Final Lab Report

Students:

* Youssef Mohamed Attia (8164)
* Karim Mohamed Almahrouky (8237)
* Abdelrahman Shabaan (8041)

A computer screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedFactory & Singelton Design Pattern:

We used the factory design pattern to provide a decision-making class that returns one of several possible concrete subclasses (Bomb ,Clown ,Plate ,Cup) of an abstract shape class depending on the input provided.

Singelton design pattern was used to ensure that only one instance of the shape Clown was created by making the class constructor private and calling it after checking if an instance is created yet.

Observer Design Pattern:

A diagram of a class

Description automatically generated

Our observer design pattern consists of 2 interfaces(observer, observant) , a concrete observer ”Main window” and a concrete observant ”Gamestart” so when the Gamestart state changes the observer class main window is notified and automatically updated.

Iterator Design Pattern:

A screenshot of a computer

Description automatically generated

Our Iterator design pattern consists of 1 interface class “Iterator” , a concrete Iterator ”ShapesData” and a client ”Game” so we used the iterator pattern to loop over all shapes collected in either clown hands or to edit those collected shapes while keeping the class private, so no changes are needed in client code.

Strategy Design Pattern:

A screenshot of a diagram

Description automatically generated

We used the Strategy design pattern to provide different algorithms for a specific task (game level) and to facilitate adding another level at any time without interfering with the main game code in the Client class (Game).