

The drone has all the capabilities mentioned in the assignment sheet. I chose that a picture can only be saved immediately after it is taken (the client must decide in advance whether a picture is to be saved), and the camera unfocuses with every movement. Landings and take-offs can occur in the middle of the sequence, you cannot take-off if the drone is already in the air, and cannot land if the drone is on the ground. The drone cannot move if on the ground. If a faulty move is added to the move sequence (e.g. trying to move when not in the air), the drone will report an error but continue executing the next moves, simply ignoring the erroneous error.

Drone is a subclass of RemoteControl in order for Drone and all its methods to remain private, this way the client can’t create and or control a drone without a RemoteControl and move sequence. All of the RemoteControl methods are public because the client needs the ability to access and manipulate moves, however the actual moves sequence is final and private so that it cannot be edited without use of RemoteControl methods. I created MoveType and FileFormat enums so that the client can only use predetermined file formats and moves. I made a Moves class so that moves can be created with necessary parameters. The trade-off here is that the Moves class is not fully encapsulated, but it does not need to be private because it is useless without a Drone and RemoteControl, so it is not dangerous for the client to have access.