

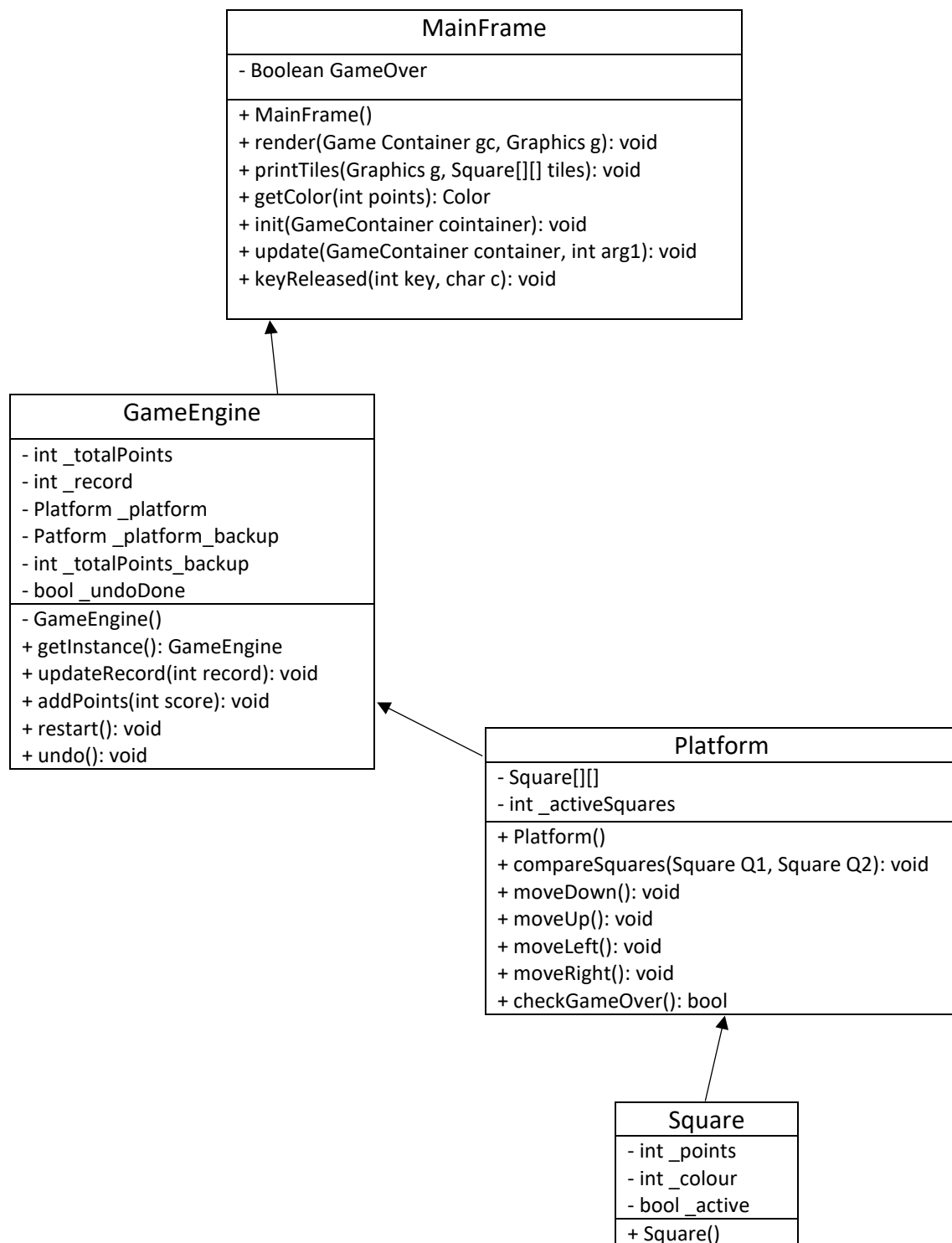
### **Design Patterns: Singleton**

1. Write a natural language description of why and how the pattern is implemented in your code.

Our game is 2048 and its code is organized in this form: a Square class, a Platform class with 16 Squares and a GameEngine class. We also have a MainFrame class to provide interface for the game and the main thread. Main Frame will create one instance of GameEngine which will have two instance of Platform Class (one for back up).

We have decided to implement the Singleton Pattern in our code to limit the GameEngine class to a single instance. Because we want to make sure that the main thread will be able to make the game platform only one time during the match. The form that we have implement that is only by doing the GameEngine constructor private and making a new method called getInstance(). So, that way, every time that the constructor is called is given the same GameEngine instead of giving a different game.

2. Make a class diagram of how the pattern is structured statically in your code.



3. Make a sequence diagram of how the pattern works dynamically in your code.

