Optimal PUBG

Пасько Максим
pasko.mi@phystech.edu

Project Proposal

Прдеставьте, что вы резко захотели поиграть в PUBG. Но так как вы ~~нуб~~ совершаете только первые шаги в этой дисциплине, вы боитесь, что все вас будут убивать первым. Но ведь вы тоже хотите поиграть подольше и полутаться как следует. Тогда встаёт вопрос: куда лучше прыгнуть в начале игры, чтобы вас сразу не убили и вы насобирали крутой лут?

1 Идея

Введём функцию

$$f(x,y) = \frac{\mathbb{P}[\text{быть убитым в окрестности точки}(x,y)]}{\mathbb{P}[\text{найти хороший лут в окрестности точки}(x,y)]}$$

Тогда, наша задача сводится к минимизации этой функции на области

$$\Omega = \{(x,y) : (x,y) \text{ достижима при прыжке для данной траектории}\}$$

1.1 Problem

Скорее всего, наша функция будет иметь очень сложный вид, поэтому к ней нельзя будет применить даже методы 1-го порядка, поэтому надо будет пользоваться какими-то другими методами. Например, мы можем использовать симплекс-метод Нелдера-Мида, описанный в [1].

Также есть вариант использовать алгоритм имитации отжига, который хорошо описан (и определяется его сходимость) в [2], или же использовать алгоритм пчелиного роя (Bee algorithm), описанный в [3].

При этом, у игрока будет не больше одной минуты от момента подключения к игре (в этот момент он может увидеть траекторию самолёта), до момента, когда самолёт полетит и уже надо будет совешать прыжок. То есть, нам необходимо, чтобы наша программа выдавала пользователю ответ как можно быстрее, поэтому мы рассмотрим все эти алгоритмы оптимизации (а возможно и ещё какие-то!), и уже на основе полученных данных о времени работы (скорости сходимости), выберем какой-то как основной.

2 Outcomes

На выходе хочется получить точку приземления, где наиболее вероятнее заиметь ценный лут, и наименее вероятнее быть убитым. Удобно будет, чтобы игрок сразу получал точку на карте (см. рис 1).

3 Литературный обзор

Итак, нам нужно добыть информацию о двух темах: лут и киллы. Начнём с лута.
Есть достаточно качественная карта лута от [4] - это сайт, который посредствам сбора инфы от каждого игрока пытается помочь игрокам в победе (огромный процент игроков используют этот сайт, поэтому данных у него предостаточно и его информации можно верить), но всё же именно численных данных там нет, но зато немного таких данных можно найти в developer blog официального (!) сайта PUBG [5].

Теперь перейдём к киллам. Сайт [6], аналогичный сайту [4], как-то открыл данные по матчам, откуда их сразу же скачали, получив приличный датасет из 720.000 игр. Важно, что в нём есть отдельно показатели смертей игроков (такие данные, как координаты жертвы, убийцы, время смерти и тп.), поэтому из

Figure 1: Выход нашей программы - оптимальная точка приземления бойца (красная точка)

Algorithm 1 Nelder-Mead

Sort $f(x_1) \leq f(x_2) \leq ... \leq f(x_{n+1})$
$x_o = \frac{1}{n} \sum_{k=1}^{n} x_k$
while $||x_i - x_j|| \geq \varepsilon \quad \forall i, j$ **do**
  $x_r = x_0 + \alpha(x_0 - x_{n+1}), \quad \alpha > 0$
  if $f(x_r) < f(x_1)$ **then**
    go to expansion
  else if $f(x_1) \leq f(x_r) < f(x_n)$ **then**
    Update $x_{n+1} \leftarrow x_r$, go to step 1
  else if $f(x_n) \leq f(x_r)$ **then**
    go to contraction
  end if
  $x_e = x_0 + \gamma(x_r - x_0), \quad \gamma > 1$
  if $f(x_e) < f(x_r)$ **then**
    update $x_{n+1} \leftarrow x_e$, go to step 1
  else
    update $x_{n+1} \leftarrow x_r$, go to step 1
  end if
  $x_c = x_0 + \beta(x_{n+1} - x_0), \quad 0 < \beta \leq 0.5$
  if $f(x_c) < f(x_{n+1})$ **then**
    update $x_{n+1} \leftarrow x_c$, go to step 1
  else
    go to step 23
  end if
  $x_i = x_1 + \sigma(x_i - x_1) \quad \forall i \neq 1, \quad 0 < \sigma < 1$
end while

него можно попробавть получить полезную инфу, чтобы считать вероятность смерти в окрестности данной точки. При этом, что-то уже проанализировано. Например, в работе [7] можно наблюдать, что большинство убийств в игре на карте ERANGEL (это собственно та карта, для которой мы считает оптимальную точку) сделано с расстояния менее 30 метров. Карта (имеющая размер 8 на 8 км) разделена на 64 больших квадрата, каждый из которых разделён ещё на 64 квадрата. Отсюда делается вывод, что можно рассматривать вероятность смерти в рамках одного маленького квадрата (длина стороны которого 125м). Такие же выводы можно сделать и из работы [8].

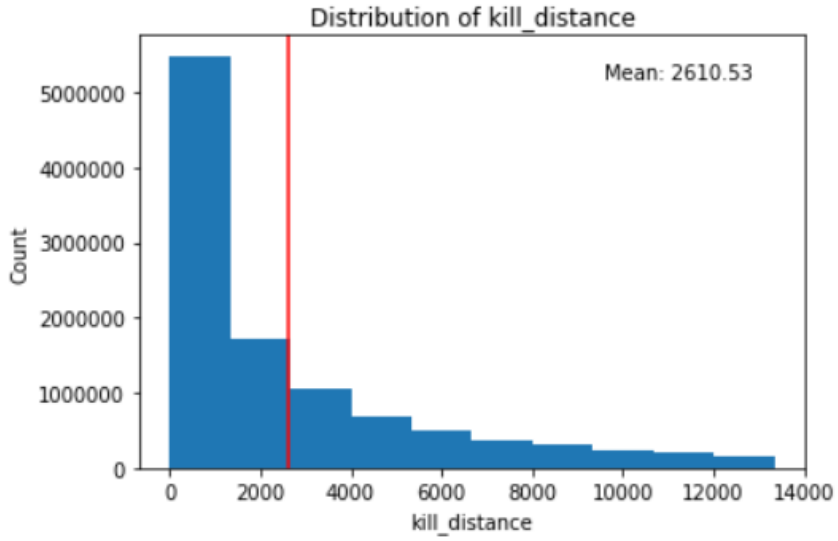

Figure 2: Анализ дистанции киллов на карте

4 Метрики качества

Для оценки качества работы программы, будем проводить запуски PUBGa на разных устройствах (скорее всего ~ 5), при этом все, кроме одного будут прыгать в случайное (или не совсем случайное) место, а один - используя совет программы. На каждой итерации будем фиксировать время жизни и количество (а ещё, что важнее, и качество) найденного лута, потом сравним между собой.

5 Примерный план

- Сначала получу распределение киллов по карте из датасета, потом надо будет получить карту лута (есть опасение, что её придётся заполнять вручную, от чего может пострадать точность), срок - где-то 16-18 апреля.
- Попробуем оптимизировать функцию несколькими алгоритмами, сравним времена работы на одной траектории, срок - 3-5 дней после предыдущего пункта.
- Попытаемся модернизировать вычисление вероятности смерти в квадрате с помощью приближения математическими распределениями для каждой траектории, посмотрим что из этого получится, протестируем программу на этом.
- Если успеется, попробуем добавить остальные карты (их всего 4, две из них размеров 8 на 8, оставшиеся размером 4 на 4).

References

[1] R. Mead J. A. Nelder. A simplex method for function minimization. The Computer Journal, 7(4):308–313, 1965.

[2] Simulated annealing algorithm kernel description. https://ieeexplore.ieee.org/document/295910, 1994.

[3] Ernesto Mastrocinque Duc Truong Pham Baris Yuce, Michael S. Packianather and Alfredo Lambiase. Honey bees inspired optimization method: The bees algorithm. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4553508/, 2013.

[4] Gosu.ai. https://gosu.ai/platform/ru/pubg/maps/erangel, 2020.

[5] Pubg official website. https://www.pubg.com/ru/category/dev-blog-ru/, 2020.

[6] Pubg.op.gg. https://pubg.op.gg/, 2020.

[7] Pubg kill analysis. https://www.kaggle.com/etsc9287/pubg-kills-analysis, 2020.

[8] Pubg finish placement prediction. https://www.kaggle.com/deffro/eda-is-fun#The-Killers, 2019.