

# FRUIT NINJA



# **The game design is described as follows:**

The player is made using **Singleton design pattern** and it contains the player's current score, best score and remaining lives.

When the player chooses new game, an object of ClassicScreen is created through GameScreen with the aid of **Command design pattern** depending on the player's choice. GameScreen also calls GameEngine where the game itself takes place. GameEngine is the class responsible for making the game objects, fruits and bombs, appear to the player.

GameEngine then uses DifficultyState class to specify the difficulty level the player should face. Difficulties are made using **State design pattern** and each difficulty vary according to the number and speed of the fruits and bombs.

GameEngine then gets the game objects from their specified factory using **Factory design pattern**, FruitFactory or BombFactory, and make them appear on screen to the player using transitions. Each fruit and bomb implements GameObject interface to implement its methods accordingly.

All the player's data are stored in an xml file whenever the player pauses the game or his/her score becomes a multiple of 20 as checkpoint so that he/she can continue his unfinished game any time after closing the game. This was made by the aid of **Memento design pattern**.

Throughout the whole game design we made sure to apply **SOLID concepts** and **MVC**. In order to achieve this we divided our code into different packages, the most important two of them are the **Logic package** which contains all the logic code and classes made with the aid of design patterns and the **View package** which contains all the classes that contain nothing but GUI.