**A nested object hierarchy is required for rotations to work properly – this practice is the recommended solution for nested, additive rotation (objects ‘within’ objects) by Unity. For two objects to rotate without affecting each other, they must be on the same level, such as Main Camera and PlayerShip as shown below.**

Ship (or other object to be animated) object hierarchy example:



CameraBank.cs, CameraPan.cs

ShipController.cs, ShipBoost.cs, ShipRotate.cs, ShipMove.cs

ShipPassiveRotation.cs

ShipRoll.cs

ShipBank.cs

ShipWing.cs