

José Luis Madrigal A01745419

César Emiliano Palome A01746493

Jorge Isidro Blanco A01745907

Alejandra Cancio A01745339

Proyecto Final Programación Avanzada

Primero que nada, este proyecto es un videojuego personalizado en Pygame, el cual está basado en un tutorial de Julian Meyer. La temática de nuestro juego es de fútbol, y se basa principalmente en que unos delanteros atacan las porterías y tú debes detenerlos, dándoles un balonazo. Pero lo que es verdaderamente interesante de la jugabilidad es que se tendrá un valor inicial de puntos que es cero, y cada vez que un villano toque una de tus tres porterías, se te restará un punto, además de que cuando le des a alguien con un balonazo, sumarán un punto. Por lo tanto, se tendrá un valor máximo de goles (5) que te hará ganar y un valor mínimo (-5) que te hará perder. Asimismo, se tendrá un tiempo en el juego, el cual es de tres minutos y medio, y cuando se acabe, entonces también perderás. Además de la jugabilidad, se tienen los personajes, el fondo y los sonidos, que le dan un toque único al juego. Las porterías, el jugador y los villanos son iconos sacados de la página Flaticon. En cuanto a los sonidos, cada vez que el jugador lanza un balón se oye el icónico “SIU” de Cristiano Ronaldo, además de que cada vez que llegan a tu portería se oye la palabra “Golazo” redactada por el Perro Bermúdez. Por último, la canción de fondo es “Wavin Flag”, la cual es para muchos una clara representación del soccer.

Cabe mencionar que elegimos esta temática porque la verdad nos gusta mucho el fútbol, ya que nos parece un deporte entretenido y desafiante. Por otro lado, los personajes y sonidos fueron elegidos ya que nos parecía que era importante que el videojuego pudiera transmitir lo que se puede encontrar en un partido de fútbol, por lo que nuestros principales consumidores

serían los fanáticos del deporte, y para ser más específicos, los de México, ya que en este país el soccer es muy reconocido y popular. También decidimos utilizar la jugabilidad de sumar y restar puntos para que el juego no fuera tedioso y se volviera realmente retador, ya que buscamos que las personas desarrollen habilidades de rapidez, concentración y reflejos.

Uno de los problemas que tuvimos al realizar el código fue encontrar las imágenes correctas para los personajes y objetos, ya que se requería que no tuvieran fondo y también debían tener un número específico de píxeles. Asimismo, hacer que los villanos quiten puntos y que los balonazos sumen no fue tan complicado, pero lo que fue un tanto difícil de entender fue la manera en mostrar ese contador en la pantalla, para que así el jugador pudiera ver lo cerca que está de ganar o perder. Pero al final pudimos arreglarlo porque nos dedicamos a entender cómo se ponía el tiempo en la pantalla, lo cual fue mostrador en el tutorial.

Para finalizar, este proyecto nos gustó mucho ya que aprendimos nuevas funciones y comandos en Pygame. Además de que pudimos explotar un poco más nuestra creatividad para crear algo que pudiera entretener a otros. Asimismo, fue bastante interesante saber la manera en la que se realiza un videojuego, ya que habíamos jugado varios, pero nunca pensábamos que el código detrás conlleva mucha lógica y trabajo.

Código

```
#1 Import library
import pygame
from pygame.locals import *
import math
import random

#2 Initialize the game
pygame.init()
width, height = 640, 480
screen=pygame.display.set_mode((width, height))
keys= [False, False, False, False]
playerpos=[100,100]
acc=[0,0]
balls=[]
badtimer=100
badtimer1=0
badguys=[[640,100]]
goals=0
pygame.mixer.init()

#3 Load images
player= pygame.image.load("resources/images/football-player.png")
grass = pygame.image.load("resources/images/grass2.png")
porter = pygame.image.load("resources/images/goalkeeper.png")
ball = pygame.image.load("resources/images/ball.png")
badguyimg1 = pygame.image.load("resources/images/villian.png")
badguyimg=badguyimg1
gameover = pygame.image.load("resources/images/gameover.png")
youwin = pygame.image.load("resources/images/youwin.png")
# 3.1 - Load audio
hit = pygame.mixer.Sound("resources/audio/azoo.wav")
enemy = pygame.mixer.Sound("resources/audio/enemy.wav")
shoot = pygame.mixer.Sound("resources/audio/siu.wav")
hit.set_volume(3.5)
enemy.set_volume(0.45)
shoot.set_volume(0.12)
pygame.mixer.music.load('resources/audio/wavinflag.mp3')
pygame.mixer.music.play(-1, 0.0)
pygame.mixer.music.set_volume(0.2)

#4 Keep looping through
running = 1
exitcode = 0
```

```

while running:
    badtimer-=1
    #5 clear the screen before drawing it again
    screen.fill(0)
    #6 draw the screen elements
    for x in range(width/grass.get_width()+1):
        for y in range(height/grass.get_height()+1):
            screen.blit(grass,(x*100,y*100))
    screen.blit(porter,(0,30))
    screen.blit(porter,(0,188))
    screen.blit(porter,(0,345))
    #6.1 set player position and rotation
    position = pygame.mouse.get_pos()
    angle = math.atan2(position[1]-(playerpos[1]+32),position[0]-(playerpos[0]+26))
    playerrot = pygame.transform.rotate(player, 360-angle*57.29)
    playerpos1 = (playerpos[0]-playerrot.get_rect().width/2, playerpos[1]-
playerrot.get_rect().height/2)
    screen.blit(playerrot, playerpos1)
    #6.2 draw balls
    for bullet in balls:
        index+=0
        velx=math.cos(bullet[0])*10
        vely=math.sin(bullet[0])*10
        bullet[1]+=velx
        bullet[2]+=vely
        if bullet[1]<-64 or bullet[1]>640 or bullet[2]<-64 or bullet[2]>480:
            balls.pop(index)
        index+=1
    for projectile in balls:
        ball1 = pygame.transform.rotate(ball, 360-projectile[0]*57.29)
        screen.blit(ball1, (projectile[1], projectile[2]))
# 6.3 - Draw badgers
if badtimer==0:
    badguys.append([640, random.randint(50,430)])
    badtimer=600-(badtimer1*2)
    if badtimer1>=35:
        badtimer1=35
    else:
        badtimer1+=5
index=0
for badguy in badguys:
    if badguy[0]<-64:
        badguys.pop(index)
    badguy[0]-=7
# 6.3.1 - Attack porter
badrect=pygame.Rect(badguyimg.get_rect())

```

```

badrect.top=badguy[1]
badrect.left=badguy[0]
if badrect.left<64:
    hit.play()
    goals -= 1
    badguys.pop(index)
#6.3.2 - Check for collisions
index1=0
for bullet in balls:
    bullrect=pygame.Rect(ball.get_rect())
    bullrect.left=bullet[1]
    bullrect.top=bullet[2]
    if badrect.colliderect(bullrect):
        enemy.play()
        acc[0]+=1
        goals += 1
        badguys.pop(index)
        balls.pop(index1)
        index1+=1
# 6.3.3 - Next bad guy
index+=1
for badguy in badguys:
    screen.blit(badguyimg, badguy)
# 6.4 - Draw clock
font = pygame.font.Font(None, 30)
survivedtext = font.render(str((210000-pygame.time.get_ticks())/60000)+":"+str((210000-
pygame.time.get_ticks())/1000%60).zfill(2), True, (0,0,0))
textRect = survivedtext.get_rect()
textRect.topright=[635,5]
screen.blit(survivedtext, textRect)
#6.4.1 - Draw Goal Counter
font2 = pygame.font.Font(None, 80)
survivedtext2 = font2.render(str(goals), True, (0,0,0))
textRect2 = survivedtext2.get_rect()
textRect2.topright=[315,5]
screen.blit(survivedtext2, textRect2)
#7 update the screen
pygame.display.flip()
#8 loop through the events
for event in pygame.event.get():
    #check if the event is the X button
    if event.type==pygame.QUIT:
        #if it is quit the game
        pygame.quit()
        exit()
    if event.type==pygame.KEYDOWN:

```

```

    if event.key==K_w:
        keys[0]=True
    elif event.key==K_a:
        keys[1]=True
    elif event.key==K_s:
        keys[2]=True
    elif event.key==K_d:
        keys[3]=True
    if event.type==pygame.KEYUP:
        if event.key==pygame.K_w:
            keys[0]=False
        elif event.key==pygame.K_a:
            keys[1]=False
        elif event.key==pygame.K_s:
            keys[2]=False
        elif event.key==pygame.K_d:
            keys[3]=False
    if event.type==pygame.MOUSEBUTTONDOWN:
        shoot.play()
        position=pygame.mouse.get_pos()
        acc[1]+=1
        balls.append([math.atan2(position[1]-(playerpos1[1]+32),position[0]-(
(playerpos1[0]+26)),playerpos1[0]+32,playerpos1[1]+32])
#9 Move player
    if keys[0]:
        playerpos[1]-=5
    elif keys[2]:
        playerpos[1]+=5
    if keys[1]:
        playerpos[0]-=5
    elif keys[3]:
        playerpos[0]+=5
#10 - Win/Lose check
    if goals>=5:
        running=0
        exitcode=1
    if goals<=-5 or pygame.time.get_ticks()>=210000:
        running=0
        exitcode=0
    if acc[1]!=0:
        accuracy=acc[0]*1.0/acc[1]*100
    else:
        accuracy=0
# 11 - Win/lose display
    if exitcode==0:
        pygame.font.init()

```

```

font = pygame.font.Font(None, 24)
text = font.render("Accuracy: "+str(accuracy)+"%", True, (255,0,0))
textRect = text.get_rect()
textRect.centerx = screen.get_rect().centerx
textRect.centery = screen.get_rect().centery+24
screen.blit(gameover, (0,0))
screen.blit(text, textRect)
else:
    pygame.font.init()
    font = pygame.font.Font(None, 24)
    text = font.render("Accuracy: "+str(accuracy)+"%", True, (0,255,0))
    textRect = text.get_rect()
    textRect.centerx = screen.get_rect().centerx
    textRect.centery = screen.get_rect().centery+24
    screen.blit(youwin, (0,0))
    screen.blit(text, textRect)
while 1:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            exit(0)
    pygame.display.flip()

```

