Documentación para el Desarrollo del Frontend Checogame

Desarrollo de la Interfaz de Usuario

Diseño del Menú Principal

El menú principal es la primera pantalla que se muestra al iniciar el juego. Consiste en un fondo de pantalla, un botón de inicio y música de fondo.

1. **Fondo de Pantalla:**
    - Cargar la imagen de fondo.
    - Redimensionar la imagen al tamaño de la pantalla (800x600).
2. **Botón de Inicio:**
    - Cargar la imagen del botón de inicio.
    - Redimensionar el botón a 200 x 100 píxeles.
    - Centrar el botón en la parte inferior de la pantalla.
3. **Música de Fondo:**
    - Cargar y reproducir la música de fondo en bucle.

Pantalla del Juego

La pantalla del juego incluye el fondo de la pista, el jugador, los enemigos y el marcador de puntuación.

1. **Fondo de la Pista:**
    - Cargar la imagen de la pista.
    - Implementar un bucle para hacer que el fondo se mueva y se repita continuamente.
2. **Jugador:**
    - Cargar la imagen del jugador.
    - Posicionar al jugador en el centro de la pantalla en la parte inferior.
3. **Enemigos:**
    - Cargar imágenes de los enemigos desde un directorio.
    - Posicionar enemigos aleatoriamente en la pista permitida.
    - Configurar el movimiento de los enemigos hacia abajo.
4. **Marcador de Puntuación:**
    - Mostrar la puntuación del jugador en la esquina inferior izquierda de la pantalla.

Implementación de la Lógica de Presentación

Control del Jugador

El jugador se controla mediante el movimiento de la cabeza detectado por la webcam.

1. **Detección de Movimiento:**

    - Utilizar la biblioteca MediaPipe para detectar el movimiento de la cabeza.
    - Mapear el movimiento de la cabeza a los movimientos del jugador en la pantalla.
2. **Restricciones de Movimiento:**
    - Limitar el movimiento del jugador dentro de un área específica de la pista.

Generación de Enemigos

Los enemigos se generan aleatoriamente y se mueven hacia abajo en la pantalla.

1. **Generación Aleatoria:**
    - Crear enemigos en posiciones aleatorias dentro del área permitida.
    - Aumentar la velocidad de los enemigos a medida que avanza el juego.
2. **Colisiones:**
    - Detectar colisiones entre el jugador y los enemigos utilizando máscaras de colisión.
    - Finalizar el juego si ocurre una colisión.

Integración de Componentes Visuales

Video de Introducción

Reproducir un video de introducción antes de iniciar el juego.

1. **Reproducción de Video:**
    - Utilizar MoviePy para cargar y reproducir el video.
    - Redimensionar y centrar el video en la ventana del juego.

Sonido

Incorporar música de fondo y efectos de sonido.

1. **Música de Fondo:**
    - Cargar y reproducir música durante el juego.
    - Detener la música cuando el jugador pierde.
2. **Efectos de Sonido:**
    - Reproducir efectos de sonido para eventos importantes (como colisiones).

Pruebas de Usabilidad y Experiencia de Usuario

Pruebas de Usabilidad

Realizar pruebas para asegurar que el juego sea intuitivo y fácil de usar.

1. **Navegación del Menú:**
    - Verificar que el botón de inicio funcione correctamente.
    - Asegurar que la música de fondo se detenga al iniciar el juego.

2. **Control del Jugador:**
    - Probar la detección de movimiento de la cabeza y su respuesta en el juego.
    - Asegurar que el jugador no pueda moverse fuera de los límites permitidos.

Experiencia de Usuario

Evaluar la experiencia del usuario para garantizar una jugabilidad fluida y agradable.

1. **Interfaz de Usuario:**
    - Asegurar que todos los elementos de la interfaz sean visibles y funcionales.
    - Evaluar la claridad y el diseño del marcador de puntuación.
2. **Rendimiento:**
    - Verificar que el juego se ejecute sin problemas en diferentes sistemas operativos.
    - Asegurar que no haya retardos ni problemas de rendimiento.

Entregables

Código Fuente del Frontend

```
import pygame

import sys

from moviepy.editor import VideoFileClip

from pygame.locals import *

# Constants

SCREEN_WIDTH = 800

SCREEN_HEIGHT = 600

class Background(pygame.sprite.Sprite):

    def _init(self):  # Corregir la definición del método __init_

        super(Background, self)._init_()



        self.surf =
pygame.image.load(r"C:\Users\UnKnown\Documents\Proyecto_juego\pista.png")

        background_width = SCREEN_WIDTH  # doble ancho
```

```
        background_height = (background_width / self.surf.get_width()) *
self.surf.get_height()

        self.surf = pygame.transform.scale(self.surf, (background_width,
background_height))

        self.rect = self.surf.get_rect(

            bottomleft=(0, SCREEN_HEIGHT)

        )

        # Usamos 2 surface para hacer un buen loop del fondo

        # Se mueven juntos y cuando uno desaparece de pantalla se mueve

        # hasta abajo, y así se van repitiendo

        self.surf2 = self.surf

        self.rect2 = self.surf2.get_rect(

            bottomleft=self.rect.topleft

        )

        self.ypos = 0

        self.ypos2 = background_height - SCREEN_HEIGHT

    def update(self, delta_time):

        self.ypos += .05 * delta_time

        self.ypos2 += .05 * delta_time

        self.rect.y = int(self.ypos)

        self.rect2.y = int(self.ypos2)

        if self.rect.y > SCREEN_HEIGHT:

            self.ypos = self.rect2.y - self.surf2.get_height()

            self.rect.y = self.ypos
```

```
        if self.rect2.y > SCREEN_HEIGHT:
            self.ypos2 = self.rect.y - self.surf.get_height()
            self.rect2.y = self.ypos2
    def render(self, dest):
        dest.blit(self.surf, self.rect)
        dest.blit(self.surf2, self.rect2)
def main():
    # Reproducir el video con sonido antes de iniciar el juego
    play_video()
    # Mostrar el menú y esperar a que el usuario inicie el juego
    show_menu()
    # Iniciar el juego
    game_loop()
def play_video():
    pygame.display.set_caption("ChecoGame")
    # Ruta del archivo de video
    video_path = r"C:\Users\UnKnown\Documents\Proyecto_juego\f1_intro.mp4"
    # Reproducir el video con sonido
    video_clip = VideoFileClip(video_path)
    # Redimensionar el video al tamaño deseado y centrarlo en una ventana
de 800x600
    video_clip_resized = video_clip.resize((SCREEN_WIDTH, SCREEN_HEIGHT))
    video_clip_resized = video_clip_resized.set_position(('center',
'center'))
```

```
    # Reproducir el video con sonido

    video_clip_resized.preview()

def show_menu():

    pygame.init()

    screen = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))

    pygame.display.set_caption("Menú")

    # Cargar imagen de fondo del menú

    background_image =
pygame.image.load(r"C:\Users\UnKnown\Documents\Proyecto_juego\back_menu.jp
g").convert()

    background_image = pygame.transform.scale(background_image,
(SCREEN_WIDTH, SCREEN_HEIGHT))  # Redimensionar la imagen al tamaño de la
pantalla

    # Cargar imagen del botón de inicio

    start_button_image =
pygame.image.load(r"C:\Users\UnKnown\Documents\Proyecto_juego\boton.png").
convert_alpha()

    start_button_image = pygame.transform.scale(start_button_image, (200,
100))  # Redimensionar el botón

    # Inicializar el módulo de sonido de pygame

    pygame.mixer.init()

    # Cargar el archivo de audio

    background_music =
pygame.mixer.Sound(r"C:\Users\UnKnown\Documents\juego-python-ia\sprites\in
tro.wav")

    # Reproducir el audio en bucle

    background_music.play(-1)
```

```
    clock = pygame.time.Clock()

    while True:

        for event in pygame.event.get():

            if event.type == QUIT:

                pygame.quit()

                sys.exit()

            elif event.type == MOUSEBUTTONDOWN:

                # Detener la reproducción del audio cuando se hace clic en
el botón de inicio

                background_music.stop()

                # Verificar si el botón de inicio fue presionado

                if start_button_rect.collidepoint(event.pos):

                    return  # Salir de la función para iniciar el juego

        # Dibujar la imagen de fondo del menú

        screen.blit(background_image, (0, 0))

        # Obtener el rectángulo del botón de inicio

        start_button_rect = start_button_image.get_rect()

        # Centrar el botón de inicio en la parte inferior de la pantalla

        start_button_rect.center = (400, 550)

        # Dibujar el botón de inicio

        screen.blit(start_button_image, start_button_rect)

        pygame.display.flip()

        clock.tick(30)

def game_loop():
```

```
    pygame.init()

    screen = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))

    pygame.display.set_caption("ChecoGame")

    # Crear una instancia de Background

    background = Background()

    # Inicializar el módulo de sonido de pygame

    pygame.mixer.init()

    # Cargar el archivo de audio

    background_music =
pygame.mixer.Sound(r"C:\Users\UnKnown\Documents\juego-python-ia\sprites\ca
rros.wav")

    # Reproducir el audio en bucle

    background_music.play(-1)

    clock = pygame.time.Clock()

    running = True

    while running:

        for event in pygame.event.get():

            if event.type == pygame.QUIT:

                running = False



        # Actualizar el fondo

        background.update(clock.get_time())

        # Dibujar todo en la pantalla

        screen.fill((0, 0, 0))  # Llenar la pantalla con color negro
```

```
        background.render(screen)  # Dibujar el fondo

        pygame.display.flip()  # Actualizar la pantalla

        clock.tick(60)  # Mantener 60 FPS

    pygame.quit()

    sys.exit()

main()
```

Guía de Programa

Se importan los módulos necesarios:

- pygame: Librería para desarrollo de videojuegos.
- sys: Proporciona acceso a algunas variables y funciones del intérprete.
- moviepy.editor: Para manejar y reproducir videos.
- pygame.locals: Para importar constantes de Pygame.

Se definen las constantes SCREEN_WIDTH y SCREEN_HEIGHT para establecer el tamaño de la ventana del juego.

Se crea un constructor para la clase Background, que hereda de pygame.sprite.Sprite. Se detecta un error en el método _init que debería ser __init__.

Carga una imagen de fondo (pista.png) y la redimensiona para ajustarse a la pantalla del juego.

Se duplican las superficies (surf y surf2) para crear un bucle de fondo continuo. Las posiciones iniciales de ypos y ypos2 se configuran para lograr este efecto.

El método update actualiza las posiciones de las superficies para crear el efecto de desplazamiento continuo. Si una de las superficies sale de la pantalla, se reposiciona para seguir el ciclo.

El método render dibuja las dos superficies del fondo en la pantalla de destino (dest).

La función main coordina la secuencia del juego: reproduce un video, muestra un menú y luego inicia el bucle del juego.

La función play_video reproduce un video de introducción (f1_intro.mp4) antes de que el juego comience. Utiliza moviepy para manejar y redimensionar el video al tamaño de la ventana del juego.

La función show_menu inicializa Pygame y configura la ventana del menú. Carga las imágenes de fondo y del botón de inicio, además de reproducir música de fondo en bucle. Este bucle de eventos maneja la interacción del usuario con el menú. Si el usuario hace clic en el botón de inicio, la música se detiene y el bucle se rompe para iniciar el juego.

La función game_loop inicializa Pygame para el bucle principal del juego, crea una instancia de la clase Background y configura la música de fondo. El bucle principal del juego maneja eventos, actualiza el fondo, dibuja todo en la pantalla y mantiene una tasa de 60 FPS. Si el usuario cierra la ventana, el bucle termina y el juego se cierra.

Funcionalidades Personalizadas

- **Detección de Movimiento con la Webcam:** Utilización de MediaPipe para detectar el movimiento de la cabeza y mapear los movimientos del jugador.
- **Generación y Movimiento de Enemigos:** Implementación de enemigos con posiciones y velocidades aleatorias.
- **Loop de Fondo:** Implementación de un bucle de fondo para crear un efecto continuo en la pista.

Archivos de Configuración

Archivos necesarios para la configuración y ejecución de la interfaz de usuario.

1. **Configuración de Pygame:**
    - Iniciar Pygame y configurar la pantalla.
    - Inicializar los módulos de sonido y fuente de Pygame.
2. **Configuración de MediaPipe:**
    - Inicializar el modelo de MediaPipe FaceMesh para la detección de movimiento.
3. **Archivos Multimedia:**
    - Incluir rutas a los archivos de imágenes, video y audio utilizados en el juego.