Programming Project 4 Report

Collaborators:

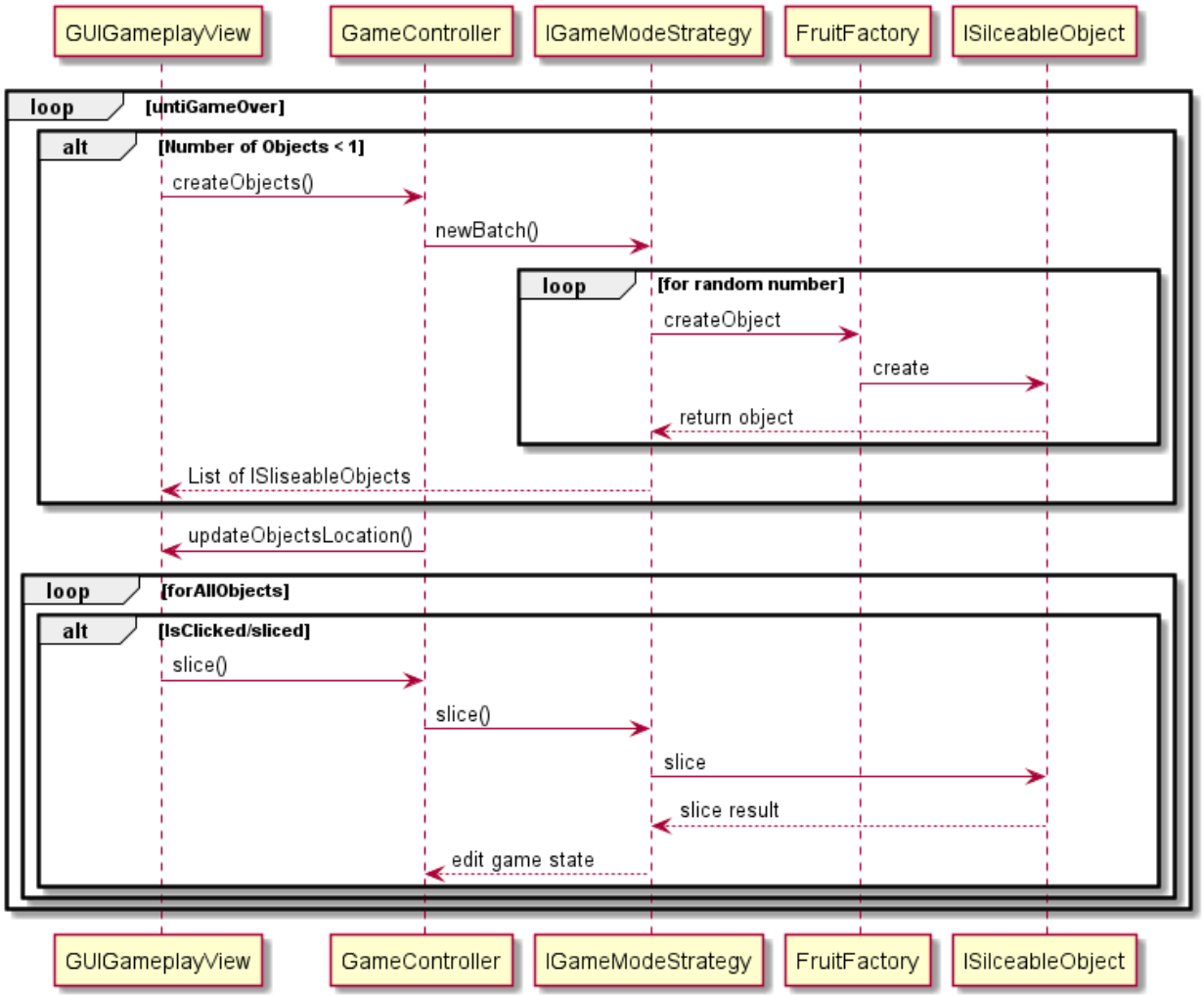
Karim M. Elsayad	6023	Bishoy Refaat	5804
Mahmoud Essam Sharshera	5786	Mohannad Ayman	5936

Diagrams

Class Diagram:

Diagram is provided in a separate folder with full resolution.

Sequence Diagram:



Design

There are 3 main components to the game that follow an **MVC** design pattern:

- The game engine, Controller, It controls the data flow into model object and updates the view whenever data changes. It keeps view and model separate. The Controller acts on both model and view.

- The GUI, View, which only uses the methods provided by the game engine to communicate with the game logic and models.

- The Objects, Models, who communicate with the controller and change its data through strategies.

Design patterns and their intent:

Factory: Provide an interface for creating families of related or dependent objects without specifying their concrete classes. It allows the creation of sliceable objects (fruits, bombs) subclasses when needed for each throw, it takes a random integer and choses a returns a random object depending on that integer, creating a random feel.

Command: used to encapsulate information needed to trigger the saving and loading (FileCommand) actions to issue requests to objects without knowing anything about the operation being requested or the receiver of the request.

Singleton: ensures only one instance of the Controller class and provide a global point of access to it, and thus it helps coordinate actions
across the project.

Strategy: It define a family of algorithms, encapsulate each one, and make them interchangeable. Strategy lets the algorithm vary independently from the clients that use it. Used to create multiple game modes with each strategy represent a game mode with its own rules, flow of sliceable objects, reaction to objects being sliced or falling off screen, and win/loss conditions. Each family is a game mode, and the game controller uses them independently.

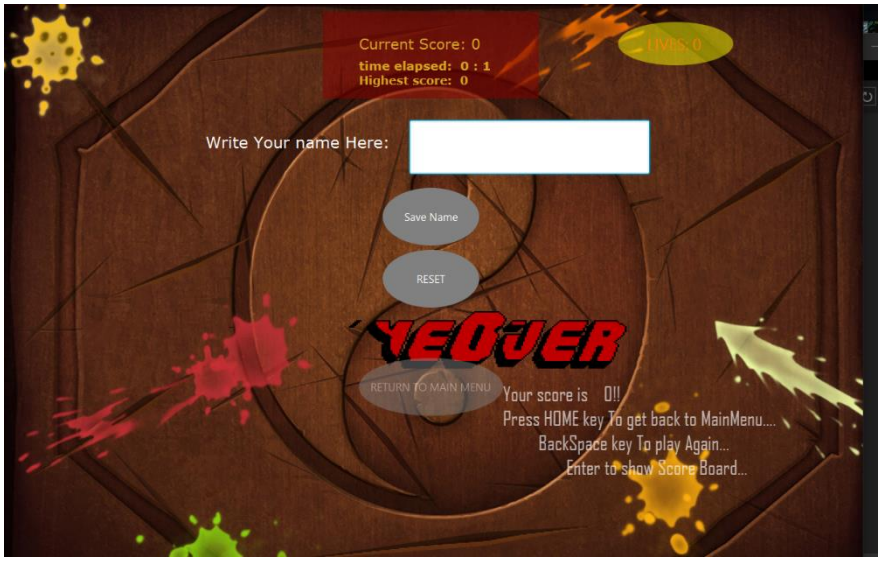
Decorator: Attaches additional responsibilities to an object dynamically. Decorators provide a flexible alternative to subclassing for extending functionality. Used to change difficulty of a strategy by modifying the amount of objects thrown.

Observer: used to update the score label in real time. When one object changes state (The score), all its dependents are notified and updated automatically (The labels which show the score).

Snapshots of GUI:

Main Menu

During Gameplay



Game Over screen

How To Play

1- When you launch the game, choose a game mode, or click on “Score board” to the highest saved scores. When selecting a game mode, you’ll be promoted to choose difficulty.

2- During gameplay, simply move the mouse across a fruit to slice it.

3- Beware of bombs in Classic mode! They reduce your lives. Fatal bombs causes an instant game over. In Classic modes, not slicing a fruit before it falls off screen causes you to lose a life. There is a special fruit that can help you recover 1 life! Be sure not to miss it.

4- There are no bombs in Arcade mode, but the game ends right after 60 seconds, so try to get the highest score you can. There is a special fruit that gives you a 10 second boost, don’t miss it!