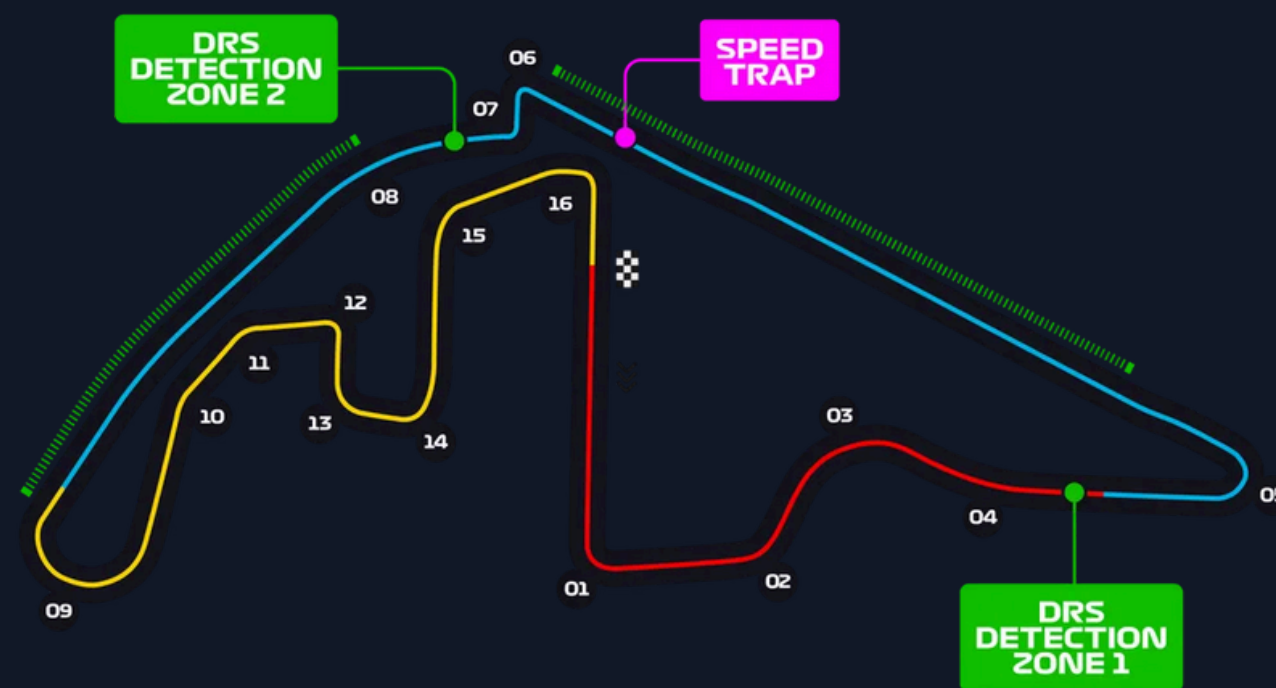


F1 AI 드라이버

유전 알고리즘과 전이 학습을 이용한 자율주행 AI 개발

아부다비 <야스 마리나 서킷>에서
가장 빠르게 달리는 AI를 만들어보자!



프로젝트 개요

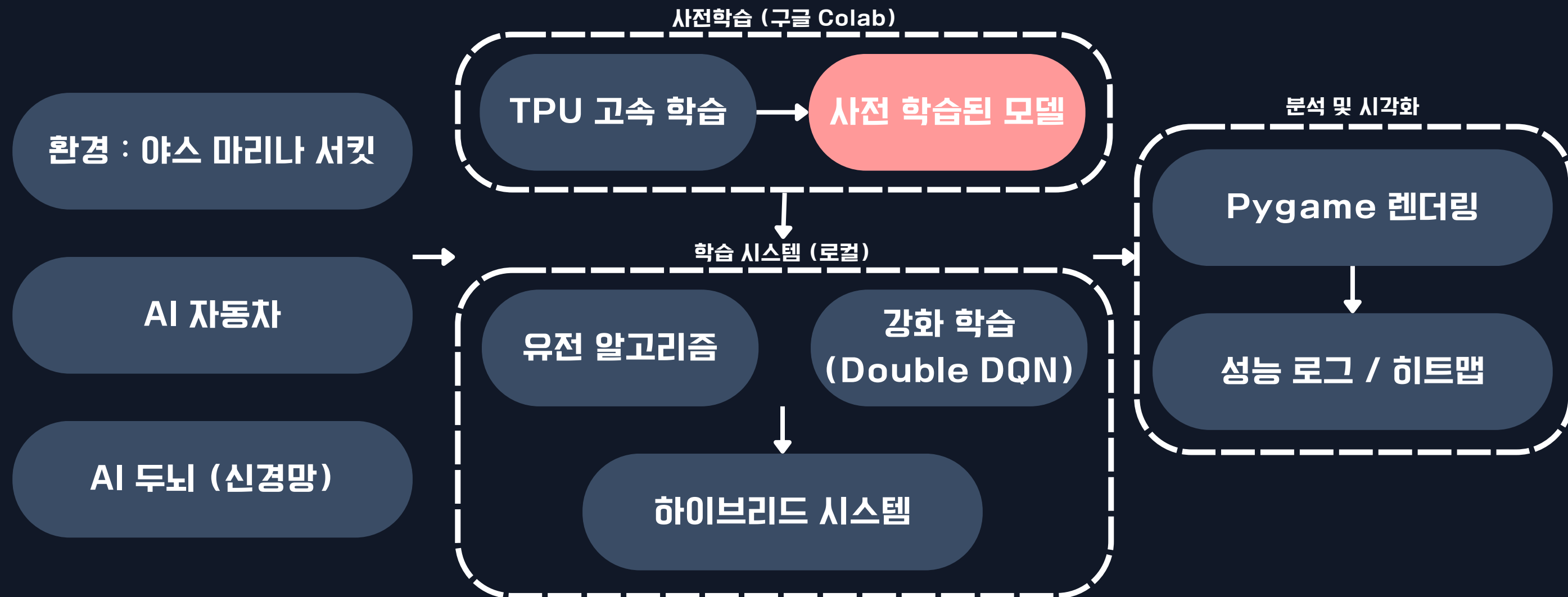
기술 스택

- Python
- Pygame
- NumPy
- PyTorch (RL)

AI 핵심 이론

- 신경망 (Neural Network)
- 유전 알고리즘 (Genetic Algorithm)
- 강화학습 (Reinforcement Learning)
- 전이 학습 (Transfer Learning)

시스템 아키텍처



핵심 기능 및 개발 과정

1단계 : 기본 시스템 구축

- Pygame으로 트랙과 자동차 구현
- Ray-Casting 센서로 벽 감지
- 기본적인 물리 엔진 적용

2단계 : 유전 알고리즘 적용

- 다세대 진화를 통한 성능 향상
- 적합도 함수 설계가 핵심
 - 주행 거리, 평균 속도
 - 레이스라인 추종 점수

두 알고리즘, 어떻게 같이 쓸까?

각 알고리즘 역할 분담

강화학습(Double DQN)

- 역할: 하나의 해법 깊게 파고들기
- 장점: 특정 상황에서 최적의 액션 찾기
- 단점: 더 좋은 해법이 있어도 찾지 못하고 한 곳에 갇힐 수 있음

유전 알고리즘(GA)

- 역할: 다양한 가능성 찾기
- 장점: 생각 못한 새로운 해법 찾기
- 단점: 가장 좋은 해법을 정교하게 다듬기에 비효율적임

DQN와 GA 장점 합치기!

사용한 주요 기법들

유전 알고리즘

- 적응형 변이율
- 다양성 보존
- 다양한 교차 연산 방식
- 병렬 처리로 자손 생성

강화학습

- Double DQN 아키텍처
- 배치 정규화
- F1 데이터 특징 추출

성능 최적화

- Lazy Rendering
- 렌더링 결과 캐싱
- 메모리 사용량 모니터링
- 모듈화 설정 클래스

학습 방식

⚠ 문제점: 로컬 컴퓨터 학습은 너무 느림

해결책 1: Colab 전이학습

- Colab의 TPU로 기본 모델 빠르게
사전 학습 후, 학습된 가중치를 로컬로
가져와 적용

해결책 2: 하이브리드 학습 시스템

- Colab으로 학습한 모델과 로컬 GA
의 환경 적응력을 결합하여 파인튜닝

최종 성능 30% 향상

시연

결론 및 향후 과제

성과 및 배운 점

- GA와 RL을 결합한 하이브리드 AI 시스템 구현
- 전이 학습 파이프라인 구현
- ai 모델링 외에도 전체 시스템 설계 경험

향후 과제

- 다른 트랙(멀티 트랙) 기능
- 다른 모델과 경쟁하는 기능
- RL 알고리즘 발전(ex: PPO, SAC)