Wingman Design

Observer

MyPlane

collision()

Player

Abstract Game Object
void update()
void draw()

Boat

Island

Weapon

Power level

Score bar

Explosion

Enemy
reset()

Bullet

Enemy
Bullet

Big Bullet

HardEnemy

EasyEnemy

Shooting
Enemy

Key Adaptor

Observable

Collision Detector

Key Control

Key Pressed()

Game Events

setValue()

Sound Player
playSound()

AirPlane

init()
paint()

Tank Game

Observer

Player

Abstract Game Object
void update()
void draw()

Tank

Weapon

Power Up

Health Bar

Explosion

Wall

Tank Bullet

Key Adaptor

Observable

Collision Detector

Key Control
Key Pressed()
Key
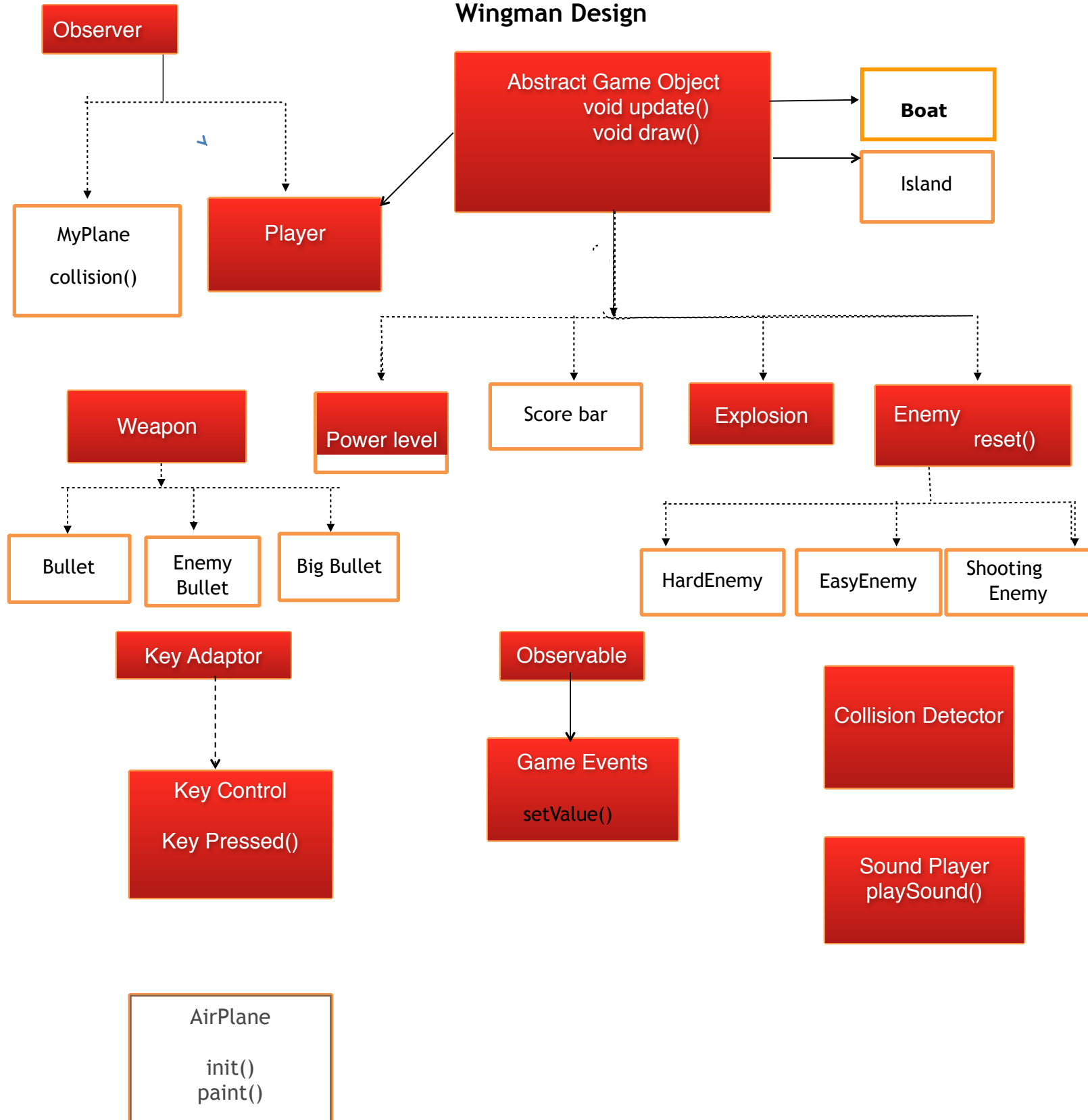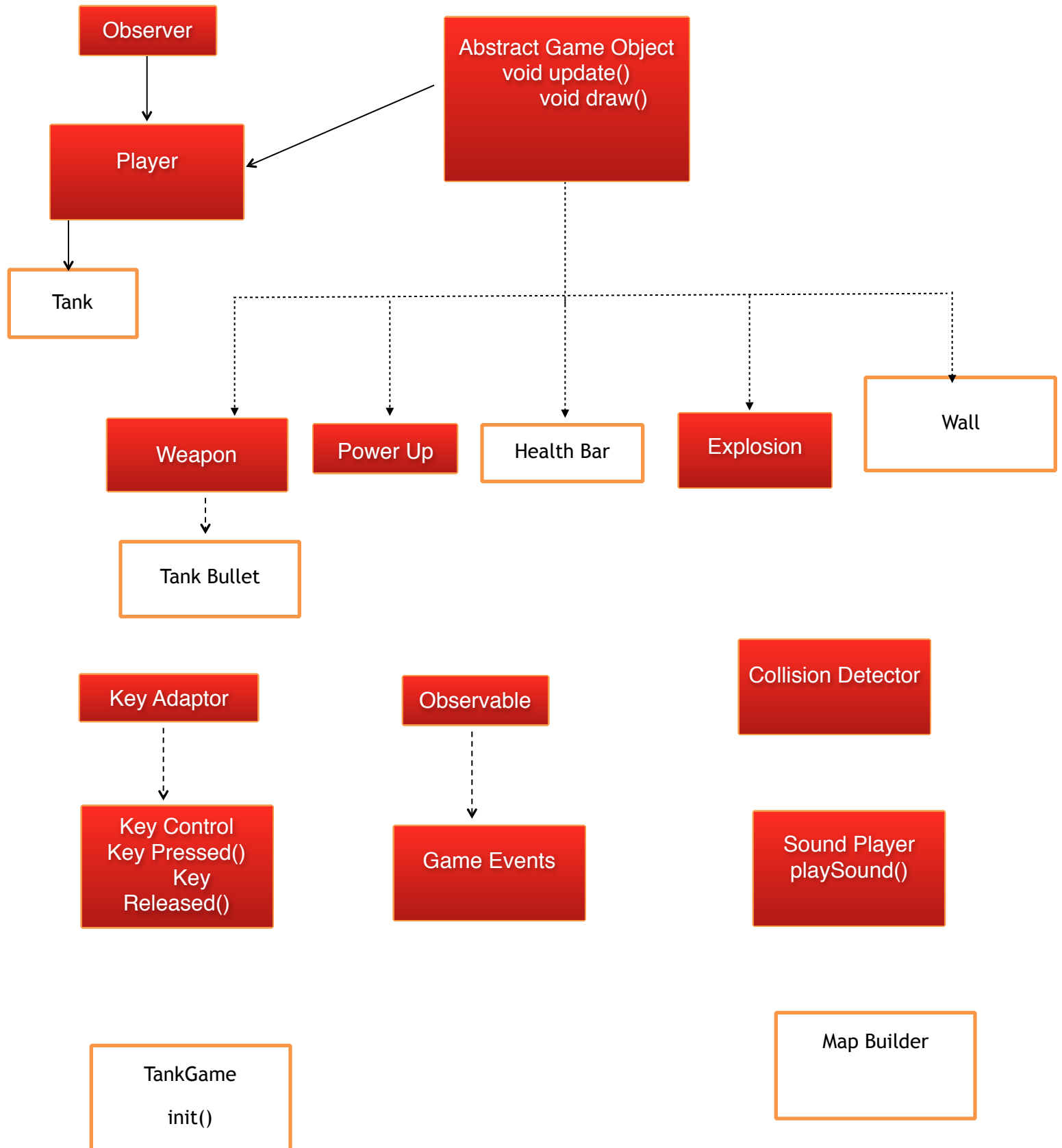Released()

Game Events

Sound Player
playSound()

TankGame

init()

Map Builder

CLASS DESCRIPTION

WINGMAN GAME:

- MyPlane- initialises variables and objects;runs main game.

- Player- is an interface that controls the two planes, shoots bullets and can collides with enemy bullets and planes.

- GameObject- This is an abstract class that defines draw() and update() method which is implemented by its subclasses

- Boat - draws boat in the background of the game screen.
- Island- draws islands in the background of the game screen.

- ScoreBar-displays player's score and number of lifes.

- Bullet-different types of bullets created: enemy and player bullets.

- KeyControl-detects when keys are pressed and released.

- GameEvents-detects when and event occurs in the game.

- Enemy-different types of enemies created: come from behind, stronger, diagonal bullets,etc.

- Explosion-displays explosion image and plays sound when player or enemy dies.

- SoundPlayer-This class gives the background music in the game and also the music when an event occurs. For example when a collision takes place. It implements playSound() method.

TANK GAME

- Wall- has a lifetime and dissappears after taking certain number of bullets.

- MapBuilder- This class is used to show the map of the layout to the player. It implements the buildMap() method.

REUSABLE CLASSES:

- Observer
- GameObject
- Player
- Weapon
- PowerUp
- Explosion
- Enemy
- KeyAdaptor
- KeyControl
- Observable
- GameEvents
- CollisionDetector
- SoundPlayer