man.hitbox[0] + man.hitbox[2] > goblin.hitbox[0] and man.hitbox[0] < goblin.hitbox[0] + goblin.hitbox[2]:

man.hit()

score -= 5

if shootLoop > 0:

shootLoop += 1

if shootLoop > 3:

shootLoop = 0

for event in pygame.event.get():

if event.type == pygame.QUIT:

run = False

for bullet in bullets:

if bullet.y - bullet.radius < goblin.hitbox[1] + goblin.hitbox[3] and bullet.y + bullet.radius > goblin.hitbox[1]:

if bullet.x + bullet.radius > goblin.hitbox[0] and bullet.x - bullet.radius < goblin.hitbox[0] + goblin.hitbox[2]:

hitSound.play()

goblin.hit()

score += 1

bullets.pop(bullets.index(bullet))

if bullet.x < 500 and bullet.x > 0:

bullet.x += bullet.vel

else:

bullets.pop(bullets.index(bullet))

keys = pygame.key.get\_pressed()

if keys[pygame.K\_SPACE] and shootLoop == 0:

bulletSound.play()

if man.left:

facing = -1

else:

facing = 1

if len(bullets) < 5:

bullets.append(projectile(round(man.x + man.width //2), round(man.y + man.height//2), 6, (0,0,0), facing))

shootLoop = 1

if keys[pygame.K\_LEFT] and man.x > man.vel:

man.x -= man.vel

man.left = True

man.right = False

man.standing = False

elif keys[pygame.K\_RIGHT] and man.x < 500 - man.width - man.vel:

man.x += man.vel

man.right = True

man.left = False

man.standing = False

else:

man.standing = True

man.walkCount = 0

if not(man.isJump):

if keys[pygame.K\_UP]:

man.isJump = True

man.right = False

man.left = False

man.walkCount = 0

else:

if man.jumpCount >= -10:

neg = 1

if man.jumpCount < 0:

neg = -1

man.y -= (man.jumpCount \*\* 2) \* 0.5 \* neg

man.jumpCount -= 1

else:

man.isJump = False

man.jumpCount = 10

redrawGameWindow()

pygame.quit()

**ABOUT OUR PROJECT**

Are you tired of all the new over-complex computer games?

Do you want to unwind after a long hard day?

Well, Man vs Alien is an easy and fun addictive game made just for you!

The objective is simple: shoot the alien before your score becomes negative!

And, as a bonus, you can jump around, listen to entertaining music and have fun!

So what are you waiting for?

Give it a try, get a high score and beat your friends!!

**LIMITATIONS**

* Does not exit automatically
* Only one alien
* Alien does not respawn
* We were unable to fix some small bugs in the game

**BIBLIOGRAPHY**

* youtube.com
* stackoverflow.com
* codecademy.com
* google.com

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **TOPIC** | **PAGE** |
| Acknowledgement |  |
| Source Code |  |
| About our project |  |
| Limitations |  |
| Bibliography |  |