Заликов Руслан БВТ 2106

v 0.2.3 – программа работает за (k + 6)n + (k + 1)n^2 где n количество точек а k рандомное число так как оно ровно попытки с которой алгоритм смог подобрать такие n точек чтобы они не накладывались друг на друга. При ошибке, что какая то точка наложилась, рандом начинает заново подбирать, поэтому k [1 , +) следовательно сложность алгоритма равна O(k), и если избавится от k то сложность станет O(n^2) что тоже не самое лучшее решение, так что после надо будет избавляться от n^2

Цель на v 0.2.4:

Избавиться от k

v 0.2.4 – Избавился от k, теперь задается не n случайных пар чисел и проверяется друг с другом, а весь экран делится на n участков, и у для каждой кнопки на участке i выбирается рандомная точка х от левой границы + радиус до правой границе – радиус, а по у просто рандомное значение от верха экрана + радиус до нижнего – радиус потому что в pygame счет по y идет сверху вниз, теперь функция которая работала за 2n + k(n^2 + n) работает за 2n

теперь сложность кода О(n^2)

исправлен вывод результата, теперь выводится моментально после прохода раунда

Цель на v 0.2.5:

Избавиться от n^2

v 0.2.5 – Пофиксил направление шаров, раньше они могли появляться со скорость 0, 0, теперь же такого произойти не может, и статичные шары не будут спавниться. Поставил задержку после попадания, чтобы пользователь успевал понять что происходит вокруг

Так же удалось в функции, которая обновляет картинку и соответственно перемещает точки, уменьшить количество проверок с n^2 до n(n+1)/2 хоть и порядок сложности все равно квадрат, все равно прогресс очевидно виден

Цель на v 0.2.6: уменьшить сложность еще, есть пару идей, протестировать, и реализовать код который уменьшит сложность до n(n-1)/2

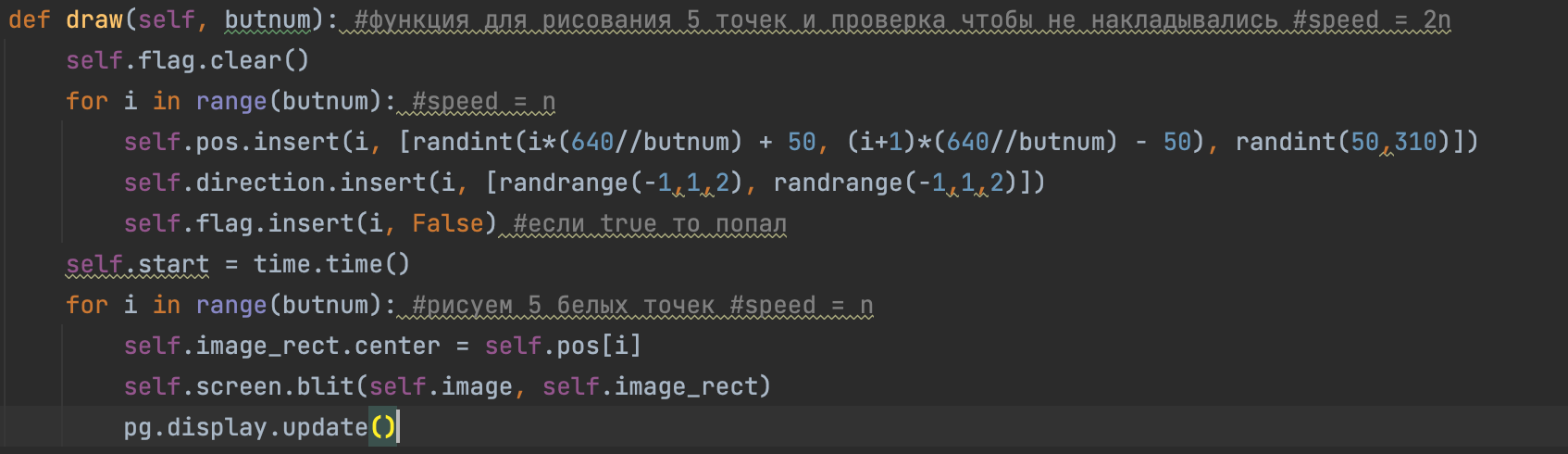
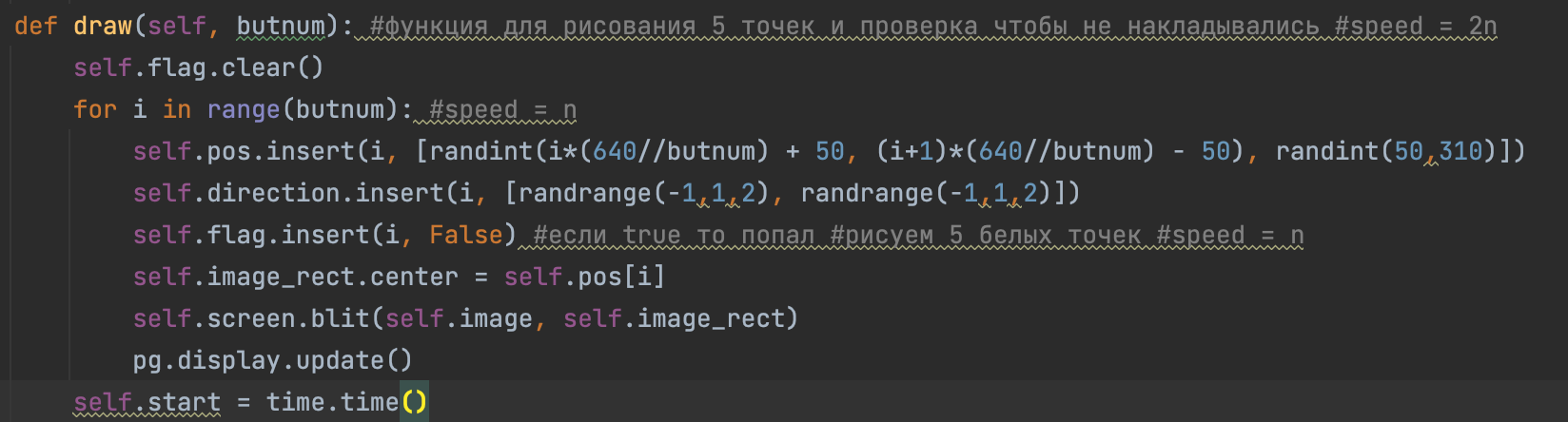
v 0.2.6 – уменьшил сложность до n(n-1)/2 добавив лишь два символа, суть оптимизации заключается в том, программа проверяла столкновение каждого с каждым

v 0.2.4 v 0.2.5 v 0.2.6

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| R(ab) | 1 | 2 | 3 | 4 | 5 | R(ab) | 1 | 2 | 3 | 4 | 5 | R(ab) | 1 | 2 | 3 | 4 | 5 |
| 1 | 0 | 12 | 13 | 14 | 15 | 1 | 0 | 12 | 13 | 14 | 15 | 1 | 0 | 12 | 13 | 14 | 15 |
| 2 | 21 | 0 | 23 | 24 | 25 | 2 | 21 | 0 | 23 | 24 | 25 | 2 | 21 | 0 | 23 | 24 | 25 |
| 3 | 31 | 32 | 0 | 34 | 35 | 3 | 31 | 32 | 0 | 34 | 35 | 3 | 31 | 32 | 0 | 34 | 35 |
| 4 | 41 | 42 | 43 | 0 | 45 | 4 | 41 | 42 | 43 | 0 | 45 | 4 | 41 | 42 | 43 | 0 | 45 |
| 5 | 51 | 52 | 53 | 54 | 0 | 5 | 51 | 52 | 53 | 54 | 0 | 5 | 51 | 52 | 53 | 54 | 0 |

По сторонам номер точки а внутри расстояние между центрами которое проверяет программа, и расстояние должно быть больше двух радиус, иначе направление обеих точек меняется на противо положное. Зеленым цветом закрашены те расстояния которые проверяются, на примере всего 5 точек, но программа может принимать любое значение точек.

Так же заметил что лишний раз хочу по точкам в методе рисования точек в момент появления

v 0.2.5v 0.2.6

не вижу смысл ходить два раза, так как я изменил создание точек теперь я могу сразу после создания точки рисовать ее, так что теперь метод draw класс точки, выполняет n операций вместо 2n

цель на v 0.2.7:

Сейчас нахожусь в процессе изучения pygame.sprite, я думаю это поможет мне с эффектом взрыва, сейчас в принципе есть идеи как его сделать, но раз уж делать, то делать правильно, д и кадров взрыва я никак найти не могу, возможно нарисую сам какой нибудь условный взрыв, так же на буду будущие версии, цель подключить tkinter, хочу чтобы перед игрой выскакивало окно, с кнопками и пользователь выбирал сам количество шаров, заготовка уже готова, пока проблемы с понимание tkinter не могу запустить функцию и передать в нее значение кнопки после нажатия на нее, tkinter позволяет вызвать функцию после нажатия но не позволяет закинуть в нее какое либо значение