# **Концепция оценки командного взаимодействия человека на примере командных компьютерных игр**

# Идея

Мы решили выбрать такую модель, при которой человеку в конкретно выбранной игре ставится оценка его командных действий из отрезка [0, 1], исходя из его результатов в этой игре. Мы составили алгоритм, который рассчитывает эту оценку из данных статистики игрока, которую мы можем получить из открытых источников. Для демонстрации мы взяли популярные игры нескольких жанров: *MOBA* (League of Legends), *FPS* (CS:GO, Rainbow Six Siege) и *Battle Royale* (Fortnite, COD:Warzone). Исходя из нашего обширного игрового опыта, мы отобрали те показатели, которые максимально объективно, насколько это возможно, отражают командные навыки игрока и рассчитали, как сильно эти показатели влияют на оценку, а также алгоритм расчета общей оценки человека из совокупности его оценок в разных играх.

# Подсчет по дереву

Принцип работы нашего алгоритма основан на использовании специального корневого дерева для оценки уровня командной игры каждого отдельного игрока *p*.

Каждый лист *L* в дереве имеет ассоциированную с ним константу *wL*, называющуюся Весом листа, со значением лежащем в (0, 1], и функцию *fL(p)*, называющуюся Выражением листа, ограниченную снизу нулём и сверху единицей. Оценкой листа *L* для игрока *p* называется функция *vL(p)* = *fL(p) \* wL*.

Каждый узел ветвления *U* в дереве имеет ассоциированную с ним константу *wU*, называющуюся Весом узла ветвления, со значением лежащем в (0, 1], и функцию   
*fU(D, p)*, где D – вектор Оценок всех потомков данного узла, называющуюся Выражением узла ветвления, ограниченную снизу нулём и сверху единицей. Оценкой узла ветвления *U* для игрока *p* называется функция *vU(p)* = *fU(D, p) \* wU*.

Вот функции, которые мы используем:

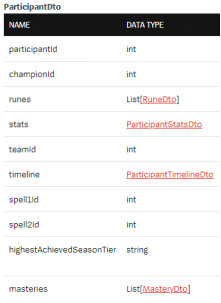
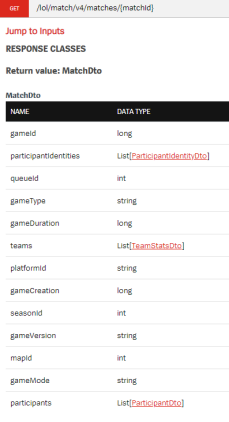
*max(D, p)* – наибольшее значение Оценки для игрока *p* среди *D*;

*sum(D, p)* – сумма значений Оценок для игрока *p* среди *D*;

*n(G) = norm(p, S, G) = pG / SG*, где *S* – Наибольшие значения среди всех наших игроков по всем величинам, *p* – Игрок, *G –* Графа в статистике (чтобы превратить значение параметра, лежащее в диапазоне луча, например от нуля до бесконечности, заключить в отрезок).

## League of legends

1) Получаем матч по его айди



Для статистики по каждому игроки нас интересует поле participants.

Это список информаций о каждом игроке.

Нас интересует поле **stats**

3) Общая статистика игрока в таком виде:

|  |  |  |
| --- | --- | --- |
| totalDamageTaken / min | long | Полученный урон |
| totalTimeCrowdControlDealt / min | int | Общее время нанесения контроля |
| totalDamageDealtToChampions / min | long | Урон, нанесенный по игрокам |
| ***totalMinionsKilled / min*** | float | Крипов в минуту |
| win | boolean | Победа или проигрыш |
| totalHeal / min | long | Вылеченное здоровье (всем игрокам) |
| visionScore / min | long | Счет по обзору (в минуту) |
| ***kda*** | float | = (kills+assists)/deaths |

Каждый матч вычисляется по схеме:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***РОЛЬ*** | Топ | Лес | Мид | Нижняя линия | Саппорт |
| ***totalDamageTaken / min*** | MAX - 20%  20% \* (x/1000)  (>=1000 - 20%) | MAX - 20%  20% \* (x/1000)  (>=1000 - 20%) | 0% | 0% | MAX - 10%  10% \* (x/700)  (>=700 - 10%) |
| ***totalTimeCrowdControlDealt / min*** | MAX - 10%  10% \* (x/4s)  (>=4s - 10%) | MAX - 10%  10% \*  (x/4s)  (>=4s - 10%) | MAX - 10%  10% \* (x/4s)  (>=4s - 10%) | MAX - 5%  5% \* (x/2s)  (>=3s - 5%) | MAX - 20%  20% \* (x/6s)  (>=6s - 20%) |
| ***totalDamageDealtToChampions / min*** | MAX - 20%  20% \* (x/800)  (>=800 - 20%) | MAX - 20%  20% \* (x/800)  (>=800 - 20%) | MAX - 30%  30% \* (x/1100)  (>=1100 - 30%) | MAX - 30%  30% \* (x/1100)  (>=1100 - 30%) | MAX - 10%  10% \* (x/500)  (>=500 - 10%) |
| ***totalMinionsKilled / min*** | MAX - 20%  20% \* (x/10)  (>=10 - 20%) | MAX - 10%  10% \* (x/6)  (>=6 - 10%) | MAX - 30%  30% \* (x/12)  (>=12 - 30%) | MAX - 30%  30% \* (x/12)  (>=12 - 30%) | 0% |
| ***totalHeal*** / min | 0% | 0% | 0% | MAX - 5%  5% \* (x/200)  (>=200 - 5%) | MAX - 15%  15% \* (x/700)  (>=700 - 15%) |
| ***visionScore / min*** | MAX - 10%  10% \* (x/1)  (>=1 - 10 %) | MAX - 20%  20% \* (x/2)  (>=2 - 20%) | MAX - 10%  10% \* (x/1)  (>=1 - 10 %) | MAX - 10%  10% \* (x/1)  (>=1 - 10 %) | MAX - 30%  30% \* (x/3)  (>=3 - 30%) |
| ***kda*** | MAX - 20%  20% \* (x/3.5)  (>=3.5 - 20%) | MAX - 20%  20% \* (x/3.5)  (>=3.5 - 20%) | MAX - 20%  20% \* (x/3.5)  (>=3.5 - 20%) | MAX - 20%  20% \* (x/3.5)  (>=3.5 - 20%) | MAX - 15%  15% \* (x/3)  (>=3 - 15%) |

Общая статистика равна среднему арифметическому всех значений по каждому матчу по следующей схеме

20% - WIN

80% - STATS

## CS:GO

Чтобы получить статистику игрока в CS:GO (самая распространённая игра в агрегаторе игр Steam) необходимо сделать несколько простых запросов к официальному API Steam(), которое может возвращать статистику игрока в тех играх, которые дают доступ к своим статистикам. Соответственно CS:GO даёт доступ к статистике игрока через API Steam. Токен пользователя Steam можно получить с помощью внешней аутентификации. Вот какие параметры мы можем получить с помощью API и выделить в качестве тех, которые отражают навыки командного взаимодействия игрока:

|  |  |
| --- | --- |
| **ПАРАМЕТР** | **ЗНАЧЕНИЕ** |
| **total\_time\_played** | общее время игры |
| **total\_wins** | суммарное количество выигранных раундов |
| **total\_rounds\_played** | суммарное количество раундов |
| **total\_mvps** | количество раундов, когда игрок становился MVP(most valuable player) |
| **total\_matches\_won** | суммарное количество выигранных матчей |
| **total\_matches\_played** | суммарное количество матчей |
| **total\_weapons\_donated** | суммарное количество случаев пожертвования своего оружия |

Соответственно пересчет баллов следующий:

* 30% если total\_matches\_played > 10, иначе 0:
  + 100% total\_matches\_won / total\_matches\_played winrate матчей
* 15% total\_mvps
* 10% total\_weapons\_donated
* 15% если total\_rounds\_played > 40, иначе 0:
  + 50% total\_wins / total\_rounds\_played winrate раундов
  + 20% total\_rounds\_player / total\_matches\_played avg раундов на матч
  + 30% total\_wins / tota\_matches\_won avg выигранных раундов на матч
* 30% total\_matches\_won

## Rainbow Six Siege

Статистику в игре R6 можно получить с помощью готовых сайтов со статистикой(на примере <https://r6.tracker.network>), а также с помощью неофициального API, доступного для Python(<https://pypi.org/project/r6sapi/>) и Java(<https://github.com/billy-yoyo/RainbowSixSiege-Python-API>). Токен игрока можно получить из внешней аутентификации. С помощью API нам доступны следующие параметры:

|  |
| --- |
| time played |
| rank |
| MMR |
| **RANKED / UNRANKED:** |
| wins / losses |
| matches played |

В игре предусмотрено два типа матчей: RANKED и UNRANKED.

Соответственно пересчет баллов следующий:

* 70% для RANKED, если matches\_played(r) > 10, иначе 0:
  + 40% wins / losses
  + 40% wins
  + 20% MMR / wins
* 20% для UNRANKED, если matches\_played(unr) > 5, иначе 0:
  + 60% wins
  + 40% wins / losses
* 10% :
  + 40% rank
  + 60% MMR

## Fortnite

Статистику можно получить через неофициальный API-сервер <https://fortnite-api.com>. Для получения необходимой статистики для игрока по его никнейму нужно обратиться к методу GET [https://fortnite-api.com/v1/stats/br/v2?name={username](https://fortnite-api.com/v1/stats/br/v2?name=%7Busername)}. В качестве ответа придет JSON содержащий данные.

|  |  |
| --- | --- |
| **ПАРАМЕТР** | **ЗНАЧЕНИЕ** |
| **duo.matches** | Количество сыгранных в паре матчей |
| **duo.wins** | Количество выигранных в паре матчей |
| **duo.top5** | Количество топ-5 в паре матчей |
| **duo.top12** | Количество топ-12 в паре матчей |
| **trio.matches** | Количество сыгранных в тройке матчей |
| **trio.wins** | Количество выигранных в тройке матчей |
| **trio.top3** | Количество топ-3 в тройке матчей |
| **trio.top6** | Количество топ-6 в тройке матчей |
| **squad.matches** | Количество сыгранных в скваде матчей |
| **squad.wins** | Количество выигранных в скваде матчей |
| **squad.top3** | Количество топ-3 в скваде матчей |
| **squad.top6** | Количество топ-6 в скваде матчей |

### Схема распределения баллов

* 25% - duo, если duo.matches > 6, иначе 0
  + 50% - wins / matches
  + 25% - top5 / matches
  + 25% - top12 / matches
* 35% - trio, если trio.matches > 5, иначе 0
  + 50% - wins / matches
  + 25% - top3 / matches
  + 25% - top6 / matches
* 40% - squad, если squad.matches > 4, иначе 0
  + 50% - wins / matches
  + 25% - top3 / matches
  + 25% - top6 / matches

### Пояснение к схеме

Для подсчета баллов по статистике Fortnite необходимо сложить баллы за игру в duo, trio и squad, умножая их на соответствующие коэффициенты. В свою очередь, баллы за каждую из этих категорий определяется таким же образом, то есть суммированием, однако уже напрямую считаемых величин.

Каждая величина (например, top5 / matches из категории duo) определяется следующим образом: находится наибольшее и наименьшее значение этой величины среди всех игроков, затем значение величины у конкретного игрока мапируется из диапазона найденных границ у величины в диапазон [0; 1]; мапированная величина top5 / matches получит обозначение n(top5 / matches). Пример: среди всех зарегистрированных нами пользователей наибольшее значение величины top5 / matches составило 0.2, а наименьшее значение – 0.0, тогда человек с top5 / matches = 0.15, будет иметь n(top5 / matches) = 0.75.

## Call of Duty: Warzone

Статистику можно получить с игровых серверов callofduty.com с помощью обертки, написанной под nodeJS. Чтобы использовать этот модуль из python, мы будем использовать js2py. Для получения необходимой статистики для игрока по его никнейму нужно обратиться к методу API.MWBattleData(<gamertag>, <platform>?). функция вернет JSON содержащий данные.

|  |  |
| --- | --- |
| **ПАРАМЕТР** | **ЗНАЧЕНИЕ** |
| **br\_brduos.matches** | Количество сыгранных в паре матчей |
| **br\_brduos.wins** | Количество выигранных в паре матчей |
| **br\_brduos.top5** | Количество топ-5 в паре матчей |
| **br\_brduos.top12** | Количество топ-12 в паре матчей |
| **br\_brtrios.matches** | Количество сыгранных в тройке матчей |
| **br\_brtrios.wins** | Количество выигранных в тройке матчей |
| **br\_brtrios.top3** | Количество топ-3 в тройке матчей |
| **br\_brtrios.top6** | Количество топ-6 в тройке матчей |
| **br\_brsquads.matches** | Количество сыгранных в скваде матчей |
| **br\_brsquads.wins** | Количество выигранных в скваде матчей |
| **br\_brsquads.top3** | Количество топ-3 в скваде матчей |
| **br\_brsquads.top6** | Количество топ-6 в скваде матчей |

### 

### Схема распределения баллов

* 25% - br\_brduos, если br\_brduos.matches > 6, иначе 0
  + 50% - wins / matches
  + 25% - top5 / matches
  + 25% - top12 / matches
* 35% - br\_brtrios, если br\_brtrios.matches > 5, иначе 0
  + 50% - wins / matches
  + 25% - top3 / matches
  + 25% - top6 / matches
* 40% - br\_brsquads, если br\_brsquads.matches > 4, иначе 0
  + 50% - wins / matches
  + 25% - top3 / matches
  + 25% - top6 / matches

## Пояснение

Для большей ясности алгоритма формирования оценки для игрока, мы нарисовали древовидную схему, где описано распределение коэффициентов для каждого параметра статистики. Она содержится в файле stat\_tree.pdf.