# Racing Game Algorithms

## Artificial Intelligence in Games

Not only the level design but also the enemies should challenge the player in games. Section~\ref{sec:rel:enjoymentingames} determines the key factors to model opponents in games for optimal \textit{Enjoyment}. The key factors are: balanced, diverse and aggressive behaviour. This section focuses on modelling diverse behaviour by creating human-like opponents.

AI in games refers often time to intelligent agents which act autonomously towards a goal. Artificial Intelligence research found multiple ways to model intelligent agents. They are used to cover complex problems in computer science e.g. autonomous cars, speech recognition. The agents in videos games often times need to satisfy specific requirements \citep{Nareyek}:

* Real Time: Processing needs to be fast.
* Dynamics: Computer games have a dynamic environment.
* Resources: System resources are sometimes restricted.

Many computer game developers circumvent the problem of applying sophisticated AI techniques by allowing agents to cheat \citep{Nareyek}. The authenticity of cheating agents is very hard to ensure and creates often times static behaviour which makes it not suited for *Education* where players should learn from their opponents \citep{Lent2007}. The design of artificial intelligence in computer games is an important component of the players *Enjoyment*. As computational hardware improves and games are becoming more life-like the need for more realistic game AI increases. In the next segment we discuss universal AI models for games and specific approaches for race games.

An often used method of creating human-like opponents in games is evolutionary learning. Evolutionary learning approaches can be applied to all kinds of games. A lot of research on learning in games has been done on board and card games. \textcite{Fogel1993} created a simple AI able to play tic-tac-toe. \textcite{Richards1997} showed that Neuronal Networks can be used to model an opponent for GO. It`s one of the most complex board games and very difficult to master, even for computers. With the increased computational power in recent years, the generated opponents are capable of beating even expert humans in a multitude of games. Today’s best Computer GO program AlphaGo \footcite{AlphaGo2018} uses a Monte Carlo algorithm based on learned knowledge. It was the first algorithm to consistently beating the world No.1 ranked player at the time. Modern computer game AI research focuses mainly on real-time strategy (RTS) and first-person shooter (FPS) games due to their popularity. \textcite{Khoo2002} developed a simple and computationally inexpensive AI mechanism to produce engaging character behaviour. The system uses behaviour based action selection techniques taken from robotics. It showed mixed results, some of the testers could not defer the AI from humans, others could not be deceived. \textcite{Cole2004} used a generic algorithm to balance parameters for bots. \textcite{Ponsen2004} is using RTS games to propose adaptive game AI with dynamical scripting. Their approach significantly improves the performance of adaptive game AI. \textcite{Thurau2004} learned strategies by observing human players. The investigated movement patterns resulted in a wide range of situation-dependent human-like strategic movements. Their research presents a first step towards the development of more human-like computer game bots.

In conventional games, non-player moving objects are controlled by predetermined algorithms. The problem in racing games is that automatically controlled cars tend to bunch together. Different performing race car algorithms can improve the problem considerably but produces monotonous race results. For this reason, Nintendo introduced the rubber banding algorithm mainly for arcade games \citep{Nintendo2004}. The artificial intelligence is designed to prevent computer-controlled opponents to get too far ahead or fall back. When done well, rubber banding can provide a consistent level of challenge. But in many cases, it becomes evident that the player can’t escape regardless of skill and effort. This completely ruins the experience for the player. This approach is similar to the cheating agents not suitable for Education. More complex algorithms are used for autonomous vehicles. Autonomous vehicles are developed to construct driverless transport systems, essentially revolutionising the way we live. The vision is to make driving safer and more efficient. A lot of car manufacturers and start-ups are working to make the vision reality. Self-driving software is simulated on powerful computers for testing and validation purposes. Photorealistic simulation runs on GPUs simulate cameras and sensors. It allows to processes the data as if it were actually driving on the road \citep{Nvidea2019}. This method would also be suited to generate a variety of diverse autonomous vehicle scenarios for racing games but it requires powerful hardware.

This section introduced some AI mechanics, generating diverse behaviour for agents. Despite all these complex algorithms, there is little research done on how these behaviours contribute to the player experience (\cite{Yannakakis2007}). There is no evidence that by generating human-like opponents we can create more satisfaction. It exists research in general board games. For chess \textcite{Iida2003} defined a metric of entertainment. The metric is based on average game length and the number of possible moves per turn. The discussed algorithms focus on diverse and aggressive behaviour. To improve Enjoyment we need to also focus on balanced AI based on the player skill level. The next section examines player’s skill level. Balancing AI based on players skill is key to improve \textit{Enjoyment} and \textit{Motivation} (see Section~\ref{ sec:rel:gamedesignprinciples}).

Intelligent Agents for Computer Games – Nareyek

Entertainment game AI vs. Serious game AI - Lent2007

Modeling Player-like Behavior for Game AI Design – Conroy