

**Secret Feature Implementation**  
**Priyanka Ghosh Dastidar**  
**CPTS-587**

Pattern that can be used to implement (A) in the game design could be the State Pattern. The State Pattern is a behavioral design pattern that allows an object to alter its behavior when its internal state changes. In the context of a game, the State Pattern could be used to represent the different states of the character during combat, such as attacking, defending, or evading.

Created an LaserStrategy class that defines the basic properties and methods that all states should have with methods :- setLaserMovement(String movement), setLaserTexture(String filename) , laserFire(float deltaTime, Rectangle hitBox, List<EnemyLaser> list, List<EnemyLaser> heavyList) and boolean canFire().

Then I have created concrete classes for each of the different strategies that the enemy can be in during combat. For example, there is LaserStrategyNormal class, LaserStrategyCircle class, LaserStrategyString class, LaserStrategyHeavy class .

I have created a LaserBinding class that adds LaserStratedy from JsonFiles and the same is displayed in the game engine during enemy attacks.

UML diagram :

