Explain the use of design pattern for Can’t Stop Game:

Design patterns are proven solutions to commonly occurring problems in software development. They provide a standard approach to solving specific design problems, making it easier for developers to write maintainable and reusable code. In the case of the Can't Stop game, we have used the Information Expert pattern in the Can'tStopGame class. This pattern suggests that a class should have all the necessary information required to carry out its responsibilities. In this case, the Can'tStopGame class has all the required information for starting, loading, saving, quitting, and changing the difficulty of the game. By following this pattern, we have created a clear and logical structure for our software that is easy to manage and extend over time.

However, there are some areas where we could improve the design of the game. For example, we could implement the Low Coupling and High Cohesion pattern, which is a key element for good software maintenance and code reuse. By using this pattern, we could create more independent and reusable classes, such as the DicePanel class, which would not be dependent on the Can'tStopGame class. This would make it easier to modify and reuse code in the future.

Another design pattern that we could implement is the Pure Fabrication pattern. This pattern suggests that we create classes solely for the purpose of representing an abstract concept, rather than representing a physical entity. In the context of the Can't Stop game, we could have created an artificial class to manage the player instance and other information, instead of directly storing it in the Can'tStopGame class. We could also use Persistent Storage to add an extra layer of indirection between the player information and the database, making it easier to manage the storage of game data.

Overall, by using design patterns in our software development, we can create a more efficient and maintainable codebase for the Can't Stop game.