SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Quality Diversity for Racing Track Design

Laurea Magistrale in Computer Science Engineering - Ingegneria Informatica

Author: Martino Piaggi

Advisor: Prof. Pierluca Lanzi

Co-advisor: Prof. Daniele Loiacono

Academic year: 2024-2025

1. Introduction

Video game creation, especially level design, requires substantial resources for content production, making procedural content generation (PCG) a valuable approach for automating digital asset creation. A key challenge in PCG is generating diverse yet high-performing content, which traditional optimization methods often struggle with due to their focus on single objectives.

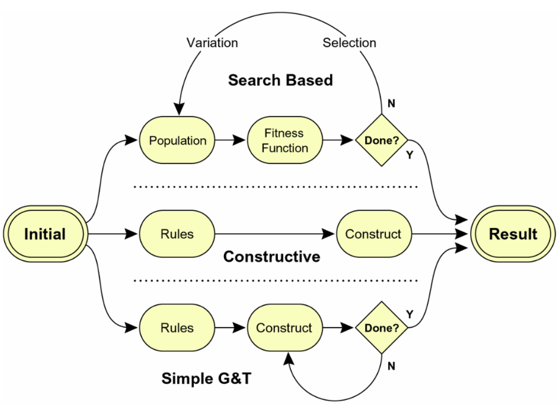


Figure 1: Conceptual diagram of the Search-Based Procedural Content Generation (SBPCG) loop, the general paradigm this research follows.

Quality Diversity (QD) algorithms offer a solution by seeking a wide range of high-performing solutions across a behavior space, inspired by biological evolution. This thesis investigates the application of QD, specifically the Multi-dimensional Archive of Phenotype Elites (MAP-Elites) algorithm [2], to the procedural generation of racing tracks for the open-source simulators TORCS [5] and Speed Dreams [3]. The system automatically evaluates generated tracks using functions derived from simulated racing competitions with existing AI drivers.

Our methodology involved developing novel genotype representations based on Voronoi diagrams and convex hulls, an end-to-end pipeline for generation and evaluation, and the use of dimensionality reduction for robust behavioral characterization. We also conducted preliminary experiments to quantify the inherent noisiness of emergent gameplay features, informing the selection of reliable behavioral descriptors and the refinement of the fitness function.

2. Methodology

This research leverages The Open Racing Car Simulator (TORCS), an open-source 3D racing platform chosen for its modularity, portability, robust AI support, and non-graphical mode, which significantly accelerates simulation times for large-scale experiments.

2.1. Track Representations and Genetic Operators

The core of our PCG approach lies in encoding abstract track structures (genotypes) into functional game assets (phenotypes) compatible with TORCS's XML format. We explored two primary genotype representations:

- **Convex Hull Method:** Adapted from Maciel [1], this method defines a track's genotype as a set of 2D points. The convex hull of these points forms a polygonal boundary, which is then smoothed using spline interpolation to create a drivable track.
- **Voronoi Diagram Method:** This novel approach derives the track's structure from the edges of a Voronoi tessellation generated from a set of seed points (Figure 2). By selecting a subset of these cells, a continuous track path is formed, later smoothed with spline interpolation. This method inherently produces closed-loop paths and offers high geometric expressiveness.

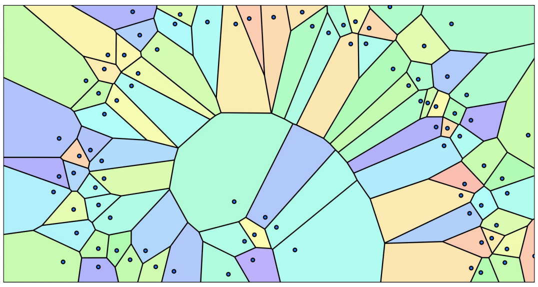


Figure 2: Example of a Voronoi diagram, which partitions a plane based on proximity to a set of seed points. The edges form the basis for our novel track representation.

Both representations utilize mutation and crossover operators. Mutation introduces small, random displacements to seed points to create minor variations. For the Voronoi-based genotypes, several custom crossover strategies were developed and evaluated, including a *Random-Line Partitioning Method* and a *Relative Reconstruction Method*, to effectively combine parent tracks.

2.2. Evaluation Pipeline and MAP-Elites Implementation

Our end-to-end pipeline (Figure 3) integrates several technologies. Python orchestrates the MAP-Elites algorithm and data analysis. A Node.js backend manages track generation and a custom HTTP API, complemented by a web-based front-end using p5.js, which provided crucial real-time visualization for developing and debugging the geometric operators. The entire system uses Docker to containerize TORCS simulations for parallel execution and consistency.

These containerized simulations collect raw telemetry data, which is processed to derive descriptive features characterizing both the track's static geometry and dynamic gameplay. Geometric features provide a structural baseline, including properties like total *track length* and the density of *left and right bends*. More sophisticated emergent features capture the driver's experience. We use Shannon entropy to quantify dynamic variability: high *speed entropy* signals a mix of fast and slow sections, while high *curvature entropy* reflects a geometrically diverse layout. Crucially, we engineered a normalized *total overtakes* score, which divides the raw overtake count by the track's geometric closure error. This robust metric effectively filters out simulation artifacts caused by imperfect track closure, providing a reliable measure of competitiveness. A Jupyter Notebook serves as the primary control interface.

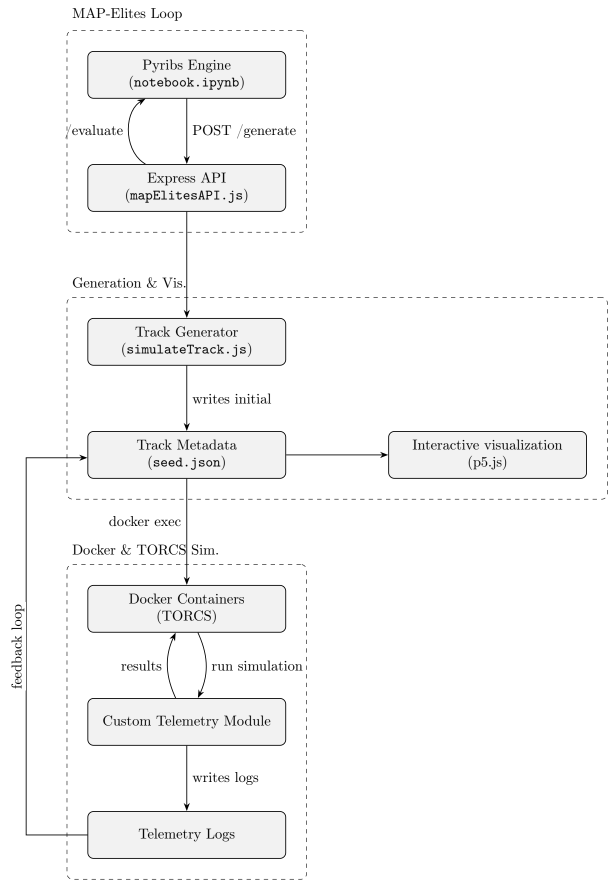


Figure 3: Overview of the end-to-end generation and evaluation pipeline, integrating Pyribs, Node.js, and Dockerized TORCS simulations.

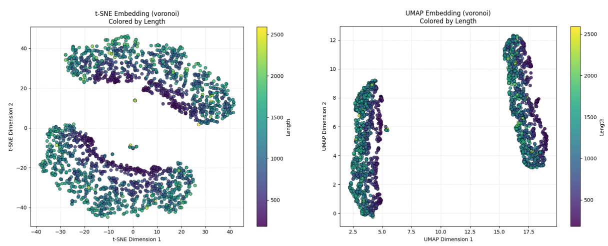
The QD search is powered by the Pyribs library [4]. We used a `SlidingBoundariesArchive` to dynamically adjust to the evolving range of track characteristics and a `CustomEmitter` to bridge Pyribs with our Node.js-based geometric operations.

3. Results and Analysis

The experimental phase was structured as a logical progression, beginning with foundational analysis to establish robust evaluation methods, followed by a series of MAP-Elites experiments to compare the generative techniques.

3.1. Foundational Analysis and Metric Validation

Initial analysis revealed that standard gameplay metrics are unreliable for evaluation. We discovered that raw overtaking counts were highly sensitive to simulation artifacts caused by minor geometric errors in track closure. To solve this, we engineered a robust, normalized overtaking score. We then used dimensionality reduction (UMAP) on raw track splines to generate a data-driven behavioral space (Figure 4), ensuring our main experiments were guided by stable, meaningful, and low-dimensional descriptors.



(a) t-SNE Embedding (b) UMAP Embedding

Figure 4: Dimensionality reduction embeddings of preprocessed track splines, which served as data-driven behavioral descriptors for the main experiments.

3.2. MAP-Elites Experiments and Findings

Our experiments proceeded in two stages. First, we conducted initial runs with a simplified fitness function focused solely on the normalized overtaking score. These tests, performed on both Convex Hull and Voronoi representations, consistently showed that tracks with low dynamic variability (measured by *speed_ entropy*) could not achieve high overtaking scores. This validated our hypothesis that a track's dynamic complexity is fundamental to its competitive potential and confirmed the utility of our chosen descriptors.

With this foundation, the main experiments used a more comprehensive fitness function rewarding length, curvature, and normalized overtakes to directly compare the generative methods. These experiments delivered a clear verdict: the Voronoi representation, when paired with the custom **Relative Reconstruction** crossover operator, demonstrated superior expressive power. It populated a significantly larger and more diverse archive compared to both the more constrained set of tracks from the Convex Hull method and the alternative Voronoi crossover operator (Figure 5).

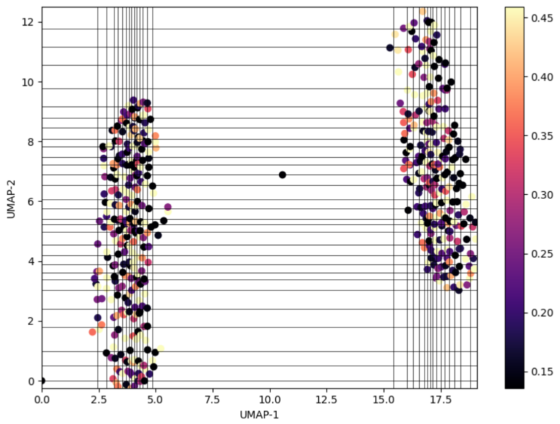


Figure 5: Final MAP-Elites archive for Voronoi tracks, showing wide coverage of the UMAP-derived behavioral space and high-fitness solutions (yellow).

The final archive contains a diverse collection of high-quality tracks, demonstrating the method's ability to explore the design space effectively (Figure 6).

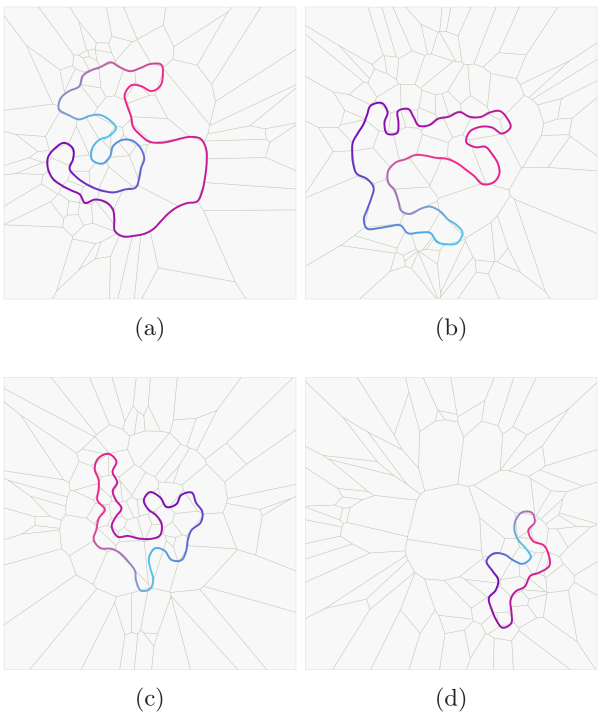


Figure 6: Another sample of diverse, high-fitness tracks generated using the Voronoi method, taken from the final MAP-Elites archive.

However, qualitative analysis of this high-performing archive also uncovered a critical limitation. A track with a very high fitness score was found to be topologically invalid, with a self-intersection that the TORCS engine did not detect (Figure 7). This discovery is significant because it shows that high performance in a simulation does not guarantee geometric validity and underscores the necessity of integrating robust geometric checks into the generation pipeline.

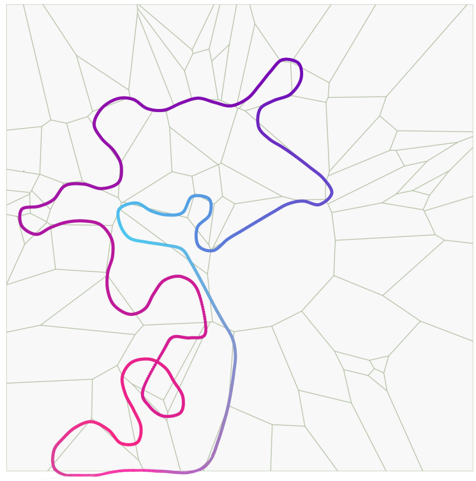


Figure 7: A high-fitness track (ID: 937.93) that is topologically invalid due to self-intersection, revealing a key limitation in simulation-based evaluation.

4. Conclusions and Future Developments

This thesis successfully demonstrated the power of MAP-Elites for the procedural generation of diverse, high-quality racing tracks. We developed a robust end-to-end pipeline, leveraging novel Voronoi and Convex Hull genotype representations, and a data-driven approach to define robust behavioral descriptors and fitness functions. Our work pioneered the use of UMAP for track characterization and provided critical insights into the challenges of simulation-based evaluation, including metric noisiness and the need for geometric validation.

While the current framework provides a strong foundation, several limitations and future directions exist:

- **Overcoming Framework Limitations:** The TORCS XML format restricts tracks to 2D. Future work should migrate the pipeline to a modern game engine to enable true 3D track generation with robust geometry handling.

- **Human-in-the-Loop Evaluation:** Current evaluation relies solely on AI agents. Integrating human feedback through an interactive evolution setup would allow players to directly guide the search towards more enjoyable designs.
- **Alleviating the Simulation Bottleneck:** The extensive simulation time is a major bottleneck. Developing surrogate models (e.g., neural networks) to predict fitness scores could significantly accelerate evaluation.
- **Exploring Advanced QD Algorithms:** Investigating cutting-edge QD algorithms like DCG-MAP-Elites could offer significant efficiency gains, especially if the track generation process can be made differentiable.

In conclusion, this research provides a solid foundation for enhancing the creative process of racing track design through Quality Diversity.

5. Source Code and Live Demo

The complete source code for the procedural generation pipeline developed in this thesis is publicly available on GitHub. The web-based visualization front-end, used for interactive debugging of the geometric operators, is also deployed as a live application.

GitHub Repository:
github.com/martinopiaggi/Quality-Diversity-for-Racing-Track-Design

Web-based visualization tool:
pcgtrack.netlify.app

References

[1] Gustavo Maciel. Generating procedural racetracks. Gamasutra Blog, December 2013. URL `https://www.gamedeveloper.com/programming/generating-procedural-racetracks`.

[2] Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, April 2015. doi: 10.48550/arXiv.1504.04909.

[3] Speed Dreams Development Team. Speed dreams, 2010. URL `http://speed-dreams.sourceforge.net/`.

[4] Bryon Tjanaka, Matthew C Fontaine, David H Lee, Yulun Zhang, Nivedit Reddy Balam, Nathaniel Dennler, Sujay S Garlanka, Nikitas Dimitri Klapsis, and Stefanos Nikolaidis. Pyribs: A bare-bones python library for quality diversity optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '23, page 220–229, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701191. doi: 10.1145/3583131.3590374. URL `https://doi.org/10.1145/3583131.3590374`.

[5] Bernhard Wymann, Christos Dimitrakakis, Andrew Sumner, Eric Espie, and Christophe Guionneau. TORCS: The open racing car simulator, 2001. URL `http://torcs.sourceforge.net/`.