A blue and white logo

AI-generated content may be incorrect.

National College of Ireland

Project Proposal

Global Motorsport Management Simulator  
21/09/2025

Computing  
Software Development  
2025/2026

Eoin Fitzsimons  
X23151374  
X23151374@student.ncirl.ie

Contents

[1.0 Objectives 3](#_Toc209632613)

[2.0 Background 3](#_Toc209632614)

[3.0 State of the Art 4](#_Toc209632615)

[4.0 Technical Approach 5](#_Toc209632616)

[5.0 Technical Details 6](#_Toc209632617)

[6.0 Special Resources Required 7](#_Toc209632618)

[7.0 Project Plan 7](#_Toc209632619)

[8.0 Testing 8](#_Toc209632620)

# 1.0 Objectives

The project aims to design and implement a prototype motorsport management simulator that captures the scale, depth, and unpredictability of real-world racing ecosystems. Unlike existing titles, which restrict players to a narrow scope (e.g., Formula 1 and a limited feeder system), this project seeks to create a dynamic multi-tiered structure that includes top-level championships, regional series, and junior karting categories.

The objective is to simulate a persistent game world where events unfold across all leagues, regardless of whether the player is directly involved. By doing so, the system mirrors the emergent quality of *Football Manager*, where countless teams and athletes evolve simultaneously, ensuring unique career trajectories every time the game is played.

A further objective is to offer a high degree of customisation. Players will be able to modify or generate new teams, drivers, and competitions. Procedural generation will be a cornerstone, enabling large-scale fictional ecosystems that feel authentic but avoid licensing restrictions.

The project also aims to demonstrate robust implementation in terms of coding, simulation design, and AI-driven decision-making. The final deliverable will be a working prototype capable of managing driver growth, team finances, league standings, and race outcomes through computationally efficient algorithms.

Ultimately, the project’s objective is to produce a management simulator that can realistically be extended or monetised, filling a gap in the current market for motorsport fans who value simulation depth and customisation over graphical fidelity.

# 2.0 Background

The motivation for this project is rooted in a long-standing personal interest in both motorsport and sports management games. Having spent extensive hours in *Football Manager*, I have experienced the appeal of large, complex simulations that generate emergent stories. Despite having a stronger interest in Formula 1 and motorsport generally than in football, I find myself drawn more to *Football Manager* than to existing motorsport management titles. This reveals an important insight: the attraction lies not in the sport itself, but in the richness of the simulated world.

By comparison, current motorsport titles such as *F1 Manager* and *Motorsport Manager* are limited. Their focus is largely on presentation and visuals rather than world depth. The lack of customisation and the narrow scope of simulated leagues lead to repetitive experiences over time.

This project aims to bridge that gap by combining the thematic appeal of motorsport with the systemic depth of *Football Manager*. The objectives outlined in Section 1.0 will be met by designing algorithms that support persistent multi-league simulation, driver development models, and financial/strategic systems. Procedural content generation will be used to overcome licensing limitations and create a fictional but believable motorsport ecosystem.

The motivation also includes an academic challenge. Developing a simulation of this complexity requires advanced algorithm design, structured data modelling, and effective user interface planning. This stretches beyond the standard programme scope, making it both a technically and personally meaningful project.

# 3.0 State of the Art

Current motorsport management titles include *Motorsport Manager* (2016) and *F1 Manager* (2022–2024). These games present detailed races and authentic visuals, but their simulation depth is fundamentally limited. Driver development is linear and largely predetermined, making long-term play predictable. Once a player has learned the optimal strategies, the outcomes become repetitive. In contrast, *Football Manager* maintains replayability by simulating an immense global ecosystem of leagues, players, and staff, where no two careers are the same. This difference highlights the gap between motorsport management games and the gold standard of emergent sports simulations.

Another limitation of existing motorsport titles is their lack of customisation. Players cannot easily expand leagues, create fictional scenarios, or reshape the competitive environment. The games are tightly constrained by licensing agreements and static structures, which prevents the freedom that makes *Football Manager* engaging. Furthermore, the game worlds are small in scale, restricted to a handful of categories such as Formula 1 or Formula 2, with little simulation occurring outside the top tiers. By comparison, *Football Manager* tracks hundreds of leagues and thousands of clubs across multiple nations, with events unfolding regardless of direct player involvement.

This project aims to distinguish itself by adopting that philosophy. The simulation will extend beyond Formula 1 into lower levels such as Formula 4, regional series, and even junior karting, producing a large interconnected ecosystem. Every league will progress whether the player manages there or not, creating a sense of a living world. Because official licensing is not available, all teams, drivers, and championships will be procedurally created, which also allows experimentation with unique structures, rivalries, and narratives. By prioritising breadth, customisation, and emergent systems, this project will attempt to deliver the first motorsport management simulation with the scale and unpredictability of *Football Manager*.

# 4.0 Technical Approach

The development will be structured in phases, following a requirements-driven approach. Requirements will first be gathered by analysing existing sports management games to identify their essential mechanics (e.g., resource management, competition structures, AI decision-making). Additional requirements will be drawn from the objectives in Section 1.0, particularly the need for emergent systems and procedural content generation.

Requirements will be broken into four categories:

1. **World Simulation Requirements** – including hierarchical league structures, seasonal calendars, and persistent progression across multiple tiers.
2. **Entity Management Requirements** – covering drivers, teams, staff, finances, and facilities.
3. **Race Simulation Requirements** – algorithms for probabilistic race outcomes influenced by driver skill, car performance, and random events.
4. **Customisation and Procedural Generation Requirements** – systems for creating fictional but realistic teams, drivers, and championships.

Tasks and milestones:

* **Phase 1: Core Data Model (Weeks 1–3):** Implement relational or graph-based structures to represent leagues, teams, and drivers. Ensure scalability for thousands of entities.
* **Phase 2: Procedural Generation (Weeks 4–6):** Develop algorithms to create initial worlds, including unique attributes for drivers and teams.
* **Phase 3: Simulation Engine (Weeks 7–11):** Build algorithms for races, progression, and season updates. Incorporate stochastic elements for unpredictability.
* **Phase 4: Management Systems (Weeks 12–15):** Implement financial models, staff recruitment, training, and facilities.
* **Phase 5: User Interface Prototype (Weeks 16–18):** Provide a functional but minimal UI for testing and demonstration.
* **Phase 6: Testing and Refinement (Weeks 19–22):** Conduct rigorous testing (see Section 8.0).

This approach balances complexity with feasibility, ensuring each milestone delivers a testable artefact. The structured breakdown allows for incremental development while maintaining alignment with project objectives.

# 5.0 Technical Details

The implementation will use **Python** as the primary language, due to its strong support for data structures, algorithm design, and rapid prototyping. Core libraries will include:

* **Pandas/Numpy** for data modelling and simulation.
* **NetworkX** (or similar) for representing relationships between entities.
* **Matplotlib/Plotly** for visualising race outcomes and season statistics.
* **SQLite** or another lightweight database for persistent storage of the game world.

Race simulation algorithms will be probabilistic. For example, each race will combine weighted random distributions based on driver skill, car performance, weather, and chance events. A Monte Carlo approach may be used to simulate multiple possible race states before selecting a final outcome.

Driver development will be implemented using growth and decline functions influenced by age, training investment, and natural variability. This ensures no two careers follow identical paths. Team finances will be simulated through income (sponsorship, prize money) and expenses (staff wages, facilities, car development).

Procedural generation will assign attributes to drivers and teams at world creation. Names, nationalities, and performance traits will be generated using algorithmic methods, ensuring diversity and replayability. Leagues will be designed hierarchically, with promotion and relegation models similar to football but adapted to motorsport structures.

The user interface will be lightweight, potentially using a desktop GUI toolkit such as **Tkinter** or **PyQt**, or a web-based UI via **Flask/Django**. The emphasis will be on functionality and clarity rather than graphics.

Overall, the technical approach combines data-driven modelling with probabilistic algorithms to create a robust, extensible simulation engine.

# 6.0 Special Resources Required

This project primarily requires standard computing resources. Development will be performed on a personal computer with sufficient memory and processing power to run multi-threaded simulations of large datasets. Open-source libraries such as Pandas, Numpy, and SQLite will be utilised, meaning there are no licensing costs.

To create realistic procedural generation, datasets of names, nationalities, and race circuits may be required. These can be sourced from publicly available open datasets or created manually. For example, lists of common surnames by nationality can be combined with fictionalised track locations to generate a believable but original environment.

No specialised hardware (e.g., GPUs) is essential, since the project does not involve high-performance graphics. However, optional use of cloud computing services (e.g., Google Colab or AWS free tier) may be beneficial for scaling large simulations during testing.

The most significant resource requirement is time, particularly for iterative development and testing of emergent systems. Given the complexity of simulating thousands of entities across multiple leagues, computational efficiency will also be a focus.

# 7.0 Project Plan

**Phase 1: Research and Requirements (Weeks 1–2)**

* Review existing management games.
* Define functional and non-functional requirements.
* Produce initial UML diagrams and data models.

**Phase 2: Core Data Structures (Weeks 3–5)**

* Implement entities: drivers, teams, staff, facilities.
* Build league structures and calendar representation.
* Ensure database persistence for world state.

**Phase 3: Procedural Generation (Weeks 6–8)**

* Develop algorithms for name, nationality, and team generation.
* Test attribute distributions to ensure balance and realism.
* Implement initial fictional leagues.

**Phase 4: Race Simulation Engine (Weeks 9–13)**

* Design probabilistic race outcome models.
* Integrate driver skill, car performance, random events.
* Validate outcomes against expected statistical distributions.

**Phase 5: Management Systems (Weeks 14–17)**

* Implement finances: sponsorship, expenses, prize money.
* Build driver development model (training, decline).
* Add recruitment and staff management.

**Phase 6: User Interface (Weeks 18–19)**

* Develop functional GUI or web interface.
* Display league tables, results, and driver/team profiles.

**Phase 7: Testing and Refinement (Weeks 20–22)**

* Conduct system, integration, and unit testing.
* Optimise for performance in large-scale worlds.
* Collect informal user feedback (where ethical).

**Phase 8: Documentation and Submission (Weeks 23–24)**

* Write final report to research thesis standard.
* Ensure Harvard Referencing compliance.
* Prepare showcase demonstration.

Milestones will be tracked weekly with a Gantt chart, ensuring progress is visible and measurable. This plan is realistic but ambitious, aiming to balance innovation with feasibility.

# 8.0 Testing

Testing will be comprehensive, covering system, integration, and unit levels.

**Unit Testing:**

* Each component (e.g., driver progression function, race simulation) will be tested with controlled inputs to verify outputs.
* Edge cases will be included, such as extreme driver attributes or financial collapse.

**Integration Testing:**

* Subsystems (race simulation + league tables, or finances + recruitment) will be tested together to confirm interactions.
* Tests will ensure persistent data across seasons is correctly updated.

**System Testing:**

* Full simulations will be run over multiple virtual seasons.
* Results will be checked for plausibility (e.g., no team winning every race indefinitely, realistic driver ageing).
* Performance benchmarks will ensure scalability with thousands of entities.

**Validation Testing:**

* Outcomes will be compared to statistical models of real motorsport (e.g., distribution of race winners, financial variance).
* Informal user evaluation may be conducted to assess usability and realism, subject to ethical guidelines.

Evidence of testing will be documented with test cases, results, and analysis. Testing is critical given the emergent nature of the simulation, where unexpected behaviours may arise. Structured and rigorous validation will ensure credibility and robustness.