

]

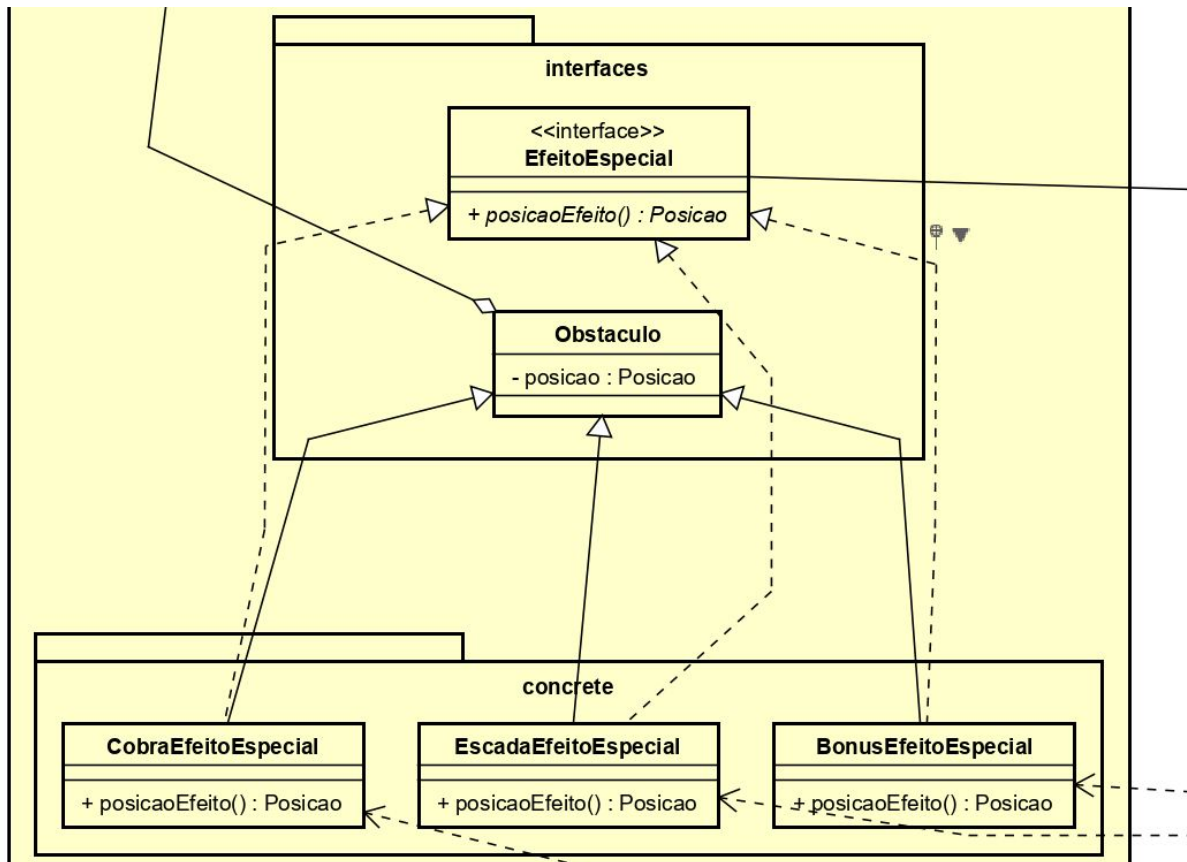
Documento de justificação de escolhas

Disciplina de Projeto Detalhado de Software

Victor Lucas da Silva Monteiro

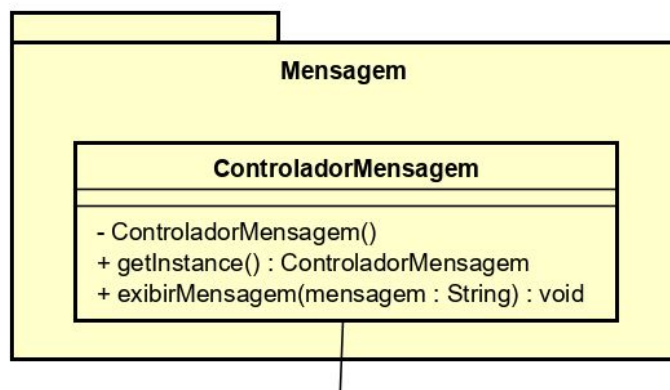
Princípio da substituição de Liskov

Este padrão/princípio foi utilizado ao longo de boa parte do projeto, em principal nas classes que necessitam de polimorfismo para executar ações. Fora utilizado, por exemplo, para aplicar diferentes tipos de **EfeitoEspecial**, bem como para fabricá-lo.

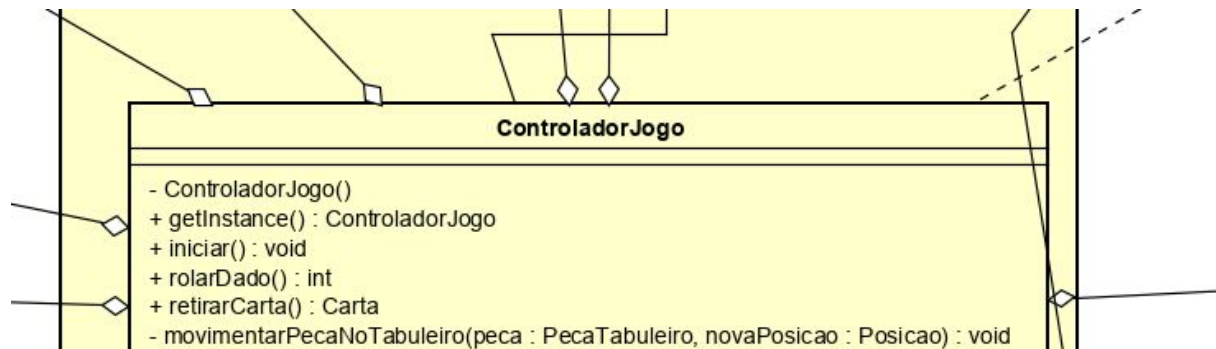


Singleton

Foi utilizado Singleton para algumas partes do projeto, uma delas é para exibição de mensagens, pois esta classe (a **ControladorMensagem**), disposta abaixo, deve poder ser acessada de qualquer outra classe, para que o jogador possa ser notificado do que está acontecendo no jogo.

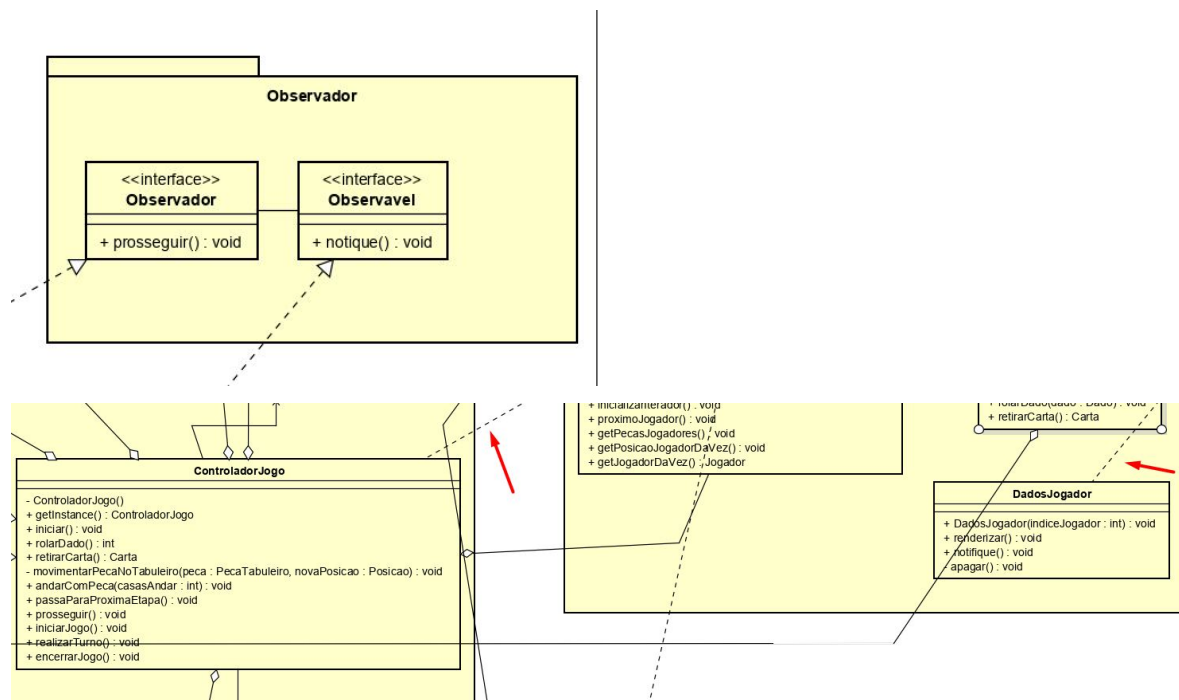


Um outro lugar que utiliza deste padrão é a classe **ControladorJogo**, pois é necessário que alguns métodos possam ser acessados pelas interfaces e, para isto, não iremos instanciar um controlador de jogo em cada interface, mas sim utilizar do padrão para facilitar.



Observer

Este padrão foi utilizado para algo bastante específico e único. Ao solicitar os nomes dos jogadores é necessário que se renderize uma tela que, justamente, pede o nome do jogador e esta tela deve ser renderizada uma a uma. Para isto, foi criado o observador para saber quando a tela é fechada e notificar o **ControladorJogo** para que renderize a próxima tela.



Factory Method

Por fim foi utilizado o padrão fábrica para poder produzir as classes que implementam **Casa** e **EfeitoEspecial**.

