Design Overview for Space Shooter

Name: Shah Nawaz Chowdhury
Student ID: 103830682

Summary of Program

The Space Game is a 2d shooter game where the player controls a ship and battles against enemy ships. The objective is to shoot down as many as possible and get a high score with only 3 lives.

Meeting HD Requirement

The program will have implemented 3 design patterns.

1. Template method pattern: This is followed by the Bullet and Ship class wherein they provide a base class by creating a skeleton with helper methods for certain roles and responsibilities all bullets and ships must have, then the concrete implementation is provided by the subclasses.
2. Factory pattern: The creation of bullets by ships follow a factory method approach. The ship class defines a factory method for bullet creation without specifying the exact type of bullet (or which class to instantiate), and the subclasses override that method to create bullets for a specific type.
3. Strategy pattern: The IsColliding interface allows a strategy for collision detection. Any collidable object can implement their own logic for detecting collision, allowing the program to have better decoupling by using composition rather than inheritance for this role as many objects can be collidable without necessarily having any other similar role or responsibility.

The program can be scaled easily since the main features have been decoupled so there is a foundation for the program. Power ups, levels, boulders, etc, can be easily added to the program.

Required Roles

Describe each of the classes, interfaces, and any enumerations you will create. Use a different table to describe each role you will have, using the following table templates.

Table 1: PlayerShip

Responsibility	**Type Details**	**Notes**
Shoot bullets	Bullet type	
Decrease lives	integer	Decrease player's lives
Track score	Integer	Keep track of player's score
Move ship	Void(x,y)	Move ship based on the x and y coordinates

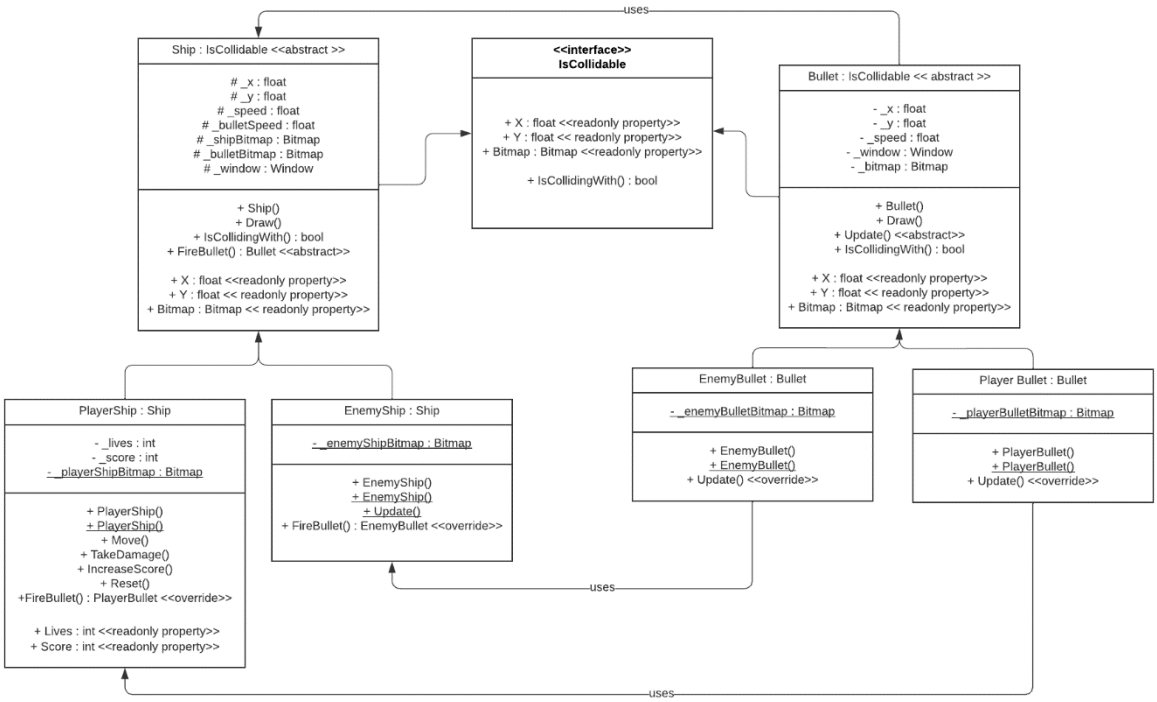
Table 2: EnemyShip

Responsibility	Type Details	Notes
Shoot bullets	Bullet type	
Move ship	Void()	Randomize movement

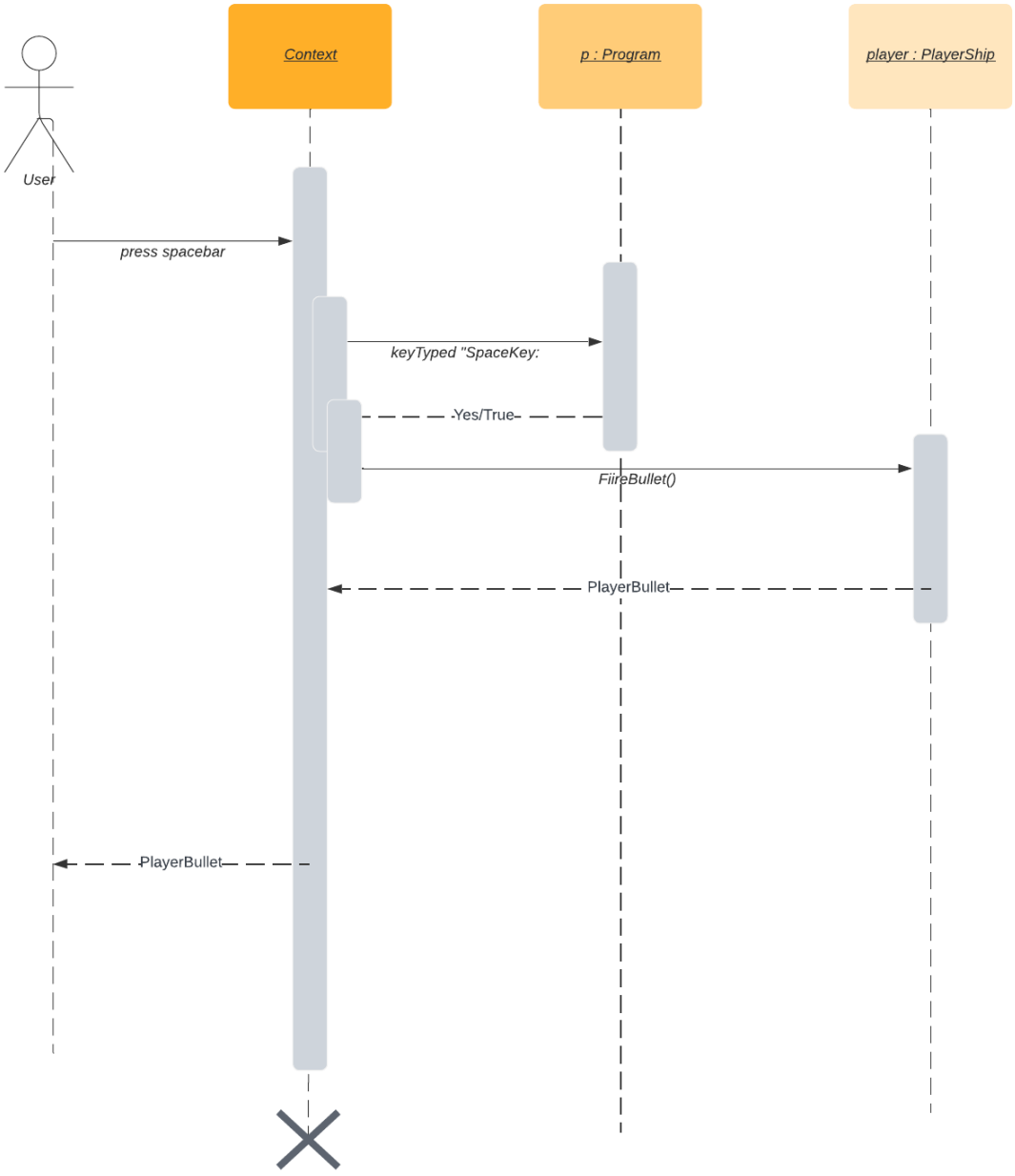
Table 3: Bullet

Responsibility	Type Details	Notes
Move bullet	Void()	Make bullet move in a direction

Class Diagram



Sequence Diagram


Context
p : Program
player : PlayerShip
User
press spacebar
keyTyped "SpaceKey:
Yes/True
FiireBullet()
PlayerBullet
PlayerBullet