**Модуль test**

### Метод game\_init

Тест №1 (normal)

Цель: проверка успешной инициализации игры

Тип: позитивный

Входные данные: inGame = false

Ожидаемый результат: одно решение, флаг inGame = true

### Метод game\_objects

Тест №1 (normal)

Цель: проверка успешной инициализации игровых объектов

Тип: позитивный

Входные данные: dots = 0;

Ожидаемый результат: поле полученной структуры s: dots != 0

### Метод apple\_spawn

Тест №1(normal)

Цель: проверка генерации координат

Тип: позитивный

Входные данные: apple\_x = -1, apple\_y = -1;

Ожидаемый результат: два решения, поля полученной структуры s: apple\_x != -1, apple\_y != -1;

### Метод apple\_check

Тест №1 (eaten)

Цель: проверка соприкосновения змеи с яблоком, после рост змеи увеличивается и генерируются новые координаты яблока

Тип: позитивный

Входные данные: dots = 5, apple\_x = 5, apple\_y = 5, x[0] = apple\_x , x[0] = apple\_y;

Ожидаемый результат: три решение, поля полученной структуры s: dots = 6, apple\_x != 5, apple\_y != 5;

Тест №2 (not\_eaten)

Цель: проверка соприкосновения змеи с яблоком, то змея не увеличивается

Тип: позитивный

Входные данные: dots = 5, apple\_x = 3, apple\_y = 3, x[0] = apple\_x , x[0] = apple\_y;

Ожидаемый результат: три решение, поля полученной структуры s: dots = 6, apple\_x = 3, apple\_y = 3;

### Метод ***collision\_check***

Тест №1 (***small\_snake\_in\_borders\_and\_not\_bites\_itself***)

Цель: проверка состояния короткой змеи без соприкосновений с коллизиями стен или хвоста

Тип: позитивный

Входные данные: inGame = true, dots = 3, x[0] = 50 , y[0] = 50, x[1] = 40 , y[1] = 50, x[2] = 30 , y[2] = 50;

Ожидаемый результат: одно решение, флаг полученной структуры s: inGame = true;

Тест №2 (***snake\_in\_borders\_and\_not\_bites\_itself***)

Цель: проверка состояния змеи при коллизии с хвостом

Тип: позитивный

Входные данные: inGame = true, dots = 5, y[0] = 40 , x[0] = 70, y[1] = 40 , x[1] = 60, y[2] = 50 , x[2] = 60, y[3] = 60 , x[3] = 60, y[4] = 60 , x[4] = 70, y[5] = 50 , x[5] = 70, y[6] = 40 , x[6] = 70;

Ожидаемый результат: одно решение, флаг полученной структуры s: inGame = true;

Тест №3 (***in\_borders\_and\_bites\_itself***)

Цель: проверка состояния змеи при коллизии с хвостом

Тип: позитивный

Входные данные: inGame = true, dots = 5, y[0] = 40 , x[0] = 70, y[1] = 40 , x[1] = 60, y[2] = 50 , x[2] = 60, y[3] = 60 , x[3] = 60, y[4] = 60 , x[4] = 70, y[5] = 50 , x[5] = 70, y[6] = 40 , x[6] = 70;

Ожидаемый результат: одно решение, флаг полученной структуры s: inGame = false;

Тест №4 (***out\_of\_left\_border***)

Цель: выход за левую границу

Тип: позитивный

Входные данные: inGame = true, dots = 3, x[0] = -10 , y[0] = 50, x[1] = 0 , y[1] = 50, x[2] = 10 , y[2] = 50;

Ожидаемый результат: одно решение, флаг полученной структуры s: inGame = false;

Тест №5 (***out\_of\_right\_border***)

Цель: выход за правую границу

Тип: позитивный

Входные данные: inGame = true, dots = 3, x[0] = 310 , y[0] = 50, x[1] = 300 , y[1] = 50, x[2] = 290 , y[2] = 50;

Ожидаемый результат: одно решение, флаг полученной структуры s: inGame = false;

Тест №5 (***out\_of\_up\_border***)

Цель: выход за верхнюю границу

Тип: позитивный

Входные данные: inGame = true, dots = 3, x[0] = 50 , y[0] = -10, x[1] = 50 , y[1] = 0, x[2] = 50 , y[2] = 10;

Ожидаемый результат: одно решение, флаг полученной структуры s: inGame = false;

Тест №5 (***out\_of\_down\_border***)

Цель: выход за нижнюю границу

Тип: позитивный

Входные данные: inGame = true, dots = 3, x[0] = 50 , y[0] = 310, x[1] = 50 , y[1] = 300, x[2] = 50 , y[2] = 290;

Ожидаемый результат: одно решение, флаг полученной структуры s: inGame = false;

Тест №6 (***game\_has\_alredy\_ended***)

Цель: проверка реакции уже на завершённую игру

Тип: негативный

Входные данные: inGame = true, dots = 3, x[0] = 70 , y[0] = 50, x[1] = 60 , y[1] = 50, x[2] = 60 , y[2] = 40;

Ожидаемый результат: одно решение, флаг полученной структуры s: inGame = false;