a) Como se llama y donde está ubicado el fichero base en lua para inicializar las naves?
Es el fichero “manager.lua” ubicado en “assets/StreamingAssets/Lua”, este fichero llama a métodos de CS.

b) Como se indica a que nave va a seguir la cámara del juego?
Se indica en el fichero “manager.lua” con el método “manager:SetFollowCamera("SpaceShip1")”

c) A que hace referencia el segundo parámetro de la función que sirve para crear naves (CreateSpaceShip)?
El segundo parámetro utilizando en “manager.lua” del método “manager:CreateSpaceShip(⋯)” introduce como valor “seek_mouse” por lo que podría darnos impresión de ser un parámetro para controlar el comportamiento de la nave.
Entrando en esta caja negra, podemos ver lo que hace exactamente ese “seek_mouse”, encontramos un fichero en C# donde podemos ver como carga el comportamiento de la nave a un fichero lua con el nombre del parámetro, en este caso “seek_mouse”
En “assets/StreamingAssets/Lua” tenemos un fichero llamado “seek_mouse.lua” el cual es el utilizado en el comportamiento de la nave.
Este fichero tiene un método “update” que será el ejecutado para el comportamiento de la nave.

d) Explica el concepto de Proxy en MoonSharp. ¿Qué clases proxy hay en el proyecto MoonSharpUnityTest?
El concepto de proxy hace referencia a un agente intermedio entre el código en Lua y el programa de Unity y su ejecución, este intermediario ofrece seguridad al proyecto y compatibilidad.
En el proyecto se registran 2 tipos de proxy, uno entre las clases Manager y GameManager y el otro entre las clases SteeringObject y SteeringController.

e) Escribe la línea que tendríamos que añadir en manager.lua para crear una nave llamada “Ship1” que use el comportamiento seek.lua, para seguir a otra nave llamada “Ship2”. (La línea debe funcionar, no olvidéis ningún parámetro ni valor)

```
manager:CreateSpaceShip("Ship1", "seek", 0, 0, {mass=10, max_velocity=12,
max_force=12, max_speed=12, stop_distance=10}, {target="SpaceShip1"})
```

f) Explica los dos últimos parámetros de la función CreateSpaceShip.
Los dos últimos parámetros del método “CreateSpaceShip(⋯)” visto en “manager.lua” parecen ser 2 tablas con información sobre la nave y sobre le objetivo. El primer parámetro, por los valores que introducimos, es claramente la definición de la nave que estamos creando, su masa, su velocidad, su fuerza, etc. Y el segundo parámetro, parece que hace referencia al objetivo que tiene.
Indagando en el código podemos ver que estas dos tablas se introducen en los métodos “SetNumberData(⋯)” y “SetStringData(⋯)” de la calse SteeringController. El primer parámetro que estamos analizando, guarda la tabla creada en “manager.lua” dentro de un diccionario en C# y el segundo parámetro almacena el nombre del “target” en una variable llamada “mTargetName”
Más adelante en los ficheros de Lua basados en el comportamiento de las naves, podemos ver como se utilizan los métodos “GetNumberData(⋯)” y “GetTarget()” para recuperar los valores de la nave y el target antes de hacer el “update” del comportamiento, de tal forma que los valores introducidos en la creación de la nave en “manager.lua” se almacenan en “SterringController.cs” y se utilizan en los comportamientos definidos en Lua, como en por ejemplo en “seek.lua” y “seek_mouse.lua”

g) Porque la función print dentro de lua funciona como un Debug.Log de Unity?
Cada vez que se crea una nave y se establece su comportamiento dentro del método “SetBehaviour(⋯)” en la clase “SteeringController.cs”, se puede apreciar como con esta línea: “mScript.Globals["print"] = (Action<string>)((msg) => Debug.Log(msg));” establece la global “print” de Lua como una “Action<String>” y le indica que al utilizarse la global “print” como método con parámetro de texto, utilize el “Debug.log(⋯)” de Unity.

h) Que son los Custom converters? ¿Para qué se utiliza esto en el proyecto MoonSharpUnityTest?
MoonSharp tiene una forma de hacer la conversión de valores entre Unity y Lua, sin embargo, podemos modificar los procesos de conversión de valores para adaptarlos a las necesidades de nuestro proyecto. Para este proyecto se utiliza para la conversión entre C# y Lua de los objetos Vector2 y Vector3