SOLID Principle	Description	Applied in the Code	Comments
Single Responsibility	A class should have only one reason to change, meaning it should have a single responsibility.	No	The class handles UI, timing management, and business logic. It should be split.
Open/Closed	Entities should be open for extension but closed for modification.	No	The current class is not designed to allow extensions without modifications.
Liskov Substitution	Objects of a derived class should be replaceable with objects of the base class without altering the functioning of the program.	Not applicable (no inheritance in the shown code)	This principle is not applicable since inheritance is not used in the code.
Interface Segregation	Clients should not be forced to depend on interfaces they do not use.	Not applicable (no interfaces in the shown code)	This principle is not applicable since interfaces are not used in the code.
Dependency Inversion	Dependencies should be on abstractions, not concrete details. Abstractions should not depend on concrete details.	No	The CyberManagementPanel class depends directly on the concrete implementation of CyberManager.