CHAPTER I – INTRODUCTION

In this game project, I applied the knowledge learned in theoretical OOP class to make this game. Java is the only programming language we have used and I implemented some essential libraries such as Swing, and KeyEvnt, for game display, and game inputs.

In this report, I will provide a detailed explanation of my project. The structure of my project is as follows:

Chapter I: Introduction

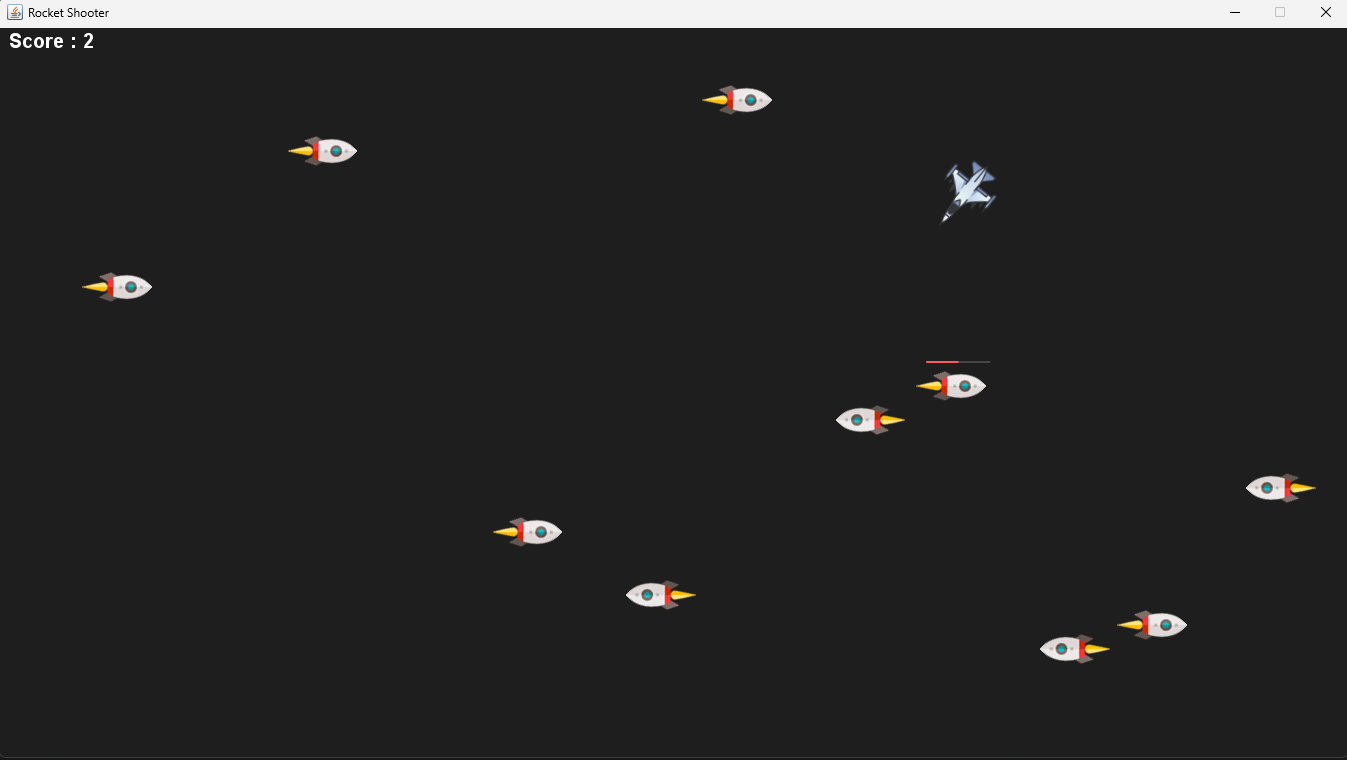
Chapter II: Rules and gameplay

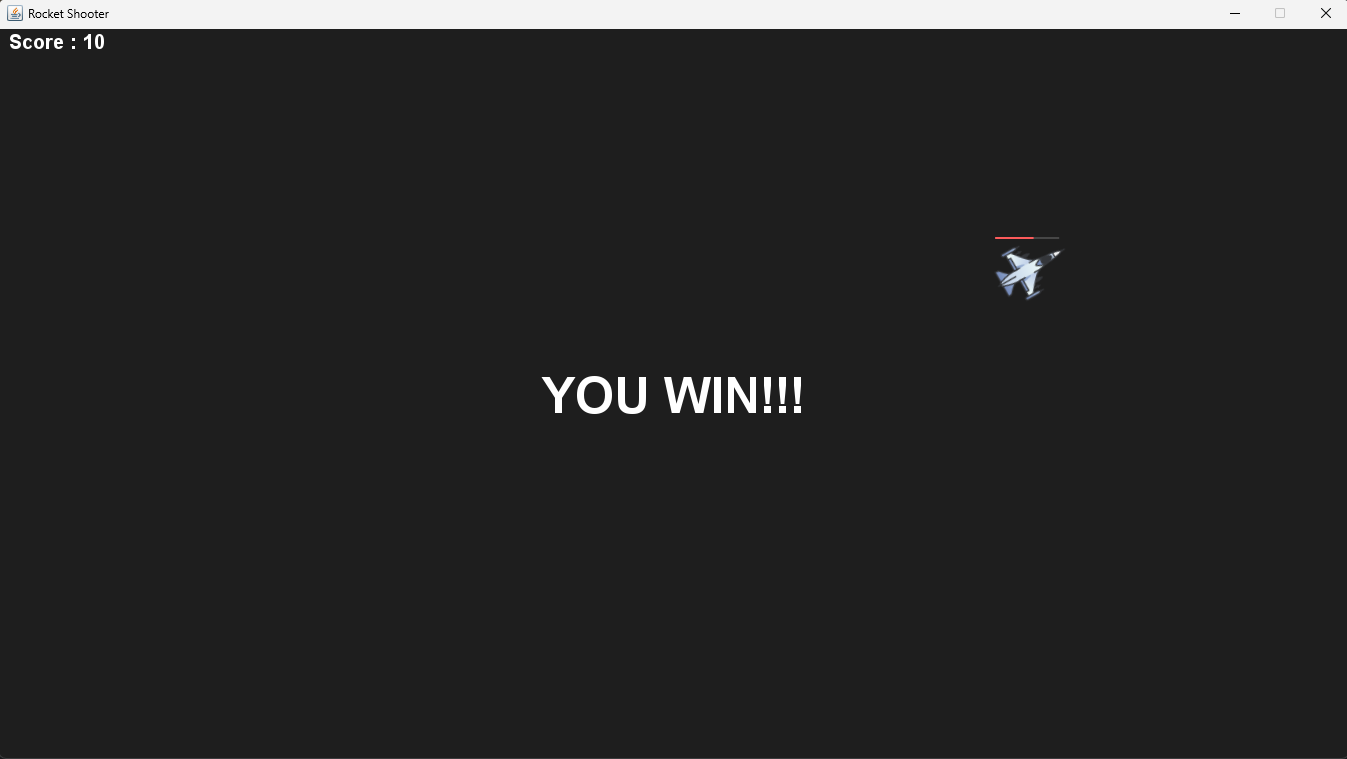
Chapter III: The details of the game technique

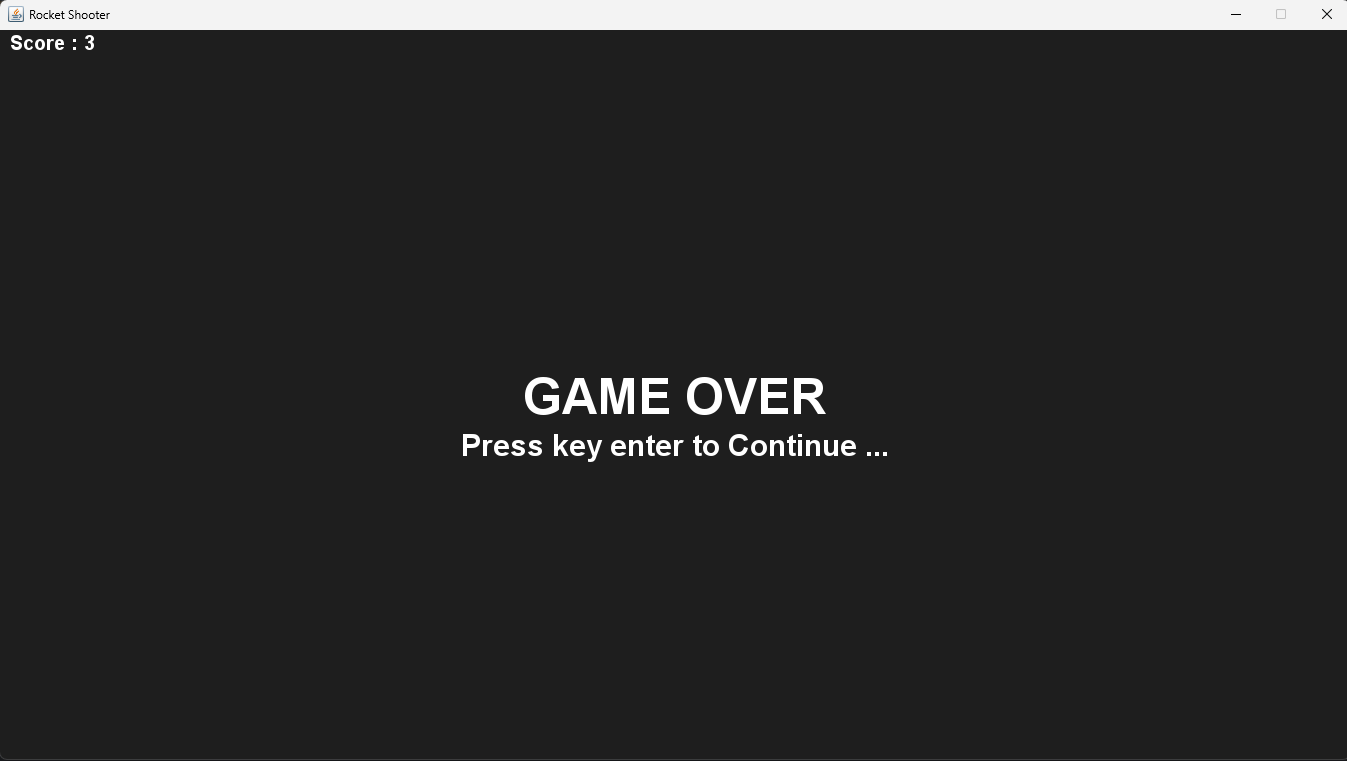
Chapter IV: UML class diagram

CHAPTER II – RULES AND GAMEPLAY

* This is a shooter game in which players will control a plane to shoot randomly generated rockets.
* The player uses the “D” button to turn right, the “A” button to turn left, the“W” button to accelerate, and the “J” button to shoot.
* To win the game, the player has to destroy 10 rockets.
* The game is over when the player’s plane gets hit by the rocket 3 times.







CHAPTER III – THE DETAILS OF THE GAME TECHNIQUE

1. Function of used class

* Package Component:
  + class Key():
    - Implement getters and setters for keyboard inputs.
  + class GamePanel():
* Package Main:
  + class Main():
    - Use to generate the frame UI for the game.
* Package Objective:
  + class Bullet():
    - Implement methods for the location, direction, and shape of the bullet.
  + class Effect():
    - Use to make the explosion effect when destroying a rocket.
  + class HP():
    - An abstract class that has 2 fields “max HP” and “current HP” that were inherited by HPRender().
  + class HPRender():
    - Render the HP bar when an object loses HP.
  + class ModelBoom():
    - An abstract class that has 2 fields “size” and “angle” that were inherited by Effect().
  + class Player():
    - Implement methods for the player’s location, angle, speed, and HP.
  + class Rocket():
    - Implement methods for the rockets’s location, angle, speed, and HP.

CHAPTER IV – UML CLASS DIAGRAM

