Описание игры: *Есть три полосы движения корабля, которым управляет игрок. Начальное положение - центральное. С течением времени на экране будут появляться различные объекты (астероиды или вражеские корабли). Цель игрока - уклоняться от надвигающихся объектов (перемещаясь между этими полосами): столкновение - проигрыш. Корабль игрока имеет возможность стрелять и “разбивать” вражеские корабли. С течением времени сложность игры будет увеличиваться : в какой-то момент вражеские корабли начнут стрелять, их будет все сильнее “разбить”.*

В игре есть три типа объектов : персонаж, которым игрок будет управлять (в дальнейшем - *земной корабль*); предметы-препятствия (*астероиды*)
и объекты, атакующие персонажа (*вражеские корабли*).

1) Земной корабль - объект, задаваемый классом **Character**. Может быть создан только единожды, как уникальный персонаж игры. Имеет атрибут (bullet_speed) - скорость пуль (неизменная на протяжении всей игры).

2) Астероиды - объекты с характеристикой размер (size). Появляются на экране по мере движения корабля. В зависимости от времени игры могут принимать разный размер :
большой или малый.

- Создаются путем вызова метода create подклассов (**SmallAsteroidFactory** и **BigAsteroidFactory**) большого класса **AsteroidFactory** в зависимости от того, какого размера препятствие требуется.
- Задаются подклассами (**SmallAsteroid** и **BigAsteroid**) большого класса **Asteroid** в зависимости от размера.

3) Вражеские корабли - объекты с характеристиками : цвет (color), прочность (health), возможность стрелять (weapon), скорость пуль (bullet_speed). Появляются на экране по мере движения корабля. В зависимости от времени игры могут повышать свой уровень : будет увеличиваться наносимый урон и прочность.

- Создание происходит следующим образом : создается объект класса **Shipyard**, от которого вызывается метод construct_ship с параметром - тип желаемого корабля. В ходе него вызывается метод create одного из подклассов (**ShipBuilderA**, **ShipBuilderB**, **ShipBuilderC** и т.д.) большого класса **ShipBuilder** в зависимости от требуемых характеристик.
- Задаются классом **Ship** с атрибутами, в которых хранятся значения характеристик.

Графика:

- Класс **ImageOfObject** отвечает за изображение определенного объекта в определенных координатах. Соответственно, имеет метод draw, принимающий любой объект и рисующий его в заданных координатах. Помимо вышеописанных классов существуют классы **Bullet**, **Lines** и **Window**, необходимые для графической части реализации.

- Класс **DrawAdapter** отвечает за необходимость рисовать множество различных объектов единым интерфейсом. Объект этого класса имеет метод draw_object, принимающий в качестве аргументов объект, который необходимо нарисовать, и координаты.
- Класс **ImageLayer** отвечает за поэтапное(по слоям) изображение объектов. Так, например, фон (объекты классов Window и Lines) должен рисоваться в первую очередь. В итоге процесс построения изображения следующий:
    - Фон
    - Движущиеся объекты
        - Земной корабль
            - Пули, “выпущенные” земным кораблем
        - Другие движущиеся объекты
            - Астероиды
            - Вражеские корабли
                - Пули, “выпущенные” вражеским кораблем