

League of Legends における初心者向け上達補助可視化システム A Visualization System for Supporting Beginners of League of Legends in Improving their Skills

土屋 和

Kanau Tsuchiya

法政大学情報科学部コンピュータ学科

E-mail: kanau.tsuchiya.6u@stu.hosei.ac.jp

Abstract

League of Legends has been one of the most popular multi-player online battle arena games. However, the complexity of this game makes it difficult for beginners to keep playing. Previous studies analyzed the statistics of the game such as coordinates and other parameters of players. Some studies estimated how well players performed in matches and which action would lead them to win. Another study used storyline visualization that included temporal data and other information, and another analyzed an entire match. However, research on players' skills at the early time of the match did not exist. In this paper, we create a visualization system for game beginners to improve their playing skills using heatmaps. We focus on damage trades that happen early in the match. A damage trade is the action of widening a gap in health or resources between the player and the opponent by using the player's skills and automatic attacks. Beginners often feel it difficult to understand the skill ranges and damage of each character. By reviewing their damage trades, they can improve their skills at the lane phase and play the matches longer to enjoy the game.

1. はじめに

近年、若年層の間でPCゲームが急速に普及した。マルチプレイヤーオンラインバトルアリーナ (MOBA) である League of Legends (LoL) もその一例である (図 1(a))。MOBA 内には多くの変数が存在し、その複雑性によってなされるゲーム体験が魅力の 1 つとなっている。しかしその複雑性はゲームを始めた初心者にとっては理解が難しく、初級者との間に大きな差ができ、少しのミスをするだけで有利を取られる。これにより初心者は MOBA ゲームの魅力を体感することなくゲームをやめてしまう。

LoL では 2 つのチームが対戦し、相手チームの最終拠点を破壊することを目指す。プレイヤーは一定時間ごと、またはユニットを倒すことで経験値やゴールドを得られ、それによりステータスが上昇し、戦闘を優位に進められる。LoL は試合前半のレーン戦と後半の集団戦に分けられる。初心者はレーン戦で大きな差をつけられ諦めることが多い。レーン戦とは、LoL のマップ上にあ

る 3 つの道でそれぞれのチームのプレイヤーが戦闘を行うイベントである (図 1(b))。

本研究では、初心者の上達を支援するシステムを開発することで、初心者がゲームの魅力を知り、プレイを続けてもらうことを目的とする。レーン戦の要素の 1 つであるダメージトレードに着目する。初心者はこのダメージトレードに勝てないことが多く、そのまま相手に有利を広げられ、試合途中で勝利を諦めてしまう。本研究では、各座標におけるキャラクターの任意のコンボのダメージ量をスキルの射程を重ね合わせ表示してヒートマップで可視化し、入力したコンボのダメージ量が時間経過でどのように変化したかを可視化することで、プレイヤーがダメージトレードの結果の要因を理解し、レーン戦における立ち回りを振り返られる技術上達補助のシステムを開発する。

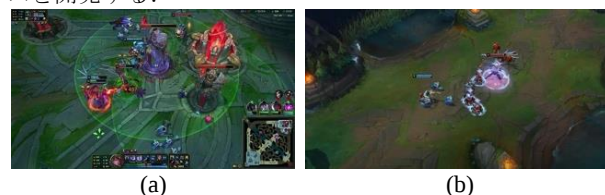


図 1 League of Legends の(a)プレイ風景, (b)レーン戦の様子

2. 関連研究

Wallner らはイベントのタイミングなどの時間的要素とプレイヤーの位置などの空間的要素を統合したストーリーラインを用いて、チームの戦略や協調性の分析を行い、プレイヤーの技術上達支援を試みた [1]. Maymin は LoL におけるチャンピオンの位置、マップの見える範囲やマナ、体力、与えたダメージなどをリアルタイムで追跡、また新たにスマートキルとワースレスデスの 2 つの指標を追加して各プレイヤー行動がどれほどチームの勝利に影響したかを分析し、その時点での勝率を予測する勝率モデルを作成した [2]. これらのように試合全体の情報から試合を分析し、技術向上支援を試みた研究はあったが、本研究のようにレーン戦に着目した研究は著者が知る限り存在しない。

3. LoL のキャラクターと攻撃方法

LoL ではキャラクターごとに自分で発動させる 4 種類のスキルと常時発動するパッシブスキルが存在する (図 2) . 相手にダメージを与えるものや自分自身を強化するもの、逆に相手を弱体化させるものなど多岐にわたる。自分で発動させるスキルは初期設定で Q, W, E, R のキーが割り当てられており、しばしばスキルはその対応するキーで呼ばれる。また通常攻撃は AA (Auto Attack) と呼ばれる。

ダメージトレードとはスキルや通常攻撃を駆使して自分と相手が互いに攻撃し合う行為であり、基本的にダメージトレード終了時点で体力が多いほうが、ダメージトレードに勝利したことになる。ダメージトレードで有利を得るには自分と相手のそれぞれのスキルや通常攻撃のダメージや射程などを把握し、立ち回ることが重要である。

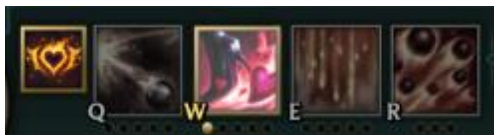


図 2 キャラクターのスキル

4. 提案手法

4.1. 可視化手順

相手が自分に与えられるダメージ量をステータスや装備、スキルの有無から計算し、自分と相手のどちらがダメージトレードで勝利するかを特定して可視化する。これにより自分がどのあたりに位置を取ればダメージトレードに勝利できるかを把握でき、良いプレイの再現性が向上すると考える。計算したダメージ量を比較して、各座標のダメージ量をマップ上にヒートマップとして出力する (図 3) . またヒートマップとは別に任意の座標で入力したコンボの時間経過によるダメージの変化を表すグラフを作成する (図 4) . このグラフは先に作成したヒートマップからダメージの推移を知りたい座標を入力することでその座標でのそれぞれのキャラクターのダメージ推移が表示される。

具体的な手順は以下の通りである。

- ① 試合データやキャラクター、アイテムのデータを取得する。
- ② ユーザーが座標やコンボを入力、射程を考慮し、各座標のダメージ量を計算する。
- ③ 計算したダメージ量から差を求めヒートマップとして出力する。
- ④ 出力したヒートマップ上から座標をユーザーに入力させ、その地点でのダメージ推移をグラフとして出力する。

ダメージ推移グラフに関しては、ヒートマップでの計算と同じものを任意の座標で行い、時間とダメージ量を記録しプロットを行う。

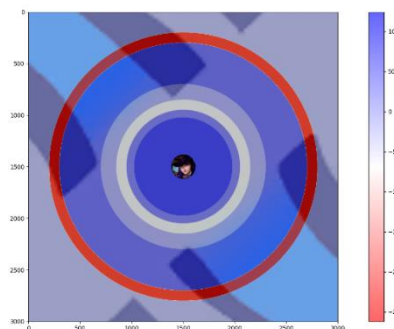


図 3 エズリアル対ケイトリンのヒートマップ

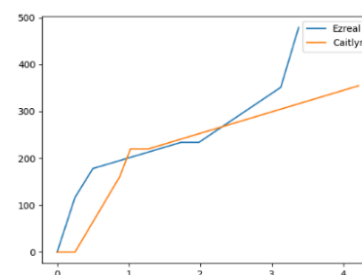


図 4 エズリアル対ケイトリンのダメージ推移グラフ

4.2. ダメージの計算

自分が操作するキャラクターの座標 (x_1, y_1) , 相手が操作するキャラクターの座標を (x_2, y_2) とする。ただし x_2, y_2 は定数とし、ヒートマップ上では相手が図の中央に位置する。キャラクター間の距離を

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

とすると、自分のダメージ量 $\text{damage_all}_1(d)$ は入力したコンボより以下のように計算する。

$$\text{damage_all}_1(d) = \sum_i (\text{skill_damage}_i \times \text{skill_count}_i) + \text{AA_damage} \times \text{AA_count}$$

ここで、 skill_damage_i は各スキルのダメージ量、 AA_damage は通常攻撃のダメージ量、 skill_count_i はスキルの使用回数、 AA_count は通常攻撃の回数を表す。

$$\text{skill_damage}_i = \begin{cases} \text{skill_damage}_i & (d \leq \text{skill_range}_i) \\ 0 & (\text{skill_range}_i < d) \end{cases}$$

$$\text{AA_damage} = \begin{cases} \text{AA_damage} & (d \leq \text{AA_range}) \\ 0 & (\text{AA_range} < d) \end{cases}$$

スキルの射程内に相手のキャラクターが位置していればダメージを計算し、射程外の場合は当たらないので、 skill_damage_i は 0 になる。 skill_damage_i や AA_damage は常に一定ではなくスキルの使用やパッシブスキルで変化する。例えばジンというキャラクターのパッシブスキルは AA の 4 回ごとに相手の減少体力割合のダメージを追加で与える。ヒートマップを作成するために、以下の damage_diff を計算して各座標にプロットする。

$$\text{damage_diff} = \text{damage_all}_1 - \text{damage_all}_2$$

コンボが入力された際の例を示す。プレイヤー 1 をエズリアル、プレイヤー 2 をケイトリンとする。これらのキャラクターのコンボは両方とも AA, Q とする。各スキ

ルの射程, ダメージは表 1 の通りであり, $d = 600$ としたときを計算する. この時エズリアルは AA は射程外で Q は射程内であるから,

$$\text{damage_all}_1(600) = 98 \times 1 + 0 \times 1 = 98$$

となる. 同じようにケイトリンは AA と Q とともに射程内なので,

$$\text{damage_all}_2(600) = 125 \times 1 + 60 \times 1 = 185$$

よって以下が得られる.

$$\text{damage_diff} = 98 - 185 = -87$$

表 1 AA と Q のそれぞれの射程

	エズリアル AA	エズリアル Q	ケイトリン AA	ケイトリン Q
射程	550	1200	650	1300
ダメージ	60	98	60	125

4.3. 例

自分をエズリアル, 相手をケイトリンとし, エズリアルコンボ Q, E, AA, W, AA, Q, ケイトリンコンボ W, Q, E, W, AA, AA を例にヒートマップとダメージ推移グラフを出力した (図 3, 図 4). 中央に表示されているアイコンはケイトリンである. 出力したヒートマップから今回のコンボ設定でエズリアルが有利を取れるのは, エズリアル Q, W の射程内かつケイトリンの W, E が射程外の位置, エズリアル E が射程内の位置のときである. 前者と後者のダメージ量の差を比較すると, 前者の方が大きな差をつけられる. また後者の場合は相手に近づく必要があり, その過程で相手が有利な領域を通過するため, 良い選択ではないと考える. そのため前者の位置を維持することでレーン戦を優位に進められると考察する. またケイトリン側は W, E の射程内からトレード, エズリアルに近づかれ過ぎないようにし, 加えて W, E の射程外にエズリアルが行かないような立ち位置の方針を取る.

ダメージ推移グラフから読み取れるのは, 0 から 1 秒の時はエズリアルが大きく有利を取っていて, 1 から 2 秒の間ではケイトリンが微かに有利, その後はまたエズリアルが有利を取り戻していることである. これよりエズリアルはダメージトレードを早めに切り上げる, もしくは一度 Q を使った後にもう一度 Q が使えるようになるまで戦闘を行うことでダメージ量で優位に立てる.

5. 実装

実装には Python, ライブラリは主に Matplotlib, NumPy, PIL, tkinter, OpenCV を使用する. 試合データとキャラクターのデータは LoL を開発, 運営している Riot Games 社が提供している Riot Games API と Data Dragon より取得する. 本研究ではボットレーンでよく使用されるアッシュ, エズリアル, ケイトリン, ジンの 4 体のキャラクターを実装した.

GUI は 6 ページある. 1 ページ目ではユーザーのゲーム内 ID や API キーの入力を行う. API キーについては自動で取得することが難しいため, ユーザー自身が実際に取得し, 入力してもらう. 1 ページ目でユーザー情報を入力すると, そのユーザーの直近 20 試合のマッチ ID が取

得でき, どのマッチを扱うか選択する. 選択したマッチ ID から, 自分がキルした, キルされたタイムスタンプをタイムライン上に表示する. 上側は自分がキルしたタイムスタンプで, 下側は自分がキルされたタイムスタンプである. 自分が扱いたい戦闘のタイムスタンプをクリックすることでチャンピオンの情報が登録され次に進む. このページではキャラクターの選択, キャラクター自身やスキルごとのレベルの選択を行う. また相手の座標は API で取得できなかったため, このページで入力を行う. このページではキャラクターの持っているアイテムの選択, コンボの入力を行う. その後計算を行い, ヒートマップとダメージ推移グラフを表示する.

6. 実験

22 歳から 25 歳までの 6 人の被験者に実際にシステムを使ってもらい, その前後でどのような変化を分析し, またいくつかのアンケートに回答してもらった. 被験者のランクは, ランクを未プレイが 1 人, アイアンが 1 人, ブロンズが 3 人, プラチナが 2 人である. 4 体のキャラクターを実装したが, ジンのみ AI 戦で選択ができなかったため, 残りの 3 体で 6 通りの組み合わせを作成し, 被験者 1 人につき, 1 つの組み合わせで実験を行った. 本来は実際の試合に近づけるためにボットレーンで実験を行うべきだったが, ボットレーンで行うと AI が他のレーンに行ってしまう不具合があったため, 本実験はミッドレーンで行った.

6.1. 手順とアンケート内容

具体的な手順は以下の通りである.

- ① 実装したキャラクターを用いて LoL のゲーム内の AI と対戦を行う. この時プレイ画面を録画する.
- ② 対戦終了後に作成したシステムを自由に利用してもらい, 対戦を振り返ってもらう.
- ③ その後再度同じ条件で対戦を行う. ここでもプレイ画面を録画する.
- ④ 録画したプレイ画面と各種値からどのような変化があったか分析をする.
- ⑤ 被験者にアンケートを行う.

加えて, 使用前と後の変化をわかりやすくするためにルーン, 購入するアイテム, サモナーズスペルを固定する制限をした.

アンケートの質問は以下の 5 段階評価の 4 問と自由記述のコメントである.

- Q1: 図からスキルの射程やダメージを理解できたか
- Q2: 図から入力したコンボがどの地点で有利か理解できたか
- Q3: 使用後どのように立ち回りを変えればよいか理解できたか
- Q4: 使用前の自分の改善点が理解できたか

6.2. 結果

アンケートの結果 (図 5) は 4 問すべてで 4 以上の評価を過半数得た. 自由記述では被験者 A から「理解できて実行が難しい」, 被験者 C から「積極的な立ち回りによってタワーの破壊時間が早くなったが, デス数が増え

た」，被験者 E から「スキルを使い切ることでそのキャラクターの真価を発揮できることがわかった」などのコメントを受けた。

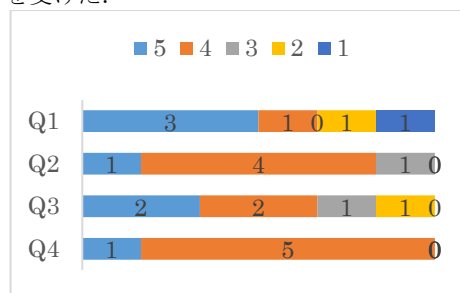


図5 アンケート結果

映像からの分析結果（表2）での各値は以下の通りである。

- ・キルは自分が相手のキャラクターを倒した回数，デスは自分が倒された回数である。キルが多く，デスが少ないほうが有利となる。
- ・所要時間は，相手のタワーを破壊するまでの時間である。相手のタワーを破壊するとゴールドを得られる。所要時間は短いほうが先に有利を取れる。
- ・CSは自分がキルしたミニオン数のことである。ミニオンをキルすることで経験値やゴールドを得られ，LoLでは倒したミニオンを時間で割った分間CSがよく指標として用いられる。ダメージトレードで勝利することはCSの値に直結する。
- ・リコールとは自分の本拠地に戻ることである。自分の本拠地では自分の体力を最大まで回復することができ，アイテムを購入できる。相手をキルできなくてもダメージトレードをして体力を削り，リコールさせることで有利を得られる。

表2 映像からの各値

	キル	デス	所要時間	CS (分間)	リコール回数
被験者 A(前)	5	1	14:15	76 (5.3)	3
被験者 A(後)	4	1	13:30	91 (6.7)	3
被験者 B(前)	6	0	14:24	97 (6.7)	1
被験者 B(後)	6	0	14:58	108 (7.2)	3
被験者 C(前)	7	0	14:16	95 (6.7)	1
被験者 C(後)	4	2	12:47	87 (6.8)	2
被験者 D(前)	6	0	11:17	97 (8.6)	3
被験者 D(後)	7	0	11:47	109 (9.3)	1
被験者 E(前)	3	0	12:56	94 (7.3)	3
被験者 E(後)	6	0	10:25	76 (7.3)	2
被験者 F(前)	6	0	10:25	79 (7.6)	3
被験者 F(後)	6	0	10:23	83 (8.0)	2

これら4つの項目に関して6人全員いずれかの値で上昇が見られた。キルは2人が上昇，所要時間は4人が上昇，CSは5人が上昇，リコール回数は2人が上昇した。

しかし，これらについて対応のあるt検定を行った結果，キル ($p = 0.84$)，所要時間 ($p = 0.27$)，CS ($p = 0.053$)，リコール回数 ($p = 0.79$) にはいずれにも統計的に有意な差はなかった。

7. 議論

アンケートの結果について，1または2の評価を受けたQ1とQ3に関して考察する。Q1に関しては，入力をコンボとしていたため，1つのスキルのみを入力すればスキルの射程やダメージが分かるが，それに気づかなければ個々のスキルの射程やダメージを把握することが難しかったためだと考える。Q3については，コンボをいくつか入力して図を見てみなければ立ち回りの方針を立てられないため，試行回数が少なかったと考える。

被験者の映像からの分析について考察する。キル，デス，リコール回数については有利なポジションを取り続けたことで，それらの値を増やしたと考える。所要時間の短縮やCSについて，ダメージトレードで体力有利を作り，所要時間に関しては相手を前に出にくくすることでタワーを攻撃する時間を増やしたと考える。CSに関しては，相手がスキルやAAをミニオンに当てられない距離まで退かせることで有利を得たと考える。

1人だけデス数が増えていた被験者Cの結果について考察する。キル，デスの値は下がっているが，これは被験者Cがコメントで述べたように，振り返りから積極的に立ち回ったが，慣れておらずキルされてしまったと推測される。しかし積極的に立ち回ったため，リコール回数の値は上昇している。ダメージ量の内訳を表示し，どのスキルを当てればよいか，避ければよいかを表示することで，積極的に立ち回った場合のリスクを軽減できると考える。

現状のシステムでは，マップの地形については考慮しないで計算している。これは過半数のスキルが射程内であれば地形を無視して，相手を攻撃できるからである。また現在の提案手法ではすべてのキャラクターのスキルに対応することは難しい。理由としてはチャンピオン以外のユニットに作用するスキルなどは今回の手法で実装することが難しいためである。また，ルーンやサモナーズスペル，一部ステータスや一部アイテムについて実装ができなかった。今後，それらを実装することで，より正確な可視化を実現できると考える。

8. おわりに

本研究ではLoL内のレーン戦の要素であるダメージトレードの技術上達を支援することを目的とし，APIやゲーム運営会社が提供するデータを利用して，入力した情報からダメージ量を計算し，ヒートマップとグラフに可視化するシステムを構築した。

今後の課題は，現状で扱っていない要素の実装，ダメージ量の内訳の表示，コンボだけでなく各スキルのみもヒートマップで可視化することである。これにより効率的にプレイヤーの上達を支援できると考える。

文 献

- [1] G. Wallner, L. Wang and C. Dormann, "Visualizing the Spatio-Temporal Evolution of Gameplay using Storyline Visualization: A Study with League of Legends," *Proceedings of the ACM on Human-Computer Interaction*, vol. 7, no. CHI PLAY 2023, pp. 412:1-23, 2023.
- [2] P. Z. Maymin, "Smart kills and worthless deaths: eSports analytics for League of Legends," *Journal of Quantitative Analysis in Sports*, vol. 17, no. 1, pp. 11-27, 2020.