

Entwicklung eines Visual Analytics Tools zur Vor- und Nachbereitung für Casual League of Legends Spieler auf Basis von Match-Daten

Sandra Kiefer*

Ferdinand Meyer†

Jan Niklas von Zwehl‡

Hochschule RheinMain - Fachbereich Design Informatik Medien
Studiengang Informatik (Master of Science)

ZUSAMMENFASSUNG

League of Legends ist ein Computerspiel mit einer sehr großen und stetig wachsenden Spielerbasis. Um Anfängern und Freizeitspielern eine Option anzubieten, mit der sie ihre eigenen Spiele analysieren können, wurde im Kontext einer Projektarbeit eine Webanwendung zur Vor- und Nachbereitung der eigenen Spiele entwickelt. Diese Anwendung wurde anschließend von League of Legends Spielern im Hinblick auf ihre Nützlichkeit und Benutzerfreundlichkeit getestet. In diesem Paper wird der gesamte Ablauf des Erstellungsprozesses der Anwendung, von der Hintergrundrecherche über die Designkonzeption bis zur tatsächlichen Implementierung, beschrieben. Anschließend werden die Ergebnisse des Nutzertests präsentiert. Die Auswertung der Befragung zeigt, dass das entwickelte Analysetool sowohl hilfreiche Informationen als auch eine geeignete Darstellungsform hat.

1 EINFÜHRUNG

League of Legends ist eines der weltweit beliebtesten Online-Multiplayer-Videospiele mit rund 150 Millionen Spielern pro Monat [1]. Da es sich bei League of Legends um ein Live-Service-Spiel handelt, das sich ständig weiterentwickelt und wächst, ist es für Spieler, die das Spiel nicht täglich spielen oder gerade erst mit dem Spiel begonnen haben, schwierig, den Überblick darüber zu behalten, was in ihrem Spiel passiert und wie sie sich verbessern können. Hier können visuelle Analyse-Tools eine entscheidende Rolle spielen, indem sie den Spielern helfen, ihre Leistung besser zu verstehen und zu lernen, wie sie ihr Spiel verbessern können.

In diesem Beitrag stellen wir ein visuelles und interaktives Analyse-Tool vor, das speziell für League of Legends auf der Karte „Summoners Rift“ entwickelt wurde. Das Tool nutzt verschiedene Datenvisualisierungstechniken, um Spielern einen umfassenden Überblick über ihr Spiel und Einblicke in ihre Stärken und Schwächen zu geben. Das Tool bietet dem Nutzer Spielerstatistiken, eine interaktive Karte zur Veranschaulichung von Ingamedaten und Tipps für zukünftige Spiele.

Durch eine Nutzerbefragung einer kleinen Gruppe von League of Legends Spielern haben wir herausgefunden, inwiefern diese Anwendung einen positiven Mehrwert für Spieler haben kann. Es wird untersucht, ob dieses Tool auch auf dem bestehenden Markt der Analysetools einen echten Mehrwert bieten kann.

Im Rahmen der Entwicklung dieses Analysetools und dieser Ausarbeitung haben wir uns zum Ziel gesetzt, herauszufinden, ob in einem solchen Analysetool die raum-zeitliche Auswertung von ereignisgesteuerten Daten einen Mehrwert für den Benutzer bietet. Ebenso wollten wir herausfinden, welche Daten aus der Riot API verwendet werden können, um ein Analysetool zu entwickeln.

*E-Mail: sandra.kiefer@student.hs-rm.de

†E-Mail: ferdinand.meyer@student.hs-rm.de

‡E-Mail: janniklas.vonzwehl@student.hs-rm.de

1.1 Lösungsansätze und Methodiken

Ausgangspunkt für die Entwicklung des Visual Analytic Tools sind die von der Riot API zur Verfügung gestellten Daten. Über eine eindeutige User-ID können beliebig viele Spiele abgefragt werden. Diese Daten sind unterteilt in „Spieler-Daten“, die Gesamtwerte für jeden Spieler eines Spiels enthalten, und „Match-Daten“, die in chronologischer Reihenfolge zu jedem Ereignis im Spiel die entsprechenden Informationen enthalten. Ziel ist es, den größtmöglichen Nutzen für die Spieler aus den gesammelten und abrufbaren Daten zu ziehen.

Zu diesem Zweck wurde ein erster Prototyp implementiert. Dieser beschäftigt sich mit der Frage, wie die Daten aufbereitet und für den Nutzer ansprechend visualisiert werden können. Insbesondere für die komplexeren raum-zeitlichen Informationen wird eine geeignete Darstellungsform gefunden. Die Daten werden mit Python-Programmcode abgefragt und mit dem Python-Framework Streamlit [20] in Form einer Webanwendung visualisiert. Zudem wird die Webseite in mehrere Unterseiten mit unterschiedlichen Themen schwerpunkten aufgeteilt, die wiederum „Dashboard-artige“ Visualisierungen enthalten. Auf jeder Unterseite hat der Nutzer die Möglichkeit, mit den Daten und Visualisierungen zu interagieren und diese so an seine individuellen Bedürfnisse anzupassen.

Mit dem Prototyp wurde bereits ein erster Benutzertest durchgeführt, um herauszufinden, ob die realisierten Visualisierungen für die Zielgruppe relevant und geeignet sind, aber auch, ob die Interaktionsmöglichkeiten sinnvoll sind. Die Ergebnisse dieses Tests und der anschließenden Befragung werden entsprechend ausgewertet. Dadurch werden wichtige Informationen für die Verbesserung und Weiterentwicklung des Prototyps gesammelt.

1.2 Aufbau der Arbeit

Die Arbeit beginnt mit einer Einleitung, in der die Struktur der Arbeit, die Lösungsansätze und die Methodiken beschrieben werden. Im zweiten Kapitel werden verwandte Arbeiten vorgestellt, wobei eine Literaturrecherche und eine Marktanalyse durchgeführt wurden. Die Marktanalyse umfasst drei spezifische und bestehende Visual Analytic Tools: Blitz, Mobalytics und Porofessor.

Das dritte Kapitel beschreibt die Funktionen des Visual Analytic Tools, das als Dashboard mit mehreren Unterseiten und verschiedenen Themenschwerpunkten konzipiert ist. Es wird ein Datenanalyse-Workflow vorgestellt, der die Aufbereitung von Spieler- und Matchdaten umfasst. Des Weiteren werden die einzelnen Visualisierungskomponenten des Tools erläutert.

Das vierte Kapitel beschreibt die Implementierung des ersten Prototyps. Hier werden die verwendeten Tools und Bibliotheken sowie die Architektur und konkrete Umsetzung der einzelnen Komponenten vorgestellt. Dabei wird besonders auf die Besonderheiten der einzelnen Komponenten eingegangen, wie zum Beispiel die Interaktionsmöglichkeiten oder die Schwierigkeiten mit den vorliegenden Daten.

Das fünfte Kapitel befasst sich mit der Evaluation des Prototyps. Hierbei wurden sowohl Benutzertests mit anschließenden Befragungen als auch Laufzeitevaluationen durchgeführt.

Abschließend wird im sechsten und letzten Kapitel ein Fazit gezogen, das die Ergebnisse der Arbeit zusammenfasst und einen Ausblick auf zukünftige Entwicklungen gibt.

2 VERWANDTE ARBEITEN

Das wohl bekannteste Videospiel im Bereich der Multiplayer Online Battle Arena (MOBA) ist das Spiel League of Legends (LoL). Dieses Spiel wird immer wieder in wissenschaftlichen Arbeiten thematisiert. Das folgende Kapitel ordnet das Thema in den aktuellen Forschungsstand ein und gibt einen Überblick über die wichtigsten Ergebnisse und Erkenntnisse der relevanten Arbeiten. Darüber hinaus gibt es eine Vielzahl von Analysetools auf dem Markt, die es dem Anwender erleichtern sollen, eigene Schwachstellen zu erkennen, zu analysieren und anschließend zu beheben. Es wird näher erläutert, auf welche Themenfelder sich die bekanntesten Anwendungen konzentrieren und welche möglichen Erweiterungen denkbar und wünschenswert sind.

2.1 Literaturrecherche

Zahlreiche wissenschaftliche Publikationen zum Thema League of Legends beschäftigen sich mit der Frage, wovon ein Sieg oder eine Niederlage im Spiel abhängt (beispielsweise „Using Machine Learning to Predict Game Outcomes Based on Player-Champion Experience in League of Legends“ [6]) oder wie die Spieler auf bestimmte Situationen reagieren und agieren (beispielsweise „Playing with Streakiness in Online Games: How Players Perceive and React to Winning and Losing Streaks in League of Legends“ [12]).

Da von Riot, den Entwicklern des Spiels, nur eine begrenzte Menge an Daten zur Verfügung gestellt wird, beschäftigen sich einige Dokumente mit einer alternativen Extraktion von Daten aus dem Spiel. Im Paper „Implementation of user playstyle coaching using video processing and statistical methods in league of legends“ [11] werden die Daten aus Videos in Form von Bildschirmaufnahmen des Spiels extrahiert und mit verschiedenen Machine Learning Algorithmen analysiert. Für die Visualisierung dieser Daten wurden bestehende Coaching Tools analysiert und eine Applikation mit verschiedenen kleineren Dashboards implementiert. Im Gegensatz dazu wird in der Publikation „Smart kills and worthless deaths: eSports analytics for League of Legends“ [14] beschrieben, wie die Daten durch das direkte Erfassen und Aufzeichnen der Positionierungen und Aktionen der Spieler in einem Spiel gesammelt werden. Ziel ist es, auf Basis dieser Statistiken ein Framework zur Spielerbewertung zu erstellen. Eine Visualisierung der gesammelten Daten ist dabei nicht vorgesehen.

Ein weiterer häufig genannter Aspekt ist die Datenverarbeitung in Echtzeit. Die Vor- und Nachteile von physischen und digitalen Implementierungen werden im Artikel „Real-Time Dashboards to Support eSports Spectating“ [5] diskutiert. Der Fokus liegt dabei auf der Berichterstattung und Live-Übertragung von eSport-Events, da ein zunehmendes Interesse an datengesteuertem Feedback bei den Zuschauern besteht. Dabei geht es vor allem um die Frage, welche Informationsdichte ein solches Dashboard haben darf und wie man der Vielzahl unterschiedlicher Nutzerbedürfnisse gerecht werden kann. Mit einer ähnlichen Fragestellung beschäftigt sich die Publikation „Live Feedback for Training Through Real-Time Data Visualizations: A Case Study with League of Legends“ [19]. Ziel ist es, den Zuschauern ein effektives Verfolgen des Spielgeschehens zu ermöglichen und die Komplexität des Spiels adäquat aufzubereiten. Darüber hinaus werden Designziele für Zuschauer-Dashboards definiert, um eine transparente Gestaltung zu gewährleisten. Beide Artikel basieren auf Nutzerbefragungen und der Einschätzung von eSports-Profis.

Im Gegensatz dazu beschreibt die Publikation „Performance analysis in esports: part 1 - the validity and reliability of match statistics and notational analysis in League of Legends“ [16], wie öffentlich

zugängliche Spielstatistiken verbessert werden können. Das Ergebnis der Studie ist, dass Spielverläufe und Notizen von erfahrenen Spielern austauschbar für alle beobachteten Maße verwendet werden können, um die Leistungsanalyse in professionellen League of Legends zu erleichtern.

Schließlich ist der Artikel „VisuaLeague: Player performance analysis using spatial-temporal data“ [2] von großer Bedeutung für diese Arbeit. Der Fokus liegt auf raum-zeitlichen Informationsdaten und wie diese mit geeigneten Visualisierungstechniken aufbereitet werden können. Auf animierten Karten werden die Ereignisse während eines LoL Matches visualisiert, mit dem Ziel, die Spielerleistung zu optimieren. Dazu wurde ein Prototyp entwickelt und in einer ersten Nutzerstudie die Bedeutung der raum-zeitlichen Daten für die Entwicklung der eigenen Fähigkeiten erfasst.

Zusammenfassend lässt sich also sagen, dass sich bereits eine Vielzahl von wissenschaftlichen Arbeiten mit der Verarbeitung und Aufbereitung von Daten aus einem LoL-Match beschäftigt hat. Dabei stand jedoch häufig nicht die Verbesserung der Leistung des einzelnen Spielers im Vordergrund. Wie verschiedene Visualisierungsmethoden auf den Nutzer wirken und welche Faktoren dabei von Bedeutung sind, wurde häufig vernachlässigt. Zudem wurde selten versucht, den größtmöglichen Nutzen aus den von Riot zur Verfügung gestellten Daten zu ziehen.

2.2 Marktanalyse

Bevor konkrete Funktionalitäten des Analyse-Tools ausgearbeitet wurden, wurde eine Marktanalyse von bestehenden Tools durchgeführt. Diese Analyse diente dazu, herauszufinden, welche Analyse-Funktionen Anfängern bereits zur Verfügung gestellt werden. Hierbei wurde notiert, welche dieser Funktionen noch ausbaufähig und welche besonders hilfreich waren, um eine Orientierung für die eigene Anwendung zu erhalten. Für diese Marktanalyse wurden drei beliebte Analysetools (Blitz, Mobalytics und Porofessor) berücksichtigt.

2.2.1 Blitz

Blitz [3] ist hauptsächlich ein Tool, dass während dem Spiel eingesetzt werden kann, um Informationen zu den aktuellen Mitspielern und Gegnern zu erhalten. Darüber hinaus bietet Blitz jedoch ebenfalls Post-Match Analyse an. Hierbei werden Statistiken aus drei verschiedenen Kategorien angezeigt und mit anderen Spielern aus allen Rängen verglichen. Die drei Kategorien sind Combat, Income und Vision, und es werden einem für jede Hinweise zu guten und schlechten Eigenschaften des eigenen Gameplays gegeben. Die Hinweise sind oft auf die gespielte Rolle angepasst und werden mit dem direkten Gegner auf der Rolle verglichen. Manche Hinweise beziehen sich auch auf einzelne Events aus den letzten Spielen, zum Beispiel, dass man früh in einem Teamfight gestorben oder mit über 4000 Gold gestorben ist. Insgesamt sind die Hinweise von Blitz sinnvoll und hilfreich, da sie sich auch auf die konkreten Ereignisse beziehen. Gut ist auch der Vergleich mit der gesamten Playerbase, da dieses einen seine eigenen Fähigkeiten besser einordnen lässt. Die Post-Match Analyse von Blitz ist jedoch nicht besonders umfangreich, da sie sich auf die wenigen genannten Bereiche beschränkt und einem somit kein umfänglicheres oder detaillierteres Bild von den eigenen Fähigkeiten und Statistiken liefert.

2.2.2 Mobalytics

Mobalytics [15] ist im Gegensatz zu Blitz ein reines Post-Match Analysetool und bietet dementsprechend auch detailliertere Informationen an. Mobalytics hat zwei Hauptbereiche zur Analyse der eigenen Spiele, diese sind die Profilanalyse und der Gamer Performance Index. Die Profilanalyse zeigt einem an, mit welchen Champions man am besten abschneidet und gegen welche Champions man die größten Schwierigkeiten hat. Die eigenen Champions werden als Champion Pool dargestellt, bei dem die meistgespielten Champions

und die jeweilige Win Rate und das individuelle Matchup gegen andere Champions angezeigt werden. Des Weiteren sieht man simple Statistiken wie KDA und Vision Score für jeden gespielten Champion. Neben dem Champion Pool gibt es den Matchup Pool, der einem die Champions zeigt, gegen die man am häufigsten spielt und die jeweilige Win Rate und Statistiken, die man gegen diesen Champion im Schnitt erzielt.

Der Gamer Performance Index besteht aus Gameplay Bewertungen aus diversen Kategorien, welche einem die Stärken und Schwächen des eigenen Gameplays aufzeigen sollen. Die Kategorien sind Fighting, Farming, Vision, Aggression, Survivability, Objectives, Consistency und Versatility und sind jeweils sehr detailliert aufgebaut. Für jede Kategorie erhält man eine Bewertung, die sich aus mehreren kleineren Bewertungen, welche für diese Kategorie relevant sind, zusammengesetzt. Jeder Wert aus diesen Kategorien kann wiederum mit den Durchschnittswerten von Spielern aus den verschiedenen League of Legends Rängen verglichen werden. Dieses gibt einem einen guten Überblick der eigenen Fähigkeiten, jedoch ist oft die Berechnung der Werte etwas intransparent.

2.2.3 Porofessor

Porofessor [17] ist ein weiteres In-game Analyse Tool, ist jedoch mit einer weiteren Seite namens LeagueOfGraphs [13] verlinkt, um auch Post-Match Analyse anzuzeigen. LeagueOfGraphs zeigt einem mehrere Statistiken aus der gesamten Match History. Insbesondere liegt bei LeagueOfGraphs der Fokus auf die eigene Performance im Verhältnis zur durchgehenden Spieldauer, also eine Aneinanderreihung von mehreren Spielen. Es wird die durchschnittliche Winrate des ersten und letzten Spiels der Session angegeben, sowie die Winrate nach einem gewonnen oder einem verlorenen Spiel. Zu den gespielten Champions werden ebenfalls simple Statistiken wie Winrate, KDA oder Gold angezeigt, die auch auf diversen anderen Tools vorzufinden sind. Zusammenfassend sind die angebotenen Informationen zwar sinnvoll und hilfreich, aber bis auf wenige Ausnahmen nicht besonders einzigartig.

2.3 Rückschlüsse

Die in Unterabschnitt 2.1 erzielten Erkenntnisse aus der Literaturrecherche decken sich größtenteils, jedoch nicht vollständig mit den Ergebnissen aus Unterabschnitt 2.2. Die Echtzeitverarbeitung von Match-Daten ist tatsächlich ein beliebtes Mittel von bestehenden Analysetools, jedoch nicht, um Zuschauern ein besseres Spielverständnis zu bieten, sondern um Spielern während einem Spiel Hinweise und Statistiken zu liefern. Der geringe Fokus auf Visualisierungsmethoden deckt sich auch mit den Analysetools, welche einen größeren Schwerpunkt auf das Anzeigen von relevanten Statistiken setzen, ohne diese besonders zu visualisieren. Die aus der Riot API vorhandenen Daten werden ähnlich zur Literatur auch bei den Analysetools nicht vollumfänglich verwendet. Manche Tools sind auch nicht immer transparent, was die Zusammenstellung von angezeigten Werten angeht. Die Spielernote wird beispielsweise oft von einem Tool angezeigt, obwohl diese nicht von der Riot API ausgegeben wird. Im Gegensatz zur Literatur ist jedoch bei den vorhandenen Analysetools der Fokus deutlich auf die Verbesserung des einzelnen Spielers gesetzt.

3 FUNKTIONEN DES VISUAL ANALYTIC TOOLS

Für diese Anwendung haben wir uns für eine Visualisierung in Form einer Dashboard Webanwendung entschieden. Konkret besteht diese Anwendung aus 5 Dashboards. Diese Dashboards werden weiter unten genauer beschrieben. Wir haben uns aus folgenden Gründen für die Visualisierung in Form eines Web Dashboards entschieden.

- Dashboards sind ein geeignetes Mittel, um dem Nutzer alle relevanten Informationen zum Spiel vollständig, kompakt und übersichtlich darzustellen. Damit soll dem Nutzer die

Möglichkeit gegeben werden, die für ihn relevanten Informationen effizient abzurufen.

- Dashboards lassen sich konfigurierbar entwickeln, sodass der Benutzer die Filterung bestimmter Informationen oder die Hervorhebung bestimmter Informationen vornehmen kann und dadurch das Dashboard nach seinen individuellen Bedürfnissen anpassen kann.
- Dashboards ermöglichen eine einfache und verständliche Darstellung von Daten. Dabei werden die für die Daten am besten geeigneten Visualisierungsmethoden verwendet.
- Die Bereitstellung von Anwendungen im Web hat den Vorteil, dass der Nutzer möglichst niederschwellig mit der Anwendung in Kontakt kommt. Auch Updates und Erweiterungen der Anwendung werden automatisch allen Nutzern zur Verfügung gestellt.

3.1 Anwendung mit Dashboard-Charakter

Während der Spiele von League of Legends wird eine große Menge an unterschiedlichen Informationen gesammelt und über die Riot API zur Verfügung gestellt. Diese große Menge an Informationen muss in der Webanwendung strukturiert und übersichtlich dargestellt werden. Es bietet sich daher an, diese große Menge an Informationen übersichtlich in verschiedenen kleineren Dashboards mit unterschiedlichen thematischen Schwerpunkten darzustellen. Dashboards sind ein leistungsfähiges Gestaltungskonzept für grafische Benutzeroberfläche, da viele wichtige Informationen auf einen Blick erfasst werden können. Ähnlich dem Armaturenbrett im Auto werden mehrere Informationen aus verschiedenen Quellen übersichtlich im Blickfeld des Fahrers visualisiert. Daher auch die Wortherkunft „Dashboard“ vom englischen „instrument panel“ (Armaturenbrett). Wichtig ist dabei, dass das Dashboard nicht zu komplex wird und die einzelnen Darstellungen abstrahiert und auf das Wesentliche reduziert werden. Durch die Aufteilung der Anwendung in mehrere Unterseiten mit unterschiedlichen thematischen Schwerpunkten kann die große Menge an Informationen strukturiert und die Übersichtlichkeit der einzelnen Unterseiten im „Dashboard-Charakter“ gewährleistet werden [8]. Darüber hinaus spielt die Möglichkeit der Interaktion und Personalisierung auf den einzelnen Unterseiten eine große Rolle.

Von besonderer Bedeutung im Spiel ist die Positionierung auf dem Spielfeld (Karte) zu verschiedenen Zeitpunkten im Spiel. Aus diesem Grund soll eine Karten-Komponente dem Benutzer die Bedeutung der Positionierung näher erläutern. Wie in Abbildung 1 zu sehen ist, werden dem Benutzer durch verschiedene Punkte auf

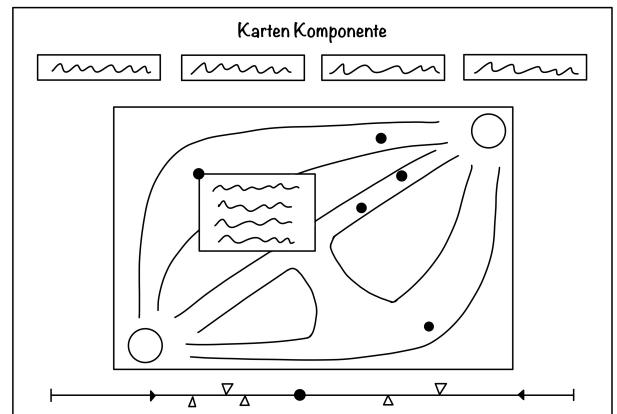


Abbildung 1: Wireframe der Karten-Komponente

der Karte Ereignisse im Spiel angezeigt. Zusätzlich kann über eine Zeitleiste ein bestimmter Zeitpunkt oder eine Zeitspanne für die anzuzeigenden Ereignisse ausgewählt werden.

Eine Statistik-Komponente soll die Entwicklung einzelner Werte in den Spielen und deren Verhältnisse zueinander näher erläutern. Dabei soll der Benutzer selbst bestimmen können, welche Werte visualisiert werden sollen, wie im Wireframe in Abbildung 2 mit einzelnen Buttons angedeutet.

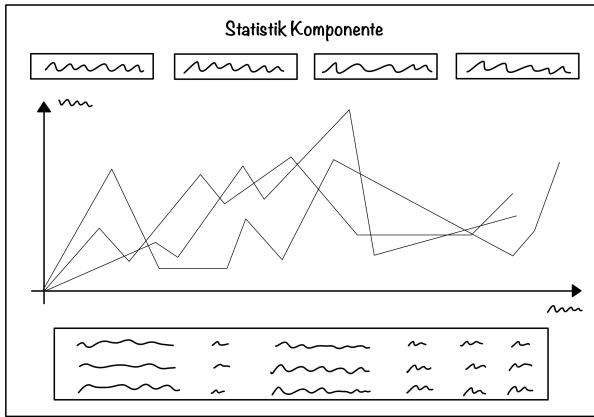


Abbildung 2: Wireframe der Statistik-Komponente

Das Spiel kann auch in verschiedene Rollen, oder auch Lanes genannt, aufgeteilt werden. Dabei werden jeder Lane unterschiedliche Aufgaben zugewiesen und unterschiedliche Werte sind von Bedeutung. Von besonderer Bedeutung ist in jeder Lane die eigene Leistung im Vergleich zum direkten Gegner auf derselben Lane. Wie in der digitalen Skizze in Abbildung 3 zu sehen ist, sollen mehrere Spiele und deren Werte in einer Tabelle aufgelistet werden, um einen Vergleich der Leistung über mehrere Spiele hinweg zu ermöglichen. Durch einfache Pfeile neben den Zahlen, soll hervorgehoben werden, ob ein Wert gut (grüner Pfeil hoch) oder schlecht (roter Pfeil runter) ist.

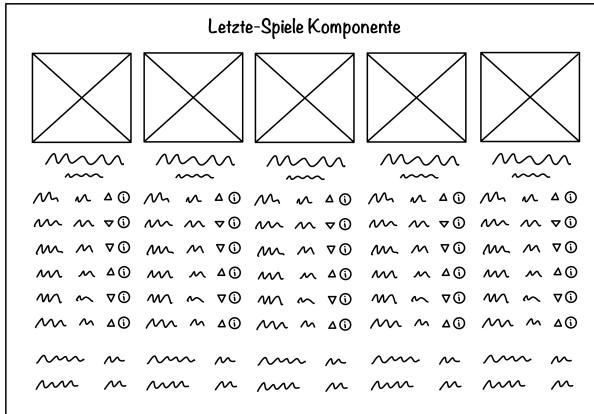


Abbildung 3: Wireframe der Letzte-Spiele-Komponente

Auch Spiele werden häufig in drei Zeitabschnitte unterteilt, die als Early Game, Mid Game und Late Game bezeichnet werden. Da in diesen Zeitabschnitten in der Regel unterschiedliche Taktiken verfolgt werden, soll in einer Zeitphasen-Komponente die persönliche Leistung in diesen Zeitabschnitten verglichen werden. Eine dreispaltige Gegenüberstellung der einzelnen Werte (siehe Abbildung 4) soll den Vergleich erleichtern.

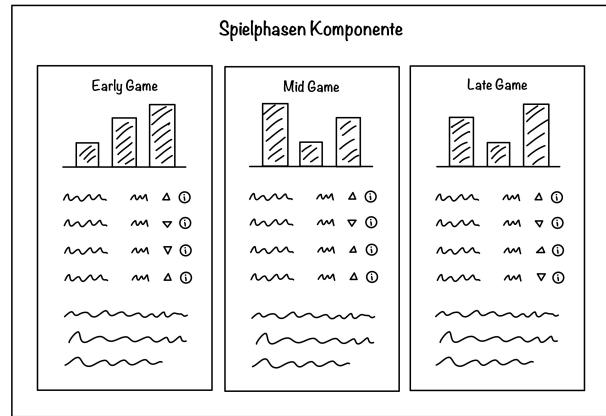


Abbildung 4: Wireframe der Spielphasen-Komponente

Zuletzt sollen dem Benutzer verschiedene Tipps für die nächsten Spiele angezeigt werden. Eine kurze Erläuterung soll dabei helfen, die eigenen Schwächen besser zu verstehen, wie im Wireframe in Abbildung 5 dargestellt.

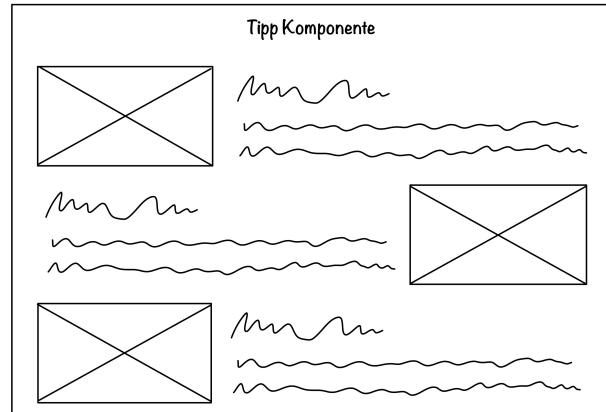


Abbildung 5: Wireframe der Tipp-Komponente

3.2 Datenanalyse-Workflow

Die Daten für die Anwendung werden alle von der Riot API zur Verfügung gestellt. Diese werden im JSON-Format über die Riot REST API bereitgestellt. Diese wird klassisch über eine HTTP-Anfrage angesprochen. In der Applikation werden vier geschlossene Endpunkte der API angesprochen. Drei der vier Endpunkte gehören zum MATCH-V5 Endpunkt.

Der einzelne Endpunkt gehört zum SUMMONER-V4 Endpunkt. Dieser Endpunkt „/lol/summoner/v4/summoners/by-name/summonerName“ wird initial bei der Nutzung der Applikation verwendet. Hier gibt der Benutzer seinen „summonerName“ an, mit dem wir die „puuid“ des Spielers abfragen. Die „puuid“ ist die eindeutige ID, die jedem Account von Riot zugewiesen wird.

Der erste Endpunkt von MATCH-V5 ist „/lol/match/v5/matches/by-puuid/puuid/ids“. Dieser Endpunkt wird verwendet, um die Spiele des Benutzers über die API abzufragen.

Der zweite Endpunkt ist „/lol/match/v5/matches/matchId“. Diesen Endpunkt verwenden wir, um über den Request mit weitergehend definierten Parametern Spiele mit der zugehörigen „matchid“ abzurufen. Weitere Parameter sind die Anzahl der letzten Spiele (max. 100) und ob nur Ranglistenspiele abgefragt werden sollen.

Hier werden die Enddaten jedes Spielers übergeben, der an dem Spiel teilgenommen hat.

Der dritte Endpunkt ist „/lol/match/v5/matches/matchId/timeline“. Dieser Endpunkt wird verwendet, um über das Ereignis „matchId“ Informationen aus dem entsprechenden Spiel zu erhalten. Auch hier können die weiteren Parameter des zuvor beschriebenen Endpunktes angegeben werden, sowie auch die Startzeit.

Die Arbeit mit den Daten der Riot API hat einen klaren Vorteil, aber auch einen Nachteil. Der Vorteil ist, dass man für eine Anwendung wie die unsere nicht selbst etwas entwickeln muss, um Informationen aus dem Spiel zu erhalten, die man dann weiterverarbeiten kann. Das erleichtert die Arbeit und man kann sich auf das Wesentliche konzentrieren. Es hat aber auch den Nachteil, dass man von Dritten abhängig ist und auf die Informationen beschränkt ist, die der Betreiber der API, also Riot, zur Verfügung stellt. Diese Erfahrung haben wir auch bei der Entwicklung unserer Anwendung gemacht. Zum Beispiel gibt Riot jedem Spieler am Ende des Spiels eine Bewertung über seine Leistung. Diese Information wird nicht über die API weitergegeben und es wird auch nicht kommuniziert, wie diese Bewertung berechnet wird. Ebenso kann es, wie in unserem Fall, vorkommen, dass wir das „Wards Placed“ Event abfragen wollten, aber ein Fehler in der Riot API dazu geführt hat, dass im Patch 13.3 [7] kein „Wards Placed“ Event ausgelöst wird. Daher konnten wir diesen Wert in der uns zur Verfügung stehenden Entwicklungszeit nicht berücksichtigen.

3.2.1 Aufbereitung Spieler-Daten

Aus den Spieldaten extrahieren wir verschiedene Spieldaten zu den einzelnen Spielern. Für unsere Anwendung verwenden wir aus den Spieldaten Informationen über den gespielten Champion in Form der gespielten Rolle, des Namens des Champions und des Levels des Champions. Ebenso verwenden wir Informationen über die Anzahl der Kills, Deaths und Assists, sowie getötete Diener und neutrale Monster im Jungle. Weitere Informationen sind der verursachte und erhaltene Schaden, sowie die verursachte Heilung und der Vision Score. Wir erhalten auch Informationen darüber, ob das Spiel von einem der Teams aufgegeben wurde und ob der Benutzer im siegreichen Team ist.

Die KDA des Spielers wird aus den Informationen über Kills, Deaths und Assists berechnet. Der Gesamtfarm wird aus den getöteten Dienern und neutralen Monstern berechnet. Für jeden der ermittelten Werte wird zusätzlich ein Durchschnittswert über die letzten 5 Spiele der jeweiligen Rolle im Spiel gebildet.

3.2.2 Aufbereitung Match-Daten

Match-Daten sind, wie bereits erwähnt, Informationen über Ereignisse, die während eines Spiels ausgelöst werden. Events sind zu wichtige Ereignisse, die während des Spiels stattfinden, sowie auch jede volle Minute wird automatisch ein Event für den Datensatz ausgelöst. Für jedes Ereignis pro Minute werden für jeden Spieler die gleichen Informationen bereitgestellt. Wir greifen hier auf deren Level, Gesamt Gold, getötete Diener und Jungle Diener sowie Gesamt Erfahrungspunkte zu.

Ebenso greifen wir auf Battle Stats zu, welche die verschiedenen für den Kampf relevanten Statistiken enthalten. Diese beinhalten unter anderem das Leben, die Resistenzen und die Schadenswerte des Champions. Sowie die bis zu diesem Zeitpunkt erlittenen Schadenswerte, aufgeteilt in magischen und normalen Schaden, aber auch addiert.

Events, die ansonsten ausgelöst werden können, sind Champion Kills, Gebäude Zerstörungen sowie das Besiegen der verschiedenen epischen Monster in Form von Dragon, Rift Herald und Baron Nashor. Hierbei erhalten wir den Zeitpunkt des Ereignisses, Informationen über die Position des getöteten Champions, Gebäudes oder Monsters sowie Informationen darüber, welche Einheiten mit einem

Assist an dem jeweiligen Ereignis beteiligt waren. Für Gebäude-Kills und Rift Heralds werden keine Assists übergeben.

3.3 Aufbau der einzelnen Visualisierungs-Komponenten

Das League of Legends Analysis Tool ist im Stile eines Dashboards aufgebaut. Es besteht aus verschiedenen Unterseiten, welche jeweils bestimmte Aspekte des zu analysierenden Spielerprofils behandeln. Die Navigation zu den einzelnen Unterseiten gelingt über eine auf der linken Seite gelegene Auswahlleiste.

3.3.1 Startseite

Beim erstmaligen Öffnen der Anwendung landet man auf der Login Seite. Diese besteht aus einer Überschrift, welche den Namen der Anwendung darstellt und einem Formular. Im Formular können die Region und der Nutzernname des League of Legends Kontos des Benutzers eingegeben werden. Die Region wird mithilfe eines Dropdown Menüs bestimmt, der Nutzernname wird in ein Textfeld eingegeben. Des Weiteren gibt es eine Checkbox zum Bestimmen, ob nur Ranglisten Spiele oder alle Spiele aus der Match History genommen werden sollen. Die zu analysierenden Spiele müssen jedoch auf der Summoner's Rift Map und im fünf gegen fünf Modus gespielt sein.

3.3.2 Karten-Komponente

Die Karten-Seite besteht aus einem interaktiven Bild der Spielkarte Summoner's Rift, welche mehrere farbige Markierungen enthält und verschiedenen Eingabeelementen. Die Eingabeelemente sind ein Multiselect, ein Slider und in der linken Sidebar befinden sich mehrere Checkboxen. Multiselect, Slider und Checkboxen bestimmen zusammen, welche Punkte auf der Karte angezeigt werden.

Das Multiselect beinhaltet eine Auswahl zwischen Kills, Deaths, Assists, Dragons, Heralds und Barons, welche jeweils eine Art von Event darstellen und eine Farbe von Punkten repräsentieren. Die Checkboxen repräsentieren jeweils eines von 5 Spielen und zeigen durch ihre Form an, zu welchem Spiel die Events gehören. Der Slider grenzt eine Zeitregion zwischen 0 und 60 ein, welches bestimmt, aus welcher Zeitphase des Spiels Events angezeigt werden sollen.

Die Events auf der Karte sind Knöpfe, die beim Drücken ein Pop-up-Fenster öffnen und schließen. Das Pop-up-Fenster beinhaltet Informationen zum Event wie die Art von Event, Beteiligte Champions und Zeitpunkt im Spiel des Events. Die Karte selber kann durch Scrollen mit der Maus rein- und rausgezoomt werden und durch Klicken und Ziehen auf der Karte kann das Bild in beliebige Richtungen geschwenkt werden.

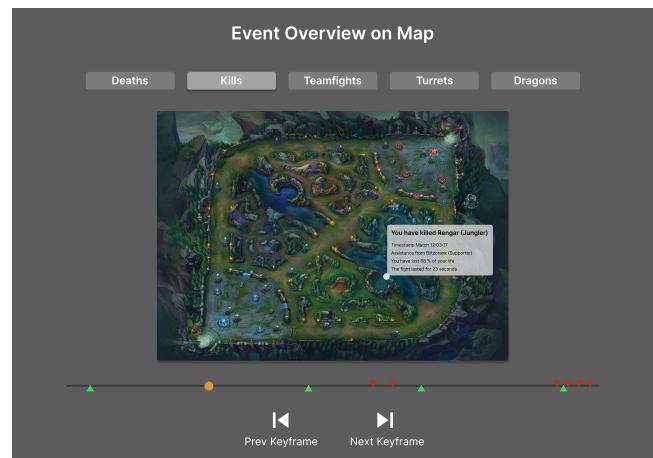


Abbildung 6: Mockup der Karten-Komponente

3.3.3 Statistik-Komponente

Die Statistik Unterseite besteht aus einem zweiseitigen Layout. Die erste Seite enthält ein Liniendiagramm, die zweite ein interaktives Balkendiagramm. Unter den Diagrammen befindet sich ein Slider und über ihnen befinden sich jeweils drei Radiobuttons. Der Slider bestimmt auf beiden Komponenten die Anzahl der Spiele, welche auf dem jeweiligen Graph angezeigt werden sollen. Die Radiobuttons bestimmen jeweils die Art von Daten, die auf den Diagrammen angezeigt werden.

Das Liniendiagramm zeigt den Verlauf des gesammelten Goldes, getöteten Vasallen oder zerstörten Augen verglichen mit den jeweiligen Zeitpunkten im Spiel an. Jede Linie repräsentiert ein Spiel und unterscheidet sich farblich und durch Strichverlauf von den anderen Linien. Rechts vom Diagramm ist eine Legende, die jeder Linie einem Spiel und dem gespielten Champion zuordnet.

Das Balkendiagramm vergleicht pro Spiel die absoluten Werte aus den Kategorien Gold, Damage und Vision. Die Kategorien sind auch durch Farbe der Balken voneinander getrennt. Durch Klicken auf eines der farbigen Balken wird der Wert in die einzelnen Bestandteile, die den Wert ausmachen, aufgeteilt und erzeugt somit ein neues Balkendiagramm. Beim Klicken auf den Hintergrund gelangt man wieder auf die vorherige Ansicht.



Abbildung 7: Mockup der Statistik-Komponente

3.3.4 Letzte-Spiele-Komponente

Die Letzte-Spiele Unterseite setzt einen Fokus auf die verschiedenen Rollen im Spiel, die ein Spieler auswählen kann. Sie ist in einem fünf Spalten Layout ausgelegt, wobei jede Spalte ein Spiel auf einer bestimmten Rolle repräsentiert.

Jede Spalte zeigt an, welcher Champion im Spiel verwendet wurde und wann das Spiel gespielt wurde. Diverse Statistiken des Spielers, die am Ende des Spiels generiert wurden, werden angezeigt und mit dem Gegner oder mit dem eigenen Durchschnitt verglichen. Durch einen farbigen Pfeil wird angezeigt, ob man besser oder schlechter als sein Gegenüber performt hat. Jede Statistik hat einen Informationspunkt, welcher durch Hovern detailliertere Informationen zu dieser Statistik anzeigt. Zum Schluss wird das Spielergebnis angezeigt und die Information, ob das Spielergebnis durch eine Aufgabe bestimmt wurde. Abhängig davon, welche Rolle ausgewählt wurde, werden einem nur Spiele zu dieser Rolle angezeigt und relevanten Statistiken fett hervorgehoben.

Zwei Dropdown Menüs befinden sich oberhalb der fünf Spalten neben der Seitenüberschrift. Die erste bestimmt die zu betrachtenden Rolle, die andere bestimmt, ob mit dem Gegner oder mit dem eigenen Durchschnitt verglichen werden soll.

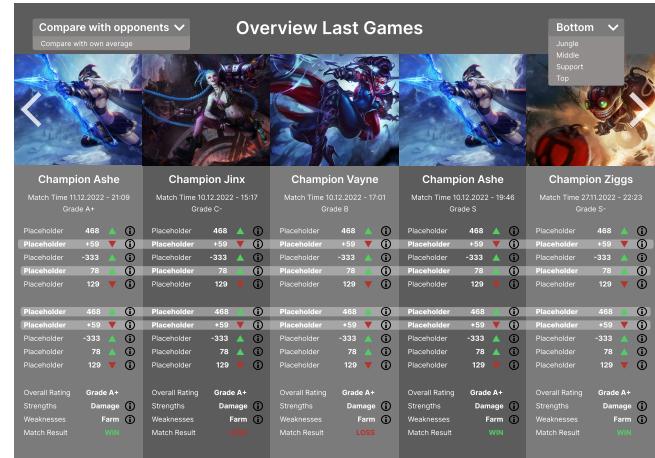


Abbildung 8: Mockup der Letzte-Spiele-Komponente

3.3.5 Spielphasen-Komponente

Die Spielphasen Unterseite hat ein drei Spalten Layout, mit einer Spalte für jede Match Phase. Pro Spalte wird der zeitliche Rahmen der Phase angegeben. Statistiken wie das Level des Spielers und das eingesammelte Gold werden für jede Phase präsentiert, sowie die während dieser Phase erworbenen Items. Vollumfängliche Statistiken zum Charakter und dem verursachten Schaden befinden sich ebenfalls in jeder Spalte.

Oberhalb von den Spalten befindet sich eine Auswahl, welche bestimmt, welches Spiel aus den letzten fünf betrachtet werden soll in Form eines Dropdown Menüs. Ein Bild des in dem Spiel verwendeten Champions sowie die zeitliche Information, wann das Spiel stattgefunden hat, werden ebenfalls angezeigt.



Abbildung 9: Mockup der Spielphasen-Komponente

3.3.6 Tipp-Komponente

Die Tipp Seite besteht aus bis zu fünf verschiedenen Hinweisen, auf die man bei seinen nächsten Spielen achten sollte. Die Hinweise sind durch Trennlinien voneinander getrennt und bestehen jeweils aus einer Überschrift, einem passenden Bild und einem Text, welcher den Hinweis etwas detaillierter beschreibt und erklärt, warum der Hinweis für einen wichtig ist. Die Seite, auf der sich Bilder und der Text der Hinweise befinden, werden immer von einem Tipp zum nächsten alterniert. Konnten keine relevanten Hinweise zu den letzten Spielen eines Accounts bestimmt werden, wird dies dem Nutzer über eine Hinweissnachricht mitgeteilt.

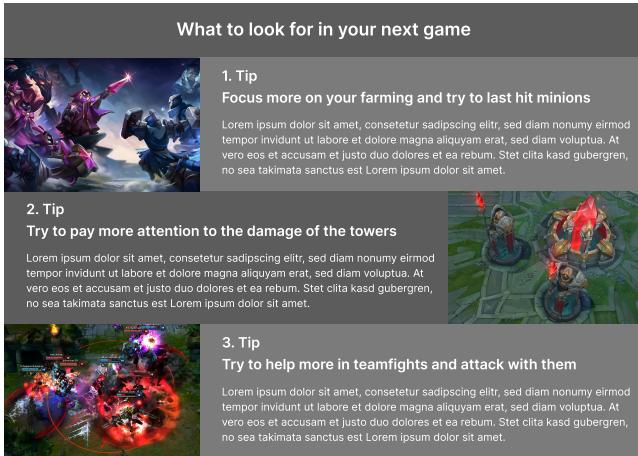


Abbildung 10: Mockup der Tipp-Komponente

4 IMPLEMENTIERUNG

Für unsere prototypische Anwendung haben wir ein webbasiertes, interaktiv nutzbares Tool geplant, das mit Live-Daten aus der Riot API versorgt wird. Da die Riot API eine große Menge an Daten zur Verfügung stellt, die wir in dieser Anwendung nutzen und darstellen wollen, haben wir uns entschieden, mehrere Dashboards zu erstellen, die sich jeweils auf eine Darstellungsform und einen Teil der zur Verfügung gestellten API-Daten beschränken.

Wir haben uns für fünf Komponenten entschieden, die wir umsetzen. Die erste Komponente befasst sich mit den Events, die während des Spiels stattfinden. Hier betrachten wir die Raum-Zeit-Koordinaten der verschiedenen Events, die während des Spiels stattfinden. Die Events, die wir betrachten, sind Champion Kills, Gebäude Zerstörungen und das Töten von epischen Monstern. Diese Raum Zeit Koordinaten werden auf der Spielkarte dargestellt und mit zusätzlichen Informationen visualisiert.

Die zweite Komponente ist die Statistik-Komponente. Hier werden ausgewählte Metriken in Graphen dargestellt. Wir arbeiten mit zwei verschiedenen Arten von Graphen. Zum einen haben wir ein Liniendiagramm mit mehreren Kategorien, wobei die Kategorien verschiedene Spiele sind. Hier sehen wir uns die Goldhistorie, besiegte Minions und den Vision Score an. Zweitens haben wir ein Balkendiagramm. Hier wird der Gold-, Vision- und Damage-Score angezeigt. Mit diesem Balkendiagramm kann der Benutzer interagieren und so die einzelnen Scores detaillierter aufschlüsseln.

Die dritte Komponente befasst sich mit der Endbewertung der einzelnen Spieler. Hier betrachten wir verschiedene ausgewählte Werte, die von der API zur Verfügung gestellt werden, und bieten dem Nutzer die Möglichkeit, diese Werte entweder mit seinen eigenen Durchschnittswerten oder mit den Werten des Gegners auf der gleichen Rolle im Spiel zu vergleichen.

In der vierten Komponente betrachten wir die relevanten Spiel- und Kampfwerte des Nutzers in den drei Spielphasen. Diese werden in Early, Mid und Late Game unterteilt. Hierbei soll der Nutzer zwischen verschiedenen Spielen auswählen können, die angezeigt werden sollen.

In der fünften und somit letzten Komponente geben wir dem Nutzer Tipps, was er für die nächsten Spiele beachten sollte, um sich zu verbessern. Hierfür werden die letzten fünf Spiele auf bestimmte Metriken untersucht und dann die häufigsten Tipps für diese Spiele werden dann ausgegeben.

4.1 Verwendete Tools und Bibliotheken

Für diese Anwendung haben wir uns für Python als Hauptsprache entschieden. Python bietet sich hier an, da wir in dieser Anwendung

hauptsächlich Daten auswerten und manipulieren. Um die Anwendung möglichst einfach an den Nutzer bringen zu können, wurde das Framework Streamlit gewählt, mit dem eine auf Datenanalyse fokussierte Python-Anwendung web fähig gemacht werden kann.

Um die Karten- und die Statistik-Komponente zu erstellen, musste die Streamlit-Anwendung um je eine React-Komponente erweitert werden. Diese werden von Streamlit nativ unterstützt. Bei der Statistik-Komponente wird das Balkendiagramm in der Restkomponente ausgegeben, da das hier beschriebene Balkendiagramm von dem d3.js Framework bereitgestellt wird. Das Liniendiagramm hingegen wird mit Pandas und Altair erzeugt.

4.2 Architektur & Implementierung der Komponenten

Jede Unterseite der Anwendung hat die von Streamlit zur Verfügung gestellten Features und die von der Riot API erhaltenen Daten auf unterschiedliche Art und Weise umgesetzt. Um das Layout gemäß den zuvor geplanten Skizzen und Diagrammen zu realisieren wurde entschieden, die Anwendung in der breiten Streamlit Ansicht zu entwickeln. Diese ist zwar nicht für mobile Ansichten optimiert, jedoch war es nicht das Ziel der Anwendung eine mobile Applikation zu erstellen, sondern eine Web-Anwendung, welche auf Desktop Browsern angezeigt wird.

4.2.1 Startseite

Die Startseite besteht aus einer Streamlit "form" Komponente, welche beim Betätigen des Submit Buttons eine Request an die Riot API schickt, um aus Username und Region die für den Account entsprechende Player Unique Id (puuid) zu erhalten. Die puuid wird bei erfolgreichem Request in den Streamlit session state geschrieben. Von dort aus können alle weiteren Seiten auf diese zugreifen und ihre eigenen Requests an die Riot API verschicken, um die Match Daten eines bestimmten Spielers zu erhalten.

4.2.2 Karten-Komponente

Die ursprüngliche Idee zur Implementierung der Karten-Komponente war es, eine der in Streamlit eingebauten Kartenelemente zu verwenden. Diese würden das Einfügen von Datenpunkten ermöglichen und allgemeine Funktionalitäten wie Scrollen und Schwenken bereitstellen. Für die League of Legends Karte war dieser Ansatz jedoch nicht geeignet, da die eingebauten Kartenelemente nur SVGs als Kartenbilder erlauben und darüber hinaus keine Interaktion mit den Kartenpunkten ermöglichen.

Die erste konkrete Umsetzung erfolgte dementsprechend mit reinem HTML/CSS/Javascript, welches in Streamlit mittels der Components API möglich ist. Der erste Prototyp bestand also aus einem Hintergrundbild von der Summoners Rift Karte und ersten Beispieldaten, welche durch Anklicken ein zugehöriges Informationspopup anzeigen. Die Koordinaten der einzelnen Events konnten aus den Daten der Riot API ausgelesen werden. Diese wurden auf einer Skala von 1 bis 16.000 angegeben und mussten daher in prozentuale Werte umgewandelt werden, um sie an der richtigen Stelle im Bild anzuzeigen. Das Hintergrundbild der Summoners Rift Karte musste zur korrekten Platzierung der Punkte skaliert und gestreckt werden, um eine echte „top-down“ Perspektive der Karte zu haben. Diese Implementierung konnte zwar mit Beispieldaten in Streamlit eingebaut werden, jedoch konnte nicht mit Live-Daten gearbeitet werden, da dies mit der reinen HTML-Implementierung nicht von Streamlit angeboten wird.

Um Kommunikation zwischen Streamlit und der Karten-Komponente zu ermöglichen, musste die Karte erneut mit React umgesetzt werden. Diese erlaubt letztendlich Kommunikation in beide Richtungen, das heißt von Streamlit zur Komponente hin und zurück. Die von Streamlit übergebenen Informationen setzen sich aus den Riot API Daten zusammen, welche nach Spiel und Art des Events unterschieden werden. Diese werden aus der Match-Timeline jedes der fünf zu analysierenden Spiele herausgelesen, mit allen für

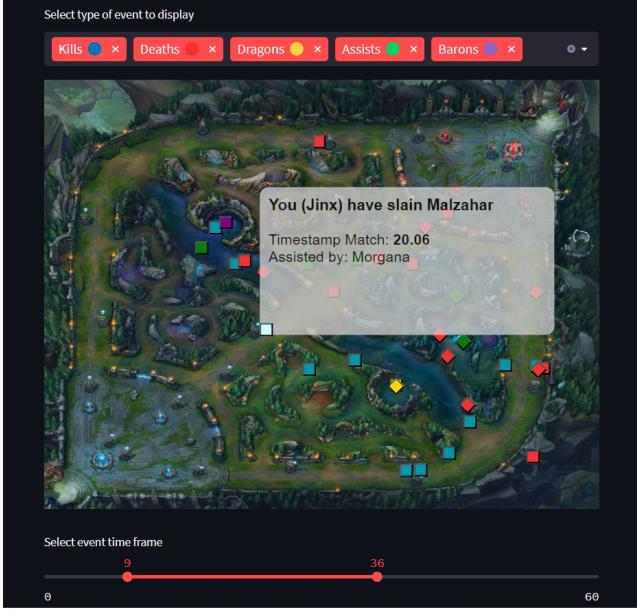


Abbildung 11: Screenshot der Karten-Komponente des Prototypen

das Event relevanten Informationen. Darüber hinaus werden von den Slidern, Multiselects und Checkboxen Informationen übergeben, welche bestimmen, welche Punkte auf der Karte tatsächlich angezeigt werden sollen.

Jedes Tooltip der React Komponente besteht aus einem Button und Pop-up. Farbe, Form und Inhalt werden von den Informationen bestimmt, die zu dem Tooltip gehören und von Streamlit an die Komponente übergeben werden. Die Informationen bestehen aus Positionsdaten, Zeitpunkt im Spiel, an dem das Event passiert ist, Spielzugehörigkeit und den Daten, welche für das jeweilige Event relevant sind, wie zum Beispiel beteiligte Champions. Jedes Tooltip stellt eine eigene React Komponente dar und besteht aus einem farbigen Button und dem aufklappbaren Text. Bei Betätigung des Buttons wird ein Signal an die Tooltip Komponente übermittelt, welches das Öffnen und Schließen des zum Tooltip gehörenden Textes bewirkt. Der Text des Tooltips hängt von der Art des Events ab und stellt sich aus den Informationen zusammen, die von Streamlit an die oberen Komponenten übergeben werden.

4.2.3 Statistik-Komponente

Die Aufteilung der Statistik Komponenten Seite in zwei Unterseiten wird mithilfe von Streamlit Tabs ermöglicht. Sowohl der Graph als auch das Balkendiagramm greifen auf die Variable vom Slider zu, um die Anzahl angezeigter Spiele zu bestimmen.

Der Graph wurde mit einer in Streamlit eingebauten Chart Bibliothek namens „Altair“ erstellt. Die Daten werden aus der Match-Timeline gelesen, um für jede Spielminute einen Eintrag im Graphen zu erhalten. Die Daten werden in einem Python Dictionary gespeichert, bevor sie zu einem Pandas Dataframe konvertiert werden. Hierbei ist es wichtig, die Daten im „Tidy data“ [21] Format abzuspeichern, das heißt jede Variable hat ihre eigene Spalte, jede Zeile ist eine Beobachtung und jede Art von Beobachtungseinheit bekommt ihre eigene Tabelle. Beim Erstellen des Graphen konnte eingestellt werden, dass jeder Linienverlauf seine eigene Farbe und ein eigenes Linienmuster erhält.

Das interaktive Balkendiagramm wurde wie die Karten Komponente als Streamlit Komponente umgesetzt. Hierbei wurde die D3 Bibliothek verwendet, um das Modell des Balkendiagramms zu erstellen. Das Model wurde größtenteils von einem Beispiel auf



Abbildung 12: Screenshot des Liniendiagramms der Statistik-Komponente des Prototypen

der offiziellen D3 Website [4] übernommen und für die individuellen Bedürfnisse der Anwendung angepasst. Um beispielsweise das D3 Modell in React einzubinden und eine Anpassung der von der Komponente angezeigten Daten zu ermöglichen, wurden sogenannte „React Hooks“ verwendet, die bei jedem verändern der Daten ein wiederholtes Laden der Komponente auslösten. In die „useEffect“ Hook wurde also der gesamte Aufbau des interaktiven Charts eingefügt und von den übergebenen Daten und der Farbe der Komponente abhängig gemacht.

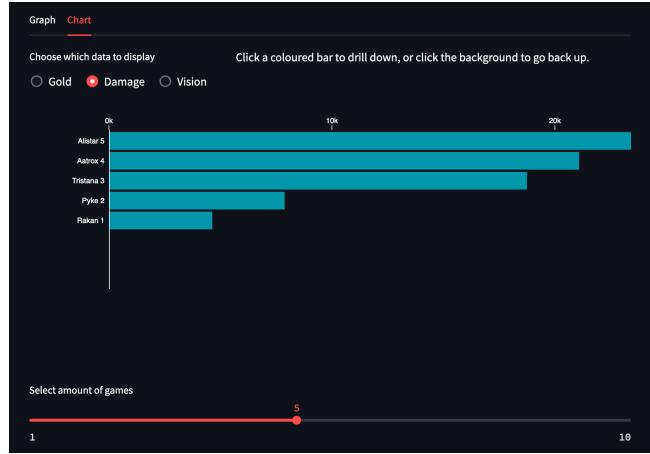


Abbildung 13: Screenshot des Balkendiagramms der Statistik-Komponente des Prototypen

Die Größe der Hintergrundfläche und Beschriftung der x und y Achsen ist auch von äußeren Parametern abhängig wie der Anzahl Spiele, die angezeigt werden sollen. Aus diesem Grund werden diese auch außerhalb des useEffect Hooks als Elemente definiert und werden von innerhalb des Hooks nur referenziert. Die Daten werden

nach entnehmen aus der API zu einem Dictionary zusammenge- setzt, welches für jede Kategorie (Gold, Damage, Vision) einen eigenen Aufbau hat. Jeder Eintrag im Dictionary besteht aus einem Namen und entweder einem konkreten Wert oder einem Array namens „children“, welches selber aus Dictionaries mit dem gleichen Aufbau besteht. Somit wird jedes der drei Dictionaries zu einer Art Baumstruktur überführt, bei dem nur an den Blättern konkrete Werte stehen. Diese konkreten Werte werden von der Chart Komponente dann zusammengezählt, um den angezeigten Wert des Elternknotens zu bestimmen.

4.2.4 Letzte-Spiele-Komponente

Die Letzte-Spiele-Komponente konzentriert sich auf den Vergleich der persönlichen Leistung über mehrere Spiele in Bezug auf eine bestimmte Lane. Für diese Visualisierung werden nur die Spielerdaten und nicht die einzelnen detaillierten Matchdaten mit dem genauen Zeitverlauf benötigt.

Zu Beginn wird mithilfe der registrierten Spieler-ID eine festgelegte Anzahl von Benutzerspielen über die API abgefragt. Leider ist es nicht möglich, die Abfragen nach verschiedenen Kriterien wie zum Beispiel den Spielmodus oder der gespielte Lane zu filtern. Aus diesem Grund muss zunächst eine vordefinierte Menge an Spielen abgefragt werden und erst danach kann nach den relevanten Kriterien wie Lane und Spielmodus gefiltert und sortiert werden. Zur Bestimmung der Lane wird auf die vom Spielserver berechnete Teamposition zurückgegriffen, die unter Berücksichtigung der Mitspieler und deren Position die wahrscheinlichste Spielposition eines Spielers berechnet [9]. Dabei wird eine durchschnittliche Genauigkeit von 87,5% erreicht [18]. Auf komplexere Berechnungen mithilfe von Spielquoten oder maschinellen Lernalgorithmen wurde wegen der Rechenzeit bewusst verzichtet. Eine Fehlklassifikation von ca. 10% ist in diesem Anwendungsbeispiel akzeptabel, da für die anvisierte Zielgruppe in der Praxis eine höhere Genauigkeit erreicht wird, da seltener Lane Wechsel stattfinden. Für jedes Spiel werden zusätzlich die gleichen Daten des direkten Gegners (Spieler des gegnerischen Teams auf der gleichen Lane) gespeichert. Abschließend wird für jede Lane und jeden Einzelwert der Mittelwert berechnet.

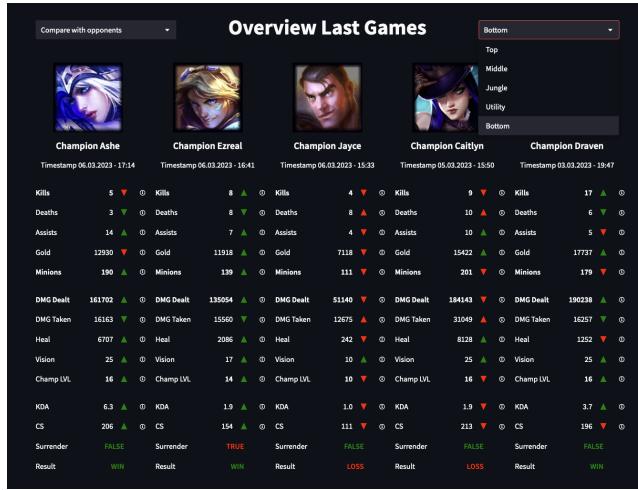


Abbildung 14: Screenshot der Letzte-Spiele-Komponente des Prototypen

Abbildung 14 zeigt die implementierte Komponente des Prototyps. Mit Streamlit wird ein Layout mit fünf Spalten erzeugt, wobei jede Spalte einem Spiel entspricht. Über zwei Auswahlboxen neben der Überschrift kann die Lane und die Art des Vergleichs ausgewählt werden. Der Inhalt der Seite wird dann dynamisch geladen. Durch

einfache, farblich hervorgehobene Pfeile wird dem Nutzer der Vergleich seiner persönlichen Leistung mit dem eigenen Durchschnitt oder dem direkten Gegner visuell aufbereitet. Klickt der Nutzer auf das direkt daneben befindliche Info-Icon, erhält er nähere Informationen, was der Wert bedeutet und dem verglichenen Wert.

4.2.5 Spielphasen-Komponente

In der Spielphasen-Komponente werden die Benutzerstatistiken an den ungefähren Breakpoints des Spiels angezeigt. Da es keine genaue Definition gibt, wann diese Breakpoints stattfinden, haben wir uns an unserer eigenen Erfahrung und der gängigen Meinung der Community orientiert.

So haben wir das Early-Game bis zur 15ten Minute begrenzt. Diese Zeitmarke wird vom Großteil der Community als solcher akzeptiert. Zu diesem Zeitpunkt haben die meisten Spieler ein Item fertig ausgebaut und meist ist mindestens ein Turm auf der Karte bereits zerstört oder steht kurz vor der Zerstörung.

Das Mid-Game haben wir bis zur 25. Minute begrenzt. Dieser Zeitmarker kann kritischer betrachtet werden, da Spiele je nach Meta, Teamzusammensetzung und Elo der Spieler natürlich kürzer oder länger sein können. Dadurch verschiebt sich auch der Zeitpunkt, zu dem das Mid-Game endet. Wir haben uns für die 25-Minuten-Marke entschieden, da die durchschnittliche Spieldauer [13] über alle Elo-Stufen hinweg über dieser Marke liegt und somit davon ausgegangen werden kann, dass hier die meisten Partien in das definierte Late-Game fallen.

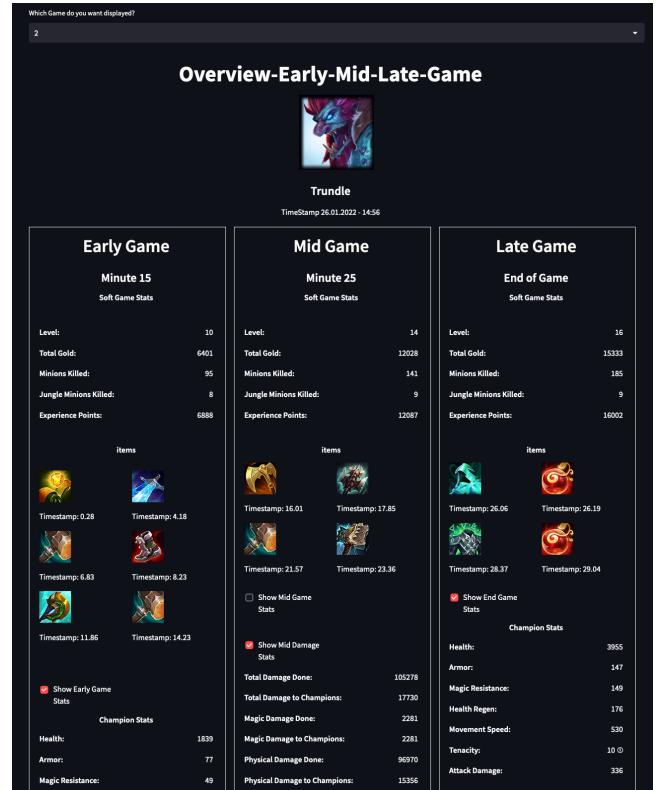


Abbildung 15: Screenshot der Spielphasen-Komponente des Prototypen

Das Ende des Late-Games ist sehr einfach und klar zu definieren, da mit dem Ende des Late-Games die Partie beendet ist. Für die Champion- und Itembilder wird die aktuelle Patch-Version des Spiels über folgende URL „<https://ddragon.leagueoflegends.com/api/versions.json>“ abgerufen. Da-

mit wird sichergestellt, dass beim Hinzufügen oder Ändern von Items im Spiel keine falschen Informationen abgefragt werden. Dies gilt auch, wenn neue Champions zum Spiel hinzugefügt werden.

Für diese Komponente müssen drei API-Endpunkte für die Spielstatistiken angefordert werden. Sie stammen alle aus MATCH-V5. Der Endpunkt „/lol/match/v5/matches/by-puuid/puuid/ids“ wird verwendet, um die matchId des Benutzers abzufragen. Der Endpunkt „/lol/match/v5/matches/matchId“ wird verwendet, um die End-of-Game-Informationen aller am Match beteiligten Spieler abzufragen. Dies ist notwendig, um im dritten Endpunkt jedem Spieler den richtigen Champion zuzuordnen zu können. Denn der dritte Endpunkt „/lol/match/v5/matches/matchId/timeline“ gibt nicht an, welchen Champion die einzelnen Spieler spielen. Hier wird nur mit „puuid“ und „participantId“ gearbeitet.

Um den Nutzer beim Aufrufen dieser Komponente mit Informationen zu erschlagen, lassen sich in jeder der drei Zeitphasen die Game- und Damagestats mit dem Anklicken einer Checkbox ein- und ausblenden.

4.2.6 Tipp-Komponente

Die Tipp Seite enthält bis zu fünf Hinweise, die angezeigt werden können, welche aus einer Auswahl von neun verschiedenen Tipps ausgewählt werden. Die Tipps werden für jedes der letzten fünf Spielen bestimmt und nach Häufigkeit des Vorkommens sortiert. Informationen aus den Match-Daten werden verwendet, um zu bestimmen, ob ein bestimmter Hinweis angezeigt werden soll.

Manche der Hinweise können nur dann angezeigt werden, wenn eine bestimmte Rolle gespielt wurde. Ein Beispiel dafür ist der „monster objective tip“, welcher einen darauf hinweist, dass Monster wie Dragon, Herald und Baron wichtig für das gesamte Team ist und nicht an den Gegner verschenkt werden sollten. Dieser Hinweis kann nur dann angezeigt werden, wenn der Spieler die Rolle des Junglers oder des Supporters ausgewählt hat, da diese Rollen am meisten von diesem Hinweis betroffen sind und am meisten Einfluss auf das Einnehmen von solchen Objectives haben.

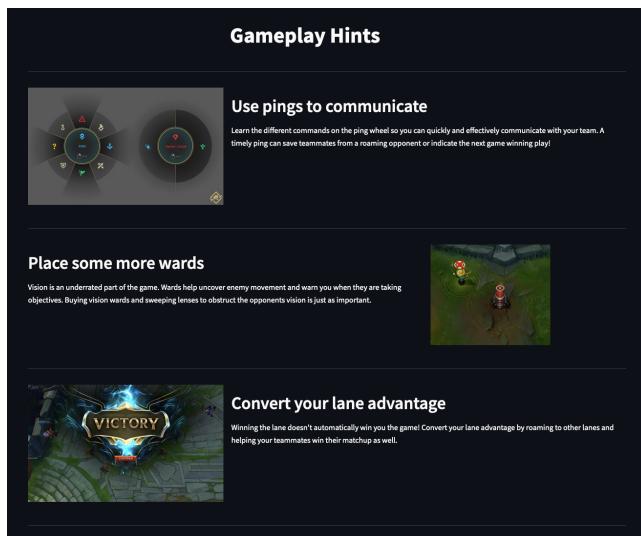


Abbildung 16: Screenshot der Tipp-Komponente des Prototypen

Viele der Tipps machen auch einen direkten Vergleich mit dem Gegner, der die gleiche Rolle wie man selber gespielt hat. Der „farming tip“ vergleicht beispielsweise den eigenen erzielten Farm mit dem des Spielers, der auf dem gegnerischen Team die gleiche Rolle gespielt und sich somit auf der gleichen Lane befunden hat.

5 EVALUATION

Ziel der Evaluation ist es, den implementierten Prototypen mit den verschiedenen Datenvisualisierungen von realen Nutzern testen zu lassen und das daraus resultierende Feedback zu analysieren. Die Ergebnisse der Evaluation sollen Aufschluss darüber geben, ob die vorgestellten Ansätze zur Datenaufbereitung und Datenvisualisierung für den beschriebenen Anwendungsfall geeignet sind beziehungsweise welche Erkenntnisse sich daraus für die Zukunft ableiten lassen.

5.1 Aufbau Nutzertest

Um den zuvor implementierten Prototypen zu testen und ein erstes Feedback zu erhalten, wurde ein Nutzertest mit 9 Testpersonen durchgeführt. Die Probanden wurden so ausgewählt, dass sie die angestrebte Zielgruppe bestmöglich repräsentieren. Der Großteil der Testpersonen spielt League of Legends seit mehr als fünf Jahren und durchschnittlich einmal pro Woche. Dabei werden sowohl Ranglistenspiele als auch normale Spiele mit Freunden gespielt. Der Fokus beim Spielen liegt bei allen auf dem Spaß und den Verbesserern ihrer eigenen Fähigkeiten. Auf einer Skala zur Einschätzung der eigenen Spielstärke von 1 (Anfänger) bis 5 (Profi) liegt der Durchschnitt bei 3.4 und somit in der Zielgruppe der Casual League of Legends Spieler. Darüber hinaus nutzt etwa die Hälfte der Tester bereits auf dem Markt befindliche Analysetools (Blitz, Professor, OP.GG und In-Game-Statistiken), allerdings sehr unregelmäßig und eher selten.

Jeder Tester hatte maximal 15 Minuten Zeit, um den Prototyp in Form einer Webanwendung selbstständig zu betrachten und zu bedienen. Die einzige vorgegebene Aufgabe war es, jede Unterseite zu besuchen und anzusehen. Während dieser Zeit wurden Notizen über Auffälligkeiten während des Tests, sonstige Beobachtungen und die Verweildauer auf den jeweiligen Unterseiten gemacht. Anschließend wurden die Tester gebeten, zwei Fragebögen auszufüllen. Der erste Fragebogen konzentrierte sich auf die einzelnen Unterseiten und wie welche Informationen dargestellt wurden. Der zweite Fragebogen ist der sogenannte „AttrakDiff“, ein standardisierter Fragebogen zur Usability von Webseiten.

5.2 Auswertung Nutzertest

Die Auswertung des Nutzertests gliedert sich in drei Sektionen. Zunächst werden die Auffälligkeiten und Probleme, die während des 15-minütigen Tests des Prototyps aufgetreten sind, diskutiert. Anschließend erfolgt die Auswertung der beiden Fragebögen, um die Stärken und Schwächen des Prototyps zu identifizieren und mögliche Einschränkungen und Fehlerquellen aufzuzeigen.

5.2.1 Beobachtungen

Während des Tests sind uns nur kleinere Probleme und Schwierigkeiten der Nutzer aufgefallen. Grundsätzlich kamen alle Tester mit der Bedienung der Anwendung gut zurecht und hatten keine Probleme. Einige technische Probleme traten jedoch auf, wie zum Beispiel die Kompatibilität der Webanwendung mit verschiedenen Browsern. Aber auch der Wechsel zwischen dunklem und hellem Systemdesign und die entsprechende Anpassung der Hintergrundfarbe ließ an vereinzelten Stellen den Text mit wichtigen Informationen „im Hintergrund verschwinden“. Außerdem kam es zu Überlappungen der Tooltips auf der Karten-Komponente, was die Sichtbarkeit der Informationen deutlich erschwerte.

Im Durchschnitt benötigten die Tester 9 Minuten und 59 Sekunden, um sich die Anwendung vollständig anzusehen. Am längsten wurde mit mehr als zwei Minuten die Statistik- und Spielphasen-Komponente betrachtet. Am kürzesten verweilten die Befragten bei der Tipp-Komponente. Besonders auffällig war der häufige Wechsel zwischen den Unterseiten, sodass es sich bei den in Tabelle 1 angegebenen Zeiten teilweise um aufsummierte Werte handelt und somit eine gewisse Messungenauigkeit bestehen kann.

Komponenten Name	Durchschnittliche Besuchszeit
Startseite	00:31 Minuten
Karten-Komponente	01:45 Minuten
Statistik-Komponente	02:26 Minuten
Letzte-Spiele-Komponente	01:54 Minuten
Spielphasen-Komponente	02:06 Minuten
Tipp-Komponente	01:17 Minuten
Gesamtzeit Anwendung	09:59 Minuten

Tabelle 1: Durchschnittliche Besuchszeit der einzelnen Komponenten

Zuletzt ist negativ aufgefallen, dass die Ladezeiten und die damit verbundene Datenaufbereitung auf einigen Unterseiten sehr lang waren. Dies ist auf die von uns gewählte Methode zur Veröffentlichung des Prototyps zurückzuführen. Die Website wird auf einer freien Plattform des verwendeten Python-Frameworks Streamlit gehostet und verfügt daher nur ein begrenztes Kontingent an Rechenleistung.

5.2.2 Fragebogen

Zu jeder Unterseite wurden zwei identische Fragen gestellt. Zum einen, wie hilfreich die dargestellten Informationen waren, zum anderen, wie geeignet die Darstellungsform für die Informationen war. Wie in Abbildung 17 zu sehen ist, schnitt die Letzte-Spiele-Komponente hinsichtlich des Nutzen der Informationen mit einem Median von 5 am besten ab. Es ist jedoch erwähnenswert, dass der Median aller anderen Komponenten bei 4 liegt, was sehr gut ist.

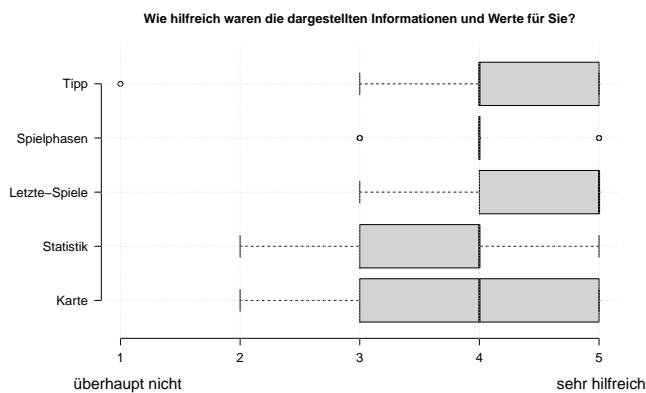


Abbildung 17: Box-Whisker-Plot über jede Komponente wie hilfreich die dargestellten Informationen für die Nutzer waren

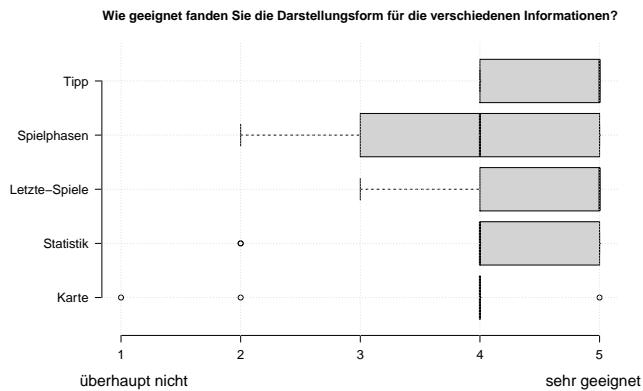


Abbildung 18: Box-Whisker-Plot über jede Komponente wie hilfreich die Darstellungsform für die gegebenen Informationen waren

Abbildung 18 zeigt das Ergebnis bezüglich der Eignung der gewählten Darstellungsformen. Am besten schneiden hier die Letzte-Spiele-Komponente und die Tipp-Komponente mit einem Median von 5 (sehr hilfreich) ab. Der Median der anderen Komponenten liegt wiederum bei 4 und ist somit gut. In der Grafik fällt besonders auf, dass die Karten-Komponente die meisten Meinungsverschiedenheiten hervorruft und somit auch die meisten Ausreißer bei den Werten hat.

Die Hinweise der Karten-Komponente wurden überwiegend als hilfreich bis sehr hilfreich eingestuft und es waren keine überflüssigen Informationen vorhanden, jedoch hätten sich einige Teilnehmer mehr Informationen zu den einzelnen Hinweisen, wie zum Beispiel die Teamfarbe, gewünscht. Die Darstellungsform der Karte und der Pop-up Events wurde ebenfalls als „geeignet“ bewertet, jedoch wurde bemängelt, dass sich die Pop-up-Fenster an den Zoom angepasst haben und sich oft gegenseitig überlappen. Darauf hinaus war manchen Nutzern der Zusammenhang zwischen den Punkten auf der Karte und den dazugehörigen Ereignissen nicht immer eindeutig. Die Bedienung war für die Nutzer größtenteils kein Problem, einige hatten jedoch Schwierigkeiten damit, die Pop-up-Fenster nach dem Öffnen wieder zu schließen.

Bei der Statistik-Komponente wurden die dargestellten Informationen größtenteils als hilfreich eingestuft und keines der Informationen wurde als überflüssig empfunden, es fehlte einigen Nutzern jedoch Statistiken wie „Wards placed“, verursachter Schaden und „Objectives“. Die meisten Nutzer empfanden auch die Darstellung von Informationen mit Graphen/Linien besser im Vergleich zur tabellarischen Darstellung. Die Darstellungsform wurde überwiegend als geeignet eingeschätzt. Die Animationen der Balkendiagramme wurden von der Hälfte der Befragten als hübsch, von der anderen Hälfte als neutral/irrelevant angesehen. Des Weiteren empfanden die meisten Nutzer, dass ihnen diese Art von Visualisierung relativ selten begegnet. Mit der Bedienung des Graphen und der Charts kamen die meisten Nutzer auch gut zurecht.

Die tabellarische Darstellung der vielen Zahlen in der Letzte-Spiele-Komponente wurde überwiegend positiv aufgenommen, nur eine Person fühlte sich von der Menge der Zahlen überfordert. Es wurden keine Informationen als überflüssig empfunden oder Werte vermisst. Der Vergleich der Werte mit dem eigenen Durchschnitt und der Leistung des direkten Gegners wurde sehr positiv aufgenommen. Drei Tester wünschten sich eine Erweiterung dieser Idee, zum Beispiel in welchen Metriken man der Beste im Team war oder wie der Durchschnitt des direkten Gegners ist. Die Hinweise zu den einzelnen Werten wurden von ca. 70% der Befragten genutzt und als hilfreich empfunden. Lediglich die Hervorhebung einzelner Werte für bestimmte Rollen war für die Nutzer visuell nicht deutlich genug. Sie wurde zum Teil gar nicht wahrgenommen oder konnte nur schwer gelesen werden. Eine Alternative wäre eine deutliche farbliche Hervorhebung der Werte.

Der Nutzerumfrage zur Folge wurde die Spielphasen-Komponente insgesamt als hilfreich und geeignet in ihrer Darstellungsform bewertet. Wobei ein Drittel der Befragten die Eignung der Darstellungsform als neutral bis eher ungeeignet bewertete. Weiterhin zeigte sich, dass für die überwiegende Mehrheit der Nutzer die Unterscheidung zwischen Early-, Mid- und Late Game tendenziell relevant bis sehr relevant ist. Lediglich ein Nutzer bewertete die Relevanz als neutral. Wie bereits in Kapitel Unterunterabschnitt 4.2.5 beschrieben, bestätigt unsere Umfrage, dass die zeitliche Einordnung von Early-, Mid- und Lategame nicht klar definiert ist. Vor allem der Zeitpunkt, wann das Midgame endet, wird auch von unseren Befragten unterschiedlich gesehen, bzw. die statische zeitliche Trennung ist nicht optimal. Knapp die Hälfte der Befragten stimmt der zeitlichen Einteilung zu, jedoch sieht der Rest diese Einteilung aber anders.

Die Hinweise der Tipp-Komponente wurden überwiegend als hilfreich bis sehr hilfreich eingestuft. Die meisten Nutzer empfanden kei-

ne der angezeigten Tipps als überflüssig, manche empfanden einen oder sogar alle Tipps als überflüssig. Die meisten Teilnehmer hatten auch das Gefühl, dass die Hinweise gut auf sie und ihre Schwächen angepasst waren. Die Tipps wurden auch größtenteils als ansprechend empfunden und die meisten Teilnehmer meinten, dass sie für ihr nächstes Spiel zwischen drei und sechs Tipps berücksichtigen könnten. Passende Negativbeispiele zu den Tipps hätten einem Teilnehmer weitergeholfen, um die Hinweise besser einzuordnen, während ein anderer Teilnehmer die Tipps als zu allgemein verfasst empfand.

Zusammenfassend kann gesagt werden, dass der Nutzertest für die Evaluierung des Prototyps sehr hilfreich war. Die Rückmeldungen in den Fragebögen waren positiv und die Kommentare und Kritiken sehr aufschlussreich. Die abschließende Frage des ersten Fragebogens, wie die Tester den Prototyp mit einer Schulnote bewerten würden (siehe Abbildung 19), ergab eine Durchschnittsnote von 2,1. Etwas mehr als die Hälfte der Nutzer würde die Anwendung weiter nutzen.

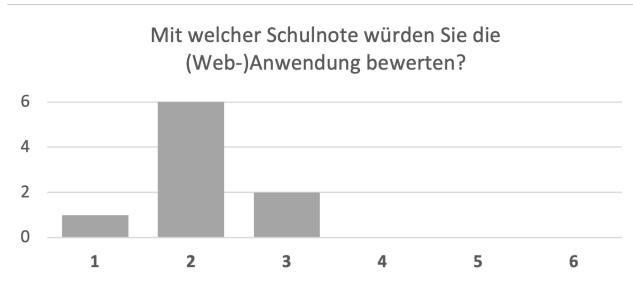


Abbildung 19: Gesamtbewertung des Prototypen mit Schulnoten

5.2.3 AttrakDiff

Mit Hilfe von AttrakDiff ist es möglich, die Bedienung und das Aussehen eines interaktiven Produktes subjektiv von den Benutzern beurteilen zu lassen. In unserem Nutzertest haben wir die Kurzform des standardisierten Tests verwendet. Dieses Evaluationsverfahren erfasst sowohl die wahrgenommene pragmatische als auch die hedonische Qualität und Attraktivität des Prototyps [10].



Abbildung 20: AttrakDiff Ergebnisüberblick in Portfolio-Darstellung

Abbildung 20 zeigt die Portfoliodarstellung. Vertikal wird die Ausprägung der hedonischen Qualität und horizontal die Ausprägung der pragmatischen Qualität beschrieben. Der Prototyp fällt demnach in den sogenannten Charakterbereich „handlungsorientiert“ mit Tendenz zu „begehr“. Das Konfidenzrechteck (hellblau) ist nicht zu groß und spiegelt damit wider, dass sich die Nutzer in der Bewertung der Anwendung im Großen und Ganzen einig sind.

Um die Bewertung über die Ausprägung der verschiedenen Qualitäten besser nachvollziehen zu können, sind in Abbildung 21 die einzelnen Wortpaare und ihre Zugehörigkeit zu den Ausprägungen genauer dargestellt. Dabei steht PQ für pragmatische Qualität, HQ für hedonische Qualität und ATT für Attraktivität. Die einzelnen Punkte symbolisieren den von den Befragten angegebenen Mittelwert. Es ist gut zu erkennen, dass alle Punkte auf der positiven Seite liegen, aber unterschiedlich gut bewertet wurden.

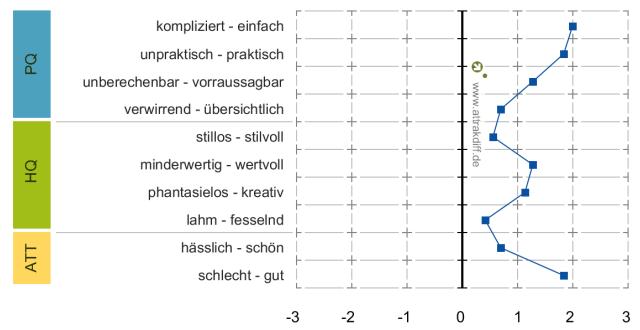


Abbildung 21: AttrakDiff Profil der Wortpaare

Zusammenfassend kann gesagt werden, dass auch der AttrakDiff Fragebogen über den Prototypen positiv aufgefallen ist. Darüber hinaus erhält man durch die Wortpaare Hinweise, an welchen Punkten verstärkt gearbeitet werden muss.

5.3 Laufzeitevaluation

Die Laufzeit der Anwendung war eines der Kritikpunkte der Nutzer, die die Umfrage ausgefüllt hatten. Beim Aufruf jeder Unterseite mussten die für die Seite relevanten Informationen per Request aus der Riot API angefragt werden, welches einen Großteil der Ladezeit beanspruchte. Wurden die Daten einmal für eine Unterseite geladen, konnten diese mit dem in Streamlit eingebauten Cache abgespeichert und schnell zugegriffen werden.

Komponenten Name	Ohne Cache	Mit Cache
Startseite	01:30 Sekunden	00:62 Sekunden
Karten-Komponente	19:32 Sekunden	00:76 Sekunden
Statistik-Komponente	11:33 Sekunden	00:88 Sekunden
Letzte-Spiele-Komponente	23:52 Sekunden	03:72 Sekunden
Spielphasen-Komponente	11:23 Sekunden	01:29 Sekunden
Tipp-Komponente	10:08 Sekunden	00:70 Sekunden

Tabelle 2: Durchschnittliche Aufrufzeit der einzelnen Komponenten innerhalb der Webanwendung im Laufzeittest

Wie in Tabelle 2 zu sehen ist, waren die Ladezeiten zum Teil 10 bis 20 Sekunden länger, wenn die Daten nicht bereits im Cache vorhanden waren. Die Anzahl Requests, die jede Unterseite abschicken musste, variierte auch, weil jede Seite unterschiedlich viele Match Daten benötigte. Da jede Seite voneinander unabhängig ist, konnten auch nicht die Daten einmal für alle Unterseiten geladen werden. Die einzige Möglichkeit, zwischen verschiedenen Seiten Informationen zu teilen ist der Streamlit Session State, in welchen nur einzelne Variablen wie die userId geschrieben werden können.

Komponenten Name	Lokale Ausführung
Startseite	00:89 Sekunden
Karten-Komponente	04:69 Sekunden
Statistik-Komponente	02:15 Sekunden
Letzte-Spiele-Komponente	05:15 Sekunden
Spielphasen-Komponente	02:69 Sekunden
Tipp-Komponente	04:58 Sekunden

Tabelle 3: Durchschnittliche Aufrufzeit der einzelnen Komponenten innerhalb der Webanwendung im lokalen Test, ohne Daten im Cache

Eine Verbesserung der Laufzeit wäre fast ausschließlich von der Dauer des Datenladens abhängig. Beim lokalen Ausführen der Anwendung konnten die verschiedenen Methoden wesentlich schneller durchgeführt werden, welches in Tabelle 3 zu sehen ist. Dieses ist auch ein Indikator dafür, dass die Seite nicht auf einem besonders leistungsfähigen Server gehostet wird.

6 KONKLUSION UND ZUKÜNTIGE ARBEITEN

Zusammenfassend lässt sich sagen, dass unser Analysetool auf dem richtigen Weg ist. Die Nutzerumfrage deutet darauf hin, dass die Komponenten des Analysetools einen guten Einblick in die von der API gesammelten Daten bieten, um ihre Spieldaten zu analysieren und Einblicke in ihre Leistung zu erhalten. Zusätzlich zu den üblichen Datenanalysemethoden und statistischen Auswertungen bietet unsere Anwendung die Möglichkeit, den Spielern durch die raum-zeitliche Auswertung von ereignisgesteuerten Daten neue Einblicke in ihr Spiel zu geben. Spieler können ihre Leistung analysieren und datenbasierte Entscheidungen treffen, um ihr Gameplay zu verbessern.

Nach Auswertung der über die Riot API bereitgestellten Daten konnten wir feststellen, dass eine Vielzahl nützlicher Daten zur Verfügung gestellt wird. Allerdings gibt es auch Lücken. Vor allem wenn es um raum-zeitliche Daten geht. So haben wir bei verschiedenen Events vermisst, dass die Koordinaten aller Spieler beim Auslösen eines Events übermittelt werden. Wir denken, dass daraus viele nützliche Informationen für den Nutzer generiert werden können.

Basierend auf der von uns durchgeführten Nutzerumfrage und dem daraus erhaltenen positiven Feedback, nehmen wir an, dass unser Tool einen wichtigen Schritt nach vorne darstellen kann. In dem wir die Möglichkeiten der Datenvisualisierung nutzen, können wir den Spielern neue und innovative Möglichkeiten bieten, ihre Leistung zu verbessern und zu verstehen.

Dennoch sehen wir noch viel Potenzial, um die Anwendung weiter zu verbessern. Sei es durch mehr direkte Daten aus der API, kombinierte und aggregierte Werte, aus denen wir zusätzliche Informationen generieren können, oder besser auf die Rollen zugeschnittenen Informationen durch die Einführung einer besseren Filterlogik. Ebenso kann durch eine verbesserte und optimierte Laufzeit dem Nutzer eine bessere Erfahrung mit der Anwendung bereitgestellt werden.

DANKSAGUNG

An dieser Stelle möchten wir uns bei allen Teilnehmern unseres Nutzertests und der anschließenden Befragung für die konstruktive Kritik und auch das Lob bedanken.

LITERATUR

- [1] activeplayer.io. Active league of legends players. <https://activeplayer.io/league-of-legends/>, 2023. besucht am: 2023-03-09.
- [2] A. P. Afonso, M. B. Carmo, T. Gonçalves, and P. Vieira. Visualeague: Player performance analysis using spatial-temporal data. *Multimedia Tools and Applications volume 78*, July 2019. doi: 10.1007/s11042-019-07952-z
- [3] I. Blitz App. Blitz, supercharge your gameplay. <https://blitz.gg/>, 2023. besucht am: 2023-03-09.
- [4] M. Bostock. Hierarchical bar chart. <https://observablehq.com/@d3/hierarchical-bar-chart>, 2020. besucht am: 2023-03-09.
- [5] S. Charleer, K. Gerling, F. Gutierrez, H. Cauwenbergh, B. Luyckx, and K. Verbert. Real-time dashboards to support esports spectating. *SportRxiv Preprints*, Oct. 2018. doi: 10.1145/3242671.3242680
- [6] T. D. Do, S. I. Wang, D. S. Yu, M. G. McMillian, and R. P. McMahan. Using machine learning to predict game outcomes based on player-champion experience in league of legends. *FDG '21: Proceedings of the 16th International Conference on the Foundations of Digital Games*, Aug. 2021. doi: 10.1145/3472538.3472579
- [7] S. Duan. Bug report riot api. <https://github.com/RiotGames/developer-relations/issues/740>, 2023. besucht am: 2023-03-09.
- [8] S. Few. *Information Dashboard Design: The Effective Visual Communication of Data*. O'Reilly Media, 2006.
- [9] R. Games. Riot developer portal match-v5 getmatch documentation. <https://developer.riotgames.com/apis#match-v5/GET-getMatch>, 2021. besucht am: 2023-02-26.
- [10] U. I. D. GmbH. Attrakdiff. <https://www.attrakdiff.de>, 2023. besucht am: 2023-03-02.
- [11] S. Kim, D. Kim, H. Ahn, and B. Ahn. Implementation of user playstyle coaching using video processing and statistical methods in league of legends. *Multimedia Tools and Applications volume 80*, Aug. 2020. doi: 10.1007/s11042-020-09413-4
- [12] Y. Kou, Y. Li, X. Gui, and E. Suzuki-Gill. Playing with streakiness in online games: How players perceive and react to winning and losing streaks in league of legends. *CHI '18: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, Apr. 2018. doi: 10.1145/3173574.3174152
- [13] leagueofgraphs. Average play time. <https://www.leagueofgraphs.com/stats/game-durations>, 2023. besucht am: 2023-03-09.
- [14] P. Z. Maymin. Smart kills and worthless deaths: esports analytics for league of legends. *Journal of Quantitative Analysis in Sports*, pp. 11–27, Sept. 2020. doi: 10.1515/jqas-2019-0096
- [15] Mobalytics. Mobalytics - the all-in-one companion for every gamer. <https://mobalytics.gg/>, 2023. besucht am: 2023-03-09.
- [16] A. R. Novak, K. J. M. Bennett, M. A. Pluss, and J. Fransen. Performance analysis in esports: part 1 - the validity and reliability of match statistics and notational analysis in league of legends. *SportRxiv Preprints*, Oct. 2019. doi: 10.31236/osf.io/sm3nj
- [17] porofessor.gg. Porofessor.gg -league of legends live game search and real-time player statistics. <https://porofessor.gg/>, 2023. besucht am: 2023-03-09.
- [18] R. A. C. Revision. Identifying champion positions. <https://riot-api-libraries.readthedocs.io/en/latest/roleid.html>, June 2019. besucht am: 2023-02-26.
- [19] F. Rijnders, G. Wallner, and R. Bernhaupt. Live feedback for training through real-time data visualizations: A case study with league of legends. *Proceedings of the ACM on Human-Computer Interaction*, Oct. 2022. doi: 10.1145/3549506
- [20] SnowflakeInc. Streamlit - the fastest way to build and share data apps. <https://streamlit.io/>, 2023. besucht am: 2023-03-02.
- [21] H. Wickham. Tidy data. *Journal of Statistical Software, Volume 59, Issue 10*, Aug. 2014. doi: 10.18637/jss.v059.i10