Programming project

Name: Ahmed Talaat Nosser ID: 05

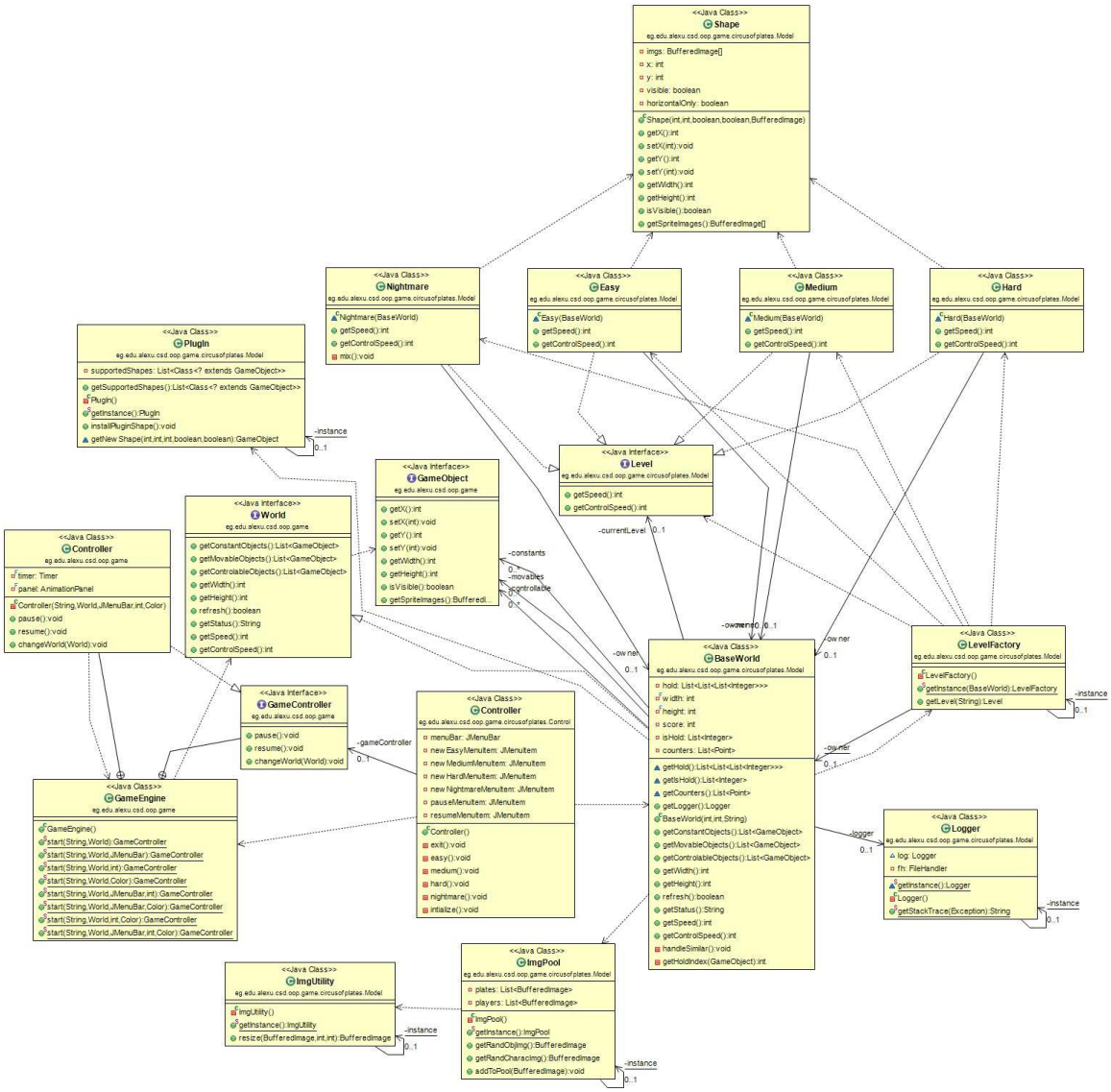
Name: Salma Ragab Gad ID: 32

Name: Mohamed Zakaria ID: 53

Project describtion:

- Winnie and tigger tries to catch the falling objects.
- When they catch three same-colored objects, the objects disappeared and the score increased by one.
- There are four levels:
  - Easy level: there is one hero and the objects fall slowly.
  - Medium level: one hero, the objects fall faster and number of them increased.
  - Hard level: there are two heroes and objects fall faster.
  - Nightmare level: two heroes, objects fall faster and the objects change their color.
- The game is over when objects hit the roof, then all objects fly to the roof.

Class diagram:

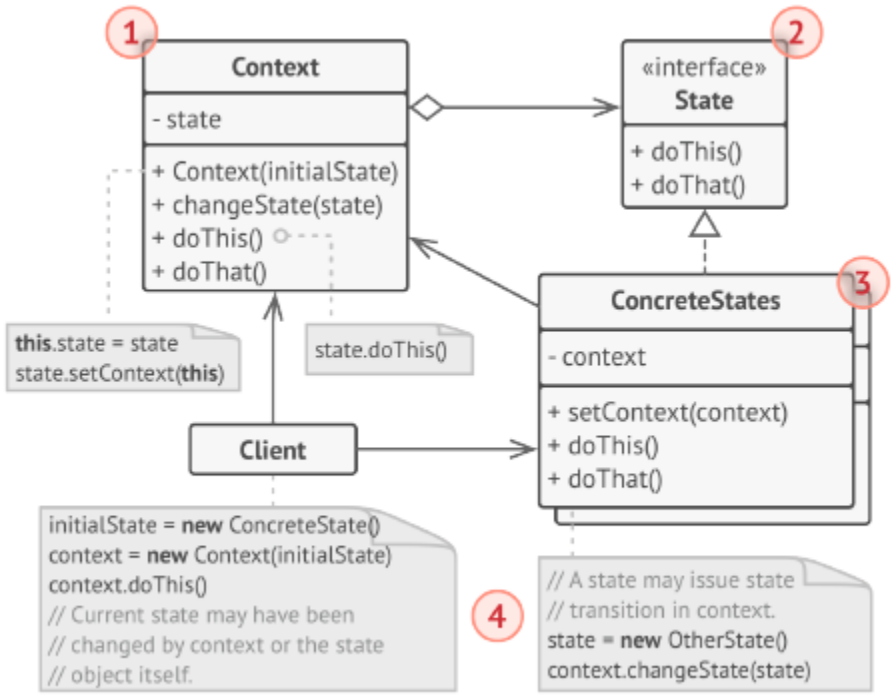


Sequence diagram:

Patterns:

1. State:

- Usage:
  - The context is the (BaseWorld) class.
  - The interface is (level) interface.
  - The concrete states are (easy, medium, hard and nightmare) classes.
  - We use it so we can add new levels without breaking the open close principle.
- class diagram:



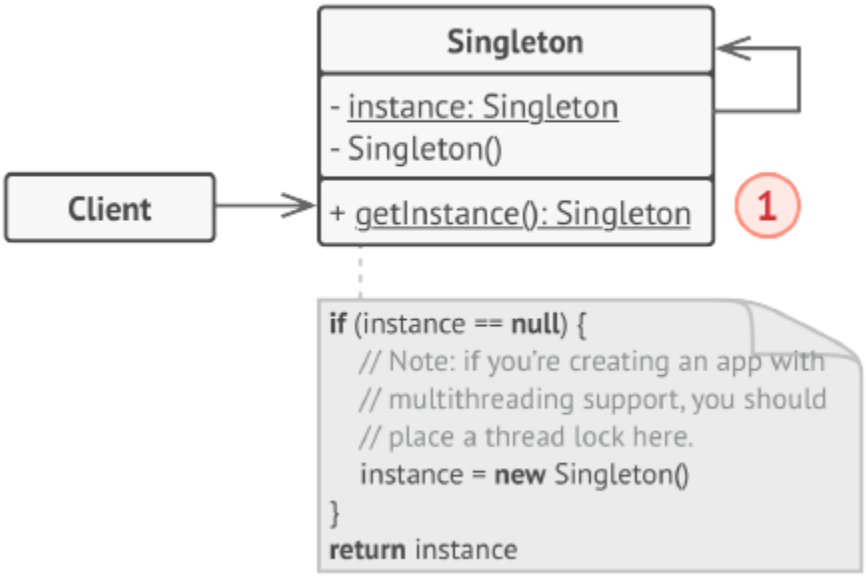
2. Singleton:

- Usage:

  -In the logger class, we need only one logger along the program.

  -We made the constructor private and the method get instance return an object of the class.

- class diagram:



3. Dynamic Linkage:

- Usage:

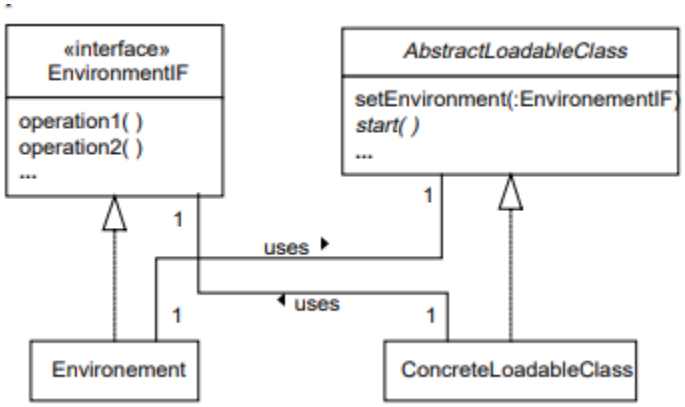
  -To enable the user to load a shape class that implements the game object interface.

  - we made class (plugin) to load the external shape.

  - the abstract loadable class is the game object interface.

  - the loadable class is packed inside the jar.

- class diagram:



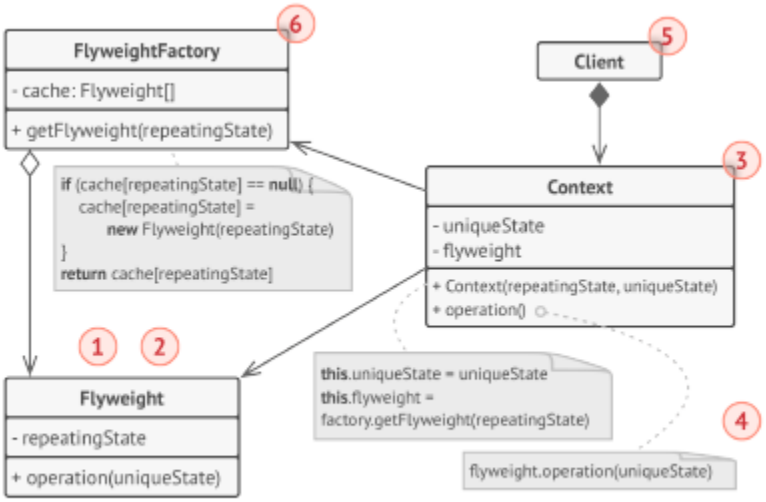
4. Flyweight:

- Usage:

  -we use it to save the memory: instead of creating new image object for each movable object, we make similar objects share the same image object.

  - Flyweight factory is the (imgpool) class.
  - Flyweight is the (shape) class.

- class diagram:



5. Factory:

- Usage:

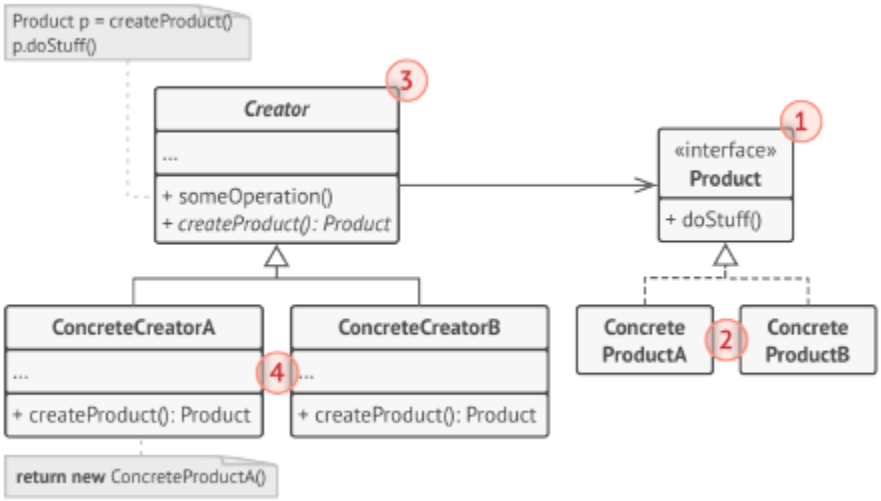
  -we use it to separate level construction code from the code that actually uses the level.

  - the product interface is the (level) interface.

  - concrete products are (easy, medium, ….) classes.

  - the creator is (level factory) class.

- Class diagram:



6. Pool:

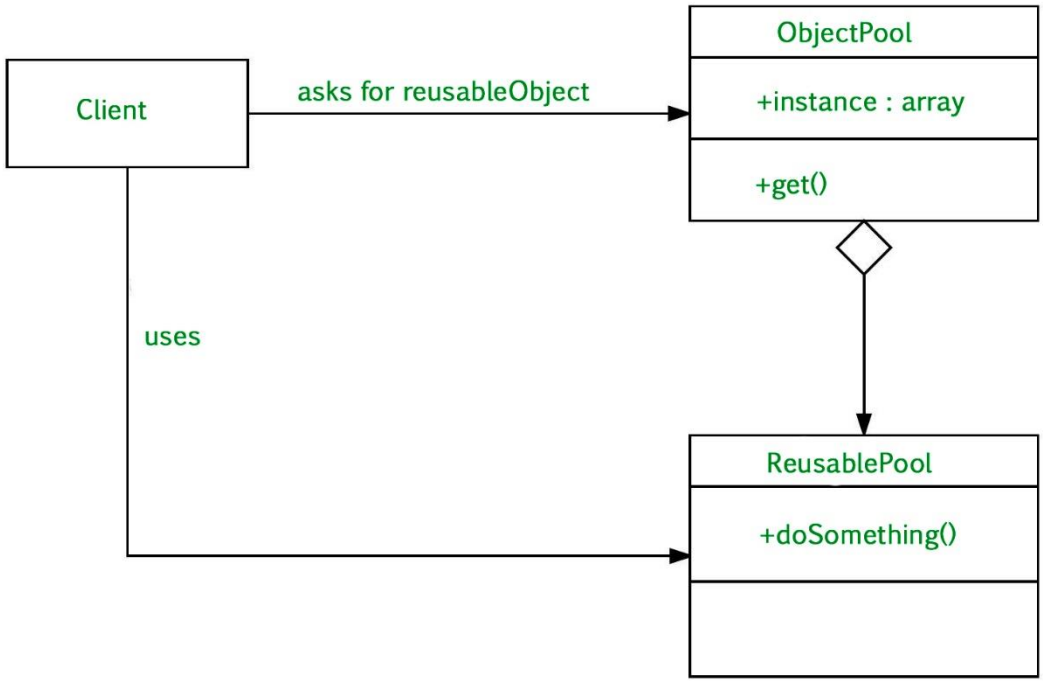
- Usage:

  -we used it in (ImgPool) class to save the images creation and loading time.

  -object pool is the (ImgPool) class.

  - the reusable pool is the (BufferedImage).

- Class diagram:

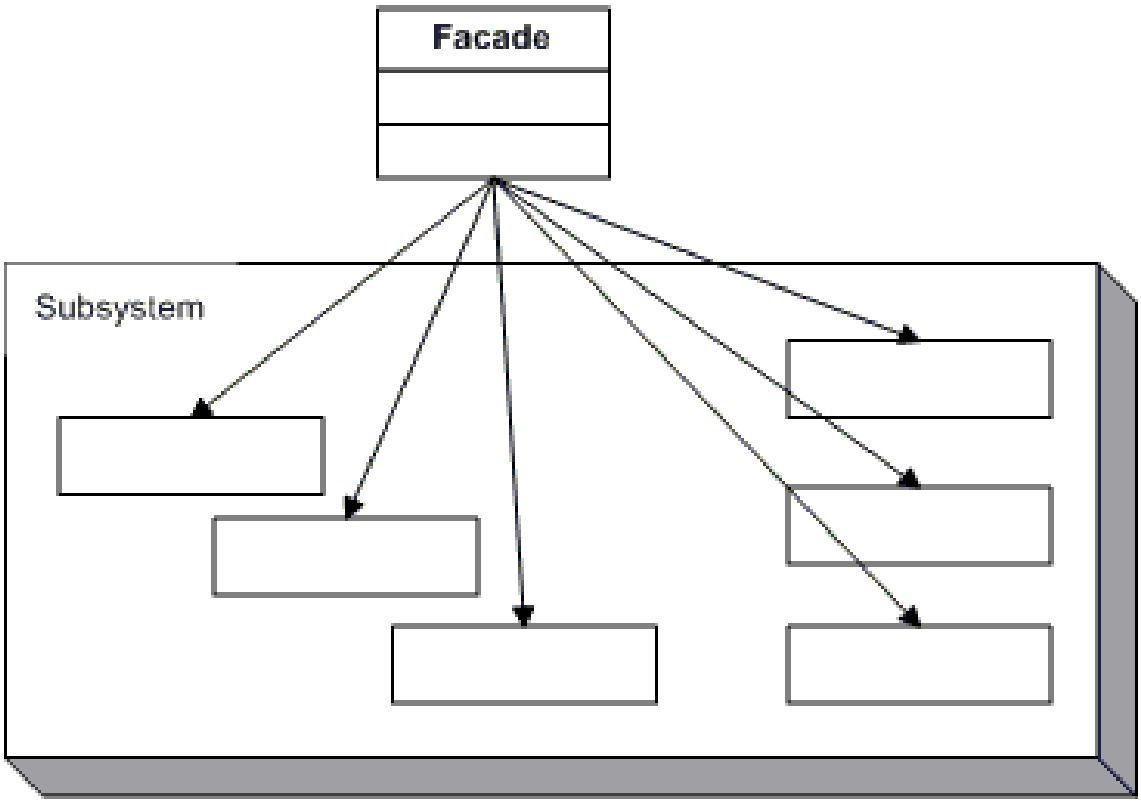


7. Facade:

- Usage:

  Used to encapsulate a complex structure of code and providing simpler interface to deal with in different classed like CustomLog, ImgUtility.

- Class diagram:



8. Iterator:

- Usage:
  Built-in java pattern, used in iterating over lists in different classes like BaseWorld.

Design decisions:

- There are 4 levels of difficulties, the user chooses the level at first.
- The design follows the (MVC) pattern.
- The user can add plugin to the game by adding jar file to the plugin folder contains class that implements (GameObject) interface along with its resources.
- New game can be started while the game is running or after the game is over and the level is chosen from the internal menu.