



# Code Review : Invaders

김승환

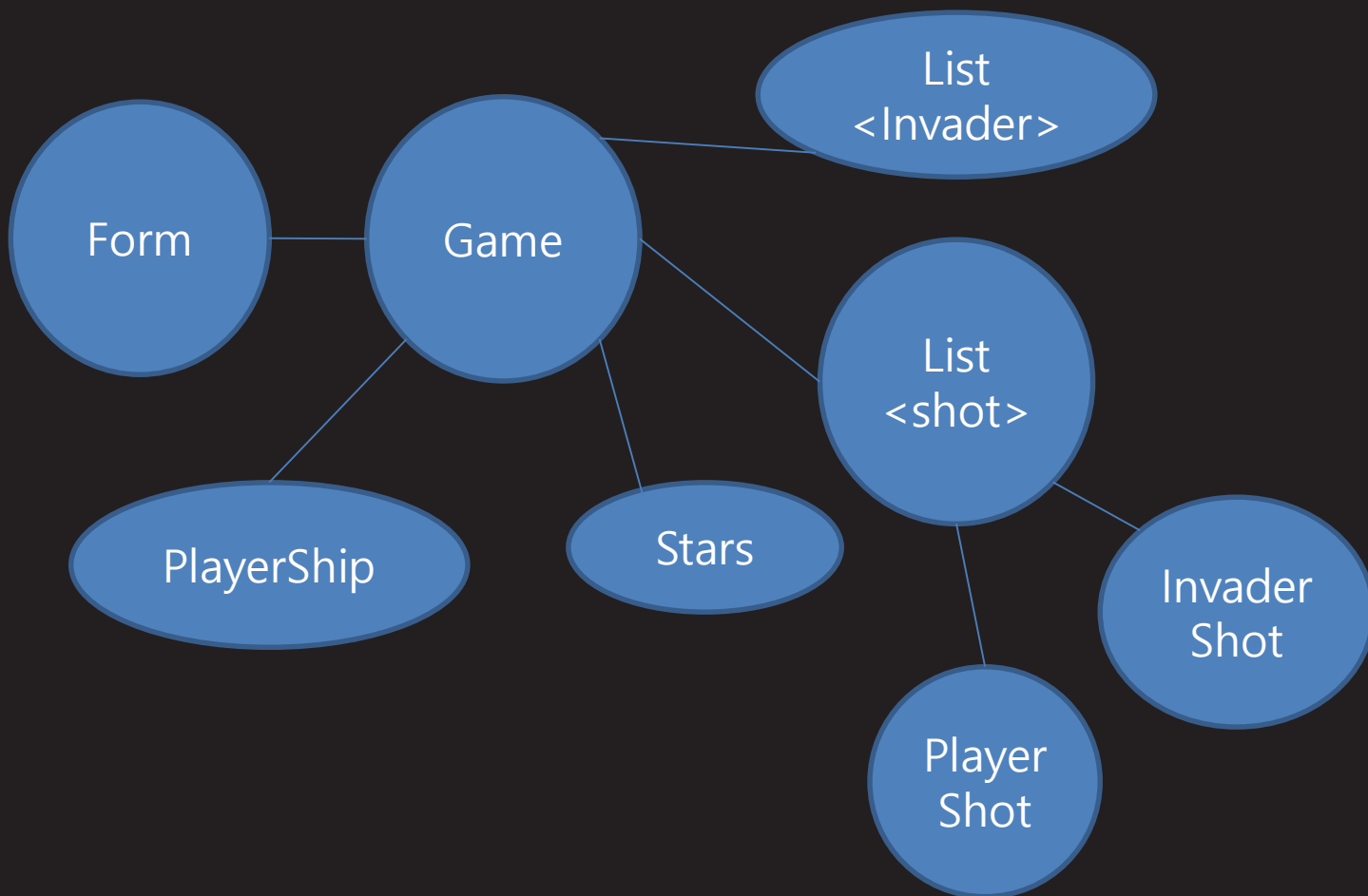
# Contents

- 설계 구조 및 클래스 다이어그램
- 목표 및 구현
- 코드 분석
- 개선할 점



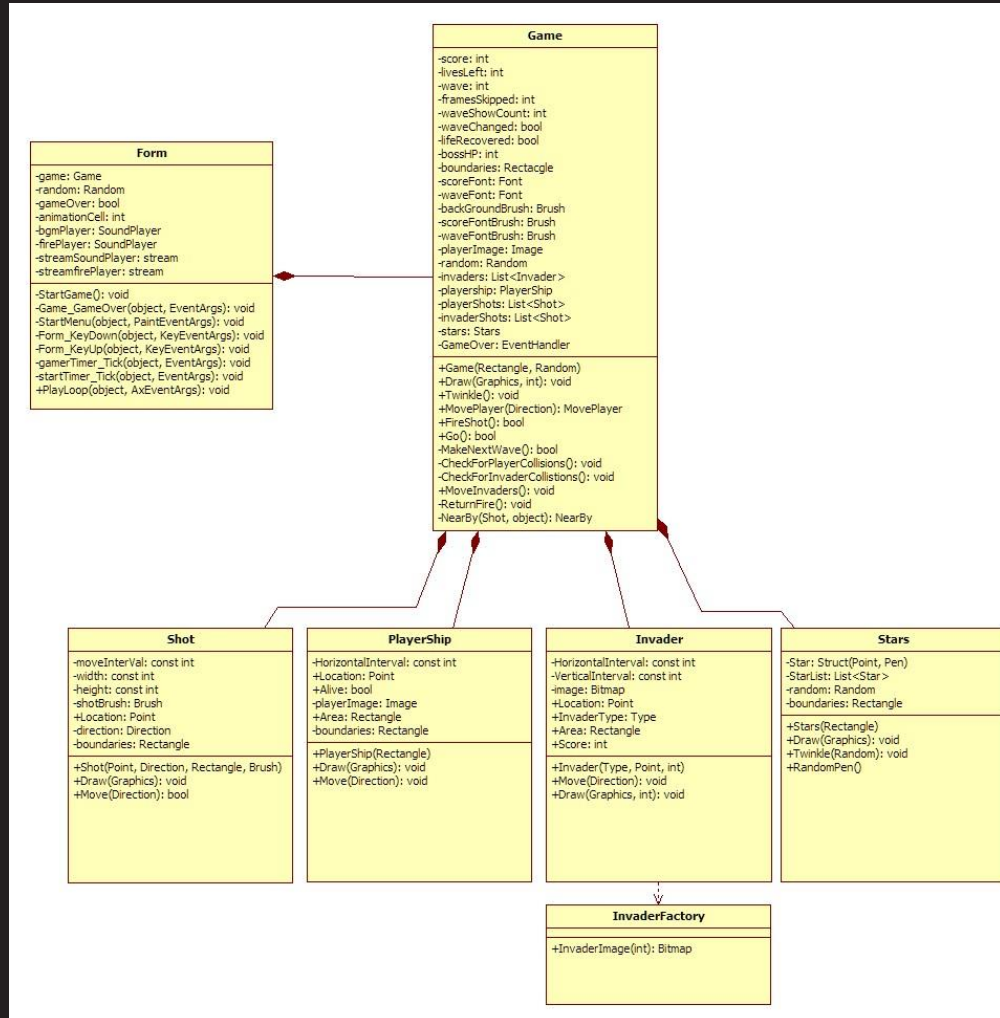
# Invaders

## 설계 구조



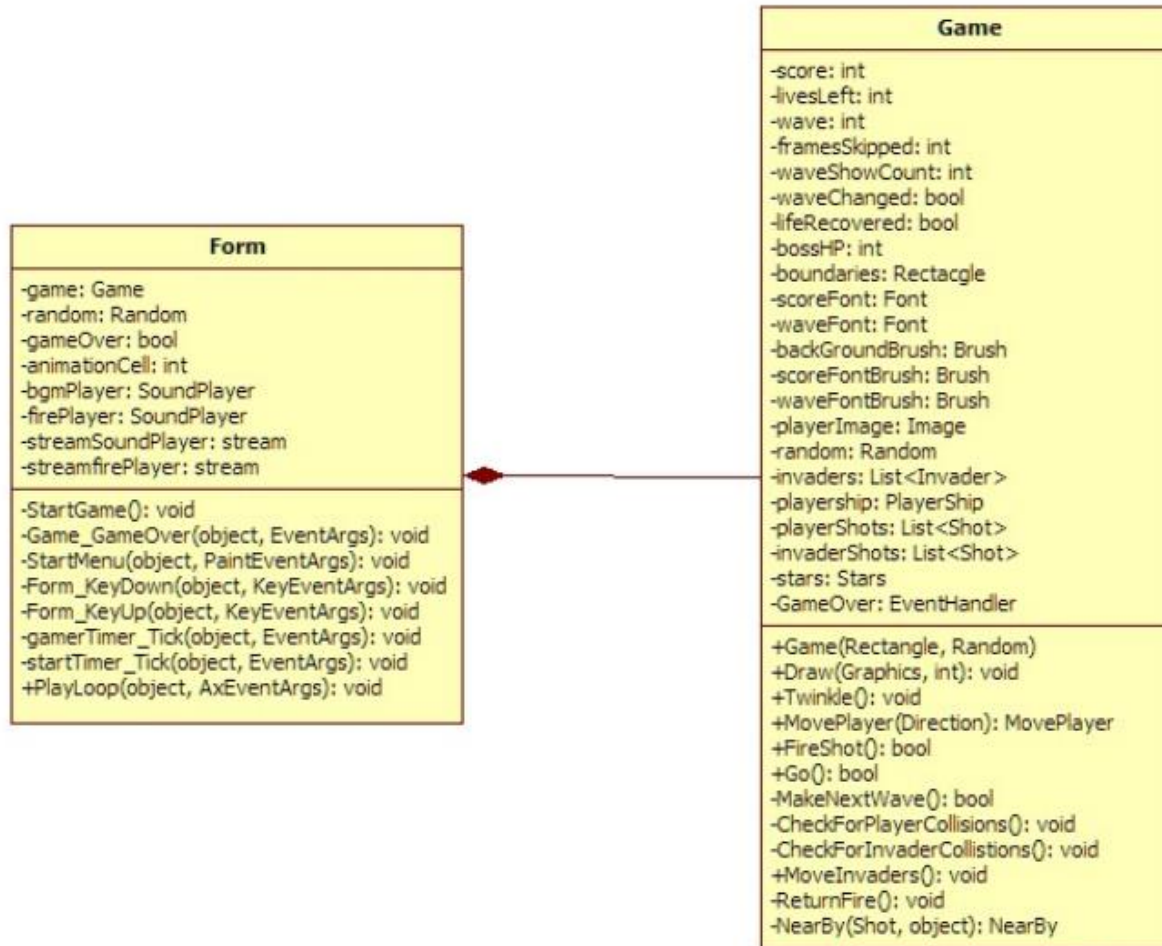
# Invaders

## Class Diagram



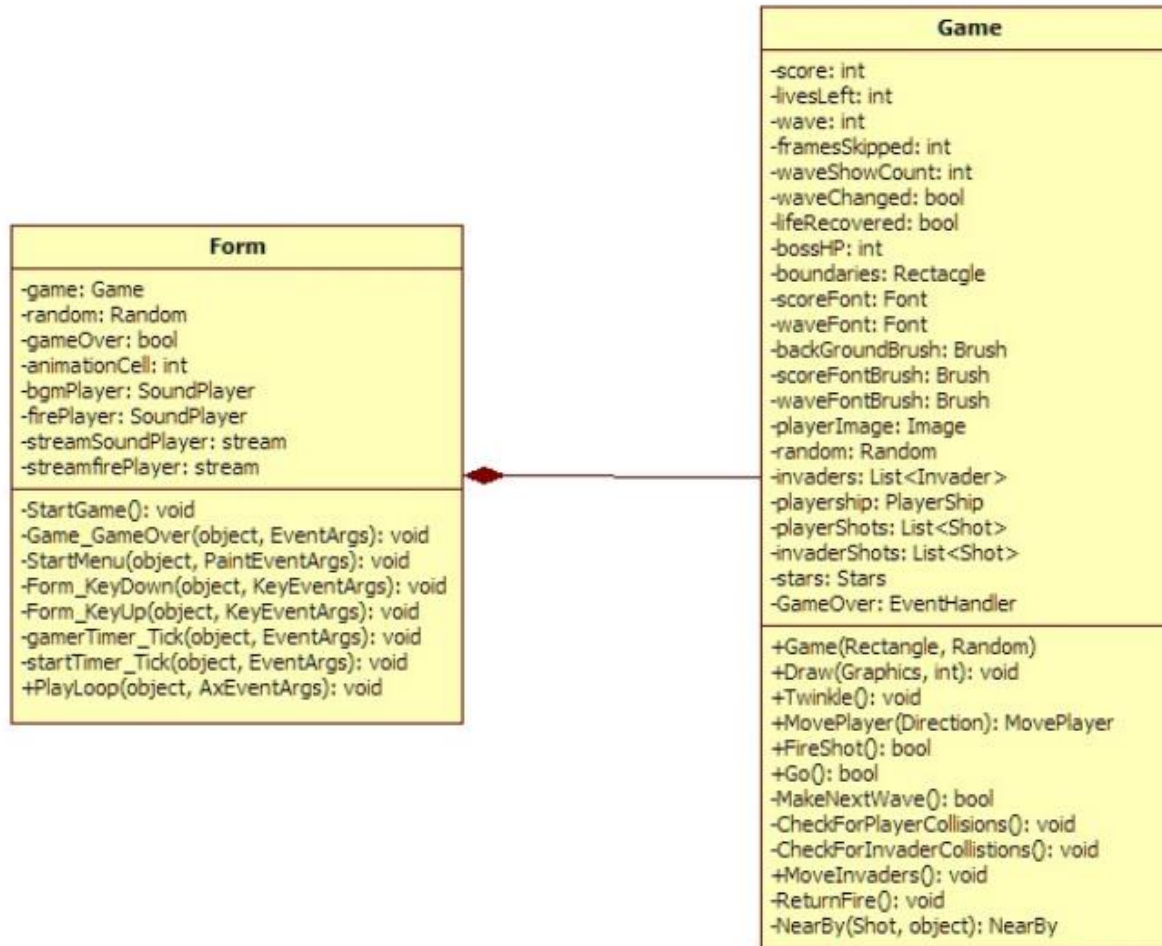
# Invaders

## Class Diagram



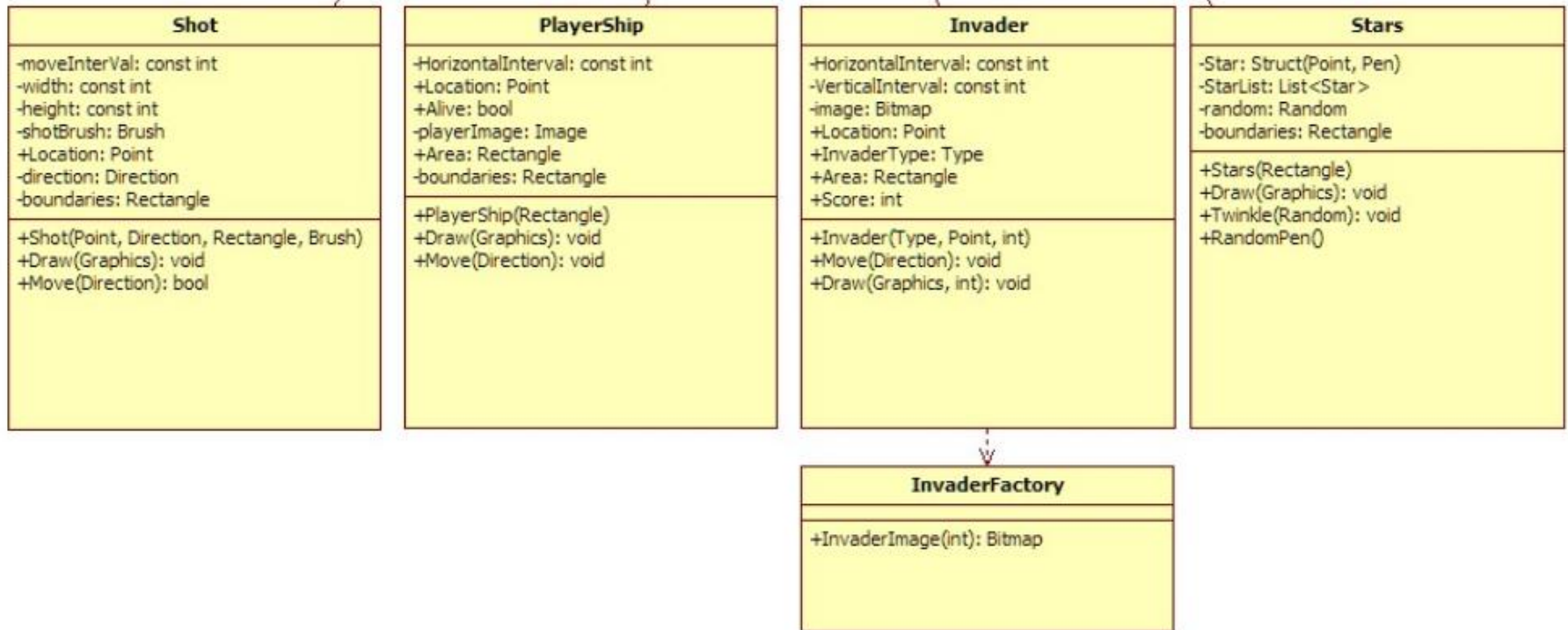
# Invaders

## Class Diagram



# Invaders

## Class Diagram



# Invaders

---

## 목표 및 구현

- 함수 모듈화 및 단순화
- 스토리텔링
- 변수, 메소드 작성법 준수
- 버그 최소화
- 커리큘럼 내용 리마인드(파일 입출력, 예외 처리, LINQ, 디자인 패턴)





# Invaders

---

## 목표 및 구현

- 게임 시작 화면
- 메인 게임 화면
- 키 입력
- 게임 구성(총 9스테이지)
- 보스 스테이지 제작
- BGM, Effect Sound



# Invaders

---

## 코드 분석

- 게임 세팅
- 게임 화면 그리기
- 배경음악, 효과음
- 충돌 체크, Invader 이동
- 총알 발사



# Invaders

## 코드 분석

- Invader 객체 리스트에 집어넣기

```
private bool MakeNextWave()
{
    wave++;
    if (wave > 9)
    {
        return true;
    }
    else if (wave > 8)
    {
        invaders.Add(new Invader(Type.Boss, new Point(200, 100), 1000));
    }
    else
    {
        for (int i = 0; i < 30; i++)
        {
            int col = i / 6 + 1;
            int row = i % 6;
            invaders.Add(new Invader((Type)col, new Point(250 + row * 60, col * 60), (6 - col) * 10));
        }
    }
    if (livesLeft < 2)
    {
        livesLeft = 2;
        lifeRecovered = true;
    }
    framesSkipped = 4 - wave;
    invaderDirection = Direction.Right;
    return false;
}
```



# Invaders

## 코드 분석

- Flyweight의 Factory Method

```
public abstract class InvaderFactory
{
    public static Bitmap getInvaderImage(Type InvaderType, int animationCell)
    {
        switch (InvaderType)
        {
            case Type.Bug:
                if (animationCell == 0) return Properties.Resources.bug1;
                else if (animationCell == 1) return Properties.Resources.bug2;
                else if (animationCell == 2) return Properties.Resources.bug3;
                else return Properties.Resources.bug4;
            case Type.Satellite:
                if (animationCell == 0) return Properties.Resources.satellite1;
                else if (animationCell == 1) return Properties.Resources.satellite2;
                else if (animationCell == 2) return Properties.Resources.satellite3;
                else return Properties.Resources.satellite4;
            case Type.Saucer:
                if (animationCell == 0) return Properties.Resources.flyingsaucer1;
                else if (animationCell == 1) return Properties.Resources.flyingsaucer2;
                else if (animationCell == 2) return Properties.Resources.flyingsaucer3;
                else return Properties.Resources.flyingsaucer4;
            case Type.SpaceShip:
                if (animationCell == 0) return Properties.Resources.spaceship1;
                else if (animationCell == 1) return Properties.Resources.spaceship2;
                else if (animationCell == 2) return Properties.Resources.spaceship3;
                else return Properties.Resources.spaceship4;
            case Type.Star:
                if (animationCell == 0) return Properties.Resources.star1;
                else if (animationCell == 1) return Properties.Resources.star2;
                else if (animationCell == 2) return Properties.Resources.star3;
                else return Properties.Resources.star4;
            case Type.Boss:
                return Properties.Resources.Boss;
            default:
                return null;
        }
    }
}
```



# Invaders

---

## 개선할 점 및 아쉬운 점

- State Pattern을 사용하지 않아 조건문이 너무 많다
- 스토리텔링 기법이 남에게 와 닿지 않을 수 있다
- 시간을 너무 조금 투자했다(파일 입출력 미구현)
- 추상화를 사용하지 않았다(물체들의 기본 베이스가 되는 클래스)

