**Flappy Bird Game**

**Project Overview:** The project aimed to develop a Flappy Bird game using the Pygame library in Python. Flappy Bird is a simple yet addictive game where a player controls a bird to navigate through gaps between pipes, avoiding collisions with pipes or the ground. The project utilized Pygame for game development and MySQL for storing and retrieving player data, such as high scores and user information.

**Pygame Integration:** Pygame, a Python library for game development, was extensively used in this project. Key Pygame functions and classes employed include:

* **pygame.init()**: Initializing the Pygame library.
* **pygame.display.set\_mode()**: Creating a display surface for rendering graphics.
* **pygame.time.Clock()**: Managing the game's frame rate.
* **pygame.event.get()**: Retrieving events from the event queue, such as key presses or mouse movements.
* **pygame.sprite.Sprite**: A base class for creating game objects.
* **pygame.draw.rect()**: Drawing rectangles on the display surface.
* **pygame.mixer.Sound()**: Loading and playing sound effects.

**MySQL Integration:** The project integrated MySQL for storing and retrieving player data. The **mysql-connector-python** library facilitated interaction with the MySQL database. Key functions and methods utilized from this library include:

* **mysql.connector.connect()**: Establishing a connection to the MySQL server.
* **connection.cursor()**: Creating a cursor object to execute SQL queries.
* **cursor.execute()**: Executing SQL queries or commands on the database.
* **cursor.fetchall()**: Retrieving all rows from the result set of a query.
* **connection.commit()**: Committing the current transaction to make changes permanent.
* **connection.close()**: Closing the connection to the database.

**Integration Workflow:** The integration workflow involved several steps:

1. Importing the **mysql.connector** module.
2. Establishing a connection to the MySQL server using **mysql.connector.connect()** with necessary credentials.
3. Creating a cursor object using **connection.cursor()**.
4. Executing SQL queries and commands using **cursor.execute()**.
5. For INSERT and UPDATE queries, committing changes using **connection.commit()**.
6. For SELECT queries, retrieving the result set using **cursor.fetchall()** or **cursor.fetchone()**.
7. Closing the cursor and connection when finished using **cursor.close()** and **connection.close()**.

**Code Overview:**

* **game.py**: Implements the Flappy Bird game using Pygame. It includes game logic, user interface, and database integration for storing high scores.
* **bird.py**: Defines the Bird class representing the player-controlled bird in the game.
* **pipe.py**: Defines the Pipe class representing the obstacles in the game.

**Conclusion:** The integration of the Flappy Bird game with MySQL database management system enhances the gaming experience by enabling persistent storage and retrieval of player data. By leveraging Pygame for game development and mysql-connector-python for database interaction, the project successfully achieves its objectives of creating an addictive game with high score tracking capabilities.