Documentación de la Aplicación - Código Fuente y su Ejecutable

“FantasyMichiHats”

Asignatura: Taller de aplicaciones

Sección: N3P12C1

Nombre del docente: Emilio Yáñez Veloso

Nombre de los integrantes del grupo: Diego Quiñones, Esteban Venegas, María Javiera puga

Tablas SQL.

Consultas y creacion de tablas SQL para base de datos  
  
  
CREATE TABLE User ( user\_id INT PRIMARY KEY, username VARCHAR(255), password VARCHAR(255), email VARCHAR(255), created\_at TIMESTAMP );

CREATE TABLE Character ( character\_id INT PRIMARY KEY, user\_id INT FOREIGN KEY REFERENCES User(user\_id), name VARCHAR(255), class VARCHAR(255), level INT, experience INT, created\_at TIMESTAMP );

CREATE TABLE Game ( game\_id INT PRIMARY KEY, character\_id INT FOREIGN KEY REFERENCES Character(character\_id), current\_level INT, current\_score INT, created\_at TIMESTAMP, updated\_at TIMESTAMP );

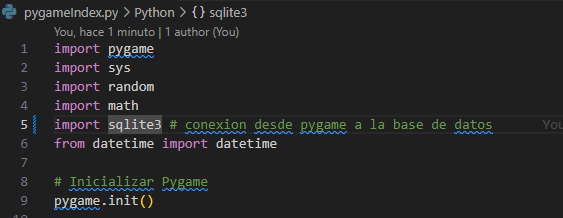
CREATE TABLE Hat ( hat\_id INT PRIMARY KEY, name VARCHAR(255), description VARCHAR(255), class VARCHAR(255), image\_url VARCHAR(255) );

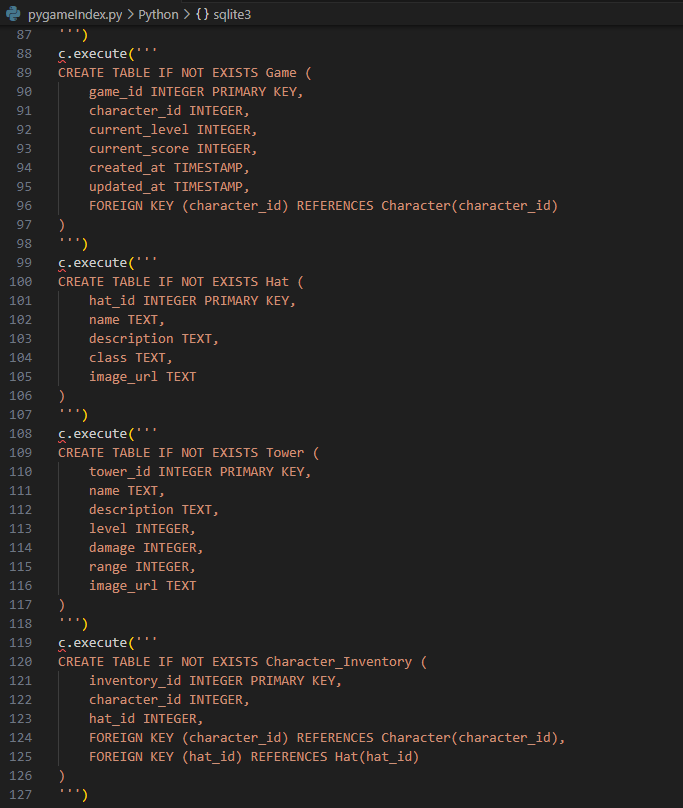
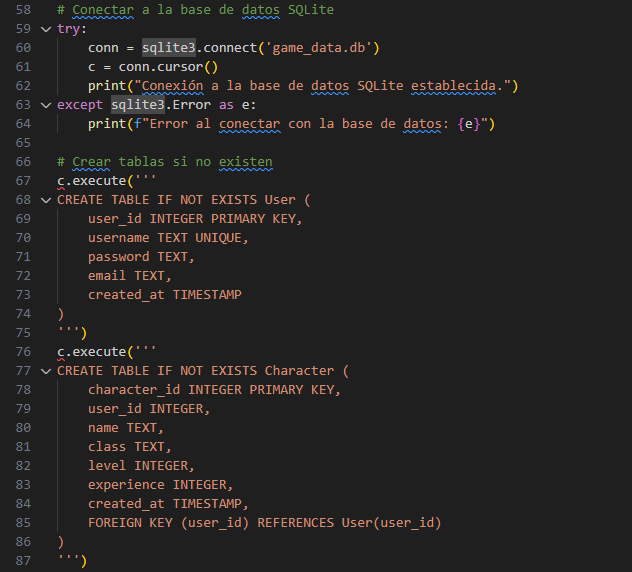
CREATE TABLE Tower ( tower\_id INT PRIMARY KEY, name VARCHAR(255), description VARCHAR(255), level INT, damage INT, range INT, image\_url VARCHAR(255) );

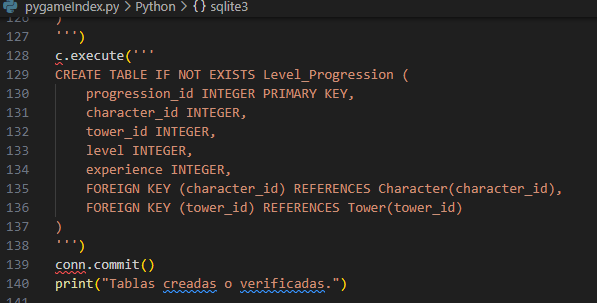
CREATE TABLE Character\_Inventory ( inventory\_id INT PRIMARY KEY, character\_id INT FOREIGN KEY REFERENCES Character(character\_id), hat\_id INT FOREIGN KEY REFERENCES Hat(hat\_id) );

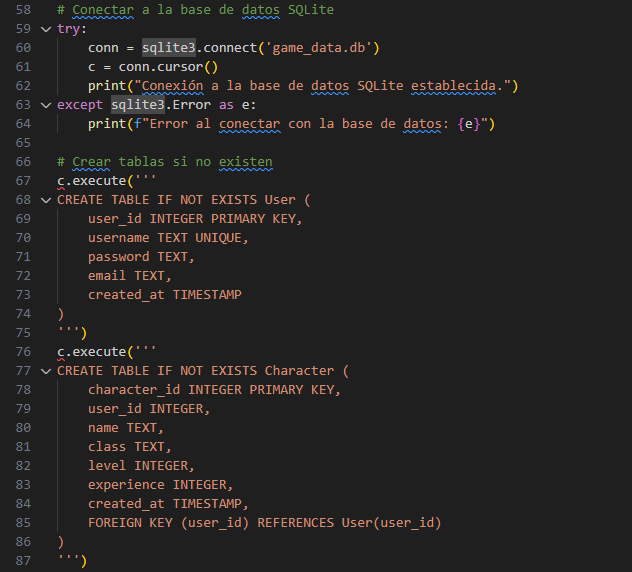
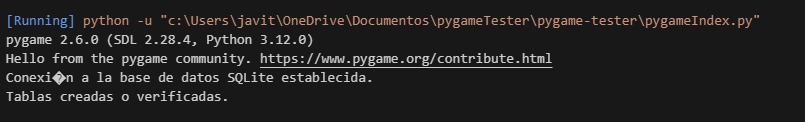
CREATE TABLE Level\_Progression ( progression\_id INT PRIMARY KEY, character\_id INT FOREIGN KEY REFERENCES Character(character\_id), tower\_id INT FOREIGN KEY REFERENCES Tower(tower\_id), level INT, experience INT );

Conexión a base de datos desde python:

Para establecer conexión desde la aplicacion Python en Pygame, a la base de datos Sqlite, se uso la libreria sqlite3.  
En la siguientes imagenes se muestra la conexión desde le codigo:  


Validación de la Estructura de las tablas  




  
Conexion establecida, mensaje en consola:  


Inserción de datos en tablas:  
 - Al registrar un usuario:   
def register\_screen():

username = get\_input\_text("Ingrese nombre de usuario:")

password = get\_input\_text("Ingrese contraseña:")

email = get\_input\_text("Ingrese email:")

try:

c.execute('INSERT INTO User (username, password, email, created\_at) VALUES (?, ?, ?, ?)',

(username, password, email, datetime.now()))

conn.commit()

print(f"Usuario {username} registrado correctamente.")

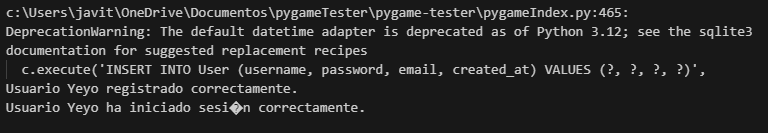
login\_screen() # Ir a la pantalla de inicio de sesión después de registrar

except sqlite3.IntegrityError:

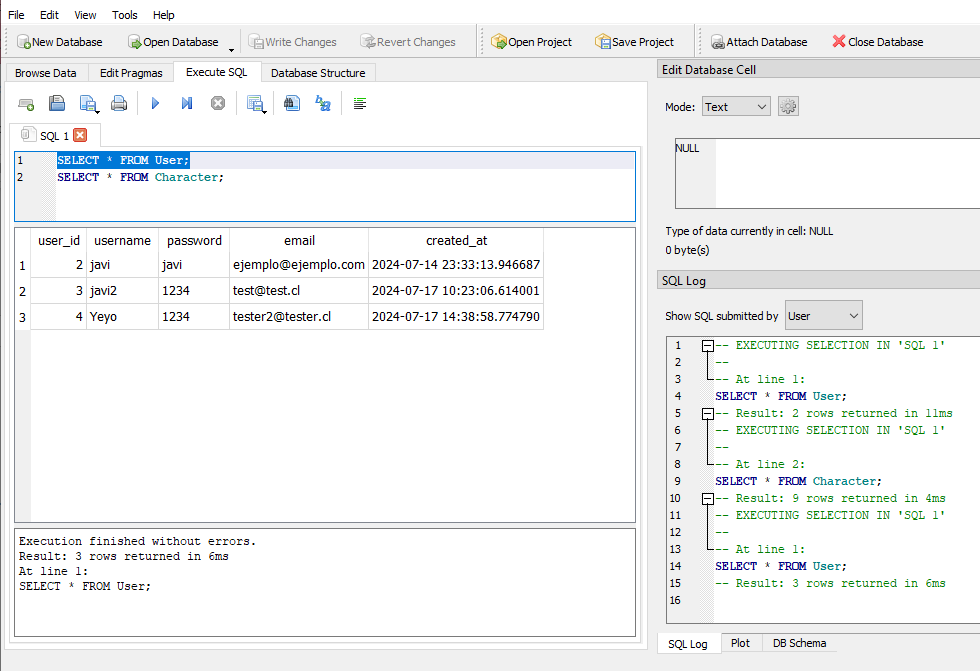
print(f"El nombre de usuario {username} ya existe.")

register\_screen() # Volver a intentar el registro si el nombre de usuario ya existe

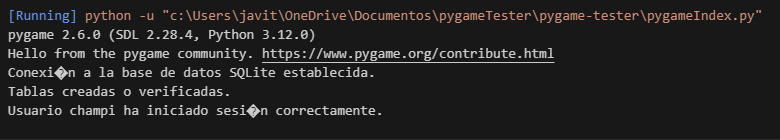
Consola



DB Brownser



* Inicio de sesión.  
  Codigo:  
  def login\_screen():
* username = get\_input\_text("Ingrese nombre de usuario:")
* password = get\_input\_text("Ingrese contraseña:")
* c.execute('SELECT user\_id FROM User WHERE username = ? AND password = ?', (username, password))
* user = c.fetchone()
* if user:
* print(f"Usuario {username} ha iniciado sesión correctamente.")
* main\_menu(user[0])
* else:
* print("Nombre de usuario o contraseña incorrectos.")
* login\_screen() # Volver a intentar el inicio de sesión si los datos son incorrectos
* consola



Creacion de personaje:  
codigo:  
class Character:

def \_\_init\_\_(self, name, char\_class, image, equipment, character\_id=None):

self.name = name

self.char\_class = char\_class

self.image = image

self.equipment = equipment

self.level = 1

self.experience = 0

self.rect = self.image.get\_rect(center=(SCREEN\_WIDTH // 2, SCREEN\_HEIGHT // 2))

self.character\_id = character\_id

def draw(self, screen):

screen.blit(self.image, self.rect.topleft)

if self.equipment:

screen.blit(equipments[self.equipment], (self.rect.x + 50, self.rect.y))

def shoot(self):

if enemies:

closest\_enemy = min(enemies, key=lambda enemy: math.hypot(enemy.rect.x - self.rect.x, enemy.rect.y - self.rect.y))

if self.equipment in ["Casco 1", "Casco 2", "Casco 3"]:

projectile\_image = projectile\_images["Espada"]

elif self.equipment in ["Capa 1", "Capa 2", "Capa 3"]:

projectile\_image = projectile\_images["Báculo"]

elif self.equipment in ["Gorro 1", "Gorro 2", "Gorro 3"]:

projectile\_image = projectile\_images["Arco"]

new\_projectile = Projectile(self.rect.centerx, self.rect.top, projectile\_image, closest\_enemy)

projectiles.append(new\_projectile)

def gain\_experience(self, amount):

self.experience += amount

if self.experience >= experience\_needed:

self.level\_up()

def level\_up(self):

self.level += 1

self.experience = 0

print(f"{self.name} ha subido al nivel {self.level}!")

create\_multiple\_enemies(self.level) # Crear múltiples enemigos cuando el personaje sube de nivel

# Actualizar el nivel del personaje en la base de datos

c.execute('UPDATE Character SET level = ?, experience = ? WHERE character\_id = ?',

(self.level, self.experience, self.character\_id))

conn.commit()

print(f"Nivel del personaje {self.name} actualizado en la base de datos.")

def create\_character(name, char\_class, equipment):

if char\_class == "Guerrero":

char\_image = warrior\_image

elif char\_class == "Mago":

char\_image = mage\_image

elif char\_class == "Arquero":

char\_image = archer\_image

# Insertar el personaje en la base de datos

c.execute('INSERT INTO Character (user\_id, name, class, level, experience, created\_at) VALUES (?, ?, ?, ?, ?, ?)',

(1, name, char\_class, 1, 0, datetime.now()))

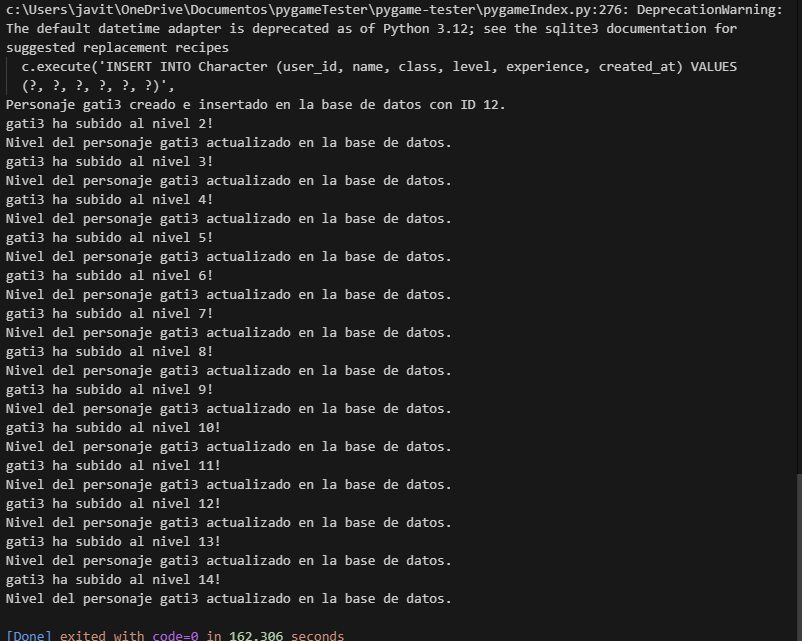
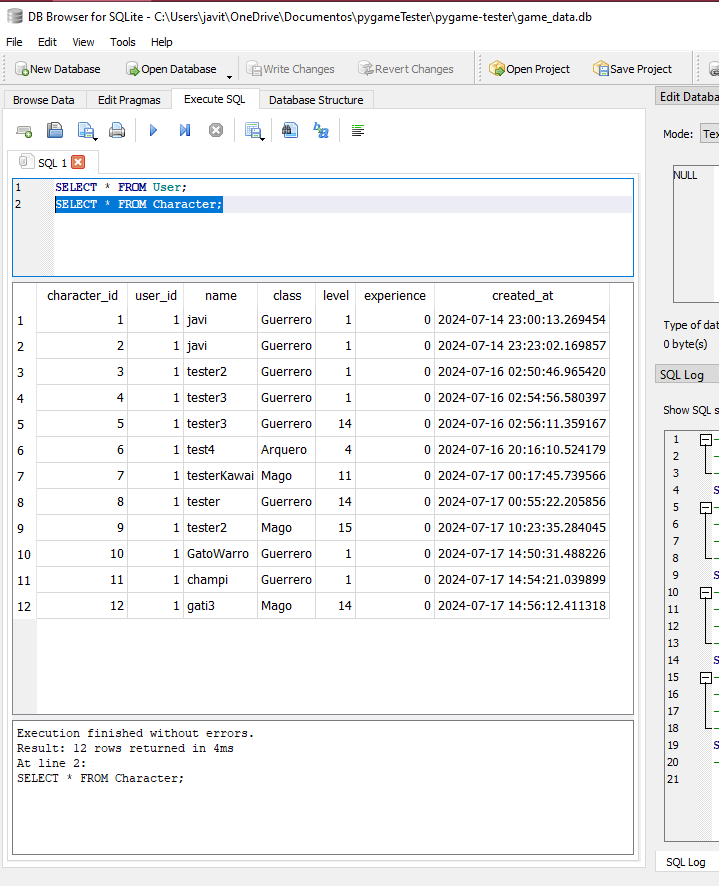
conn.commit()

character\_id = c.lastrowid

new\_character = Character(name, char\_class, char\_image, equipment, character\_id)

characters.append(new\_character)

print(f"Personaje {name} creado e insertado en la base de datos con ID {character\_id}.")

consola:  
DB Brownser:  


Documentación de pruebas  
  
**Registro de usuario**

|  |  |  |  |
| --- | --- | --- | --- |
| **Prueba** | **Resultado esperado** | **Resultado obtenido** | **Estado** |

|  |  |  |  |
| --- | --- | --- | --- |
| Registro de usuario | El usuario se registra correctamente y se almacena en la BD | Usuario registrado correctamente y almacenado | Aprobado |

|  |  |  |  |
| --- | --- | --- | --- |
| Registro de usuario duplicado | El sistema rechaza el registro de un usuario duplicado | El sistema muestra un error y rechaza el registro | Aprobado |

**Inicio de sesión**

|  |  |  |  |
| --- | --- | --- | --- |
| **Prueba** | **Resultado esperado** | **Resultado obtenido** | **Estado** |

|  |  |  |  |
| --- | --- | --- | --- |
| Inicio de sesión correcto | El usuario inicia sesión correctamente | El usuario inicia sesión y accede al menú principal | Aprobado |

|  |  |  |  |
| --- | --- | --- | --- |
| Inicio de sesión incorrecto | El sistema rechaza las credenciales incorrectas | El sistema muestra un error y rechaza el inicio de sesión | Aprobado |

**Creación de personaje**

|  |  |  |  |
| --- | --- | --- | --- |
| **Prueba** | **Resultado esperado** | **Resultado obtenido** | **Estado** |
| Creación de personaje | El personaje se crea y se almacena en la BD | Personaje creado y almacenado correctamente | Aprobado |

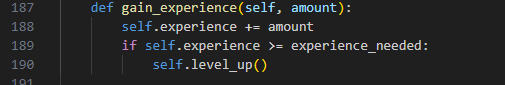
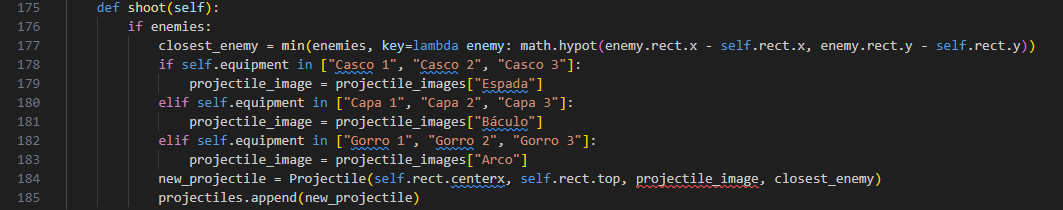
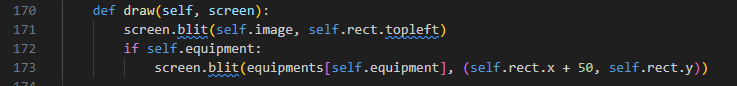
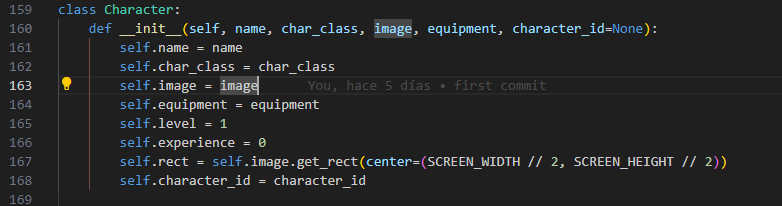
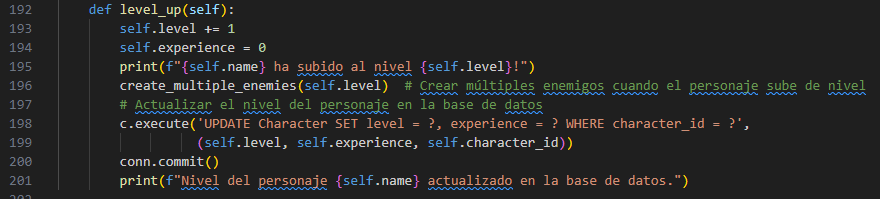
**Juego y nivelación**

|  |  |  |  |
| --- | --- | --- | --- |
| **Prueba** | **Resultado esperado** | **Resultado obtenido** | **Estado** |

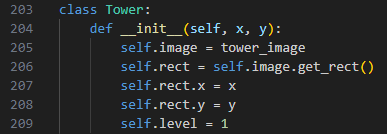
|  |  |  |  |
| --- | --- | --- | --- |
| Ganancia de experiencia | El personaje gana experiencia y sube de nivel | El personaje sube de nivel correctamente | Aprobado |

Documentación de clases  
  
Existen en el código, distintas clases y funciones que hacen funcionar la aplicación.  
A continuación, se documentan en orden:  
  
Clases

**Character**

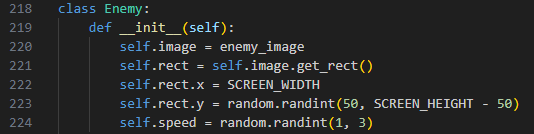
* + 
  + 

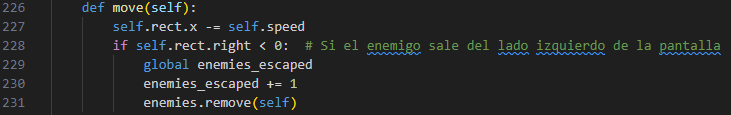
**Tower**



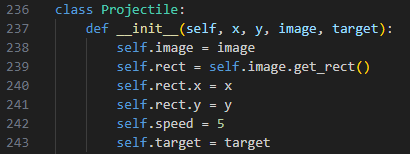
* + 
  + 

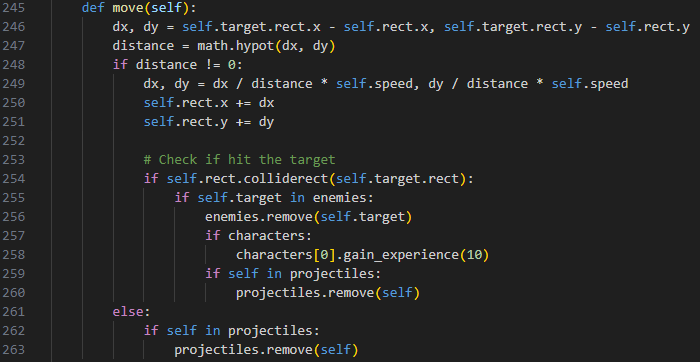
**Enemy**



* + 
  + 

**Projectile**



* + 
  + 

### **Funciones**

1. **create\_character:** Crea un personaje con la clase y el equipamiento seleccionados, lo inserta en la base de datos y lo añade a la lista de personajes del juego.
2. **draw\_text:** Renderiza y dibuja texto en la pantalla en las coordenadas especificadas.
3. **select\_class:** Permite al jugador seleccionar la clase de su personaje mediante la pulsación de teclas.
4. **select\_equipment:** Permite al jugador seleccionar el equipamiento de su personaje mediante la pulsación de teclas.
5. **create\_enemy:** Crea un nuevo enemigo y lo añade a la lista de enemigos.
6. **create\_multiple\_enemies:** Crea múltiples enemigos especificando el número de enemigos a crear.
7. **show\_start\_screen:** Muestra la pantalla de inicio con opciones para registrar, iniciar sesión y salir del juego.
8. **get\_input\_text:** Muestra una caja de entrada de texto en la pantalla y devuelve el texto introducido por el usuario.
9. **register\_screen:** Registra un nuevo usuario pidiendo nombre de usuario, contraseña y correo electrónico. Si el nombre de usuario ya existe, se vuelve a intentar el registro.
10. **login\_screen:** Permite al usuario iniciar sesión pidiendo nombre de usuario y contraseña. Si los datos son correctos, se muestra el menú principal.
11. **main\_menu:** Muestra el menú principal con opciones para jugar, modificar la cuenta y salir del juego.
12. **modify\_account\_screen:** Permite al usuario modificar su contraseña y correo electrónico. Los cambios se guardan en la base de datos y se vuelve al menú principal.
13. **get\_player\_name:** Permite al usuario introducir su nombre de jugador. Devuelve el nombre introducido.
14. **draw\_pause\_menu:** Permite al usuario pausar el juego y volver al menú principal o salir.
15. **switch\_map:** Cambia el mapa actual al siguiente en la lista y reinicia las listas de enemigos y proyectiles.
16. **Main:** Función principal del juego. En esta función, están insertas las “win condition” , “lost condition” y el conteo de segundos en juego.

Manual de instalación y de usuario

### **Instrucciones de instalación**

Presionar en CODE arriba a la derecha en github y descargar como Zip.

Extraer y seguir las instrucciones.

### Ejecutar Juego

Entrar en la carpeta extraida.

Entrar a la carpeta dist/pygameIndex

Ejecutar pygameIndex.exe Nota Puede que Windows Defender de alerta, si es así, presionar en "Más información" y luego en "Ejecutar". No es un virus, puede leer el código fuente cuando quiera.

### **Registro de Usuario**

### Paso 1: Registrar

Presiona el botón de “Registrar”.

Ingresa nombre de usuario, contraseña y email.

### Paso 2: Iniciar Sesión

Después de registrarte, se te pedirá que ingreses tu nombre de usuario y contraseña.

### Menú Principal

En el menú principal, puedes realizar las siguientes acciones:

**Jugar**

**Modificar Cuenta**

**Salir**

### Modificar Cuenta

Puedes cambiar tu contraseña y email desde esta sección.

### Jugar

Escribe tu nombre que verás dentro del juego.

Con los botones 1, 2, 3 elige tu clase de Michi y el sombrero que llevará puesto.

### Controles del Juego

Disparar: Presiona la tecla Espacio para disparar.

No dejes que se te escape ningún enemigo. ¿Hasta qué nivel llegarás?

**Nota**: Asegúrate de que todos los archivos de recursos estén en el directorio correcto para evitar errores de carga.

**¡Diviértete jugando a FantasyMichiHats!**

Términos y Condiciones de Uso

**1. Aceptación de los Términos**

Al descargar, instalar y utilizar el juego "FantasyMichiHats" (en adelante, "el Juego"), usted acepta cumplir y estar sujeto a los siguientes términos y condiciones (en adelante, "los Términos"). Si no está de acuerdo con estos Términos, no debe utilizar el Juego.

**2. Descripción del Juego**

"FantasyMichiHats" es un juego de defensa de torres y RPG en el que los usuarios pueden crear personajes, gestionar equipamiento y defender torres contra oleadas de enemigos. Los jugadores pueden avanzar en niveles, ganar experiencia y modificar sus personajes a medida que progresan en el juego.

**3. Registro y Cuentas de Usuario**

Para acceder a ciertas características del Juego, debe registrarse y crear una cuenta de usuario. Usted es responsable de mantener la confidencialidad de su nombre de usuario y contraseña y de todas las actividades que ocurran bajo su cuenta. Usted acepta notificar de inmediato cualquier uso no autorizado de su cuenta.

**4. Uso del Juego**

**4.1. Licencia de Uso**: Se le otorga una licencia limitada, no exclusiva y revocable para utilizar el Juego únicamente para su uso personal y no comercial.

**4.2. Restricciones de Uso**: No puede modificar, distribuir, vender, alquilar, arrendar, prestar o sublicenciar el Juego. Tampoco puede descompilar, realizar ingeniería inversa o intentar obtener el código fuente del Juego.

**5. Contenido Generado por el Usuario**

**5.1. Propiedad del Contenido:** Usted conserva todos los derechos sobre el contenido que cree o cargue en el Juego. Sin embargo, al cargar contenido, usted otorga al desarrollador del Juego una licencia no exclusiva, mundial, libre de regalías, sublicenciable y transferible para usar, reproducir, distribuir, crear trabajos derivados, mostrar y ejecutar dicho contenido en relación con el Juego.

**5.2. Normas de Conducta:** Usted acepta no publicar contenido que sea ilegal, ofensivo, difamatorio, o que infrinja los derechos de terceros.

**6. Privacidad y Datos Personales**

La privacidad de sus datos es importante para nosotros. Por favor, consulte nuestra [Política de Privacidad](enlace a la política de privacidad) para obtener información sobre cómo recopilamos, usamos y divulgamos sus datos personales.

**7. Limitación de Responsabilidad**

El Juego se proporciona "tal cual" y "según disponibilidad". No garantizamos que el Juego esté libre de errores o que su uso sea ininterrumpido. En la máxima medida permitida por la ley, no seremos responsables de ningún daño indirecto, incidental, especial, consecuente o punitivo que surja del uso o incapacidad de usar el Juego.

**8. Modificaciones a los Términos**

Nos reservamos el derecho de modificar estos Términos en cualquier momento. Las modificaciones serán efectivas inmediatamente después de su publicación en el Juego. Su uso continuado del Juego después de la publicación de las modificaciones constituye su aceptación de los Términos modificados.

**9. Terminación**

Podemos suspender o terminar su acceso al Juego en cualquier momento, sin previo aviso, por cualquier motivo, incluyendo si creemos que ha violado estos Términos.

**10. Ley Aplicable y Jurisdicción**

Estos Términos se regirán e interpretarán de acuerdo con las leyes de Chile. Acepta someterse a la jurisdicción exclusiva de los tribunales de Chile para resolver cualquier disputa relacionada con estos Términos o el uso del Juego.

**11. Contacto**

Si tiene alguna pregunta sobre estos Términos, por favor contáctenos en soporte@fantasymichihat.cl

Fecha de última actualización: 19-07-2024