Pygame Implemented Space Invaders

For the completion of the assignment, I use 4 classes, one for the player, one for the enemy, one for the missile and one for the game. Inside of the main class, critical instance attributes are fabricated in order to account for the screen, ship, enemy and the missiles. The rungame() loop, contains various events such as the keyboard interaction which handles the ship movement as well as the missile fire. Collisions are handled as well within the game loop, by calling the method isCollidingWith() upon the instance of enemy. The scoreboard is also drawn through use of various pygame modules.

The next class that gets created is the RocketShip class which inherits from the main my game class. Within this class, various methods for handling the movement of the player, as well as getting both the y and x position are addressed.

I use Inheritance from the rocket ship class for the enemieship class and the missile class for the getXpos() and getYpos() methods while method overriding the draw methods in each class. This therefore underlines the use of polymorphism.

Subsequently another class for the Missile is created, which inherits from MyGame, and has the colliding method as well as a method to get the icon of the missile and moving the missile as well.

I declare all my private variables inside the game class thereby ensuring encapsulation.

Thus, essentially the game has one big application while loop that collects key presses, creates missile instances and enemy instances, has the quit game clause, calls move methods for missiles and enemies, and calls the collision method and removes enemies if true, and refreshes the screen

The game starts automatically when the code is run. For every kill you are rewarded 5 points, once you have killed all aliens in one spawn, a further 6 will appear immediately. If one of the enemies reaches the bottom of the screen, you loose, however if you can destroy the rare green ship that appears, you win the game instantly.