Homework 4, Design Patterns

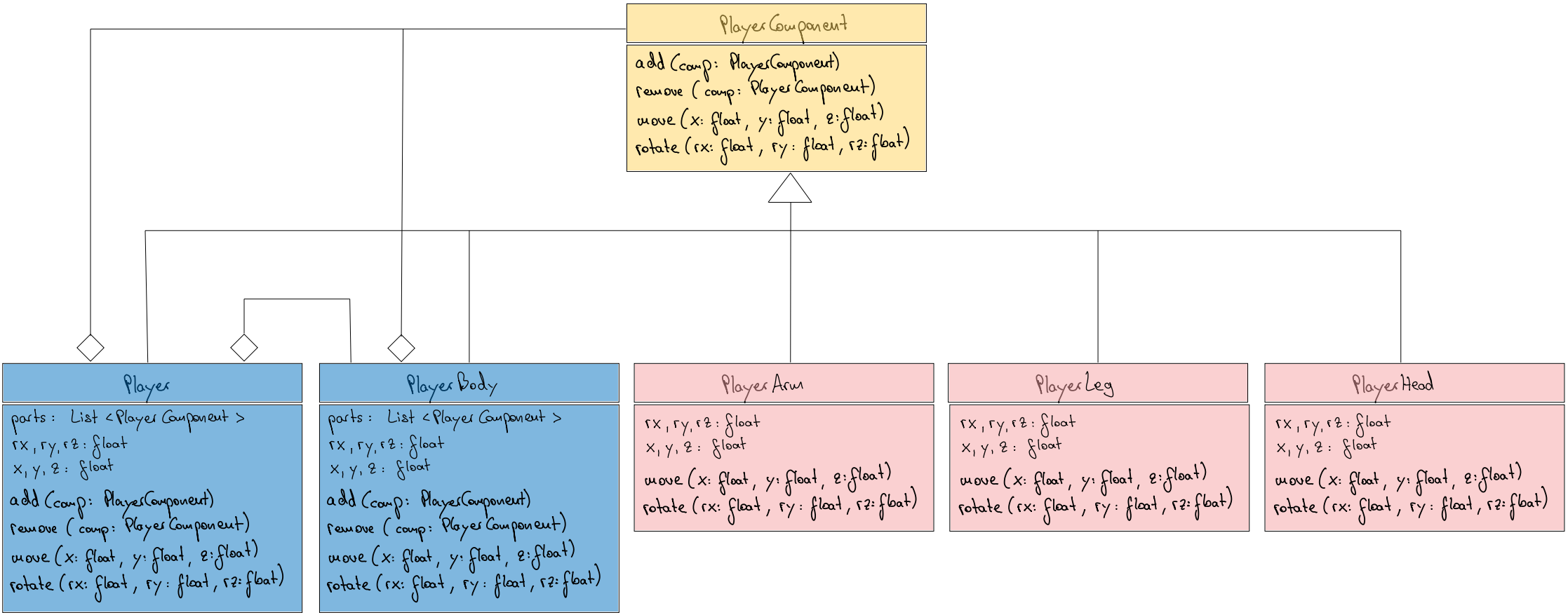
Exercise 1

The code implements the Composite Pattern, which is evident from the following characteristics:

- Component Interface: The PlayerComponent interface defines common methods (add, remove, move, rotate) for all components, ensuring a uniform interface for both simple and complex elements.
- Composite Classes: The Player and PlayerBody classes act as composite objects. They maintain a list of PlayerComponent objects (parts), allowing them to store and manage child components. They implement the PlayerComponent interface and provide implementations for adding, removing, and performing operations (move, rotate) on their child components.
- Leaf Classes: The PlayerArm, PlayerHead, and PlayerLeg classes are leaf components. They also implement the PlayerComponent interface but do not maintain child components. Their add and remove methods are empty, reflecting that they cannot contain other components.
- Hierarchical Structure: The pattern allows the creation of a tree structure where Player can contain PlayerBody, and PlayerBody can contain PlayerArm, PlayerLeg, and so on. This hierarchy enables complex compositions of simple and composite objects, demonstrating the core principle of the Composite Pattern.

By following these principles, the code allows for treating individual objects and compositions uniformly, which is the essence of the Composite Pattern.

See also the pattern diagram on the next page.


PlayerComponent
add (comp: PlayerComponent)
remove (comp: PlayerComponent)
move (x: float, y: float, z: float)
rotate (rx: float, ry: float, rz: float)
Player
parts: List <PlayerComponent>
rx, ry, rz: float
x, y, z: float
add (comp: PlayerComponent)
remove (comp: PlayerComponent)
move (x: float, y: float, z: float)
rotate (rx: float, ry: float, rz: float)
PlayerBody
parts: List <PlayerComponent>
rx, ry, rz: float
x, y, z: float
add (comp: PlayerComponent)
remove (comp: PlayerComponent)
move (x: float, y: float, z: float)
rotate (rx: float, ry: float, rz: float)
PlayerArm
rx, ry, rz: float
x, y, z: float
move (x: float, y: float, z: float)
rotate (rx: float, ry: float, rz: float)
PlayerLeg
rx, ry, rz: float
x, y, z: float
move (x: float, y: float, z: float)
rotate (rx: float, ry: float, rz: float)
PlayerHead
rx, ry, rz: float
x, y, z: float
move (x: float, y: float, z: float)
rotate (rx: float, ry: float, rz: float)