

Práctica 3

Análisis y Diseño de Software

Miguel Ibáñez González y Sergio Hidalgo Gamborino

Apartado 1:

1. ¿Qué pasaría si dejáramos cambiar el nombre del estado una vez creado?
No se podría buscar cuál es el estado correspondiente.

¿Cómo lo has evitado?

Haciendo privada la variable name en State y no haciendo un setter.

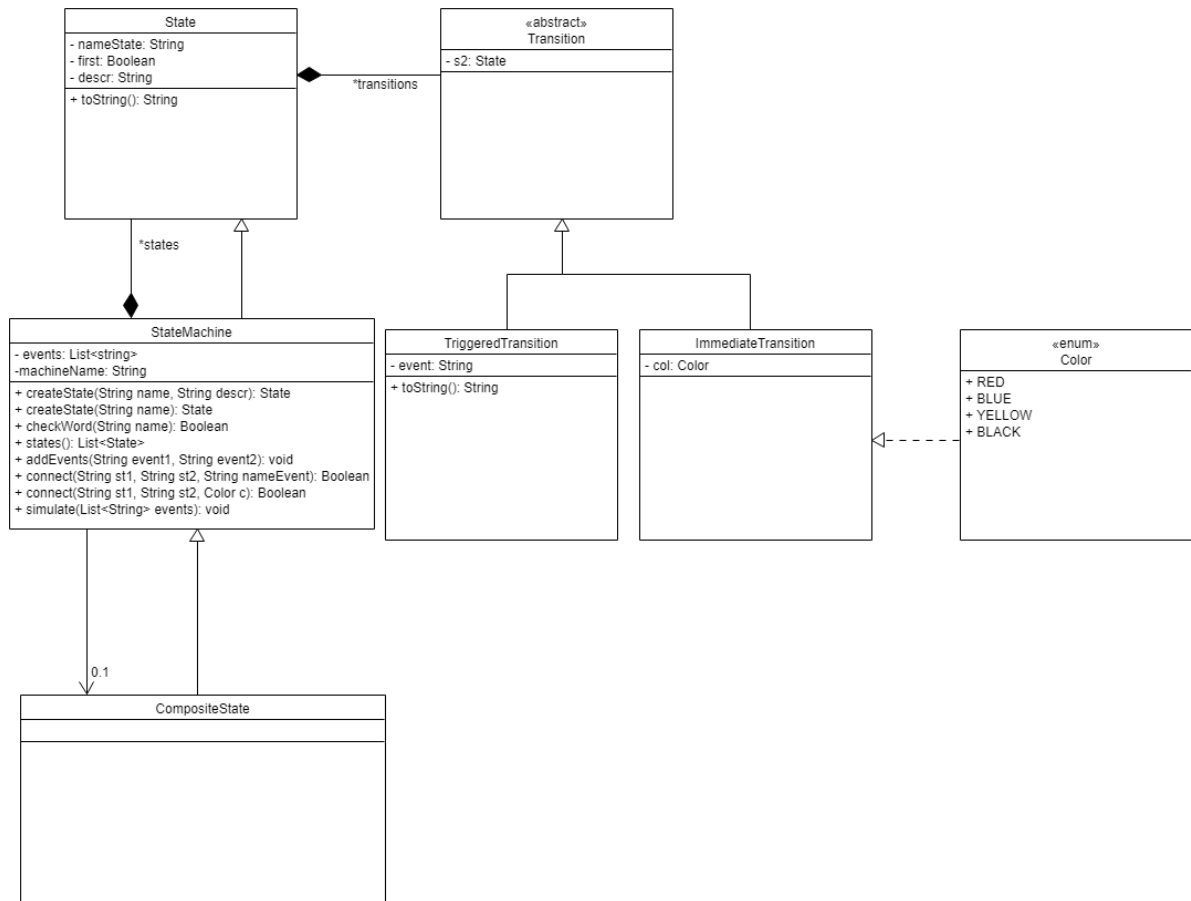
2. ¿Qué has hecho para evitar crear objetos de tipo State fuera del paquete stateMachine?

Hacer el constructor de tipo Protected.

Apartado 2:

Con este diseño ¿qué podría ir mal si dejáramos conectar un estado con otro directamente desde la clase State?

Al no tener acceso al resto de estados, no se tiene tanto control a cerca de con que estado se conecta cada estado.



Dadas las similitudes entre StateMachine y CompositeState se decidió que la segunda heredase las funciones y características de la primera. El problema más notable ha sido añadir los eventos de StateMachine a CompositeState, que se ha solucionado pasándole una variable tipo StateMachine para poder tener acceso a los eventos y copiarlos en la array de eventos del CompositeState. Además, en la simulación ha habido que añadir ciertas condiciones lógicas para hacer que funcionase de manera correcta con los estados de tipo CompositeState. Además, para usar ciertos métodos ha habido que añadirlos de forma protegida a State, de tal manera que solo serán visibles y ejecutables dentro de la propia lógica del paquete, no fuera, por lo que no ponen en peligro la jerarquía de clases.