|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА Системы обработки информации и управления \_\_\_\_\_\_\_\_\_

**ОТЧЕТ**

**К ЛАБОРАТОРНОЙ РАБОТЕ № 4**

**По дисциплине «Методы поддержки принятия решений»**

**РЕШЕНИЕ ОПТИМИЗАЦИОННЫХ ЗАДАЧ С ПОМОЩЬЮ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ**

Студент ИУ5-73Б  Д.К. Пермяков

(Группа) (Подпись, дата) (И.О.Фамилия)

Преподаватель  Ю.Н. Кротов

(Подпись, дата) (И.О.Фамилия)

Москва – 2024

**Цель**

Научиться решать оптимизационные задачи с помощью генетических алгоритмов.

**Задание**

Лабораторная работа состоит из двух частей:

I. Использование генетического алгоритма для оптимизации раскладки клавиатуру для заданных слов;

II. Использование генетических алгоритмов для обучения интеллектуальных агентов.

Для выполнения части I. требуется проделать следующие шаги:

1. Загрузить необходимые файлы (вариант со списком слов, макет клавиатуры и набор шрифтов).
2. Загрузить файл клавиатуры и шрифта в GoogleColab
3. Выбрать 3 слова согласно варианту и для каждого найти оптимальные раскладки клавиатуры, и визуализировать их с помощью макета и набора шрифтов. Если в слове присутствует **дефис**, то либо заменить на пробел, либо писать слитно.

Для выполнения части II. требуется выполнить следующие шаги:

1. Выбрать одну из предложенных игр с реализацией на JavaScript.
2. Обучить интеллектуального агента с помощью генетического алгоритма в среде без графического интерфейса на Python. В случае необходимости модифицировать код для получения лучших результатов.
3. Проверить качество агента в первоначальной среде на JavaScript.

**Ход работы**

**Часть I**

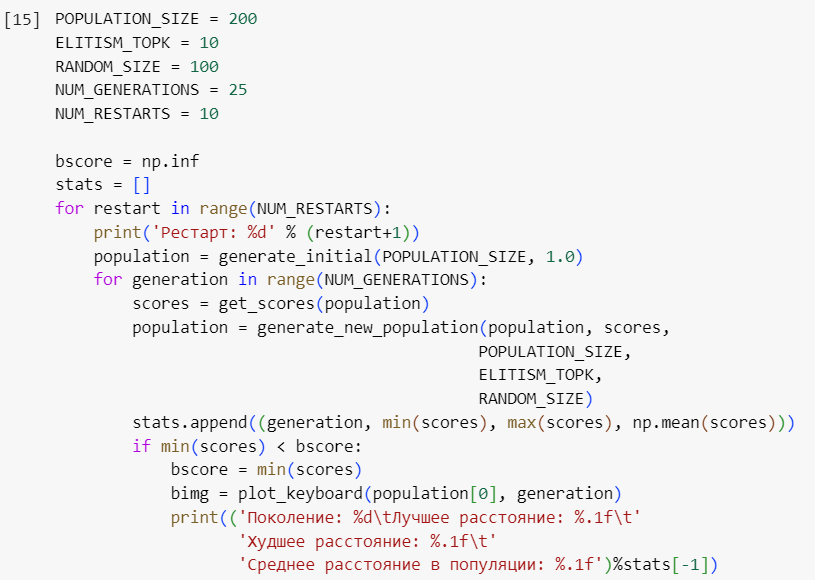
Вариант 13. Слова: Уменьшительно-ласкательное, транслитизировать, примитивщина.

Используем алгоритм для оптимизации раскладки клавиатуры сразу для всех трёх слов, потому что они имеют похожий буквенный состав.

Создадим и отрисуем клавиатуру с обычной раскладкой:



Далее задаём функцию приспособленности, функцию мутации и скрещивания. Затем создаём функцию генерации популяций. Осуществим оптимизацию раскладки с помощью генетического алгоритма:



Отобразим получившуюся раскладку для слова “уменьшительноласкательный”:



Как можно увидеть, все буквы, из которых состоят слова " уменьшительноласкательный ", собрались в одном месте. Кроме того, ГА расположил "е", "с" и "т" по центру, так как они встречаются наиболее часто, таким образом обход по буквам во всех словах минимизируется. После 10 повторных запусков ГА минимальное найденное расстояние оказалось равным 206097.5. Это почти в два раза меньше, чем у стандартной раскладки клавиатуры.

“Транизитировать”



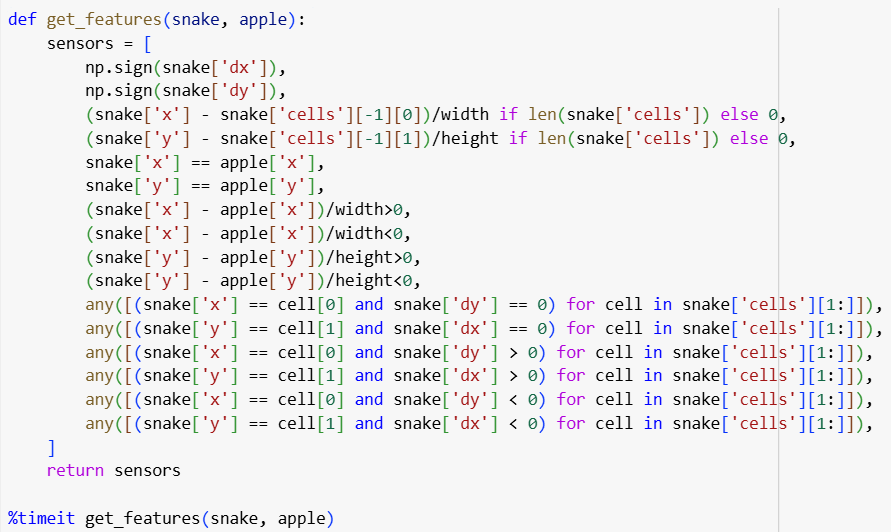
“Примитивщина”

137027.8

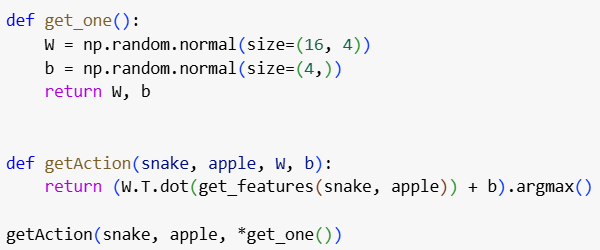


**Часть II**

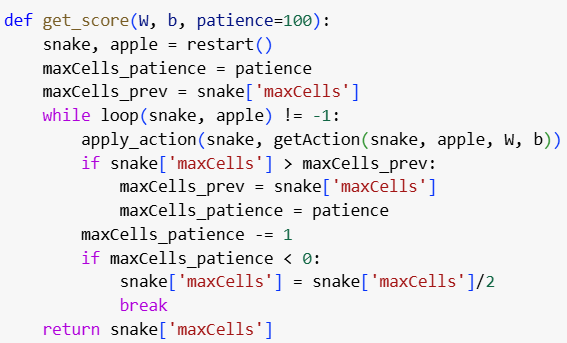
Для выполнения задания выбрал «Змейку». Сначала реализуем рабочую версию игры без использования генетического алгоритма. Далее займемся интеллектуальным агентом. Определим признаки, помогающие агенту лучше понимать окружение, чтобы избегать столкновений и лучше собирать яблоки:



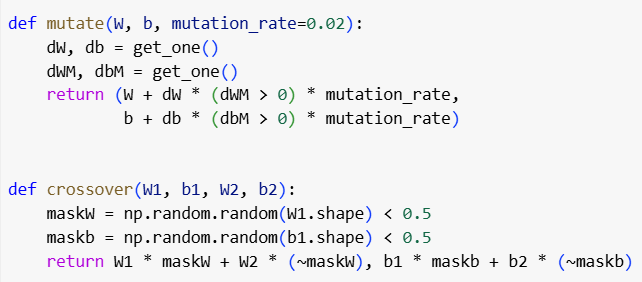
Зададим модель принятия решений на основе линейной комбинации признаков, которая нужна для обучения агента предсказывать действие, основываясь на текущем состоянии. Здесь определяется матрица весов и вектор смещений:



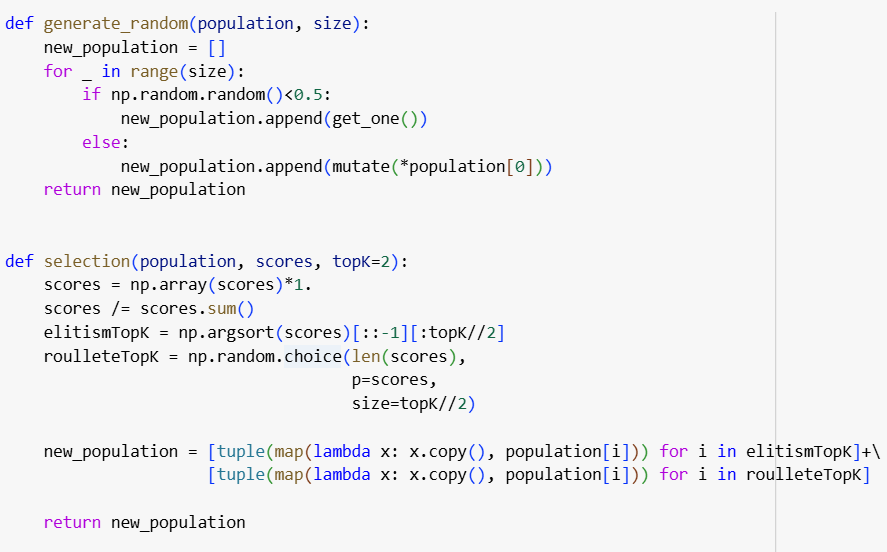
Напишем функцию приспособленности, которая будет проводить оценку и сравнение различных агентов на степень близости к оптимальному выполнения задачи:



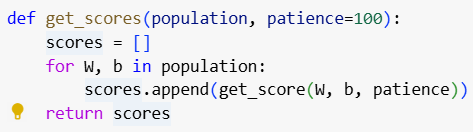
Далее добавим этапы генетического алгоритма. Сначала функции, реализующие мутацию и скрещивание, которые, соответственно, вносят случайные изменения и комбинируют родительские параметры для поиска оптимальных решений:



Затем функции, реализующие создание новой популяции, содержащей лучших потомков:



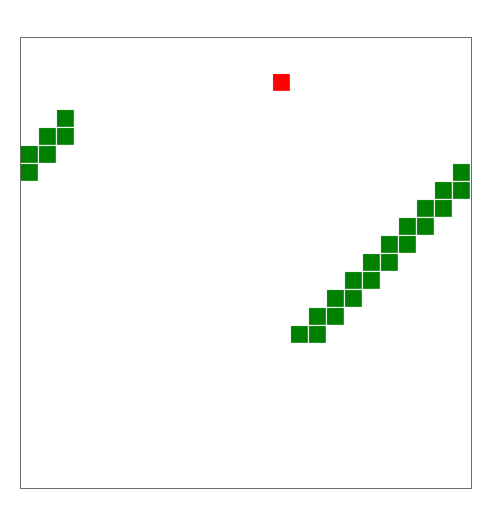
Введём функцию, возвращающую оценку текущей популяции:



И, наконец, обучим интеллектуального агента:



Запустим игру с агентом и посчитаем среднюю продолжительность жизни агента по результатам 3 партий:



В первой партии время 02:25, во второй 02:10, в третьей 03:30. Среднее время – 02:35. Отличный показатель.

**Вывод**

Разобрались в принципах работы генетического алгоритма, изучили, что представляет собой генетическое определение задачи и как оно связано с реализацией алгоритма. Разобрали функции, выполняющие мутацию, скрещивание, отбор и визуализацию результата программы. Приобрели навыки чтения и анализа программ, использующих генетические алгоритмы.