

## Design plan “secret features”

The new “secret” feature chosen is attack B. This feature is an extension of the games bullet collision detection with extra behavior. More specifically, larger sized bullets are shot and eventually explode after colliding with edges/and or borders of the game frame; smaller bullets are expelled from each explosion. In order to successfully imitate attack B into our project design, impacts to our existing design, as well as useful patterns concerning the new feature should be discussed and planned out.

The current architectural design for handling bullet collision detection involves classes ‘Bullets’ and game component ‘Game1.’ Refactoring plans are still ongoing to provide the collection of all objects in the game (such as bullets) an entity subsystem decomposition to increase abstraction.

It might be necessary to incorporate a new class say ‘BigBullet’ to differentiate the different types of bullets (big/small). With our current implementation, randomly assigned x and y velocities are attached to instances of bullets letting them essentially randomly traverse the playing field; this is similar to the aftereffects of attack B’s big bullets exploding and releasing smaller bullets. In terms of collisions, we currently have an entity-to-entity collision check to determine whether an entity (player/enemy) has collided with a bullet. This however appears different with attack B; collision is determined when the bullet collides with a screen edge. As stated above, it might be beneficial in this case to add a big bullet type class that inherits from our base bullet to account for this slight behavioral change.

With all of this in mind, it makes sense to consider incorporating a new pattern into our design to handle the new behaviors featured in attack B. Currently our bullet entities are represented as model components in the Model-View-Controller (MVC) pattern. However, structural patterns Composite and Decorator also seem applicable here since additional responsibilities need to be implemented in order to handle the functionality of different types of bullets. In the case of a Decorator pattern, it would make sense to utilize the ‘BigBullet’ class as a concrete decorator to extend the functionality and behavior of the bullet entity. This may include additional operations to detect bullet collisions with an edge and checking the state of the bullet (dead or alive).