ChatGPT

Project Specification: Formula 1 Management Simulation Game (Browser-Based F1 Manager)

Introduction

This project is a browser-based, single-player Formula 1 management simulation game inspired by the **F1 Manager** series. It aims to let players step into the role of a team principal, managing all aspects of an F1 team. The application will use a **React + TypeScript** frontend for a dynamic, responsive UI, and a **Node.js** backend to handle game logic, data management, and simulations. A strong emphasis is placed on clean and sophisticated **UI/UX**, ensuring the interface is intuitive and visually appealing. All game features will be introduced in stages, with each version building on the last. This iterative approach allows starting from a minimal viable product and progressively adding depth and complexity over multiple versions. Each version described below includes the features of previous versions plus the new enhancements.

Real-World Data Integration and Sources

To enhance realism, the game will incorporate real-world data for drivers, teams, and circuits. We will leverage public data APIs and datasets to source this information:

- **Ergast F1 API / Jolpica** – For historical and current Formula One data, we can use the Ergast Developer API, which provides a comprehensive record of F1 results, driver details, constructors (teams), and circuits [1] [2]. (Note: Ergast is being succeeded by the Jolpica API beyond 2024 [3], which can serve as a drop-in replacement.) These APIs give access to driver names, nationalities, team names, circuit info, and even race results, which can populate the game's database.
- **Circuit Data Repositories** – For track maps and details, community-maintained datasets are available. For example, there is a GitHub repository that provides all F1 circuits in JSON/GeoJSON format (including locations and layouts) [4]. This can support the "live map" race visualization by supplying coordinates or outlines of each track.
- **Wikipedia/Wikidata** – As a supplementary source, wiki data can be used to fetch additional info like circuit lengths, driver birthdays/ages, etc., if needed for flavor. These would be processed into the game's database for offline use.
- **OpenF1 (Real-time Data)** – Optionally, for the latest seasons or more detailed live data, the open-source OpenF1 API provides real-time telemetry and timing information [5]. While not necessary for early versions, this could enable advanced features like live timing charts in later versions.

Using these sources ensures the game starts with authentic 2023–2025 season data (drivers, teams, tracks) and can be updated easily each season. The data will be cached or imported into the Node backend (e.g., as JSON or in a database) to avoid heavy external calls during gameplay. With real names and stats in place, players get an immersive experience managing familiar F1 personalities and venues.

Version 1: Basic Team Management & Season Simulation (MVP)

Overview: The first version is a minimal viable product focusing on core team management and a basic season simulation. It establishes the foundation of the game: the player takes control of one of the official F1 teams (or a fictional team if licensing is an issue) and plays through a season. The emphasis is on simplicity and playability, ensuring that this initial build is feasible and functional. The UI will be clean and straightforward, presenting essential information like team roster, races, and standings without overwhelming detail.

Features in Version 1:
- *Team Selection & Roster:* At game start, the player selects a team to manage (from 10 real-world F1 constructors, e.g., Ferrari, Mercedes, etc.). Each team comes with two drivers and basic car performance stats reflecting their real-world competitiveness. Driver and team data are loaded from the real-world dataset (e.g., 2024 season driver lineup). For the MVP, driver abilities are simplified to an overall rating that influences race results.
- *Calendar and Race Simulation:* The game includes a simplified race calendar (e.g., 16–22 Grand Prix events). For each Grand Prix, a **simple race simulation** generates results based on team and driver ratings plus a bit of randomness (to allow upsets). There is no user intervention during races in this version – the outcome is auto-simulated when the player proceeds to the next race. Results could be determined by a weighted random algorithm where higher-rated teams finish near the front. The race result is presented as a text/table showing finishing positions and points scored.
- *Standings & Progress:* After each race, the championship standings (drivers' and constructors' points tables) update. The UI should provide a standings screen so the player can track their progress through the season. The points system follows the real F1 format (e.g., 25-18-15… points). At season end, the game could declare a championship result and perhaps allow restarting or continuing to a new season (in later versions, continuing will be more meaningful as features like R&D and transfers come in).
- *Basic Financial Overview:* Although detailed finances come later, Version 1 can include a very basic budget just for context. Each team starts with a set budget (not heavily used yet). The UI might show the team's current balance and a basic income/expense overview (e.g., income from prize money after each race, expenses like a flat operational cost per race). This is mostly to lay groundwork for future finance features. The board's season objective (e.g., "finish 5th or higher in the Constructors' Championship") might be shown, but consequences of failing it are minimal in MVP.
- *User Interface (UI/UX):* Focus on a **clean layout** – e.g., a dashboard screen with your team info (drivers, car performance rating, budget), a calendar or "Next Race" button, and a standings screen. Use a modern, responsive design (perhaps a component library like Material UI or a custom design system) to ensure it's visually appealing. Even at this early stage, the UI should reflect an F1 theme (team colors, minimalist modern fonts, etc.) to engage the player. All interactions (like advancing to the next race) should be obvious and easy.

Rationale: Version 1 is kept minimal – essentially a single-season simulator with limited interaction – to ensure it's achievable. It establishes the data structures and flow of the game (teams, drivers, races, results) which future versions will expand upon. By including only light simulation and management, we can build a stable base and then progressively layer on complexity.

Version 2: Driver Market & Contracts

Overview: Version 2 introduces dynamic team management through a **driver market** with contract negotiations and transfers. In addition to managing races, the player can now interact with the driver roster over multiple seasons. This adds depth to team management: drivers can be hired, fired, or have contracts renewed, and their morale or patience can become factors. The game evolves from a one-season simulator into a multi-season career where off-track decisions matter. The UI/UX will expand to include screens for scouting and contract negotiation, while still remaining user-friendly and clear.

New Features in Version 2:
- *Multi-Season Continuity:* Instead of ending after one season, the game now supports continuing into subsequent seasons. This persistence allows the effects of management decisions (like who your drivers are) to carry over. Real-world data (from the sources above) can seed the initial season's driver roster, and fictional young drivers or retirements can be introduced in future seasons to keep the world dynamic.
- *Driver Contracts:* Every driver has a contract with a team, including a duration (number of seasons) and salary. The game will track contract end dates. At season end (or during a specified transfer window), drivers with expiring contracts become available to sign. The player can also attempt to buy out contracts for drivers currently in other teams (with a financial penalty) if they want to sign them early.
- *Negotiation System:* Signing a driver is not automatic – the player must negotiate terms. Each driver has expectations (salary level, contract length, role, etc.) influenced by their skill and the prestige of the team. For example, a world-class driver will demand a high salary and might only join a lower team if sufficiently paid or promised the #1 driver role. In F1 Manager 2024, contract offers have an *appeal meter* that shows how interested the driver is [6] – we can implement a simpler version of this. The negotiation UI lets the player set terms: e.g., contract length (1-3 years), annual salary, a signing bonus, and the driver's status (first or second driver). Based on these, a driver will either accept, reject, or make a counteroffer. A happy driver (coming from a smaller team or without a seat) will be more patient and willing to negotiate, whereas a top driver may reject low offers outright [6]. If the player low-balls an offer, the driver might get "insulted" and end talks, simulating the patience mechanic (in F1 Manager, a rejected offer shrinks the appeal bar making the next offer harder to succeed) [7].
- *Driver Market UI:* A new **"Driver Market" or "Scouting" screen** is added. Here, all drivers (and possibly F2/F3 drivers or free agents, if data allows) are listed with key info (age, overall rating, current team, contract expiry). The player can filter and select a driver to see details and an option to approach for a contract. Drivers who are not interested (e.g., currently under contract with top teams) will be indicated as such or have a low chance to sign. This screen will use real data for known drivers and can generate fictional stats for new entrants as seasons progress.
- *Transfers & Team Changes:* After negotiation, if a deal is struck, the driver will join the team in the next season (or immediately if it's a mid-season transfer and rules allow). The game should also simulate AI team transfers – other teams will also sign drivers when they have vacancies. Over multiple seasons, this creates a dynamic grid where drivers switch teams, new rookies appear, and veterans retire. This adds a strategic layer: the player must plan driver line-ups for the long term.
- *Contract Management:* The player also has to manage their own drivers' happiness. A driver who is consistently treated as "second driver" (team orders always against them) or paid low might have lower morale and could refuse to renew a contract. Conversely, a satisfied driver (good results, fair team orders, decent pay) will be more willing to stay. (We will formalize morale and driver happiness in a later version with a "staff morale/mentality" system, but even in Version 2 we can simulate it simply via negotiation difficulty).
- *Basic Staff Roles:* (Optional for this version) If desired, we can include the concept of hiring technical staff

(like a Technical Director or Race Engineer) using a similar contract system. However, to keep scope small, Version 2 might stick strictly to drivers. Staff management can be a future addition beyond our listed scope.
- *UI/UX Considerations:* The negotiation interface should be **clear and interactive**. For example, when offering a contract, use sliders or input fields for salary and contract length, and visually indicate the driver's interest (e.g., a progress bar or color-coded response, similar to the appeal meter in F1 Manager [6]). Provide feedback like "Driver X has made a counteroffer" or "Driver X is unwilling to negotiate further." The driver market screen can have profile popups showing a driver's stats and contract status. Throughout these screens, maintain a polished look consistent with the F1 theme, using team colors or driver portraits if available. The complexity of multiple screens is growing, so ensuring a **consistent navigation** (using React Router for different views like Dashboard, Driver Market, Standings, etc.) is important.

Rationale: Adding the driver market and contracts in Version 2 dramatically increases player engagement by allowing long-term planning and team building. It turns the game from a static season simulator into a living career. This sets the stage for subsequent features (like car development and finances) because now that multiple seasons are in play, managing resources and planning for future seasons becomes meaningful.

Version 3: Car Research & Development (R&D) System

Overview: Version 3 introduces a Car R&D feature, allowing the player to develop their car over time. This adds a technical management aspect: allocating resources to improve the car's performance. Each car is composed of various components (chassis, engine, aerodynamics, etc.), and the player can initiate development projects to design or upgrade parts. The focus is on balancing performance gains with time and cost. The game will model how car performance evolves, which directly affects race results in the simulation. The UI will get a new **Car Development** section with interactive elements for designing parts and viewing performance stats.

New Features in Version 3:
- *Car Performance Attributes:* Define a set of performance attributes for cars. For example, attributes could include **Engine Power**, **Aerodynamic Downforce**, **Drag Efficiency**, **Reliability**, **Tire Wear**, etc. These can be simplified into a few key stats (e.g., straight-line speed, cornering ability, fuel efficiency, reliability). In reality, F1 cars have many nuanced stats (F1 Manager 2024 uses 12 attributes feeding into 10 performance stats [8] [9]), but for the game we might start with ~5–6 core stats to keep it manageable (we can expand later). Each team's car starts with base values for these stats (reflecting real-world 2024 performance – e.g., Team A has the highest engine power, Team B has the best downforce, etc.). These stats influence the race simulation outcomes (higher stats = faster lap times, better reliability = fewer failures, etc.).
- *R&D Projects:* The player can initiate R&D projects to improve the car. A project could be *designing a new part* (e.g., a new front wing to improve downforce) or *researching next year's car*. Each project will have a **cost (in money)** and a **time to complete** (e.g., number of races or in-game weeks). For instance, designing a new aerodynamic part might take 4 weeks and $1 million. Once completed, the new part yields an improvement in certain attributes. The player can then *manufacture* and fit that part to the car. All teams (AI-controlled teams as well) will be doing R&D, so if the player doesn't develop, they may fall behind as other teams improve.
- *Development UI:* A dedicated **Car Development** screen is added. This screen shows current car performance stats and allows starting new projects. For example, it could list part categories (Engine, Aero, Chassis, etc.) – clicking one opens options like "Improve Engine Power (minor gain, short time)" or "Design new Front Wing (major gain in downforce, longer time)". When a project is running, a progress bar or timer

displays how many weeks until completion. Multiple projects could be allowed in parallel if resources suffice (e.g., one aerodynamic and one chassis project concurrently). However, to introduce strategy, we might limit the number of simultaneous projects based on staff or facility level (which can be upgraded in future versions).
- *Resource Points or Staff:* Optionally introduce the concept of **engineering team capacity**. For instance, allocate "engineers" or "wind tunnel time" to projects. A simplified approach is to use *R&D points* that accumulate each race (similar to how some games have "research points"). The player spends these points (plus money) to start projects. This ensures that the progression is paced (can't max out car immediately).
- *In-Season vs Off-Season Development:* The player can decide whether to focus on the current car or start researching next year's car (especially useful if current season is lost cause). For simplicity, this can be abstracted: any R&D done late in the season can either apply immediate upgrades or be directed to next year's design (improving the base stats for the next season's car). This mirrors real F1, where teams balance in-season upgrades and next-year's car development [10].
- *Regulation Changes:* (Optional stretch) Not in this prompt's list, but to mention: for realism, future versions could include rule changes that reset or alter certain performance aspects (e.g., an aerodynamic rules reset might diminish everyone's aero stats). We note this as a future consideration beyond Version 3.
- *Impact on Races:* With R&D, car performance is now dynamic. The race simulation logic will factor in the updated car stats: e.g., if Team X significantly improved engine power, their cars will have higher top speeds and likely better results on power-sensitive tracks (if track data includes attributes, or we can define track types like "high-speed" vs "twisty" to influence how stats translate to performance). This adds strategic depth – a player might develop specifically to gain an edge at upcoming tracks.
- *Cost Cap and Budget (Preparation):* All projects cost money, and this version ties into finances. In real F1, a cost cap limits how much can be spent on car development [11]. We will soon introduce a budget management system (in Version 4), but even here we can hint at a cost cap: for example, prevent the player from spending beyond a certain threshold on R&D in a season. This encourages efficient spending.
- *UI/UX Considerations:* The Car Development interface should be **visual and informative**. For instance, show the car outline with part icons (engine, wings, etc.) highlighting which areas are weak or strong. Use charts or bars to compare current car stats to the grid average or to the best team (e.g., a bar chart showing your aerodynamic level vs the best, to know where you stand). This comparison gives context for the player's decisions [12] [13]. When starting a project, present a confirmation modal summarizing cost, time, and expected gain (e.g., "New Front Wing: +5% High-speed Downforce, completion in 3 races, cost $2M"). Upon completion, notify the player (could be via an in-game email system or notification banner). Maintain the UI theme with technical styling (perhaps blueprint-like visuals to represent engineering). The UX must simplify complex data – using tooltips or info icons to explain what each stat means will help players manage R&D without confusion.

Citations: In F1 Manager 2024, each car consists of 9 parts (6 aero parts, 3 powertrain) which collectively determine 12 performance attributes like various downforce levels, drag, engine power, etc [14] [8]. Development projects require balancing performance gains against time and cost, since all projects take time and money, and you must *"strike the balance between a cutting-edge car… and delivering parts at the right moment"* [11]. The game will emulate this by forcing the player to plan upgrades wisely under resource constraints.

Rationale: The Car R&D system is a major enhancement that introduces an engineering challenge to the game. It ensures that success is not only about having the best drivers, but also about developing the best car. This feature also creates a sink for the budget (coming in Version 4) and ties into sponsor goals (coming in Version 5, e.g., a sponsor might demand a certain level of car development). Implementing R&D now sets the stage for a richer simulation where the player feels the technological race that parallels the on-track race.

Version 4: Team Finances and Budget Management

Overview: Now that the game features driver contracts and R&D projects, managing money becomes critical. Version 4 introduces a detailed financial system, including budgets, income, and expenses. The player must keep the team financially healthy, balancing spending on car development, driver salaries, staff, and facilities (if any), against income from sponsors, prize money, and other sources. Financial management will enforce tough choices – you can't buy every upgrade or hire all the best staff without running out of money. This version will also incorporate **sponsors and their contracts** (as sponsors are a primary income source in motorsport). The UI will have a **Finance Dashboard** and a **Sponsors screen**, giving clear visibility into the team's monetary situation. *(We combine the finances and sponsor features in this version, as they are closely linked.)*

New Features in Version 4:
- *Detailed Budget & Cost Cap:* Each team has a yearly budget (e.g., starting budget provided by team owners or previous season's savings). Income and expenses are tracked per race and per season. A **cost cap** rule can be implemented – for example, a spending limit (say $135M per year, matching F1 regulations) that teams cannot exceed on certain expenditures. The Finance system will alert the player if they are nearing the cap.
- *Income Sources:* The game will simulate multiple income streams [15] :
- **Sponsor Income:** Money from sponsors (details below in Sponsor Management). This includes upfront signing bonuses and periodic payments.
- **Prize Money:** At season end (and possibly some per-race bonuses), teams receive prize money based on their finishing position in the Constructors' Championship. This reflects real F1 where higher-placed teams get more funds for next year. The game can use a sliding scale or preset values for 1st through 10th place.
- **Budget from Owners/Chairman:** Some games give a fixed amount each season from the team owner [15] (especially for smaller teams). We can include a pre-season lump sum or monthly stipend from a "Chairman". For instance, a top team might not get much extra (expected to be self-sufficient), but a backmarker might get a boost to help them improve.
- **Other:** If we have *pay drivers* (drivers who bring sponsorship), that could contribute (e.g., a driver with high marketability might bring additional funds [16]). Also, if facilities produce income (like a team museum for visitors), but that might be beyond scope for now.
- *Expense Categories:* The game will track major expenses [17] :
- **Driver and Staff Salaries:** Deduct per race or per month from budget. When signing a driver, their negotiated annual salary is now taken from the budget across the season. Similarly, if we have key staff like Technical Director, their salaries count.
- **R&D and Car Part Costs:** All those projects from Version 3 consume funds. Designing/manufacturing new parts, as well as yearly design of a new chassis, have costs [17] . The finance system will log these expenses when projects are started or completed.
- **Facilities and Operations:** If we consider facilities (like HQ buildings), upgrades to them cost money too [18] . In this version, we might introduce a simple facility upgrade system (e.g., upgrade factory to allow 2 projects at once, cost $5M). Even if not, we should account for general operational costs – e.g., a fixed cost per race for logistics, or per season for engine supply (in F1, customer teams pay for engines).
- **Miscellaneous:** Other expenses could include pit crew salaries, travel costs, etc., but these can be

abstracted as a fixed “race weekend cost” so the player knows each race deducts, say, $500k for running the team.

- *Sponsor Management:* Sponsors are companies that fund the team in exchange for exposure. In this game, sponsors will be a key part of the financial gameplay (and also add interesting objectives for the player). **Sponsor features:**
- **Sponsor Slots:** The team will have several sponsor slots to fill. For example, one primary sponsor (biggest deal, appears on the car livery prominently) and a few secondary sponsors. Each slot can hold one sponsor contract at a time.
- **Sponsor Offers:** At the start, the player gets a selection of sponsor offers to choose from. Sponsors differ in their payment structure and goals. For instance, one sponsor might pay a large upfront amount plus a small amount per race, another might pay a moderate amount per race with hefty bonuses for podium finishes. Sponsors also have contract durations (e.g., 2 seasons).
- **Dynamic Goals/Targets:** Each sponsor provides objectives to earn bonuses [19] . For example: “Qualify in top 10 at least 5 times this season for a $5M bonus” or “Finish P8 or higher in a race to earn $1M (repeatable for each race)” [20] . Some sponsors have easy goals with small rewards; others have tough goals (“Win a race”) but high pay-outs. The player can pick which sponsor best aligns with their team’s expected performance. This creates a mini-strategy: a lower team might prefer a sponsor that offers guaranteed per-race money for finishing 15th or above [21] , whereas a top team might take one that gives huge bonuses for podiums.
- **Marketability and Sponsor Quality:** In real management games, a team’s attractiveness (often called marketability) affects what sponsors they can get [16] . We can implement a simple marketability rating (based on team results and driver popularity) which gates the best sponsors. For example, five-star sponsors (big brands) only approach you if you’re a top team with highly marketable drivers [16] . Initially, as Williams-like underdogs, players might only get low-tier sponsors. As they achieve success, better sponsor deals become available, reflecting growing team reputation.

- **Sponsor UI:** A new **Sponsors** screen lists current sponsors, their payments, and objectives. It also allows browsing available sponsors when a slot is open. The interface should highlight the key terms of each sponsor: upfront payment, per race payment, bonus conditions, contract length. For clarity, use a table or cards for offers, and perhaps icons or colors to denote difficulty of objectives (e.g., green = easy goal like finish 15th, red = hard goal like podium). The sponsor logos (if fictional logos are used) can add visual flavor. The Finance Dashboard should summarize total sponsor income per race and potential bonuses achieved so far.

- *Financial UI:* The **Finance Dashboard** will present the team’s financial status at a glance. Key elements to display: current balance, projected end-of-season balance (considering known incomes and costs), and a breakdown of income vs expenses (perhaps a line or bar chart). For example, show a bar for total income (split into sponsors, prize money, etc.) versus total expenses (split into salaries, R&D, etc.) so the player can see if they’re profitable [22] [23] . A balance trend chart over the season could be useful as a visual analytic. Warnings should be given if the balance is about to go negative or if the player is exceeding the cost cap.

- *Economic Difficulty:* There should be consequences for running out of money. If the budget goes negative, the game could restrict new spending (no new R&D projects or contract signings until finances improve). In a severe case, maybe the player could get fired or have to take a high-interest loan (loans could be a feature if needed). This pressures the player to not overspend.

- *Integration with Prior Features:* Driver contract negotiations now directly tie into finances – you can't offer a huge salary if you don't have the budget. We may include an **indicator of affordability** when negotiating (e.g., showing how a proposed salary impacts the budget). R&D choices are also constrained by cost now, adding challenge to development decisions. The player must prioritize: do they spend on a car upgrade or save money for an upcoming driver contract? This interplay is the heart of management strategy.

Citations: Motorsport management games emphasize finances as *"absolutely vital if you want to be successful"* [22]. Income comes from sponsors, prize money, chairman investments, etc., while major expenses include car development, salaries, and HQ upgrades [15] [17]. Sponsors in particular are crucial: *"Sponsors are one of the most important financial sources. Each sponsor has an Upfront Payment... you get instantly when the contract is signed"* [16]. Many sponsors also offer per-race or performance bonuses for hitting objectives [19], so choosing the right sponsor deal can significantly affect a team's budget. Our game's financial system is built to mirror these principles, requiring careful budgeting and strategic sponsor selection.

Rationale: With finances and sponsors, the game now simulates the real-world challenge of running an F1 team within economic constraints. This feature makes every decision more meaningful: the player must consider the monetary impact of R&D and hiring decisions. It also introduces a new "game" of securing sponsor deals and meeting targets, which adds variety to the gameplay loop beyond just racing. By implementing this in Version 4, we ensure the upcoming features (like race strategy) will be influenced by resource management (e.g., you might take a risk in a race to secure a sponsor bonus). In short, this financial layer enriches the strategic depth and realism of the simulation.

Version 5: Car Setup Tuning and Race Weekend Preparation

Overview: Version 5 brings the action to the race weekends by introducing **car setup tuning** and related practice session elements. Before each qualifying and race, the player can tweak the car's setup (wing angles, tire pressures, suspension settings, etc.) to suit each circuit and driver preferences. The goal is to find an optimal setup "sweet spot" that maximizes performance. This feature adds a hands-on, engineering mini-game to each race weekend, making the lead-up to qualifying interactive rather than just passive. It leverages drivers' feedback stats and track-specific characteristics, deepening the realism. The UI will include a **Practice/Setup screen** where these adjustments are made with visual indicators for performance windows.

New Features in Version 5:
- *Practice Sessions & Feedback:* Each race weekend now has a Practice phase (could be simulated in the background or interactive if we choose). During practice, the player sends their drivers out to gather feedback on the car setup. As in real life, the driver will report how the car feels (e.g., too much understeer, gearing too short, etc.). We can abstract this: each setup is defined by a set of sliders or values (for example: Front Wing Angle, Rear Wing Angle, Suspension Stiffness, Tire Pressure, Gear Ratios). For each driver and track, there is an "optimal range" for each parameter. The player's task is to adjust these to find the optimal combination. Drivers have a feedback rating that determines how quickly and accurately they give setup feedback. After a certain number of practice laps, the game can display clues such as "Front wing: far from optimal (too high)" or "Gear ratios: a bit low geared" to guide the player. The feedback system could be represented by 5 stars or checkmarks that fill in as the driver learns the setup [24] – for example, a driver might need ~15-18 laps to reach full feedback on all aspects [25], after which they'll comment on each area.

The player can choose to run practice in segments, pitting to adjust once some feedback is received (e.g., at 2/5 feedback points you get partial info and can already tweak) [25] .
- *Setup Parameters:* For simplicity, we might start with 3–5 adjustable parameters: **Aerodynamics (front/ rear wing)**, **Balance (suspension stiffness or anti-roll)**, **Gear Ratio**, **Tire Camber/Pressure**, and **Suspension (ride height)**. Each parameter has a continuous range, but the optimal setting is track-specific and driver-specific (drivers may prefer slight differences). The "performance window" is essentially how close you get to the optimal – a perfectly tuned car might get, say, a 100% performance utilization, while a poor setup might only extract 90% of the car's potential pace.
- *Performance Gain:* A better setup yields better lap times in qualifying and the race. For example, every percentage off the optimal might cost a tenth of a second per lap (just as a concept). So if you nail the setup (within the sweet spot), your drivers can maximize the car's stats; if you miss badly, your car underperforms relative to its potential. In management terms, this gives the player another lever to outperform rival teams with equal machinery.
- *Driver Setup Skill:* Each driver could have a "Feedback" or "Setup Skill" attribute (common in such games). A driver with high feedback will get to optimal setup faster and maybe has a broader tolerance. A less experienced driver might struggle, requiring more laps to hone in. This makes practice more valuable for some drivers than others.
- *Track Characteristics:* Each circuit will have inherent traits (some known from data sources or can be defined): e.g., Monza is low-downforce, Monaco is high-downforce, etc. We will use these to determine optimal aero levels and gear ratios for that track. Weather (if implemented in future) also could affect ideal setups (e.g., wet requires softer suspension), but weather can be introduced later. For now, assume mostly dry.
- *Setup UI:* The **Setup Tuning interface** should be interactive and user-friendly. For example, show sliders for each parameter with markers for current setting. After a practice run, highlight ranges as "Too low / Too high / Good" based on feedback. Some games represent this with bars and colored ranges – the idea is to visually indicate the sweet spot range for each parameter after enough feedback. If 5/5 feedback is achieved, the optimal range might be narrowed down very tight (e.g., a highlighted green zone on the slider). The player then adjusts sliders into those green zones for an optimal setup. Additionally, display an **overall setup confidence** or rating (e.g., 95% confidence once near optimal). This gives a sense of progress.
- *Time Management:* The player has limited practice time (say 60 minutes, or a certain number of runs). This means they can only test a setup so much. They must balance spending time to perfect the setup vs. preserving engine life or practicing qualifying sims. However, to keep it simple, we might just allow a fixed number of adjustments (like 2 or 3 rounds of feedback) per driver. This is enough to make it a challenge without being tedious.
- *Qualifying Preparation:* After practice and setup, the player enters Qualifying with the setup locked in. A great setup from practice will give the driver a better chance to set a competitive time. We haven't detailed Qualifying itself yet (that's coming in Version 6 with session management), but even if Quali is simulated, we now incorporate setup as a factor. If implementing Quali in this version: perhaps an automated one-lap simulation where time = base car performance + driver skill + setup bonus + random.
- *Integration:* The setup mini-game ties into R&D: no matter how good the parts are, a bad setup can undercut performance. It also adds a use for drivers' feedback stat. Additionally, this feature increases player engagement between races – now each race weekend has an interactive phase (practice tuning) before results are decided.
- *UI/UX Considerations:* Despite multiple sliders and data points, the UI must remain **clear and uncluttered**. One approach: use a tab or modal for "Car Setup" where all sliders are visible, and next to them, small gauge or bar showing current vs optimal range discovered. Use driver dialogue boxes or text hints for feedback (e.g., "Driver says: Car is unstable in corners (oversteer)" to hint front wing is too low). The design should use consistent icons (maybe a wing icon for downforce, a tire icon for handling, etc.) to aid quick

recognition. Keep the color coding intuitive: red = bad setup, green = optimal. Also ensure it's responsive (sliders should be easy to use on both desktop and mobile, given browser-based nature).

Citations: Realistic management games emphasize extracting performance via setups. For instance, **Motorsport Manager** had a similar concept where you aim for a percentage setup rating (like 95%+) by adjusting settings based on driver feedback. In F1 Manager (Frontier) 2022/23, practice feedback is slower – *"It takes about 18 laps to get 5/5 feedback points… at 2/5 you will get feedback for 2 aspects of the setup"* [25]. This indicates our design of progressive feedback is on point. A good setup is vital because an optimal setup can give a significant performance boost, while a wrong setup can degrade lap times and increase tire wear. As one guide notes, *"Getting the most out of a certain tyre is vital to strategy, and when they [the tires] reach a certain grip percentage, lap times drop"* [26] – a part of that is having the right setup to keep tires in their ideal window. Our game will incorporate these principles, making setup tuning an important factor in race performance.

Rationale: Adding car setup tuning enriches the realism and gives the player more agency during race weekends. It simulates the technical skill of race engineering, not just high-level decisions. This feature also creates variation – even if you have the best car, a missed setup can let weaker rivals beat you, which mirrors real F1 unpredictability. It keeps the player engaged in between the managerial tasks of hiring and finances by providing a puzzle to solve each race. By mastering setups, players feel more connected to on-track performance, which will tie nicely into the next version where they will directly manage qualifying and race events.

Version 6: Race Weekend Management – Qualifying and Race Strategy

Overview: Version 6 is a major milestone that turns the game into a full race weekend simulation. It introduces interactive **Qualifying sessions and Race day management**. The player will not just simulate results, but actively make decisions during qualifying and races: when to send cars out in qualifying, race strategies (pit stops, tire choices), real-time commands (push, conserve, overtaking aggression), and team orders. The simulation of the race will run in a timed manner (either real-time or accelerated), allowing the player to respond to events. This feature adds excitement and hands-on control, fulfilling the fantasy of being a team principal on the pit wall. The UI now gains a **Race Control screen** with live data (timing, positions) and controls for strategy, as well as a **Qualifying screen** for managing qualifying runs.

New Features in Version 6:
- *Qualifying Sessions:* Implement a qualifying format (for realism, the modern F1 Q1/Q2/Q3 knockout can be used, or a simpler one-shot or short session for manageability). The player will oversee qualifying and decide when and how to run their cars:
- **Qualifying Management:** In Q1 (if multi-session), the player must send each driver out at least once to set a lap time. They choose the timing (which affects track conditions and traffic) and the tire compound for the run. For example, waiting until late in the session might give better track grip but risks running into traffic or session stoppages. The interface could have a "Send Out" button for each driver, and show a timer countdown of the session. If a driver is sent, their out-lap, flying lap, in-lap will be simulated with a resulting lap time. The player can potentially recall a driver (abort lap) if needed (e.g., if it started raining).
- **Tire and Run Choices:** The player manages tire usage – using softer tires yields faster times but you have a limited number of sets. Also, any tire rules (like starting tire rules) can be considered if we want that

realism.
- **Result and Advancement:** After Q1, the slowest cars are knocked out (if using knockout). The player aims to get both drivers into the final qualifying. All this is presented with a timing screen showing current provisional times, a cutoff line, etc. At the end of qualifying, the starting grid is set. (If this is too complex initially, we could implement a simplified one-session qualifying where all cars run at once or sequentially, but ideally the official format is more engaging.)

- *Race Simulation Engine:* The core of this version is a **live race simulation**. We need to simulate each lap of the Grand Prix with reasonable fidelity: cars have lap times based on their performance (car stats + driver + setup) and affected by fuel load, tire condition, and race events. A simple physics model can determine lap times and overtakes, or we can use probabilistic checks for overtaking based on pace difference and track difficulty. Key aspects to simulate:
- **Lap-by-Lap Progress:** Iterate through laps (perhaps in a time-accelerated loop). At each time step (could be each quarter-lap or so for granularity), update positions.
- **Overtaking:** If a car is significantly faster and close behind another, an overtake chance is calculated. Driver stats (overtaking vs defending) influence the success [27] . Aggression settings (player controlled) also factor in – higher aggression increases overtake likelihood but also tire wear and risk.
- **Tire Wear and Performance:** Each tire compound has a lifespan and performance curve. For example, Soft tires start fast but degrade quicker, losing pace after X laps. We model tire wear as a percentage; pushing hard (aggressive driving) makes it drop faster [26] . When a tire is worn (below, say, 30% grip), lap times slow significantly and risk of mistakes or punctures rises [28] . The player must monitor this and decide when to pit.
- **Fuel and ERS (optional):** If fuel is a factor (modern F1 has no refueling, but fuel weight affects early pace), we can simulate fuel burn and perhaps allow engine mode changes that trade fuel for speed. ERS (battery energy) could be a later addition; initially we might skip or include as a simple push-to-pass boost toggle.
- **Pit Stops:** Players plan pit strategies: number of stops, which lap to pit, and which tire to switch to. A strategy screen before the race can allow setting a default strategy (e.g., one-stop vs two-stop) for each driver. During the race, the player can also call an audible and pit earlier or later depending on circumstances (safety car, rain, etc. — safety car and rain are advanced events, possibly added in a future version). A pit stop will cost some time (like ~20 seconds) and then reset that car's tire wear on new tires. Quick pit reactions can win the player time over AI opponents.

- **Real-Time Decisions:** The player has a suite of commands for each driver during the race:

    - **Pace/Aggression:** e.g., set to *Push*, *Balanced*, or *Conserve*. Pushing makes the driver drive faster at the cost of higher tire (and fuel) consumption and greater risk of mistakes. Conserve does opposite. This corresponds to how hard the driver is pushing, similar to "engine modes" or "tire saving" in other games [26] . Aggression also impacts overtaking: a driver on high aggression will attempt riskier passes but might wear tires or cause incidents [27] .
    - **Team Orders:** The player can issue orders between teammates, such as *"Do not fight teammate"* (so the faster car can overtake easily) [29] , or a direct *"Swap Positions"* if one driver is on a different strategy. Also, *"Hold Position"* could be used to avoid battles in the closing laps. Team orders help maximize team result and avoid crashes between teammates. Some drivers might have traits (like "Hates Team Orders") that cause them to ignore or be unhappy with such commands [30] .

    - **Pit Instructions:** Beyond scheduled stops, the player can pit a driver at any time (perhaps via a “Pit Now” button and selecting tire compound). The UI should display each driver’s tire wear and perhaps predicted position after pit to help decisions.
    - **Emergency Calls:** e.g., if a driver has damage or tires about to puncture, an alert pops up and the player should react (pit for repairs).

- *Race Control UI:* The interface for the race will be fairly complex and thus needs careful design for clarity. Key components:

- **Live Timing/Leaderboard:** A list of all cars with their current position, intervals to the car ahead/behind, current tire, number of stops, etc. This updates every few seconds or each lap. The player can glance here to see the race status.
- **Track Map (2D)**: While the *full* live map with moving dots is slated for Version 7, even in Version 6 we might have a basic track map with static info (or a textual representation of gaps). But likely we wait for Version 7 for the visual map. Instead, we can have a simplified positional chart (like a bar that shows relative positions on track or a running order list).
- **Driver Panels:** For each of the player’s drivers, have a panel showing their current status (position, tire wear graphic, fuel if used, current command setting). From here, the player issues commands: e.g., toggle push/normal/conserve, set overtaking to aggressive/normal, etc., and a pit button. These panels should also show alerts (like “Tires overheating” or “Stuck behind slower car”).
- **Messages/Commentary:** A text feed that logs important events: overtakes (“Driver X passed Driver Y for P5”), pit stops, crashes, etc. This gives life to the race and provides context for what competitors are doing. It can also include engineer-radio style messages for flavor (e.g., “Driver: My tires are gone!” when wear is high, prompting the player to pit).

- **Strategy View:** Possibly a sub-screen or overlay where the player can visualize strategies (like expected pit window laps, maybe a simple chart of tire performance over laps). But this might be advanced; at minimum, tooltips or small indicators (like an estimated pit window lap) can be shown.

- *Simulation Speed:* The player should be able to adjust the simulation speed (pause, 1x, 2x, 4x, etc.) since a full race in real-time would be long. We will allow pausing (to issue orders) and maybe an acceleration so a race can be done in a manageable time (e.g., 10-15 minutes). Key moments like the start or final laps might be watched at normal speed for excitement.

- *AI and Difficulty:* The AI-controlled teams will also execute strategies and react to some extent. They will pit when tires are worn, use aggression when chasing, and obey the same rules. We might include different difficulty levels affecting AI intelligence (and maybe their inherent pace).

- *Session transitions:* After qualifying, the results set the grid. On race day, the player can adjust the starting fuel load (if used) and starting tire (with rules like top 10 maybe locked to their Q2 tire if we implement that rule, or we can ignore that rule for simplicity in modern era since 2022 it’s not required). Then the race starts and the above simulation plays out. After the race, results are recorded (along with stats like fastest lap, etc., if desired) and championship standings update.

- *UI/UX Considerations:* This is the most complex interface so far, essentially like a control center. **Clarity and minimal clutter** are paramount. We should use panels, color coding, and maybe modals smartly: for example, clicking on a driver in the leaderboard could bring up detailed stats or controls for that driver, to avoid everything showing at once. Hotkeys or quick-buttons (like “push for

2 laps" as a quick command) could be useful for power users. The design should mimic a real F1 pit wall screen aesthetic – dark backgrounds, multi-column data, possibly customizable widgets. It should feel like a techy, data-rich environment but with important info highlighted (maybe your team's drivers are emphasized in the timing screen). We will also ensure it runs smoothly – heavy calculations on Node backend with updates sent via WebSocket to the React frontend for real-time updates could be used to handle the live nature. The user experience should be thrilling: for example, a subtle animation when positions change, or tire icons that gradually change color as they wear (green to red). Sound alerts (if applicable in a browser context) could notify when a command is needed (like a radio message sound).

Citations: This feature is essentially bringing the game to parity with the management aspect of F1 Manager and Motorsport Manager during races. As the F1 Manager 2022 description puts it: *"From lights-out to the chequered flag, you're in control. Own every decision from pit strategy to tyre choices to driver callouts"* [31]. That's exactly what we implement here. Effective strategy means monitoring tires and adjusting aggression, because *"Aggressive driving will result in tires wearing down faster... Getting the most out of a tyre is vital to strategy"* [26]. The player can also use team orders; for instance, in F1 Manager 2023 you can tick "don't fight teammate" to have one driver yield to the other [29]. We allow similar orders. By referencing known practices (like adjusting defensive aggression [27]), we ensure our simulation behaves believably. With this live management, the game crosses into a fully immersive experience.

Rationale: Version 6 is arguably the core of the "manager" experience, turning the game from a management sim that mostly happens between races into one that fully involves races. It is complex, but it's what truly makes it an F1 management game rather than a spreadsheet simulator. By implementing it after establishing all the foundational systems (team, drivers, R&D, finances, setups), we ensure the player has context and stakes for the decisions they make on race day. This version will be the most engaging, as the player's tactical prowess in each race can directly influence results, not just the long-term strategy.

Version 7: Live Race Visualization (Track Map & Telemetry)

Overview: In Version 7, we enhance the race presentation by adding a **live map-based visualization** of the race. Instead of relying only on text and tables, the player will see the cars (represented as icons or colored dots) moving around a 2D circuit map in real-time. This visual element greatly improves the user experience, making it easier to follow the action (who is where on track, gaps visually, etc.) and adding excitement as battles and track position become more intuitive. In addition, we will introduce more visual data like real-time telemetry or performance graphs during the race (for example, a tire wear graph or timing chart) to fulfill the desire for visual performance analytics. This version is about polish and immersion, giving the game a broadcast-style quality.

New Features in Version 7:
- *2D Track Map Display:* For each circuit, we will use either a stylized track outline or a simple schematic map. Cars will be rendered as moving dots or small car icons on this map, updating their position every few seconds (or every simulated "sector"). Each car's icon can be labeled with the driver number or an abbreviation (e.g., HAM for Hamilton) for identification. The player's own team might be highlighted (e.g., using team colors or a halo around their dot). The map will show the track layout and maybe key markers (start/finish line, pit entry). This replicates the "helicopter strategic view" that some management games offer [32]. In fact, F1 Manager 2024 added a "Strategic Helicam" which is essentially a top-down view to plan strategy [32] – our map serves a similar purpose, giving a clear overview of where everyone is.

- *Smooth Animation:* The movement of cars on the map should be smooth to the eye. We can interpolate positions between updates for a fluid motion, or update fast enough that it looks continuous. The cars' speed can be reflected by their movement speed on the map (though we might cap it visually to not zip too fast when accelerating time). If a car is in the pits, it could be shown in a pit lane path or removed from track and shown as "PIT" status. If a car crashes or retires, maybe its icon stops on track or is removed with a marker.
- *Interactive Elements:* The map could be interactive – the player might click a car's icon to select that driver (instead of using the panels). This would be a nice UX addition. Also, hovering on a car could show a tooltip with driver name, position, tire, gap to car ahead. This makes the map not just eye-candy but a functional tool for information.
- *Visual Race Events:* We can add simple visuals for events: e.g., yellow flag sectors highlighted on the map if an incident occurs, or a safety car icon if deployed (should we introduce safety car at this point for completeness). However, even without going that far, the map itself showing cars bunching up behind a safety car would be evident.
- *Charts and Graphs:* In addition to the map, we implement **visual performance analytics** during and after sessions. Examples:
- **Lap Times Graph:** A real-time line graph plotting the player's drivers' lap times and perhaps a few rivals, to see performance trends and tire degradation.
- **Gap Graph:** A chart that shows the gap between two cars over time (like the "race duel" graphs used in broadcasts). If the player is chasing another car, they can visually see the gap closing or widening each lap.
- **Tire Wear Graph:** We can display the wear percentage of tires over laps for each stint, possibly after the race as part of analysis.
- **Driver Stats Table:** A table updated live or post-race showing stats like number of overtakes, top speed, sector times, pit stop duration, etc.
- **Championship Projections:** Maybe a dynamic display of what the championship standings would look like if the race finished as is (for drama).

These analytics satisfy the requirement of visual performance data reporting. After the race (or anytime), the player could access a **Data Analysis** screen to review these charts for their team and others. This is useful for making decisions (e.g., seeing that your car consistently loses time in Sector 2 could hint an aero weakness to address). As per the question, graphs and data tables should be included, so we ensure at least a few clear charts are present. We might use a charting library to implement these, ensuring they are interactive (hover for values) and can be toggled.

- *Enhanced UI Presentation:* The overall UI, especially during races, will feel more polished with these visuals. It's akin to watching the race from a bird's-eye perspective rather than just reading numbers. We should also refine any remaining UI elements for consistency. Perhaps add **replay or highlights** functionality (not fully animated, but maybe key event markers you can jump to in a replay timeline – this could be an advanced touch if time permits, otherwise skip). Also, ensure the color scheme and design align with a professional look; by now, we can incorporate user feedback from earlier versions to smooth any rough edges in usability.

- *Performance Considerations:* Rendering moving elements and charts in real-time needs to be optimized. We will leverage React's canvas or SVG capabilities for the map (or even WebGL if needed for performance, but likely not necessary). The Node backend will feed positions (maybe as distances along lap) to the frontend at a set interval. We must ensure this doesn't bog down slower machines – possibly providing an option to simplify or turn off live map for lower-end systems.

Citations: The idea of a live track map with moving dots has precedent in older management games and current ones. For instance, early versions of Motorsport Manager on PC showed cars as simple icons on a 2D track, which one reviewer noted as *"tiny models of cars instead of dots on the maps"* in later versions [33]. This kind of visualization greatly helps in understanding race context. Furthermore, real F1 teams rely on tons of data and telemetry visualizations; as an article on LightningChart notes, *"Through data analysis, teams can improve race performance, fine-tune strategies, and enhance car development"* [34]. By adding graphs for lap times, tire wear, and so on, our game is giving the player similar analytical tools to assess and improve their performance. This fulfills the request for visual performance analytics in the spec.

Rationale: Version 7 doesn't add a lot of new gameplay mechanics but elevates the *presentation* and *information* available. This is critical for a sophisticated UI/UX experience. By seeing the cars move on a map, players can more intuitively grasp strategic situations (like how far apart cars are, when to pit to avoid traffic, etc.). The charts and data satisfy the more hardcore players' desire to dive into numbers, but can also be ignored by casual players who just enjoy the visuals. Essentially, this version transforms the simulation from text-based to visually rich, making the game more immersive and closer to a professional simulation tool. At this stage, our game has all major features implemented as per the specification, and Version 7 ensures it's presented in a polished, user-friendly, and engaging way.

Version 8: Driver & Team Development, Stats, and Further Realism

(Final planned version in this specification; focuses on rounding out the game with detailed stats and any remaining features to mirror a complete F1 management experience.)

Overview: Version 8 adds comprehensive **driver and team performance data tracking**, as well as development of personnel over time. This means drivers will have a range of stats and can improve or decline, and teams can invest in staff and facilities for long-term gains. It also includes a robust statistics system for player enjoyment (records, history, etc.). By now, all core systems are in place, so this version polishes the long-term gameplay and realism. It ensures that multiple seasons of play remain engaging with driver aging/regression, new talent, and the ability to see the story of your career in data form. The UI will include **Driver Profile pages**, **Hall of Fame/Records**, and possibly **Staff Management screens** if we add those elements.

New/Enhanced Features in Version 8:
- *Expanded Driver Stats:* Instead of a single overall rating, drivers now have multiple attributes that affect their performance. Typical attributes might include: **Pace** (outright speed), **Consistency**, **Overtaking skill**, **Defending skill**, **Wet Weather skill**, **Tire Management**, **Feedback**, **Morale**, and **Experience**. These can be scaled 0-100 or a star rating. The overall rating (OVR) can be an aggregate. These stats influence all the systems: e.g., a driver with high Overtaking is more likely to pass others in race [27], high Consistency means fewer mistakes, high Feedback means quicker to optimize setup, etc. Drivers also have **personality traits** (optional but adds flavor) such as "Mentor" (helps teammate improve) or "Hates Team Orders" (resists team orders) [30], which we already hinted at.
- *Driver Development:* Drivers will improve or decline each season based on age and performance. Young drivers gain stats with experience, up to their potential. Older drivers (say over 32) might slowly decline in reflexes/pace [35]. We implement a system where after each race or season, stats are adjusted slightly – good performances could accelerate growth. Also, the player could dedicate some of the offseason or resources to training a driver (maybe an event like "send driver to training camp for +2 Pace, cost $X"). This gives another dimension to management: nurturing talent vs hiring established stars. The **Driver Profile** UI

will show their stat progression over time (like a line chart or historical stat values) and their contract, age, etc. It should also highlight career achievements (wins, podiums, titles). Frontier's F1 Manager shows driver stat categories and notes that they are starting values that change as the career progresses [36] ; we'll do the same, ensuring drivers in our game are not static [37] .
- *Team Staff and Facilities:* To further mirror reality, introduce management of key staff (Technical Director, Head of Aerodynamics, Race Engineer, etc.) and facilities (Factory, Wind Tunnel, Simulator). These contribute to car development speed, driver training, pit stop efficiency, etc. For example, a better Factory (upgraded via money) could allow more simultaneous R&D projects or faster part fabrication. A better Pit Crew facility or training program could reduce average pit stop times. These elements provide additional uses for finances in late-game and allow the player to shape the team's strengths (maybe you build a strong pit crew to gain time in pits, or a state-of-art simulator to boost driver development). Managing staff could involve hiring those with ratings, similar to drivers. While detailed staff management is extra, even a simplified version (just upgrade levels of departments) would add depth.
- *Records and Stats Tracking:* Over multiple seasons, a lot of data accumulates. The game should celebrate and utilize this by tracking statistics such as: most wins for a driver, all-time records (e.g., fastest lap ever at a track), team records (championships won), etc. A **History/Records** screen can list each season's champions and allow the player to review past seasons' results. This is great for immersion – as you play 10 seasons, you see the legacy you built or how the balance of power changed. If using real drivers, by season 5 new fictional drivers may be dominating if old ones retired, so these new names become part of the history.
- *Additional Realism Features:* At this stage, we can consider any remaining F1 features not yet included, if scope allows:
- **Regulation Changes:** Every few seasons, a rule change could shake up car performance. For example, a new aero regulation resets aero R&D by some percentage, teams with better facilities adapt quicker. This adds variability to long-term play.
- **Random Events:** e.g., driver injury (so you need a reserve driver temporarily), teams get penalized for rule breaches, etc. These can be optional flavor events to keep the player on their toes.
- **Media and Reputation:** Not in original scope, but could be fun – press interviews affecting morale, or board confidence level. However, this might be beyond what's needed since focus is more on management mechanics.

- *UI/UX Improvements:* The interface gets refined to handle these new data views. The **Driver Profile** page should use charts and tables to show stats and development (e.g., a radar chart of their skill set, or line graphs over seasons). The **Team Overview** might display facility levels with progress bars and upgrade buttons. The **Hall of Fame** or Records could be a simple list or even an in-game news article style summary of each season. We should ensure navigation through historical data is smooth (perhaps a dropdown to select season archives). Also, tooltips explaining new stats or features (like what each facility does) will help users master the complexity. At this final version, the UI should feel comprehensive but still navigable – likely by organizing sections (Main Dashboard, Team (subsections: Drivers, Staff, Facilities), R&D, Finances, Race Weekend, etc.). A consistent top nav bar or menu can allow jumping between these sections. Given we've added a lot, having a good menu is crucial now. The visual design should be coherent; possibly by now we have a custom F1-styled theme across all components (with team color accents, etc.).

Citations: By incorporating driver stat growth, we acknowledge that *"young drivers will gain skills and older drivers will start getting worse"* as noted in Motorsport Manager [38] . Also, marketability (which we used for sponsors) can be a driver stat too, impacted by traits (one trait "Media Shy" reduces marketability [39] , whereas "Narcissist" might increase it [40]). These nuances add realism. Tracking improvements is key; as the F1 Manager 2024 official site shows, they list drivers with categories (like "CNR, BRK, RCT…" presumably different skills) and note those *"ratings will change as the player progresses throughout their career"* [37]. For team development, older games like Grand Prix World had staff and engine supplier negotiations [41], and modern games emphasize facilities (e.g., investing in HQ). We reflect that in our design. Lastly, including rich stat tracking and analytics aligns with the earlier point that data analysis is crucial in F1 [34] – now the player can do that for their own team and league over many seasons.

Rationale: Version 8 ensures longevity of the game. With drivers improving/retiring and the ability to upgrade infrastructure, a 10-season playthrough will feel like an evolving world, not a static repetition. Players can form attachments to drivers they've developed from rookie to champion, which is very rewarding. The detailed stats and history give a sense of accomplishment and context ("In year 2030 my team finally won the title, and my lead driver broke Schumacher's wins record!"). By completing these features, our game achieves a comprehensive management simulation status, combining breadth (many aspects to manage) and depth (detailed systems in each aspect).

Data Sources for Drivers, Teams, and Tracks

(Recap and additional detail, if needed, on how we'll use real-world data in the game implementation.)

To support realism, the game will integrate data for real F1 entities:

- **Drivers:** We will start with real driver names, nationalities, and baseline skill ratings reflecting their real prowess (e.g., top drivers have higher stats). Sources like the Ergast API provide lists of drivers per season, including basic info [1]. Additionally, community-created rating databases (for example, the F1 Manager 202X game's driver stats or mods) could guide our initial ratings. Since our game might extend into future seasons, we can also generate fictional drivers for when real ones retire, but even those can be inspired by F2/F3 driver data (which can be fetched or manually input).
- **Teams (Constructors):** The 10 real F1 teams and their data will be used. Ergast provides constructor info [42], and we can use real performance data from the past season to set their initial car performance and budgets (e.g., the team that won last year starts with the best car stats but perhaps lowest wind tunnel time due to regulations, etc.).
- **Tracks:** All circuits on the F1 calendar will be included. We will use data like track length, number of laps (for race distance), and maybe corner counts to influence tire wear and overtaking difficulty. The circuit GeoJSON repository [4] can provide track layouts for our 2D map and perhaps coordinates to map car movement. We might simplify each track to a path for cars to follow on the canvas. Track-specific parameters such as fuel consumption rate or downforce level needed can be hardcoded or sourced from racing data websites (e.g., race previews often mention if a track is high-downforce or tough on brakes).
- **Historical Data:** If we include past seasons or allow starting in, say, 2023 or 2024, we can use historical results from Ergast to set initial conditions. For example, performance of teams could be derived from their 2024 standings, and driver stats could consider career wins/podiums.
- **Updates and Maintenance:** Since F1 data changes every year, we would design the game so data like drivers/teams are not hardcoded but rather loaded from external JSON or a database. This way, updating to a new season (2026, 2027, etc.) would be a matter of inputting new data (which could be scraped from Wikipedia or accessed via an updated API like the new Jolpica API).

Using these sources ensures authenticity while reducing the manual effort of data entry. All usage will respect licensing (Ergast data is free for non-commercial use, Wikipedia is CC-BY-SA, etc.). For any real-world data we present (like driver names, team names), we assume this fan game usage is acceptable under fair use (non-commercial, etc.), or we have fictional stand-ins if not.

In summary, real-world data will power the game's early experience – from driver abilities to track characteristics – creating an immersive simulation that feels connected to real F1 [1]. As the player progresses beyond the current season, the game world will organically diverge, but always rooted in an initially accurate representation of Formula 1.

Conclusion

This comprehensive specification outlined an F1 Management Simulation game that evolves from a simple season manager to a deep, multi-faceted simulation across many versions. Starting with a basic but playable foundation in Version 1, each subsequent version layered in new dimensions: driver contracts, car R&D, finances and sponsors, detailed setups, live race strategy, rich visualizations, and detailed stats. At each step, the focus remained on **clean UI/UX** and logical integration of features, ensuring the game grows in complexity without overwhelming the player. Real-world data sources like the Ergast API and community datasets will anchor the simulation in authenticity [1] [4]. By the final version, the game offers a sophisticated experience where players can manage every aspect of a Formula 1 team – from boardroom negotiations to on-track wheel-to-wheel battles – all within their web browser. The iterative development approach allows testing and refining at each stage, ultimately driving towards a polished product that will appeal to F1 enthusiasts and management game fans alike.

Each version of the project can be treated as a milestone, delivering a playable game with increasing depth. This not only makes development manageable but also means the game could potentially be released in stages (early access), gathering user feedback to fine-tune the UI and mechanics (especially important for complex features like live race control). Emphasizing a modern tech stack (React/TypeScript frontend, Node.js backend) ensures the game is accessible (just a web browser needed) and maintainable, with possibilities for future expansion (like online multiplayer or mobile adaptation, beyond this scope).

In conclusion, this specification provides a roadmap to create a fully-fledged Formula 1 management simulator. By following the versioned plan and leveraging the cited data and design principles, the development team can progressively build a game that is both **realistic** in simulating the intricacies of F1 and **engaging** in terms of gameplay and presentation. The end result will let players live out the dream of being an F1 team principal – making the tough decisions that lead their team to glory or defeat, with all the excitement that entails.

Sources:

- Ergast Developer API – Historical F1 data (drivers, constructors, circuits) [1] [2]
- OpenF1 API – Real-time and historical data (telemetry, timing) [5]
- F1 circuits GeoJSON repository – Track maps and locations [4]
- Motorsport Manager PC Wiki – Game mechanics (sponsors, finances, tires) [16] [26] [15]
- F1 Manager (Frontier) insights – Contract negotiation and team management features [6] [32]
- The-Race.com – Reference to past F1 management games and features (Grand Prix Manager series) [41]

- F1Manager.com (Frontier official) – Car development guide, driver database for stat categories [8] [37]
- Reddit r/F1Manager – Discussions on practice feedback and setup optimization [25]

---

[1] [2] [3] [42] Ergast Developer API – A public open source Formula One API
http://ergast.com/mrd/

[4] A repository of Formula 1™ circuits in GeoJSON format. - GitHub
https://github.com/bacinger/f1-circuits

[5] OpenF1 - An API for real-time F1 data : r/F1DataAnalysis
https://www.reddit.com/r/F1DataAnalysis/comments/16w84uz/openf1_an_api_for_realtime_f1_data/

[6] [7] F1 Manager 2024: How to sign and swap drivers | Traxion
https://traxion.gg/how-to-sign-and-swap-drivers-in-f1-manager-2024/

[8] [9] [10] [11] [12] [13] [14] Car Development & Research Guide - F1® Manager 2024
https://www.f1manager.com/en-US/2024/news/car-development-research-guide

[15] [17] [18] [22] [23] Finances | Motorsport Manager PC Wiki | Fandom
https://motorsportmanagerpc.fandom.com/wiki/Finances

[16] [19] [20] [21] Sponsors | Motorsport Manager PC Wiki | Fandom
https://motorsportmanagerpc.fandom.com/wiki/Sponsors

[24] [25] Regarding the Feedback from Drivers for Set-up : r/F1Manager
https://www.reddit.com/r/F1Manager/comments/wy7jsp/regarding_the_feedback_from_drivers_for_setup/

[26] [28] Tires and Fuel Strategy | Motorsport Manager PC Wiki | Fandom
https://motorsportmanagerpc.fandom.com/wiki/Tires_and_Fuel_Strategy

[27] Team Orders: How do I make the "second driver" not just stop - Reddit
https://www.reddit.com/r/F1Manager/comments/1jkv1eb/team_orders_how_do_i_make_the_second_driver_not/

[29] Team orders (Don't Fight Teammate) :: F1® Manager 2023 General ...
https://steamcommunity.com/app/2287220/discussions/0/3816291531977018589/

[30] [35] [39] [40] Drivers | Motorsport Manager PC Wiki | Fandom
https://motorsportmanagerpc.fandom.com/wiki/Drivers

[31] F1® Manager 2022 - PS4 & PS5 Games | PlayStation (US)
https://www.playstation.com/en-us/games/f1-manager-2022/

[32] Features - F1® Manager 2024
https://www.f1manager.com/en-US/features

[33] Motorsport Manager - Wikipedia
https://en.wikipedia.org/wiki/Motorsport_Manager

[34] Create a formula 1 data analysis with LightningChart Python
https://lightningchart.com/blog/python/formula-1-data-analysis/

[36] [37] Drivers - F1® Manager 2024
https://www.f1manager.com/en-US/drivers

[38] Motorsport Manager (PC) - RPG Codex
https://rpgcodex.net/forums/threads/motorsport-manager-pc.113607/

[41] When F1 management games had their surreal ‘golden era’ - The Race
https://www.the-race.com/formula-1/when-f1-management-games-had-their-surreal-golden-era/