

**Data Preparation: Input from Screen Data  
AI Agent for Car Racing Game**

Version: 1.0  
Author: José Henrique Nóbrega Pereira  
Fontys ICT  
Date: June 3, 2025

|  |
| --- |
| A black background with purple text  AI-generated content may be incorrect. |

**Introduction**

To be able to train the AI to play the videogame, it needs to be given some inputs about the game to make decisions based on meaningful information. If the AI is put into the game without any inputs, it would basically be “blind” and only be able to random and uneducated guesses on its next moves.

This document will outline the input data processing used to train the AI and how the agent perceives the environment through a combination of visual inputs and state indicators extracted from the game frames.

**Game State Input**

The main input received by the AI are screenshots of the game state. These frames represent the visual state of the car and the track at each timestep. To capture motion and pass it to the AI, a stack of four consecutive frames is used as a single input, transmitting the idea of movement. Each frame is first converted to grayscale and then resized to 84x84 pixels (from 96x96 pixels previously) to reduce computational overhead. The stacking of frames allows the agent to have a short-term movement feel, such as the movement direction and changes in velocity.

In addition to the game frames, each screenshot includes 5 fixed-position visual indicators that provide extra information about the state of the car.

The first indicator is the reward value which works as a score for the current episode for the AI. The rewards are constantly updating and reflects the feedback from the environment. This is essential because it guides the agent’s behaviours updates by indicating how good or bad certain actions are in context of the goal of going around the track.

The second indicator is the true speed. This displays the current speed of the vehicle with a bar that raises vertically with the increase of speed. This allows the agent to always know how fast it is going.

The third indicator is the 4 ABS sensors. Each is represented with a vertical bar as well and lets the agent know the lock-up condition of each wheel. This informs the agent about wheel traction and braking performance.

The fourth indicator is the steering wheel position which indicates the current angle that the steering wheel is. This provides information about the agents steering inputs and how sharply the car is currently turning with a horizontal green bar that goes right and left of a centre point.

The fifth and final indicator is a gyroscope represented by a similar bar to the steering wheel but in red. This shows angular velocity data, which informs the AI how quicky the car is rotating, allowing the AI to react in case of drifting and oversteering.

The picture below shows these indicators:

A video game of a race car

AI-generated content may be incorrect.

Figure 1. Screenshot of the five indicators the AI can read from each game state.  
The number “0548” is the current reward score the AI has hit.  
The white vertical bar is the true speed of the car.  
The 4 blue vertical bars are the 4 ABS sensors for each wheel.  
The green horizontal bar is the steering wheel position.  
The red horizontal bar is the gyroscope.

These indicators and embedded in the game visual output and are not provided as separate numerical inputs to the AI. Instead, they appear in a fixed position on the screen which allows the AI to learn their position and what kind of information it represents by interpreting it as a image input. This information is not given separately to the AI, it must learn through training what it means and associate it with the driving.

These inputs are passed to two separate neural networks, an Actor and a Critic. These networks extract the information from the information they are given and choose their actions based on it. The combination of the raw game frames combined with the embedded indicators about the car state allow the AI to develop a deep understanding about the environment, the vehicle and how to drive it to extract the most performance possible.

**Conclusion**

To conclude, the input processing process transforms the visual game frames into structured and compact observations that capture both environmental information and vehicle information. This allows a visually driven agent to make informed decisions based on what they “see”, without having the need to access internal game variables.

**Personal Reflection**

On a personal level what stood out to me the most was realizing that AI does not need direct access to internal game data to understand the environment and the vehicle. It can learn everything directly from the videogame frame inputs. Before this project I assumed that the visual frames would only be used for knowing the car position relatively to the track and where it is facing and the remaining factors like true speed and the wheel position would need to be given as separate numerical inputs but with this process I learned that those indicators can be transformed into a visual form and fed to the AI that way. Designing the game this way felt like that AI was trying to learn a new language from scratch, and it did!

This approach of teaching an agent through vision alone opened my eyes to a new way of thinking about input processing and agent design. This is something I will be looking to implement in futures projects, either in video games or my future field of work. I also learned that the AI performance does not depend solely on how smart the algorithm is but also how clearly the environment is presented to the AI.