PUCK-MAN

User interaction

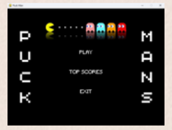
- PUCK-MAN HAS A MULTIFUNCTIONAL INTERACTION SYSTEM ,WHERE THE USER ALWAYS BE ON KNOWLEDGE OF WHAT IS HE DOING:THE MAIN WAY OF INTERACTION WILL BE THE MENUS:

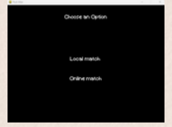
-MAIN MENU: WHERE THE USER COULD SELECT WHAT WILL HE DO HE CAN PLAY OR WATCH THE TOP SCORES OF OUR DATA BASE.
-MATCH MENU: WHERE THE USER COULD SELECT WHAT KIND OF MATCH IS HE WANNA PLAY , A LOCAL MATCH OR A ONLINE MATCH.

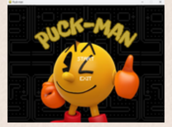
- ANOTHER WAY OF USER INTERACTION ARE THE DIFFERENTS KIND OF "SCREENS" , WHERE THE GAME USUALLY DO A CONNECTION BETWEEN MENUS OR OTHER "SCREENS":

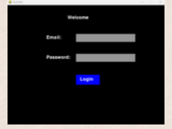
-LOG IN : THE USER REGISTER ON OUR DATA BASE TO SAVE HIM PROGRESS.
-ONLINE ALERT: THE GAME REMEMBER TO BE ONLINE TO THE USER.

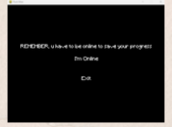
- "GAMING SCENE" IS HOW WE CALL TO THE DIFFERENT KIND OF GRAPHIC ASSITANT ON OUR GAME , LIKE VIDEOS OR IMAGES , EVEN THE SCREEN WHEN U WIN OR LOSE A MATCH.



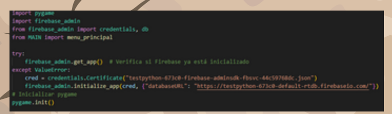




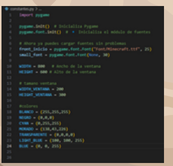



REMEMBER, u have to be online to save your progress

CODE EXPLANATION

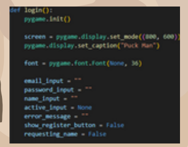


```python
import pygame
import firebase_admin
from firebase_admin import credentials, db
from MAIN import menu_principal

try:
    firebase_admin.get_app()  # Verifica si Firebase ya está inicializado
except ValueError:
    cred = credentials.Certificate("testpython-673c0-firebase-adminsdk-fbsvc-44c59768dc.json")
    firebase_admin.initialize_app(cred, {"databaseURL": "https://testpython-673c0-default-rtdb.firebaseio.com/"})
# Inicializar pygame
pygame.init()
```



constantes.py > ...

```python
import pygame

pygame.init()  # Inicializa Pygame
pygame.font.init()  # ✔ Inicializa el módulo de fuentes

# Ahora ya puedes cargar fuentes sin problemas
front_inicio = pygame.font.Font("Font/Minecraft.ttf", 25)
small_font = pygame.font.Font(None, 30)

WIDTH = 800   # Ancho de la ventana
HEIGHT = 600 # Alto de la ventana

# tamano ventana
WIDTH_VENTANA = 200
HEIGHT_VENTANA = 300

#colores
BLANCO = (255,255,255)
NEGRO = (0,0,0)
CYAN = (0,255,255)
MORADO = (138,43,226)
TRANSPARENTE = (0,0,0,0)
LIGHT_BLUE = (100, 100, 255)
BLUE = (0, 0, 255)
```



```python
def login():
    pygame.init()

    screen = pygame.display.set_mode((800, 600))
    pygame.display.set_caption("Puck Man")

    font = pygame.font.Font(None, 36)

    email_input = ""
    password_input = ""
    name_input = ""
    active_input = None
    error_message = ""
    show_register_button = False
    requesting_name = False
```

HOW WE DO?

-ALMOST OF THE USER INTERACTION,WAS MADE USING DIFFERENT KIND OF FUNCTIONS , LIKE "CONSTANTES" OR "IMPORT" AND THE MAIN LIBRARY WHO HELP US WAS "PYGAMES" , WITH THIS LIBRARY WE MADE OUR GRAPHIC SCREENS .

LOG IN AND OUR DATA BASE

-ON THE "LOG IN" SCREEN WE USE FIREBASE , IT IS OUR REAL TIME DATA BASE, AND WITH OUR CREDENTIALLS OF FIREBASE WE COULD GET ACCES TO THE DATA BASE AND MODIFICATE THE USER IMFORMATION TO CREATE OR REGISTER OUR USER.

CONSTANTES

-THIS IS OUR SCREEN BASE , WITH THIS AND THE "IMPORT" FUNCTION WE DID THE MODIFICATION AND THE WHOLE GAME, BECAUSE THIS FUNCTION HELP US TO STAY THE PARAMETERS OF THE SCREEN AND DEFINE THE DIFFERENT KIND OF KEYS TO CREATE THE GAME

WHAT IT IS?

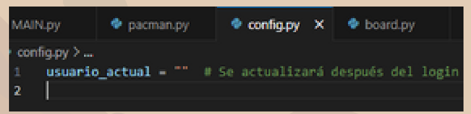
PUCK-MAN IS A MAZE ARCADE GAME , WHERE WE USE THE BEST OF TWO VIDEOGAMES AGE´S , WE USE THE ORIGINAL ARCADE INTERPHASE WITH THE NEW ONLINE MODES.

HOW WE DO ?

USING PYTHON AS THE LANGUAGE OF PROGRAMMING AND VS CODE AS TERMINAL AND CODE WRITTER, AS DATA BASE WE USE FIREBASE.

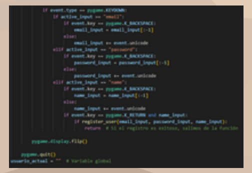
WHAT WAS THE BIGGEST PROBLEM?

WELL , IF WE TELL THE TRUTH THERE WERE ALOT OF PROBLEMS , BUT THE REAL PROBLEM ,WAS THE UPDATE OF SCORES ON OUR DATA BASE SINCE THE GAME CONSOLE , BECAUSE WHEN WE TRIED TO UPDATE USING THE FIREBASE CREDENTIALS ON THE PACMAN CODE , AT THE SAME TIME THE MAIN CODE WAS USING THIS ONE ,SO WE HAD TO USE A GOBLAL FUNCTION WHO BRING US THE SCORE AND THE USER TO ALL MOMENT.

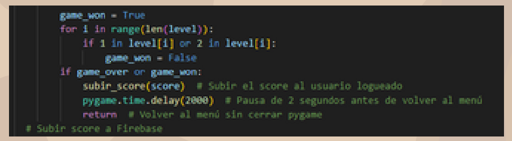


config.py > ...

```python
usuario_actual = ""  # Se actualizará después del login

```



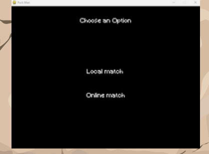
```python
                if event.type == pygame.KEYDOWN:
                    if active_input == "email":
                        if event.key == pygame.K_BACKSPACE:
                            email_input = email_input[:-1]
                        else:
                            email_input += event.unicode
                    elif active_input == "password":
                        if event.key == pygame.K_BACKSPACE:
                            password_input = password_input[:-1]
                        else:
                            password_input += event.unicode
                    elif active_input == "name":
                        if event.key == pygame.K_BACKSPACE:
                            name_input = name_input[:-1]
                        else:
                            name_input += event.unicode
                        if event.key == pygame.K_RETURN and name_input:
                            if register_user(email_input, password_input, name_input):
                                return  # Si el registro es exitoso, salimos de la función

        pygame.display.flip()

    pygame.quit()
usuario_actual = ""  # Variable global
```



```python
        game_won = True
        for i in range(len(level)):
            if 1 in level[i] or 2 in level[i]:
                game_won = False
        if game_over or game_won:
            subir_score(score)  # Subir el score al usuario logueado
            pygame.time.delay(2000)  # Pausa de 2 segundos antes de volver al menú
            return  # Volver al menú sin cerrar pygame
# Subir score a Firebase
```

What is an **API?**

A set of protocols that allow different software applications to communicate and exchange data.

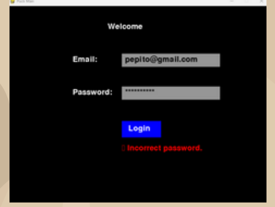


API

APPLICATION PROGRAMMING INTERFACE

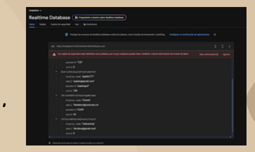
FIRST OF ALL , WE MUST TALK ABOUT WHAT KIND OF "API" ARE WE USING ON OUR GAME ;
WE CAN CLASSIFY THEY ON TWO DIFERRENT KIND OF IT .

- "API" LOCAL :WHERE THE USER MAKES A SELECTION AND OUR GAME GIVE WHAT HE WANTS , LIKE SELECT A LOCAL MATCH.
- "API" FIREBASE:THIS OUR MAIN "API" BECAUSE , THIS WORKS LIKE OUR DATA BASE AND LIKE OUR DATA INTERMEDIA PASSANGER , IT HELP US TO BRING THE DATA TO THE USER ,LIKE THE TOP SCORES OR THE LOG IN INFORMATION

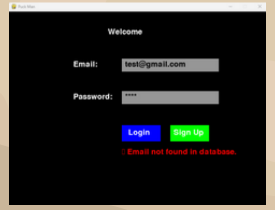


Email: pepito@gmail.com
Password: **********
Incorrect password.

IN THIS CASE , THE USER PUT A WRONG PASSWORD , SO THE DATA BASE DONT ALLOW PLAY ON THE GAME



BUT IF THE INFORMATION IS CORRECT , THE GAME WILL LET U PLAY ON YOUR ACCOUNT



Email: test@gmail.com
Password: ****
Email not found in database.

EVEN IF U DONT HAVE AN ACCOUNT THE GAME WILL LET U CREATE A NEW ONE WITH A PASSWORD AND A NICKNAME. THIS INFORMATION WILL BE ON THE DATA BASE

Faryd walteros