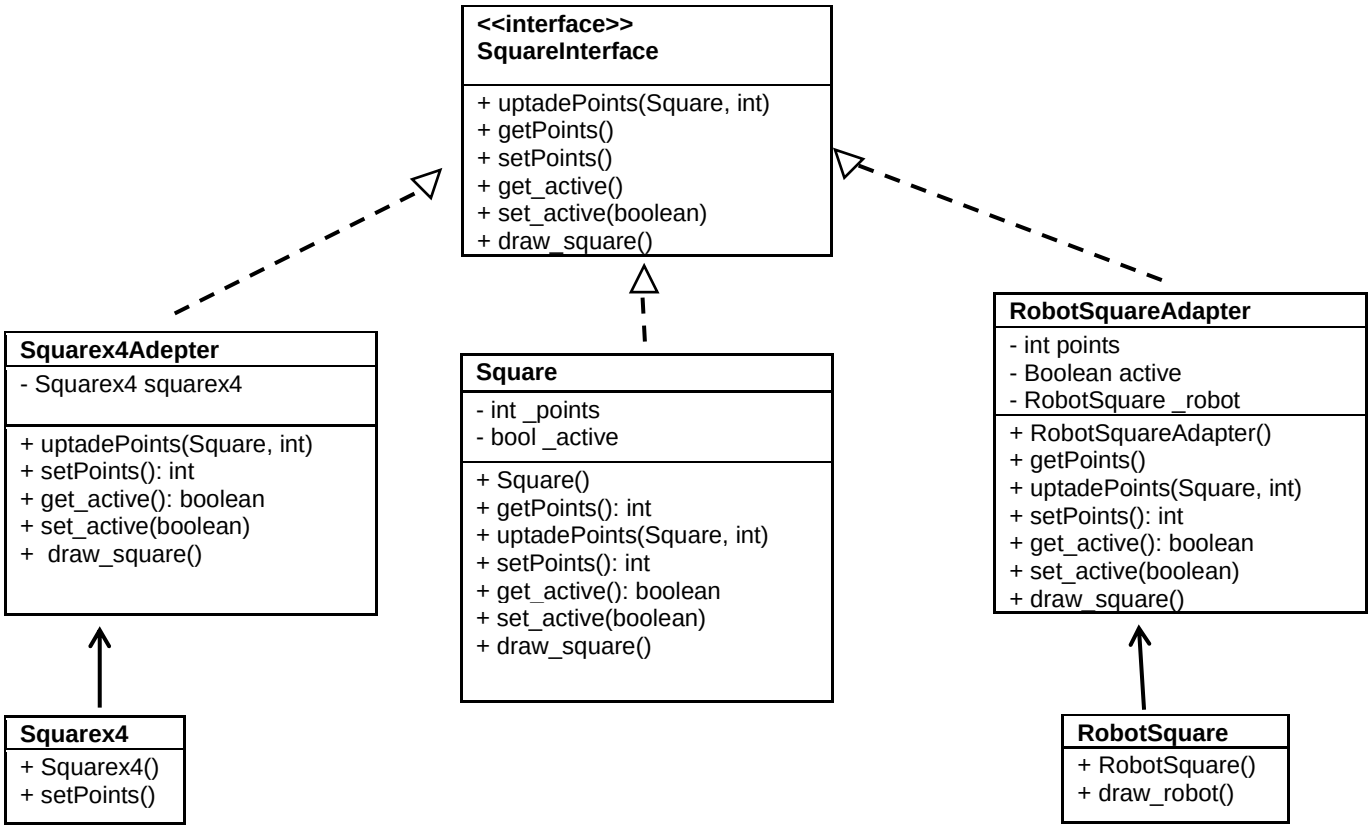
Design Patterns: Adapter Pattern

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

We have decided to implement the Adapter Pattern to extend the functionalities of the game. We doesn't want to change Platform class, adapter help us to 'adapt' new squares modes to our existing game. Indeed now there are three game modes: the "easy" mode where the points are multiplied for 4, the "hard" mode (normal 2048 game) where the points are multiplied for 2 and the robot version, where there's no points showed, just images of robots.
We created an interface called SquareInterface and classes implementing it : Square, Square4Adapter and RobotSquareAdapter. There are another classes called SquareX4, RobotSquare, with the functions that we have to adapt to our game. When the player choose the game mode, the correct class is used, respectively Square for the "hard" game, SquareX4 for the "easy" game and RobotSquare for robot game mode.

Before using the Adapter Pattern, in the package Game we had only a class called Square, but this didn't allow to extend the game. In this way, with appropriate classes "adapter" it is possible to extend the game functionalities.

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