**Snake Game – Console-Based Python Project Using Data Structures**

**📌 Overview**

This project is a **console-based Snake game** developed entirely in **Python**, using the **Pygame** library to handle graphics, keyboard input, and the game loop. Unlike many modern games that require a full stack (front-end and back-end) development approach, this project is built without any front-end frameworks (like HTML/CSS or GUI libraries) or back-end systems (like databases or servers). Instead, it relies purely on Python's capabilities, logic, and core **data structures** such as **lists and queues** to control the game's internal mechanics.

The primary goal of this project is to **demonstrate the application of data structure concepts** in a fun and interactive way. It also serves as an excellent learning exercise for beginners or intermediate Python programmers who want to understand how games work internally and how game state can be managed through structured programming.

### 🛠️ Technologies Used

* **Programming Language:** Python
* **Graphics Library:** Pygame
* **Data Structures:** Lists, Queues
* **Interface:** Console / Terminal-based (No GUI or web-based frontend)

### Game Description

The Snake game is a classic arcade-style game where the player controls a snake that moves around the screen, eats food, and grows longer. The objective is to score as many points as possible by consuming food items while avoiding collisions with the wall and the snake’s own body.

#### Key Features:

* Snake movement is controlled using arrow keys.
* Each time the snake eats food, it grows longer.
* Collision with the wall or the snake’s own body ends the game.
* The current score increases with each food item eaten.
* The game speed increases gradually to make it more challenging.

### Application of Data Structures

This project highlights how simple data structures can be effectively used in a real-world game development scenario:

1. **List for Snake Body:**
   * The snake's body is stored as a list of coordinates.
   * Each movement updates the head and tail of the list accordingly.
2. **Queue-like Behavior:**
   * As the snake moves, new positions are added to the front (head) and older positions are removed from the back (tail), mimicking queue behavior.
3. **Collision Detection:**
   * The snake’s position is checked against the wall boundaries and its own body list to detect a collision.
4. **Food Position:**
   * The food's position is randomly generated and compared with the snake’s body to avoid overlap.

This combination of data structures creates an efficient and logical way to manage game state without complex algorithms.

**How It Works**

1. The Pygame window initializes a screen with defined width and height.
2. The snake is initialized as a list of coordinates representing its body.
3. A game loop starts, handling:
   * Keyboard inputs (for direction)
   * Updating the snake’s position
   * Checking for collisions
   * Displaying graphics and text (score, food, snake body)
4. Food appears randomly on the screen.
5. When the snake eats the food:
   * The food is repositioned
   * The snake grows longer
   * The score is increased
6. If the snake hits a wall or its own body:
   * The game ends and a "Game Over" message is displayed.

**Objectives of the Project**

* To understand the basics of **game loop implementation**.
* To apply **data structures** like lists and queues in a practical use case.
* To demonstrate how games can be developed **without front-end and back-end**.
* To create a fun, console-based Python project that can be used as a portfolio item.

**Project Structure (Example)**

SnakeGame

│

├── snake\_game.py # Main game file

├── README.md # Project explanation and documentation

├── assets/ # (Optional) Images or sound files

└── requirements.txt # Pygame or other dependencies

**How to Run the Project(IN Command Prompt)**

1. Install Python (preferably version 3.8 or above).
2. Install Pygame using pip:

pip install pygame

1. Run the Python file:

python snake\_game.py

**Learning Outcomes**

By completing this project, you will:

* Understand how a game loop works.
* Learn how to control game state using **lists and queues**.
* Get hands-on experience using **Pygame** for simple 2D graphics and event handling.
* Improve logical thinking and structured problem-solving.

**Contributions**

This project is open to suggestions and contributions. If you have ideas to improve the game (like adding levels, increasing difficulty, or adding a scoreboard), feel free to fork the repository and submit a pull request!

**License**

This project is open-source and available under the MIT License.

**Final Notes**

This Snake game project is a simple yet powerful example of how you can **build functional and entertaining applications using only Python and basic data structures**. It’s a great starting point for students, hobbyists, and anyone looking to explore game development without diving into heavy frameworks or complex architectures.