2020

Plane War

DESIGN DOCUMENT

WAI HONG CHONG

Contents

1 Introduction

1.1 Purpose

This document outlines the specifications and design pattern implementations used to create the Plane War game. This document will describe the overall design and high-level view of the game, the various components, and design pattern implemented, how the implementations function within the game and gameplay, and opportunities for additional improvement.

1.2 Goals

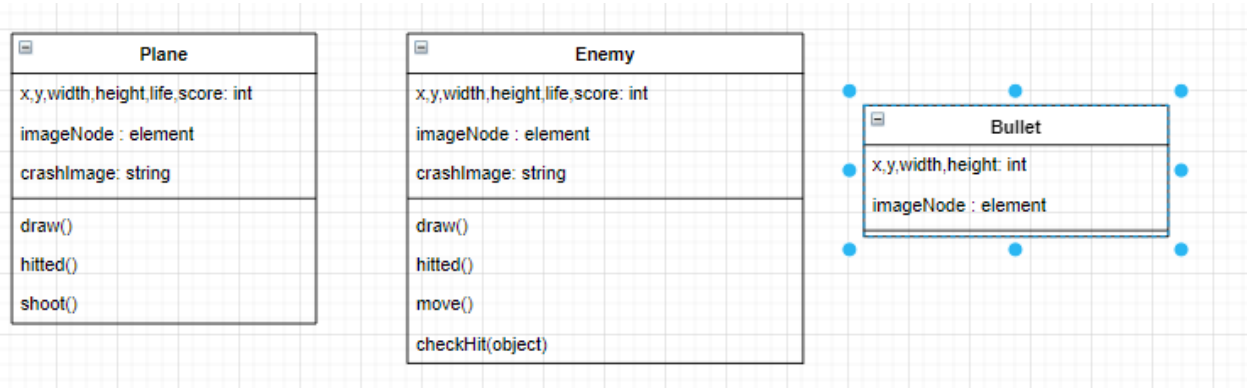
The goal of this project is to develop a similar online game “plane war” based on the space invaders arcade game, while using design pattern to write more effective code. The game will automatically generate the player’s plane, and the plane can be operated by the mouse. Players’ plane will automatically launch missiles to try to destroy enemy planes one by one. At the same time, the game will randomly generate three different sizes of enemy planes to increase the difficulty. The player has a total of three lives to get the highest score.

2 Design patterns

2.1 Constructor Pattern

In JavaScript, constructor pattern provides us more convenient and more scalable implementation way than classes, because we can crate many objects of the same “type” by using a “blueprint”, which is the constructor, even the classes can do the same feature. But the constructor also offer another adding a property to an object, which means you can add a new property to an existing object (not the “blueprint”), also you are allow to add a new method to an existing object, that’s why we said that constructor pattern offer more scalable than classes. So, in future, you can extend more function or property to an object that created by using constructor.

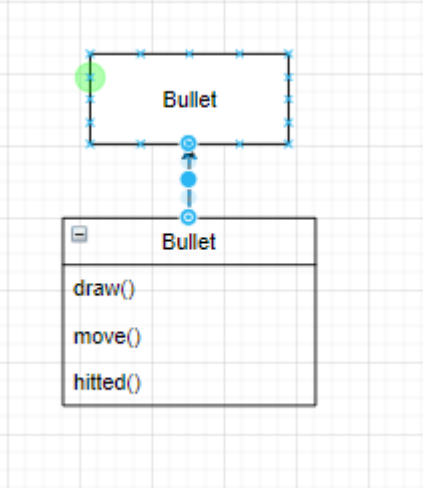
In the figure below, we will just create the Plane, Enemy and Bullet by using the constructor design pattern, as you can see the Plane and Enemy is very similar, so that we can improve those in future by inherited from the Plane and just simply extend the different object by using the constructor pattern.



2.2 Prototype pattern

The prototype pattern is to expand on the original basis of the constructor pattern with more function and variables, which the constructor is only allowed to expand on an object, and the prototype is to expand on the “blueprint”.

In here, we only use the prototype pattern in Bullet, even we can just create the function by using the constructor pattern, but I think these will make the code look cleaner.



3 Discussion

The process of designing this game is to ensure that the game can have the best performance by using multiple design patterns. And considering that more functions may be expanded in the future to make the game more fun, such as more levels, menu functions, and score records, we spent a lot of time integrating these modes into the project, and as more With the addition of more components, these design patterns have proved more useful. If we do not carry out these conceptual planning in advance, it will be more difficult when we want to add more components.