What is Unit Testing?

Unit testing is a method of software testing where individual "units" of a program are tested to verify they behave as required. In the context of the Yatzy game, a "unit" is the smallest testable unit, i.e., a method of the Yatzy class (e.g., Ones(), TwoPairs(), Yatzy()).

- **Key Characteristics**:
  - **Isolation**: Each unit is tested individually independent of the rest of the program. For instance, testing TwoPairs () has nothing to do with the overall game logic, just the dice value.
  - **Automation**: is typically automated and thus can be executed repeatedly with minimal effort (e.g., via a testing framework like pytest).
  - **Granularity**: Focused on single functionality in contrast with the whole application.
- **Purpose**:
  - Confirm that each method returns the correct output for given inputs.
  - Detect issues early during development.

Why is Unit Testing Important?

1. **Reliability**: Ensures your Yatzy strategies compute scores accurately (e.g., Yatzy() returns 50 points only if the five dice are identical).
2. **Maintainability**: Enables code changes (e.g., enhancing Pair() logic) by preventing updates from introducing new bugs.
3. **Automation in CI/CD**: Automated tests can be integrated with tools like GitHub Actions, running on every code commit to maintain quality.

4. **Documentation**: Tests provide clear examples of what methods should accomplish (e.g., a test for LargeStraight() shows that it expects the sequence 2-3-4-5-6 to appear).
5. **Teamwork**: Unit tests allow other team members to test your code, so methods work as intended when inserted into the bigger picture.

How Unit Testing Applies

- Implementing a Yatzy class with specific methods.
- Writing unit tests for each method to verify their scoring logic (e.g., Ones (), TwoPairs() etc).
- Configuring automated testing with GitHub Actions to execute tests.