Observer Pattern

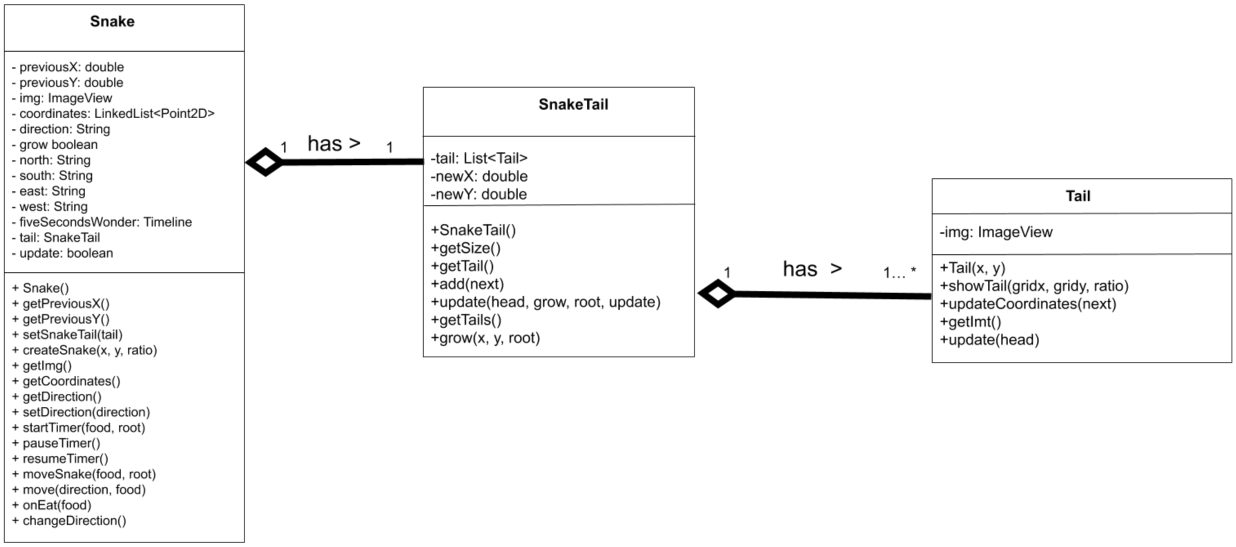
How is it used in our snake implementation?

We used the observer pattern in our game for the interaction between the head of the snake and the tail parts of the snake. The head of the snake is moved by the user, the user doesn't directly interact with the tail of the snake, however we do need to keep track of where our tail is through the SnakeTail class and the tail should move in the same direction as the head moved before it. To this end we used the observer pattern to keep a list of tail elements in our SnakeTail class and update their locations when the head itself moves. We do this by iterating over all the tail elements once the head moves and calling their method to make them move as well.

What downsides did this implementation bring?

Instead of all the tail elements being able to determine themselves if they hit a wall or another tail element in parallel we now need to traverse the entire list of tail elements from our head in order to check if we hit a wall or another tail element.

Class Diagram of our implementation



Implementation Code

The code for our implementation can be found in the Snake, SnakeTail and Tail classes.