Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Department of Humanities, Social and Political Sciences
Prof. Dr. U. Brandes

Tracking Data and Tactical Diagrams in Football

Master Thesis

Thomas Gence

August 16, 2025

Advisors: Prof. Dr. Ulrik Brandes, Hugo Fabrègues
Department of Computer Science, ETH Zürich, ETH Zürich

Abstract

Tactics boards are widely used in football to visualize, communicate, and develop strategic ideas, from professional clubs to grassroots teams. While digital tools have become more common, existing solutions often lack both intuitive interaction and support for domain-specific features that reflect the complexity of football tactics. This thesis addresses these gaps by developing a concrete, functional prototype of a digital tactical diagram tool focused on multi-touch interaction. The design and implementation are grounded in both practical workflow needs and established visualization principles. The work identifies which critical features remain underdeveloped in current software, demonstrates what can be achieved with a user-centered design approach, and offers guidelines for future improvements in digital tactics platforms.

Acknowledgments

Firstly, I would like to thank my supervisors, Prof. Dr. Ulrik Brandes and Hugo Fabrègues, for their guidance and support throughout this research. I am equally thankful to my family and friends for their encouragement during this thesis. Special thanks also go to Lilian Bonnet and Leo Das, two students who had a project with the Social Networks Lab of ETH Zürich, for their helpful input. Lastly, I would like to acknowledge the contribution of artificial intelligence tools, which were utilized at various stages of the development and writing of this thesis.

Contents

Chapter 1

Introduction

Tactics boards have long been essential tools for visualizing and communicating football strategies at all levels of the game. Traditional magnetic whiteboards and manual sketches, while accessible and popular, are fundamentally limited: they cannot store scenarios, replay actions, scale representations, or accurately model complex tactical relationships such as formations and coordinated movements [1, 2].

In response to these limitations, digital tactical boards have emerged, ranging from commercial products [3, 4, 5] to research prototypes [6, 7]. Digital platforms now offer features such as interactive scenario editing, session storage, and, in some cases, the integration of spatio-temporal match data [8, 9]. However, most commercial tools remain focused on manual drawing and general usability, often lacking analytical or data-driven capabilities and rarely supporting advanced football-specific structures [10]. In contrast, academic systems tend to prioritize novel visualizations and analytics, but frequently overlook usability, workflow integration, and requirements for widespread practical adoption [11, 12].

Another persistent challenge is the lack of standardization for visual conventions, symbology, or interaction models in commercial and academic tools [13, 14]. This fragmentation creates inconsistencies in tactical communication, hinders comparison between approaches, and complicates the integration of advanced analysis or visualization techniques [15, 16].

This thesis addresses these gaps on two fronts. First, it surveys and critiques the current ecosystem of digital tactic boards, identifying key user needs and open problems, particularly around multitouch interaction, domain-specific representations, and exploratory workflows. Second, it presents the design and implementation of a fully functional prototype, highlighting both implemented features and those identified as essential but missing from the current landscape. The outcome is both a demonstration of current possibilities and a set of practical design guidelines to inform the next generation of digital tactical board software for football.

Chapter 2

Review of Existing Tools and Related Work

2.1 Tactics Boards and Diagram Tools

Tactics boards have long been fundamental instruments for planning and communicating football strategy. Traditional magnetic whiteboards remain common at every level of the game, but are fundamentally limited by their analog nature: they cannot store, replay, or easily share scenarios, nor support nuanced representations such as dynamic formations or coordinated team movements [1, 2]. The last decade has seen the rise of digital tactical board, both as commercial products (e.g., TacticalPad [3], Easy2Coach [4], Tactical Board [5], Tacticalista [10]) and as academic systems [6, 7, 11].

Commercial tools generally focus on enabling coaches to create, edit, and share tactical diagrams via direct manipulation. They typically support drag-and-drop player tokens, drawing tools (arrows, zones), pre-defined formations, and sometimes simple animation features (e.g., frame-by-frame movement or stepwise scenario building). However, these platforms prioritize usability over analytical depth: there is limited support for advanced data import (such as tracking data), analytical overlays, or dynamic representations beyond basic “whiteboard” metaphors. Integration with external match datasets or event data is rare, and conventions for diagram annotation vary widely between products [3, 4, 10, 17].

Academic systems take a more experimental approach, often exploring how advanced visualization or interaction techniques can support new kinds of tactical insight. For instance, Smartboard integrates LLM-based exploration to generate tactical options and scenario variations [6]; SoccerMap explores deep learning approaches for interpretable visual analysis [7]; and some tools focus on interactive dashboards for exploring spatial event data [11]. However, many research prototypes remain difficult to adopt in day-to-day coaching workflows, reflecting a broader gap between visualization research and production-quality tools, a phenomenon observed across domains [13].

Despite this diversity, there is still no widely-accepted standard for digital tactical diagramming. The space is fragmented, and user needs, especially for supporting nuanced tactical communication,are only partially met by current tools.

2.2 Visual Language and Representation

The visual encoding of tactics, how positions, movements, and relationships are represented, remains a major challenge. Unlike sports like American football, football lacks a standardized visual language for tactics diagrams. Most commercial and academic tools rely on a set of ad-hoc conventions: circles for players, arrows for movement, and lines for passes or zones, with color or shape used to distinguish teams or roles. However, symbology and encoding choices are inconsistent, both across and within tools [14]. Few platforms offer flexible or extensible visual vocabularies, and there is little consensus on how to represent tactical elements such as coordinated pressing, player roles, or interaction zones. This lack of standardization complicates the comparison of diagrams, limits interoperability, and can even hinder communication between analysts and coaches [15, 16].

The scientific literature in visual analytics and information visualization has long emphasized the importance of consistent visual encoding and semiology[14, 18, 19], yet practical adoption in football tactics software remains limited. Despite widespread use of circles, arrows, and zones, there is no domain-wide standard for football tactics symbology. Most tools adopt ad-hoc encodings, with limited agreement on how to depict roles, coordinated pressing, or interaction zones [15, 16]. Visualization theory strongly argues for consistent encodings and a clear visual grammar [14, 19, 18], yet football-specific conventions remain fragmented and largely tool-specific.

2.3 Interactive Diagram Creation and Annotation

The core function of a digital tactical board is to enable interactive creation and annotation of diagrams: defining player positions, movement, and tactical plans, ideally in a way that supports both clarity and flexibility. Commercial platforms typically enable basic interaction through drag-and-drop placement, click-to-assign roles or numbers, and manual drawing of arrows or zones. Some tools (e.g., TacticalPad, Easy2Coach) offer limited animation support via frame-by-frame scenario building, but this remains rare. Academic research explores more advanced interactive features, including scenario libraries, stepwise tactical planning, and direct manipulation interfaces tailored to the specific requirements of football [6, 11]. For example, Smartboard investigates how large language models can generate tactical scenario variations based on user input [6], and several tools experiment with gestures or contextual menus for more efficient annotation workflows.

However, interactive support for complex tactical structures, such as group behavior, dependencies, or nested formations, remains weak. Most platforms require laborious manual editing for anything beyond simple movement lines or position changes, and only a minority allow saving and reusing user-defined situations as templates or libraries. The potential for more advanced, football-specific interaction (e.g., grouped movement, dynamic role assignment) is recognized but remains underdeveloped in the current ecosystem.

Illustrative example: keyframe authoring and playback (TacticalPad). To illustrate the type of animation support available in commercial diagramming tools, Figures 2.1 and 2.2 present two common interactions in TacticalPad [3]: (i) authoring keyframes with trajectory previews, and (ii) playing the authored sequence with a single click. Similar stepwise scenario-building functions can also be found in other platforms such as Easy2Coach and Tactical Board [4, 5].

These examples show how commercial platforms typically approach animation: users manually position players across successive frames, connect them with arrows, and then preview the sequence as a simple animation. This workflow is useful for producing quick illustrative sketches, but it comes with clear limitations. Every scenario has to be created from scratch, trajectories are imagined rather than grounded in match data, and editing becomes cumbersome when dealing with more complex situations. In short, current commercial tools enable basic playback of user-authored sequences, yet remain detached from real game contexts and offer little integration with analytical data sources. This gap between illustrative animation and data-driven analysis is one of the challenges that motivates the design of more advanced systems.

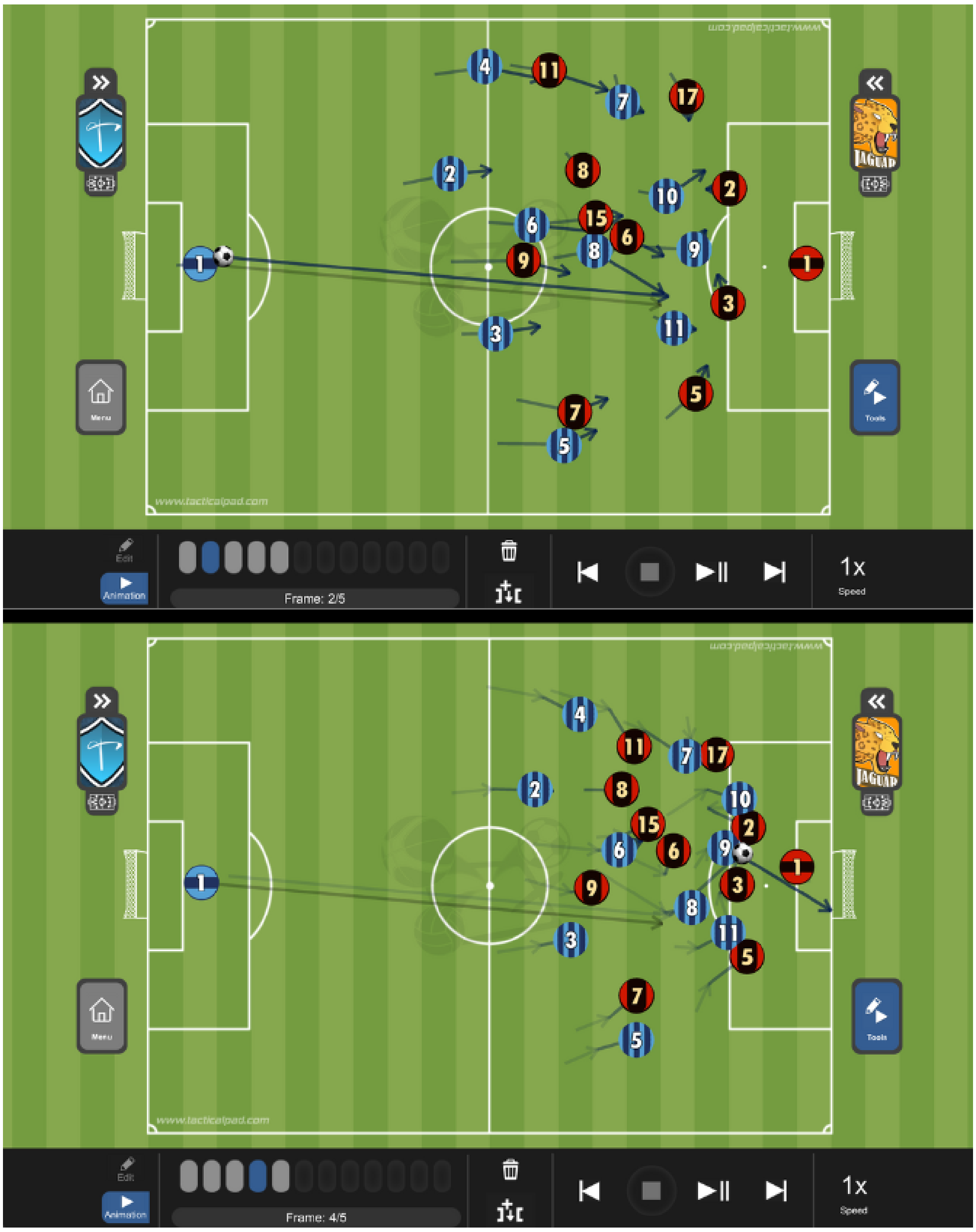


Figure 2.1: TacticalPad : keyframe authoring with trajectory previews. Each panel corresponds to a successive *frame* (label in top-left). Arrows indicate movements between frames: faded strokes represent earlier motion, while more opaque strokes denote intended upcoming motion. This progressive opacity provides a visual cue for temporal context and intent.

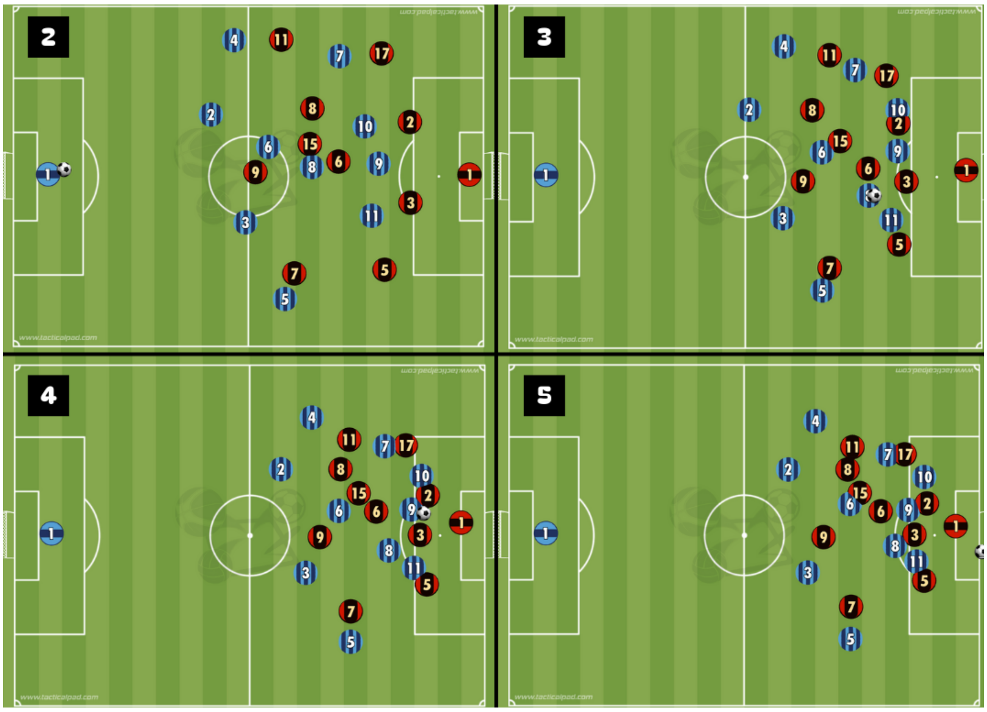


Figure 2.2: TacticalPad : one-click playback of authored sequences. After placing players and drawing arrows between frames, pressing *Play* runs the authored animation.

2.4 Integration of Match Data and Automated Diagram Generation

While some commercial platforms support basic video annotation, the integration of match data (tracking, event data) directly into tactical diagram tools is rare. Most commercial products do not support importing spatiotemporal datasets or generating diagrams from raw data [3, 4]. Instead, event and tracking data are typically visualized in specialized analytics platforms, not in tactical diagram tools [8, 20, 21, 22].

Recent academic work, however, has pushed the boundaries. Systems such as Forvizor [9], PassVizor [21], and the Sketchplan approach [12] demonstrate that interpretable diagrams and animations can be generated directly from tracking or event datasets. PassVizor, for example, analyzes pass dynamics via trajectory pattern mining and interactive visual analytics [21]. Where pass feasibility is concerned, probabilistic models of ball control ("pitch control") provide a principled basis for estimating interception risk along candidate pass paths [23]. Furthermore, these academic systems remain largely proof-of-concept, and lack user-friendly workflows or robust interfaces for practical coaching and routine analysis.

As a result, there is a clear gap between what is theoretically possible with automated or data-driven tactical diagramming, and what is implemented in mainstream coaching tools. The lack of interoperability and unified conventions continues to hinder broader adoption and integration of advanced analytics into everyday coaching workflows.

Chapter 3

System Design and Functionalities

The design of a modern tactical diagramming tool for football must reconcile two realities: the increasing availability of spatio-temporal match data, and the continued need for intuitive, expressive manual diagram creation. This chapter presents the complete set of functionalities for such a tool, structured to follow the natural workflow of a coach or analyst. The goal is not merely to list features, but to present a coherent design blueprint where each choice is deliberately justified by established principles in Human-Computer Interaction (HCI), cognitive psychology, and the state of the art in sports analytics [1, 6, 9, 16, 24].

3.1 Scenario and Team Initialization

3.1.1 Creating & Analyzing Scenarios

The first step in any tactical workflow is to establish a context. An effective tool must allow a coach to either analyze a past performance or prepare for a future one. To accommodate this duality, the system is designed to offer two primary entry points:

- **Match-Based Mode ("Tactical View"):** The user can load a complete, processed match. This mode is the foundation for reconstructive analysis, allowing a coach to review key sequences, understand tactical patterns, and prepare sessions grounded in real events. The system is designed to handle datasets containing synchronized player positions and event logs, such as the one provided by the Deutsche Fußball Liga (DFL) [8]. This dataset contain player and ball positions, speed, minute of play and synchronized event tags (passes, shots, fouls, etc...).
- **Empty Pitch Mode ("Strategic View"):** Alternatively, a user can start

with a blank canvas. This mode is essential for creative and preparatory tasks: designing drills, establishing a team's default tactical formation, or simulating "what-if" scenarios against a potential opponent.

This dual-entry approach is crucial, as coaching workflows frequently alternate between reconstructive analysis and creative planning [16].

3.1.2 Team and Formation Setup

Once in the "Strategic View," setting up the teams must be a fluid and efficient process.

- **Player Placement and Group Management:** A key design principle is to allow users to drag and drop player tokens directly onto the pitch. To support typical coaching workflows, the system is conceived to allow multi-selection of players, enabling the coach to move entire defensive lines or midfield blocks as a single unit. This aligns with the Gestalt principle of **Common Region**, where selected items are perceived as a group, simplifying complex manipulations [25].

- **Template and Playbook Library:** To accelerate setup, the tool provides a library of common tactical formations (e.g., 4-3-3, 4-4-2). A coach can load a template instantly and then make minor adjustments. Furthermore, any custom formation can be saved back into this library for future reuse. This adheres to Nielsen's fourth heuristic, **Consistency and Standards**, by providing familiar starting points [26]. The library is also designed to store entire animated plays, creating a "playbook" of set pieces or tactical drills.

- **Split-Screen Comparison:** A key feature for strategic planning is the ability to compare two tactical setups side-by-side. The split-screen mode allows a coach to display two different formations or variations of a single play simultaneously, facilitating A/B testing of tactical ideas and clarifying instructions for players.

3.2 Core Visualization and Visual Language

The visual foundation of the tool is paramount. Every design choice aims to maximize clarity and minimize the user's cognitive load, ensuring that the interface itself becomes "invisible" so the coach can focus on the tactics [27].

3.2.1 Pitch and Player Representation

The system renders a proportionally accurate pitch with all standard markings. In the "Tactical View," players are represented as abstract bicolored discs, a deliberate choice over photorealistic avatars.

- **Visual Encoding of Players:** The top half of the disc shows the shirt color, the bottom half the shorts color, and the jersey number is displayed in the center: 

  This abstract representation is effective because it removes extraneous visual details, allowing the coach and players to focus solely on positioning and structure. This is a direct application of Tufte's principle of maximizing the "data-ink ratio" by removing non-essential visual elements [18].

- **Orientation and Speed Indicator:** A small arrow is attached to each player token. Its direction indicates the player's orientation (calculated from smoothed trajectory vectors), and its length is directly proportional to their speed. This design is scientifically grounded in the theory of **pre-attentive visual processing** [19]. The human brain can decode variations in length and orientation almost instantly, without conscious effort. Displaying speed as a number (e.g., "25 km/h") would require the user to read and process each number individually, drastically increasing cognitive load and cluttering the interface. The player's number and the attached arrow rotate in unison with the player's real orientation, providing an immediate and intuitive visual cue of where the player is looking and moving.
- **Why orientation matters:** In tactical analysis, a player's body orientation is one of the most critical variables for interpreting marking, pressing, and passing options. The direction a player is facing influences their effective field of view, available passing lanes, and reaction time to pressing triggers. Encoding this visually through an arrow is significantly more efficient than relying on numerical metrics or textual labels: orientation can be decoded in less than 200 milliseconds via pre-attentive visual processing, whereas reading values takes conscious effort and interrupts tactical reasoning.

3.2.2 A Proposed Visual Language for Tactical Actions

To address the lack of standardization in football diagrams noted in the literature [13], a central goal of this design is to propose a consistent visual language. This language distinguishes between the data-driven "Tactical View" and the more abstract "Strategic View."

- **Player States in "Strategic View":** For purely hypothetical scenarios, a simplified representation is proposed to clearly distinguish roles:
  - Attacker (non-ball carrier)
  - Ball carrier
  - Defender

- **Arrow Types:** Different arrow styles correspond to specific actions, creating a clear visual grammar across all modes:
  - Pass
  - Dribble
  - Attacking Off-ball run
  - Defensive run / Pressing

3.2.3 Data-Driven Analytical Overlays

To move beyond simple description, an effective tool must help the coach to see the game's hidden dynamics. The system is therefore designed to superimpose data-driven analytical layers onto the pitch.

Pressure Visualization: The tool can visualize the defensive pressure experienced by the ball carrier. This is represented visually as a dynamic "aura" around the player. Its color and intensity change based on a real-time calculation of the proximity, speed, and angle of nearby defenders, using models inspired by recent academic work [28]. This provides an immediate, intuitive understanding of the player's available options and the stress they are under, translating complex data into a simple visual heuristic.

Why an aura is more effective than numbers: Defensive pressure is a dynamic, rapidly changing variable. Displaying it as a real-time numeric value forces the analyst to read, interpret, and mentally map it to the spatial situation — a slow, cognitively costly process. By contrast, a colored aura directly leverages the visual system's ability to process gradients and intensities instantly, making it possible to "feel" the pressure level at a glance. This mirrors the way players themselves perceive pressure on the pitch: not as a number, but as a spatial sense of opponents closing in.

Contextual Tactical Zones: For advanced analysis, the tool is designed to render real-time tactical zones that reveal hidden opportunities and risks. These overlays transform the tool from a descriptive platform ("what happened") to a prescriptive one ("what could happen"), actively guiding the coach's tactical decision-making.

- **Passing Channels:** Blocked passing options are visualized as flexible ellipses based on opponent positioning, similar to concepts of "pitch control" explored in 'PassVizor' [21]. The visualization uses a clear color code:
  - Safe channels
  - Risky channels
  - Impossible/Blocked channels

- **Shooting Zones:** A cone projected from the player indicates the areas with the highest probability of scoring. The cone's gradient from green (high probability) to red (low probability) is envisioned to be calculated in real time based on the player's distance to goal, shooting angle, and the positioning of the goalkeeper and nearby defenders. This design is inspired by spatial shot effectiveness patterns identified in academic analyses such as [29], which highlight how these variables influence scoring likelihood in match situations.

3.3 Temporal Navigation and View Management

3.3.1 The Timeline System

The timeline is the central hub for temporal control, designed for both high-level Browse and micro-analysis. It provides a powerful workflow for navigating to key moments, as illustrated in the figures that follow.

- **Slider and Event Markers:** The main slider allows for fluid, frame-by-frame scrubbing through the match. Initially, key match events are all displayed as clickable icons on the timeline. This design respects **Jakob's Law**, as users are already familiar with this interaction pattern from standard video editing software, reducing the learning curve [30].

- **Event Filtering:** In a typical match, the sheer number of events can lead to a cluttered and overwhelming timeline. To address this, the tool provides an event filter panel. A user can select specific event types (e.g., ”Goal”), and the timeline instantly updates to show only those relevant markers. This is a direct application of **Hick’s Law**; by drastically reducing the number of choices presented to the user, the system makes it significantly faster and easier to find important moments, thus reducing cognitive strain [31].

- **Direct Navigation and Micro-Zoom:** Clicking on a specific event marker on the timeline serves two functions simultaneously. First, it instantly navigates the main view to that precise frame in the match. Second, it displays a zoomed-in, secondary timeline focused on the moments immediately preceding and following the selected event. This micro-navigation view is crucial for detailed analysis of the build-up and consequences of a key action, providing context without requiring the user to manually scrub back and forth.

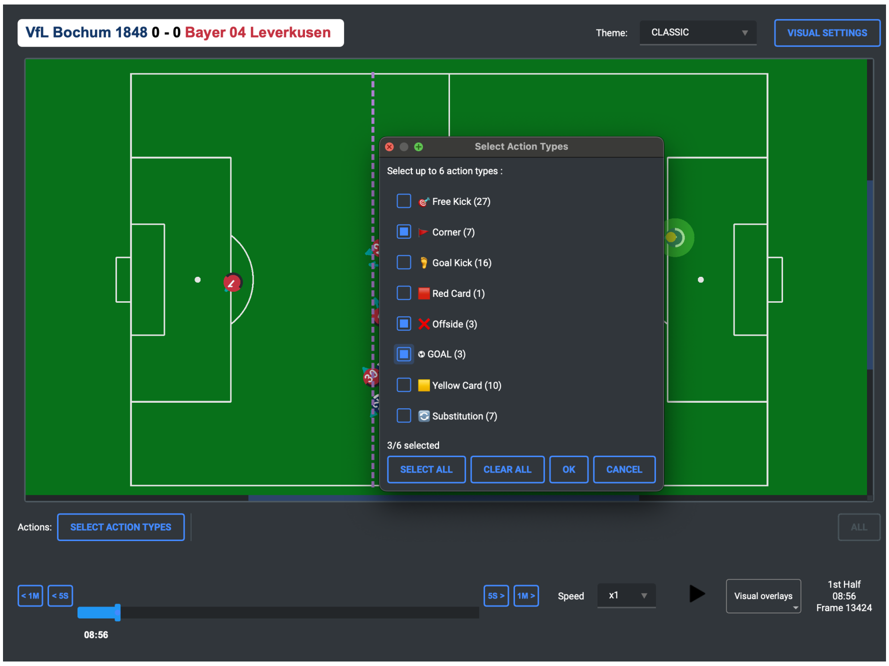


Figure 3.1: Step 1 - Filtering : Filtering what type of events we want to have above the timeline.

Figure 3.2: Step 2 - Pre-selection : Events are filtered and now the user chooses what action to select.

Figure 3.3: Step 3 - Selection : When clicking on a specific event (here, "Blue team scores"), we zoom in so that the user can better see the other events around this goal. After clicking we are projected at the goal moment.

3.3.2 Camera Management

The camera system is designed to focus attention effectively, providing a flexible viewport onto the tactical canvas. A dedicated control panel offers a suite of ten specialized buttons, each addressing a specific analytical need. The panel is organized into three logical groups: manual controls, automated views, and presets for key zones.

- **Manual Controls:** The default view shows the entire pitch. The user has direct control via three essential buttons: standard '+' and '-' buttons allow for manual zoom, while a 'Reset' button instantly restores the full-pitch view. These controls are designed to be large and immediately accessible, respecting **Fitts' Law** to minimize interaction time and effort [32].

- **Automated and Intelligent Views:** To reduce the manual burden on the user, two intelligent modes are provided:
  - **Ball Tracking ('BALL'):** A dedicated mode that automatically centers the camera on the ball and follows its movement, freeing

the coach to focus on the tactical context rather than manual navigation.

  - **Context-Aware Auto-Zoom:** The tool is designed with an intelligent camera that adapts to the play. It can automatically zoom in on the ball carrier during close control and then zoom out to a wider view when a long pass is anticipated. This reduces the need for constant manual adjustment, lowering the user's extraneous cognitive load.

- **Preset Views for Set Pieces and Key Zones:** To accelerate the analysis of recurring situations, the system includes six preset camera views that instantly focus on critical areas of the pitch:

  - **Corners ('TLC', 'TRC', 'BLC', 'BRC'):** These four buttons instantly focus the camera on the top-left, top-right, bottom-left, and bottom-right corners, essential for detailed analysis of offensive and defensive corner-kick organizations.

  - **Penalty Areas ('LP', 'RP'):** These two buttons zoom in on the left and right penalty areas, crucial for analyzing dangerous attacking situations, finishing, or defensive organization in the final third.

  This set of dedicated presets acts as a series of "accelerators" in the sense of Nielsen's heuristics, allowing expert users to navigate to critical zones with a single click instead of multiple zoom and pan operations [26].

3.4 Tactical Annotation and Simulation

A central component of the tool is its ability to translate tactical ideas into clear, structured visual annotations. This functionality serves two purposes: it enables coaches to convey concepts rapidly to players, and it provides a visual record that can be revisited, adjusted, and compared over time.

3.4.1 Annotation Tools

The annotation system is deliberately designed to be both expressive and precise. It offers a range of tools that balance the creative freedom needed for tactical ideation with the clarity required for effective communication.

Cognitive efficiency of visual annotations: Each annotation type (arrow, zone, label) is designed to carry meaning through shape, style, and position without requiring the viewer to consult a legend constantly. This reduces working memory load and speeds up comprehension during live or fast-paced tactical reviews.

- **Arrow Creation and Properties:** Beyond simply drawing arrows, the user can customize a comprehensive set of properties:
  - *Color and Style:* Passes, dribbles, and runs follow the visual grammar established earlier (solid, dashed, wavy lines), but users can override these defaults to emphasize specific tactical ideas. Color selection is offered via a perceptually uniform palette to ensure high contrast and accessibility, particularly in projected presentations.
  - *Thickness and Curvature:* Line thickness can encode the importance or intensity of an action (e.g., a high-speed pass rendered thicker). Curvature can be adjusted to reflect realistic ball trajectories or to depict off-ball runs that bend around defenders.
  - *Arrowheads and Endpoints:* Multiple arrowhead styles are available (classic, triangular, open), enabling coaches to visually distinguish between action types or player intentions.

  These properties are not merely aesthetic: they serve as cognitive cues, allowing players to interpret a diagram faster and with less ambiguity. By encoding action type, importance, and trajectory directly into the shape and style of an arrow, the tool exploits **pre-attentive visual attributes** [19], reducing the need for explanatory text.

- **Zone Creation and Properties:** Zones can be drawn as rectangles, circles, ellipses, or free-form polygons. Each zone supports:
  - Fill color and opacity (to avoid obscuring underlying data).
  - Border style and thickness.
  - Optional labels for naming tactical areas (e.g., “pressing trap zone”, “shooting pocket”).

  This flexibility supports both schematic training diagrams and precise, data-driven overlays.

- **Post-Creation Editing:** All annotations are fully editable after creation. Users can drag endpoints, rotate arrows, adjust curvature, or recolor elements in real-time. A live preview ensures that changes are immediately visible.

- **Advanced Interaction Features:** To support complex tactical drawings:

  - *Multi-Selection & Grouping:* Multiple annotations can be selected and moved or modified together, useful when adjusting an entire defensive line or a coordinated pressing pattern.
  - *Layer Management:* Annotations are organized into layers, which can be shown or hidden. This allows, for example, a coach to toggle between offensive and defensive phases without redrawing.
  - *Annotation Presets:* Frequently used styles (e.g., a standard "pressing arrow") can be saved as presets for rapid reuse, accelerating workflow in live sessions.
  - *Layer Visibility Toggles:* Each annotation layer (e.g., `Offense Layer`, `Defense Layer`, `Set Pieces Layer`) can be individually toggled on or off via a dedicated "eye" icon in the `Layers Panel`. This allows the coach to instantly switch between viewing only offensive annotations, only defensive ones, or any desired combination. The visibility state is updated in real time without reloading the scene. This selective display is particularly valuable when illustrating different phases of play: hiding non-relevant layers keeps the visual scene clean and prevents cognitive overload, avoiding the "visual noise" problem that occurs when all tactical information is displayed simultaneously.

- **Undo/Redo & Cognitive Safety Nets:** A robust undo/redo system encourages exploration by removing the fear of making irreversible mistakes, aligning with the principle of **User Control and Freedom** [26].
- **Grouped Deletion Tools:** Instead of relying solely on "Clear All," the tool supports deletion by annotation group. In the `Layers Panel`, a coach can select an entire layer (e.g., all pressing arrows, or all set-piece zones) and remove it with a single click. This prevents accidental loss of unrelated work and allows for fast iteration during tactical discussions: an entire scenario can be reworked without having to manually erase each element.

3.4.2 Simulation Mode

The simulation mode extends the tool beyond static diagrams, enabling dynamic "what-if" analyses. Its design choices reflect the need to experiment with tactical variations without overwhelming the user.

- **Loop-Based Scenario Editing:** A coach selects a time interval (e.g., five seconds before a goal), which then plays in a continuous loop. This fixed temporal context supports repeated observation and modification without constant manual navigation.

- **From Annotation to Action:** In simulation mode, attaching an arrow to a player overrides their recorded trajectory. The player then moves according to the drawn instruction, and the resulting new play unfolds on screen. This **direct manipulation** approach [33] creates an immediate cause-effect link between coach input and visual output, strengthening tactical understanding.
- **Proportional Timing:** The time taken to execute a simulated action scales with the length of the drawn arrow, providing an intuitive, non-verbal way to encode tempo and intensity.
- **Trajectory Fading for Clarity:** Both real and simulated trajectories gradually fade as the play progresses, instead of remaining fully visible. This is a deliberate decision grounded in cognitive load theory: displaying all past paths simultaneously produces excessive visual clutter (often called "spaghetti diagrams" in sports analytics). The fading preserves short-term context—so the user can still see the immediate origin of a movement—while keeping the scene readable. This design follows Tufte's principle of reducing non-essential "data-ink" [18] and reduces cognitive load by emphasizing the latest movements.
- **Branching Scenarios:** The system supports multiple alternative simulations from the same starting point. A coach can compare Variant A (e.g., pass to left wing) against Variant B (pass into the box) without overwriting the original data.
- **Trigger Points:** Simulated actions can be set to trigger only when specific conditions are met (e.g., a forward run begins only once a midfielder enters a certain zone), allowing for more realistic tactical modeling.
- **Export and Communication:** Simulations can be recorded and exported as short video clips, facilitating clear communication to players or inclusion in scouting reports.

Action model driven by annotations (common to Tactical & Strategic views). Beyond simple motion overrides, the simulation treats an annotated arrow as a typed action that binds one or two players to a football event (e.g., *pass*, *shot*, *dribble*, *tackle*). In Tactical View (match data), the user may slightly reposition players and draw a typed arrow; this produces a *user-authored action* that (i) overrides trajectories locally for playback and (ii) prepares an updated event entry aligned to the edited frames. In Strategic View (blank pitch), the same authoring pattern creates hypothetical actions with the exact same schema. This unified interaction keeps the mental model identical across views: draw → bind player(s) → choose action type → simulate.

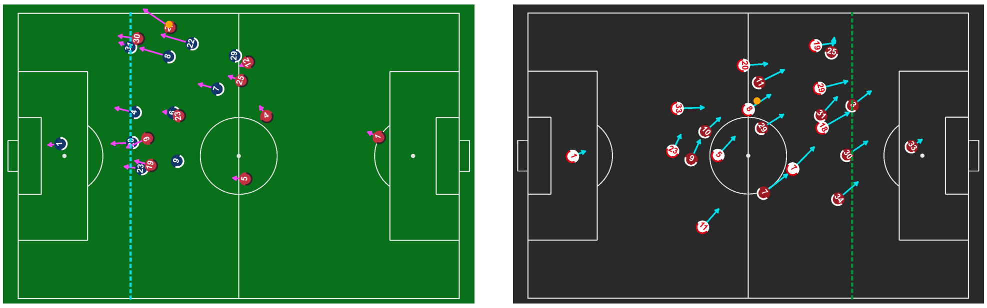
In designing the simulation system, the guiding principle has been to make complex tactical ideas **fast to draw, easy to read, and faithful to the intended meaning**, while minimizing the risk of misinterpretation. The trajectory fading mechanism, in particular, illustrates how a small visual design choice can have a large impact on the usability of a sports analysis tool.

3.5 Customization and Project Management

Finally, the tool provides options for personalization and saving work.

3.5.1 Interface Customization

A well-designed tactical tool must be visually clear in a variety of contexts, from a high-resolution monitor in an analyst's office to a low-quality projector in a team meeting room. To address this, the tool's design incorporates a robust system for interface customization, centered on perceptually effective visual themes.



(a) The "Classic" visual theme. **(b)** The "Black & White" theme.

Figure 3.4: The two primary visual themes available in the application.

As introduced earlier with the orientation arrows, our colour choices are optimised only for certain visual overlays rather than for the entire scene. In practice, this means focusing on the offside line and player-orientation arrows, while keeping the pitch, pitch lines, and team kits fixed to match the real match or training context. The reason is simple: these overlays carry critical analytical information and must remain easy to read regardless of the colours on the pitch. If, for example, the offside line or arrows were too close in colour to the grass or a team's kit, they would blend into the background and lose their function. By optimising only overlays, we target the elements where colour choice has the greatest impact on tactical clarity, without altering the recognisable appearance of the game.

- **Visual Themes:** As shown in Figure 3.4, the design includes two primary themes. The rationale for this is not merely aesthetic. Professional team kits often feature colors that have poor contrast against each other or against the green of the pitch, which can make player identification difficult. This is a critical issue for coaches who need immediate clarity, and especially for users with color vision deficiencies.

  To solve this, the themes are generated by an algorithm whose goal is to maximize legibility. The system is designed to operate in a **perceptually uniform color space**, which models human color vision far more accurately than standard screen-based color models like RGB [19]. By optimizing for the highest possible perceived difference between colors, the themes guarantee strong contrast and accessibility.

- **User Control:** While the system provides an algorithmically optimized default, the principle of **User Control and Freedom** is paramount [26]. The user retains final control and can manually override the theme's colors or change the global scale of all visual elements on the pitch via a dedicated settings panel. This allows for adaptation to personal preferences, specific presentation needs, or unexpected lighting conditions.

3.5.2 Saving and Exporting

- **Saving and Loading:** The system allows for saving the full project state, including loaded data, all annotations, camera positions, and filter settings. In "Strategic Mode," coaches can also save specific team line-ups as reusable templates.
- **Exporting (Envisioned):** For communication, the ability to export is crucial. The tool is envisioned to export annotated frames as static images (PNG, PDF) and animated sequences as videos (MP4), a need consistently highlighted in surveys of professional coaching workflows [34].

Chapter 4

Implementation

This chapter explains how the features introduced in Chapter 3 were implemented in the working prototype. It focuses on the **technical execution**: data ingestion, transformation, scene rendering, algorithm design, and integration into an interactive application.

Programming environment and dependencies The implementation targets **Python 3.12.2** and relies on a combination of general-purpose and domain-specific libraries:

- **PyQt6** for the GUI (widgets and `QGraphicsScene` used to render the pitch, players, overlays, and the timeline).
- **qt_material** to style Qt widgets (buttons, sliders, menus). This only affects the GUI chrome and is *independent* of the in-pitch color logic handled by the internal `ThemeManager`.
- **floodlight** to parse and read the dataset.
- **NumPy** for efficient array operations on frame-indexed positional data.
- **SciPy** for selected numerical routines (e.g., temporal smoothing of orientations).
- **colormath** for perceptual color conversions (CIELAB/LCH) and **CIEDE2000** contrast computations in the theme optimization.

4.1 System Architecture Overview

The application follows a **modular manager-based architecture**. Each functional domain is encapsulated in a dedicated manager, instantiated and wired by `MainWindow` at startup. The GUI is implemented with **PyQt6**.

- **MainWindow** – Loads match data, instantiates all managers, and coordinates their interactions through Qt's signal/slot system.
- **PitchWidget** – Owns the `QGraphicsScene` where static pitch lines, dynamic player/ball icons, and overlays are rendered.
- **Timeline** – Custom slider widget showing key events and supporting zoom to segments.
- **Managers:**
  - `ThemeManager` – Perceptual theme generation and contrast adjustments (using **colormath** for CIELAB/LCH transforms and CIEDE2000).
  - `CameraManager` – View position/zoom (manual, ball-following, presets).
  - `TrajectoryManager` – Drawing of real and simulated trajectories with temporal fading.
  - `ScoreManager` – Match score/time display.
  - `ArrowAnnotationManager` - Manages creation, preview, selection, and editing of arrows (straight, dotted, zigzag, curved)
  - `RectangleZoneManager` - Manages rectangular zones: create/select/move/resize/rotate, with stroke/fill/opacity/dash styling
  - `EllipseZoneManager` - Same as `RectangleZoneManager` but for ellipse zones
  - `TacticalSimulationManager` – Overrides tracked movements with user-defined paths in looped intervals.
  - `SettingsManager` – Stores UI preferences (visible overlays, active theme).

4.2 Data Handling Pipeline

All match data is loaded with the **floodlight** library from **Floodlight XML files** containing *tracking* (player/ball positions per frame) and *events* (passes, shots, goals, etc.). The code does not manually parse XML for positions/events: floodlight provides frame-synchronized structures that the application consumes directly. For auxiliary *DSAM* fields (distance, speed, acceleration, minute) that are not exposed by floodlight's high-level objects, the loader falls back to `xml.etree.ElementTree` to extract them from the same XML feed (see `data_processing.py`).

4.2.1 Parsing and indexing

- The dataset, once read via floodlight, exposes synchronized, frame-indexed arrays for positions (players and ball). These arrays are the basis for rendering and for all analytical overlays.
- Event timestamps are projected to the nearest frame index within a tolerance so that (i) the **timeline icons** and direct navigation remain *frame-accurate* (see Subsection 4.5.2 and the timeline design in Chapter 3), and (ii) overlays recompute on the exact animation frame that the user jumps to (see Section 4.4). (Procedure in Appendix A.4, *Event Projection to Frames*.)

4.2.2 Derived quantities and where they are used

Beyond raw positions, some quantities are computed on the fly (NumPy/SciPy) and then consumed by specific UI/analytics modules:

- **Orientation vectors** — Computed from frame-to-frame player displacements and converted to angles with $\arctan 2$. To avoid wrap-around artefacts, the angle series is mapped to its cosine/sine components, each component is smoothed (Savitzky–Golay filter [35]), then the angle is reconstructed via $\arctan 2$. (see Appendix A.3)

  Used for: Drawing per-player direction/speed arrows; pressure computation (projection of defender speed/acceleration toward the carrier).

- **Ball carrier identification** – Inferred as the nearest eligible player to the ball within a 3.5 m threshold. This avoids flagging carriers while the ball is traveling (e.g., long passes).

  Used for: pressure visualization (aura centered on the carrier) and contextual overlays (e.g., offside line toggling).

4.3 Rendering Cycle

Rendering uses a layered model in `QGraphicsScene`:

1. **Static layer** – Pitch lines, goals, constant elements (cached).
2. **Dynamic layer** – Player and ball icons updated every frame (positions, rotations/orientations).
3. **Overlay layer** – Tactical visualizations (pressure auras, passing channels, annotations), recomputed when inputs change.

Per-frame draw cycle:

1. Update positions and rotations (orientation vectors) for players and ball.
2. Recompute active overlays from the latest frame data.
3. Apply the current theme to updated items.
4. Redraw only layers that changed since the previous frame.

4.4 Key Algorithms

4.4.1 Perceptually-Driven Visual Theme Generation

Goal Ensure strong legibility regardless of team kits and the color of the pitch.

Implementation The theme optimizer searches locally in LCH space for a palette that yields the best worst-case visibility across all critical pairs (teams vs pitch, teams vs each other, lines vs pitch), with a secondary preference for higher overall average contrast. Candidate colors are first filtered to exclude those too close in perceptual distance, insufficiently separated in hue, or lacking minimum luminance contrast against key backgrounds. The optimisation score is a weighted combination of worst-case and average contrast, ensuring that no critical pair is too close while favouring palettes with strong global visibility. The exact candidate filtering logic and the scoring procedure are detailed in Appendix A.1, where step-by-step pseudocode mirrors the algorithm used in the prototype to select perceptually distinct colors.

4.4.2 Pressure Computation

Goal Estimate the defensive pressure on the ball carrier at each frame and provide a scalar intensity in $[0, 1]$ suitable for visualization.

Implementation For every defender, the algorithm computes a time-to-intercept (TTI) relative to the ball carrier. This estimate accounts for distance, speed, acceleration, and current orientation. Each TTI is mapped to a pressing probability through a logistic function, and the contributions of all defenders are aggregated to yield a single intensity value. The aggregation assumes independence between defenders for tractability, which simplifies the model but does not capture potential coordination effects. The output is a smooth, interpretable signal that updates frame by frame. Whereas Bekkers et al. [28] compute pressing intensity from tracking-based kinematic models using time-to-intercept metrics and logistic mapping, our approach adopts a simplified kinematic approximation that omits dependencies between defenders.

By projecting velocity and acceleration onto the line to the carrier, the model captures directionality: defenders moving away from the carrier contribute little or no pressure. The logistic mapping ensures gradual transitions, while the aggregation naturally accounts for multiple defenders closing in simultaneously. The full pseudocode is given in Appendix A.2.

4.5 Implementation of Other Functionalities

4.5.1 Annotation Tools

- **Arrows** – Straight/dashed/curved with adjustable thickness, curvature, arrowheads, and color; styles encode semantics (e.g., pass, dribble, run).
- **Zones** – Rectangles, circles, ellipses, and polygons with configurable fill/opacity/border, color, and optional labels.
- **Editing** – Post-creation drag/rotate/reshape; multi-selection and layer management; presets for frequently used styles.
- **Undo/Redo** – Allows the user to revert or reapply the most recent edits (e.g., position, size, or color changes).

4.5.2 Timeline and Event Filtering

- The custom timeline shows key events; clicking an icon seeks the animation to the corresponding frame (see Section 4.2).
- Event-type filters focus the timeline on intervals where the selected actions occur (zoom-to-type), reducing clutter.
- Selecting a specific event also shows a local micro-zoom window centered on that event to inspect its immediate context.

4.5.3 Camera Control

- **Manual** – Pan/zoom via mouse/shortcuts; reset to full-pitch.
- **Ball tracking** – Keeps the ball centered while play unfolds.
- **Context-aware auto-zoom** – Zoom adapts to local density/spread.
- **Presets** – One-click views for corners (TLC/TRC/BLC/BRC) and penalty areas (LP/RP).

4.5.4 Trajectories and Simulation

- In **Simulation Mode**, real trajectories from the tracking data are rendered with **temporal fading** to prevent the visual overload commonly known as "spaghetti clutter" (see Figure 4.1). When full trails of player movements are rendered simultaneously, the resulting tangle of lines quickly becomes unreadable. To maintain clarity, our approach displays only the most relevant segments: past positions are progressively erased as the player advances, while upcoming positions are shown with decreasing opacity the further they lie in the future.
- To keep rendering efficient, trajectories are **sampled and cached** per (frame, interval) pair. This prevents unnecessary recomputation during playback or looping and ensures fluid interaction even when horizons span several seconds.
- In the same mode, the user can already **draw arrows as annotations** and assign them an *action type* (pass, shot, dribble, tackle, . . .), binding them to one or more players. At this stage these arrows remain descriptive only: they do not yet alter the simulation or update the event log. In §5.1.1, we discuss how these user-authored arrows will be extended to produce meaningful modifications of the underlying data.

Figure 4.1: Trajectories with temporal fading in Simulation Mode. Past movements gradually fade out while future projections are displayed with decreasing opacity.

Chapter 5

Future Work

While the current prototype demonstrates the feasibility of a modular, data-driven tactical diagramming tool, several functionalities identified during the design phase remain to be implemented. Addressing these gaps would significantly extend the analytical and pedagogical capabilities of the system.

5.1 Missing Functionalities from the Strategic/Tactical Views

5.1.1 User-Authored Event Editing from Annotations

In **Simulation Mode**, users can draw arrows directly on the pitch to represent intended actions. These arrows can already be typed (pass, shot, dribble, tackle, etc.) and bound to one or two players through their properties. At present, however, they remain purely descriptive: they do not permanently alter the underlying tracking or event data. Instead, any modification of player positions or actions is applied only locally and temporarily for the duration of the simulation, leaving the raw data untouched.

The planned extension is to link these user-authored arrows to the underlying data model so that authored actions (a) alter trajectories in the simulation itself and (b) yield a proposed update in the event log aligned to the edited frames. In **Strategic View** (blank pitch), the exact same authoring pattern would then create entirely new hypothetical events.

Scope and behavior (planned).

- **Action typing:** each user-drawn arrow carries an action label from a fixed set $\{$*pass*, *shot*, *dribble*, *tackle*, $\ldots\}$.
- **Actor binding:** one origin player is required; a target player is required for passes and optional otherwise.

- **Light coordinate edits:** in match mode, small drags of player markers are allowed to reflect "what-if" micro-adjustments; these edits are scoped to the selected interval.
- **Event update:** saving the authored action would insert or amend the corresponding event in the timeline (time range derived from the arrow length and playback tempo). Edited markers would be visually outlined and filterable.
- **Reversibility:** the edit could be reviewed and reverted from the inspector without touching the raw tracking file.

Next steps. Stabilize the action schema (type, actor(s), frame range, geometry), harmonize commit/revert flows in both views, and finalize timeline regeneration so authored events remain frame-accurate while preserving source data integrity.

5.1.2 Strategic View Implementation

The current version supports data-driven Tactical View but lacks a fully interactive Strategic View for hypothetical scenario building. **Implementation path:**

- Integrate the existing annotation and simulation modules into a "blank-pitch" mode (Strategic View), where the pitch is initially empty and the user can manually place players, draw tactical movements, and simulate hypothetical scenarios without loading match data. This mode would reuse the same modular components used in the Tactical View (e.g., `PitchWidget`, annotation managers, camera controls) but initialised with an empty state, ensuring code reuse and minimising additional complexity. From an analytical standpoint, this would also allow coaches to design and test set-piece variations or pressing schemes in isolation before cross-referencing them with match data.
- Allow full manual player placement, with multi-selection and grouped movement.
- Provide a formation library and "playbook" for reusable tactical sequences.

5.1.3 Split-Screen Comparison

The envisioned ability to compare two tactical setups or scenarios side-by-side is not yet implemented. **Implementation path:**

- Instantiate two `PitchWidget` instances in a synchronized container.

- Implement linked timelines and camera controls so that both views remain temporally aligned.
- Support drag-and-drop of annotations between the two panes.

5.1.4 Custom Composition Creation

The current prototype can display formations from match data but does not allow saving fully custom line-ups. **Implementation path:**

- Extend the project save format to include arbitrary player roles, positions, and kit designs.
- Implement a formation editor with role naming, role–position binding, and export to the formation library.

5.2 Analytical Extensions

5.2.1 Passing Channel Feasibility Prediction

While the UI concept is defined, the system currently lacks an algorithm to categorise passing options. The proposed approach builds on the *pitch control* framework introduced by Spearman [23], which models the probability of each team controlling the ball at a given location.

Visual analytics systems such as PassVizor [21] have shown the value of interactive exploration of passing patterns, but also highlight a key limitation: they provide little modelling of the opposing team and thus cannot fully assess whether a passing lane is realistically viable. Our contribution addresses this gap by incorporating explicit measures of opponent influence—through pitch control and interception likelihood—into the evaluation of passing channel feasibility.

Proposed approach:

- **Input:** Current positions and velocities of all players, ball carrier ID, target player ID.
- **Feature extraction:**
  - *Line-of-sight occlusion* — detect if any opponent's intercept path crosses the straight line from passer to receiver.
  - *Pitch control model* — compute each team's probability of controlling the ball at multiple points along the passing path using the probabilistic approach of Spearman [23].
  - *Receiver context* — distance to nearest defender at estimated reception time, receiver's orientation relative to ball.

- **Classification:**
  - Train a logistic regression or gradient-boosted tree on historical labelled passes (easy = completed with no pressure; risky = contested or forced back; impossible = intercepted).
  - Output category and associated confidence.
- **Visualization:**
  - Green solid line for "Easy", yellow dashed line for "Risky", red crossed line for "Impossible".
  - Optional hover tooltip with contributing factors (e.g., "High interception risk: defender 2.3 m away at reception").

These features are updated in real time from the tracking stream and evaluated only for plausible targets—filtered by passer orientation, line-of-sight, and distance thresholds—to preserve UI responsiveness.

5.2.2 Enhanced Shooting Zone Prediction

Current cones are static in shape and rely on distance/angle heuristics. **Implementation path:**

- Incorporate data-driven expected goals (xG) models conditioned on shot context.
- Vary cone width and gradient dynamically based on xG distribution.

5.2.3 Context-Aware Auto-Zoom Enhancements

The current auto-zoom considers ball location and player spread. We propose adding simple triggers that adapt the zoom and framing to the tactical context, while keeping transitions smooth and unobtrusive.

Implementation path:

- **Wing switch (long diagonal):** When the ball's x-coordinate crosses the pitch midline at high horizontal speed and most teammates are on the far side, smoothly zoom out, pan, and zoom in toward the new attacking side.
- **Shot likelihood (final third focus):** If the ball carrier is close to goal and facing it, zoom in slightly and frame the view ahead of the carrier to capture the attacking movement.
- **Crossing context:** When the ball is near the sideline in the attacking third, adjust the framing to include the penalty area and far-post lane.

- **Fast transition:** If the ball speed is high or possession changes in midfield, briefly zoom out to show the new team shape, then return to the normal view.
- **Set-piece presets:** If basic event tags are available (corner, free kick), move to a pre-defined camera position suited to that situation.

All changes should transition smoothly, avoiding sudden jumps, and include short cooldowns to prevent constant zoom adjustments.

Chapter 6

Conclusion

This work introduced an interactive tactical analysis tool for football that merges the robustness of professional tracking data processing with the creativity of dynamic visualization and annotation. The design integrates essential, intuitively usable features, such as smooth navigation, precise event synchronization, and adaptable overlays, with more innovative concepts, including perceptually optimized themes and real-time pressure visualization inspired by advanced sports analytics research.

The goal throughout has been to combine reliability, clarity, and efficiency, while leaving space for exploratory and creative tactical experimentation. By grounding the interaction model in established usability principles: consistency, feedback, and minimal cognitive load—the prototype ensures that advanced analytical capabilities remain accessible even under real-time constraints.

Beyond its current implementation, the system offers a strong foundation for the integration of more ambitious functionalities, such as predictive passing feasibility, multi-view comparison modes, and composition planning. As such, it not only delivers a practically usable prototype, but also points toward future directions for football analysis tools that are both analytically rigorous and intuitively engaging.

Appendix A

Appendix

A.1 Pseudocode for Theme Optimization

Inputs

mode One of `CLASSIC` or `BLACK&WHITE`.

teamColors Home and away main/secondary kit colors.

ballColor Color used to render the ball.

Fixed elements Pitch and pitch-line colors are taken from the selected mode (classic green with white lines, or a black&white preset adapted to overall kit brightness) and are not optimised.

Outputs

theme Dictionary with keys `grass`, `line`, `offside`, `orientation_arrow` where `offside`, `orientation_arrow` are optimized colors.

Algorithm 1 Overlay-Focused Theme Generation

Require: mode, teamColors, ballColor

$(grass, line) \leftarrow \textsc{PresetFromMode}(mode,\ teamColors)$ ▷ fixed by preset

$F \leftarrow \{grass, line, ballColor\} \cup teamColors$ ▷ forbidden set

$offside \leftarrow \textsc{PickColor}(F)$

$F \leftarrow F \cup \{offside\}$ ▷ keep overlays distinct

$orientation\_arrow \leftarrow \textsc{PickColor}(F)$

return $\{\texttt{grass} : grass,\ \texttt{line} : line,\ \texttt{offside} : offside,\ \texttt{orientation\_arrow} : orientation\_arrow\}$

Algorithm 2 PICKCOLOR(F, $grass$)

$C \leftarrow$ GENERATECANDIDATESLCH() ▷ compact LCH grid: varied hues, a few lightness/chroma tiers
$C \leftarrow \{c \in C \mid \text{INGAMUT}(c)\}$ ▷ candidate is representable in sRGB without clipping
$C \leftarrow \{c \in C \mid \text{PERCEPTUALLYDISTINCT}(c,\ F)\}$ ▷ reject if perceptual distance to any forbidden colour is too small
$C \leftarrow \{c \in C \mid \text{HUESEPARATED}(c,\ F)\}$ ▷ avoid near-family hue clashes
$C \leftarrow \{c \in C \mid \text{VISIBLEAGAINST}(c,\ grass)\}$ ▷ ensure overlay legibility against the pitch
if $C = \varnothing$ **then**
    return FALLBACK()
end if
$best \leftarrow$ **None**; $bestScore \leftarrow -\infty$
for $c \in C$ **do**
    $min \leftarrow \min\big(\text{VISIBILITY}(c,\ t) \text{ for all } t \in F\big)$ ▷ worst-case contrast
    $avg \leftarrow \text{average}\big(\text{VISIBILITY}(c,\ t) \text{ for all } t \in F\big)$ ▷ overall contrast
    $score \leftarrow min + 0.2 \cdot avg$
    if $score > bestScore$ **then**
        $bestScore \leftarrow score$; $best \leftarrow c$
    end if
end for
return $best$

Notes on implementation Conversions between sRGB, CIELAB, and LCH as well as perceptual distance and contrast computations are handled by utility functions (see `utils/color_utils.py`) .

A.2 Pseudocode for Pressure Visualization

Inputs (frame t)

carrierId ID of the ball carrier.

defenders List/Set of opponent player IDs to evaluate.

positions Map or arrays giving (x, y) for every player at frame t.

speeds Arrays giving speed S (m/s) for every player at frame t.

accelerations Arrays giving acceleration magnitude A (m/s^2) for every player at frame t.

orientations Arrays giving orientation θ (radians) for every player at frame t.

t_{thresh} Time horizon for "pressable" situations (default 1.2 s).

σ Logistic width (default 0.5 s).

Output

pressureIntensity Scalar in $[0, 1]$ for the frame.

Algorithm 3 PRESSUREINTENSITY (frame-level aggregation)

Require: `carrierId`, `defenders`, `positions`, `speeds`, `accelerations`, `orientations`, t_{thresh}, σ

$(p_x, p_y) \leftarrow$ `positions`[`carrierId`]
$P_{\text{none}} \leftarrow 1$ ▷ probability that nobody is pressing
for all $id \in$ `defenders` **do**
  $(d,\ \mathbf{u},\ v_0,\ a_{\|}) \leftarrow$ PROJECTKINEMATICS$(id,\ (p_x, p_y)$, `positions`, `speeds`, `accelerations`, `orientations`$)$
  $t \leftarrow \text{SOLVETTI}(d,\ v_0,\ a_{\|})$
  $p_i \leftarrow \text{Logistic}\big((t_{\text{thresh}} - t)/\sigma\big)$
  $P_{\text{none}} \leftarrow P_{\text{none}} \cdot (1 - p_i)$
end for
return `pressureIntensity` $\leftarrow \text{clip}\big(1 - P_{\text{none}},\ 0,\ 1\big)$

Algorithm 4 PROJECTKINEMATICS$(id,\ (p_x, p_y),\ positions,\ speeds,\ accelerations,\ orientations)$

$(x, y) \leftarrow positions[id]$; $S \leftarrow speeds[id]$; $A \leftarrow accelerations[id]$; $\theta \leftarrow orientations[id]$
if any of (x, y, S, A, θ) is missing **then**
  continue
end if
$(dx, dy) \leftarrow (p_x - x,\ p_y - y)$; $d \leftarrow \text{HYPOT}(dx, dy)$
$\mathbf{u} \leftarrow (dx/d,\ dy/d)$ ▷ unit direction defender→carrier
$\mathbf{h} \leftarrow (\cos\theta,\ \sin\theta)$ ▷ heading unit vector
$v_0 \leftarrow S\ (\mathbf{h}\cdot\mathbf{u})$; $a_{\|} \leftarrow A\ (\mathbf{h}\cdot\mathbf{u})$ ▷ signed components toward carrier
return $(d,\ \mathbf{u},\ v_0,\ a_{\|})$

Algorithm 5 SolveTTI$(d,\ v_0,\ a_{\|})$

if $d = 0$ **then**
    return $t \leftarrow 0$
end if
if $|a_{\|}| < 10^{-9}$ **then** ▷ constant-speed closure
    return $t \leftarrow d / \max(v_0,\ 10^{-6})$
else ▷ constant-acceleration closure along the line of sight
    $\Delta \leftarrow v_0^2 + 2\, a_{\|}\, d$
    if $\Delta \geq 0$ **then**
        return $t \leftarrow \dfrac{-v_0 + \sqrt{\Delta}}{a_{\|}}$ ▷ physically meaningful root
    else
        return $t \leftarrow d / \max(|v_0|,\ 10^{-6})$ ▷ fallback when no real solution
    end if
end if

Rendering When the ball is active at the global frame, a single translucent disc (fixed radius) is drawn *centered on the carrier*. Its fill color is obtained from `get_pressure_color` (see data_processing.py); only color encodes intensity (radius and opacity are kept constant to avoid overlapping between visual elements on the pitch).

A.3 Orientation Estimation (from tracking positions)

Inputs

$\mathbf{pos}(\boldsymbol{id}, \boldsymbol{t})$ **:** Player 2D position at frame t.

$\boldsymbol{w}$ **:** Window length for smoothing (frames).

$\boldsymbol{p}$ **:** Polynomial order for Savitzky-Golay filter.

Output $\boldsymbol{orientation}(\boldsymbol{id}, \boldsymbol{t})$: Orientation of player `id` at frame `t`.

Algorithm 6 Orientation Estimation from Frame-to-Frame Displacements

Require: $\mathrm{pos}(id, t)$ for all t; smoothing window w; polynomial order p

$\Delta x_t \leftarrow x(id, t) - x(id, t-1)$; $\quad \Delta y_t \leftarrow y(id, t) - y(id, t-1)$

$\theta_t \leftarrow \arctan 2(\Delta y_t,\ \Delta x_t)$ ▷ raw orientation series

$c_t \leftarrow \cos(\theta_t)$; $\quad s_t \leftarrow \sin(\theta_t)$ ▷ avoid wrap-around artifacts

if length of series $\geq w$ **then**

  $c^{\mathrm{smooth}} \leftarrow \mathrm{SavitzkyGolay}(c,\ w,\ p)$

  $s^{\mathrm{smooth}} \leftarrow \mathrm{SavitzkyGolay}(s,\ w,\ p)$

  $\theta_t^{\mathrm{smooth}} \leftarrow \arctan 2\big(s_t^{\mathrm{smooth}},\ c_t^{\mathrm{smooth}}\big)$

else

  $\theta_t^{\mathrm{smooth}} \leftarrow \theta_t$

end if

return $\theta_t^{\mathrm{smooth}}$

A.4 Event Projection to Frames

Inputs

$\{\tau_i\}$: **list(float)** Event timestamps (s).

$\{t_j\}$: **list(float)** Frame times (s).

ε : **float** Matching tolerance (s).

Output Mapping event $i \rightarrow$ frame $j^\star$.

Algorithm 7 Event Projection to Frames

Require: eventTimestamps $\{\tau_i\}$, frameTimes $\{t_j\}$, tolerance ε

for all τ_i **do**

  $j^\star \leftarrow \arg\min_j |t_j - \tau_i|$

  if $|t_{j^\star} - \tau_i| \leq \varepsilon$ **then**

    attach event i to frame $j^\star$

  end if

end for

Bibliography

[1] E. Morgulev, O. H. Azar, and R. Lidor, Sports analytics and the big-data era, *International Journal of Data Science and Analytics*, vol. 5, no. 4, pp. 213–222, 2018.

[2] F. Vidal-Codina, N. Evans, B. El Fakir, and J. Billingham, Automatic event detection in football using tracking data, *Sports Engineering*, vol. 25, no. 1, p. 18, 2022.

[3] TacticalPad. URL : https://www.tacticalpad.com/new/.

[4] Easy2Coach: Online Soccer Tactic Board. https://www.easy2coach.net/en/online-soccer-tactic-board/.

[5] Tactical Board. URL : https://tactical-board.com/uk/big-football.

[6] Z. Liu *et al.*, Smartboard: Visual exploration of team tactics with llm agent, *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[7] J. Fernández and L. Bornn, Soccermap: A deep learning architecture for visually-interpretable analysis in soccer, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2020, pp. 491–506.

[8] M. Bassek, R. Rein, H. Weber, and D. Memmert, An integrated dataset of spatiotemporal and event data in elite soccer, *Scientific Data*, vol. 12, no. 1, p. 195, 2025.

[9] Y. Wu *et al.*, Forvizor: Visualizing spatio-temporal team formations in soccer, *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 65–75, 2018.

[10] Tacticalista. URL : https://tacticalista.com/en/.

[11] E. Delibas, A. Uzun, M. F. Inan, O. Guzey, and A. Cakmak, Interactive exploratory soccer data analytics, *INFOR: Information Systems and Operational Research*, vol. 57, no. 2, pp. 141–164, 2019.

[12] D. Seebacher, T. Polk, H. Janetzko, D. A. Keim, T. Schreck, and M. Stein, Investigating the sketchplan: A novel way of identifying tactical behavior in massive soccer datasets, *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 4, pp. 1920–1936, 2021.

[13] G. Walsh, N. S. Andersen, N. Stoianov, and S. Jänicke, The lack of specialized symbology and visual interaction design guidance for subsea military operations, in *VisGap-The Gap between Visualization Research and Visualization Software*, The Eurographics Association, 2023, pp. 19–26.

[14] J. Bertin and W. Berg, Semiology of Graphics: Diagrams, Networks, Maps. ESRI Press, 2011, ISBN: 9781589482616.

[15] C. Perin, R. Vuillemot, and J.-D. Fekete, Soccerstories: A kick-off for visual soccer analysis, *IEEE transactions on visualization and computer graphics*, vol. 19, no. 12, pp. 2506–2515, 2013.

[16] H. Sarmento, R. Marcelino, M. T. Anguera, J. CampaniÇo, N. Matos, and J. C. LeitÃo, Match analysis in football: A systematic review, *Journal of sports sciences*, vol. 32, no. 20, pp. 1831–1843, 2014.

[17] Philippka Graphic Tool. URL : https://ft-graphics.fussballtraining.com/.

[18] E. R. Tufte, Envisioning information, *Optometry and Vision Science*, vol. 68, no. 4, pp. 322–324, 1991.

[19] C. Ware, Information Visualization: Perception for Design (Information Visualization: Perception for Design). Elsevier Science, 2013, ISBN: 9780123814647.

[20] R. Metulini, Spatio-temporal movements in team sports: A visualization approach using motion charts, *arXiv preprint arXiv:1611.09158*, 2016.

[21] X. Xie *et al.*, Passvizor: Toward better understanding of the dynamics of soccer passes, *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 1322–1331, 2020.

[22] A. Cortez, B. Gonçalves, J. Brito, and H. Folgado, Improving event data in football matches: A case study model for synchronizing passing events with positional data, *Applied Sciences*, vol. 15, no. 15, 2025, ISSN: 2076-3417. DOI: 10.3390/app15158694.

[23] W. Spearman, A. Basye, G. Dick, R. Hotovy, and P. Pop, Physics-based modeling of pass probabilities in soccer, in *Proceeding of the 11th MIT Sloan Sports Analytics Conference*, Boston, MA, vol. 1, 2017.

[24] T. Munzner, Visualization analysis and design. CRC press, 2014.

[25] Gestalt Principles. URL : https://www.gestaltprinciples.com/.

[26] J. Nielsen, 10 usability heuristics for user interface design, 1994.

[27] S. Krug, Don't Make Me Think: A Common Sense Approach to the Web (2nd Edition). New Riders Publishing, 2005, ISBN: 0321344758.

[28] J. Bekkers, Pressing intensity: An intuitive measure for pressing in soccer, *arXiv preprint arXiv:2501.04712*, 2024.

[29] M. Perez-Arroniz, J. Calleja-González, J. Zabala-Lili, A. Crespo, and A. Zubillaga, Shooting and goalkeepers response analysis in a professional football league, *Apunts Sports Medicine*, vol. 60, no. 226, p. 100 480, 2025, ISSN: 2666-5069.

[30] Jakob's Law. URL : https://lawsofux.com/jakobs-law/.

[31] Hick's Law. URL : https://lawsofux.com/hicks-law/.

[32] Fitt's Law. URL : https://lawsofux.com/fittss-law/.

[33] B. Shneiderman, Direct manipulation: A step beyond programming languages, *Computer*, vol. 16, no. 8, pp. 57–69, 1983.

[34] L. Lolli *et al.*, Data analytics in the football industry: A survey investigating operational frameworks and practices in professional clubs and national federations from around the world, *Science and Medicine in Football*, vol. 9, no. 2, pp. 189–198, 2025.

[35] R. W. Schafer, What is a savitzky-golay filter?[lecture notes], *IEEE Signal processing magazine*, vol. 28, no. 4, pp. 111–117, 2011.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every written paper or thesis authored during the course of studies. **In consultation with the supervisor**, one of the following two options must be selected:

☐ I hereby declare that I authored the work in question independently, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used no generative artificial intelligence technologies[1].

☒ I hereby declare that I authored the work in question independently. In doing so I only used the authorised aids, which included suggestions from the supervisor regarding language and content and generative artificial intelligence technologies. The use of the latter and the respective source declarations proceeded in consultation with the supervisor.

Title of paper or thesis:

Tracking Data and Tactical Diagrams in Football

Authored by:
If the work was compiled in a group, the names of all authors are required.

Last name(s):	**First name(s):**
Gence	Thomas

With my signature I confirm the following:

- I have adhered to the rules set out in the Citation Guidelines.
- I have documented all methods, data and processes truthfully and fully.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for originality.

Place, date	**Signature(s)**
Zürich, 16.08.2025	Gence

If the work was compiled in a group, the names of all authors are required. Through their signatures they vouch jointly for the entire content of the written work.

[1] For further information please consult the ETH Zurich websites, e.g. https://ethz.ch/en/the-eth-zurich/education/ai-in-education.html and https://library.ethz.ch/en/researching-and-publishing/scientific-writing-at-eth-zurich.html (subject to change).