Tracking Tool

Inhalt

Überblick 1
Bedienung 1
Vorbedingung 1
Benutzung 2
Tracking 3
Layer Control 4
Implementierung 5
Struktur 5
Logik 5
Ausblick und Anmerkungen 6
Kartenmaterial 6
Teilnehmerhierarchie 6
Icons 6
Potential: Leaflet 7

Überblick

Beim Tacking Tool handelt es sich um eine Webanwendung, welche sich eine WebSocket-Schnittstelle der Excon zu Nutze macht und darüber Übungsinformationen und Positionsdaten bezieht. Das Tool erlaubt somit die Anzeige aller aktiven Teilnehmer und Echtzeit-Verortung auf einer Leaflet-Karte.

Bedienung

Vorbedingung

Unter Firefox muss unter der URL about:config die Einstellung
security.fileuri.strict_origin_policy aktuell auf false gesetzt werden.

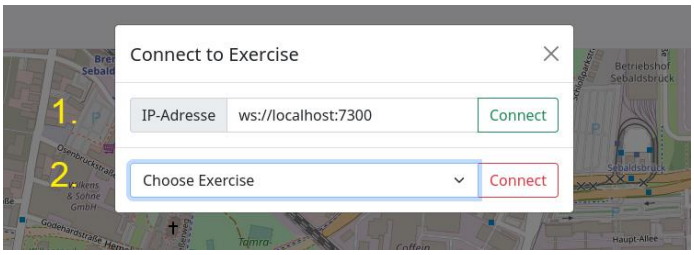
Für eine erfolgreiche WebSocket-Verbindung ist zunächst sicherzustellen, dass das Service Management gestartet wurde. Dies lässt sich in der Konsole durch folgenden Command überprüfen:

```
ps ax|grep svc
```

Ist dies nicht der Fall, lässt sich im Verzeichnis TDS/TDS-build-iteration/bin der Server mit folgenden Command starten:
./svcServiceManagement
Sollte es bei der Verbindung dennoch zu Komplikationen kommen, kann die Konsolenausgabe Hinweise auf das Fehlverhalten einzelner Services geben.

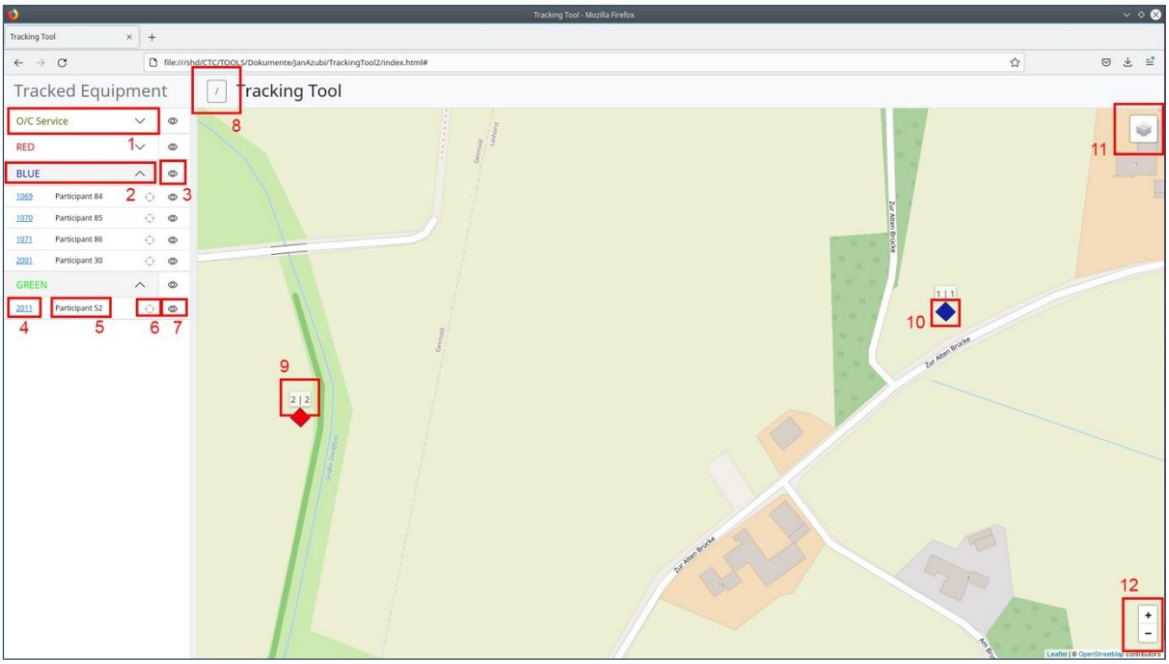
Benutzung

Zunächst muss eine Verbindung zum Service-Port 7300 hergestellt werden, wozu nach Laden der Anwendung ein Popup öffnet. Die korrekte Port-IP ist standesgemäß eingetragen (1). Nach erfolgreicher Verbindung, die durch den Farbwechsel des Connect-Buttons ersichtlich wird, kann eine zugehörige Übung gewählt werden (2).



Ist auch diese erfolgreich angewählt verfärbt sich der zugehörige Button abermals und das Popup-Fenster schließt sich.
Die Funktionalität des Tools ist nun gegeben, indem alle notwendigen WebSocket-Verbindungen eingegangen wurden. Das Tool kann nun Positionsmeldungen empfangen und legt für neue Parteien und die dazugehörigen Simulations-Teilnehmer in der linken Sidebar einen Eintrag an. Ein Marker zur Teilnehmer-Position wird der Karte hinzugefügt. Die Anwendung ist responsive und blendet die Sidebar bei Unterschreitung einer Fensterbreite ein. Sollte diese also verborgen sein, kann sie mit dem Button oben links ein- und ausgeklappt werden (8).

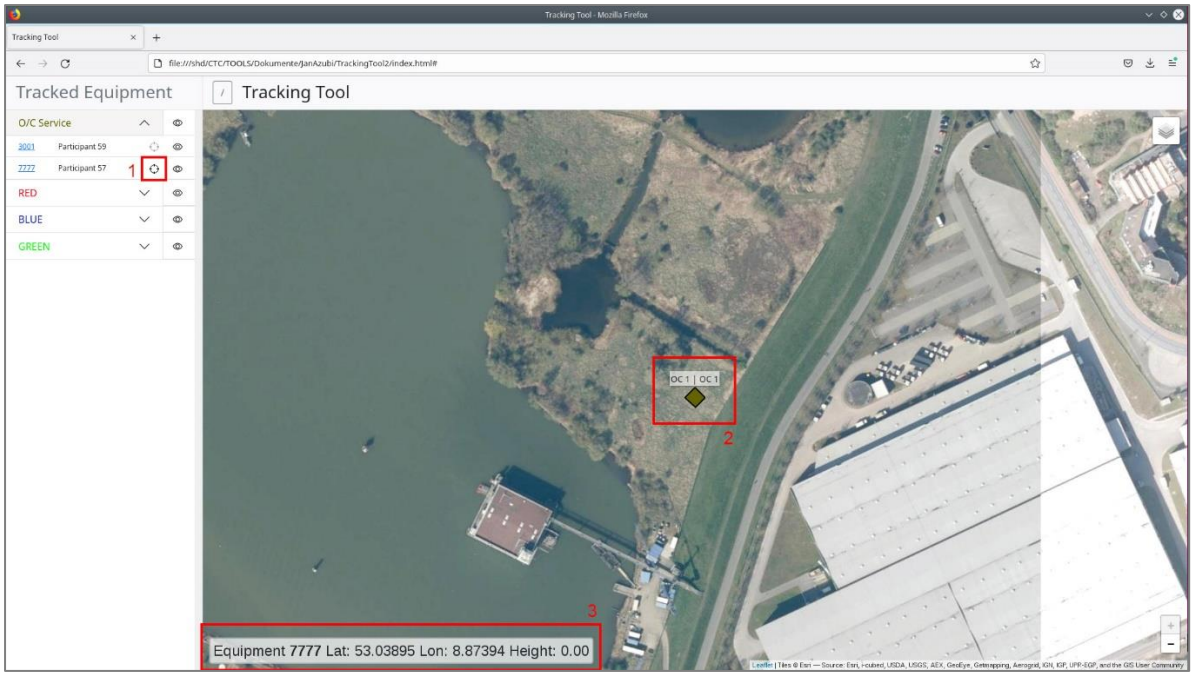
Weitere Funktionen sind im folgenden Screenshot markiert und betitelt.



1. Partei-Accordion mit Party ID (eingeklappt)
2. Partei-Accordion mit Party ID (ausgeklappt)
3. alle Teilnehmer der Partei ein- bzw. ausblenden
4. Equipment ID, Teilnehmer wird durch Klick zentriert
5. Participant ID
6. Tracking-Mode starten bzw. beenden (sh. Tracking)
7. Teilnehmer ein- bzw. ausblenden
8. Sidebar ein- und ausklappen
9. Teilnehmer-Informationen (Designator | Tactical Identifier)
10. Teilnehmer-Icon in Farbe der jeweiligen Party
11. Ebenenkontrolle (sh. Layer Control)
12. Hinein- und Herauszoomen der Karte

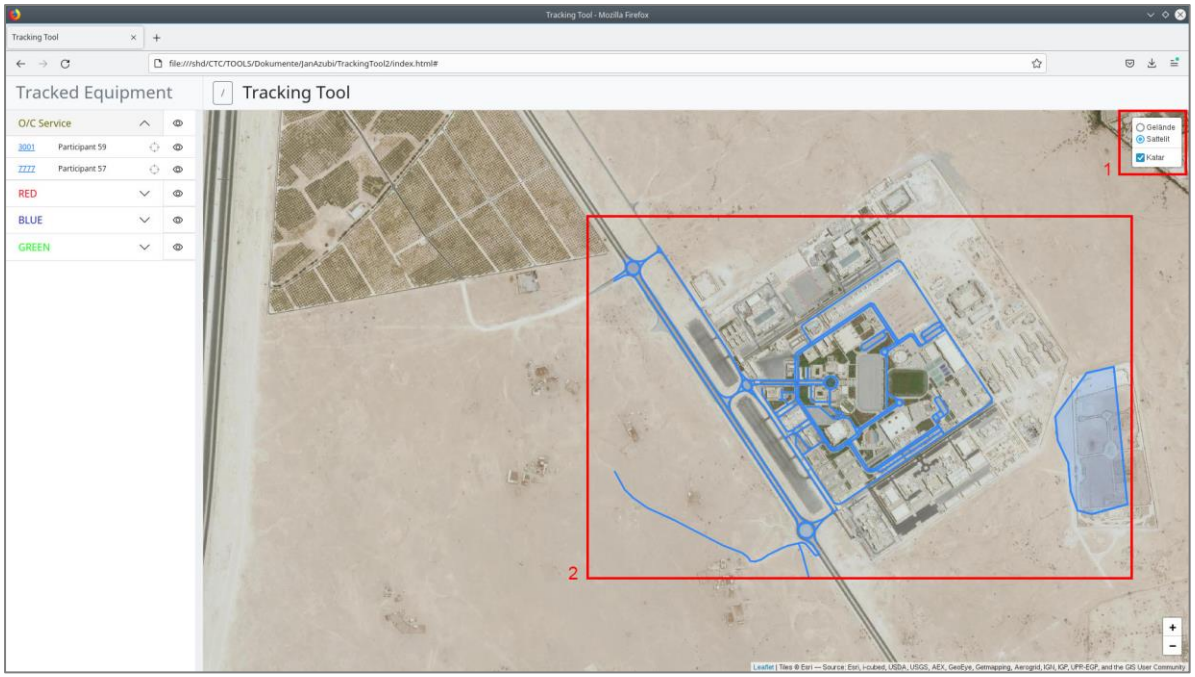
Tracking

Das Tool bietet die Möglichkeit einzelne Teilnehmer zu fokussieren und zu verfolgen. Somit wird bei jeder Bewegung des Teilnehmers die Karte neu zentriert. Die Tracking-Funktion wird mit dem Fadenkreuz neben der Participant ID in der Teilnehmer-Liste eingeschaltet (1). Wird ein Teilnehmer verfolgt, ist das Fadenkreuz-Symbol hervorgehoben und die Positionsdaten werden im unteren Bereich der Karte eingeblendet (2). Außerdem ist die Outline des Icons fett, was in vor Allem von benachbarten Teilnehmern abhebt (3).
Das Tracking lässt sich durch erneutes Klicken des Fadenkreuzes oder das Tracken eines anderen Participants aufheben



Layer Control

Leaflet bietet die Möglichkeit zwischen mehrere Karten und Overlays zu wechseln (1). Aktuell sind OSM-karten eingebunden. Für den Gebrauch der Anwendung im Feld wären lokale Kartendaten allerdings sinnvoller, da diese nicht nachgeladen werden müssen. Overlays können aus Vektor- und Pixelgrafiken bestehen. Zu Testzwecken ist bereits eine GeoJSON-Datei für Katar eingebunden (2), die sich in der Ebenenkontrolle einschalten lässt.
Weitere Informationen zu den sog. „Control.Layers“ als Ebenenkontrolle und Layer-Formaten finden sich in der ausgezeichneten Leaflet-Dokumentation unter folgendem Link: https://leafletjs.com/SlavaUkraini/reference.html
Es wäre denkbar Karten und Overlays nicht einfach einzublenden, sondern die Kontrolle so zu struktiveren und zu verknüpfen, dass bestimmte Übungsgebiete direkt mit zugehörigen Overlays eingeblendet werden und sich auch die Karte auf den Bereich zentriert.
Die Karten-Links und GeoJSON-Datei sind derzeit in einer separaten JS-Datei abgelegt und werden in die script.js importiert.



Implementierung

Struktur

Die Anwendung besteht neben HTML und CSS-File aus der script.js, welche die Equipment.js (bildet den Bauplan für jede neue Equipment-Instanz, die durch das Tool verortet wird) und map.js (enthält Karten-Links und GeoJSON-Datei) und helpers.js (enthält Hilfsfunktionen zur Umrechnung von Koordinaten und Farbwerten) importiert.
Externe Module bilden die Leaflet-Bibliothek als Gerüst für Karteneinbindung und Marker-Erstellung (Version 1.7.1) sowie das Bootstrap-Framework für das HTML-Gerüst und Styling (Version 5.1.3).

Logik

Alle Informationen werden in Programm als Key-Value-Paare in JSON-Objekten gespeichert, welche der Datenstruktur von Maps entsprechen. Den Key bildet meist die eindeutige Equipment ID, den Value ein weiteres Objekt. So lassen sich die dynamischen eintreffenden Daten von verschiedenen WebSockets über die Equipment ID zusammenführen.
Mit erfolgreicher WebSocket-Verbindung bezieht das Tool einmalig allgemeine Informationen zur den Partein (partyBaseData) und den angemeldeten Teilnehmern (participantBaseData) Diese Informationen werden wie beschreiben jeweils in einem JSON-Objekt gespeichert. Ein Verständnis über den Informationsgehalt bietet der Socket-Sniffer, ein Verständnis über die Objektstruktur die dazugehörige Konsolenausgabe.
Treffen nun Positionsmeldungen zu einzelnen Teilnehmern ein, wird geprüft, ob diese bereits erfasst sind. Wenn nicht wird eine Instanz der Equipment-Klasse erzeugt, die weitere Informationen aus den zuvor genannten Quellen hinzuzieht. Die Positionsmeldung allein enthält nämlich nur Equipment ID, Lat, Lon und Height. Somit lassen sich unveränderliche

Attribute wie Parteizugehörigkeit und der Ausstattungstyp mit der aktuellen Position zusammenführen.
Daraufhin wird, falls noch nicht vorhanden ein Objekt mit der Party ID angelegt. Die Party wird im Zuge dessen auch der Sidebar mit zugehörigem Namen und Farbe hinzugefügt. Danach wird die bereits erzeigte Equipment-Instanz in das Party- Objekt einsortiert und auch in der Sidebar unter der Party sortiert nach Equipment-ID untergeordnet. Die Datenstruktur entspricht somit auch der hierarchischen Struktur

Für alle folgenden Positionsmeldungen wird zuerst geprüft, ob es bereits Objekte zu mit entsprechender Party ID und darin Objekte mit Equipment ID gibt. Falls ja werden alle Instanziierunsschritte und das Rendern eines Icons übersprungen und lediglich die Positionsattribute überschrieben und der Marker auf der Karte versetzt.
Die Funktionen zum Ausblenden, Zentrieren und Verfolgen ergänzen diese Grundverzeigungen um weitere Bedingungen und sind im Code jeweils kommentiert. Einige Methoden sind auch Bestandteil der Equipment-Klasse.

Ausblick und Anmerkungen

Kartenmaterial

Überlegungen zur Anpassung des Kartenmaterials finden sich unter dem Kapitel Bedienung/ Layer Control.

Teilnehmerhierarchie

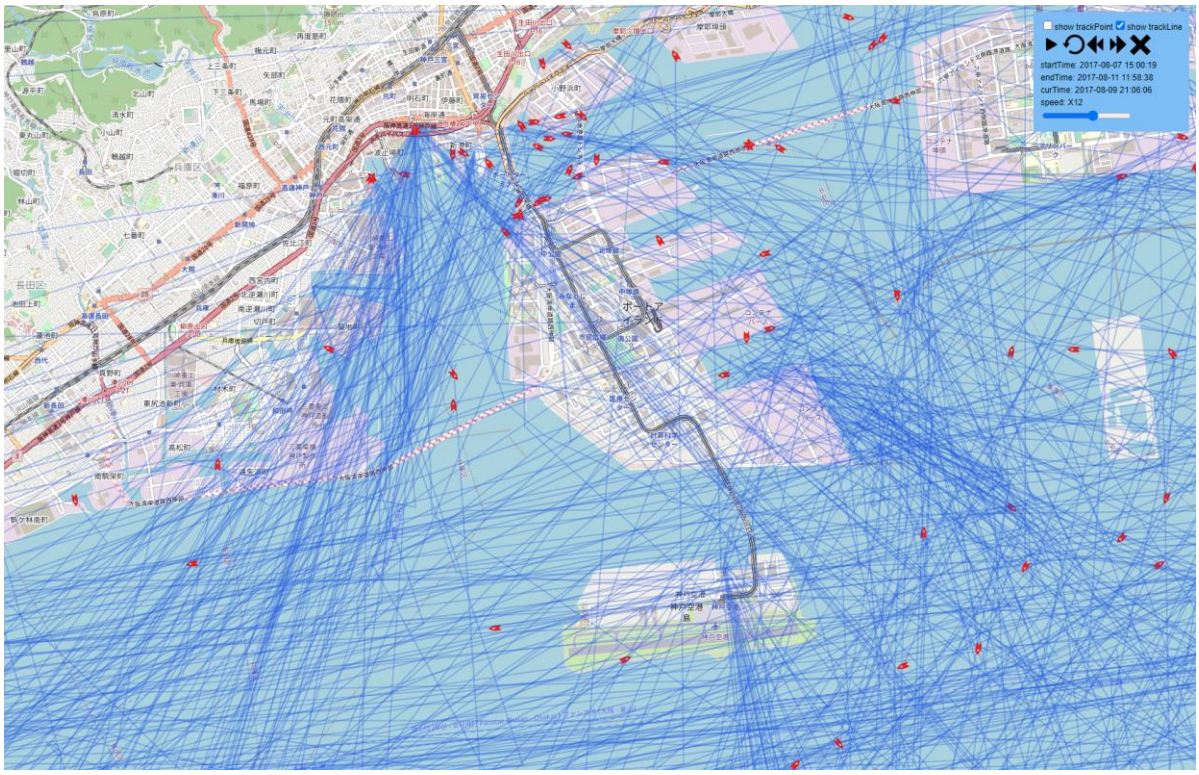
Aktuell sind die Teilnehmer lediglich ihren Parteien untergeordnet. Gemäß der echten Gliederung wäre eine weitere Hierarchiestufe in Squads sinnvoll, die wiederum ausgeblendet und eingeklappt werden können. Dazu ist zu überlegen, ob die Squads auch in der Datenstruktur eine weitere Objekteben zwischen Party und einzelnem Equipment darstellt, was eine grundlegende Überarbeitung der Programmstruktur nach sich zieht.

Icons

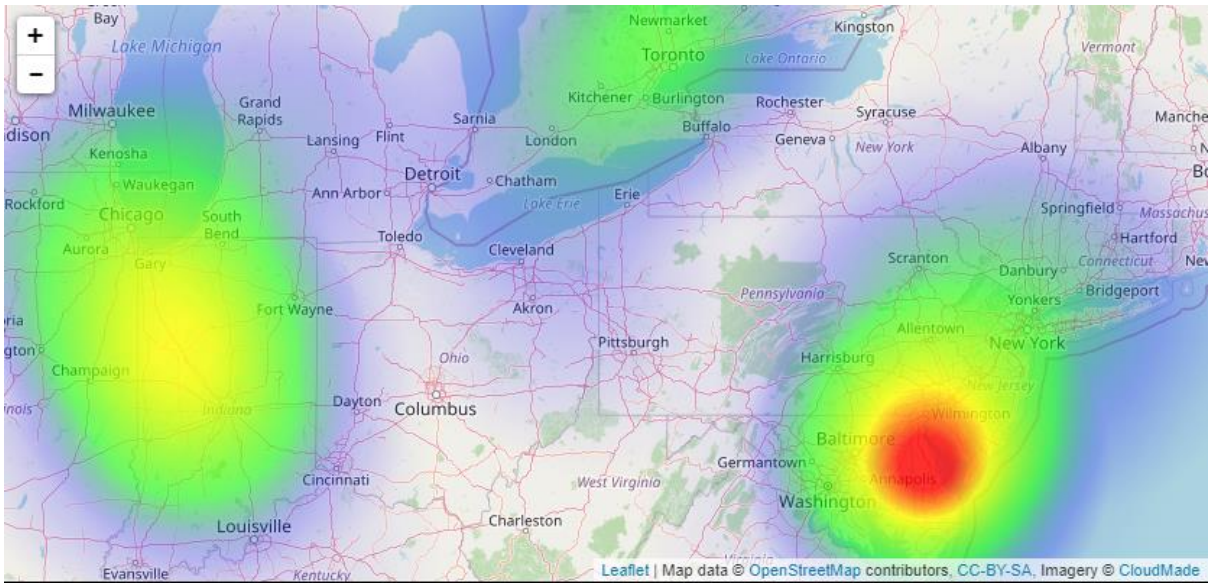
Die Marker-Icons der Teilnehmer variieren aktuell nicht nach Ausstattungstyp. Die .svg-Datei ist derzeit statischer Bestandteil des Icons in der Equipment-Klasse und wird lediglich nach Party Color eingefärbt, indem das Fill-Attribut dynamisch vergeben wird. Es ist allerdings so, auch Symbol-Informationen über den WebSocket übertragen werden und in Zukunft die Verwendung für die echten taktischen Symbole ermöglichen könnten. Die jetzigen Formate verweisen meines Wissens entweder auf ein lokales Symbolverzeichnis bzw. in Legatus eigens angelegte Symbole und lassen sich daher im Vergleich zu allgemein gültigen Farbwerten nicht übersetzen.
Gäbe es in Zukunft entweder die Option die SVGs der Symbole abzufragen oder wären diese direkt in der participantBaseData enthalten, ließen sich diese stattdessen den Markern als benutzerdefinierte Icons übergeben. So könnte die Anwendung an taktischem Tiefgang gewinnen.

Potential: Leaflet

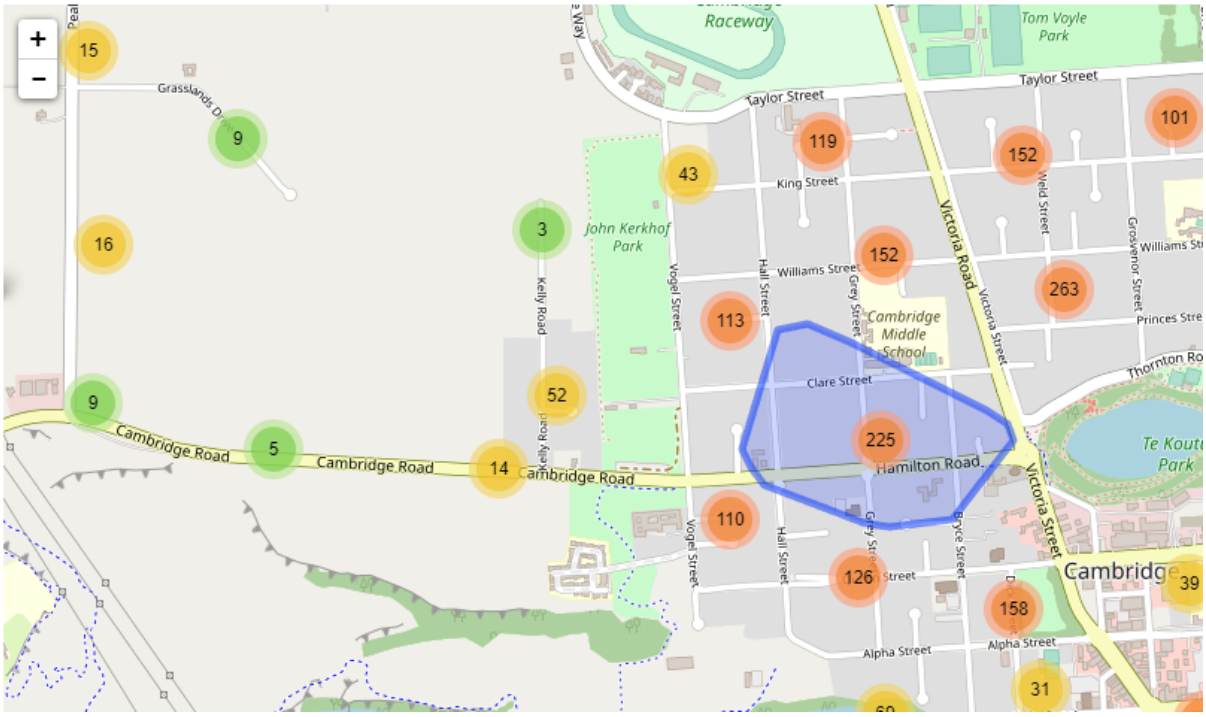
Leaflet bietet Möglichkeiten über die Live-Verortung hinaus. Im Kontext der Simulation betrifft dies das Playback oder die allgemeine Auswertung von Archivdaten. Ergänzend zur Überarbeitung der Timeline (Noah-Tool) könnten Events auch in für Zeitpunkte oder Zeiträume auf der Karte betrachtet werden. Dazu zählen Laufwege oder die räumliche Ballung von bestimmten Vorkommnissen wie Schüssen. Die folgenden Beispiele sollen einen visuellen Eindruck des Potentials vermitteln:



Playback, Live-Demo: https://linghuam.github.io/Leaflet.TrackPlayBack/



Heatmap, Live-Demo: https://www.patrick-wied.at/static/heatmapjs/example-heatmap-leaflet.html



Cluster, Live-Demo: https://jsfiddle.net/glenn/yu1a1zzo/