Padrões de Desenho

Color Ninja

# *Abstract Factory*

Com o nosso jogo a basear-se no lançamento de formas para o ecrã do dispositivo, decidimos usar este padrão para a criação dessas mesmas formas. Como este padrão permite usar vários “construtores” de objetos, podemos elaborar qualquer forma. Assim, podemos definir comportamentos semelhantes para todas as formas, criando-as com as propriedades que bem entendermos.

# *Singleton*

Este padrão de desenho será usado especificamente para a classe do nosso jogo que reunirá todas as informações da instância de jogo que está a decorrer. Como só decorre um jogo de cada vez, não haverá mais do que um objeto deste tipo.

# *Null Object*

Para evitar que algo no nosso programa seja completamente vazio, faremos com que um objeto que seria equivalente a *null* tenha certas propriedades predefinidas, para que ainda possa ser utilizado como instância do programa.

# *Iterator*

Para o caso de implementarmos a funcionalidade de o nosso jogo lidar com mais do que uma forma ao mesmo tempo (atirar duas formas simultaneamente, por exemplo), usaremos este padrão de desenho para percorrer a coleção onde as formas estarão armazenadas.

# *State*

Sendo o nosso programa efetivamente um jogo, este padrão será importantíssimo para as transições de estado que se vão efetuar dentro do *Game Loop*, que será o *singleton* que representará o estado atual do jogo.

*Frameworks*

Color Ninja

# LibGDX

Com base nos exemplos dados, em relação a projetos de estudantes do ano passado, decidimos usar a framework LibGDX para a criação do nosso jogo. Esta framework tem características que a tornam apropriada para a programação de jogos que utilizem tanto a plataforma Android como desktop. Além disso, permite incluir bibliotecas (como a Box2D), que facilitam a implementação de um “motor” de jogo à medida do que queremos.