Fruit Ninja
Project Report

By: Abdelrahman Ahmed Shawky (6353)
Zeyad Hesham Saad Mohamed (6601)

Introduction:

This project is an implementation of the game “Fruit Ninja” using java. It is not a copy of the game but only a trial in order to develop further coding skills in java. It is not to be used commercially with all rights reserved to Halfbrick Studios.

It is a single player game in which the player slices fruits with a blade controlled by mouse swipes. As the fruits are thrown onto the screen, the player swipes the mouse to create a slicing motion, attempting to slice the fruit into two halves. A player tries to avoid bombs which either end the game or make player lose a life according to the type of bomb.

Description of the design:

The design follows object-oriented programming concepts such as polymorphism and inheritance. Firstly, RegularBomb, FatalBomb, SpecialFruit1, SpecialFruit2, Fruit1, Fruit2 and Fruit3 are all inherited from interface “GameObject”. Also, every fruit has two states, either sliced or unsliced. That is why, the interface “FruitState” is implemented by two classes “SlicedState” and “UnslicedState”.

As for the actions done by the code, “GameController” is the main controller that controls all game actions and therefore implementing “GameActions”. It also contains “Save”, “GameModel”, “MementoCareTaker”, “RemoteControl” and “FactoryProvider” classes.

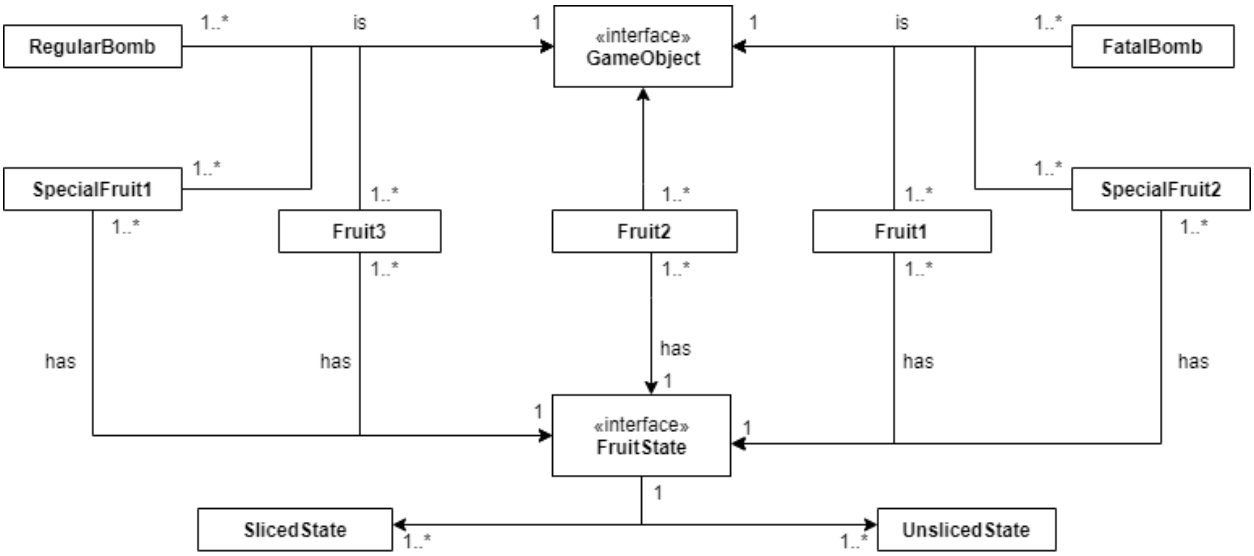
Plenty of design patterns were used to aid and simplify the code and ensure SOLID principles are thoroughly applied throughout. As an example, singleton classes were used for “GameController” to ensure only one controller exist to avoid any misuse of code. Also, the use of the factory method which simplifies the creation of objects and aids in

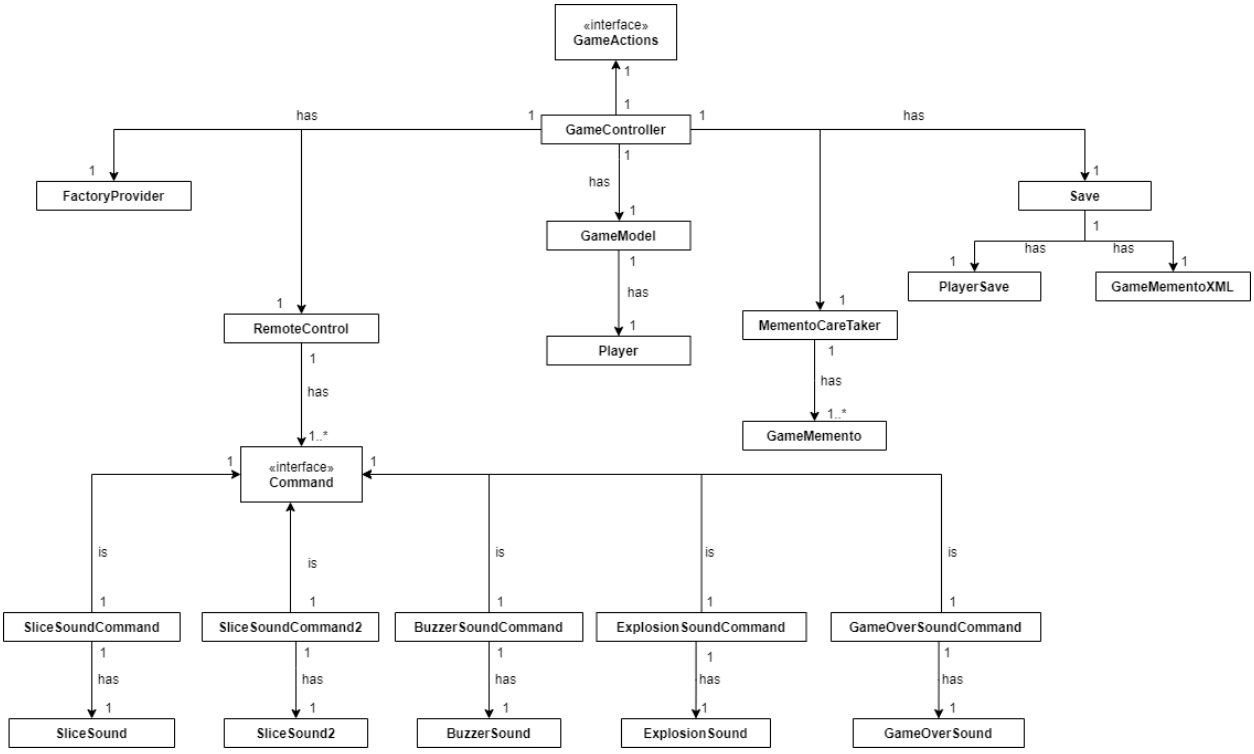
maintaining all object creations in one place for further extensions. In addition, the use of command which organizes the code by adding every task to its own command with the aid of “RemoteControl” as seen in the class diagram. Moreover, the observer pattern is used to maintain consistency throughout the code. For example, when one object changes all its dependents are notified and updated accordingly.

Continuing in the design patterns, the state pattern is used to identify the two states of the fruits which are either sliced or unsliced in order to change their image accodingly. Finally, the adapter pattern is present to adapt one object to another object which allows objects to work together that could not before due to incompatible data types or interfaces. This is represented here in adapting of classes in order to save to an XML file or reading from an XML file.

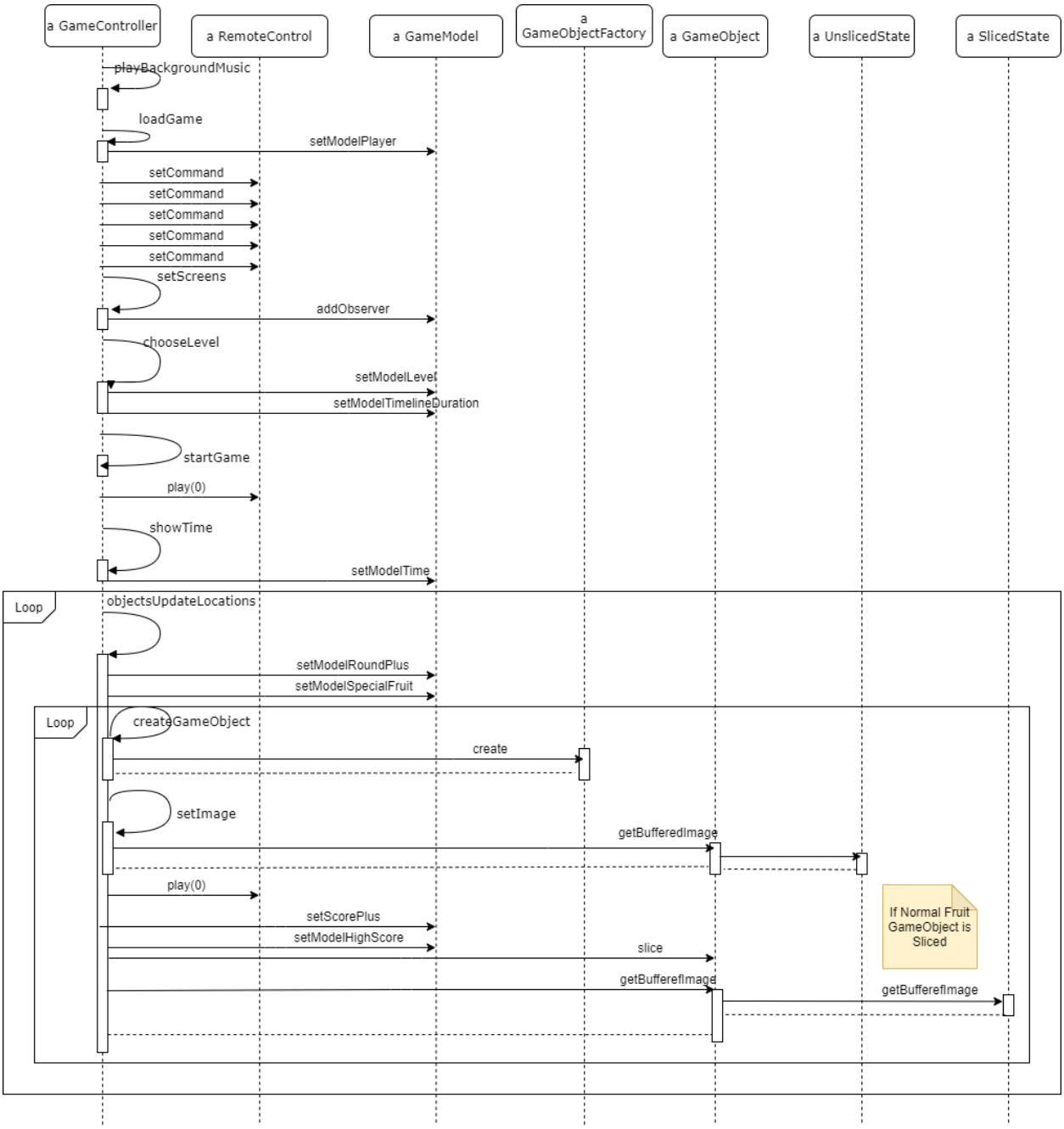
Combing all these design patterns with simple yet powerful OOP concepts like polymorphism was the main infrastructure of this project. Those simple OOP concepts are what made this project possible and led to this final application.

Class Diagram:





Sequence Diagram:



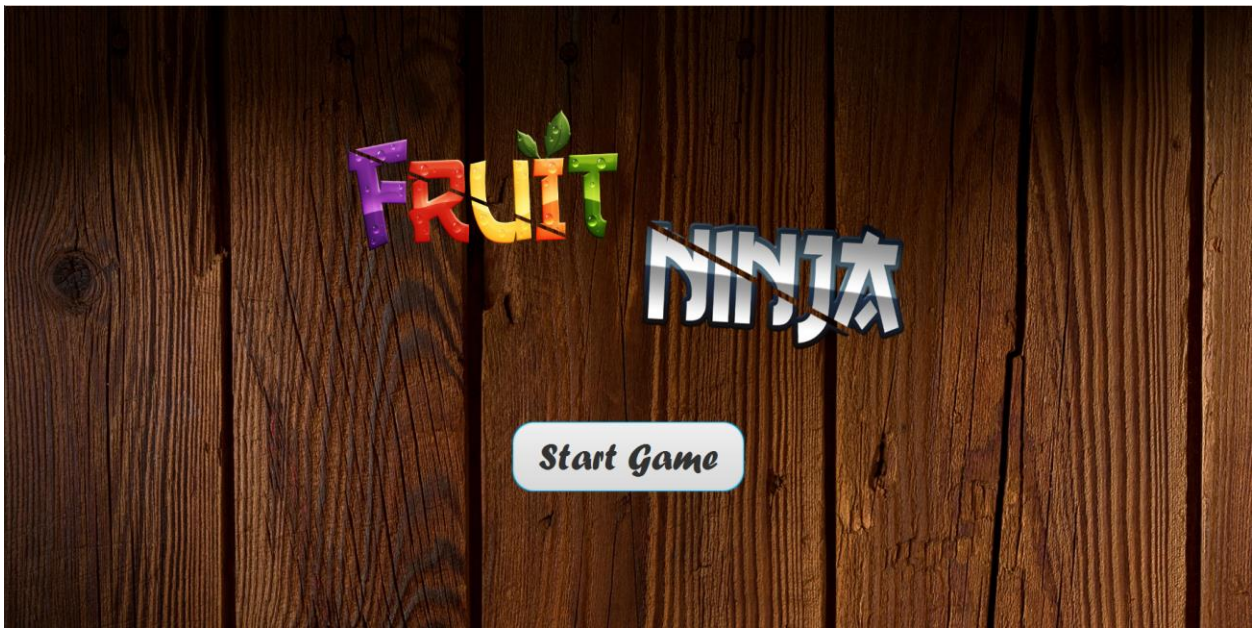
Design Patterns:

1. MVC:
   MVC design pattern is the main structure of the application and it is embedded in the infrastructure of the project. It is divided into Model-View-Controller whereas the GameModel class represents the “Model” of the MVC and the GameController class represents the “Controller” and all the GUI classes represent the “View”. The GameController controls the game in collaboration with the GameModel where all the game data is sufficiently stored.
2. Singelton:
   Singelton design pattern is used throughout the application for any class where only one instance is needed. This was to avoid any misuse of classes and ensuring object creation is at its optimal.
3. Factory:
   Factory design pattern was used for two tasks in this application. Firstly, it is used for game object creation such as, fruits and bombs. Also, it is used for screen creation of the GUI for instance, GameScreen and LevelSelectScreen. It is mainly used to ensure proper creation of these classes and maintain creation code in one specific area for any further extensions.
4. State:
   State design pattern is mainly used for the fruit game objects. Each fruit has two states a SLICED state, as well as, an UNSLICED state. The UNSLICED state allows the method getBufferedImages to return the unsliced fruit image. On the other hand, the SLICED state allows the same method to return the sliced fruit image instead. This allows easy manipulation of sliced and unsliced fruit images through calling the same method.

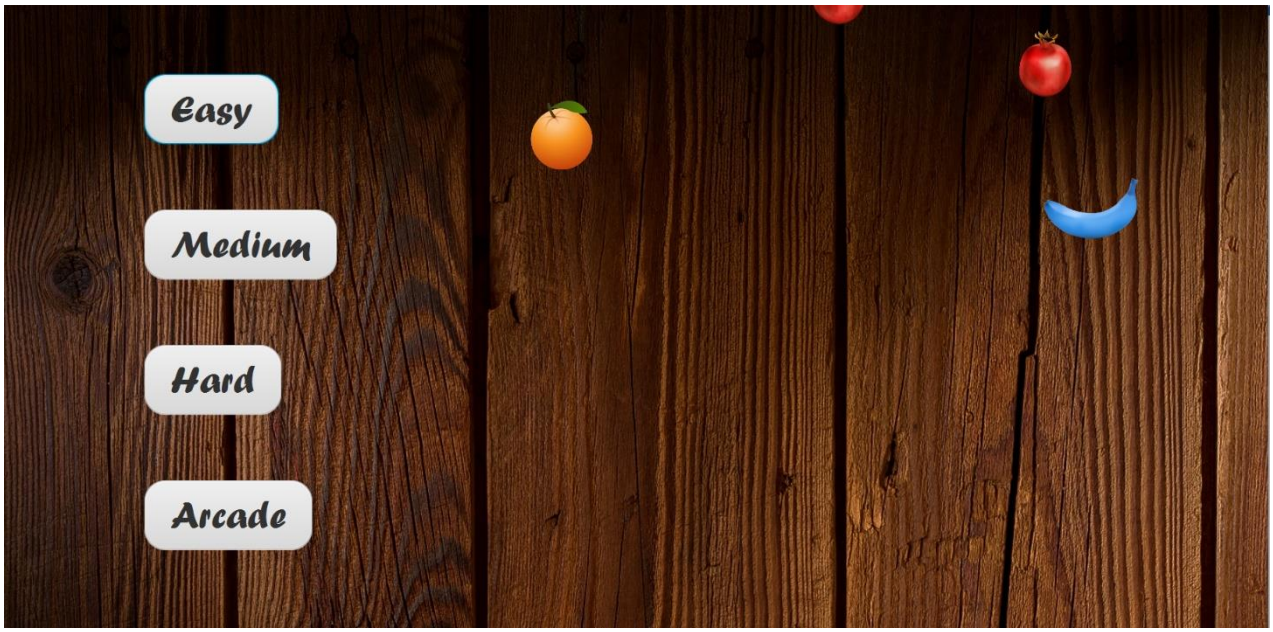
5. Observer:
   Observer design pattern is composed of two main parts which are the “Observer” represented by the GameScreen GUI class, as well as, the “Subject” represented by the GameModel class. Its main advantage is updating the GameScreen with the latest scores and high scores of the player while the game is being played. This is done through capturing any changes in the GameModel then notifying the GameScreen of these changes to update the data accordingly.
6. Memento:
   Memento design pattern is used for saving the game mid-play where a player can close a game whenever they want and continue where they left off once they restart the application. It is mainly composed of MementoCaretaker class responsible for creating and retrieving saved data, as well as, GameMemento class where the data is stored of a specific state of the game.
7. Command:
   Command design pattern is used in this application for sound effects, where each sound is triggered through a different command such as, slice sound, bomb sound and game over sound. This is done through the RemoteControl class where all the commands are set at initialization enabling use of the remote in any part of the application to trigger different sound effects.
8. Adapter:
   Adapter design pattern is used for converting the GameMemento calss to an XML class thus enabling saving to an XML file. This allows the code to be viable for further extension with different saving methods other than XML. Also, adapter is used for the same purpose to save the player’s high scores on every game mode, as well as, reading saved data from XML file.

Snapshots of GUI:

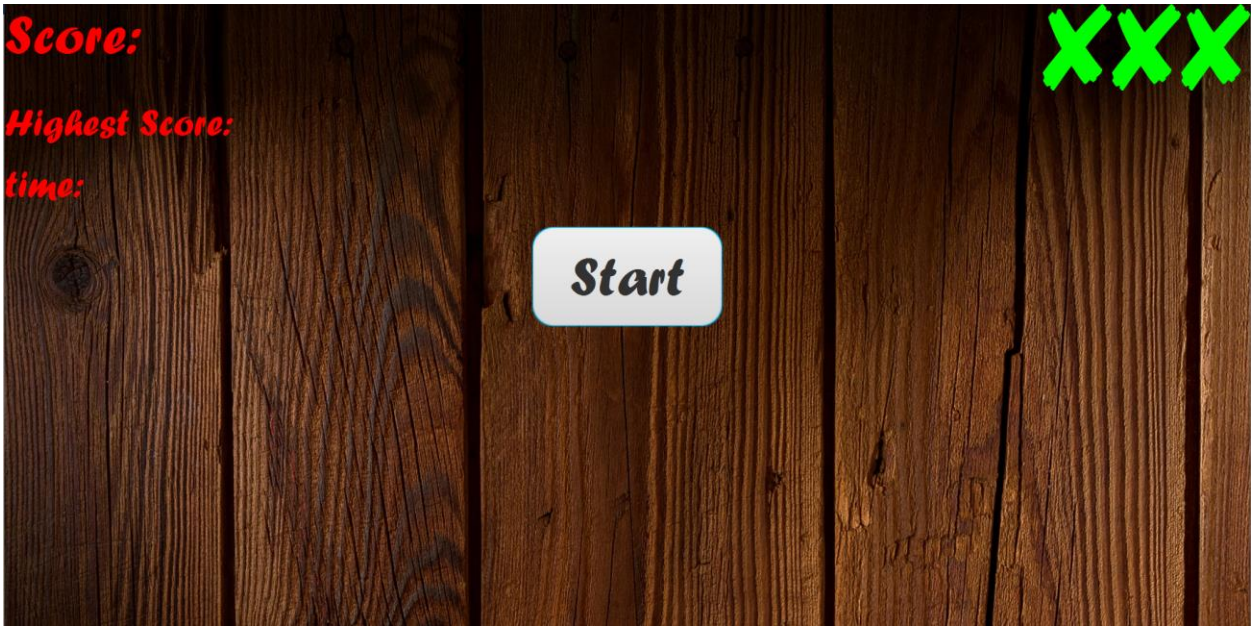
1) Main Menu:



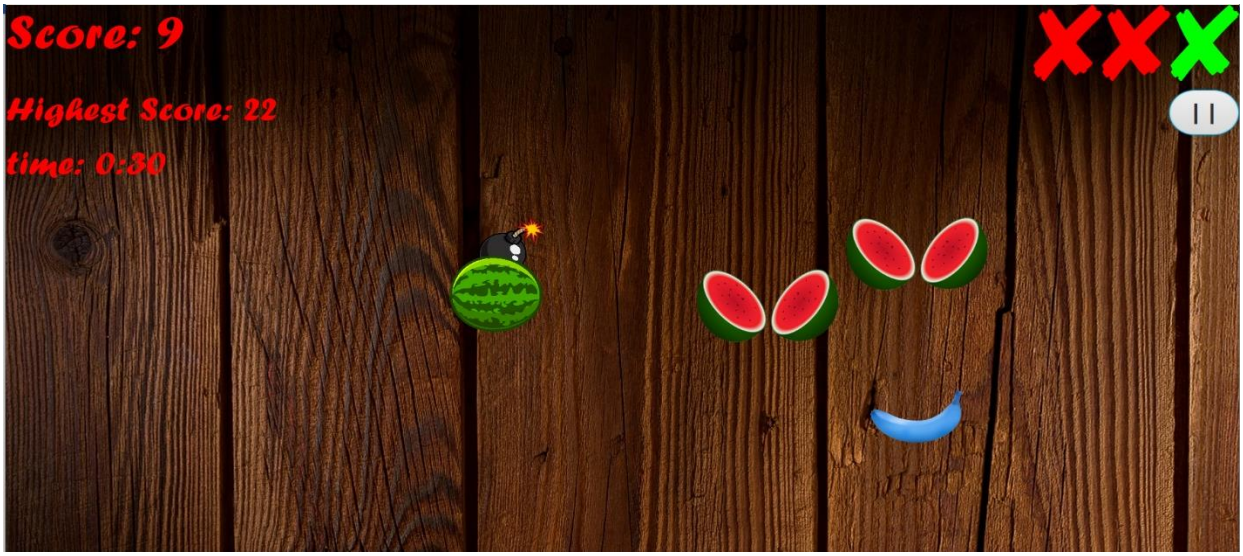
2) Different difficulties and game modes menu:



3)After selecting game mode/difficulty:



4)After pressing “Start” snapshot which shows the high score, current score, lives remaining, time since start of the game and Sliced Fruits:

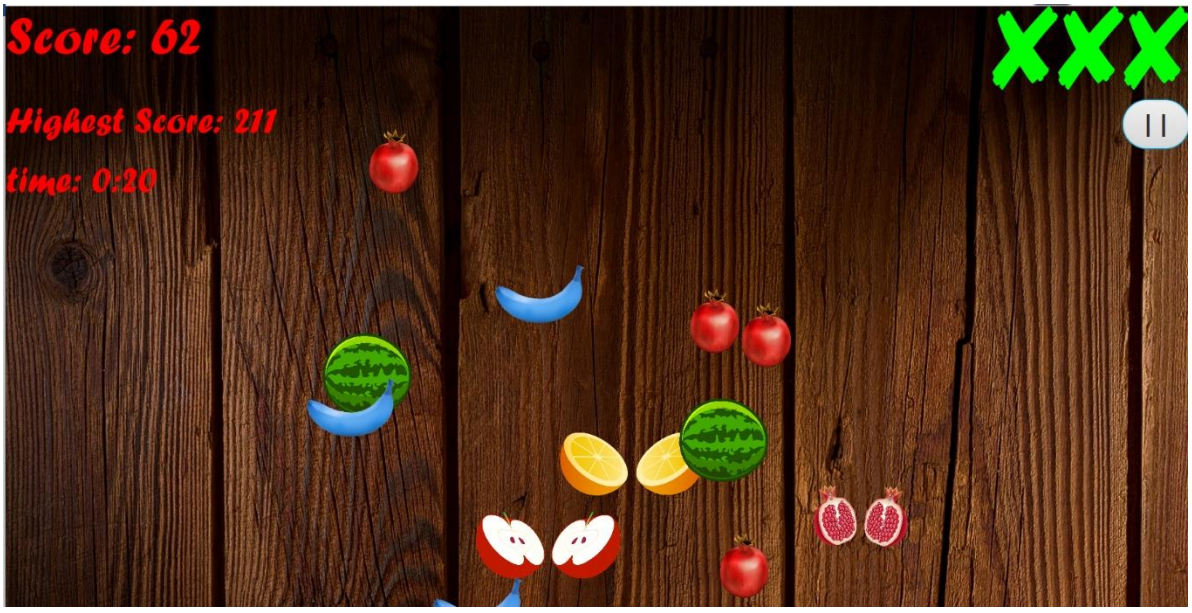


5)Pausing in the middle of a game:

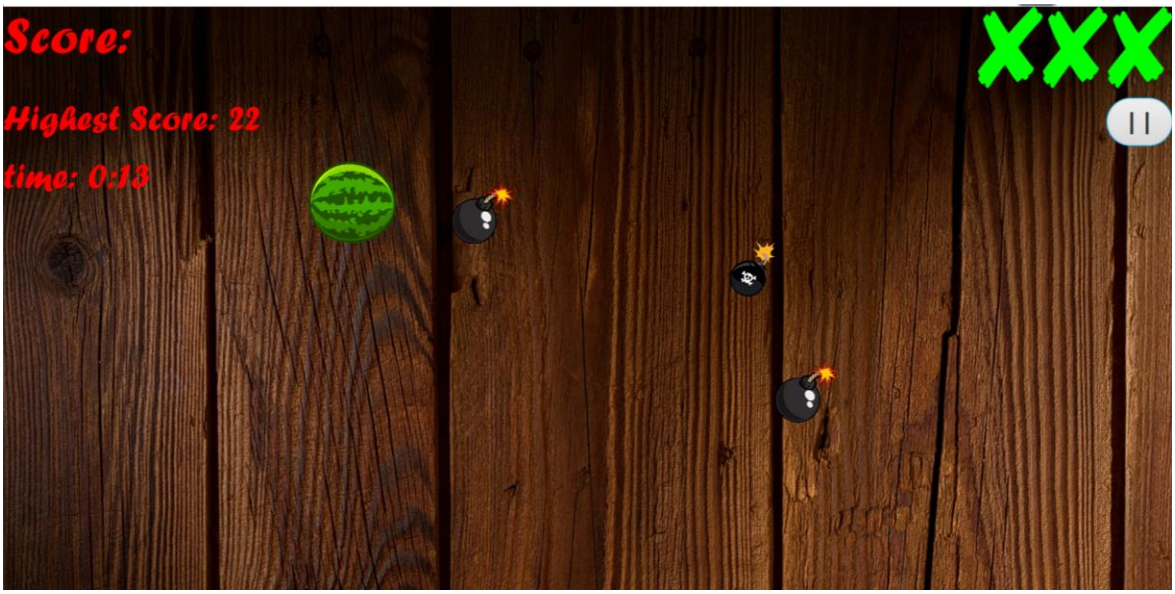
6)Screen when losing a game showing high score and current score:

7)Snapshot when getting a new high score:

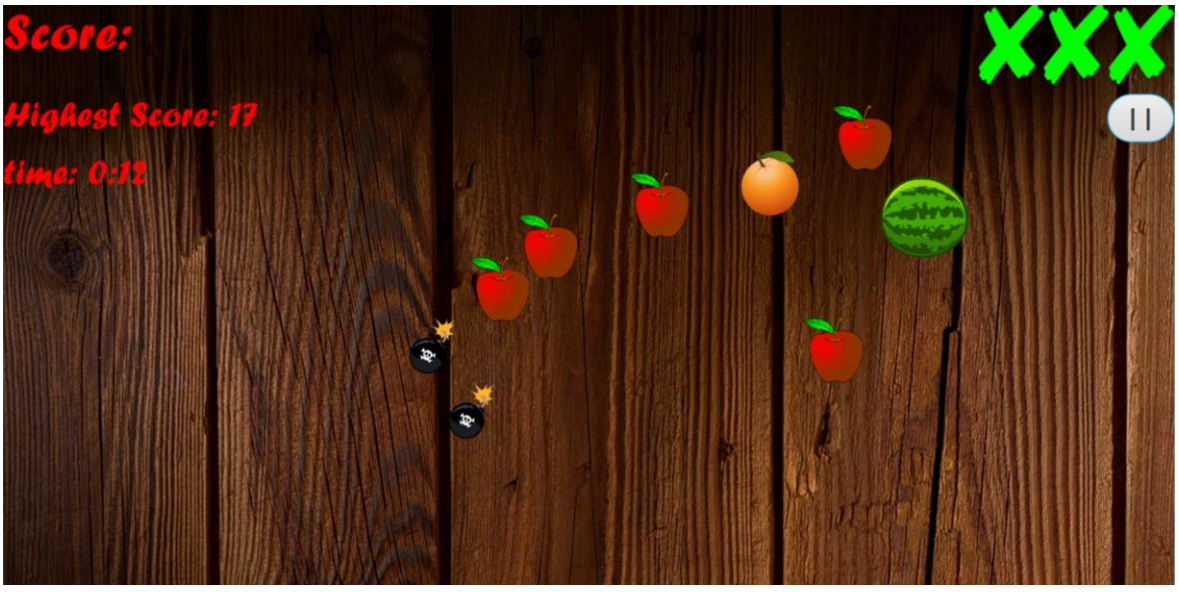
8)Arcade mode:



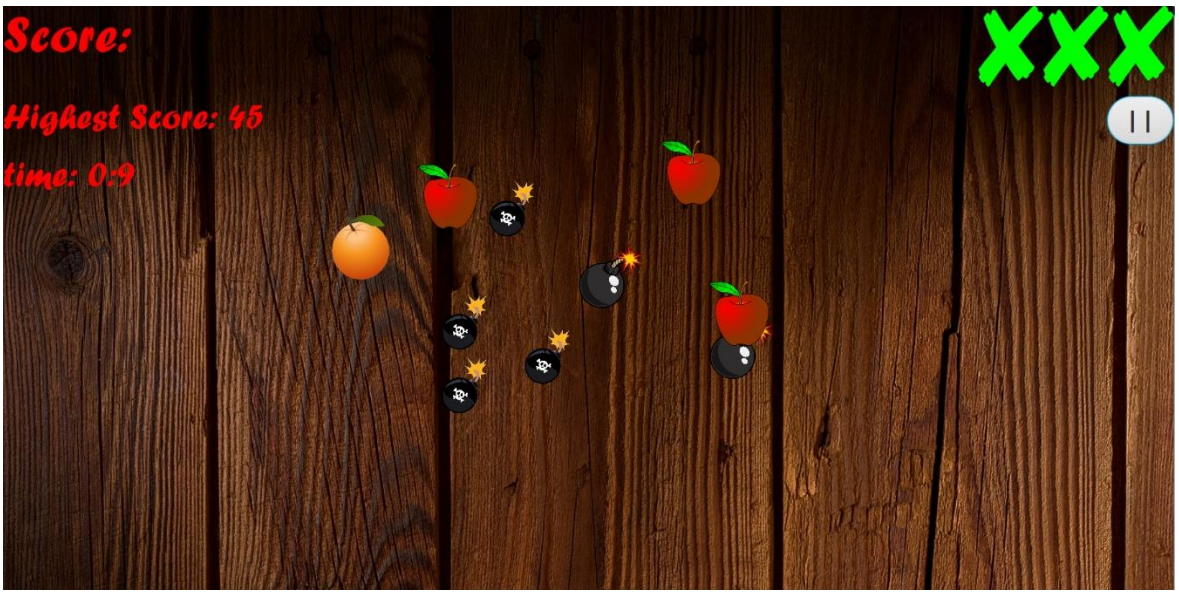
9)Easy mode with minimal fruits appearing:



10)Medium Mode:



11)Hard Mode with maximum fruits and bombs and faster speeds:



User Guide

Firstly, to start the application the user has to simply double click the self-executable icon named ”FruitNinja.bat” which will automatically start the application. A menu will appear with only one button which is “Start Game”, pressing this button will take the user to another screen which displays all the available difficulties and game modes which are:

1) Easy (for beginners)
2) Medium (for users who gained a bit of experience playing the game)
3) Hard (the most difficult setting)
4) Arcade (different game mode than usual that has no bombs or lives and gives the user a minute to slice as much fruits as possible)

After the user chooses the most suitable game mode/difficulty for them, they will be shown a button “Start’ which when pressed starts the game. Upon starting the game, the user will be shown the high score, current score and amount of time passed in the top left and number of lives left in the top right. In order to succeed in the game, the user has to slice the fruits while avoiding the bombs by clicking on the fruits with their mouse. There are two types of bombs that the user has to avoid clicking on, one bomb decreases a life while the other ends the game. The user can pause the game at any given time and exit or retry, the pause menu will be found in the top right. After the user has finished the game, the current score and high score will be displayed and if the user proceeds with pressing the screen they will be returned to the original main menu which displays when running the game and the user can play again or quit as they want.

There are three types of fruits each grant a player with one point but there are also two kinds of special fruits. The blue banana removes one cross adding a life already lost by the player. In addition, the pomegranate adds 5 extra point to the player score unlike normal fruits.