Fruit Ninja Game

User Manual:

When you open the game, first select the difficulty, then select the player or create a new one, then click play.

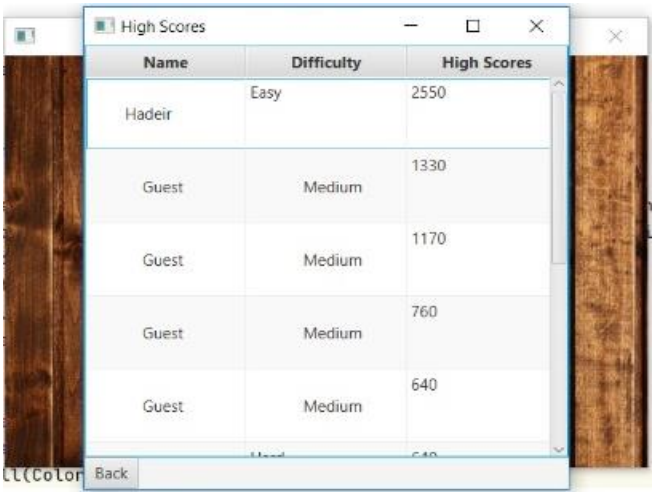
When the game starts, click the mouse button and drag to slice. Try to slice the fruits before they fall, if they fall down you will lose a life. You should not slice the bombs, if you slice them you will lose a life, or the game will end depending on the bomb.

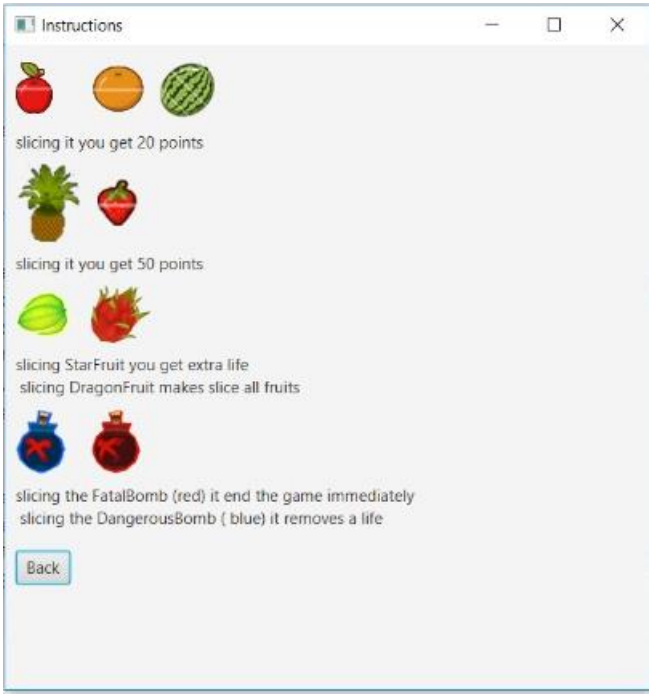
You can pause the game and resume any time. If you exit a paused game the game will be saved. When you finish your lives, you can restart or go back to main menu to change player or difficulty.

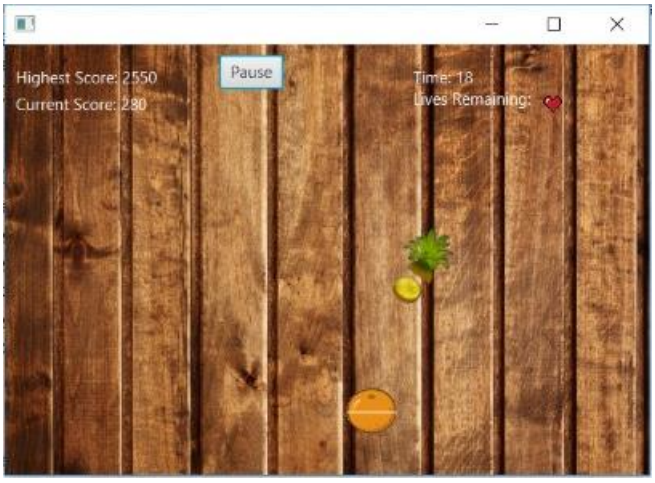
You can open the instructions to see what every fruit does when it is sliced.

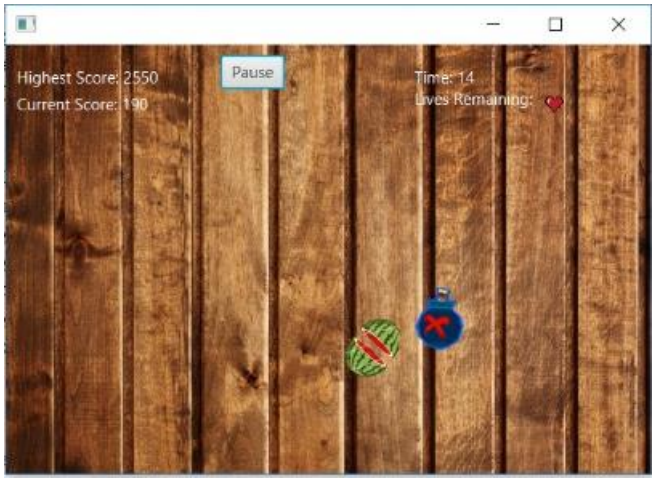
You can click on High scores to see top scores or exit to exit the game.

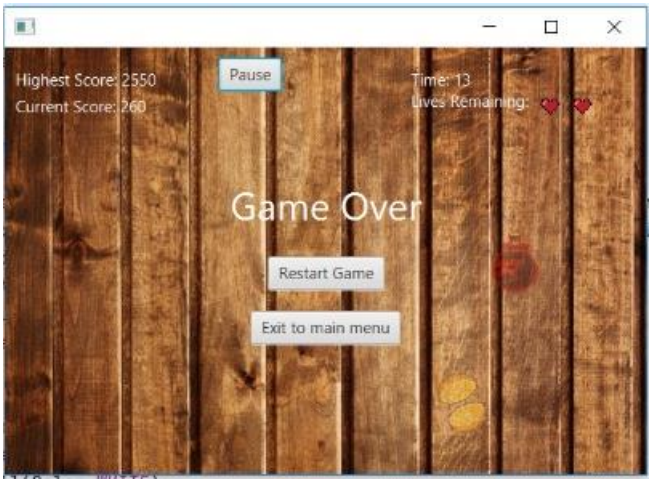
Snapshots of Game:

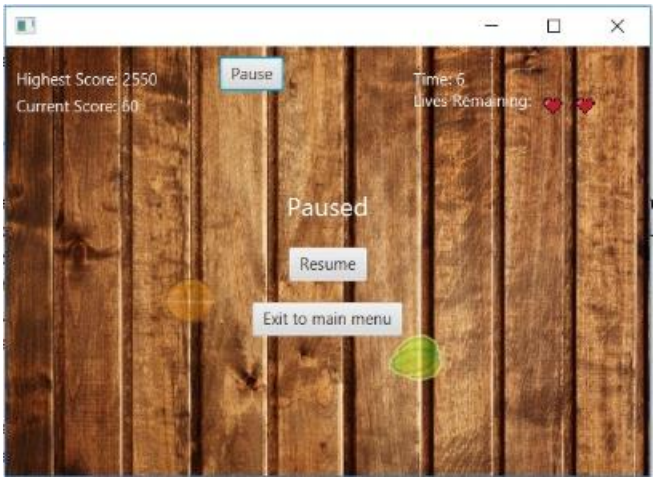


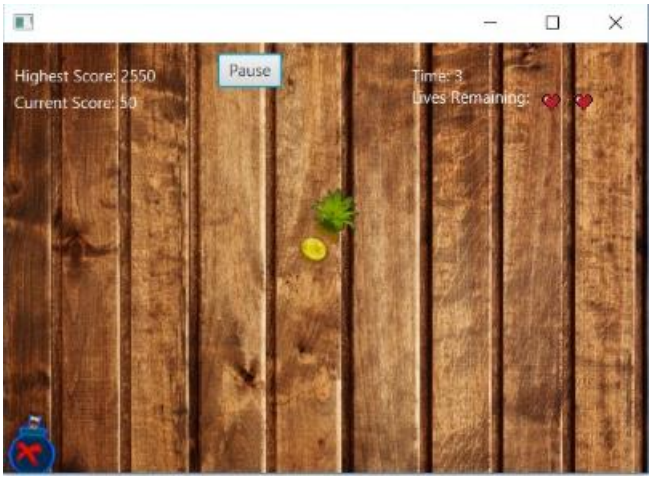


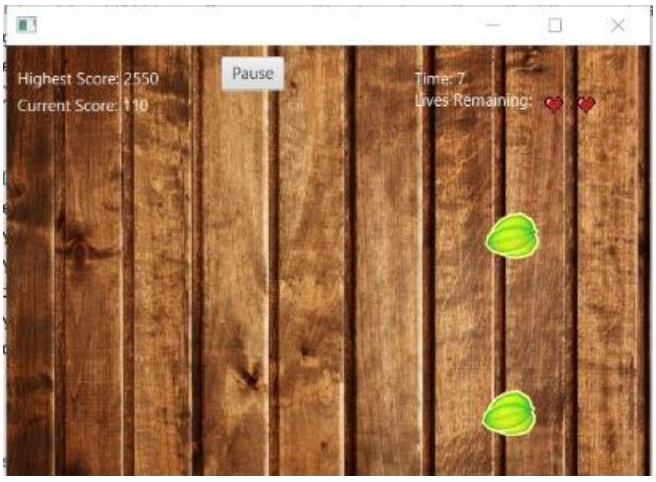

Highest Score: 2550
Current Score: 280
Pause
Time: 18
Lives Remaining:


Highest Score: 2550
Current Score: 190
Pause
Time: 14
Lives Remaining:


Highest Score: 2550
Current Score: 260
Pause
Time: 13
Lives Remaining:
Game Over
Restart Game
Exit to main menu


Highest Score: 2550
Current Score: 60
Pause
Time: 6
Lives Remaining:
Paused
Resume
Exit to main menu


Highest Score: 2550
Current Score: 50
Pause
Time: 3
Lives Remaining:


Highest Score: 2550
Current Score: 110
Pause
Time: 7
Lives Remaining:

UML Diagrams in separate PDFs

Description:

- The class ENUM contains enumerated values for the Game Objects. The GameController creates a random GameObject by generating a random number then calling the ObjectFactory to create the object corresponding to the random number. The time of creation is saved in the object.
- Once created the object location is set at the bottom of the scene at a random x location. The GameController calls the move() method of all objects on screen to determine the objects new locations based on the time passed since object creation.
- The GameContoller checks the slice region and calls the slice() methods of any object inside the region.
- Inside the Game, a timeline is used to call the GameController to create an object every certain time interval. Then a timer is used to update object locations each frame and pass the slice region using a mouse dragged event to the GameContoller.

Design Patterns:

1- Singleton:
    - Used in Game class and in RemoteControl class.
    - Game: to be able to easily get the game variables such as object list for actions on them and game scene height for use in max height calculation. Also only one instance of game should be available each run.
    - Remote Control: Only one instance of Remote Control needed so to avoid creating instance each time a command is executed.
2- Factory:
    - Used in ObjectFactory class.
    - Object Factory: to be able to create any GameObject instance given only name of object.

3- Command:
    - Commands: AddLifeCommand, RemoveLifeCommand, SliceAllObjectsCommand.
    - Used by certain objects to affect the state of the game when they are sliced.
4- State:
    - Used for Difficulty
    - Difficulty variable in Game changes the state of the game between (Easy, Medium, Hard). This affects the spawnRate of game objects and the speed of game objects.
5- Memento:
    - Used to save and load game
    - Class GameState stores the state of the game. The GameController can then store the GameState in a memento to so that it can be saved and loaded. The save method saves the game state in GameState.xml and is called when you exit from pause but the load method does not work so it is not used (could not unmarshall the enum object type).